ITC19/ Performance Challenges for Efficient Next Generation Networks 1737-1746
LIANG X.J. and XIN Z.H.(Editors)
V.B. IVERSEN and KUO G.S.(Editors)
Beijing University of Posts and Telecommunications Press

FluNet: A Hybrid Internet Simulation/Emulation Environment for a Fast Queue Regime*

Yung Yi[a] and Sanjay Shakkottai[b]

[a]Department of Electrical and Computer Engineering,
University of Texas at Austin,
yi@ece.utexas.edu

[b]Department of Electrical and Computer Engineering,
University of Texas at Austin,
shakkott@ece.utexas.edu

Abstract:

Motivated by the scale and complexity of simulating large-scale networks, recent research has focused on hybrid fluid/packet simulators, where fluid models are combined with packet models in order to reduce simulation complexity. However, these simulators still need to track the queuing dynamics of network routers, which generate considerable simulation time-complexity in a large-scale network model.

In this paper, we propose a hybrid simulator – FluNet – where queueing dynamics are not tracked. The FluNet simulator is predicated on a fast-queueing regime at bottleneck routers, where the queue length fluctuates on a time-scale that is much faster than the time-scale of end systems. FluNet does not track queue lengths at routers, but instead, uses an equivalent rate based model at the router queue; and queue-based AQM schemes (such as RED) are replaced by equialent rate-based models. *This allows us to simulate large-scale systems, where the simulation "time-step-size" is governed only by the time-scale of the end-systems, and not the intermediate routers;* whereas a fluid model based simulator that *tracks* queue-length would require decreasingly smaller step-sizes as the scale size of system increases. We validate our model using a Linux based implementation with real traffic. Our results indicate a good match between packet systems and the associated FluNet system.

1. INTRODUCTION

The Internet has experienced tremendous growth in both scale and speed, and the control and management of the Internet is becoming an ever more important issue. To model and understand the behavior of such networks, several widely-used discrete event-driven simulators are available [1, 11, 23] in the area of simulation. However, event-driven simulation of large scale network systems with a significant number of users and flows passing over multiple autonomous systems with a large number of routers and complex routing patterns, is difficult due to computational complexity (leading to excessive time to carry out simulation).

Recently, there have been significant efforts on developing (approximate) fluid model based simulators to address the time-complexity of discrete event simulators. These simulators can be classified into *(i)* pure fluid model based studies, and *(ii)* hybrid fluid model based studies. Pure

---

*This research was supported by NSF Grants ACI-0305644, CNS-0325788, and CNS-0347400.

fluid model based research includes [4, 18, 20, 22], where the authors are primarily interested in rate based modeling of TCP sources, AQM algorithms, and their interactions. Recent work in [17] applies network calculus based on the mathematical theory of Min-Plus (or Max-Plus) algebra to fluid modeling of network dynamics. On the other hand, [5, 12, 19, 24, 28] integrate packet models along with fluid models to enable hybrid simulation. Hybrid simulators have the advantage of accurately tracking source dynamics (as the sources in the simulator are typically modeled using packet networks), while simultaneously using fluid approximations in the core network, where the system scale (i.e., a large number of flows and a large network capacity) permits fluid approximations to be accurate [25].

An important source of time-complexity is due to the simulation of queueing dynamics at the core network routers. Most existing hybrid simulators however still need to track the queuing dynamics of network routers (by means of fluid queues). In this paper, we propose a hybrid simulator – FluNet – where queueing dynamics are not tracked. Instead, queue-length based AQM schemes (such as RED) at intermediate routers are replaced by an equivalent fluid-rate based model.

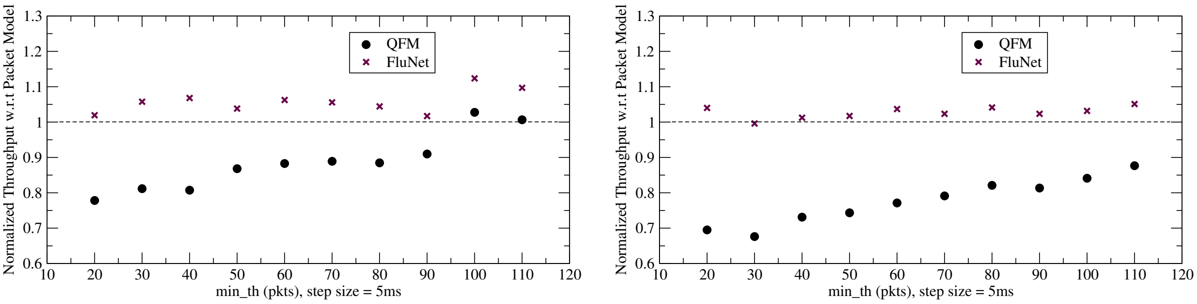
The main idea in FluNet is to replace the Internet core by a fluid model based network where router queues are replaced by equivalent rate-based models, while keeping the dynamics of end-systems unchanged. The main features of FluNet are summarized below.

(i) *Dimensional collapse*: Multiple packet flows between each pair of end-systems (such as between a pair of LANs/WANs) are represented by a single fluid-flow within FluNet, since congestion controllers at the intermediate routers need only “aggregate (over flows) rate information” to respond to occurring congestion.

(ii) *Absence of queueing dynamics*: Queue-length based AQM schemes (e.g., RED [10]) at intermediate routers are replaced by an equivalent fluid-*rate* based model, which leads to simpler modeling of the associated packet network. Further, FluNet has no control-theoretic approximation of source controllers, but uses actual end-systems in a Linux platform.

(iii) *Fast queue regime*: The FluNet simulator is predicated on a fast-queueing regime at bottleneck routers, where the queue length fluctuates on a time-scale that is much faster than the time-scale of end systems. This regime is reasonable to study, especially for large-scale systems where sufficient randomness (generated by end-systems, unresponsive flows, as well as intermediate routers) is present, and sufficient traffic aggregation occurs. FluNet does not track queue lengths at routers, but instead, uses an equivalent rate based model that depends on the (stochastic) stationary behavior of the router queue. *This allows us to simulate large-scale systems, where the simulation “time-step-size” is governed only by the time-scale of the end-systems, and not the intermediate routers;* whereas a fluid model based simulator that *tracks* queue-length would require decreasingly smaller step-sizes as the size of system increases (see Section 2 for details).

By implementing FluNet in a popular discrete event simulation (ns-FluNet) and in a real operating system (real-FluNet), we validate our model and its feasibility for both simulation as well as real-time emulation with real traffic. The simulation/measurement results show a good match between a packet system and the associated FluNet system under various network



(a) Scale 1: 200 Mbps, 100 TCP and 100 unresponsive (total 30 Mbps)

(b) Scale 2.5: 750 Mbps, 250 TCP and 250 unresponsive (total 75 Mbps)

Figure 2. Normalized throughput of QFM and FluNet with different system scales.

ing dynamics are fast (i.e., the queue-length process changes rapidly over a round-trip time), whereas a fluid-queueing simulator (such as in [12]) will be better when the queueing dynamics are slow.

A fast queueing regime corresponds to a small queue regime in a large-scale system (the system scale corresponds to the number of flows and link capacity). Note that the term 'small queue' corresponds to the queue length when normalized with the system capacity. Such a regime seems reasonable for large-scale systems based on arguments presented in [2]. The authors in [2] argue that the required buffer size *need not* scale linearly with the system size (i.e., with respect to the link capacity increase). This implies that in large systems, the buffer fluctuations will be fast, because the buffer size normalized to the link capacity shrinks [2]. Thus, the queue length *normalized* with the capacity will be small, leading to fast dynamics.

We also remark that results in [17] suggest that fluid queue based simulation could perform poorly when the bottleneck buffers are not saturated. This can be understood from the fact that unsaturated buffers correspond to a system with fast queueing dynamics, where tracking queue length trajectories (i.e, a fluid queue based approach) may not be feasible.

To illustrate the effect of fast and slow queueing regimes, in Figure 2, we have plotted the normalized average throughput of FluNet and QFM [12] (normalized with respect to the throughput measured with a pure packet-only simulation) for different RED parameters (`min_th` and `min_th`), and with simulation time-step-size being 5 msec. RED [10] is a queue based AQM mechanism at routers, that marks or drops packets depending on the queue length. The RED parameters (`min_th`$= a$, `max_th`$= b$) correspond to the case where marking/dropping occurs when the queue-length exceeds a packets/b packets, respectively. Throughout this paper, `max_th` is set to be three times as `min_th` [9]. In Section 3, we describe a rate-based equivalent model of RED that has been implemented in FluNet.

We remark that a *fast-queueing regime* results when the RED queue-threshold parameters are small. Thus, as the queue-size is moderately small, the queue-length fluctuates faster, leading to a fast-queue regime. As the queue-threshold parameters of RED increase, the queue length is allowed to build up to a larger value before marking/dropping occurs, thus leading to a *slow-queue regime*. We observe from Figure 2-(a) that when the queue parameters are small, the throughput measured from FluNet is close to that of a packet implementation (no fluid approximations); whereas when the RED parameters are large, QFM outperforms FluNet, and the throughput

Under such a regime on sizing of router buffer and a large amount of randomness in the system, we could have a considerable number of “cycles” in the queue dynamics of the intermediate routers even over a small interval of time (see the simulation results in [8]), where one “cycle” corresponds to the time interval over which an empty router buffer fills up and empties again (technically, the regeneration time). In other words, the queue dynamics occur on a much *faster* time-scale than that of the end system controller [8, 16]. In order to understand this intuitively, consider a router of capacity $n \times c$ accessed by n TCP flows and n unresponsive flows. Then, the time scale of a TCP source rate update is the order of $1/c$ (since its rate update is clocked by the ACK packets from the receiver), whereas the time scale of a router queue “cycle” is in the order of $1/(nc)$. Thus, it is reasonable to expect that queueing dynamics are not visible to the end system controller. Instead, the queueing behavior at the router affects the end system controller only through *the statistical behavior of the queue.* The authors in [8, 26] quantified the above heuristic by showing that *the queue based marking and the associated queueing dynamics can be approximated by a rate based marking function*.

3.2. Refinement: Queue Averaging Effect

The ERBM model considers a scenario where only packet marking occurs based on the instantaneous queue-length, whereas popular AQM algorithms such as RED [10] use *queue averaging* to filter the effect of short packet bursts due to TCP window dynamics [10]. In this section, we outline results that show that the ERBM model is valid even with queue averaging, under suitable assumptions. Due to space constraints, we provide only a summary of the model and the results. The details of the ERBM model as well as proofs, assumptions, and the system model used in the analysis summarized in this section are available in [27].

The system model can be summarized as follows. The system consists of a single bottleneck router fed by n TCP flows and n unresponsive flows (web-mice or other short flows), and with a queue based marking function (denoted by $p^q(\bar{Q}_n(t))$, where $\bar{Q}_n(t)$ is the weight-averaged queue length) is employed at the router. With this system, we will derive the equivalent rate based marking function for a given queue-based marking function. For a fixed $T > 0$, and for large n, we are interested in studying the queue length process (which measures the volume of data at the router) over the time-interval $[0, \frac{T}{n}]$. Thus, we are interested in the queue dynamics at the router over a short interval of time. Even over this small time interval, we will show that the queue reaches “steady-state” behavior. This occurs due to the fact that the capacity is very large (nc), and causes the queue to “regenerate” an arbitrarily large number of times over the interval $[0, \frac{T}{n}]$. However, from a single *end-system* (the user) point of view, this corresponds to a very short interval of time. Thus, one can expect that the end-user will only perceive the statistical “steady-state” queueing behavior. The results in this section quantify the above heuristic.

Let us denote the instantaneous queue-length process at the router by $Q_n(t)$ (the subscript n indicates that the capacty is nc), and the exponential moving averaged process by $\bar{Q}_n(t)$, which is given by $\bar{Q}_n(t) = w_n\bar{Q}_n(t-\delta_n) + (1-w_n)Q_n(t)$, where $0 < w_n < 1$ is the queue-averaging parameter for n-th system and $\delta_n = 1/(nc)$. In [10], the authors provides a guideline on how to choose the parameter w_n. Essentially, the authors in [10] argue that w_n is chosen such that a fixed burst of packets (i.e., L back-to-back packets from a single flow) should be allowed into the router without this burst being marked. This burst tolerance is chosen to account for TCP window behavior and cumulative ACKs, which lead to a burst of packets being transmitted from a single TCP source, instead of the packets being spaced apart. However, observe

that is Poisson with parameter λ (even if the actual system does not have Poisson arrivals). We define an equivalent rate based marking function $p^r(x,\lambda)$ as follows:

$$p^r(x,\lambda) \quad = \quad \begin{cases} E_{\pi^x}[p^q(Q)] & \text{if} \quad \frac{\lambda}{c-x} < 1 \quad \text{and} \quad x < c, \\ 1 & \text{if} \quad x \geq c \quad \text{or} \quad \frac{\lambda}{c-x} \geq 1, \end{cases} \tag{3}$$

where Q is the stationary queue length random variable and π^x is the stationary distribution of an M/D/1 queue with capacity $(c-x)$ and arrival rate λ. In other words, the congestion controller dynamics with a queue-based marking function $p^q(\cdot)$ can be well approximated by a *equivalent* system with only a rate-based controller $p^r(x,\lambda)$ at the router, where x and λ are simply the average arrival rate from the TCP and unresponsive flows (averaging over flows, not time) to the router queue, respectively.

In such a case, popular queue based marking schemes such as RED [10] and REM [3], can be approximated by an equivalent rate based marking, resulting in simpler system dynamics. The limiting system consists of a fluid model (rate based system update) and no (asynchronous) queueing dynamics. A natural implementation of ERBM model is to define a small measurement interval (time step size) that depends only on the end-system time-scale and the time-scale of the randomness, over which the average TCP and unresponsive arrival rates are measured, and to apply those rates to the equation (3).

4. EXPERIMENTAL RESULTS WITH REAL IMPLEMENTATION

Due to space limitations, we focus on measurements from real-FluNet implemented over Linux (see [27] for more simulation/emulation results with various network configurations, as well as an implementation within ns-2).

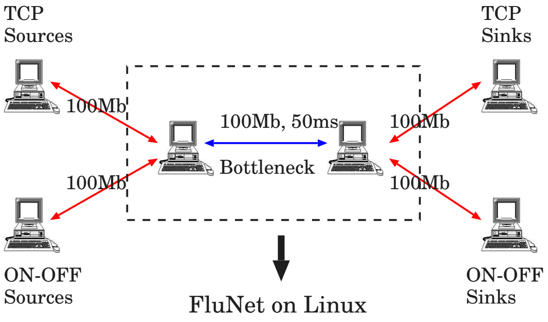


Figure 4. Network configuration with real-FluNet

The network topology for measurement is shown in Figure 4. We consider a simple topology in this section so that we can implement an actual packet network with the identical topology and provide base-line measurements for comparison. Our real-FluNet implementation can be configured for other topologies are well.

Two hosts are responsible for generating 50 TCP sources and a variable number of unresponsive ON-OFF sources. Two routers reside between source and destination pool, and all links are

8. S. Deb and R. Srikant. Rate-based versus Queue-based models of congestion control. In *Proceedings of ACM SIGMETRICS*, 2004.
9. S. Floyd. Red: Discussions of setting parameters. http://www.icir.org/floyd/REDparameters.txt.
10. S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
11. GloMoSim. http://pcl.cs.ucla.edu/projects/glomosim/.
12. Y. Gu, Y. Liu, and D. Towsley. On Integrating Fluid Models with Packet Simulation. In *Proceedings of IEEE INFOCOM*, March 2004.
13. C. V. Hollot, Y. Liu, V. Misra, and D. Towsley. Unresponsive flows and AQM performance. In *Proceedings of INFOCOM*, volume 1, pages 85–95, San Francisco, CA, April 2003.
14. Iperf. http://dast.nlanr.net/Projects/Iperf/.
15. H. Jiang and C. Dovrolis. The origin of tcp traffic burstiness in short time scales. Technical report, 2004. Available at ”http://www.cercs.gatech.edu/tech-reports/tr2004/git-cercs-04-09.pdf”.
16. F. P. Kelly. Models for a self-managed Internet. *Philosophical Transactions of the Royal Society*, A358:2335–2348, 2000.
17. H. Kim and J. C. Hou. Network Calculus Based Simulation for TCP Congestion Control: Theorems, Implementation and Evaluation. In *Proceedings of IEEE INFOCOM*, March 2004.
18. Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu. Fluid models and solutions for large-scale ip networks. In *Proceedings of ACM SIGMETRICS*, June 2003.
19. B. Melamed, S. Pan, and Y. Wardi. Hybrid discrete-continuous fluid-flow simulation. In *Proceedings of ITCOM, Scalability and Traffic Control in IP Networks*, August 2001.
20. V. Misra, W.-B. Gong, and D.Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *Proceedings of ACM SIGCOMM*, 2000.
21. NIST Net. http://snad.ncsl.nist.gov/itg/nistnet/.
22. D. M. Nicol and G. Yan. Discrete event fluid modeling of background TCP traffic. *ACM Transactions on Modeling and Computer Simulation*, 14(3), July 2004.
23. QualNet. http://www.scalable-networks.com.
24. G. Riley, R. Fujimoto, M. Ammar, K. Permula, and D. Xu. Distributed network simulations using the dynamic simulation backplane. In *Proceedings of International Conference of Distributed Computing Systems*, 2001.
25. S. Shakkottai and R. Srikant. How good are deterministic fluid models of Internet congestion control? In *Proceedings of IEEE INFOCOM*, New York, NY, June 2002.
26. Y. Yi, S. Deb, and S. Shakkottai. Short queue behavior and rate based marking. In *Proceedings of the 38th Conference on Information Sciences and Systems*, March 2004.
27. Y. Yi and S. Shakkottai. Flunet: A hybrid simulation/emulation environment at a fast queue regime. Technical report, WNCG, Department of Electrical and Computer Engineering, The University of Texas at Austin, June 2005.
28. T. Yung, J. Martin, M. Takai, and R. Bagrodia. Integration of fluidbased analytical model with packet-level simulation for analysis of computer networks. In *Proceedings of SPIE*, 2001.