Towards a Framework for Defining Internet Performance Metrics

Vern Paxson
Network Research Group
Lawrence Berkeley National Laboratory*
vern@ee.lbl.gov

LBNL-38952

June 20, 1996

Abstract

The Internet's tremendous growth represents a triumph of standardization, since it is only through standardization that so many different networks using so many different designs can smoothly exchange data. The standardization of Internet measurement, however, has not matched the explosive growth of the network as a whole. Even such basic notions as how to measure the throughput or delay along an Internet path lack a standardized framework. Instead it has become increasingly difficult to diagnose problems or determine whether one is receiving promised performance.

In this paper we outline how a measurement framework might be developed to support Internet diagnosis and performance evaluation. We propose terminology to use in defining standards, including the key notions of *metric* as the fundamental property we wish to measure, *methodology* as a way to attempt to measure the property, and *measurement* as the result of a specific application of a methodology. We develop a basic contrast between *analytically-specified* metrics, which emphasize viewing network properties in analytic terms, and *empirically-specified* metrics, which correspond to properties that are generally too complex to discuss analytically but still very important for practical measurement. Each has its place in the framework.

We further discuss the notion of *composition* (how a property we wish to measure might be fruitfully viewed in terms of a collection of simpler, underlying properties), the crucial issues of measurement errors and uncertainties, and the pros and cons of different measurement *strategies,* including the degree of *cooperation* they require. We also sketch a proposal for how one might architect a *measurement infrastructure* for the Internet. We finish with proposed formalisms for defining Internet metrics and methodologies, illustrated with an example of defining an *Internet route* metric and an accompanying methodology based on the `traceroute` utility.

1 Introduction

The Internet is a huge collection of interoperating networks, with 9.5 million computers at last count. As such, it represents a triumph of standardization, since it is only through standardization that so many different networks using so many different designs can smoothly exchange data. The standardization of Internet measurement, however, has not matched the explosive growth of the network as a whole. Even such basic notions as how to measure the throughput or delay along an Internet path lack a standardized framework.

To address this shortcoming, the Benchmarking Methodology Working Group of the Internet Engineering Task Force has created an IP Provider Metrics (IPPM) effort aimed at developing such a framework. One of the main goals of this effort is to provide a basis for evaluating the performance of different Internet components, particularly "IP clouds" that provide Internet connectivity to external networks in an opaque fashion. Such standardized performance evaluations can serve a number of needs, including:

- aiding trouble-shooting and capacity planning in a complex world of tens of thousands of networks and interconnecting links;
- providing market incentives for network service providers to optimize their networks, by giving Internet customers sound techniques for evaluating the service they are receiving and comparing the performance of different providers;
- enabling Internet research geared towards a better understanding of the behavior of network traffic and how the Internet evolves.

---

*This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. An earlier version of this paper appeared in the *Proceedings of INET '96.*

In this paper we outline how such a measurement framework might be developed. The discussion is necessarily heavy on terminology, since a large element of successful standardization is unambiguous descriptions of the standards. We also emphasize that the discussion here is *preliminary*. This version of the paper is only a *draft* of the underlying ideas, and does *not* reflect any standardization produced by the IPPM, though the intent is to influence that standardization process.

The discussion proceeds as follows. We first define a number of terms, both those concerning the Internet in general (§ 2) and those concerning the measurement framework (§ 3). In the latter we develop the key notions of *metric* as the fundamental property we wish to measure, *methodology* as a way to attempt to measure the property, and *measurement* as the result of a specific application of a methodology.

We then develop a basic contrast between *analytically-specified* metrics (§ 4), which emphasize viewing network properties in analytic terms, and *empirically-specified* metrics (§ 5), which correspond to properties that are generally too complex to discuss analytically but still very important for practical measurement. Each has its place in the framework (§ 6). Because the notion of an empirically-specified metric is quite similar to the notion of "methodology," § 7 briefly expands on the distinction between the two.

§ 8 discusses the notion of "composition": how a property we wish to measure might be fruitfully viewed in terms of a collection of simpler, underlying properties. We follow this with a discussion in § 9 of the crucial issues of measurement errors and uncertainties; these in general cannot be avoided and must instead be quantified whenever possible if we are to make sound measurements. We then turn to a high-level discussion of different measurement "strategies" (§ 10) including a taxonomy of methodologies as "passive" or "active", and whether they require "soft" or "hard" cooperation (or no cooperation at all). These facets of methodologies directly influence some of the errors inherent in the methodology, and also the ease of deploying the methodology in the Internet. This discussion leads in turn to a proposal for how one might architect a "measurement infrastructure" for the Internet (§ 11), which could provide for sophisticated measurements in a relatively inexpensive (and realistic) fashion. We also discuss what sort of factors could lead to widespread deployment of the infrastructure.

We finish with proposed formalisms for defining Internet metrics and methodologies (§ 12), illustrated with an example of defining an "Internet route" metric and accompanying methodology based on `traceroute`. As with the rest of this paper, these are intended not as final standards but to stimulate further discussion.

2 Some basic terms

In this and the next section we define terminology used in the remainder of the discussion. This section covers basic networking terms not specific to measurement, and the next session covers measurement-specific terms.

Internet host or **host** A computer capable (if all is working properly) of communicating using the Internet protocols. Includes **routers**.

Link The link-level abstraction of a "virtual direct connection" between two or more Internet hosts. Often thought of in terms of a single underlying physical connection, though it need not be so realized.

Router An Internet host that facilitates communication between other Internet hosts by forwarding packets from one link to another.

Internet path or **path** The network-level abstraction of a "virtual link" from host A to host B. That is, the Internet Protocol (IP) makes it appear to higher levels as though A has a *direct* connection to B. This apparent direct connection is a "path." The notion of "path" is a unidirectional concept—that is, the path from A to B is distinct from the path from B to A.

Route A sequence of links and routers comprising an Internet path.

IP Cloud or **Cloud** A collection of routers viewed as a "black box." Packets enter the cloud at well-defined entry points and later exit at well-defined exit points, if they were not dropped by the cloud. In principle, all of the routers within a cloud are internally connected.

3 Some measurement terms

We begin by defining the general notion of an "Internet component" as any element of an Internet network whose properties we wish to quantify. An Internet component may for example be a single computer such as a router, a large collection such as the links and routers comprising an IP cloud, or a service group such as a Network Operations Center (NOC).

We refer to different properties of an Internet component as *metrics*, with the term implying the use of standardized *units* when quantifying the metric. Quantified values of metrics as termed *measurements*. So, for example, a metric of a router might be its *forwarding rate*, defined as the number of packets per second it can receive from one link and send out on another link. A measurement of the forwarding rate metric might be "123,444 packets per second."

Unless care is taken, the notion of "metric" can prove quite slippery. For example, the above definition of forwarding rate neglects to mention the *size* of the packets being forwarded. This may or may not be relevant. If the router copies each incoming packet to a temporary buffer, then that copying might dominate its processing time, and the size of the packets is relevant. If however the router copies incoming packets but does so in parallel with the forwarding lookup, and if the lookup takes longer than copying a maximal packet, then the packet size does not matter.

We term metrics *well-defined* if they include all such pertinent factors, and *ill-defined* if they fail to do so. Unfortunately it will not generally be apparent whether a metric is indeed well-defined until it has been subjected to considerable use.

Equally important when attempting to measure metrics is the surrounding *context* of the measurement. We define a measurement's context as those elements of the complete system used to make the measurement that are in addition to the component being measured.

For example, a router fed by a single T1 circuit will never be able to forward faster than 1.544 Mbps, since that is the upper bound on the rate of arriving packets. If the minimal sized packet is 64 bytes, then the measured forwarding rate will be at most 3,015 packets per second, regardless of the speed of the router's internals. Furthermore, suppose the router does forwarding lookups in parallel, as discussed above, and that these lookups exceed the cost of the internal copying. The *measured* forwarding rate will still vary according to the packet sizes, because the incoming T1 circuit can deliver fewer large packets per second than small ones. In general, understanding the effects of context is *crucial* for making accurate measurements.

We term a process for quantifying a measurement for a metric as a "methodology." For a methodology to be *sound*, it must take into account all the relevant effects of the measurement context. (There may be additional requirements for soundness.) We refer to a methodology that produces erroneous results as *unsound*.

It is valuable to distinguish between two basic types of metrics, "analytically-specified" and "empirically-specified." We will refer to these as "analytical" and "empirical" metrics, for short.

Analytical metrics refer to those defined in terms of the theoretical, abstract properties of the components. These are the properties used to analyze the component mathematically. Empirical metrics, on the other hand, refer to properties directly defined by measurement. Each type of metric plays an important role in network measurement, and each type has its advantages and disadvantages. We expand on both types of metric in the next two sections.

4 Analytically-specified metrics

As discussed above, analytical metrics are those that view a component in terms of its abstract, mathematical properties. We limit the scope of metrics (both analytical and empirical) to properties defined in terms of the Internet "network layer," or at a higher layer. Some examples of analytical metrics:

Propagation time of a link The time difference in seconds between when host X on the link L begins sending 1 bit to host Y, and when host Y has received the bit.

Transmission time of a link The time required to transmit b bits from X to Y on the link L, as opposed to 1 bit.

Bandwidth of a network link A network link's data-carrying capacity, measured in bits per second, where "data" does not include those bits needed solely for link-layer headers. For an ATM link using AAL5 encapsulation, this metric would be 48 bytes/cell (= 384 bits/cell) times the cell rate. For links with variable sized transmission units, this metric is ill-defined unless a transmission size is also specified.

Flow capacity of a network path For a given Internet path from A to B, the maximum rate at which data can in principle be transferred along the path. (This will often be the "bottleneck" bandwidth of the slowest link in the chain of links comprising the path.) Note that here we have defined a fairly "high level" metric, but it remains an analytical metric because it is an analytic property of the component (the network path).

Maximum flow capacity of an IP cloud For a given entry and exit point, the greatest transmission rate achievable, in bits per second, if we had all of the links and routers in the cloud at our disposal (so we could send our data along multiple, parallel routes).

Instantaneous route of a network path The sequence of links and routers comprising the path from A to B at a given instant in time. Here the "units" might be IP addresses of the traversed router interfaces, with the links between them being implicit (but see § 12 for problems with this definition).

Hop count of a route How many routers a packet from A to B will visit along a particular route. (We might want to know this number in order to compute store-and-forward delays, for example.)

Buffer size of a router How many bits the router has available for buffering queued packets. Here we are modeling the router as a queueing server. In practice, the buffering might be specific to the outgoing interface, or it might be shared between the different interfaces, or it might be different for different flows or types of flows. These differences are often crucial, and illustrate some of the difficulties of devising well-defined analytical metrics.

Instantaneous queue size of a router interface At a given moment in time, the number of bits consumed at a router by packets queued for transmission on a particular interface. Again, we model the router as a queueing server.

Instantaneous connectivity of an Internet path Whether at a particular instant host A is able to send IP datagrams to host B. (The units here are Boolean.)

Epoch connectivity of an Internet path Whether over a given interval of S seconds starting at time T host A is able to send any IP datagrams to host B and have B receive one or more with non-zero probability. Note that this is significantly different from the previous definition. At any given moment, if an Internet path includes a router with completely full buffers, then there is no instantaneous connectivity along that path; but there may well be epoch connectivity during an interval containing that instant.

Maximum jitter along an Internet path The maximum amount of variation, measured in seconds, that packets sent from A to B might experience in their end-to-end transmission time.

Availability of an Internet path The unconditional probability that for any S second interval host A will have epoch connectivity to host B. Clearly a function of S. Since the metric is defined as a probability, it is unitless.

Mean NOC turn-around time The expectation of the amount of time in seconds between when a trouble ticket is submitted to a NOC and when the problem is resolved. An empirical version of this metric would be "the average amount of time historically taken by a NOC to resolve a trouble ticket."

Note that as phrased above, this metric is not conditioned on the *type* of trouble ticket. Clearly certain types of problems will be resolved more quickly than others. This shortcoming highlights the great degree of latitude that must be surveyed before standardizing performance metrics—as it is defined above, this metric is probably almost worthless.

These examples illustrate the wide scope of analytical metrics. Indeed, any parameter in a mathematical analysis of an Internet component immediately lends itself to an analytical metric. We are not advocating any of the above as necessarily *good* analytical metrics, just as examples, and it is important to keep in mind the utility of having a minimal set of useful metrics rather than trying to define every possible interesting metric. Similarly, while the emphasis in this section is on precise, analytic notions of metrics, we do *not* want the framework for Internet performance metrics to become bogged down in unwieldy formalism. But we believe that an emphasis on analytical metrics can play a crucial role in shaping measurement methodologies so that they achieve both generality and relevance for the future.

5 Empirically-specified metrics

In contrast to analytical metrics are empirical measurements, which are defined directly in terms of a measurement methodology. One example was given in the previous section, a metric of "the average amount of time historically taken by a NOC to resolve a trouble ticket." Here the measurement methodology is to record a running average of the time taken by the NOC to resolve each trouble ticket, and this number is also the empirical metric.

Another example is "the output of the `traceroute` program run on host A with the argument B." One might initially think that this empirical metric is the same as "Instantaneous route of a network path" discussed in the previous section, but it is not: the empirical metric is defined by a particular program (methodology), which actually does *not* measure the instantaneous route of a network path but only a close approximation to it. In a situation like this we would argue it is better to think of the `traceroute` program as a measurement methodology for assessing an analytical metric, rather than as directly producing an empirical metric, as that way the notion of "close approximation but not exact" is preserved (see § 12).

These first two examples of empirical metrics might appear to merely reflect pedantic hair-splitting. If these examples were all that one could do in terms of empirical metrics, the notion of empirical metric would not be worth developing. To illustrate its utility, we now turn to a more complex example.

Consider the very pragmatic question of how much throughput a user can expect to achieve across an IP cloud. Here we mean that the user has a data source transmitting at one of the cloud's entry points, and a data sink at an exit point, and wants to know how fast they can move data from the source to the sink.

We might initially think that this problem fits well into the analytical metric framework: the analytical metric of interest is "IP cloud throughput," and it is defined as "the number of bits per second that can be transmitted from a given entry point A of an IP cloud to a given exit point B." While this metric appears to capture the desired notion (by definition), it turns out this metric is not a useful concept, when expressed as an analytical metric. The reasons are as follows.

We assume that the user wants *reliable* transmission across the IP cloud, as is often the case. Generally, the way to do this is to use the TCP protocol. TCP, quite wisely, has built into it notions of congestion avoidance, in which it effectively adapts its transmission rate to current network conditions. The exact algorithms used for the adaption are subtle and vary between different implementations. Thus, the throughput the user can expect to see across the IP cloud is heavily shaped by:

- the use of TCP;
- how the flow and congestion control algorithms adapt

to the network conditions;

- the TCP software used (at both the source and sink; these two may be different);
- and the current network conditions in the cloud.

Regarding this last item, predicting how the cloud's network conditions will interact with the TCP implementations remains an open research area.

Thus we are in a quandary if we want to tackle throughput measurement using analytical metrics: the definition of the analytical throughput metric does not give us any insight into how to accurately measure it.

Other measurement communities, when confronted with the need to measure complex systems, have used the *benchmark* approach, in which some standardized applications are used to stress the system. The system's performance in executing the benchmark is then used as a metric for how well the system performs in general. With benchmarks, two key difficulties are the relevance of the benchmark to what the user really wants to know (i.e., how *their* application will perform), and the related possibility of systems being artificially tuned to perform well on specific benchmarks without a subsequent gain for more general applications. Nevertheless, when a more analytic description of a system's performance metrics remains elusive, benchmarks can provide valuable tools.

For measuring user throughput, one fruitful approach might be to use a "state of the art" TCP implementation to transfer data across an IP cloud and carefully measure the resulting throughput (and the factors that contributed to it). Here "state of the art" refers to the best current practices both for high-performance TCP and for congestion control. An example would be the `treno` tool, which attempts to do just this, and is described in [MM96]. We could then define an empirical throughput metric as something like "the throughput reported by version $X.Y$ of the `treno` tool when run with the following arguments . . ." (Even this definition is problematic because, for example, the tool could produce different measurements under identical network conditions due to other context factors such as host clock resolution.)

Thus, in our taxonomy, empirical metrics correspond to benchmarks, while analytical metrics reflect an attempt to evaluate the system in more analytically fundamental terms.

6 Analytical vs. Empirical metrics

Analytical metrics have a number of appealing properties. Foremost of these is that they include the possibility of developing an analytic framework for understanding different aspects of Internet behavior. Such a framework should not be sold short: without it, keeping the Internet functioning, improving its performance, and extending it for future traffic become seat-of-the-pants engineering problems, likely to prove disastrous when applied to such a huge system. Empirical metrics are much more likely to prove difficult to compose (see below) or to generalize how they will be affected by changes in network parameters.

Analytical metrics are also easier to define than empirical metrics. The latter necessarily can only be specified using a program, with all the attendant possibilities of hard-to-spot bugs in the definition, while the former can be specified in analytic terms.

A key problem with analytical metrics, however, relates to exactly this easier form of definition. At first blush it might seem straight forward to define an analytical metric, but often the underlying notions prove slippery. Consider the following subtleties associated with the analytical metric defined above:

Transmission time of a link The time required to transmit b bits from X to Y.

What if the link uses data compression (such as some modems do)? Then the transmission time for b bits depends on the bits being sent. The analytical metric might be redefined as "b zero bits," for example. This new definition is more well-defined, but still perhaps not too useful for measuring modem links, because the zero bits might compress extremely well, while "real-world" data might not. Another definition would be "b random bits," where presumably the randomization defeats any compression. This metric might better reflect real-world data (or might not, depending on the sophistication of the compression scheme). It is however a more cumbersome definition, since we must now introduce a suitable notion of "random."

Related to this example, analytical metrics present two distinct problems: the first is whether the metric is well-defined, meaning that its definition includes all of the relevant particulars necessary for capturing the notion of interest. The second is that, even if it is well-defined, it may be significantly removed from the real-world notion of interest. It is easy to overlook that an analytic notion does not match a real-world notion, such as transmission time in the face of possible compression. If an analytical metric fails in either of these ways, then not only is the metric not useful, but it may well mislead us because its failure is subtle.

Empirical metrics can suffer from these defects, but in a different way. An empirical metric being ill-defined occurs if the program defining the metric has a bug in it. Given that non-trivial programs in general always contain bugs, we perhaps must resign ourselves to the fact that our empirical metrics will generally be ill-defined, though hopefully in ways that are for the most part insignificant. Empirical metrics can also be misleading if they do not match the real-world notions of interest (see the discussion of benchmarking above), but this flaw is likely to be less common than in the case of analytical metrics, because analytical metrics are in general further removed from the real world.

A final problem with analytical metrics is that even when the metric is well-defined and correctly reflects a "real-world" notion, it may be difficult to measure the analytical metric. Consider the analytical metric defined above:

Instantaneous route of a network path The sequence of links and routers comprising the path from A to B at a given instant in time.

The best-known methodology for measuring this metric is the `traceroute` utility [Ja89]. `traceroute`, however, elicits the network route one hop at a time. If, while hop N is being discovered, hop K ($\neq N$, i.e., either upstream or downstream) changes, then this change will be missed, and the sequence of hops reported by `traceroute` will reflect the instantaneous route neither as it was when the measurement began, nor when it ended. Thus, this analytical metric is hard to measure using current tools, limiting its practical utility. (See § 12 for further discussion.) Empirical metrics do not have this problem, since the measurement methodology by definition accurately measures the metric.

7 Empirical metric vs. methodology

The difference between an "empirical metric" and a methodology for that metric is subtle. In this section we briefly discuss the intent behind the two notions, to illustrate the distinction between them.

We imagine the process of assessing some aspect of Internet performance as beginning with an informal notion of "what we would really like to be able to measure." Sometimes this notion has a natural counterpart in the analytic framework used for mathematical analysis of networks. If so, the notion should be expressed as an analytical metric. But sometimes the notion has an inherent complexity that considerably removes it from this analytic framework. Defining it in a form useful for mathematical analysis would then be difficult or clumsy. In this case, it makes sense to express the notion directly as an empirical metric—something that cannot be profitably reduced or simplified beyond "construct the following measurement."

Thus, there will be a one-to-one relationship between an empirical metric and a methodology for the metric. *They are two different names (and concepts) for the same thing.* What we are trying to avoid here is introducing what are essentially useless analytical metrics—those that serve a purpose only on paper, but don't actually help develop measurement methodologies. We would instead like our framework to explicitly point up the inherent complexity of the metric.

8 Composing metrics

Many interesting metrics can be viewed as compositions of simpler metrics. For example:

Transmission time of a link The time required to transmit b bits from X to Y

is often viewed as a combination of:

Propagation time of a link The time difference in seconds between when host X on the link begins sending 1 bit to host Y, and when host Y has received the bit, *and*

Bandwidth of a network link A network link's data-carrying capacity, measured in bits per second, where "data" does not include those bits needed solely for link-layer headers.

If the propagation time is P and the bandwidth is ρ then the transmission time T for b bits is viewed as:

$$T = P + b/\rho.$$

A significant advantage of analytical metrics is the possibility of defining such compositions. With empirical metrics, it is more difficult to know how two separate measurements compose, because the underlying meaning of the measurements is more complex.

With composition comes the possibility of introducing errors. As illustrated in § 6, if the network link of interest uses compression, then the definition of T above is ill-defined. Instead, we have to think of T as a function of both b and of the pattern of those b bits. Note that neither P nor ρ offer an opportunity to introduce the notion of compression: P doesn't because it's defined in terms of a single bit, and ρ doesn't because it's defined without any notion of the pattern of the bits.

Related to the possibility of introducing errors, composition also offers an opportunity for detecting problems in the definition of a metric. For example, the above composition points up the need to include in ρ a notion of the pattern of the bits (for links using network-layer compression). If when defining a new metric one attempts as much as possible to define it as a composition of existing metrics, one can often see problems with the new definition (or perhaps the existing ones), because the composition doesn't quite work. Thus, thinking in terms of composition provides a valuable self-consistency check that we would like to always apply to a suite of metrics to keep it sound.

9 Errors and uncertainties

In general we do not want to discard a measurement methodology because of imperfections, because measurement is almost always imperfect, even in a digital world. Instead, we ask that a methodology *analyze* its sources of measurement error and *quantify* the corresponding effects.

For example, many methodologies include a measurement of elapsed time, necessitating the use of some form of clock. All clocks have imperfections, both in keeping absolute time (synchronization), and in measuring relative time (drift). Furthermore, clocks generally have a granularity below which they cannot measure any passage of time (resolution). The clocks in computers are sometimes poorly kept simply due to operating system deficiencies, or due to hardware shortcomings. Thus for any measurement involving a clock we would like to know how both the clock's resolution and its inaccuracies affect the measurement. This analysis ideally takes a form like: "A clock resolution of $\pm t$ μsec and drift error of $\pm x$ % translates into a measurement error of $f(t, x)$ %," though this may be difficult to achieve.

Another common source of timing error, more difficult to quantify, is that arising from measurement overhead occurring on the computer making the measurement, as opposed to delays due to the network component being measured. The former is a measurement error, while the latter is generally related to the metric of interest. We note that one technique often valuable in reducing host-related measurement errors is the use of a packet filter (ideally running on a separate machine) that records all of the pertinent network traffic with high timing accuracy. The trace produced by the filter can then be analyzed to assess exactly when the network traffic occurred, minimizing the effect of measurement host delays, or at least allowing them to be accounted for.

Finally, just as defining metrics by composition (§ 8) provides an opportunity to debug the metric's definition, it also provides an opportunity for considering how errors and uncertainties compose. Naturally a goal is then to choose a definition that minimizes the propagation and escalation of errors. The general question of how errors propagate during composition has been long studied by numerical analysts in general, and proponents of interval arithmetic in particular—see [Kn81] for an overview.

10 Measurement strategies

In this section we discuss different types of measurement methodologies. We term each type a "measurement strategy" because the different types take different high-level approaches to measurement.

Many methodologies are *active*, meaning that part of the measurement process is to generate new network traffic. This traffic might be to elicit a special response from network components (e.g., `traceroute`), or to see what sort of performance the network provides for the traffic (e.g., `treno`). There are two drawbacks to active methodologies: they add potentially burdensome load to the network, especially if the methodology is not carefully designed to minimize the amount of traffic generated, and they can suffer from "Heisenberg" effects, in which the additional traffic perturbs the network and biases the resulting analysis. For example, if the methodology for measuring bottleneck link bandwidth within an IP cloud is to time huge `ttcp` transfers, then the resulting additional traffic may congest the path through the cloud to the point where packets are dropped, and the measured throughput is appreciably lower than the bottleneck link bandwidth (we have, unfortunately, observed researchers doing this). Along these lines, note that some methodologies are "more" active than others: the `ping` program, while active, only very lightly loads the network, so distortions in `ping` measurements due to Heisenberg effects are likely to be much smaller than those in the above `ttcp` measurement.

An alternative type of methodology is to make *passive* measurements, in which existing network traffic is recorded and analyzed. Because packet filters can capture network traffic without any perturbative effects (if they have local disks for recording the traffic), it is possible using passive techniques to completely eliminate both additional traffic load and Heisenberg effects. These are major advantages, leading us to prefer passive techniques to active ones. On the other hand, for many metrics it is exceedingly difficult to see how one might measure them passively: for example, determining the route taken by packets. The benefits of passive monitoring are in some cases sufficiently large, however, that it behooves us to think carefully before eliminating passive approaches as an option.

For example, if what we care about are not complete Internet routes but merely inter-autonomous system (AS)[1] routes, then we actually *can* measure the routes passively if we are able to monitor traffic between two BGP peers, since over time that traffic contains full inter-AS routing information.

One drawback with passive techniques that is vital to address is ensuring privacy and security [Ce91]. Since most Internet traffic is sent unencrypted, passive measurement programs will often have the potential to capture sensitive traffic. One major step in addressing these concerns is the recently written `tcpdpriv` program [Mi96], which takes packets captured by the popular `tcpdump` program [JLM89] and removes or scrambles a configurable amount of the information present in them, including source and destination hosts and ports, and packet contents. By incorporating `tcpdpriv` into passive measurement methodologies, the privacy and security concerns can largely be satisfactorily addressed.

Another axis of measurement strategy concerns measurement *location*. Some measurements can be done at a single point, without any external cooperation by other network components. For example, to estimate the effective bandwidth of a link utilizing network-layer data compression (§ 8), a program running on a single computer could passively monitor the traffic on a link and perform the necessary computations to form its estimate.

Most measurements, however, and all active measurements, require at least some form of participation by multiple network components. For example, the `ping` program for estimating the round-trip time from host A to B requires that host B respond to ICMP "ECHO request" messages [Po81a, Po81b].

Several forms of such cooperation are already present and widely deployed in the Internet, such as response to certain ICMP requests, generation of "Time exceeded" ICMP er-

---

[1] An autonomous system is a collection of Internet hosts and routers controlled by a single administrative authority that is responsible for internal routing (between any hosts within the AS); often viewed as a single "cloud."

ror messages by Internet routers, and the near ubiquity of "anonymous" FTP servers, which allow throughput measurements between a host A and a remote site S (even if not allowing a throughput measurement to a particular host B at site S).[2] We will term such cooperation as "soft cooperation," indicating that it is easy to come by.

Other forms of cooperation are not widespread, and in general require special arrangements to perform the corresponding measurements. For example, to estimate the route corresponding to the Internet path from A to B requires only soft cooperation, namely that the intervening routers support the generation of "Time exceeded" ICMP error messages, and that the endpoint B supports the generation of "UDP port unreachable" messages. But to determine whether the route is *symmetric*, that is, the same from B to A as from A to B, requires making a routing measurement from A to B and then making one from B to A. The cooperation of host B is "hard": if host B does not provide the means for making the second measurement, and in general it does not without prior agreement with the administrator of B, then the measurement cannot be made.[3]

Methodologies requiring only soft cooperation are naturally to be preferred to those requiring hard cooperation, because the former can be applied on a much wider basis than the latter.

We should note, however, that sometimes a metric can be measured using two different methodologies, one requiring only soft cooperation and one requiring hard cooperation, with the tradeoff being ease of widespread applicability vs. accuracy of measurement. For example, one way to estimate the one-way transmission time T_1 from host A to host B is to measure the round-trip time T from A to B and back to A, using a tool such as `ping`, and then to apply the assumption of symmetric routing and divide the result by two. This technique introduces considerable uncertainty into the measurement of T_1, because of the symmetric routing assumption (and, in reality, routes tend not to be symmetric—see [Pa96a]). Another technique, requiring hard cooperation, is to time the departure of the packet from A and its arrival at B using synchronized clocks, and then using that direct measurement (plus the error due to imperfect synchronization of the clocks) in estimating T_1.

The line between soft and hard cooperation is not fixed but fluid over time, as new network services become more widely deployed. For example, we are presently advocating for the addition to routers of a "fast timestamp" option, in which upon receiving a particular ICMP request, the router generates a high-precision timestamp which it sends in response. If such functionality becomes widely deployed, then a number of throughput measurements presently requiring hard cooperation for high accuracy will become possible with soft cooperation. If the routers were to run with globally synchronized clocks (not so outrageous to contemplate today as in the past, with the growing availability of inexpensive GPS receivers) then the one-way transmission time measurement discussed above could be done using only soft cooperation.

Finally, the drawbacks of hard-cooperation methodologies would be lessened if "measurement platforms," as described in the next section, become deployed. In that case, even though network components at large might not support a particular methodology, if the much smaller set of measurement platforms does then the methodology effectively requires only soft cooperation.

11 Measurement infrastructure

Many of the most interesting measurements are those of the performance of large IP clouds. These are the measurements that potentially have the largest effect on the most people, by providing the necessary information to help network service providers diagnose faults, understand how their capacity is being used, determine how to plan for future growth, and assess their performance versus that of their competitors (as well as allowing third parties to do so).

Another very interesting class of measurements concerns those of traffic over what have historically been referred to as "backbone links": highly aggregated high-speed links used to carry large volumes of Internet traffic over large distances. Such traffic measurements can prove bountiful for answering key network research questions.

For example, the case for "self-similar" network traffic [LTWW94], which has profound implications for networking performance, has been made most solidly for local area networks and for links connecting "stub" networks to the Internet backbone. It has proven harder to make for "backbone links" (where it would have the most serious implications, but also where conditions might be sufficiently different from the other locations that the phenomenon might instead be diminished), because of the great difficulty in attaining measurements from backbone links.

Another example concerns Internet performance in the post-NSFNET world. There is widespread anecdotal evidence that Internet performance in the large has significantly degraded since the National Science Foundation ceased to manage the core of the network (measurement evidence that routing has degraded appears in [Pa96a]), and widespread speculation that the degradation is principally due to the growth of the World Wide Web. Traffic traces from backbone links would shed invaluable light on this issue, which has major implications for Internet engineering.

We envision providing for both IP cloud measurement and backbone link measurements by developing a "measurement infrastructure," something like the following. The infrastructure would consist of a number of cooperating "mea-

---

[2] Though accounting for throughput bottlenecks due to the FTP server itself complicates such measurement.

[3] In principle, this measurement can also be made using "third-party" `traceroute`, which employs IP Loose Source routing. But this routing is often disabled for security reasons, so reliance on it is a form of hard cooperation.

surement platforms." Each platform would be a single host, or perhaps a pair of hosts (to address some of the measurement inaccuracies discussed in § 9), dedicated to performing Internet measurements. Access to the platforms would be controlled by the administrative entity hosting the platform, which would also own the platform and define the usage and privacy policies for making measurements using the platform.[4] This entity might for example be a network service provider, hosting a platform at its border with another provider; or a government entity such as the NSF, as part of a contract to buy service from a provider.

The platforms would be designed to support a wide variety of network measurement (again controlled by the administrative entity). Some of the measurement might be passive, for example support for the network research questions discussed above. Other measurement might be active, both for soft-cooperation methodologies such as `traceroute`'s to arbitrary Internet locations, and hard-cooperation methodologies, usually involving other measurement platforms. Two cooperating platforms on either side of an IP cloud would serve admirably for performing a number of different measurements of the cloud's performance.

A key point concerning the measurement infrastructure is the "N^2" effect: every time a new platform is added to the infrastructure, the number of Internet paths now conducive to sophisticated, hard-cooperation measurement methodologies grows by N, the total number of platforms (or $2N$, since each path can be measured in two different directions). Overall, if the infrastructure is comprised of N platforms then on the order of N^2 different paths can be probed. This sort of scaling property has an attractive "snowball" effect: as more sites add platforms, the utility of the overall infrastructure increases superlinearly, making it considerably more appealing for still more sites to participate.[5]

Most likely such a "snowball" can only come about using both "carrot" and "stick" approaches: network service providers need to both find hosting the measurement platforms to their own benefit, because of the utility of being able to better analyze their networks, and an economic necessity, because customers will demand the ability to better measure the service they are receiving. The ability for customers to compare the service offered by different providers in turn will create the necessary market incentives for the providers to optimize their networks.

See [Pa96a] for a discussion of a measurement experiment carried out using a crude measurement infrastructure approach, which for $N = 34$ was able to probe a significant number of different routes through the Internet.

12 Formalism

We finish with a proposal for a formal structure for defining new metrics and measurement methodologies. As mentioned in the introduction, this paper reflects work-in-progress, so this proposal is both incomplete and subject to change. We present it here as a starting point to encourage discussion.

The definition of a performance metric should include:

- a **name** for the metric
- a discussion of the **underlying notion** that the metric is supposed to capture
- a discussion of **related metrics**
- if the metric is empirically-specified, discussion of why this is the correct choice
- the metric **expressed as a composition** of more basic metrics (some of which may not have yet been formally defined)
- standard **unit** of measurement
- general **measurement issues** (for example, a type of measurement error that most likely no methodology can overcome)
- **cross references** to one or more corresponding methodology definitions, AND/OR
- a discussion of **methodology issues** (measurement assumptions, an example of a measurement approach, an analysis of sources of measurement errors and uncertainties, likely measurement context issues, likely methodology taxonomy)
- **known problems** with the definition
- **future work** (other metrics suggested by this one)

The definition of a methodology should include:

- a **name** for the methodology
- the **corresponding metric**, and a reminder of the measurement units
- the **assumptions** behind the methodology (those aspects of the measured component behavior that, if different, render the methodology inaccurate)
- a discussion of **how the methodology works**
- an **analysis** of how what's measured by the methodology relates to the metric
- an analysis of **measurement errors and uncertainties**, including, in quantitative terms if at all possible, the corresponding inaccuracy of the resulting measurement

---

[4] Clearly, ensuring that the platforms are themselves secure is also a major requirement, to prevent malicious use.

[5] One has to be careful, however, to assure that the snowball does not grow into an avalanche—i.e., that the volume of active measurements between platforms does not unduly burden the network.

- measurement **context issues** (§ 3), including how the measurement location affects the results
- how the methodology fits in the **taxonomy** of passive or active, location/cooperation, and **implications of these** for measurement accuracy and Heisenberg effects
- what **insights** (if any) the methodology sheds on the formal definition of the metric
- any **new metrics or methodologies** suggested
- REQUIRED: a **reference implementation**

We envision performance metric and methodology definitions undergoing the same standardization process as other Internet standards: that is, formal definitions are developed using the “RFC” process [HG94].

Here is an example of a formally defined performance metric:

Name “Instantaneous route of a network path”

Underlying notion Whenever packets are sent from an Internet host A to another host B, each packet traverses on its journey a sequence of Internet links and routers. We term this sequence the *route from A to B*. The particular route affects the time required for the journey; the bandwidth available at the moment from A to B; the congestion levels and available queueing buffer encountered by the packets; and the administrative domains encountered by the packets (with implications for billing, trouble-shooting, privacy, and security).

We are interested in defining the route at the *IP network layer*, so our definition should not allow the notion that the component bits of a single IP packet might encounter different routes—at the network layer, the smallest routable quantity is a full IP packet.

The route from A to B may change over time, on time scales as short as from one packet to the next, so it is important to introduce the notion of an “instantaneous” snapshot of the route. Care must be taken with this notion, however. If “instantaneous” is taken to mean: “view the routing tables at a single instant in time and compute the route taken by a packet from A to B,” then the definition is flawed as follows. It may be that at a single instant in time the route from A to B is comprised of the interface addresses A, R_1, B, but that if a packet were actually sent from A to R_1, by the time it completely reaches R_1, the route may have