A Scalable System for Sharing Internet Measurements

Mark Allman, Ethan Blanton, Wesley M. Eddy

Abstract— **This paper proposes a system for storing and sharing Internet measurement data amongst researchers. The Scalable Internet Measurement Repository (SIMR) is centered around a database of measurements, tools, experiments, users and datasets. From this set of databases users can search for particular measurements, download the tools used to make and analyze those measurements, and quickly ascertain the relationships between various measurements. The goal of this system is to facilitate the sharing of data within the research community. This sharing will allow researchers to validate the results obtained by others, as well as answer new questions that require large and diverse datasets that would be difficult or impossible for a single researcher to collect alone.**

Keywords— **Internet measurement, sharing, database.**

I. Introduction

In this paper we propose the Scalable Internet Measurement Repository (SIMR) system for disseminating Internet measurements and the tools used to collect and analyze these measurements. The system is not an attempt to cure all of the current problems with ad-hoc network measurement and home-brew data management. However, we believe that using SIMR will significantly reduce some of the current problems facing the measurement community.

The following is a list of some of the problems with collecting, sharing, and analyzing Internet measurement data that SIMR attempts to mitigate.

- Sharing measurements with the research community is currently done in an ad-hoc fashion (e.g., with a link on some individual researcher's web page). By putting a system in place that allows researchers to post their data collections in a standard way, we hope to encourage more scientists to share their data with their colleagues.
- Much of our understanding about the network is currently limited by our individual abilities to collect data. For instance, [1] studies TCP connections to a single WWW server. While the data presented in such papers may be useful, the results would be stronger and more compelling if the conclusions were based on measurements taken at numerous points throughout the network, since the network is immensely heterogeneous and no one site can be thought of as "typical". With SIMR we hope to encourage these kinds of large-scale studies by providing a way that data can be easily accumulated from multiple vantage points.
- A large percentage of the Internet measurement studies currently published are not verified by the community due to the inability of researchers to access others' data and measurement/analysis tools. SIMR not only tracks the measurements themselves, but also the tools used to take and analyze the measurements. Finally, SIMR contains a way to group tools and measurements together to easily share everything that a researcher needs to reproduce a set of results (e.g., the collection of all the data, tools and scripts used by a given researcher for a given paper).
- Even when raw measurements are released, currently there is no standard method for doing so. Therefore, while a researcher can download a particular measurement, some important piece of information may be missing. For instance, a *ping* measurement may be made available, but the measurement may be useless to another researcher unless a timestamp (indicating when a measurement was taken) is stored with the measurement. Alternatively, we gather lots of meta-data about our measurements but track this data in an ad-hoc fashion (e.g., encoding the time a measurement was taken in the filename of the output). The SIMR system requires basic meta-information about each measurement to be stored in a standard way in the database, hoping to encourage collection of potentially useful ancillary information.
- Even if individual researchers release measurement data on their own, there is no accurate method of searching the global set of data available for specific measurements. For instance, a researcher may want to find all available DMZ packet traces from January–March 2000 that have been released. The SIMR system's database will make such searches viable.
- As outlined in [2] the community has a lack of coherent longitudinal datasets for analyzing the changes in the network. A system like SIMR may encourage the collection of such datasets and will provide a method for tracking the data in a standard way over time so that researchers can easily make use of long-term data collected from various points in the network.

We want to stress that we are not proposing the SIMR system as *the way* to design a distributed system for sharing measurements (etc.) within the research community. We are forwarding the system as a *strawman* and a starting point in the conversation about whether the community believes such a system is desirable and how to design such a system[1]. We believe input from the community will likely make the SIMR system outlined in this paper stronger and more useful.

Finally, we note that the SIMR system, as specified in this paper, is intended to track *Internet* measurements.

M. Allman is with BBN Technologies/NASA Glenn Research Center (mallman@grc.nasa.gov); E. Blanton and W. Eddy are with Ohio University ({eblanton,weddy}@irg.cs.ohiou.edu).

[1] In fact, we are still thinking of additional meta-data that should be included in the system three days before the final version of this paper is due and are resigned to the fact that we will likely not incorporate enough context for some situations.

While there certainly is value in making testbed measurement tools and methodologies available to the community the value of a centralized system for tracking these is not as compelling as for live Internet measurement data. First, we believe that by combining testbed measurements with live Internet measurements we would add confusion to the system. For instance, assume a researcher is attempting to download measurements of bulk transfers made with *ttcp* across the Internet. A search that resulted in a large number of *ttcp* measurements taken across a LAN would have to be further winnowed (either manually or using some heuristic) to get the set of Internet measurements the scientist wishes to analyze. Also, in most cases, networking measurements taken in testbed environments are highly reproduceable. Assuming scientists completely specify their experiments in a paper or report (as they should) another scientist can easily reproduce the measurement data. The tools used to make the measurements may be custom, but as long as the report gives a pointer to the location of these tools a second researcher should have little problem recreating the experiment in most cases. Therefore, we conclude that including testbed measurement data in SIMR is of only marginal benefit and so we assume that the system is for tracking only live Internet measurements in the rest of this paper.

This paper is organized as follows. § II discusses the overall architecture of the SIMR system. § III discusses the general characteristics of the databases that make up the central database in the SIMR system. Meanwhile, § IV – § VIII discuss each database in detail. § IX provides a general discussion on the various repositories in the system. § X outlines the measures the SIMR database will take to manage the meta-data and attempt to avoid database pollution. § XI discusses some security considerations for the system. Finally, § XII summarizes our conclusions and outlines future work in this area.

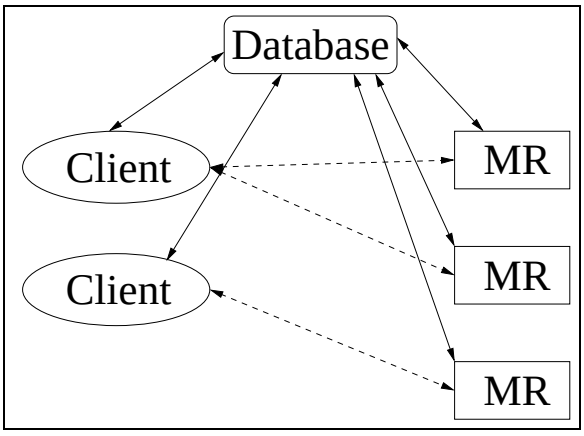

Fig. 1. SIMR system architecture.

II. Architecture

The SIMR system consists of three high-level components, as illustrated in figure 1. The *database* is at the center of the SIMR system. The database actually breaks down into five separate databases that track the measurements themselves, the users that take the measurements, the tools used to collect and analyze the data, descriptions of the experimental methodology and finally a database for tracking datasets. These various databases are discussed in § III – § VIII. The second major component is the measurement repositories (MR). These servers store the actual measurements collected and distribute this data to researchers upon request (see § IX). Finally, the clients shown in figure 1 represent researchers using their web browsers to interact with the system by making queries to the various databases and requesting measurements from measurement repositories based on the results returned by the database.

The design of SIMR is not necessarily novel. The architecture is similar to that of the music sharing Napster service [3] – where users query a central database of available songs and then retrieve those songs from other users who have made them available from their own hosts. We believe this approach suits the problem of distributing Internet measurement data well for several reasons, as discussed below.

An alternative to the architecture proposed in figure 1 would be a fully peer-to-peer system whereby each researcher makes their own database, measurements and tools available and then advertises the availability of this information in some way. We chose an architecture involving a centralized component for SIMR for several reasons. First, interacting with a single entity is likely to be easier and requires no special software for the users querying SIMR. Additionally, name clashes could be a concern if no central authority ensures the unique naming of measurements and tools. Next, a search for measurements on a distributed set of hosts may be skewed by sites that are not currently reachable (for whatever reason). We believe these problems could likely be addressed adequately with enough thought and work.

Our largest concern about using a distributed database is the quality of the measurements contained within the system. For instance, some measurement M may show a *ping* run with an average RTT of 100 ms if downloaded today. Requesting the same measurement tomorrow may return a *ping* with an average RTT of 150 ms. This change would likely go undetected unless the same researcher happened to download the same measurement twice and found a discrepancy. By using a central database the policy that meta-data *never* changes[2] can be enforced. Also, as outlined below, we force the meta-data for each measurement to include an MD5 hash [4] of the measurement itself. Therefore, if a measurement is changed in a researcher's repository there is a straightforward method for detecting the discrepancy between the meta-data and the measurement. Also, a central database can be configured to enforce strict rules about which meta-data must be provided en-

[2]We really mean that *most* of the meta-data does not change. For instance, the time a measurement was initiated should never change. But, a pointer to where the data is currently archived may change every once in a while.

suring that all measurements of a certain type have some minimal meta-data.

The overriding concern is that a distributed database opens up the case that a sloppy researcher can pollute the global dataset. And, while we could detect this after obtaining the meta-data and the measurements this requires unnecessary checks and resources for every researcher. With a more centralized approach researchers still have to be careful, but they can be assured that certain safe guards have been put into place. See § X for a discussion of the database's handling of records.

Next we turn our attention to the measurement/tool repositories. Distributed repositories serve to spread out the load on the SIMR system. One possible design would place the measurements on a central server with the database (similar to the Internet Traffic Archive [5]). However, that requires the central server to be able to store and serve a large amount of data which may not scale. Using distributed repositories offloads both of these requirements to the researchers wishing to make the measurements available or to third parties such as the Internet Traffic Archive whose mission is to serve these measurements. By distributing the repositories we could run into the problem of measurements and the meta-data in the database becoming unsynchronized. As discussed above, we mitigate this by including an MD5 hash of the measurement in the meta-data stored in the database. Therefore, a researcher can verify that the measurement and the meta-data correspond.

Finally, note that the SIMR system does not call for developing a specialized application layer protocol for its communication. We envision the measurement repositories being hosted on standard HTTP or FTP servers. The communication with the database will involve using standard HTTP and simple CGI scripts. While this may provide a less than optimal solution in some cases, the benefit in terms of accessibility is great. Researchers will not need specialized tools to access the database or the measurement repositories. In addition, users will be able to automate the downloading of measurements with the use of standard tools (e.g., a perl script that queries the database and retrieves new measurements using *wget* [6] every night).

III. Database Overview

The database is the key component of the SIMR system. The "database" is actually broken into five separate databases to track users, tools, measurements, experiments and datasets. *The key task with regards to the database (and the entire SIMR system) is fully specifying the meta-data required for each database entry such that other researchers will have the needed context for using the contents of the repositories for their own research.* In laying out the information to be kept in SIMR the touchstone we use is to minimize the amount of interaction between a researcher wanting to use some measurement and the researcher who submitted the measurement in question[3]. In the following sections we give a brief overview of the purpose of each database followed by a description of the meta-data held by the particular database in question.

The database entries will be exchanged using XML [7]. This provides a standard way for researchers to exchange and process the information. In addition, XML leaves room for extending meta-data as the needs of the community change. Finally, XML allows the database to easily check entries provided by researchers to ensure the required meta-data is provided.

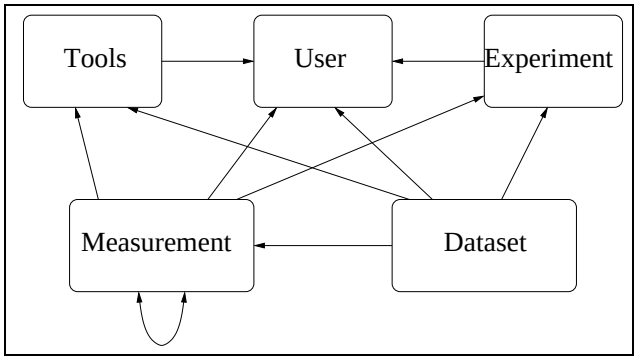

Fig. 2. SIMR database layout.

The overall layout of the various databases is shown in figure 2. The arrows in the figure represent pointers kept from one database to another. For instance, there is a pointer in the tool database to the user database. This indicates that each tool record tracks the person who submitted the tool. The interactions between the various databases are many and can get quite complicated. However, we believe the task of creating the meta-data for inclusion into SIMR can be automated fairly easily. And, as discussed further in § X some of the meta-data can likely be generated automatically from measurement output. Finally, all of the interactions shown in the figure will likely not apply to every measurement submitted to the system.

IV. User Database

The user database is kept to allow for the automated submission of information to the database by known scientists. In addition, each measurement (tool, etc.) can be attributed to a particular researcher. Thus, this attribution can be used as part of the search criteria when querying the database for certain kinds of measurements. Also, keeping track of which researchers submitted which measurements provides scientists an easy way to contact the originator of the data if questions arise. Finally, identifying the researcher who took a measurement can be used for attribution purposes. Each user record consists of the following fields:

- **Username** (required). A unique username that identifies a particular researcher. In various places within the database unique tags are required. The database itself will avoid possible clashes between user's naming conventions by prepending the tags supplied with submissions with the

[3]Ideally SIMR would contain enough context to avoid all communication about the measurements, although this seems like a very tall order in practice.

username of the person submitting the measurement or tool (so that, for instance, two different researchers could each independently submit a measurement with the tag "ping-2" without creating a collision).

- **Email Address** (required). An email address for contacting the originator of a particular measurement. In addition, this provides an easy way for SIMR to notify users of failures.
- **Comment** (optional). Generally the user's name, but could also contain other information (phone number, etc.).
- **PGP Key** (required). The user's public PGP[4] key. This allows the system to authenticate each user and their submissions to the database.
- **Administrative Information** (optional). This field appears in every database and is used to store administrative information that is not propagated to users of SIMR. This information is expected to be required to enforce SIMR policy (which is beyond the scope of this paper). For instance, a note that some user can submit notes for existing measurements but no new measurements (per some policy).

The administrator of the central database server will be charged with creating all records in the user database. In principle this could be an automated process whereby each user provides the necessary information and a record is created on-the-fly. Alternatively, the records could be created by hand by the administrator. Each method has its costs and benefits, but as these costs and benefits are largely nontechnical in nature this policy question is not further addressed in this document.

V. Tool Database

The tool database is used to track programs used to make and analyze network measurements. Tracking the tool used to take a particular network measurement is important so that known bugs in the utility can be taken into account when analyzing the data. In addition, different versions of the same tool may produce different output. Finally, including a pointer to the actual tool used to take a measurement allows others to use the same tool to extend the work by taking additional measurements. Each tool in the database will consist of the following fields:

- **Tool Tag** (required). Each version of each tool will have a unique tag (e.g., "tcpdump-3.5"). The database will prepend the tag with the username of the person submitting the record to avoid clashes between researchers. If multiple tools with the same tag are submitted by the same person the database will reject all but the first entry.
- **User** (required). The username of the researcher submitting the tool record.
- **Version** (optional). The version of the tool. While this field is optional, we *strongly recommend* that researchers fill in this field. We make this field optional because in some cases a researcher may not know the exact version of a particular tool that was used to take the measurement. For instance, if a researcher takes *tcpdump* packet traces on a mesh of hosts (e.g., NIMI [8], [9]), he or she may not know the version of the tool used on each host in the mesh (and, may not be able to easily find out).
- **URL** (required). The URL that a researcher can use to download the tool. We *highly recommend* that researchers mirror the tools they use in a reliable repository and list that URL in the tool record. (See § IX for a discussion on repositories.) Listing the official URL where one can download the current version of some tool may not help when a new version comes out or when the web site for that particular tool moves. Therefore, we recommend keeping the tools used to take the measurements and analyze the data in the same repository with the measurement.
In addition, SIMR should be flexible enough to allow for the use of automated gathering of tools for researchers. Therefore, the URL should yield the actual tool described in the record. In other words, the URL should not point to a web page that must be further navigated to obtain the desired tool since this would hinder automatic reaping of data.
- **MD5 Hash** (required). This is an MD5 hash of the file pointed to in the "URL" field. The distributed nature of the SIMR system means that tools and data can potentially change without corresponding changes to the database (i.e., a new version of a tool may have the same URL, but work in a drastically different manner). This is mitigated by keeping an MD5 hash of the tool in question so that researchers can detect tools that have changed since the database entry was submitted (which should not happen).
- **Notes** (optional). The notes field contains miscellaneous information about a particular tool. Notes can be used by the original submitter to pass along information known about a tool that could be useful to others (e.g., a known bug in running the tool under some operating system). In addition, notes can be added by any SIMR user at any time. This allows other scientists to note additional oddities they may find in the tools. Each note consists of a text string and a user ID.
- **Administrative Information** (optional).

There are a number of standard tools that are popular in the research community, such as *tcpdump* [10]. It is expected that the tool database will be seeded with several such tools, mirroring them on the database server. Therefore, researchers can use the standard entries without creating duplicates for their own measurements (while being assured that the tools are stored in a reliable place).

Adding tools to the database is somewhat problematic. As will be discussed in § VI, certain classes of measurements have their own specific set of meta-data. For example, a *tcpdump* measurement will likely have different meta-data than a *traceroute* measurement (in addition to sharing some common fields). When adding measurement tools to the database we must consider whether the tool is part of a current class of tools (e.g., "packet sniffers"). If it is not, a new specification of meta-data will need to be created to ensure SIMR captures the key meta-data for this new tool. On the other hand, analysis tools can be inserted

[4]See http://www.gnupg.org/ for information on PGP.

into the database without such rigorous checking. Therefore, we propose that the SIMR system will accept any tool from any user into the tool database. However, before measurements using a given tool can be added to the database the SIMR administrator (likely in consultation with the researcher adding the new tool) will have to "approve" the new tool, adding new meta-data requirements as needed. The exact mechanisms of this selection and approval are a policy issue, and will not be further discussed in this document. This rule is in place to ensure that all the meta-data for a particular kind of measurement is captured in SIMR.

VI. Measurement Database

The next portion of the database we discuss holds an entry for each measurement to be made available via SIMR. We expect that each invocation of a measurement tool will correspond to a single database entry. In addition, we include several methods for linking measurements together in the database (e.g., a *ping* executed immediately following a *traceroute*). We believe that by requiring each invocation of a measurement tool to be described separately in the database we can require the most accurate meta-data and provide researchers with powerful and accurate searching capabilities.

An alternative method for tracking measurements is to allow more free-form records to be submitted about measurements stored by a researcher in whatever form he or she feels is useful or convenient. For instance, a record could describe a set of *traceroute* measurements taken over some measurement infrastructure (e.g., NIMI) for a year, including a pointer to a tar file containing the output of thousands of *traceroutes*. However, ensuring that the database obtains accurate meta-data for these sorts of measurement sets is difficult. For instance, we would have to make sure that meta-data was available for each *traceroute* – and hence we would have to know exactly how to dig through the tar file. Further, it is highly likely that no two researchers would put their tar files and meta-data together in the same manner. Therefore, an automated method for verifying the contents of the tar file would be nearly impossible. On the other hand, accepting measurements as is without verifying the meta-data returns us to many of the problems outlined in the introduction. Therefore, we strive to nail down as much of SIMR's requirements and interactions as possible to allow the system to work with little or no human interaction. We believe the overhead of generating the meta-data is worth the long-term benefits of doing so.

However, we do make one concession due to the fact that we do not believe that every set of measurements can be neatly placed into the database. Therefore, we include a method for including a "mess" of measurements together in one record when the coupling pointers included in SIMR are not sufficient to capture the interactions of various measurements. However, we discourage researchers from using the "mess" type instead of taking the time to submit the measurements in a more coherent manner. (And, the SIMR system may do well to allow scientists to add "mess" measurements only with special permission.)

The next subsection details the common fields that all measurement records will contain. In addition, each type of measurement will require additional, unique meta-data. We examine several different kinds of measurement data and define meta-data for those examples in the following subsections.

Figure 3 shows an example of a measurement record that might appear in the SIMR system. We will refer to this example in the following subsections.

A. Common Fields

The following fields contain information that is common to measurements of any type and should be included in all records, as appropriate.

- **Record Type** (required). Each kind of measurement supported by the SIMR system will have a record type. In our example record the measurement type is *packet-trace*. While all record/measurement types will share a number of common fields in their database entry (as outlined in this subsection) the record type will define an additional set of meta-data that should be stored with a given measurement. The record type is closely related to the tool used to take the measurement, but is slightly more generic. For instance, a packet trace could be taken by any number of different tools (*tcpdump*, *ethereal*, etc.). However, since all packet traces have the same qualities regardless of the collection tool they can be grouped into one record type in SIMR (i.e., they will need the same meta-data).
The SIMR system will reject any database submissions with an unknown record type. Further, there should be no automatic way for users to enter new record types. Each record type and the fields it contains must be carefully considered such that all needed information is contained in each entry. Therefore, to keep the database of high quality the creation of a new record type must be a manual process – ideally a collaboration between a number of researchers – to ensure the database is not polluted with useless data.
- **Unique Tag** (required). This field is a string that is used to uniquely identify a measurement within the SIMR system. For instance, the unique tag in our example is "weddy-tcpd-snd-1011369283". Note that even though the tag contains information about the researcher who took the measurement (weddy) and the time the measurement was taken (1/18/2002 at 10:54 AM EST) this meta-information must still be submitted in the appropriate database fields. Encoding this information in filenames or the unique tag name of a measurement does not aid other researchers in searching the database and therefore is not sufficient for SIMR.
The purpose of the unique tag is to provide a way for researchers to discuss various measurements with their colleagues in such a way that everyone can easily find the same data. For instance, a paper might comment: "The weddy-tcpd-snd-1011369283 trace shows a particularly large number of retransmitted segments." and an interested researcher could easily download the measurement and have a look.

The uniqueness of the tags will be enforced by the SIMR system. As a first pass at avoiding clashes the user's ID will be prepended to all measurements submitted. Hence, each user ID has its own namespace and users need not worry about clashing with someone else's labeling scheme.

• **User** (required). The ID of the user that took this measurement. This field is a pointer into the user database and can be used to find other information about a user (e.g., their email address). The user "weddy" submitted the example record shown in figure 3.

• **Date** (required). This field represents the time the given measurement was initiated. The date will be represented as the number of seconds (and partial seconds) since January 1, 1970 (a "Unix timestamp"). In our example the timestamp the measurement was started is "1011369283.386290".

We *strongly encourage* researchers to collect highly accurate timestamps (as shown in the example record). However, this is not a hard requirement and courser grained timestamps will be accepted.

• **URL** (required). This field is a pointer to the location of measurement itself. Similar to the tool database, the URL should yield the actual measurement described in the record. That is, the URL should not point to a web page that must be further navigated to obtain the desired data since this would hinder automatic reaping of data.

• **MD5 Hash** (required). This field contains the MD5 hash of the measurement obtained by accessing the given URL. This information is required to ensure that the database and the repository stay synchronized and that the description of a measurement, in fact, correlates with the measurement at the given URL.

• **Tool** (required). This field represents a pointer to the entry in the tool database used to take the measurement. The string that appears in this field is the unique tool tag. The example in figure 3 indicates the measurement was taken with "mallman-tcpdump-3.4.8-2".

• **Supporting Software** (optional). In some cases measurements not only rely on the tool used to take the measurement, but also one or more pieces of supporting software. For instance, a customized kernel used to test some new TCP mechanism may be used in conjunction with *ttcp* to determine the usefulness of the proposed change to TCP. This field is used to point at entries in the tool database that describe this supporting software (e.g., patches to a FreeBSD 4.4 kernel). This field may appear multiple times in a single record.

• **Keywords** (optional). This field is a list of keywords that may help other scientists search for particular measurements. For instance, a *ttcp* connection that is using a custom kernel with the NewReno loss recovery strategy [11], [12], [13] may list "NewReno" in this field.

• **Notes** (optional). The notes field contains miscellaneous information about the particular measurement. Notes can be added by any SIMR user at any time. This allows other scientists to note oddities in the measurements. For instance, consider a packet trace from the link between some university and the wide area network. The researcher who took the measurement might be looking at the prevalence of TCP options. However, someone else may be looking at the duration of connections and notice that the clock on the tracing host jumps backward at some point in the trace. The second researcher can submit a note that will be attached to this measurement to alert others to this anomaly in the trace. Each note consists of a text string and a user ID. Note that claims made in the notes field will not be verified by anyone. Therefore, if a researcher believes that a note is false it may behoove the community for that researcher to add a new note stating this disbelief of the existing note.

In our example there are two notes added to the record. The first note (added by the originator of the record) indicates the number of packet drops reported by the kernel on the tracing machine. The second note, by a second scientist, indicates a clock jump was found in the trace.

• **Administrative Information** (optional).

In addition to the above data, each record in the measurement database contains a *host identifier* field that indicates where the measurement was taken. By providing the host on which a measurement was taken, researchers are providing additional search criteria for other scientists. For instance, if a researcher has determined that a given machine has a particular TCP bug that should be factored out a suitable search can be employed such that no measurements from the host in question are returned.

The host identifier specified here is not specific to the **measurement host** field, but rather is used anywhere the database requires host identification. The fields in the host identifier are:

• **IP Prefix** (required). This field indicates either the IP address or a reasonably specific IP prefix for the given host. An IP prefix of "132.235.1.1" indicates a specific IP address, while a prefix of "132.235.0.0/16" indicates the network to which the host is connected. We allow for researchers to use prefixes rather than specific IP addresses to mitigate security concerns that may arise in providing specific addresses (especially in conjunction with the information in the following fields).

In some cases a host may be located behind a Network Address Translator (NAT) [14]. In these cases, the IP address of the host may not indicate the network to which the host was actually attached. For instance, a laptop whose "home location" is at NASA GRC in Cleveland, Ohio taking a few measurements from a hotel network in California that performs NAT functions may assume that it is still on its home address. Additionally, some passive measurement hosts may not have IP addresses. For instance, an intrusion detection system may not have an IP address in an effort to escape attacks. In either of these cases, we urge researchers to determine the network prefix the host is attached to and enter that prefix into the SIMR system (along with a flag indicating that the IP address reported is not the actual IP address used, but does represent the network on which the host resides). This provides the community with some idea about where the host was located (network-wise).

• **Host Name** (optional). This field gives the fully-

qualified domain name (FQDN) of the host in question. When the name of a host is required for analysis of the data provided this field *must* be provided as DNS entries change and therefore we cannot count on the name/IP address mappings to be constant over time. However, it is expected that in many cases the name of the machine is immaterial to the measurement. Further, if the above field only provides an IP address prefix rather than a specific address, divulging the FQDN will defeat the purpose of specifying a prefix rather than an address. Therefore, we specify that host names beginning with a "." are domain names (for instance ".ohiou.edu").

• **Platform** (optional). This field is expected to contain a string that indicates the operating system and type of machine for the given host (e.g., "Solaris/2.8/SPARC"in our example). This can be useful to another researcher in trying to untangle the peculiarities in a given measurement. For instance, there are many known TCP bugs [15] and knowing the OS that was used to make a TCP transfer may help a researcher better analyze the behavior shown in a packet trace. This field is optional because there could be cases when the exact OS version or platform may not be known (and difficult to determine). However, populating this field is highly recommended in the general case.

• **Location** (optional). This field represents a record indicating the geographic location of a given host. The record contains fields for a *place* (e.g., "Ohio University"), a *city*, a *territory/province/state* and a *country* (which is expected to be from the standard list of country codes[5]). Researchers are encouraged to fill out as much of this record as possible (e.g., some countries may not have any sort of territories and therefore that field cannot be populated).

The next five fields represent pointers that allow measurement records to be linked together, as appropriate. These pointers are optional because every measurement does not need every pointer (or, any pointer, in fact). However, in some cases linking measurements can be crucial to providing the needed context for other scientists to use measurements. Therefore, we *strongly encourage* researchers to add these pointers to their submissions when appropriate.

• **Experiment** (optional). This field points to a record in the experiment database (see § VII) that explains the methodology used to gather this and other related measurements. In the example measurement record the measurement shown is a part of the "weddy-ping-ttcp-trials-1" experiment.

• **Association** (optional). This field represents a generic association between two measurements. For instance, assume a scientist used *ttcp* to transfer some amount of data over a TCP connection and traced that connection using *ethereal*. The same data transfer will appear as two records in SIMR (one for the *ttcp* output and one for the *ethereal* trace). The association pointer in the *ttcp* record should point at the *ethereal* record, and vice versa.
In figure 3 the generic association is set to "weddy-ttcp-1011369283" indicating the *ttcp* measurement that corresponds to the packet trace shown in the example.

• **Dependency** (optional). This field is used to show that one measurement is in some way dependent on another measurement. For instance, in [16] two bulk data transfers were conducted back-to-back. The second transfer may experience different network conditions because the first transfer "blazed a trail". Therefore, the record of the second transfer in the SIMR database should include a dependency pointer to the first transfer. In the example record the packet trace is dependent on a ping measurement ("weddy-ping-1011369270").

• **Derived-From** (optional). This field is used to indicate that a particular measurement is derived from some other measurement in the database. For instance, a given measurement might be end-to-end round-trip times taken from the last hop output of *traceroute* measurements. Or, a *tcp-reduce* measurement may have been derived from a *tcpdump* packet trace that is contained elsewhere in the database.

• **Distilled-To** (optional). This field is the converse of the previous field. This shows that the given measurement has been massaged into a different form and gives a pointer to the derived measurement. A record can have multiple "Distilled-To" fields. The example record in figure 3 shows that the packet trace has been distilled to some new form (likely *tcp-reduce*, based on the record name) by another researcher in record "elb-tcpred-snd-1011369283".

B. Simple Measurement Entries

In this section we will look at the meta data required for two "simple" measurement types: *traceroute* and a common WWW log file. These two measurement types are examples. SIMR will be required to contain many more types of measurements. Each type of measurement submitted to SIMR will likely need its own specific meta-data. The known measurement types will be listed, in detail, on the SIMR web page so that researchers do not re-invent types just because they are unaware that someone has already done the work. In addition, researchers should consult the list of required meta-data *before* running their experiments, such that they collect the required bits of ancillary information while the experiment is running.

B.1 Traceroute

For *traceroute* measurements we add two fields to the group of fields required for every measurement. The first additional, required field is a host identifier that indicates the host that is the ultimate **target** of the *traceroute*. In the case of *traceroute* we know the sender will be the same as the machine that took the measurement. The second additional, required field is the **maximum TTL** used by *traceroute*. This field indicates the maximum path length that *traceroute* will record. This allows a scientist to determine whether the entire path was determined.

[5] See http://www.iana.org/cctld/cctld-whois.htm for a list of country codes.

B.2 WWW Common Log File

For a common log format (CLF) file[6] we need to append two fields to the standard fields required by all measurements (defined above). The first field is the **duration** of the log file found in the SIMR system. This will be used by SIMR in searching the database for specific measurements. For instance, say a researcher requests log files from Saturdays in March. SIMR has to not only know when the log file was started (which is accessible from the generic "Date" field), but also the timeframe encompassed by the log file to satisfy the search request. The second added field, denoted **anonymized**, indicates whether the log file has been anonymized to protect the identity of the clients accessing the web server that made the log file.

C. Packet Traces

Storing enough context about packet traces to be useful to other researchers is a difficult task due to the vast differences in the contents of packet traces. [1] gives an analysis of packet traces taken on a moderately busy production web server. On the other hand, [17] analyzes packet traces (both sender-side and receiver-side) of synthetic TCP connections involving a mesh of hosts. SIMR needs to be flexible enough to accommodate both of these types of packet dumps by storing enough meta-data that the traces can be effectively used by other scientists (with little or no communication with the researcher who took the original measurement).

First, we define several fields that must be stored in every packet trace record:

• **Synthetic Traffic** (required). This field indicates whether the traffic contained in the trace file is synthetic or whether the trace is simply a passive look at traffic as it happened "in the wild". The packet trace shown in figure 3 is of synthetic traffic.

• **Duration** (required). This field indicates the length of the trace file (in seconds). This allows searching for traces based on length as well as based on the time encompassed by the trace. In our example the trace file spans just over 63 seconds.

• **Packet Filter** (required). This field defines the packet filter used to take the packet trace measurement. For instance, a *tcpdump* filter of "tcp and port 1234" would indicate the trace contains all TCP traffic to or from port 1234 observed. This field is required, but may be blank if no packet filter was used (i.e., the measurement contains all traffic on the observed link). In the example record presented in figure 3 the packet filter used is listed as "tcp and port 5555 and host 192.55.91.71".

• **Snapshot Length** (required). This field defines the maximum number of bytes saved for each packet in the trace. This can be useful, for example, in searching for traces that include enough packet header to contain the TCP options. In our example record the snapshot length is set to 120 bytes.

• **Anonymized** (required). This field indicates whether the addresses appearing in the trace have been anonymized. The measurement in our example record has not been anonymized.

• **Vantage Point** (required). This field indicates where the packet trace was taken: near a host acting as a data *sender*, near a host acting as a data *receiver*, near a host acting as *both* a sender and a receiver, or in the *middle* of the network. The example record given in figure 3 indicates the packet trace was taken near the data sender.
Depending on the application being traced a clear sender or receiver may not be present (e.g., a packet trace of a voice conversation between two people across the network). We suggest that researchers not think of these fields as absolutes. For instance, consider an HTTP connection where one host sends a small request and receives a large amount of data in return. We can think of the host that sent the request as the "receiver" since it received the large majority of the data bytes even though it did send *some* data bytes (the request).

For many packet traces the above general meta-data is all that can be reasonably specified about the trace in the SIMR system. However, for traces of packet streams between two known endpoints we can include additional meta-data in the database to aid other researchers in searching for and using the measurement data. These additional fields are:

• **Hosts** (required). These fields indicate the two hosts that are involved in a flow. Our example shows that the first host has an IP address of "192.55.91.71" which resolved to the FQDN of "porsche.grc.nasa.gov" at the time the measurement was taken. The second host has an IP address of "132.235.1.1" and is running the Linux operating system.

• **Sender** and **Receiver** (optional). These flags indicate the sender and receiver of the data stream in the flow. As discussed above (see "vantage point") there is not always a clear cut sender or receiver. Therefore, we allow for these fields to be omitted. We note that in our example record the data sender is "192.55.91.71" and the data receiver is "132.235.1.1".

• **Single Conversation** (required). This field indicates whether the packets in the trace can all be considered as from the same conversation (e.g., TCP connection)[7]. This may help others find measurements they are interested in (e.g., if they are more interested in finding traces of single bulk transfers than of web pages retrievals that span multiple TCP connections). The example packet trace record shown in figure 3 indicates that the packet trace contains traffic from one conversation.

• **Packet Trace Association** (optional). This field is a pointer that can be used to point to other packet trace records that contain a trace of the same conversation(s) from a different vantage point. For instance, a trace taken near the data sender might include a pointer to a trace of

[6] http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format

[7] We deliberately did not use the work "connection" in this field as a group of UDP packets (for instance) between two processes on two different hosts should qualify as a conversation.

the same conversation taken near the receiver (as is shown in figure 3). This would allow for easy pairing of measurements for better analysis (e.g., see [18] for a study involving packet traces from both the data sender and the data receiver).

VII. Experiment Database

The experiment database is meant as a place where descriptions of a set of measurements can be held. This allows other researchers to more quickly and easily untangle the potentially messy set of pointers attached to a group of measurements. The experiment database contains three fields, as follows:

- **User** (required). A pointer to the researcher in the user database that submitted the measurements and tools for a particular experiment.
- **Experiment Tag** (required). This is a unique tag that identifies a particular experiment. As with all other tags in the SIMR system the first portion of the tag is the researcher's ID.
- **Description** (required). A description of the details of the experiment. For instance, a simple set of *ping* and *traceroute* measurements might have an experiment record containing: "This set of measurements contains back-to-back *ping* and *traceroute* measurements taken between NASA GRC and Ohio University over the course of 7 days. Each measurement consists of 10 pings, separated by 1 second, followed by a *traceroute*. The time between measurements is determined using a Poisson process with a mean of 60 seconds."
- **Notes** (optional). This field represents a set of notes that researchers can add to the experiment database.
- **Administrative Information** (optional).

Some experiments may not require a description in the experiment database. For instance, a simple set of independent *traceroute* measurements taken over the course of a month likely does not require an elaborate description.

VIII. Dataset Database

The final database contained in the SIMR system is a "dataset" database. This database contains pointers to records in the user, tool, experiment and measurement databases and is meant to allow for easy tracking of everything involved in a particular study/project. For instance, a researcher could construct a dataset record that outlines all the measurements and tools involved in a particular paper or presentation and then add a footnote to a paper indicating that the dataset used in the paper is available from SIMR (with a pointer to the dataset tag). A second researcher could then access this data to try out new ideas that extend the original work. Hence, SIMR provides the ability to conduct an "apples to apples" comparison of two mechanisms.

The fields contained in this database are:

- **User** (required). The user ID of the researcher putting together the dataset.
- **Dataset Tag** (required). A unique tag that is used to reference the record. For instance, "elb-pam-2002-1" might be used for the record holding a dataset used in a PAM 2002 paper. As with all other tags in the SIMR system the first portion of the tag is the researcher's ID.
- **Tool** (optional). This field may appear any number of times and contains a pointer to the tool database for a particular tool used in either the gathering or the analysis of the dataset.
- **Experiment** (optional). This field may appear any number of times and contains a pointer to the experiment database indicating the measurement methodology.
- **Measurement** (optional). This field may appear any number of times and contains a pointer to a measurement used in this dataset.
- **Usage** (optional). This field is used to indicate any particular directions that a scientist may need to run a similar set of measurements or run the analysis of the dataset (e.g., "run the *go* script to perform the analysis and then the *plot* script to generate the figures").
- **Paper** (optional). If this dataset is presented in a paper the paper's full bibliographic entry should be included in this record. This field may be repeated.
- **Notes** (optional). This field represents notes that researchers wish to add to this dataset.
- **Administrative Information** (optional).

IX. Repositories

The architecture presented in § II calls for repositories for measurements and tools. These repositories are servers that can be accessed by a URL that is stored in the measurement database (e.g., HTTP [19] or FTP [20] servers). While the system presented in this paper allows for repositories to be anywhere on the network we encourage researchers to take efforts to make sure that the URLs exported to the central database are *stable and reliable*. There may be a continued role for WWW sites such as the Internet Traffic Archive [5] even with the system proposed in this paper, to ensure that a stable and reliable location for measurement tools and results is available to researchers who may not be able to provide such a service themselves. As outlined in § X, SIMR will take some efforts to ensure its databases are not overly polluted with invalid or persistently unavailable measurements.

Using the system presented in this paper, researchers are encouraged to provide references to the tools used to both take their measurements and analyze the data. If a standard tool is used and a *stable and reliable* URL for the particular version of the tool is available (for instance, a particular version of *tcpdump* at http://www.tcpdump.org/) the researcher may use that URL. However, if a tool was written or modified for the measurements in question the researcher is strongly encouraged to make the tool available on their own server. In addition, we suggest that any programs or scripts used to analyze the data also be placed in the repository (and the tool database, as discussed in § V).

The storage format of the measurements is immaterial to the system itself. As long as the database contains an appropriate type for the measurement, the database can provide access to the measurement results. However, we

```
<measurement>
  <type>packet-trace</type>
  <utag>weddy-tcpd-snd-1011369283</utag>
  <user>weddy</user>
  <date>1011369283.386290</date>
  <url>http://irg.cs.ohiou.edu/~weddy/ttcpdata/weddy-tcpd-snd-1011369283</url>
  <md5hash>973c15eb4225c2cc2f7210c3432274ac</md5hash>
  <tool>mallman-tcpdump-3.4.8-2</tool>
  <supporting-software>elb-freebsd-4.4-56</supporting-software>
  <notes>
    <note 1>
      <user>weddy</user>
      <contents>
        tcpdump reported 3 kernel drops during this trace
      </contents>
    </note 1>
    <note 2>
      <user>elb</user>
      <contents>
        The timestamp goes backward at about 6 seconds into the
        trace.
      </contents>
    </note 2>
  </notes>
  <meas_host>
    <ip-prefix>192.55.91.0/24</ip-prefix>
    <fqdn>.grc.nasa.gov</fqdn>
    <location>
      <place>NASA Glenn Research Center</place>
      <city>Cleveland</city>
      <territory>Ohio</territory>
      <country>us</country>
    </location>
    <platform>Solaris/2.8/SPARC</platform>
  </meas_host>
  <experiment-ptr>weddy-ping-ttcp-trials-1</experiment-ptr>
  <association>weddy-ttcp-1011369283</association>
  <dependency>weddy-ping-1011369270</dependency>
  <distilled-to>elb-tcpred-snd-1011369283</distilled-to>
  <packet-trace>
    <synthetic>yes</synthetic>
    <duration>63.238545</duration>
    <filter>tcp and port 5555 and host 192.55.91.71</filter>
    <snaplen>120</snaplen>
    <anonymized>no</anonymized>
    <vantage-point>sender</vantage-point>
    <host1 opts=sender>
      <ip>192.55.91.71</ip>
      <fqdn>porsche.grc.nasa.gov</fqdn>
    </host1>
    <host2 opts=receiver>
      <ip>132.235.1.1</ip>
      <platform>Linux/2.2.12/i386</platform>
    </host2>
    <single-conv>yes</single-conv>
    <packet-trace-assoc>weddy-tcpd-rcv-1011369283</packet-trace-assoc>
  </packet-trace>
</measurement>
```

Fig. 3. Example entry in the measurement database.

would encourage researchers to provide as much information as possible. For instance, one could use *tcp-reduce* [21] to distill information from packet traces and provide that output to the community. However, in applying such a transformation to the packet trace some information is lost. For instance, *tcp-reduce* does not report options found in TCP SYN segments – which may be useful to another researcher. Another advantage of providing the raw output of the measurement tool is that the community can verify that the data was not changed in some way during the massaging process. At a minimum we suggest researchers provide the community with the measurements in the form they used for their analysis. In other words, providing the community with *tcp-reduce* output when the raw *tcpdump* output was used in the original analysis does not offer a direct method to validate the results obtained or extend the work using the same analysis tools.

X. Managing Meta-Data

This section discusses several small items that must be managed by the database server with regards to the meta-data. The key goal of the mechanisms we discuss is to avoid *database pollution*, which means that the number of useless records in the database is a significant portion of the number of overall records. This will end up causing researchers to sift through the measurements by hand and will ultimately reduce the viability of SIMR as a way to find useful Internet measurement data. Therefore, the SIMR system should do everything possible to guard the databases against *pollution* – including erring on the side of manual intervention. Even if an administrator must intervene to ensure the meta-data is of high quality, that ends up saving each scientist using SIMR time in the long run.

A. Submissions

The server must ensure that only measurements from known researchers are accepted into the system. This is accomplished by requiring all submissions to be signed with the user's PGP key (the SIMR user database keeps a copy of each user's public key). Accepting entries from anyone could open the database to (*i*) *pollution* – whereby sloppy entries end up obfuscating coherent entries to the point where the database becomes useless to researchers, and (*ii*) denial of service attacks – where some attacker fills the database with junk records.

B. Meta-Data Validation

The database is vulnerable to sloppy researchers inserting data into the database that is wrong. In many cases, we believe that there is nothing that can be done about this situation. For instance, if a researcher notes that a *ping* measurement was run at 2 AM and it was really executed at 1 AM, how can SIMR figure this out? However, we believe that there are several mechanisms that can be implemented to help prevent database pollution, as follows:

- Ensure that required fields are included in the submitted records. Allowing records that lack required meta-data weakens the global dataset and does not aid in sharing useful Internet measurements.
- In some cases SIMR can weed out obviously bogus meta-data. For example, timestamps of zero or timestamps in the future can easily be flagged and cause the record to be excluded from the database. Another case would be when a IP prefix of "139.0.0.0/8" is given. Since there have been no "/8" allocations of IP addresses in this range SIMR can quickly determine that this prefix is unnecessarily large and will likely provide no useful information to other scientists. There are many cases where such clear-cut validations can be easily applied.

Another class of validations may involve some heuristics. For instance, measurements that purport to be older than some threshold could be held for approval before inserted into the database. This class of validations likely ends up involving specific policy decisions about what data to accept, what to reject and what to hold for moderation.

- Some measurements lend themselves to some easy validation by analyzing the measurement. For instance, packet traces generally contain a timestamp for each packet. So, if the time a packet trace was reported to have started is significantly different from the time the first packet was recorded in the trace a flag could be raised[8]. There are a number of these validations that SIMR may be able to preform in an attempt to keep the quality of the database high.

Further, if we can use tools to validate certain measurements by looking at the measurement data itself, we could use similar tools to help researchers generate the meta-data for their submissions. This would aid researchers in putting together their submissions, as well as improve the accuracy of the submissions.

C. URL Verification

As discussed above, each measurement and tool entry submitted must contain an MD5 hash of the file in question. This ensures that the meta-data submitted can be exactly synchronized with the measurement data or tool later. When a new measurement is submitted, the server should request the measurement from the repository to check that the URL given is valid. In addition, the server can then check to ensure that the MD5 hash is correct. If either of these actions fails, the entry can be rejected (with an email to the originator). (This task could also be offloaded to another host if the load on the database server itself is an issue.)

Verification of all measurement and tool availability should be completed periodically by the database server (or a delegate) with failures resulting in records being marked as "invalid" – but not removed from the database[9]. This

[8]Such a situation could be legitimate if, for example, the packet sniffer is looking for rare network events. However, in the general case, this likely indicates a problem in the meta data.

[9]Note that the check for availability, while not fully specified here, should be robust to temporary network outages. Records should only be marked as "invalid" after requests for those measurements have failed for a sufficiently long time ($O(weeks)$) or after a successful retrieval and then a failed MD5 check.

verification will help to keep the database up-to-date and minimize the failures the user's of the system will experience. This process also reinforces the need for researchers to find reliable servers for their data.

XI. Security Considerations

SIMR is meant for the public dissemination of measurement data. Hence, security of measurement data (or meta-data) is not a design goal. However, one could envision an access control list (of PGP public keys, for instance) embedded in the meta-data for each measurement if privacy of measurements is required. The one facet of SIMR that does include security is submission. When submitting a measurement (or tool, note, etc.) a researcher will be required to produce a known credential so that the system knows how to name the incoming measurements and knows they are coming from a valid source. Opening up the database for unlimited write access by anyone would open the system up to an attacker filling the database with worthless records. In addition, requiring a credential to submit measurements provides another disincentive for "cluttering" the database with useless data. For these reasons, SIMR will only be available to update by known users.

Of additional concern is a malicious user submitting a Trojan horse to the tool database. For instance, an attacker could write a version of *ping* that takes network measurements and appears to be working "normally" while gathering information about the victim's system and sending that information back to the attacker or opening a backdoor into the victim's system that can be exploited at a future point. This would be especially concerning for binary measurement or analysis tools submitted to SIMR. Therefore, we *strongly encourage* researchers to submit tools in the form of source code.

Additionally, work on "safe" measurement techniques should be undertaken by the community. An example of this is *pcapd* [22], [23], [9]. This tool allows the granting of access to a packet filter to specific users for some specific purpose. So, some users would be allowed to watch all ICMP packets while other users might be limited to UDP packets sent to/from port 5454. If a researcher could setup *pcapd* before running someone else's packet capturing tool the researcher could ensure that the packet capture tool was only observing appropriate traffic. These sorts of tools that allow scientists to bound unknown measurement tools to known operations will aid in researcher's ability to trust the tools downloaded from SIMR.

While infrastructure like *pcapd* may help alleviate some of the issues with running other people's measurement gathering software, these sorts of systems do not help when running other people's analysis code, which presents a much trickier and more difficult problem. One possible way to mitigate the security implications of such code is to run the analysis in a "sandbox". For instance, using the Restricted Korn Shell[10] will prevent the tools downloaded via SIMR from accessing critical data on a researcher's system and running arbitrary tools on that system.

The SIMR system itself could be extended in the future to include meta-data regarding the safeness of the tools. A field in the tool database may convey whether a tool is "provably safe". Or, the community may be able to rate the safeness of a tool, with the overall score being stored in SIMR for other's to consult before running a particular piece of code. However, in our opinion, while we may be able to somewhat mitigate the security problems associated with running other people's code, each researcher will have to remain vigilant when executing code that they did not write[11].

XII. Conclusions and Future Work

In this paper we have outlined a new architecture for making network measurements available to the community for verification and re-examination. The benefits of such a system are numerous. We do not claim that the system presented in this paper is exactly right or covers every possible contingency. However, we hope that we have provided a starting point for a community discussion on this topic, such that a SIMR-like system can be implemented to aid in network measurement efforts in the future.

In addition, we feel that the system can likely be extended to track other sorts of data. For instance, the networking community may also benefit from a system for tracking simulation scripts and the tools used to analyze the subsequent simulation runs. Finally, non-networking data also could benefit by using a system such as the one presented in this paper. For instance, a database for keeping track of physics results or photos taken by a Mars rover could be useful for other communities. While this is clearly out of scope of this paper we note that all that would really have to change would be the definitions of the meta-data that is kept about each record.

Acknowledgments

This work benefited from discussion with Joseph Ishac and Vern Paxson. In addition, several items that SIMR tries to address were first outlined in [2].

[10] http://web.cs.mun.ca/~michael/pdksh/

[11] And, in fact, some of us need to be careful when using code that we do write!

References

[1] Mark Allman, "A Web Server's View of the Transport Layer," *Computer Communications Review*, vol. 30, no. 5, pp. 10–20, Oct. 2000.

[2] Vern Paxson, "Some Not-So-Pretty Admissions About Dealing With Internet Measurements," May 2001, Invited talk at the Workshop on Network-Related Data Management (NRDM 2001).

[3] "Napster Homepage," http://www.napster.com/.

[4] Ronald Rivest, "The MD5 Message-Digest Algorithm," Apr. 1992, RFC 1321.

[5] Vern Paxson, "Internet Traffic Archive," http://ita.ee.lbl.gov/.

[6] "Gnu wget," http://www.gnu.org/.

[7] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)," Tech. Rep., World Wide Web Consortium, Oct. 200.

[8] Vern Paxson, Jamshid Mahdavi, Andrew Adams, and Matt Mathis, "An Architecture for Large-Scale Internet Measurement," *IEEE Communications*, 1998.

[9] Vern Paxson, Andrew Adams, and Matt Mathis, "Experiences with NIMI," in *Proceedings of Passive and Active Measurement*, 2000.

[10] "*tcpdump* Homepage," http://www.tcpdump.org.

[11] Janey Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP," in *ACM SIGCOMM*, Aug. 1996.

[12] Kevin Fall and Sally Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *Computer Communications Review*, vol. 26, no. 3, July 1996.

[13] Sally Floyd and Tom Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," Apr. 1999, RFC 2582.

[14] Kjeld Borch Egevang and Paul Francis, "The IP Network Address Translator (NAT)," May 1994, RFC 1631.

[15] Vern Paxson, Mark Allman, Scott Dawson, William Fenner, Jim Griner, Ian Heavens, Kevin Lahey, Jeff Semke, and Bernie Volz, "Known TCP Implementation Problems," Mar. 1999, RFC 2525.

[16] Mark Allman, "Measuring End-to-End Bulk Transfer Capacity," in *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

[17] Vern Paxson, "End-to-End Internet Packet Dynamics," in *ACM SIGCOMM*, Sept. 1997.

[18] Vern Paxson, "Automated Packet Trace Analysis of TCP Implementations," in *ACM SIGCOMM*, Sept. 1997.

[19] R. Fielding, Jim Gettys, Jeffrey C. Mogul, H. Frystyk, and Tim Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," Jan. 1997, RFC 2068.

[20] Jon Postel and Joyce Reynolds, "File Tranfer Protocol (FTP)," Oct. 1985, RFC 959.

[21] Vern Paxson, "tcp-reduce," 1995, http://ita.ee.lbl.gov/html/contrib/tcp-reduce.html.

[22] Vern Paxson, "Personal Communication," Jan. 2002.

[23] Jose Gonzalez, "pcapd, work in progress," 2002.