A Survey of Empirical Measurements of Networked Software Systems

Frederick M. Korz

Department of Computer Science
Columbia University
New York, NY 10027

Technical Report No. CUCS-025-92
June 24, 1992

Abstract

Empirical measurement is an important research tool in networked systems, providing concrete answers to questions including latency, bandwidth, utilization, and communications patterns. The presence of networks, however, complicates the task of measuring software systems. Networks introduce a host of often uncontrollable factors – message latency, network reliability, etc.

We examine issues in empirical measurement, particularly in the measurement of networked software systems. We then present a survey of representative measurement literature using a framework derived from these measurement issues. Finally we suggest the probable direction of future work in the field.

Contents

1 Introduction

Empirical measurement is an important research tool in the study of networked software systems. It provides concrete answers to questions such as latency, throughput and utilization. Empirical answers can verify, supplement or replace assumptions.

A networked software system (NSS) is a system in which computational processes, executing on separate processors, communicate to perform a task. In an NSS, communications costs contribute significantly to a task's cost (e.g. time). An NSS can, in turn, be part of a larger system. Examples of networked software systems include distributed databases, remote procedure call (RPC) implementations, and replicated data management subsystems.

Model validation is one major application of empirical measurement. A model of an NSS is an abstraction of that NSS. It provides a mechanism to predict the operational properties, such as performance, of the modeled system. Empirical measurements, as opposed to simulations or evaluations, provide concrete data to validate or falsify a model. Examples of measurement supporting modeling include [GL91] and [CDJM91].

System performance determination is another major application of empirical measurement. System construction is considered by many an art([Lam84]). It includes an iterative cycle of thinking, building (or perhaps simulating), testing, and evaluating against goals. Empirical measurement has a role in this process. A system builder often employs models and assumptions when thinking about system design. Measurement's role in the modeling aspects is outlined above. In addition to these models, the builder needs performance information as feedback to guide the implementation. Empirical measurement provides system builders this quantitative feedback. It allows the verification of assumptions used in system design and allows the testing and tuning throughout implementation. Finally, when a system is complete, empirical measurement results provide the backing for system performance claims.

The development of the Network Time Protocol (NTP) ([Mil89b, Mil89a]) provides an example of the use of measurement in design and early implementation while other papers surveyed in sections 3.2 and 3.3 provide examples of measurements of final products.

Measurements of NSSs have additional complicating factors not present in measurements of centralized (unnetworked) systems. The chief complicating factor is the presence of a communications network. The communications network linking the elements of the system introduces unpredictability. None of the research reviewed found the network a deterministic, invariant factor. Instead, all found that the network contributed significantly to the system and was a major source, if not the major source, of variability in the system. Investigators must contend with the network when measuring performance at various stages of a system's life – design, implementation, tuning, and final evaluation.

This paper addresses two related areas. Firstly it identifies and organizes a set of issues in measuring NSSs. Secondly it presents a representative survey of current and/or distinguished work in the field, organized and evaluated along the lines of the issues identified.

In section 2 we explore issues in the design and conduct of network software measurement. In section 3 we formulate an organizing framework from the issues of section 2 and survey several groups of networked software empirical measurement papers. Finally in section 4 we extrapolate the uses and trends in the issues and approaches laid out in the previous sections.

2 Empirical Measurement Issues

Studying a system involves a sequence of several steps. First one formulates an initial model of the system, identifies the performance characteristics of interest, and devises means of measuring these characteristics. Then one performs the experiments, analyzes the data, and evaluates the results. The results may not adequately describe the system's behavior making revisions and repetition necessary as more is learned.

One can identify three areas of importance in carrying out the abstract sequence, mentioned above. They are expressed as questions in the first column of Table 1. The questions have, in turn, subordinate aspects, shown in the second column. Together the questions serve in at least two ways. They highlight issues in the design of experiments, clarifying goals and methods. They also can be used as the basis of a taxonomy of empirical measurement of NSSs. Appendix A explores the dimensions and subdimensions of Table 1.

Dimension	**Subdimension**	**Question(s)** / **values or range**
	⋮	
What is measured?	Quality Focus	What behavioral qualities are of interest?
		Latency, Throughput, Traffic Distribution, ...
	Layer Focus	What layer(s) are being studied?
		primary layer(s) and spread
	Environment	What is the test environment?
		live, controlled, private
How is it measured?	Traffic	Source of the traffic for the measurement?
		Existing to introduced
	Involvement	How are measurements picked off?
		Participant to external observer
	Intrusiveness	Is the software instrumented?
		Intrusive or non-intrusive
How is it processed?	Acquisition	What processing is needed to acquire data?
		Amount required.
	Analysis	How is the analysis processing done?
		Amounts online and offline.

Table 1: Measurement Issues

3 Survey of Representative Work

The questions and subquestions discussed in section 2 and appendix A can serve as a detailed framework for a taxonomy. They are, however, too detailed to give a top level grouping of work in the field.

There *are*, however, discernible grouping of papers. In reading and evaluating the work in terms of section 2, the papers selected form three cohesive groups – traffic characterization, support-protocol performance, and system performance. The grouping most closely coincides with the abstraction of intent or goal. They serve to provide a first partitioning of the work in system measurement.

A fourth group, other, comprises notable works that lie outside the other three groups or span multiple groups. The latter happens when a study undertakes to do many things and includes multiple experiments, some of which would place it in one group, and others in another group but neither set of experiments dominates the study.

The papers surveyed below were selected to illustrate the background, growth and current status of the field and are not a comprehensive collection. They are arranged in chronological order within each section – 3.1 through 3.4.

Terminology Empirical measurement literature uses the terms *experiment* and *measurement* loosely and interchangeably. Sometimes the terms indicate single observations. Other times these words indicate series of observations. A similar situation exists with the terms *study* and *experiment* referring to either a series of observations or a set of series of observations.

To clarify matters for the survey, we make three definitions. Given a series of observations, call each observation a *measurement*. Call series of measurements toward a single goal an *experiment*. Similarly, call a series of independent experiments toward understanding a system a *study*.

$$\text{observation} \equiv \text{measurement} \subseteq \text{experiment} \subseteq \text{study}$$

3.1 Traffic Characteristics

The common element of this group of papers is the study of the communications traffic. This is done at one or more layers of peer to peer communication. The papers range in ambition from the simple to the complex. Simple studies often investigate aggregate quantities such as mean packets per second at the data link level. In contrast, complex studies are more detailed and span multiple protocol layers. To reach conclusions and insights, these detailed studies often correlate observations across different protocol layers or in one layer across time.

There are many types of traffic characterization. One type is characterizing traffic the pattern of events as a function of time. For example, observing the number of Internet Protocol packets per minute for a week. Another characterization type is by pattern of events with respect to some classification scheme, independent of when those events occur. For example, one could establish a classification by Ethernet protocol field and count all Ethernet packets for an hour, determining how many fall into each category. This second type of characterization is called a traffic breakdown.

While the quality and layer focuses vary from paper to paper, all papers calculate some distribution information on the traffic. Some papers aim at simply recording data in a fashion similar to memory reference traces. Others aim at detecting traffic characteristics. Still others are directed at demonstrating improvements over existing protocols or protocol implementations. In any case, the data is usually obtained in a production environment.

Most work in traffic characteristics obtains its data by passive observation of existing traffic. Researchers tap the communications medium, promiscuously copying packets. No alterations of the studied software are necessary to obtain the observations.

Promiscuous packet capture requires significant processing power to obtain accurate results. There must be adequate buffering to hold new packets during the processing of the current packet. Until recently (about 1990), timestamping the packets was a high overhead process, putting additional demands on the acquisition system. High precision memory-mapped clocks have since lowered this cost. Once the costs of obtaining packets is considered, a tradeoff is made between online and offline analysis. If sufficient, the remaining processing power at acquisition time can be used for online analysis. On the other hand, if online analysis is too costly, one must examine the costs of archiving some part of the data stream for offline analysis. Of course, mixtures of online and offline processing can be used.

In choosing the measurement traffic load and network environment, a question comes up. Are the traffic load and environment representative of the networked software's typical operating conditions? Another way of looking at this is, is the data collected representative? The fear is that measurements may be conducted under conditions that limit the applicability of the results or, even worse, invalidate them. This is a matter of continued debate which we will not attempt to answer here.

3.1.1 DECnet Study

Chiu and Sudama [CS88] studied DECnet protocol performance and applications. The study describes DECnet traffic characteristics in a live environment. Using this data they compare the performance of several DECnet protocol implementations.

The authors built a traffic model to evaluate the performance of different protocol implementations. The model has multiple pieces – one for each of the network, transport, session and application layers. At each protocol layer, the study categorizes the DECnet traffic according to the sub-model for that layer. Since only the improved protocol implementations use the new features, the authors can determine the improvements' effects. They do this by direct inspection or comparison with known characteristics of the old implementations.

For example, at the transport layer, Chiu and Sudama model three characteristics: flow control overhead, acknowledgement overhead and out of sequence packet behavior. Packets are examined for their transport level contents. Each packet contains either link control information or user data. Link control information is either flow control or other link control (such as link initialization and periodic keep-alive packets (on otherwise idle links)). The authors define two measures of performance. The first is overall transport overhead. This is the ratio of link control packets to data packets. The second is flow control overhead which is the ratio of flow control packets to data packets.

Old DECnet implementations generated one flow control packet for each data packet. That means if all sites used old implementations, then the flow control overhead would be 1.00. If all sites used new implementations, the flow control overhead would far less. This is because new implementations allow essentially open ended sending of packets. The receiver sends link control packets only to start or stop the flow if there is a speed mismatch between sender and receiver. The results show overall control overhead ratios of 0.04 to 0.26 and flow control overhead ratios of between 0.03 and 0.24. This indicates several things. First, most of the sites use the new implementation and that far less network bandwidth is used for transport overhead. Second, of the remaining overhead, most of the overhead is flow control.

All results were gleaned from four traces. The parameters of the measurements are listed in table 2.

What is measured?	Quality Focus	Traffic Distribution and Overhead
	Layer Focus	Network through Application
	Environment	Production-use LANs
How is it measured?	Traffic Source	Existing Network Traffic
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 2: Chiu and Sudama – A Case Study of DECnet Applications and Protocol Performance

3.1.2 Packet Trains in NSFnet

"Traffic Characterization of the NSFNET National Backbone" by Heimlich [Hei90] is a traffic study in two uncontrolled Internet settings (a University LAN and a backbone WAN). The study starts with traditional traffic breakdowns. Using traffic breakdown statistics, Heimlich found the phenomenon of *packet trains* [JR86] in his traces. He modified the routing algorithms of NSFnet to improve their performance by taking advantage of packet trains.

Packet trains are groups of packets between the same source and the same destination occurring in rapid succession. Each packet train is separated from successive trains by inter-train gaps. To set the maximum allowable inter-packet gap (MAIG), one observes the distribution of packet arrival times. From this distribution the MAIG is set as the 90^{th} percentile of all inter-packet gaps. Any inter-packet gap larger than the MAIG causes the packet following the gap to be considered the first car of the next train.

Heimlich modified the software on the router's monitoring machine to clip and record packets' network information. He divided the traffic traces by several criteria including network loading (low or high) and protocol (e.g. FTP, login, TCP, and UDP). He was able, only under conditions of high network loading, to fit each to a Poisson (exponential) distribution. When network loads were low, only piecewise Poisson approximations fit.

Since Poisson models were unable to fully describe the traffic captured, Heimlich looked for packet trains in the traces. He analyzed the packet traces for locality (the probability that the next packet is part of the same train as the current packet). The NSFnet data showed values of 30% under low network loads falling to 10% under high network loads. He characterized packet trains in terms of lengths, inter-car gaps, and inter-train gap thresholds. For example, for the file transfer protocol at times of low network load the mean inter-car gap was 304ms and length 23 while at times of high network load this the gap fell to 139ms and length to 6.8.

Heimlich then compared two routing algorithms in a simulator, using the traces as input. The old algorithm is the production version used in NSFnet routers. The new algorithm assumes the existence of packet trains. It pre-fetches routing information at the beginning of a packet train in anticipation of the following messages. The pre-fetching improved simulated router cache hit ratios by between -9 and 29 percent, depending on cache size and traffic trace examined. For all but the smallest cache sizes (2 and 8 routes) performance of the pre-fetching algorithm equalled or exceeded the old algorithm on the full traffic mixture.

The mechanics of Heimlich's measurements are similar to the other traffic characterization studies in this section. The main parameters are shown in table 3.

What is measured?	Quality Focus	Traffic Distribution, Packet Trains
	Layer Focus	Network and Transport
	Environment	Production-use LANs
How is it measured?	Traffic Source	Existing Network Traffic
	Acquisition	Externally observed
	Intrusiveness	Effectively Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 3: Traffic Characterization of the NSFNET National Backbone

3.1.3 LAN Traffic

"LAN Traffic Analysis and Workload Characterization" by Khalil, et al [KLW90] has two aspects. The first is a detailed traffic classification, similar to work by Cáceres et al [CDJM91] (section 3.1.5) but is restricted to internal LAN traffic. The second examines the applicability of Poisson models as traffic models, similar to [Hei90] (section 3.1.2) and extends Heimlich's results.

The authors capture packet headers and analyse these for packet size and user access distributions. The first group of distributions are packet counts and byte counts by host, identifying communications patterns and showing them to be non-uniform. The second group of distributions looks at packet length breakdowns by transport protocol (`TCP, UDP, and ND`) and discovers them to be trimodal rather than bimodal (the common assumption). The third group of distributions examines LAN utilization and throughput, finding them to be highly variable from moment to moment but that their minima, maxima and time average have a distinct pattern. In particular this pattern is similar to that discovered by Cheriton (Section 3.2.1), with peaks at 9AM, 3–5PM and 3AM.

The authors proceed to check standard modeling assumptions that overall traffic follows a Poisson distribution. Their results are mixed. They confirm the validity of the assumptions for periods of heavy loads. At times of light loading, however, their results demonstrate the inadequacy of Poisson distribution models.

The trace data was obtained by a specially constructed multiprocessor. The first processor, a high-performance single board computer, managed the promiscuous interface, obtained timestamps and recorded the header information. The second processor, a workstation, handled the management of a buffering system – memory, dual disks, dual 8mm tape. The accuracy of the timestamps and the ability to capture all packets in the trace were necessary for the Poisson modeling results. The rest of the measurement is classified in table 4.

What is measured?	Quality Focus	Traffic Distribution, Poisson Model Assertions
	Layer Focus	Network(IP), Transport(TCP and UDP) and Application
	Environment	Production-use Ethernet
How is it measured?	Traffic Source	Existing Network Traffic
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 4: LAN Traffic Analysis and Workload Characterization

3.1.4 High Time-resolution Measurement

In "High Time-Resolution Measurement and Analysis of LAN Traffic" [LW91], Leland and Wilson address the time domain nature of *external* LAN traffic. They characterize the time domain nature of this traffic, analyze current models and use the traffic trace as input to a high-speed wide-area LAN interconnection network simulation. The measurement parameters are given in table 5.

The authors used the equipment described in section 3.1.3 to *accurately* capture traffic. The 50 microsecond accuracy timestamping and heavy buffering were vital for their results. They analyzed the traffic trace for arrival rates, burstiness, and interarrival rates. We examine each of their results in characterizing the traffic's time dependent aspects.

Arrival rates varied widely, showing peaks at 10AM, 1PM and 5PM. At these times internal LAN traffic exceeded 30% of the Ethernet's available bandwidth. External traffic (traffic for which one host is not on the LAN) peaked as high as 14.7% of the available Ethernet bandwidth. This 14.7% corresponded to 95% of the available bandwidth on the external T-1 line that connected the LAN to the regional network. Previous studies by other authors (e.g. [SH80]) had much lower average and peak utilizations.

Burstiness analysis shows that the traffic is bursty over a wide range of time scales. There was a "striking" similarity between the mean packet arrival rate averaged by hour for 307 hours and the same rate averaged by minute for the burstiest 6 hours. Looking at bytes per second, the maxima within 5 second periods were 152 times greater than the means. This ratio of maximum to mean is called the peak-to-mean ratio. At smaller intervals, 5 milliseconds, the ratio was even higher - 715 to 1. Packet rate peak-to-mean ratios were more extreme than the byte rates. Using 5 second averaging windows, the ratio was 73 to 1. Shrinking the windows to 5 milliseconds, this ratio rose to 861 to 1. This burstiness was found over periods varying by 6 orders of magnitude – from "milliseconds to hundreds of seconds."

Various researchers have tried to model traffic using batched Poisson, deterministic batch and Markov-modulated Poisson models. The authors note that this high degree of variability over such a wide domain of time scales, milliseconds to hours, is the reason for the failure of these models to capture the coefficient of variation of the observed traffic. In particular, Leland and Wilson observe that these models "have very little long-range variability: the coefficient of variation for arrivals rapidly converges to zero outside the characteristic time domain of the model." In other words, while these models can be tuned to generate traffic that matches the observed traffic at one time scale, they will fail to match the structure of the observed traffic when examined at other time scales.

The authors note that interarrival times showed sharp peaks in the distribution corresponding to particular protocol implementations. Furthermore, "The peaks ... become less well-defined as the traffic load increases, so that a memoryless (Poisson) arrival model may achieve an acceptable fit to the inter-arrival data for the busiest periods." This behavior is a result of the source of the traffic increase which stems from an increased number of active machines active on the network, more than a few machines becoming more active.

The packet trace was applied to a SMDS℠ interface model. SMDS is a proposed wide-area interconnection service. It provides connectionless, packet delivery service employing small, fixed size data units called cells. The authors wished to determine what size buffering would be needed at the entry points to a SMDS network used to link LANs over a T-1 telephone service. The SMDS link parameters specify a packet loss rate of less than 0.01% of the presented packets and that 90% of all packets will be delivered in 140ms. Analytic simulation models predicted exponential reduction in packet loss and a leveling of delay as buffer pool size increased. The authors discovered, using the external traffic trace as the traffic model, no reduction in packet losses and increasing packet delays with larger buffer pools.

What is measured?	Quality Focus	Time Distribution of Traffic
	Layer Focus	Data Link
	Environment	Production-use LAN
How is it measured?	Traffic Source	Existing External Network Traffic
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 5: High Time-Resolution Measurement and Analysis of LAN Traffic

3.1.5 Individual Conversations in Stub Networks

"Characteristics of Individual Application Conversations in TCP/IP Wide-Area Internetworks" by Cáceres, Danzig, et al [CDJM91] studies traffic on stub networks. Stub networks are organizational networks which are connected to only one wide area internetwork. The authors characterize the traffic generated by network applications protocols (`FTP`, `SMTP`, `UUCP`, `Telnet`, `rlogin`, etc.) between these stub networks and their wide area networks. Their results refute some commonly held assumptions about traffic models and suggest alternative explanations.

For example, traditional modeling assumes that bulk data sources transfer large amounts of data, use large packets in one direction, and are primarily unidirectional. Bulk data traffic was found to be frequently bidirectional and small. `FTP` transfers were less than 10Kbytes per conversation in 80% of all cases. Other bulk sources uses tended to send even less data per conversation. `SMTP` (electronic mail) was shown to transfer bulk data at a 10:1 rather than 20 or 30:1 ratio.

Traditional models assume interactive traffic can be modeled accurately by bidirectional Poisson streams of small packets, where approximately equal amounts of data pass in each direction. Interactive traffic was more unidirectional than assumed. "Interactive applications routinely generate 10 times more data in one direction than the other, using packet sizes ranging from 1 [data] byte to MTU."[1]

The results include other interesting phenomena. The authors observed synchronization points in bulk data transfers. These are points where, because of the transactional nature of the transfer protocol, all outstanding data packets in one direction are acknowledged before any more are sent. An example of this is `SMTP`. `SMTP` flushes all data for the current piece of mail and obtains a receipt before proceeding with any subsequent pieces of mail for the remote host. This synchronization introduces gaps between trains of packets, giving rise to the formation of packet trains. The authors do not, however, present any quantitative analysis of packet train formation as a result of synchronization events.

The authors also generalize their experience analyzing trace data. They expound a meta-algorithm for workload measurement of wide-area internetworks and for building corresponding traffic models.

The authors obtained packet traces from several stub networks. These traces were first classified by network application protocol. Within each application protocol, the authors computed distributions by packets, bytes and *conversations*. A conversation is a series of packets exchanged over a TCP connection between applications on different hosts. The authors selected conversations over packet trains because they were more representative of the structure of the packet exchanges. The remaining mechanics of this study are similar to the other traffic characterization studies in this section. Table 6 gives the main parameters.

What is measured?	Quality Focus	Distribution of Traffic
	Layer Focus	Transport & Application
	Environment	Production-use LANs connected to WANs
How is it measured?	Traffic Source	Existing External Network Traffic
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 6: Characteristics of Individual Application Conversations in TCP/IP Wide-Area Internetworks

3.2 Performance of Supporting Protocols

This group of papers concentrates on aspects of the performance of protocols which are used to support applications. A connectionless transport protocol used to support

[1]An MTU is the maximum size data unit that can be transmitted intact, without the need for fragmentation and subsequent reassembly at the network layer.

an RPC based operating system is an example of a supporting protocol. One can view support protocol implementations as subsystems to final applications. Frequent goals in measuring supporting protocol performance include measuring capabilities (e.g. Section 3.2.2), understanding fault modes and improving performance (e.g. section 3.2.4) and testing alternate implementations or protocol variants (section 3.2.6).

Measurements address some quality, X, of the service. Usually X has dimensions of $[count]/[time]$. Measurements are focused narrowly on the level at which the service is provided and are conducted in either a controlled environment or production environment. Controlled environments are employed for best case results while production environment for typical case results.

Measurements are most often conducted using active traffic generation. The measurement component is usually integrated with the source and/or sink, sitting outside but right on top of the protocol stack. Intrusive measurement techniques insert themselves into the software measured. This creates problems with the mechanics of insertion (need for source code usually). It also impacts the studied system's performance because of the time spent executing the extra code. For these reasons, intrusive measurement techniques are usually avoided.

Acquisition often competes with the protocol for processor cycles. If done correctly, however, the impact of the competition can be minimized by using the processor to make observations outside the times the processor is needed for protocol processing. The analysis is usually done online, saving results in compact form in main memory until the test is over.

3.2.1 Network Measurement of VMTP

In "Network Measurement of the VMTP Request-Response Protocol in the V Distributed System"[CW87] the authors study the relationship between a communications protocol and the distributed operating system it supports. Cheriton and Williamson's stated goal is to characterize the behavior of an implementation of VMTP (Versatile Message Transaction Protocol) in its current environment and extrapolate their results to "the next generation of communication systems." Additional but unstated goals are to confirm design assumptions and check the attainment of design goals.

The paper reports results in latency, throughput, and traffic distributions and their time series. The findings are organized in three sections – Ethernet traffic, VMTP traffic and VMTP performance.

The first stage of analysis is traffic characteristics. The authors determine for the protocols present in the trace, the packet rate, packet size, and data rate distributions as well as the aggregate communication patterns between hosts. An unusual feature of the paper is the profile of several parameters over a 24 hour period. For example the profile of packet rate verses time-of-day shows peaks of activity at 11AM and 3–6PM. This acknowledges that the uncontrolled environment is not necessarily constant. The packet size was found to be essentially bimodal with many small (<100 bytes) packets and large (1050 bytes). The data rate dropped off sharply, with rates of at or under 10Kbytes/second occurring 74% of the time while rates of above 40 Kbytes/second occurring only 6% of the time.

The second stage focuses on VMTP and delves into details of the captured packets. Captured packets bearing requests and responses are correlated to determine message transaction rates and durations, and distributions of message types and sizes. For example, message transaction durations peak at 10ms covering 20% of all message transactions. They fall off with a roughly exponential shape with only 13% taking 100ms or longer. Message size distributions were similar for requests and responses with many short (under 50 bytes) and a peak at 1050 bytes. Responses showed some additional peaks at 100 and 300 bytes. The authors relate their measurement results to specific system activities.

The third stage of the analysis assesses VMTP performance — interprocess communication (IPC) times and data rates, the distribution of traffic by VMTP packet type, and retransmission behavior. In contrast to the first and second stages, these investigations employed pairs of machines "connected by an idle 10 Mbit Ethernet." Depending on the size of the data block transferred (ranging from 1 to 16Kbytes), VMTP throughput between Sun-3 workstations ranged from 2.18 to 4.46Mbits/second. VMTP request packets made up 51% of all VMTP packets, with Response and RequestAck packets making up 23% and 13%. Remaining packet types making up 13% were under 6% each.

Going beyond presenting the performance measurements, the authors checked average throughput measured under worst case conditions – high-load with multiple packets per VMTP transaction. They theorized that loss was due to overruns and retransmission. To test this they introduced inter-packet pauses to allow the slower machine processing time. Throughput improved, (from 1.96Mbits/second to 2.61Mbits/second) confirming the hypothesis.

Tables 7, 8 and 9 give the general measurement parameters. Table 7 covers the three groups of traffic distribution based results – *network use*, *VMTP use* and *VMTP distribution*. Table 8 covers the VMTP performance results measured under best case conditions on a private LAN. Finally, table 9 covers the VMTP performance results measured under simulated worst case conditions – again a private LAN but heavily loaded. This last experiment was intrusive in the sense that the protocol was modified to tune a parameter (interpacket gap delay).

3.2.2 Atlantic SATNET

In "Distributed Testing and Measurement across the Atlantic Packet Satellite Network (SATNET)"[SCS+88], Seo et. al. investigate the performance of the TCP transport protocol over geostationary satellite links. Their primary goals were to understand the link's performance, IP performance over the satellite link and TCP performance using IP over the link. Their secondary goals were to improve performance where possible. Research progressed upward through the protocol layers and outward from the center as more communications links and processors are introduced between the traffic source, the echoer, and the traffic sink. Table 3.2.2 gives the links studied.

At the data link level they measure the performance of the two 64Kbit/second

What is measured?	Quality Focus	Traffic Distribution at network, VMTP, and ...
	Layer Focus	Data link, Network & Transport
	Environment	Production LAN
How is it measured?	Traffic Source	Existing Traffic
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture & transfer to storage
	Analysis	Offline

Table 7: Network Measurement of VMTP – Network traffic

What is measured?	Quality Focus	VMTP latency & throughput
	Layer Focus	Transport
	Environment	Isolated LAN
How is it measured?	Traffic Source	generated loads
	Acquisition	endpoint active involved
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Outside of protocol cycles
	Analysis	Offline and online

Table 8: Network Measurement of VMTP – VMTP traffic

channels. The channels are managed by a *Priority Oriented Demand Assignment* allocation scheme. PODA uses part of the link bandwidth to schedule the remaining bandwidth among the ground stations. The signal propagation delay was 0.26 seconds. Adding reservation and queuing, the minimum packet delay was 0.6 seconds and with all factors included the average delay was 0.8 seconds. Overall the channels performed with an effective bandwidth of 37Kbits/second and an error rate of 0.3% for maximum size packets.

At the network (IP) level, the researchers studied performance of IP from one ground station to another and, adding in ground station gateways, from gateway to another gateway. Performance at these levels was linked to the data link characteristics. For example, they experimented with different size packets and compared the gateway to gateway theoretical maximum performance, measured peak performance and measured steady performance. The measured steady rate results were 59% to 78% of the theoretical rates, depending on packet size. The gateway to gateway host throughput was limited by 50Kbit/second connecting lines.

The authors then examined host to host IP performance for traffic through the gateways and satellite link. They measured throughput rates, round trip time distributions, packet sequence disordering and packet loss. Among their results they found that RTT, pack disordering and packet loss all increased with distance. The packet

What is measured?	Quality Focus	Distribution and causes of retransmissions
	Layer Focus	Transport
	Environment	Private LAN
How is it measured?	Traffic Source	generated loads
	Acquisition	endpoint active involved
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Outside of protocol cycles
	Analysis	Online

Table 9: Network Measurement of VMTP – VMTP performance

IP	Local SIMP both to Satellite echo and through to remote SIMP
IP	Local Gateway to local SIMP, remote SIMP, and remote gateway
IP	Local Host to local gateway, local SIMP, remote SIMP, remote gateway
TCP	Local Host to local SIMP, remote SIMP, remote HOST

Table 10: SATNET Protocols and links

disordering ratio was quite close to the predicted value derived for 2 channels. The packet loss rate, between 9 and 106 lost per 10000, was in a agreement with previously measured bit error rates.

In the transport level investigations, the authors measured the existing TCP implementation's performance in terms of throughput, delay, loss, etc. This provided a baseline. They then implemented Karn's *Round Trip Time estimation* (RTT) ([KP87]) and Van Jacobson's *Slow Start* ([Jac88]) algorithms. The performance improvements were significant (from 3–4Kbits/sec to greater than 12 Kbits/sec) meriting the permanent adoption of the algorithms.

The research was performed by a distributed team of researchers at 6 organizations in 5 nations. They quickly discovered the need for standardized measurement and reporting practices as well as calibrating benchmarks to make independent results comparable.

Two types of measurements were done – active and hybrid passive. The active ones provide involved measurements using sources, sinks and reflectors (echoers). The semi-passive observations used active traffic generation plus passive packet capturing. The some of the TCP performance results were intrusive to the TCP code. The intrusive ones were used to determine relative characteristics of alternative strategies rather than absolute performance numbers. Table 11 contains the remaining measurement parameters.

What is measured?	Quality Focus	Average and peak throughput of TCP over SATNET Channel
	Layer Focus	Transport and relation with data link and network
	Environment	Production-use and idle channels
How is it measured?	Traffic Source	Infinite source/sink
	Acquisition	Active involvement
	Intrusiveness	(mostly) Non-intrusive
How is it processed?	Acquisition	Run Round Trip test drivers
	Analysis	Online

Table 11: Distributed Testing and Measurement across the Atlantic Packet Satellite Network (SATNET)

3.2.3 Firefly RPC

"Performance of Firefly RPC" by Schroeder and Burrows[SB89], investigates Firefly Remote Procedure Call (RPC) running on DEC Firefly multiprocessors. Firefly RPC can be used for communications between address spaces on the same processor, on different processors on the same machine or on different machines. Table 12 gives the measurement parameters.

Each aspect of RPC operation in the intra- and inter-machine cases is examined and an experiment designed to measure it. For example, they measured parameter martialing and unmartialing time, underlying transport protocol (UDP on IP) overhead and ethernet interface overhead. Once these experiments were completed the authors could make extrapolations of what would happen if a component were replaced with a faster one. For example, a 10-fold local area network speed increase would only improve RPC performance by 4% to 18%. If, on the other hand, the processor speed were tripled, performance would rise by 36 to 52%. This showed that CPU use was the bottleneck.

What is measured?	Quality Focus	RPC Transport Latency & Throughput
	Layer Focus	Transport protocol with details down to the hardware level
	Environment	Laboratory LAN
How is it measured?	Traffic Source	Existing External Network Traffic
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Client
	Analysis	online

Table 12: Performance of Firefly RPC

3.2.4 Congestion Avoidance and Control

In October of 1986, on a link between Lawrence Berkeley Laboratory and the University of California at Berkeley, TCP connections experienced a sudden, inexplicable reduction of throughput from 32Kbits/second to 40bits/second. "Congestion Avoidance and Control"[Jac88] describes Van Jacobson investigations. Jacobson recreates *congestion collapse* in a laboratory setting, measures and analyzes it. He devises and implements corrections which have since become standard in TCP implementations.

The author presents the principle of packet conservation for a TCP connection. He identifies the three ways to violate this principle which we review together with his results below.

The first way to violate packet conservation is to fail to reach equilibrium. Jacobson shows that TCP is self clocking and stable once started but it is hard to start. To solve this he presents the *slow-start* algorithm[Jac88]. Slow-start avoids a sudden rush of packets when a transmission begins on a TCP connection. Without slow-start, throughput was only 35% of the available bandwidth (7Kbytes/second). Each packet was retransmitted at least once and packets 108 through 116 were sent five times. With slow-start, throughput was low during the first 2 seconds but increased to yield 16Kbytes/second throughput over the same period as the previous test. The throughput, after the slow start, approached the 20Kbytes/second link speed.

The second way to violate packet conservation is to violate equilibrium by injecting a packet into a link before one has exited. Packet injection for new packets is controlled by a sliding window algorithm so this is already controlled. Packets can, however, also be injected into the link by retransmission. Jacobson shows that a good algorithm is needed to set retransmission timers correctly. The algorithm in use at the time of the collapses, which was based on the mean round trip time, had several problems. It was slow to respond to round trip (packet transmission to acknowledgement reception) time changes. It also used a fixed constant in the calculation that caused instability above a network load of 30% and did not handle backoff after retransmitting correctly. Typical timer settings with the then-current algorithm were 2 to 4 seconds more than actual round trip times which were typically 3 seconds. The improved algorithm, based on the mean and variance of the round trip times, corrected these problems. Typical timer settings were only 0.1 to 2 seconds more than actual round trip times.

The third way to violate packet conservation is to fail to adapt to the resource limits of the connection path. If retransmission timers are correctly set, a timeout means either a packet was damaged in transit or the link has become congested. Since error rates are typically low ($\ll$1%), a retransmission is an indication of congestion. The old implementation of TCP did not use the loss information as an indication of congestion. Without a congestion detection and avoidance algorithm, four TCP connections using one link experienced 50% retransmissions, a waste of 25% of TCP data bandwidth, and uneven sharing (8Kbps, 5Kbps, 5Kbps, and 0.5Kbps) of the link. In all, the senders sent 25% more packets than the link could handle. With the introduction of congestion avoidance, the retransmission rate was only 1% of the packets, there was no waste of bandwidth and the sharing was more even (8Kbps, 8Kbps, 4.5Kbps, and

What is measured?	Quality Focus	Throughput & Latency
	Layer Focus	Transport & Network
	Environment	Laboratory simulation of small internet
How is it measured?	Traffic Source	Simulated Network Load
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Client
	Analysis	Minimal processing, usually online yielding direct results

Table 13: Congestion Avoidance and Control

4.5Kbps). The senders in this case did not exceed, on average, the link's bandwidth. The imbalance between the connections was attributable to another difference in the TCP implementations.

The recreation of the congestion collapse and measurements were conducted on a laboratory internet of two 10Mbps Ethernets linked by a a 230.4Kbps microwave link plus appropriate routers linking into and out of the microwave link. The other parameters are given in table 13.

3.2.5 Transport Service in LANs

Svobodova's paper, "Measured Performance of Transport Service in LANs"[Svo89] is a survey of existing measurement work on four reliable transport protocols – OSI TP4/CLNS, TCP/IP, NETBLT and VMTP. It reviews throughput and delay measurement results, noting boundary conditions, discontinuities and anomalies, and their causes. Although there are so many combinations of hardware and software, the measurement parameters, given in table 14, were relatively constant.

The author notes several common threads running through the measurement papers she surveys. First, the performance of transport protocols is improving as CPU speeds and implementations improve. Second, network adaptors(interfaces) can have a high impact on overall performance when their limitations cause (a) bottlenecks or (b) extra work at other protocol layers. Third, customized protocols typically perform better than general protocols. Fourth, the trend in transport protocols is toward offloading some protocol processing from the main processor(s) onto smart network adaptors or auxiliary processors. Finally, almost all this work was done using isolated, controlled laboratory networks.

Throughput varies widely. Even within the same protocol the effects of different CPUs, network adaptors (interfaces), network and implementation parameters, and implementations is seen. For example, OSI TP4/CLNS with 256 byte packets on a 10Mbit/s token bus and a 6MHz Intel 80186 processor achieved 120Kbits/s. An implementation on an 8 MHz processor, using much larger packets (10Kbytes), and employing several protocol implementation improvements had a throughput of 2.8Mbits/s –

far more than expected from a 33% processor speed increase.

Transport protocol delays were only available for OSI TP4/CLNS and VMTP. They vary less widely than the throughput results. The highest latencies were on the order of 4 times the lowest.

What is measured?	Quality Focus	Throughput and latency
	Layer Focus	Transport
	Environment	Laboratory LANs
How is it measured?	Traffic Source	Generated loads
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Live
	Analysis	Online

Table 14: Measured Performance of Transport Service in LANs

3.2.6 High Speed Networking at Cray Research

In this paper, "High Speed Networking at Cray Research"[N+91], Nicholson et. al. use performance measurement to support improvements in their TCP/IP implementation. These improvements are motivated by the characteristics of high speed, long delay network links for TCP and the limitations of the basic TCP/IP implementation. For example, with the 64Kbyte sliding window of TCP and a 30ms link round trip time, the fastest TCP can transmit data is 17Mbits per second. This is only 39% utilization of a T-3 line.

The authors engage in three types of improvements. The first type comes from protocol options already in the literature. They adopt options that expand the sliding window ("WINDOW SCALE") and handle potential sequence number wrap-around("ECHO"). The second type comes from implementation improvements, many of which are also already published but not widely adopted. These include slow-start (see section 3.2.4), and header prediction[Jac90]. The third type of improvement came from tuning system parameters such as kernel buffer sizes and kernel to user space transfer sizes.

The authors arranged these changes as a series of steps. After each step they measure the throughput. The original throughput of their TCP/IP implementation was 350Mbits per second on a single processor prototype Cray-2. They improved performance to 461Mbits/second (31.7%).

The Cray team did their initial tests over a 800Mbps point-to-point channel between a CRAY-2 and a CRAY-YMP located at NASA Ames Research Center. This verified the benefits of the implementing the "WINDOW SCALE" TCP option. Subsequent testing was done on a single processor Cray-2 prototype, using a loopback interface. The remaining measurement parameters are specified in table 15.

What is measured?	Quality Focus	Throughput
	Layer Focus	Transport
	Environment	800Mbit/second point-to-point channel
How is it measured?	Traffic Source	Generated load
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Outside protocol time
	Analysis	Online

Table 15: High Speed Networking at Cray Research

3.3 Application Performance

This group of papers comprises studies of systems and applications – software with which people directly interact. A distinguishing characteristic these papers is the desire to thoroughly understand the operation of the *whole* system. This desire leads to very goal oriented studies that use numerous experiments, each with its own focus(es), style, methodology, etc.

The aim of measurement usually performance, often at the top level, where an application makes a service available. Since a high degree of understanding is the goal, the measure and its variance must be explained. When a measurement is made with explicable variance levels, the breadth of focus is adequate. If the measurement variance in an experiment is higher than expected or inexplicable, the experiment is redesigned, where possible, to measure subparts of an application and identify and isolate the source of the variance. The overall measurement is then compared to the sum of the parts as a check. As a result of this looking deeper, the breadth of the layer focus of a study can increase. A study that was once tightly focused on one layer can end up spanning many additional layers.

All this is typically done in a “real-life” environment. Isolated networks are sometimes used as a comparative environment to control otherwise unexplained communications related variances in results, to determine whether or not these variances stem from network characteristics.

There is no predominant approach to measurement in this group. Measurements are carried out using a mixture of approaches – whatever works for the particular aspect of the system decomposition that the researcher expects will explain the system's behavior. Unlike the traffic profile and support protocol groups, hybrid styles are likely to show up here.

Typically systems builders have the source code for the system under study, making intrusive measurement techniques significantly easier. They also can, for intrusive measurements, quantify and correct or isolate induced overhead for individual experiments, by means of control experiments. e.g. Measuring an overall performance metric of both the intrusively instrumented system and the uninstrumented system.

Often measurement acquisition uses resources on the same machine as the application. The experimenter tries to be deft during acquisition – not using cycles that will be charged to the application. This minimizes the need for or difficulty of separating the artifacts of acquisition from the data. The degree of analysis and time depends on methods. Performance aspects are generally analyzed online while others either on or offline as dictated by resources.

3.3.1 Analysis of Transaction Management Performance

In "Analysis of Transaction Management Performance"[Duc89], Duchamp investigates the CAMELOT transaction facility. He examines the design and performance of the transaction manager component. Duchamp's performance measures are the transaction latency and throughput. In particular the author studies the performance effects of the inter-site communication, an optimization of 2 phase commit, non-blocking commit, and multithreaded design. The general measurement parameters appear in table 16.

Duchamp measures Camelot RPC latency and its components. The overall latency is 28.5ms per call. It is exactly accounted for by the measured underlying RPC mechanism (19.1ms) plus the contributions of contacting the underlying RPC mechanism (2×1.5ms) and the Camelot communications manager component (3.2ms per call at each site).

The author measures the latency of transactions as a function of the number of subordinates for three variants of two-phase commit protocol. For example, with two subordinate sites, the unoptimized write transaction takes 210ms with a variance of 33ms. In contrast the optimized version takes 162ms with a variance of 19ms.

A non-blocking commit protocol was measured on the same transaction as the two-phase commit test described above. Write transactions took 230ms with a variance of 67ms. The greater time for this protocol is due to an extra step needed to provide the non-blocking behavior.

Next Duchamp investigated the effects of using multiple threads in the transaction manager to improve performance when multiple transactions are active. He increased the number of client/server pairs running transactions until saturation when throughput ceased to increase. For both update and read only transactions, more threads meant equal or higher transaction rates (transactions per second) and delayed saturation. For example, with update transactions, 5 threads reached saturation at 2 client/server pairs executing 7.3 transactions per second and peaked at 3 pairs and 7.5 transactions per second. When run with 20 threads, the saturation and peak points coincided at 3 client/server pairs and 8.5 transactions per second.

3.3.2 Network Time Protocol

Together "Measured Performance of the Network Time Protocol in the Internet System (RFC-1128)" [Mil89b] and "Internet Time Synchronization: the Network Time Protocol (RFC-1129)" [Mil89a], both by Mills, show the role of experimentation in the development of a protocol, clients and servers, over several years (1985–1989).

What is measured?	Quality Focus	Application Throughput and Latency
	Layer Focus	Application but extended as necessary down to hardware
	Environment	4Mbit/second private token ring
How is it measured?	Traffic Source	Custom load generator
	Acquisition	Participate from above
	Intrusiveness	Effectively non-intrusive
How is it processed?	Acquisition	Timestamps obtained outside measured activities
	Analysis	Online, outside of measured activities

Table 16: Analysis of Transaction Management Performance

The first RFC concentrates on performance measurements time protocols. It describes several experiments. First Mills obtained a file of 112,370 internet hosts and interrogated them using three time distribution protocols – `ICMP Timestamp`, `TIME`, and `NTP`. Of those 107,799 addresses were reported reachable, 94,260 responded in some way and 20,758 returned some valid time indication. Some post-processing was done to remove other entries for hosts that gave `NTP` or `TIME` responses leaving a list of 8455 hosts. The breakdown was 3694 `Timestamp`, 7666 `TIME`, and 789 `NTP`. "...at least 30 percent of the hosts in all three protocols make some attempt to maintain accurate time to about 30 ms with `NTP`, a minute with `TIME` and a couple of minutes with `ICMP Timestamp`. Between this regime and the one-percent regime the accuracies deteriorate; however, in general, NTP hosts maintain time about a thousand times more accurate than either of the other two protocols."

Mills then proceeded to obtain more refined, long term results for the systems using `NTP` to determine their performance and error characteristics. Performance is defined in terms of the error between actual time and reported time. One of the results `NTP` provides is a delay measure for each interaction with a server. The offset vs. delay results provided a crucial piece of information for designing the filtering algorithm currently used. Mills observed that the offset was smallest for the lowest delay measurements and the bounds of the offset increased linearly with increasing delay. At a delay of approximately 0.2 seconds the offset was 0.0 seconds while at 0.4 seconds delay the offset ranged from -0.08 to 0.08 seconds.

Mills reports additional results in the accuracy of time standards distributed by HF band (3–30MHz) radio clocks which are affected by propagation path changes through the day and the stability of local oscillators phase locked to radio clocks.

The second paper, RFC-1129, complements the first, addressing more of the background, concepts, and evolution of the current version of NTP. It provides a lighter treatment of the measurements of RFC1128, placing those results in the context of the protocol development.

Mills's work in validating the operational properties of NTP has begun to move it into the domain of support protocols. For example, Lixia Zhang's *Virtual Clock Protocol*, [Zha91], is a transport protocol which has been proposed and simulated and is now being implemented. It uses clocking to distribute resources. The better the synchronization of the clocks, the lower its overhead and better its performance.

The measurements were acquired using systems running the Fuzzball operating system. The Fuzzball operating system was designed with timekeeping in mind and keeps time accurate to 1ms. The other measurement parameters are given in table 17.

What is measured?	Quality Focus	Distribution of time service results and performance of NTP
	Layer Focus	Application
	Environment	Internet
How is it measured?	Traffic Source	Active queries
	Acquisition	Client observation
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Modest, offline

Table 17: Network Time Protocol – RFC 1128, 1129

3.3.3 Replicated Distributed Data

In "Accessing Replicated Data in a Large Scale Distributed System"[GL91], Golding and Long use empirical network measurements to support algorithm design. They propose four algorithms for accessing distributed replicated data. To study their performance of the algorithms, the authors build a simulation model of two components. The first component is the candidate algorithms. The second component is the network model. We look at the measurement aspect of their results. The general measurement parameters of Golding and Long's work are given in table 18.

The network model needed two parameters that were best set from host availability and network latency. The authors had previously studied internet host availability based on host polling (see section 3.4.1). They used data from this previous work to obtain availability rates. They found that systems typically had availabilities of 93% to 97%. Long and Golding examined the hypothesis that failures to reach a host were independent and could be modeled by Bernoulli trials. They were able to reject this hypothesis because, looking at the lengths of failures (numbers of successive failures to reach a host), it predicted that failures of length=3 were 1.17% and length=11 were $< 10^{-8}$. In reality failures of length=3 were 1.80% and 2.70%.

The authors conducted a second set of experiments determining the latency distribution for each host polled periodically over a 48 hour period. They characterized their results by the mean latency. Latencies ranged from 24ms to 1600ms for hosts outside the local site.

The measurement results were used to set the site-availability and site-response-time network parameters for their simulation studies. The simulations were then repeated using a mathematical rather than empirical traffic model to simulate the network values. They found that, using an exponential network latency distribution, the results "were sufficiently close to those of [our] trace-based simulations ...."

What is measured?	Quality Focus	Distribution of responses and response times (latencies)
	Layer Focus	Transport
	Environment	Internet
How is it measured?	Traffic Source	Created
	Acquisition	Client
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 18: Accessing Replicated Data in a Large Scale Distributed System

3.3.4 Adaptive Remote Paging

"Adaptive Remote Paging for Mobile Computers"[SD91] by Schilit and Duchamp investigates the feasibility of using remote storage accessed via a radio network as backing store for a portable computer. To accomplish this, the work is split into two parts. The first part evaluates the performance aspects of the transport system. The second part evaluates demands placed on that transport system, using a prototype implementation. Table 19 lists the general measurement parameters.

The authors measured RPCs consisting of small requests and responses of 0, 2, 4, and 8Kbytes. They also measure the components of the RPC transactions and compare those with the overall results. For example, for 8Kbyte RPC's the measured time was 52ms. The independent measured components were 18.1ms bus transfer plus 7.2ms Ethernet plus 19ms TCP loopback plus 4ms Mach IPC, totalling 48.3ms. Similarly close agreements were present for 0, 2 and 4Kbyte RPCs.

Schilit and Duchamp measured paging activity by instrumenting the backing store manager to collect time histories (traces) of access behavior while running various test suites on the machines using the backing store manager. The two suites were editing & compiling and a complete OS recompilation. They analyzed these for object access patterns – object store usage vs. time and sizes vs. lifetimes. Their results show, for their applications, "there is a good-sized boot of large, long-lived objects: among the objects of above-median size, 75% live longer than 10 minutes(11% in the heavy load case)."

The authors created a paging benchmark to evaluate the performance of six forms of backing store. They executed random 1Kbyte reads and measured average time. The results ranged from 44ms for using local files, through 45ms to 50ms for using the Mach pager, remote adaptive pager (main memory) and local adaptive pager, and increased to 94ms for remote adaptive pager (disk) and 118ms for NFS files.

What is measured?	Quality Focus	Throughput over link & Distribution of paging
	Layer Focus	Transport & Application
	Environment	Laboratory RLAN
How is it measured?	Traffic Source	Introduced Traffic & Workload
	Acquisition	Client & Externally observed
	Intrusiveness	Non-intrusive & Intrusive
How is it processed?	Acquisition	Time online & Capture and transfer to storage
	Analysis	Online results & Offline

Table 19: Adaptive Remote Paging for Mobile Computers

3.4 Other Studies

This group of papers has two subgroups – those related because of some aspect of their work crossing empirical measurement of systems and those which are so broad that they do not neatly fall in any of the preceding three groups.

3.4.1 Estimated Host Reliability

In "Estimating the Reliability of Hosts Using the Internet"[LCP90], Long, Carroll and Park establish estimated mean-time-to-failure (MTTF) and mean-time-to-repair (MTTR) distributions for hosts on the Internet.

Typically an exponential distribution is used to model host up-times. The first observation was that if this is correct then sampling could derive MTTF and MTTR estimates. They first tested this hypothesis in an experiment capable of disproving an exponential uptime distribution. They observed that part of the results were exponential but that overall the test statistic disproved an exponential hypothesis. More careful review of the data to remove duplicates and wild outliers greatly reduced the value of the statistic computed leaving no conclusive evidence against an exponential distribution of up-times.

Assuming a roughly exponential uptime distribution, the authors sampled uptime distributions using SUN RPC `rstat`.[2] They first derived an estimate for all responding systems and for all Sun-3 and Sun-4 computers. Again the posts of probability of failure vs. uptime in days revealed an exponential like shape.

They further refine their results by using host specific characteristics to segregate polling data by hardware architecture and operating system type. This narrowed the confidence intervals on their results. For example, a Sun 4/490 had a MTTF of 15.07 days with a 95% confidence interval running from 10.58 to 19.55 days.

The authors computed availabilities for specific Sun models and systems from other vendors. As an example, a Sun 4/490 had a 0.928 availability with the 95% confidence

[2]Sun RPC is implemented for many other operating systems including those by DEC, NeXT and HP.

interval from 0.876 to 0.977.

Finally the author's MTTR, a function of their previous analyses of MTTF and availability. The value of MTTR for a Sun 4/490 was 1.17 days. They noted that "servers, such as the Sun 4/280, 4/390, 3/180 and 3/280 have a lower MTTR than the workstations."

For their first analysis, the authors identify a set of 1000 random internet hosts from an initial pool of over 350,000 hosts. Every 20 minutes for two days they polled these hosts and recorded the pattern of responses and failures. Two months later, after analysis of the first results, they repeated the work in more detail using 68,000 hosts. The other basic measurement parameters are given in table 20.

What is measured?	Quality Focus	Distribution of successful/unsuccessful host polls
	Layer Focus	N/A
	Environment	Internet
How is it measured?	Traffic Source	Polled
	Acquisition	Client
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 20: Estimating the Reliability of Hosts Using the Internet

3.4.2 Measurement of Organizational E-Mail

In "A Measurement Study of Organizational Properties in the Global Electronic Mail Community"[SW90], Schwartz and Wood use a traffic study methodology. They are interested in the high level communications patterns of people using electronic mail.

The authors have post processed the electronic-mail logs from 3700 organizations around the world. They looked at the communications patterns of approximately 50,000 people who sent a total of approximately 1.2 million messages among 183,000 pairs of people. From the data they formed graphs where the nodes represent people, hosts, or administrative domains (depending on the level of detail desired in the graph) and the edges represent communication between nodes. Table 21 gives the general measurement parameters.

Among their results they found that at the macroscopic level the graph was very sparse. The typical interconnect densities were on the order of 2×10^{-4} that of a fully connected graph. They also found that removing the host name part of the e-mail address summarized the graph edge count by a factor of 11.7. This reflected the workstation nature of the environments at those sites. One person could send mail from or receive mail at many equivalent machines. At the user level (most detailed) they found that although the diameter of the graph was 21, the average length between users was 5.96 and maximum 13.99.

Looking at the microstructure of the graph also produced interesting results. They examined outdegree of nodes categorizing nodes as "Loose Web" (0–10), "Main Population" (10–100) or "'Star' Nodes" (100–900). The activity (count of messages vs. outdegree) within each category was roughly exponentially distributed. They note that while the graph had essentially one component, if the "'Star' Nodes" and their edges were removed the graph now had 178 components. This indicated the major role of these nodes in the communications patterns.

What is measured?	Quality Focus	Access pattern
	Layer Focus	Application
	Environment	Internet
How is it measured?	Traffic Source	Existing Application Traffic
	Acquisition	Externally observed
	Intrusiveness	Non-intrusive
How is it processed?	Acquisition	Capture and transfer to storage
	Analysis	Offline

Table 21: A Measurement Study of Organizational Properties in the Global Electronic Mail Community

4 Directions of Future Work

Empirical measurement of network software serves many purposes, as outlined in section 1. Measurement is a means of detecting, locating and correcting problems performance problems. Measurement also has a role in the use of models. In both areas of empirical measurement of NSSs, the presence of a network as a system component adds a potential bottle-neck and source of variability.

Measurement studies are growing more subtle and refined. This is not to say that fundamental measurements are being ignored. Rather the community's experience in measurement allows basic studies to performed more easily or concurrently with the more advanced work.

The past decade has seen the wide-spread acceptance of local area network physical protocols including 3 and 10 megabit per second Ethernets and 4 megabit per second token rings. These set upper bounds on the measurement data rates. Higher level protocols employing these physical protocols may further limit the data rates.

While the LANs and to some extent protocols used on them remained fixed, the equipment available to investigators has improved and become less expensive. Hardware improvements include accurate memory-mapped clocks for timestamping, higher performance CPU's and network interfaces capable of handling all packets in realtime, and larger and faster main and disk memory for larger buffering capacity. All have enabled better measurements. This has enabled more complete, detailed or deeper studies with using general purpose computers for acquisition rather than building or obtaining specialized equipment.

The introduction of DS-3 links, HIPPI and FDDI may cause the next round of escalation. Once capable measurement systems setups may again be pressed for processing power and buffer space when processing every packet present on their attached networks.

4.1 Directions in Traffic Analysis

Given an adequately precise recording of the desired information by some combination of a promiscuous tap and a filter, the direction over the past 10 years has been toward greater complexity and subtleness in extracting the desired information. The desired information tends toward either input for models or confirmation/refutation of models that have been advanced and might be useful. [CDJM91], [LW91] and [Hei90] are instances of the front end of this trend.

4.1.1 Model Building

Models of traffic are improving too. Early studies seemed to focus on either the timing of packets ([JR86, Hei90, LW91]) or the characteristics of the packets ([CS88, Ć89, CDJM91] but not both. These two threads are converging towards model building measurements that address both aspects ([KLW90]). The most recent work is also taking increasing notice of longer term (day, week, month) trends ([Pax91]).

4.2 Directions in Support Performance

For support software such as RPC systems and other transport and lower level protocols, traffic generators, sinks and echoers will continue to hold sway. For both, concurrent passive observation (hybrid) has an often underutilized role but is growing ([CW87], [SB89] and [SD91]).

4.3 Directions in Application Performance

A bifurcation as to when the measurements are done. In older work in this group, measurements were done when the system came up for tuning and then more complete ones done for conference and journal presentations. This is changing. While people continue to do measurements late in the design and build cycle, Mills's ([Mil89b, Mil89a]) and Golding's ([GL91]) work show the growing trend of doing measurements earlier to provide bases or support previous assumptions in the development of new systems or algorithms.

References

[C89] Ramón Cáceres. Measurements of Wide Area Internet Traffic. Technical Report 89/550, University of California, Berkeley, December 1989. PROGRES Report No. 89.12.

[CDJM91] Ramón Cáceres, Peter B. Danzig, Sugih Jamin, and Danny J. Mitzel. Characteristics of Individual Application Conversations in TCP/IP Wide-Area Internetworks, march 1991.

[CS88] D.-M. Chiu and R. Sudama. A case study of DECnet applications and protocol performance. In *Proceedings of the 1988 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 47–55. ACM/SIGMETRICS, May 1988.

[CW87] D. Cheriton and C. Williamson. Network Measurement of the VMTP Request-Response Protocol in the V Distributed System. In *Proceedings of the 1987 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, May 1987.

[Duc89] Dan Duchamp. Analysis of Transaction Management Performance. In *Proceedings of the Twelfth Symposium on Operating Systems Principles*, Arizona, December 1989.

[GL91] R. Golding and Darrell D. E. Long. Accessing Replicated Data in a Large-Scale Distributed System. Technical Report UCSC-CRL-91-01, Concurrent Systems Laboratory, Baskin Center for Computer Engineering & Information Sciences, University of California, Santa Cruz, January 1991.

[Hei90] S.A. Heimlich. Traffic characterization of the NSFNET national backbone. In *Proceedings of the 1990 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 257–258. ACM/SIGMETRICS, May 1990.

[Jac88] Van Jacobson. Congestion Avoidance and Control. In *Proceedings of SIGCOMM 88*, pages 314–329, February 1988.

[Jac90] Van Jacobson. 4BSD Header Prediction. *ACM Computer Communication Review*, 20(2):13–15, April 1990.

[JR86] Raj Jain and Shawn A. Routhier. Packet Trains – Measurements and a New Model for Computer Network Traffic. *IEEE Journal on Selected Areas in Communications*, 4:986–995, September 1986.

[KLW90] K. M. Khalil, K. Q. Luc, and D. V. Wilson. LAN Traffic Analysis and Workload Characterization. In *Proceedings of the 15th Conference on*

Local Computer Networks, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264, September 1990. IEEE Computer Society Press.

[KP87] Phil Karn and Craig Partridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. In *Proceedings of SIGCOMM 87*. ACM, August 1987.

[Lam84] Butler W. Lampson. Hints for Computer System Design. *IEEE Software*, pages 11–28, Janury 1984.

[LCP90] D. D. E. Long, J. L. Carroll, and C. J. Park. A Study of the Reliability of Internet Sites. Technical Report UCSC-CRL-90-46, Computer & Information Sciences, U Cal. Santa Cruz, September 1990.

[LW91] Will E. Leland and Daniel V Wilson. High Time-Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection. In *INFOCOM '91*, 1991.

[Mil89a] D. L. Mills. Internet Time Synchronization: the Network Time Protocol. Technical Report 1129, University of Delaware, October 1989.

[Mil89b] D. L. Mills. Measured Performance of the Network Time Protocol in the Internet System. Technical Report 1128, University of Delaware, October 1989.

[N+91] A. Nicholson et al. High Speed Networking at Cray Research. *Computer Communication Review*, 21(1):99–110, January 1991.

[Pax91] Vern Paxson. Measurements and Models of Wide Area TCP Conversations. Technical Report LBL-30840, Comp. Sys. Dept, Lawrence Berkeley Lab., University of California, Berkeley, May 1991.

[SB89] Michael D. Schroeder and Michael Burrows. Performance of Firefly RPC. In *Proceedings of the Twelfth Symposium on Operating Systems Principles*, Arizona, December 1989.

[SCS+88] K. Seo, J. Crowcroft, P. Spilling, J. Laws, and J. Leddy. Distributed testing and measurement across the Atlantic packet satellite network (SATNET). In *Proceedings of the 1988 ACM/SIGCOMM Symposium*, pages 235–246. ACM/SIGCOMM, August 1988.

[SD91] Bill N. Schilit and Dan Duchamp. Adaptive Remote Paging for Mobile Computers. Technical Report CUCS-004-91, CS Department, Columbia University, February 1991.

[SH80] J. F. Shoch and J. A. Hupp. Measured Performance of an Ethernet Local Network. *Communications of ACM*, 23(12):711–721, December 1980.

[Svo89] Liba Svobodova. Measured Performance of Transport Service in LANs. *Computer Networks and ISDN Systems*, 18(1):31–45, November 1989.

[SW90] M. F. Schwartz and D. C. M. Wood. A Measurement Study of Organizational Properties in the Global Electronic Mail Community. Technical Report CU-CS-482-90, Department of Computer Science, University of Colorado, August 1990.

[Zha91] Lixia Zhang. VirtualClock: A New Traffic Control Algorithm for Packet-Switched Networks. *Transactions on Computer Systems*, 9(2):101–124, May 1991.

A Measurement Issues

A.1 What is measured?

Within this identification there are three aspects: the quality focus, the layer focus, and the environment. For this section we will use a running example to help explain what is meant by these three aspects. For our example we consider the measurement of the "best case average latency of remote procedure calls over an unloaded token ring between idle machines". The first piece, "best case average latency," is the quality focus. The second piece, "remote procedure calls," is the layer focus. The third piece, "an unloaded token ring between idle machines," constitutes the environment.

A.1.1 Quality Focus

Quality focus refers to which area of performance are being measured. In the running example the quality focus is "best case average latency". This can, in turn, be further refined into three elements that describe the focus of an experiment – the *basic quality*, the *conditions* and the *summarization*.

The *basic quality* is the characteristic for which one wants quantitative results. In the running example, "latency", is the desired quality the experimenter wishes to quantify. Other typical basic qualities. (e.g. latency, throughput, availability, reliability, and traffic distribution)

The *condition* specifies the circumstances under which the quality is desired. This can be, for example, normal, average, or worst case. The basic quantities can be highly impacted by the conditions. One might imagine an experiment measuring the latency of a communications protocol may show exceptional performance on an unloaded network but when that same network is heavily loaded it may suffer from timeouts and retransmissions.

The *summarization* is the way in which the results are reduced for presentation. To bound and reduce experimental error, most studies make numerous observations of a quality, during one or more intervals. They then employ descriptive statistics to condense their data, for example means and variances. In our example, the term "average" suggests that the summarization involves taking and reporting the mean of a number of observations.

A.1.2 Layer Focus

Networked software systems are often built in layers. The layer focus identifies which layer is most directly the target of investigation and to what extent other layers are studied. Layer focus can range from a narrow concentration on a single layer to broad interest in several layers and their interaction.

The ISO seven layer OSI model might appear to provide a handy organization but difficulties arise when trying to map between is and systems that were not built to the model. A loose ordering running from the data link level at the bottom to the final application at the top suffices to organize literature in the field.

In the running example, the layer focus in the "remote procedure call". Casting this in OSI terms, the focus is on the session or transport layer and the breadth is unstated.

A.1.3 Environment

The environment interacts with system under study. The interaction, in turn, affects the measurement results. Two sub-questions come up: what is the environment and to what extent may it be controlled. In the running example, the environment is "an unloaded token ring between idle machines." One can see that the environment is closely related to the measurement conditions discussed above.

We can identify two extremes: isolated laboratory settings, having dedicated resources, and uncontrolled setting in which machines and/or networks are shared. In the isolated setting, measurements give an upper bound on performance rather than revealing typical behavior. In the uncontrolled (live) setting best case values may be rare events and hard to isolate from raw data but general results may be more indicative of typical behavior.

A.2 How is the measurement made?

Measurements results are derived from observable events. One needs to identify these events and how to acquire them. What are they? How are they produced? How are they observed? More concretely, when one measures, is passive observation or active probing or exercising? If the approach is active, to what degree? Will their observation affect the system studied, perhaps requiring alterations?

It is possible to separate the active vs. passive question into two pieces – traffic production and involvement. Overall, both sets of questions can be organized into three areas: traffic, involvement and intrusiveness.

A.2.1 Traffic

Does the measurement rely solely on existing traffic or does it introduce traffic? Another way of looking at this is, what is the degree of involvement in data acquisition?

Using existing traffic raises concerns about its quantity and quality. Is there enough to perform the measurements in a reasonable amount of time and is it representative (normal or typical)?

Artificial traffic can supplement or replace existing traffic. An example of the latter is introducing an occasional transaction into a much larger transaction stream. An example of the latter is an inexhaustible source of transactions into a transaction processing system or packets into a transport protocol.

In the case of replacement, a model of the desired traffic is built and a generator presents that load to the system under study. That load can be anywhere from minimal load to a saturation load designed to stress or even overstress the system. Once the traffic model is established there is no worry about perturbing the system – the model is deliberately doing that. There is, of course, the need to ensure the validity of the traffic model.[3]

In the case of supplemental traffic, one inserts probes along with other, existing, traffic. If one is trying to measure the system without perturbing it greatly or only to a known amount, care is necessary.

A.2.2 Involvement

Involvement relates to the degree in which the experiment interacts with the system being studied. At one end of the scale is passive observation. At the other end, the experimental system interacts directly with the system under study and no passive observations are made.

A.2.3 Intrusiveness

Intrusiveness considers whether or not observations require modification of the measured system. If so then the methodology is considered intrusive. In either case, one must consider the side effects of measurement on the system.

Intrusive measurements generally require source code although it may be possible to graft measurement code into executables. An example is altering an operating system kernel to log packet transfers or system calls. Intrusive measurement perturbs execution and system behavior to some degree.

Non-intrusive measurements require one or more points of access. Typically this involves intercepting or observing information transfer events across layer boundaries or on a physical medium. Examples include using traffic sources and sinks outside a transport protocol being studied and capturing Ethernet packets.

A.3 How is processing handled?

There are two types of processing involved in empirical measurement, acquisition and analysis. The former is used to make observations. The latter is applied to reduce the observations to results.

A.3.1 Acquisition

A certain amount of processing capability is needed to acquire observations. If the acquisition is on the observed system, it competes with the observed system for processing power. Outside observation, such as packet tracers, avoids this contention. In either case, however, the processing power must be up to the task. If the peak or average data arrival rate exceeds the acquisition processing capacities raw observations may be lost. This leaves the measurements suspect.

[3] One area of measurement, benchmarking, is continually embroiled in this question.

Processing power needed varies with the measurement method and the load. One particular factor that can affect load is communications patterns. With shared communication media many pairs of communicating machines contribute to the traffic. If a single host is involved in most of these conversations, that host or the communications medium but not acquisition processing is usually the bottle-neck. If no one host dominates the medium and traffic is spread across many machines, each of which proceeds at its own rate, then it is possible for the aggregate traffic to exceed the acquisition capability.

If analysis, discussed below, has an online component, both compete for the processor but output is reduced. If analysis is totally offline then the acquisition processing also includes the overhead of recording the raw data stream.

A.3.2 Analysis

Analysis can be done online, at the time of the measurement or offline, after the measurement is complete, or some combination. Sometimes it is natural to combine acquisition and analysis processing, doing the analysis online because it can be performed within the time constraints presented by the arrival of new observations.

Offline processing is not a panacea. Because some processing has to be done for acquisition and more to write the raw data to the archival medium for later analysis, it too if vulnerable to overrun of its input, processing and output capacities.

Online processing, if done on a host involved in the communications, rather than a passive monitor, perturbs the system CPU usage but typically has less I/O processing. Offline processing requires the movement of larger amounts of data to some storage medium which also affects the host performance.

Typically offline processing is desirable because it records more information allowing new or multiple theories to be tested with one set of information. Online processing does not allow for new ideas – every test must be built in. Particularly where the signal to noise ratio or resulting data rate is low and experiments are conducted over longer periods of time, current work leans toward offline processing.