Internet Performance Modeling: The State of the Art at the Turn of the Century

Mark Crovella [a] Christoph Lindemann [b] Martin Reiser [c]

[a] *Boston University, Department of Computer Science, 111 Cummington St., Boston, MA 02115, USA, e-mail: crovella@cs.bu.edu*

[b] *University of Dortmund, Department of Computer Science, August-Schmidt-Str. 12, 44227 Dortmund, Germany, e-mail: cl@cs.uni-dortmund.de*

[c] *GMD, Institute for Media Communication, Schloß Birlinghoven, 53754 St. Augustin, Germany, e-mail: martin.reiser@gmd.de*

Abstract

Seemingly overnight, the Internet has gone from an academic experiment to a world-wide information matrix. Along the way, computer scientists have come to realize that understanding the performance of the Internet is a remarkably challenging and subtle problem. This challenge is all the more important because of the increasingly significant role the Internet has come to play in society. To take stock of the field of Internet Performance Modeling, the authors organized a workshop at Schloß Dagstuhl. This paper summarizes the results of discussions, both plenary and in small groups, that took place during the four-day workshop. It identifies successes, points to areas where more work is needed, and poses "Grand Challenges" for the performance evaluation community with respect to the Internet.

1 Workshop Organisation

The past ten years have seen researchers in performance evaluation begin to turn their attention toward the Internet as an object of study. As part of this process, in October 1999, 21 researchers met at Schoß Dagstuhl, Germany (http://www.dagstuhl.de) to examine the current state of affairs with respect to performance evaluation of the Internet. Our purpose was to take stock of the field of Internet Performance Modeling, and we focused on a core set of three questions:

- What has been achieved to date in modeling the Internet?
- What are the important research questions in the near and long term?

- Are there already any success stories (some real impact) from work to date in Internet performance modeling?

The authors were organizers of the Dagstuhl workshop. Workshop participants were: **Virgilio Almeida,** *Federal University of Minas Gerais, Brazil;* **Martin Arlitt,** *Hewlett-Packard Laboratories, USA;* **Paul Barford,** *Boston U., USA;* **Antonia Erni,** *ETHZ, Switzerland;* **Anja Feldmann,** *AT&T Research, USA;* **Günter Haring,** *U. Vienna, Austria;* **Boudewijn Haverkort,** *U. Aachen, Germany;* **Sugih Jamin,** *U. Michigan, USA;* **Zhen Liu,** *INRIA, France;* **Arnold Neidhardt,** *Telcordia Research, USA;* **David Nicol,** *Dartmouth College, USA;* **Kihong Park,** *Purdue U., USA:* **Vern Paxson,** *ACIRI, USA;* **Misha Rabinovich,** *AT&T Research, USA;* **Jennifer Rexford** *AT&T Research, USA;* **Mark Squillante,** *IBM T.J. Watson Labs, USA;* **Carey Williamson,** *U. Saskatchewan, Canada;* and **Walter Willinger** *AT&T Research, USA.*

The format of the workshop consisted of intensive small group discussions during which paper sections were developed, interspersed with meetings in plenary session to take stock and chart direction. As a result, workshop participants did not have the opportunity to contribute to all paper sections, and attribution is indicated under individual section headings.

Work in Internet performance modeling is diverse and touches on many related aspects of computer science. The organization of the area that emerged during the workshop was along three dimensions, and at two levels. The three dimensions of the problem are subfields of performance evaluation in general:

Measurement: Accurately capturing quantitative measures of the Internet and its activity, along with the related problems of workload modeling and characterization;

Modeling: The central activity of performance evaluation— building formal descriptions and simulations of the Internet, and the efficient use of such models to provide insight into expected behavior of the Internet; and

Control: Using the insights gained from measurement and modeling to effect better use of Internet resources, and more desirable system behavior.

While these three dimensions captured the process of Internet performance modeling, we found that such processes typically occur at two distinct levels:

Systems and Application: the level in which user and application actions are the objects of interest; and

Network: the level at which system components (hosts, links, and routers) are the objects of interest.

Clearly these two levels, and the three dimensions, do not unambiguously classify all of the issues in Internet performance modeling. However we found these

distinctions useful in organizing our discussion and in presenting our results, and the format of this paper reflects this taxonomy. Successive sections of this paper treat each of the three dimensions of the Internet performance evaluation problem in order. Within each section, discussion is separated according to the to levels.

In addition to this taxonomized review, workshop participants found it helpful to crystallize the state of the art by defining a number of "Grand Challenge" problems for Internet performance evaluation. These challenges are set out as problems whose solution would lead to a radical improvement in the state of our knowledge about the Internet, our ability to predict its performance, or our potential to improve its behavior. Thus, each section of this paper, corresponding to a dimension of the problem, concludes with a set of Grand Challenges.

2 Measurement Methods

The essential foundation of effective performance modeling is a comprehensive set of accurate, well-understood measurements of system inputs and properties. In this section we present an overview of the state of the art in Internet measurement.

2.1 System Level (Almeida, Arlitt, Haring, Liu, Squillante)

Building on previous work on network level measurement, a range of studies have considered the potential causes of the novel behavior found in Internet measurements. The goal of these studies was to get a better understanding and to help explain the corresponding system and workload characteristics. Some of the characteristics deemed to be important were: file size distributions, file popularity distribution, self similarity in Web traffic, reference of locality and user request patterns. A number of studies of different Web sites found file sizes to have heavy-tailed distributions and objects popularity to be Zipf-like in nature. Other studies also found that representations of reference locality in the Web were self-similar. These studies also confirmed that the cause of self-similar behavior in Web traffic was due to the superposition of many ON/OFF periods, which were found to follow heavy tailed distributions. Still other studies of different Web site environments demonstrated long-range dependency in the user request process, primarily resulting from strong correlations in the user requests. These patterns also exhibited both light tail and heavy tail marginal distributions, which appear to be influenced in part by geographical and cultural effects.

These and subsequent studies resulted in the development of tools that provide more accurate and representative measurements and models at the system level by incorporating the above characteristics and statistical distributions for workload characterization, workload generation, benchmarking, etc. Such research on workload characterization has led to a better understanding of Web server and system performance. Current and future challenges in this area include: tracking workload evolution such that the understanding of the workload follows the changes in technology, user behavior, and evolution of the Internet as a whole; obtaining representative data; insuring confidentiality when analyzing user behavior; performing analysis and characterization in a timely fashion to keep pace with the fast evolution of the Internet; and characterizing emerging and future Web-based applications such as e-commerce, teleconferencing, etc. Another advance required is proper data measurement to support workload characterizations (e.g., defining what is needed, and how it is to be collected).

2.2 Network Level (Barford, Feldmann, Jamin, Paxson, Williamson)

There are a number of goals that drive network measurement: performance evaluation and debugging of distributed systems; assessing network provisioning needs; adapting the use of resources to fit with current network capacities; measuring compliance with service agreements; and general scientific exploration of network behavior.

One class of properties to measure are those *intrinsic* to the network: for example, its topology, link capacities, and latencies. Another class reflects the current state of the network, such as queueing delays, link utilization, and routing dynamics. There are two basic approaches to making network measurements: *active* techniques inject additional traffic into the network and measure the resulting effects, while *passive* techniques monitor the existing state of the network, ideally in such a way that the traffic is completely unperturbed.

A number of measurement techniques have been developed to date. For active measurements, some examples are ping (measures Internet connectivity), traceroute (Internet path routing), TReno (bulk transfer capacity), and pathchar (Internet link characteristics). Tools for performing passive measurements are classified by the granularity of their measurements: per-packet, per-flow, router statistics (counters), router configuration information (e.g., forwarding tables), fault alarms, modem records (e.g., number dialed, bytes transferred), routing tables. Examples of passive measurement tools are the tcpdump packet filter, the mtrace tool for measuring multicast routing, the NetFlow router flow statistics, and HP OpenView (router counters and alarms).

Some measurement tools facilitate protocol analysis across multiple layers. For example, a Web header measurement tool might record network packets at the IP layer, reassemble the TCP data stream embedded in those packets, and then extract the individual HTTP header lines carried in that data stream. More general frameworks have been developed to support whole classes of such measurements—for example, Windmill (see below).

Some significant measurement success stories include the nearly-ubiquitous ping and traceroute tools (mentioned above), used in numerous studies; the high-resolution packet collector developed by Leland and Fowler; the Berkeley Packet Filter for high-speed in-kernel packet capture (used, for example, by tcpdump); the Network Probe Daemon framework used to measure Internet path properties between 37 sites; and the Multi-threaded Routing Toolkit used to instrument routers to gather measurements of routing information.

The state of the art has gone beyond the basic approaches outlined above by pushing the limits of what can be measured using just a network end system. For example, pathchar (mentioned above) analyzes subtle timing variations to extract hop-by-hop network properties nominally invisible to end-system measurement, and sting can extract packet loss information along a network path and accurately separate the loss into those occurring on the forward and reverse directions. Packet filtering has turned to the problem of passive high-speed packet capture. The OCxMON tool is an inexpensive stand-alone computer that can monitor very high speed optical links and extract either short term packet-level traces or longer term flow-level measurements. PacketScope supports long-term packet-level measurement by capturing the packet stream to a high-capacity tape drive directly attached to a workstation with sufficient power to anonymize the data in real-time to address privacy concerns. Finally, IDMAPS aims to provide an Internet "distance" service based on integrating widespread end system measurements.

A number of prototype measurement infrastructures have been developed and deployed, including: WAWM for Web measurement, Surveyor for active connectivity, delay and loss measurements for Internet2 and IEPM for similar measurements between scientific laboratories, ISMA for instrumenting Internet routing, and NIMI, which emphasizes a general notion of "measurement modules" (such as unicast and multicast delay, loss, path capacity and routing measurements) and a flexible access control model. Finally, software architectures have been developed to facilitate intricate synthesis of higher-level information from heterogeneous measurement sources: Windmill supports on-the-fly composition of measurements using plug-in modules, and NetScope supports post-measurement composition of multiple disparate data sets into a unified data model, including topology, routing information, and link-level measurements.

Looking at the state of the art in terms of capabilities rather than classes of measurement tools, we first consider the issue of measurement accuracy. For time resolution, GPS receivers offer extremely accurate timestamping of packet arrivals, but use of GPS is not yet ubiquitous in measurement studies, due to difficulties in siting antennae and the need to customize operating system kernels to actually take advantage of the highly accurate time source.

For measurement integrity, kernel-resident packet filters generally facilitate loss-free packet capture at 100 megabit/sec speeds on common workstation hardware (providing that the traffic volume is indeed being reduced by the filter), but speeds much faster than that require customized measurement platforms, even if the measurements only entail capturing packet headers. The general movement away from broadcast media and towards point-to-point media, and towards increasingly sophisticated layer-2 clouds, also complicate measurement efforts, requiring specialized hardware and software.

Finally, the state of the art in terms of large-scale measurement is currently on the order of a few dozen coordinated measurement points. The administrative difficulties with scaling measurement infrastructures up the next order of magnitude remain a challenging research problem.

2.3 Grand Challenges

2.3.1 Global Measurement Infrastructure (Barford, Feldmann, Jamin, Paxson, Williamson)

We now turn to a possible "Grand Challenge" problem as an exercise in thinking about where it might prove most fruitful to focus future measurement research. Historically, measurement of network systems has been done as an afterthought, rather than integrated into the system's basic design. The vision underlying the Global Measurement Infrastructure (GMI) is ubiquitous measurement capability. The infrastructure would consist of both active and passive measurement capabilities. Some infrastructure elements would be explicitly added to the Internet to support measurement; others would be embedded into the design of Internet components and protocols. While the heart of the GMI is a deployed, physical infrastructure, a key element of that infrastructure is an emphasis on APIs for fostering modularity, extensibility, and heterogeneity of data sources. For example, an API might be defined for controlling measurement elements, and another to structure how measurement results are retrieved or disseminated.

For something as large scale as the GMI to possibly succeed, it must incorporate coherence as a central architectural principle. That is, GMI components must be accessed in a consistent way; the GMI must accommodate new types

of measurement components seamlessly; and measurements produced by the GMI must be expressible in a uniform data model, such that different types of measurements produced by different sources can be correlated together into a cohesive whole.

The GMI must include certain forms of analysis. It must be capable of assessing the quality and soundness of its own measurements; characterizing the properties of its components, their interconnections, and their placement within the general Internet topology; and include the analysis necessary for aggregating measurements such that they reflect higher-level abstractions of Internet behavior.

The GMI would enable a number of central Internet measurement goals:

(1) Performance debugging of Internet systems
(2) Large-scale measurement of Internet properties such as topology or routing convergence
(3) Meaningful measurement in the face of the vast heterogeneity of the Internet
(4) Longitudinal studies of how the Internet evolves over time

There are, however, a number of very hard problems that must be solved in order to realize the GMI. Any form of publicly accessible measurement immediately raises difficult access control issues, which must be solved or the GMI is doomed. Another difficult problem is dealing with the immense diversity of the measurement components, in functionality, software versions, and the characteristics of the measurements they produce (for example, accuracy, granularity, time scale, and spatial scale). In addition, it appears likely that the scale of the GMI will mandate that the GMI components be capable of self-configuration and adaptive measurement based on their earlier measurements. A related problem is simply trying to understand the scale of the GMI: how to know when it comprises a sufficient set of elements such that it is capable of performing a particular measurement to a degree that is representative of Internet behavior as a whole. Finally, the effectiveness of the GMI will be greatly enhanced if new Internet protocols and systems are explicitly designed to contribute to the GMI.

3 Models and Solution Techniques

Given the availability of necessary measurements, the core task of performance modeling is the construction of appropriate models (formalisms and simulations) and the pursuit of their solutions to gain insight into system behavior. In this section we examine the issues in developing models for the Internet.

3.1 System Level (Almeida, Arlitt, Haring, Liu, Squillante)

Formal models and methods can play an important role in the analysis of Internet workloads and performance at the system and application level. This includes traffic modeling, workload and performance forecasting, profiling customer behavior, capacity planning and transforming workload representations. Various formalisms and methods are available, such as graph models, Petri nets, formal grammars, queueing networks, fluid models, stochastic optimization and control techniques.

Some of the basic and elementary methods in these areas cannot directly incorporate some of the key aspects and complexities of the Internet. Hence, to be successful in the analysis of Internet workloads and performance at the system and application level, we need methods to map fundamental Internet workload and performance problems to a sufficiently representative abstraction that is amenable to formal analysis. We also need methods to map the corresponding solution from formal models and methods to a solution for the original Internet problems. This further requires the exploitation and development of sophisticated mathematical methods to solve these formal abstractions of fundamental Internet problems at the system and application level. In some cases, it will be necessary to go back to first principles so that the analysis and solution sufficiently reflect the key aspects of the real Internet problem. Recent examples include the mathematical analysis of queueing models under heavy-tailed distributions with correlations, and the impact of these effects on key characteristics of the user request response time distribution. To be most successful, the analysis of fundamental Internet problems requires close collaborations across multiple disciplines. This is one of the biggest challenges in the area of system performance modeling and analysis for the Web. These formal models and methods can also support and provide a foundation for the development of optimal resource management and control, as well as pricing strategies.

3.2 Network Level (Haverkort, Jamin, Neidhardt, Nicol, Willinger)

The activity of low-level network modeling and analysis has enjoyed quite a number of notable successes already. The enthusiasm with which packet networks were deployed was based in no small part on the fact that the models predicted adequate performance. Modeling of CSMA protocols (*e.g.,* Ethernet) provided the understanding of phenomena and solutions (*e.g.,* collisions and exponential back-off) pre-requisite to their wide-spread adoption.

More recently, simulations of data networks have revealed emergent behavior,

such as ACK compression and the importance of phase effects or inadvertent synchronization. Likewise, mathematical modeling has provided a perspective from which to understand self similarity and long-range dependence.

The adoption of an empirical or phenomenological approach to modeling has allowed practical engineering to proceed with some confidence, though at the price of lacking the greater confidence that deeper models would provide.

Despite these successes, there are a number of current concerns with network-level modeling, which we summarize in the rest of this section.

Difficulties in modeling Internet performance. Here we use "demand" to refer to what the user really wants from the network, as opposed to "offered load" which refers to what measurements can be made directly. Demand and service quality are both masked. Naively, one might imagine that performance estimation would have the simple outline of calculating estimates of service quality from assumptions about demand and network design, where, at least in principle and after the fact, both the assumptions and the service quality can be checked by measurements. Instead, neither check is direct. First, concerning service quality, the reactions of end-to-end protocols can change the symptoms of congestion without relieving the damage to the service. For instance, the reactions might ensure the loss of fewer packets while degrading the service with a smaller rate of packet transmission. Without the help of a model that includes these reactions, however, measurements within the network cannot distinguish this kind of service degradation from a lack of demand. Thus, service quality cannot be defined simply in terms of quantities that can be measured within the network. Second, concerning assumptions about demand, measurements at a network bottleneck can give traffic traces, at least in principle, but this observed traffic is typically not the traffic that would have been generated by the users if the network had not been so congested. For instance, the observed traffic could have been modulated both by explicitly advertised reactions built into end-to-end protocols and by the implicit reactions of users abandoning sessions after painful delays. This potential for modulation has two implications for constructing a demand model from traffic observations. First, the observed traffic is simply not the pattern of traffic arrivals that customers would generate if their demands were satisfied. Second, demand is not simply a hypothetical arrival process, but is a more elaborate structure that includes a strategy for reacting to congestion. Because demand is an elaborate structure, it is not something that has an obvious definition in terms of quantities that can be measured within the network. Thus, instead of just one calculation of performance from models, one must do two others: one to infer a source model from raw measurements, and another to relate service quality to the network's raw measurements of performance indicators.

Temporal vs spatial vs across-layers. To date, the majority of work concerned with characterizing and modeling the dynamics of actual Internet traffic has focused on its temporal nature at the packet level. However, to fully understand the dynamics of the Internet and of the traffic that it carries, it is necessary to study its behavior in time, space, and across the different layers in the networking hierarchy. For example, to gain a basic understanding of routing behaviors, or to effectively and efficiently design techniques for the placement of Web proxies, it is crucial to know the underlying topology of the network. Similarly, to come up with realistic workload characterizations, it is necessary to understand and capture the essence of the subtle interactions that exist in today's Internet and are due to the feedback and control mechanisms acting at and interacting with the different layers in the networking hierarchy. At the same time, when trying to provide a physical explanation for a recently observed temporal characteristic of measured Internet traffic, namely, its multi-fractal scaling behavior over small time scales, there exists empirical evidence that a basic understanding of certain characteristics of the underlying network topology will play a key role. While the discovery and characterization of network topologies has already become a very active area of research, much future work can be expected in the areas of characterizing, modeling and analyzing the spatio-temporal-multi-layer dynamics of Interent traffic.

Abstraction. The "performance of the Internet" cannot be modeled nor analyzed in a general sense, even if we would like to do so. First of all, "the performance of the Internet" is a ill-defined notion in itself. But even if we concretize its notion, then still, depending on the performance question at hand ("what do I want to learn from the performance evaluation?") large parts of the Internet need to be abstracted in order to come up with performance predictions at all, be they derived using simulation or analytic/numerical techniques. Hence, there is no single way to approach Internet performance modeling, *i.e.,* there is no single "correct" level of abstraction; we need to have several levels of abstraction.

The main problem (and challenge) is to find the right level of abstraction for the performance modeling issue at hand. If the abstraction is "right" it covers these system (Internet) aspects that are important for the objective of the study, and leaves out any other (complicating but now- not-relevant) issues. The key issue then is validation: is the chosen abstraction indeed valid, in the sense that it does not reduce the system or model to a too trivial stage, so as to become useless? Validation of a model "against the Internet", *e.g.,* to verify the correctness of the assumptions made or to verify the insensitivity of the model for changes in some not-explicitly modeled system parameters, is an open and yet unresolved issue.

Similar to finding the right level of abstraction for the system model, one should find the right level of abstraction (modeling detail) for the users of the Internet, as they constitute the workload to be handled by the system. As an example of this, should we model single or aggregate users, should we explicitly model sessions, flows or even individual packets?. It is questionable whether traditional "static" workload models do suffice; indeed, we might need to consider more explicitly the interplay between the network dynamics and its users (*e.g.,* the response times perceived by the Internet users has their impact on their future behavior and vice versa).

3.3 Grand Challenges

The issues in current network modeling suggest a number of "Grand Challenge" problems, whose solution would represent significant advances in low-level Internet modeling.

3.3.1 Multi-Layer Workload Characterization (Almeida, Crovella, Haring)

Characterization can be accomplished at many levels: user level, application level, protocol level, and network level. However the relationships between these levels are currently not well understood. Mechanisms that transform characterizations from one level to another are lacking. The presence of network feedback to the user complicates this problem.

Multi-layer characterization allows insight into how higher levels (user or application behavior) affect network behavior. In addition it enables engineering prediction of the impact of new applications or systems. Multi-layer characterization provides a method for predicting the impact of design changes to the network infrastructure. Such studies will diffuse knowledge of the implications of characterizations into industrial practice.

One challenge for achieving a solution is a consistent framework that transforms characterizations from higher level to lower level, or vice versa. For example, the use of consistent formalisms across layers would be helpful. It appears to be easier to go from the upper layers to the lower layers because of the causality relationship, but going from lower layers to upper layers is useful as well and represents an important challenge. Workload characterization at the user level should consider the system behavior. For instance, models should be able to represent the changes in user behavior according to the various levels of system performance.

3.3.2 Internet-Equivalent Workloads For Live Experimentation and Benchmarking (Almeida, Crovella, Haring)

Currently, testing a server, client, proxy, or other device/system under laboratory workloads that mimic conditions in the actual Internet is not practical. Generation of proper delays on a per-flow basis, loads on simulated machines, and scaling up of accurate load generation to thousands or tens of thousands of flows is not feasible.

The development of Internet-equivalent workloads would provide the ability to engineer better systems. It would allow for test system modifications to be done in a controlled environment without disturbing real systems. Furthermore, it would allow for more accurate benchmarking of systems.

To achieve this capability, an understanding of how workloads are affected by spatial location/local infrastructure, cutural behavior, orgarnizational role, time of day/week, and usage profile is needed. For example, the distribution of requests across content providers varies with cultural setting and with time of day/week, and with the interactions between the two as well. Creating a concise description or model of how workloads are dependent on these issues is a challenge. Engineering issues surrounding efficient, accurate creation of tens of thousands of flows must be solved, and should be incorporated in a general workloads generating system.

3.3.3 Access Technologies (Almeida, Crovella, Haring)

Ubiquitous computing (*e.g.,* PDAs and appliances attached to the Internet) will add at least an order of magnitude to the number of traffic sources, and will change the characteristics of traffic sources. In addition, local access technologies like Universal Mobile Telecommunications System (UMTS), cable modems, Digital Subscriber Lines (DSL), and Local Multi-Point Distribution Services (LMDS), will influence workloads arriving from traditional end systems. Understanding the impact of these changes, in terms of workload characteristics and system performance, is challenging. Modeling the aggregated traffic from millions of such sources is another problem that must be solved. Since important network characteristics already arise from the interaction of large numbers of components, additional scaling has the potential to introduce new performance phenomena. Traffic patterns may shift in intensity and characteristics, and may become more local, changing network loading characteristics.

Solution of this problem will allow us to solve problems in advance of deployment, anticipate problems with network performance. ISPs, network providers, network equipment designers, will need to understand impacts of new workloads to properly design and operate networks.

The first step is to understand the traffic generated by new IP-connected devices individually. More challenging is the characterization of aggregate traffic resulting from these new devices. Understanding of how local bandwidth improvements will change user behavior and application behavior. Tractable models (either analytic or simulation) capable of representing millions of components are needed. An additional dimension that complicates mobile workload characterization and system performance is the dependence of the workload on physical location.

To bridge the gap between theory and practice mature modeling procedures have to be incorporated in appropriate software packages. The interface/front-end system of these tools has to be adequate for supporting the solution of real Internet engineering problems.

3.3.4 Methodologies (Haverkort, Jamin, Nicol, Park, Williamson, Willinger)

We also see "Grand Challenges" in the methodology use to model and analyze the Internet. The challenges arise principally from the fact that the Internet is a large-scale complex system. This facet has enormous methodological impact. Mathematical, statistical, and simulation techniques in common use today to analyze computer and communication systems frequently lose their applicability when applied to the Internet. Techniques used in other mathematical disciplines may find application; the challenge is to make Internet modeling a multi-disciplinary activity, so as to bring in other view-points and a larger aggregate knowledge of potentially applicable methods and techniques.

Developing the methodology to deal with emergent behaviors (behaviors induced by large scale in systems) is a challenge. Problems exist in developing the models (mathematical and simulation) that give rise to it and problems exist in recognizing it. Important from a methodological perspective is the challenge to follow through: it is not enough to recognize emergent behavior when it exists, we can and must push through to understanding its causes. From a methodological perspective it is important that the modeling framework used be able to explain it, not just create it.

Thorough validation of modeling is another challenge. Problems exist here at many levels, from formal validation of protocol correctness, to development of simulation frameworks capable of empirical behavioral validation of protocols, to validation of mathematical models with simulation models, and—critically—validation using the Internet itself. Relevance of modeling work is best demonstrated when proven in the harsh light of reality.

4 Resource Management and Control

The goal of effective performance modeling is informed development of better resource manangement and system controls. In this section we discuss outstanding issues of manangement and control in the Internet.

4.1 System Level (Erni, Lindemann, Rabinovich, Reiser)

The main application of the Internet is the World Wide Web. The Web's large scale and increasingly important role in society make it an important object of study and evaluation. The Web is useful for as a means of publishing and delivering information in a wide variety of forms such as text, graphics, audio, and video. Web applications are ranging from surfing for online information and product catalogues, e-commerce, teleteaching/telelearning, to video conferencing.

To effectively implement these Web applications, lower-level services called intermediaries are required which may constitute applications in their own. Examples of the intermediaries are agents like shopbot, bargainfinder etc. as well as search engines such as AltaVista and MetaCrawler. Beside the search and indexing software, the main part of these search engines is a Web crawler. Today's crawlers successively visit Web pages without taking into account their popularity. Recently developed search engines such as Google and Clever contain crawlers that exploit the hyperlink structure of the Web for deriving popularity measures of pages. As a consequence, the search engines can answer most queries satisfactory even if their database contains only a very small portion of the more than 350 millions Web pages. Selective Web crawling based on document popularity can be employed not only in search engines, but also in determining documents for prefetching by proxy caches.

Today's internet heavily relies on caching and replication to reduce response time as the load of servers and network. For this purpose extensive hierarchies of proxy caches have been deployed. Several protocols for inter-cache cooperation have been developed and successfully employed in heterogeneous systems comprising Web proxies from multiple vendors. The intercache protocol, ICP, originally used in Harvest and Squid has been standardized and is widely used. The advent of transparent proxies has allowed the user to benefit from proxy caching without being aware of their existence. Therefore, network providers can include proxy caching as part of the network infrastructure.

In addition to caching, Web site replication has emerged as important technology for providing scalability. In early days of the Web, the burden was placed upon users to select between different mirror sites. Similar to the advances in

caching, transparent technologies for replication have emerged. State-of-the-art algorithms for replica selection include a variety of metrics such as network proximity, server utilization, and availability. The transparent redirection of requests to site replicas can be performed at several levels. At the level of the domain name service, DNS, the DNS server chooses a server replica for processing a given request by returning its IP address. At the IP level, clients are given the IP address of a switch serving as multiplexor to forward a given request to one of the server replicas. Web replication can become especially important for Web hosting service providers, since it will allow them to efficiently manage their resources by dynamic replication and migration.

The original Web-server of the mid 90's stored and retrieved static HTML documents. The incredible fast growth of the Web and the availability of methods for generating pages dynamically caused the need for integrating databases into Web servers. Besides storing and managing Web documents, such an integrated database allows the deployment of already existing information system technology in the Web. Databases can either be accessed by Common Gateway Interface, CGI, scripts, Java servlets or by including code into HTML pages. Recently, CGI scripts are being increasingly replaced by Java servlets because of performance and portability reasons.

HTTP was conceived as a "state-less" protocol. Its merit is simplicity, however, achieved at the expense of performance. In HTTP 1.0, a new TCP connection is established and released for each transfer of a HTML page and the embedded inlines with obvious overhead for connection establishment and teardown. In addition, slow-start commences anew each time, preventing the window to achieve a satisfactory size. The problem is alleviated in HTTP 1.1 where the TCP connection is kept for a certain time-window. Efforts to further improve HTTP continue in the project HTTP-NG of the W3C.

4.2 Network Level (Park, Rexford, Williamson)

The Internet is governed by a collection of control algorithms and mechanisms spanning from MAC protocols at the link layer to user-level controls at the application layer. The end-to-end paradigm, which espouses a simple internal network design at the expense of more complex end systems, has been immensely successful at facilitating high-level network services, as evidenced by their widespread use and by the exponential growth of the Internet. In particular, the three protocols—TCP at the transport layer, IP at the network layer, and CSMA/CD at the data link layer—together have shouldered the bulk of the network control responsibility. Early work on queueing theory influenced the use of packet-switching as an alternative to traditional circuit-switching. In the late 1980s, TCP was extended to perform adaptive congestion control.

The design of congestion avoidance and related mechanisms drew heavily on physical insights into system dynamics and control, as well as simulation and empirical evaluation. The resulting mechanisms have proven to be robust. On the quality of service (QoS) front, the works on generalized processor sharing (GPS) have provided much insight into the design of scheduling and queueing disciplines for packet-switched networks. The scheduling aspect of GPS made two key contributions: establishing the (currently) best-known theoretical bound for worst-case end-to-end packet transfer delay across a multi-hop internetwork; and providing insight into the complexity of the mechanisms required for effective management of delay-sensitive traffic. These ideas continue to influence the design and analysis of QoS mechanisms for the next-generation Internet, including recent work on differentiated services.

In spite of these successes, the Internet faces new challenges brought about, in part, by these very successes. The demands of the emerging networked information society—the volume, the range of QoS requirements, and their commercial viability—are beginning to strain the current best-effort Internet.

Scalable QoS. Two realities bring pause to continuing simply with the tried-and-tested control strategy of throwing bandwidth at the TCP/IP/switched Ethernet support substrate: system size and heterogeneous payload. The transport of data, image, voice, audio, and video over the same network, and their use as components in a commercial environment, has resulted in QoS demands for which the best-effort Internet was not specifically designed. Aggravating the problem is the explosion of system size; when coupled with QoS requirements, it is not at all evident how these new services can be effectively realized over the existing control infrastructure. Turning all services into guaranteed services via the use of admission control at the network edges and weighted fair queueing routers per hop is infeasible due to inefficiency and resulting prohibitive cost. Per-flow control inside the network for potentially millions of concurrent users employing guaranteed service mechanisms puts a significant computational burden on routers, increases signalling complexity and overhead, amplifies concerns about reliability and fault-tolerance due to statefulness, and can result in poor resource utilization due to the self-similar burstiness of network traffic. The migration to aggregate-flow control for all but the most stringent classes of services addresses the mechanism aspect of scalability, but it is only a first step toward solving the "how" or algorithmic aspect of the QoS provisioning problem. How much complexity to incorporate into the network, how much to push to the edges, what components to perform in a closed-loop vs. open-loop fashion—delay-bandwidth product being the Achilles' heel of reactive controls—are some of the design decisions that will influence the next generation Internet control structure. Solutions that are not scalable are of limited utility.

Time Scale. A second, but perhaps more subtle, network control challenge is time scale. Different protocols—often because of layering, but not necessarily so—act at time scales orders of magnitude apart. Reactive controls are bounded by the round-trip time (RTT) or length of the feedback loop, but even open-loop proactive controls are affected by the length of the control path when allocating resources on behalf of flows that are in the millisecond to second range. At the two extremes are resource capacity dimensioning and network management tasks that act at the time scale of minutes, hours, or longer, and MAC protocols that operate in the microsecond range. Pricing—dynamic and static—is envisioned as yet another dimension in the control plane, with congestion-dependent pricing expected to operate across a range of time scales. The relationship between protocols spanning several time scales is often not well understood, and potential performance gains due to selective coupling and cooperativeness are not fully exploited. The superposition of protocols across layers and time can result in unexpected consequences, including stability problems for multi-layered feedback controls. The need for time scale sensitivity in network protocol design is also prompted by traffic characteristics, especially for self-similar traffic, where burstiness persists across a wide range of time scales. Reactive protocols have been traditionally designed to operate at the time scale of the feedback loop only, being impervious to information such as correlation structure in network state at significantly larger time scales which may be exploitable for control purposes. The recently advanced framework of multiple-time-scale congestion control is an avenue of research that tries to address this problem. Also of relevance are facts from workload characterization that show that most connections are short-lived whereas the bulk of data traffic volume is contributed by relatively few long-lived connections. Protocol design that is sensitive to the probable duration of connections is yet another dimension that is in its infancy of exploration.

Integrated Control. A third aspect of network control of growing importance is the integration of various control activities, such as routing, congestion control, admission control, end system scheduling, and network security. The provisioning of a target end-to-end QoS can be accomplished in a number of alternative ways (*e.g.,* by putting more responsibility on QoS routing to find a desirable path, or assigning higher priority to a flow on a crowded, less desirable path). The relative cost and trade-off between different control actions can have direct bearing on the efficiency of resource allocations. Real-time CPU scheduling at end systems is needed to provide sufficient processor time to end-to-end protocols and relevant application processes, and if security services are requested of the network, their computational overhead must be reflected in the overall resource provisioning decisions. As an extreme example, denial-of-service attacks that can significantly disrupt network services may have an impact that matches or exceeds that of inefficient network control. The correct and efficient functioning of each subsystem is a necessary but

not sufficient condition to achieving effective control that marshals network resources to achieve a common goal. Middleware in the form of Web caching, multicast management, and a host of other proposed functionalities can exert a significant impact on network performance, and an active incorporation of their capabilities in the overall network control process is desirable. The integration of wireline and wireless services, including mobility concerns, is yet another feature that engages a spectrum of control functionalities requiring intimate cooperation.

In addition to these challenges to the traditional best-effort Internet, the properties of IP traffic and workloads, network protocols, and communication technology have important implications for resource control mechanisms. Following are traffic control dimensions—new and old—that are expected to influence future protocol design.

Fractal traffic. Traffic streams in the Internet exhibit high variability across a wide range of time scales due to user workload patterns and their collective manifestation as self-similar burstiness at multiplexing points in the network. Interaction with feedback congestion controls can cause further fragmentation at lower time scales admitting a multifractal characterization of its overall properties. Scale-invariant burstiness implies concentrated periods of over-utilization and under-utilization of network links, and long-range dependence leads to slow decay of queue-length distribution, which can translate to amplified queueing delays in packet-switched networks. As a result, adding more buffer space to routers offers limited returns, and link bandwidth must be over-provisioned to deliver high (and predictable) performance. In addition, variability of link utilization complicates feedback congestion controls, measurement-based admission controls, and dynamic routing which depend on accurate estimates of network state. On the flip side, long-range dependence—i.e., the presence of nontrivial correlation structure at "large" time scales—implies a measure of predictability which may be exploitable for traffic control purposes. An example of this is multiple-time-scale congestion control, which has been applied to existing protocols such as TCP and rate-based congestion controls, yielding significant performance gains.

Heavy-tails. The high variability of aggregate traffic arises principally from the diversity of transfer sizes and connection durations in the Internet. Although most TCP/UDP connections are very short (on the order of a few packets), most of the packets belong to long-lived flows. These properties persist across several levels of aggregation, from individual TCP and UDP flows to traffic between a pair of hosts or subnets. The large number of short flows has important implications for network congestion. In the case of TCP, short transfers typically engage only the slow-start feature whereas long transfers lead to a "steady-state" during the congestion avoidance phase. Nonetheless, a collection of these flows can contribute a large number of packets in a short

period of time, particularly when these flows arrive in a bursty manner (as is common in HTTP/1.0). Fortunately, the heavy-tailed flow-size distribution facilitates new control mechanisms that focus their attention on the small number of long-lived flows that carry the majority of the traffic. Long-lived flows admit to effective identification (i.e., on-line classification) and this property has been exploited to reduce signaling overhead in IP-over-ATM networks, reduce the work involved in load balancing in Web servers, increase the stability of load-sensitive routing, and improve congestion control. Heavy-tailed distributions, though intimately tied to the physical causality of self-similar traffic, are of interest in themselves.

High delay-bandwidth product. The rapid increase in link capacity over the past few years has no impact on propagation delays, which are bounded by the speed-of-light. This is problematic for control of short transfers, where latency is dominated by propagation delay and some form of open-loop control is unavoidable. More importantly, high propagation delay limits the effectiveness of reactive controls which rely on feedback to modulate traffic control actions. Outdated feedback information, due to increased latencies, can result in untimely actions that adversely impact the in-flight traffic in the pipe. This typically manifests itself as under-utilization or over-utilization of network resources. Consequently, proactive recovery schemes, such as packet-level forward error correction, become attractive alternatives. In general, open-loop control schemes that reserve resources are needed to assure stringent levels of protection and quality of service. In tandem, traffic policing and shaping, due to their smoothing and bounding effect on input traffic, can (to some extent) alleviate the burstiness problem. The latter, however, is restricted to short-range variations—which, of course, can significantly impact performance—as self-similar burstiness is an aggregate phenomenon that is largely unaffected by detailed on-goings at smaller time scales. An approach to reactive traffic control that can mitigate the delay-bandwidth product problem is multiple-time-scale traffic control which, by exploiting correlation structure and predictability at time scales exceeding the RTT, can limit the outdatedness of feedback information.

Hop-by-hop vs. end-to-end control. The Internet operates in a decentralized manner, with hop-by-hop buffering, scheduling, and routing interacting with end-to-end congestion control and application adaptation that operate at the time scale of round-trip times (100-500 msec). The interplay between hop-by-hop and end-to-end controls preclude studying either in isolation. For example, the appropriate buffer sizes and discard policies depend on TCP congestion control, as well as on the number of active flows and their round-trip times. In addition, since the number of flows and their round-trip times vary with time, it is difficult to select the appropriate parameters for the hop-by-hop policies. The interactions are complex, and are difficult to study analytically. In addition, an individual router may not have access to all of the necessary

information (e.g., round-trip times) to adapt the control parameters locally, even if appropriate rules for setting these parameters were known.

Multiple protocol levels. The Internet operates with multiple layers of protocols. For example, Web transfers involve HTTP sessions that run on top of TCP, which in turn controls the transmission of individual packets. Similarly, the Border Gateway Protocol (BGP) for inter-domain routing uses TCP to exchange reachability information between neighboring domains. Likewise, TCP often runs over ATM or wireless networks, with diverse performance properties. The complex layering of multiple protocols often degrades performance in subtle ways. For example, TCP interprets a packet loss as a congestion indication, even though wireless networks may experience loss due to transient noise or fading. Similarly, rate-based feedback control in an ATM network (e.g., in Available Bit Rate service) interacts with TCP congestion control. Unraveling the interactions between layers requires a detailed understanding of each of the protocols, and results in complex performance models with a large number of parameters.

Although the Internet owes much of its success to its decentralized, heterogeneous, and stateless nature, these very properties introduce significant complexity in network control. Understanding, and perhaps exploiting, the interaction between IP workloads and control mechanisms is crucial to evolving the Internet into a more stable, robust, and efficient network.

4.3 Grand Challenges

4.3.1 Scalable Multilevel Network Control (Lindemann, Neidhardt, Rexford, Williamson)

IP networks are controlled by a variety of resource allocation techniques that operate on different time scales. For example, buffer management and link scheduling typically operate at a fine time scale, congestion control operates at the a medium timescale of round-trip times, and routing and admission control operate at a coarser time scale. These parameters driving these mechanisms are typically set as part of network provisioning and operations at a much larger time scale, ranging from hours and days to weeks and months. For example, routers are configured with parameters for buffer management (*e.g.,* buffer sizes, Random-Early-Detection probabilities), link scheduling (*e.g.,* weighted fair queuing weights), and routing (*e.g.,* link weights for Open Shortest-Path First routing, Border Gateway Protocol policies). The parameters do not change very often, and are typically set in isolation. Furthermore, the design of the network topology traditionally occurs at a very coarse timescale, though recent research has resulted in new technologies for

topology reconfiguration (e.g., wavelength assignment in Wave Division Multiplexing). The large time gap between network-level control and operational control in today's IP networks can result in inefficient use of network resources, and degraded user-perceived performance.

Narrowing the gap between network-level and operational control, and deriving scalable and robust control mechanisms, requires solving a number of challenging problems in network monitoring and modeling, including:

- Multiple control loops: Understanding the interaction between different control loops and protocol levels is crucial to efficient usage of network resources. Better models that capture these interactions would facilitate robust control. These models are crucial to determining which sets of control mechanisms should be applied together, and in what way. For example, the interaction between pricing policies and network provisioning, or network routing and topology reconfiguration, is not well understood.
- Network models: Online monitoring and control requires effective models of the workload and topology, and a clear formulation of the resource optimization problems. These problems are likely to be very complicated, and require new techniques for finding effective and robust solutions. For real-time changes to network configuration and policies, these techniques must be computationally efficient.
- Networking monitoring: Online network operations requires efficient systems for acquiring configuration and usage data, to feed the topology and workload models. Similarly, online operations require efficient systems for reconfiguring network elements with new control parameters. The systems (and their underlying protocols) must scale to large networks and high link speeds.

4.3.2 Optimized Responsiveness and Resource Consumption (Arlitt, Erni, Rabinovich, Reiser)

Bridging the time scale between local computing and remote access of information promises to enhance the responsiveness of the Internet. Not only would this latency hiding improve user satisfaction but it would also enable new applications involving multiple accesses of highly distributed information sources and compute services.

A large amount of redundant data is sent over the Internet that consumes bandwidth and puts load on routers and servers. Examples are repeated DNS queries and HTTP requests. Handling this traffic consumes network resources and places extra load on end-servers. While the Internet capacity is increasing at a rapid rate, so are the demands of new applications. Moreover, as Internet access improves, the performance bottleneck will increasingly shift towards

the backbones and end servers. Reducing resource consumption is therefore an important way to increase the effective capacity of the Internet.

Caching and replication are classic ways to address the problems above. However, there are some important challenges in getting the most out of caching and replication in the context of the Internet.

Transparency problem: provide transparent caches that can be placed freely in the network. To achieve transparency, protocol support is required to provide better way to funnel requests into the caching infrastructure. Today, traffic is hijacked or browser is statically configured. Neither solution is satisfactory. Hijacking breaks TCP connections when packets from a client take different paths to the destination. Configuring browsers places too much of an administrative burden on ISPs, and is not feasible at all for a higher-tier ISP whose customers are other ISPs and not end users.

Placement problem: Once the transparency problem is solved so that we are free to put transparent caches anywhere in the network, the next question is what are the right strategic places to deploy them? Similarly, where in the network should content servers be put and what is the replication degree for replicated servers? At a higher level, for a given deployment of caches and servers in the network, what content should be placed on which caches and servers? Being able to dynamically and automatically reconsider content placement is important to deal with changing demands. All these issues constitute a placement problem. Addressing the placement problems requires good models for network topology, traffic matrix, and access model. It also requires continuous measurement of traffic, network and server loads, and content usage.

Dimensioning problem: Another challenge is understanding the dimensioning of proxies. How powerful a CPU should a given proxy server have, how much main memory, how many disk drives and what size? The problem is somewhat analogous to storage hierarchy issues of the past two decades.

Further challenges include putting intelligence and appropriate protocol support into caches and servers for prefetching, cooperation with other caches, maintaining consistency, and increasing the portion of Internet traffic that can be cached. For prefetching, caches and servers could exchange hints allowing better prediction of future demand. For inter-cache cooperation, better mechanisms are needed for location management (i.e., allowing a cache to find out which other caches may have a requested object) and for server selection, including an intelligent decision on when to obtain an object from a remote cache and when to fetch it directly from the end server. Regarding the uncacheable information and consistency, part of the challenge lies in educating content providers in proper design of Web sites and adopting certain technologies that have already been proposed.

5 Conclusion

A number of themes emerged in the workshop, and are reflected in this paper.

First, measurement techniques for the Internet have begun to develop, but significant unsolved problems remain. Workload measurement is challenging, and needs to reflect the influence of the heavy distributional tails present in many workload metrics. One of the biggest challenges appears to be to understand whether and how workload characteristics are changing as new applications arise and are deployed in the Internet. Measurement approaches at the network level are varied, and both passive and active measurement techniques have been developed. While a number of active network measurement techniques can now provide a window into some network properties (round trip time, routing paths, packet loss) there are many other questions about network state that are not yet directly measurable; furthermore, the challenge of discerning network state from passive measurements is to date largely unmet.

Second, techniques for modeling the Internet's performance are not yet well developed. A significant complication is the closed-loop nature of most Internet applications, in which user behavior, protocol adaptations, and network conditions all simultaneously interact. A related problem is that most modeling methods tend to focus on a single abstraction level (packets, routes, flows, transactions) and to abstract away temporal and spatial variability of network conditions; yet it is just these properties that seem to be most important in the performance of today's Internet.

Finally, methods that improve performance and resource manangement in the Internet are being developed very rapidly, and are incredibly varied. These methods (such as caching, indexing, data replication, new protocols and new service models) are often not subjected to rigorous performance evaluation, both because of the rapid pace of change and because of the performance evaluation difficulties mentioned above. However, to move the Internet to the next level of functionality, one in which quality of service and resource consumption are bounded and understood, the ability to effectively and accurately model Internet performance will become indispensible.

Based on these observations, workshop participants developed concrete proposals for the steps necessary to advance the state of the art in Internet performance modeling, which are listed as "Grand Challenges" in this paper. Progress on these proposals will help move the Internet forward in the new century with a sound foundation, based on measurement, analysis, and modeling.