Reducing the Complexity of Realistic Large Scale Internet Simulations

Kai Below
Airbus Germany GmbH
Cabin Communication Systems
Kreetslag 10, 21129 Hamburg, Germany
kai.below@airbus.com
Phone: +49 40 74 3 – 8 58 20, Fax: – 8 15 00

Ulrich Killat
Department of Communication Networks
Technical University Hamburg-Harburg (TUHH)
Denickestr. 17, 21071 Hamburg, Germany
killat@tu-harburg.de, www.tuhh.de/et6/
Phone: +49 40 4 28 78 – 30 49, Fax: – 29 41

***Abstract*— Computer simulation is a powerful tool for performance evaluation of new network protocols or dimensioning of communication networks. Current processors allow us to drive realistic simulations for medium to large network topologies with more than 110,000 HTTP/TCP clients, as shown in this paper.**

We reduce the complexity of the simulations by using a smaller number of clients as compared to reality while increasing the activity of each client and thus keeping the traffic load approximately constant.

When evaluating how far the number of clients can be reduced without introducing large errors, we are focusing on the following measures: accuracy of matching a specified (measured) traffic load for each link, coefficient of variation, Hurst parameter, end-to-end delay and loss probability. We show that the only parameter affected by the reduction is the loss rate.

Our results show that a significant reduction of the required memory, a factor $4-8$, can be gained by reducing the number of clients and increasing the client activity. A side-effect is a gain in simulation speed of up to 30 %. Nevertheless, the reduction of the required memory makes it possible to perform realistic simulations of large networks topologies on computer systems with hard per process memory limitations.

I. Introduction

The target of performance evaluations with computer simulations is to model reality as close as possible while minimising simulation effort, such that the results of the simulation show the characteristics of the real system. The problem is always to find where and how much detailed modelling can be reduced by approximations without changing or even losing the characteristic behaviour of the system under study. We focus here on HTTP/TCP traffic in large scale Internet simulations.

The reduction of simulation effort can be performed in at least two different ways: 1. usage of source models for aggregated traffic or 2. usage of a realistic, reactive user model with a reduced number of clients. We will focus on the second method since the intrinsic features of the WWW traffic are the user behaviour and the use of the HTTP/TCP protocols. Therefore it is important to use source models that reflect this behaviour as opposed to aggregated traffic [1].

Reducing the number of clients by keeping the traffic load approximately constant can be realized by increasing the activity of each single client [2], [3], [4]. In the HTTP/TCP source model we have two parameters that control the activity: the average HTTP volume and the average off-time. The average HTTP volume is usually quite small (e.g. ≈ 60 kB [5]) which means that TCP mostly operates in slow-start. Increasing the average HTTP volume has a major impact on the behaviour of TCP: it is more likely that the congestion-avoidance phase is reached, which means that the stochastic features of the generated traffic are changed. Reducing the average off-time in order to increase the activity of the clients, has a drawback, that we regard as less significant and that is also present with the first approach: we serialise transmissions which otherwise would have occurred in parallel, therefore multiplexing properties like average queueing delay and loss rate could be affected.

The reduction of the simulation complexity by reducing the number of customers goes along with corresponding changes in the parameters to be measured. We evaluate the changes of the following parameters in a large simulation study while reducing the number of clients: average traffic load, coefficient of variation of the inter-departure times, Hurst parameter of a byte-counting process and the end-to-end delay of TCP packets.

We perform this extensive simulation study with a network model of the German research network B-WiN, a model where

Published at GLOBECOM December 2003 San Francisco, CA, USA.

the simulation with realistic parameters and therefore a realistic number of clients is still feasible, i.e. the memory requirements are below the 3 GB per process limit of 32 bit systems. Observing the effect of reducing the number of clients gives information of how to scale down the number of clients for huge network simulations, where the simulation with realistic parameters is not practicable.

The rest of the paper is organised as follows. The source model is discussed in Sec. II. The B-WiN simulation scenario used for the evaluations is illustrated in Sec. III. The simulation results are shown in Sec. IV. We conclude the paper with a summary of the findings in Sec. V.

II. HTTP/TCP Client Model

The target of a realistic simulation can only be reached using a realistic model for the user behaviour. The traffic measured in the Internet is known to be self-similar [6], [7], [8]. Self-similar traffic can be generated with power-tail distributed random variables for the HTTP object size distribution [7]. The measurements of the traffic generated by the HTTP/TCP source model demonstrate the existence of self-similarity with a Hurst parameter $H > 0.5$. The Hurst parameter of the traffic can be adjusted with the shape parameter α of the power-tailed HTTP object size distribution [9], [2].

The HTTP/TCP client model is similar to the one described in [5]. We use the simulation software Ptolemy [10] for which we implemented a full TCP "New-Reno" protocol stack (with connection setup and release) and HTTP 1.1 with pipelining, that is, the full HTTP transfer volume is transmitted in one TCP connection which is being closed at the end of the transfer, see Fig. 1. We do not model the usage of parallel connections here. The TCP implementation was validated with ns-2 [11].

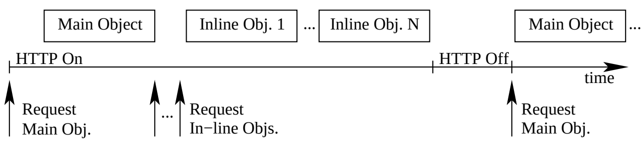


Fig. 1. HTTP/TCP behaviour, no parallel connections.

The HTTP request of a client opens a TCP connection via the three way handshake. Some requests for the embedded objects (inlines) are sent from HTTP client to server after receiving the main HTTP object. The TCP connection is closed by the client after receiving all requested objects. After successful transmission the user spends a certain amount of time reading. Further discussion of the distributions and their parameter fittings of the user model can be found in [9], [12].

The HTTP object size is drawn from a so called "truncated power-tail distribution" (TPT) [13] with a truncation level $T = 20$. We additionally limited the maximum HTTP object size to 100 MB to ensure convergence of the simulation. The major contribution to the self-similarity of the traffic comes from the distribution of the HTTP object sizes [9], [12]. Therefore we deviate from [5] in using a geometric distribution for the number of HTTP inline objects and a negative exponentially distributed off-time.

We have derived a simple approximation for the average throughput r_{src} of such an HTTP/TCP client in [2], [3]:

$$r_{src} = \frac{v}{t_{on} + t_{off}} \qquad (1)$$

where v is the average HTTP volume to be transmitted (main object and all inlines), t_{off} is the average off-time. The on-time is described by

$$t_{on} = (1 + n_{RTT_{ss}} + n_{RTT_{ca}}) \cdot rtt_{avg}, \qquad (2)$$

with $n_{RTT_{ss}}$ and $n_{RTT_{ca}}$ denote the number of round-trip times (required to transmit v) in slow-start and congestion-avoidance, respectively. We assume as an approximation that no packet loss occurs in slow-start and that the window size in congestion avoidance is $\text{CWND} = 0.75 \cdot \text{CWND}_{\text{max}}$. The derivation of the parameters $n_{RTT_{ss}}$ and $n_{RTT_{ca}}$ is detailed in [2]. The average round-trip time is denoted by rtt_{avg} and approximated by the sum of the propagation delays. The client parameters v and t_{off} control the average throughput. We use $v = 60$ kB and $t_{off} = 40$ s as realistic parameter set [5]. The amount of protocol overhead (header, acknowledgements and HTTP request volume) is taken into account in order to increase the accuracy of the estimations.

III. B-WiN Simulation Scenario

The B-WiN scenario (Fig. 2) is a complex, realistic model of the German research network. It models (at the state of February 2000) the network that connects all German universities and research institutes and for which a complete traffic matrix and some Hurst parameter values have been measured. The results of simulations with this model can be regarded as a test case to evaluate real network behaviour. The model consists of 11 nodes connected via 18 bi-directional links with a total capacity of 3.9 Gbit/s. The link capacities range from 53 Mbit/s to 167 Mbit/s and the link delays from 0.5 ms to 18.5 ms. Flows from all nodes to all adjacent nodes result in 110 flows in total.

The measured traffic matrix (throughput per flow in Mbit/s) is depicted in Fig. 3. The range of values is very large, the minimum value is 0.44 Mbit/s and the maximum value is 126.78 Mbit/s. The routing was optimised by the providers in order to keep the traffic load below $70\,\%$ on all links except for those coming from the USA (node "US") showing a load of more than $90\,\%$ (max. $98.4\,\%$). The minimum traffic load was measured on link "Ka->M" as $26.7\,\%$. The total average throughput was 1.44 Gbit/s in January/February 2000.

We have solved the problem of finding the number of clients per flow required to generate the measured throughput matrix depicted in Fig. 3 based on using (1) for all 110 flows [2], [3]. The solution is a prerequisite for being able to judge the effect of scaling down the number of clients while keeping the traffic load approximately constant.

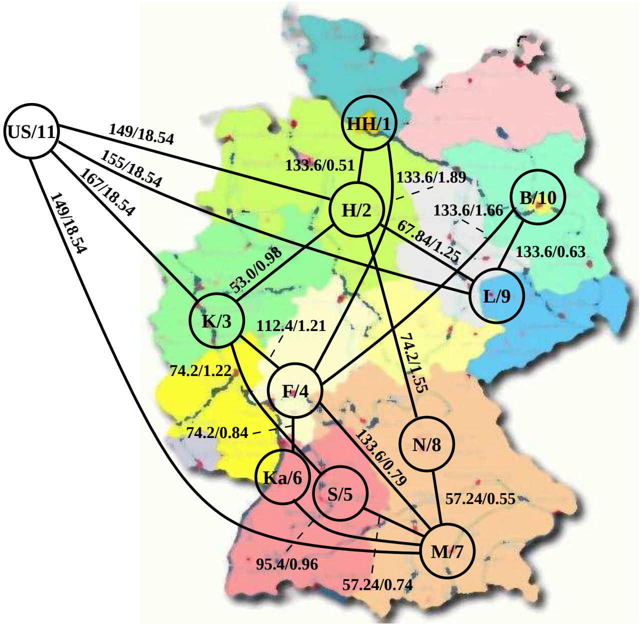


Fig. 2. B-WiN, capacities [Mbit/s] / delays [ms].

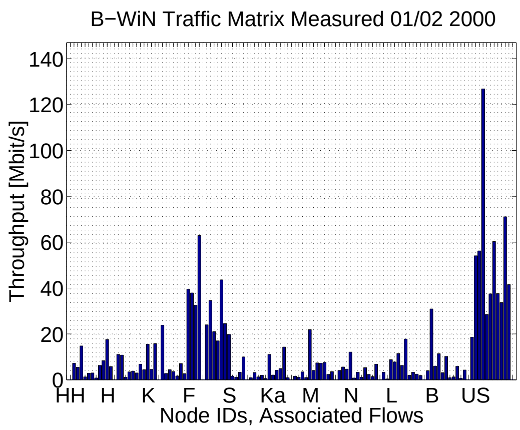


Fig. 3. B-WiN throughput matrix 2002/02.

The connection of subnets to the core network is modelled as follows: every node is equipped with two internal links with a capacity of 1 Gbit/s and propagation delay of 0.1 ms to which the local clients and web servers are connected, respectively. Each web server has a throughput limitation of 100 Mbit/s. We allocated 20 web servers at node "US", 10 web servers at node "F" and 5 web servers at all remaining nodes. All router buffer sizes were limited to 2500 packets.

The traffic measurements at the B-WiN in [14] show Hurst parameter values in the range $[0.53, 0.82]$. We try to cover this range by using three different values for the shape parameter $\alpha = \{1.1, 1.5, 2.0\}$ for the HTTP object size distribution.

IV. Simulation Results

We expect to gain efficiency for the simulations by scaling down the number of clients as discussed in Sec. I. Mainly two parameters are affected: the required memory and the simulation speed. We performed simulations for each of the combinations of off-time (seven values) and α (three values) with four different seeds. This results in a total of 84 simulations, each with a measurement time of 10 minutes. We spent additionally 100 s for each simulation for the transient phase, that is before the measurements are started.

The efficiency of the simulations (memory requirements and simulation speed) is discussed in Sec. IV-A. We discuss the changes associated with the reduction of the number of clients based on measurements of the following parameters: the average traffic load (Sec. IV-B), the coefficient of variation of the inter-departure times (Sec. IV-D), the Hurst parameter (Sec. IV-E) and the average end-to-end delay (Sec. IV-F). Due to space limitations we can only show some representative figures for selected parameters.

A. Simulation Efficiency

As a result of the client model (Sec. II, Eq. (1)) and the simulation scenario (Sec. III) the total number of HTTP/TCP clients changes with the off-time chosen. The relationship between off-time, total number of clients and the measured memory requirements of our simulator is displayed in Tab. I. The memory savings for smaller off-times are of major importance since the current 32-bit computer systems do not allow processes to use a larger address space than 3 GB. The theoretical limit of the address space of 32-bit systems is 4 GB, but Linux and Windows reserve 1 GB for the kernel [15], [16].

TABLE I

Total number of clients and measured memory requirements as a function of the off-time.

Off-time [s]	# Clients	RAM [MB]
40	112976	2000
20	56695	1150
10	28560	520
5	14495	270
2	6052	120

The empirical upper bound on the required memory for the simulations in Tab. I was found to be a linear relation:

$$M_{req}\ [MB] \leq N_{clients} \cdot 0.02 + 10. \qquad (3)$$

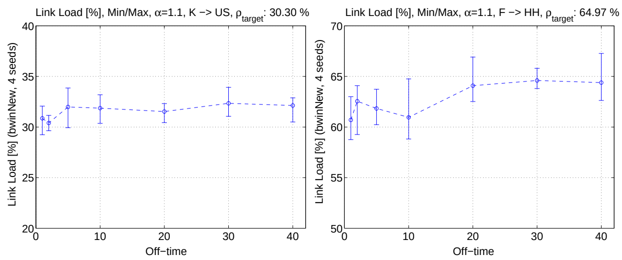
The maximum gain in simulation speed is smaller than 40 % (in our implementation) which is a rather small profit compared to the memory usage, which can be explained as follows: keeping the traffic load approximately constant means that the number of generated events in the simulator is also approximately constant and therefore the simulation speed remains approximately constant. We believe that the observed increase in speed is a result of smaller operating system overhead (memory allocation and deallocation) and more cache hits in the CPU for the processes with lower memory utilisation.

We show in the following that most measured parameters are approximately the same for $t_{off} = 5$ s as compared to $t_{off} = 40$ s. This means that for a memory limitation of 3 GB an equivalent of 1.2 million clients can be simulated (for $t_{off} = 5$ s) instead of $150,000$ (for $t_{off} = 40$ s). The total average throughput can be approx. 15 Gbit/s as compared to 1.86 Gbit/s for $t_{off} = 40$ s.

B. Average Link Load

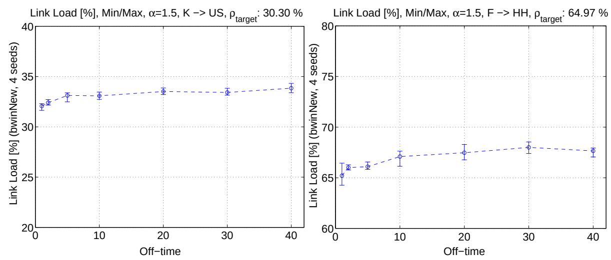
We compare the average traffic load measured in the simulations with the target traffic load (from measurements at the real network, see Sec. III). Although our estimation of the required number of clients based on (1) tries to keep the load constant at the measured values (Fig. 2 (b)), small deviations from the ideal behaviour can be observed.

The measurements depicted in Fig. 4 and Fig. 5 show that the average load is approximately constant for $t_{off} > 5$ s (please note the different ordinate scales). The minimum and maximum measurements (vertical bars) have a larger difference from the average value for smaller values of α, as can be expected since the traffic is more bursty in this case. The load value is in most cases slightly larger than the target load values[1] of $30.3\,\%$ and $64.97\,\%$, respectively.



(a) K->US (b) F->HH

Fig. 4. Average traffic load over off-time, $\alpha = 1.1$.



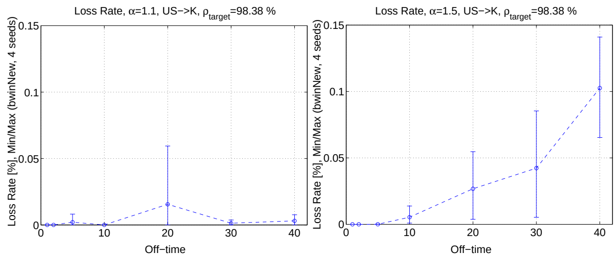
(a) K->US (b) F->HH

Fig. 5. Average traffic load over off-time, $\alpha = 1.5$.

We can summarise that the average traffic load is approximately constant for $t_{off} > 5$ s with only few exceptions and matches almost the measurements in the real world network. Thus a reduction of the number of modelled clients by a factor of 8 seems to be feasible.

C. Loss Probability

The packet loss probability of the link with the highest load of $\rho = 98.4\,\%$ is shown in Fig. 6 (router buffer: 2500 packets). The loss probability depends for $\alpha = 1.5$ on the off-time and thereby on number of clients as can be seen in Fig. 6 (b), but the value remains below $0.15\,\%$ and therefore the significance of this dependency is rather low.



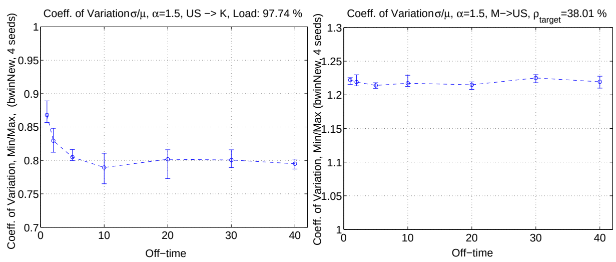
(a) $\alpha = 1.1$ (b) $\alpha = 1.5$

Fig. 6. Loss probability over off-time, “US->K”.

D. Coefficient of Variation of Inter-Departure Times

We measured the coefficient of variation, defined as $CV = \sigma/\mu$, with standard deviation σ and mean value μ. The coefficient of variation is a measure of the relative dispersion. It is used here to evaluate the change of the variability of the inter-departure times when the number of clients is decreased. The inter-departure times are measured at the queueing module that is used to limit the bandwidth of the links. The queueing module is located at every outgoing port of the router.

The behaviour of the coefficient of variation for different off-times (number of clients) is shown in Fig. 7 (a) for $\alpha = 1.5$, link “US->K”. The value decreases with increasing off-time (number of clients), but it is constant for $t_{off} > 5$ s. The coefficient of variation at link “M->US” (Fig. 7 (b)) does not show any clear tendency, the value is more or less constant, independent of the number of clients.



(a) US->K (b) M->US

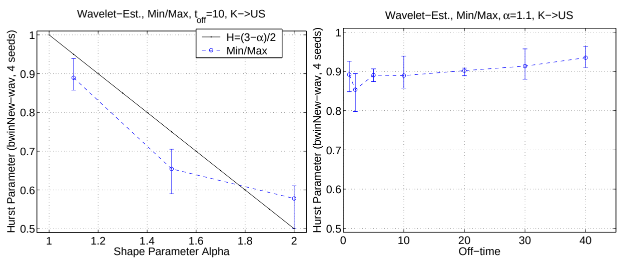
Fig. 7. Coefficient of Variation over off-time, $\alpha = 1.5$.

[1]The target load value can be calculated as the sum of all measured flow throughput values from Fig. 3 that are routed of the corresponding link divided by the link capacity.

We would like to note that recent measurements of inter-arrival times conducted in [17], [18] showed values very similar to our simulation results. The coefficient of variation of the inter-arrival times in those measurements was in a similar range as the values in our simulations ($[1.01, 1.26]$), which confirms that the simulations match the reality quite well. We can conclude that the coefficient of variation of the inter-departure times is nearly not affected by changing the number of clients in the system, the changes are very small (less than 2.5 %) and the absolute values are in ranges confirmed by recent measurements.

E. Hurst Parameter

The Hurst parameter represents the degree of self-similarity of the network traffic. We have used the Wavelet estimator [19] to estimate the Hurst parameter of the counting process (number of bytes per measurement window) on each link. The measurement window of the counting process was set to 2.5 ms. The Hurst parameter can be tuned via the shape parameter α [9] and follows in this case approximately $H = (3-\alpha)/2$ (infinite superposition of on/off sources), as shown in Fig. 8 (a) and discussed in [2].



(a) $t_{off} = 10$ s (b) $\alpha = 1.1$

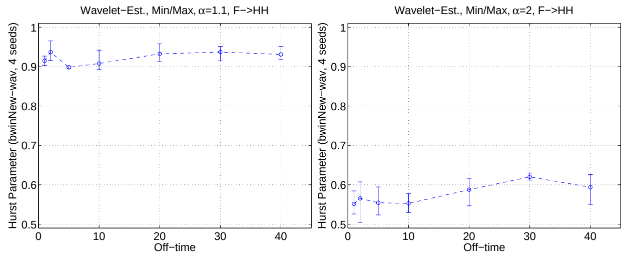
Fig. 8. Hurst Parameter, K->US.

The influence of the off-time / number of clients on the Hurst parameter for $\alpha = 1.1$ at link "K->US" is shown in Fig. 8 (b). The value is approximately constant for $t_{off} > 5$ s. The measurements on link "F->HH" in Fig. 9 (a) and Fig. 9 (b) show a nearly constant line for all off-time values. The measurements on other links and for different values of α did not show a qualitatively different behaviour from the figures shown here.

We can summarise that the Hurst parameter is almost independent on the off-time and therefore on the number of clients.

F. Average End-To-End Delay

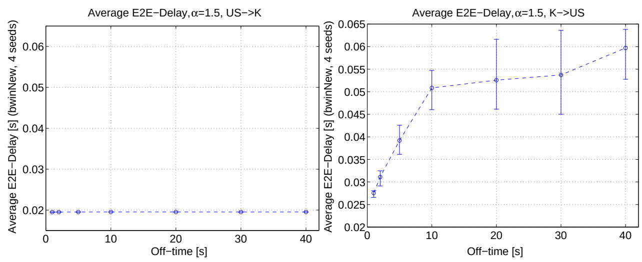
The average end-to-end delay is a very important measure since it has a major impact on the user perceived quality of service. The end-to-end delay of the data packets was measured and averaged over all clients in the same flow. The end-to-end delay of flow "US->K" is constant for all number of clients, as can be seen in Fig. 10 (a). The measurement in the opposite direction (flow "K->US", Fig. 10 (b)) show the



(a) $\alpha = 1.1$ (b) $\alpha = 2.0$

Fig. 9. Hurst Parameter over off-time, F->HH.

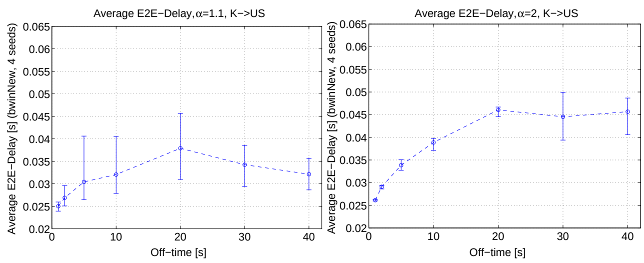
strongest dependency on the number of clients of all flows. However, the value is approximately constant for $t_{off} > 10$ s.



(a) US->K (b) K->US

Fig. 10. Average end-to-end delay over off-time, $\alpha = 1.5$.

The average end-to-end delay depends strongly on the degree of self-similarity: the case of very high Hurst parameter ($\alpha = 1.1$, Fig. 11 (a)) is dominated by very large variability (bars of min. and max. range) but the range of the values is nearly constant for $t_{off} > 5$ s. The dependency on the number of clients is much stronger In the case of very low Hurst parameter ($\alpha = 2.0$, Fig. 11 (b)), the value is only constant for $t_{off} > 20$ s.



(a) $\alpha = 1.1$ (b) $\alpha = 2.0$

Fig. 11. Average end-to-end delay over off-time, "K->US".

We summarise that the average end-to-end delay is approximately constant for $t_{off} > 10$ s. One exception is the case of very low degree of self-similarity ($\alpha = 2.0$), which is only of marginal relevance for current internet traffic.

V. CONCLUSION

We have considered the problem of reducing the complexity of network simulations with HTTP/TCP clients by reducing the number of clients and increasing their activity at the same time by reducing the off-time (Sec. I). The target was to evaluate the effect of reducing the number of modelled clients in a simulation scenario, where its complexity still allowed us to simulate the network also with a realistic client population. Non-critical scale-down parameters are then considered to be applicable to larger networks where the simulation with a realistic number of clients is not possible.

We have discussed the measurements of the following parameters: average link load, coefficient of variation of the inter-departure times, Hurst parameter and average end-to-end delay. We have shown that the these parameters are approximately constant for $t_{off} \geq 10$ s in all relevant cases, that is when only 28560 clients are used instead of $112,976$. The gain is that the simulation requires only 520 MB RAM instead of 2000 MB RAM; simultaneously simulation time was reduced by up to $30\,\%$. For many cases even simulations with $t_{off} = 5$ s (14495 clients, 270 MB) yield a sufficient accuracy for the shown measures.

We have shown in this study that a realistic simulation of large network topologies is possible by reducing the number of clients by a factor of $4 - 8$ as compared to reality. The savings allows to drive realistic simulations corresponding to an equivalent of up to 1.2 million modelled clients with an average total throughput of approximately 15 Gbit/s.

REFERENCES

[1] S. Floyd and V. Paxson, "Difficulties in simulating the internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, Aug. 2001. [Online]. Available: http://www.icir.org/fbyd/papers/simulate_2001.pdf

[2] K. Below and U. Killat, "Generating prescribed traffic with HTTP/TCP sources for large simulation models," in *Kommunikation in Verteilten Systemen (KiVS), accepted for presentation*, Leipzig, Germany, Feb. 2003.

[3] ——, "Internet traffic generation for large simulations scenarios with a target traffic matrix," in *Advances in Communications and Software Technologies (WSEAS ICOSMO 2002)*. Skiathos, Greece: WSEAS Press, Sept. 2002, pp. 146–151, ISBN: 960-8052-71-8. [Online]. Available: http://www.tu-harburg.de/et6/papers/documents/Below_Kai/below-ICOSMO2002-c.pdf

[4] ——, "On the configuration of simulations of large network models with HTTP/TCP sources," in *15th ITC Specialist Seminar, Internet Traffic Engineering and Traffic Management*, P. Tran-Gia and J. Roberts, Eds., Würzburg, July 2002, pp. 143–149. [Online]. Available: http://www.tu-harburg.de/et6/papers/documents/Below_Kai/below-ITC-SPEC-Seminar2002-c.pdf

[5] H.-K. Choi and J. O. Limb, "A behavioral model of web traffic," in *Proceedings of the Seventh Annual International Conference on Network Protocols, Toronto, Canada*, Nov. 1999. [Online]. Available: http://www.cc.gatech.edu/computing/Telecomm/people/Phd/hkchoi/paper/icnp99.pdf

[6] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," in *ACM SIGCOMM*, D. P. Sidhu, Ed., San Francisco, California, 1993, pp. 183–193. [Online]. Available: citeseer.nj.nec.com/leland93selfsimilar.html

[7] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Transactions on networking*, vol. 5, no. 6, pp. 835–846, Dec. 1997. [Online]. Available: http://www.cs.bu.edu/faculty/crovella/paper-archive/self-sim/journal-version.ps

[8] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, June 1995. [Online]. Available: ftp://ftp.ee.lbl.gov/papers/WAN-poisson.ps.Z

[9] K. Park, G. Kim, and M. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proc. IEEE International Conference on Network Protocols*, Oct. 1996, pp. 171–180. [Online]. Available: http://citeseer.nj.nec.com/article/park96relationship.html

[10] U. of California at Berkeley, "Ptolemy classic," http://ptolemy.eecs.berkeley.edu/ptolemyclassic/body.htm.

[11] UCB/USC/LBNL/VINT, "Network simulator ns (version 2)," Avaliable from http://www.isi.edu/nsnam/ns/.

[12] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed propability distributions in the world wide web," in *A Practical Guide yo Heavy Tails: Statistical Techniques and Applications*, R. J. Adler, R. E. Feldmann, and M. S. Taqqu, Eds., 1998, pp. 3–25. [Online]. Available: http://math.bu.edu/people/murad/pub/www4-posted.ps

[13] M. Greiner, M. Jobmann, and L. Lipsky, "The importance of power-tail distributions for modeling queueing systems," *Operations Research*, vol. 47, no. 2, 1999. [Online]. Available: http://www.eng2.uconn.edu/~lester/papers/pow198.ps.gz

[14] H. Gogl, M. Greiner, and H.-P. Schwefel, "Model calibration," Technische Universität München, Institut für Informatik, Tech. Rep., Dec. 1997, http://citeseer.nj.nec.com/gogl97model.html.

[15] "The linux kernel archives, kernel 2.4, include/asm-i386/page.h, PAGE_OFFSET," Jan. 2003, http://www.kernel.org/pub/linux/kernel/v2.4/.

[16] "Microsoft corporation, memory support and windows operating systems," Jan. 2003, http://www.microsoft.com/hwdev/platform/server/PAE/PAEmem.asp.

[17] "Abilene-I data set, IPLS-KSCY-20020814-102000-1," Aug. 2002, http://pma.nlanr.net/Traces/long/ipls1.html.

[18] "NZIX-II trace archive, NZIX-20000710-120000," July 2000, http://pma.nlanr.net/Traces/long/nzix2.html.

[19] P. Abry and D. Veitch, "Wavelet analysis of long-range-dependent traffic," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 2–15, 1998. [Online]. Available: citeseer.nj.nec.com/abry98wavelet.html