Application level measurements: Trends and directions

Balachander Krishnamurthy, AT&T Labs–Research

June 1, 2003

The growing area of network measurement indicates increasing interest in trying to gather empirical data across various protocol layers and examining techniques that support such measurement. One way to categorize the work in measurement [TUT] is into hardware vs. software based tools, passive vs. active measurement approaches, on-line vs. off-line analysis, LAN vs. WAN measurements, or protocol level. A quick examination based on popular literature in this area over the last few years shows that measurement research has been largely in the following areas:

1. Characterization (snapshot of traffic/behavior, general Internet)
2. Active and passive measurements
3. Specific protocols and applications (DNS, BGP, Web, streaming, P2P)
4. Routing, topology, and modeling issues
5. Statistical properties and inferencing
6. Reverse engineering and tomography: learning from outside and inferring connectivity information through indirect measurements
7. Examining anomalies

Some researchers focus exclusively on the underlying protocols and lower level details (varying algorithms dealing with congestion control) while others work at a high level of examining specific application or problem (e.g., are CDNs being used efficiently). Quite a bit of work is in the interaction between the layers: work at an application level that takes into account the issues at the network layer. Often lessons and techniques from one are

applied to the other (use of old ideas such as piggybacking at higher layers) or characterizing variability in intra-domain and inter-domain traffic across different protocols.

1 Breakdown of steps

At least four steps/concerns are involved in the measurement process. We briefly discuss them with a look at commonalities across the wide spectrum of measurements carried out today.

1. Measurement infrastructure software/hardware: local or remote data acquisition. These range from *tcpdump* and friends to packet monitors such as Gigascope [GIGA] which combine hardware and software mechanisms. Remote data acquisition or measurement has to deal with security concerns as the measurement process itself may be considered intrusive. This is especially true if it triggers bugs and demonstrates flaws in implementation (e.g., Request-URI too large causing Web servers to crash [KA01]).

2. Transmission and storage: Large amounts of data gathered over long periods of time have to be transmitted over the network and stored in an accessible manner for future analysis. Except for on-line analysis, most data is stored. On-line analysis has much stricter temporal requirements and a smart combination of hardware and software may be the best answer. Typically, the speeds are still limited to OC-48, for doing any serious on-line analysis for streamed data.

   Transmission limitations have already led to dependence on sampling which raises related issues (sampling accuracy, hidden traffic, validation, moving beyond single protocol). Alternatively filtering on data can be done before transmission by pre-compiling queries (aggregation, selection etc.) into monitors. Storage can be in compressed flat files (as is typically done with *netflow*), databases, warehouses, or application-specific structures. Recently there has been work in using graphs as a storage structure [DNSG] which has enabled quick identification of anomalies. Some potential for creating common formats and exploiting reasonable storage techniques.

3. Processing and analysis: Once data has been captured, transmitted, and stored, software typically takes over for processing and analyzing

the data. Typically this is not viewed as a serious issue in measurement research and the details of the work are not presented. Often the complexity is hidden from view and there is considerable overlap/redundancy in efforts expended at this step. We will discuss this in more detail in Section 2.

4. Validation and sharing: A final step, not often followed is actual validation of the data and analysis. A typical stopping point is publication of a paper or an Internet-Draft. Validation requires comparison with other approaches that have captured similar data. One way to validate is to share data but this runs into concerns of privacy and organizational difficulties. An alternative is to anonymize the data and make it available and some strides have been made in this arena.

2 Role of Software

Measurement data is gathered at various levels, protocols, locations, and duration intervals. Infrastructures like NIMI [NIMI], and PlanetLabs [PLAB] have been set up to facilitate common measuring sites and enable reasonable comparisons of results. But there is no mechanism to compare results from similar studies that are even carried out on similar topics. A study on the effectiveness of CDNs, say, can be biased based on a variety of factors: locations, times, access to additional data, traffic pattern differences etc.

Worse yet, currently there is not a lot of *software* that is commonly available. Clearly, each experiment is going to require different bits of base, glue, and analysis software. However, various measurement experiments do tend to have many common characteristics that can be mapped to a set of scripts, libraries, and toolsets. Given that a few hundred researchers are involved in this effort, and there is significant duplication of effort, one would expect a toolbase to evolve in support of the effort. The PlanetLab effort has focused on creating a set of machines and environment where network measurements can be carried out and reported in a uniform fashion. A software tool collection to augment that would reduce the amount of work done by many and improve the quality of the tools and scripts used. The set of tools available in PlanetLab is very limited at present [PLTL]. Categories of tools include scripts (Perl, shell), contributed openly available code (e.g., for parsing netflow like flow-tools set or FlowScan) etc.

Examining the role of software we find that the skill sets of researchers and students ends up dictating the tools used rather than the following of a

systematic process. A tool or library that can be applied to diverse datasets uniformly would be preferable over a set of hand crafted (often quickly and dirtily) scripts and programs. Interpreted languages like Perl are indeed useful to test out ideas quickly but often end up as the final artifact. Thus the notion of being able to ask difficult questions at the exploratory stage and obtain quick answers on large volumes of data appears to be alien to many. An approach which inverts this and allows one to construct queries quickly (without the need for any significant programming) and yet be applied to large and diverse data sets would be preferable. I will give examples of this approach in my talk.

References

[PLTL] `http://www.planet-lab.org/php/contrib/contrib.php/`.

[TUT] Carey Williamson, "A Tutorial on Internet Traffic Measurement," in *IEEE Internet Computing*, Vol. 5, No. 6, pp. 70-74, November/December 2001

[GIGA] Chuck Cranor, Theodore Johnson and Oliver Spatscheck. "Gigascope: a stream database for network applications" in *Proceedings of SIGMOD 2003*. June 2003.

[KA01] Balachander Krishnamurthy and Martin Arlitt. PRO-COW: Protocol Compliance on the Web—A Longitudinal Study. In *Proc. USENIX Symposium on Internet Technologies and Systems*, March 2001.
`http://www.research.att.com/~bala/papers/usits01.ps.gz`.

[DNSG] Charles D. Cranor, Emden Gansner, Balachander Krishnamurthy, and Oliver Spatscheck. Characterizing Large DNS Traces Using Graphs. In *Proceedings of ACM Sigcomm Internet Measurement Workshop*, Nov 2001.
`http://www.research.att.com/~bala/papers/imw01-dns.ps`

[NIMI] Vern Paxson, Andrew Adams, and Matt Mathis. Experiences with NIMI. In *Proc. Passive and Active Measurement: PAM-2000*, April 2000.
`http://www.aciri.org/vern/papers/nimi-pam-00.ps.gz`.

[PLAB] `http://www.planet-lab.org/`