Report from Dagstuhl Seminar 13472

Global Measurement Framework

Edited by

Philip Eardley[1], Marco Mellia[2], Jörg Ott[3], Jürgen Schönwälder[4], and Henning Schulzrinne[5]

1 **BT – Suffolk, GB,** philip.eardley@bt.com
2 **Politecnico di Torino, IT,** mellia@tlc.polito.it
3 **Aalto University, FI,** jorg.ott@aalto.fi
4 **Jacobs University – Bremen, DE,** j.schoenwaelder@jacobs-university.de
5 **Columbia University, US,** hgs@cs.columbia.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13472 "Global Measurement Framework".

Seminar 17.–20. November, 2013 – www.dagstuhl.de/13472
1998 ACM Subject Classification C.2.3 Network Operations, C.4 Performance of Systems
Keywords and phrases Internet measurements, Quality of experience, Network management, Traffic engineering
Digital Object Identifier 10.4230/DagRep.3.11.144

1 Executive Summary

Philip Eardley
Marco Mellia
Jörg Ott
Jürgen Schönwälder
Henning Schulzrinne

The Internet has a history of unexpected and often unpredictable behaviors due to manifold interactions of thousands of networks, and billions of components and devices and users. The resulting complexity requires measurements to understand how the network is performing, to observe how it is evolving, and to determine where failures or degradations occur. Especially with constantly evolving applications and their interaction paradigms, new phenomena occur and need to be factored into operations and management: one example is the substantial effort going into defining interfaces to assist peer-to-peer applications so that the amount of cross-ISP traffic is reduced. Measurements thus form an integral part of network operator tool sets to keep the net up and running.

But measurements are equally important for the research community to understand network traffic as well as protocol and application dynamics and their evolution. And they assist in quantifying application and (access) network performance and thus provide a tool for end users and regulators to monitor operators and their service level agreements. Tools such as speedtest.net have become widely used for individual measurements and basic ISP rating. Measurement service providers such as SamKnows or RIPE offer networks of probes,

i.e., separate devices or embedded software on access routers, for continuous background measurements at the end users. These help ISPs and regulators in their work. Standards bodies such as the IETF and the Broadband Forum have established working groups to define a global measurement architecture and common interfaces and to extend the set of metrics describing communication properties.

This Dagstuhl seminar brought together researchers from industry, academia, and regulators to discuss the state of the art in measurements and their exploitation, measurement and analysis techniques, privacy and anonymization, and to contribute to a common understanding in a number of areas, including:

- improving the expressiveness of measurement metrics (and develop appropriate new ones) beyond throughput, loss rate, and RTT so that the actual application-specific user *quality of experience* can be assessed;
- expanding the reach, scale, and diversity of measurements and the corresponding data analysis to obtain a more comprehensive view on the performance of networks and applications;
- structuring the otherwise mostly disconnected measurement activities to allow interfacing between them and/or providing defined access methods to them, for both carrying out measurements and accessing measurement results (offline and in real-time);
- providing ways to better instrument and more broadly utilize measurement infrastructure, inside operators, for end users, and at third parties.

Because the means for taking steps towards achieving the above goals was on learning about and from each other and developing joint perspectives, the seminar chose an extremely interactive organization comprising three elements:

1. Individual presentations were limited to an initial round of introductions (1 slide each) covering a set of questions for the participants to get know each others background and interests.[1]
2. Panel discussions (with ample involvement of the "audience") set the stage for the discussion topics of the day.
3. Extensive group work to dive into a number of topics and also for presenting and discussing the group outcome on the next day.

A side effect of this organization is that there were virtually no individual talks and hence no talk abstracts were collected.

We focused on two complementary aspects of a global measurement framework: 1) *creating* a global measurement framework and 2) *using* such a framework. Both were introduced by panels, with a lot of discussion contributing to these overviews, as described in the respective introduction to the following two sections.

[1] The complete slide set is available on the seminar web page at http://www.dagstuhl.de/13472/.

2 Table of Contents

3 Building a Global Measurement Framework

Arthur Berger (Akamai), Benoit Claise (Cisco), Sam Crawford (SamKnows), and Daniel Karrenberg (RIPE) introduced aspects and issues of building a (commercially) viable network measurement infrastructure and its constituents, emphasizing their respective angle on the problem. Today, we have many different measurements systems in operation at different scale and reach. Each of these systems exhibits a bias in some form, e.g., whom they are measuring for (ISPs vs. regulators vs. users vs. researchers), whether measurements are active or passive or a combination of both, where the measurement points are located, etc. A truly global measurement framework has to bridge theses biases. Different target groups may be interested in different metrics and different granularity of reporting (which may or may not be compatible). Active measurements are good for reference measurements, as the test traffic is defined, but will only offer limited insight as measurement traffic type and timing won't perfectly mimic real users. Passive measurements may be more inclusive; on the other hand, they can only be made on the traffic that exists at that moment and also raise issues concerning user traffic observations and thus privacy concerns. And the number and location of vantage points will be different depending on the questions we are trying to answer. We also covered pitfalls in designing and operating large-scale measurement systems and privacy aspects as cross cutting themes.

One key topic for subsequent discussion, metrics and measurements, was of such broad interest that it was covered in plenary style. In addition, we identified four working groups to follow up on selected sub-topics: 1) Doing it wrong: worst practices, 2) Privacy, 3) Latency measurements, 4) infrastructure and interfaces. We will briefly recap their results below.

3.1 Measurements and metrics

We identified different use cases and, as already mentioned, measurements and metrics may differ depending on which use we are looking at. In general, however, it is important that metrics are clearly defined – how they are measured (tests, vantage points, math, number and intervals of repetitions, time span, etc.) and what their semantics are (what they are useful for and what not) – so that they can be implemented by different parties (in different environments) but yield reproducible and comparable outcomes. To counter implementation errors, reference code may help. To prevent interpretation errors, metrics should be clearly and unambiguously observable. Whenever measurements are carried out and documented in data sets, it is important to record the conditions for the measurements and provide sufficiently detailed documentation in the data sets to allow for later re-use, comparison, etc. without running the risk of misinterpretation.

Concerning standardization, we have to start with the metrics that are understood well enough and are of broad interest; quite a few (simple ones) are already addressed by IETF or ITU specifications. Operational measurement infrastructures (SamKnows, RIPE ATLAS) measure some 15 metrics (as active measurements), focusing on those motivated user and regulatory use cases. Metrics are evolving to become more sophisticated (which creates a tension between standardization and innovation): from measuring simple download speeds and maybe round-trip-times (as is still dominant when judging ISPs) to application-dependent impact (e.g., web page load time, rebuffering events for video streaming) and the more "elusive" quality of experience in general, including availability and robustness. This increasing complexity requires careful and thorough definitions. Many subtleties matter,

which need to be identified: to avoid implementation and measurement errors, to control bias, to include in documentation of results and data sets, etc. The potential for (arbitrary) rise in complexity may call for modesty in what we try to achieve so that metrics and measurement definitions aren't overdone beyond what can be meaningfully documented, interpreted, and compared.

Metrics are used for two closely related yet distinct purposes that would be nice to disentangle: network measurements per se to understand network behavior and benchmarking for performance comparison or rating. The former probably calls for a larger number and more fine-grained metrics, whereas the latter may require fewer metrics and coarser levels.

3.2 Doing it Wrong: Worst Practices

The introduction panel and the general discussion about metrics already captured system design aspects regarding failures. This group work focused on data gathering, processing, documentation, and to some extent interpretation. The first observation emphasized was the importance of metadata (already alluded to above): the environment (e.g., cross-traffic, outages), the context (e.g., applications, usage), and information about the measurement infrastructure (e.g., version and patch numbers). Such metadata are equally important for proper interpretation of results as the measurements themselves.

Ideally the measurement data should always be kept in raw form, not just results after applying an interpretation function, because this function may change over time and one may want to go back and look at old results in the light of new insights. Before measuring, measurement setups (including specific implementations) need to be (repeatedly) calibrated; ideally experiments include such calibration steps. Lack of calibration would lead to uncertainty of the results. Once measurement data is collected, correction factors may have to be applied to raw data and sanity checks ("repair") should be performed to discard data that are obviously broken. Care needs to be taken to not create a bias during such steps.

Finally, data needs to be interpreted in the context of, e.g., a specific application to understand if the results are "good" or "bad". A recent trend has been towards developing quality of experience metrics, but this is work in progress and really hard to get right. Particular caution is required when attempting to map observable network characteristics to subjective experience metrics.

3.3 Privacy

Privacy is a tricky topic when it comes to measurements, especially when performing passive measurements (when user traffic patterns or even user data are observed to understand network performance), but also to some extent when carrying out active measurements (e.g., when measurement systems try to avoid colliding with user activity and thus time stamps from measurements yield insights into when users are active). This becomes particularly relevant when data is not (just) used for internal evaluation (e.g., of an ISP or a measurement platform) but when (raw or pre-processed) data is anonymized and shared, e.g., for research. Mistakes, e.g., during the anonymization process cannot be undone (examples from the past can be found in literature).

Defining the problem space requires understanding against whom to protect the users,

businesses, or other entities (e.g., against service providers, applications, governments) and who could serve as the trusted entity offering this protection (e.g., ISPs, governments, privacy service providers, or even the crowd). Measurements and privacy can be at odds with each other (and when data collection is carried out by governmental organization they indeed are, as recent history has shown). Yet, in many cases, people are constantly giving their usage and performance data away to service providers anyway (or they would need to pay for a service). An extreme position would be to argue that, under present circumstances, there is not much of privacy left in the first place. A "privacy as a service" provider could help changing this when interacting with individual services or ISPs, but this would come at a cost for the user.

For data collection using measurement along the lines of this seminar, rules could be defined for maintaining user-related data once collected. The first aspect is minimizing the amount of data collected, anonymizing to a good degree, and then discussing mechanisms how to achieve this. We note that there may be some tension between following these ideas of user privacy protection and keeping raw and encompassing metadata as discussed in the previous section.

3.4 Latency measurements

Measuring latency, while straightforward at the first glance, features numerous sophisticated subtleties when looking at different protocols and applications: latency may be defined in different ways depending on the intent: for example, round-trip time of an ICMP or UDP ping, TCP connection establishment, application layer latency. When concerned with network measurements, we typically try to measure the latency imposed by the network, but other sources of delay exist: in the operating system, in the server or data center, and then in different segments of a network path (home network, access network, etc.).

Latency measurements can be carried out using a number of tools operating at different layers (the simplest ones being *ping* and *traceroute*). They may carry out measurements end-to-end between two hosts or they may receive ISP support (e.g., for timestamping packets when they pass through) so that finer-grained resolution along the measured path becomes possible. Measurements can determine the base RTT (some flavor of calibration), the latency under load (max RTT), and delay variation. Latency-related metrics include RTT and one-way delay variation. Tools for these basic metrics are available.

What is missing includes: being able to identify the source of latency (which requires cooperation of the ISPs), transforming basic latency measurements into semantically richer metrics that reflect the user experience (which is highly application specific and tricky to achieve, as noted above), extensive latency measurements in mobile networks, and support for passive latency measurements at a single point in the network (e.g., when requests and responses or TCP segments can be mapped).

3.5 Infrastructure and interfaces

The group distinguished four different architectures for large-scale measurements, ranging from ubiquitous, but fixed-function devices, to fully programmable custom applications, typically on general-purpose computing equipment. The components of a measurement platform may be owned by the ISP or a dedicated measurement entity or by the subscriber (e.g., third party

modems). Networks to be measured (and whose contributions to measurement results may have to be dissected) include the public networks (ISPs) and private networks (enterprises, universities, etc.)

We differentiate three classes of measurement use cases that differ in scale and purpose:

1. (continuous) large-scale measurements to understand network performance representative of a specific population,
2. monitoring (sampling) intra- and inter-domain operation, and
3. trouble shooting (on-demand) at the scale of individual users or ISPs.

To support these classes of operation, we define a number of logical components: measurement agents (as the active entities carrying out measurements), measurement servers (as the entities that act as peer points for the measurement agents to perform measurements), one or more controllers (as the instance(s) directing the operations of the measurement agents), and a collector (as a data sink to which the measurement agents upload their results). The operation requires several protocols: between the measurement agents and the controllers to retrieve instructions (schedules, tests to be carried out, servers to be contacted, etc.); the measurement protocols used to execute the tests between the measurement agents and servers; and the upload/collection protocols to store the measurement results. These protocols could be complemented by data formats (for measurement data and metadata) and possibly query formats to access the results database. Finally, mechanisms for software upgrades may be provided to update the measurement agents.

When measurements are carried out not just against measurement servers, but by contacting hosts of service providers (to get a more accurate reading of application performance), we also foresee the necessity of a "do not probe" mechanism by means of which sites can indicates that they do not wish to be measured (conceptually similar to robots.txt for web servers). Other mechanisms may be defined to indicate the willingness of sites of participate in measurements (e.g., using DNS SRV records) as well as to limit the volume of measurement traffic incurred to a given site.

4 Using a Global Measurement Framework

Al Morton (AT&T), Henning Schulzrinne (Columbia University, FCC), Andrea Soppera (BT), Fabian Bustamante (Northwestern University) introduced the topic of use cases for a global measurement framework. We originally considered three use cases defining who the measurement results are targeted at:

1. The operator use case, in which operators use measurements for monitoring and optimizing their networks;
2. the regulator use cases, in which a government entity wants to oversee that the operators fulfill their obligations and do not overclaim the services they are offering;
3. the end user use case, in which measurements assist the end users, e.g., in validating the services they are obtaining and, (in conjunction with operator support) in resolving access or performance problems.

One special case related to end users are application designers, whose applications could learn from measurements about the expected performance (or changes therein) and react accordingly at runtime.

Across the use cases, the "target" for measurement results may differ. On the one hand, there is a technical audience (engineers, researchers, etc.) interested in improving the (cost

effectiveness of) network services and performing trouble shooting when needed. On the other hand, we have regulators and (company) lawyers and further less technical people who also need or want to understand and work with results from network measurements, e.g., to ensure compliance with government regulations, compare networks, etc. While probably all metrics can be gamed in one way or another, the risk of being caught (since users, peers, competitors, and third parties are monitoring as well) is substantial, so that there is little incentive for cheating – metrics won't need to be protected from this perspective. Nevertheless, it would be nice if we could define metrics in a way that if an operator attempts gaming them, this would result in performance (or other) benefits for the user.

Two working groups were formed: 1) One use case covering trouble isolation for operators (which also covers elements of the end user use case) and 2) one addressing data analysis in general.

4.1 Use Case: Service Provider Trouble Isolation

End users carry out measurements because they do care about their network performance – this is reflected in speedtest.net having seen more than 5bn measurements. There are many reasons for this, including: a user's experience may be unsatisfactory; a user may have a new service subscription (ISP or content) and wants to see if it lives up to expectations; a user may have bought new equipment and wants to see its (improved) performance; or a user may carry out tests, possibly as a byproduct of another activity.

However, carrying out such user-invoked measurements using some of the most prominent test platforms may actually not help very much: the user only makes a single measurement point at a time, without calibration to a baseline as discussed above. Performing a ping-based latency (RTT) measurement and performing then an end-to-end file transfer to a point of the measurement system fails to localize the issues (they may not even show any issues if the problem is in a network segment not traversed by the test) and are of unknown accuracy. While ISPs have carefully managed networks, some segments of the path are not managed at all: this includes especially the user's home network. This would require separate measurements, especially when WLANs are involved, given that the WLAN channels used overlap in many buildings with unpredictable performance impact.

If we are able to deploy measurement points at the edge of the network and coordinate measurements from endpoints or home network devices with such embedded measurement point deployments, we can help customers isolate whether the problems is in their home network or the access network. Carrying out measurements may influence future quality of service and quality of experience for users and can yield a positive experience for the users and improve satisfaction with the subscribed service.

There is a tension concerning privacy: the more data is available (instantaneous and historic data) about a user, the more effective trouble shooting can be carried out; yet, at the same time, there is a legitimate desire to maximize user privacy, e.g., by minimizing the amount of data collected and stored. In some scenarios there may also be a tension with business sensitivity.

4.2 Data Analysis

This group addressed the mechanics of the data processing required for data analysis. First of all, generic cloud computing services (by third parties) should not be used because moving around all the large (and constantly growing) volumes of data may be hard because there are issues of trust with the cloud service provider (and well as the network), among other reasons. The consequence is that entities running measurement platforms build their own (post)processing cloud. We look at three case studies.

Akamai collects data for billing purposes as well as for optimizations. For billing, they collect data in quasi real-time (1 min delay), moving the data from the caches to their data center, perform aggregation using Hadoop (HDFS, hbase), and keep the data for diagnosis for two weeks and those data needed for legal purposes for two months. For optimization, DNS to cache allocations are recomputed once per minute based upon the observed performance. RIPE collects data for statistics purposes and long-term observations. They also use Hadoop (HDFS, hbase). Data collected is aggregated, the volume is reduced, and the then preprocessed to make the data sets accessible to tools such as R.

Ftw and Polito collect data from passive measurements, so the resulting data volume gets really large. They store data in SQL with a custom data warehouse solution or Hadoop, respectively, with customized post-processing. In all cases, the collected data is used for reporting purposes (structured repetitive tasks), data mining (more ad-hoc and relying on individual ingenuity for analysis). All have in common that they (have to) use custom-developed processing and evaluation solutions. What is missing is a common toolset / platform that offers a basic set of functionality applicable for the needs across the different platform described above. This also extends towards visualization and to a framework (and formats) for sharing data.

5 Impressions and Next Steps

This Dagstuhl seminar saw 2.5 days (and evenings) of lively and extensive discussions among the participants. The different stakeholders were well represented and also the mix of academia and industry was just right. Sharing perspectives and experience from their respective viewpoints was extremely valuable. We clearly made progress in understanding the issues at hand and important steps to be taken, which we will also feed into the discussion of the different working groups at the IETF. We also foresee work on a joint scientific publication documenting the insights gained in this seminar. Finally, the participants expressed strong interest in continuing our discussions as a follow-up seminar in the future.

Acknowledgements. Two EC FP7 research projects, Leone and mPlane, kindly supported the social event of the seminar.

Participants

- Saba Ahsan
Aalto University, FI
- Vaibhav Bajpai
Jacobs Universität – Bremen, DE
- Arthur W. Berger
Akamai Technologies – Cambridge, US
- Ernst Biersack
EURECOM – Biot, FR
- Trevor Burbridge
British Telecom R&D – Ipswich, GB
- Fabian E. Bustamante
Northwestern University – Evanston, US
- Pedro Casas
FZ Telekommunikation Wien, AT
- Benoit Claise
CISCO Systems Belgium, BE
- Sam Crawford
SamKnows Ltd. – London, GB
- Philip Eardley
British Telecom R&D – Ipswich, GB
- Daniel Karrenberg
RIPE NCC – Amsterdam, NL
- Mirja Kühlewind
Universität Stuttgart, DE
- Abdelkader Lahmadi
INRIA – Nancy – Grand Est, FR
- Jukka Manner
Aalto University, FI
- Marco Mellia
Polytechnic Univ. of Torino, IT
- Al Morton
AT&T – Middletown, US
- Jörg Ott
Aalto University, FI
- Fabio Ricciato
AIT – Wien, AT
- Dario Rossi
Télécom Paris Tech, FR
- Ramin Sadre
Aalborg University, DK
- Jürgen Schönwälder
Jacobs Universität – Bremen, DE
- Henning Schulzrinne
Columbia Univ. – New York, US
- Andrea Soppera
British Telecom R&D – Ipswich, GB
- Anna Sperotto
University of Twente, NL
- Burkhard Stiller
Universität Zürich, CH
- Tivadar Szemethy
Netvisor – Budapest, HU
- Brian Trammell
ETH Zürich, CH)