A personal journey towards more reproducible networking research

Olivier Bonaventure

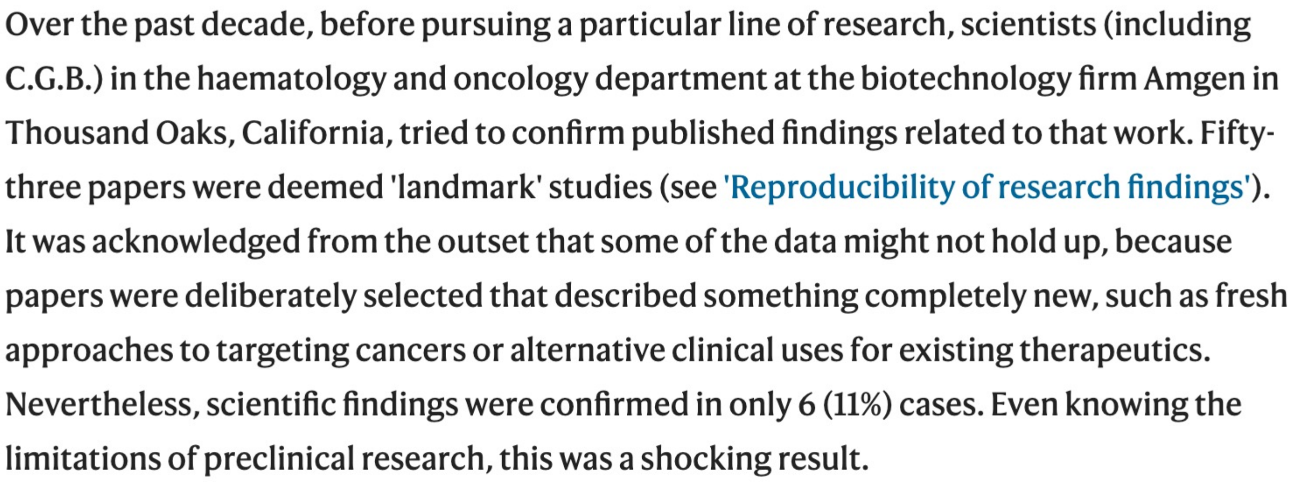
UCLouvain

https://inl.info.ucl.ac.be

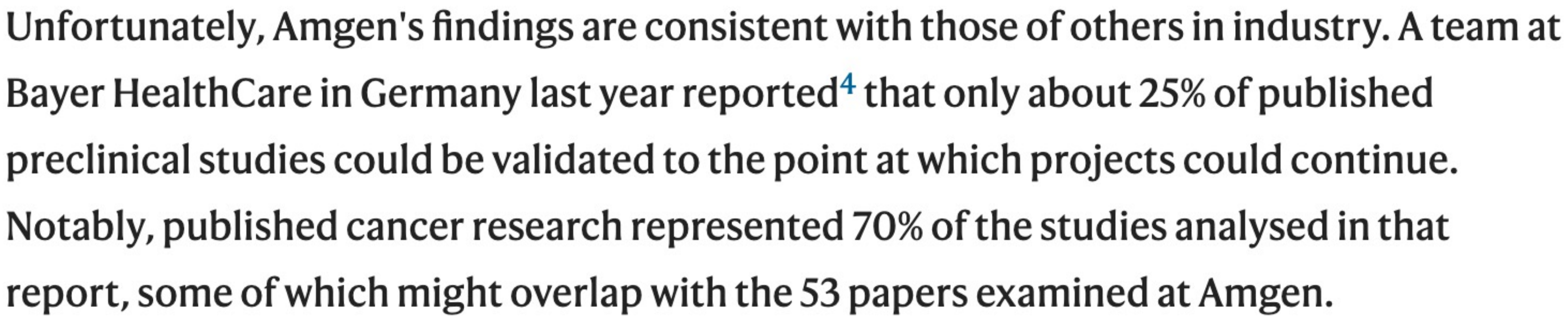
Agenda

- **The reproducibility crisis**
- Dagsthul’s guide to Reproducibility for Experimental Networking Research
- Artefacts Evaluations

The reproducibility crisis in medicine



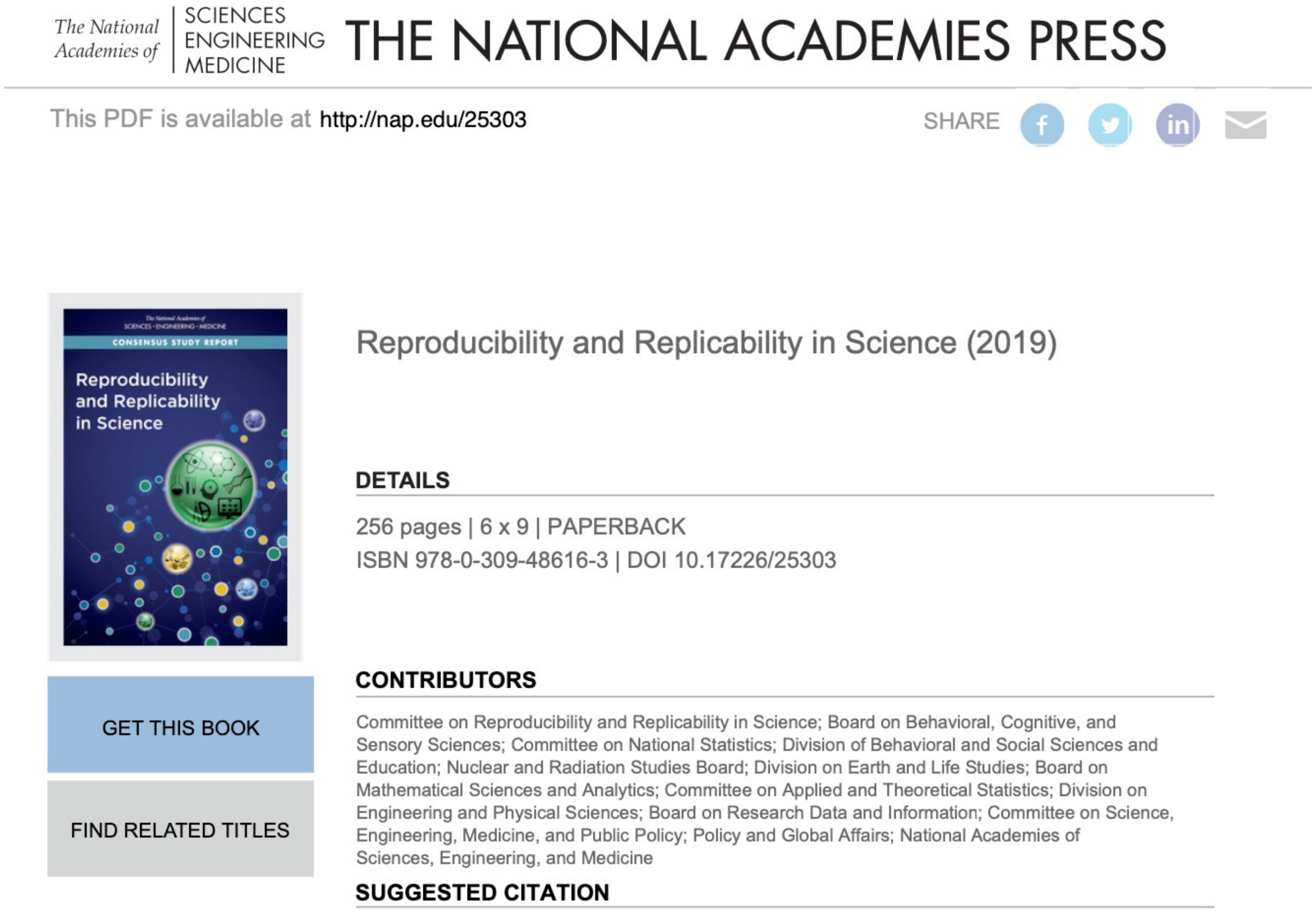
Over the past decade, before pursuing a particular line of research, scientists (including C.G.B.) in the haematology and oncology department at the biotechnology firm Amgen in Thousand Oaks, California, tried to confirm published findings related to that work. Fifty-three papers were deemed 'landmark' studies (see 'Reproducibility of research findings'). It was acknowledged from the outset that some of the data might not hold up, because papers were deliberately selected that described something completely new, such as fresh approaches to targeting cancers or alternative clinical uses for existing therapeutics. Nevertheless, scientific findings were confirmed in only 6 (11%) cases. Even knowing the limitations of preclinical research, this was a shocking result.



Unfortunately, Amgen's findings are consistent with those of others in industry. A team at Bayer HealthCare in Germany last year reported[4] that only about 25% of published preclinical studies could be validated to the point at which projects could continue. Notably, published cancer research represented 70% of the studies analysed in that report, some of which might overlap with the 53 papers examined at Amgen.

Source: Begley, C., Ellis, L. Raise standards for preclinical cancer research. *Nature* **483,** 531–533 (2012). https://doi.org/10.1038/483531a

The reproducibility crisis



The National Academies of SCIENCES ENGINEERING MEDICINE THE NATIONAL ACADEMIES PRESS

This PDF is available at http://nap.edu/25303

SHARE
The National Academies of SCIENCES · ENGINEERING · MEDICINE
CONSENSUS STUDY REPORT
Reproducibility and Replicability in Science
GET THIS BOOK
FIND RELATED TITLES

Reproducibility and Replicability in Science (2019)

DETAILS

256 pages | 6 x 9 | PAPERBACK
ISBN 978-0-309-48616-3 | DOI 10.17226/25303

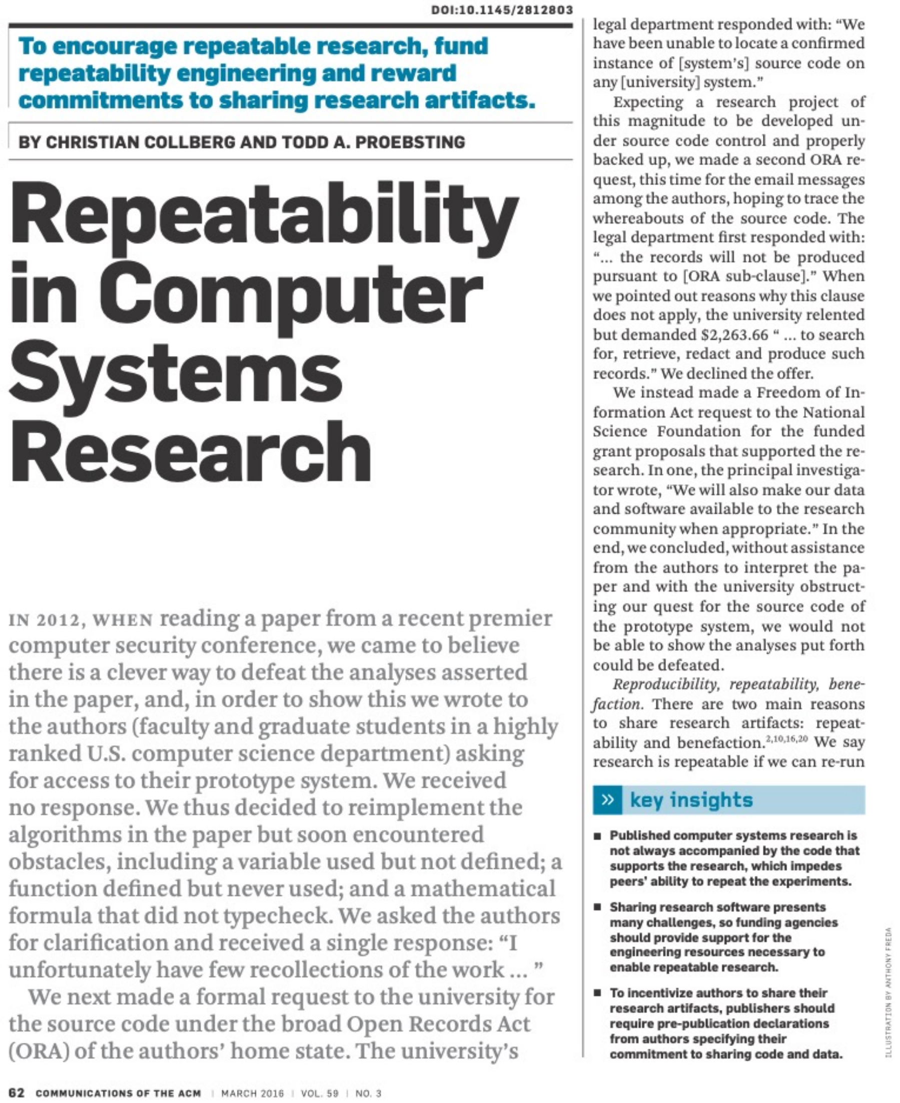
CONTRIBUTORS

Committee on Reproducibility and Replicability in Science; Board on Behavioral, Cognitive, and Sensory Sciences; Committee on National Statistics; Division of Behavioral and Social Sciences and Education; Nuclear and Radiation Studies Board; Division on Earth and Life Studies; Board on Mathematical Sciences and Analytics; Committee on Applied and Theoretical Statistics; Division on Engineering and Physical Sciences; Board on Research Data and Information; Committee on Science, Engineering, Medicine, and Public Policy; Policy and Global Affairs; National Academies of Sciences, Engineering, and Medicine

SUGGESTED CITATION

Reproducibility in Computer Science

- An interesting study



DOI:10.1145/2812803

To encourage repeatable research, fund repeatability engineering and reward commitments to sharing research artifacts.

BY CHRISTIAN COLLBERG AND TODD A. PROEBSTING

Repeatability in Computer Systems Research

IN 2012, WHEN reading a paper from a recent premier computer security conference, we came to believe there is a clever way to defeat the analyses asserted in the paper, and, in order to show this we wrote to the authors (faculty and graduate students in a highly ranked U.S. computer science department) asking for access to their prototype system. We received no response. We thus decided to reimplement the algorithms in the paper but soon encountered obstacles, including a variable used but not defined; a function defined but never used; and a mathematical formula that did not typecheck. We asked the authors for clarification and received a single response: "I unfortunately have few recollections of the work … "

We next made a formal request to the university for the source code under the broad Open Records Act (ORA) of the authors' home state. The university's legal department responded with: "We have been unable to locate a confirmed instance of [system's] source code on any [university] system."

Expecting a research project of this magnitude to be developed under source code control and properly backed up, we made a second ORA request, this time for the email messages among the authors, hoping to trace the whereabouts of the source code. The legal department first responded with: "… the records will not be produced pursuant to [ORA sub-clause]." When we pointed out reasons why this clause does not apply, the university relented but demanded $2,263.66 " … to search for, retrieve, redact and produce such records." We declined the offer.

We instead made a Freedom of Information Act request to the National Science Foundation for the funded grant proposals that supported the research. In one, the principal investigator wrote, "We will also make our data and software available to the research community when appropriate." In the end, we concluded, without assistance from the authors to interpret the paper and with the university obstructing our quest for the source code of the prototype system, we would not be able to show the analyses put forth could be defeated.

Reproducibility, repeatability, benefaction. There are two main reasons to share research artifacts: repeatability and benefaction.[2,10,16,20] We say research is repeatable if we can re-run

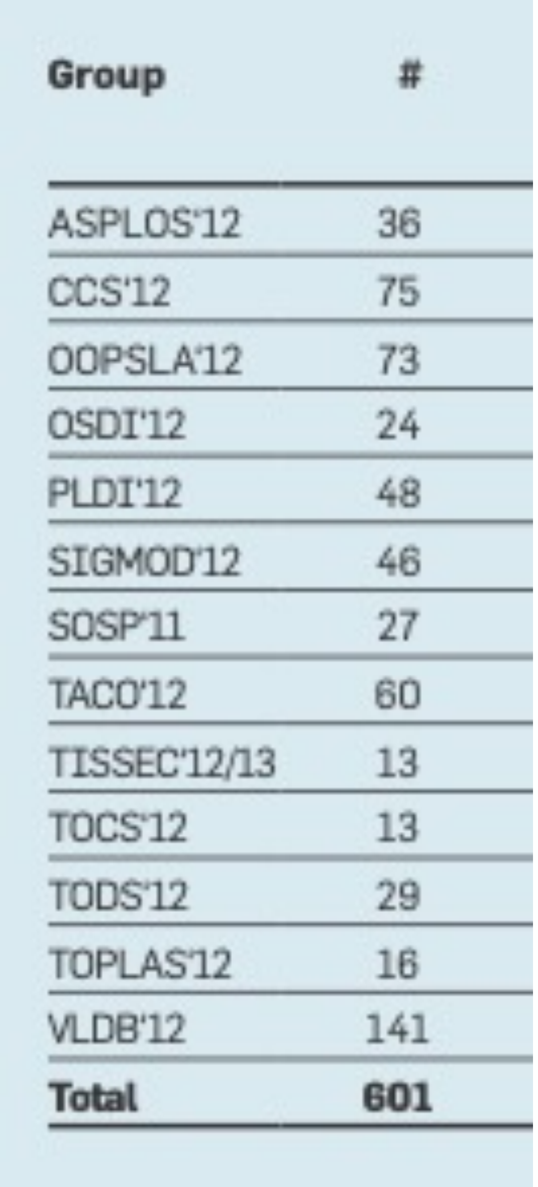
» **key insights**

- **Published computer systems research is not always accompanied by the code that supports the research, which impedes peers' ability to repeat the experiments.**
- **Sharing research software presents many challenges, so funding agencies should provide support for the engineering resources necessary to enable repeatable research.**
- **To incentivize authors to share their research artifacts, publishers should require pre-publication declarations from authors specifying their commitment to sharing code and data.**

ILLUSTRATION BY ANTHONY FREDA

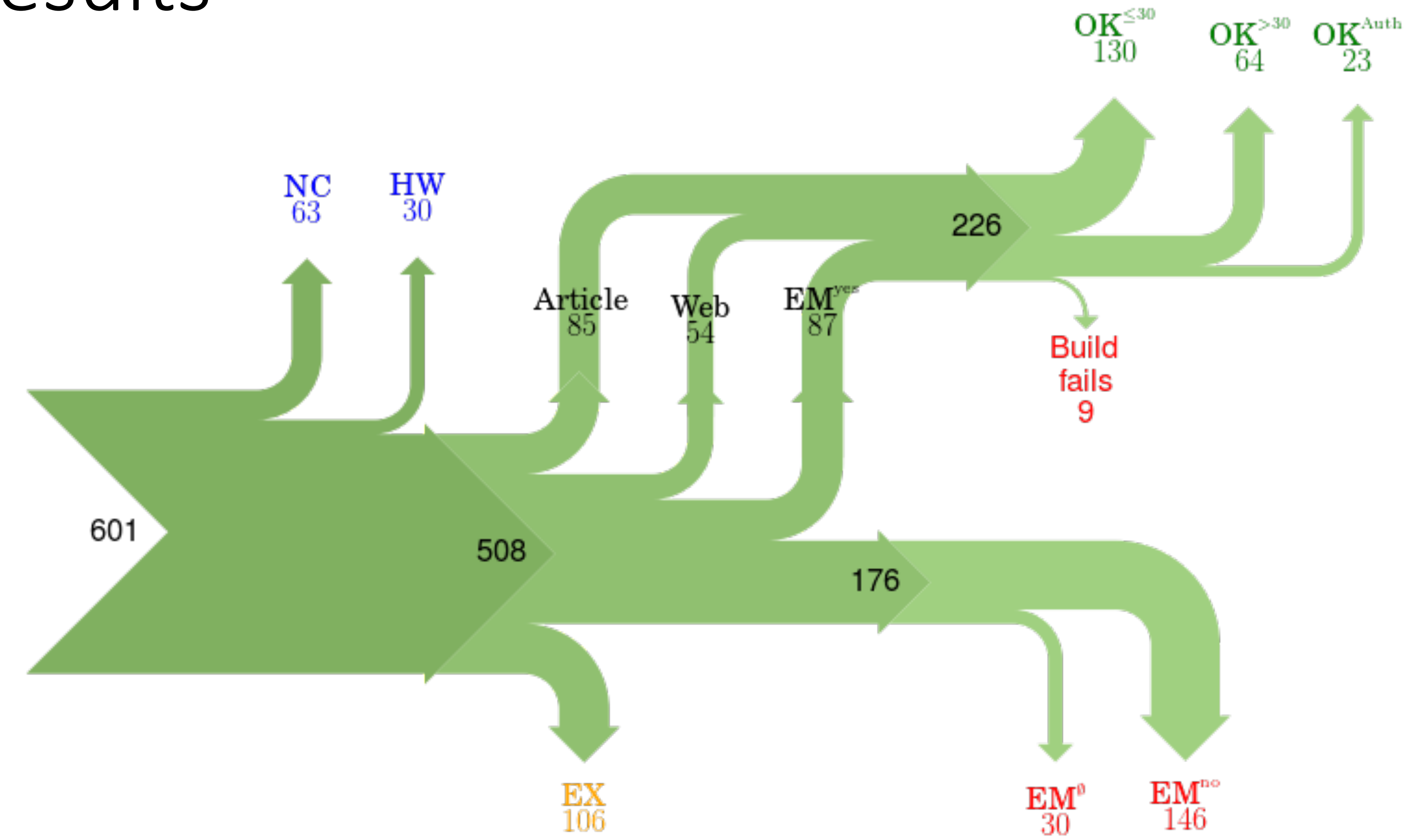
Software artfeacts

- Methodology
  - Select papers from top conferences and journals
  - Check if the they use software that does not require special hardware
  - Try to obtain software
    - Directly with URL in paper
    - By contacting corresponding authors by email
      - avoid contacting the same author several times
  - Try to recompile software
    - With modifications that do no require more than 30 minutes



Group	#
ASPLOS'12	36
CCS'12	75
OOPSLA'12	73
OSDI'12	24
PLDI'12	48
SIGMOD'12	46
SOSP'11	27
TACO'12	60
TISSEC'12/13	13
TOCS'12	13
TODS'12	29
TOPLAS'12	16
VLDB'12	141
Total	**601**

Survey results



Proposal : sharing contract in papers

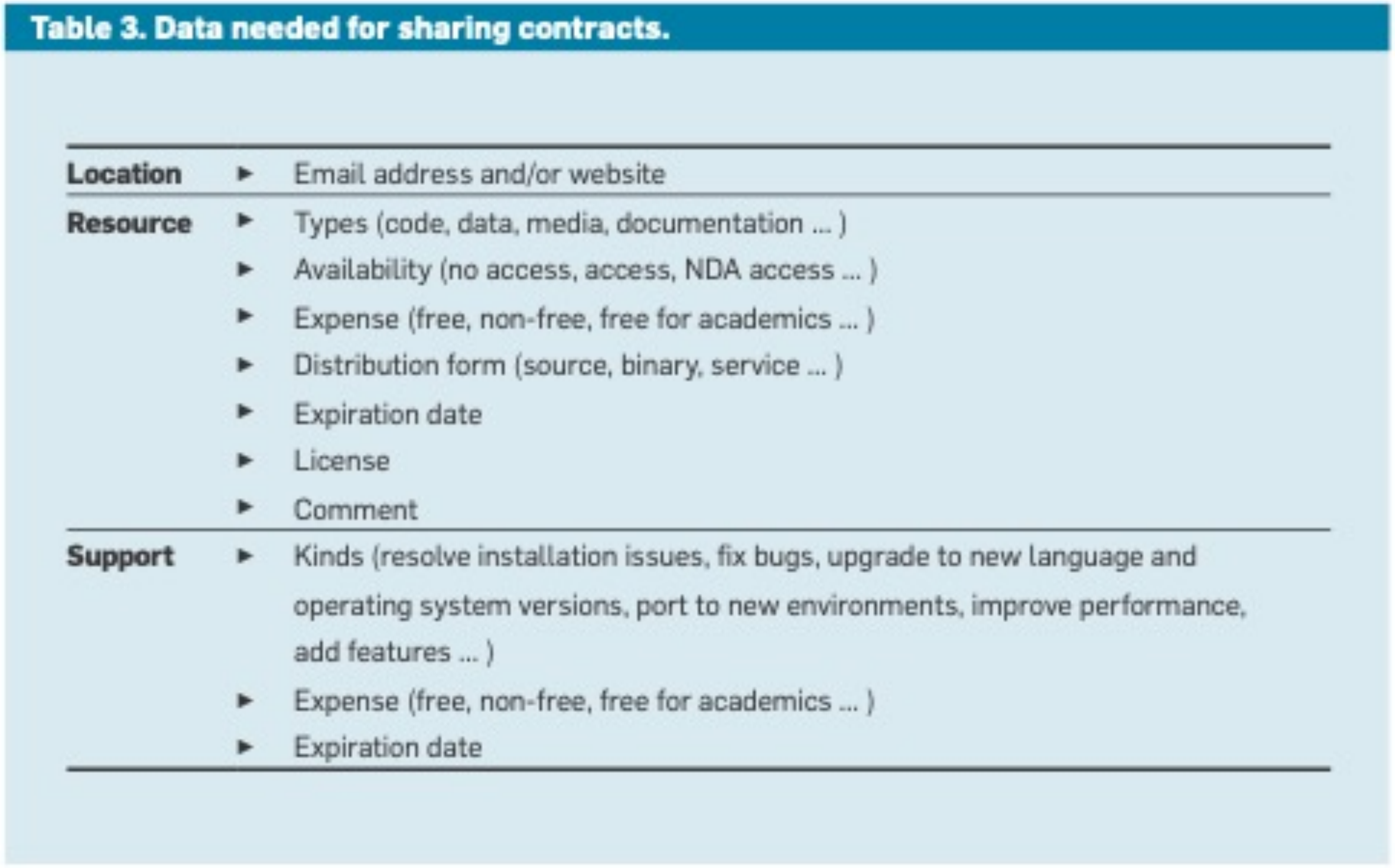


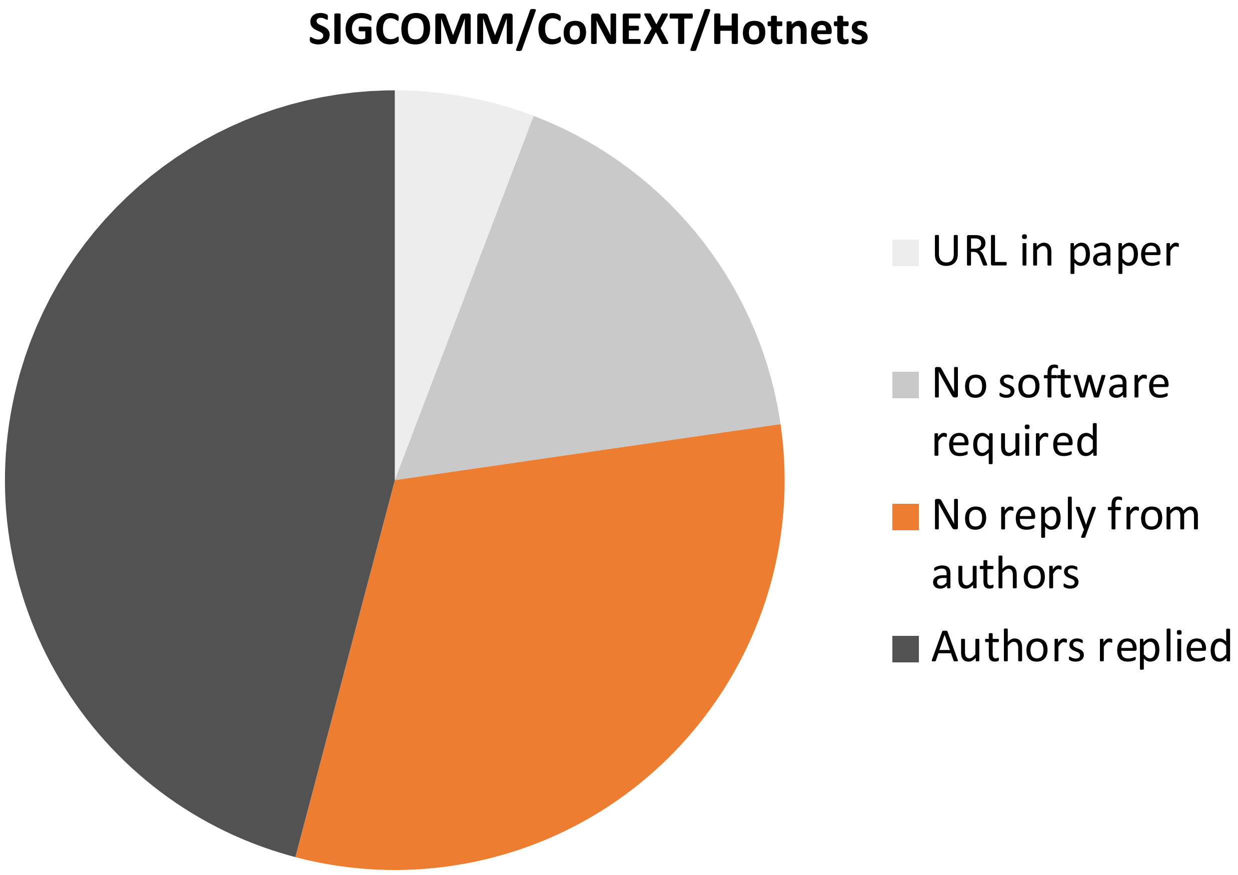
Table 3. Data needed for sharing contracts.

Location	►	Email address and/or website
Resource	►	Types (code, data, media, documentation ...)
	►	Availability (no access, access, NDA access ...)
	►	Expense (free, non-free, free for academics ...)
	►	Distribution form (source, binary, service ...)
	►	Expiration date
	►	License
	►	Comment
Support	►	Kinds (resolve installation issues, fix bugs, upgrade to new language and operating system versions, port to new environments, improve performance, add features ...)
	►	Expense (free, non-free, free for academics ...)
	►	Expiration date

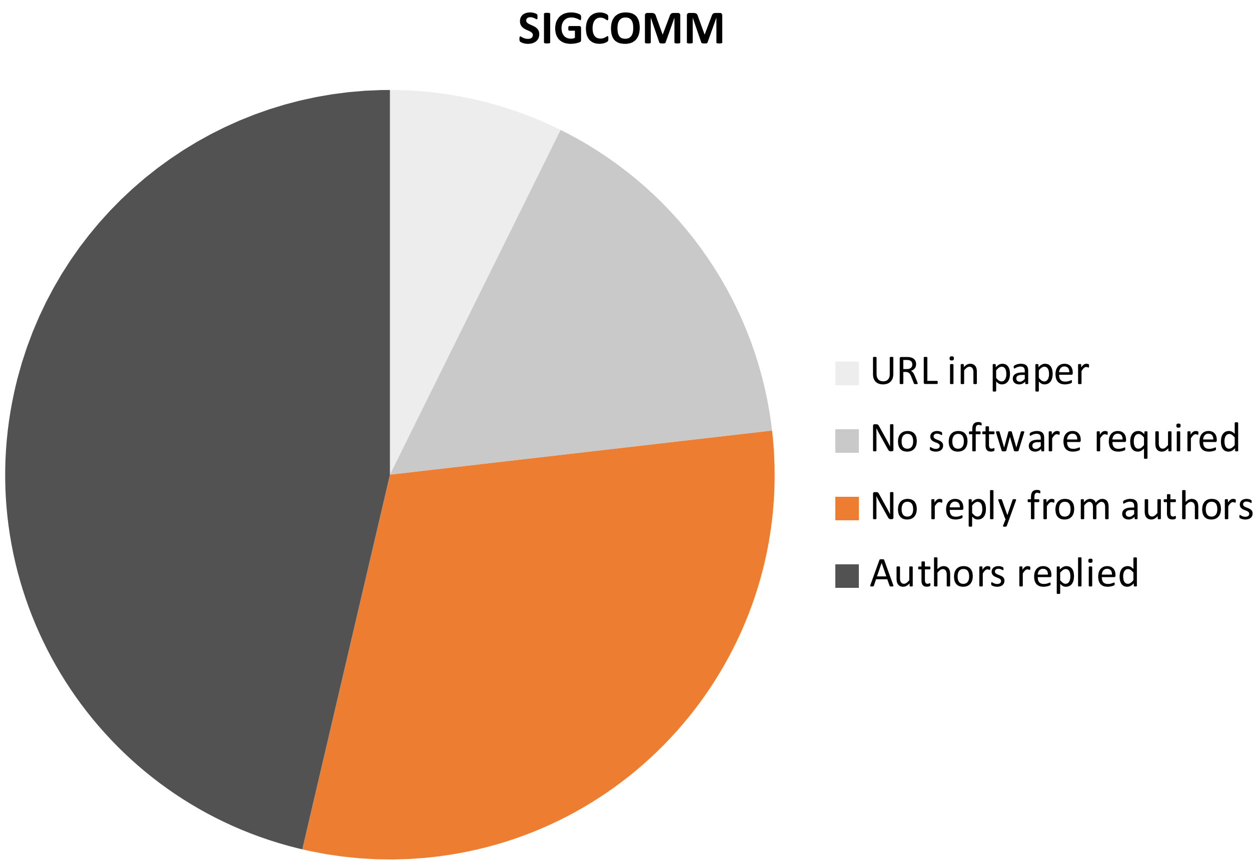
Reproducibility in networking research

- A short study conducted with Nicolas Houtain on software artefacts presented during the **SIGCOMM'15** Community session
- Methodology and dataset
    - Conference programmes
        - SIGCOMM, 2013 and 2014
        - CoNext, 2013 and 2014
        - Hotnets 2013 and 2014
    - Read each paper to identify
        - Whether software was used to produce results
        - If yes, is this software publicly available ?
            - Directly mentioned in the paper
            - Available after contacting the authors by email

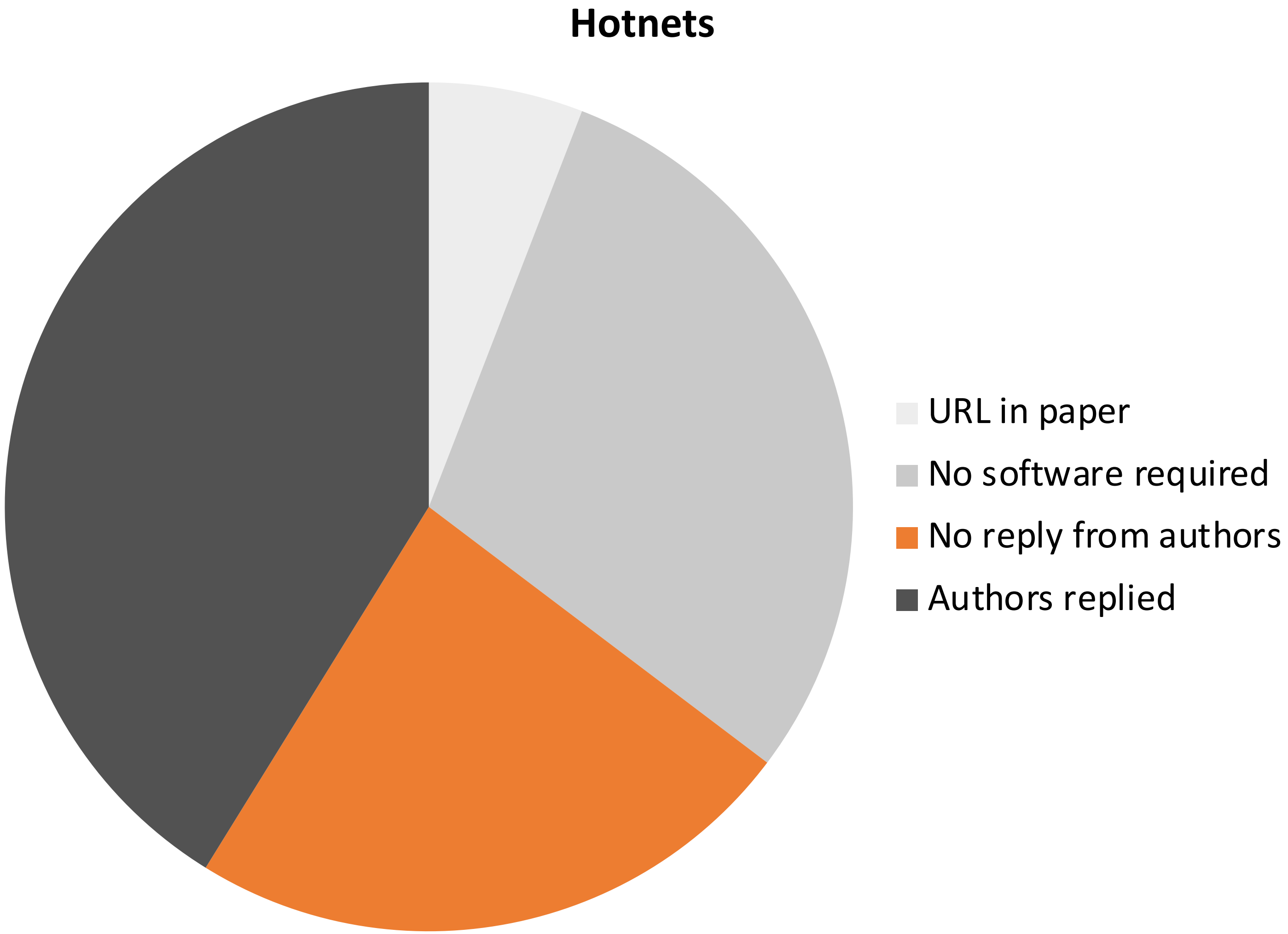
How accessible is software written for scientific papers ?



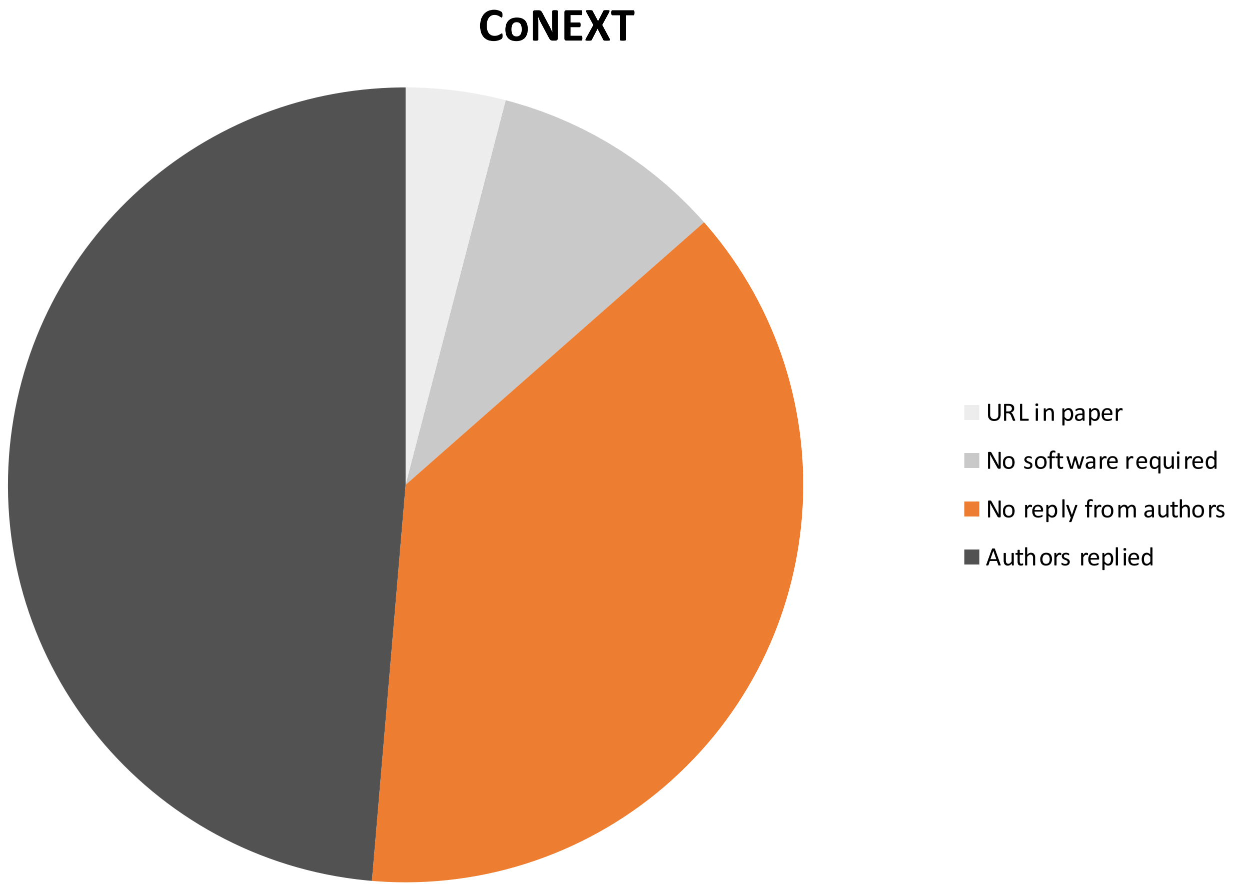
Software from SIGCOMM papers



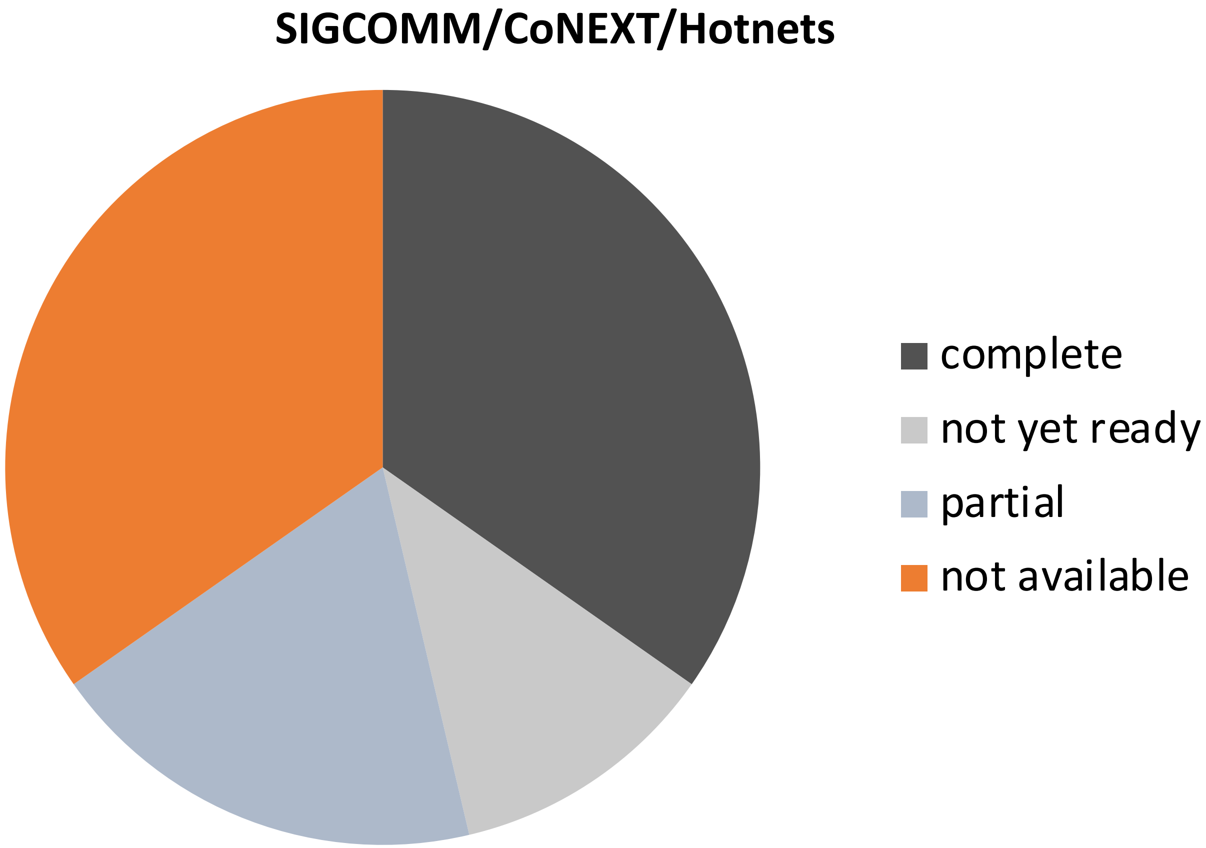
Software from Hotnets papers



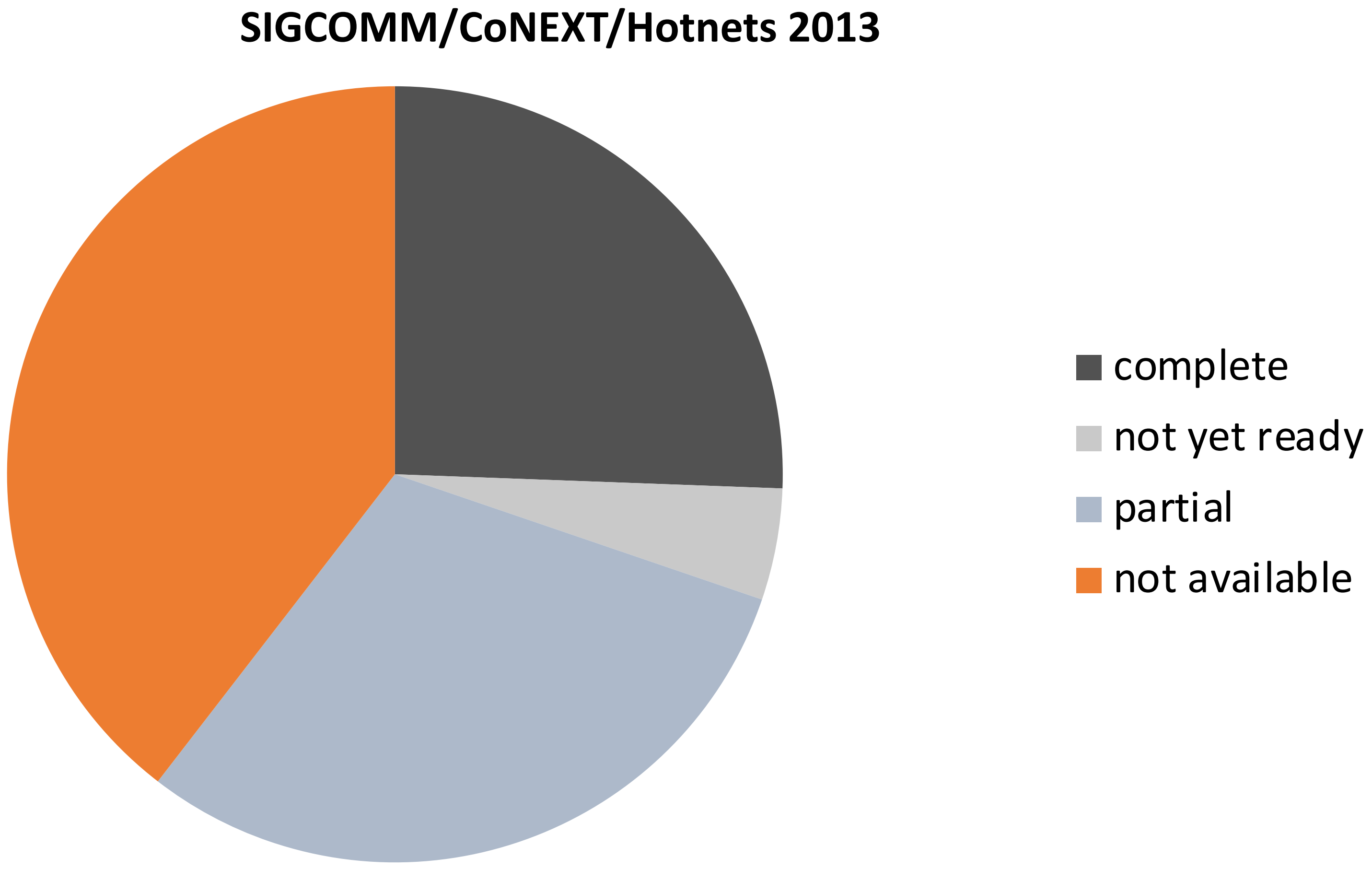
Software from Conext papers



Authors' responses

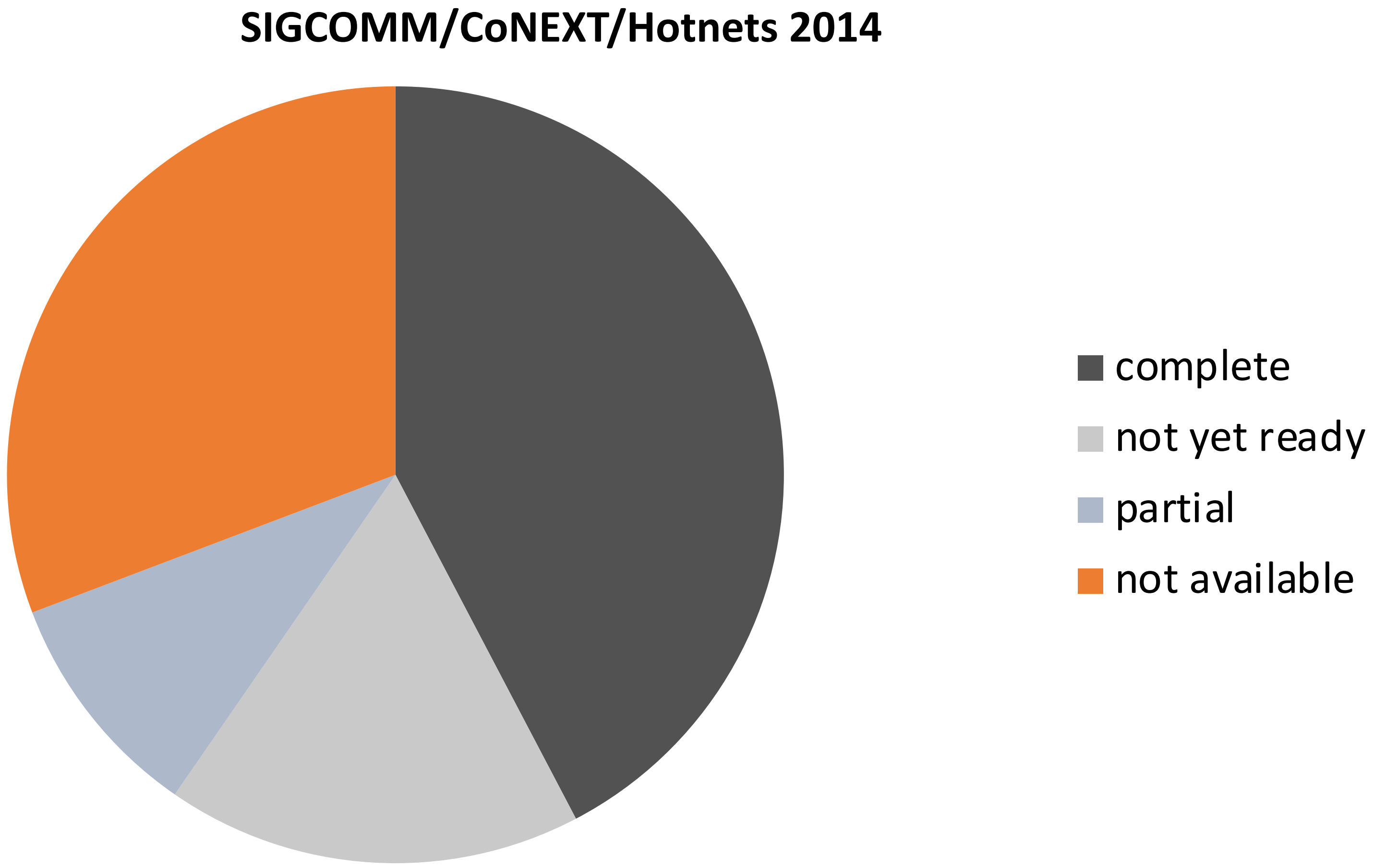


2013 conferences



SIGCOMM/CoNEXT/Hotnets 2013

2014 conferences



Recommendations at SIGCOMM 2015

- Encourage authors to release the software used for accepted papers
    - Source code preferably (open-source, closed source with "research" license), if not binaries or VMs
    - Datasets used for experiments/analysis
        - There are sometimes privacy issues, but not always and some part of the paper should be evaluated with publicly accessible datasets
- Encourage authors to post additional data in the ACM digital library next to each paper

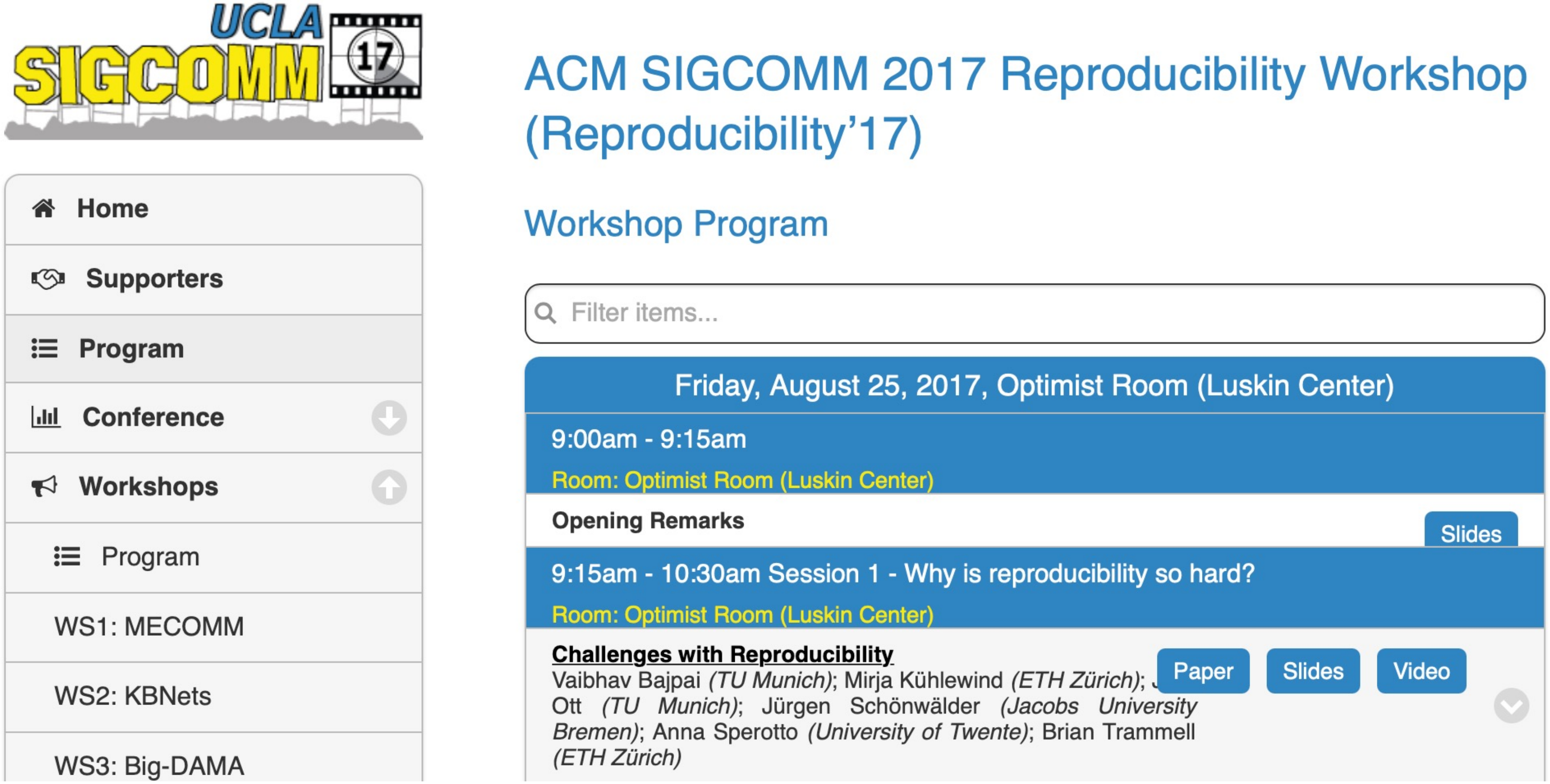
Recommendations at SIGCOMM 2015

- Encourage TPC members to consider the reproducibility of the papers while reviewing
  - Could become a standard question of review forms and maybe used for decisions

- Encourage fully reproducible papers through awards or some form of recognition

- Encourage steering committees to think about ways to improve the reproducibility of our papers

SIGCOMM 2017 Reproducibility workshop



ACM SIGCOMM 2017 Reproducibility Workshop (Reproducibility'17)

Workshop Program

Friday, August 25, 2017, Optimist Room (Luskin Center)

9:00am - 9:15am

Room: Optimist Room (Luskin Center)

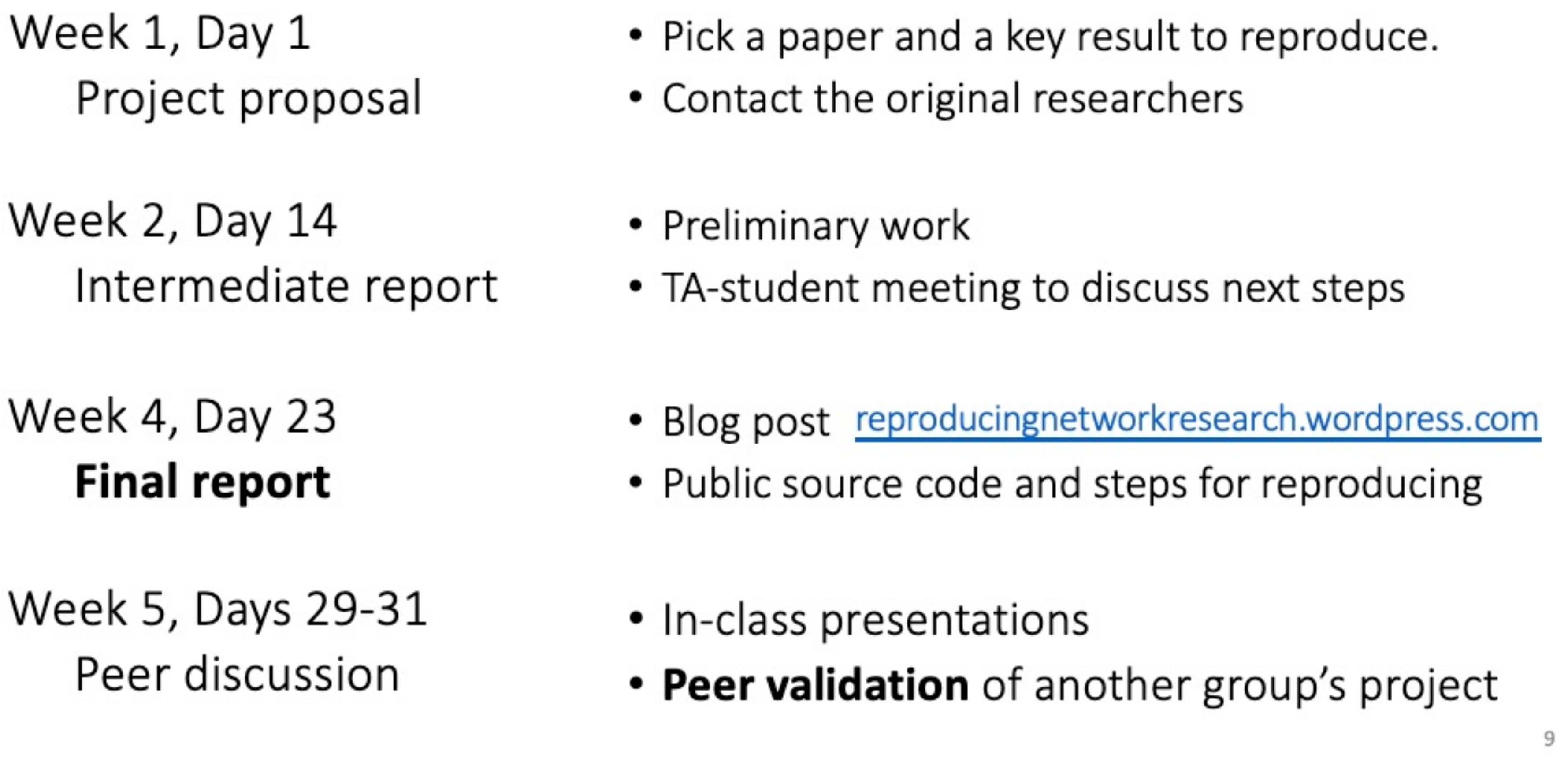
Opening Remarks

9:15am - 10:30am Session 1 - Why is reproducibility so hard?

Room: Optimist Room (Luskin Center)

Challenges with Reproducibility
Vaibhav Bajpai *(TU Munich)*; Mirja Kühlewind *(ETH Zürich)*; J Ott *(TU Munich)*; Jürgen Schönwälder *(Jacobs University Bremen)*; Anna Sperotto *(University of Twente)*; Brian Trammell *(ETH Zürich)*

Stanford CS244 – Graduate networking class

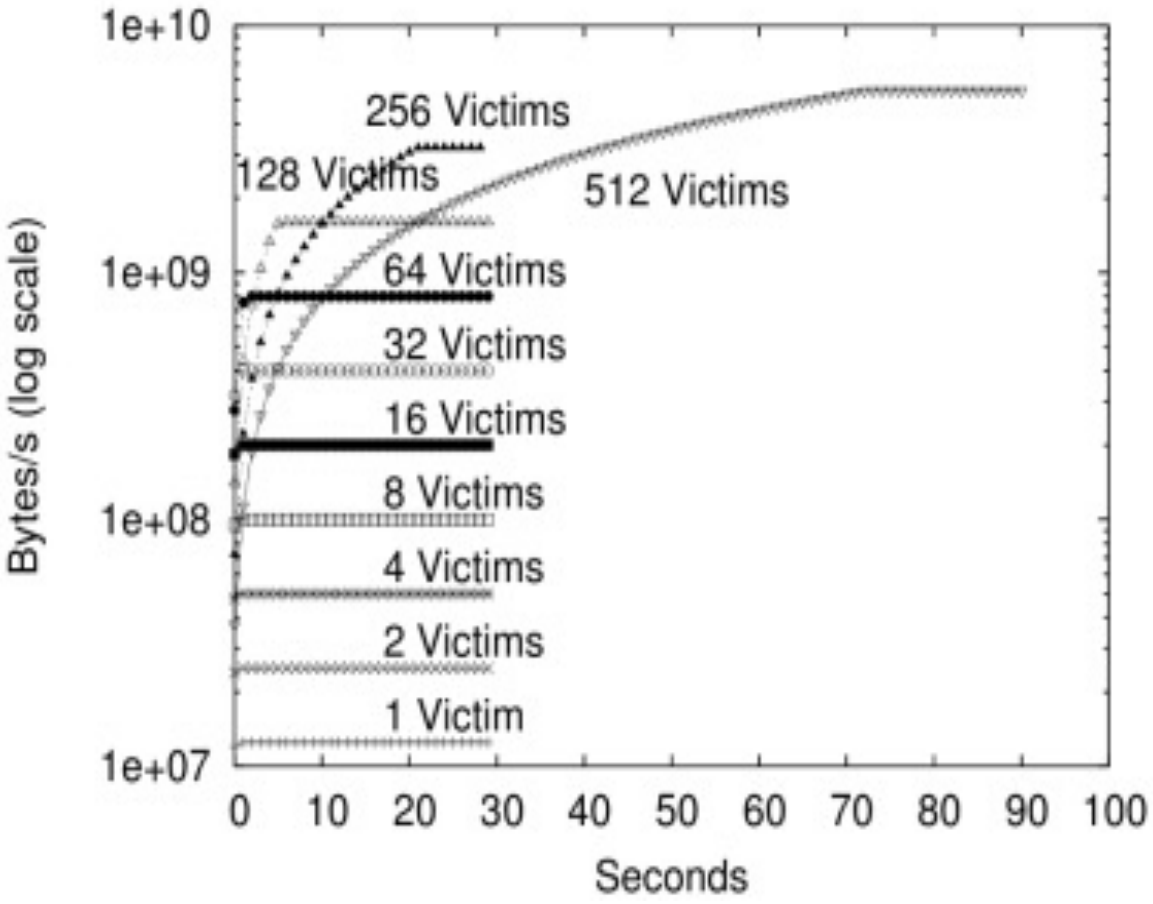


Week 1, Day 1 Project proposal	- Pick a paper and a key result to reproduce. - Contact the original researchers
Week 2, Day 14 Intermediate report	- Preliminary work - TA-student meeting to discuss next steps
Week 4, Day 23 **Final report**	- Blog post reproducingnetworkresearch.wordpress.com - Public source code and steps for reproducing
Week 5, Days 29-31 Peer discussion	- In-class presentations - **Peer validation** of another group's project

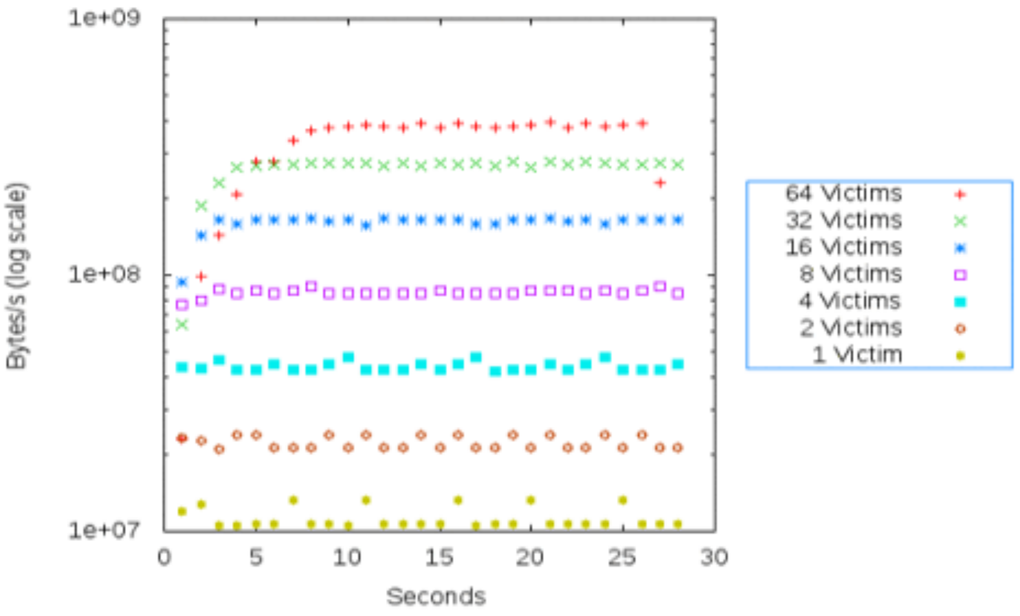
Source: **Lisa Yan** and Nick McKeown, Learning Networking by Reproducing Network Results, SIGCOMM'17 workshop

Reproduced TCP opt-ack attack

Original result from paper



Alexander and Trey’s reproduced result
(blog post)



R. Sherwood, B. Bhattacharjee, and R. Braud. Misbehaving TCP receivers can cause internet-wide congestion collapse. CCS 2005.

Source: **Lisa Yan** and Nick McKeown, Learning Networking by Reproducing Network Results, SIGCOMM’17 workshop

What kinds of reproductions?

40+ papers

Publication	# student reproductions
TCP opt-ack attack	8
Increasing TCP init cwnd	7
TCP Fast Open	7
MPTCP	6
DCTCP	5
Hedera	4
pFabric	3
Sprout	3
(24 other papers)	30

- Congestion control
- Topologies
- Security attacks
- Applications

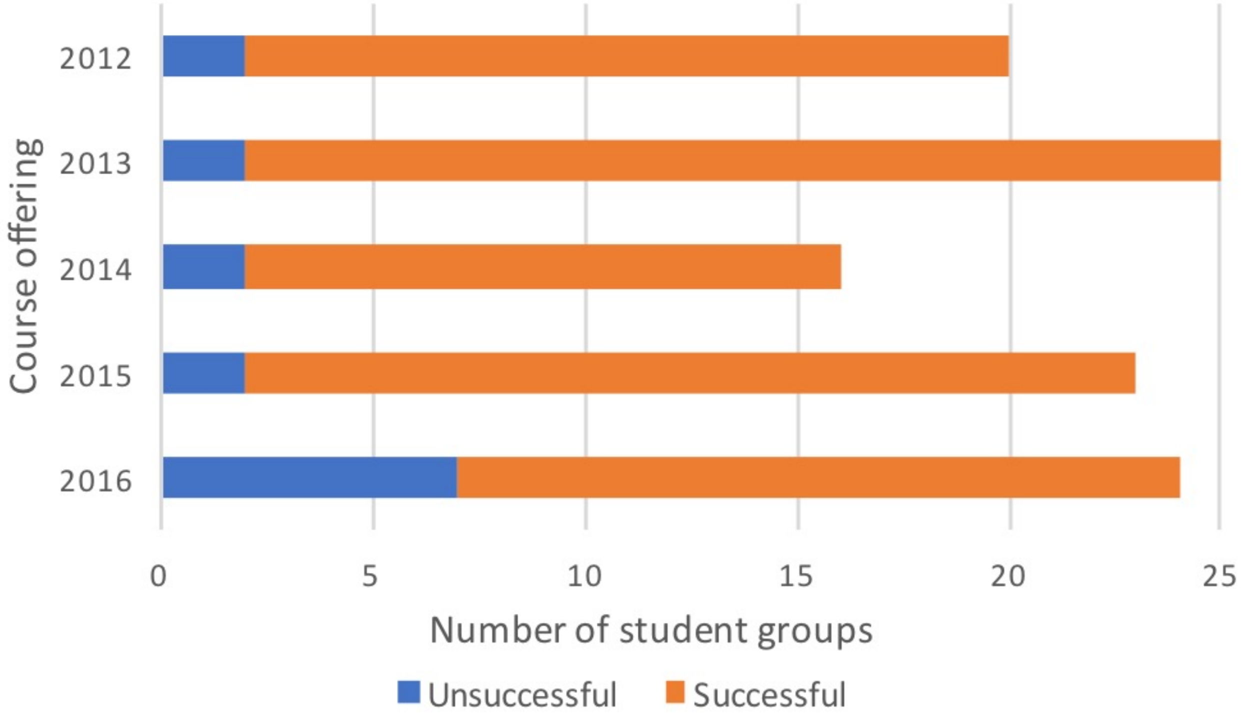
Source: **Lisa Yan** and Nick McKeown, Learning Networking by Reproducing Network Results, SIGCOMM'17 workshop

5 years of student projects

40+ papers

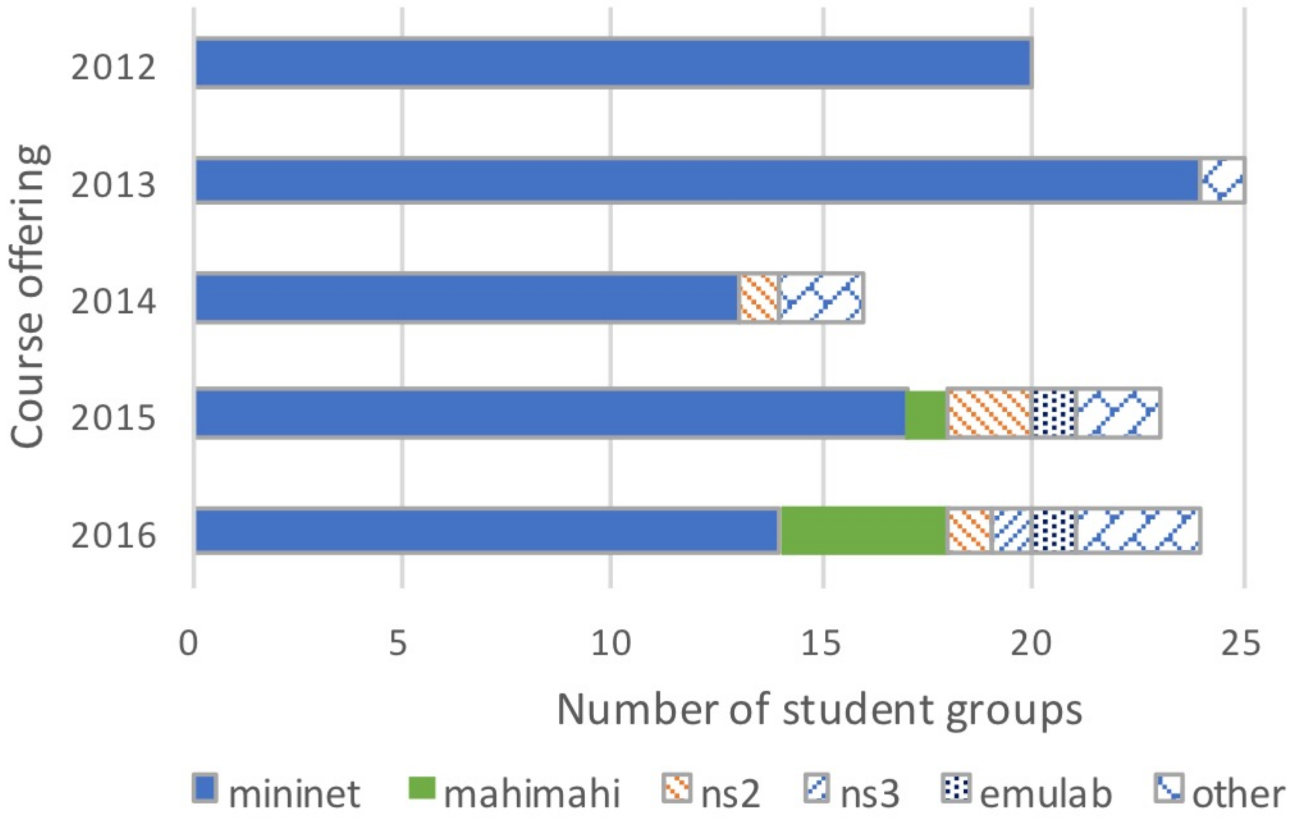
Publication	# student reproductions
TCP opt-ack attack	8
Increasing TCP init cwnd	7
TCP Fast Open	7
MPTCP	6
DCTCP	5
Hedera	4
pFabric	3
Sprout	3
(24 other papers)	30

200+ students



Source: **Lisa Yan** and Nick McKeown, Learning Networking by Reproducing Network Results, SIGCOMM'17 workshop

Emulators/simulators used by students

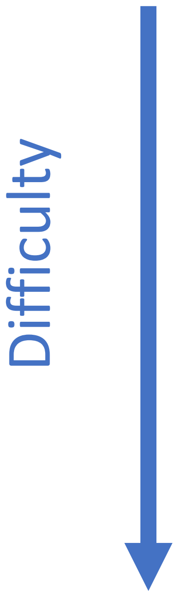


Python

Source: **Lisa Yan** and Nick McKeown, Learning Networking by Reproducing Network Results, SIGCOMM'17 workshop

Repeat, Replicate, Reproduce, Reuse

- Four different words that are not completely synonyms



Difficulty

  - Repeat
    - A team of researcher can reexecute the same experiments/analysis using their artefacts and obtain the same results
  - Replicate
    - Another team of researchers can perform the same experiments as those of the original paper with the paper’s artefacts and obtain the same results
  - Reproduce
    - Another team of researchers can collect similar data or develop similar software and obtain the same results as the original paper without using the paper’s artefacts
  - Reuse
    - Another team of researchers extends the paper’s artefacts to improve them or solve other research problems

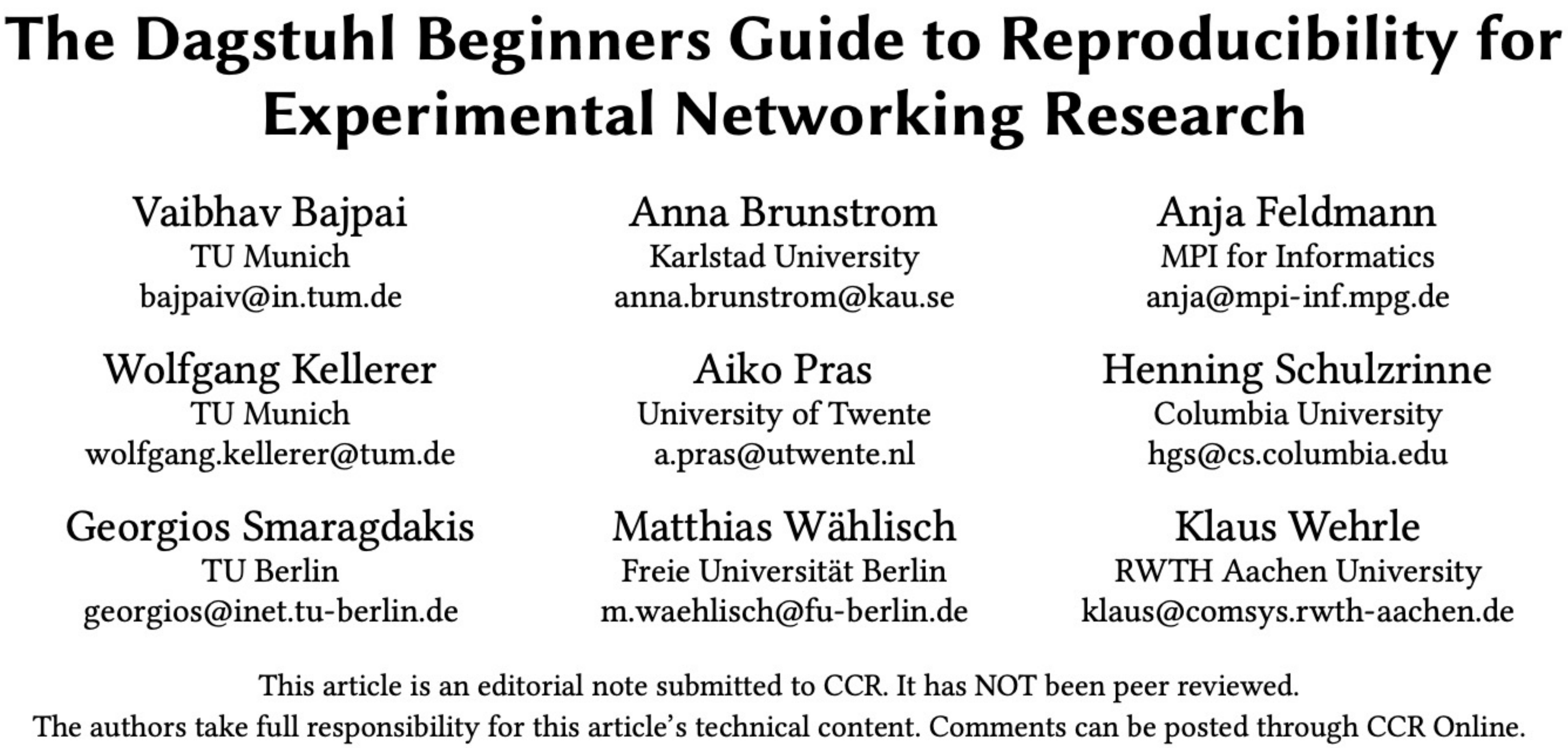
Repeatability

- Should be mandatory for **all** Ph.D. students
  - This is a basic element of the scientific process, takes notes and ensure that you can repeat your experiments
    - As computer scientists, your experiments are much more deterministics than in natural sciences
    - Don’t underestimate the importance of repeatability
    - A few years from now, questions from your jury members might force you to rerun some experiments

Agenda

- The reproducibility crisis
- **Dagsthul's guide to Reproducibility for Experimental Networking Research**
- Artefacts Evaluations

Dagsthul’s guide to reproducible research



The Dagstuhl Beginners Guide to Reproducibility for Experimental Networking Research

Vaibhav Bajpai
TU Munich
bajpaiv@in.tum.de

Anna Brunstrom
Karlstad University
anna.brunstrom@kau.se

Anja Feldmann
MPI for Informatics
anja@mpi-inf.mpg.de

Wolfgang Kellerer
TU Munich
wolfgang.kellerer@tum.de

Aiko Pras
University of Twente
a.pras@utwente.nl

Henning Schulzrinne
Columbia University
hgs@cs.columbia.edu

Georgios Smaragdakis
TU Berlin
georgios@inet.tu-berlin.de

Matthias Wählisch
Freie Universität Berlin
m.waehlisch@fu-berlin.de

Klaus Wehrle
RWTH Aachen University
klaus@comsys.rwth-aachen.de

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article’s technical content. Comments can be posted through CCR Online.

Lots of very useful advice for researchers

- Problem formulation and design
    - Hypothesize
        - Think before, experiment after
            - Try to predict the results of an experiment before running it
    - Plan and solicit early feedback
        - Plan your experiments and discuss with colleagues before executing them
            - Have you considered all the key parameters ?
            - What is the lesson that you want to learn from each experiment ?
    - Iterate
        - You will rerun your experiments several times, automate them
            - Put everything in scripts that can execute autonomously at night
            - Rerun each experiment several times to check if results are stable
            - Check for external interference (e.g. nightly backup running on the server you use for experiments and which consume CPU,...)

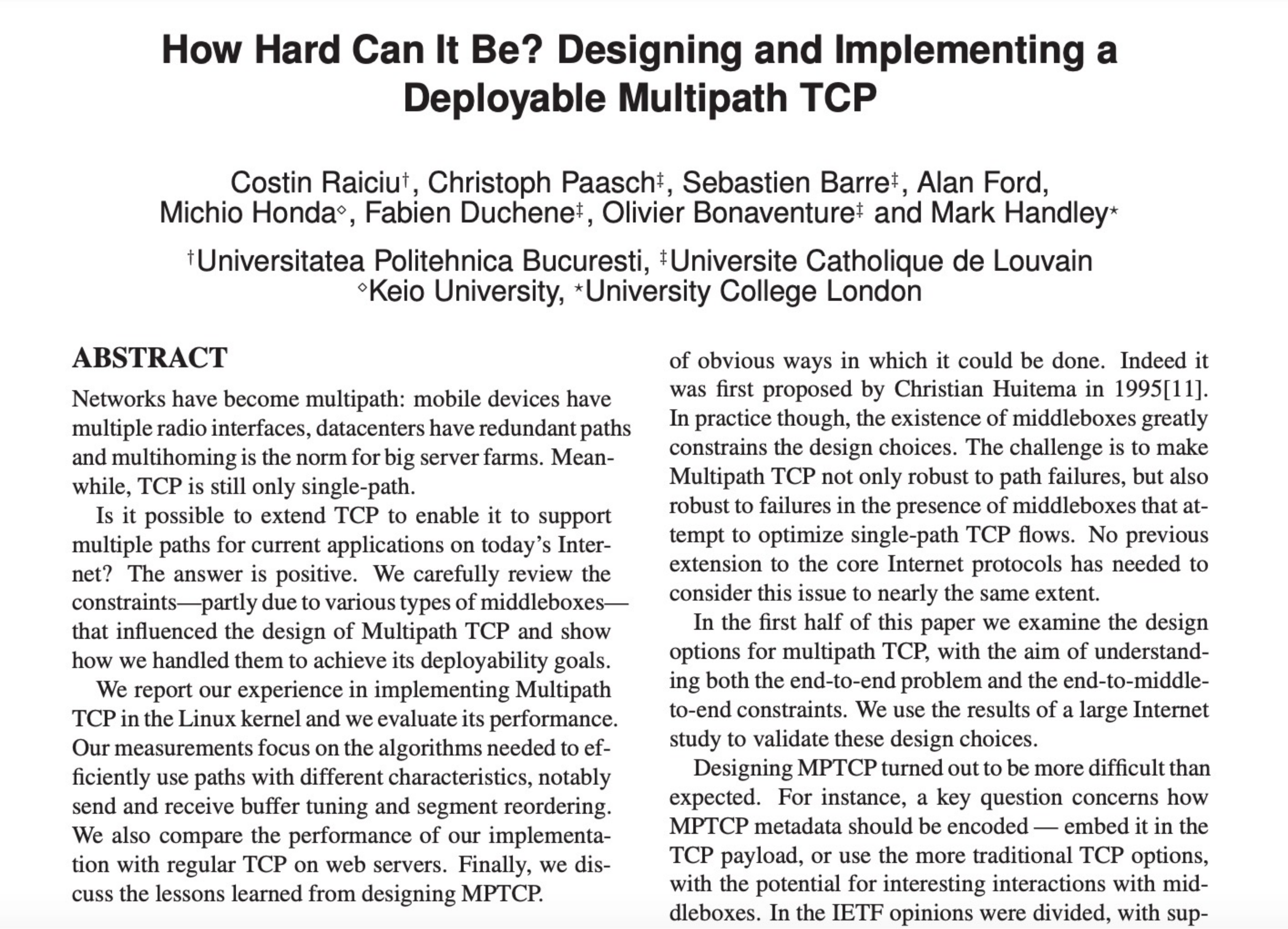
Documentation

- Use a version control system
    - For all the code you use and the measurement scripts
- Provide detailed logs of your experiments
    - The measured data is important, but also collect
        - The system state (networkin configuration, OS, libraries, version of measurement software)
        - The raw data produced by the measurement tool
            - shoul include the version of the measurement script you used so that you precisely know which script was used to collect a given measurement
    - Metadata is as important and sometimes more important than data
- Keep regular backups
    - Of everything that is important, your disk will fail, you do not know when

Experimentation

- Always start with a basic validation
  - simple experiment whose result is unsurprising to check that there are non configuration errors. If it fails, script stops and you investigate
- Do no reinvent the wheel
  - Check for well-maintained tools that you could use for your experiments
    - tool developers have already coped with many details that affect measurements and you could ignore at first attempt
    - this is reassuring for reviewers, because they know what the tool measures
  - If you develop your own tool, start from a test suite
    - Write a technical report to convince other researchers of the quality of your tool
- Monitor your experiment
  - Check for problems that could affect your measurements (disk full, competing processes, network reconfiguration, ...)

Multipath TCP



How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP

Costin Raiciu[†], Christoph Paasch[‡], Sebastien Barre[‡], Alan Ford,
Michio Honda[⋄], Fabien Duchene[‡], Olivier Bonaventure[‡] and Mark Handley[⋆]

[†]Universitatea Politehnica Bucuresti, [‡]Universite Catholique de Louvain
[⋄]Keio University, [⋆]University College London

ABSTRACT

Networks have become multipath: mobile devices have multiple radio interfaces, datacenters have redundant paths and multihoming is the norm for big server farms. Meanwhile, TCP is still only single-path.

Is it possible to extend TCP to enable it to support multiple paths for current applications on today's Internet? The answer is positive. We carefully review the constraints—partly due to various types of middleboxes—that influenced the design of Multipath TCP and show how we handled them to achieve its deployability goals.

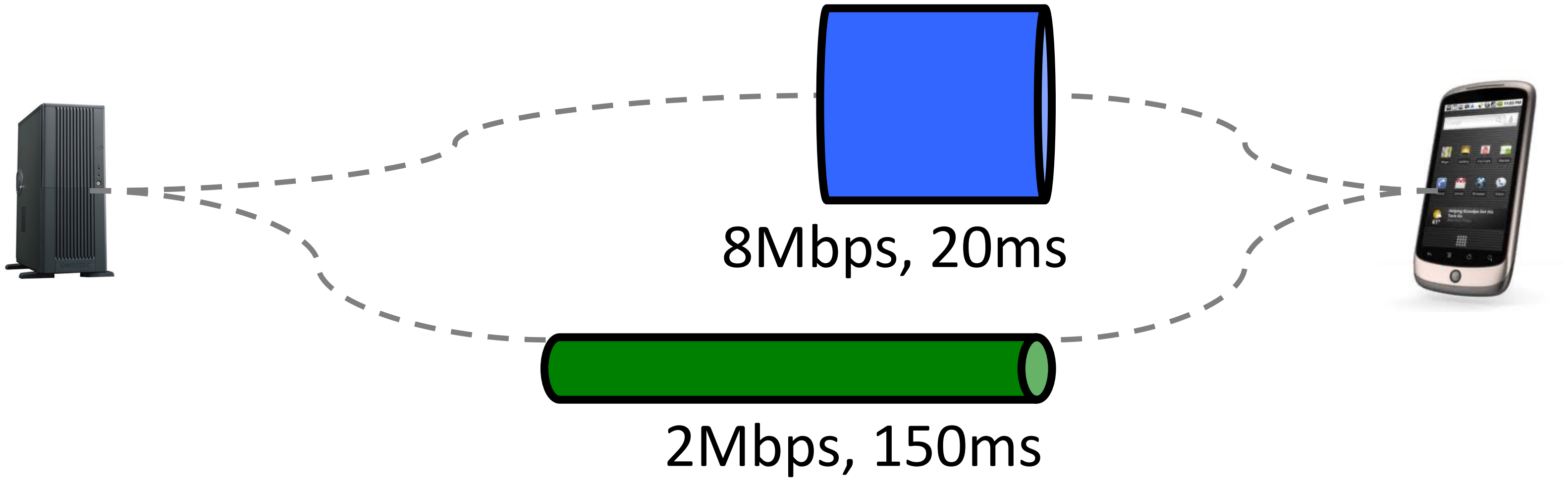
We report our experience in implementing Multipath TCP in the Linux kernel and we evaluate its performance. Our measurements focus on the algorithms needed to efficiently use paths with different characteristics, notably send and receive buffer tuning and segment reordering. We also compare the performance of our implementation with regular TCP on web servers. Finally, we discuss the lessons learned from designing MPTCP.

of obvious ways in which it could be done. Indeed it was first proposed by Christian Huitema in 1995[11]. In practice though, the existence of middleboxes greatly constrains the design choices. The challenge is to make Multipath TCP not only robust to path failures, but also robust to failures in the presence of middleboxes that attempt to optimize single-path TCP flows. No previous extension to the core Internet protocols has needed to consider this issue to nearly the same extent.

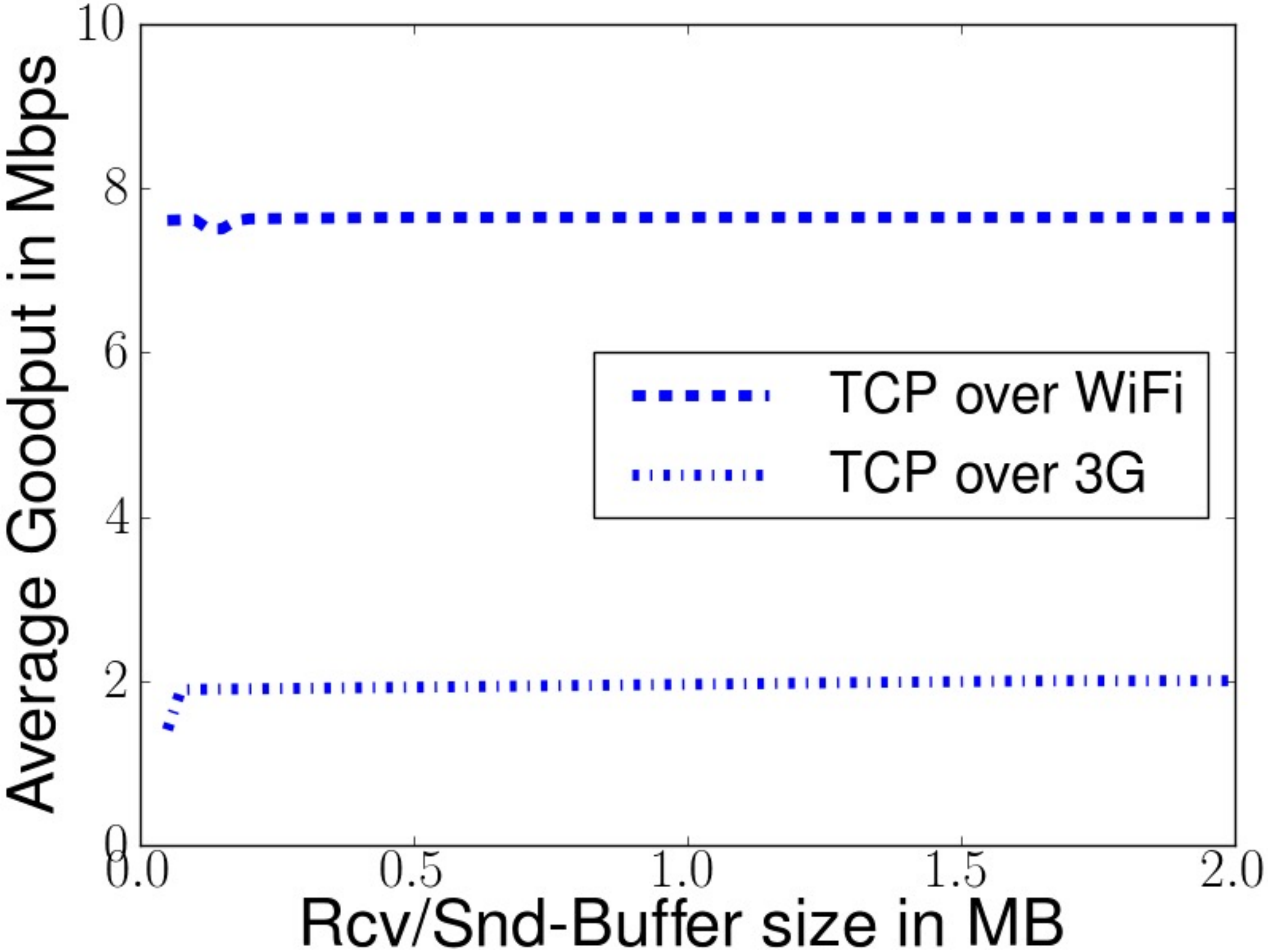
In the first half of this paper we examine the design options for multipath TCP, with the aim of understanding both the end-to-end problem and the end-to-middle-to-end constraints. We use the results of a large Internet study to validate these design choices.

Designing MPTCP turned out to be more difficult than expected. For instance, a key question concerns how MPTCP metadata should be encoded — embed it in the TCP payload, or use the more traditional TCP options, with the potential for interesting interactions with middleboxes. In the IETF opinions were divided, with sup-

MPTCP over WiFi/3G

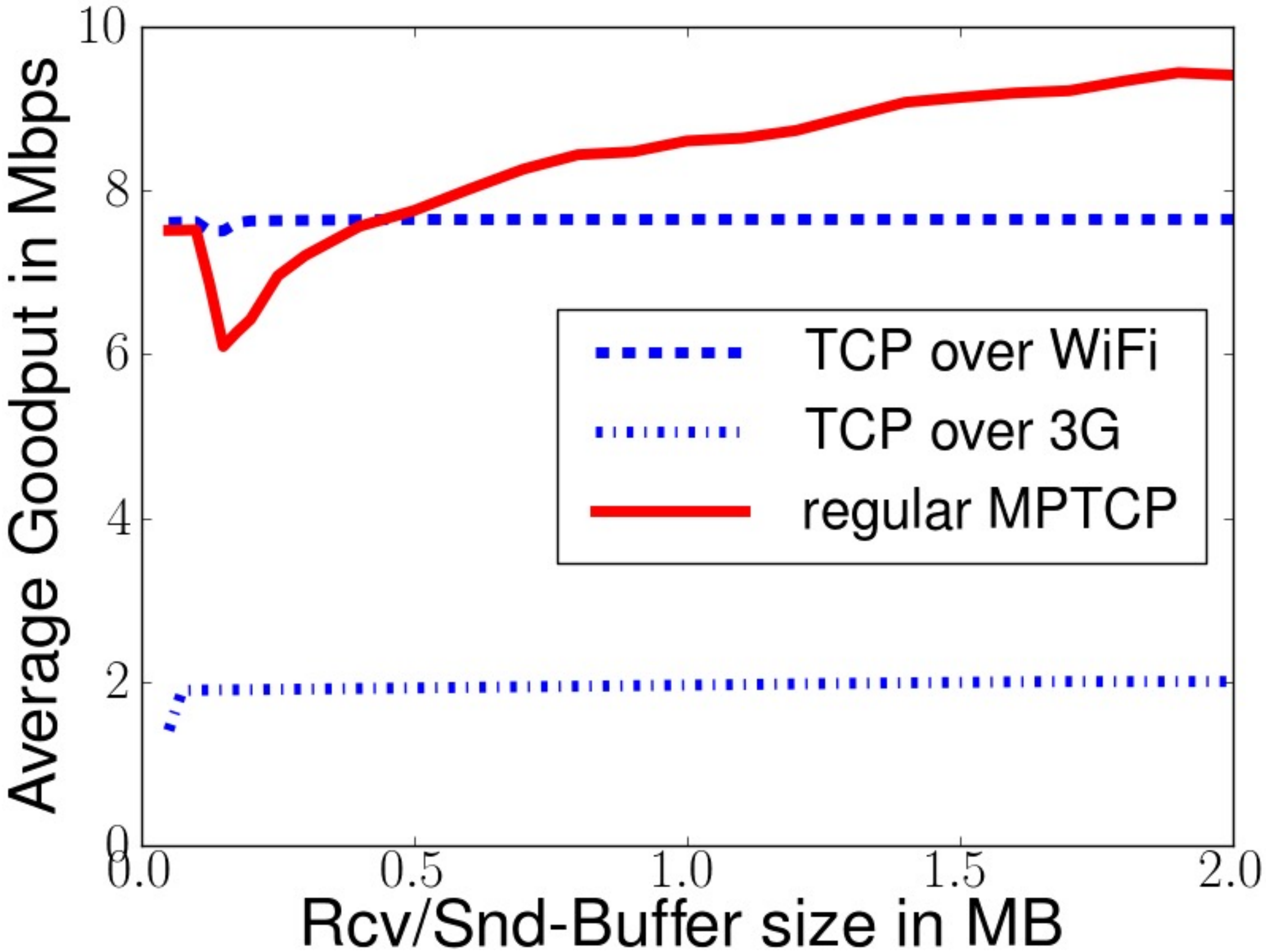


TCP over WiFi/3G



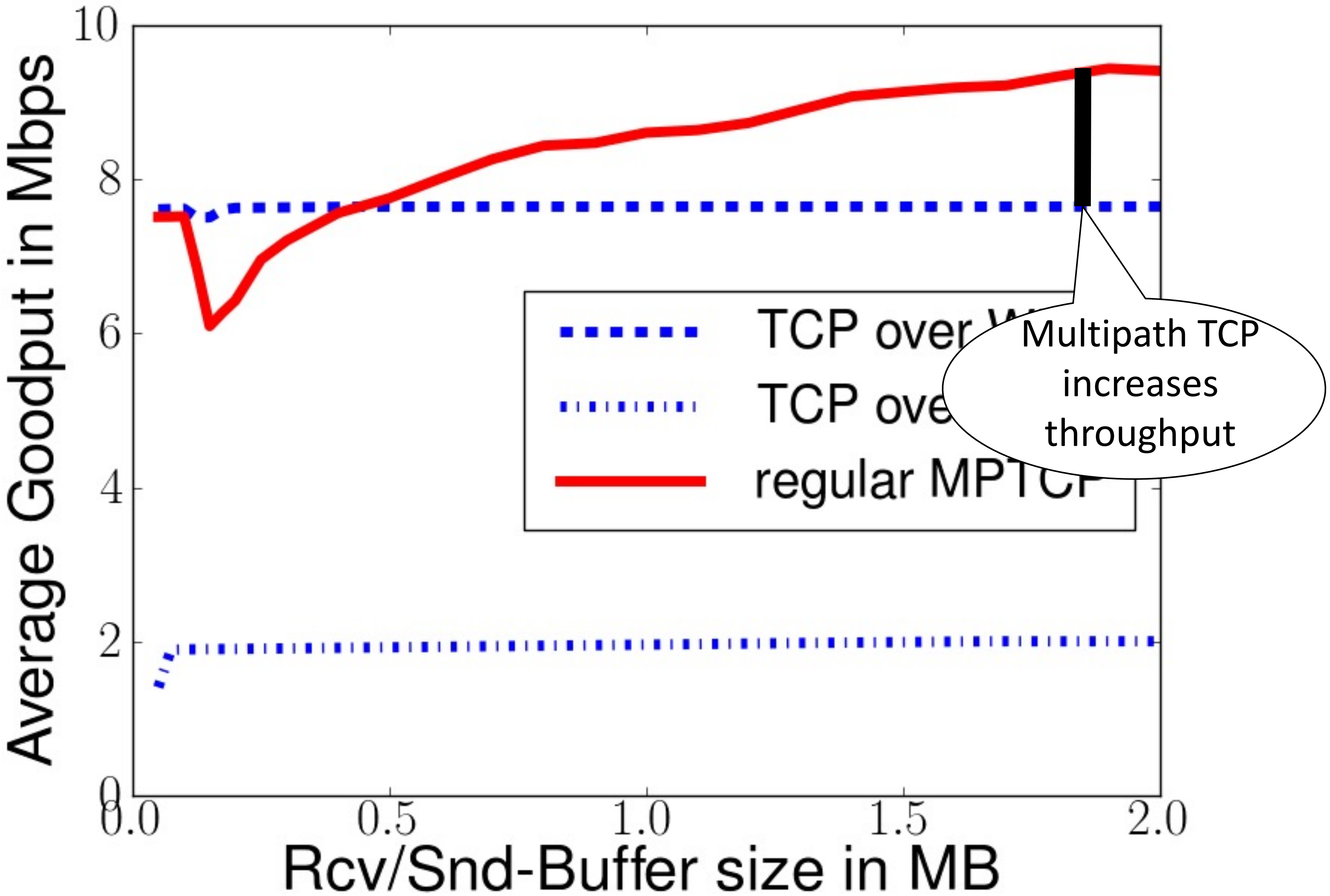
C. Raiciu, et al. “How hard can it be? designing and implementing a deployable multipath TCP,” NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, 2012.

MPTCP over WiFi/3G



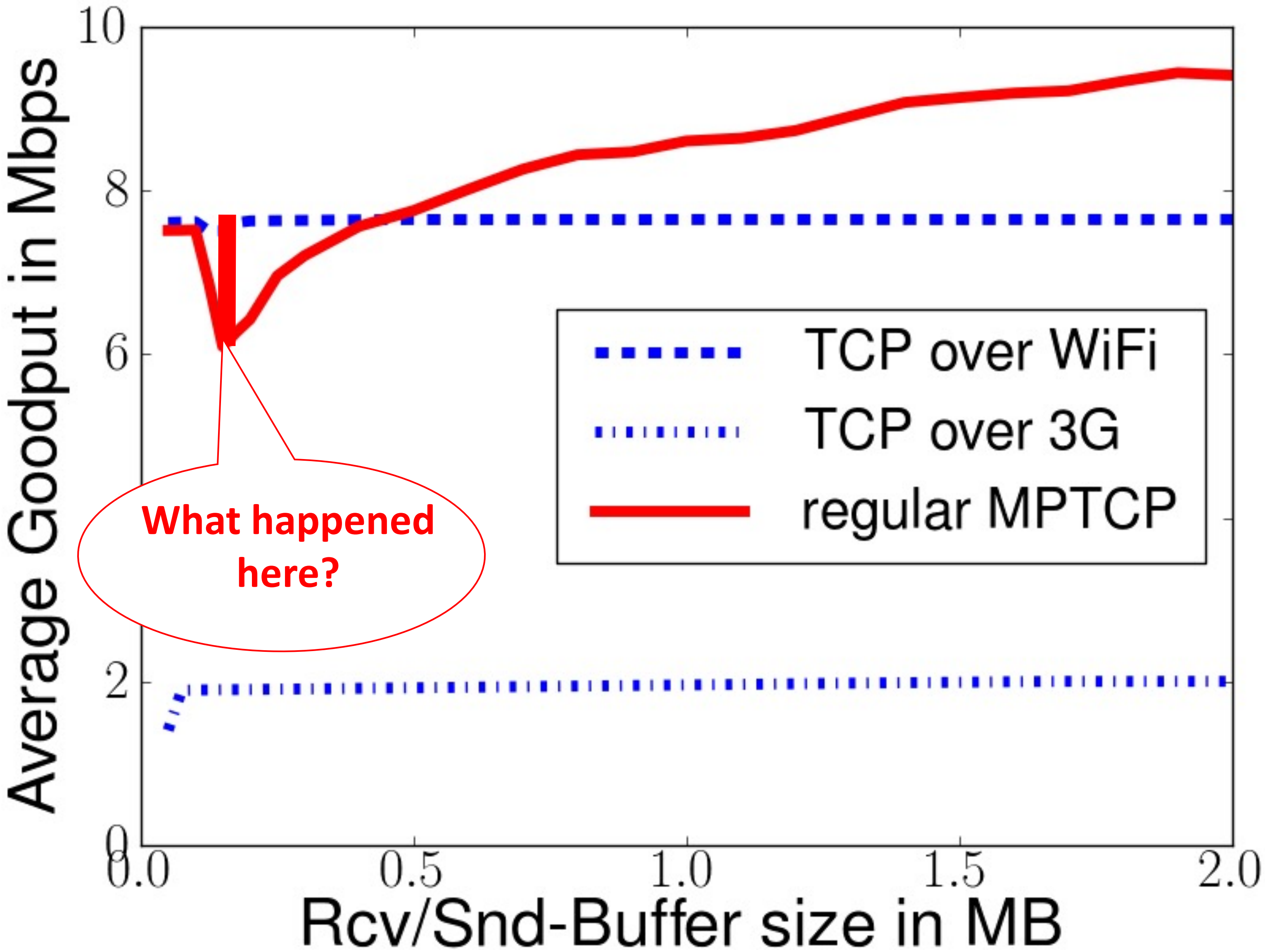
C. Raiciu, et al. “How hard can it be? designing and implementing a deployable multipath TCP,” NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, 2012.

MPTCP over WiFi/3G



Multipath TCP increases throughput

MPTCP over WiFi/3G

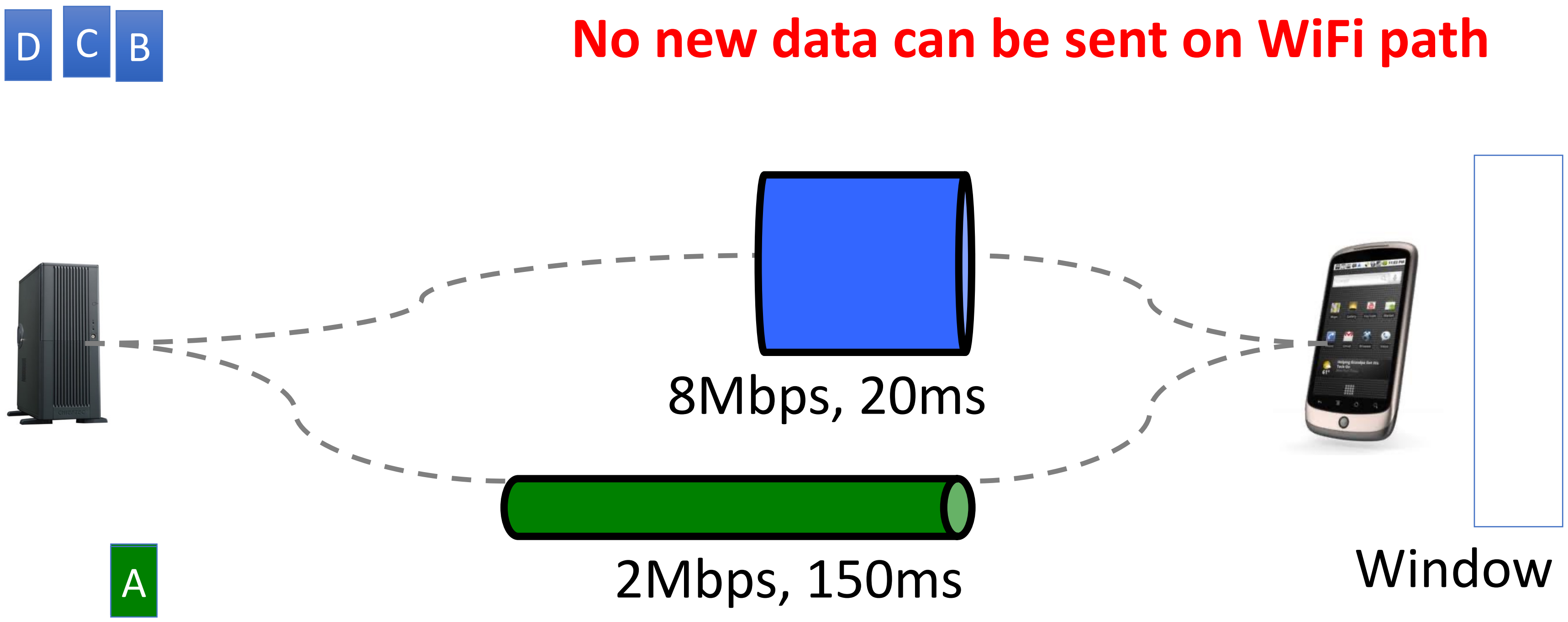


What happened here?

Understanding the performance issue

Window full !

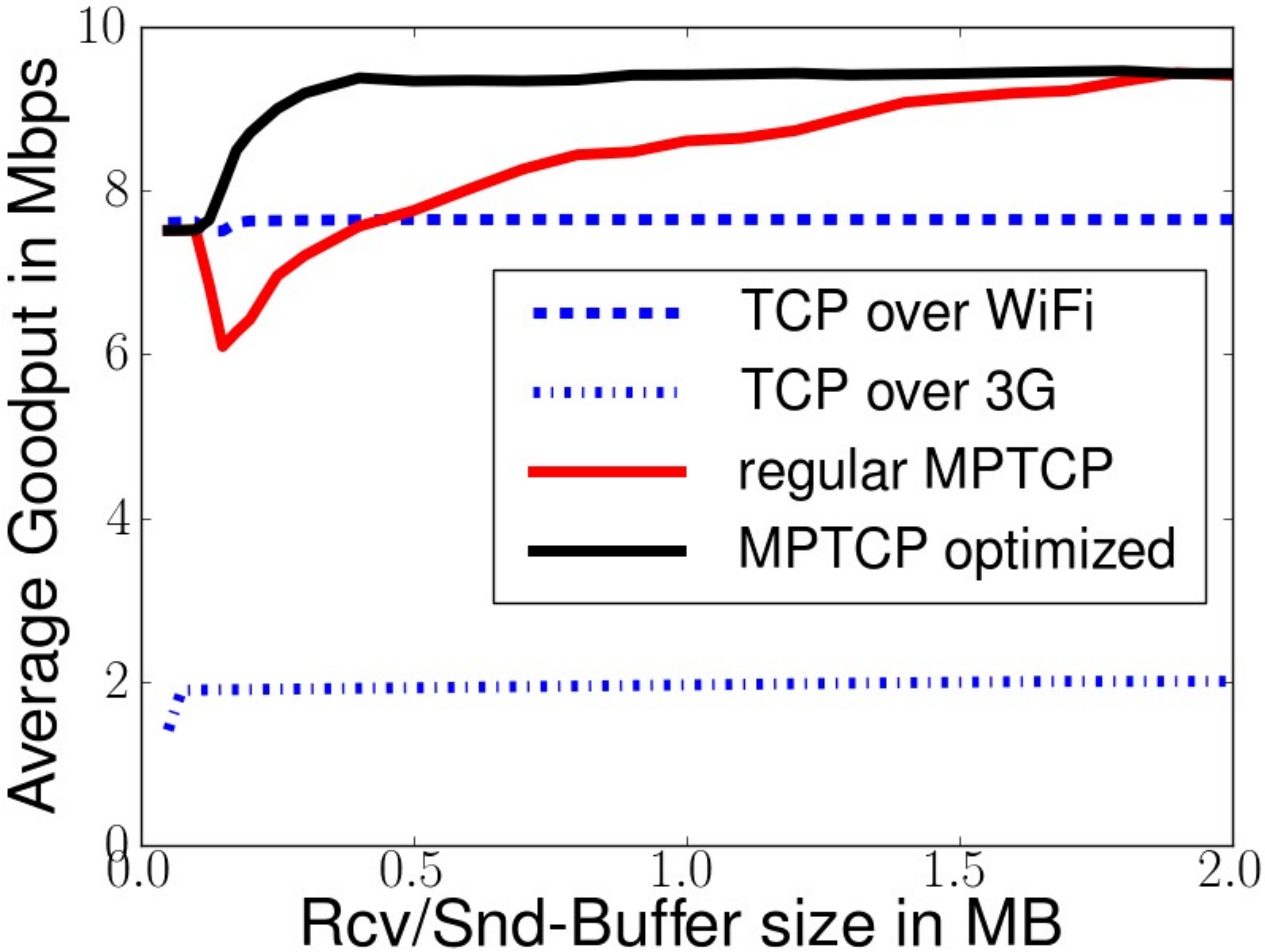
No new data can be sent on WiFi path



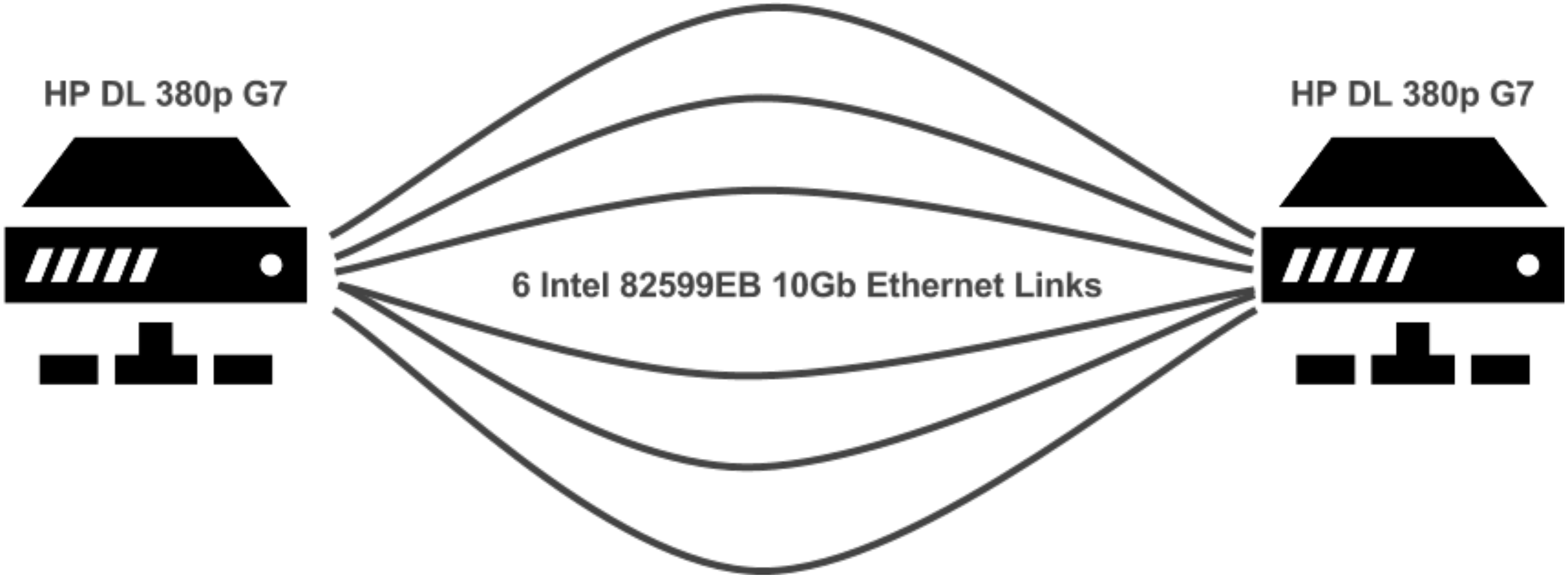
Reinject segment on fast path

Halve congestion window on slow subflow

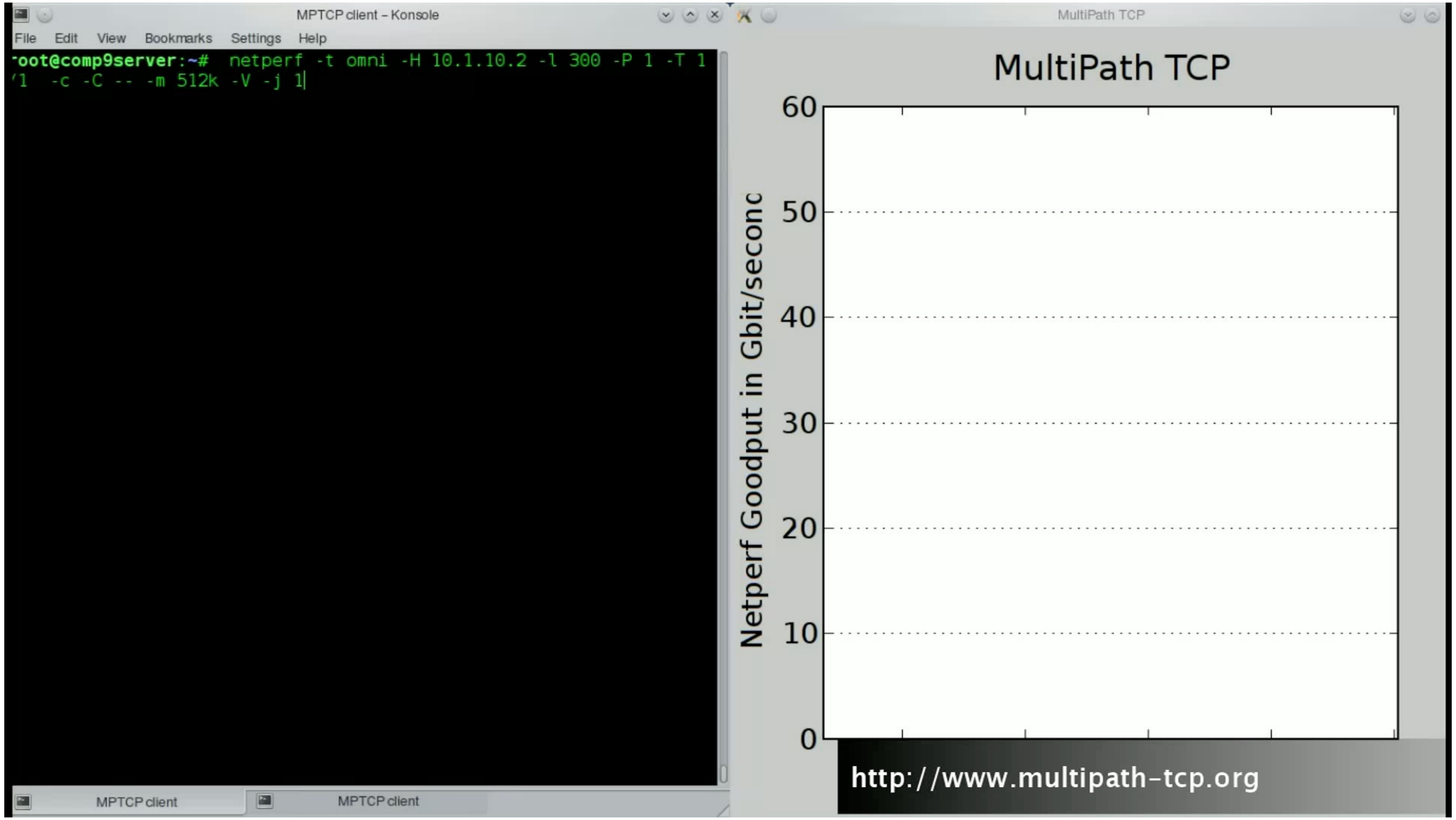
MPTCP over WiFi/3G



Example: how fast can MPTCP go ?



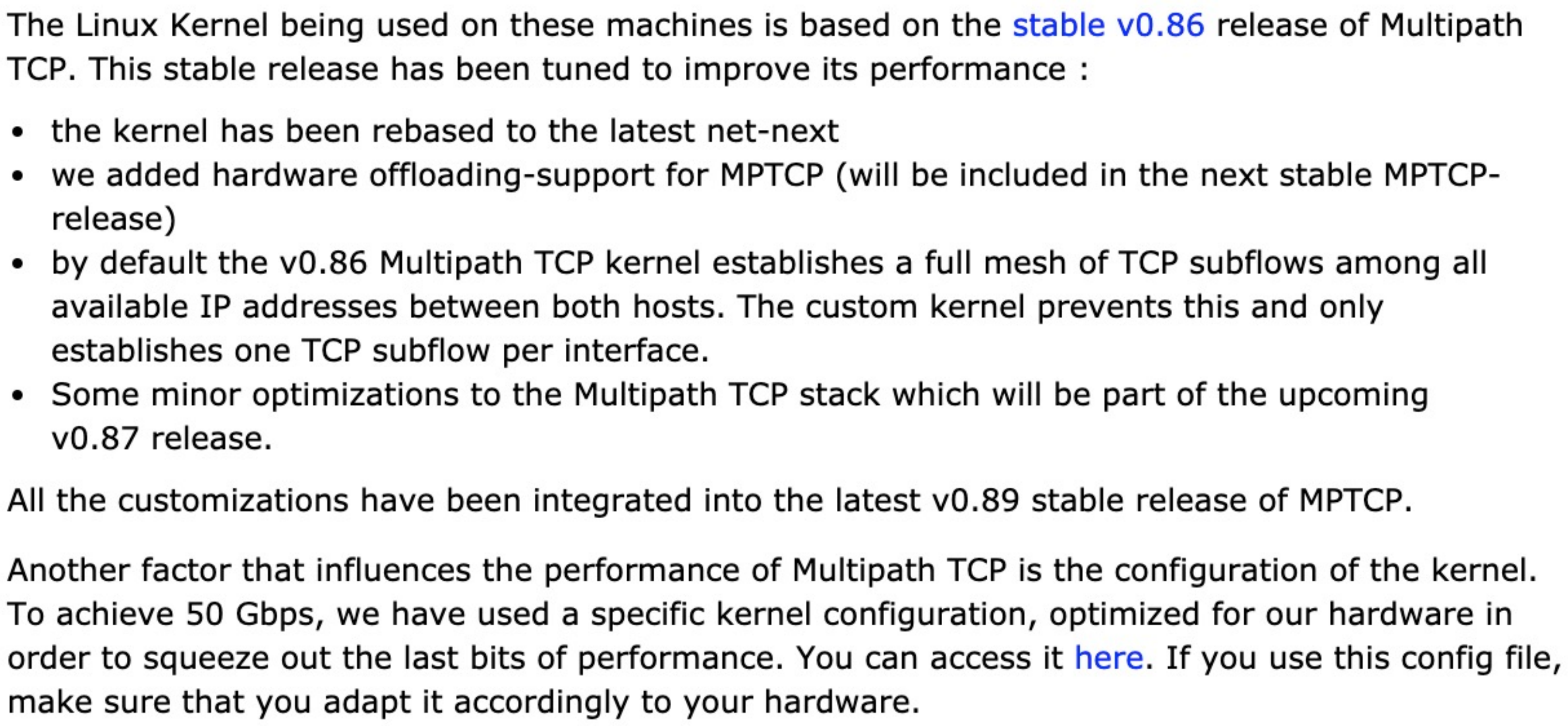
How fast can Multipath TCP go ?



```
'oot@comp9server:~#  netperf -t omni -H 10.1.10.2 -l 300 -P 1 -T 1
'1  -c -C -- -m 512k -V -j 1
```

Reproducible measurements

- Software used for the measurements



The Linux Kernel being used on these machines is based on the stable v0.86 release of Multipath TCP. This stable release has been tuned to improve its performance :

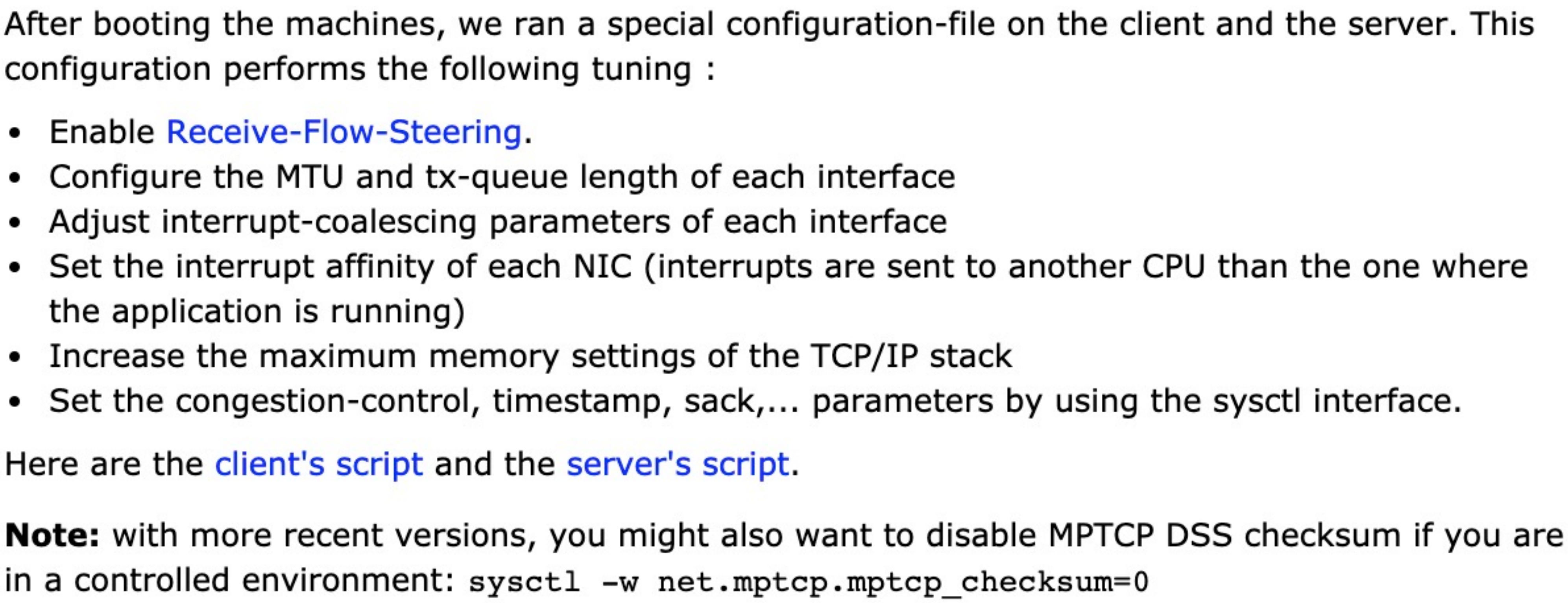
- the kernel has been rebased to the latest net-next
- we added hardware offloading-support for MPTCP (will be included in the next stable MPTCP-release)
- by default the v0.86 Multipath TCP kernel establishes a full mesh of TCP subflows among all available IP addresses between both hosts. The custom kernel prevents this and only establishes one TCP subflow per interface.
- Some minor optimizations to the Multipath TCP stack which will be part of the upcoming v0.87 release.

All the customizations have been integrated into the latest v0.89 stable release of MPTCP.

Another factor that influences the performance of Multipath TCP is the configuration of the kernel. To achieve 50 Gbps, we have used a specific kernel configuration, optimized for our hardware in order to squeeze out the last bits of performance. You can access it here. If you use this config file, make sure that you adapt it accordingly to your hardware.

For more details, http://multipath-tcp.org/pmwiki.php?n=Main.50Gbps

Reproducible measurements (2)



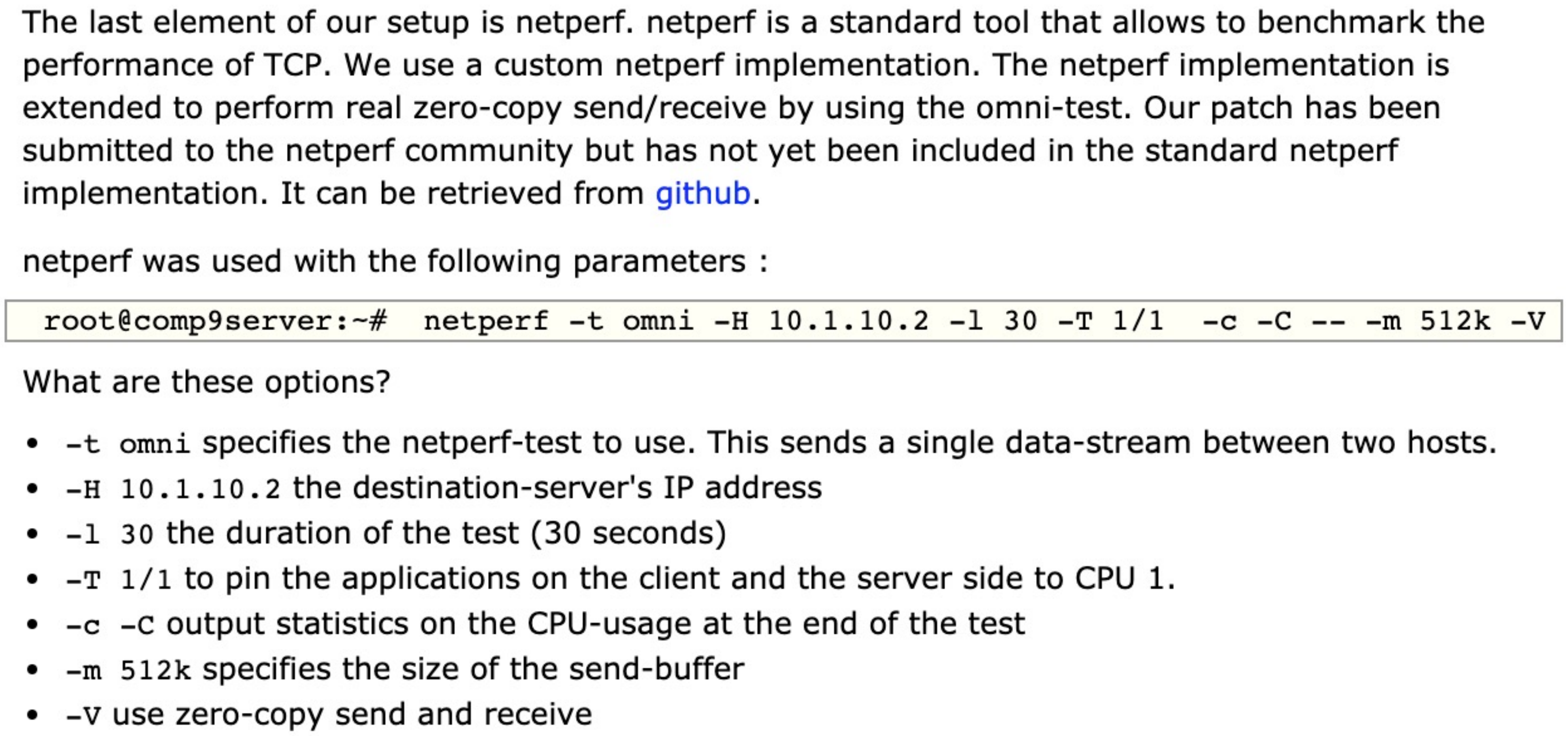
After booting the machines, we ran a special configuration-file on the client and the server. This configuration performs the following tuning :

- Enable Receive-Flow-Steering.
- Configure the MTU and tx-queue length of each interface
- Adjust interrupt-coalescing parameters of each interface
- Set the interrupt affinity of each NIC (interrupts are sent to another CPU than the one where the application is running)
- Increase the maximum memory settings of the TCP/IP stack
- Set the congestion-control, timestamp, sack,... parameters by using the sysctl interface.

Here are the client's script and the server's script.

Note: with more recent versions, you might also want to disable MPTCP DSS checksum if you are in a controlled environment: `sysctl -w net.mptcp.mptcp_checksum=0`

Reproducible measurements (3)



The last element of our setup is netperf. netperf is a standard tool that allows to benchmark the performance of TCP. We use a custom netperf implementation. The netperf implementation is extended to perform real zero-copy send/receive by using the omni-test. Our patch has been submitted to the netperf community but has not yet been included in the standard netperf implementation. It can be retrieved from github.

netperf was used with the following parameters :

```
root@comp9server:~#  netperf -t omni -H 10.1.10.2 -l 30 -T 1/1  -c -C -- -m 512k -V
```

What are these options?

- `-t omni` specifies the netperf-test to use. This sends a single data-stream between two hosts.
- `-H 10.1.10.2` the destination-server's IP address
- `-l 30` the duration of the test (30 seconds)
- `-T 1/1` to pin the applications on the client and the server side to CPU 1.
- `-c -C` output statistics on the CPU-usage at the end of the test
- `-m 512k` specifies the size of the send-buffer
- `-V` use zero-copy send and receive

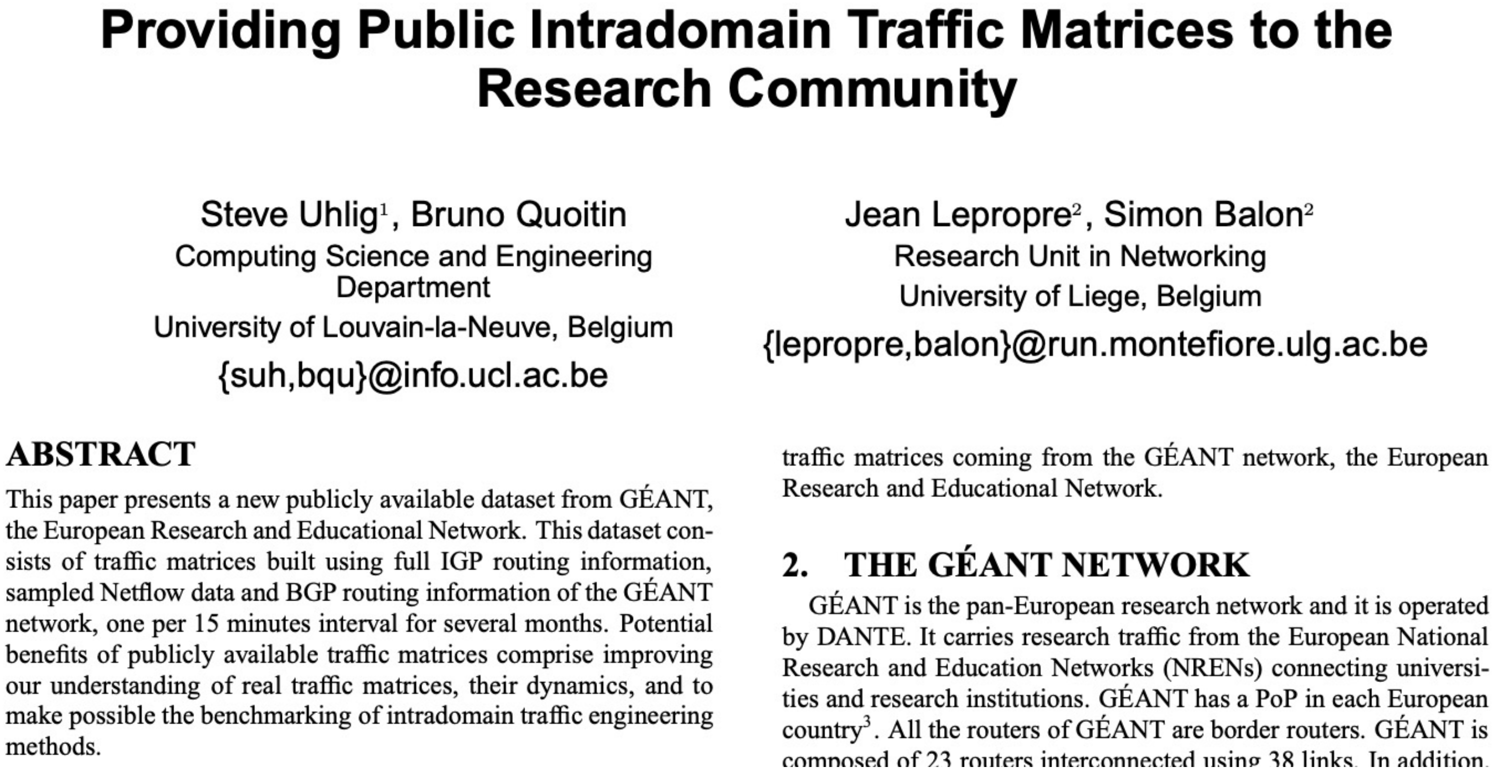
Handling data

- Privacy, anonymization and ethics
  - Control plane measurements are usually fine
  - Data plane measurements can be more difficulty
    - You probably need authorization to collect packets/Netflow from real users
    - If you probe external servers, you probably need their authorization

- Data integrity
  - Define checks that allow you to verify that the data has been correctly captured
    - Can you miss some data ? Can your measurements be polluted by caches ?

GEANT Traffic matrices

- The project



Providing Public Intradomain Traffic Matrices to the Research Community

Steve Uhlig[1], Bruno Quoitin
Computing Science and Engineering Department
University of Louvain-la-Neuve, Belgium
{suh,bqu}@info.ucl.ac.be

Jean Lepropre[2], Simon Balon[2]
Research Unit in Networking
University of Liege, Belgium
{lepropre,balon}@run.montefiore.ulg.ac.be

ABSTRACT

This paper presents a new publicly available dataset from GÉANT, the European Research and Educational Network. This dataset consists of traffic matrices built using full IGP routing information, sampled Netflow data and BGP routing information of the GÉANT network, one per 15 minutes interval for several months. Potential benefits of publicly available traffic matrices comprise improving our understanding of real traffic matrices, their dynamics, and to make possible the benchmarking of intradomain traffic engineering methods.

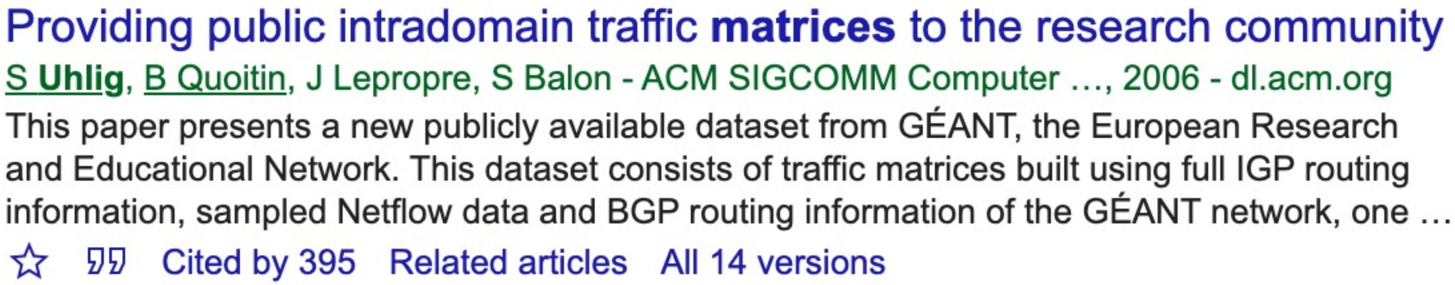
traffic matrices coming from the GÉANT network, the European Research and Educational Network.

2. THE GÉANT NETWORK

GÉANT is the pan-European research network and it is operated by DANTE. It carries research traffic from the European National Research and Education Networks (NRENs) connecting universities and research institutions. GÉANT has a PoP in each European country[3]. All the routers of GÉANT are border routers. GÉANT is composed of 23 routers interconnected using 38 links. In addition,

Repeatability/Replicability/Reproducibility

- Repeatability
  - Dataset collected by GEANT (Netflow) and post-processed
    - We could not release the Netflow data, only the post processed information
  - We reused the dataset in several PhD theses
- Replicability
  - The dataset has been reused by other researchers, but not the tool used to process the data
- Reproducibility
- Reusability
  - Regularly used dataset

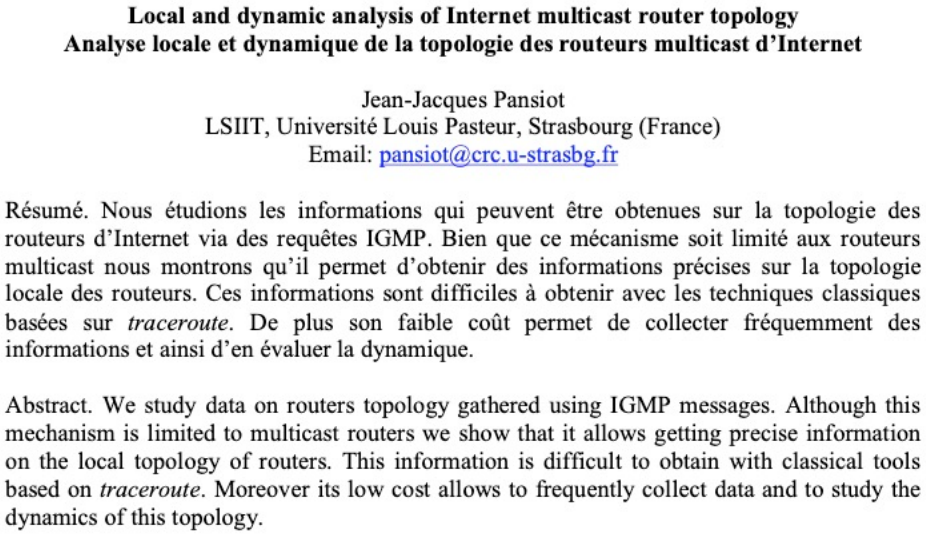


Providing public intradomain traffic **matrices** to the research community

S **Uhlig**, B Quoitin, J Lepropre, S Balon - ACM SIGCOMM Computer …, 2006 - dl.acm.org

This paper presents a new publicly available dataset from GÉANT, the European Research and Educational Network. This dataset consists of traffic matrices built using full IGP routing information, sampled Netflow data and BGP routing information of the GÉANT network, one …

When a dataset brings new ideas



Local and dynamic analysis of Internet multicast router topology
Analyse locale et dynamique de la topologie des routeurs multicast d'Internet

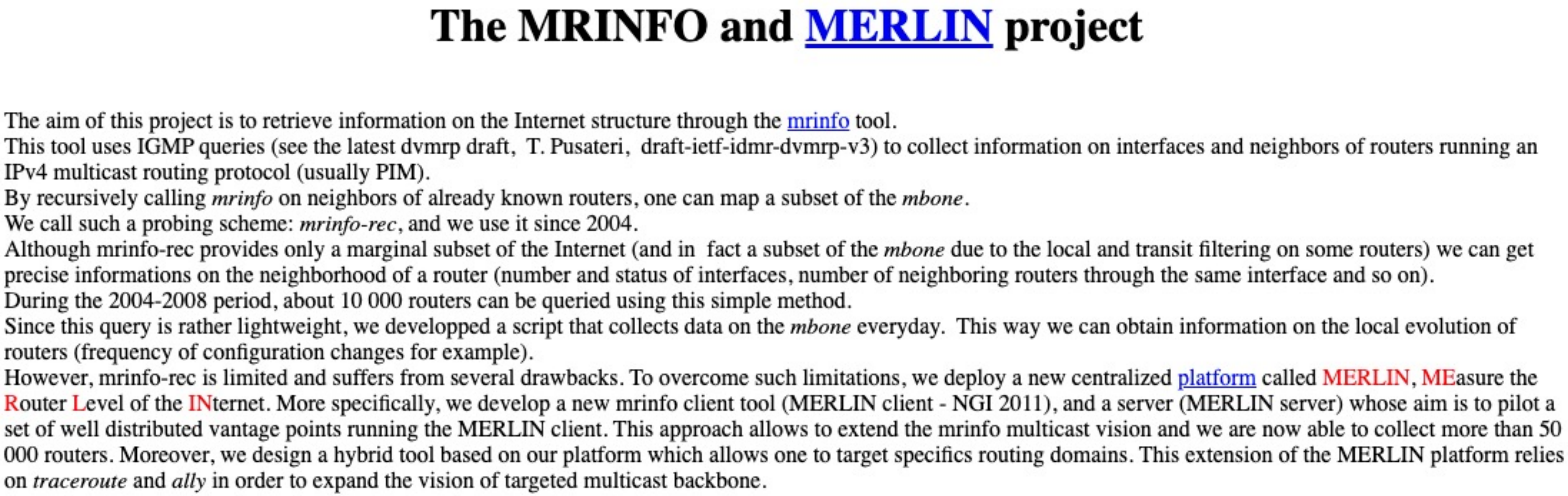
Jean-Jacques Pansiot
LSIIT, Université Louis Pasteur, Strasbourg (France)
Email: pansiot@crc.u-strasbg.fr

Résumé. Nous étudions les informations qui peuvent être obtenues sur la topologie des routeurs d'Internet via des requêtes IGMP. Bien que ce mécanisme soit limité aux routeurs multicast nous montrons qu'il permet d'obtenir des informations précises sur la topologie locale des routeurs. Ces informations sont difficiles à obtenir avec les techniques classiques basées sur *traceroute*. De plus son faible coût permet de collecter fréquemment des informations et ainsi d'en évaluer la dynamique.

Abstract. We study data on routers topology gathered using IGMP messages. Although this mechanism is limited to multicast routers we show that it allows getting precise information on the local topology of routers. This information is difficult to obtain with classical tools based on *traceroute*. Moreover its low cost allows to frequently collect data and to study the dynamics of this topology.

http://svnet.u-strasbg.fr/mrinfo/

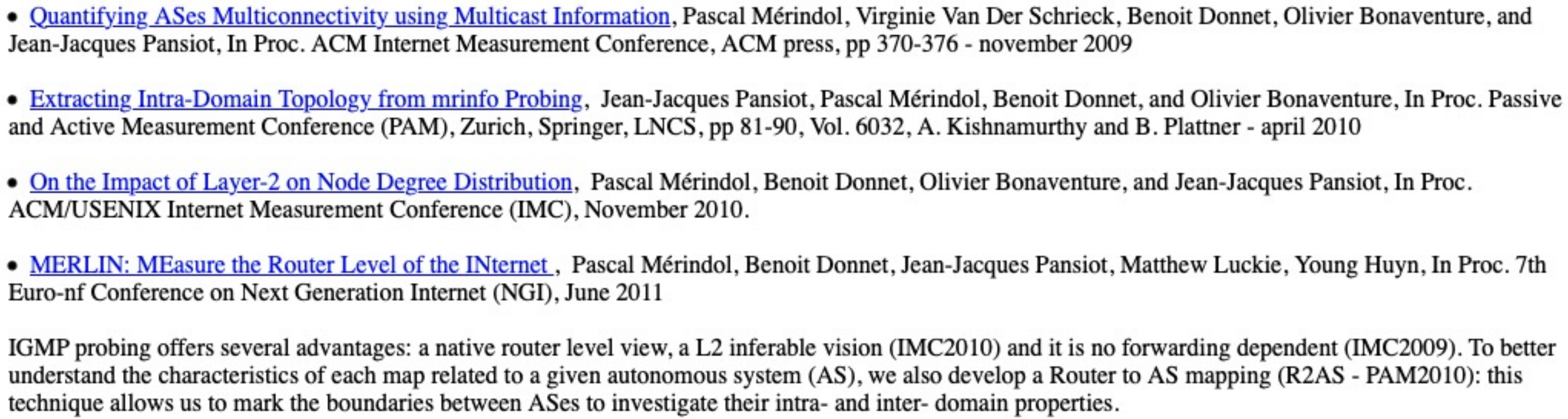
The mrinfo project



The MRINFO and MERLIN project

The aim of this project is to retrieve information on the Internet structure through the mrinfo tool.
This tool uses IGMP queries (see the latest dvmrp draft, T. Pusateri, draft-ietf-idmr-dvmrp-v3) to collect information on interfaces and neighbors of routers running an IPv4 multicast routing protocol (usually PIM).
By recursively calling *mrinfo* on neighbors of already known routers, one can map a subset of the *mbone*.
We call such a probing scheme: *mrinfo-rec*, and we use it since 2004.
Although mrinfo-rec provides only a marginal subset of the Internet (and in fact a subset of the *mbone* due to the local and transit filtering on some routers) we can get precise informations on the neighborhood of a router (number and status of interfaces, number of neighboring routers through the same interface and so on).
During the 2004-2008 period, about 10 000 routers can be queried using this simple method.
Since this query is rather lightweight, we developped a script that collects data on the *mbone* everyday. This way we can obtain information on the local evolution of routers (frequency of configuration changes for example).
However, mrinfo-rec is limited and suffers from several drawbacks. To overcome such limitations, we deploy a new centralized platform called MERLIN, MEasure the Router Level of the INternet. More specifically, we develop a new mrinfo client tool (MERLIN client - NGI 2011), and a server (MERLIN server) whose aim is to pilot a set of well distributed vantage points running the MERLIN client. This approach allows to extend the mrinfo multicast vision and we are now able to collect more than 50 000 routers. Moreover, we design a hybrid tool based on our platform which allows one to target specifics routing domains. This extension of the MERLIN platform relies on *traceroute* and *ally* in order to expand the vision of targeted multicast backbone.

Projet results



- Quantifying ASes Multiconnectivity using Multicast Information, Pascal Mérindol, Virginie Van Der Schrieck, Benoit Donnet, Olivier Bonaventure, and Jean-Jacques Pansiot, In Proc. ACM Internet Measurement Conference, ACM press, pp 370-376 - november 2009

- Extracting Intra-Domain Topology from mrinfo Probing, Jean-Jacques Pansiot, Pascal Mérindol, Benoit Donnet, and Olivier Bonaventure, In Proc. Passive and Active Measurement Conference (PAM), Zurich, Springer, LNCS, pp 81-90, Vol. 6032, A. Kishnamurthy and B. Plattner - april 2010

- On the Impact of Layer-2 on Node Degree Distribution, Pascal Mérindol, Benoit Donnet, Olivier Bonaventure, and Jean-Jacques Pansiot, In Proc. ACM/USENIX Internet Measurement Conference (IMC), November 2010.

- MERLIN: MEasure the Router Level of the INternet , Pascal Mérindol, Benoit Donnet, Jean-Jacques Pansiot, Matthew Luckie, Young Huyn, In Proc. 7th Euro-nf Conference on Next Generation Internet (NGI), June 2011

IGMP probing offers several advantages: a native router level view, a L2 inferable vision (IMC2010) and it is no forwarding dependent (IMC2009). To better understand the characteristics of each map related to a given autonomous system (AS), we also develop a Router to AS mapping (R2AS - PAM2010): this technique allows us to mark the boundaries between ASes to investigate their intra- and inter- domain properties.

The code that you develop

- Licensing and credits
    - What are the rules imposed by your institution for releasing code ?
        - Is this allowed ?
        - Are there special declarations or IPR that you need to take into account ?
            - Check this before thinking about submitting paper artefacts
        - If you decide for open-source, understand the different types of licenses
            - Public domain, BSD/MIT, GPLv2, GPLv3, Apache, ...
    - Do you rely on existing code ?
        - In-house
            - Are you allowed to release the in-house code that you reused ?
        - External
            - What are the rules imposed by the code that you reuse ?

Experience with open-source licenses

- GPLv2
  - Used by our shim6, MPTCP and SRv6 implementations in the Linux kernel
    - Does not restrict the ability to create startups
  - If you implement kernel modules, try to merge your code upstream
- MIT License
  - Used by quic-go and inheritted by Multipath QUIC
  - Used by picoquic and inheritted by PQUIC
- Creative commons
  - Used for the Computer Networking: Principles, Protocols and Practice ebook
    - https://www.computer-networking.info

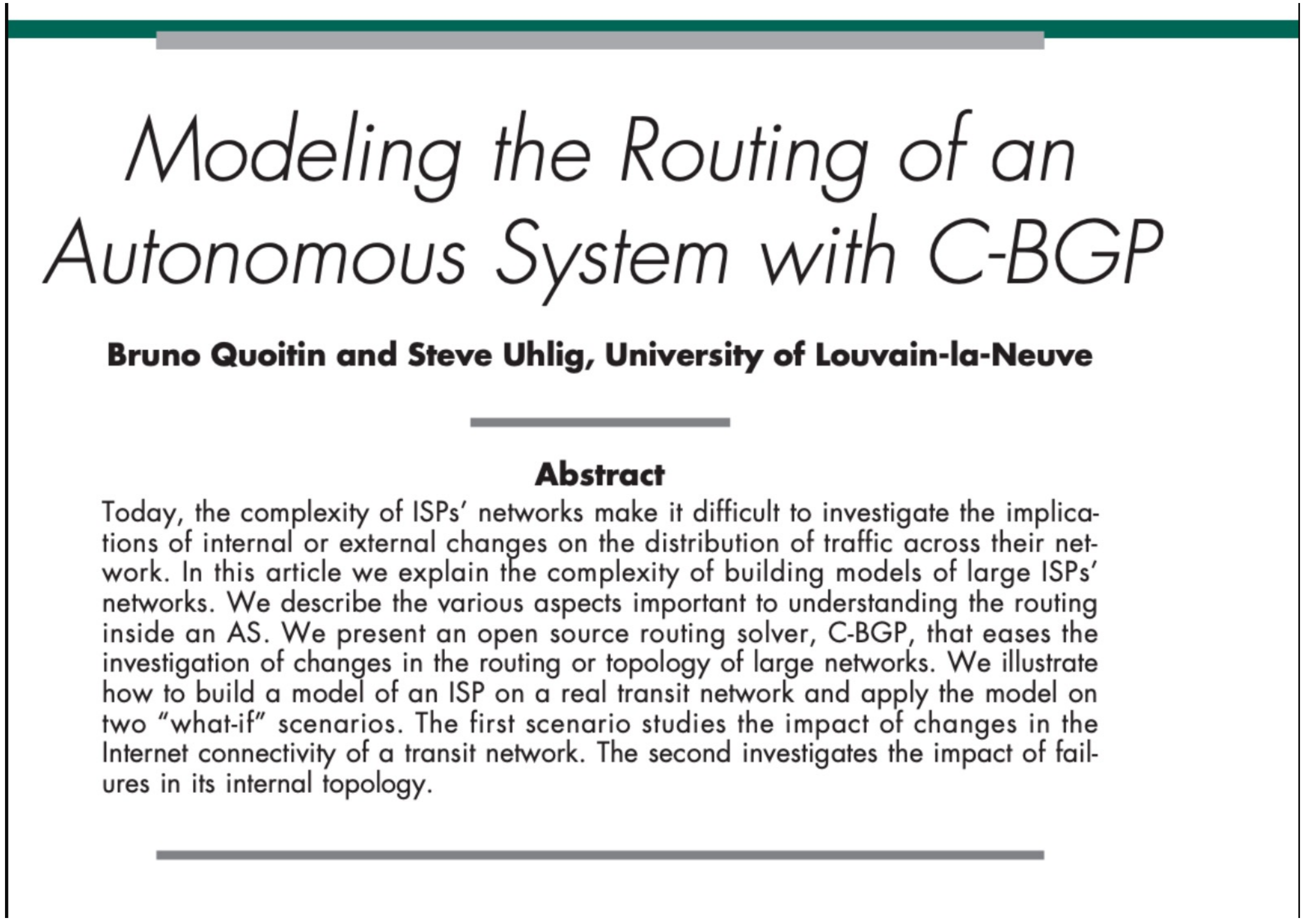
Simulations

- Your own special simulator
  - Plan to spend more time to validate your simulator than to execute your own measurements

- An existing (and well-maintained) simulator
  - Check papers that already used this simulator
  - Check how the different models need to be configured to obtain accurate results

C-BGP

- The project



Modeling the Routing of an Autonomous System with C-BGP

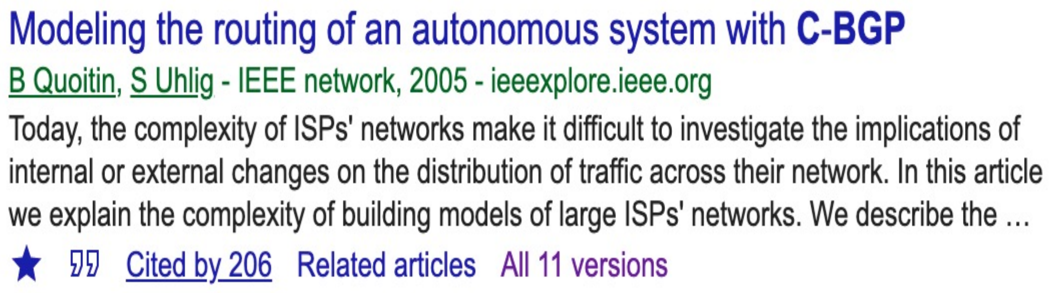
Bruno Quoitin and Steve Uhlig, University of Louvain-la-Neuve

Abstract

Today, the complexity of ISPs' networks make it difficult to investigate the implications of internal or external changes on the distribution of traffic across their network. In this article we explain the complexity of building models of large ISPs' networks. We describe the various aspects important to understanding the routing inside an AS. We present an open source routing solver, C-BGP, that eases the investigation of changes in the routing or topology of large networks. We illustrate how to build a model of an ISP on a real transit network and apply the model on two "what-if" scenarios. The first scenario studies the impact of changes in the Internet connectivity of a transit network. The second investigates the impact of failures in its internal topology.

Repeatability/Replicability/Reproducibility

- Repeatability
  - C-BGP was created and maintained by Bruno Quoitin
  - https://sourceforge.net/projects/c-bgp/
- Replicability
  - Various researchers were able to use the simulator for their own research, they were able to replicate simple examples
- Reproducibility
  - Nobody tried to re-implement a similar simulator. Since it was open-source and well documented, there was no incentive to reproduce such work
- Reusability



Modeling the routing of an autonomous system with **C-BGP**

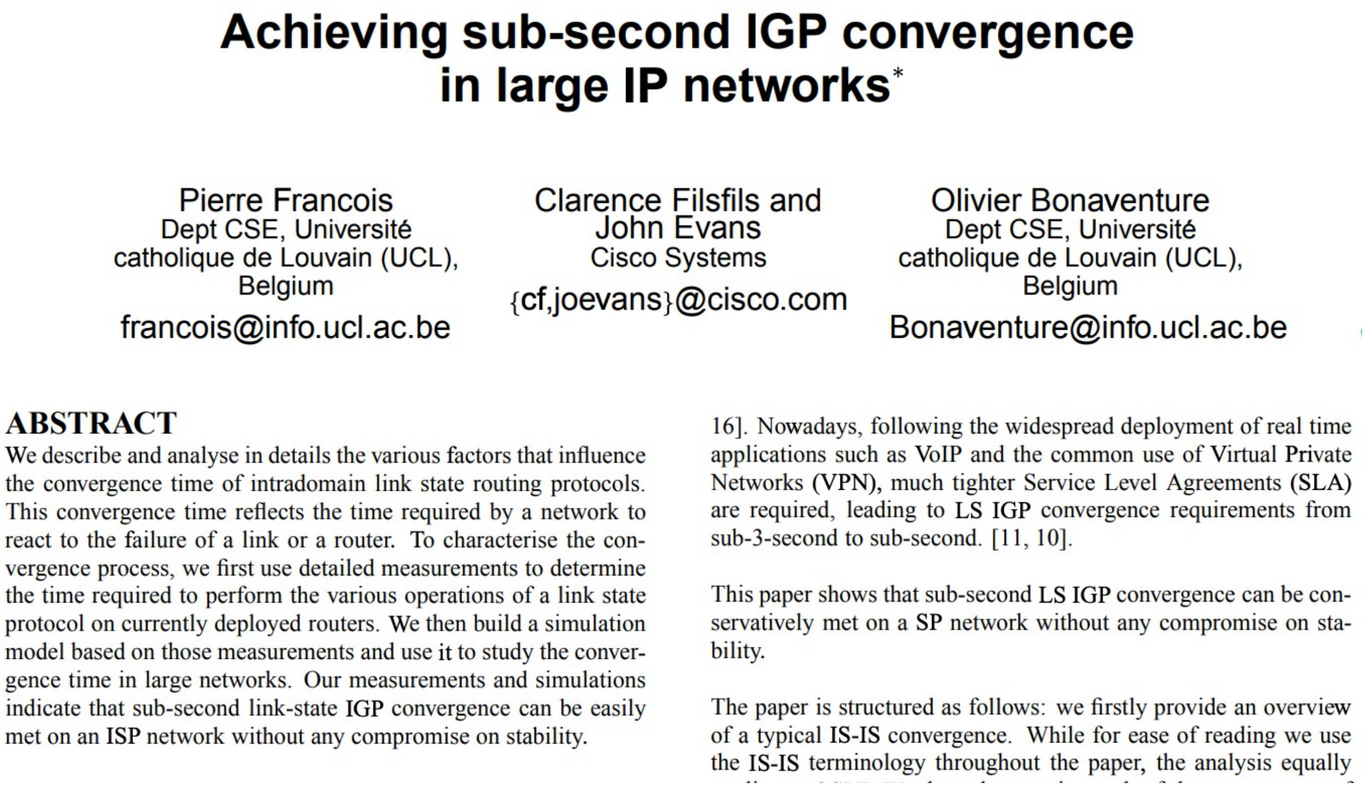
B Quoitin, S Uhlig - IEEE network, 2005 - ieeexplore.ieee.org

Today, the complexity of ISPs' networks make it difficult to investigate the implications of internal or external changes on the distribution of traffic across their network. In this article we explain the complexity of building models of large ISPs' networks. We describe the ...

★ ❞ Cited by 206 Related articles All 11 versions

Faster routing convergence

- The project



Achieving sub-second IGP convergence in large IP networks*

Pierre Francois
Dept CSE, Université catholique de Louvain (UCL), Belgium
francois@info.ucl.ac.be

Clarence Filsfils and John Evans
Cisco Systems
{cf,joevans}@cisco.com

Olivier Bonaventure
Dept CSE, Université catholique de Louvain (UCL), Belgium
Bonaventure@info.ucl.ac.be

ABSTRACT

We describe and analyse in details the various factors that influence the convergence time of intradomain link state routing protocols. This convergence time reflects the time required by a network to react to the failure of a link or a router. To characterise the convergence process, we first use detailed measurements to determine the time required to perform the various operations of a link state protocol on currently deployed routers. We then build a simulation model based on those measurements and use it to study the convergence time in large networks. Our measurements and simulations indicate that sub-second link-state IGP convergence can be easily met on an ISP network without any compromise on stability.

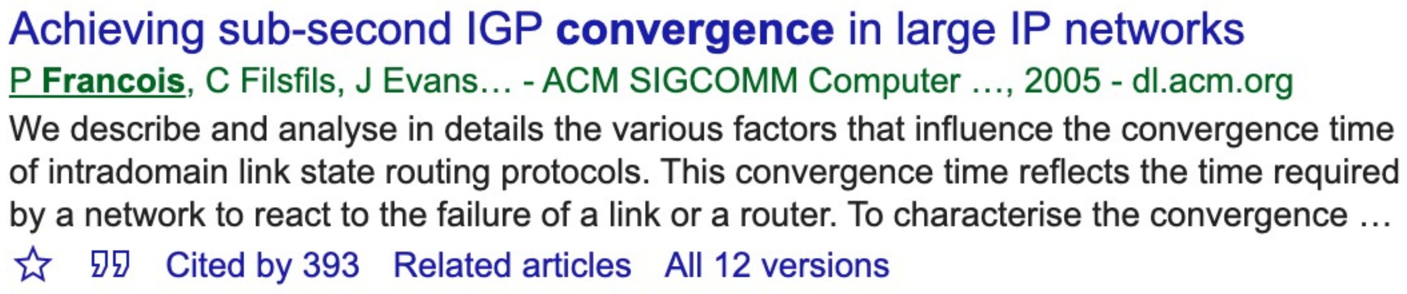
16]. Nowadays, following the widespread deployment of real time applications such as VoIP and the common use of Virtual Private Networks (VPN), much tighter Service Level Agreements (SLA) are required, leading to LS IGP convergence requirements from sub-3-second to sub-second. [11, 10].

This paper shows that sub-second LS IGP convergence can be conservatively met on a SP network without any compromise on stability.

The paper is structured as follows: we firstly provide an overview of a typical IS-IS convergence. While for ease of reading we use the IS-IS terminology throughout the paper, the analysis equally

Repeatability/Replicability/Reproducibility/Reusability

- Repeatability
  - Custom simulator written only for this paper
  - Private dataset (ISP topologies) provided by co-authors
- Replicability
  - Unfortunately, the custom simulator was never released
- Reproducibility
  - Although the paper has been widely cited, I'm not aware of any attempt to reproduce the paper results
- Reusability
  - Lots of citations,
  - but no real reuse



Achieving sub-second IGP **convergence** in large IP networks

P **Francois**, C Filsfils, J Evans… - ACM SIGCOMM Computer …, 2005 - dl.acm.org

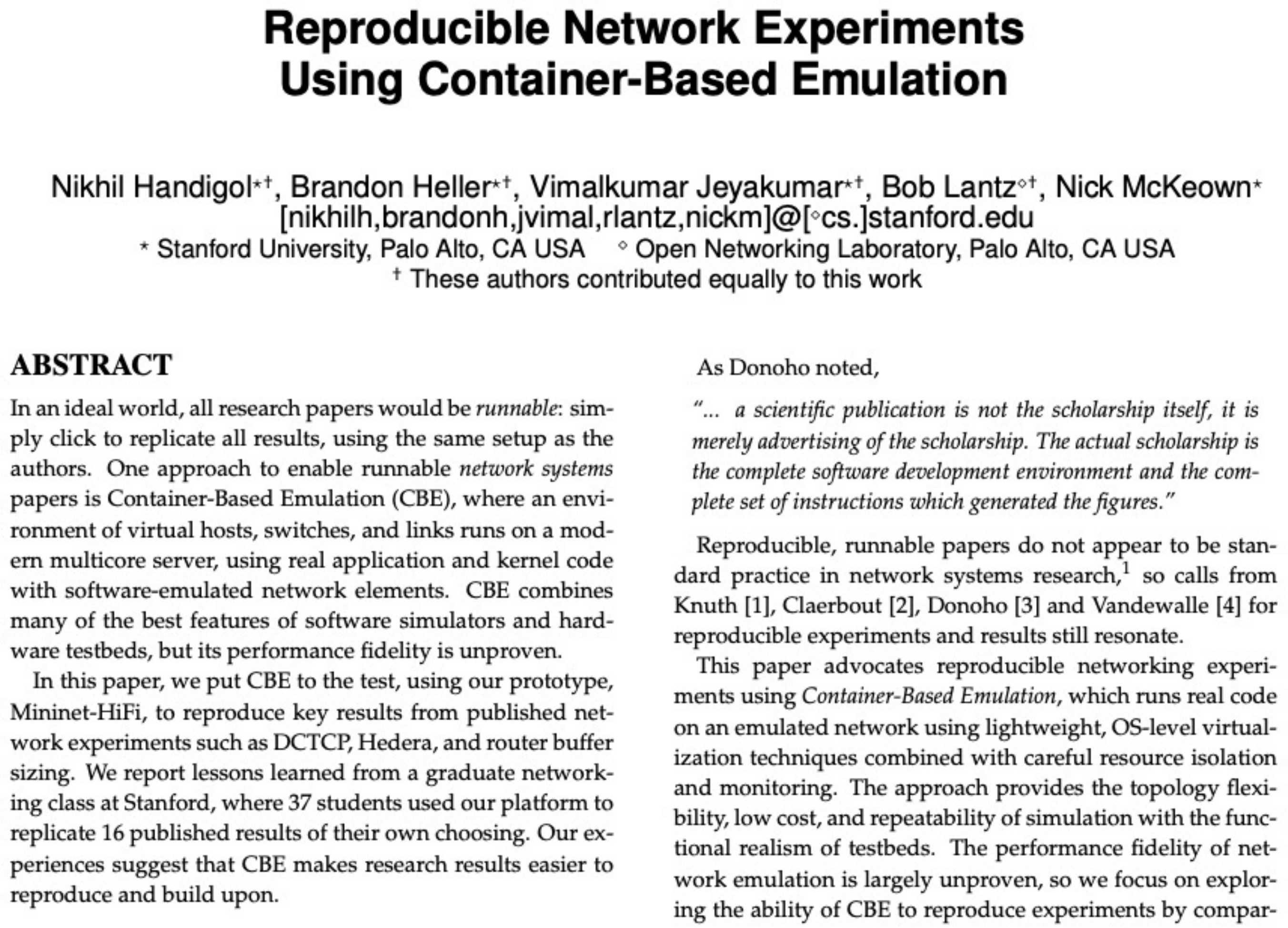
We describe and analyse in details the various factors that influence the convergence time of intradomain link state routing protocols. This convergence time reflects the time required by a network to react to the failure of a link or a router. To characterise the convergence …

☆ 99 Cited by 393 Related articles All 12 versions

Check list with existing simulators

- Is the simulator well-maintained ?
  - Can you get support and bug fixes
- Is the simulator open-source ?
  - Can you develop your own models and fix problems in its internals ?
- Is the simulator used in your research community ?
  - If not, develop your own tests and plan some space in your papers to convince the reviewers of the quality of the simulator
- Does it contain good models for *your* research problem ?
  - You should develop your model for the core of your research, but reuse existing models for the protocols that you do not change

Replacing simulation by emulation



Reproducible Network Experiments Using Container-Based Emulation

Nikhil Handigol⋆†, Brandon Heller⋆†, Vimalkumar Jeyakumar⋆†, Bob Lantz◇†, Nick McKeown⋆
[nikhilh,brandonh,jvimal,rlantz,nickm]@[◇cs.]stanford.edu
⋆ Stanford University, Palo Alto, CA USA ◇ Open Networking Laboratory, Palo Alto, CA USA
† These authors contributed equally to this work

ABSTRACT

In an ideal world, all research papers would be *runnable*: simply click to replicate all results, using the same setup as the authors. One approach to enable runnable *network systems* papers is Container-Based Emulation (CBE), where an environment of virtual hosts, switches, and links runs on a modern multicore server, using real application and kernel code with software-emulated network elements. CBE combines many of the best features of software simulators and hardware testbeds, but its performance fidelity is unproven.

In this paper, we put CBE to the test, using our prototype, Mininet-HiFi, to reproduce key results from published network experiments such as DCTCP, Hedera, and router buffer sizing. We report lessons learned from a graduate networking class at Stanford, where 37 students used our platform to replicate 16 published results of their own choosing. Our experiences suggest that CBE makes research results easier to reproduce and build upon.

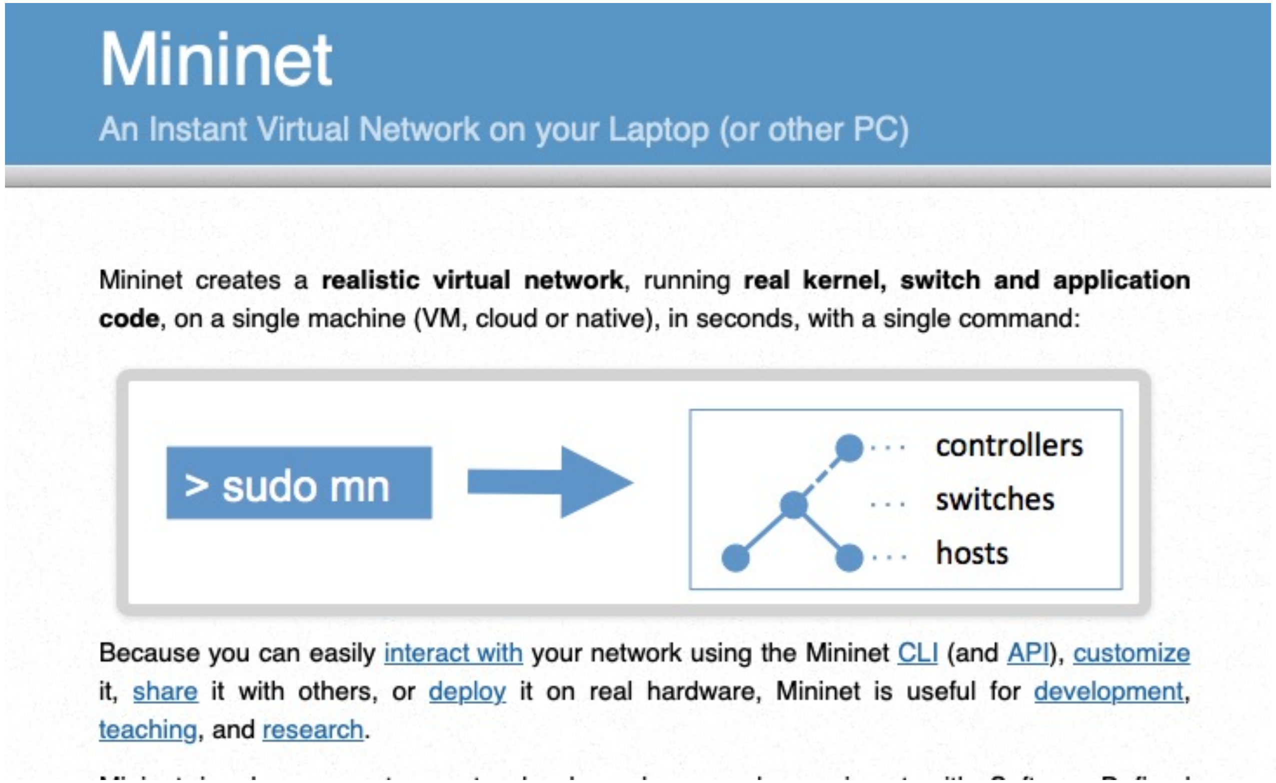
As Donoho noted,

> *"... a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."*

Reproducible, runnable papers do not appear to be standard practice in network systems research,[1] so calls from Knuth [1], Claerbout [2], Donoho [3] and Vandewalle [4] for reproducible experiments and results still resonate.

This paper advocates reproducible networking experiments using *Container-Based Emulation*, which runs real code on an emulated network using lightweight, OS-level virtualization techniques combined with careful resource isolation and monitoring. The approach provides the topology flexibility, low cost, and repeatability of simulation with the functional realism of testbeds. The performance fidelity of network emulation is largely unproven, so we focus on exploring the ability of CBE to reproduce experiments by compar-

Mininet...



Mininet

An Instant Virtual Network on your Laptop (or other PC)

Mininet creates a **realistic virtual network**, running **real kernel, switch and application code**, on a single machine (VM, cloud or native), in seconds, with a single command:

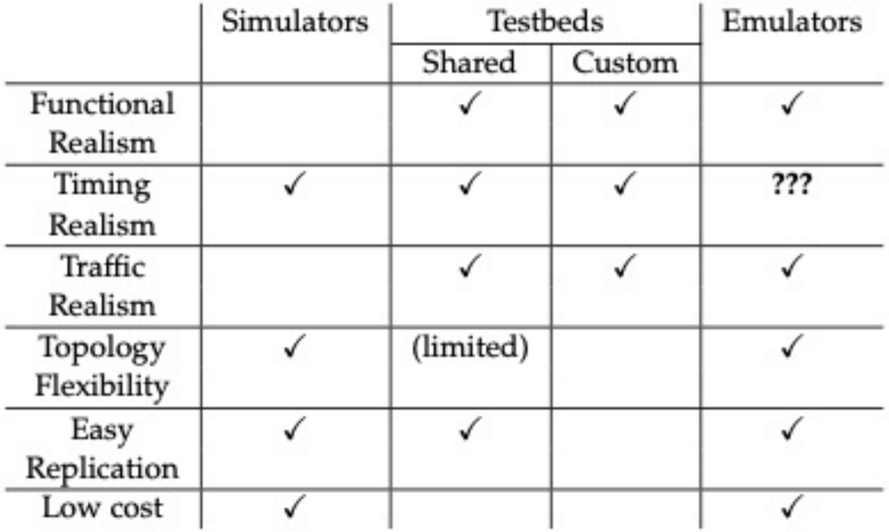
> sudo mn

controllers

switches

hosts

Because you can easily interact with your network using the Mininet CLI (and API), customize it, share it with others, or deploy it on real hardware, Mininet is useful for development, teaching, and research.



	Simulators	Testbeds		Emulators
		Shared	Custom	
Functional Realism		✓	✓	✓
Timing Realism	✓	✓	✓	???
Traffic Realism		✓	✓	✓
Topology Flexibility	✓	(limited)		✓
Easy Replication	✓	✓		✓
Low cost	✓			✓

Table 1: Platform characteristics for reproducible network experiments.

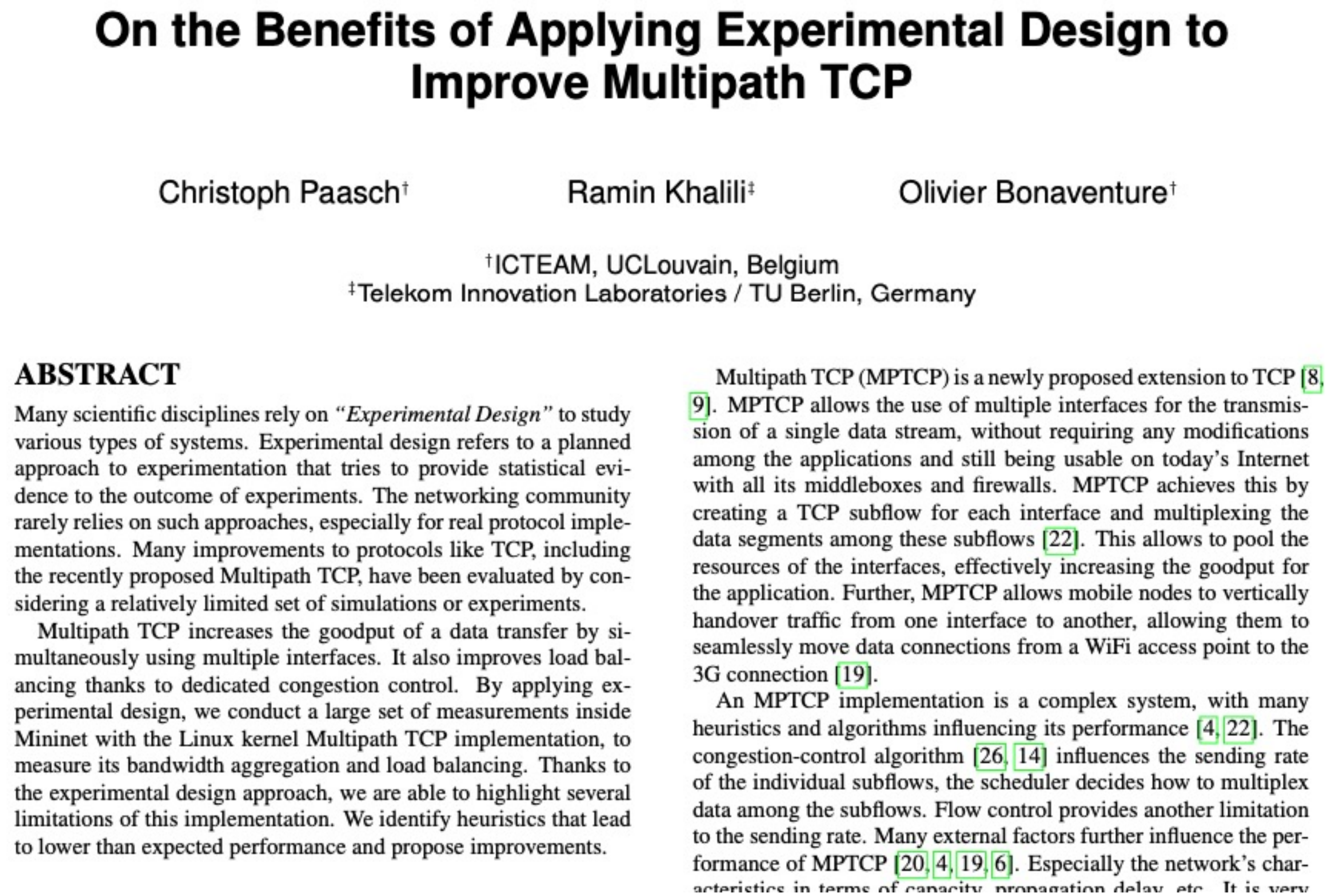
Mininet...

- The good
  - You can easily model a small network on your laptop and execute your own code on routers and servers
    - This is real code and not a simplified model, all details are considered
- The bad
  - Everything (routers, servers, applications) compete for the CPU and memory on your laptop
    - If CPU load is too high, you will get random results
    - Don’t run other applications on your laptop while mininet is running
- The ugly
  - Different runs of the same experiment can produce very different results
    - repetition can help, but in some cases, you will need to scale down

Our experience with Mininet

- Advantages
  - Works well for rapid prototyping and to test simple scenarios
  - Allows to experiment with patched Linux kernels with limited efforts
  - Nice teaching tool, check ipmininet
- Drawbacks
  - Works well for low bandwidth flows that are not CPU bound
  - Only works for very small networks on our computers
  - Be careful with workloads that are CPU bound

Experimental design with Mininet



On the Benefits of Applying Experimental Design to Improve Multipath TCP

Christoph Paasch[†] Ramin Khalili[‡] Olivier Bonaventure[†]

[†]ICTEAM, UCLouvain, Belgium
[‡]Telekom Innovation Laboratories / TU Berlin, Germany

ABSTRACT

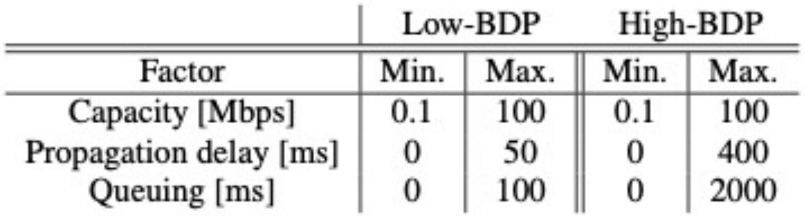
Many scientific disciplines rely on *"Experimental Design"* to study various types of systems. Experimental design refers to a planned approach to experimentation that tries to provide statistical evidence to the outcome of experiments. The networking community rarely relies on such approaches, especially for real protocol implementations. Many improvements to protocols like TCP, including the recently proposed Multipath TCP, have been evaluated by considering a relatively limited set of simulations or experiments.

Multipath TCP increases the goodput of a data transfer by simultaneously using multiple interfaces. It also improves load balancing thanks to dedicated congestion control. By applying experimental design, we conduct a large set of measurements inside Mininet with the Linux kernel Multipath TCP implementation, to measure its bandwidth aggregation and load balancing. Thanks to the experimental design approach, we are able to highlight several limitations of this implementation. We identify heuristics that lead to lower than expected performance and propose improvements.

Multipath TCP (MPTCP) is a newly proposed extension to TCP [8, 9]. MPTCP allows the use of multiple interfaces for the transmission of a single data stream, without requiring any modifications among the applications and still being usable on today's Internet with all its middleboxes and firewalls. MPTCP achieves this by creating a TCP subflow for each interface and multiplexing the data segments among these subflows [22]. This allows to pool the resources of the interfaces, effectively increasing the goodput for the application. Further, MPTCP allows mobile nodes to vertically handover traffic from one interface to another, allowing them to seamlessly move data connections from a WiFi access point to the 3G connection [19].

An MPTCP implementation is a complex system, with many heuristics and algorithms influencing its performance [4, 22]. The congestion-control algorithm [26, 14] influences the sending rate of the individual subflows, the scheduler decides how to multiplex data among the subflows. Flow control provides another limitation to the sending rate. Many external factors further influence the performance of MPTCP [20, 4, 19, 6]. Especially the network's characteristics in terms of capacity, propagation delay, etc. It is very

The experiments



	Low-BDP		High-BDP	
Factor	Min.	Max.	Min.	Max.
Capacity [Mbps]	0.1	100	0.1	100
Propagation delay [ms]	0	50	0	400
Queuing [ms]	0	100	0	2000

Table 1: Domains of the influencing factors for the measurement of aggregation benefit.

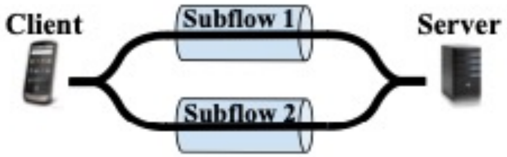
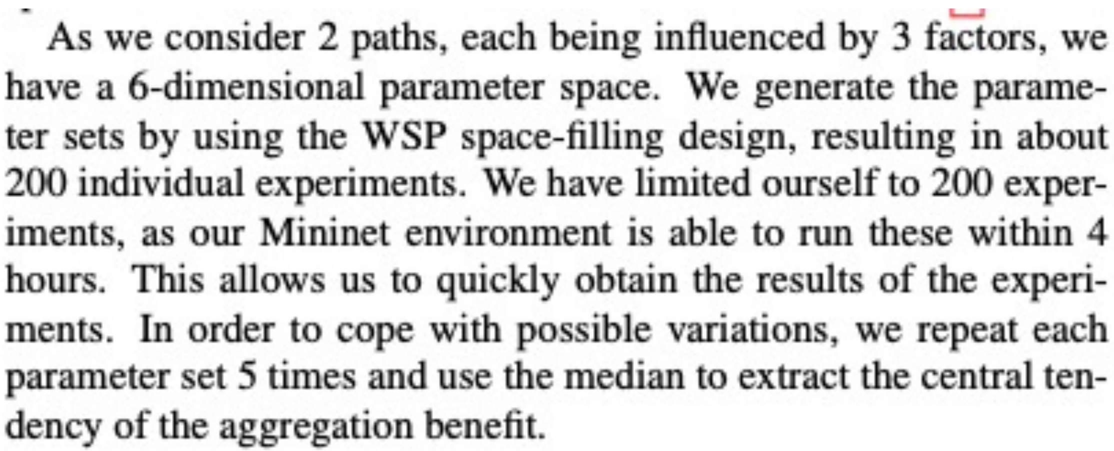


Figure 2: Our topology to evaluate aggregation benefit: MPTCP creates one subflow across each bottleneck. This allows to evaluate multipath-scenarios like a mobile phone connecting to WiFi/3G at the same time.



As we consider 2 paths, each being influenced by 3 factors, we have a 6-dimensional parameter space. We generate the parameter sets by using the WSP space-filling design, resulting in about 200 individual experiments. We have limited ourself to 200 experiments, as our Mininet environment is able to run these within 4 hours. This allows us to quickly obtain the results of the experiments. In order to cope with possible variations, we repeat each parameter set 5 times and use the median to extract the central tendency of the aggregation benefit.

The results

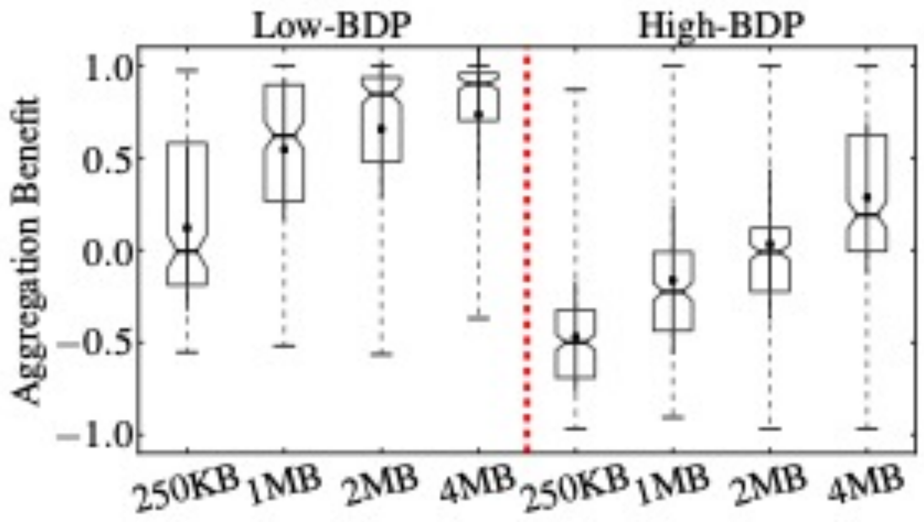


Figure 3: The aggregation benefit increases with the buffer size. If the buffer is small, MPTCP is limited by the receive window and does not fully utilizes the network's capacity.

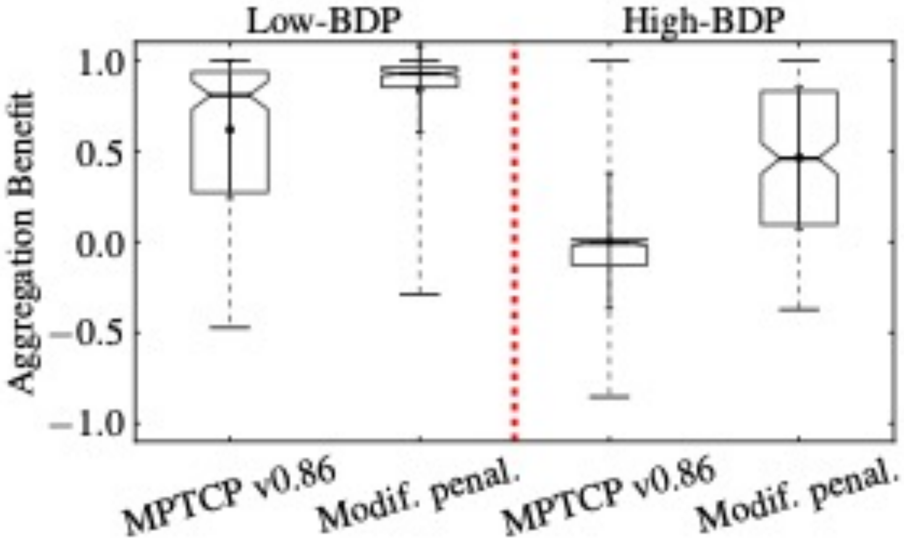
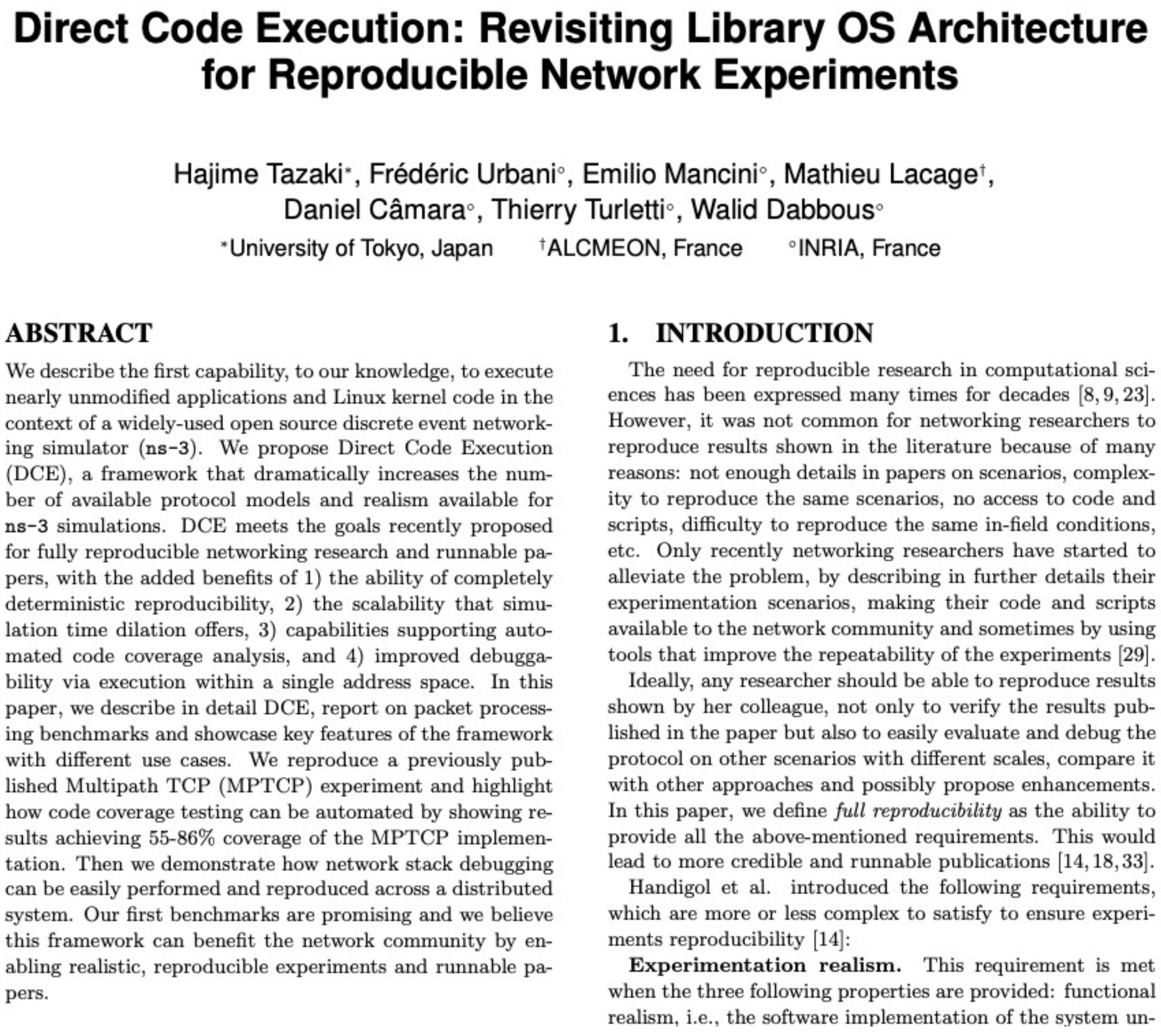


Figure 5: MPTCP v0.86 has a weak performance with auto-tuning. Our modification to the penalization algorithm significantly improves the aggregation benefit.

ns3-DCE : best of both worlds ?



Direct Code Execution: Revisiting Library OS Architecture for Reproducible Network Experiments

Hajime Tazaki[*], Frédéric Urbani[◦], Emilio Mancini[◦], Mathieu Lacage[†],
Daniel Câmara[◦], Thierry Turletti[◦], Walid Dabbous[◦]
[*]University of Tokyo, Japan [†]ALCMEON, France [◦]INRIA, France

ABSTRACT

We describe the first capability, to our knowledge, to execute nearly unmodified applications and Linux kernel code in the context of a widely-used open source discrete event networking simulator (**ns-3**). We propose Direct Code Execution (DCE), a framework that dramatically increases the number of available protocol models and realism available for **ns-3** simulations. DCE meets the goals recently proposed for fully reproducible networking research and runnable papers, with the added benefits of 1) the ability of completely deterministic reproducibility, 2) the scalability that simulation time dilation offers, 3) capabilities supporting automated code coverage analysis, and 4) improved debuggability via execution within a single address space. In this paper, we describe in detail DCE, report on packet processing benchmarks and showcase key features of the framework with different use cases. We reproduce a previously published Multipath TCP (MPTCP) experiment and highlight how code coverage testing can be automated by showing results achieving 55-86% coverage of the MPTCP implementation. Then we demonstrate how network stack debugging can be easily performed and reproduced across a distributed system. Our first benchmarks are promising and we believe this framework can benefit the network community by enabling realistic, reproducible experiments and runnable papers.

1. INTRODUCTION

The need for reproducible research in computational sciences has been expressed many times for decades [8, 9, 23]. However, it was not common for networking researchers to reproduce results shown in the literature because of many reasons: not enough details in papers on scenarios, complexity to reproduce the same scenarios, no access to code and scripts, difficulty to reproduce the same in-field conditions, etc. Only recently networking researchers have started to alleviate the problem, by describing in further details their experimentation scenarios, making their code and scripts available to the network community and sometimes by using tools that improve the repeatability of the experiments [29].

Ideally, any researcher should be able to reproduce results shown by her colleague, not only to verify the results published in the paper but also to easily evaluate and debug the protocol on other scenarios with different scales, compare it with other approaches and possibly propose enhancements. In this paper, we define *full reproducibility* as the ability to provide all the above-mentioned requirements. This would lead to more credible and runnable publications [14, 18, 33].

Handigol et al. introduced the following requirements, which are more or less complex to satisfy to ensure experiments reproducibility [14]:

Experimentation realism. This requirement is met when the three following properties are provided: functional realism, i.e., the software implementation of the system un-

ns3-DCE…

- The good
  - ns-3 is a simulator and thus your application can simulate **very large networks** if you have enough RAM/CPU
    - Time is simulated and it usually runs much slower than real-time
  - Simulation is fully reproducible
    - if you find a bug in your code, you can fix it and then rerun exactly this experiment
- The bad
  - unless you specify it, your application executes in zero time
    - this ignores CPU consumption effects which could be important in some scenarios
- The ugly
  - In some cases porting applications (or worse kernel code) to ns3-DCE can be a tough software engineering challenge

Our experience with ns3-DCE

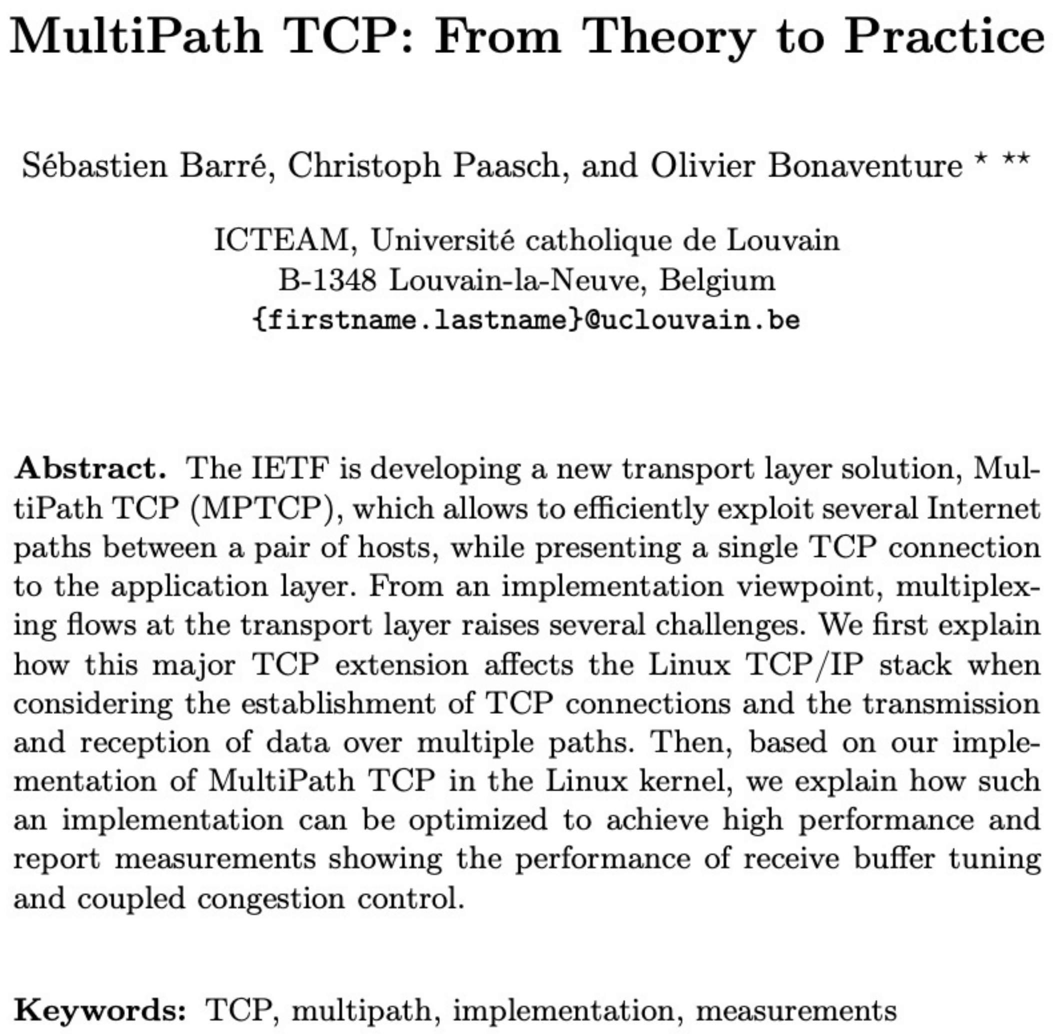
- Advantages
  - Works well with QUIC implementations in userspace
  - Used it extensively as a test suite to avoid regressions in PQUIC
    - Since ns3-DCE is deterministic, once you've found a scenario with problems in retransmissions or congestion control, you can check that your fixes solve them and do not create new ones

- Drawbacks
  - Requires very good systems knowledge to port existing applications
    - plan to spend some time to get upto speed
  - Not widely used by the networking community

Real experiments

- The good
  - You execute your entire code on real machines
    - Your code needs to handle most corner cases and works well
- The bad
  - Getting the required hardware ressources can be costly or difficult
    - If you use shared machines, be careful of interferences
      - planetlab, grid5000, ...
- The ugly
  - Beware of overoptimizing your solution to your hardware setup
    - It might be much smaller or very different from deployments

Multipath TCP



MultiPath TCP: From Theory to Practice

Sébastien Barré, Christoph Paasch, and Olivier Bonaventure * **

ICTEAM, Université catholique de Louvain
B-1348 Louvain-la-Neuve, Belgium
{firstname.lastname}@uclouvain.be

Abstract. The IETF is developing a new transport layer solution, MultiPath TCP (MPTCP), which allows to efficiently exploit several Internet paths between a pair of hosts, while presenting a single TCP connection to the application layer. From an implementation viewpoint, multiplexing flows at the transport layer raises several challenges. We first explain how this major TCP extension affects the Linux TCP/IP stack when considering the establishment of TCP connections and the transmission and reception of data over multiple paths. Then, based on our implementation of MultiPath TCP in the Linux kernel, we explain how such an implementation can be optimized to achieve high performance and report measurements showing the performance of receive buffer tuning and coupled congestion control.

Keywords: TCP, multipath, implementation, measurements

First experimental results

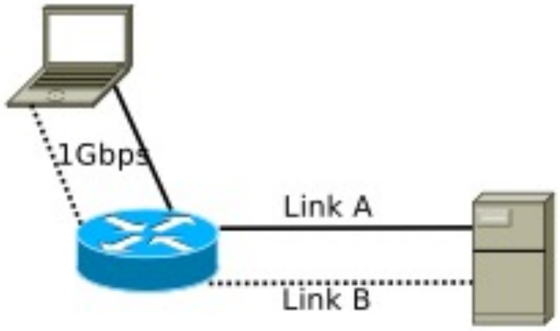


Fig. 3. Performance testbed

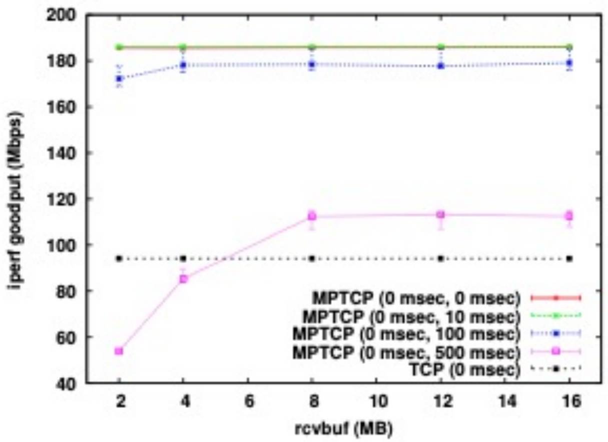


Fig. 6. Impact of the maximum receive buffer size

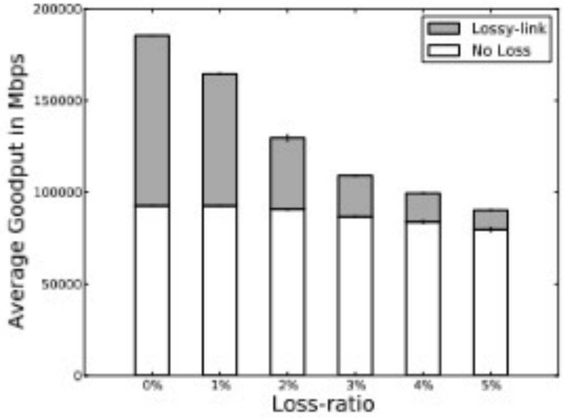
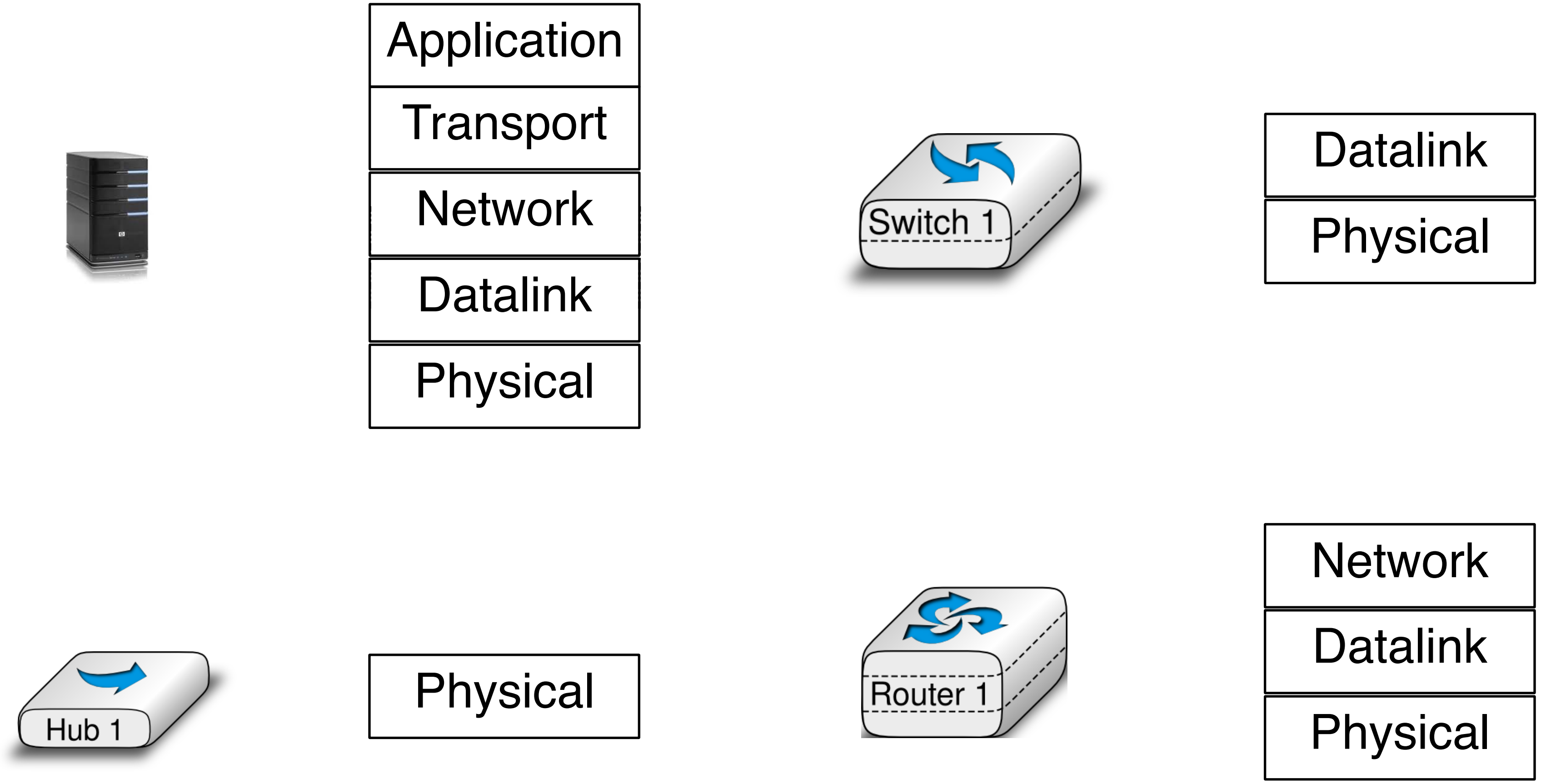


Fig. 7. Impact of the packet loss ratio

Benefits of trying to replicate MPTCP results

- What about WAN measurements ?
    - Colleagues in Finland installed MPTCP kernel to replicate our results
        - Worked well in their lab
    - MPTCP connections could be established between Finland and Belgium, but no data was ever transferred

The Internet architecture that we explain to our students



Application
Transport
Network
Datalink
Physical

Datalink
Physical

Physical

Network
Datalink
Physical

O. Bonaventure, Computer networking : Principles, Protocols and Practice, open ebook, http://inl.info.ucl.ac.be/cnp3

In reality

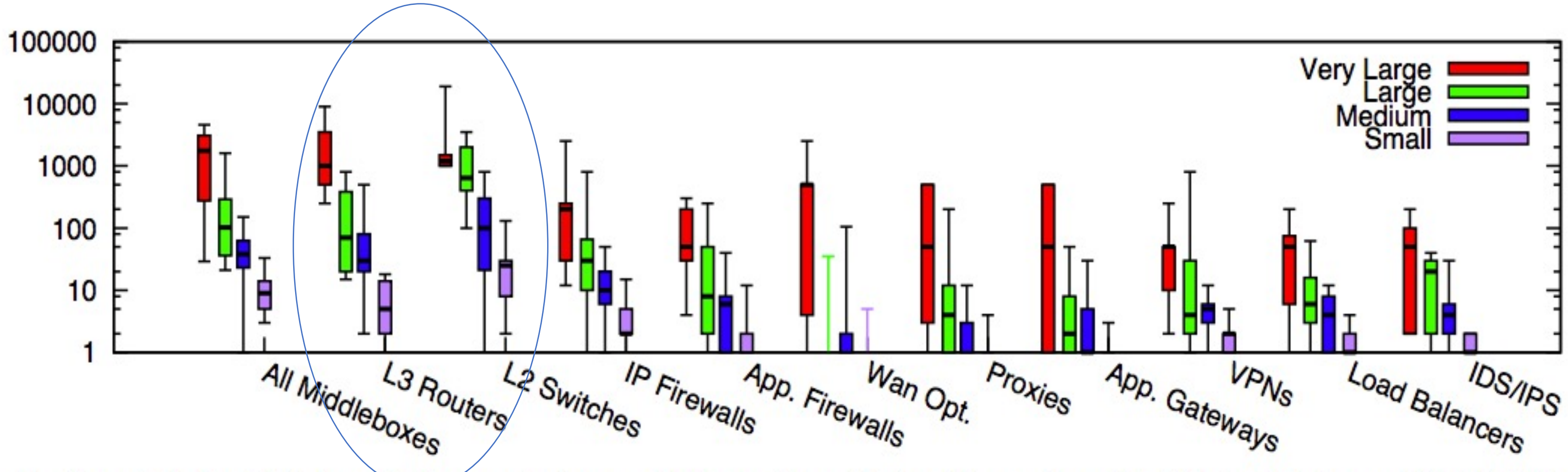
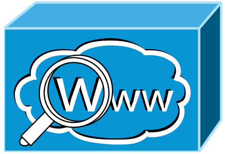


Figure 1: Box plot of middlebox deployments for small (fewer than 1k hosts), medium (1k-10k hosts), large (10k-100k hosts), and very large (more than 100k hosts) enterprise networks. Y-axis is in log scale.

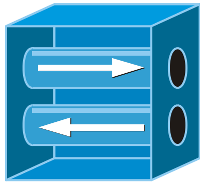
- almost as many middleboxes as routers
- various types of middleboxes are deployed

Sherry, Justine, et al. "*Making middleboxes someone else's problem: Network processing as a cloud service*." Proceedings of the ACM SIGCOMM 2012 conference. ACM, 2012.

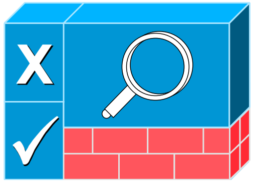
Internet devices according to Cisco



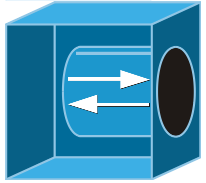
Web Security Appliance



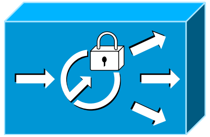
VPN Concentrator



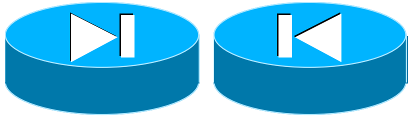
NAC Appliance



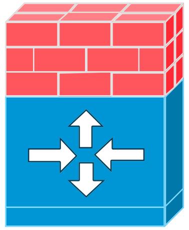
SSL Terminator



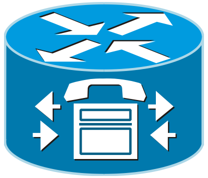
ACE XML Gateway



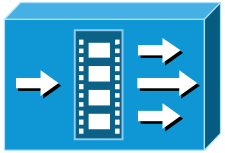
PIX Firewall Right and Left



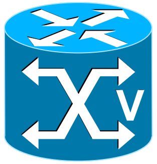
Cisco IOS Firewall



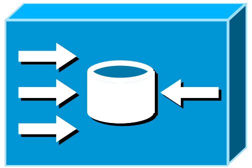
IP Telephony Router



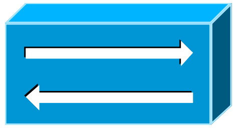
Streamer



Voice Gateway



Content Engine

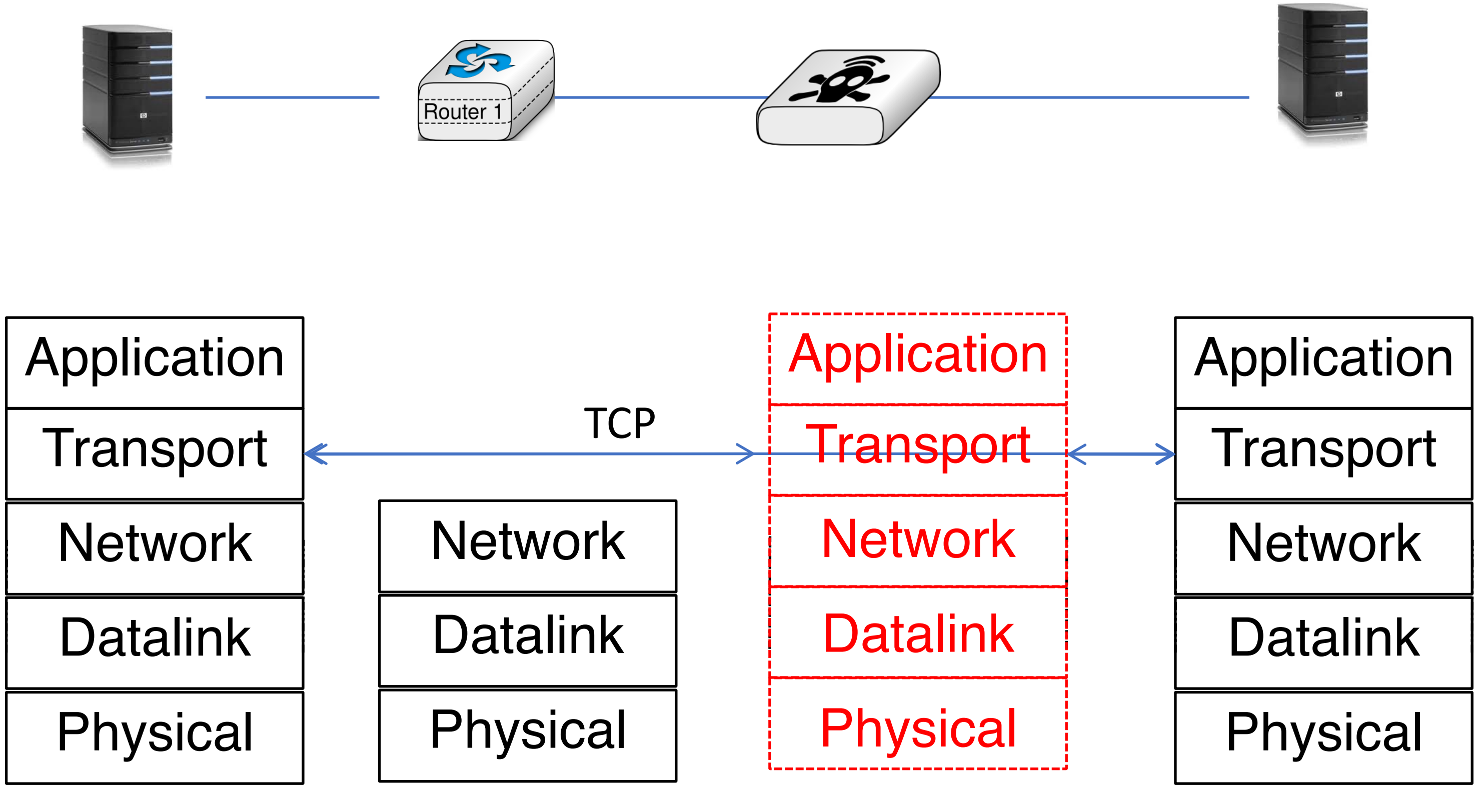


NAT

http://www.cisco.com/web/about/ac50/ac47/2.html

Middleboxes in the architecture

- In the official architecture, they do not exist
- In reality...



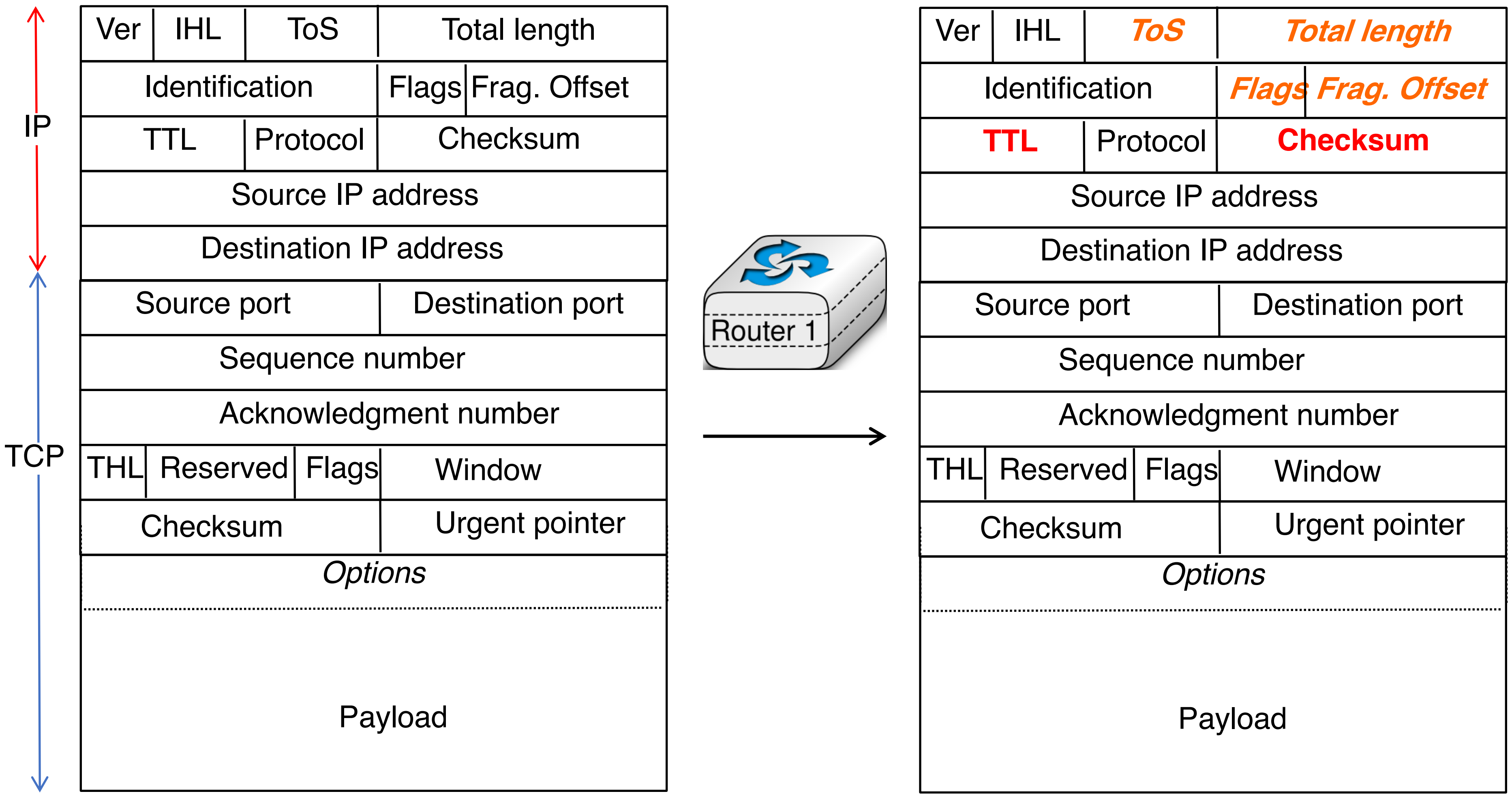
Application
Transport
Network
Datalink
Physical

Network
Datalink
Physical

Application
Transport
Network
Datalink
Physical

Application
Transport
Network
Datalink
Physical

TCP segments processed by a router



IP

TCP

Ver | IHL | ToS | Total length

Identification | Flags | Frag. Offset

TTL | Protocol | Checksum

Source IP address

Destination IP address

Source port | Destination port

Sequence number

Acknowledgment number

THL | Reserved | Flags | Window

Checksum | Urgent pointer

Options

Payload

Ver | IHL | ***ToS*** | ***Total length***

Identification | ***Flags*** | ***Frag. Offset***

TTL | Protocol | **Checksum**

Source IP address

Destination IP address

Source port | Destination port

Sequence number

Acknowledgment number

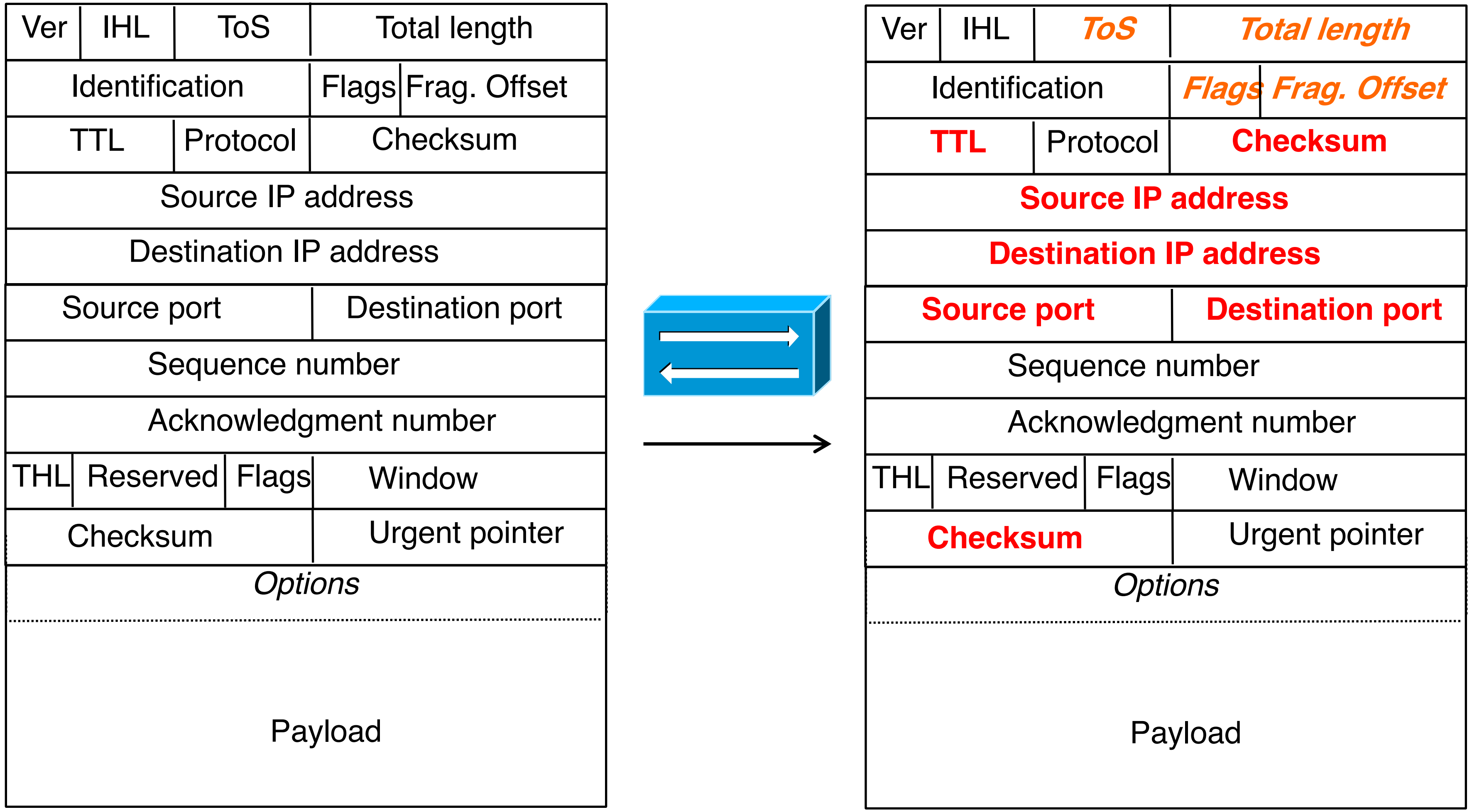
THL | Reserved | Flags | Window

Checksum | Urgent pointer

Options

Payload

TCP segments processed by a NAT



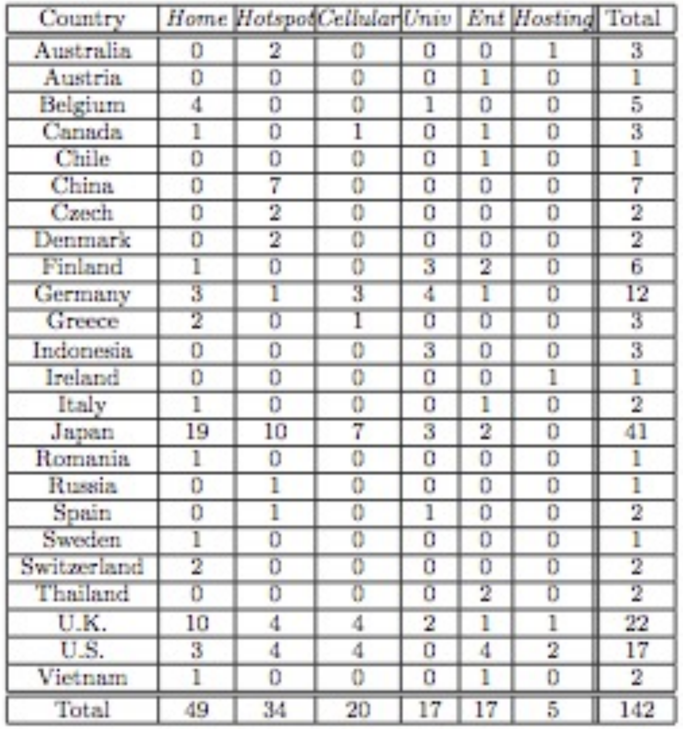
Ver	IHL	ToS	Total length
Identification	Flags	Frag. Offset	
TTL	Protocol	Checksum	
Source IP address			
Destination IP address			
Source port	Destination port		
Sequence number			
Acknowledgment number			
THL	Reserved	Flags	Window
Checksum	Urgent pointer		
Options			
Payload			

Ver	IHL	***ToS***	***Total length***
Identification	***Flags***	***Frag. Offset***	
TTL	Protocol	**Checksum**	
Source IP address			
Destination IP address			
Source port	**Destination port**		
Sequence number			
Acknowledgment number			
THL	Reserved	Flags	Window
Checksum	Urgent pointer		
Options			
Payload			

How transparent is the Internet ?

- 25th September 2010 to 30th April 2011
- 142 access networks
- 24 countries
- Sent specific TCP segments from client to a server in Japan

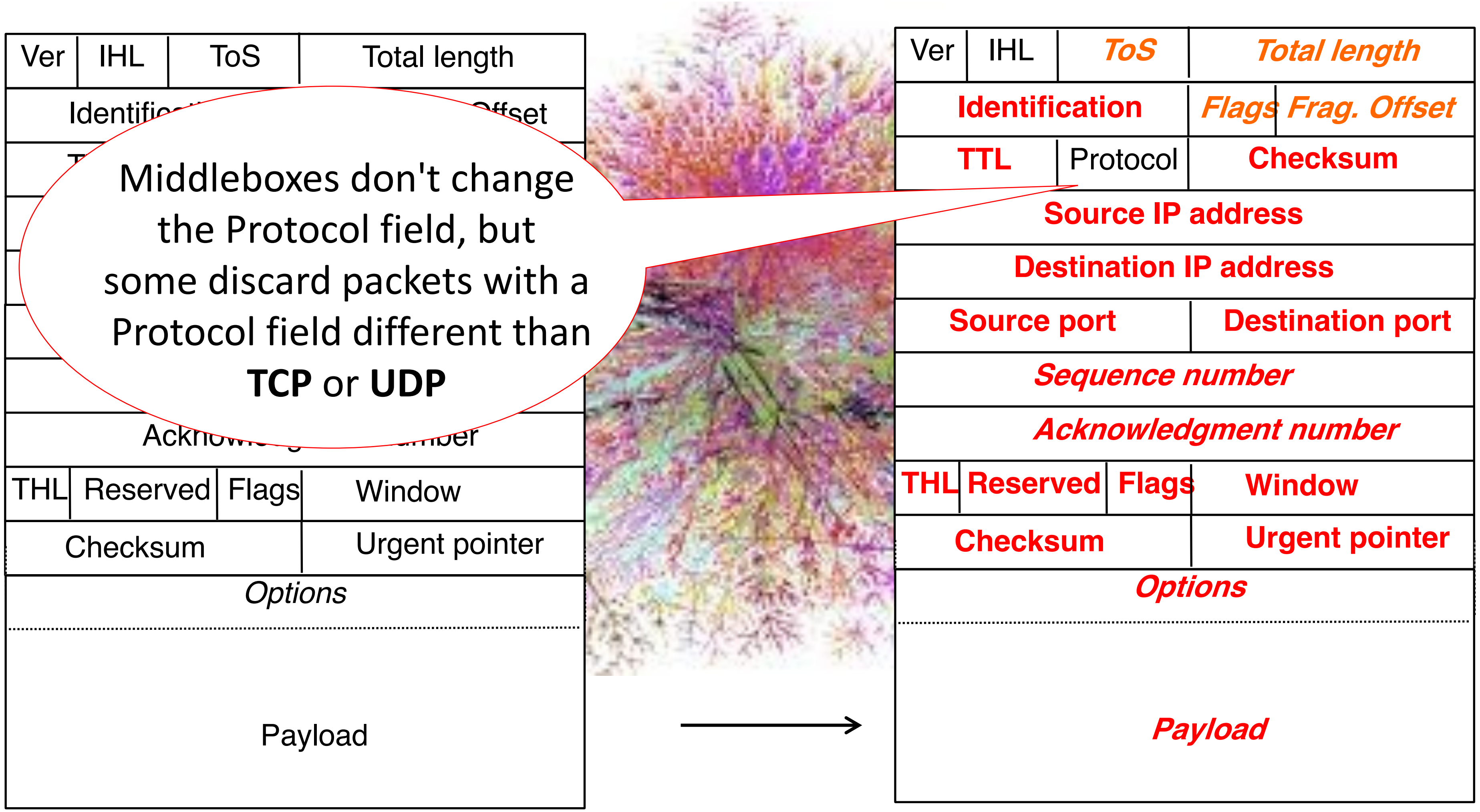
Table 2: Experiment Venues



Country	*Home*	*Hotspot*	*Cellular*	*Univ*	*Ent*	*Hosting*	Total
Australia	0	2	0	0	0	1	3
Austria	0	0	0	0	1	0	1
Belgium	4	0	0	1	0	0	5
Canada	1	0	1	0	1	0	3
Chile	0	0	0	0	1	0	1
China	0	7	0	0	0	0	7
Czech	0	2	0	0	0	0	2
Denmark	0	2	0	0	0	0	2
Finland	1	0	0	3	2	0	6
Germany	3	1	3	4	1	0	12
Greece	2	0	1	0	0	0	3
Indonesia	0	0	0	3	0	0	3
Ireland	0	0	0	0	0	1	1
Italy	1	0	0	0	1	0	2
Japan	19	10	7	3	2	0	41
Romania	1	0	0	0	0	0	1
Russia	0	1	0	0	0	0	1
Spain	0	1	0	1	0	0	2
Sweden	1	0	0	0	0	0	1
Switzerland	2	0	0	0	0	0	2
Thailand	0	0	0	0	2	0	2
U.K.	10	4	4	2	1	1	22
U.S.	3	4	4	0	4	2	17
Vietnam	1	0	0	0	1	0	2
Total	49	34	20	17	17	5	142

Honda, Michio, et al. "*Is it still possible to extend TCP?*" Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, 2011.

End-to-end transparency today



Middleboxes don't change the Protocol field, but some discard packets with a Protocol field different than **TCP** or **UDP**

Ver | IHL | ToS | Total length

Identific… Offset

Ackno… umber

THL | Reserved | Flags | Window

Checksum | Urgent pointer

Options

Payload

Ver | IHL | *ToS* | *Total length*

Identification | *Flags* | *Frag. Offset*

TTL | Protocol | **Checksum**

Source IP address

Destination IP address

Source port | **Destination port**

Sequence number

Acknowledgment number

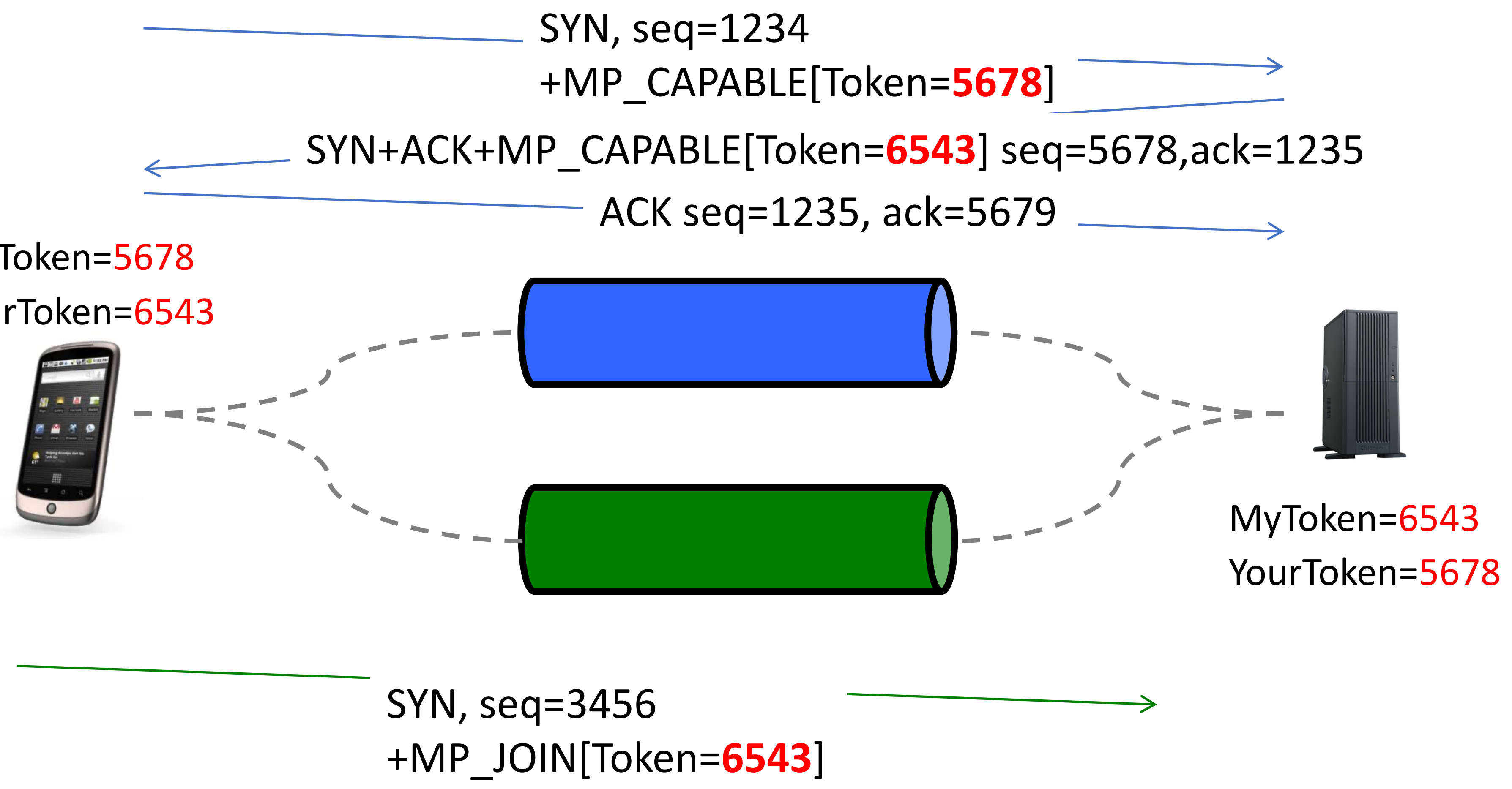
THL | **Reserved** | **Flags** | **Window**

Checksum | **Urgent pointer**

Options

Payload

How to create an MPTCP connection ?



SYN, seq=1234
+MP_CAPABLE[Token=**5678**]

SYN+ACK+MP_CAPABLE[Token=**6543**] seq=5678,ack=1235

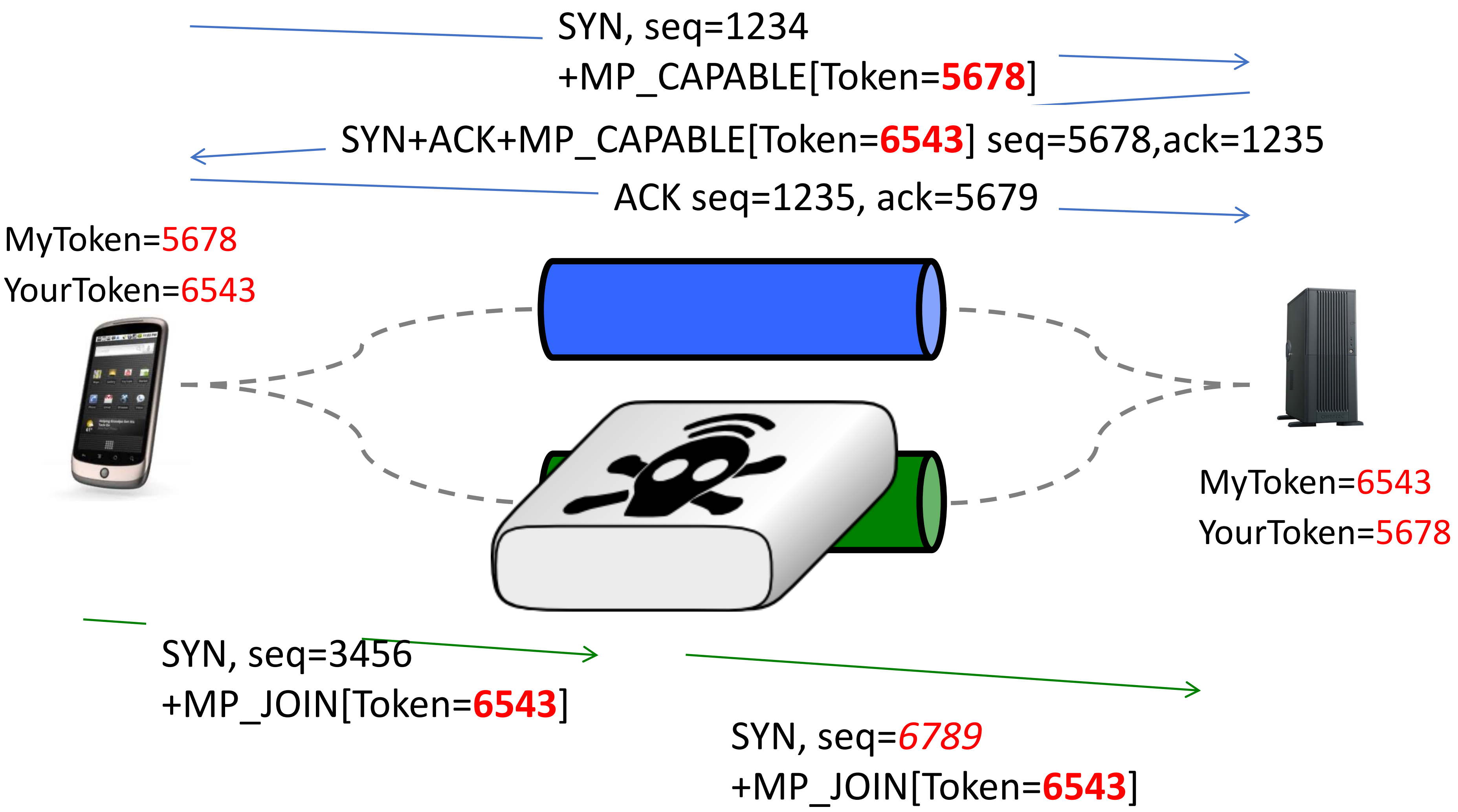
ACK seq=1235, ack=5679

MyToken=5678
YourToken=6543

MyToken=6543
YourToken=5678

SYN, seq=3456
+MP_JOIN[Token=**6543**]

How to create an MPTCP connection ?



SYN, seq=1234
+MP_CAPABLE[Token=**5678**]

SYN+ACK+MP_CAPABLE[Token=**6543**] seq=5678,ack=1235

ACK seq=1235, ack=5679

MyToken=5678
YourToken=6543

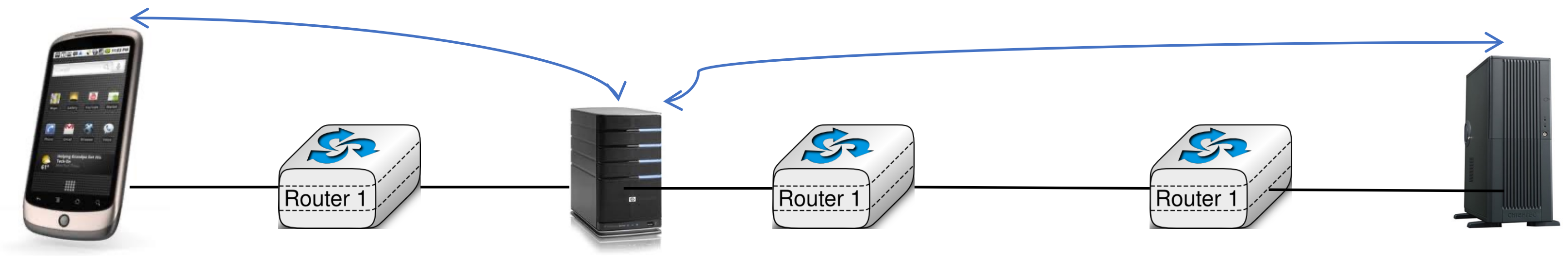
MyToken=6543
YourToken=5678

SYN, seq=3456
+MP_JOIN[Token=**6543**]

SYN, seq=*6789*
+MP_JOIN[Token=**6543**]

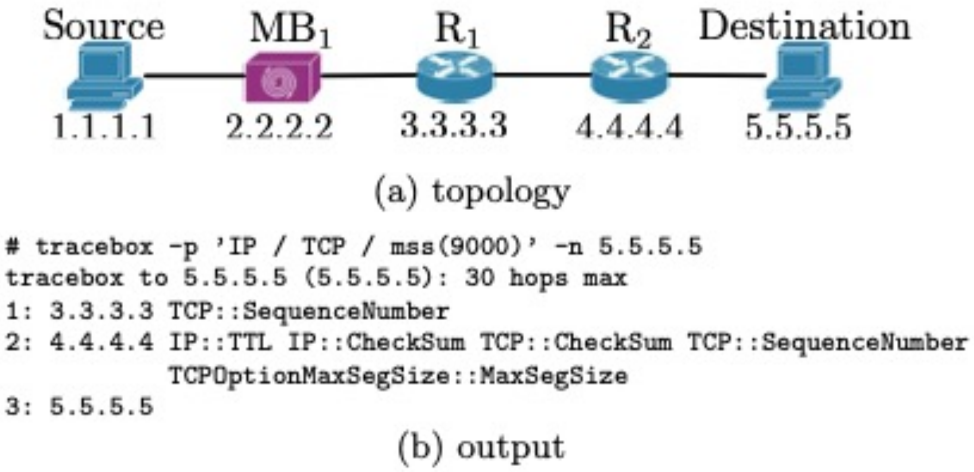
Which middleboxes change TCP sequence numbers ?

- Some firewalls change TCP sequence numbers in SYN segments to ensure randomness
  - fix for old windows95 bug
- Transparent proxies terminate TCP connections
  - Widely used in cellular networks and in some enterprises



Consequences of this replicability study

- Multipath TCP was redesigned to cope with all these middleboxes
  - Multipath TCP works through most middleboxesand falls back to TCP when there is severe interference to preserve connectivity
- Michio Honda et al. published the measurement study *Is it possible to extend TCP?*
- We developed `tracebox`, a traceroute like that detects middlebox interference



(a) topology

```
# tracebox -p 'IP / TCP / mss(9000)' -n 5.5.5.5
tracebox to 5.5.5.5 (5.5.5.5): 30 hops max
1: 3.3.3.3 TCP::SequenceNumber
2: 4.4.4.4 IP::TTL IP::CheckSum TCP::CheckSum TCP::SequenceNumber
           TCPOptionMaxSegSize::MaxSegSize
3: 5.5.5.5
```

(b) output

Next steps with Multipath TCP

- Repeatability
    - Linux kernel code, moved to http://www.multipath-tcp-org
    - Mailing list to discuss with other researchers
- Replicability
    - Multipath networks pushed us to optimize reaction to link failures
        - Convinced two PhD students to start a company focused Multipath TCP
    - Created a community with other MPTCP researchers
        - Actively encouraged them to submit patches and reviewed them for inclusion
    - New releases of the kernel patch at every IETF meeting

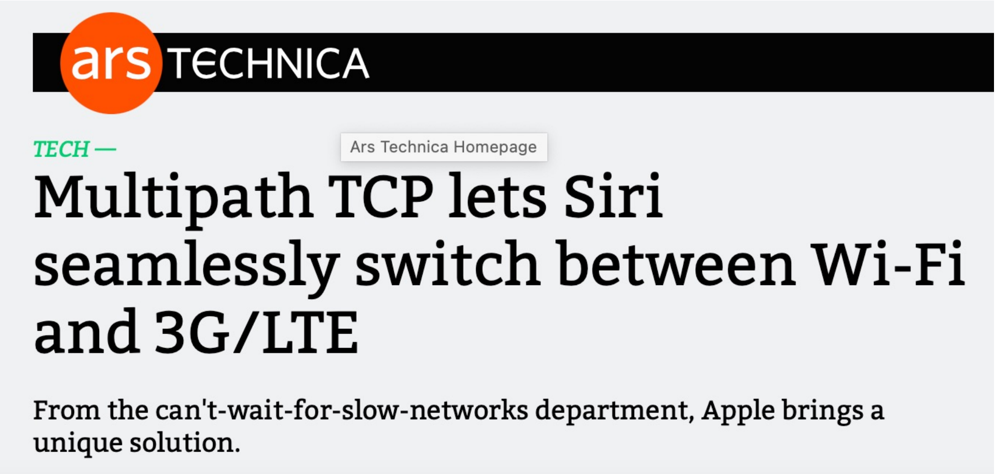
Reproducibility/Reusability

Multipath TCP for FreeBSD

Nigel Williams

njwilliams@swin.edu.au

- Reproducibility
  - Nigel Willimas and colleagues started to implement MPTCP in FreeBSD
  - Anumita Biswas informed us that Apple was interested by MPTCP
    - In September 2013 we still surprised to see MPTCP officially launched by Apple



TECH —

Multipath TCP lets Siri seamlessly switch between Wi-Fi and 3G/LTE

From the can't-wait-for-slow-networks department, Apple brings a unique solution.

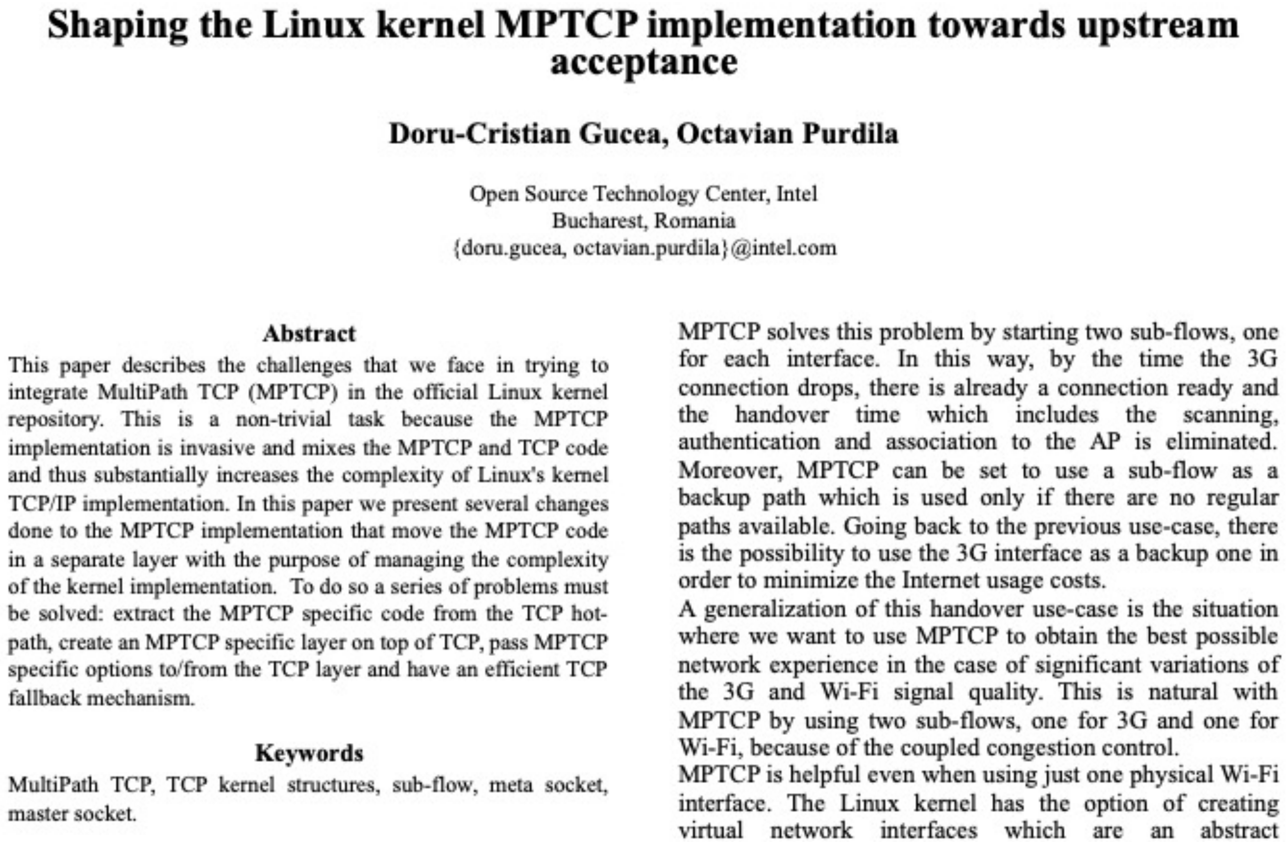
- Reusability
  - Getting MPTCP in the official Linux kernel proved to be more difficult, but
  - MPTCP received the 2019 SIGCOMM Networking Systems Award

Getting MPTCP in the mainline Linux kernel

- In 2013, we thought that everything was ready to bring MPTCP in the mainline Linux kernel and get wider deployment
  - MPTCP was published in RFC6824
  - Apple was using it on all iPhones for Siri
  - Samsung and LG started to use MPTCP in Android smartphones in Korea
- Unfortunately, MPTCP made a lot of changes to TCP and TCP is a key part of the Linux kernel code
  - Linux developers did not want to risk any regression in performance in the Linux kernel and did not consider the smartphones to be a strong enough use case at that time

MPTCP in the mainline Linux kernel (2)

- A slowly progressing engineering project
  - Netdev 2015



Shaping the Linux kernel MPTCP implementation towards upstream acceptance

Doru-Cristian Gucea, Octavian Purdila

Open Source Technology Center, Intel
Bucharest, Romania
{doru.gucea, octavian.purdila}@intel.com

Abstract

This paper describes the challenges that we face in trying to integrate MultiPath TCP (MPTCP) in the official Linux kernel repository. This is a non-trivial task because the MPTCP implementation is invasive and mixes the MPTCP and TCP code and thus substantially increases the complexity of Linux's kernel TCP/IP implementation. In this paper we present several changes done to the MPTCP implementation that move the MPTCP code in a separate layer with the purpose of managing the complexity of the kernel implementation. To do so a series of problems must be solved: extract the MPTCP specific code from the TCP hot-path, create an MPTCP specific layer on top of TCP, pass MPTCP specific options to/from the TCP layer and have an efficient TCP fallback mechanism.

Keywords

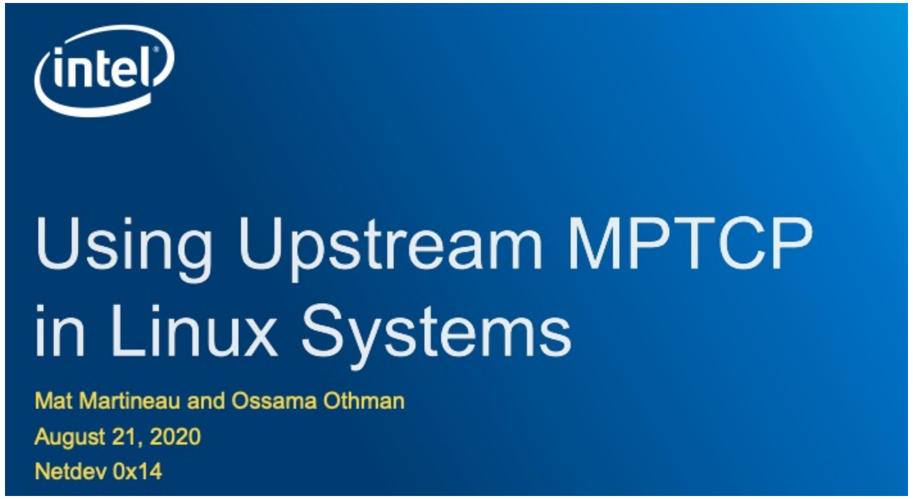
MultiPath TCP, TCP kernel structures, sub-flow, meta socket, master socket.

MPTCP solves this problem by starting two sub-flows, one for each interface. In this way, by the time the 3G connection drops, there is already a connection ready and the handover time which includes the scanning, authentication and association to the AP is eliminated. Moreover, MPTCP can be set to use a sub-flow as a backup path which is used only if there are no regular paths available. Going back to the previous use-case, there is the possibility to use the 3G interface as a backup one in order to minimize the Internet usage costs.

A generalization of this handover use-case is the situation where we want to use MPTCP to obtain the best possible network experience in the case of significant variations of the 3G and Wi-Fi signal quality. This is natural with MPTCP by using two sub-flows, one for 3G and one for Wi-Fi, because of the coupled congestion control.

MPTCP is helpful even when using just one physical Wi-Fi interface. The Linux kernel has the option of creating virtual network interfaces which are an abstract

MPTCP in the mainline Linux kernel (3)

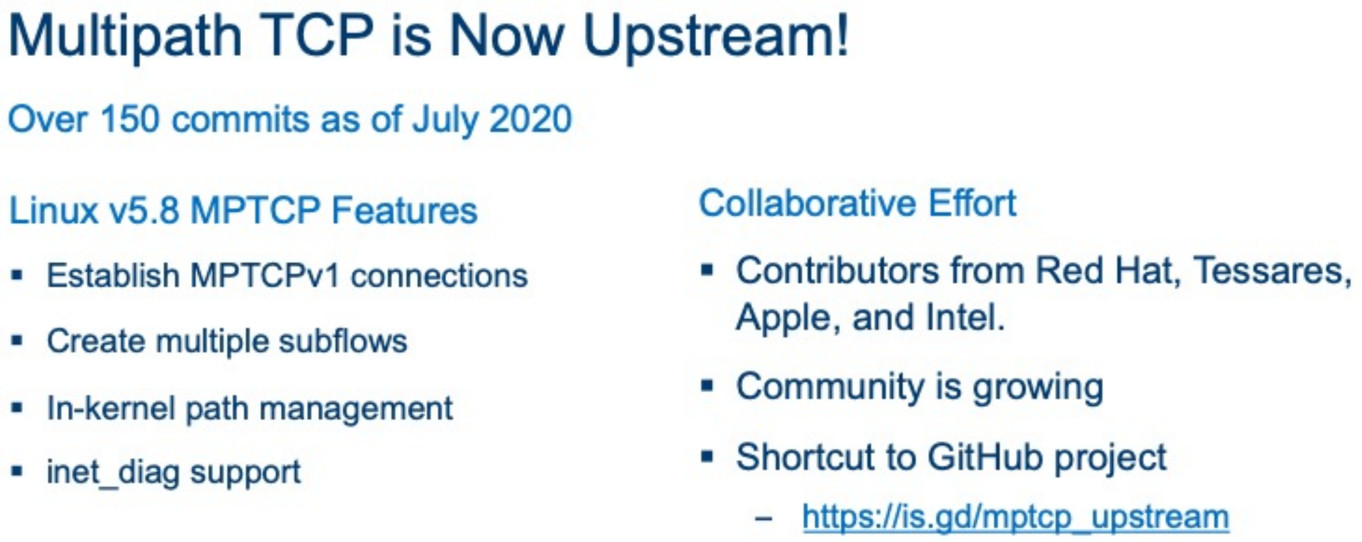


Using Upstream MPTCP in Linux Systems

Mat Martineau and Ossama Othman

August 21, 2020

Netdev 0x14



Multipath TCP is Now Upstream!

Over 150 commits as of July 2020

Linux v5.8 MPTCP Features

- Establish MPTCPv1 connections
- Create multiple subflows
- In-kernel path management
- inet_diag support

Collaborative Effort

- Contributors from Red Hat, Tessares, Apple, and Intel.
- Community is growing
- Shortcut to GitHub project
  - https://is.gd/mptcp_upstream

Some lessons learned

- Repeatability is key
  - Start at the beginning of your PhD work, don’t take wrong habits
- Releasing artefacts enables other researchers to reuse or expand your work results
  - should contribute to your citations, which is unfortunately a key metric
- Releasing artefacts takes time
  - Packaging software, documentation, providing support
- Software and artefacts are not always well valued by community
  - main research output metric remains citations and many authors forget to correctly reference the artefacts they use in their papers

Agenda

- The reproducibility crisis
- Dagsthul’s guide to Reproducibility for Experimental Networking Research
- **Artefacts Evaluations**

Earlier attempts at evaluating artefacts

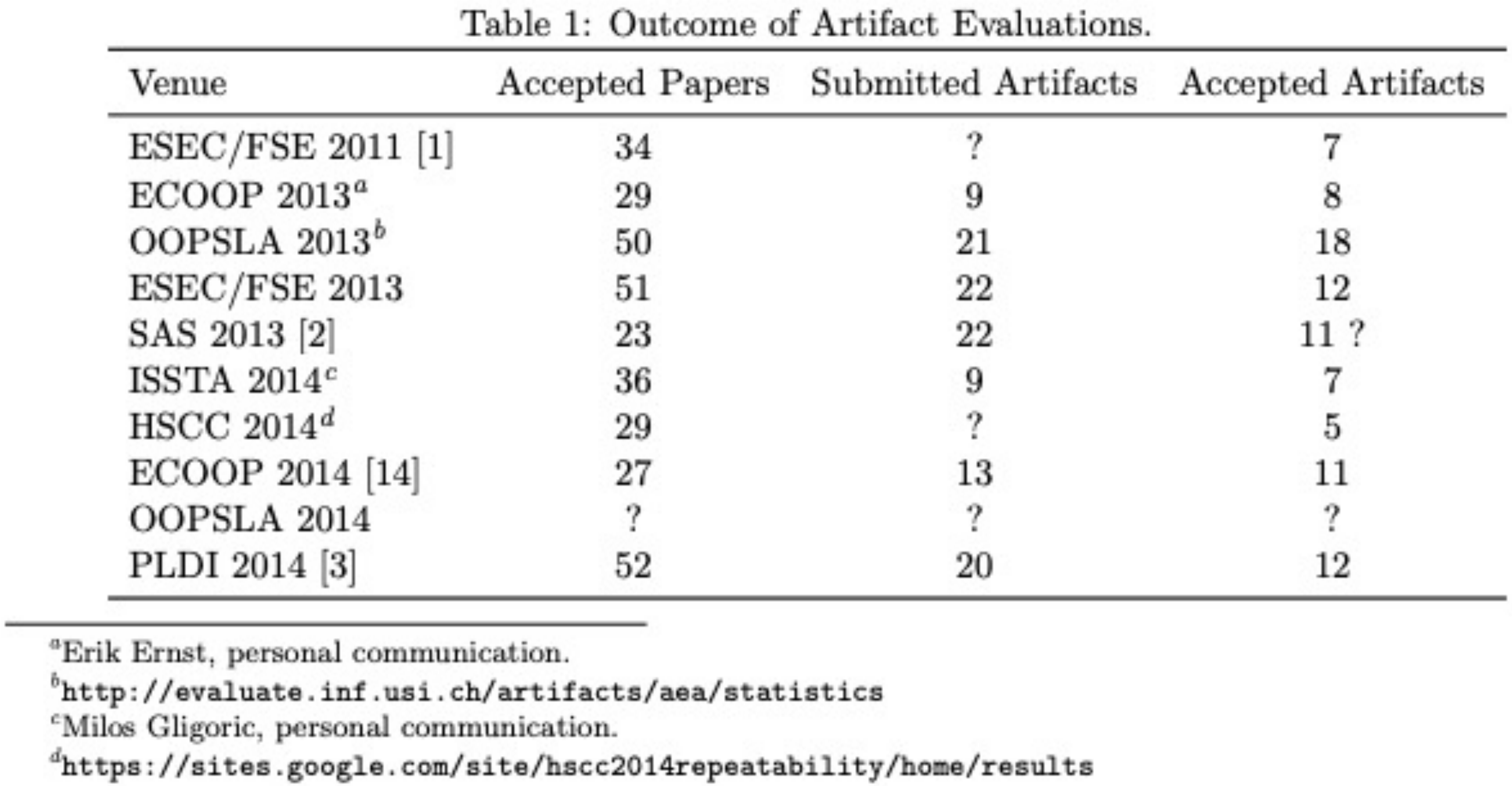


Table 1: Outcome of Artifact Evaluations.

Venue	Accepted Papers	Submitted Artifacts	Accepted Artifacts
ESEC/FSE 2011 [1]	34	?	7
ECOOP 2013[a]	29	9	8
OOPSLA 2013[b]	50	21	18
ESEC/FSE 2013	51	22	12
SAS 2013 [2]	23	22	11 ?
ISSTA 2014[c]	36	9	7
HSCC 2014[d]	29	?	5
ECOOP 2014 [14]	27	13	11
OOPSLA 2014	?	?	?
PLDI 2014 [3]	52	20	12

[a] Erik Ernst, personal communication.

[b] `http://evaluate.inf.usi.ch/artifacts/aea/statistics`

[c] Milos Gligoric, personal communication.

[d] `https://sites.google.com/site/hscc2014repeatability/home/results`

ACM Artefacts Evaluation

- Initiated by different ACM SIGs
- Adopted by ACM with artefacts badges

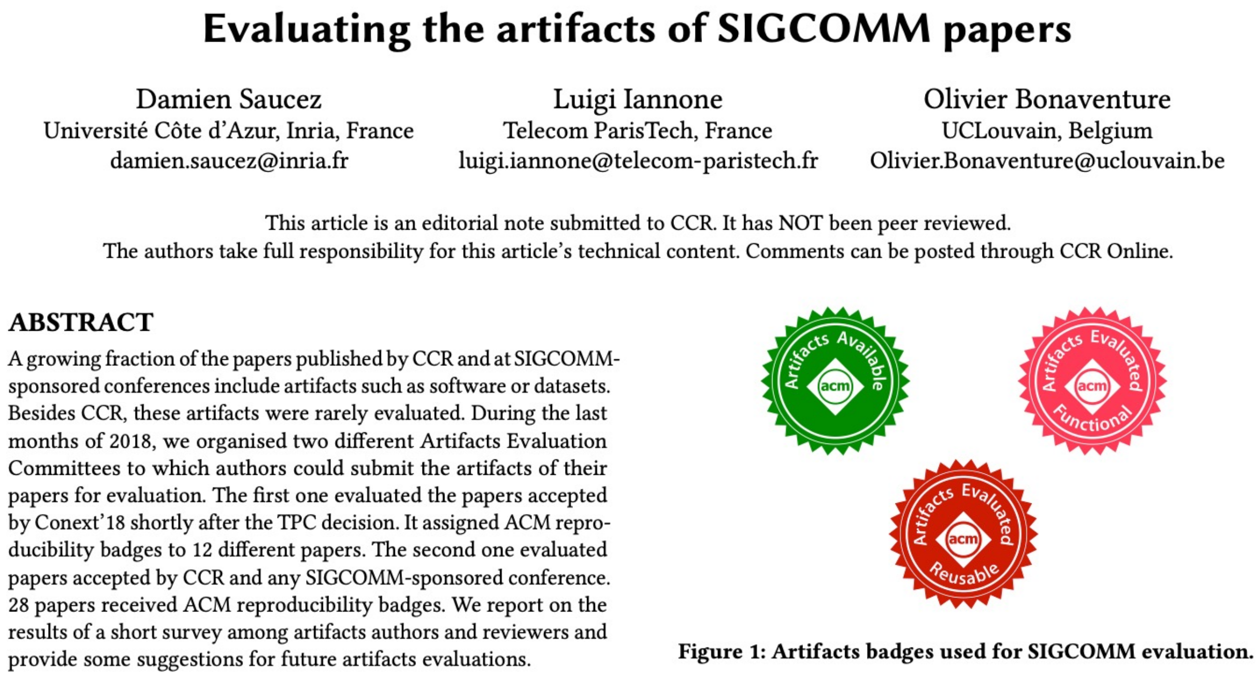
Artifacts Available
acm

Artifacts Evaluated
acm
Functional

Artifacts Evaluated
acm
Reusable

Artefacts evaluations in 2018

- Experiment with different conferences
  - Conext'18
  - SIGCOMM CCR
    - all papers published in 2018
  - SIGCOMM conferences



Evaluating the artifacts of SIGCOMM papers

Damien Saucez
Université Côte d'Azur, Inria, France
damien.saucez@inria.fr

Luigi Iannone
Telecom ParisTech, France
luigi.iannone@telecom-paristech.fr

Olivier Bonaventure
UCLouvain, Belgium
Olivier.Bonaventure@uclouvain.be

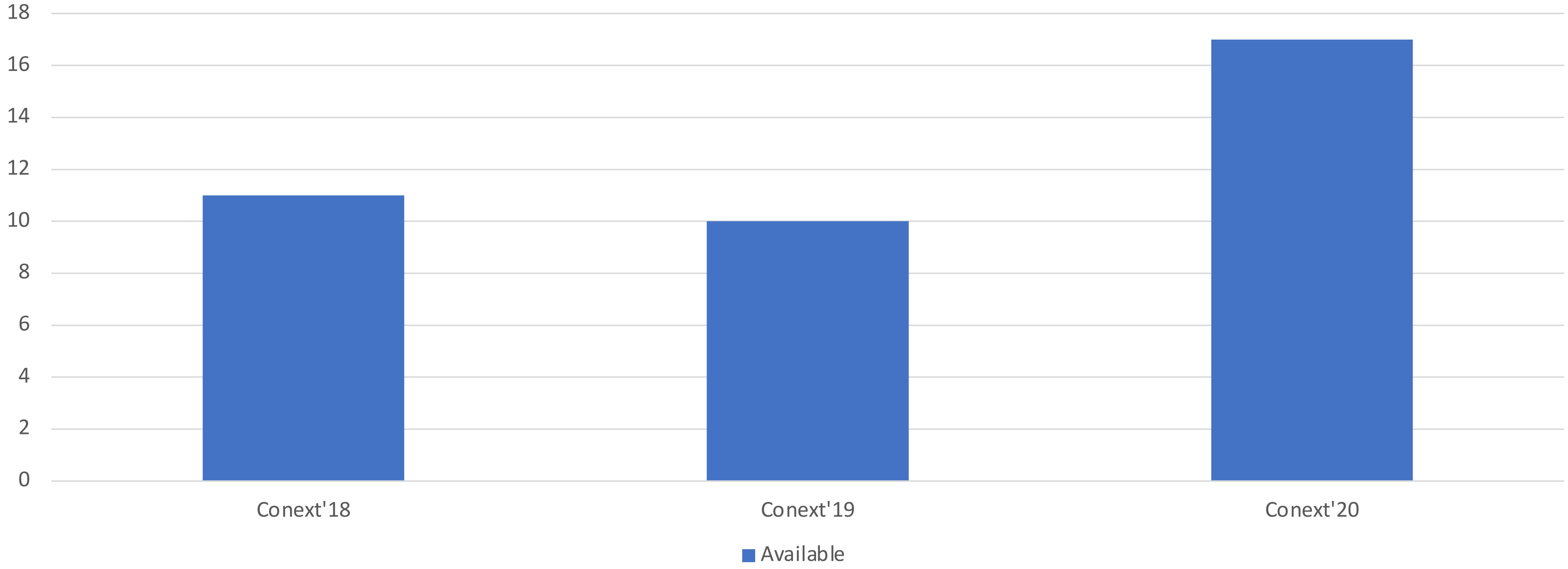
This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

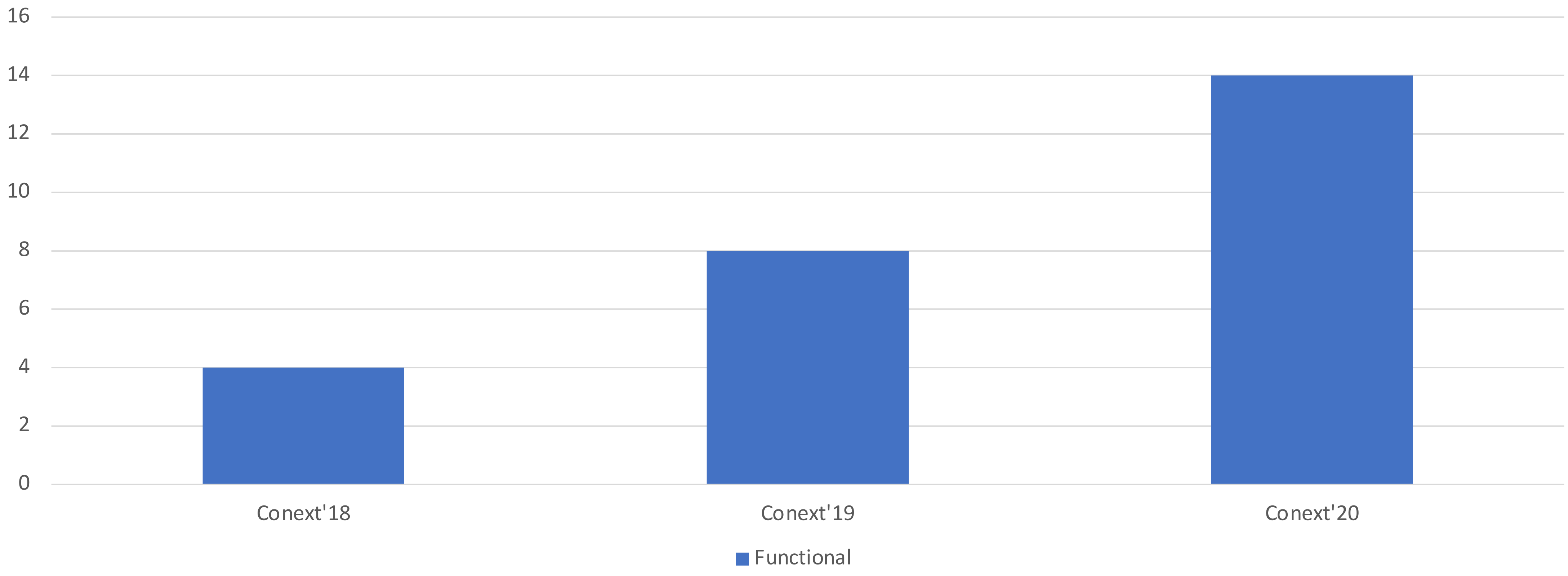
A growing fraction of the papers published by CCR and at SIGCOMM-sponsored conferences include artifacts such as software or datasets. Besides CCR, these artifacts were rarely evaluated. During the last months of 2018, we organised two different Artifacts Evaluation Committees to which authors could submit the artifacts of their papers for evaluation. The first one evaluated the papers accepted by Conext'18 shortly after the TPC decision. It assigned ACM reproducibility badges to 12 different papers. The second one evaluated papers accepted by CCR and any SIGCOMM-sponsored conference. 28 papers received ACM reproducibility badges. We report on the results of a short survey among artifacts authors and reviewers and provide some suggestions for future artifacts evaluations.

Figure 1: Artifacts badges used for SIGCOMM evaluation.

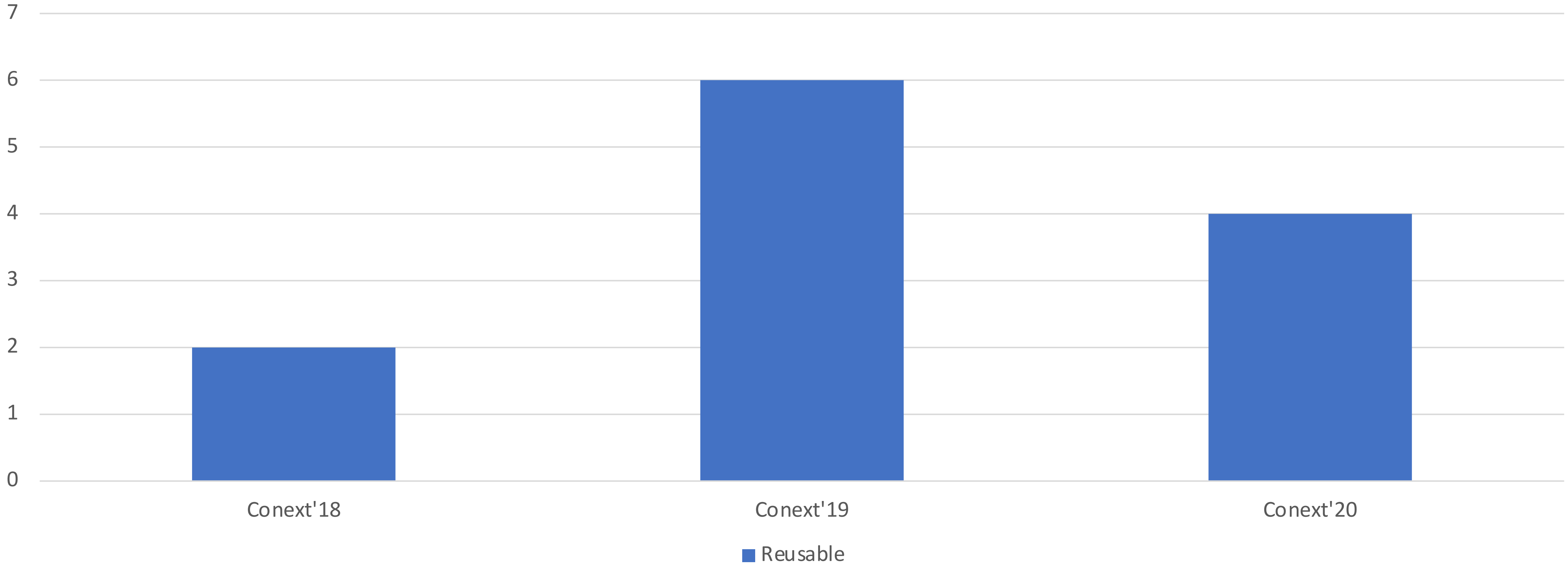
Available artefacts at Conext



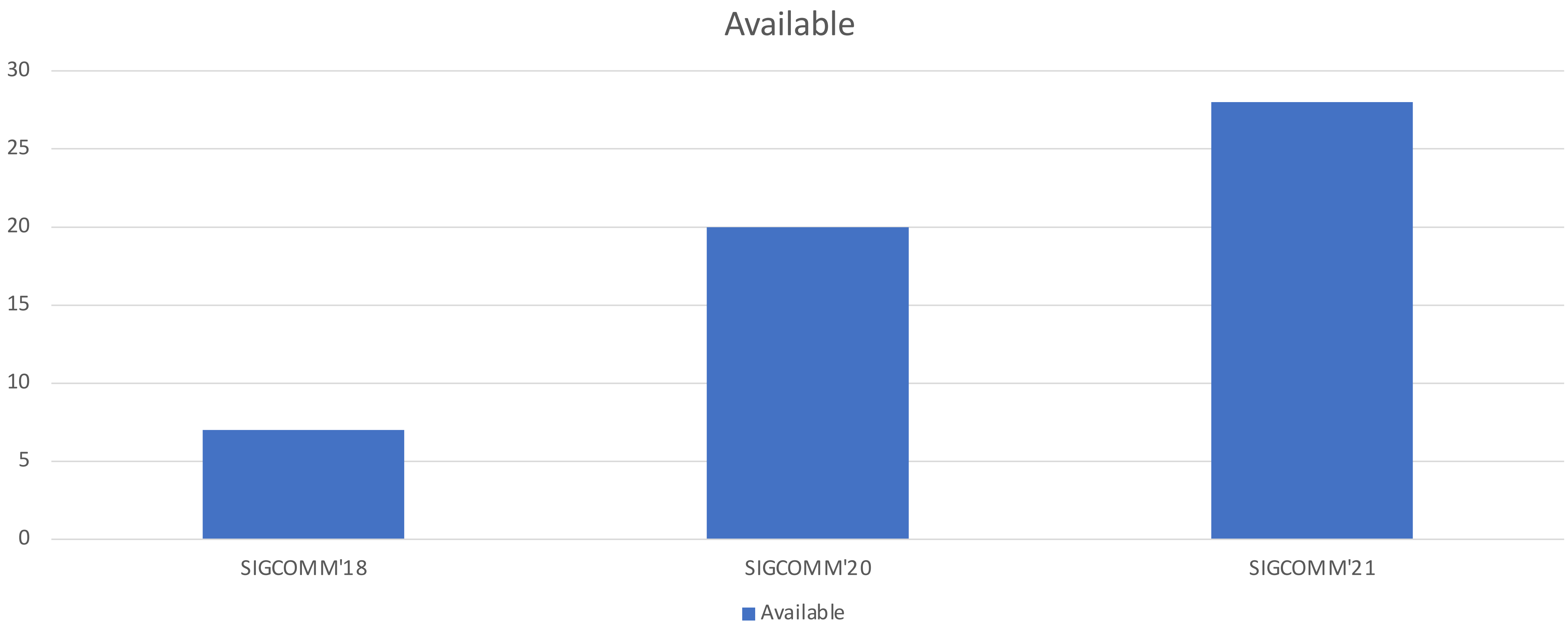
Functional artefacts at Conext



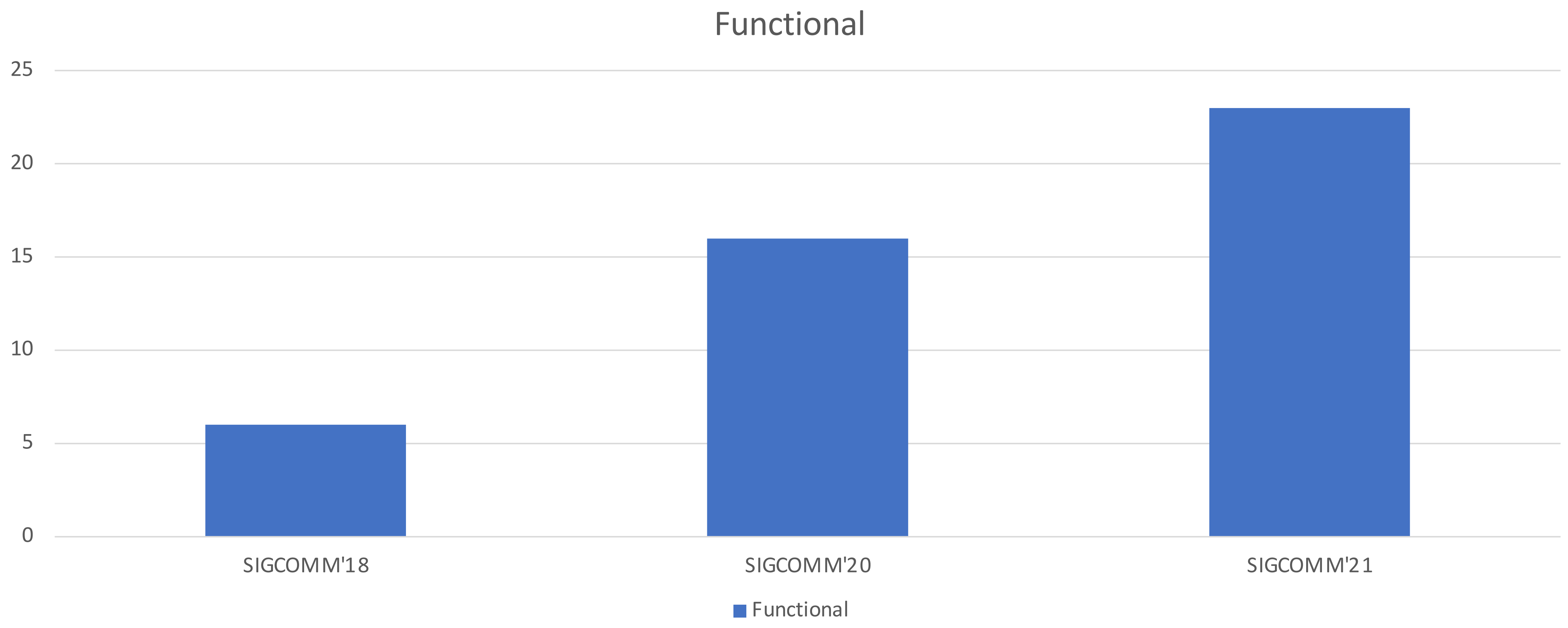
Reusable artefacts at Conext



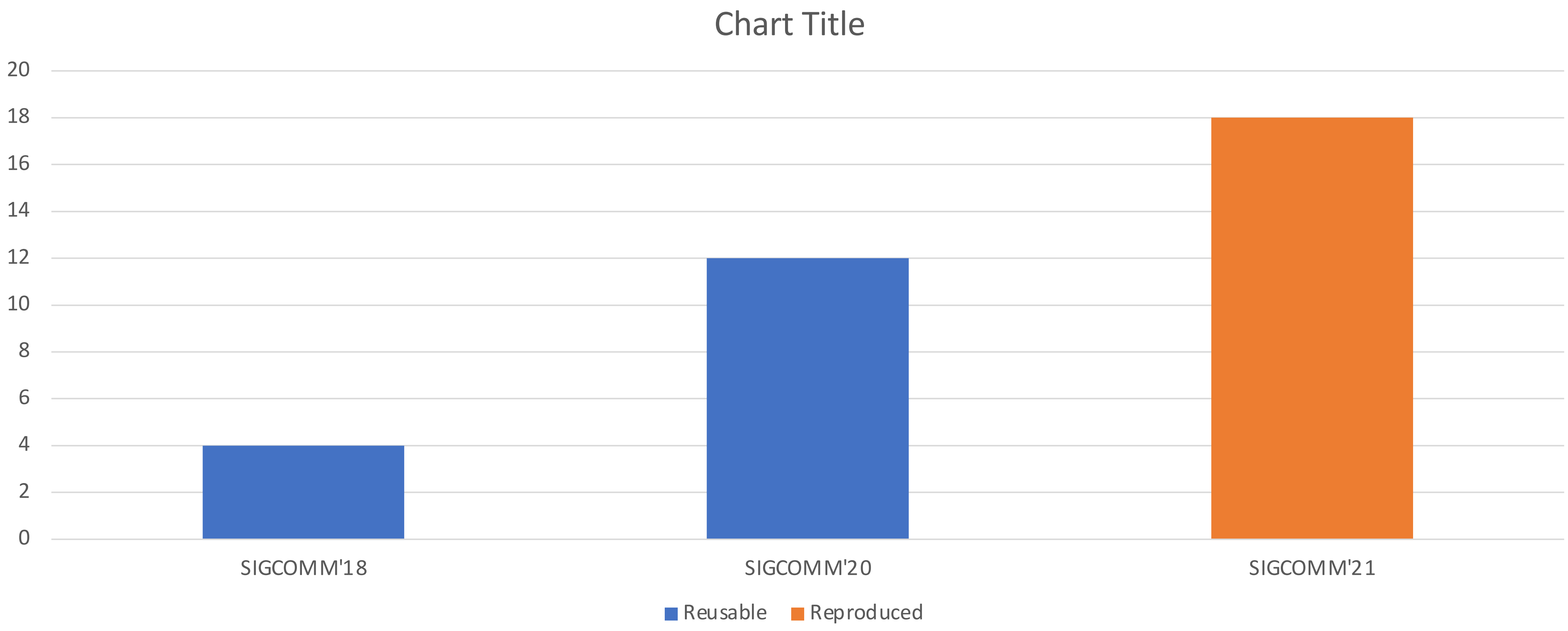
Available artefacts at SIGCOMM



Functional artefacts at SIGCOMM



Reusable/reproduced artefacts at SIGCOMM



Artefacts evaluation committees

- Objective
  - Verify that the artefacts provided by the authors of *accepted* papers support the claims of the paper
    - a posteriori evaluation, assigns badges
- Artefacts review
  - Starts shortly after paper acceptance
    - Why not during the paper review ?
  - Reviewers are volunteers who agree to spend time
    - You can (should) volunteer
    - Allows you to interact with researchers who are working on related topics
    - Allows you to understand how to provide good artefacts
  - Decision
    - Reviewers suggest badges for each paper based on their experience with the artefacts

Recent artefacts evaluation committees

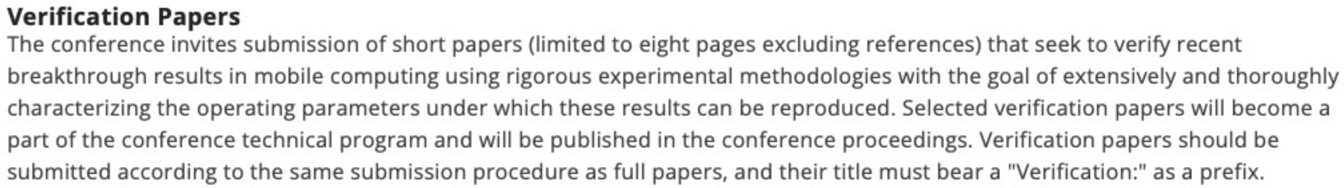
- Conext 2021
  - Chairs: Anna Brunstrom, Ken Calvert, Gianni Antichi
  - https://conferences2.sigcomm.org/co-next/2021/#!/artifactcommittee

- SIGCOMM 2021
  - Chairs: Aaron Gember Jacobson, Arpit Gupta
  - https://conferences.sigcomm.org/sigcomm/2021/cf-artifacts.html

- Internet measurements (IMC, PAM, TMA)
  - Not yet

Recent artefacts evaluation committees

- Mobicom
  - Not yet, but



Verification Papers
The conference invites submission of short papers (limited to eight pages excluding references) that seek to verify recent breakthrough results in mobile computing using rigorous experimental methodologies with the goal of extensively and thoroughly characterizing the operating parameters under which these results can be reproduced. Selected verification papers will become a part of the conference technical program and will be published in the conference proceedings. Verification papers should be submitted according to the same submission procedure as full papers, and their title must bear a "Verification:" as a prefix.

- Mobisys
  - *To honor the system nature of ACM MobiSys, this year we integrated an artifact evaluation program in the review process. Submissions could opt in for such a program by expressly indicating that at the time of submitting the paper. By doing so, if the paper was eventually accepted, the authors were committed to providing implementations, models, test suites, benchmarks, and data used to derive the results presented in the paper to the artifact evaluation committee. This was formed by a subset of PC members who volunteered for this job. A total of 12 papers have ultimately applied for the program. The artifact evaluation committee examined the provided artifacts and validated authors' requests of one or more "Artifact Evaluated" badges available as part of the corresponding ACM initiative.*

- ICNP, INFOCOM
  - Not yet

Artefacts evaluators

- You will work with the artfeacts of recently accepted papers
  - Nice way to see new research results
  - Nice way to interact with authors
  - Allows you to learn how to prepare the artefacts for your future papers
- You will encourage authors to release artefacts
  - artefacts will be available if you want to compare your solution with previous papers