Internet Delay Measurements using Test Traffic

Design Note

Henk Uijterwaal[*]
Olaf Kolkman[†]

RIPE NCC
June 2, 1997

Document: RIPE-158.ps

Abstract

This document discusses the test traffic project proposed as a new activity in RIPE-144 [1]. The document gives an overview of the project, discusses the implementation and mentions a couple of points that have to be addressed in future design documents. This document is intended to solicit input from interested parties.

1 Introduction

This document describes the design and implementation of the test traffic project. This project has been proposed as a new RIPE-NCC activity in [1]. The project is similar to the project proposed in [2] for a different community. The goal of the project is to do independent measurements of connectivity parameters, such as delays and routing-vectors, in the Internet.

The general idea behind this project is shown in figure 1. The local network of each Internet Service Provider (ISP) can be divided into two parts: an internal part, that deals with the traffic between the machines of the customers in this network, and an external part, that deals with the traffic between this and other service providers. This project focusses on the external networks only.

Two of the parameters that determine the network performance between two providers are: the total bandwidth and the delay. The bandwidth is the

*Email: henk@ripe.net
†Email: okolkman@ripe.net

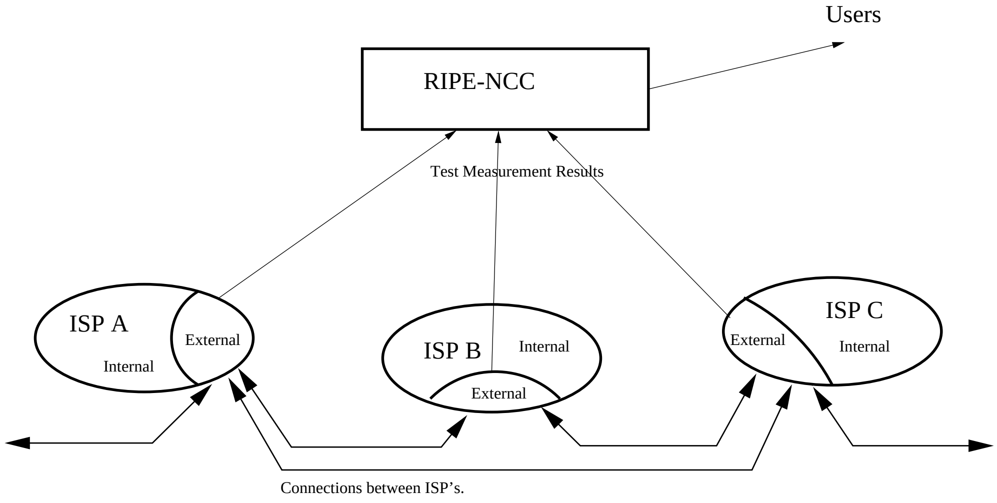

Figure 1: Overview.

number of bits that can be transferred between two providers per unit of time. The delay is defined as the time elapsed between the moment that a packet leaves the network of one provider and the moment that the packets arrives at its destination. Associated with these parameters is a routing-vector, this vector describes how traffic travels from this provider to another.

In order to measure the delays and determine the routing, measurement-boxes will be installed at each participating provider. These boxes collect data. The data is then transferred to a central machine at the RIPE-NCC. Here the data is processed and made available to the users of the networks.

The outline of the remainder of this note is as follows: Section 2 discusses the measurements that we plan to use in order to determine the network performance. Section 3 gives an overview of the hardware and software infrastructure needed for these measurements. Section 4 shows how the data can be presented to both experts and casual users of the Internet. An important point is to prove that our measurements actually reflect the performance of the network. Section 5 discusses how we can verify our results. Finally, section 6 discusses the implementation, milestones and deliverables of this project.

2 Measurements

There are (at least) two ways to monitor the characteristics of the Internet:

- Passive: by monitoring the amount of traffic that passes a certain point and,
- Active: by generating test traffic and measuring how much time it takes to ship the test traffic over the network.

The advantage of an active measurement over a passive measurement, is that for the active method, the test traffic can be generated under well-defined and

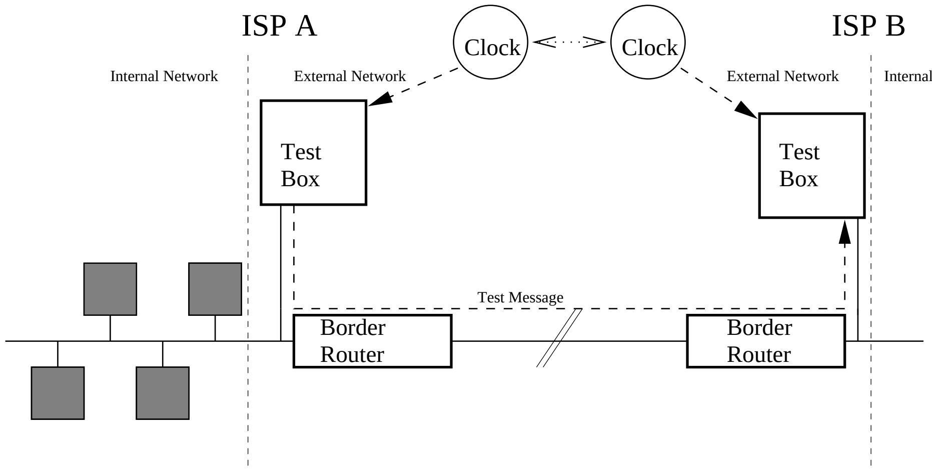

Figure 2: Experimental Setup

controlled conditions, whereas the passive measurement depends on the traffic that happens to pass a certain point. Another important advantage of active measurements is that it avoids the privacy concerns associated with passive measurements. As we want to deliver an objective measurements of the performance of the Internet, we have decided to focus on active measurements only.

The principle of our measurements is shown in figure 2. A test-box is connected 0 hops away from (or, if that is not feasible, as close as possible to) the border router of each participating ISP. By connecting the test-box 0 hops away from the border router, we exclude effects of the internal network from our measurements.

If the ISP has more than 1 border router, we will either connect the test-box in such a way that the delays to each border router are similar or we will install multiple test-boxes. This way, we avoid a bias for traffic in certain directions in our measurements.

The box at ISP-A generates a pre-defined pattern of test messages. The test messages are send through the border router and the network to a similar box at ISP-B. Both test-boxes are connected to a clock. By looking at these clocks when a test message is sent and when it arrives, one can determine the delay between these two providers. There are other measurements that one can do with this setup, this will be discussed in section 2.1.

The test-boxes at both ISP's are identical, thus the process can be reversed to measure the delay between ISP-B and ISP-A. Also, the boxes at both ISP's can be used to send test traffic to similar boxes at other ISP's.

It should be obvious from figure 2 that the clocks at the two ISP's should be *synchronized* (in other words, the time offset between the two clocks should be known and remain constant) and provide the time with a high accuracy compared to the time difference that we want to measure. This will be discussed in more detail below.

In the extreme case where all $\mathcal{O}(1000)$ ISP's in the RIPE-NCC (geographical) area participate in the project, the test-boxes can be receiving test traffic from and sending it to 1000 other boxes, thus testing all 1000^2 possible connections. However, the topology of the Internet is such that it is likely that one can still do reliable measurements of the performance of the Internet without having to send test traffic for all 1000000 possible connections. This will be studied.

Although this method can be used to measure the performance of the internal network of each provider, we want to limit our work to the external networks only. For this reason, the test-boxes should be located as close as possible to the border routers of the ISP's, in order to eliminate any effects caused by the internal networks.

2.1 Data Collection

There are several possibilities for the test traffic that we can generate:

- **One-way:** A packet is sent from ISP-A to ISP-B, as described in the previous section.
- **Two-way:** A packet is sent from ISP-A to ISP-B and then returned to the sender. This principle is used in "ping" and other tools that determine the round trip time.

  However, assuming that the paths are known, the results obtained with one two-way traffic measurement, will simply be the sum of two the one-way traffic measurements. For this reason, we plan to use two-way traffic only for independent checks of the one-way results.
- **Real life applications:** A measurement that a user sees in a real application, like fetching a WWW-page or downloading a file by FTP, using a well defined TCP benchmark connection.

In the first instance (phase 1, section 2.1.1), we plan to do one-way and two-way measurements. At a later stage (phase 2, section 2.1.2), this will be expanded to performance measurements for real life applications. Our measurements will, already in phase 1, produce several observables, this will be discussed in section 2.1.3.

2.1.1 Phase 1

The one way test traffic will consist of UDP data packets of 3 different sizes: small (56 bytes, as used by ping and similar tools), medium (576 bytes, the minimum packet size that any router must handle as a single packet) and large (say, $\gtrsim$ 2048 bytes, to see the effect of (possible) splitting data-packets into smaller units and related effects).

The packets will contain:

- The address of the sender.

- A time-stamp that shows when the packet was sent, together with the dispersion of the clock in the sending machine. The dispersion is needed to determine the overall error in the measurement.

- A reference number, in order to keep track of the number of packets send and lost.

- A hop-count (number of routers that the packet passed between sender and receiver).

- Administrative information, such as the version number of the software and a checksum.

- Padding to give the packet the desired size.

The receiving process will remove the padding and then add the following information to the packet:

- The address of the receiver.

- A time-stamp that shows when the packet was received, together with the dispersion of the clock in the receiving machine.

The result is raw-data that contains all information about this particular delay measurement.

The path or routing-vector between ISP-A and ISP-B is defined as the collection of machines and network between the border routers of these two ISP's. This path may vary as a function of time. A tool like "traceroute" will be used to determine the path for the test traffic at any given time. The path information will be made available.

There are several potential problems while doing these measurements:

- **Set-up effects:** even for Internet traffic, the routers have to be setup before a connection between two points is established. If this is not taken into account, then the measured delay will be larger than the delay that can be attributed to the network. Two possible solutions to circumvent this problem are:

  - Precede any measurement by a "traceroute". This determines the path and provides the routing information that we are interested in anyway.

  - Run the test more than once and compare the results. If the first result is significantly different from the other measurements, discard it.

  Set-up effects are interesting in themselves and this data will be recorded and studied.

- **No connectivity.** The fact that there is, contrary to what one expects, no connectivity between two points is interesting information in itself. However, our software should be written such that it will survive this case without operator interference.

For the two-way measurements, we plan to use data-packets that are similar to the one-way measurements. These results will be used for consistency checks only.

2.1.2 Phase 2

In the second phase of the project, we plan to do measurements with:

- **TCP-streams.**
- **Simulations of applications.**
- **Packet Trains.** A packet train is a number of test messages that are sent with very short intervals. They provide a way to study if the packets are delivered out of order and/or merged together into larger packets along the way.

The implementation of this will be discussed in a future design note.

2.1.3 Observables

These measurements will provide several observables:

- **Delay Information:** The results of the delay measurements can be stored in a $n \times n$ matrix $\mathbf{D}(t, s)$ where each element $\mathbf{D}_{sd}(t, s)$ represents the time that a packet needed to travel from a source s to a destination d. If there is no connection between two points, then $\mathbf{D}_{sd}(t, s)$ will be ∞.

  The elements of $\mathbf{D}$ are a function of the time t, as the network characteristics will change over time, and the packet size s.

  The elements $\mathbf{D}_{sd}$ all have an error $\Delta\mathbf{D}_{sd}(t, s)$ which gives the total error in this delay measurement.

- **Routing Vector or Path Information:** Each delay measurement is accompanied by a determination of the path between the two locations using a tool like "traceroute". These results can be written as a $n \times n$-matrix $\mathbf{P}(t)$ of vectors.

  This matrix provides, assuming that our test-boxes are installed at a significant fraction of the ISP's, an up-to-data map of the Internet as well as the history of of the network.

  If there is no connectivity between two points i and j, then the element P_{ij} will be 0. The number of zeroes therefore gives a measure of the total connectivity. Also, the two elements P_{ij} and P_{ji} should either both be zero (no connection between these two points) or non-zero. If only one of them is zero, this indicates a network configuration error.

Finally, the elements P_{ij} can be used to detect routing loops.

It should be noted that on any particular test-box, only the results of delay measurements *to* that box and the path information *from* that box are available (in other words: the *columns* of the matrix $\mathbf{D}$ and the *rows* of the matrix $\mathbf{P}$): the results of a delay measurement between ISP-A and ISP-B will be collected at ISP-B, whereas the traceroute information is only available at ISP-A. It is undesirable to do the traceroute at ISP-B, as the path from A to B is not guaranteed to be the same as the path from B to A.

In order to get the full matrices, the results have to transferred to a central point. This is one of the reasons for collecting all test results on a single machine.

From these observables we can determine if there are trends in the transfer times as a function of time and isolate special or suspicious events. During the initial phase of the project we will concentrate on developing (statistical) tools to analyze the matrices. In a later phase we, or other researchers, may develop tools to correlate the data in the matrices, analyze the effect of the size of the packets or perform analysis of the routing vector matrix itself.

Once enough data has been collected we must be able to define meaningful metrics and we might tune the measurements to optimally sample this metric.

2.2 Frequency of the test traffic

There are 5 basic requirements:

1. The test traffic should be small compared to the load on the connection under consideration. If not, then the test traffic will affect the performance and the measurement becomes useless.

2. The intervals should be small enough to study (interesting) fluctuations in the performance of the network.

3. As the network changes over time, the amount and type of test traffic should be easily configurable.

4. The interval between two measurements should be small enough to detect and eliminate set-up effects.

5. As suggested by the IPPM [2] the measurements should be randomly distributed to prevent synchronization of events due to weak coupling as demonstrated in [3]. The IPPM suggest using a so-called Poisson sampling rate. We will investigate the IPPM suggestion and other possibilities to prevent weak coupling.

The first two requirements are contradictory: smaller time intervals means more test traffic, but more test traffic means a higher load on the network.

For the initial settings, suppose that we generate the following amount of test traffic:

- 1 small packet per minute.
- 1 medium sized packet per minute.
- 1 large packet every 10 minutes. These packets are used to see the effects of splitting large packets into smaller ones, presumably this effect is constant over time.

This then generates a data-volume of approximately 14 bytes/s for each measurements. This number does not include overheads and the data for the "traceroute" program.

If this amount of data is too high, then one might consider increasing the interval between two packets and add a burst mode (a short time with a higher test traffic volume) for occasional measurements with a smaller interval between two test packets.

This leads to a number of questions that have to be answered

1. Can we generate this amount of data without the ISP's objecting? Probably not for 1 connection, but what if we have installed test-boxes at 10, 100 or even 1000 sites, with 10^2, 100^2 or 1000^2 possible connections?
2. Are we sure that the test traffic does not affect the performance of the network?
3. How do we check that a packet has left the test-box before the next one is sent out? We do not want packets to sit in a buffer if the local network is at its maximum load.

2.3 Error Analysis, Required Clock Accuracy

In this section, we want to estimate the errors on the final results. As we mentioned before, our plan is to do both one and two way measurements.

One way measurements involve sending a time-stamped message from one machine to another. The second machine compares its arrival time with its local clock and determines the transfer delays from that. This requires that the local clocks are synchronized and provide the time with a high enough accuracy. The synchronization is the difference between the clock on this machine and an absolute time standard. The accuracy is the error in the time measurement. The errors in synchronization and accuracy determine the overall error in the delay measurement.

Two way measurements are a bit simpler in this respect: a message is sent to another machine and echoed there. All time measurements can be done on the same machine, so the clock has to be accurate and the the resolution of the clock has to be high enough. The resolution is the smallest time interval that can be measured by this clock.

In both cases, we will be subtracting two times, t_1 and t_2. If we call the total error on these times Δt_i, then the error in the final result (Δt) is equal to:

$$\Delta t = \sqrt{\Delta t_1^2 + \Delta t_2^2} \tag{1}$$

It should be noted that, even with an high resolution clock, Δt can become relatively large for small time intervals. For example, if we measure a 50 ms time interval with a clock with a resolution of 10 ms, then the error in the result will be 14 ms.

Systematic effects of the equipment on the results have to be studied and should be understood.

3 Experimental Setup

3.1 General Idea

The general setup is shown in figures 1 and 2.

The central machine at the RIPE-NCC controls the test-boxes at the ISP's. The data is collected by the test-boxes and then transferred to the central machine.

It is planned that the data collected at the local machines will be kept there for a few days, so that it can be viewed and analyzed locally. The main processing of the data will be done on the central machine.

Of the order of 1000 ISP's are active in the geographical area serviced by the RIPE-NCC. Although we certainly do not foresee that all 1000 ISP's will participate in the project or that test traffic will be sent for all 1000^2 possible connections, we do plan to design the software such that these numbers can be handled.

3.2 Data Flow Diagram

Figure 3 shows a first version of the Data Flow Diagram (DFD) for this project. This diagram will be the starting point for the software design for the project.

3.3 The Central Machine

The central machine (or machines) performs a number of tasks, including:

- It controls the configuration, which determines which connections are being tested,
- It collects the data from all test-boxes,
- It acts as the software repository,
- It will be the platform where the software is being developed,
- Finally, this machine will be used for data-analysis.

The main problem for the central machine is the amount of data to be stored.

The results of each delay measurement will consist of the packet sent by one test-box to another, plus information added by the receiving test-box such as the arrival time. If we assume that this results in (at most) 100 bytes of data

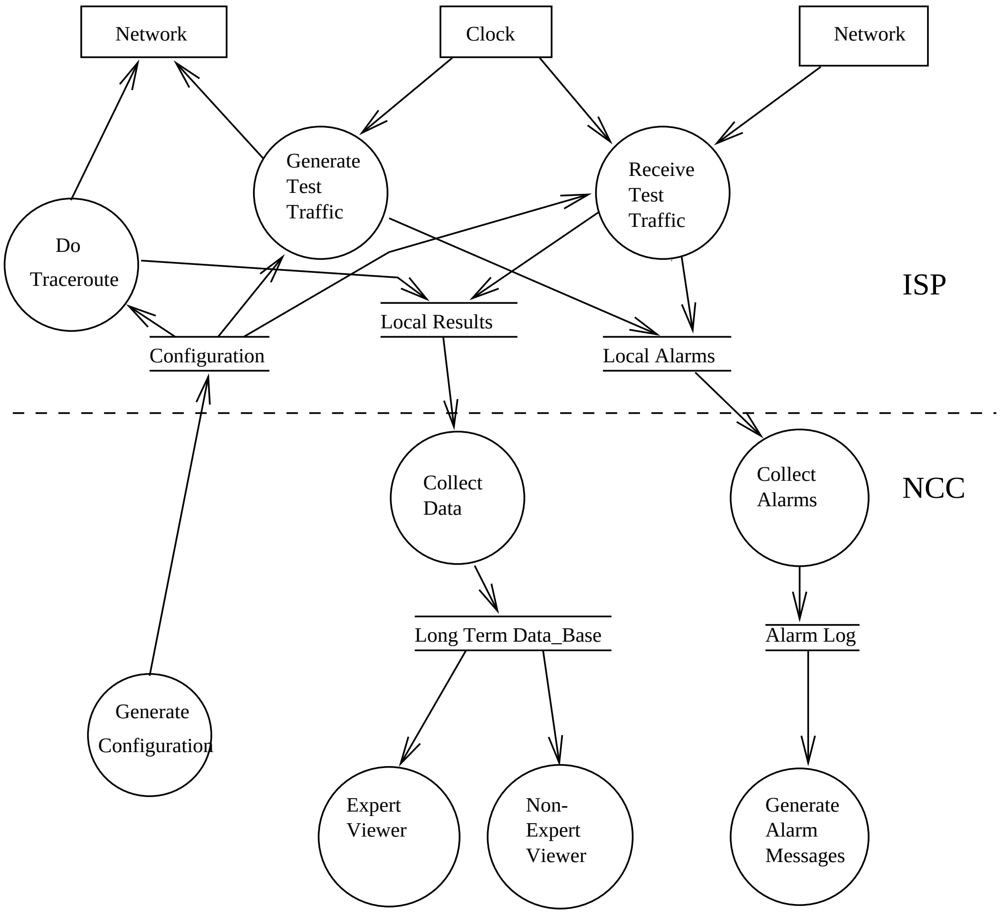

Figure 3: The Data Flow Diagram for this project.

for each delay measurement, then measuring at a rate of once a minute will produce 150 kbyte/day or 55 Mbyte/year of data for each connection.

In addition to that, one also has to store the routing-vector information. For a stable network, the amount of data for this will be less, as one can store a routing-vector with a validity range, rather than the routing vector for each measurement.

So, allowing for short intervals with high rates and a significant amount of routing-vector information, a safe upper limit of the amount of data to be stored is 250 kbyte/day or 90 Mbyte/year of data per connection.

This means that in a setup where $\mathcal{O}(100)$ connections are being tested, one needs several Gbytes/year of disk space to store the raw data. If the number of connections that is being tested goes up by an order of magnitude, one will need a tape-unit or similar mass storage device.

In the extreme case, where 1000 providers participate in this project and we test all 1000^2 possible connections between them, the data volume will be of the order of several Tbytes/year. While this volume is not unmanageable, it will require a tape robot for storage, a major investment, or a significant compression of the raw data before it is stored.

A database program will be used to store the data. This database should be able to handle data volumes up to several Tbytes, distributed over several physical volumes, and read the data in a reasonable amount of time. As the data might be used for several different analysis, it should be possible to generate Data Summary Tapes (DST's), with a subset of the data.

For presenting the data, we need a graphical analysis tool. This tool should be able to plot and histogram data, and, of course, have an interface to the database.

3.4 The test-boxes at the ISP's

3.4.1 Hardware Setup

The test-boxes consist of an industrial PC, which can be mounted in a 19 inch rack. Inside the box, there will be a CPU, a disk and a high precision clock. The disk should be large enough to store several days worth of data, assuming the 250 kbyte/day/connection, 1 Gbyte should be sufficient. The box should be hooked up to the border router of the ISP.

The machine will be "plug and play", in order to make installation easy and to avoid tampering with the machine by a local ISP. After the box has been installed, all maintenance will be done remotely from the NCC. Only in the case of major hardware failures that cannot be solved remotely, we will need some support from the local ISP's.

The design of the box will be such that the ISP cannot affect the performance of the computer or reconfigure his network such that the results will be affected ("Design the network for the benchmark"). This will include installing machines at the customers of this ISP and do consistency checks.

3.4.2 Software Setup

The software on the machine will read a configuration file that specifies which measurements it has to do, perform the measurements and collect the data.

The machine load should be small, to avoid that the measurements are affected by processes competing for CPU-time. This will also put a limit on the number of connections that can be simultaneously tested.

As the number of machines will be large, the software should be easy to maintain:

- One script should restart all processes after a reboot or power failure.
- The software should survive common errors and restart itself without operator interference.
- An automated procedure, like rdist, will be used keep the software up to date.
- As the machines will be located in the heart of the networks of the ISP's, they should be hacker-proof. In order to accomplish this, the number of

accounts on the machines will be limited to the absolute minimum, all software will run in secure shells and no un-encrypted passwords will be sent over the net. Finally, all the software running on the machines will be made available to the participating ISP's, so that they can convince themselves that the test-boxes do not introduce any security holes.

All the software written for this project as well as the design of the test-boxes will be made available to the participating ISP's. They can use it for measurements on the internal part of their networks.

3.5 The clock

The clock at the remote machines is the most critical part of the whole project. Before we start to work on any other part of the project, we should prove that a clock with sufficient accuracy can be built.

We aim for a accuracy that is at least 1 order of magnitude better than the smallest delays that will be measured by the box, including drifts. In a typical network environment, the delays will be of the order of 10 ms, this then translates to a required accuracy of 1 ms or less. The overall error in a 10 ms delay measurement will then be of the order of 1.4 ms.

We are considering 3 solutions for this problem:

3.5.1 NTP/Network Time Protocol

This is a software protocol [4] that uses timing servers all over the world. With atomic clocks as references and suitable hardware, accuracies of down to a few hundred ps can be obtained. With off-the-shelf products, the accuracy will be less.

The current list of servers [5] lists primary servers in: France, Germany, Holland, Italy, Norway, Switzerland, Sweden and the UK, and secondary servers in: Portugal, Poland and Slovenia.

This approach uses free software, so the costs for this solution are small, assuming that the internal PC clock in a standard environment is stable enough for our purposes.

A potential problem is that there are large areas without a nearby server. Also, the accuracy might be affected by unstable networks. A test-setup will be used to determine if this solution can provide a stable clock under these circumstances. In these cases, an external reference clock has to be used to obtain the necessary resolution.

3.5.2 Radio Time Servers

This is a system of long-wave radio stations broadcasting the current time. A receiver can collect these timing signals and provide the local time with an accuracy of the order of a few ms. Typical receivers cost of the order of $ 2500.- in 1992 [6]. There are, however, several problems with this approach:

First of all, timing signals are only available in the UK, Germany and surrounding areas. This means that this approach can only provide a clock signal for a small part of the global area covered by the RIPE-NCC. For the remaining part, one needs another approach.

Then, reference [6] mentions seasonal effects and other corrections which are needed in order to get the accuracy of a few ms. This implies that the clocks need constant attention in order to keep the high accuracy.

Finally, the costs for each clock is much higher then for a solution that uses GPS receivers (discussed below).

For all these reasons, we will not consider this solution any further.

3.5.3 GPS Receivers

This approach uses the Global Position System (GPS), a satellite navigation system developed by the US Department of Defence (DoD). Receivers collect a timing and position signal from up to 24 satellites. With the system, the time can be obtained with accuracies down to 200 μs.

There are several receivers on the market that can be mounted as an extension card inside a PC. These cards have to be connected to an antenna and can be read out by the PC. The NTP package [4] can then be used to synchronize the internal clock to the global time. The GPS receivers will therefor provide an easy to maintain and reliable clock.

A test showed that an antenna mounted in the window of our offices was able to pick up the signals from several satellites, which is sufficient to provide a timing signal. This antenna has to be located, depending on the type of card and antenna, within 10 to 100 m of the receiver. This puts some constraints on the local infrastructure at the ISP's.

Cards typically cost of the order of $1000 [7].

A side-effect of this solution is that we create a network of high-precision clocks (so-called stratum-1 NTP servers) through the area where our test-boxes are located. These clocks can be used for other purposes. Also, the GPS receiver will provide its global position with an accuracy of about 100 m.

3.5.4 Conclusion on clocks

We believe that GPS receivers combined with the NTP software will provide a suitable clock signal for all test-boxes. However, this is the most critical problem in the project and we should focus on this first.

3.6 Costs

The funding for the prototypes and initial deployment of the test-boxes is discussed in [1]. Large scale deployment and maintenance may require additional funding which will be sought separately.

4 Presentation of the data

We realize that the presentation of the measurement results is a delicate matter. We plan to organize this in close coordination with the ISP's concerned and the relevant RIPE working groups. The schemes presented below represent a first idea which certainly needs refinement as the project progresses.

4.1 Disclosure of the data

We foresee 3 different levels of disclosure of the data:

1. **Experts:** The experts at the NCC will, of course, have access to all data that is collected with our test-boxes.
2. **Participating ISP's:** Each ISP that installs a test-box will get access to all data related to his network (but not to data related to somebody else's network).
3. **Non-participating ISP's:** They will only have access to global quantities derived from the data.
4. **The general public:** The general public will have access to global quantities derived from the data, but with a more detailed explanation describing how the data should be interpreted.

During the initial phases of the project, the data will be distributed under a non-disclosure agreement. Both the experts at the RIPE-NCC and the participating ISP's can use the raw data for any analysis that they find interesting but both sides agree not to publish any results until the results have been verified (see section 5) and both sides agree that that they are meaningful, correct and can be published. The details of this agreement will be worked out with the participating ISP's.

4.2 Access to the data

There will be several ways in which the data can be accessed:

1. **Test-box:** The test-box will have a mechanism that will return the results of the last set of measurements from this box to the ISP where the box is located.
2. **Server:** All information about an ISP will be available for that ISP only on the central computer for the project at the RIPE-NCC.
3. **Reports:** Finally, we will use automated tools to extract summary plots from the data. These plots will be made available on a regular basis.

   Two different kinds of reports may be made available: Reports with data relevant to a single ISP and reports with data relevant to all ISP's. The plots in the first category will be made to that particular ISP only, whereas the other plots will be made available to the entire Internet community. In the latter case, the plots will be presented in such a way that it is not possible to trace the data back to a particular ISP.

The format in which the data will be presented will be decided in mutual agreement with the participating ISP's.

4.3 Reports

4.3.1 Step 1

In the initial phase, we foresee the following plots:

- Delays between two test-boxes.
- Length of the routing-vector. This information can be cross-checked against the number of hops that the delay measurement packets saw between sender and receiver.
- Packet loss (# messages received/# messages sent). From the reference numbers in the packets, the receiving machine can determine which packets actually arrived and estimate the packet loss along the way.
- Percentage of the connections working. A connection between two points is considered broken if less than a pre-defined fraction of the test traffic messages sent over that connection actually arrives. From that, we can determine the percentage of the connections in the entire sample that is actually working.
- Time of arrival of the last message from host X. This provides another way to determine if a connection is working and to calculate the fraction of connections that are working.
- Number of changes in the routing vector.
- More plots will be defined as the project progresses.

All these quantities will be plotted as a function of time of the day. In addition, the path information will be made available in some sort of data-base.

4.3.2 Step 2

As soon as the individual plots are understood, we will start generating summary plots. These summary plots will contain information about 1 particular ISP (for example, the average delay to or from this ISP) or even more global quantities (like the average delay on the Internet).

As soon as we are confident that exceptional behavior (for example, the delay between two points suddenly doubles) is an indication of a network problem rather than a problem with our test-boxes, we can start to generate warning messages to flag these conditions. There are several ways to detect these conditions, including quantities exceeding a certain threshold, χ^2 test, Kolgomorov tests against known usage patterns, and so on.

4.3.3 The far future

There is much more information that can be extracted from this data-set. For example:

- Delays inside one country or geographical region, or from one region to another.
- Correlations, if the traffic between points A and B is slow, is this correlated with heavy traffic from A to C, or C to B or even C to D?
- Trend analysis, how do the delays develop as a function of time?
- Optimization of the routing in the network.
- Relation between the delays and the geographical distance between two test-boxes.

These and other subjects will be studied.

5 Verifying the results

5.1 Local Test Bed

Our plan is to first build a test bed consisting of two test-boxes in a well known setup at the NCC and a third test-box at a remote ISP. This test bed will be used for:

- Software development.
- Software performance (timings, latencies, machine load, etc).
- Tests of the clock cards and their performance, in particular:
  - Development of the interface for GPS clocks.
  - Measurements of the accuracy, drift and resolutions that can be obtained with both GPS and NTP clocks as well as tests of hardware modifications and filter algorithms to reduce drift and increase the accuracy of the clocks.
  - Measurements of the differences between two identical clocks.
  - Tests of the effects of the environment (temperature!) on the clock.
  - Testing possible locations of the antenna and their effect.
- Tests of the hardware stability.
- Tests of the system maintenance procedures.
- External checks: for example by connecting a scope to two clocks and comparing the clock-pulses with an independent, reliable, source.

In short, with this setup we should convince ourselves that the hardware does what we expect it to do.

5.2 Internal consistency

We will check the internal consistency of the data by comparing one-way and two-way measurements. For example, the sum of the one-way delays in traffic sent from A to B and B to A should, assuming that the paths are the same, be equal to the delay measured with a two-way measurement.

Also, a delay measured for traffic from A to C via B, should be about the same as the sum of the delays for traffic sent from A to B and B to C.

There are more tests like this possible, they should give us a handle on the in-situ performance of the test-boxes.

5.3 Error Analysis

The treatment of statistical errors is well-known and should not present any problems.

In order to estimate the systematic errors, we will do tests where the clocks are artificially set of by x ms and see if we can detect this from the results.

6 Implementation

6.1 Tools

We are in the process of deciding which tools will be used to implement this project.

- **Clock:** we are currently investigating which cards are available on the market.
- **Test-boxes:** our plan is to use off-the-shelf industrial PC's running BSD Unix.
- **Plot package:** Two packages are being considered:
  - The CERN-library tools (which includes a DST-mechanism) and
  - PGPerl, a PERL extension to the PGPlot package.
- **Database:** For the first prototypes, we plan to use a home-grown data-base. At a later stage, a commercial product will be considered.

6.2 Time Schedule

The basic approach is that we want to start with a test bed (see section 5). This setup will be built during the summer of 1997. Data taking will start in the late summer.

If the test bed is a success, we will try to find a handful (10. . . 30) ISP's who are interested in having a test-box installed at their sites. These boxes will be used to test of the order of 100 connections, where the number of 100 is set by the data volume that can be handled without major investments in storage

	Date		
*	Apr 14	‘97	Start of project.
	Apr	‘97	Write design note (this document).
	May	‘97	Write design note.
			Order 1 or 2 clock cards for evaluation.
			Design and order the hardware for 2 prototype test-boxes.
	May 16	‘97	Draft design note should be finished.
*	May 21	‘97	RIPE-meeting, presentation of the project, get feedback from the RIPE customers.
	May 31	‘97	Produce final version of this design note including feedback from the RIPE-meeting.
	Jun	‘97	Start on software design. Write note with implementation details.
			Test clock cards.
			Build and test prototype test-boxes.
	Jun 30	‘97	First version of the software design should be ready.
	Jul	‘97	Start implementing the software.
	Aug	‘97	Build test setup with 3 prototype test-boxes.
			Do first measurements.
			Analyze results.
	Aug 11	‘97	IETF-meeting, consider presenting something there.
*	Sep	‘97	RIPE-funding meeting. Present project, including some data collected in the test setup.
	Fall	‘97	Continue with software development.
			Design final version of the test-boxes.
			Analyze results (phase 1 measurements).
			Present results at the RIPE meeting, find a some ISP's (say, $\mathcal{O}(25)$), that are interested in hosting a test-box.
*	Dec	‘97	First version of the system ready to be used in a medium sized environment.
			Build and install test-boxes.
	Jan	‘98	Start taking data with the test-boxes.
			Analyze and verify results.

Table 1: The time schedule for this project. The milestones are marked with a $*$.

devices. This setup will start running early 1998. Several ISP's, both from the RIPE community and from those participating in the TERENA TF-ETM working group [8], have shown interest in hosting our test-boxes.

After this setup has proven to be a success, we will develop a plan to include more ISP's and connections.

A more detailed plan of deadlines and milestones can be found in table 1.

7 Conclusions

In this document, we gave an overview of the test traffic project, discussed the implementation and mentioned a number of questions that have to be studied. The project was presented at the RIPE-27 meeting and the feedback from the RIPE-community has been added to this design document.

We will now start with the design of the software and the prototype test-boxes (see section 6). Future notes will discuss the implementation details of the project. If the development of the prototype test-boxes goes according to plan, we expect to be able to present the first results of the project in the late summer of 1997.

References

[1] D. Karrenberg *et al.*, "RIPE NCC Activities & Expenditure 1997", RIPE-144.

[2] G. Almes *et al.*, Framework for IP Provider Metrics, draft-ietf-bmwg-ippm-framework (work in progress), and related documents.

[3] S. Floyd and V. Jacobsen, "The synchronization of Periodic Routing messages", IEEE/ACM Transactions on Networking, Vol. 2, No 2, April 1994.

[4] See `http://www.eecis.udel.edu/~ntp/`, and references therein.

[5] See `http://www.eecis.udel.edu/~mills/ntp/servers.html`.

[6] D. L. Mills, "Network Time Protocol (Version 3), Specification, Implementation and Analysis", RFC-1305.

[7] The W3IWI/TAPR TAC-2 (Totally Accurate Clock) Project, see `http://www.tapr.org/tapr/html/tac2.html`. Other cards are also being considered.

[8] Minutes of the TERENA TF-ETM meeting, April 24, 1997, see `http://www.terena.nl/task-forces/tf-etm/meetings/minutes-240497.txt`