A Study of Networks Simulation Efficiency: Fluid Simulation vs. Packet-level Simulation

Benyuan Liu, Daniel R. Figueiredo, Yang Guo, Jim Kurose, Don Towsley [1]
Department of Computer Science
University of Massachusetts

Abstract—**Network performance evaluation through traditional packet-level simulation is becoming increasingly difficult as today's networks grow in scale along many dimensions. As a consequence, fluid simulation has been proposed to cope with the size and complexity of such systems. This study focuses on analyzing and comparing the relative efficiencies of fluid simulation and packet-level simulation for several network scenarios. We use the "simulation event" rate to measure the computational effort of the simulators and show that this measure is both adequate and accurate. For some scenarios, we derive analytical results for the simulation event rate and identify the major factors that contribute to the simulation event rate. Among these factors, the "ripple effect" is very important since it can significantly increase the fluid simulation event rate. For a tandem queueing system, we identify the boundary condition to establish regions where one simulation paradigm is more efficient than the other. Flow aggregation is considered as a technique to reduce the impact of the "ripple effect" in fluid simulation. We also show that WFQ scheduling discipline can limit the "ripple effect", making fluid simulation particularly well suited for WFQ models. Our results show that tradeoffs between parameters of a network model determines the most efficient simulation approach.**

Keywords—**fluid simulation, performance evaluation, traffic model**

I. Introduction

Traditionally, packet-level simulation has been widely used for performance evaluation of computer networks. However, this technique does not scale well as the size and complexity of networks increases. The fast growth of data communication networks over the past decade makes this approach computationally expensive, if not infeasible, for truly large scale models. Consequently, efficient simulation techniques for such networking models have become an important issue.

Many methods have been proposed to speed up network simulation. These methodologies can be categorized into three different and orthogonal types (Figure 1): computational power; simulation technology; and simulation model. In the direction of computational power, simulations can be sped up by using faster and more powerful machines. In the simulation technology direction, new enhanced algorithms for implementing the simulation can further speedup simulation. Algorithms such as the calendar queue algorithm and splay tree algorithm have been proposed in order to improve the efficiency of event list manipulation. Another technique in this direction that has received much attention in the literature, is the RESTART mechanism that explores rare event simulation [1]. A third approach is to use models with a higher level of abstraction, simplifying the simulation and improving its efficiency. The tradeoff in this case, is the accuracy of the desired measures of interest obtained by the more abstract model. For example, the packet-train simulation technique models a cluster of closely-spaced packets as a single "packet-train" [2].

[1]This research has been supported in part by the NSF under awards ANI-9809332 and CDA-9502639, and in part by CAPES (Brazil)

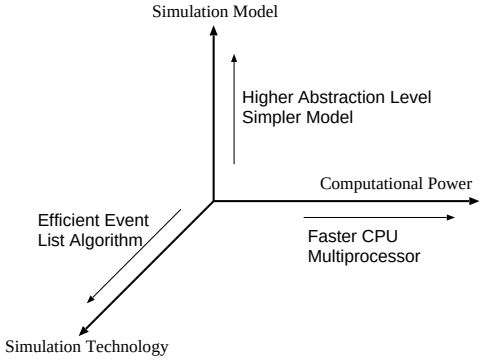

Fig. 1. Different ways to speed up simulation

Another modeling technique making simplified assumptions about the real system is the fluid model, which was first proposed by Anick et al. in [3] to model data network traffic. In the fluid simulation paradigm, network traffic is modeled in terms of a continuous fluid flow, rather than discrete packet instances. A cluster of closely-spaced packets may be modeled as a single fluid chunk with a constant fluid rate, with small time-scale variations in the packet stream being abstracted out of the model. A fluid simulator keeps track of the fluid rate changes at traffic sources and network queues. An equivalent packet-level simulator would keep track of all individual packets in the network. In fluid simulation, the higher level of abstraction suggests that less processing might be needed to simulate network traffic. Intuitively, this is not surprising as a large number of packets can be represented by a single fluid chunk. For simple network components, where traffic flows do not compete for resources, the fluid simulator outperforms the packet-level simulator. An example would be a link that connects two nodes and never experiences queueing; this component only introduces a constant propagation delay.

However, for other components where different traffic flows meet and contend for limited resources, it does not easily follow that fluid simulation always outperforms packet-level simulation. The management of the limited resource can significantly increase the total processing required by the fluid simulator. In previous work [4], we found that the fluid simulation can sometimes be less efficient than packet-level simulation due to this reason.

In this paper, we investigate several common networking scenarios and compare the amount of computational effort required by fluid simulation and packet-level simulation. As a measure of computational effort, we use the notion of event rate and show

that this is both adequate and accurate. We derive analytical results to characterize the event rate for some of the scenarios studied. We identify the major factors that contribute to the event rate of both fluid and packet-level simulation approaches and establish the tradeoffs between the different factors. The flow aggregation is considered as a technique to reduce the fluid simulation event rate. We also show that WFQ scheduling discipline can reduce the impact of the "ripple effect" in fluid simulation. We designed and implemented a fluid simulator and a packet-level simulator in order to explore their functionalities and validate analytical results.

One drawback of a fluid model is that the accuracy of the interest measures is compromised due to the abstraction. In this study we will not address the accuracy issues of the fluid simulator, but rather focus on the relative efficiencies of fluid and packet-level simulation. Nicol et al. [5] claim that despite the high level of abstraction of fluid models, the error of estimated measures obtained with fluid simulation is very small compared to the results of packet-level simulation. Another important issue that will not be addressed in this paper is how performance measures can be obtained via fluid simulation. However, we note that in [5], [6], [7], the authors show how performance measures such as end-to-end delay and loss characterization can be computed in fluid simulations.

The rest of the paper is structured as follows. Section II gives a brief review of fluid models and previous work. In Section III feed-forward network models are presented and analyzed. Section IV investigates the performance of feedback network models. In Section V we show that flow aggregation can significantly reduce the fluid simulation event rate. Preliminary results concerning WFQ are provided in Section VI. Finally, concluding remarks and discussions are presented in Section VII.

II. Background and previous work

In this section we describe the networking model components that will be used as building blocks to create the scenarios investigated throughout this paper. The two basic building blocks are the source model and the multiplexer model. More complex network models are constructed using these two components. In the following we consider the dynamics of the two basic components for fluid and packet models.

Markovian on-off source models are often used in network research to capture the bursty nature of the network traffic. Both packet and fluid versions of these source models are widely encountered in the literature. The source transits between an on and off state, remaining in each state for an exponentially distributed amount of time. Throughout this paper, the transition rate from on state to off state and vice versa will be denoted by λ and μ, respectively. When in the on state, a packet source transmits packets according to a Poisson process with rate γ, while a fluid source sends out fluid at a constant rate. No packet or fluid is sent during the off period. Without loss of generality, assume that packets have a fixed size x. In order for a fluid source to be considered equivalent to a packet source, we would minimally require that both have the same average data rate. Given this condition, the fluid peak rate should be set to be $h = \gamma x$. Figure 2 illustrates the behavior of the on-off fluid packet and fluid sources.

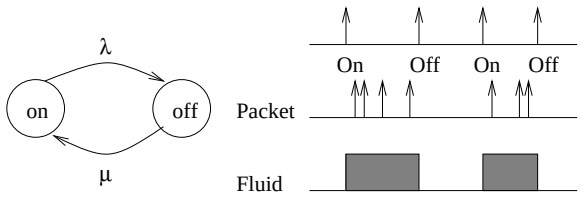

Fig. 2. The on-off fluid source and packet source

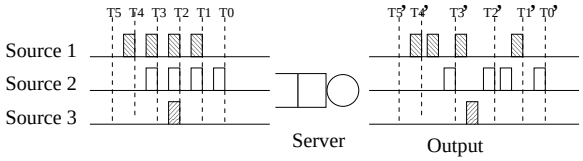

Fig. 3. Dynamics of a FIFO packet multiplexer

The second basic component of network models is the multiplexer. An important characteristic of a multiplexer is its scheduling discipline. In this paper, two scheduling disciplines are considered: FIFO and WFQ. The dynamic of FIFO scheduling is described here and WFQ is presented in Section VI.

For a given scheduling policy, the dynamics of the multiplexer depends on the underlying traffic model (fluid or packet). In the FIFO packet model, packets are placed into and served from a queue according to the order of their arrival. Figure 3 depicts a FIFO queue being fed by three different sources. Note that the departure process of the packets is similar to the arrival pattern. The only difference is the spacing between packets.

In the FIFO fluid model, the dynamics of the multiplexer are more subtle [4]. First, note that fluid from two different sources are distinct and do not mix, just as packets transmitted by two sources can be differentiated at the queue. Figure 4 depicts the dynamics of a FIFO fluid multiplexer being fed by three independent and identical sources. For this example, assume that the arrival rate of one source is equal to the service rate of the multiplexer. At time $T0$ the second source enters the on state and immediately starts to receive service at full capacity. At time $T1$, source 1 turns on and both flows must now share the service rate equally. At time $T2$, source 3 starts transmitting and eventually, at time $T2'$, all of them will share the service rate equally. Note that the arrival of flow 3 at time $T2$ causes flow 1 and flow 2 to change their departure rates at $T2'$. In general, the arrival of a new flow can cause a departure rate change in many other flows. Note that the output process of a given source can contain many *more* rate changes than the arrival process. For a larger network model, these rate changes will be propagated downstream causing even more rate changes in the departure process of other flows. This propagation and amplification of rate changes has been observed before and is known as the *ripple effect* [8], [4]. We will shortly see that the ripple effect has a profound impact on the efficiency of fluid simulation.

However, not every flow rate change affects the output rates of all the other flows currently being serviced. If the sum of the rates of all arriving flows is smaller than the service rate of the queue, then a flow rate change does not affect other flows currently being serviced. In this case, we say that a flow rate change does not *"interfere"* with other flows in the queue. An

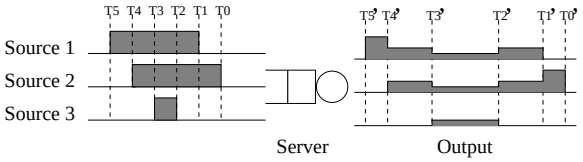

Fig. 4. Dynamics of a FIFO fluid multiplexer

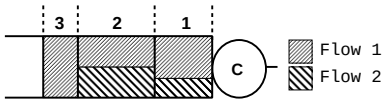

Fig. 5. Different fluid chunks in a FIFO queue

arrival flow rate change causing other flow rate changes in the departure of the queue is called *"flow interference"*.

To better understand how different flows are stored and served in a multiplexer, we introduce the concept of fluid chunk. A fluid chunk is a contiguous well-defined amount of fluid stored in the queue. Figure 5 illustrates three chunks of fluid formed by two different flows. Note that within a chunk the ratio of the arrival rate among different flows remains constant. Chunks are determined and placed in the queue in the order that they arrive. A chunk is served once it reaches the head of the queue. All the flows in the chunk are served simultaneously, with service rates proportional to the flow arrival rates within the chunk. Figure 5 depicts three fluid chunks that are formed by the rate changes of two flows. In the first chunk the arrival rate of flow 1 was greater than that of flow 2. In the second chunk the arrival rates are equal, and in the last chunk flow 2 goes off.

Another intrinsic characteristic of fluid simulation is flow merging, and consequently, the flow merging probability. Flow merging can be best understood with a single source single queue model. It occurs when the queue is back-logged and the source transits from the off to the on state. In this case, the output rate of the flow will not change at the end of the current chunk, and the two consecutive fluid chunks will merge in the queue. The probability that such a scenario occurs during simulation is denoted as the merging probability, i.e., the probability that a rate change on an input flow does not introduce a new chunk.

In order to compare the relative efficiencies of fluid simulation and packet-level simulation, we must establish a common measure of computational effort. In a simulation, basic events such as packet arrival/departure or fluid rate change, are fundamental units of work and (as we shall see) are directly related to the computational effort involved in a simulation. Therefore, we use the simulation event rate as a measure of computational effort. The event rate of a simulator is defined as:

$$e = \lim_{t \to \infty} \frac{N(t)}{t} \tag{1}$$

where $N(t)$ is the total number of events processed by the simulator by simulation time t.

In a packet-level simulation, events include source transitions, packet generation and packet departures from queues. For fluid simulation, events include source transitions and flow rate changes at sources and queues. We note that different events usually require different amounts of computational work, and these differences may change from one simulator to another. However, since the event rate includes all the events within the simulator and each event represents a basic computational effort, we expect the event rate to be an adequate measure of the overall computational effort of the simulator. We will indeed see that the execution time of a simulation is proportional to the corresponding event rate.

In order to compare the actual efficiency between fluid simulation and packet simulation, and validate analytical results obtained, two simulation frameworks were designed and implemented. The framework developed consists of simple and independent network components such as sources, links, routers and queues that form the building blocks for the network model. These building blocks can be instantiated by specifying their specific parameters, and connected together to form a much larger complex model. Each simulator framework corresponds to a modeling technique, packet or fluid, and both have a very similar interface so that a model can be executed in the packet or fluid simulator with minimum modification. Both simulators were constructed using the SSF (Scalable Simulation Framework) framework, which itself allows the construction of general purpose event-driven simulation [9]. Several APIs are publicly available in different programming languages. In our work, we used the DaSSF implementation which provides enhanced simulation features such as scalable parallel techniques [10]. Both packet and fluid simulator frameworks developed as part of this study are publicly available [1].

All simulation experiments conducted in this report were carried out using the simulation frameworks mentioned above. The simulations were performed on a dual processor Pentium-III 730MHz machine with 1Gb of physical memory running Red Hat Linux release 6.1 with kernel version 2.2.14-mosix SMP.

In all experiments involving a comparison between fluid and packet-level simulation, the sample path of the on and off periods of corresponding sources is the same. The purpose is to highlight other types of events occurring in the simulation and establish a fair comparison between the two simulators.

III. Feed-Forward FIFO Networks

In this section, we compare the simulation event rate between the fluid simulation and packet-level simulation, in the context of feed-forward FIFO networks.

In order to investigate the impact of the ripple effect on the amount of computational effort required to simulate a network model, we first present a system consisting of a single on-off source and a single FIFO queue. This case does not exhibit a ripple effect since there is only one flow going through the queue. In our previous work [4], an analytical approximation for the event rate of a single FIFO queue was presented. The analysis in [4] contains a parameter to represent the merging probability, which was not solved. Using the technique presented in [3], we can obtain an exact analytical expression for the event rate of this model. The precise expression for the fluid simulation event rate is now given by:

[1] The simulator can be obtained at http://gaia.cs.umass.edu/fluidsim

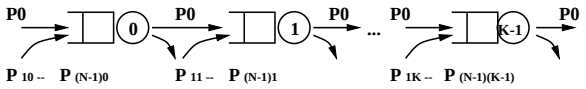

Fig. 6. A tandem queueing network

$$e^f = \frac{2\lambda\mu}{\lambda+\mu} + 2(1 - \frac{\lambda h}{c(\lambda+\mu)})\lambda \quad (2)$$

The derivation of this formula can be found in [11] and the result can be interpreted as follows. The first term in (2) is the event rate generated by the source transitions. Every time the source changes it state, a new event is counted, representing the flow rate change of that source. The second term is the queue departure event rate. This term accounts for the flow rate changes generated by the queue and the effect of chunk merging.

Compared with the exact packet-level simulation event rate $e^P = 2\lambda\mu/(\lambda+\mu)+2\mu\gamma/(\lambda+\mu)$ [4], with e^f from equation (2), we note that the fluid simulation outperforms packet-level simulation if $\gamma > \lambda/\mu(\lambda+\mu-\lambda h/c)$. This analysis was validated by simulation results. Therefore, for a single source, single queue case, a fluid simulation will require less computing than a corresponding packet-level simulation whenever the above condition for γ holds. We will see shortly however, that this advantage is mitigated by the ripple effect in the case of multiple queues.

Now we consider a larger and more complicated feed-forward network. The tandem queueing network was chosen as a concrete example to represent a feed-forward network because of its simple and regular topology. This permits tractable analytical models of simulation efficiency to be constructed and also allows easy parameterization of the model. At the same time, a more complicated feed-forward queueing network can be decomposed into many tandem queues with different parameters. We will see that by understanding the simple tandem queueing system, considerable insight can be gained into more complicated feed-forward networks.

The tandem queueing network model considered in this section is depicted in Figure 6. In this particular model, there are N input flows at each queue, with flow 0 starting as an on-off source at node 0, and traversing through all the nodes. The other $(N-1)$ flows, which are all identical and independent on-off sources, leave the system after passing through one node. Although, this system has many parameters that can be tuned, two of them - the number of sources entering each node and the number of nodes in the system - are the most important, since they have direct influence on the ripple effect. Thus, both will be explored and discussed in the following.

We first consider the number of nodes in the tandem queue. The model investigated has 15 sources (14 new entering each node) per node with a constant load of 0.8 and $\lambda = \mu = 1$. Figure 7 presents the results obtained from the simulators. We observe that the fluid simulation event rate increases quadratically with the number of nodes in the system. This quadratic behavior is caused by the ripple effect, which propagates further as the number of nodes increases. The packet-level simulation results in Figure 7 are for $\gamma = 3$, and, not surprisingly, presents a linear increase. However, the fluid simulation can still

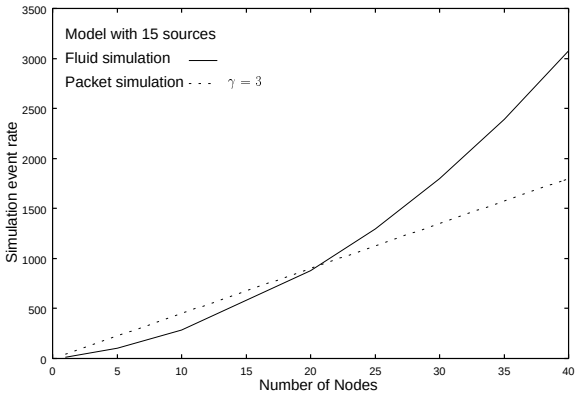

Fig. 7. Simulation event rate of a tandem system

be more efficient when the number of nodes is small or when γ, the packet rate, is large. Note that the fluid simulation does not depend on γ, enabling it to outperform its counterpart when γ is large. Other models with different number of sources entering each node also exhibit the same behavior.

In previous work, we derived analytical results for the tandem queueing system in order to compare the efficiency between packet-level and fluid simulation [4]. That analysis focused on studying the behavior of the tandem queue under different number of nodes in the system under simplifying assumptions. The analytical results predict the quadratic growth of the fluid simulation event rate, which is corroborated by the simulation results presented here.

The second important parameter is the number of flows entering each node in the tandem system. This parameter affects the amount of interaction among different flows which gives rise to the ripple effect. The scenario we consider here has 20 nodes with a constant load of 0.8 and $\lambda = \mu = 1$. The service rate of the queues is scaled to maintain a constant load for different number of sources. The packet-level simulation results are shown for $\gamma = 3$. Figure 8 presents the simulation results obtained for this model. Note that the fluid simulation event rate increases sub-linearly with the number of flows, while the packet-level simulation increases linearly. Again, the fluid simulation event rate does not depend on γ, and will eventually outperform the packet-level simulation. Note that the results obtained in this case are opposite to the ones obtained when varying the number of nodes. Here, the fluid simulation will eventually outperform the packet-level simulation due to its sub-linear increase. A possible explanation for the sub-linear increase in the event rate of the fluid simulation lies in the behavior of the ripple effect. Since the service rate is increased to maintain the load constant, the number of flows that can be serviced by the queue without interfering with each other increases. This can reduce the ripple effect, since less interference among flows will occur.

The tradeoff between these different parameters leads to the conclusion that the parameter space can be partitioned into regions such that, in each region, either fluid or packet-level simulation is always superior. A simple analysis will allow us to get a rough (approximate) idea of these regions. Even though the location of these boundaries may not be precise due to approxi-

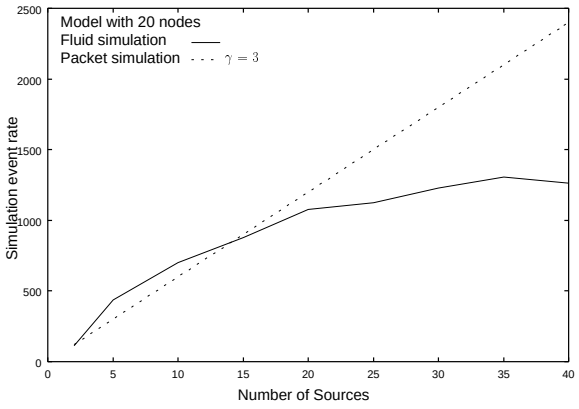

Fig. 8. Simulation event rate of a tandem system

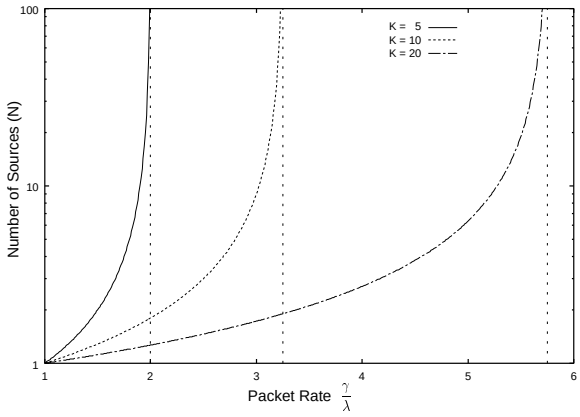

Fig. 9. Boundary condition for efficiency and regions between fluid and packet simulation

mations in the analysis, it is important to note that such regions do exist.

Let $R^f(K)$ and $R^p(K)$ denote the fluid and packet simulation event rates for a K node tandem queueing system. The events associated with the simulation of the sources are not considered, since these are identical for both paradigms. Based on the propositions in [4], the fluid event rate is given by:

$$R^f(K) = \frac{2\lambda\mu}{\lambda+\mu}\Big[NK + (N-1)\alpha_{00}\frac{(K-1)K}{2}\Big] \qquad (3)$$

where the term α_{00} is the probability that source 0 is in the on state.

The packet-level simulation event rate is determined by the number of packets sent by each source and the number of nodes each packet traverses. This leads to:

$$R^p(K) = NK \cdot \frac{2\mu\gamma}{\lambda+\mu} \qquad (4)$$

To investigate the relative efficiency of both simulation paradigms and determine a boundary condition, let $R^f(K) = R^p(K)$. Solving for N leads to:

$$N = \frac{\alpha_{00}\frac{(K-1)K}{2}}{\alpha_{00}\frac{(K-1)K}{2} + K - K\frac{\gamma}{\lambda}} \qquad (5)$$

If this equation is satisfied, then the fluid and packet simulation event rates are equally efficient. Figure 9 shows the N-γ boundary condition curves with different number of nodes K in the system. For a given value of K, the tradeoff curve carries the following interpretation: if the point (N_0, γ_0) lies to the right of the curve, fluid simulation is more efficient than its packet counterpart; otherwise, packet simulation is more efficient. We observe that increasing K shifts the tradeoff curve to the right, increasing the space in which packet simulation is more efficient (has a smaller event rate). The asymptotic line can be obtained from (5) and is given by:

$$\frac{\gamma}{\lambda} = \frac{\alpha_{00}}{2}K - \frac{\alpha_{00}}{2} + 1$$

The fluid simulation is always more efficient if γ/λ, the number of packets transmitted per on period, is greater than $(\alpha_{00}/2K - \alpha_{00}/2 + 1)$, regardless of the size of the tandem queueing system. This is expected since the event rate of fluid simulation does not depend on γ, while the packet-level event rate is proportional to γ. This result confirms the existence of boundaries of regions where one simulation paradigm (fluid or packet-level) is always more efficient than the other. Boundaries and regions based on other parameters such as N-K, were also be established and can be found in [11].

For more complicated feed-forward queueing networks, where different flows can traverse more nodes and interact with more flows, we conjecture that the ripple effect will be even more pronounced. However, the event rate should still be finite because a flow rate change traverses only a finite number of nodes in the network before it leaves the system. Thus the number of extra events caused by ripple effect is finite. Even these more complicated networks are expected to have well-defined regions in which either the fluid or the packet-level simulation is more efficient.

IV. Feedback queueing networks

In the previous section our analytical and simulation results for a tandem queueing network demonstrated that the event rate for a fluid simulation is finite for any stable feed-forward queueing networks. In this section we consider a network that has a cycle. We saw earlier that with fluid simulation a flow rate change can cause a change in the output rate of other flows, thus affecting other flows at downstream nodes. This "ripple effect" could be particularly worrisome in a cyclic network. A flow rate change can modify the rate of several other flows and all these rate changes can traverse the nodes in a cycle and spawn more rate changes, resulting in a cascade of flow rate changes. In such a scenario, the event rate of the simulation could grow without bound.

In this section we investigate the simulation event rate of a feedback queueing system using the cyclic queueing model, depicted in Figure 10. We consider the effect of different parameters on the simulation event rate. The relationship between the event rate and the corresponding execution time is also presented to support our use of the simulation event rate as the measure of computational effort in a simulation.

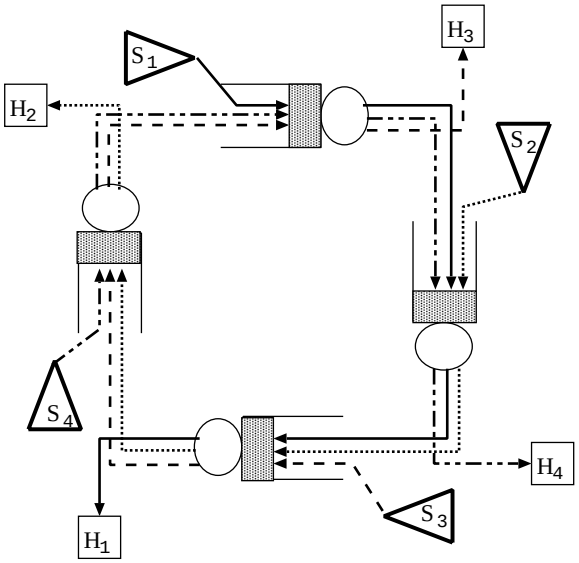

Fig. 10. A feedback queueing model

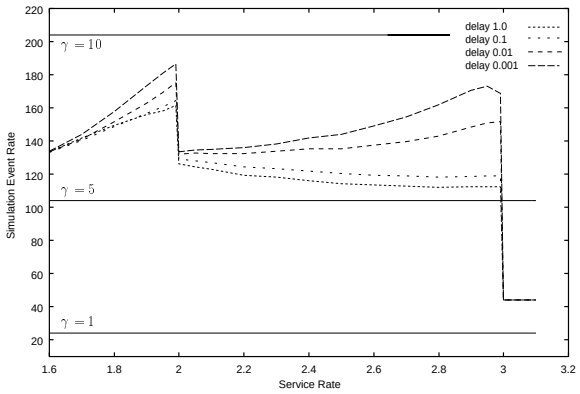

Fig. 11. Simulation event rate for the feedback queueing model

A. A feedback queueing model

The scenario illustrated in Figure 10 contains four identical FIFO queues with infinite storage capacity that are connected together to form a cycle. A single on-off source is injected into each of the queues. The routing of the flows is predefined so that each flow traverses exactly three queues before departing the system. In this setting, three different flows pass through each queue. The four traffic sources are identical, with equal on and off periods and the peak rates are set to 1. The stability condition of this system requires the service rate to be greater than 1.5, which is the average aggregate input rate at each of the queues. In order to evaluate this system under different loads, the service rate will be varied. We will see that another important parameter is the link propagation delay, which will also be varied, taking values of 0.001, 0.01, 0.1 and 1 time units. The simulation event rate and execution time were collected for service rates ranging from 1.6 to 3.1, which corresponds to system loads of 0.9375 to 0.5.

B. Results of fluid simulation

The fluid simulation event rate of the feedback queueing model is plotted in Figure 11, as a function of service rate. The four staircase curves correspond to four different link propagation delays. In the same figure, the event rates of the packet-level simulation of the same model are plotted, with different packet transmission rates (γ).

We first note that Figure 11 shows that the fluid simulation event rate of the simulation converges (is finite) for all different service rates and link delays. This can be explained informally by considering the dynamics of the FIFO queue. Note that an input flow rate change does not cause new rate changes among output flows at a queue if the queue is empty and, at the same time, the aggregate arrival rate into the queue is smaller than the queue's service rate. So even for a closed-loop queueing system, a flow rate change can only cause a finite number of new rate changes at "downstream" queues, if the probability that a queue is empty and the aggregated arrival rate is smaller than the service rate is greater than zero. If this condition holds, then a flow rate change will eventually arrive at a queue in which this rate change will pass through without causing rate changes in other flows.

This convergence condition is directly related to the service rate of the queues, which interestingly has two opposite effects on the simulation event rate. First, as the service rate increases, the average queueing delay decreases, making the fluid rate changes propagate faster through the closed loop. This effect tends to increase the event rate, since more fluid rate changes occur within a fixed time interval. On the other hand, a larger service rate implies that the queue is more likely to be empty and at the same time, the aggregate arrival rate is smaller than the service rate, resulting in a larger probability that a rate change will not cause a ripple effect. This effect tends to reduce the simulation event rate. It is the tradeoff between these two factors that determines the simulation event rate. From Figure 11, we see that in this model, when the service rate is below 2, the first factor dominates, and the event rate increases as the service rate increases. For service rates greater than 2, the second factor becomes more salient, leading to the flattening of the event rates.

Another important factor affecting the event rate in the closed loop system is the link propagation delay between adjacent queues. The smaller the link delay, the less time it takes for a rate change to propagate around the network. Smaller link delays will result in more flow rate changes in a unit of time, causing the overall simulation event rate to increase. Therefore, for a given service rate, the event rate increases as the link delay decreases.

Another interesting observation can also be noted in Figure 11. Each event curve exhibits two discontinuities at service rates of 2 and 3, where event rate drops dramatically. These discontinuities can be explained as follows. When a flow rate change arrives at a queue, if the queue has a backlog, the rate change will cause new rate changes in all the other flows with a non-zero flow rate. When the queue is empty, this only happens when the aggregated arrival rate is larger than the service rate. Therefore, the probability that a rate change causes a ripple effect is the sum of the probability that either the queue is not empty *or* the queue is empty but the aggregated arrival rate is now larger than

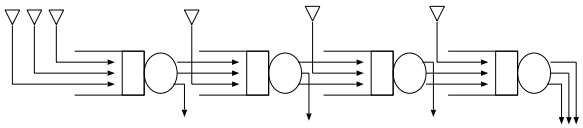

Fig. 12. A similar model with no feedback loop

service rate.

To explain the simulation event rate drop when the service rate is two, we consider two service rates, one slightly less than two, the other slightly greater. We denote these two service rates as $(2-\epsilon)$ and $(2+\epsilon)$, where $\epsilon \ll 1$. The probability that a queue is empty can be taken to be the same for these two cases. Now we consider the probability that aggregated arrival rate is greater than service rate. For the case when the service rate is $(2-\epsilon)$, the aggregated arrival rate is larger than the service rate when two or more sources are simultaneously on. In our model this probability is 11/16. For the case when the service rate is$(2+\epsilon)$, the aggregated arrival rate is larger than the service rate only when at least three sources are simultaneously on; the probability of this event is 5/16. The probability that a rate change causes a ripple effect is much smaller when the service rate reaches two from the left and this probability jump causes a major drop in event rate, which leads to the discontinuity. The discontinuity at service rate three can be explained via similar arguments.

To examine how a feedback system magnifies the fluid simulation event rate, we next analyze simulation event rates in the model illustrated in Figure 12, which does not contain a closed loop but otherwise is quite similar to the model in Figure 10. The number of queues remains the same but a link in the original model was removed in order to break up the loop. To preserve the degree of interaction at each queue, two extra sources were added to the first queue so that the number of flows passing through each queue remains the same. The purpose of this model is not to represent a feedback system but rather try to capture the impact that a closed loop has on the fluid simulation event rate. The simulation event rate for both models is presented in Figure 13. The link propagation delay in both models is 0.001, although this parameter has no impact on the feed forward model. It can be observed that the event rate for the closed loop model is considerably higher than for the open loop model. This is due to the fact that a rate change may traverse the closed loop many times, causing new rate changes in other flows. This result indicates that the events generated by looping around account for a large fraction of the overall number of events in the simulation.

For a cyclic queueing network, one would expect that more interaction among flows causes more ripple effects, resulting in a higher event rate. To show how different interaction levels affect the fluid simulation event rate, we extended the cyclic queueing model in Figure 10 to contain six nodes. In the simulation, we varied the number of queues a flow traverses, decreased the service rate to scale the load, and obtained the results shown in Figure 14. As expected, the event rates increases as the interaction level increases, i.e., a flow traverses more hops. The event rates also exhibit discontinuities and behaviors as a function of load similar to that for the four node model.

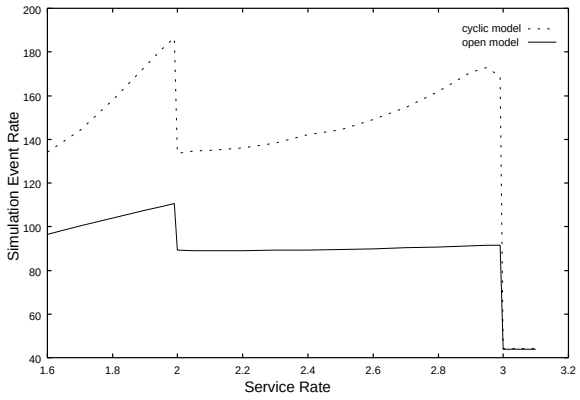

Fig. 13. Simulation event rates

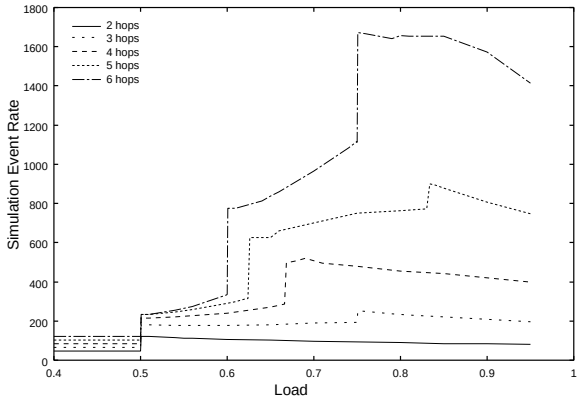

Fig. 14. Simulation Event rate for a 6-node cyclic queue

C. Results of packet-level simulation

A closed loop does not introduce any special consideration into a packet-level simulation model. A packet is simply generated at a source, traverses through three queues, and leaves the system without introducing any extra events to other packet flows. When the system is stable, the simulation event rate is independent of the service rate and link propagation delay. The packet-level simulation event rate for the model in Figure 10 is thus fully determined by the packet rate of the source and the number of nodes a flow traverses and is given by:

$$e^P = (e_s + e_q N_h) N_n \tag{6}$$

where e_s represents the event rate of a single source; e_q is the event rate of the queue and any other components between adjacent queues (i.e., link, router) per flow; N_h is the number of hops traversed by each flow; and N_n is the total number of nodes in the system. The simulation event rate at a node is the sum of the event rate of the sources and the event rate of components between adjacent nodes $(e_s + e_q N_h)$. The event rate of the model is just the sum of event rates of the N_n nodes.

This formula was verified with simulation results (not presented here), and was used to plot the event rates for $\gamma = 1, 5, 10$ in Figure 11. This yields a comparison between the two simulation techniques. Notice that the fluid simulation is more efficient if the packet transmission rate of the source, γ, is greater than 10. Note that for a on-off packet voice model, the typical on period

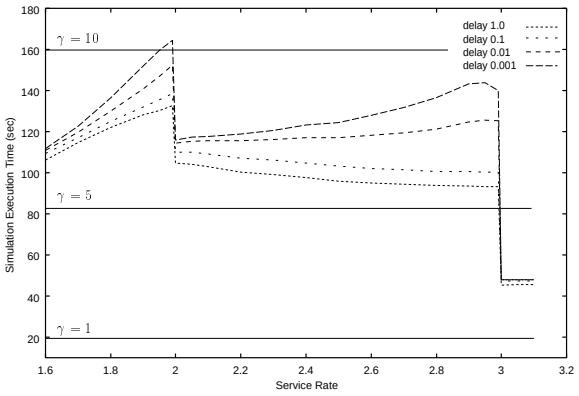

Fig. 15. Measured simulation execution time vs service rate

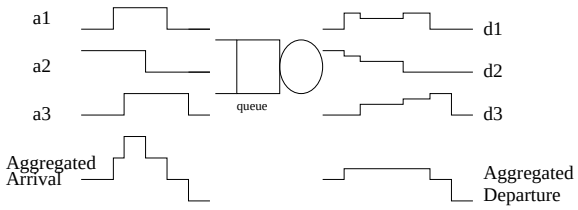

Fig. 16. Example of flow aggregation

averages between 0.4 and 1.2 seconds [12]. Study [13] suggests that a voice packet should be transmitted every 20 ms, which has been widely adopted by many voice applications tools, such as vat [14]. Using these parameters, a typical value for the packet rate (γ) is 30. For models having these parameters, the fluid simulation is likely to be more efficient.

D. CPU time versus event rate

In order to evaluate the appropriateness of our using the simulation event rate as a measure of computation effort, the execution times of the simulations were collected. In Figure 15, the simulation execution time corresponding to the event rate curves are presented. Comparing Figure 11, which plots event rate as a function of service rate, with Figure 15, which plots execution time as a function of service rate, we see that both exhibit the same behaviors. This suggests that the basic event rate defined in Equation 1 is a good measure of a simulation's computational cost and can be used to establish a comparison.

V. Flow Aggregation in Fluid Simulation

The analysis of the tandem queue model in Section III shows that the interaction between different flows can cause a ripple effect in a queueing network, greatly increasing the computational effort needed by fluid simulation when the network is large. A standard method for combatting this is to isolate the flow of interest and aggregate the remaining flows into a single flow. In this section we study the effect of flow aggregation on the simulation event rate for both packet-level and fluid simulation and discuss the implications.

In packet-level simulation, flow aggregation does not reduce the number of events simulated, since the simulator must still keep track of each individual packets being generated by the source. It does not matter if the packet belongs to the aggregated background traffic flow or to the flow of interest. In fluid simulation, however, a flow rate change can cause many rate changes in other flows - changes that occur at the same simulation time. If the flows are aggregated together, only one rate change will occur, representing the overall effect of the individual rate changes on the aggregate flow. Figure 16 illustrates the savings of flow aggregation. In this example, three flows (a_1, a_2, a_3) are fed into a FIFO queue. After the interaction among the flows at the queue, the corresponding departure processes is given by d_1, d_2, d_3, respectively. The aggregated arrival process and departure process are plotted below the individual flows. In the figure we observe that the number of rate changes in the aggregated arrival process is equal to the sum of the rate changes of all the individual arriving flows. Hence, no savings are obtained in the arrival process with source aggregation. However, the total number of rate changes in the departure process of the individual flows is 11, while, the aggregated departure process exhibits only 3 rate changes. There are two important factors that lead to the large event rate reduction. One is that a rate change in the aggregated flow may represent many simultaneous rate changes in the individual flows, which are counted as only a single event in the aggregate flow. The other factor is that after aggregation, all the individual flows become a single flow. Therefore, the probability of merging between fluid chunks increases.

To measure the improvements obtained with flow aggregation, we present an analysis of a model with a single queue being fed with multiple identical on-off sources. Since the sources make the same contribution to the overall simulation event rate in both the aggregated and non-aggregated scenarios, the following analysis only considers the event rate of the departure processes. The difference between the two cases would be solely due to the flow aggregation.

Consider a FIFO queue fed by N independent and identical Markovian on-off sources, the probability that a single source is in the on state is $\alpha = \mu/(\lambda+\mu)$ and the event rate of each source is $2\lambda\mu/(\lambda+\mu)$. Ignoring the merging of fluid chunks and other subtleties intrinsic to fluid simulation [4], the event rate of the departure process without flow aggregation, denoted by e^f, can be approximated by:

$$\begin{aligned} e^f &= \sum_{i=1}^{N} \left(\frac{2\lambda\mu}{\lambda+\mu} + \frac{2\lambda\mu}{\lambda+\mu}(N-1)\alpha \right) \\ &= \frac{2\lambda\mu}{\lambda+\mu}(1-\alpha+\alpha N)N \end{aligned} \tag{7}$$

The first term inside the summation is the event rate generated by the transitions of a source. The second term accounts for the rate changes in this flow caused by interference of other flows.

Suppose next that one is interested in the N-th flow and treats all the other flows as a single aggregated background traffic flow. In this case, each rate change of the N-th flow may cause one rate change in the aggregated flow, and each rate change in the aggregated flow may cause one rate change in the N-th flow. Let α' denote the probability that at least one source in the aggregated traffic in on; we have $\alpha' = 1 - (1-\alpha)^{N-1}$. Again, ignoring the merging of flow chunks and the events due to the special chunk effect, the approximation for event rate e_a^f is:

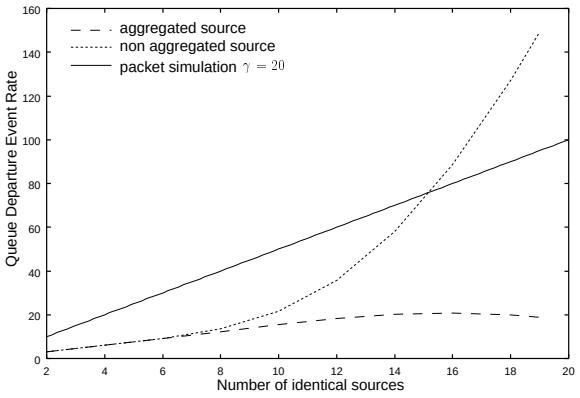

Fig. 17. Event rate of flow aggregation and no aggregation

$$e_a^f = \left[\frac{2\lambda\mu}{\lambda+\mu} + \frac{2\lambda\mu}{\lambda+\mu}(N-1)\alpha\right] + \left[\frac{2\lambda\mu}{\lambda+\mu}(N-1) + \frac{2\lambda\mu}{\lambda+\mu}\alpha'\right] \quad (8)$$

The term in the first bracket is the event rate of the N-th flow. The second term is the event rate of the aggregate flow, which represents the other $N-1$ flows.

The above analysis indicates that in the worst case, the event rate increases quadratically with the number of sources feeding the queue if the identity of each flow is preserved. With flow aggregation, the event rate becomes to increase linearly with the number of flows.

Simulation results presented in Figure 17 show exactly how flow aggregation can considerably reduce the simulation event rate. In this simulation, the sources are all identical, with $\lambda = \mu = 1$, and peak rate equal to 1. The service rate of the queue of the infinite capacity FIFO queue is 5. The two curves in the figure show how the simulation event rate varies as the number of identical sources increases from 2 to 19. Note that the load increases with the number of sources and the system becomes unstable with 20 sources.

We observe that in Figure 17 the event rate without flow aggregation increases quadratically with the total number of sources, as predicted by the analysis. The event rate with flow aggregation increases linearly at the beginning and eventually decreases after the number of sources reaches 12. The reason for this behavior is that as the number of flows aggregated together increases, the merging probability becomes larger and can further reduce the event rate. For real network models, the number of flows going through a queue is usually large enough that the flow aggregation can greatly reduce the event rate and the execution time of the fluid simulator.

The event rate of the packet-level simulation for the above scenario is just the aggregated packet transmission rate of the sources:

$$e^p = \frac{\mu}{\lambda+\mu} \cdot \gamma \cdot N \quad (9)$$

where γ is the packet transmission rate when the source is in the on period. Note that the first term represents the fraction of time the source is on.

While the ripple effect can induce a large number of extra events in fluid simulation, which can be prohibitive for large scale network, the flow aggregation technique provides an instrument to reduce the effect of interaction among flows, hence reducing the overall event rate. Also note that the event rate of fluid simulation with flow aggregation exhibits the same asymptotic linear behavior as packet-level simulation. If the packet rate (γ) is large, then the fluid simulation can outperform the packet-level simulation for tandem network models, which would not be true without the flow aggregation. An example of this potential is illustrated in Figure 17, where the aggregated fluid simulation outperforms the packet-level simulation with $\gamma = 20$, and the simulation without flow aggregation performs the worst.

VI. Weighted Fair Queueing in Fluid simulation

All the scenarios studied thus far have involved FIFO queues. The dynamics of FIFO queues directly cause the ripple effect, which can significantly increase the event rate of fluid simulation for a large network model. Weighted fair queueing (WFQ) [15] provides a certain degree of separation between flows from different sources, and we will see that this separation will significantly decrease the impact of the ripple effect in WFQ queues. A comparison between the fluid and packet-level simulation event rates of WFQ queueing network models is presented.

A fluid WFQ server is work conserving and operates at a fixed rate c. The server will thus be busy if there is fluid stored in any of its subqueues. A WFQ node consists of a set of K classes or subqueues, each having the FIFO policy and being fed by one or more different flows. The service rate is distributed among the classes according to a set of positive partition parameters $\{\phi_1, \phi_2, \cdots, \phi_K\}$. The details of the dynamics of a WFQ node can be found in [16].

In a WFQ node, if the service rates distributed to each subqueue is kept constant, a flow only interacts with those flows that merge into the same class, instead of all the flows arriving at the node. In this case, interference among flows that belong to different classes is eliminated by the isolation mechanism provided by the WFQ scheduling discipline. Less interference among flows in WFQ node reduces the ripple effect, and hence, the simulation event rate.

However, due to the work conserving policy of WFQ, the service rate allocated to each class can change. These changes occur when some subqueues require less service rate than their partitions while other subqueues can use more service rate than their partitions. The service rate of a class depends on the size of the corresponding FIFO subqueue, the fluid arrival rate, as well as the queue length of other classes. The dynamics of WFQ are illustrated in Figure 18. The weight for each queue is 0.5 and the service rate is c. Initially flow 1 grabs the entire service rate c. At time t_1 flow 2 turns on, and the two queues begin to share the service rate equally, resulting in a departure rate change for both flows at that moment. When flow 1 turns off at t_2 and its queue empties at a later time t_3, the service rate for flow 2 jumps to c, resulting in a fluid departure rate change in flow 2. This dynamic allocation of the service rate is called the "service rate redistri-

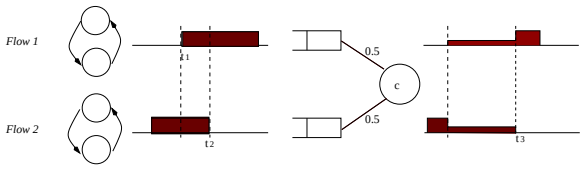

Fig. 18. A two queue weighted fair queuing system

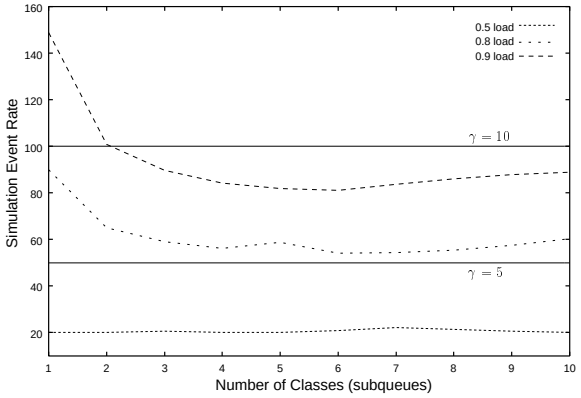

Fig. 19. Simulation event rate vs. the number of classes in a WFQ node

bution process". Therefore, a service rate change of a class can cause rate changes not only on all flows in its class, but also on flows currently being serviced in other classes. In contrast to the effect of flow isolation, the service rate redistribution process actually causes extra events which contributes to the fluid simulation event rate. However, for a moderately loaded WFQ node, we expect the impact of the service rate redistribution process not to be significant.

As mentioned above, the flow isolation and service rate redistribution process have opposite impacts on the fluid simulation event rate. The event rate for a WFQ node will depend on how the input flows are divided into different classes and how often the service rate is recomputed and distributed among classes.

Figure 19 shows the simulation event rate of a WFQ node as a function of the number of classes in the WFQ. In this simulation, 20 independent and identical on-off sources were used as input, with $\lambda = \mu = 1$. The peak rate while in the on state is $\gamma = 1$. The sources are equally allocated to the classes, so that each class is fed by the same number of sources. The partition of the service rate is distributed equally among the different classes. The system load was varied by changing the service rate, and loads of 0.5, 0.8. and 0.9 were investigated.

Note that when the number of classes is one, the WFQ node behaves exactly the same as a single FIFO queue. This corresponds to the leftmost points in Figure 19. If we choose the load of 0.9, the event rate of the fluid simulation decreases by a factor of 32% when a single class FIFO is replaced by a 2-class WFQ node. The event rate further decreases as the number of classes in the WFQ node increases, since increased isolation among different flows diminishes the ripple effect. However, as the number of classes increases past 7, the event rate due to the service rate redistribution starts to become significant, increasing the overall simulation event rate. Note that for all loads, the event rate to simulate the WFQ node with any number of classes, is always less than the event rate to simulate a single FIFO node with the same input flows.

As expected, decreasing the load will lead to a higher probability that the fluid flowing from the sources will pass through the node without queueing. Hence, less ripple effect is incurred and the overall fluid simulation event rate is smaller. Note that for the load of 0.5, no queueing takes place, since the service rate is equal to sum of the peak rate of all the sources. Therefore, the number of classes has no impact on the fluid simulation event rate, as observed in Figure 19. The event rate in this case is given solely by the behavior of the sources and can be computed analytically. Thus, the fluid simulation of a WFQ node is likely to be more efficient, in terms of event rate, than the fluid simulation of a FIFO node, specially at higher loads.

For packet-level simulation, the event rate of the WFQ node is given by the rate at which packets arrive at the node, which is determined by the rate at which packets are generated by the sources. The isolation between classes in WFQ does not reduce the packet-level simulation event rate. Moreover, the packet-level simulation event rate in this model can be easily computed using equation (9). The comparison between fluid simulation and packet-level simulation still depends on the tradeoffs of the network and traffic rate parameters, especially γ. In Figure 19 the packet-level event rate for $\gamma = 5, 10$ are presented to illustrate the tradeoffs. Note that fluid simulation can outperform packet-level simulation for some parameter configurations. A quantitative analysis of the WFQ fluid simulation event rate, to establish a better understanding of the parameter space and tradeoffs, is still an ongoing research topic.

VII. Conclusions

In this paper, we evaluated the relative performance of fluid simulation over packet-level simulation by analyzing several different networking scenarios. The simulation event rate was used as the basic measure of comparison between the two simulation techniques. The results obtained show that the simulation execution time is proportional to the simulation event rate, making this measure both adequate and accurate.

An important issue is the impact of the ripple effect on the fluid simulation event rate. This characteristic of fluid simulation is a major contributor to the simulation event rate, particularly, in large and complex models. The ripple effect is intrinsic to fluid simulation and does not appear in packet-level simulation. In this case, the event rate is fully determined by the rate at which packets are generated and their routes inside the network model. Therefore, the relative efficiency between the two simulation approaches depends critically on the impact of the ripple effect and the rates at which packets are generated.

We have derived analytical results to characterize the simulation event rate for the tandem queueing model. For all the scenarios investigated, we have provided simulation results and discussed the trends of the event rates. We have shown the tradeoffs between parameters of the models, which determines the most efficient simulation technique. In general, fluid simulation outperforms packet-level simulation when the packet rate (γ) is large. We have also considered two methods that can reduce the impact of the ripple effect, namely, flow aggregation and WFQ policy. A fluid simulator and a packet-level simulator was designed and implemented in order to explore their functionalities

and validate analytical results.

An alternative approach to improve the performance of fluid simulation is the time-driven fluid simulation scheme [17]. In this approach, the continuous fluid flow is discretized into fixed length of fluid chunks, where each chunk has a constant rate and represents many packets. This discretization is done throughout the network being simulated. The simulation advances at the granularity of the discretization, which can be adjusted to accommodate different resolution requirements in modeling. Flow rate changes within a time step are averaged out, reducing the number of rate changes resulting from the ripple effect. Also the uniform discretization interval facilitates parallelism in the simulation. This scheme is undergoing both theoretical and experimental investigation.

REFERENCES

[1] M. Villén-Altamirano and J. Villén-Altamirano, "RESTART: a straightforward method for fast simulation of rare events," in *Proceedings of the 1994 Winter Simulation Conference*, Lake Buena Vista, Florida - USA, Dec. 1994, pp. 282 – 289.

[2] Jong Suk Ahn and Peter B. Danzig, "Packet network simulation: Speedup and accuracy versus timing granularity," *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, pp. 743 – 757, Oct. 1996.

[3] D. Anick, D. Mitra, and M.M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *The Bell System Technical Journal*, vol. 61, no. 8, pp. 1871 – 1894, Oct. 1982.

[4] B. Liu, Y. Guo, J. Kurose, D. Towsley, and W. Gong, "Fluid simulation of large scale networks: issues and tradeoff," in *PDPTA '99*, Las Vegas, NV, June 1999, pp. 2136 – 2142.

[5] David Nicol, Michael Goldsby, and Michael Johnson, "Fluid-based simulation of communication networks using SSF," in *Proc. 1999 European Simulation Symposium*, Erlangen-Nuremberg, Germany, Oct. 1999.

[6] D. Ros and R. Marie, "Estimation of end-to-end delay in high-speed networks by means of fluid model simulations," in *13th European Simulation Multiconference*, Warsaw, June 1999.

[7] D. Ros and R. Marie, "Loss characterization in high-speed networks through simulation of fluid models," in *SPECTS'99*, Chicago, IL, USA, July 1999.

[8] G. Kesidis, A. Singh, D. Cheung, and W.W. Kwok, "Feasibility of fluid-driven simulation for ATM network," in *Proc. IEEE GLOBECOM 96*, Nov. 1996, vol. 3, pp. 2013–2017.

[9] S3 Consortium, "Scalable simulation framework API reference manual," Documentation Draft, Jan. 1999.

[10] J. Cowie, D. Nicol, and A. Ogielski, "Modeling the global internet," *Computing in Science & Engineering*, pp. 30–38, 1999.

[11] Yang Guo, *On Fluid Modeling of Networks and Queues*, Ph.D. thesis, University of Massachusetts, Amherst - MA, 2000.

[12] Mischa Schwartz, *Broadband Integrated Networks*, Prentice Hall, 1996.

[13] N. S. Jayant, "Effects of packet loss on waveform coded speech," in *Proc. Fifth Int. Conference on Computer Communications*, Oct. 1980, pp. 275 – 280.

[14] Van Jacobson and Steven McCanne, "URL http://www-nrg.ee.lbl.gov/vat," Lawrence Berkeley National Laboratory - University of California, Berkeley.

[15] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," *Internet Res. and Exper.*, vol. 1, 1990.

[16] Abhay K. Parekh and Robert G. Gallager, "A generalized processor sharing approach to flow control in integrated services network: The single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344 – 357, June 1993.

[17] A. Yan and W.B. Gong, "Time-driven fluid simulation for high-speed networks with flow-based routing," in *Proc. of the Applied Telecommunications Symposium*, Boston, MA, Apr. 1998, pp. 153 – 158.