A Fluid Model for Aggregate TCP Connections

Raffaele Bolla, Roberto Bruschi, Matteo Repetto

DIST - Department of Communications, Computer and Systems Science

University of Genoa

Genoa, Italy

Via Opera Pia 13, 16145 Genova, Italy

Email: raffaele.bolla, roberto.bruschi,matteo.repetto@dist.unige.it

***Abstract*— It is well known that the large majority (more than 90%) of IP traffic on the Internet is today carried by TCP connections. Moreover, it is quite realistic to suppose that the largest part of best effort traffic will use TCP also in the future. So, this protocol is and will be a fundamental element for the evaluation and forecast of IP network performance. TCP has been studied for many years, and a lot of models have been proposed for it that can be found in the scientific literature. However, most of them are related to a single connection behavior. In this respect, the aim of this work is to propose a model for aggregate TCP flows that could give a good representation of TCP performance in a network with an acceptable level of precision but also of complexity. Our model can be consider a "fast" simulation model, i.e. it is based on a sort of simulation (event generations and performance measures), which, however, is very fast, because it operates at aggregate level by using a fluid approximation, so that the computational time it requires is comparable to that required by a relative by complex analytical model. The paper reports several tests and comparisons with NS2 simulations, which show the quite good precision of the proposed TCP model.**

Keywords—TCP connections modeling, Networks modeling, TCP performance evaluation.

I. INTRODUCTION

It is well known that the large majority (more than 90%) of IP traffic on the Internet is carried by TCP connections. Even if real-time services over IP should use the UDP protocol, today this kind of services represents a small part of the global traffic traveling on the "normal" IP worldwide network. Moreover, in the future the assured quality services should be treated in a different way with respect to normal best effort traffic, e.g. by using the IETF Differentiated or Integrated Services architectures [1][2][3]. So it is quite realistic to suppose that the largest part of best effort traffic will use TCP also in the future. Another indication of this trend can be found in the Internet Service Provider (ISP) traffic type measures [4]; these measures show that an ever increasing part of today's IP traffic (sometimes more than 50%) is generated by peer-to-peer applications (like Gnutella, Kazaa, … actually freely available on the network), which actuate the data transfer by using (more or less always) the TCP protocol.

So the TCP is and will be a fundamental element for the evaluation and forecast of IP network performance [5]. This protocol has been studied for many years, and a lot of models have been proposed for it that can be can be found in the scientific literature [6-21]. However, most of the proposals are related to a single connection behavior, like [6], [7], [8] and [9] where the details of the congestion control are precisely modeled. In [10] three different versions of the TCP control mechanism (Tahoe, Reno and SACK) are compared and, again, a single connection model for each different mechanism version is proposed. [11] and [12] report a characterization of the statistical parameters of the connections, while [13] is a statistical study of connection aggregates. In [14], [15] and [16] the interaction among more connections is analyzed, while in [17] a fluid model for TCP is proposed that differs from our proposal, because it maintains a sort of detailed description of single connection behavior (with a quite high computational effort). In [18], [19] and [20] there are some interesting indications for the modeling of aggregate TCP traffic. Finally, in [21] Roberts et al. have studied the statistical bandwidth sharing of many random TCP connections, and they propose a fluid model to analyze the performance and the level of fairness of TCP congestion control, with respect to different values of Round Trip Time (RTT) and of average sizes of data transferred with each single connection. This last approach is somehow similar to our proposal (the main differences will be described in the following Sections).

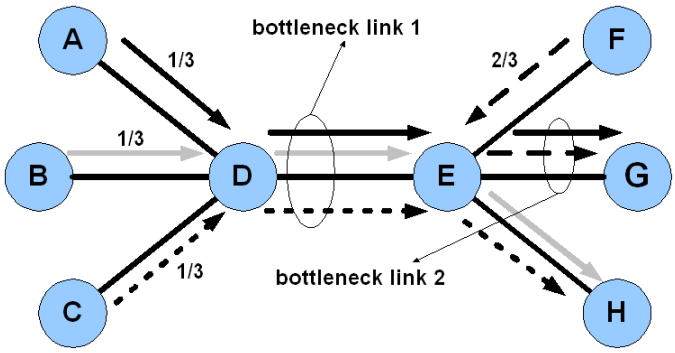
Our effort is to propose a model for aggregate TCP flows that could give a good representation of the TCP performance in a network with an acceptable level of precision but also of complexity. Our model can be consider a "fast" simulation model, i.e. it is based on a sort of simulation (event generations and performance measures) which is very fast, because it operates at aggregate level by using a fluid approximation. Note that the computational time that is required is comparable to that of a relative complex analytical model. The data that can be acquired by this simulator are the link utilizations and the service time of each data transfer.

The paper is organized as follows. The next Section reports the general structure of the proposed system, while Section III describes the data source model. The detailed description of the

whole aggregate TCP connections model can be found in Section IV. Section V shows the numerical results and, finally, Section VI contains the conclusions.

II. General Model Description

The model is based on the flow concept. A flow models all the TCP connection data exchanges between the same source and destination pair as a "liquid" data exchange. The flow representation does not take into account the packet dynamics. The incoming traffic is represented as blocks of data with random size, which become available (for transmission) at a source node at a random instant. Every source–destination pair is modeled as a tank, and the network links represent the pipes which drain the tank, while the data blocks fill it. Where more than one "liquid data" source (either coming directly from a tank or from a previous link) has to share a single pipe, this sharing is guided by a specific law. This law will be completely detailed in Section IV but, simplifying, it divides the capacity of the link in proportion to a sort of "pressure" of the different flows. The source-destination path of each flow is decided by the routing algorithm. The rate (the number of bits per second) associated to each flow is fixed by the link along the path, which assigns the smallest bandwidth to that specific flow. These rates are always dynamic ones; they can change when a tank in the network becomes void, or when it is void and receives a block of data, or when the routing changes one or more paths.



Simplified model behavior representation.

Fig. 1 shows a simplified representation of the model behavior; each flow is supposed to have the same "pressure" and all the links have the same capacity. The colored arrows are the flows, while the numbers a side are the fractions of link capacity that each flow can utilize. All the "tanks" are supposed to have some data inside (so every flow is active). It is easy to see that there are two bottleneck links: the first between nodes D and E which limits the size of black, gray and dotted flows to 1/3; the second between nodes E and G, which limits the size of the sketched flow to 2/3, because the black one is already limited to 1/3. This kind of model results in very fast simulation and it gives the possibility to compute numerical performance indexes in a quite precise way. Moreover, every routing algorithm can be used with it without modifying anything in the system. For the notation aspects, we indicate with N the set of the nodes, with P the set of flows p between each node pairs (i,j): $i \neq j$, $i,j \in N$. t_k, k=, 0,1,2,…, being the event instants (i. e. the instants when there is a new block arrival, or a buffer empties or the routing changes a routing table), $r_{t_k}^{(p)}$ is the rate associated to the flow $p \in P$ during the period $[t_k, t_{k+1})$.

III. TCP Sources Model

The statistical characterization of TCP sources is a complex problem that has been studied for years in the technical and scientific literature. A TCP traffic source can be modeled at different levels of "resolution", i.e. packets, bursts, connections, aggregated connections. In our case every node in the network can contain a "data generator" for every source-destination pair. This generator can be represented as a tank (a fluid infinite buffer) which collects the data when they become available and it is emptied by the links by following the model laws. The data are generated as blocks of different sizes, which became completely available at a certain instant. They represent, somehow, the original data that the user needs to transport over the network.

This structure can model, in principle, three different types of situations with different grades of approximation. The first situation is the case in which the node is the real source of the data (i.e. a user PC); in this case the model is very realistic. The second one is the situation in which the node is a router, which collects the traffic coming from LANs directly connected (or connected with high speed links). Also in this case the approximation is quite good, if we suppose the LANs to be high-speed and switched ones; in this case the data are virtually instantaneously available at the router. The last case is that in which a node is a router receiving the traffic from other routers outside the network (i.e., in a part of the network which is not modeled). In this last situation the approximation is less precise, due to rate limitation and delays arising to the non modeled network part and specific actions should be applied, at least to limit the output flow rate in accordance to the average hypothetical feed speed of the router tank.

To realize our data generator we have to generate two quantities: the arrival instants of the blocks between pairs and their sizes.

The first quantity is a quite "random" parameter, because it strictly depends on the user behavior and on the interactions between users and applications. Taking into account this characteristic we have decided to use a Poisson process to generate the arrival instants, i. e. we use an exponential interarrival distribution, where the interarrival time Δt of the flow $p \in P$ has distribution:

$$f_{\Delta T}^{(p)}(\Delta t) = e^{-\lambda^{(p)} \cdot \Delta t}, \quad p \in P \tag{1}$$

with arrival rates $\lambda^{(p)}$, $p \in P$.

Concerning the size of data blocks, many studies and measures reported in the literature (see [22] and [23], among others) suggest the use of heavy-tail distributions, which originate a self-similar behavior, when a traffic coming from many generators is mixed in the network links. We have decide to use a Pareto-Levy distribution, i. e.:

$$f_X^{(p)}(x) = \alpha^{(p)} (x_{INF}^{(p)})^{\alpha^{(p)}} x^{-(1+\alpha^{(p)})} \quad , \quad x \geq x_{INF}^{(p)} \quad p \in P \tag{2}$$

where x is the block size in bits, $\alpha^{(p)}$ is the shape parameter and $x_{INF}^{(p)}$ the location parameter, i. e. the minimum size of a block. This distribution is characterized by an infinite variance when $\alpha^{(p)} \leq 2$ and an infinite mean for $\alpha^{(p)} \leq 1$. The average block size $\bar{x}^{(p)}$, with $\alpha^{(p)} > 1$, is:

$$\bar{x}^{(p)} = E\left\{x^{(p)}\right\} = \frac{\alpha^{(p)} x_{INF}^{(p)}}{\alpha^{(p)} - 1} \quad p \in P \tag{3}$$

In the model we utilize a truncated version of equation (2). This choice has two origins. From one side, it makes the generator more realistic, since the real maximum data block size is finite. On the other hand, this choice makes the model able to give stable average performance indexes, which is not really possible with a generator with infinite variance (note that the realistic values for $\alpha^{(p)}$ are between 1 and 2).

This approach gives us the possibility of representing both short connections (mices), which do not exit from the slow-start phase of TCP control flow, and very long data transfers (elephants). It is evident, anyway, that phenomena related with the packet level (like synchronization of elephant connections) are not observable with our representation

IV. TCP AGGREGATE FLOW MODEL

An aggregate flow is supposed to be formed by many TCP connections, each one characterized by its own adaptive congestion control. The behavior of every single connection is guided by "micro" temporal conditions (in terms of RTT and packet loss), which each flow control mechanism tries to follow (compatibly with the destination buffer sizes and status). Due to the high non linearity and adaptability of the TCP congestion control, it is very difficult to describe the instantaneous behavior of such aggregates along the network. Instead, our model tries to describe their average behavior, by considering them as fluid flows that occupy a constant bandwidth for a period of time (i.e., between two successive fluid simulator events) on all the links crossed. This approximation is acceptable if the packet loss probability is low. In particular, if a node is overloaded and then it looses a lot of packets, the link downstream this node sees a throughput equal to the number of non lost packets ("goodput") while the links upstream the node see the total traffic. The throughput difference between the two links is the "badput". In our model, substantially, we consider only the "goodput" of the aggregates, which results in a good approximation when the packet loss is low.

The model acts at each event instant t_k, k = 1, 2, 3, … by re-computing the value of rate $r_{t_k}^{(p)}$ for every aggregate flow $p \in P$. Thus, the core of the procedure is represented by the computation algorithm for these rates. This algorithm is substantially based on the *min-max* rule described in [24]. Let us drop, for the sake of simplicity, the time step index t_k, and define the following quantities:

- A, the set of all the links in the network;
- $C^{(a)}$, the capacity of the link $a \in A$;
- A_i, the set of all the links that are not already completely utilized at step i;
- $\tilde{P}$, the set of all the active aggregates (i.e. the aggregates with the buffer not empty);
- $\tilde{P}_i$, the set of all the aggregate flows which have not reached the maximum possible rate value at algorithm step i;
- $\tilde{P}_i^{(a)}$, the set of all the flows $p \in \tilde{P}_i$ which cross the link $a \in A$ at step i;
- $S^{(p)}$, the set of the links on the path followed by flow $p \in \tilde{P}$;
- $U_i^{(a)} \leq C^{(a)}$, the used capacity on the link $a \in A$ at step i;
- $r_i^{(p)}$, the rate of the flow $p \in \tilde{P}$ at step i;

At step i = 0 the variables are initialized as follows: $A_0 = A$; $\tilde{P}_0 = \tilde{P}$; $U_0^{(a)} = 0, \forall a \in A$; $r_i^{(p)} = 0, \forall p \in \tilde{P}$.

Then, the algorithm applied to find the rate of each flow ($r_{t_k}^{(p)}, \forall p \in \tilde{P}$) during the period $[t_k, t_{k+1})$ (for i=1,2,..) is:

1. Compute the percentage of link capacity sharing for each flow as

$$\gamma_i^{(a,p)} = \frac{\alpha^{(p)} \bar{x}^{(p)}}{\sum_{j \in \tilde{P}_i^{(a)}} \alpha^{(j)} \bar{x}^{(j)}}, \quad \forall p \in \tilde{P}_{i-1}^{(a)}, \quad \forall a \in A_{i-1}$$

2. Then compute the incremental rate for all the flows that do not have already reached the maximum value as

$$\Delta r_i^{(p)} = \min_{a \in S^{(p)}} \left\{ \left(C_a - U_{i-1}^{(a)} \right) \gamma_i^{(a,p)} \right\}, \forall p \in \tilde{P}_{i-1}$$

3. The new rates become

$$r_i^{(p)} = r_{i-1}^{(p)} + \Delta r_i^{(p)}, \quad \forall p \in \tilde{P}_{i-1}$$

4. Then the value of the utilized capacities within the sets of active links and flows must be updated as

$$U_i^{(a)} = \sum_{p \in \tilde{P}_{i-1}^{(a)}} r_i^{(p)} \quad \forall a \in A_{i-1}$$
$$A_i = \left\{ a : C^{(a)} - U_i^{(a)} > 0 \right\}$$
$$\tilde{P}_i = \left\{ p : S^{(p)} \cap A_i \right\}$$

5. $i = i + 1$;
6. If $\tilde{P}_i \neq \varnothing$ go to step 1; otherwise the procedure ends.

The critical part of this procedure is related to the computations at step 1. We have observed the way by which the different aggregate flows share a bottleneck link capacity, and we have seen that this sharing is directly proportional to the number of "active TCP connections", i.e., connections

which are transmitting. An example of this behavior can be seen in Fig. 3, which has been obtained with the test network of Fig. 2. In our model the concept of connection has no sense, but this observation also suggest that this sharing should be proportional to the offered load (if we suppose that the instantaneous throughput of each connection be the same, on average). So, we have defined a sharing parameter $\gamma^{(p,a)}$, which represents the maximum bandwidth portion that an aggregate flow p on a link a can use as:

$$\gamma^{(p,a)} = \frac{\phi^{(p)}\bar{x}^{(p)}}{\sum_{j \in \tilde{P}^{(a)}} \phi^{(j)}\bar{x}^{(j)}} \quad (4)$$

which is proportional to the offered load of the different active flows on the link.

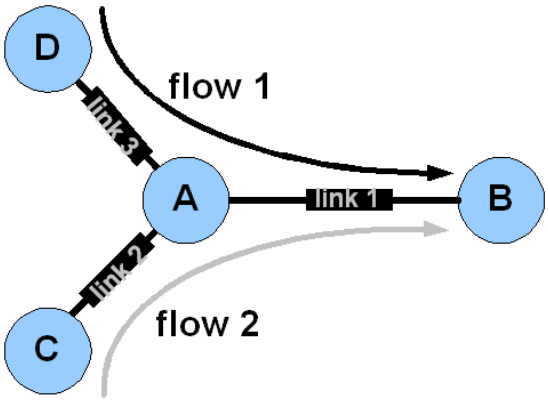


Figure 2. Simple test network topology with a single bottleneck link, which is shared by two aggregate flows.

Another element should be evaluated: this approach does not take into account the RTT (Round Trip Time). What we have observed in this respect is that if there is a sufficient number of active TCP connections in every aggregate flow, their superposition hides all the different behaviors with respect to the RTT. This effect can be easily observed in Fig. 4, which shows some results obtained with the simple example network of Fig. 2 with all the link bandwidth equal to 30 Mbps, $\bar{x}^{(1)} = \bar{x}^{(2)} = 0.39$ Mbyte, $\phi^{(1)} = 5$ burst/s, $\phi^{(2)}=7.5$ burst/s, and $\varphi^{(1)} = \varphi^{(2)} = 2$. Fig. 5 is a comparison between the flow simulator and ns, which shows the absolute throughput error over time versus link 3 delay. All the results in Fig. 5 are obtained with $C^{(1)}=C^{(2)}=C^{(3)}=30$ Mbps, link 1 and 2 delays equal to 5 ms, $\bar{x}^{(1)} = \bar{x}^{(2)} = 0.39$ Mbyte, $\phi^{(1)} = 5$ burst/s, $\phi^{(2)}=7.5$ burst/s, and $\varphi^{(1)} = \varphi^{(2)} = 2$. Note that the percent standard deviation is calculated with respect to the NS average flow.

The different value of RTT does not change the aggregate behavior, and this happens also in more complex conditions and networks not reported here. The different values of RTT, as described in [21], influence the performance of each single connection, i. e. for higher RTT values, we have connections with smaller average throughputs, and then longer average transmission times. This is because, on average, the throughput of each connection decreases, while the average number of connections in progress (i.e. the connections that are transmitting data) increases (they need a longer time to transfer the same quantity of data). In general, with a certain minimum number of connections, these two effects tend to compensate with respect to the aggregate throughput. So, if the average RTT value of each connection increases, then the aggregate flow should be composed by more TCP connections, each one with a lower average throughput, but the aggregate throughput remains almost the same.

Therefore, we can conclude that, only at aggregate flow level, there is a sort of independence in statistical bandwidth sharing with respect to the different values of RTT. The only critical situations can be found when a flow is composed by few TCP connections.

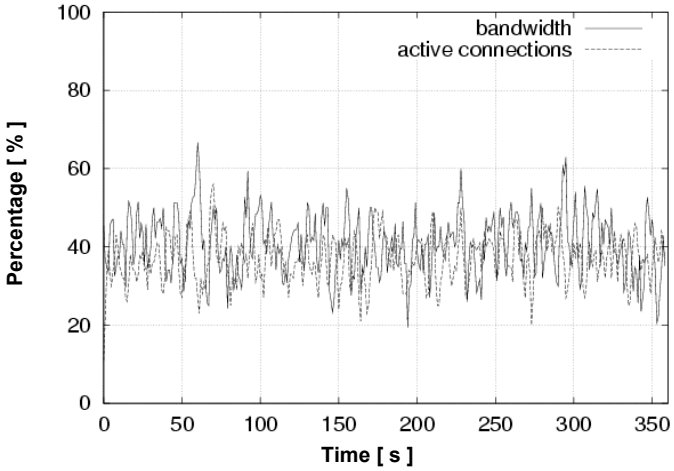


Figure 3. Percentage of utilized bandwidth for flow 1 (between node A and D) on link 1 and percentage of TCP active connections over the total ones for the same flow on the same bottleneck link between C and D in the network of Fig. 2, with C(a)=30 Mbps □a□ A, link 1 and 2 delays equal to 5 ms, $\bar{x}^{(1)} = \bar{x}^{(2)} = 0.39$ Mbyte, □(1) = 5 burst/s, □(2)=7.5 burst/s, and □(1) =□(2) = 2.

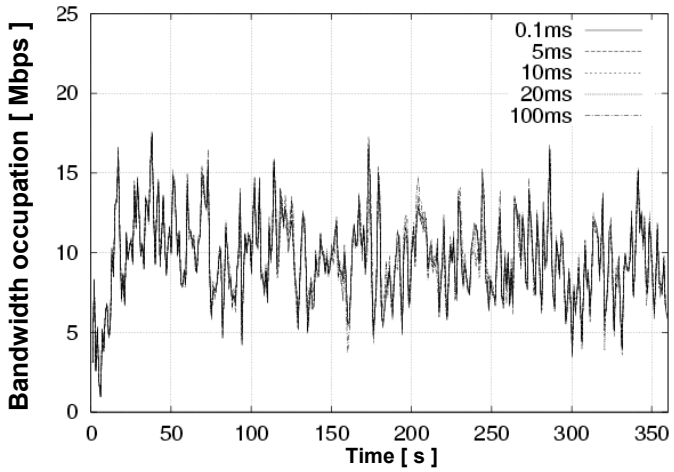


Figure 4. Average bandwidth utilization (obtained with NS2) for aggregate flow 1 (with respect to the network in Fig. 2, with $C^{(a)}$=30 Mbps □a□A, link 1 and 2 delays equal to 5 ms) with different delay values of link 3, namely: 0.1 ms, 5 ms, 20 ms, 10 ms and 100 ms. The traffic source of flow 1 has $\phi^{(1)}$=5

burst/s with $\bar{x}^{(1)}$=0.39 Mbyte, the traffic source of flow 2 has $\lambda^{(2)}$=7.5 burst/s and $\bar{x}^{(1)} = \bar{x}^{(2)} = 0.78$ Mbyte, while $\lambda^{(p)} = 2$ $\forall p \in P$.

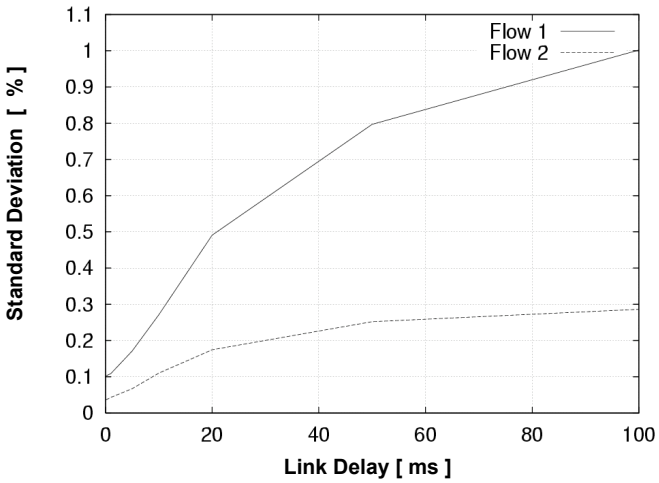


Figure 5. Comparison, in terms of percent standard deviation, between fluid model and NS2 versus different delay values of link 3 (with respect to the network in Fig. 2 with $C^{(a)}$=30 Mbps $\forall a \in A$, link 1 and 2 delays equal to 5 ms), namely: 0.1 ms, 5 ms, 20 ms, 10 ms and 100 ms. The traffic source of flow 1, from node C to B, has $\lambda^{(1)}$=5 burst/s, $\bar{x}^{(1)}$= 0.39 Mbyte and $\beta^{(1)}$ = 2, while the traffic source of flow 2, from node D to B has $\lambda^{(1)}$=8 burst/s, $\bar{x}^{(2)}$= 0.39 Mbyte and $\beta^{(2)}$ = 2.

V. NUMERICAL RESULTS

To obtain an acceptable validation of the numeric fluid simulator's results, we have decided to use the well-known NS2 tool [25] as comparison.

Given the particular structure of our fluid model, the natural term of comparison is the utilized bandwidth (averaged over a window of 0.5 sec.) in every link for each aggregate flow along time. To make the comparison meaningful between the two simulators, we have always used both the same topology and the same traffic load realizations. The comparison has been realized starting with a simple network situation, to reach then more complex environments. To obtain clear indications about the feasibility of the applied approximations, many parameters not considered in the fluid model (like propagation delay, or different version of TCP flow control) have been changed in the NS2 simulations.

In the first set of simulations, we have used the simplest possible environment: one peer (aggregate flow), which crosses a single link (Fig. 6).

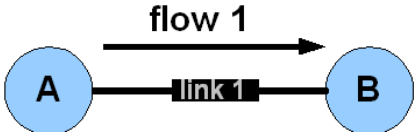


Figure 6. Simple network topology with a single link.

The first objective is to obtain an indication about the behavior of the model with respect to different offered loads, link capacities, delays, mean burst sizes and Pareto form values. Some of the results obtained with these tests are reported in Figs. 7 and 8, where it can be seen that the fluid and NS2 results are very close. The same results have been obtained with all the other (not reported) tests. Note that the maximum value of bandwidth occupation is about equal to the total capacity of the link. Then, according to these results, it is reasonable to conclude that a generic TCP aggregate flow can totally use the available resources on a link.

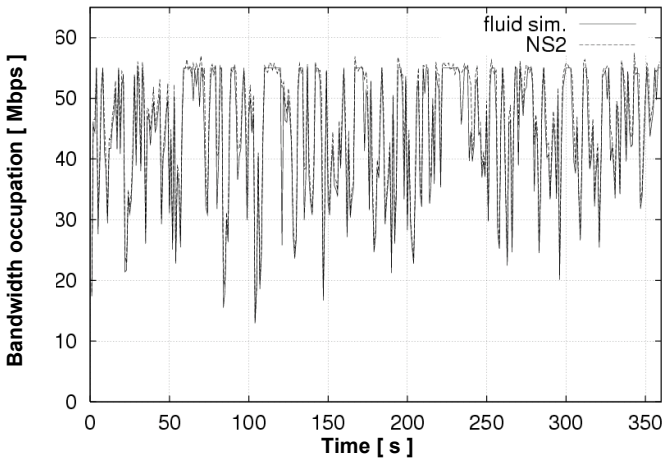


Figure 7. Utilized bandwidth with an aggregate flow, with $\lambda^{(1)}$ =15 burst/s, $\bar{x}^{(1)}$ = 0.39 Mbyte and $\beta^{(1)}$ = 2; the link capacity is $C^{(1)}$=55 Mbit/s, with delay time of 1 ms.

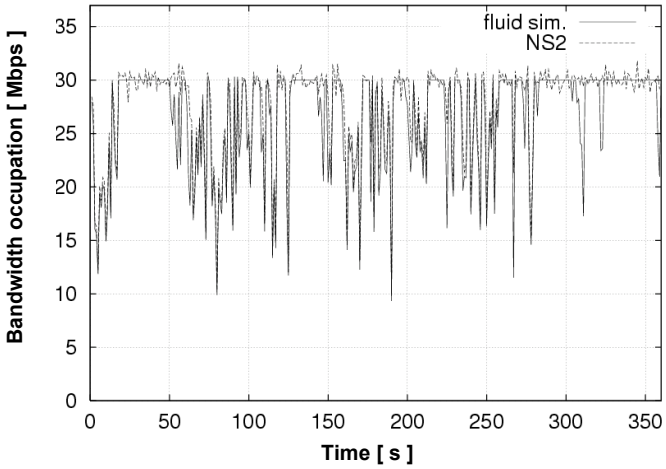


Figure 8. Utilized bandwidth with an aggregate flow, with $\lambda^{(1)}$ =13 burst/s $\bar{x}^{(1)}$ = 0.39 Mbyte and $\beta^{(1)}$ = 2; the link capacity is $C^{(1)}$=30 Mbit/s, with delay time of 0.1 ms.

Using the previous network, we have investigated if the usage of some of the most widespread TCP versions generates different behavior. We have used the same offered load generation for four different NS2 simulations, each of them with a specific TCP congestion control version, namely Reno, NewReno, Tahoe and Sack version. All the simulations are characterized by a packed drop probability equal to 0.01. Fig. 9 reports the result of this comparison: as one can observe, there are no perceptible differences. Fig. 10 is a comparison, in term of percent standard deviation, between the fluid simulator and

NS2 with different TCP versions (Tahoe, Reno, New Reno, Vegas and Sack), versus the offered load. To obtain this comparison we used the network in Fig. 6, with $C^{(1)}$ = 45 Mbps and the delay time of link 1 of 5 ms, while the traffic source of flow 1 is characterized by $\lambda^{(1)}$=20 burst/s, $\bar{x}^{(1)}$= 0.39 Mbyte and $\sigma^{(1)}$ = 2. Note that the percent standard deviation is calculated with respect to the NS average flow.

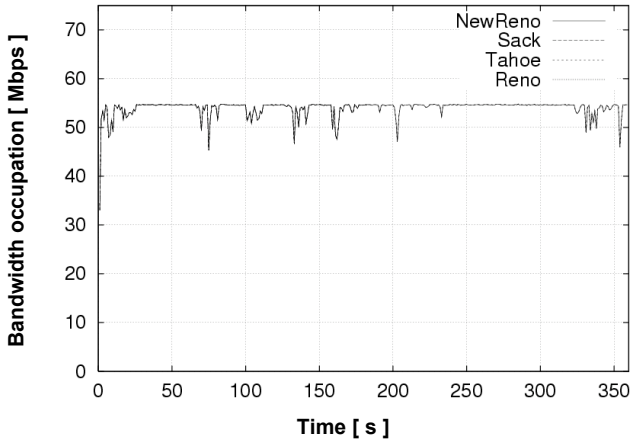


Figure 9. Utilized bandwidth in NS2 for different TCP versions (Reno, New Reno Tahoe and Sack) with the same source generation (λ(1) =26 burst/s, $\bar{x}^{(1)}$ = 0.37 Mbyte and σ(1) = 2), using the network in Fig. 6 with C(1) equal to 55 Mbit/s, and delay time of 10 ms.

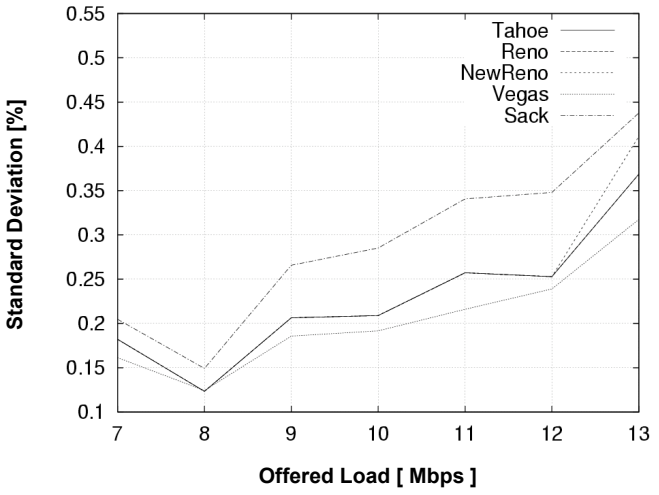


Figure 10. Comparison, in term of percent standard deviation, between fluid model and NS2 versus different TCP versions, namely: Tahoe, Reno, New Reno, Vegas and Sack. The network used in this comparison is represented in Fig. 6, $C^{(1)}$=45 Mbps and the link 1 delay time is 5 ms. The traffic source of flow is characterized by $\lambda^{(1)}$=20 burst/s, $\bar{x}^{(1)}$ = 0.39 Mbyte and $\sigma^{(1)}$ = 2.

The next results have been obtained by using the simple test network of Fig. 2. Figs. 11 and 12 show the behavior of the fluid model in a shared link situation with a high offered load. We have used a link 1 and 2 capacity of 20 Mbps, a link 3 capacity of 12 Mbps and a total offered load of quite 11 Mbps. In this case, with heavy congested links, the accuracy of the fluid model decreases, because the fluid model does not tend to follow the real bandwidth dynamics of the aggregate flows, but tends only to approximate the average behavior. This is so, because the model is guided by the fluid buffer's state (i.e., the bandwidth sharing algorithm is applied when a fluid buffer becomes empty or when an empty fluid buffer receives data to transmit): so, when some flows cross a heavy congested links, the fluid buffers of those aggregates tend to empty very slowly, and the bandwidth sharing algorithm is not applied for a long period of time. The throughputs obviously remain constant during all this time interval, and equal to the average values of bandwidth occupation calculated by the criteria of (4).

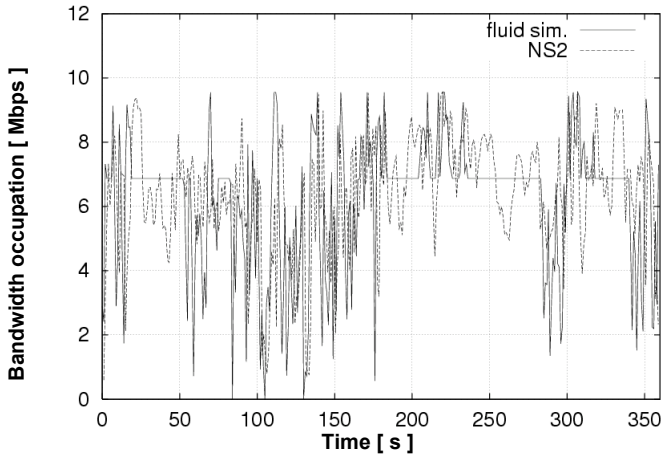


Figure 11.Utilized bandwidth for aggregate flow 1 (with respect to the network in Fig. 2), with $\lambda^{(1)}$= 3 burst/s, $\lambda^{(2)}$=1 burst/s, $\bar{x}^{(1)}$ = 0.29 Mbyte, $\bar{x}^{(2)}$ = 0.44 Mbyte and $\lambda^{(p)} = 2 \lambda p$; $C^{(2)}$= $C^{(3)}$ = 20 Mbit/s, with both link 1 and 2 delay times equal to 20 ms, $C^{(1)}$ = 12 Mbit/s, with link 1 delay time fixed to 15 ms.

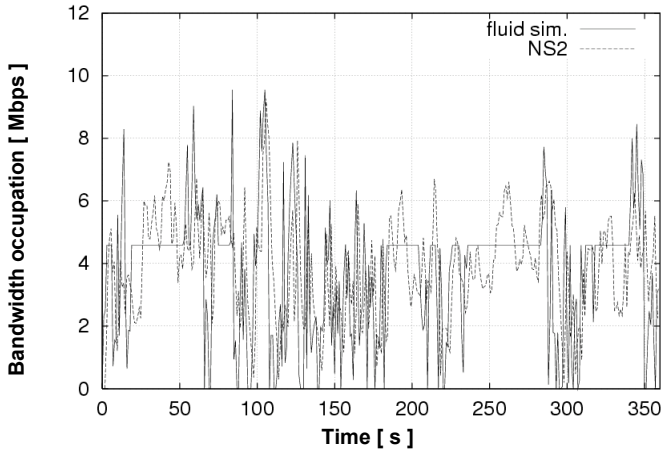


Figure 12.Utilized bandwidth for aggregate flow 2 (with respect to the network in Fig. 2), with $\lambda^{(1)}$= 3 burst/s, $\lambda^{(2)}$=1 burst/s, $\bar{x}^{(1)}$ = 0.29 Mbyte, $\bar{x}^{(2)}$ = 0.44 Mbyte and $\lambda^{(p)} = 2 \lambda p$; $C^{(2)}$= $C^{(3)}$ = 20 Mbit/s, with both link 1 and 2 delay times equal to 20 ms, $C^{(1)}$ = 12 Mbit/s, with link 1 delay time fixed to 15 ms.

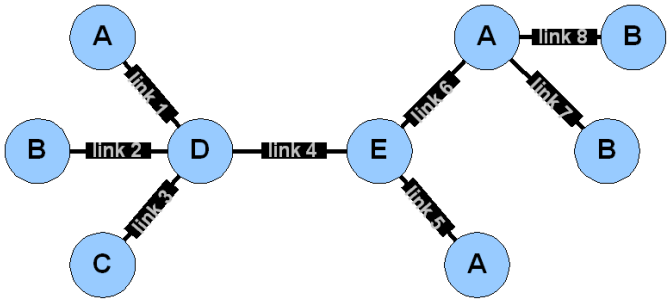


Figure 13. Network topology with two bottleneck links.

Finally, we have carried out some tests with the same traffic load of all peers, the topology of Fig. 13, and by changing only the bandwidth in the first bottleneck link (link 4 with reference to Fig. 13).

TABLE I. PARAMETERS OF THE LINKS OF THE NETWORK TOPOLOGY IN FIGURE 13.

	Bandwidth	delay
Link 1	50 Mbps	1 ms
Link 2	50 Mbps	1 ms
Link 3	50 Mbps	0.2 ms
Link 4	Variable	5 ms
Link 5	50 Mbps	1 ms
Link 6	80 Mbps	10 ms
Link 7	5 Mbps	0.5 ms
Link 8	10 Mbps	1 ms

TABLE II. SOURCE AND DESTINATION NODES AND STATISTICAL PARAMETERS OF TRAFFIC SOURCES USED IN THE LAST VALIDATION SET FOR THE FLUID MODEL.

	Node Tx	Node Rx	_	$\bar{x}$	□
Flow 0	A	H	30	1.5	1
Flow 1	B	G	20	2	1.5
Flow 2	C	E	15	1.5	2

By acting on the bottleneck is bandwidth and by keeping the offered load unchanged, we can observe the fluid model performance with respect to different average utilizations of the bottleneck link 4. In fact, as one can observe in Fig. 14, the behavior of the average error and of the error standard deviation (both normalized with respect to the average load of the network) shows that the fluid model is closer to NS2 when the offered load of all peers, which cross the same link, is lower than the available bandwidth of the latter. This result depends again (as in the situation of Fig. 2) on the fact that the model is working with all the flows active for long periods of time. Also in this conditions, anyway, the global precision of the model remains acceptable for most of the possible usages.

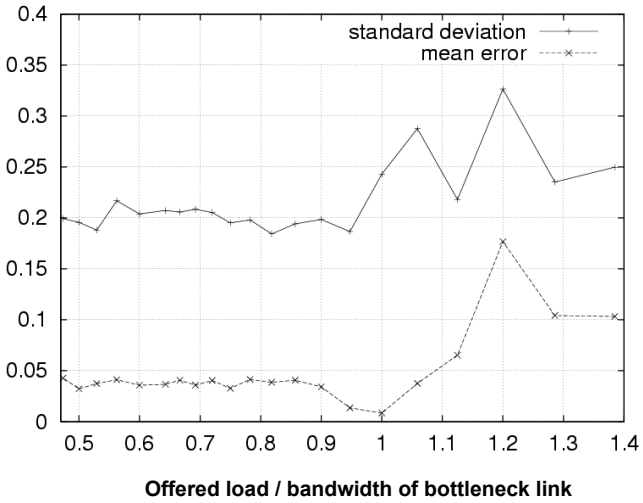


Figure 14.Values of average error and average quadratic error (both normalized to network load), between NS2 and the fluid model, versus the bandwidth of the bottleneck link. For the input parameters of this simulation see Tabs. 1 and 2.

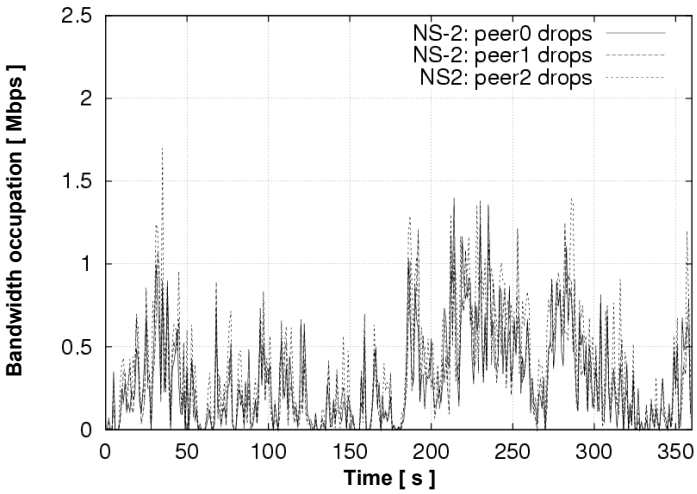


Figure 15. Throughput of drop packets for aggregate flows 0,1 and 2 at node 3 (with Fig. 12 topology); the bandwidth of the bottleneck link 4 is 105 Mbps.

Another cause of reduction in the performance of the fluid model may be a non negligible packet drop probability, which could appear with high traffic loads and with a multi-bottleneck network topology (as, for example, a "parking-lot" topology). In fact, in these cases, the fluid simulator may overvalue the throughputs of those peers which cross only a part of the bottleneck link's chain, because our model does not consider the bandwidth resources taken up by the dropped traffic throughput (or "badput") of the other aggregate flows.

Otherwise, there are no problems in the presence of significant "badput" components only in those links which are ahead of bottlenecks. For instance, Fig. 15 shows the quantity of dropped traffic at the node before the first bottleneck link (node C). Despite from this relatively high "badput", the

average error and error standard deviation values are both quite low (normalized average error = 0.035 and normalized standard deviation ≈ 0.2).

The execution times depend on the number of significant aggregate sources in the simulated network. To give a concrete idea, all the simulations shown in this paper have been realized with a AMD Athlon XP 1800+, and the longest simulation time (for one network condition) has been about 5 minutes.

VI. CONCLUSIONS

We have proposed a model for aggregate TCP flows that could give a good representation of the TCP performance in a network with an acceptable level of precision, but also of computational effort. Our model can be consider a "fast" simulation model, i.e., it is based on a sort of simulation (event generations and performance measures) which is quite fast, because it operates at the aggregate level by using a fluid approximation, so that the computational time it requires is comparable to that of a relative by complex analytical model.

We have presented the technique in some detail and we have shown several results, which confirm the relative precision of the model, but also underline its actual limits. The global results are anyway good and this approach can be useful for both control techniques (bandwidth allocation) and planning mechanisms which want to take into account the performance of the best effort traffic.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang , W. Weiss, "Request for Comments 2475: An Architecture for Differentiated Services", Informational. December 1998, URL:http://www.ietf.org/rfc/rfc2475.txt..

[2] C. Weider, P. Deutsch, "Request for Comments 1727: A Vision of an Integrated Internet Information Service". Informational. December 1994, URL: http://www.faqs.org/rfcs/rfc1727.html.

[3] Xiao, X., Ni, L. M., "Internet QoS: A Big Picture". IEEE Network. August 1999.

[4] J. Leyden, "P2P swamps broadband networks". On-line article available on Register USA. URL: http://www.theregus.com/content/6/26287.html.

[5] R. Bolla, F. Davoli, M. Repetto, "A control architecture for quality of service and resource allocation in multiservice IP networks". Proc Art-QoS 2003, Workshop on Architectures for Quality of Service in the Internet. 24-25 March 2003, Warsaw, Poland.

[6] M. Mathis, J. Semke, J. Madhavi. "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm". Pittsburgh Supercomputing Center. ACM SIGCOMM. June 1997, vol. 27, no. 3.

[7] X. Yang, "A Model for Window Based Flow Control in Packet-Switched Networks". Proc. of the IEEE Conference on Computer Communications (INFOCOM) , (New York), Mar. 1999, pp., URL: www.ieee-infocom.org/1999/papers/04a_02.pdf.

[8] J. Padhye, V. Firoiu , D. Towsley, J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation". SIGCOMM. August 1998, pp..

[9] E. Altman, K.E. Avrachenkov, C. Barakat, "A stochastic model of TCP/IP with stationary random losses". ACM SIGCOMM 2000, Stockholm, Sweden, also in Computer Communication Review. October 2000, vol. 30, no. 4, pp.231-242.

[10] B. Sikdar, S. Kalyanaraman and K. S. Vastola, "Analytic models and comparative study of the latency and steady-state throughput of TCP Tahoe, Reno and SACK", GLOBECOM 2001 - IEEE Global Telecommunications Conference. November 2001, no. 1, pp. 1781 – 1787.

[11] V. Misra, W. Gong and D. Towsley. "Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior". Proceedings of Performance '99. October 1999.

[12] A. Arvidsson, P. Karlsson, "On Traffic Models for TCP/IP". Int. Teletraffic Congress 1999 - ITC-16, pp. 457- 466.

[13] J. Kilpi and I. Norros, "Testing the Gaussian approximation of aggregate" Proceedings Internet Measurement Workshop. 2002, Marseille , France.

[14] Baccelli, F. and Hong, D., Interaction of TCP Flows as Billiards, Proc. of the IEEE Conference on Computer Communications (INFOCOM), San Francisco, April 2003.

[15] A. Misra, T. Ott, J. Baras, "Predicting bottleneck bandwidth sharing by generalized TCP flows". Computer Networks: The International Journal of Computer and Telecommunications Networking. November 2002, vol. 40, issue 4, pp. 557 – 576.

[16] M. Garetto, R. Lo Cigno, M. Meo, M. Ajmone Marsan, "A Detailed and Accurate Closed Queueing Network Model of Many Interacting TCP Flows". INFOCOM 2001, pp. 1706-1715.

[17] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, P. Owezarski "A flow-based model for Internet backbone traffic". Proceedings Internet Measurement Workshop 2002. 6-8 November 2002.

[18] V. Misra, W. Gong, D. Towsley, "A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED". ACM SIGCOMM 2000. August / September 2000.

[19] Y. Joo, V. Ribeiro, A. Feldmann, A. C. Gilbert, W. Willinger, "TCP/IP traffic dynamics and network performance: a lesson in workload modeling, flow control, and trace-driven simulations". ACM SIGCOMM Computer Communication Review. April 2001, vol. 31, issue 2.

[20] Yeom, A. L. Narasimha Reddy, "Modeling TCP Behavior in a Differentiated Services Network". IEEE/ACM Transaction on Networking. February 2001, vol. 9, no. 1.

[21] S. Ben Fredj, T. Bonald, A. Proutiere, G. Régnié, J. W. Roberts, "Statistical bandwidth Sharing: a study of congestion at flow level", Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications. August 2001, San Diego, California, United States, pp.111-122.

[22] W.Willinger, V. Paxon, M. S. Taqqu "A practical guide to heavy tails: statistical techniques and applications". 1998, pp. 27 – 53.

[23] Kihong Park, Walter Willinger, Self-Similar Network Traffic and Performance Evaluation. John Wiley & Sons, Inc., 2000, New York, NY,.

[24] D. Bertsekas, R. Gallager, Data Networks, 2nd Ed., Prentice-Hall, 1992.

[25] The Network Simulator – Ns2. Documentation and source code from the home page: http://www.isi.edu/nsnam/ns/.