Traffic Measurements

By
Tapani Nieminen
29.11.1999 in course S-38.149
Email:Tapani.Nieminen@hut.fi

1. Introduction

Vern Paxson starts his most recent paper with claim: “Historically, the Internet has been woefully under measurement and under-instrumented. The problem is only getting worse with the network’s ever-increasing size.” The following resume tries to lighten some of the measurements reported from the Internet. There are many differences both in the methods used and the goals aimed. Since it is quite understandable, that the results are not very well comparable. The purpose of this presentation is more view different approaches than declare the ultimate solution for question "How Internet should be developed?"

Erramilli and Wang [1] looked for better tools for the monitoring of the packet traffic levels since traditional methods developed for circuit switched voice traffic could not reliably identify capacity exhaust in packet networks.

Jerkins and Wang [2][3] analyzed ATM cell-level aggregate traffic measured for busy hour from a commercial ATM network. They found that at least in 1997 the traffic was dominated by a few VPI/VCIs and only third of them showed long range dependence. Majority of most active VPI/VCIs had Hurst parameter close to 1 possibly indicating a level-shift or non-stationarity.

Paxson[8] measured and analyzed the end-to-end Internet dynamics by utilizing the N^2 scaling property of measurement framework to measure a sufficiently diverse set of Internet paths for reflecting general Internet behavior. To cope with such large-scaled measurements requires attention to calibration using self-consistency checks; robust statistics to avoid skewing by outliers; and automated “micro-analysis,” such as that performed by tcpanaly, that we might see the forest as well as the trees. He removed packet filter errors and TCP effects, and shows how TCP-based measurement provides a viable means for assessing end-to-end packet dynamics.

Maxenschuk and Lo measured delay and lost packets [4] from intrastate, cross-country and international Internet connections to estimate achievable voice quality. Siler and Walrand [5] present a collection of algorithms for estimating QoS parameters using on-line measurements. Claffy et al. [6] present a methodology for profiling IP-traffic flows and discuss about optimizing cost of creating flows. Both Claffy [7] and Paxson [9] describe the present tools and architectures for Internet measurements in CAIDA and NIMI. New Abry-Veitch estimator [11] is proposed to estimate long-range Dependence also for non-stationary processes.

2. Measuring traffic to identify capacity exhaust

Measurements of traffic levels are an essential input to performance models and traffic engineering algorithms and are also necessary to trigger congestion controls and operator alarms. Packet measurement analysis have shown that packet traffic is fundamentally different from circuit switched voice traffic as well as many standard theoretical traffic models such as Poison, batch-Poisson and Markov Modulated Poisson Process (MMPP). Packet traffic exhibits fractal or self-similar properties, which are associated with the well known burstiness of packet traffic an shown to be central to the resolution of several unresolved problems in the monitoring of packet traffic levels.

The choices of measurements and time scales depend on the purpose of the measurement. Traffic measurements that serve as triggers for congestion control need to be made on finer time scales than measurements in support of traffic engineering.

Erramilli and Wang [1] studied the monitoring of the packet traffic levels for operation and traffic engeneering in different packet networks including X.25, ISDN, Common Channel Signaling, Frame relay, switched multimegabit data Services (SMDS) and ATM networks. They noticed that measurement time scales were in order of 15 to 30 minute as a carry over from circuit switched voice traffic. Traffic levels were monitored measurements of either packet counts or switch element occupancies.

They concentrated to measurements needed to identify capacity exhaust. That is to identify the usage levels above which the packet network resource is so occupied, that grade of service objectives such as the average queueing delay can no longer be met. Measurement studies have indicated the presence of significant statistical properties in packet traffic that are more characteristics of fractal processes than conventional stochastic processes. The only apparent common feature in packet traffic data sets is that traffic is bursty. So one could argue, that fractal features are associated with burstiness. The engineering challenge is to describe such complexity parsimoniously i.e. with as few parameters and measurements as possible.

- what is the minimum set of operational measurements that have to be collected to accurately asses network load and performance?
- Over what time scales should operational rate and utilization measurements be made for a given packet network?
- How should these measurements be interpreted?
- Is there an alternative to fine timescale measurements?

2.1 Peak-to-Mean Ratio (PMR)

Several existing packet switch systems do measurements in shorter timescales (e.g. 1 to 10 s.) but due the huge amount of data they save only the peak value and report it every 15 or 30 minutes. PMR is calculated by taking the ratio of peak rate to an average rate over specified peak and average rate measurement intervals. It is shown to give so indication of capacity exhaust and augment long term average rate measurements. For processors PMR is computed taking peak arrivals rate over intervals and for packet trunks by taking the peak byte count.

2.2 Squared Coefficient of Variation (CSQ)

If X is packet interarrival time random variable with mean $\overline{X}$ and variance σ_x^2, then the squared coefficient of variation is defined

$$CSQ = \sigma_x^2 / \overline{X}^2 \qquad (2.1)$$

Estimation of CSQ requires measurement of interarrival times and typically it is not included in the set of on-line network measurements. CSQ for Poisson traffic is 1 and higher values correspond to higher burstiness.

2.3 Correlation Dimension (D_C)

A measure of burstiness proposed is fractal correlation dimension. Smooth traffic processes like Poison have a dimension of 1. Analyzed traffic streams have dimensions less than 1. Decreasing dimension corresponds to increasing burstiness. D_C doesn't directly provide information about the size of a set, but how non-uniformly mass is distributed.

It is possible to consider several types of fractal dimension. Here used correlation measure C(t) defined as the fraction of the number pairs points that are less than a distance t apart. It can be calculated

$$C(t) = \sum_{i}^{M} \sum_{j}^{M} \frac{1}{M^2} \cdot C_{ij} \qquad (2.2)$$

Where C_{ij} is 1 or 0 depending whether the arrival points indexed i and j are within t of each other and M is the total number of arrivals. Associated with the correlation measure is the correlation dimension D_C defined as follows:

$$D_C = \lim_{t \to 0} \frac{\log C(t)}{\log t} \qquad (2.3)$$

In practice C(t) is calculated for a range of t values and it is seen if C(t) follows a scaling relation of the type for some range of t

$$C(t) \sim \alpha \cdot t^{D_C} \qquad (2.4)$$

The slope of the best linear fit to the log C(t) vs. log t is then taken to be an estimate of D_C.

Other measures proposed are index of dispersion counts (IDC), peakedness and the Hurst parameter. Peakedness (z) of an arrival process is defined as the variance of busy servers from an infinite group of servers and there is no natural choice of the distribution or the service time for the range of packet networks. Also IDC and Hurst parameter were not analyzed [1]. The later was reasoned with the unfeasibility to routine measurements.

In the analysis based on simulation of actual Ethernet and ISDN D-channel packet traffic traces CSQ and D_C were found insufficient to estimate queuing performance. PMR with carefully selected timeintervals (Ethernet ~1 second and ISDN ~10 seconds) could provide information on capacity exhaust. The explanation is that queueing delays (even averages) are dominated by a few heavy intervals. This is supported by the result, that average waiting times are typically many times the average busy period. The capacity exhaust was estimated to occur, when the long-term utilization level is about 1/PMR.

3. Measuring and analyzing ATM cell-level aggregate traffic

Jerkins and Wang analyzed ATM cell-level aggregate traffic measured for busy hour from a commercial ATM network. The ATM data device used is non-intrusive, lossless and can store ATM traffic cell-by-cell for many hours with timestamp resolution of 50 nanoseconds. Traffic on an OC-3c trunk connecting two ATM switches was collected over 2,5 billion cells translating to about 150 GB of recorded data.

The traffic was dominated by a few VPI/VCIs. The whole aggregate traffic had Hurst parameter roughly 1, which could mean a level-shift or non-stationarity. Also there was a regular structure ~60 Hz possibly caused by some video applications in the dominating VPI/VCI named A and contributing > 80 % of traffic. The non-A aggregate traffic appeared to be self-similar with Hurst parameter between 0,8 and 0,9. There was only 2 VPI/VCIs (0,16 % utilization) subscribed as CBR service with very "regular" structure and Hurst parameter close to 1.

The time scales of self-similar traffic behavior was said to have cut-offs, lower below 10 ms for Ethernet where short-range correlation dominate and upper when time-off-day effects begin to govern.

The results provide evidence that aggregate ATM traffic can have long-range dependent features and individual VPI/VCI can exhibit ON-OFF behaviors. An experience has shown that traffic parameter values can vary drastically between networks and even within the same network as the network evolves. This necessitates the task of continued monitoring and analysis of high-speed packet networks.

In Further analysis[3] of ten most active individual VPI/VCIs varying temporal and spatial correlation were found. Three demonstrated strong positive temporal correlation even in the long term, six exhibited periodicity suggesting signaling applications and one had strong negative correlation of lag 1 suggesting anti-persistence.

4. End-to-end Internet dynamics

Accurately characterizing end-to-end Internet dynamics—the performance that a user actually obtains from the lengthy series of network links that comprise a path through the Internet—is exceptionally difficult, due to the network's immense heterogeneity. It can be impossible to gauge the generality of findings based on measurements of a handful of paths, yet logistically it has proven very difficult to obtain end-to-end measurements on larger scales.

At the heart of Paxson's study [8] lies the NPD (network probe daemon) measurement framework, in which a number of sites around the Internet run a specialized daemon that provides measurement services to authenticated users. The key scaling property of this framework is that, for N participating sites, it can probe $O(N^2)$ Internet paths. This scaling enabled to probe over 1,000 Internet paths, due to the participation of 37 sites. Consequently, the data for his analysis is more than an order of magnitude richer than that available for previous end-to-end studies and a serious argument can be made that we can indeed extrapolate his findings to conclusions about Internet paths in general.

4.1 End-to-end Internet routing behavior

The first part of Paxson's work looks at the behavior of end-to-end routing the series of routers over which a connection's packets travel. Measurements were done using two experimental runs, one at the end of 1994 and one at the end of 1995. Each measurement comprised a single traceroute from a randomly selected site to another randomly selected site. Based on 40,000 measurements he analyzes routing “pathologies” such as loops, outages, and flutter; the stability of routes over time; and the symmetry of routing along the two directions of an end-to-end path

First we must identify anomalies present categorised to number of routing pathologies:

- unresponsive routers, routing loops, routing changes in the middle of measurement, erroneous routes, omission of TTL decrement, and infrastructure failures, all of which were rare;
- host and stub network outages, which were fairly common
- “fluttering,” in which the path rapidly alternated between two different routes. In 1994 fluttering was quite common, and sometimes had great impact on the routes of consecutive packets sent by a host. But, like outages, samples are not persuasively representative, and fluttering was rare in 1995

The likelihood of encountering a major routing pathology more than doubled between the end of 1994 and the end of 1995 rising from 1.5% to 3.4%, indicating that routing degraded over the course of 1995.

There are two distinct types of stability that are of interest. The first is *prevalence*: whether we are likely to observe the same route in the future as at the present. The second is *persistence*: whether the route we observe at the present is likely to remain *unchanged* for a considerable period of time. Most Internet paths are heavily dominated by a single dominant route, but that the length of time over which routes persist varies greatly, from seconds to many days.

Routing asymmetries have little direct impact on end-to-end performance, but they introduce significant measurement problems, because they cloud the accuracy of the easiest form of measurement, “sender-only” measurement, in which no receiver cooperation is required. It was found that about half of all Internet routes exhibited a major asymmetry, in which at least one city differed between the route from A to B versus that from B to A.

4.2 End-to-end Internet packet dynamics

The second part of Paxson's work studies end-to-end Internet packet dynamics. He analyzed 20,000 TCP transfers of 100 Kbytes each to investigate the performance of both the TCP endpoints and the Internet paths. The measurements used for this part of study are much richer than those for the first part, but require a great degree of attention to issues of *calibration*, which he addressed by applying *self-consistency checks* to the measurements whenever possible. He found that packet filters are capable of a wide range of measurement errors, some of which, if undetected, can significantly spoil subsequent analysis. He further find that network clocks exhibit adjustments and skews relative to other clocks frequently enough that a failure to detect and remove these effects will likewise pollute subsequent packet timing analysis.

Each transfer was traced using the tcpdump utility at both the sender and the receiver, resulting in two trace files called a “trace pair.” All findings are based on analyzing trace files and trace pairs.

Using TCP transfers for network path “measurement probes” gains a number of advantages, the chief of which is the ability to probe fine time scales without unduly loading the network. However, using TCP also requires us to accurately distinguish between connection dynamics due to the behavior of the TCP endpoints, and dynamics due to the behavior of the network path between them. To address this problem, he developed an analysis program, tcpanaly, that has coded into it knowledge of how the different TCP implementations in the study function. In the process of developing tcpanaly, was in tandem developed detailed descriptions of the performance and congestion-avoidance behavior of the different implementations. Some of the implementations suffered from gross problems; the most serious of which would devastate overall Internet performance, were the implementations ubiquitously deployed.

The analyzing of packet dynamics began by characterizing packet-forwarding pathologies: out-of-order delivery, packet replication, and packet corruption. The frequency with which packets arrive in a different order than sent varies enormously among Internet paths. Reordering often occurs in conjunction with the route “flutter” pathology, but also there was numerous instances in which it occurred in the absence of flutter, and some instances in which massive reordering events occurred due to “pauses” in router forwarding. The possibility of reordering limits how quickly a TCP sender can infer a packet loss using the “fast retransmission” mechanism. Paxson found that this mechanism could be altered to retransmit more efficiently only if both the TCP sender and receiver would be changed.

Packet replication—the network delivering a single packet more than once—does indeed occur, but it is exceptionally rare. On the other hand about 1 Internet data packet in 5,000 arrives with different data than what was originally sent. This rate is high enough that, given TCP' s 16-bit checksum, about one packet in 300,000,000 will be accepted with undetected errors. The Internet carries many more packets than this each day.

4.2.1 Estimating bottleneck bandwidth

Next we identify a network path' s *bottleneck bandwidth* before analyzing packet loss and delay because it determines what we call the “self-interference time constant,” Q_b. Two data packets of size b sent less than an interval Q_b apart must necessarily queue at the bottleneck element of the network path. Thus, knowledge of Q_b enables to determine which of measurement probes were perforce correlated and to distinguish between the loss of data packets that had to queue behind their predecessors (“self-interference”), versus those lost even they did not have to queue due the connection's own loading.

This “packet pair” technique could produce incorrect estimates in the presence of: excessive noise; packet reordering; changes in the bottleneck bandwidth; or network paths in which the bottleneck is comprised of multiple, separate channels or links. This last case is particularly interesting, because it leads to erroneously large bottleneck estimates even if the network is completely quiescent. The problem lies in the fundamental assumption made by packet pair that packets must queue behind one another at the bottleneck and be served by it one at a time. For a multi-channel or multi-link bottleneck, however, this assumption does not in fact apply, and a pair of packets can traverse the bottleneck without it altering the spacing between them.

A robust algorithm was developed for estimating bottleneck bandwidth, based on “packet bunch modes” (PBM). PBM works by stepping through an increasing series of packet bunch sizes, and, for each, computing from the receiver trace all of the corresponding bottleneck estimates. The bunch size, termed as the *extent* is denoted by k. for each extent, a window over the arrivals at the receiver is advanced. By focussing on identifying multiple modes in the distribution of the estimated bottleneck bandwidth, PBM can accommodate errors introduced by noise, as well as detecting changes in bottleneck bandwidth and the presence of multi-channel links. By using receiver-based measurement, it also can cope with packet reordering, and with the possibility of asymmetries in the bottleneck bandwidths along the two directions of a network path.

PBM was calibrated by testing whether we could associate known, common link speeds with its estimates. We found that we could almost always do so. Once we had faith in PBM' s accuracy, we could then test other estimation methods against PBM to see how well they perform. We found that receiver-based packet pair performs almost as well, if we can tolerate failing to detect shifts in bottleneck bandwidth or multi-channel links, both of which prove rare. Sender-based packet pair, however, does not perform nearly as well, due to the additional noise incurred by measuring timings that reflect the traversal of packets in both of a path's directions. Finally, we find that about 20% of the time, a path's two directions have *asymmetric* bottleneck bandwidths, but that, along a single direction, the bottleneck generally remains constant over lengthy periods of time.

One drawback with PBM is that it is ad hoc to an unsatisfying degree. It uses a considerable number of heuristics that can only be defended on the basis that they appear to work well in practice.

4.2.2 Robust statistics

The final problem we must address with our analysis strategies is that of widespread noise. For example, if we wish to summarize a connection's round trip times (RTTs), we might at first think to express them in terms of their sample *mean* and *variance* (or *standard deviation*, the square root of variance). However, in practice we find that often a connection observes one or two RTTs that are *much* higher than the remainder. These extreme values greatly *skew* the sample mean and variance, so that the resulting summaries do not accurately reflect “typical” behavior. To address these sorts of problems, statisticians have developed the field of *robust statistics*, that remain resilient in the presence of extremes, or “outliers.”

Examples are use of the *median*, or 50th percentile, as a statistic for summarizing a distribution' s central location, rather than the mean. A robust statistic for measuring variation rather than standard deviation is the *interquartile range* (IQR), which is the difference between a distribution' s 75th percentile and its 25th percentile and characterizes the distribution' s “central variation.” Unlike the mean and standard deviation, the median and IQR are virtually unaffected by the presence of outliers.

4.2.3 Packet loss

In analyzing patterns of packet loss in the Internet, it was found that over the course of 1995, packet loss rates *nearly doubled*, indicating a marked degradation in service. However, these rates required further inspection to understand their implications.

We distinguish between losses of “loaded” data packets, meaning those which necessarily queued behind a predecessor at the bottleneck; “unloaded” data packets, which did not do so unless cross traffic had delayed them; and the small “acknowledgement” packets returned to a TCP sender by the TCP receiver. We find that network paths are well characterized by two general states, “quiescent,” in which no loss occurs, and “busy,” in which one or more packets of a connection are lost. The prevalence of quiescent connections remained about 50% in both our datasets, but for busy connections, packet loss rates increased significantly over the course of 1995.

We found that loaded packets are much more likely to suffer high loss rates than unloaded packets. This is not surprising, since they encounter not only the ambient network load but that of their predecessors; and that acks are more likely to be lost than unloaded packets (or even loaded packets, for high loss rates). We interpret these findings as reflecting the fundamental difference between data packets being sent at a rate that *adapts* in an effort to diminish packet loss, and acks being sent at a rate that does *not* adapt to the rate at which acks are lost. This finding highlights how the loss rates observed by a TCP connection's data packets *differ* from the unconditional loss rates along the path they traverse.

The last comparison between data packet and ack loss rates we made was to determine the degree of correlation between the two rates for a single connection. We found that the two are nearly uncorrelated, indicating that this fundamental property of a network path is *asymmetric*.

It was found that different major regions of the Internet—the United States, Europe, and connections from one to the other—experienced very different loss rates. Then, after showing that loss rates follow the well-known diurnal cycle reflecting working hours and off-work hours, we analyzed variations in the time of day during which our measurement apparatus succeeded in executing a measurement. For North American sites, these successes were uniformly spread over the 24 hours of each day. For European sites, though, the frequency of successes dipped to low points in patterns that closely matched the loss-rate cycle, indicating that our European measurements suffered from a discernible *bias* towards underestimating loss rates.

Another question was whether packet loss events are well modeled as independent, since this assumption is sometimes made when theorizing about network behavior. We found that loss events are instead strongly correlated. Furthermore, the duration of loss “outages” exhibits clear Pareto distributions with infinite variance, which accords with a recent model of how individual connection behavior can give rise to “self-similar” aggregate traffic behavior.

We then looked at the question of *where* packets are lost along an Internet path. In particular, whether they are lost before or after the bottleneck element. From careful analysis of timing information we can sometimes distinguish between these two. We found that, while most losses occur at or before the bottleneck, a significant minority (roughly 25%) occurs after.

We next evaluated how packet loss rates evolve over time, with an eye towards gauging the efficacy of caching packet loss statistics associated with a path in order to predict future path performance. We found that a path' s state, in terms of “quiescent” or “busy,” is a good predictor of its future state for many hours, but a path's observed loss rate is *not* a good predictor of its future loss rate.

We find that most TCPs retransmit fairly efficiently, and that deploying the proposed “selective acknowledgement” option would eliminate almost all of their remaining unnecessary retransmissions. However, some TCPs incorrectly determine how long to wait before retransmitting, and these can suffer large numbers of unnecessary retransmissions.

4.2.4 Packet delay

In an analysis of end-to-end packet transit delays both round-trip times (RTTs) and one-way transit times (OTTs) were found to exhibit great “peak-to-peak” variation, meaning that maximum delay far exceed minimum delays. OTT variations for the most part are asymmetric. The only clear correlation occurs between the order-of-magnitude (logarithm) variation in the two directions. On the other hand, OTT variation is clearly correlated with packet loss rates, but it is not a good predictor of future OTT variation, in accord with the finding that packet loss rates are not good predictors of future loss rates.

Packet *timing compression means that* a group of packets arrives at their receiver more closely spaced than when they were sent. We identify three types of compression: ack compression, data packet compression, and receiver compression. Each requires somewhat different assessment considerations. Overall, none of the three types occur frequently enough to pose a significant problem in terms of network performance and stability. Their presence does, however, complicate path measurement efforts, which must use judicious filtering to avoid mistaking compression events for different network effects, such as a temporary increase in bottleneck bandwidth.

Paxson investigated the *time scales* over which queueing occurs, by determining on which time scales the maximum sustained and peak OTT variations were observed. He found that both occur most frequently on time scales of about 100–1000 msec, though, as with many Internet phenomena, we also found a wide range of behavior beyond this region. (In particular, we sometimes found maximal queueing occurring on much longer time scales.)

The last aspect of packet delay we analyzed was the degree to which it reflects *available bandwidth*. We did this by studying the ratio between the delay a packet incurred due to its connection's own loading of the network path, versus the total delay it incurred. This ratio correlates well with the overall throughput achieved by a connection. However the accuracy of the ratio is diminished by the presence of errors in estimating the bottleneck bandwidth.

Paxson observed a distinct decrease in available bandwidth over the course of 1995, though there is significant regional variation, with U.S. sites enjoying considerably more available bandwidth than European sites. Finally, we investigated how available bandwidth evolves over time. We found that a connection's available bandwidth is a fairly good predictor of future available bandwidth out to time scales of minutes to hours, but diminishes significantly for longer time periods.

The delay variation is an interpretation of how it reflects the available bandwidth. Let T_i be the time when the sender transmits *i*th packet. A notion of data packet *i*’s “waiting time,” $\lambda_i = Q_b + \max[(T_{i-1} + \lambda_{i-1}) - T_i, 0]$ meaning the total delay it incurs due to both queueing at the bottleneck behind its predecessors, and the time required for its own transmission across the bottleneck Q_b. For simplicity, we assume that is the same for each data packet. Since every packet requires time Q_b to transit the bottleneck, variations in OTT do not include Q_b but will reflect $\psi_i = \lambda_i - Q_b$. ψ_i is the expected additional delay that packet i will experience because it will have to queue behind its predecessors at the bottleneck.

Let γ_i denote the difference between packet i's measured OTT and the minimum observed OTT. We interpret γ_i as reflecting queueing delays. If the network path is completely unloaded except for the connection's load itself, then we should have $\psi_i = \gamma_i$ That is, the measured extra delay (γ_i) can all be accounted for by the expected extra delay due to *i* queueing behind its predecessors. More generally, define

$$\beta = \frac{\sum_i (\psi_i + Q_b)}{\sum_j (\gamma_j + Q_b)} \tag{4.1}$$

β then reflects the proportion of the packet's delay due to the connection's own loading of the network. If $\beta \approx 1$ then overall, we have a situation approximating namely all of the delay variation is due to the connection's own queueing load on the network. On the other hand, if $\beta \approx 0$ then the delays experienced by the packet are much higher than those due to their own transmission times across the bottleneck and their own queueing behind their predecessors. In this case, the connection's load is insignificant compared to that of other traffic in the network. This observation provides the basis for hoping that we might be able to use to estimate available bandwidth without fully stressing the network path, unlike other available bandwidth estimation techniques.

5. Delay and lost packet measurement

Maxenschuk and Lo measured delay and lost packets[4] from intrastate, cross-country and international Internet connections to estimate achievable voice quality. If the line drops out for a short period in a conversation, we can ask the other party to repeat, but if 2-3 packets are lost every second the received speech is very difficult to understand. The "quality" of the connection is defined as the fraction of the time that a channel is free of distortion for intervals that are long enough (Minimum Loss Free Interval from 0,5 - few seconds) to transmit active speech segments. Every interval, where at least one packet was missing was considered distorted.

For two weeks they transmitted every hour 10 minutes of 64 B UDP packets every 20 ms. Packets carried timestamps and session and sequence numbers and if packet had not arrived in time for disassembly, it was considered lost. Receiver had a delay of 100, 200 or 400 ms (and on international connection also 6,4 s as reference) to compensate the variable delay from Internet, which occasionally rise up to ten of seconds.

When the receiver was assumed to be able to correct 1, 2 or 3 packets during an interval, the main improvement came from correcting single packets. If several packets are lost then there is usually missing more than three. The results indicate that the Internet is best suited to carry voice for short distances, while most applications use it as a long distance alternative.

6. References

[1] A. Erramilli and J.L. Wang, Monitoring packet traffic levels, Proc. Globecom'94, San Francisco, CA, pp. 274-280, November/December 1994

[2] J.L. Jerkins and J.L. Wang, A measurement analysis of ATM cell-level aggregate traffic, Proc. Globecom'97, Phoenix, pp. 1589-1595, November 1997

[3] J.L. Jerkins and J.L. Wang, Establishing broadband application signatures through ATM network traffic measurement analyses, ICC 98, vol. 2 , pp. 837 -843 vol.2, 1998

[4] N.F. Maxemchuk and S. Lo, Measurement and interpretation of voice traffic on the internet, Proc. ICC'97, Montreal, Canada, pp. 500-507, June 1997

[5] M. Siler and J. Walrand, On-line measurements of QOS for call admission control, Sixth International Workshop on Quality of Service, pp. 39-48, 1998

[6] K.C. Claffy, H.-W. Braun, and G.C. Polyzos, A parametrizable methodology for Internet traffic flow profiling, IEEE JSAC 13, 8, 1481-1494 , October 1995

[7] K. Claffy and T. Monk, What's next for Internet data analysis? Status and challenges facing the community Proc. IEEE 85, 10, 1563-1571, October 1997

[8] V. Paxson, Measurements and Analysis of End-to-End Internet Dynamics, Technical Report, University of California, Berkeley, April 1997

[9] V. Paxson and J. Mahdavi, An architecture for large-scale Internet measurement, IEEE Communications Magazine 36, 48-54, August 1998

[10] V. Paxson, End-to-End Internet Packet Dynamics, IEEE/ACM Networking 7, 3, 227-292, June 1999

[11] M. Roughan and Darryl Veitch, Measuring Long-Range Dependence under Changing Traffic Conditions, Proc. INFOCOM'99, New York, NY, pp. 1513-1521, 1999