A trace based method for realistic simulation

Philippe OWEZARSKI and Nicolas LARRIEU
LAAS–CNRS
7, avenue du Colonel Roche
31077 TOULOUSE Cedex 4
FRANCE
e-mail: {owe, nlarrieu}@laas.fr

***Abstract*— This paper proposes a new approach for making simulations realistic. This approach is based on the principle of "trace driven simulation", i.e. using results of actual traffic traces analysis in order to reproduce the same experimental conditions in simulation. The main principle of the approach proposed in this paper deals with making simulation traffic sources replay – under certain conditions – the actual traffic traces grabbed on actual networks. This paper describes the implementation of this approach in the NS simulator, and evaluates it by comparing the characteristics of traces obtained with our replay approach, with original data traces. The parameters that are considered for making the comparison are the usual traffic parameters as throughput, packet rate, etc., but also everything that is related to traffic dynamics, i.e. the second order statistical moments as autocorrelation of traffic or long range dependence.**

I. Introduction

Simulation is an essential tool to provide a priori evaluation of a network. It evaluates if such a network can work, i.e. if it is free of bugs, checks its liveness, its integrity, and also makes some evaluations of its performances. Of course, simulating the Internet is not an easy task, especially because of its size (number of users and equipments), its complexity (number of protocols for instance), the behavior of its users, and of course, its fast evolution in terms of technology and usage. Because of all these features, it is very difficult to get some realistic simulations of the Internet, or at least of some small parts of it.

[PAX01] that addressed this issue said that it is not enough to focus on the problem of network topology for simulating the Internet, but it is also essential to solve the problem of traffic sources to make them reproduce as much as possible the actual traffic of the Internet in terms of applications and protocols as well as their behaviors. For that, [PAX01] recommends "trace driven simulation", meaning that traffic sources have to use some results of traffic traces analysis. In addition, it is said that because of the self regulation mechanisms of protocols (congestion control mechanisms of TCP for example), the "trace driven simulation" approach has to work at flow level, the simulation environment being in charge of "shaping" packets according to the simulated network topology and protocols. The NS network simulator, for example, has been designed in the framework of the VINT [FAL96] project for this purpose.

One of the main issue for simulating traffic sources is related to the absence of an always valid traffic model and most of the time that traffic sources used in simulations are very simple and follow constant, Poisson or Markov laws. Of course, such sources do not generate traffics with all the unregular properties of actual sources in the Internet.

However, it is not easy to build a traffic model including all the characteristics and features of actual traffics. And the accuracy and realism of simulations remain poor.

Therefore, this paper proposes a new approach for building traffic sources for simulators. The principle of this approach deals with making simulation traffic sources replay the actual flow[1] traces captured with monitoring equipments. It is a simple and efficient way to create in simulators traffic sources having all the real characteristics and features, and going through the need of a flow arrival model in the Internet, that is not available yet.

In addition, as the simulation environment (topology, network agents, etc.) are in charge of the shaping of packets, it is important to create it in order to make it generate traffic having the same characteristics as actual traffic. Previous work [PAR96] and [VER00] showed that Internet traffic characteristics as long range dependence (LRD), self-similarity, etc. are due to TCP (the main protocol used in the Internet) and its congestion control mechanisms. As congestion control mechanisms are based on pre-defined responses to losses, it seems that the main characteristics of actual traffic xto enforce in simulations is related to the loss process.

This paper describes the implementation of this approach in the NS simulator (section II), and evaluates it. This evaluation is made by comparing the real traffic that has been captured and the "same" traffic replayed in the simulator (section III). The parameters that are compared are of course the usual traffic parameters (throughput, packet rate, etc.), but also everything that is related to the traffic dynamics, in particular second order statistical moments as autocorrelation of traffic or LRD.

II. A new trace based approach for simulation

As a starting point for using our approach for traffic sources, a traffic trace that can give flow level information is required. This trace can be any kind of packet level traces (as TCPdump or DAG traces [CLE00]). Then, some tools have to be used

[1] A flow is classically defined as a set of packets having the same 5-tuple, i.e. same source and destination addresses, same source and destination port numbers, and same protocol.

TABLE I

FLOW CLASSES USED TO REPLAY MONITORED TRACES

Class	Loss rate of flow class (%)
C0	0
C10	0-10
C20	10-20
C30	20-30
C50	30-50
C100	50-100

to extract the flows information. At the end of the process, it is required to have a flow file, having an entry for each flow. The parameters are:

- For a TCP flow, the beginning timestamp, the number of packets and their sizes. The NS replay agent is then going to start the flow at the right time and send all packets with their real size, the shaping of packets being realized by the sending and receiving TCP agents.
- For a UDP flow, the beginning timestamp, the number of packets, their sizes, as well as the time between every consecutive two packets. The principle to replay this flow is the same as for TCP, except that as UDP is not running any flow or congestion control algorithm, our UDP agent as to respect the sending times of all packets that depend on application and/or user.

A replay module has been developed for NS. The traces we used are traces from the WAND group (university of Waikato), captured in Auckland, NZ. They were captured on 10 Mbps Ethernet links at peak hours, and using DAG systems, thus guaranteeing a very accurate timestamping based on GPS clocks (accuracy is less than 2 microseconds).

For designing the topology of the network, the goal is to create the simplest topology able to reproduce the loss process in simulation. For that, we analyze each flow of the original trace and measure its loss ratio. The goal then is to reproduce thanks to the simulation topology the same loss ratio for each flow. In order to build a suited network topology, it is also necessary to extract from original traces other flow parameters:

- the loss rate experimented by each flow during their exchange on the network,
- the RTT (Round Trip Time) experimented by each flow,
- the average throughput got by each flow,
- the duration of each flow.

To limit the complexity of the simulation topology, and based on loss ratio analysis, we decided to define only six different loss classes of flows (cf. table I for details).

With the information extracted from original traces, we are able to deduce both bandwidth and queue length of each link of the simulation topology where the flow will be transmitted, depending on the loss class to which the flow belongs to.

The link bandwidth for class i (Bw_{Ci}) is computed thanks to equation 1:

$$Bw_{Ci} = \frac{\sum_{n=1}^{N_{flow}} d_i * Th_i}{d_{trace}} \tag{1}$$

where:

- N_{flow} is the number of flows of class i,
- d_i is the duration of flow i,
- Th_i is the average throughput of flow i,
- d_{trace} is the trace duration.

Then, the queue length for class i (QL_{Ci}) is deduced thanks to equation 2:

$$QL_{Ci} = Bw_{Ci} * (100 - rate_{loss}) \tag{2}$$

where:

- $rate_{loss}$ is the average loss rate (in %) got by each flow of the class.

Finally, the experiment topology that will be used to replay the trace considered in this example is depicted on Figure 1 (RTT_{Ci} is the average RTT of the whole flows belonging to the class i).

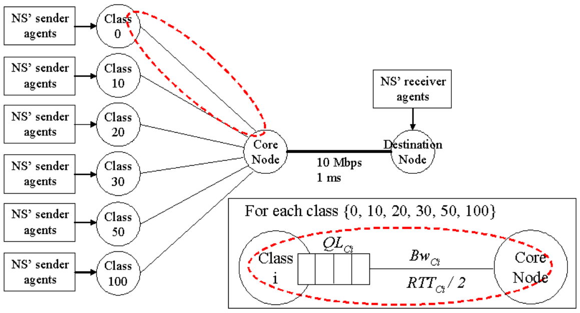


Fig. 1. Experiment topology

Traces have been monitored on a 10 Mbps Ethernet local network. That is why, in the experiment topology, the core link has a 10 Mbps capacity. The delay value for this link is 1 ms to avoid any influence on the sending of different class flows[2]. Indeed, the average RTT for each loss class has been computed and put on the different access links for the six loss classes.

In the next section, we present several experimental results to validate our replay approach. We have tested our replay method on a large number of traces, and we got very similar results with all of them. We are then just showing the results we got with one of them, comparing it with the replayed traffic. Recall that for the analysis we will mainly focus on traffic dynamics that are the most difficult parameters to reproduce and to control in simulation, and that are responsible of most the (performance) issues of the Internet. For example, it has been shown in the literature that the traffic characteristics causing the more issues are the highly oscillating nature of Internet traffic especially because of the dependence between the transmission of bytes and packets [PAX95], the dependence between losses or congestions [ZHA01], because of the heavy tailed flow size distribution that makes dependence phenomena propagate on very long range [CRO97]. Such features can be characterized with mathematical functions. In

[2] It should be 0 but it is impossible with NS.

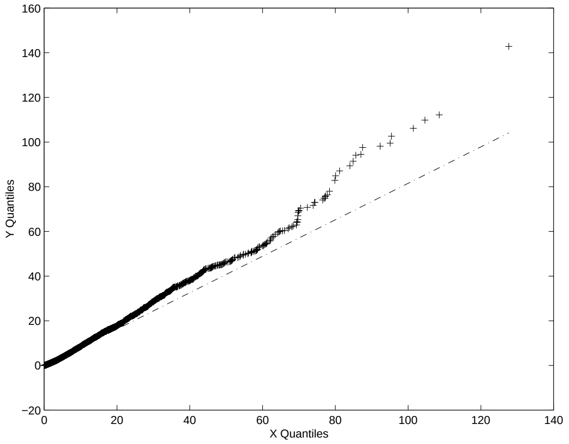


Fig. 2. QQ-plot of packets inter-arrival times (ms) – trace data (Y-axis) vs. simulation data (X-axis)

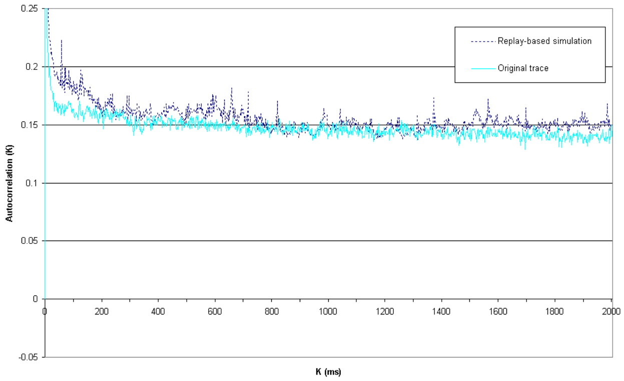


Fig. 3. Autocorrelation function of packets inter-arrival times series

the next section, dependence properties will be pointed out by the autocorrelation function, and oscillation range will be characterized by the LRD function.

III. Results analysis

First of all, the average loss ratio we got in simulation is the actual one fir each class.

But the main issue addressed in this paper deals with getting simulations where the “shaping” of packets is similar to the real case. For that, the inter-arrival times of packets are analyzed on both cases: simulation and real network.

The analysis shows that the only difference comes from the proportion of very close packets that is more important in the real trace than in the simulated one. Figure 2 depicts the QQ-plot of the two inter-arrival time series. It appears that the matching between simulation and real series is very good for the whole values except for very high quantiles. In the real case, packets separated by very short durations are the ones of flows that experiment very low RTT. In the simulation, as we defined for all flows of a loss class the same RTT, flows with short RTT are not well replayed.

Our trace replay based simulation technique proved to give good results for first level statistics (distribution function analyzed thanks to the QQ-plot matching evaluation). But to check that the two processes that generate the two traces (real and simulated) are similar, it is also required to analyzed second order statistics[3]. That is why Figure 3 shows the autocorrelation function for the 2 cases. It then clearly appears on Figure 3 that our trace based simulation gives quite good results for second order statistics, what is one of the key issue when replaying traffic.

To complete our analysis, it is required to compute the LRD of the traffic. In fact, LRD gives an evaluation of the dependence induced in the traffic at all scales. The goal now is to check that for every range, the dependence in the traffic is the same for the real and simulated trace. The LRD has been computed using the LDestimate tool [ABR98]. The results in the 2 cases are depicted on Figure 4. It appears that the LRD for our trace based simulation and the real trace are almost the same.

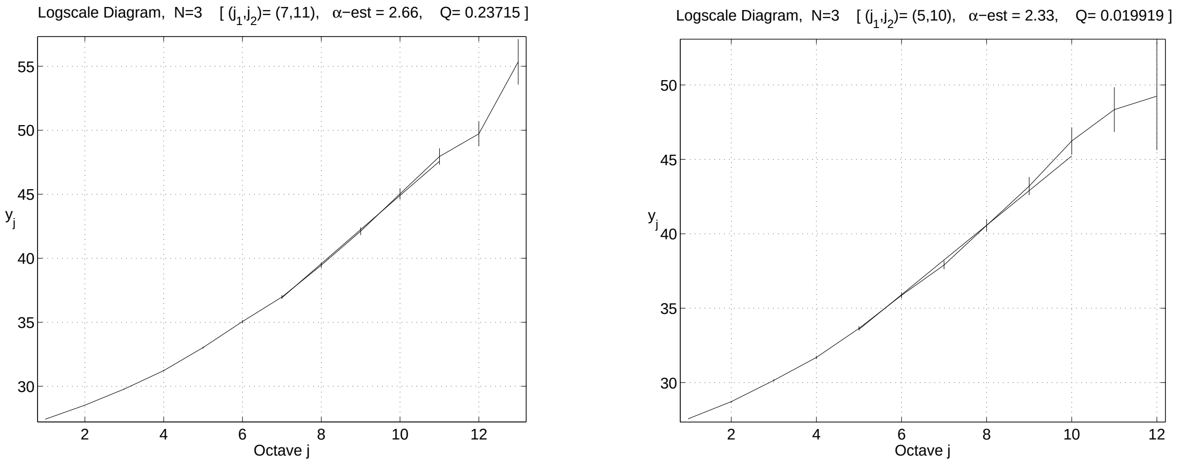
This means that the real traffic is highly long range dependent, and our simulation approach is able to perfectly reproduce that level of complexity on large scales. This result was expected as LRD is due to the log-normal or heavy tailed distribution of flow sizes [PAR96] and our simulation approach is replaying the flows with their real sizes.

IV. Conclusion

We proposed in this paper a new trace driven approach for simulating the Internet, and more particularly based on replaying traffic traces captured by a passive monitoring system. This approach has the strong advantage to build simulated traffic sources having all the characteristics and features of real flows arrival processes in the Internet. And it seems to be a good way to proceed to get realistic simulations, as long as a model for flows arrival in the Internet is not available. We also proposed a way for building the simulation topology that is based on our knowledge of the loss rate of each flow. The objective is there to reproduce as accurately as possible the loss process, as it is the essential parameter impacting the shaping of Internet traffic (at least for TCP traffic).

The results we got with our “trace based simulation” approach, showed that simulated traffics reproduce the complexity of actual traffics especially for traffic dynamics. Our simulation results are really impressive and show that simulation hugely benefits from network monitoring. However, we have seen, in section III, some small mismatches as some limits for having packets separated by short durations. This point will be addressed in future work. We will try to find a way to build the simulation topology not only on the single loss rate parameter, but also on RTT.

[3] In fact, to prove that two processes have the same behavior, it is required to show that they have the same behaviors at all statistical orders. But practically speaking, third and greater orders have very few influence. In such an experimental evaluation it is generally considered that it is enough to validate that the first and second order statistics are the same to validate the matching of the two processes.



(a) Original trace

(b) Replay-based simulation

Fig. 4. LDestimate diagram of packets interarrival times (ms)

REFERENCES

[ABR98] P. Abry, and D. Veitch, *Wavelet analysis of long range dependent traffic*, IEEE Trans. Inform. Theory, vol. 4, no. 1, 1998.

[CLE00] J. Cleary, S. Donnely, I. Graham, A. McGregor and M. Pearson, *Design principles for accurate passive measurement*, PAM (Passive and Active Measurements) Workshop, Hamilton, New Zealand, April 2000.

[CRO97] M. Crovella and A. Bestavros, *Self-similarity in World Wide Web traffic : Evidence and possible causes*, IEEE/ACM Trans. on Networking, Vol. 5, n. 6, pp. 835-846, Dec. 1997.

[FAL96] Fall, K., *Network emulation in the Vint/NS simulator*, proceedings of ISCC'99, July 1999.

[PAR96] K. Park, G. Kim and M. Crovella, *On the relationship between file sizes, transport protocols, and self-similar network traffic*, Proc. of IEEE ICNP, 1996.

[PAX95] V. Paxson and S. Floyd, *Wide-Area Traffic: The Failure of Poisson Modeling*, IEEE/ACM Transactions on Networking, Vol. 3 No. 3, June 1995.

[PAX01] S. Floyd and V. Paxson, *Difficulties in Simulating the Internet*, IEEE/ACM Transactions on Networking, Vol.9, No.4, pp.392-403, August 2001.

[VER00] A. Veres, Zs. Kenesi, S. Molnar, G. Vattay, *On the propagation of long-range dependence in the Internet*, SIGCOMM'2000, Stockholm, Sweden, September 2000.

[ZHA01] Y. Zhang, N. Duffield, V. Paxson and S. Shenker, *On the Constancy of Internet Path Properties*, Proc. ACM SIGCOMM Internet Measurement Workshop (IMW'2001), San Francisco, California, USA, November 2001.