Internet Research Needs Better Models

Sally Floyd Eddie Kohler
International Computer Science Institute
floyd@icir.org kohler@icir.org

1 Introduction

Networking researchers work from mental models of the Internet's important properties. The scenarios used in simulations and experiments (including our own) reveal aspects of these mental models, 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 traffic across the link is one-way; reverse-path traffic is rarely congested.

All of these modeling assumptions can deeply 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 [12] without the stability and relative transparency provided by a simulator (such as ns [10]), emulator (such as the University of Utah's Emulab [13]), or self-contained testbed; and experimental design for these platforms includes the design and implementation of an explicit and concrete network model.

Researchers often use network models with little relationship to Internet reality, or with an unknown relationship to Internet reality. This isn't necessarily a problem. Some divergences between models and reality are 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 entire 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 properly 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]

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

Research Problems	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 problems, with typical models and required supporting measurements.

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 considers the typical models for other research areas as well, 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 disagreements 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 [12], 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

modeling, for example.

ongoing program, so that old measurements don't ossify 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, the meat of the paper, 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 the ranges of models used today could be counted on to give similar results, and if the 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 [14].

Figure 1 (borrowed from [4]) 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 DropTail queue. Each dot on the graph represents the result of a single simulation; the *y*-axis shows the throughput

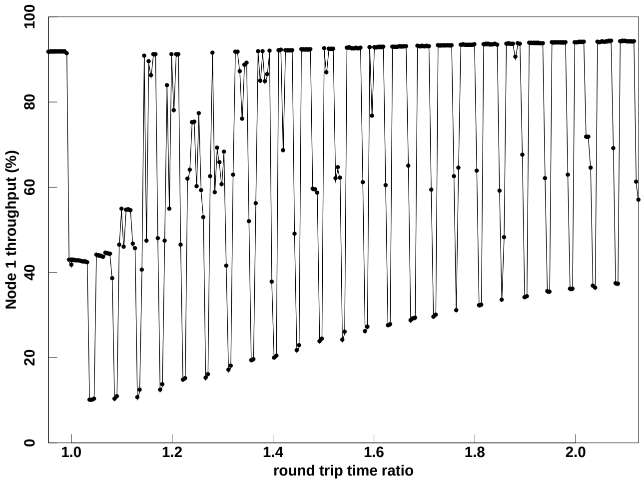

Figure 1: Flow 1's throughput as a function of the ratio of the two flows' round-trip times.

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 contains many short-lived flows, and core links have a much higher degree of statistical multiplexing, 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 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 [5] 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 explore the effects of parameter settings in a sufficiently wide range of scenarios.
- 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 [3]. 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, e.g., 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 [3]. Again, if the authors of [5] 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 [11] 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) [6] and Random Early Marking (REM) [1] perform badly. Many scenarios with Drop-Tail or Adaptive Virtual Queues (AVQ) [8] 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 of the work in active queue management mechanisms comes down to an implicit tussle about which simulation scenarios are the most important to address. E.g., [9] discusses oscillations with RED in scenarios with one-way, long-lived traffic, while [3] criticizes the reliance on these scenarios. Queue oscillations are widely considered a serious potential problem with RED active queue management. However, moderate changes in 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, Figure 2 shows two simulations with quite different results in terms of the queue dynamics at the congested link.[3] The top simulation shows the instantaneous queue size over the second half of a

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

[3] This scenario was taken from [3, Section 5.1].

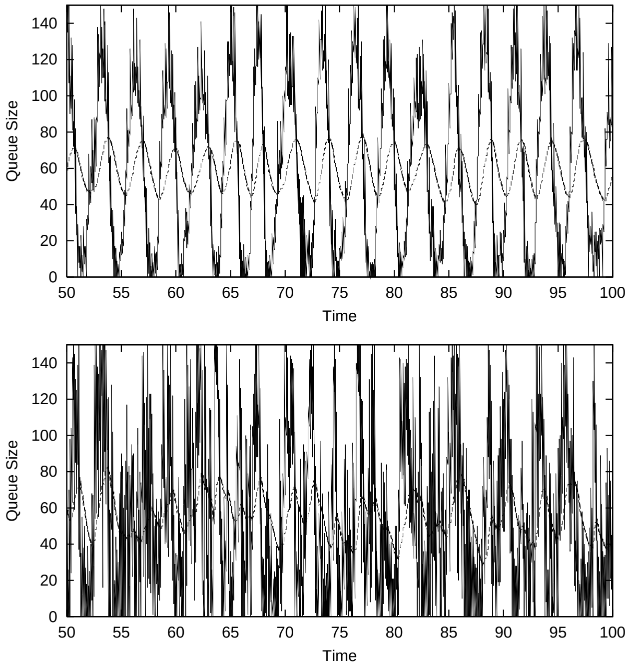

Figure 2: Scenarios with different traffic mixes and distributions of round-trip times. Top: mostly long-lived traffic, all with the same round-trip time. Bottom: mostly Web traffic, with a range of round-trip times.

100-second simulation. The simulation scenario is of a dumbbell topology with a 15 Mbps, 20 ms congested link, with all of the round-trip times equal to 180 ms. The traffic consists mostly of 80 long-lived flows with large receiver's advertised windows. The packet drop rate is 2.6%, and the throughput is 96%, but the oscillations in the instantaneous queue are quite pronounced. (The dashed line shows the estimated average queue size used by RED in determining the packet drop rate.)

In the bottom simulation scenario, two changes have been made. First, although the average round-trip time is still 180 ms, the round-trip times vary from 40 to 320 ms. The second and more critical change is that the traffic has been changed to a mix consisting mostly of traffic from the web traffic generator in NS, along with a small number (15) of long-lived flows. The demand from the web traffic generator was chosen to give the same packet drop rate of 2.6%. The throughput is also relatively high, at 93%, but the queue dynamics and the distribution of queuing delay are rather different. Which is more important to explore, the more 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 closer to that observed on real links, with flows with a distribution of transfer sizes? It is not necessarily the case that the mechanisms proposed to address the oscillations in the top graph also perform well in scenarios with more diverse traffic, as in the bottom graph.

3.4 TCP Variants

Not only AQM schemes suffer from modeling issues. The designs of several TCP variants, for example, were influenced by particular implicit network models. In the case of Reno TCP [7], 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 [2], we aren't sure how frequently the underlying model applies in practice, making evaluation difficult.

Reno TCP added Fast Recovery to TCP, which allows the TCP sender to avoid slow-starting in response to congestion—it halves its congestion window instead. Reno TCP works quite 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 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 come in bursts, and Reno TCP responds to those bursts with long timeouts. The attendant performance problems led to a spate of papers proposing a range of mechanisms in the network to reduce burstiness, where better models—e.g., including the typical burstiness of flows slow-starting at different times—might have prevented Reno's performance problems in the first place.[4]

Vegas TCP is optimized for environments with very low levels of statistical multiplexing (e.g., only a few active TCP connections), where the sending rate of a particular 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. Under different models—with higher levels of statistical multiplexing, for example, where the queueing delay experienced by a TCP connection has very little to do with the sending rate of that flow—Vegas TCP performs significantly

[4]Later TCP variants, including NewReno and SACK, fix this bug in Reno.

worse than in the environment with small-scale statistical multiplexing [2].[5]

We actually know very little about where Internet congestion occurs, or where it can be expected to occur in the future. Are most congested links lower-bandwidth access links with low levels of statistical multiplexing, or high-bandwidth transoceanic links with high levels of statistical multiplexing? Or are both types of links occasionally congested? Despite this lack of information, it is appropriate to design protocols that are optimized for one environment. The important thing is to address the issue, either by considering the full range of reasonable models or by making the choice explicit, if a choice is made.

4 Conclusions

In summary:

- Network research, and Internet research in particular, has a great need for better models, and for a 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.

But we close with questions, not statements. What would be needed for this to become a shared effort? An IRTF research group, with a collection of documents on evaluating models? Measurement tools to collect measurements? A shared repository of models and simulation scenarios? 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’.

References

[1] Sanjeewa Athuraliya, Steven Low, and David Lapsley. Random Early Marking. In *QofIS 2000*, September 2000.

[2] Lawrence Brakmo, Sean O’Malley, and Larry L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In *SIGCOMM 1994*, August 1994.

[3] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED’s Active Queue Management, August 2001. `http://www.icir.org/floyd/papers/adaptiveRed.pdf`, submitted for publication.

[4] Sally Floyd and Van Jacobson. On Traffic Phase Effects in Packet-Switched Gateways. *Internetworking: Research and Experience*, 3(3), September 1992.

[5] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, August 1993.

[6] C. Hollot, Vishal Misra, Don Towsley, and Wei Bo Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In *INFOCOM 2001*, April 2001.

[7] Van Jacobson. Dynamic Congestion Avoidance / Control (long message), 11 Feb 1988. URL http://www-nrg.ee.lbl.gov/nrg-email.html.

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

[9] Vishal Misra, Weibo Gong, and Don Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In *SIGCOMM 2000*, August 2000.

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

[11] Jitendra Padhye, Sally Floyd, and Scott Shenker. Simulation-based comparison of AQM schemes. Work in progress, citation for acknowledgement purposes only.

[12] Vern Paxson and Sally Floyd. Difficulties in Simulating the Internet. *IEEE/ACM Trans. on Networking*, 9(4): 392–403, August 2001.

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

[14] Andras Veres and Miklos Boda. The Chaotic Nature of TCP Congestion Control. In *INFOCOM 2000*, March 2000.

[5] Reno TCP also behaves badly in these scenarios when accompanied by high packet drop rates.