GWD-I (NMWG Internal Draft)
Network Measurements Working Group
http://www-didc.lbl.gov/NMWG/
http://www-didc.lbl.gov/NMWG/docs/measurements.pdf

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
UCSB
October 16, 2002

A Hierarchy of Network Measurements for Grid Applications and Services DRAFT

Status of this memo

This memo is an internal draft for GGF6 and is intended for submission to GFSG following GGF6.

This memo provides information to the Grid community regarding current practices of network measurements used for network performance. It does not define any standards or make recommendations. Distribution is unlimited.

What's changed in this mid-GGF6 draft!

This mid-GGF6 draft has revised figures to reflect the changes suggested during the first working section. Most text has not been updated to reflect those changes, but notes have been made to indicate the type of change expected. Those notes are in red. All changes since the first GGF6 draft are marked with changebars.

What's changed since GGF5

The major changes in this version of the document include the separation of network entities from the characteristics hierarchy, resulting in a different way of expressing topologies. Sections 4, 5, and 6 have been rewritten to reflect those changes. Minor changes to the existing sections on each characteristic have been made and new sections on queues, availability, and others have been added.

1 Introduction

Add material to discuss importance of simplicity and ability to represent measurements taken, rather than complete correctness or representation only of correct and useful measurements

This document describes a naming system needed for the description of network measurements 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. The application of this naming system will facilitate the creation of common schemas 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. This document is the first product of the Network Measurements Working Group (NMWG). A future NMWG document will define mappings between available tools and the measurement methodologies they implement.

The NMWG focuses on existing and currently used network measurements. It does not attempt to define new standards or to define only the best measurement methodologies to use for grid applications. It does attempt to point out the advantages and disadvantages of different measurement methodologies. The NMWG is closely related to the IETF Internet Protocol Performance Metrics (IPPM) WG, however their focus is on defining best-practices metrics of use to network engineers, whereas the NMWG and this document focus on existing practices and 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.

The NMWG focuses on network measurements, however many measurement tools' results 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].

2 Security Considerations

There may be important security concerns associated with the distribution of network measurement information. We do not address those concerns in this document, but implementers are encouraged to consider the security implications of distributing measurement information.

3 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 predictions 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 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 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 paths, 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 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 [28] project aims to extend this by making more information available to the TCP stack and to external programs.

3.1 How this document 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-level interface, such as that provided by NWS [30], provides the necessary information.

Currently there exist a variety of APIs for collecting network information from network tools and Grid Information Services, none of which are compatible. Our goal is to provide 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. We believe that our nomenclature would also be well suited to 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 [27], however we believe that this nomenclature is as well-suited for annotating information contained in other schemas 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.

The network characteristics nomenclature presented here 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 the proposed nomenclature, measurements can be classified and published. Other systems familiar with the specific measurement methodology will be able to use them as intended. Systems not familiar with that particular methodology 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 use of such a nomenclature requires continued work towards the NMWG’s goals, as well as integration with the work of other GGF working groups. We submit this nomenclature to the Grid and Internet communities as an agreed-upon standard of the types of measurements in use, to allow both current and future measurement methodologies to classify their observations to maximize their portability.

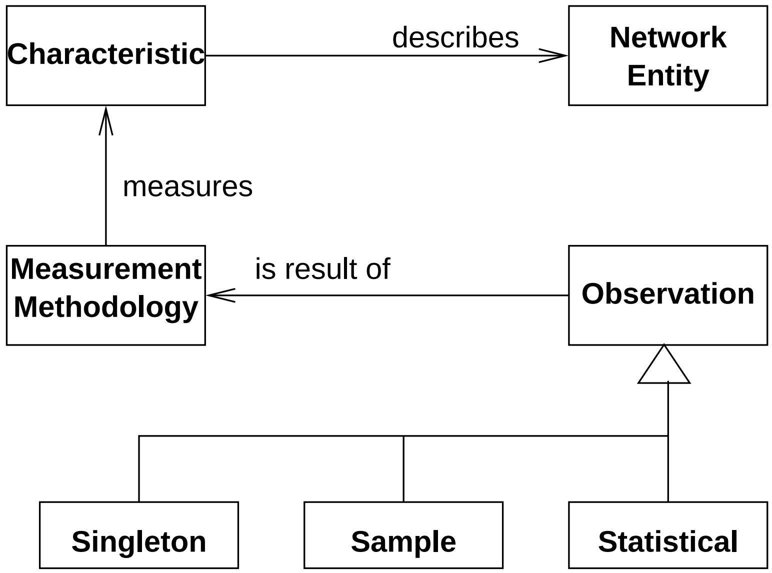

Figure 1: The relationship between the terms used in this document.

4 Terminology

Before discussion and definition of network measurements can begin, we first define the relevant terms. The different research backgrounds of people participating in GGF necessarily bring slightly different terminology. 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 Internet. *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 described by Figure 1. We use UML notation for the diagrams in this document.

It is important to note that 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 hops and paths (end-to-end) where appropriate. As illustrated in Figure 1, characteristics are applied to Network Entities, a general term that encompasses nodes, paths, autonomous systems, hops, etc.

Aside: We considered the terminology specified by the IETF IPPM RFC2330 [23]. However, in actual use we found that it wasn't ideally suited for our application. We 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 RFC2330 wherever possible.*

4.1 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. However, as discussed in Section 10, loss can be expressed generally as a fraction of all traffic sent, or specifically with detailed statistical properties.

4.2 Measurement methodology

raw or derived doesn't really matter as far as the characteristic

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." Raw measurement methodologies use a technique which 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 (as performed by,e.g., pchar and pathrate).

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 hop propagation data and queue lengths. Each of these techniques is a separate measurement methodology for calculating the roundtrip delay characteristic.

4.3 Observations

Singleton is smallest observable, not precisely atomic

An instance of output from a measurement methodology is an observation. An observation can be a *singleton*, which is an atomic observation, a *sample*, which is a number of singletons 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 needs to be attributed with timing information, indicating when a certain observation has been made. For singleton observations, a simple timestamp may be sufficient. For statistical observations, both the beginning and end of the observation time interval need to be reported.

4.4 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 measurements methodologies under separate characteristics are not directly interchangeable (although may be used along with other characteristics to derive those values).

Two common characteristics are capacity and utilization. Note the relationship between these two characteristics and available bandwidth. Available bandwidth can be defined in terms of capacity and utilization (using utilization in a general sense to represent traffic utilizing the path). Using the above definitions of characteristic and measurement methodologies, available bandwidth is a characteristic because it has a well-specified place in the hierarchy and because its measurements are not equivalent to those of any other characteristic. 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 may be a characteristic, but the techniques are measurement methodologies.

For example, consider the question of whether TCP-capacity (as opposed to link-capacity) is a characteristic or a measurement methodology. Although the maximum bandwidth that a TCP connection can achieve over a particular hop is important to many applications, it is not truly a characteristic. In particular it is a function of the path capacity and delay, the TCP implementations used by both sending and receiving machines, and the power of the machines at each endpoint. Therefore, link-level capacity is the true characteristic, while the other factors determine how an application-level observation relates to the hop capacity characteristic.

5 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, host, router, etc. Although to a typical user, the entity being measured may be obvious, it may actually be difficult to determine what is being measured. Issues in determining the network entity being measured can include:

- Choice of protocol can influence a network's behavior.
- Different QoS levels affect all aspects of network behavior, including using completely different routes.
- 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 measurement systems to interchange information about network measurements 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 the conditions under which measurements are taken, such as protocol and QoS.

not all conditions, try to include those that may change node or path

5.1 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 also

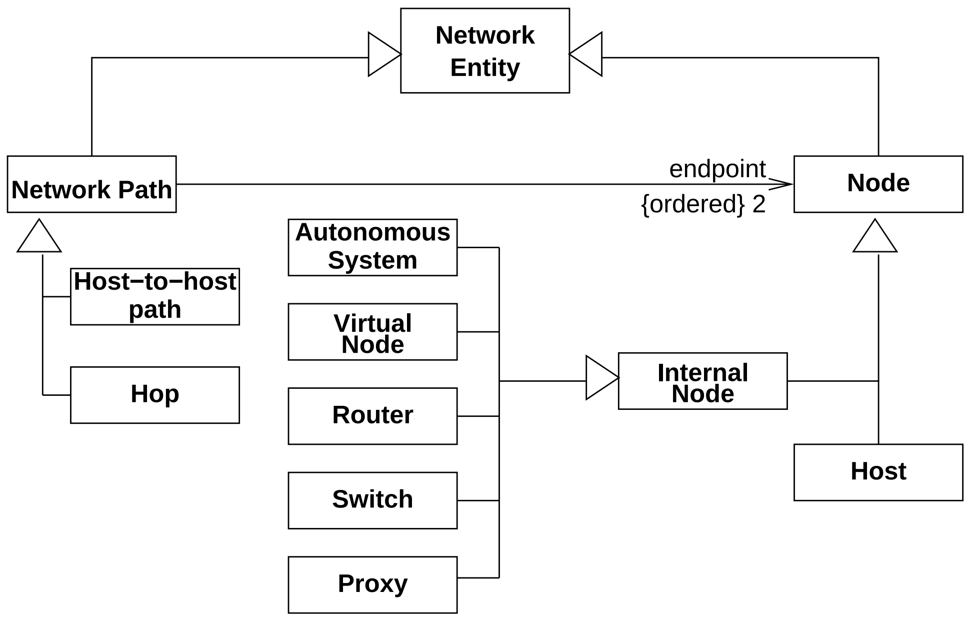

Figure 2: Network entities are divided into nodes and paths. A node can be many things, ranging from an autonomous system down to a host or switch. A path is a unidirectional connection from one node to another node. Paths and nodes are both annotated with protocol and QoS illustration so that network behavior with different types of traffic can be represented.

represent an autonomous system or a virtual node. A path exists between the two nodes used as measurement endpoints. The type of the path, and the hops it follows, is irrelevant to the definition of the entity.

Attributes are those needed to define path uniquely, not necessarily all attributes that are important to measurement. Intent is that measurement will include attributes (even those that are associated with Host may be seen differently by different tools.)

Nodes and paths are annotated with attributes including protocol stack (e.g., TCP over IPv4) and QoS level. Note that the attributes are not merely descriptive of the node or path, but form a tuple that define a unique node or path. In other words, the model superimposes multiple topologies over the actual physical nodes and paths, allowing the behavior of the network under different types of traffic to be characterized separately. Also note that some characteristics, such as capacity, may be most appropriately described without specifying either a protocol or QoS level.

5.2 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.

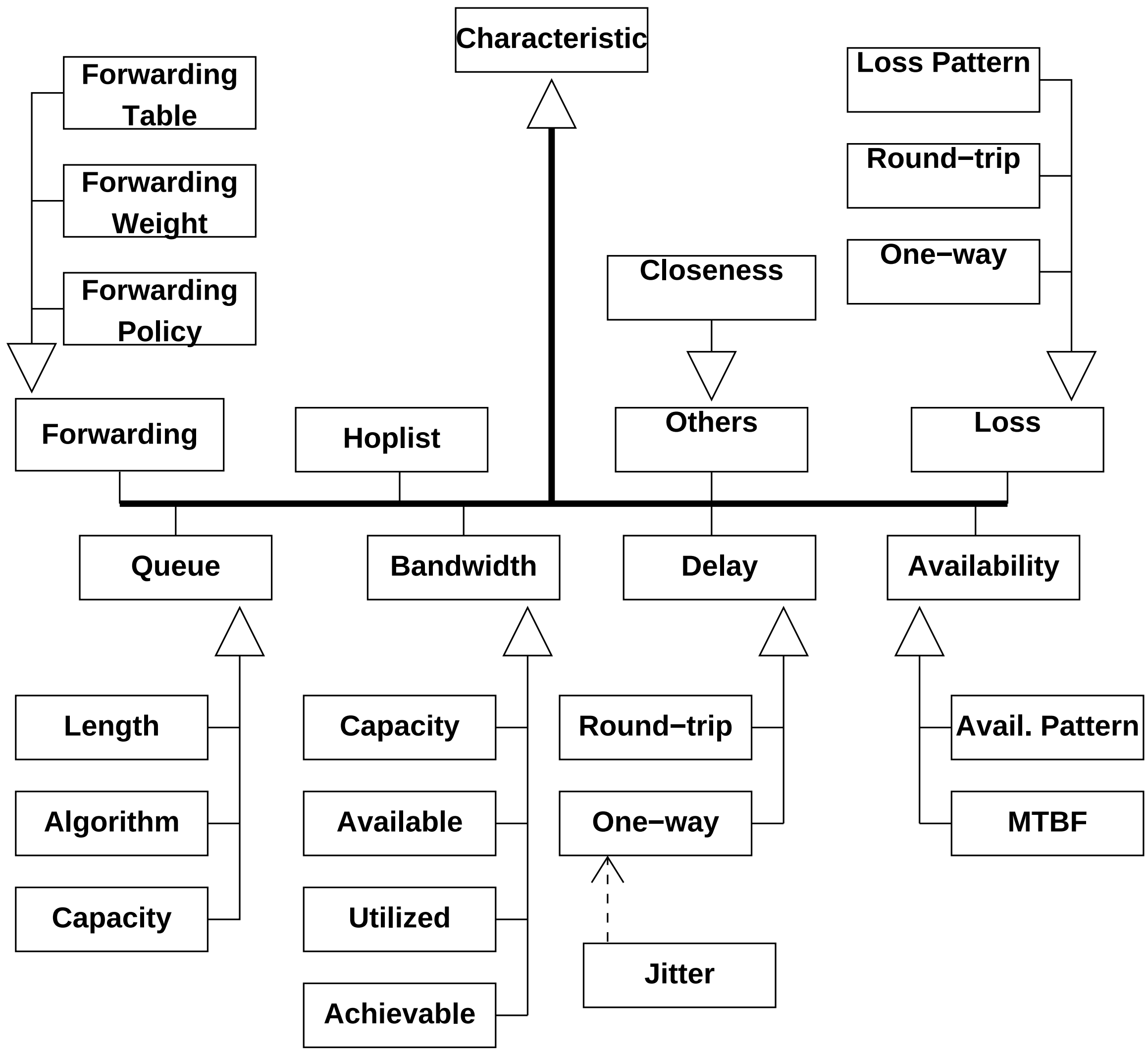

Figure 3: Network characteristics that can be used to describe the behavior of network entities.

5.3 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 this 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 behavior 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.

6 Topology and Route Characteristics

Point-to-point, not multicast or broadcast

differentiate from IPPM framework in terms of definitions

As indicated in Figure 2, a network topology is composed of nodes and paths. Nodes and paths are annotated with attributes including protocol and QoS level. Note that the attributes are not merely descriptive of the node or path, but form a tuple that define a unique node or path. In other words, the model superimposes multiple topologies over the actual physical nodes and paths, allowing the behavior of the network under different types of traffic to be characterized separately.

Paths are used to represent the connection between any two nodes in the network. This can include 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-level paths allows a detailed model of a network's structure to be built. The only characteristic describing topology is route, but because the process of describing a topology relies on the relationship between network entities and routes, this section discusses node and paths as well as routes.

6.1 Physical and Functional Topologies

There are two approaches to characterizing network topology: physical and functional. The physical approach determines the physical links that connect the network together. By determining the connections between links, along with their capacities, queuing algorithms, and traffic load, the network can be modeled and its behavior analyzed 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 modeling low-level network behavior. 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 links. This approach may be taken across a variety of sites distributed around the Internet, or using a single-source tree [7, 17, 26, 29]. 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 virtual topologies is that all of the internal nodes are virtual nodes, as shown in Figure 2.

Add attribute to indicate physical or virtual

Indicate attributes of protocol stack and QoS required.

6.2 Paths

Paths are now broken up into host-to-host paths and hops. There is no proposed difference between them in terms of attributes, but they allow specification of what type of “path” is being referred to.”

Most measurement methodologies are expected to indicate their source and destination nodes by the IP address, DNS name, (or maybe MAC address?) that they are using. If a node has multiple IP addresses assigned to it (regardless of whether they are actually separate interfaces), this technique lacks precision, but it is what is observed by most methodologies. A methodology aware of such issues should report the correct address. Full details are expected to be provided in the forwarding characteristic.

The building block of a topology graph are the paths between nodes. Each path identifies a source and destination node. Paths are unidirectional for topology purposes. In the cases where the characteristics describe a round-trip property, the path’s source is the originating node for the characteristic.

6.3 Nodes

mention about hosts that are also routers, and separate internal node vs host into separate paragraph

mention about using virtual/functional nodes inside physical topologies

Nodes are generally classified into hosts and internal nodes. Internal nodes are capable of forwarding traffic, whereas hosts are considered to be only endpoints of communication. Internal nodes can be routers, switches, or proxies, as well as more general concepts such as autonomous systems or virtual nodes in a topology. Like paths, nodes also are defined along with a set of attributes. The attributes indicate a specific set of characeristics to the node when handling a particular type of traffic. In particular, queuing algorithm, lengths, etc. may vary quite significantly according to type of traffic and QoS level.

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. Furthermore, virtual nodes may play a role in describing physical networks. For example, 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.

6.4 Hoplist

Routes is renamed hoplist. Hoplist can be a set of hops, because hops are now a type of path.

The route characteristic allows a path, such as an end-to-end path, to be subdivided into the hops that build the path. For example, a path between two end-hosts could have a route characteristic containing the paths between each of the routers along the end-to-end path. A route is an ordered set of paths, with the source node of the first path and the destination node of the last path being the source and destination node of the described path.

End-to-end paths are not the only paths that can be divided into the hops of a route. For example, one of the above router-to-router paths could be further subdivided into links between Ethernet switches in a bridged LAN.

6.4.1 Path Unification

The “route” 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 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 uniquely identify the individual routers. 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, which each indicate a single link. Assuming the router’s switching fabric is not itself a bottleneck, this detail may be unimportant. However, if traceroute measurements are being combined with other tools, difficulty may be experienced combining the information 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 it is worth noting that comparing destinations of links may be more useful than sources due to the way traceroute works [14].

7 Forwarding

Use forwarding as generic term for property to forward data, can be applied at layer 2, 3, or even application. Table would be used for routing table, forwarding database, or NAT table. Weight for routing metrics. And policy to indicate algorithm run or perhaps more detailed information. It will be necessarily up to schema implementers to create more concrete definitions—which may vary a lot from layer to layer and protocol to protocol.

Interface information is now provided by the forwarding characteristic. This is true for any device with multiple interfaces. In the case where a node could use multiple interfaces to reach another node, the forwarding table and policy should typically indicate how use choices are made, although the existence of a path object between two nodes may clairfy that.

Routing metrics can also be used as a characteristic of paths. Most routing protocols present cost metrics along with their reachability information. For instance in the Open Shortest Path First (OSPF) protocol, the cost metric for each link (path) is represented as an integer distance. In the case of interdomain 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.

8 Bandwidth Characteristics

Add layer as a mandatory attribute for each characteristic. Capacity makes sense at each level, gradually adding framing and spacing overheads

Available will be changed to mean unutilized. It can be argued it is, therefore, redundant, however it is possible to produce a measurement of unutilized bandwidth without directly measuring capacity or utilization.

Add new characteristic of “achievable” to capture what a particular protocol can do for an application. Also intended to indicate that achievable would be the characteristic where host issues (bus, protocol stack, etc) would be represented. Unclear that this is meaningful at multiple layers.

Bandwidth is defined most generally as data per unit time. However, the “bandwidth of a path” is not a precisely defined term and specific characteristics must be clearly defined prior to discussing how to measure bandwidth.

There are three characteristics that describe bandwidth:

Capacity: The maximum amount of data that a path can carry

Utilization: The aggregate traffic currently on that path.

Available Bandwidth: The maximum throughput that the path can provide to an application, given the path’s current load (utilization). Can be measured directly or estimated from capacity and utilization.

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

8.1 Capacity

Capacity is the theoretical maximum link-layer throughput that the 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 needed to determine how to set optimal TCP buffer sizes.

8.1.1 Capacity Measurements

Hop capacity can sometimes be obtained directly via SNMP queries to the network switches and routers along the path. However in general it is not possible get access to this information in a commercial ISP’s network.

There are a variety of techniques for inferring capacity, most of which are based on “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 behavior, and estimate path characteristics based on this information. For example, the bottleneck capacity hop separates the 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 closest spacing of the packets. Actual implementations based on these techniques apply a variety of statistical techniques to filter out the noise and random behavior 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 measurements. 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.

8.2 Utilization

Clarify/remove/edit the whole utilization is 0 or 1. This is clearly untrue in the limit as dt-¿ 0, anyway.

Utilization is the aggregate capacity currently being consumed on a path. As a singleton observation, utilization is merely a boolean value. More useful is a statistical observation, which summarizes utilization as a fraction between 0 and 1 over a particular time interval.

More complex representations of utilization are possible. For example, a profile of traffic on a path during a particular time interval is also a way of observing utilization, however at significantly more detail than the simple proportion of capacity in use. It is an open question for this document how to organize traffic profile vs proportion within the characteristic hierarchy. We propose that traffic profile be represented underneath utilization, because the simpler proportion can always be derived from a more detailed representation of the traffic utilizing the path.

8.2.1 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 in general it is not possible get access to this information for a commercial ISP's network.

8.3 Available Bandwidth

Much of this will be applied to achievable, instead

Available bandwidth is the maximum throughput that a path can provide to a flow or a set of flows given the current traffic load, for the given protocol and QoS parameters of a measurement. In the context of a path consisting of several hops, the hop with the minimum transmission rate determines the capacity of the path, while the hop with the minimum unused capacity limits the available bandwidth. Other groups have defined available bandwidth as the unutilized bandwidth on a hop, which is a valid definition but typically not the information an application needs, and is easily calculated from the capacity and utilization characteristics.

The IETF IPPM defines a TCP measurement of the available bandwidth characteristic: "Bulk Transfer Capacity" (BTC) [20]. The specific definition of the bulk transfer capacity is:

$$\mathrm{BTC} = \mathrm{datasent}/\mathrm{elapsedtime}$$

This effectively tries to capture the "steady state" of a long-lived flow—amortizing out the constant of overhead.

This definition underscores a problem with our generalizations. Users tend to think of a measure *Availability* as identical to "Bulk Transfer Capacity." The problem is how this characteristic is derived. There are many tools that measure what can be thought of as available bandwidth, although their results may vary significantly with their methodology. This is the strength of using a hierarchical arrangement in that measures of availability are grouped together by their common goal, but are differentiated closer to the leaves of the tree.

Tools to measure available bandwidth fall into two general categories:

- Flow based (or Connection-Oriented)
- Datagram based (typically packet trains)

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

8.3.1 Flow Measurements

One of the most commonly used flow-oriented measurement is available TCP bandwidth. Available TCP bandwidth is both one of the most useful and yet most meaningless of all network measurements. A common method of measuring available 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*, and so on. There are a number of problems with this technique:

- Results are greatly affected by TCP implementation. As is 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.

- 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 to get a reasonable estimation of available bandwidth on a WAN using iperf requires about 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.

- TCP-based tools only measure end-to-end bandwidth, where often one wants to have hop-by-hop information as well.

Other types of stream-oriented measurements use datagrams to simulate TCP flows or to quickly saturate the network. We make the distinction that if the aggregate behavior 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.

So, 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. 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 NIC used. If this were the case, 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.

8.3.2 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. There is, however, some question about their ability to measure availability in all situations [11].

9 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 metric provides an indication of the delay due only to propagation and transmission delay.
- The minimum value of this metric provides an indication of the delay that will likely be experienced when the path traversed is lightly loaded.
- Values of this metric 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:

- 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 since 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.

9.1 One-way delay measurements

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 usecs) the clocks are often synchronized with GPS. If the packet is not received at the destination within a reasonable period of time, then the one-way delay is undefined (informally, infinite according to RFC 2679), and the packet is taken to be lost.

9.1.1 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 [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 since 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.

9.2 Roundtrip delay measurements

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.

Most modern IP stacks implement an ICMP echo responder in the ICMP server [24]. Upon seeing an ICMP echo request packet, the ICMP 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 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 [28] 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.

9.3 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 a 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 maybe 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 millisecond). Thus delay measurements on short links 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. It is also useful to keep track of the occurrences of such events.

Some NIC cards coalesce interrupts to reduce the interrupt load on the CPU. This can lead to aggregations of delays since packets may be held at the NIC card, until some criteria is reached, before they are delivered to the host. In such cases it may be important to use the NIC card to do the packet timing.

10 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 metric and statistically derived quantities over time series of those singleton metrics need to be taken into account. The singleton metric 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. 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 sustainable bandwidth. Depending on the application itself, the above-mentioned threshold values can vary significantly.

The singleton packet loss is a binary metric. 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 metric 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 metric has the value 1, if S did not receive the reply packet.

10.1 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 behaviors of the two directions. Roundtrip measurements therefore mix the properties of both path directions, possibly producing misleading results. (For example, the sustained bandwidth of a file transfer may strongly depend on the packet loss on the path from source to destination, while being largely independent of the reverse 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.

10.2 Roundtrip loss

From a network-centric viewpoint, the more useful loss metric 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-level) 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 metrics 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.

10.3 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, 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.

10.4 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. Since packets of different protocols are supposed to be forwarded differently by routers in the Internet, loss properties need to 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 routers frequently treat differently than regular TCP or UDP data.

- **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 [25] 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-level 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.

11 Availability

Discuss sub-characteristics as given in document. Relate to IPPM connectivity, basically as boolean at root.

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 as "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 link 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.

12 Queueing Information

In wired networks, queue overflow in routers is the predominant reason for packet loss. To model network behavior, 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 algorithm used by the router. Derived information can be obtained by sending traffic through the router and deducing its behavior through analytical means [18]. However the information is obtained, it does not effect the characteristic it attempts to determine, although measurements should indicate how the information was derived.

To describe a queue, we include three characteristics: algorithm, 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 queueing algorithms, while this document can suggest general characteristics, the appropriateness of these characteristics and the need to add additional per-algorithm characteristics or extend our general characteristics must be evaluated according to the nature of the algorithm being represented.

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

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 behavior of different queueing algorithms may be necessary for each different algorithm.

13 Other Characteristics

There are many non-standard characteristics that are developed to represent various other properties that are considered important. We group these in the "other" category and suggest that schemas implement a technique for allowing the representation of arbitrary characteristics using an ASCII description.

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 [12]. For example, Ferrari 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, it might be better to chose a particular characteristic that weighs bandwidth, latency, long transfers, or short transfers according to the application's need.

14 Full Copyright Notice

15 Intellectual Property Rights

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.

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/internet-drafts/draft-ietf-ippm-ipdv-09.txt, 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.

[16] Internet2 end-to-end performance initiative. http://www.internet2.edu/e2eperf/.

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

[26] 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.

[27] 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 (to appear), May 2002.

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

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

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