Time-Adaptive Numerical Simulation for High Speed Networks

Suman Kumar, Seung-Jong Park, S. Sitharama Iyengar
Computer Science and
Center for Computation and Technology
Louisiana State University
Baton Rouge, LA, USA
{suman,sjpark,iyengar}@csc.lsu.edu

Jung-Han Kimn
Mathematics and
Center for Computation and Technology
Louisiana State University
Baton Rouge, LA, USA
kimn@math.lsu.edu

***Abstract*—As the bandwidth of networks is increasing exponentially, the computational cost to simulate such type of networks is also growing in a similar fashion. This paper presents a scalable simulation method, called time-adaptive numerical simulation, which can be used to represent dynamics of high speed networks using fluid-based models. The new method dynamically adjusts the size of a time step for a numerical solver which solves a system of differential equations representing dynamics of protocols and nodes' behaviors. The simulation results show that the time-adaptive method reduces the computational time while achieving the same accuracy compared to that of a fixed step-size method.**

I. Introduction

Currently, computer networks are growing exponentially in terms of size as well as bandwidth. To serve the ongoing bandwidth requirement and scalability of these networks, there has been a continuous evolution of different protocols for large scale and high speed networks [1], [2]. To satisfy the high speed requirement, several protocols have been suggested by researchers namely Fast TCP [3], HSTCP [4], S-TCP [5], BIC-TCP [6], HAMILTON-TCP [7] on high speed networks. Since networks of these types are not publicly available, a network simulator can be a much simpler way of performance evaluation of different protocols. Hence, the analysis and evaluation of different protocols using different scenarios posed a research challenge on designing a network simulator suitable for high speed networks.

It is well known that packet-based simulators like NS2 [8] and Opnet [9] cannot be used in the case of large simulations [10]. Among the main research direction in the simulation of TCP/IP networks, we would quote parallel simulation projects, such as SSFnet [11] and emulation projects such as NistNet [12]. A current packet level simulator takes a large cpu-time and memory to simulate a single link of 10Gbps bandwidth for 10000 seconds and this requirement of cpu-time and memory increase with the increase in total simulation time and the number of nodes.

Fluid simulation (through approximation of fluid dynamics) is based on a path wise description of the dynamics of the interaction between flows which takes into account discrete event phenomena that are of central importance for drop-tail queues at routers/links, such as congestion epochs, losses, synchronization of sources etc. Although it satisfies the simulation of large number of flows with low bandwidth, the current method is still far behind the high speed networks which involves a few flows. Several fluid models [13], [14], [15] have been suggested to evaluate the Additive Increase and Multiplicative Decrease (AIMD) algorithm of TCP with the parameters related to AQM router policies and different network parameters namely link-bandwidth and delay. For the AQM policies, the packet delay corresponding to each packet is very much dependent on the bottle-neck queue attached to that link. As the bandwidth increases, the queuing delay decreases. To record the packet delay and the corresponding packet drop for a short period of time in TCP, we have to decrease the time step which in turn increases the execution time of the entire simulation. In the case of high speed networks (of the order of 1Gbps), we have different scenarios and an ideal scenario includes lesser number of high speed flows. The current research work focuses on such a scenario and uses input/output structure of fluid model for the execution time reduction.

The paper is organized as follows: in section II, we discuss the motivation and challenges in this area. Section III gives a brief overview of related work. In section IV, we discuss a simple system model and idea behind this work. Section IV introduces a general framework proposed by

This work was funded in part by NSF grant (CNS-0509645) and Louisiana Board of Regents grant (LEQSF2006-08-RD-A-08).

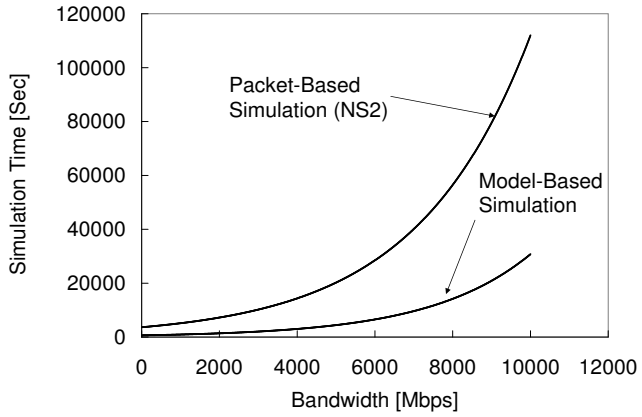

Fig. 1. Scalability as a Function of Network Bandwidth: Packet-Based Simulation Result from NS2 Simulator and Model-Based Simulation Result from a Fluid Level Simulator Using a Numerical Method, such as Euler Method.

us. In Section VI, the performance results and the validity of the proposed fluid simulator are mentioned. The conclusion and the possible future research direction can be found in section VII.

II. Motivation

A. Problem

Nowadays, there have been several research initiatives which develop and deploy high speed networks over major research institutions. These networks, such as NLR (National Lambda Rail) [1], LONI (Louisiana Optical Networks Initiative) [2], etc have bandwidths greater than 10 Gbps. By the virtue of DWDM (Dense Wavelength Division Multiplexing) technology [16] at link layer, a pair of optical fiber can transmit 30-40 simultaneous light waves of which transmission bandwidth is 10 Gbps so that we can achieve up to 400 Gbps of bandwidth at each physical link. To catch up with those large bandwidths at the physical layer, many researchers have developed protocols at higher layers, such as transport and network layers.

To develop those protocols, it is necessary to simulate the behavior of networks in order to evaluate the performance before implementing and deploying networks. Until now, development of small or mid-scale networks of which bandwidth is less than 1 Gbps has been supported by a packet-based simulation which emulates detailed behaviors of packets or queuing theory. However, as the scale of networks increases, the execution time of the packet-based simulation methods increases exponentially due to large number of packets to process. To overcome the scalability problem, we need to develop a new model-based simulation framework which takes less amount of simulation time and uses parallel computation. For the model-based simulation, we can use an approach based on fluid dynamics which represents a behavior of an individual flow in networks.

Figure 1 shows the execution time between the two simulation methods as a function of network bandwidth. The packet-based simulation experiences an exponential increase in its execution time as its bandwidth expands more than 1Gbps. However, in a fluid-based simulator, for bandwidths less than 10Gbps, the execution time is still reasonable.

Therefore, in this paper, we develop a fluid-based simulation method which predicts the behavior of transport and routing protocols over high speed networks within less amount of execution time.

B. Challenge

As shown in Figure 1, the execution time of fluid-based simulation is reasonably below that of packet-based simulation. However, its execution time is still more than several hours. Furthermore, the execution time will increase as more number of flows are getting involved in simulation. Therefore, it is necessary to develop a new fluid-based simulator which is scalable to the size of networks as well as network bandwidth.

In general, the fluid-based simulation method solves a set of Ordinary Differential Equations (ODE) which represent the dynamic behaviors of flows in networks using numerical methods, such as Euler method and Runge-Kutta method [17]. When the numerical methods solve a set of ODE, they use a time step-size, h, which is a step-size of a solution for Equation 1,

$$\frac{dy(t)}{dt} = f(t, y(t)) \tag{1}$$

to obtain a numerical estimate y_{k+1} of the Euler method.

$$y_{k+1} = y_k + h * f(t, y_n) \tag{2}$$

The numerical solver for a set of ODE is a time-stepped fluid-based network simulator. The accuracy of a solution is determined by the time step-size, h, and the network bandwidth because higher bandwidth creates more finer events in terms of time, such as packet departures and arrivals at nodes. For example, to represent the behavior of flows going through a link below 100 Mbps which roughly transmits 10^4 packets per second[1], 10^{-4} second of time step-size is a minimum time step to catch

[1]For the convenience of calculation, we assume that the size of packet is 10,000 bits

interesting events, such as packet arrival and departure, with no loss of information in the numerical sense. However, in case of 10 Gbps bandwidth networks, the step-size should be 10^{-6} second to solve equations without any loss. Therefore, simulation of high speed networks requires a shorter time step-size as the bandwidth of high speed networks increases.

Clearly, for a given link capacity say C_l, the queue service time for each packet say t_{st}. The minimum time step is such that it should capture this event. Hence, we can argue that at the interesting events we should atmost have this size of time-step to create a good snapshot of the network behavior. Therefore, the minimum step size dt_{min} is given as

$$dt_{min} = t_{st} = \frac{1}{C_l} \tag{3}$$

As the numerical simulations have a shorter time step-size, the total number of time steps increases. Since the computational time of the fluid-based simulation is proportional to the number of time steps, the execution time of the fluid-based simulation for high speed networks (e.g., more than 10Gbps bandwidth) is more than hundreds of thousand seconds (See Figure 1).

To reduce computational time for the fluid-based simulation, this paper presents a time-adaptive method which adjusts time step-size based on the dynamics of flows so that it can reduce the total time steps while achieving similar level of error.

For example, the Euler method has accumulated errors, such as $e_{k+1} = y_{k+1} - y(t_{k+1})$ and $e_k = y_k - y(t_k)$, can be represented as

$$e_{k+1} = e_k + h_k(f(t_k, y_k) - f(t_k, y(t_k))) - O(h_k^2),$$

where y_{k+1} is an estimate and $y(t_k)$ is an exact solution($t_{k+1} = t_k + h$) [17].

Therefore, the accumulated error is dependent on propagational error e_k and the time step-size h_k at each time step k. This paper will propose a new algorithm which changes the step-size h_k based on the error sensitivity while maintaining similar level of error.

For the convenience of calculation, this paper will use the Euler method as a basic method. However, the proposed time-adaptive method is still applicable to any higher order ODE solver.

III. Related Works

Fluid models give the basis of fluid level simulator and several fluid models were later introduced by researchers. Mishra [13] developed a methodology to model the TCP AIMD algorithm and obtained the expected transient behavior of networks with Active Queue Management routers supporting TCP flows. They used jump process driven Stochastic Differential Equations (SDEs) to model the interactions of a set of TCP flows and Active Queue Management routers (RED queue) in a network setting. The derived SDEs are transformed into a set of Ordinary Differential Equations (ODEs) which can be easily solved numerically. AIMD algorithm. The formulation presented in this work is quite general purpose and helpful in formulating the model for other TCP variants and also to analyze the other AQM mechanisms. Further extension of this work is presented in [14]. In their paper, the solution techniques have been presented to reduce the simulation time by simulating and solving only the queuing equations for potential congested links. This way the computation time has greatly reduced. Although the solution methodology scales well to a large number of flows with low bandwidth but no attempt has been made for reducing the computation time for the high speed network case.

In [15], Baccelli proposed a simplified representation of interacting the TCP flows via coupled evolution equations for simulating large IP networks at flow level. The basis of this approach is the joint evolution of the congestion window size of long lived (FTP type) flows controlled by TCP and sharing a single drop-tail router, in the network. The modeling is done in terms of sending rate of the source, giving the instantaneous throughput fluctuations at any point of time. The important aspect of synchronization rate has been explored effectively giving more realistic simulation results as compared to previous packet level simulator. The results obtained by this flow level simulator take into account key packet level phenomena such as the reaction delay, the scheduling and the buffer overflows, via the estimate used for the synchronization rate. In the paper they did not suggest any algorithm for further reduction of simulation with the use of system/model properties.

In [18], time-driven fluid simulation is proposed to simulate high speed networks. Here, the network elements are modeled as fluid servers where as the traffic source can be of arbitrary type, including a discrete-event and fluid source. Furthermore, usefulness of the fluid simulation with packet simulator has been explored in [18] where a hybrid method is used. Fluid models are used to represent aggregations of flows for which less detail are required and packet-level models are used to represent the individual flows for which more details are needed.

IV. Main Ideas

A. Fluid Model

In this section we are introducing the system model[2] where we summarize the fluid model equations used and the basic idea of the algorithm used to solve that fluid model for high speed network with lesser number of flows. The network is modeled as a directed graph $G = (V, E)$ where V denotes the set of routers and E is a set of links connecting those routers. Each link in E is served at a rate of C_l bps. Each link is associated with an AQM policy which is characterized by a packet drop probability $p_l(t)$, and which depends on the state of the queue associated with that link. Each link is also associated with a propagation delay for which the traffic departing from the queue associated with l arrives at the next queue after the propagation delay characterized by that link. Modeling of Advance Queue Management policy is done in such a way that any packet is discarded whenever the queue size exceeds the limit set by maximum queue limit value with probability 1, giving the drop-tail behavior in our case. It is evident that in this case, we are not distinguishing between the two identical flows which have the same path from the source to the destination. All the flows experience the same delay which can be given by the summation of the propagation and the link delay from the source to the destination associated with their path.

Without loss of generality, some of the frequently used notations in their generic form are listed here for easy references:

- F_i = A set of ordered queues traversed by the i^{th} flow in forward manner
- R_i = A set of ordered queues traversed by the i^{th} flow in backward manner (for the acknowledgment from destination)
- $W_i(t)$ = Congestion Window for i^{th} flow at time t
- $R_i(t)$ = Round Trip Time for i^{th} flow at time t
- M_i = Maximum Congestion Window limit for i^{th} flow
- $\lambda_i(t)$ = Loss Indication Rate for i^{th} flow at time t
- $q_l(t)$ = Queue Size associated with l^{th} link at time t
- C_l = Service Capacity/Bandwidth for l^{th} link
- $p_l(t)$ = Packet Drop Probability at l^{th} queue at time t
- C_l = Service Capacity/Bandwidth for l^{th} link
- a_l = Propagation Delay associated with l^{th} link
- q_l^{max} = Maximum queue size associated with l^{th} link
- n_l = Denotes number of flows traversing l^{th} link
- $A_l^i(t)$=Arrival Rate of i^{th} flow at l^{th} link at time t

[2]The basis of current work is the fluid Simulation framework given in [14], which accounts for shaping of the flow and delaying as they traverse through different links in the network

Basic equations in the form of linear and ordinary differential equations governing the flow level behavior of the network are summarized as below:

Window Size:

$$\frac{dW_i(t)}{dt} = \frac{1(W_i(t) < M_i)}{R_i(t)} - \frac{W_i(t)}{2}\lambda_i(t) \tag{4}$$

Here, $1(W_i(t) < M_i)$ is indicator function, which has binary output. If the argument is True its value is '1' and 0 otherwise. Its associated with the window function to limit the window from going beyond the maximum allowed value, hence when congestion window value reaches maximum there is no further increment in it.

Queue Size:

$$\frac{q_l(t)}{dt} = -1(q_l(t) > 0)C_l + \sum_{i=1}^{n_l} A_l^i(t) \tag{5}$$

Similarly queue size is restricted by the indicator function $(q_l(t) > 0$ and can have only positive value.

Round Trip Time:

$$R_i(t) = \sum_{l \in F_i U R_i} a_l + \frac{q_l(t)}{C_l} \tag{6}$$

Loss Indication Rate:

$$\lambda_i(t) = \sum_{l \in F_i} A_l^i(t) p_l(t) \tag{7}$$

Packet Drop Probability(For Drop-tail queue):

$$p_l(t) = \begin{cases} 0, & q_l(t) < q_l^{max} \\ 1, & q_l(t) > q_l^{max} \end{cases} \tag{8}$$

The above equations can be used to represent the entire system as a feedback system, where the loss event is used as a feedback mechanism and consecutively the AIMD adapts the congestion window to avoid the loss event by decreasing its size.

B. Main Idea

The model introduced in the last subsection is used to give network statistics for the network equipped with drop-tail queue and TCP AIMD congestion window algorithm at the transport layer.

In Figure 2 and Figure 3, shown the behavior of congestion window behavior and loss indication rate, $\lambda_i(t)$ as a function of time. Our algorithm is based on the fact

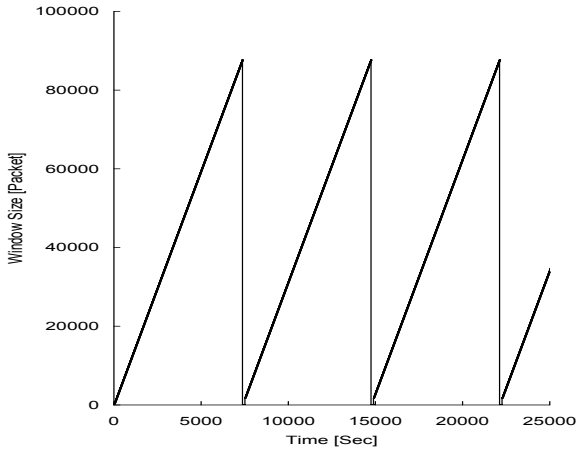

Fig. 2. Congestion window Behavior at High Speed Optical Network and 70ms Delay

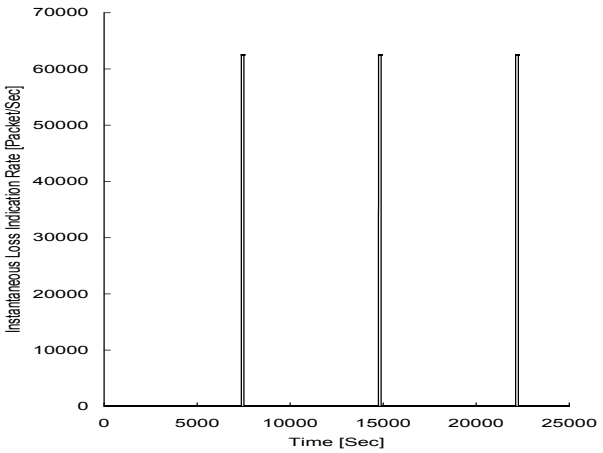

Fig. 3. Loss Indication Rate at High Speed Optical Network, 10Gbps and 70ms Delay

that the congestion window value changes with loss indication rate. As we observe from the Figure 2, the additive increment continues till we have a loss event due to the queue overflow as indicated by loss indicator function λ as shown in Figure 3. The queue overflow causes packet loss at that queue which causes multiplicative decrease in terms of λ as the packet arrival rate is quite high during that time. This can be easily understood by observing Equation 7 and Equation 4. An interesting thing to observe that before the loss event at the drop-tail queue (the packet drop probability at this queue is given by Equation 8), the window increases linearly and hence a linear solution is preferred, whenever the loss event occurs the congestion window gives a non-linear behavior and the decrease part of the AIMD comes into play.

For the case of high speed network, the loss event lasts for a very small time as compared to linear increase time, which is evident by the fact that we are working in a high bandwidth and large delay case, and so the RTT is large(Equation 6). Hence it takes a longer time (the increment part of the Equation 4) to reach to the peak value where it overshoots causing a loss event on the queue. At this point, the window value decreases by a factor of $\frac{W}{2}\lambda$(multiplicative decrease part in Equation 4). After the loss event, when the congestion window value is low, the queue is cleared immediately with the service rate C_l as in Equation 5 and remains empty until the window reaches its peak again. In order to record this fast decrease and recovery of congestion window from the loss event, we must have microscopic behavior of the network and to observe this behavior, we have to rely on a smaller time step as compared to the time where there is no loss event. Since the usual method is employed with constant time step value and as discussed in the section II, this constant value should be low enough to observe the network statistics for the high speed scenario. Prior to the loss event, we don't need to simulate the network with a smaller value of time step to solve the differential equation governing the network statistics as introduced earlier in this section. We can still do better with some reasonable value of time-step for no loss case as the only equation we deal with is linear, whereas, in the case of loss event we must decrease this value to observe minute details.

Our algorithm is based on this fact that we can use different values for no-loss and loss case. We suggest that whenever the congestion loss occurs, we should decrease the value of time step and must continue with some larger value of time step for the no-loss case.

V. Framework

In this section, the pseudo code of the algorithm is explained in Figure 5 and the corresponding flow chart of the algorithm has been introduced in Figure 4.

In the above figure 5 we have described the algorithm. first the values of dt_{min} and dt_{max} have been set as an input parameters. Further the decision as to which one out of two values should be used is made based on loss information. Since, we are dealing with the feedback system, the source window gets the loss signal after some time which is determined by the queuing and propagation delay in its path. In our algorithm this is determined beforehand from the signal received directly from the queue in the form of loss rate indicator variable λ. Whenever the value of λ any queue associated to any flow is nonzero, the algorithm switches the time step to minimum and records the window behavior for that short duration of time in which the loss event occurs. Also, before reaching to the source of that flow, the use of congestion signal for adjusting the time step is justified to minimize the possible error

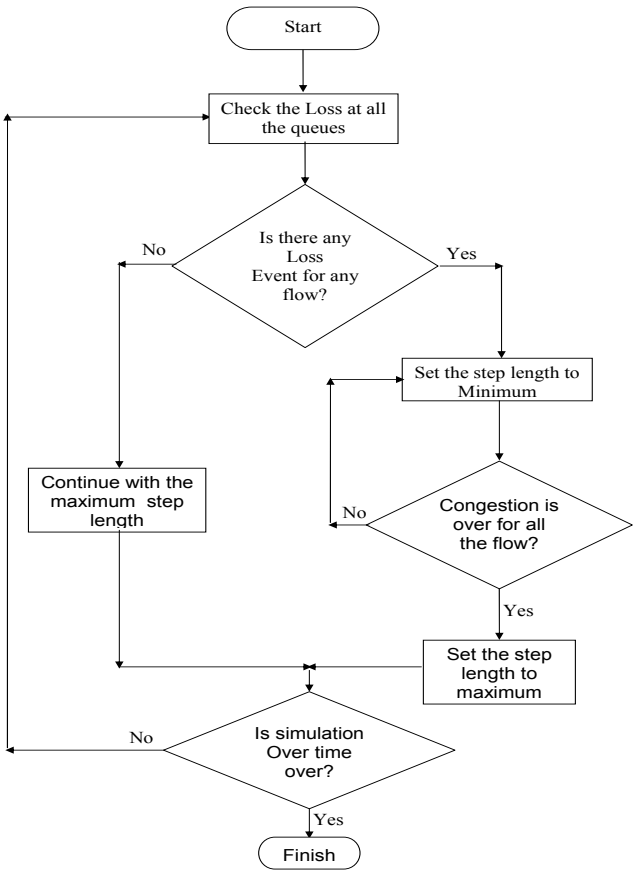

Fig. 4. Flow Chart Showing Loss Signal Flow and Decision Making for Step Length Adjustment

as much as possible. The idea is to adapt the time step to a suitable value as soon as possible, so that, we can track the network behavior accurately. We find that the congestion signal for each flow is the best method to judge the same. We will justify this argument in our next section with the support of various results which we have derived using our method and also compared with the regular fluid model solver simulation(FS) method without adapting time step length. Furthermore, the data flow from various level in the network and decision making for the step length adjustment is shown in the figure 4.

VI. Performance Evaluation

All simulations were carried out on a workstation which has dual Xeon-3GHz processors, 2GB RAM on PC 2700 board. To present our result, we are using a dumb-bell topology as shown in figure 6 to simulate an ideal scenario for a bottleneck link shared by two flows. Every link in this topology is equipped with a drop-tail queue with a maximum queue size of 500. Drop-tail queue management policy is used with a maximum queue size of 500

Define:

1 dt=Time Step Length for Simulation
2 λ=Loss Indication Rate
3 $RunTime$=Current network time in Simulator
4 $SimTime$=Time for which simulation to be run

Start:

//to start with maximum Time Step Length
5 $dt = dt_{max}$;//to start with maximum Time Step Length
6 Repeat step 7 to step 10 for every dt interval
while(RunTime$\leq$ Semitone)
7 If (for any $\lambda! = 0$)
//Set Time Step Length to minimum for any loss event
8 set $dt = dt_{min}$
9 else
//Set Time Step Length to maximum for no loss case
10 $dt = dt_{max}$

End:

Fig. 5. Adaptive Timestep Fluid Simulation Algorithm

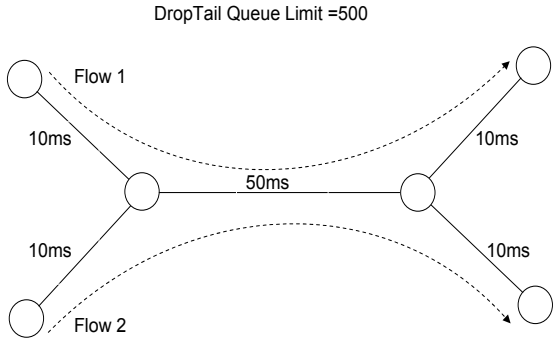

Fig. 6. Topology Used for the Simulation Showing Bottleneck Link and Two Flows

packets. The packet size is kept fixed at 1000B. The delay, as shown in the figure, is 50ms for bottleneck link and 10ms for the edge links.

A. Accuracy Validation

Since our method is based on the existing fluid model solver (FS), we compare our time-adaptive fluid solver(AFS) with the normal fluid solver(FS) which does not have a time adaptation mechanism. While solving using AFS, we start with the maximum value of time-step which we set as 0.001 and we suitably vary the minimum value keeping the minimum value for normal fluid simulation as we increase the bottleneck link bandwidth. In the figure 7, the comparison of congestion window behavior between AFS and FS has been shown for 5Gbps case, and it gives a good match. Between FS and AFS, the cor-

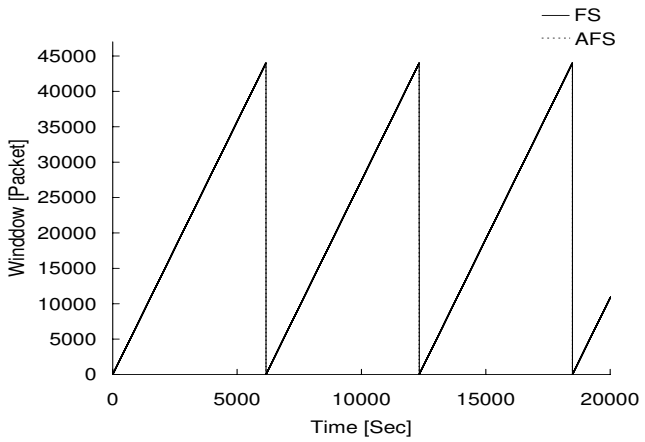

Fig. 7. Comparison of Congestion Window Using AFS and Fluid Solver(FS) for 5Gbps link

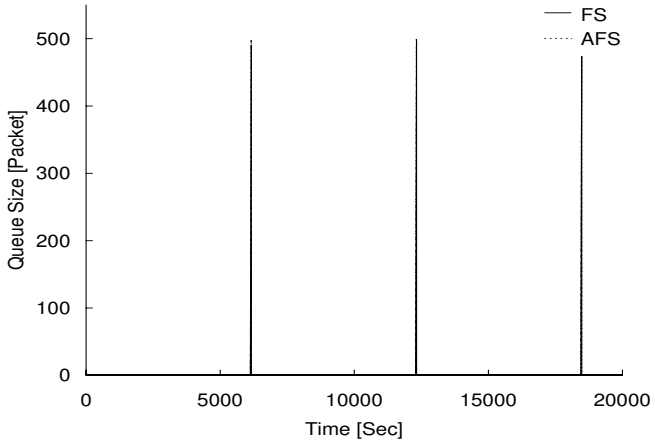

Fig. 8. Comparison of Bottle-Neck Queue Using AFS and Fluid Solver(FS) 5Gbps link

responding queue size behavior is presented in Figure 8 which in spite of a very little mismatch shows a good accuracy. As we know smaller the stepsize the more accurate it is the solution. Here we are showing how our accuracy depends on the change of step size. In the figure 9 it is clear that smaller the step length smaller the error. For the case of 10^{-3} we have larger error as compared to smaller step length.

B. Comparison

The Figure 10 shows the execution time comparison between the three simulation methods: NS2, FS and AFS. The NS2 simulation was not able to complete for the entire 20,000 sec due to a limitation in the number of packets sent and hence we had to scale it. For these simulators to reach a congestion window limit, the simulations should be carried out for a long time in the high speed case. Since we ran the simulation for the bandwidth ranging from 100Mbps to 10Gbps, with the network parameters that we have used, the 10Gbps case shows its first

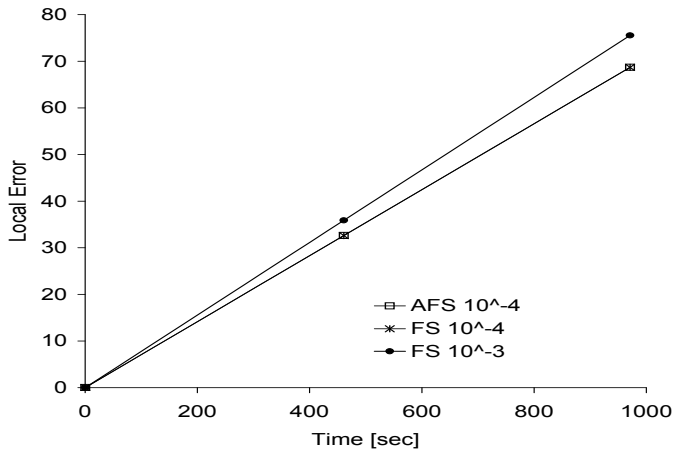

Fig. 9. Comparison of Errors for Different Step-length with respect to Step Length 10^{-6}

loss event at around 17,000 second. Hence to have a fair comparison between these methods for direct bandwidth, the simulation should be carried out for 20,000 seconds. To accommodate all the values in the graph, a log scale has been used. As we can see in the figure, the variation shows that FS achieves quite a good improvement over the packet simulator NS2. The execution time in the case of NS2 increases exponentially as the bandwidth increases and overshoots in the case of 10Gbps case. We observe that tradition fluid simulator(FS) method, although it is using a differential equation solver in its core, its execution time increases because of lower step length for the higher bandwidth case. As compared to the NS2 and FS method, our method achieves good improvements because it uses two different step lengths for the simulation. Further, as we can see our simplified method uses fixed minimum and maximum step length, which accounts for a slight decrease in the execution time, as there are lesser number of loss events in the case of higher bandwidth. We can also see in Figure 10 that the improvement through our method varies from 5 to 80 times as the bandwidth increases from 500Mbps to 10Gbps leaving NS2 far behind.

In the figure 11, we have shown the memory utilization for different methods. As we can see, the memory utilization is larger in the case of NS2 and still increases for FS too. The increment in non-swapped memory used by NS2 should be accounted for its inherent working mechanism. Since our system works on the delayed feedback mechanism, we have to store some variables to use it in later stage, as in the case of TCP source uses congestion information which reaches to it after one RTT, hence we have a increase in memory increase in the case of FS as the step size is decreased, which forces more number of data to be stored. Whereas, in the case of AFS method,

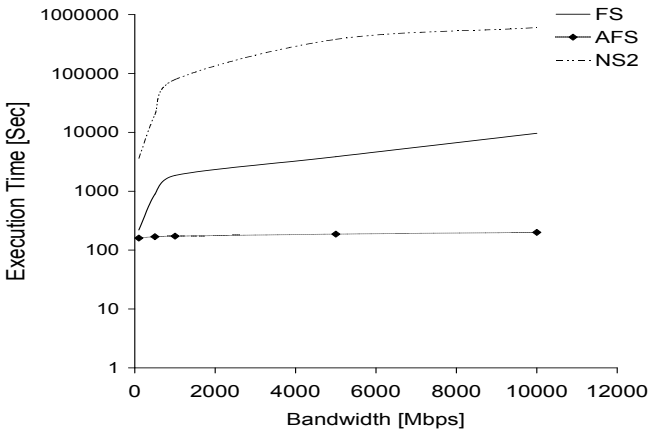

Fig. 10. Comparison of Execution Time for NS2, FS and AFS Methods with the Variation of Bandwidth

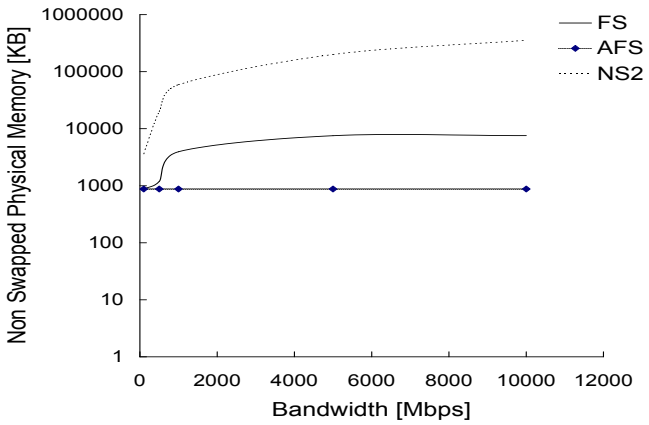

Fig. 11. Comparison of Non-Swap Memory Utilization for NS2, FS and AFS with the Variation of Bandwidth

most of the time simulation is carried out with maximum step length, hence there is no significant increase in the utilization of non-swapped memory, hence results in better performance.

VII. Conclusions

As the bandwidth of high speed networks is getting larger, fluid-based simulation method gains attention from researchers in the area of networks due to its scalability on bandwidth compared to the packet-based simulation methods. Although the fluid-based simulation method has a significant reduction in terms of computational time, it still suffers a scalability problem for the networks with bandwidth greater than 10 Gbps. Since the fluid-based simulation method uses a constant time step to numerically solve the system of differential equations, it needs to decrease the size of time step in case of a larger bandwidth. The decrease of time step produces more number of time steps, which induces more amount of computational cost.

In this paper, we have developed the time-adaptive method for the numerical solver for a system of differential equations to reduce the computational cost. The proposed method adjusts the time step-size for the numerical solver in order to reduce the computational cost while maintaining the accuracy of simulation results. The time-adaptive method uses a larger time step for the part of linear increase and a smaller time step for the part of multiplicative decrease in the event of packet loss. Since the event of packet loss is synchronized with the event of multiplicative decrease, we can adjust the time step based on the event of packet loss.

Comparisons between the time-adaptive method and the constant time step method show that the proposed method significantly reduces the computational cost while maintaining the level of accuracy compared to the constant time step method.

References

[1] NRL, "http://www.nlr.net/," .
[2] LONI, "http://www.cct.lsu.edu/news/loniforum.php/," .
[3] C. Jin, D. X. Wei, and S. H. Low, "Fast tcp: motivation, architecture, algorithms, performance," in *Proceedings of the conference on Computer Communications (IEEE INFOCOM)*, Hong Kong, Mar. 2004.
[4] Sally Floyd, "Highspeed tcp for large congestion windows," Tech. Rep. RFC 3649, Dec 2003.
[5] Tom Kelly, "calable tcp: improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 83 – 91, Apr. 2003.
[6] L. Xu, K.Harfoush, and I. Rhee, "Binary increase congestion control (bic) for fast long-distance networks," in *IEEE INFOCOM*, Mar. 2004.
[7] D.J.Leith R.N.Shorten, "H-tcp: Tcp for high-speed and long-distance networks," in *Proc. PFLDnet*, 2004.
[8] ns2, "http://www.isi.edu/nsnam/ns," .
[9] opnet, "http://www.opnet.com/.," .
[10] ns2 LargeSim, "http://www.isi.edu/nsnam/ns/ns-largesim.html," .
[11] SSFnet, "http://www.ssfnet.org," .
[12] Nistnet, "http://snad.ncsl.nist.gov/itg/nistnet/," .
[13] Vishal Misra and WeiBo Gong andDon Towsley, "A fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red," in *ACM SIGCOMM*, Sept. 2000.
[14] Y. Liu, F. L. Presti, Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu, "Fluid models and solutions for large-scale ip networks," in *ACM SIGMETRICS*, 2003.
[15] F. Baccelli and D. Hong, "Flow level simulation of large ip networks," in *INFOCOM*, 2003.
[16] DWDM, "http://www.iec.org/online/tutorials/dwdm/," .
[17] Arieh Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press, 1996.
[18] Anlu Yan and Wei-Bo Gong, "Time-driven fluid simulation for high-speed networks," *IEEE Transactions on Information Theory*, vol. 45, no. 5, July 1999.