Internet Research Needs Better Models

Sally Floyd Eddie Kohler
ICSI Center for Internet Research, Berkeley, California
{floyd, kohler}@icir.org

1 INTRODUCTION

Networking researchers work from mental models of the Internet's important properties. The scenarios used in simulations and experiments reveal aspects of these mental models (including our own), often including one or more of the following implicit assumptions: Flows live for a long time and transfer a lot of data. Simple topologies, like a "dumbbell" topology with one congested link, are sufficient to study many traffic properties. Flows on the congested link share a small range of round-trip times. Most data traffic across the link is one-way; reverse-path traffic is rarely congested.

All of these modeling assumptions affect simulation and experimental results, and therefore our evaluations of research. But none of them are confirmed by measurement studies, and some are actively wrong. Some divergences from reality are unimportant, in that they don't affect the validity of simulation results, and simple models help us understand the underlying dynamics of our systems. However, as a community we do not yet understand which aspects of models affect fundamental system behavior and which aspects can safely be ignored.

It is our belief that lack of good measurements, lack of tools for evaluating measurement results and applying their results to models, and lack of diverse and well-understood simulation scenarios based on these models are holding back the field. We need a much richer understanding of the range of realistic models, and of the likely relevance of different model parameters to network performance.

2 NETWORK MODEL PRINCIPLES

By *network model*, we mean the full range of parameters that might affect a simulation or experiment: network topology, traffic generation, end-node protocol behavior, queue drop policies, congestion levels, and so forth. Internet experiments are difficult to replicate, verify, or even understand [17] without the stability and relative transparency provided by a simulator (such as ns [15]), emulator (such as the University of Utah's Emulab [18]), or self-contained testbed; and experimental design for these platforms includes the design and implementation of an explicit and concrete network model.

Network models used in practice often have little relationship to Internet reality, or an unknown relationship to Internet reality. This isn't necessarily a problem. Divergences between models and reality can be unimportant, in that they don't affect the validity of simulation results, or useful, in that they clarify behavior in simple cases. Some divergences are necessary in order to investigate the Internet of the future instead of the Internet of the past or present. However, the research community has not yet determined which divergences are acceptable and which are not. We simply don't know whether the models we use are valid. This basic question has led to difficulties both in our own research and in our evaluation of other work.

We need better models and better tools for evaluating our own and others' models. We need to know when a model might lead to bad results, and what those results might be. In particular, we believe:

Models should be specific to the research questions being investigated. We wouldn't recommend trying to construct a single model of the global Internet, with a single set of simulation scenarios, for use by all researchers. The Internet cannot be simply and accurately modeled in the same way that one might model a machine that one could hold in one's hand. Researchers should instead concentrate on modeling properties relevant to their research, and finding valid simplifications or abstractions for other properties. The very process of deciding which properties are relevant, and testing those decisions, gives insight into the dynamics of the questions under investigation. Building a single global model, in contrast, would make people's simulations run slower without necessarily improving their precision, clarity, or applicability.[1]

For example, one area of particular interest to us is congestion-related mechanisms at a queue in a router. This includes such research topics as differentiated services, active queue management, ECN, QoS, aggregate-based congestion control, fairness, and so forth, and touches on other issues, such as design of end-host protocols. Models for these topics must include characteristics of congested links, the range of round-trip times for flows on a congested link, and the effects of congestion elsewhere on the network. A fully-worked-out topology isn't necessary, however; the range of round-trip times, and an understanding of the congestion experienced elsewhere, sufficiently represents the topology. Table 1 describes typical models used in other research areas, such as unicast and multicast congestion control, routing lookups, and peer-to-peer systems.

We need to understand how models' parameter settings affect experimental results. As a model for a given research question is built, researchers should explore the model's parameter space. For example, do some parameters change results only slightly, or are results sensitively dependent on one or more parameters? Section 3 explores this in detail for several research questions. An understanding of the realm of possibilities, and their causes, can prove invaluable for interpreting results, and should be codified and distributed as part of the research community's shared knowledge base.

Modeling must go hand-in-hand with measurement. It is necessary to fully explore the range of parameter settings, but researchers should agree on particularly important settings to facilitate comparison of results. Network research should not founder on dis-

---

First Workshop on Hot Topics in Networks, Princeton, New Jersey, October 28–29, 2002

[1] Application-specific modeling is becoming a shared agenda in the research community, with work into application-driven topology modeling, for example [20].

Research Topics	Typical Models	Supporting Measurements
AQM, scheduling, differentiated services.	A dumbbell topology, with aggregate traffic.	Characteristics of congested links, range of round-trip times, traffic characterization (distribution of transfer sizes, etc.), reverse-path traffic, effects of congestion elsewhere.
Unicast congestion control.	A single path, with competing traffic.	Characteristics of links, queue management along path, packet-reordering behavior, packet corruption on a link, variability of delay, bandwidth asymmetry.
Multicast congestion control.	A single multicast group in a large topology.	Router-level topologies, loss patterns, traffic generation by group members.
Routing protocols.	A large topology.	Router-level topologies, AS-level topologies, loss patterns.
Routing lookups.	A lookup trace, or a model of the address space.	Ranges of addresses visible at a link.
Web caching and CDNs, peer-to-peer systems.	Models of large topologies with application traffic.	Topologies, application-level routing, traffic patterns.
Controlling DDoS attacks.	Models of large topologies with aggregate traffic.	Topologies, attack patterns.
Web cache performance.	A single cache with many clients and servers, as in Web Polygraph.	Detailed client behavior, server behavior.

Table 1—Some research topics, with typical models and required supporting measurements.

agreements over the network models and simulation scenarios that should be used. (Section 3 describes cases where we are close to that state of affairs.) Measurement can help settle these disagreements by saying what parameters, or ranges of parameters, are actually observed in practice.

We want models that apply to the Internet of the future, as well as to the Internet of today. Due to the Internet's vast heterogeneity and rapid rate of change [17], we must pay close attention to what seems to be invariant and what is rapidly changing, or risk building dead-end models. Measurement, for example, should be an ongoing program, so that old measurements don't congeal into widely accepted, but inappropriate, parameter settings.

Better models will make the Internet community's research efforts more effective. Lack of agreement over models complicates comparison and collaboration, and researchers risk expending valuable effort on dead ends caused by invalid models. Better models will therefore immediately improve the state of Internet research, and perhaps the Internet itself.

3 “ROGUES’ GALLERY”

This section describes some modeling issues in our own, and others', network research. Some of the research we discuss has flaws, caused by inappropriate models, that might have been avoided given a better understanding of the network models appropriate for specific research topics. Some of it has not received a thorough evaluation because the models underlying the research have not been evaluated. The point is not to scold others (or ourselves!). Concrete examples are simply the most effective way to communicate the range of problems that can crop up when models aren't treated carefully enough.

Again, if models used today could be counted on to give similar results to one another, and if their results could be counted upon to be relevant to the current and/or future Internet, then there would not be a problem. However, different models and different simulation scenarios do give different results when used to evaluate the same research question, and have different degrees of relevance to the actual Internet.

3.1 Phase Effects

For example, some simulations demonstrate sensitive dependence on precise parameter settings. This rich behavior is not relevant to the modern Internet; it is an artifact of unrealistic simulation scenarios, such as those with long-lived traffic, packets the same size, and no reverse-path traffic. We would like to discourage researchers from investigating in depth the rich behavior of these unrealistic and irrelevant scenarios [19].

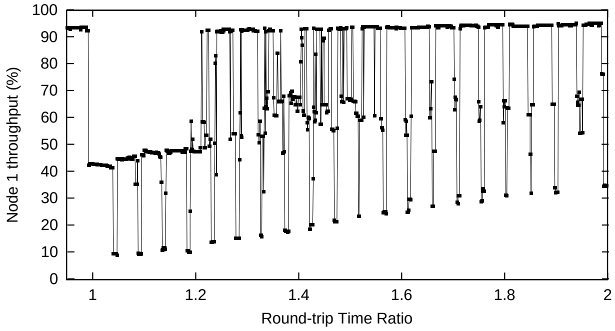

Figure 1—Flow 1's throughput as a function of the ratio of the two flows' round-trip times.

Figure 1 (taken from [6]) illustrates phase effects, where a small change in the propagation delay of a single link completely changes the fraction of link bandwidth received by one of two TCP flows sharing a Drop-Tail queue. Each dot on the graph represents the result of a single simulation; the y-axis shows the throughput of flow 1 in that simulation. The simulation topology is a simple dumbbell. When the propagation delays of the two competing flows' access links are equal, then both flows have the same round-trip time and receive the same fraction of the link bandwidth. However, as the propagation delay of one of the access links changes slightly, flow 1 can shift to receiving almost all of the link bandwidth, or to receiving very little of the link bandwidth, depending on the exact propagation delays of the two access links. In real networks, of course, the traffic mix includes short-lived flows, and small control packets as well as large data packets, and probably more than two competing flows, all making phase effects much less likely. The lesson is not that phase effects are a significant or important dynamic to address in current networks, but rather that simulations can be very tricky and unrealistic, and that the combination in a simulation scenario of DropTail queue management with one-way long-lived traffic can be deadly indeed.

3.2 Active Queue Management: Parameters

Random Early Detection (RED) was one of the first proposals for Active Queue Management, and the 1993 paper on RED [7] included a number of simulations, investigating scenarios with a range of round-trip times; varying traffic load over the life of the simulation; two-way traffic including TCP connections with a range of transfer sizes; scenarios including bursty and less-bursty traffic; and a range of values for the configured target average queue size.

However, the 1993 paper neglected to address some key issues:

- The paper did not investigate performance in scenarios with high packet drop rates.
- The paper did not investigate performance for a range of link bandwidths for the congested link.
- The paper did not explore the potential for oscillations in the average queue size, in particular for scenarios with large propagation delays and long-lived traffic.

Partly because the paper neglected to address these issues, a lengthy literature was spawned on the limitations of RED, and nine years later Active Queue Management has still not seen widespread deployment in the Internet.

For instance, all of the paper's simulations were of scenarios with small packet drop rates, so performance looked quite nice. However, it was soon pointed out that performance looked less good when the packet drop rate exceeded RED's configured parameter max_p.[2] In 1997, the default value for max_p in the NS simulator was changed from 0.02, an unrealistically optimistic value, to 0.1. In 1999 the 'gentle' variant was added to RED to give increased robustness when the average queue size exceeded the maximum threshold, and Adaptive RED was developed in 2001 to adapt RED parameters to changing network conditions [5]. All of this might have been done much sooner if the authors of the RED paper (i.e., one of the co-authors of this paper) had paid more attention in 1993 to RED performance in scenarios with high packet drop rates.

Similarly, while the original RED paper gave guidelines for the setting of the queue weight parameter w_q, all of the scenarios in the paper had a congested link of 45 Mbps. This led to work by others using NS's default value of the queue weight parameter for a range of inappropriate scenarios, e.g., with 10 Gbps links, so that the average queue size was estimated over too small of a time interval, only a fraction of a round-trip time. The use of an overly-small value for w_q, particularly in an environment of one-way, long-lived traffic, can exacerbate RED's problems with oscillations of the queue size [5]. Again, if the authors of [7] had investigated and thought carefully about a wider range of simulation scenarios in 1993, it would have reduced the amount of work necessary later on. Even now that the default NS parameters have been changed to reasonable values, the effects those parameters had on simulation results should sensitize us to the importance of understanding the models we use.

An evaluation of AQM mechanisms in progress [16] shows that, for many simulation scenarios, all considered mechanisms perform similarly. However, simulation scenarios can be devised that show each mechanism in a bad light. In scenarios with long round-trip times and mostly long-lived flows, RED and Adaptive RED exhibit queue oscillations (see the next section). In scenarios with mostly web traffic, or with changes in the level of congestion over time, the Proportional-Integral Controller (PI) [8] and Random Early Marking (REM) [2] perform badly. Many scenarios with Drop-Tail or Adaptive Virtual Queues (AVQ) [13] give competitive performance in terms of delay-throughput tradeoffs, but also give high packet drop rates. It would be helpful to have more grounding in deciding which models and simulation scenarios were critical to explore, and which are edge cases that were less likely to occur in practice. It is unsettling to feel that one could construct a simulation to show almost anything that one wanted, and that there is so little agreement within the research community about why one chooses to explore one set of simulation scenarios rather than another.

3.3 Active Queue Management: Oscillations

Much research effort in active queue management mechanisms comes down to an implicit disagreement about which simulation scenarios are the most important to address. For example, [14] discusses oscillations with RED in scenarios with one-way, long-lived traffic, while [5] criticizes reliance on such scenarios. Queue oscillations are widely considered a serious potential problem with RED active queue management. However, moderate changes in the traffic mix can strongly affect oscillation dynamics. In particular, adding short-lived flows, reverse-path traffic, and a range of round-trip times—characteristics ubiquitous on the Internet—changes simple oscillations into more complex bursty behavior. This dramatic change highlights the importance of the network model. If we understood better the ways in which different models can affect experiment dynamics, perhaps we would be further along in addressing AQM behaviors.

To illustrate, we examine three simulations with somewhat similar parameter settings, but quite different results in terms of the queue dynamics at the congested link. The simulations share a dumbbell topology with a 15 Mbps, 10 ms congested link with Adaptive RED queue management; they all have similar, small amounts of reverse-path traffic; and they all run for 100 seconds. The simulations differ in their traffic mixes and flow round-trip times.[3] Figures 2 through 4 show, for each simulation scenario, the instantaneous queue size over the second half of the simulation, with the dashed line showing the average queue size estimated by RED.

In Figure 2, traffic consists mostly of 80 long-lived flows with large receiver's advertised windows, and with all round-trip times equal to 240 ms in the absence of queueing delay. This resembles models used in literature on RED oscillations [12], and indeed, although the packet drop rate is 2.8% over the second half of the simulation and the link utilization over the second half is also good at 98.6%, oscillations in the instantaneous queue size are quite pronounced.

Traffic observed at Internet routers, however, tends to exhibit a wide range of round-trip times, including relatively short round trip times ($<$ 50 ms) [1, 11]. Figure 3 changes the model of Figure 2 by introducing a wide range of round-trip times, which now vary between 20 and 460 ms. Stable oscillations in queue size have been replaced by more irregular behavior. The simulation might actually be used to argue that oscillations are not a problem on the Internet, because of the absence of regular oscillations of the queue size. The

---

[2] The parameter max_p gives the packet dropping probability imposed when the average queue size exceeds the maximum threshold.

[3] This scenario was adapted from [5, Section 5.1].

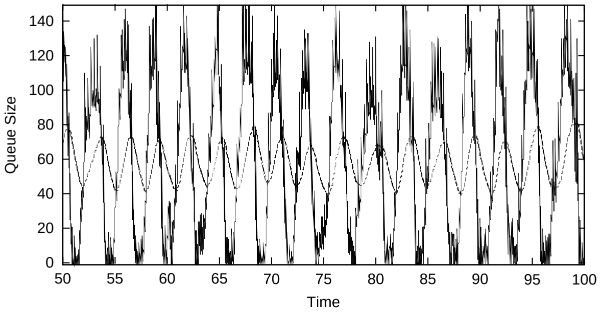

FIGURE 2—Long-lived traffic, 240 ms RTTs.

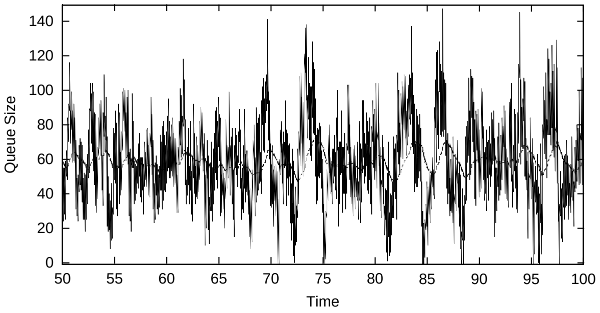

FIGURE 3—Long-lived traffic, 20–460 ms RTTs.

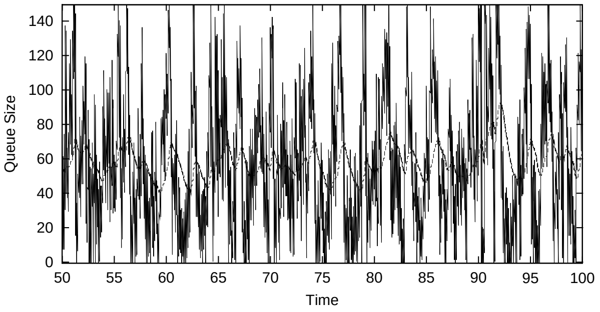

FIGURE 4—Mostly web traffic, 20–460 ms RTTs.

packet drop rate is now 4.6% over the second half (higher because of the influence of flows with very short round-trip times), and link utilization over the second half is now 99.9%.

But the traffic in Figure 3 still consists of all long-lived flows, while most flows on the Internet tend to have short lifetimes [4]. Figure 4 therefore introduces shorter-lived flows into the mix: traffic now mostly comes from the web traffic generator in NS, with a smaller number of long-lived flows (fifteen). The demand from the web traffic generator was chosen to give roughly same packet drop rate as Figure 2 over the second half of the simulation, in this case of 2.6%; the link utilization over the second half is also good, at 98.9%. The queue dynamics and the distribution of queuing delay are rather different, however. The queue size varies more extremely than in Figure 3, and unlike that simulation, the average queue size also varies significantly.

To some extent, we have lacked tools for evaluating the models our simulations actually use. For instance, do the round-trip times seen on the congested link in Figure 4 correspond to the full range we expect, and does that range correspond in a meaningful way to measured Internet data? It turns out that simple mechanisms can enable evaluation of aspects of a simulation's model.

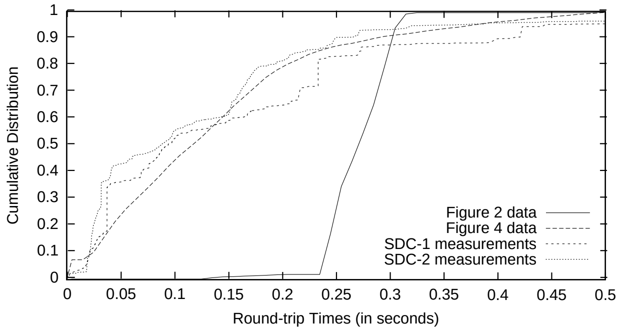

FIGURE 5—Distributions of packet round-trip times on the congested link of two simulations, with data measured on the Internet for comparison.

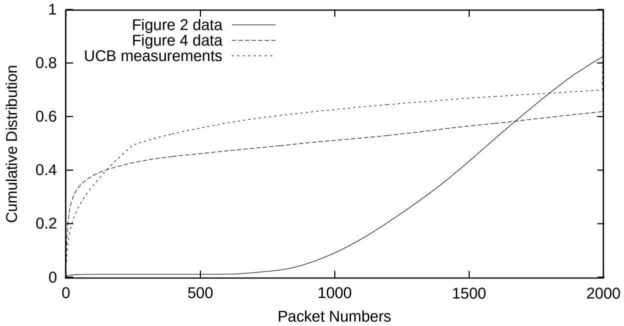

FIGURE 6—Distributions of packet numbers on the congested link over the second half of two simulations, with data measured on the Internet for comparison.

Figure 5 shows one way to evaluate the range of round-trip times in a simulation. We added mechanisms to the NS simulator to record the simulated TCP senders' estimated round-trip times for packets on the congested link. The figure shows a cumulative per-packet distribution of these measurements. It clearly demonstrates Figure 2's narrow range of round-trip times, from 240 to 310 ms, and Figure 4's much wider range, from almost 0 to more than 500 ms.[4] We have also included two representative measurements of an OC3 access link at UC San Diego, calculated by Jiang and Dovrolis using a passive TCP round-trip time estimator [11, Figure 13]. We used these measurements to guide our setting of link propagation delays, and as a result Figure 4 matches the measurements far better than Figure 2. Note that although the average round-trip time of a TCP connection in Figure 4 is still 240 ms in the absence of queueing delay, most of the packets come from the TCP connections with shorter round-trip times, as one would expect.

In order to better evaluate the mix of connection lengths in a simulation, we also added mechanisms to NS to record packet numbers seen on the congested link. The first packet sent in a flow is numbered 1, as are any of its retransmissions. The next packet is numbered 2, and so forth. Thus, graphing a cumulative distribution of packet numbers shows the fraction of packets sent during connection startup (slow start). This quantity is largely determined by the distribution of flow sizes in the simulation, but has independent interest.

Figure 6 show the cumulative distribution of packet numbers for the simulations in Figures 2 and 4, as well as from a July 2000

---

[4] Very short round-trip times are from the first packet in each connection, which reports an estimated round-trip time of 0 ms.

trace of wide-area traffic to and from UC Berkeley.[5] We used these measurements, in part, to guide our setting of the relative number of web sessions and of long-lived flows. As Figure 6 shows, almost all the packets in the second half of Figure 2's simulation were at least the 500th packet in their respective flows. This means there were no slow-start dynamics in that part of that simulation. In contrast, short-lived flows in Figure 4 gave rise to a substantial number of packets with small packet numbers in the second half of the simulation. The corresponding increase in slow-start dynamics probably influenced the simulation results.

These simulations raise the question of which is more important to explore, the pronounced oscillations in a scenario with long-lived flows all with the same round-trip time, or the variability of demand over shorter time scales that comes from a traffic mix and round-trip time distribution closer to that observed on real links? It is not obvious that the mechanisms proposed to address the oscillations in Figure 2 also perform well in scenarios with more diverse traffic (as in Figure 4), or in other scenarios that more stringently stress the responsiveness of the underlying queue management.

3.4 TCP Variants

It is not just AQM research that suffers from modeling issues. As examples of transport protocols, we show below how the designs of several TCP variants were influenced by implicit network models. In the case of Reno TCP [10], the model has proved false, and as a result Reno TCP has terrible performance in some scenarios that are common in practice. In the case of Vegas TCP [3], we aren't sure how frequently the underlying model applies in practice, making evaluation difficult.

Reno TCP added Fast Recovery to TCP in 1990, following Jacobson's introduction of congestion control in Tahoe TCP in 1988 [9]. Fast Recovery makes a key contribution of allowing the TCP sender to avoid slow-starting in response to congestion—with Fast Recovery, the TCP sender halves its congestion window and avoids a slow-start. Reno TCP works well when only one packet is dropped from a window of data, but generally requires a Retransmit Timeout, and the attendant slow-start, when multiple packets are dropped from a window. This response would be perfectly appropriate if single packet drops were the typical occurrence, and multiple packet drops in a window of data in fact represented more serious congestion calling for a more serious congestion control response. Unfortunately, this is not the case; losses often come in bursts, particularly with Drop-Tail queue management, and Reno TCP responds to those bursts with long timeouts. Reno TCP's attendant performance problems led to a spate of papers proposing a range of mechanisms in the network to reduce multiple packet drops from a window of data, while better models—for instance, including the typical burstiness of flows slow-starting at different times—might have prevented Reno's performance problems with multiple packet drops in the first place. It is straightforward to modify Fast Recovery to avoid Reno's unnecessary Retransmit Timeouts, as illustrated by later TCP variants such as NewReno TCP, which fixes this bug.

As a second example of how limitations in modeling assumptions affect transport design, we consider Vegas TCP [3]. Vegas is optimized for environments with very low levels of statistical multiplexing (e.g., only a few active TCP connections), where the sending rate of an individual TCP connection strongly affects the queue size at the router. In such a scenario, increases in the congestion window past its optimal size only increase the queueing delay, rather than increasing the connection's sending rate. Thus, once increased queueing delay is detected, Vegas TCP refrains from further increases in the congestion window.[6] However, under different models—with higher levels of statistical multiplexing, for example, where the queueing delay and packet drop rate experienced by a connection have very little to do with the sending rate of that flow—Vegas TCP performs significantly worse than in the environment with small-scale statistical multiplexing.

We actually know very little about where Internet congestion occurs, or where it can be expected to occur in the future. Are congested links lower-bandwidth access links with low levels of statistical multiplexing, or high-bandwidth transoceanic links with high levels of statistical multiplexing, or both (as would seem to be the case)? What are typical levels of congestion, or of packet reordering, or of packet corruption? The more we know about the range of realistic network conditions, and of how this range might be changing over time, the better we can make informed choices in our design of transport protocols.

4 MOVING FORWARD: A PROPOSAL

Researchers could conceivably use existing measurements and analysis methodologies to understand the models they use for their simulations. Unfortunately, those measurements and methodologies have never been synthesized into a convenient, coherent whole. We lack an agreed-upon set of best modeling practices, partially because we have not yet recognized that creating such best practices is a legitimate research goal in its own right.

We hope this paper helps broaden discussion within the research community about the models we use. In addition, we have laid out a path for our own research that leads towards more relevant Internet models. The rest of this section lays out that path in outline.

We intend to begin with specific research questions, such as questions around congestion-related mechanisms at router queues. Analysis of the research questions will lead to a description of the experimental parameters relevant for constructing models. Sections 3.2 and 3.3, for example, showed that bottleneck link bandwidth, the range of expected round-trip times of flows on the link, and the range of flow lengths are all relevant parameters for AQM models.

Next, simulation experiments will show how parameter settings affect the observed behavior of existing techniques. Relevant experiments will be based on published research in the area. For settings that do affect behavior, new measurement studies and analysis of the measurement literature will describe how the settings look on the real Internet.

We will distill this work into a set of best practices for model construction. This may include ready-made simulation setups, papers, RFC-like documents, and so forth. We eventually hope to facilitate the creation of a shared repository of models and simulation scenarios for use by all.

Of course, changes in the network might expose the importance of different parameters. Our work will not determine the complete set of interesting simulations for a research area. Rather, it will point out the parameters that have proved important in the past, provide observations of their values on the Internet, and describe expected effects of other values.

Finally, we will make the measurement programs we borrow, or create, available to the research community as high-quality, maintained software tools. This will make it easy for the community to keep the best-practice models up to date with changing Internet

---

[5] The per-byte distribution of packet numbers was calculated from a list of connections, along with the total number of packets and of bytes for each connection, derived by Ratul Mahajan from a July 2000 trace file of wide-area traffic to and from UC Berkeley.

[6] Vegas TCP can be seen in part as a reaction to the poor performance of Reno TCP in the presence of multiple packet drops.

conditions.

Note that while we welcome collaborators, we don't think we've found the only, or even necessarily the right, approach. More important is to address the problem itself: the need for better models in Internet research.

5 CONCLUSIONS

In summary:

- Network research, and Internet research in particular, has a great need for better models, and for better common evaluation of models.
- Specific research problems require their own models—problem- or application-driven modeling, rather than global Internet modeling.
- We need a better understanding of exactly which aspects of models are critical for a particular research issue.
- Models must be based on network measurement when necessary.
- We want models that apply to the Internet of the future, as well as to the Internet of today.
- We have some ideas that we plan to put into practice, but this project can only flourish with the commitment of the research community as a whole.

The simulation scenarios we used to generate figures in this paper may be found at `http://www.icir.org/models/sims.html`.

ACKNOWLEDGMENTS

We gratefully thank Hao Jiang and Constantinos Dovrolis for generously providing the data on measured round-trip-time distributions shown in our Figure 5, and Ratul Mahajan for help creating the data on measured packet-number distributions shown in our Figure 6. We also thank anonymous reviewers for comments on previous drafts, and the National Science Foundation for funding our future work on this research program.

REFERENCES

[1] M. Allman. A Web Server's View of the Transport Layer. *ACM Computer Communication Review*, 30(5), Oct. 2000.

[2] S. Athuraliya, S. Low, and D. Lapsley. Random Early Marking. In *QofIS 2000*, Sept. 2000.

[3] L. Brakmo, S. O'Malley, and L. L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In *SIGCOMM 1994*, Aug. 1994.

[4] C. Cunha, A. Bestavros, and M. Crovella. Characteristics of WWW Client-based Traces, 1995. Boston University Technical Report BUCS-95-010, 1995.

[5] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management, Aug. 2001. `http://www.icir.org/floyd/papers/adaptiveRed.pdf`.

[6] S. Floyd and V. Jacobson. On Traffic Phase Effects in Packet-Switched Gateways. *Internetworking: Research and Experience*, 3(3), Sept. 1992.

[7] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, Aug. 1993.

[8] C. Hollot, V. Misra, D. Towsley, and W. B. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In *INFOCOM 2001*, Apr. 2001.

[9] V. Jacobson. Congestion Avoidance and Control. In *SIGCOMM 1988*, Aug. 1988.

[10] V. Jacobson. Dynamic Congestion Avoidance / Control (long message). Email message, 11 Feb. 1988. `http://www-nrg.ee.lbl.gov/nrg-email.html`.

[11] H. Jiang and C. Dovrolis. Passive Estimation of TCP Round-Trip Times. *ACM Computer Communication Review*, 32(3), July 2002.

[12] K. B. Kim and S. H. Low. Analysis and Design of AQM for Stabilizing TCP. Technical Report caltechCSTR:2002.009, Caltech, Mar. 2002.

[13] S. Kunniyur and R. Srikant. Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management. In *SIGCOMM 2001*, pages 123–34, Aug. 2001.

[14] V. Misra, W. Gong, and D. Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In *SIGCOMM 2000*, Aug. 2000.

[15] NS project. The Network Simulator—ns-2 (Web site). `http://www.isi.edu/nsnam/ns/`.

[16] J. Padhye, S. Floyd, and S. Shenker. Simulation-based Comparison of AQM Schemes. Work in progress, citation for acknowledgement purposes only.

[17] V. Paxson and S. Floyd. Difficulties in Simulating the Internet. *IEEE/ACM Trans. on Networking*, 9(4):392–403, Aug. 2001.

[18] Utah Emulab project. Emulab.net, the Utah network testbed (Web site). `http://www.emulab.net/`.

[19] A. Veres and M. Boda. The Chaotic Nature of TCP Congestion Control. In *INFOCOM 2000*, Mar. 2000.

[20] E. Zegura. New Directions and Half-Baked Ideas in Topology Modeling. viewgraphs, IPAM workshop, June, 2002, `http://www.cc.gatech.edu/projects/joint/pubs/IPAM-concluding-workshop.ppt`.