GWD-R-P
Category: candidate Proposed Recommendation
Network Measurements Working Group
http://www-didc.lbl.gov/NMWG/

Bruce Lowekamp, College of William and Mary
Brian Tierney, Lawrence Berkeley National Lab
Les Cottrell, Stanford Linear Accelerator Center
Richard Hughes-Jones, University of Manchester
Thilo Kielmann, Vrije Universiteit
Martin Swany, University of Delaware

24 May ~~18~~ 2004
http://www-didc.lbl.gov/NMWG/docs/draft-ggf-nmwg-hierarchy-04~~2~~.doc

A Hierarchy of Network Performance Characteristics for Grid Applications and Services

Status of This Memo

This document provides information to the community regarding proposed standards for describing network measurements taken in Grid environments. Distribution of this document is unlimited.

Copyright Notice

Abstract

This document describes a standard set of network characteristics that are useful for Grid applications and services as well as a classification hierarchy for these characteristics. The goal of this work is to identify the various types of network measurements according to the network characteristic they measure and the network entity on which they are taken. This document defines standard terminology to describe those measurements, but it does not attempt to define new standard measurement methodologies or attempt to define the best measurement methodologies to use for grid applications, however it does attempt to point out the advantages and disadvantages of different measurement methodologies.

This document was motivated by the need for the interchange of measurements taken by various systems in the Grid and to develop a common dictionary to facilitate discussions about and specifications for measurement systems. The application of this naming system will facilitate the creation of common schemata for describing network monitoring data in Grid Monitoring and Discovery Services, and thus help to address portability issues between the wide variety of network measurements used between sites of a Grid.

Table of Contents

~~GWD-R-P~~ ~~Bruce Lowekamp, College of William and Mary~~
~~Category: Proposed Recommendation~~ ~~Brian Tierney, Lawrence Berkeley National Lab~~
~~Network Measurements Working Group~~ ~~Les Cottrell, Stanford Linear Accelerator Center~~
~~http://www-didc.lbl.gov/NMWG/~~ ~~Richard Hughes-Jones, University of Manchester~~
~~Thilo Kielmann, Vrije Universiteit~~
~~Martin Swany, University of Delaware~~
~~7 January 2004~~
~~http://www-didc.lbl.gov/NMWG/docs/draft-ggf-nmwg-hierarchy-01.doc~~
~~candidate~~ ~~18~~
GWD-R-P (candidate Proposed Recommendation) 24 May 2004

1. Introduction

This document ~~describes~~ proposes a standard set of network characteristics and a classification hierarchy for these characteristics that will be useful for Grid applications and services. The goal of this work is to identify the various types of network measurements according to the network characteristic they measure and the network entity on which they are taken. This document was motivated by the need for the interchange of measurements taken by various systems in the Grid and to develop a common dictionary to facilitate discussions about and specifications for measurement systems. The application of this naming system will facilitate the creation of common schemata for describing network monitoring data in Grid Monitoring and Discovery Services, and thus help to address portability issues between the wide variety of network measurements used between sites of a Grid.

The nomenclature and hierarchy presented in this document addresses the first step of this process by providing a common dictionary of terms and relationships between commonly used measurements. The hierarchy allows measurements to be grouped according to the network characteristic they are measuring. Fully achieving the goals of measurement portability will require one or more schemata and protocols to be developed. The hierarchy is specified separately to allow it to be used as the nomenclature for multiple schemata and other specifications, regardless of the origin of those specifications.

The NMWG focuses on existing and currently used measurements to determine network performance. This document defines standard terminology to describe those measurements. It does not attempt to define new standard measurement methodologies or attempt to define the best measurement methodologies to use for grid applications, however it does attempt to point out the advantages and disadvantages of different measurement methodologies. The results from some measurement tools are influenced by bottlenecks at the hosts making the measurements. We work to identify these effects in this document, but a more thorough approach to such bottlenecks is the focus of the Internet2 End-to-end Performance WG [16] and there is also related work from the European Union DataGrid and DataTAG projects [32].

The NMWG is closely related to the IETF Internet Protocol Performance Metrics (IPPM) WG. Whereas their work focuses on best-practices for network engineers and defining the correct way to take measurements, our goal is creating a comprehensive system that can be used to categorize all measurements that are in use, taking into account the requirements of grid applications and grid-related tools. Where possible, we adopt the terminology defined in the IPPM Framework [23], although due to the different goals of NMWG and IPPM, certain sections of that framework do not apply to this document. We have also tried to incorporate and clarify common terminology where possible (e.g. in the meanings and use of terms such as “node” and “host”, “path”, “hop” and “link” and the various terms for “bandwidth”, “capacity”, “throughput” etc.).

The NMWG is engaged in ongoing work to develop suitable schemata to enable the observations to be published and retrieved within the Grid environment. We believe that this document defines those characteristics and entities that are generally of interest to those in the grid measurement community and that it has the support of that community. However, we anticipate that changes may be required based on the experiences gained from the NMWG’s work and other implementations based on this hierarchy.

1.1 Notational Conventions

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” AND “OPTIONAL” are to be interpreted as described in RFC2119.

Because the goal of this work is to facilitate the portability of all measurements that are actually taken, rather than to specify only the correct way to take measurements, this document rarely specifies absolute requirements such as "MUST." Were the authors to specify the proper way to take measurements, this document would list REQUIRED, RECOMMENDED, and OPTIONAL attributes for each measurement. However, as our goal is to represent all measurements people take, even those that are technically incomplete, we may recommend the proper attributes for a measurement, but we only require attributes where they are strictly necessary to describe the intended meaning of the measurement. We expect that implementations and schemata that use this hierarchy will introduce stricter requirements according to their needs.

2. Sample Grid use of network measurements

As an example of how network measurements could be used in a Grid environment, we use the case of a Grid file transfer service. Assume that a Grid Scheduler [13] determines that a copy of a given file needs to be copied to site A before a job can be run. Several copies of this file are registered in a Data Grid Replica Catalogue [6], so there is a choice of where to copy the file from. The Grid Scheduler needs to determine the optimal method to create this new file copy, and to estimate how long this file creation will take. To make this selection the scheduler must have the ability to answer these questions:

- what is the best source (or sources) to copy the data from?
- should parallel streams be used, and if so, how many?
- what TCP window and buffer size should be used?

Selecting the best source to copy the data from requires a prediction of future end-to-end path characteristics between the destination and each possible source. Accurate prediction of the performance obtainable from each source requires measurement of available bandwidth (both end-to-end and hop-by-hop), latency, loss, and other characteristics important to file transfer performance.

Determining whether there would be an advantage in splitting up the copy, and, for example, copying the first half of the file from site B and in parallel copying the second half of the file from site C, requires hop-by-hop link availability information for each network path. If the bottleneck hop is a hop that is shared by both paths then ~~there is no advantage to~~the transfer time cannot be reduced by splitting up the file copy in this way.

Parallel data streams will usually increase the total throughput on uncongested paths, as shown by Hacker et al. [15]. However, on congested hops, using parallel streams may just make the problem worse. Therefore, further measurements, such as delay and loss, are needed to determine how many parallel streams to use.

Even in the case of a single stream, accurate network measurements can be used to greatly improve performance and host resource allocation. TCP has always used a "prediction" (or smoothed estimate) of the RTT to determine timeouts. Recent work in the Web100 [29] project aims to improve these estimates and make more information available to the TCP stack and to external programs.

2.1 How this recommendation helps

When a distributed application is designed, the designer makes decisions about what options the application has for adapting to the network, how to make a decision between them, and what measurement tools to use. Typically, an application-layer interface, such as that provided by the Network Weather Service (NWS) [31], provides the necessary information.

Currently a variety of APIs exist for collecting network information from network tools and Grid information services, none of which are compatible. This recommendation provides a way for both network tool and grid application designers to agree on a common classification of measurement observations taken by various systems and desired by applications. It presents a necessary basis for defining a set of schemata for disparate measurement data that allows for its discovery and presentation by a general-purpose Grid information systems interface such as that described in [28]. The nomenclature of this recommendation is as well-suited for annotating information contained in other schemata as it is for developing a new schema.

The natural strides of research support the development of multiple network measurement systems, the development of newer measurement tools, and the cooperation of various groups to share deployed infrastructure. Even with the cooperation of various groups building Grids, there will be different monitoring systems developed. The same monitoring system may be deployed using probes with different parameters. The development of new techniques as well as different needs requiring different tools or parameters will continue to guarantee the need for many different performance monitoring systems.

This recommendation is not an attempt to unify all measurement systems under a specific set of measurements, nor is it an attempt to define standard measurement techniques for everyone to use. Instead, the nomenclature is aimed at allowing monitoring systems to classify the measurements they take. The independent development and deployment of the monitoring systems can continue, but the classification of the measurements they take will allow that information to be used in other ways.

With this nomenclature, measurements can be classified according to their methodology, the characteristic(s) that they measure, and the entities being measured. Any system familiar with the specific measurement methodology will be able to use the measurements as intended. Systems not familiar with that particular methodology, but which support~~s~~ this nomenclature, can treat them as generic measurements of that particular characteristic. By maintaining both the original measurement and a generic classification, maximum information is available according to another system's ability to interpret it.

Full realization of measurement portability requires continued work towards the development of dictionaries, schemas, and protocols. We expect many such standards to follow through the work of the NMWG and other GGF groups. We submit this nomenclature to the Grid and Internet communities as a recommended hierarchy of the types of measurements in use, to provide an important base-class for future progress towards measurement portability.

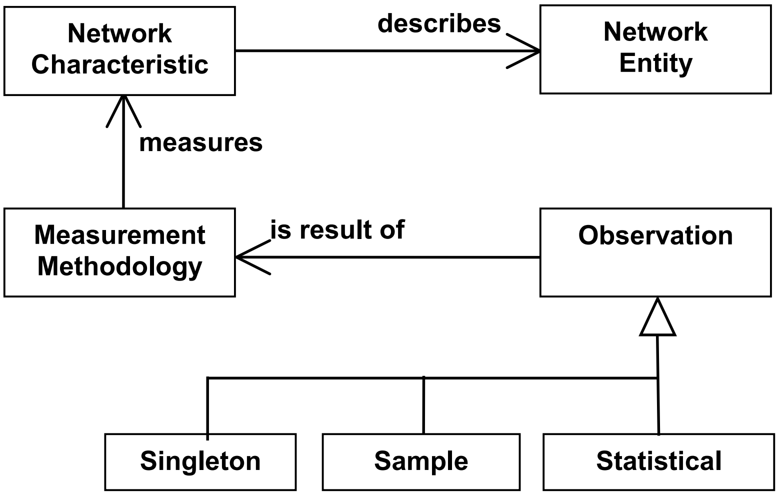

Figure 1: The relationship between the terms used in this document.

3. Terminology

The different research backgrounds of people working with networks and Grids necessarily bring slightly different terminology. To enable discussion and definition of network measurements we will define three terms, ranging from general to specific:

- *Network characteristics* are the intrinsic properties of a portion of the network that are related to the performance and reliability of the network.
- *Measurement methodologies* are the means and methods of measuring those characteristics.
- An *observation* is an instance of the information obtained by applying the measurement methodology.

The relationship between these terms is illustrated in Figure 1. We use UML notation for the diagrams in this document with the arrows with open triangles indicating an inheritance, or an "is-a" relationship, and the open arrows indicating an attribute.

Many network characteristics are inherently hop-by-hop values, whereas most measurement methodologies are end-to-end. Therefore, what is actually being reported by the measurements may be the result for the smallest (or "bottleneck") hop. In this document we distinguish between links (hop-by-hop) and paths (end-to-end) where appropriate. As illustrated in Figure 1, characteristics are applied to Network Entities, which is a general term that encompasses nodes, paths, autonomous systems, hops, etc.

> *Aside: We considered the terminology specified by the IETF IPPM Framework [23]. However, in actual use we found that it wasn't ideally suited for our application. We have defined the distinction between metric and measurement more strongly than in the IPPM framework, due to our desire to develop a hierarchy between the various characteristics and measurements, rather than simply establishing a flat dictionary of terms. IPPM has defined multiple "metrics" where our definitions would indicate only one characteristic, or subsets of a common characteristic, for instance "loss rate" versus "loss pattern". Furthermore, discussions with IPPM contributors have indicated some questions as to what the differences between metrics and measurement methodologies are. Our definitions also led to practical difficulties, where people who were unfamiliar with our*

vocabulary would misinterpret our use of the term "metric." In practice, we find that characteristic, measurement methodology, and observation are rarely misinterpreted, therefore we prefer that terminology and avoid the difficulty of conflicting uses of the word "metric." We preserve the use of the other terminology described by IPPM Framework wherever possible.

3.1 Characteristic

A characteristic is an intrinsic property that is related to the performance and reliability of a network entity. More specifically, a characteristic is a primary property of the network, or of the traffic on it. A characteristic is the property itself, not an observation of that characteristic. An example characteristic is hop capacity.

Note that a characteristic is not necessarily associated with a single number. For instance, packet loss is an important characteristic of paths and hops. However, as discussed in Section 10, loss may be expressed generally as a fraction of all traffic sent, or more specifically as a loss pattern with detailed statistical properties.

3.2 Measurement Methodology

A measurement methodology is a technique for recording or estimating a characteristic. Generally, there will be multiple ways to measure a given characteristic. Measurement methodologies may be either "raw" or "derived", but this distinction is for convenience, and doesn't really matter as far as the characteristic is concerned. Raw measurement methodologies use a technique that directly produces a measurement of the characteristic, while derived measurements might be an aggregation or estimation based on a set of low level measurements, such as using a statistical analysis of bursts of packets to estimate bandwidth capacity (for example, as performed by pchar [37] and pathrate [38]).

As an example, consider roundtrip delay as a characteristic to be measured. Roundtrip delay may be measured directly using ping, calculated using the transmission time of a TCP packet and receipt of a corresponding ACK, projected from separate one-way delay measurements, or estimated from link propagation data and queue lengths. Each of these techniques is a separate measurement methodology for calculating the roundtrip delay characteristic, and each methodology has advantages and disadvantages such as accuracy, precision, and ease-of-use.~~.~~

~~3.2.1Predictions~~

~~Prediction techniques may be applied to give a calculated estimate of the value of a characteristic at some time in the future based on historic observations. The methodologies could be based on fitting techniques using the last "n" observations, or based on scaling cyclic patterns derived from some form of averaging over several earlier periods.~~

3.3 Observations

An instance of output from a measurement methodology is an observation. An observation can be a singleton, which is the smallest individual observation, a sample, which is a number of singletons of the same characteristic together, or a statistical observation, which is derived from a sample observation by computing a statistic on the sample. The classifications of observations are taken from RFC2330 [23].

Because network characteristics are highly dynamic, each reported observation MUST be attributed with timing information, indicating when the observation was made. For singleton observations, a simple timestamp may be sufficient. For statistical observations, both the beginning and end of the observation time interval SHOULD be reported. In general, observations

SHOULD be recorded with attributes describing the conditions prevalent at the time of the observation. Representation of the measurements is discussed further in Section 5.

3.4 Characteristic vs. Measurement Methodology

Figure 1 illustrates the relationship between the terms. While the difference between a network entity and a characteristic of that entity is clear, differentiating between characteristics and measurement methodologies is frequently more difficult. The intuitive sense is that all measurement methodologies under a particular characteristic should be measuring "the same thing," and could be used mostly interchangeably, while measurement methodologies under separate characteristics are not directly interchangeable.

It is of course possible to use measurements of other characteristics to derive the value of a given characteristic. For example, available bandwidth can be defined in terms of two other characteristics, capacity and utilization (using utilization in the general sense of "traffic utilizing the path"). Available bandwidth can also be measured directly by injecting traffic into the network. Using the above definitions of characteristic and measurement methodologies, available bandwidth is a characteristic because its measurements are not equivalent to those of any other characteristic, and this gives it a well-specified place in the hierarchy. This definition is consistent with our intuitive sense of the hierarchy, which provides for characteristics being derived from a set of related characteristics. Note that available bandwidth is not unique in having measurement methodologies that measure it directly as well as methodologies used to estimate it from other characteristics.

To determine if a particular concept is a characteristic or a measurement methodology, the most important factor is whether the technique used to make the observation has any influence on the value itself. In particular, if there are different ways to observe identical or similar concepts, resulting in different values, then the concept MUST be a characteristic, but the techniques MUST be measurement methodologies.

3.5 Network Layers

For some characteristics the network layer being considered MUST be specified when reporting the measurement. We expect that the information that will be published for grid use will mainly be at layers 3 and 4. This paper follows the OSI network reference model in defining the layers [33]:

Layer 1 is the physical medium—encoding and bits on the wire
Layer 2 is the link layer dealing with framing, e.g. Ethernet or SDH/Sonet
Layer 3 is the network layer (IP)
Layer 4 is the transport layer (UDP, UDP+RTP, TCP etc.)

4. Overview of Measurements Representation

There are two elements to describing a network measurement. The first is what is being measured—the characteristic being measured. The second element is the network entity that the measurement describes—the path, hop, switch, etc. Although to a typical user the entity being measured may appear to be obvious, determining what is being measured may actually be difficult. Issues in determining the network entity being measured can include:

- Choice of protocol can influence a network's behaviour.
- Different QoS levels affect all aspects of network behaviour; in fact, some QoS policies may specify different routes between the same pair of end hosts for different service levels or classes of traffic.

- Route flaps or other instabilities mean that the same end-to-end traffic may experience a completely different environment from moment to moment.
- In high-bandwidth environments, frequently the hosts performing the measurement are the bottleneck, rather than the network path.

Developing a way for different measurement systems, implemented by different people with different goals, to interchange information about the network measurements made requires a uniform way for representing the measurement, the entity it measures, and the conditions under which the measurement is performed. Our representation combines the network entity with attributes indicating the conditions under which measurements are taken, such as protocol, QoS, network layer etc. Clearly it is not necessary to include all conditions, just those that may change and affect the observation. For example, the diameter of the optical fibre used is probably not required for many types of observations.

4.1 Network Entities

As networks are best represented in graph form, network entities are divided into nodes and paths, as illustrated in Figure 2. A node does not necessarily correspond only to a single physical entity, but can represent a range of devices including an autonomous system a switch or a virtual node. A path is a unidirectional connection from one node to another node and is represented by the ordered pair of endpoints in Figure 2. A path exists between the two nodes used as measurement endpoints.

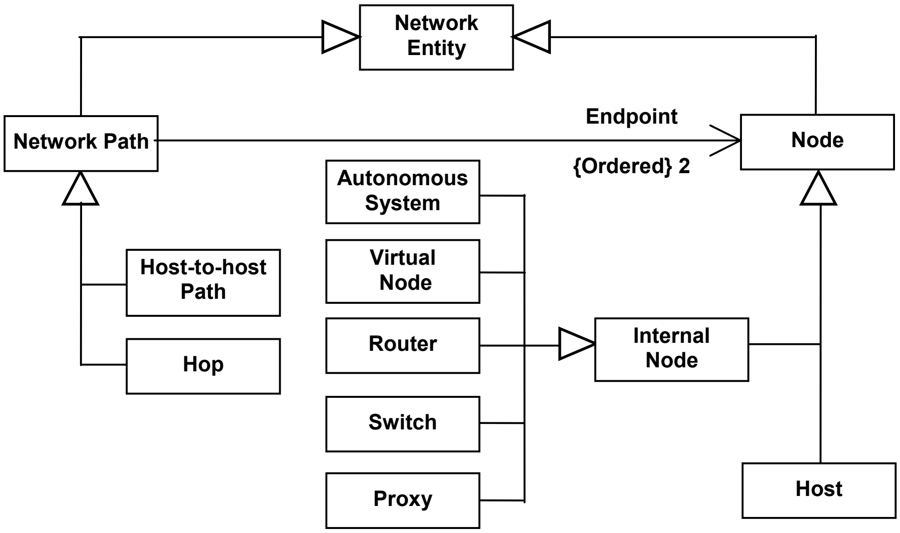

Figure 2: The relation between the node and path Network Entities. The internal nodes shown are only examples, and are not exclusive.

Nodes and paths SHOULD both be annotated with attributes such as the protocol stack (e.g., TCP over IPv4) and QoS level. These attributes describe the conditions under which a particular observation was made; use of attributes also allows the network's behaviour with different types of traffic to be represented.

4.2 Attributes and Profiles

The attributes annotating a node or path are not merely descriptive, but form a tuple that define a unique node or path. In this way, the model allows superimposition of multiple topologies over the actual physical nodes and paths, allowing the behaviour of the network under different types of traffic to be characterized separately. Note that not all possible attributes need to be specified for every measurement, only those important to the measurement and those needed to define the path uniquely need to be specified. Some characteristics, such as capacity, may be most appropriately described without specifying either a protocol or QoS level.

True portability of the characteristics requires that each characteristic SHOULD be associated with a profile that specifies which attributes are required and which are optional for the nodes and paths involved when an observation is made. Furthermore, some characteristics may be important as attributes to observations of other characteristics. For example, “Hoplist” might be an important attribute for an end-to-end observation of “achievable bandwidth.” As the NMWG develops its schema that implements this characteristics hierarchy, that schema specifies the attributes required for each characteristic. However, we are reluctant to specify mandatory attribute profiles at this stage, because differentiating between the truly mandatory attributes and optional attributes is difficult to accomplish. For now, we encourage implementers to examine the NMWG schema document and other implementations for guidance on attributes. We intend to review this issue after completion of the NMWG schema and before we consider promoting this document from Proposed Recommendation. At such time, we may include explicit lists of mandatory and recommended attributes for each characteristic. For this version, recommended attributes are listed with each characteristic when clearly appropriate.

4.3 Characteristics

The characteristics hierarchy is shown in Figure 3. Any number of these characteristics may be applied to a network entity. Note that some characteristics, such as route or queue information, are only sensible when applied to either paths or nodes. There is no requirement expressed or implied that all characteristics must be gathered for a particular network entity. The purpose of the characteristics hierarchy is merely to allow a standard way to describe what is measured about a particular entity. The current hierarchy is not intended to be complete, rather, it is expected that as more characteristics become of interest to the grid community, they will be added to the hierarchy.

4.4 Text Representation

As a hierarchical representation, the characteristics hierarchy naturally maps into a dot-separated text representation, which SHOULD be used for flat text annotations of measurements. Based on the results of GGF’s Discovery and Monitoring Event Descriptions (DAMED) working group[41], text representations SHOULD be of the form:

<entity type>.<characteristic>.<sub-characteristic>

where the <entity> may be a high-level entity such as a *path* or *node*, or a low-level entity such as a *switch*; the characteristic is one of the top-level characteristics in Figure 3, such as *bandwidth*; and the sub-characteristic is below the top-level characteristic in the hierarchy, such as *available* for *bandwidth*. There MAY be zero or more sub-characteristics. All letters MUST be lowercase except in the case of multi-word names, in which case the first letter of each word after the first MUST be capitalized. The full name of each entity or characteristic MUST be used, except for *NetworkPath*, which MUST be abbreviated as *path* for convenience. For example, here are the standard names for several combinations of entities from Figure 2 and characteristics from Figure 3 :

```
path.delay.roundTrip                    path.delay.oneWay
path.loss.oneWay                        path.packetReordering
path.loss.roundTrip                     hop.packetReordering
path.bandwidth.Utilized                 node.queue.length
hostToHostPath.bandwidth.achievable     router.queue.discipline
```

`hostToHostPath.hoplist`
`router.bandwidth.utilized`
`hop.delay.oneWay.jitter`
`switch.forwarding.forwardingTable`
`router.bandwidth.capacity`
`autonomousSystem.delay.oneWay`

4.5 Storage versus Retrieval

To this point, we have described a way to represent measurements of the network and to store them in a way that makes them available to other measurement systems. The obvious implication of this goal is that it should be possible to query for measurements as well as to store them. The flexibility of our representation may make database queries challenging to implement. For example, consider an application's query for available bandwidth between two hosts. Regardless of whether measuring systems have reported measurements for the complete end-to-end path or for each hop of the end-to-end path separately, the aggregate end-to-end result should be available to the application.

Using the protocol and QoS attributes as defining elements of the network entity also raises the issue of providing responses for queries for different sets of attributes. For example, if a query is made for UDP traffic, but only TCP measurements are available, are those reported? If a query is made for a specific QoS level, but only observations with no QoS level specified are available, are those reported?

This document does not specify the behaviour of a measurement system or archival system in answering these questions, but merely points out the challenges that exist making use of this flexible system for representing measurements and that these issues must be addressed by a measurement system using this representation for information~~. We believe that in general, the guiding principle should be return as much of the most pertinent information as possible and let the querying application make the decision what information to use.~~

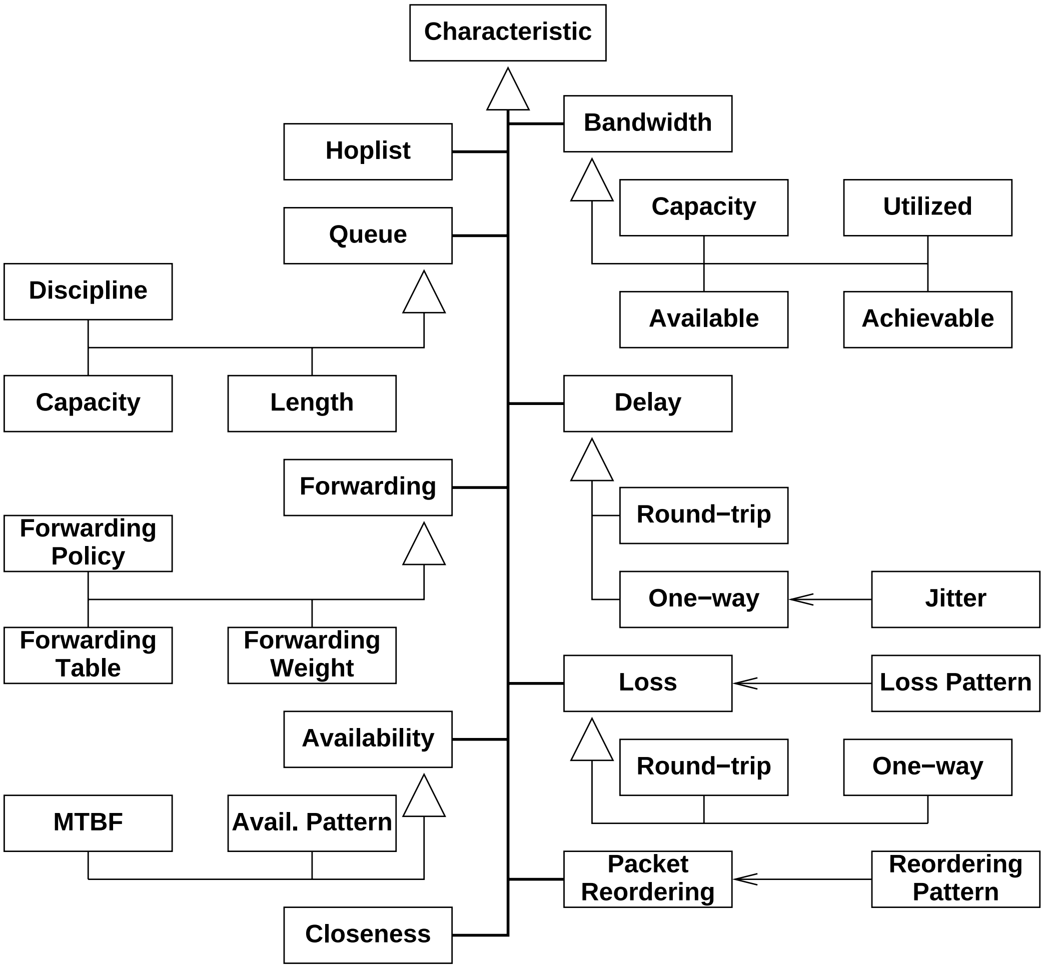

Figure 3: Network characteristics that can be used to describe the behaviour of network entities.

5. Using Nodes and Paths to Describe Topology

As indicated in Figure 2, a network topology is composed of Nodes and Network Paths. Each Network Entity SHOULD be annotated with attributes including protocol and QoS level. Network Paths are used to represent the connection between any two nodes in the network. Network Paths represent anything from an end-host to end-host connection across the Internet to a single link between two Ethernet switches. Representing a network's topology with link-layer paths allows a detailed model of a network's structure to be built.

This document does not consider multicast nor broadcast issues and focuses on point-to-point paths.

5.1 Representing Topology

Network observations can be reported for a variety of different types of network entities:

- **End-to-End Paths** are the common case of host-to-host measurements.
- **Links** between routers or switches are frequently measured for capacity, availability, latency and loss.
- **Nodes** may report useful information such as router queue discipline or host interface speed.

For systems that report network topology, the network entities themselves—hosts, routers, switches, and links—form the actual observations being reported rather than just the object being observed. Topologies are best representing using a graph structure, with nodes and edges.

Furthermore, different systems care about topology at different levels. An ISP or network manager focusing on service level agreement (SLA) satisfaction might be most interested in a topology ~~of AS to AS~~between autonomous systems (AS), reflecting the different ISP's contractual relationships. Many systems collect topology at layer-3 through traceroute, typically discovering all the routers along a path. Other systems might report only layer-2 topology between routers, or both layer-2 and layer-3 topology. Most often, a system does not have access to the layer-2 devices across an entire end-to-end path, so it might report the layer-2 topology at one or both leaf networks, but be unable to discover more than the layer-3 routers across the network core.

To design a representation capable of capturing the levels of details offered by each of these possibilities, we focused on maximizing the *flexibility* of our representation. We adopt an entity representation consisting only of *paths* and *nodes*, as shown in Figure 2. While some representations focus more on distinguishing between paths, links, and hops, we avoid making that distinction at the first level of our representation to avoid imposing structure that would be incompatible with some systems. However, our classification hierarchy captures the distinctions implied by such classifications without enforcing such terminology as “lowest-level hop” that might not be true in a system that, for example, represents layer-1 repeaters.

The most confusing aspect of this representation is the multiple uses of path. Simply, a path is a unidirectional connection between two nodes. When a measurement tool reports on the behaviour of a connection between two nodes, it is always reporting about a path. A system that reports host-to-host TCP measurements and a system that reports utilization of each physical link in a LAN both use paths. In the cases where the characteristics describe a round-trip property, the path’s source MUST be the originating node for the characteristic.

There are two keys to using this simplistic path type:

- The type of path is distinguished by the types of nodes. Because nodes are distinguished as host, router, switch, etc., the layer of the path can be readily determined.
- Each path MAY be divided into its constituent hops using the hoplist characteristic. A hoplist is an ordered set of hops. Hop is a subclass of path and therefore also a path that can be described, or subdivided, as described above. The only reason for including a separate hop type is if a particular measurement system wishes to include additional attributes for hop that are not used for path. There is no difference in the attributes or characteristics applied to each type of path.

A very common representation is to use path to describe a host-to-host connection, hop to describe a router-to-router connection, and link to describe a switch-to-switch connection. Paths can be composed of hops and hops can be composed of links. The path-only representation is capable of expressing these details because the type of a path can be determined by the source and destination nodes. Therefore, paths, hops, and links can be represented in the path-only form. Furthermore, a host-to-host path in the path-only representation can be subdivided into all of the known steps—both layer-3 and layer-2—using a single hoplist. Systems wishing to import such a representation with the multiple-type approach would need to convert this to the multiple types in their representation. However, there are significant programming advantages to being able to iterate over a list of a single type to calculate capacity, for example.

Systems compliant with this specification MUST provide a path-only representation of topology for use in exchanging measurements with other systems. They MAY provide additional interfaces, either at the user-level or to interface with other systems, that use other representations of topology.

5.2 Physical and Functional Topologies

There are two approaches to characterizing network topology: physical and functional. The physical approach determines the physical hops that connect the network together. By determining the connections between nodes, along with their capacities, queuing disciplines, and traffic load, the network can be modelled and its behaviour analysed or predicted. Physical topology can be determined for both LANs [5, 19] and WANs [14, 22].

The functional approach differs in that it makes use of end-to-end information, under the assumption that such observations are more readily available and usable than modelling low-level network behaviour. Functional topology representations attempt to group and arrange network sites according to their perceived closeness determined by traffic performance, rather than according to the actual connections of physical hops. This approach may be taken across a variety of sites distributed around the Internet, or using a single source tree [7, 17, 27~~, 30~~]. Internet distance maps [30] are among the best-known examples of functional topology. Functional topologies provide useful information that an application could use to make many of the same decisions that are made with physical information.

The only noticeable difference in the representation of physical and functional topologies is that for the latter, all of the internal nodes are virtual nodes, as shown in Figure 2. An attribute indicating whether a Network Entity is physical or functional MUST be provided.

5.3 Nodes

Nodes are generally classified into hosts and internal nodes:

- Hosts are considered to be only endpoints of communication.
- Internal nodes are capable of forwarding traffic, and can be routers, switches, or proxies (e.g. a device that implements NAT or SOCKS, or any other in-stream node that may modify the data), as well as more general concepts such as autonomous systems or virtual nodes in a topology.

- Virtual nodes are used to describe additional functionality that might be found in a physical node. Note that physical and virtual nodes do not occupy disjoint graphs in a network topology. In particular, functional networks always contain physical nodes at their edges. There may be several virtual nodes, describing different functionality, overlaid on a single physical node. Furthermore, virtual nodes may play a role in describing physical networks.

Two examples of the use of a virtual node are given:

- Because some hosts also perform routing, a virtual node might be used to represent the routing done by the host, with the host being a separate node linked to the virtual node.
- As an example of using a virtual node to describe a physical network, consider a half-duplex link such as a wireless network. Because paths are defined to be unidirectional, the half-duplex nature of such a network cannot be adequately described in a path. However, by imposing a virtual node in the network with appropriate characteristics of the half-duplex link, the virtual node can represent the transmission characteristics of the half-duplex link.

Like paths, nodes also should be annotated with a set of attributes. The attributes indicate a specific set of characteristics of the node when handling a particular type of traffic. In particular, queuing disciplines, queue lengths, etc. may vary quite significantly according to type of traffic and QoS level.

6. Forwarding and Hoplist Characteristics

Nodes and Network Paths are sufficient to define a topological structure, but they do not indicate the relationship between paths and their constituent hops, nor do they indicate how traffic is actually forwarded through a network. The Hoplist characteristic is responsible for representing the constituent hops of each path, while Forwarding is used to describe how nodes send traffic through a network.

6.1 Hoplist

The hoplist characteristic, shown in Figure 3, allows a path, such as an end-to-end path, to be subdivided into the hops that form the path. For example, a path between two end-hosts could have a hoplist containing the set of router-to-router hops between each of the routers along the end-to-end path. Each member of the hoplist is itself a hop, which is also a type of path as shown in Figure 2. End-to-end paths are not the only paths that can be divided into hops with a hoplist. For example, one of the above router-to-router hops could be further subdivided by another hoplist into hops representing the links between Ethernet switches in a LAN.

When a round-trip path is represented, a system MAY provide a single hoplist for the entire round-trip, starting and ending at the “source” node, with the “destination” node in between. If the hoplist is only listed from “source” to “destination,” it MAY be assumed to only represent the outbound hoplist (e.g., in the common case where traceroute is run from the source node).

> *NOTE: The authors view the representation of round-trip paths in this manner as a compromise between simplicity of design and accuracy of representation. Our expectation is that systems that care about the round-trip nature of the path will be designed to account for it, and those that are not designed to account for it will not notice the difference while iterating over the hops. We will re-evaluate this issue as this proposed recommendation moves toward a recommendation.*

6.1.1 Path Unification

The hoplist characteristic is easily used to divide an end-to-end path into hops. However, the inverse problem—building an end-to-end path from hops—is quite challenging. In particular if separate measurement systems are reporting the hops, it may be difficult to build the end-to-end path because the nodes may be described using different names. We refer to the process of building an end-to-end path from component paths as path unification.

Path unification is hard for even one measurement system, not to mention merging the results of multiple systems. Because routers use different IP addresses for each port, and those IP addresses typically resolve to different names, it is frequently difficult to identify the individual routers uniquely. For example, traceroute identifies only the receiving interface of each router. Measurements based on traceroute, therefore, do not provide a useful source-destination pair, but only a series of destination interfaces, where each indicates a single hop. Combining traceroute measurements with those from other tools, may be difficult because different names or IP addresses may be used to refer to the same router. Resolving these issues is beyond the scope of describing the characteristics, but we note that comparing hop destinations may be more useful than comparing hop sources due to the way traceroute works [14].

6.2 Forwarding

The forwarding characteristics are used to describe how internal nodes forward traffic from one node to another. Forwarding can occur at several layers, primarily at layer-2 and layer-3, which are generally referred to as switching and routing, but forwarding can also be performed at the application layer.

6.3 Forwarding Table

The forwarding table is the routing table, forwarding database, NAT table, or other structure used by an internal node to determine where to forward traffic when it receives it.

6.4 Forwarding Policy

The forwarding policy characteristic should be used to describe additional features of how a particular node forwards traffic. In particular, the routing algorithm used, the queuing discipline, and other details should be described as forwarding policy characteristics. Schema implementers MUST establish more concrete definitions, and such schemata may need to vary significantly from layer to layer and protocol to protocol. Portability of this characteristic will require the development of a system for denoting various forwarding policies.

6.5 Forwarding Weight

Forwarding weight is used to describe the input used by the forwarding policy. For instance, in the Open Shortest Path First (OSPF) protocol, the cost metric for each hop (path) is represented as an integer distance. In the case of inter-domain routing protocols such as the Border Gateway Protocol (BGP), each route is annotated with a vector of Autonomous Systems (AS) that must be traversed. From this information the “AS Path” length can be computed and used as the basis on which most routing decisions are made.

Note that while forwarding policy and forwarding table are generally characteristics of nodes, the forwarding weight may be either a characteristic of paths (OSPF) or nodes (BGP)." Similar to Forwarding Policy, portability of forwarding parameters requires an additional standard.

7. Bandwidth Characteristics

In the networking community, bandwidth is defined most generally as data per unit time. Some have suggested "bitrate" as an alternative, as bandwidth has conflicting meanings in the physics community. As authors, we agree that the "bitrate" is a more precise term without conflicting meanings in other communities, however, we believe bandwidth is the most generally accepted term within the networking community, and therefore we use "bandwidth" to represent the concept. Implementations of this system MUST use the term "bandwidth" and SHOULD support "bitrate" as a synonym for bandwidth.

The "bandwidth of a hop (or path)" is not a precisely defined term and specific characteristics must be clearly defined prior to discussing how to measure bandwidth. There are four characteristics that describe bandwidth, summarized here, and described in detail below:

- *Capacity*: The maximum amount of data per time unit that a hop or path can carry
- *Utilization*: The aggregate traffic currently on that hop or path.
- *Available Bandwidth*: The maximum amount of data per time unit that a hop or path can provide given the current utilization.
- *Achievable Bandwidth*: The maximum amount of data per time unit that a hop or path can provide to an application, given the current utilization, the protocol and operating system used, and the end-host performance capability. The aim of this characteristic is to indicate what *throughput* a real user application would expect as opposed to what the network engineer could obtain. Some people prefer *throughput* over *achievable,* we prefer *achievable* because it is clearer that it can be used as a characteristic applied to a path, as opposed to a specific connection across that path. Implementors MUST support "achievable bandwidth" and MAY support "throughput" as a synonym.

Each of these characteristics can be used to describe characteristics of an entire path as well as a path's hop-by-hop behaviour.

The network layer at which bandwidth is measured SHOULD be an attribute of the measurement. As an example consider capacity. At Layer 1 this would give the physical bit rate, but as one went up the layers, the capacity would apparently decrease by taking into account the framing and packet spacing at the link layer and then the protocol overheads for layers 3 and 4.

For a recent comprehensive review of types of measurements and measurement methodologies, readers are referred to Prasad et al. [25].

7.1 Capacity

Capacity is the maximum amount of data per time unit that the hop or path has available, when there is no competing traffic. The capacity of the bottleneck hop is the upper bound on the capacity of a path that includes that hop. This information is useful in many areas such as determining how to set optimal TCP buffer sizes. Again, the layer at which a capacity measurement is taken SHOULD be reported as an attribute of that measurement.

7.1.1 Capacity Measurements

Link layer capacity can sometimes be obtained directly via SNMP queries to the network switches and routers along the path. However, in general it is not yet possible to get access to this information for network switches and routers in different administrative domains. For example, an end-user is as yet unlikely to be able to access SNMP information in a commercial ISP's network.

There are a variety of techniques for inferring capacity. On an unloaded network one can send a suitable stream of packets and measure the bandwidth at the receiver. Other techniques are based on analysis of “packet trains.” Bursts of carefully-spaced datagrams can be injected into the network, and the difference between the separation of packets at the source and the destination (or other point), referred to as the dispersion, is observed. Packet dispersion techniques evaluate the dispersion of packets based on analytical models of network behaviour, and estimate path characteristics based on this information. For example, the bottleneck capacity hop separates back-to-back packets by the time each packet takes to cross that hop. If this separation is maintained through the remainder of the path, the bottleneck capacity can be derived by observing the widest spacing of the packets. Actual implementations based on these techniques use differing initial packet separations and apply a variety of statistical techniques to filter out the noise and random behaviour of real networks. Although they are quite useful, there are a number of challenges associated with implementing and applying these measurement techniques. They include, but aren’t limited to:

- Host timing issues—as link speeds increase, intra-host latencies make a larger difference in measurement accuracy. Interrupt coalescing and driver and kernel implementations can make appreciable differences.
- Differential Queuing—there are many techniques for so-called “traffic shaping,” and it is difficult to be certain that UDP or ICMP are treated the same way as TCP in the network infrastructure.
- Clock resolution—as we approach to higher speeds such as 10Gbits/s, inter-packet delay timing requires clocks with resolution of 1 µs or better, which is a challenge for today’s CPUs.

7.2 Utilization

Utilization is the aggregate capacity currently being consumed on a hop or path. As a singleton observation, utilization is the amount of data passing through the hop or path over a particular time interval. This gives an average bandwidth over the observation time. Selection of the observation time interval is important: too short an interval may cause difficulties, e.g. CPU loading issues on routers or switches, and too long an interval would mask peak bandwidth rates. Due to the burstiness of network traffic, utilization can vary widely between samples. Selecting a sampling interval requires extensive expertise and knowledge of the intended application of the utilization information.

More complex representations of utilization are possible. For example, a profile of traffic on a hop during a particular time interval is also a way of observing utilization, giving significantly more detail than the simple bandwidth in use. We represent the traffic profile in the hierarchy as a sub-characteristic of utilization, because the simpler fraction can always be derived from a more detailed representation of the traffic utilizing the hop.

7.2.1 Utilization Measurements

Utilization measurements are generally collected passively. Like capacity, utilization can sometimes be obtained directly via SNMP queries to the network switches and routers along the path. However, as mentioned above, in general, a typical user cannot get access to such information for the complete path.

7.3 Available Bandwidth

Available bandwidth is the maximum amount of data per time unit that a hop or path can provide, given the current utilization. It could be argued that available bandwidth is redundant as it can be derived from capacity and utilization, however it is possible to produce a measurement of un-utilized bandwidth without directly measuring capacity or utilization.

7.3.1 Datagram Measurements

There is general consensus that packet train and packet dispersion methodologies, described above, are well suited for measuring path capacity. More recent research has explored using similar techniques to measure available bandwidth [34]. There is, however, some question about their ability to measure this characteristic in all situations [11], in particular in providing accurate absolute quantitative estimates. Besides the problems with cross-traffic and multi-modality that Dovrolis points to [11], there are also problems with today's packet dispersion techniques at high speeds (~Gbit/s and beyond) with the basic timing needing sub-micro-second time stamps. The use of interrupt coalescing in modern gigabit network interfaces further complicates matters. It may be necessary to combine the lightweight packet/train/dispersion techniques with more intrusive techniques to provide absolute quantitative values.

7.4 Achievable Bandwidth

Achievable bandwidth is the maximum amount of data per time unit that a hop or path can provide to an application, given the current utilization, the protocol and operating system used, and the end-host performance capability and load. On a path consisting of several hops, the hop with the minimum transmission rate determines the capacity of the path. However, while the hop with the minimum unused capacity may well limit the achievable bandwidth, it is possible (and even likely on high-speed networks) that the hardware configuration or software load on the end hosts actually limits the bandwidth delivered to the application.

Tools to measure achievable bandwidth fall into two general categories:

- TCP Flow-based (or Connection-Oriented)
- Datagram-based (including both small stream and packet dispersion techniques)

Issues related to each of these measurement techniques are discussed below.

7.4.1 TCP Measurements

One of the most commonly used flow-oriented measurements is achievable TCP bandwidth. Achievable TCP bandwidth is both one of the most useful and yet most difficult to characterize of all measurements. Achievable bandwidth is, in fact, often *not a network measurement*, but a measurement of the end host capability. We include it in this document because of the fact that it is so useful. Also, a more correct term for this might be ~~*achievable*~~ *throughput*, but we are using the term achievable bandwidth because we want to group this characteristic clearly with the other bandwidth characteristics.

A common method of measuring achievable bandwidth is to open up a TCP stream and send some data, thus simulating an application data stream. A number of tools have been developed over the years that do this, including *ttcp*, *iperf* [39], and so on. There are a number of problems with this technique:

Results are greatly affected by the TCP implementation. As described in the BTC RFC, the TCP implementation on both the sending and receiving host operating systems can have a large influence on the achieved bandwidth. Any methodology that relies on a system's TCP implementation is therefore subject to its influence on its results. Furthermore, tuning the sending and receiving hosts, such as selecting the appropriate sized socket buffers, can have a profound influence on performance.

Other factors that influence how one measures achievable bandwidth include:

- Results are affected by the TCP slow-start algorithm. For a fairly typical high-speed link (e.g.: capacity = OC-12, RTT = 50 ms), slow start takes about 1 second. Therefore short tests are dominated by slow start, and longer tests are more intrusive. Some tools try to factor out the effect of slow start.

- TCP-based tools can be quite intrusive, putting a high load on the network. Experiments at SLAC have shown that, even for relatively low bandwidth delay products (e.g. 100 Mbits/s and 200 msec RTT), to get a reasonable estimation of available bandwidth on a WAN using iperf requires at least a 10 second test, which places a lot of unnecessary traffic on the network [8].
- Real applications are more bursty than most of these tools, and will therefore be more subject to router buffer overflows and queuing delays than tools like iperf.

Most of these issues should affect the network test procedures and real applications in similar ways and one would expect that the bandwidth achieved would be similar. However other issues like disk sub-system performance and application complexity often dominate the throughput achieved by the real application.

The IETF IPPM defines a TCP measurement related to achievable bandwidth, called “Bulk Transfer Capacity” (BTC) [20]. The specific definition of bulk transfer capacity is:

BTC = data_sent / elapsed_time.

This effectively tries to capture the “steady state” of a long-lived flow—amortizing out the constant of overhead. The BTC definition assumes an “ideal TCP implementation”, which, in practice, does not exist. Therefore the BTC RFC specifies that type of TCP implementation must be specified as part of BTC. There are so many TCP implementations on the Grid (i.e.: at least 20 versions of the Linux kernel contain a TCP implementation modification large enough to affect achievable bandwidth) that we feel that BTC, as defined, is not very useful to Grid applications.

7.4.2 Datagram Measurements

Other types of stream-oriented measurements use datagrams to investigate the achievable bandwidth for UDP-based protocols and to simulate TCP flows or to quickly saturate the network. We make the distinction that if the aggregate behaviour of the (connectionless) flow is considered, then this measurement technique is stream-oriented. This most correctly models UDP-based bulk transfer utilities and “streaming” applications as well.

A stream of suitably spaced UDP packets can be transmitted and the amount of data received and the time taken to receive that data measured at the destination, several tools do this including *iperf* and *UDPmon* [39]. The spacing could be regular or follow a Poisson distribution, provided that the average generated packet rate does not exceed the transmission capability of the network interface card, NIC, used. If the NIC is the bottleneck, packets could be queued or even lost in the sending operating system, distorting the sequence presented to the network. Recording the time to transmit the data may only observe the time to move data from user space to the kernel, not the time to send it over the network.

8. Delay Characteristics

This section draws heavily on RFC 2679 “One-way Delay Metric for IPPM,” G. Almes, S. Kalidindi, and M. Zekauskas and RFC 2681 “A Round-trip Delay Metric for IPPM,” G. Almes, S. Kalidindi and M. Zekauskas [1, 3]. For more details these references should be consulted.
As described in RFC 2679, delay is important because:

- Some applications do not perform well (or at all) if end-to-end delay between hosts is large relative to some threshold value.
- Erratic variation in delay makes it difficult (or impossible) to support many real-time applications.

- The larger the value of delay, the more difficult it is for transport-layer protocols to sustain high bandwidths.
- The minimum value of this characteristic provides an indication of the delay due only to propagation and transmission delay, which will ~~.~~
- ~~The minimum value of this characteristic provides an indication of the delay that will~~ likely be experienced when the path traversed is lightly loaded.
- Values of this characteristic above the minimum provide an indication of the congestion present in the path.

A general definition of delay, following RFCs 2330, 2679 and 2681, is the time between when the first part (e.g. the first bit) of an object (e.g. a packet) passes an observational position (e.g. where a host's network interface card connects to the wire) and the time the last part (e.g. the last bit) of that object or a related object (e.g. a response packet) passes a second (it may be the same point) observational point. ~~The above raises several issues including:~~Issues complicating the precise measurement of delay include:

- How the time is synchronized if 2 observational points are used;
- Packet fragmentation ~~issues~~;
- Most measurements are made by Internet hosts, which can introduce scheduling delays into the timestamps.

Delay can be measured one-way or roundtrip. ~~One-way delays are important because today's Internet connections often use asymmetric paths, or have different quality of service in the two directions. Even symmetric paths may have different characteristics due to asymmetric queuing. Also an application may depend mostly on performance in one direction. For example the performance of a file transfer may depend more on performance in the direction in which the data flows, or in a game it may be more important to get a request to the destination before another gamer does, than it is to get the response back.~~

~~In principle the round-trip delay can be composed from the one-way delay measurements in both directions. On the other hand, it is often easier to measure round-trip delay than one-way delay since only one observation point and clock is needed. Also, many applications depend mainly on round-trip delays. RFC 2679 discusses the issues of errors and uncertainties in delay measurements related to clock accuracy and synchronization.~~

8.1 One-way Delay

One-way delays are important because today's Internet connections often use asymmetric paths, or have different quality of service in the two directions. Even symmetric paths may have different characteristics due to asymmetric queuing. Also an application may depend mostly on performance in one direction. For example the performance of a file transfer may depend more on performance in the direction in which the data flows, or in a game it may be more important to get a request to the destination before another gamer does, than it is to get the response back.

One-way delay is usually measured by timestamping a packet as it enters the network and comparing that timestamp with the time the packet is received at the destination. This assumes the clocks at both ends are closely synchronized. For accurate synchronization (tens of µs) the clocks are often synchronized with GPS. If the packet is not received at the destination within a reasonable period of time, then the packet is assumed to be lost.~~one-way delay is undefined (informally, infinite according to RFC 2679), and the packet is taken to be lost.~~

RFC 2679 defines a precise measurement methodology for one-way delay. While there are other ways to measure the one-way delay characteristic, we suggest that new implementations follow the guidelines of RFC 2679.

8.1.1 Jitter

The variation in the one-way delay of packets is sometimes called the “jitter.” Because it is a partial statistical description of another characteristic, we place it under one-way delay in the characteristics hierarchy. It is represented separately because, while it is difficult to accurately measure one-way delay, it is quite easy to measure the variation in one-way delay by observing the inter-arrival times of packets sent at a fixed interval.

As described in RFC 3393 [10] the term jitter is used in different ways by different groups of people. The IP community therefore defined the term IP Packet Delay Variation (IPDV), for a selected pair of packets in a stream of packets, as the difference in the delay of the first packet and the second of the selected packets.

Jitter, in general, is very important in sizing playout buffers for applications requiring regular delivery of packets (e.g. voice or video). Other uses include determining the dynamics of queues within a network where changes in delay variation can be linked to changes in queue length. Given a stream of delay measurements, IPDV is easy to extract. Note that because it is the difference in delays between packet pairs, clocks do not need to be carefully synchronized. Given an IPDV probability distribution, one can also calculate statistics such as the Inter Quartile Range (IQR) to provide other estimates of jitter.

8.2 Roundtrip Delay

In principle the round-trip delay can be composed from the one-way delay measurements in both directions. However, this requires making the individual measurements close to one another and being able to select the appropriate 2 measurements to add together. Furthermore, measuring round-trip delay directly is often simpler than measuring one-way delay because only one observation point and clock is needed. Also, many applications depend mainly on round-trip delays. RFC 2679 discusses the issues of errors and uncertainties in delay measurements related to clock accuracy and synchronization.

~~Roundtrip delays can be composed from the individual one-way delay measurements. However, this requires making the individual measurements close to one another and being able to select the appropriate 2 measurements to add together. Since roundtrip delays are usually easier to measure than one-way delays, round-trip delays are usually measured directly.~~

Round-trip delay is usually measured by noting the time when the packet is sent (often this time is recorded in the packet itself), and comparing this with the time when the response packet is received back from the destination. This requires that the destination must be prepared to receive and respond (i.e. send the packet back to the source) to the test packet.

Analogous to RFC 2679 for one-way delay, RFC 2681 defines a precise measurement methodology for round-trip delay. While other methodologies exist for this characteristic, we suggest that new implementations adopt that approach when appropriate.

Most modern IP stacks implement an ICMP ECHO~~echo~~ responder in the ICMP server [24]. Upon seeing an ICMP ECHO~~echo~~ request packet, the ICMP ECHO~~echo~~ responder will send a corresponding ICMP echo response packet to the sender. Thus no special server has to be installed. The ubiquitous ping tool makes use of ICMP ECHO~~echo~~ and is heavily used for making round-trip delay measurements.

Another way to avoid having a server is to measure the delay between sending a TCP packet, such as a SYN packet, and then timing how long before the ACK is received, see for example [9]. The TCP stack itself also estimates the RTT and the Web100 tool [29] allows access to various TCP RTT estimates such as minimum, maximum and smoothed RTT. It is also possible to measure the TCP RTT by passively capturing the TCP packets. The tcptrace tool [21] provides

RTT reports for various passive capture tools such as tcpdump. These TCP mechanisms may be useful if, for example, ICMP echo is blocked or is suspected to be rate limited. A disadvantage of the SYN/ACK mechanism is that the frequent opening of a TCP connection via SYN packets may look like a denial of service attack.

8.3 Issues in Measuring Delay

Active measurements of delay require a probe to send probe packets and a responder to receive the probe packets and possibly return response packets. The need to have a responder/server at all the destination hosts can be major drawback to deployment. For roundtrip measurements, in many cases one can avoid this difficulty by using the ICMP echo response server built into most modern hosts.

In either case there can be problems since security may block or rate limit access to the server. Rate limiting the delay probes/responses can be tricky to determine and is sometimes suspected due to anomalous results, for example, the loss rate increases as delay packets are sent at higher frequencies. One may be able to determine whether ICMP echo rate limiting is being imposed by comparing the ICMP echo delays with those measured using TCP RTTs as mentioned above.

As is described in detail in the RFCs referenced, clock synchronization and errors are critical for one-way delay measurements, but less so for roundtrip and IPDV measurements. Many ping implementations do not provide sub millisecond timing (or in the case of Windows sub 10 ms). Thus delay measurements on short hops such as on a ~~Local Area Network (~~LAN~~)~~ are often not possible using ping.

~~Though the IETF delay RFCs suggest using infinity for the delay if a measurement times-out, several projects ignore such packets as far as delay is concerned, and simply count them as lost. The delay RFCs also tackle the problem of duplicate and out-of-order packets, and we discuss these issues in Section 12.~~

Some NICs coalesce interrupts to reduce the interrupt load on the CPU. This can lead to aggregations of delays since the arrival of packets may not be notified to the host CPU by the NIC, until some criteria is reached. In such cases it may be important to use the NIC to do the packet timing.

8.4 Representing Delay

Though the IETF delay RFCs suggest using infinity for the delay if a measurement times-out, several projects ignore such packets as far as delay is concerned, and simply count them as lost. The delay RFCs also tackle the problem of duplicate and out-of-order packets, and we discuss these issues in Section 12. The IETF RFCs adopt their approaches because, among other reasons, loss and reordering frequently affect applications in the same way that large delay does. Designers of grid measurement systems should carefully consider their design choices in these areas. In particular, disregarding the advice to represent lost packets with infinite delay because it makes averaging impossible is ill-advised, as the arithmetic mean is not useful in summarizing delay.

9. Loss Characteristics

Along a network path, packets sent out by a sender may get lost and may, in consequence, not be received by their destination. However, the loss of a single packet often has little impact on network applications, but repeated loss can have a significant effect. Therefore, the statistical properties of packet loss events are the most interesting for determining application performance.

According to RFC 2680 [2], for packet loss both a singleton characteristic and statistically derived quantities over time series of those singleton characteristics need to be taken into account. The singleton characteristic observes whether or not a packet sent from a source will be received by its destination, using a given protocol, at a certain point in time. In this section, we describe one-way loss, roundtrip loss, and statistical loss properties, all three are refinements of the general concept "loss," according to the hierarchy given in Figure 3.

This section draws heavily on RFC 2680 "A one-way Packet Loss Metric for IPPM", G. Almes, S. Kalidindi, and M. Zekauskas [2]. See that document for further details.
As described in RFC 2680, loss is important since:

- Some applications do not perform well (or at all) if end-to-end loss between hosts is large relative to some threshold value.
- Excessive packet loss may make it difficult to support certain real-time applications (where the precise threshold of "excessive" depends on the application).
- The larger the value of packet loss, the more difficult it is for transport-layer protocols to sustain high bandwidths.
- The sensitivity of real-time applications and of transport-layer protocols to loss become especially important when very large delay-bandwidth products must be supported.

Packet loss can impact the quality of service provided by network application programs. The sensitivity to loss of individual packets, as well as to frequency and patterns of loss among longer packet sequences is strongly dependent on the application itself. For streaming media (audio/video), packet loss results in reduced quality of sound and images. For data transfers, packet loss can cause severe degradation of achievable bandwidth. Depending on the application itself, the above-mentioned threshold values can vary significantly.

The singleton packet loss is a binary characteristic. The value 0 indicates successful transmission from source to destination. The value 1 indicates a lost packet.

The **singleton, one-way packet loss** from source S to destination D at time T has the value 0 if the first bit of a packet has been sent by S to D at time T, and D has received that packet. The characteristic has the value 1, if D did not receive that packet.

The **singleton, roundtrip packet loss** from source S to destination D at time T has the value 0 if the first bit of a packet has been sent by S to D at time T, D has received that packet, and subsequently, a reply packet has been sent by D to S that has been received by S. The characteristic has the value 1, if S did not receive the reply packet.

9.1 One-way Loss

In the Internet, the path from a source to a destination may be different from the path from the destination back to the source (asymmetric paths). Even in the case of symmetric paths for both directions, additional traffic from other applications may cause different queuing behaviours of the two directions. Roundtrip measurements therefore mix the properties of both path directions, possibly producing misleading results. (For example, the achievable bandwidth of a file transfer may strongly depend on the packet loss on the path from source to destination, while being ~~largely in~~less dependent on loss on~~f~~ the return path~~verse direction~~.) For these reasons, one-way loss measurements are generally preferred.

Packet loss is highly dependent on the path the traffic follows, as well as the protocols used and QoS levels selected. All of these are incorporated as parameters of the path as described in Section 6.

9.2 Roundtrip Loss

From a network-centric viewpoint, the more useful loss characteristic is one-way loss, as described in RFC2680. Roundtrip loss is, in fact, the combination of two one-way loss measurements. The focus of this document is, however, on the impact of network characteristics on grid applications. For this reason, roundtrip loss becomes an important characteristic of its own. Furthermore, as measurements of round-trip loss, such as ping, are frequently reported, it must be represented in the hierarchy.

Roundtrip loss occurs when a sender does not receive a reply for a packet, although the network (or application-layer) protocol expects such a reply. Assuming statistical independence of packet loss in both directions of a network path, roundtrip loss properties may be derived from the individual characteristics for each direction. However, for applications that use request/reply protocols (like RPC or HTTP), traffic in both directions is not independent, so roundtrip loss is more precisely measured directly.

9.3 Loss Patterns (Statistical Properties)

Currently, the Internet community primarily uses only one statistically derived quantity for loss, namely the loss average, defined as the average of the singleton loss values over a series of sent packets. The loss average corresponds to the loss rate which is given as a percentage between 0% and 100 %.

Other statistically derived quantities, like loss burstiness patterns, loss free seconds, loss or conditional loss probability [4], are of interest to specific transport protocols as well. Techniques are being developed to measure these types of loss, and should be represented under loss pattern.

9.4 Issues in Measuring Loss

The general problem with measurements of packet loss (as with other properties) is that the measurements need to be as close as possible to application data transfer, in order to produce significant results. Because many Internet routers forward packets differently according to their protocol, loss properties SHOULD be derived for each relevant protocol separately.

- **ICMP (ping)**
  The ping utility uses ICMP ECHO REQUEST / ECHO REPLY packets to determine both roundtrip delay and roundtrip loss. Ping reports the roundtrip loss rate. Typical implementations of ping send only one packet per second, possibly causing fewer loss events than application-data transfers which typically send bursts of back-to-back packets.

  Furthermore, ping’s default packet size is much smaller than the MTU of existing networks. Such small packets sometimes cause much smaller loss rates than the longer packets which are typical for application data transfers. Ping also uses ICMP, which some routers may treat differently than regular TCP or UDP data. However, routers usually pass ICMP packets not directed at that router, in the same manner as TCP or UDP packets.

- **TCP**
  Loss information for TCP as the predominant transport protocol is the most important for applications. However, TCP packet loss can only be measured at the kernel level of a protocol stack. Retrieving this information is thus dependent on the interface provided by the operating system. TCP roundtrip loss might also be measured by SYN/ACK packet pairs. However, as these packets are used in the special case of connection establishment, the significance of the collected loss data is questionable for TCP data transfers.

The sting tool [26] can accurately measure loss rates, separately for both directions of a TCP connection. Sting re-implements TCP in user space to retrieve information about lost or duplicate acknowledgments in order to derive which packets have been lost. However, the implementation relies on cooperation of the operating system kernel, limiting sting's applicability and portability.

- **UDP**

UDP lends itself for application-layer loss measurements, both one-way and roundtrip. In such measurements, a series of packets containing timestamps are sent from source to destination. The destination records for each packet the event of receiving. The receiver needs to define a timeout value for each packet after which the packet is supposed to be lost. Besides the measured UDP packets, both parties need to communicate across a (TCP) channel with enforced packet delivery to exchange control information, e.g., for determining a suitable timeout value. UDP may also be rate limited by some routers.

10. Availability

The network availability characteristic is a measure of how often the network is "up" or "down." The exact definition of "down" compared to "very congested" may vary, and probably depends on the application. For example, one definition may be if no ICMP packets get through the network for 10 seconds, then the network is considered "down."

One example of where this characteristic may be useful is to describe a microwave link that goes down during thunderstorms, but is up the rest of the time. This hop might have an availability of 90%. Availability may also be related to a "service level agreement," or SLA, which is a contract between a network service provider and a customer that specifies what percentage of the time a network services will be available.

Availability MAY also be applied to nodes, although from a networking perspective it is frequently difficult to differentiate between a node being unavailable or the path to that node being unavailable.

11. Queuing Information

In wired networks, traffic congestion manifests itself as queue overflow in routers, which is the predominant reason for packet loss. To model network behaviour, analytical models require information about the static and dynamic properties of the queues. Information about queues is described under this characteristic.

Queue information can be either raw or derived. Raw information, such as that obtained by SNMP, can indicate the precise drop rate, length, and queuing disciplines used by the router. Derived information can be obtained by sending traffic through the router and deducing its behaviour through analytical means [18]. However the information is obtained, it does not affect the characteristic it attempts to determine, although measurements should indicate how the information was derived.

To describe a queue, we include three characteristics: disciplines, capacity, and length. These characteristics are sufficient to describe a fixed-length tail-drop queue, but are inadequate for other types of queues. Therefore, in particular for queuing disciplines, while this document can suggest general characteristics, the appropriateness of these characteristics and the need to add additional per-disciplines characteristics or extend our general characteristics must be evaluated according to the nature of the disciplines being represented.

The first information stored about a queue is the disciplines used by the queue. This includes not only the drop disciplines—how the node determines what packet to drop when (or before) the queue overflows—but other details such as approximations of fair-queuing that may be implemented for this queue. A complete enumeration of all drop disciplines is beyond the scope of this document, however we suggest that the disciplines used be represented as a set of ASCII descriptions of the disciplines, such as “tail-drop,” “RED,” etc., with the mapping of disciplines to names and version numbers agreed upon by an appropriate group in a future standard.

Associated with each queue are also parameters such as capacity and current length. Our description of a queue incorporates these two terms; however we note that these are intended for general descriptions and specific parameters required to describe the behaviour of different queuing disciplines may be necessary for each discipline.

12. Packet Re-Ordering

When packets are sent over a network path, packets may be delivered to their destination in the wrong order. There could be many reasons why packet re-ordering may occur on a path, some of the more common reasons include: multiple physical trunk links have been deployed in a path, the use of parallel forwarding engines in routing equipment, and routing changes at both level-2 and level-3. It is also likely that there is some threshold in the traffic load above which re-ordering may take place. See [35] and [36] for discussions on re-ordering issues.

The occasional interchange of a single packet with its neighbour often has little impact on network applications, but repeated occurrences can have a significant effect. For example re-ordering could be important to TCP if a packet was re-ordered by more than 3 packets, triggering duplicate ACKs and causing re-transmission of the supposedly lost packet. Even though the re-ordered packet is not lost, there is the implication of the protocol stack doing more work when it gets an out of order packet. Re-ordering is also a consideration when designing an application that uses UDP; in this case the application must correctly deal with out of order packets.

Thus, the statistical properties of packet re-ordering events are the most interesting for determining application performance. Unfortunately, at the time of this writing, there is no widely accepted way to represent packet reordering. Various network components report reordering, but precise definitions of the algorithms used, and justification for those algorithms in an analytical framework, are lacking. The IPPM is working on developing a representation for packet reordering.

12.1 One-way Re-ordering

Only one-way packet re-ordering is meaningful. One straightforward representation is to represent ~~The~~ singleton packet re-ordering ~~characteristic takes~~with an integer value. The value 0 indicates successful transmission from source to destination with no re-ordering. A positive value indicates the number of packets that have displaced the packet under consideration from its correct position.

12.2 Re-ordering Patterns (Statistical Properties)

Because of the interactions between packet re-ordering and the TCP algorithm, patterns of re-ordering are more interesting than singleton observations. For example, a single packet re-ordered to arrive earlier than it should will cause no problems, as the subsequent packets will continue to deliver in-order data. However, the same packet re-ordered later in the sequence can appear as a loss as intervening packets result in duplicate ACKs sent back to the sender.

One statistic used to represent packet re-ordering is ~~ally derived quantity for re-ordering, is~~ the average number of singleton re-order events over a series of sent packets. The re-order average corresponds to the re-order rate, which is given as a percentage between 0% and 100%. Unfortunately, applying this percentage to traffic analysis is difficult, particularly because different sources calculate the percentage in different ways.

More detailed, and potentially more useful, ~~Other statistically derived~~ quantities include the re-order patterns, showing how often the re-ordering occurs and how many packets were involved each time. The IETF IPPM is developing techniques ~~Techniques are being developed~~ to quantify the patterns including calculating the average value of the re-ordering and the average spacing between re-ordering events. These should be represented under re-order pattern.

13. Closeness

A number of researchers have sought to combine multiple characteristics into a single number to represent a notion of network-distance. All such characteristics meet the requirement that they produce a single quantity that is indicative of the “distance” or “closeness” of two nodes in a network. For example, Ferrari [12] proposes closeness as a function combining available bandwidth with roundtrip delay. In principle, an application could make a choice between two data providers by using any single distance characteristic that measures each possibility. In practice, an application might choose a particular “closeness” characteristic that weighs bandwidth, latency, long transfers, or short transfers according to that particular application’s needs.

14. Security Considerations

There are important security concerns associated with the making and distribution of network measurement information. For example, ISPs frequently consider network configuration and performance information to be proprietary. Furthermore, observing traffic, and in particular collecting packet headers, is frequently considered a violation of the presumption of privacy on the network. Systems that collect the measurements described here are sometimes regarded as invasive, and indeed poorly designed or configured monitoring tools can consume a disproportionate amount of network bandwidth. Port blocking, protocol blocking, and traffic shaping can impact many measurement tools. Tools, such as traceroute, that send UDP probes to increasing port numbers can appear to be port scans and raise security alerts.

We do not address those concerns in this document, but implementers are encouraged to consider the security implications of making and distributing measurement information. While distribution of end-to-end application-level measurements is generally accepted, measurements that identify individual users or consume noticeable amounts of resources should be taken carefully, and the distribution of information to other sites that cannot be obtained readily by other users at those sites should be considered carefully.

Author Information

Contact information for authors:

Les Cottrell Stanford Linear Accelerator Center cottrell@slac.stanford.edu	Richard Hughes-Jones University of Manchester R.Hughes-Jones@man.ac.uk Tel: +44 161 275 4117	Thilo Kielmann Vrije Universiteit kielmann@cs.vu.nl
Bruce Lowekamp College of William and Mary lowekamp@cs.wm.edu	Martin Swany University of Delaware swany@cis.udel.edu	Brian Tierney Lawrence Berkeley National Laboratory bltierney@lbl.gov

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

References

[1] G Almes, S Kalidindi, and M Zekauskas. A one-way delay metric for IPPM. RFC2679, September 1999.

[2] G Almes, S Kalidindi, and M Zekauskas. A one-way packet loss metric for IPPM. RFC2680, September 1999.

[3] G Almes, S Kalidindi, and M Zekauskas. A round-trip delay metric for IPPM. RFC2681, September 1999.

[4] J. C. Bolot. Characterizing end-to-end packet delay and loss in the internet. *High-Speed Networks*, 2(3):305–323, 1993.

[5] Yuri Breitbart, Minos Garofalakis, Cliff Martin, Rajeev Rastogi, S. Seshadri, and Avi Silberschatz. Topology discovery in heterogeneous IP networks. In *Proceedings of INFOCOM 2000*, March 2000.

[6] Ann Chervenak and GGF: Data Replication Research Group. An architecture for replica management in grid computing environments. http://www.sdsc.edu/GridForum/RemoteData/Papers/ggf1replica.pdf.

[7] Mark Coates, A O Hero III, Robert Nowak, and Bin Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, May 2002.

[8] R. L. Cottrell and Connie Logg. A new high performance network and application monitoring infrastructure. Technical Report SLAC-PUB-9202, SLAC, 2002.

[9] R. L. Cottrell and M. Shah. Measuring rtt by using syn/acks instead of pings. http://www-iepm.slac.stanford.edu/monitoring/limit/limiting.html#synack, December 1999.

[10] C. Demichelis and P. Chimento. Ip packet delay variation metric for ippm. ~~http://www.ietf.org/internetdrafts/draft-ietf-ippm-ipdv-09.txt~~RFC 3393, November~~April~~ 2002.

[11] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. What do packet dispersion techniques measure? In *IEEE INFOCOM 2001*, 2001.

[12] Tiziana Ferrari and Francesco Giacomini. Network monitoring for grid performance optimization. *Computer Communications*, 2002. submitted for publication.

[13] GGF. Grid scheduling area. http://www.mcs.anl.gov/˜schopf/ggf-sched .

[14] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[15] T. J. Hacker and B. D. Athey. “The end-to-end performance effects of parallel tcp sockets on a lossy wide-area network” Proceedings of the 16th IEEE-CS/ACM International Parallel and Distributed Processing Symposium (IPDPS) 2002.

[16] Internet2 end-to-end performance initiative. http://e2epi.internet2.edu/

[17] Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang. On the placement of internet instrumentation. In *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[18] Jun Liu and Mark Crovella. Using loss pairs to discover network properties. In *ACM SIGCOMM Internet Measurement Workshop 2001 (IMW2001)*, November 2001.

[19] Bruce Lowekamp, David R. O’Hallaron, and Thomas Gross. Topology discovery for large ethernet networks. In *Proceedings of SIGCOMM 2001*. ACM, August 2001.

[20] [M. Mathis and M. Allman. A framework for defining empirical bulk transfer capacity metrics. RFC3148, July 2001.

[21] S. Ostermann. tcptrace. http://irg.cs.ohiou.edu/software/tcptrace/tcptrace.html.

[22] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An investigation of geographic mapping techniques for internet hosts. In *Proceedings of ACM SIGCOMM 2001*, pages 173–185, 2001.

[23] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP performance metrics. RFC2330, May 1998.

[24] J. Postel. Internet control message protocol. RFC792, September 1981.

[25] R. Prasad, M. Murray, C. Dovrolis, and K. Claffy. Bandwidth estimation: metrics,

measurement techniques, and tools. In IEEE Network, June 2003.

[26] Stefan Savage. Sting: a tcp-based network measurement tool. In *USENIX Symposium on Internet Technologies and Systems*, pages 71–79, Boulder, CO, October 1999.

[27] Gary Shao, Fran Berman, and Rich Wolski. Using effective network views to promote distributed application performance. In *Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, 1999.

[28] M. Swany and R. Wolski. Representing dynamic performance information in grid environments with the network weather service. 2nd IEEE International Symposium on Cluster Computing and the Grid, May 2002.

[29] Web100 Project team. Web100 project. http://www.web100.org.

[30] Wolfgang Theilmann and Kurt Rothermel. Dynamic distance maps of the internet. In *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[31] R. Wolski. Dynamically forecasting network performance using the network weather service. *Cluster Computing*, 1:119–132, January 1998

[32] EU DataGrid and EU DataTAG work on end-host performance. See for example R. Hughes-Jones et al. Performance Measurements on Gigabit Ethernet NICs and Server Quality Motherboards. PFLDnet Workshop, February 2003 http://datatag.web.cern.ch/datatag/pfldnet2003/papers/hughes-jones.pdf

[33] ISO/IEC 7498-1:1994 Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=20269&ICS1=35&ICS2=100&ICS3=1

[34] Jiri Navratil & R. Les Cottrell "ABwE: A Practical Approach to Available Bandwidth Estimation", SLAC-PUB 9622, submitted to PAM2003.

[35] M. Laor, L.Gendel "The Effect of Packet Reordering in a Backbone Link on Application Throughput" IEEE Network Sep 2002

[36] John Bellardo and Stefan Savage "Measuring Packet Reordering", http://www.cs.ucsd.edu/users/savage/papers/IMW02.pdf

[37] Source code and documentation of pchar is available at http://www.employees.org/~bmah/Software/pchar/ .

[38] [Source code and documentation of pathrate is available at http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/ .

[39] Source code and documentation of iperf is available at http://dast.nlanr.net/Projects/Iperf/ .

[40] Source code and documentation of UDPmon is available at www.hep.man.ac.uk/~rich/net .

[41] Dan Gunter, James Magowan, and the DAMED-WG, "An Analysis of 'Top N' Event Descriptions," http://www-didc.lbl.gov/damed/documents/TopN_Events_final_draft.pdf, February 2003.