Fundamentals of Internet Measurement: A Tutorial

Nevil Brownlee, CAIDA (Cooperative Association for Internet Data Analysis)
Chris Loosley, CMG (Computer Measurement Group)

Many Internet users need to understand how to measure Internet traffic and performance. The primary focus of this tutorial is the global Internet, and ways of measuring, analyzing, and reporting the services provided to a user's network via the Internet. Some sections apply to measuring any network that uses the TCP/IP protocol suite, including a private network, or intranet.

First published in the CMG Journal of Computer Resource Management, Issue 102, Spring 2001

2855 Campus Drive
San Mateo, CA 94403

This tutorial is for readers with a working knowledge of networks and the Internet, and who need to know more about network measurement. Internet users who need to measure Internet traffic and performance include:

- Corporate administrators who buy network service for their organization.
- Smaller service providers who buy network service from a larger provider and resell it to their own customers.
- Anyone buying outsourced Web services who must judge the service provider's ability to serve their end user community adequately.

Many sections of this tutorial apply to measuring any network that uses the TCP/IP protocol suite, including a private network (or intranet). However, the primary focus is the global Internet, and ways of measuring the services provided to a user's network via the Internet. Because end users are typically not on the same network as hosting service providers, an important consideration is the performance of Internet traffic that crosses network boundaries.

A Web version of this tutorial is available as the CAIDA Network Measurement FAQ. It was produced by the CAIDA Metrics Working Group [CAIDA-METRICS].

1. Networking

In this section, we introduce some basic networking terminology. You can usually discover the meaning of a networking term using your favorite search engine. For example, to find 'xxx', try searching for *network xxx definition*. Another approach is to use a Web glossary of networking terms; for examples, see [GLOS-SITES].

1.1. Networks and Internets

A **network** is a collection of **hosts** connected together so that they can exchange information. The hosts may be general-purpose computers (e.g. Mac, Unix or Windows systems) or special-purpose machines such as printers and routers. Hosts in a network communicate using a mutually-agreed network **protocol,** i.e. they exchange **packets** of information following a protocol that defines the packet types, layouts, and sequences.

As well as its physical aspects, every network needs people to look after it. Small networks may have just a **network manager** who is responsible for keeping the network running properly. Larger networks tend to have a team of operations and support staff, organized around a **Network Operations Center (NOC)**.

An **internet** is a collection of networks with links between them so as to allow hosts on any one of the networks to communicate with hosts on any network within that internet. **Routers** are devices with links to more than one network; they forward packets back and forth between the networks. A router may be a purpose-designed unit that simply routes packets and maintains tables of information to support that routing, or it might possibly be a general-purpose computer with multiple network interfaces, running routing software as well as performing other computational tasks.

1.2. Adjacency, Connectivity, Internet and Intranets

The Internet is the internet of networks that use the **TCP/IP** suite of protocols, and are linked together in a globally-connected mesh. [COMER-00] gives a good introduction to the TCP/IP protocol suite.

The Internet can be viewed as a **graph** in the sense that it is a set of nodes (networks containing routers) and edges between them (links between routers). We describe two networks as **adjacent** if there is at least one link directly between them. A network is **reachable** from another network if there is a path between them. Note that **reachability** is one-way; X may be reachable from Y while Y is unreachable from X. Lastly, two networks are **connected** if there is a path (made up of one or more routers and/or links) between them that provides reachability in both directions. For more about graphs and graph theory see [GLOS-GRAPH].

A distinguishing feature of the Internet is that it provides universal **connectivity** to its hosts—every host is connected to every other host and able to communicate with it. It is, of course, possible to run TCP/IP on an isolated network, but such an isolated network would not be part of the Internet. Similarly, individual hosts within a network may be prevented from communicating with Internet hosts by a firewall**;** such hosts do not have Internet connectivity.

A **firewall** is a hardware device or a software program running on a secure host that sits at the junction point or gateway between two networks, usually a private network and a public network such as the Internet, and has connectivity to both. The firewall examines all traffic passing between the two networks, routing only packets that meet defined criteria. Its main purpose is to protect the private network from hostile intrusions originating in the public network that could compromise confidentiality, corrupt data, or interfere with normal service.

An **intranet** is an internet using TCP/IP, but in which all the hosts belong to a single organization, for example a large company with office networks at various geographic locations. Since the hosts on an intranet are only accessible to members of the organization that owns it, an intranet is not part of the Internet. The organization may, of course, have some hosts that are Internet-connected.

1.3. Network Layers

So far we have talked about networks only in physical terms, defining **connectivity** as the ability to send

messages between hosts. Once connectivity is established, it becomes possible for a network to provide **services**; e-mail is one example of such a service.

It is useful to think of network services as being implemented in **layers.** Each layer is considered an entity in itself, providing support for higher layers. This allows each layer to be developed independently of the other layers, which is an effective simplification.

The commonly used layers are listed in Figure 1. The layer numbers were assigned by the Open Systems Interconnection (OSI) Reference Model [OSI-REF], and are widely used as shorthand for the layer names.

1	**Physical**	Provides hardware interfaces between machines
2	**Link**	Sends and receives packets on interfaces
3	**Network**	Carries packets between hosts so as to form networks
4	**Transport**	Provides well-defined information transport for applications
5-7	**Application**	Provides application services to users

Figure 1: Network layers

2. Internet Service Providers (ISPs)

Companies that provide connectivity to the Internet are known as **Internet Service Providers**. Many ISPs provide a complete range of services, e.g. they will

- Connect networks (or individual hosts) to the Internet
- Provide services for their customers such as e-mail, network news and Web access
- Provide support services such as Web hosting

Other ISPs may, however, choose to concentrate on particular services, as discussed below.

2.1. Transport

Transport providers are ISPs who run their own wide-area network and provide Internet connectivity for their customers via that network. ISPs having high-speed networks covering large geographical areas and connecting to many other ISP networks are commonly described as **backbone** providers. A special case of transport provider is a **transit provider,** i.e. one whose customers are other transport providers rather than individuals or companies. The notion of **transit** is discussed in section 2.4 (below).

ISPs who focus on end-user customers can be described as **access providers.** Access providers connect customer networks to the ISP's own network, and thus to the Internet. Customer connections may use various technologies, e.g. dial-in modems (low speed) or fixed connections (higher speeds).

One issue that customers need to be aware of is **over-subscription.** To help define this term, we introduce two others: **proximal**, meaning “very near” and its opposite, **distal,** meaning "far away".

An access provider's network (illustrated in Figure 2) connects to other transport provider networks via fixed links with a well-defined maximum capacity, for example, a T3 link, with 45 Mbps capacity. We'll call this the **proximal capacity**, because it is nearer to the Internet backbone.

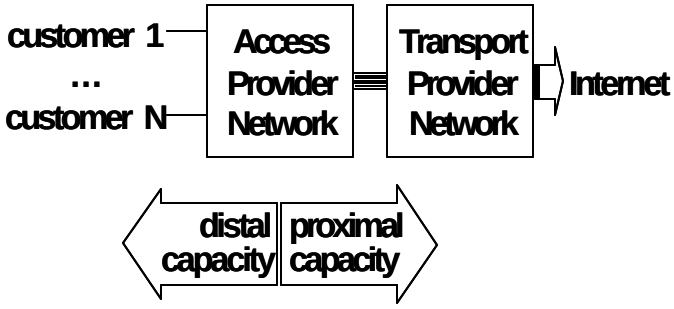

Figure 2: Example of a (small) ISP network

The access provider's network also has N **customer** ports. Each customer may connect using a specified capacity, for example, a 56 kbps modem. The network's **total distal capacity** is N * (average customer capacity).

If the total distal capacity is less or equal to the proximal capacity all will be well, even if every customer attempts to utilize all their specified capacity at the same time. This, of course, doesn't often happen in practice. Because ISPs do not expect all their customers to be online simultaneously, it is normal for an ISP's total distal capacity to exceed its proximal capacity. Customers must share the ISP's proximal capacity, which may degrade their perceived performance during periods of peak usage.

This situation is called **oversubscription**. An access provider's **oversubscription ratio**, R, is computed as:

R = (total distal capacity) / (proximal capacity)

High values of R don't necessarily imply poor performance, for example if most customers are browsing the Web it is unlikely that they will all click their mice at the same instant. Nonetheless, it is reasonable to ask an ISP what oversubscription ratio their network uses. Some ISPs offer cheaper rates if you are prepared to accept a higher ratio.

2.2 Web and Application Hosting

Web hosting providers are ISPs who provide and/or maintain Web pages for their customers. They may provide this service on the ISP's own equipment (hosts, disk farms, etc.) and Web server software, or they may provide facilities to house their customers' equipment in a location where it can be directly connected to the ISP's network.

Application providers or **application service providers (ASPs)** offer what might be called a "time-sharing" service for complex applications. They differ from Web hosting providers, in that their customers do not have to create or maintain an independent web site; the ASP provides and supports web-based applications software that their customers can use. ASPs can provide a useful service for smaller companies who may wish to use complex or expensive software like financial or business applications, but do not want the expense of deploying and supporting those applications.

2.3. Content

The term **content provider**, in its most general usage, refers to any network host that is the source of downloadable content. In this sense, the term is normally used to distinguish companies and organizations whose Web sites are the true target of users' requests for content from those whose role is to *distribute* content on behalf of others.

The need to deliver Web page content promptly to a widely distributed population of Internet users has spawned a new category of service provider, the **content delivery network (CDN)**. CDNs are used by popular commercial Web sites to reduce page download times. For more information, see [IRG-SITES].

A more limited usage of the term **content provider** is to describe organizations whose mission is to *gather and distribute* information. Some (for example, dictionary or encyclopaedia publishers) may indeed generate the information themselves. Others (for example, press cutting and news analysis services) may gather it from many sources, providing indexing and access. This type of content provider usually makes information available to their customers via the Web, and their Web sites may restrict access using usercodes and passwords.

Such a site may also be called a **portal**, these terms are not at all well defined. However if there is a distinction between a "portal" and a "content provider", it is that a portal's primary function is to furnish *links* to content on other sites, as opposed to furnishing the content itself.

2.4. Traffic exchange

Anyone purchasing Internet connectivity assumes that they can exchange traffic with ISPs other than the one to which they are directly connected. This assumption is fundamental to the operation of the Internet. To make this possible, ISPs either pay for transit or have peering arrangements, the terms of which specify how providers agree to cooperate, approximately (the terms are difficult to define) as follows:

Bilateral peering means that two providers exchange routing information, so that each becomes aware of the other's routes to their customers' networks. They may or may not also agree to carry traffic to some or all of those customers; the details are determined by a contract between the two providers.

Transit in this sense is traffic carried by one provider on behalf of another. Such traffic may be carried with or without the exchange of routing information. Again, the details are agreed between providers.

These terms embody very different concepts, and are often lumped under the generic term of **peering.** Understanding that they cover different (though closely related) activities is important when discussing things like service specifications across multiple providers. It should also help to reduce finger pointing when there are performance problems.

For a more thorough discussion of interconnection and peering see [HUSTON-99]

3. Measurement Topics

Lord Kelvin said, "when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind" [KELVIN]. Consider:

- Without measurements, you have no objective record or benchmark of how a network behaves. Measurements show whether changes improve or degrade the network's performance, and by how much.
- If you are buying Internet connectivity from an ISP you need to understand the kind of service being offered. Only by measuring actual performance can you verify that you're getting what you pay for.

Since a network operates by transmitting information packets between connected hosts, at its most fundamental level network measurement involves observing how packets progress through the network. However, not all network measurements require that level of detail, as we explain in the following sections.

3.1. Active vs. Passive Measuring

Passive measurements are carried out by observing normal network traffic, so they do not perturb the network. They are commonly used to measure traffic flows, i.e. counting the number of packets and bytes travelling through routers or links between specified sources and destinations.

Active measurements, on the other hand, are performed by sending test traffic into the network. For example, one might measure a network's maximum carrying capacity by sending packets through it and increasing the sending rate until the network is saturated. Clearly one needs to be aware that active measurements impose extra traffic onto a network and can distort its behavior in the process, thereby affecting measurement results.

You should also consider the effect of your measurements on the network you are trying to measure, and be sure to keep it minimal. Some system administrators may object to your probe packets or test downloads. In such cases you should be prepared to discuss the issues with the persons concerned.

One potential problem with passive measurements is that they rely on traffic flowing across the link being measured. For example, to verify that hosts on your network can download pages from Web server ho-hum.com, you could set up a traffic meter to observe packets to and from ho-hum.com, and produce plots of the ho-hum traffic. This approach can work well provided that users do actually download ho-hum pages often.

If your users only look at ho-hum pages now and then, there may not be enough traffic to allow reliable passive measurements. In that case, you could set up a script to download a chosen page from ho-hum.com at regular intervals—effectively a hybrid of active and passive measurements.

3.2. Measuring at One Point or at Many Points

Another aspect of network measurements concerns the place or places in the network where they are actually made. Some measurements rely on observations at more than one point in the network. For example to measure the time a packet takes to travel from host A to host B, you must record the times when the packet leaves A and arrives at B using accurate, synchronized clocks.

For measuring traffic flows through a large network you might consider observing flows at many points, so as to gather detailed information about the paths packets take through the network. This is not a good idea, since it is difficult to correlate measurements of flows taken simultaneously at even a few different places. Instead it is much simpler to measure traffic at the ingress/egress links of your network, avoiding the complexities of having to follow individual packets on their various paths through the network, while still allowing you to produce a traffic matrix showing overall traffic flows through it.

Because a host performing active measurements need only concern itself with its own (measurement) traffic, active measurements can be performed using only modest hardware, e.g. a Unix system running on a PC. A host performing passive measurements, on the other hand, must cope with all the traffic passing the measurement point, a task which gets harder and harder as traffic rates increase. Nonetheless, passive measurements can be performed very effectively at network ingress/egress points.

3.3. Network vs. Application Measurements

There is considerable interest in measuring the performance of networked applications, for instance, we suggested above that you might make active measurements of the ho-hum.com Web server by making periodic downloads of its pages.

In principle you could set up some cron jobs that run scripts to download a set of Web pages and measure the time it takes. In practice this can be a non-trivial task, and commercial monitoring packages and services are available. (See 7.5 Application Monitoring and 7.6 Visualization).

Application-level measurements are needed for a clear view of overall application performance, which cannot easily be synthesized from lower level data. They may

also offer some insights into the performance of the client and server hosts, and of the network links between. However, although Web downloads may be thought of as a network service, measuring them gives only an indirect view of underlying network behavior.

In situations where you want to compare performance of a particular service, say over consecutive days or weeks, it may be sufficient to measure the overall performance and assume that variations caused by server loading are small compared to variations caused by network congestion. This approach is certainly useful in cases where you are trying to make comparisions, e.g. between different transport providers.

Another advantage of application measurements is that some ISPs today use **traffic filtering** techniques within their networks, for example by blocking ICMP echo packets or by limiting the rate at which they are processed. Measurements using such packets (e.g. using ping, see below) are still useful, but the increasing use of traffic filtering is decreasing that utility.

3.3.1. The IPPM Framework

To clarify the differences between application and network metrics, the IETF's IP Performance Metrics (IPPM) Working Group has developed a Measurement Framework for measuring network performance, [IPPM-FRAM].

The Framework presents terms for describing networks, explains the need for metrics to be useful, concrete, well defined, and capable of being measured repeatedly and reliably. With the Framework defined, the IPPM Working Group has continued to specify network metrics such as those for one-way delay [IPPM-OWD] and connectivity [IPPM-CONN].

These metrics involve observing well-specified test packets sent through the network. They are designed to measure network performance directly, and do not depend on the presence of any application software.

3.4. Sampling Techniques and Traps

When observing packets on a network, one's goal is to use measuring tools that can keep up with the traffic at the measurement point, without missing any packets for any reason; this task gets harder and harder as the traffic rate increases.

If the rate is too high for all packets to be observed reliably, the measurement tool should at least report the number of packets that were missed. In this situation there may be no alternative but to sample the packets, i.e. to base the measurement on a specified subset of the packets—in other words, to **sample** the network traffic.

What sampling algorithm should one use? The simplest approach is to examine every nth packet, and this usually works well. Again, one could examine packets at fixed time intervals; this is harder to implement and may be affected by aliasing (synchronization) effects. Synchronizing effects, and ways to avoid them, are discussed in section 11.1 of the IPPM Framework [IPPM-FRAM].

Another point of view is that of signal processing, summed up in the Sampling Theorem, which says that

"a band-limited signal which has no frequency components higher than F Hz is completely described by sampling values of the signal at intervals of 1/2F seconds."

To paraphrase, to measure effects that happen at frequencies of F Hz or less, you must sample the signal at least 2F times per second.

Many feel that sampling granularity should be finer at higher network speeds. Ultimately, how the data is to be used determines the appropriate sampling rate; for example, billing may require a finer granularity than capacity planning. One paper covering sampling strategies is [CPB-93]. Although this paper is a bit dated, its discussion of sampling techniques is still relevant.

4. The Most Common Metrics

The metrics in this section are the ones most commonly used when assessing network performance. Unfortunately there is little agreement on exactly how they are defined; this section presents the Working Group's view of the metrics and the way they are used in practice.

4.1. Latency

In general terms, latency is a time delay while one waits for something to happen. For many kinds of network communications, once a packet (or group of packets) has been sent from one computer to another, nothing appears to happen until an answering packet is received in return. So a widely used measure of **network latency** is **round-trip time (RTT)**, the time for a packet to make the round trip from a client to a server and back.

Many component times contribute to network latency:

a. The time it takes a packet to travel along the physical links that make up its path through the Internet (transport time)
b. The time it takes to pass through routers between those links (queuing and transmission time)
c. The time required for the server to process an incoming packet and generate a response packet (server response time)

Typically we cannot measure latency for each hop on a network route; the goal should be to measure the major components of latency separately. For example:

- **Forward delay**, the sum of component times (a) and (b) for packets travelling from client to server
- **Server delay**, component time (c)
- **Reverse delay**, the sum of component times (a) and (b) for packets travelling from server to client, often not the same as forward delay.

Measuring forward and reverse delays requires measurement at both client and server, using **one-way delay** techniques such as those as described in [IPPM-OWD].

In practice, ISPs, at least the ones who publish service level agreements on the Web, e.g. [ISP-SERV], usually just add components (a), (b) and (c) together to produce a single latency or **backbone delay** value. Typical latency values range from hundreds of microseconds on a LAN to 90 ms or more on a link from New York to London.

To measure latency one should use a method that is implemented within the server's IP stack, so that it requires very little server processing to generate a response. Ping (see below) is most commonly used for this purpose. ISPs (the ones mentioned in the previous paragraph) commonly use ping to measure latency, and assume that the sever delay (c) is small compared to the forward and reverse delays (forward and reverse a+b). The resulting **latency** measure is thus a somewhat coarse indication of Internet performance, rather than a true measure of network delay. It is nonetheless a very widely-used performance metric.

Unfortunately, network latency is not fixed, it changes as conditions on the network vary. Reasons for this include

- If the server is lightly loaded it will respond quickly. Busy servers will be slower to respond, which will increase the server response time.
- If there is no congestion on the packets' path, queuing time will be minimal. If, however, the path is congested, packets will be queued within routers, increasing the queuing time.
- The paths a packet takes through the Internet and back may change due to routing fluctuations, which are often caused by link failures that cause routing protocols to recalculate paths for packets.

Route changes can produce changes in the transport time. For example the **best path** selected by the routers may take packets through more routers than before, increasing the time spent in routers and in hops between them.

To detect variations in latency it is common to produce daily plots showing average latency for short intervals, e.g. one to five minutes. Such plots often show diurnal variations, with latency increasing during the busy (congested) periods of the day. They may also show sudden increases or decreases—these can be caused by routing changes (as discussed above), by security attacks, or may simply be bursts of traffic.

ISPs usually specify the latency one can expect in their network, for examples, see [ISP-SERV]. A typical **service latency** specification goes something like this:

> *Our network latency is the average round-trip time for packets sent between any pair of our backbone routers. It will not exceed 85 ms for pairs of routers located anywhere in Europe.*

Note that

- An ISP can manage latency only for packets travelling within its own network.
- Since only average latency is specified, the maximum observed latency can be higher than the specified value. You need to ask how often the latency is measured, by how much it may vary, and over what period it is averaged.
- The ISP (see [ISP-REPT]) will measure latency using a tool such as ping, which uses the ICMP protocol. This may not accurately represent the performance of applications that use other protocols.

Streaming applications (such as Voice over IP and videoconferencing) work best when there is little variation in their transmission time delay. Although such applications can tolerate occasional packet losses, variation in delays can cause noticeable degradation in the user-perceived quality of their service. There is considerable interest in metrics for delay variation, but this work is still in its early stages.

4.2. Packet Loss

By default, Internet packet transport works on a **best-effort** basis. Routers make every reasonable effort to forward packets, but may drop them depending on the router's immediate local conditions. Best-effort forwarding is an important design feature of the IP protocol and any Internet service that needs reliable packet transport must detect packet loss and resend lost packets. TCP does this for high-level applications such as Web browsing.

Packets travelling through the Internet may be delayed by being queued in routers. If its queues become full, a router may be forced to discard packets because it has no space in which to hold them. Such packets are described as **lost.** Other network faults can also cause lost or corrupted packets, but these are much less common.

Network **packet loss** is the fraction of packets lost in transit from client to server and back during a specified

time interval, expressed as a percentage of the packets sent to the server during that interval. Packet loss rates vary from 0% (an uncongested path) through 5 to 15% (severe congestion). Higher loss rates will most likely make the network unusable for normal purposes.

A moderate level of packet loss (say a few percent) is not in itself an indication of network failure, since many services continue to operate effectively in the face of some packet loss. For example:

- Some real-time or streaming services, e.g. Voice over IP, can tolerate some packet loss, but once a packet is lost there's no point in trying to recover it.
- TCP resends lost packets, and uses packet loss as a signal that it should send data at a slower rate. This behavior is described as **network-friendly.**

The second point is particularly important. Since TCP relies on detecting lost packets to sense congestion and control the rate at which packets are sent, we expect to see occasional packets lost from TCP streams.

As mentioned above, routers discard packets when they run out of queue space to hold them. A simple way to do this is wait until the queue fills and then drop packets as they arrive, which favors streams with lots of packets already enqueued. A better algorithm is to discard packets at random from the queue before its space is exhausted, thus maintaining space for new incoming packets. This is called **Random Early Detection (RED)**.

4.3. Throughput

Throughput is the rate at which data is sent through the network, usually expressed in bits per second (bps), bytes per second (Bps) or packets per second (pps). Throughput most commonly refers to the total data transfer rate for all traffic being carried, but it can be useful to measure throughput at finer granularity, e.g. for Web transactions, for Voice over IP, to specified destinations, etc.

Throughput is measured by counting bytes transported during a specified time interval. Be careful in choosing the interval; a long interval will average out short-term bursts in the data rate. Short intervals imply a higher data collection rate, and may exaggerate the burstiness of the data. A good compromise is to use one- to five-minute intervals, and to produce daily or weekly plots.

4.4. Link Utilization

Internet service is normally provided to a corporate site via one or more physical links, each of which has a maximum data rate, known as the **access rate** of the link. **Link Utilization** over a specified interval is simply the throughput for the link expressed as a percentage of the access rate.

Some links, e.g. serial links such as T1 and T3, have a well-defined physical maximum speed (1.544 Mbps for T1). Other links, e.g. Frame Relay PVCs, have a second rate, the **Committed Information Rate (CIR).** A CIR is the data rate you are paying for, but your traffic is allowed to run for short periods at higher rates. That is, the link will carry short bursts without loss. Utilization should be calculated as a fraction of the CIR for these links.

5. Availability

A network manager's goal is to keep the network running properly all the time, which is difficult because various parts of the system inevitably fail, potentially with follow-on effects that may escalate the effect on the system. To reduce the likelihood of system-wide failure we should design our networks to be fault-tolerant systems.

"A Conceptual Framework for Systems Fault Tolerance" [FAULT-FRAM] is a good introduction to this topic, especially its section 3.2, "Faults and Failures." The essential point here is that a well-designed system can continue to run effectively even though some of its parts are faulty.

Nevertheless, even the best managed network can be subject to **outages** from time to time. Hence the need to define, measure, and report on **availability**.

5.1. Definitions and Tests

Many service descriptions do not provide an explicit definition of availability. But the implicit usage is similar that of ITU-T E.800 [E-800], namely:

> … the ability of an *item* to be in a state to perform a *required function* at a *given instant of time* or at any *instant of time within a given time interval* …

From the user point of view, **availability** over a specified time interval is the percentage of that interval during which the system was available for normal use. Of course, one must decide just what it is that is supposed to be **available.** For example, consider

- **Service availability:** being able to send packets for a specified service—say WWW request packets—to a particular Internet host, and to receive answering packets
- **Host availability:** being able to send packets, say ping packets, to a particular Internet host, and to receive answering packets
- **Network availability:** being able to send packets from your network to the Internet, and to receive answering packets

In each of these cases, one can test availability by sending suitable packets and observing the answering packets (or lack of them). For example

- **Web service availability test:** download specified pages from target Web server using Web browser, measure latency, packet loss and throughput.
- **Host availability test:** ping to the target host, having made sure that it will respond to ICMP packets. (ping is described below).
- **Network availability test:** traceroute to the target host, so as to determine whether there is connectivity to the target network. (traceroute is described below).

Measurements like this will produce latency and packet loss values for each case. For each case, one has to decide what values for maximum latency and minimum packet loss are required for effective service. Should the measured values fall outside these limits, the service will be considered **unavailable.**

For example, again from the user point of view, you could decide that network availability is lost if, when you ping your ISP's access router at one-minute intervals, ping latency is worse than 10 ms or ping packet loss is greater than 1 percent. You need to be realistic when setting these limits, for example 2% packet loss may be unacceptable for Web or ftp data transfer, but 5% packet loss could be acceptable for email.

This approach to defining and testing availability is modeled on the techniques for measuring **connectivity** discussed in [IPPM-CONN]. For the three cases above, connectivity means the ability for our chosen type of request packet(s) to travel from the client to the server, have a response generated by the server, and for the response packet(s) to return from the server to the client.

5.3. Reporting on Availability

Availabilities are usually reported as a single monthly figure giving the percentage of the time that the network was available. One minute is approximately 0.01% of a week, hence if service availability was 99.99%, the service was unavailable for about four minutes during the month.

When you discuss availability with your ISP, bear the above in mind as you work together to agree on a definition of availability. Consider how you will each monitor the availability of your Internet service, since a difference in approach will make it hard for you and your ISP to agree about the measurements or their meaning.

In practice, ISPs (again the ones who publish service level agreements on the Web, e.g. [ISP-SERV]) do not normally specify availability so carefully. Instead they use a single value of **backbone availability,** without specifying exactly what it means. For a user, it most likely means site reachability, i.e. (ii) above, to hosts within the ISPs network. ISPs often publish Web pages showing backbone availability, either a single worst-case figure, or (better) a matrix showing availability between significant interconnection points-of-presence (POPs) in the ISP's network [ISP-REPT].

Network faults are the most common cause for a loss of backbone availability, e.g. a fiber may be cut by a backhoe, or some piece of network equipment may fail. Unavailable time should start from the moment that the provider or customer detects network unavailability. Be aware, however, that some providers consider the network to be available until the user reports a network failure!

Another cause of unavailability is scheduled downtime, i.e. time specified by the provider for maintenance or upgrades. A service definition should make it clear whether or not scheduled downtime is considered **unavailable** time.

Other important quantities related to availability include

- **Mean Time To Repair (MTTR):** The time (in minutes or hours) taken to restore normal service after a loss of availability
- **Mean Time Between Failures (MTBF):** The average time (in hours or days) between the beginning of normal service and the next loss of availability

Both MTTR and MTBF are important metrics that should be a part of your availability formula. A service that fails once every 2 years but has an MTTR of a week does not seem particularly desirable. On the other hand, a service with a smaller MTBF and extremely small MTTR may have good availability numbers, but the presence of frequent short-lived transient errors may be irritating and generate many trouble reports.

5.4. Reliability

Traditional measures of quality also include **reliability**, which is closely related to availability, but not the same. It is a measure of how often you get a response back that is wrong, or get a part that is defective. A “wrong” response in a network would be a corrupted packet, which is not the same as getting no response—that would indicate a loss of availability. Note, however, that when measuring network behavior it is common to count wrong response packets as lost.

Network transport protocols provide checks on the correctness of the transferred data; if corruption is detected (by transport-layer software), the packets concerned are retransmitted, so that the user sees only a slower overall transfer rate. Such a lowered rate reduces the quality of the service, possibly to the point where that service should be considered unavailable.

6. Working with Statistics

Once you understand what network metrics exist and how to measure them, you must decide how to present the results. Since you will inevitably use some kind of statistical technique—however simple—to summarize your data, this section provides some background information on choosing and using statistics.

6.1. Summary Statistics: Mean, Median, Percentiles

Metrics such as latency are strictly valid only for specified periods, e.g. the five-minute-period latencies shown on a daily plot. Performance summaries (e.g. those published on network providers' Web pages) are often given as a single value, such as the average five-minute latency for a whole month.

Care is required in choosing a suitable statistic for this purpose. In particular, one should avoid using statistics that make assumptions about the process that generated the data being summarized. That is, one should use **non-parametric** statistics such as percentiles.

The **arithmetic mean** (or **average**) is one possibility, but before using averages one should be confident that the observed data is well-behaved, and doesn't have too many **outlier** values. A few outliers can greatly affect the arithmetic mean.

An alternative to the simple average is the **geometric mean**, which is considerably less affected by outliers. Technically, it is the Nth root of the product of a set of N data points. It is computed by averaging the logarithms of the data points, and computing the exponent of the result (reversing the log).

For most purposes, the **median** (50th percentile) is a good choice, since it is not much affected by a few outlier values. One can also use **percentile ranges** to indicate the spread of the observed values, e.g. 5% and 95%, 10% and 90% or 25% and 75%. For a further discussion of percentiles, see the IPPM Framework [IPPM-FRAM].

6.2. "Mononumerosis"

The name "mononumerosis" was coined for the Metrics Working Group by Cindy Bickerstaff to describe an undue focus on a single measured value. This is a common mistake, but for a better understanding we must usually look at a set of metrics, because tradeoffs exist between different metrics.

One example is delay and loss. You can reduce delay in a network by shortening queues in routers, but one result can be increased loss. Do you really want a lower delay at the cost of increased packet loss? In some cases, you might tolerate higher delays in order to reduce packet loss to suit some applications. At the application level, slightly longer routing delays could actually lead to faster average response times, because of fewer retransmission delays caused by lost packets.

Another example is performance vs. availability. Would you want faster Web page downloads at the cost of higher failure rates, and sometimes getting back no Web pages?

To summarize, metrics have tradeoffs, and you should look at the whole picture rather than focusing on just one dimension. How you evaluate the tradeoffs may depend on what kinds of traffic will be using the network.

6.3. Sampling and Averaging Games

Some metrics can only be measured as an average value over a specified measurement interval. For example, to measure bandwidth utilization you count the number of bytes within each interval, and compute the interval's average utilization. The period over which a metric is averaged can make a big difference to its distribution. The choice is important when an ISP computes charges based on some measured percentile, for example on the 95th percentile of bandwidth utilization.

Averaging over shorter intervals makes for higher values, averaging over longer intervals results in lower values. Thus if service provider A charges less than service provider B for 95 percentile bandwidth utilization, but service provider A takes 60 second averages and service provider B takes 1 hour averages, service provide A may justifiably cost more.

7. Common Measurement Tools

A tremendous variety of tools are available for measuring many aspects of network performance. Tools range from simple commands included in common operating systems, through free (open-source) applications, to commercial packages and systems. In this section we briefly survey what's available, and how you can use it to measure and monitor your network. We focus on measuring the **most common metrics** discussed above, using freely available tools.

7.1. ping

ping is a simple application that runs on a (client) host, normally supplied as part of the host's Operating System. *ping* sends an ICMP echo request packet to a specified (server) host, the server sends back an ICMP echo reply packet, and the *ping* program displays the time taken for the round trip. *ping* can be run on different platforms under different Operating Systems; the server host doesn't have to run any special software for this,

because ICMP packets are handled within its Operating System kernel.

The ICMP echo request/reply packets are often loosely referred to as **ping packets.** Most ping programs allow the user to send a single packet, or a series of them at specified intervals. If you ask ping to send more than one packet it will do that, and display summary statistics upon termination.

Ping's statistics include the packet loss percentage, as well as information about the round-trip times (latency). For example, on a Unix host, ping will report:

- target name
- count of packets transmitted
- count of packets received
- percentage of packet loss
- min/avg/max/stddev of round-trip times

Ping provides a simple test of reachability for the server site, at least for ICMP packets. However, being able to ping a host does not necessarily mean that you can access other services on it (e.g. www or telnet), since those services may be blocked by a firewall at the edge of the target host's network. Similarly, the lack of a ping response may only indicate that ping packets are being blocked by a firewall.

You may be tempted to select a router as a target host—especially one that is outside a firewall—since its ping server should always be running. Routers, however, usually run their ping server at a low priority, so ping will produce high round-trip times.

When measuring site availability, you must select a host at the target site (i.e. within the target site's network) which will reply to ping packets. You can then run ping at regular intervals so as to measure the site's latency and packet loss percentage; from these you can determine the times at which the site was available.

For further details of ping consult your system's documentation (e.g. its ping man page), or see [STEVENS-94].

7.2. traceroute

traceroute is probably the most commonly used diagnostic tool for determining why a target host fails to respond to ping. Like ping, traceroute is normally supplied as part of a host's Operating System, and it doesn't require any special software installed on other hosts. traceroute produces a hop-by-hop listing for each router along the path to the target host; for each hop it prints the round-trip time (latency) for the router, or an * if there was no response.

Note that traceroute shows you only the forward path, i.e. the one from the source to the target host. The return path is seldom the same, which is why forward and reverse delays differ. To trace the reverse path one must run traceroute on the remote host, with one's own host as its target. One way to do this is to use a **reverse traceroute** server. The easiest way to find reverse traceroute servers is via [REV-TRACE].

Interpreting traceroute's output accurately requires some experience, nonetheless it is a useful network diagnostic tool. You should occasionally run traceroute for hosts that are important to you, and keep the output for later comparison. [STEVENS-94] provides more details of traceroute.

To work properly, traceroute relies on receiving ICMP response packets from routers along the path being traced. If those packets are blocked by a firewall, traceroute will show the routers as **not responding.**

7.3. Network Management, SNMP

The **Simple Network Management Protocol (SNMP)** is the Internet standard system for managing network-attached devices, and today devices such as hubs, switches and routers come with built-in SNMP agents. An SNMP agent maintains a database of information specified in a MIB (Management Information Database) for the device. Hosts on the network can find out about the device by reading data from objects in its MIB, and (if authorized) can change the device's configuration by writing new values to some of those objects.

Systems that do gather SNMP data comprehensively are types of Network Management Systems (NMS). These allow a network manager to monitor network information from many devices, display it on screen and report problems, e.g. loss of connectivity, as they arise. One does not, however, need a full-blown NMS for simple everyday monitoring of a single Internet connection—SNMP APIs are available for popular programming languages such as perl, making it simple to write programs or scripts to read and display SNMP object values at regular intervals.

Given such a program, you need to decide which MIB objects to monitor. Many MIBs are available, both standard and proprietary but we will focus here on the simplest, the Internet Standard MIB-II [SNMP-MIB2], implemented on nearly every new network device. MIB-II has an Interface table, with one entry for every interface on the device. Among its variables the interface entry has counters for

- **ifInOctets**, the total number of octets received on the interface, including framing characters.
- **IfOutOctets**, the total number of octets transmitted out of the interface, including framing characters.

SNMP counters are never reset, instead you should read them at regular intervals and subtract the values. This allows you to monitor the data rates in and out of the interface.

7.4. Traffic Flows

Monitoring total traffic on links (through interfaces) using SNMP is straightforward, but what if you need a more detailed breakdown of that traffic? For that you need a flow measurement system.

One possibility is the IETF's RTFM traffic measurement architecture [RTFM-ARC]. To use this you need to

- Set up traffic meters at each point where you want to monitor traffic flows
- Create configuration files to specify which flows you're interested in
- Run a **manager** program to download meter configurations, and read flow data from the meters at specified intervals.
- Set up scripts, cron jobs, etc. to analyze and archive the flow data.

NeTraMet [NETRAMET] is an open-source implementation of RTFM. The NeTraMet meter is a program that runs on Unix systems; it is most commonly used to monitor 100 Mbps Ethernet links, but it is also used on FDDI and OC3 links.

If you have a device with multiple high-speed links to other networks, you would like to collect flow data from its interfaces directly, rather than having to install passive taps on each link. If you are using Cisco routers or switches, you can use Cisco's NetFlow data export [NETFLOW] to do this. For every interface running NetFlow, the device builds a table of information about flows. When a flow ceases, or after a specified interval, the device sends the flow information as **NetFlow data records** to a specified device for processing.

Several open-source systems are available for collecting and analyzing NetFlow data. One widely used system is cflowd, [CFLOWD] which is a purpose-designed, distributed data collection and analysis system for use with NetFlow-enabled devices. NeTraMet includes a meter that reads NetFlow records, allowing NeTraMet to be used to collect and analyze NetFlow data.

7.5. Application Monitoring

Most application monitoring tools are commercial software and/or services, and tend to be highly sophisticated. At its simplest, however, service monitoring can be achieved by writing scripts and cron jobs that periodically run an application, measure its response time, and log the results. One of the few open-source examples is ***timeit*** [TIMEIT], which monitors Web server performance by getting pages from a predefined list of Web servers.

7.6. Visualization

Having measurement tools is only the beginning of a successful network monitoring effort. You have to use those tools to gather and archive data, and to produce easily available, easily understandable reports that network support staff can actually use. To help with this you need good visualization tools.

The most widely-used visualization tool is **MRTG** [MRTG], which reads SNMP variables such as the traffic counters (inOctets, outOctets above), logs the data, and produces Web pages from it. As well as daily plots, MRTG produces weekly, monthly and yearly plots.

RRDtool [RRD-TOOL] is a generalized system developed from MRTG, for storing and displaying time-series data. A steadily growing list of **RRDtool front-ends** are appearing [RRD-FRONT]. These gather data from various sources and pass it to RRDtool for processing and display

7.7. CAIDA tools

One source of information about measurement tools is the CAIDA tools taxonomy [CAIDA-TOOLS]. The Measurement section lists tools sorted by type of measurement, together with references to the tools and their developers. The Visualization section lists tools for displaying the data, an essential step towards understanding your network. Among the tools are:

cflowd [CFLOWD] and **flowscan** [FLOWSCAN], for processing router NetFlow data (described under “Traffic Flows”). FlowScan is a network analysis and reporting tool that processes flow files generated by cflowd.

coralreef, for analyzing TCP/IP flows for optical network media (OC3-OC48).

netramet, an open-source (GPL) implementation of the RTFM architecture for network traffic flow measurement, developed and supported by Nevil Brownlee at the University of Auckland.

arts++, a binary file format specification for storing network data.

netgeo, for mapping IP addresses, domain names, and AS numbers to geographical locations.

RRDtool, for storing and displaying time-series data (such as network bandwidth, or server load average).

GeoPlot, for creating geographical images of data sets.

GTrace, a graphical front-end to traceroute that geographically depicts IP path information between source and destination hosts.

Mapnet, for macroscopic Internet visualization and measurement.

Otter, for visualizing arbitrary network data that can be expressed as a set of nodes, links, or paths.

8. Comparing Service Offerings

You may need to verify that the network is performing within its service specification. This involves long-term monitoring and reviewing of your network's performance. You will also need to know what limits your ISP specifies for latency, packet loss, etc. so that you can pinpoint the times when service was unacceptable. Occasionally, you may also need to compare the services of various ISPs. This section suggests ways to make such a comparison.

8.1. Provider Net Performance Pages

If you have a specific list of ISPs to compare, you should probably start by discovering exactly what kind of service each ISP offers, by contacting the ISP and asking them directly. As a first step, however, it is worth looking at the ISP's Web page to see whether their services are described there. A list of Web pages for ISP service descriptions is given in [ISP-SERV].

In publishing service descriptions, ISPs will usually specify maximum latency and packet loss, together with minimum availability. As discussed in **Common Metrics** above, you need to read such descriptions very carefully so as to be sure how the ISP defines the metrics. You also need to know how the ISP measures them, and whether the ISP makes those measurements available to you (perhaps on a Web page).

Some ISPs maintain Web pages showing the performance of their network; [ISP-REPT] lists some of these pages. The pages vary considerably as to how much information they contain, from the bare minimum (latency figures for the last n months) through to matrices showing metric values for routers in many of the ISPs main locations. AT&T's network status page is a particularly good example. Not only does it have matrices for **backbone delay** (latency) and **backbone loss** (packet loss), it also has a good **state of the network** page and a background **methodology** paper explaining how the measurements are made.

Most ISPs also publish information about network outages, together with contact information for reporting problems; some of these are also listed in [ISP-REPT]. To find more such pages, try searching for **xxx network status,** where xxx is the ISP of interest.

8.2. Internet Weather Maps

Internet **weather maps** are Web pages which attempt to provide an impression of Internet performance in various parts of the world. Some of these are included in [CAIDA-OTHER].

Andover News Network's Internet Traffic Report uses measurement stations at various locations around the world. These ping many routers along "major paths" on the Internet; the ping data is used to compute a **traffic index,** i.e. a number from 0 (slow) to 100 (fast) which indicates performance relative to that in the past week. As well as seeing the average traffic index for the five continents, one may also view the ping data for individual routers.

Another example is the MIDS Internet Weather Report presents animated scans of macroscopic conditions across the Internet. It displays geographical maps that show round-trip (ping) delays from MIDS offices in Austin, Texas to thousands of Internet hosts worldwide.

8.3. Rating Services

A growing number of commercial suppliers provide systems for measuring performance and comparing that of different service providers. A few of these are listed in [RATE-SERV]. As well as tools, these companies provide consulting services to help users improve the performance of their networks, as viewed from the Internet at large.

One of these companies—Matrix Information and Directory Services (MIDS)—provides a set of Web pages comparing the performance of ten large ISPs in terms of the ISPs' latency, packet loss and reachability, as observed from six MIDS **beacon** sites. These comparisons are given as tables, and plots of the metrics may also be viewed.

9. Acknowledgements

The following members of the CAIDA Metrics Working Group [CAIDA-METRICS] made significant contributions to this material: Carter Bullard, Cindy Bickerstaff, Les Cotrell, Jambi Ganbar, Gelnn Grotefeld, Daniel McRobb, Sue Moon, Jeff Sedayao, Marty Schulman, Dave O'Leary, Tanja Szeby, Henk Uijterwaal, Brett Watson.

The Working Group Co-ordinator, Nevil Brownlee, edited the online FAQ; Chris Loosley edited this version.

10. References

To improve readability, the protocol designation "http://" has been omitted from all URL's in this section.

[CAIDA-METRICS] www.caida.org/outreach/metricswg/

[CAIDA-OTHER] www.caida.org/analysis/performance/

[CAIDA-TOOLS] www.caida.org/tools/taxonomy/

[COMER-00] Comer, Douglas, "Internetworking with TCP/IP Vol. I: Principles, Protocols, and Architecture (4th edition)," Prentice Hall, 2000

[CFLOWD] www.caida.org/tools/measurement/cflowd/

[CPB-93] "Application of Sampling Methodologies to Network Traffic Characterization," Proc. SIGCOMM '93, pp 194-203, San Francisco, September 1993. Available from moat.nlanr.net/Papers/sigcomm.sampling.ps

[E-800] "ITU-T Recommendation E.800 (08/94)–Terms and definitions related to quality of service and network performance including dependability."

[FAULT-FRAM] "A Conceptual Framework for Systems Fault Tolerance," hissa.ncsl.nist.gov/chissa/SEI_Framework/framework_1.html

[FLOWSCAN] ee-staff.ethz.ch/~oetiker/Webtools/rrdtool/frontends/flowscan.html

[GLOS-GRAPH] www.utm.edu/departments/math/graph/glossary.html

[GLOS-SITES] Glossaries of networking terms:

www.mids.org/glossary
www.Webopedia.com
whatis.com

[HUSTON-99] Huston, G., "Interconnection, Peering, and Settlements," www.isoc.org/inet99/proceedings/1e/1e_1.htm

[IPPM-CONN] Mahdavi, J. and Paxson, V., "IPPM Metrics for Measuring Connectivity," RFC 2678, September 1999

[IPPM-FRAM] Paxson, V., Almes, G., Mahdavi, J. and Mathis, M., "Framework for IP Performance Metrics", RFC 2330, May 1998

[IPPM-OWD] Almes, G., Kalidindi, S. and Zekauskas, M., "A One-way Delay Metric for IPPM," RFC 2679, September 1999

[IRG-SITES]These Internet Research Group sites have papers and links to vendors in the areas of Internet caching, content distribution, and Internet traffic management.

www.caching.com
www.cddcenter.com
www.itmcenter.com

[ISP-REPT] Network Performance Report Web pages:

stats.sjc.above.net/traffic/
ipnetwork.bgtmo.ip.att.net/
traffic.cwusa.com/index.html
www.eli.net/sla-stats/index.shtml
www.jet.net/status/statushelp.html
www.sysci.org/status/statushelp.html
www.uu.net/network/latency/
www.wcom.com/terms/service_level_guarantee/t_sla_latency.phtml

Network Operations Web pages:

www.dx.net/network.html
www.earthlink.net/assistance/status/status.html
status.flash.net/

www.genuity.com/help/noc/nocintro.htm
help.mindspring.com/netstatus/
www.one.net/network/announce/
www.pictureview.com/support/PVTS2.html
www.psinet.com/netstatus/
www.sprintlink.net/netstat.html
www.noc.uu.net/CritOut.html

[KELVIN]
This statement is widely quoted, see for example
www.gla.ac.uk/publications/avenue/29/legacy4.html
www.faraday.gla.ac.uk/app_areas.htm

[ISP-SERV] Service Definition Web pages:

www.att.com/globalnetwork/mdns_agreements2.html
www.psinet.com/sla/
www.us.uu.net/support/sla/
www.wcom.com/terms/service_level_guarantee/

[MRTG]
ee-staff.ethz.ch/~oetiker/Webtools/mrtg/mrtg.html

[NETRAMET] www.auckland.ac.nz/net/NeTraMet

[NETFLOW]
"NetFlow Services and Applications" (an introduction and overview)
www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm

[OSI-REF]
www.cisco.com/cpress/cc/td/cpress/fund/ith/ith01gb.htm
www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/osi_prot.htm

[RATE-SERV] Commercial performance-rating services:

www.mids.org/
www.keynote.com/product/prod_main.html
www.visualnetworks.com/

[REV-TRACE]
www.caida.org/analysis/routing/reversetrace/

[RRD-FRONT]
ee-staff.ethz.ch/~oetiker/Webtools/rrdtool/frontends/index.html

[RRD-TOOL] www.caida.org/tools/utilities/rrdtool/

[RTFM-ARC]
Brownlee, N., Mills, C. and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, October 1999

[SNMP-MIB2]
McCloghrie, K. and Rose, M., "Management Information Base for Network Management of TCP/IP-based internets: MIB-II," RFC 1213, March 1991

[STEVENS-94] Stevens, W.Richard, "TCP/IP Illustrated, Volume 1," Addison-Wesley, 1994

[TIMEIT] IETF IPPM Working Group minutes, December 96, www2.ietf.org/proceedings/96dec/ops/ippm-bickerstaff/sld001.htm Distribution file available from ftp://ftp.va.pubnix.com/pub/uunet/timeit-2.1.tar.gz