Internet Traffic Generation for Large Simulations Scenarios with a Target Traffic Matrix

KAI BELOW, ULRICH KILLAT
Department of Communication Networks
Technical University Hamburg-Harburg (TUHH)
Denickestr. 17, D-21071 Hamburg, Germany
{below, killat}@tu-harburg.de http://www.tuhh.de/et6/
Published at WSEAS ICOSMO, Skiathos, Greece, September 2002.

Abstract: - The simulation of realistic networks is an important aspect in the development of new protocols or algorithms for communication networks. The traditional approach for such a simulation is to connect a certain amount of sources to network nodes and to measure the link load, end-to-end delay etc. in the simulation in order to evaluate the performance of the network.
Network providers have a different view: the traffic intensities in their networks are known, but their target is to optimize the current network. Therefore the first step is to configure the simulation model to generate the same traffic that was measured in the real network. We prescribe traffic intensities via a traffic matrix and propose a method for the allocation of sources for all flows that minimizes the difference between prescribed and measured traffic matrix.
We use HTTP/TCP source models for traffic generation since current Internet traffic is mostly TCP traffic. The reactivity of TCP makes it hard to estimate the average rate of a single source, especially since TCP operates not in steady state here due to the small average of the HTTP transfer volumes in the Internet. Nevertheless we derive a simple estimate on the average rate per HTTP/TCP source in the paper that only uses network and protocol properties that are known before carrying out the simulation with the actual parameters.
The performance of the proposed algorithm for the allocation of sources was evaluated by simulating three network topologies with various parameters. We show in this paper that the performance of the method is good under various conditions.

Key-Words: - HTTP, TCP/IP, Simulation, Traffic Generation

1 Introduction

The classical network simulation experiments build on the following setting: models of network nodes are interconnected to form a certain network model. The nodes are then equipped with a routing table and a set of sources (producing packets for prescribed destinations). As a result of the simulation, traffic intensities on links and delays for the various flows can be determined which are subject to different parameter setting optimizations like protocol variants used, buffer management and service disciplines in routers.

Network operators have a different problem: they know the traffic on their links and would like to know how the network state is affected when parameter settings are changed. To answer this question it is necessary to find a configuration for the simulation model so that the simulated traffic matches closely the measured traffic. In order to solve this problem the number of sources for each flow has to be determined, which would produce the known, observed traffic. This is not a trivial problem since the steady state formulas for the TCP throughput [1] are not applicable for short-time connections that are characteristic for WWW traffic.

We investigate this problem for a network which mainly supports HTTP applications and thus builds on the reactive behavior of TCP. The intrinsic features of the WWW traffic within the last years are the user behavior and the use of the HTTP/TCP protocols and therefore it is important to use source models that reflect this behavior as opposed to generating aggregated traffic [2]. The traffic measured in the Internet is known to be self-similar [3, 4]. Self-similar traffic is usually generated with power-tail distributed random variables [4]. We use the truncated power-tail distribution [5, 6] for the HTTP volume to be transmitted via TCP. The selection of the distributions and their parameter fittings for the generation of self-similar traffic can be found in [7].

We use the simulation software Ptolemy [8] for which we implemented TCP "NewReno" 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. We do not model the usage of parallel connections here. The TCP implementation was validated with ns-2 [9]. The HTTP request of a client opens a TCP connection via the three way handshake. A request for all embedded objects (includes) is 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. This on / off behavior is the key for being able to derive a simple estimation of the average rate of such a traffic source, especially since the off-time (reading time) is in general much larger than the on-time [7].

The rest of the paper is organized as follows. The proposed algorithm for the allocation of sources is described in Section 2. The three simulation scenarios used for the validation of the new method are illustrated in Section 3. The simulation results are shown in Section 4. We conclude the paper with a summary of the findings of the paper in Section 5.

2 Algorithm for the Allocation of Sources

We recommend to change stochastic distributions that contribute to the mean on-time to constant values of their respective mean values for a faster convergence of the simulation and to change back to the probability distributions after reaching a good allocation of sources. Especially power-tailed probability distributions, that are used to generate self-similar traffic, are known to converge very slowly to their corresponding mean value, so that it could be, that the achieved traffic matrix is different from the specified one not due to a wrong allocation of sources but due to a short simulation time so that the probability distributions did not converge to their respective mean values. A distribution without a power-tail, like the negative exponential distribution used in our simulations, should be used for the off-time in order to desynchronize the sources.

Due to the granularity of integer values of the number of sources, the resulting accuracy is limited and depends on the specified off-time and traffic matrix: a larger off-time lets the average rate per source decrease and therefore the granularity is finer such that the accuracy of the method increases. The smallest values of the traffic matrix encounter the largest problem with the granularity. We have presented an early state of this algorithm in [10] that focuses on the iterations and shows that the algorithm converges fast.

2.1 Estimation of the Source Allocation

We regard the traffic matrix TP as given, the elements tp_{ij} denote the throughput of the flow between node i and node j in Mbit/s. The target is to estimate a source allocation S that generates traffic as described with TP. We derive an estimation of the average rate of a single HTTP/TCP source that can be applied for all flows so that the source allocation S can be achieved. The average rate of a single HTTP/TCP source in the flow from node i to node j is

$$r_{src_{ij}} = \frac{v}{t_{on_{ij}} + t_{off}} \quad [Mbit/s], \tag{1}$$

where v represents the average HTTP transfer volume in Mbit

$$v = b \cdot 8/10^6 \tag{2}$$

and b is the average HTTP transfer volume in Bytes. The quantities $t_{on_{ij}}$ and t_{off} are the average HTTP on- and off-times, respectively (download time and reading time). The distributions and parameters of t_{off} and v are given quantities from measurements of the user behavior, e.g. from [7]. The value of $t_{on_{ij}}$ depends on v and is in general hard to estimate due to the reactivity of the TCP protocol. Measurements of HTTP traffic [7] have shown that the mean value of t_{off} is in general much larger than $t_{on_{ij}}$ so that t_{off} and v dominate the (1). Therefore a very simple estimation of $t_{on_{ij}}$ can already be sufficient for a good estimate of the average throughput per source $r_{src_{ij}}$.

Despite the dominance of v and t_{off} in 1 we derive a very simple estimate for $t_{on_{ij}}$ in order to increase estimation accuracy and applicability area of the formulas. A simple estimate of the mean on-time is

$$t_{on_{ij}} = n_{RTT} \cdot rtt_{ij}, \tag{3}$$

where rtt_{ij} is the average RTT and n_{RTT} is the number of RTTs required by TCP to transmit the volume v. The parameters v and n_{RTT} do not depend on the flow and n_{RTT} can be calculated once for the whole network. The average RTT depends on the networks dynamics and can not be known in advance. Since rtt_{ij} depends strongly on the flow, we calculate a lower bound of the RTT by summing up the corresponding link delays fore each flow.

The number of required round-trip times n_{RTT} for the transmission of v can be calculated as follows (packet loss is ignored here):

$$n_{RTT} = 1 + n_{RTT_{ss}} + n_{RTT_{ca}}. \tag{4}$$

The value of 1 RTT in (4) represents the first two packets of the TCP connection setup with the three-way handshake[1]. The on-time is finished when the client receives the last data packet (acknowledgments and closing of the connection is counted as off-time). The parameters $n_{RTT_{ss}}$ and $n_{RTT_{ca}}$ represent the number of RTTs that TCP requires to transmit b Bytes slow-start and congestion avoidance phase, respectively. The total number of packets n_P for transmitting b Bytes is

$$n_P = \left\lceil \frac{b}{MSS} \right\rceil \tag{5}$$

with MSS representing the maximum segment size of TCP packets (we used a value of $MSS = 1460$ Bytes here). The maximum number of packets that can be sent in an RTT is the ratio of the maximum congestion window and the maximum segment size:

$$n_{P,CWND_{max}} = \left\lfloor \frac{CWND_{max}}{MSS} \right\rfloor. \tag{6}$$

The maximum number of RTTs that TCP stays in slow-start phase (without losses) is:

$$n_{RTT_{ss,max}} = \lceil \log_2(n_{P,CWND_{max}}) \rceil. \tag{7}$$

The maximum number of packets that can be transferred in slow-start, can be calculated with

$$n_{P_{ss,max}} = 2^{n_{RTT_{ss,.max}}+1} - 1, \tag{8}$$

where $CWND_{max}$ denotes the maximum TCP congestion window (we use $CWND_{max} = 65535$ Bytes). The number of RTTs in slow-start follows with (5), (7) and (8) as:

$$n_{RTT_{ss}} = \begin{cases} \lceil log_2(n_P) \rceil & n_P \le n_{P_{ss,max}} \\ n_{RTT_{ss,max}} & n_P > n_{P_{ss,max}}. \end{cases} \tag{9}$$

[1] The third packet of the three-way handshake can be directly followed by the HTTP request so that it is not counted in (4).

The number of RTTs in congestion-avoidance phase can be written as the remaining number of packets after slow-start phase divided by the maximum number of packets that can be transmitted (with the assumption that no loss occurs):

$$n_{RTT_{ca}} = \left\lceil \frac{\max(0, n_p - n_{P_{ss,max}})}{n_{P,CWND_{max}}} \right\rceil. \tag{10}$$

Using (1), (3) and (4) we can write the final result for the average source rate of a single HTTP/TCP source:

$$\begin{aligned} r_{src_{ij}} &= \frac{v}{n_{RTT} \cdot rtt_{ij} + t_{off}} \\ &= \frac{v}{(1 + n_{RTT_{ss}} + n_{RTT_{ca}}) \cdot rtt_{ij} + t_{off}}. \end{aligned} \tag{11}$$

The source allocation matrix S with the elements s_{ij} representing the number of sources for the flow from node i to node j can be calculated finally by dividing the throughput tp_{ij} by the average source rate $r_{src,ij}$ for each flow and rounding the resulting numbers:

$$\begin{aligned} s_{ij} &= round\left(\frac{tp_{ij}}{r_{src_{ij}}}\right) \\ &= round\left(\frac{tp_{ij}}{v} \cdot [n_{RTT} \cdot rtt_{ij} + t_{off}]\right). \end{aligned} \tag{12}$$

3 Simulation Scenarios

We use three simulation scenarios in this study: the bottleneck scenario (Fig. 1), a so called parking-lot scenario with four main links and different RTTs and hop-counts for the flows (Fig. 2) and the B-WiN model, a large network model that is representative of an existing backbone (Fig. 3, see also the technical report [11]).

The bottleneck scenario is very simple, all flows share the same link between two routers. We installed different link delays so that the flows experience different minimum RTTs: the bottleneck link 50 Mbit/s and 8 ms, S1, S2 and S3 are connected via 100 Mbit/s and 1 ms, 5 ms and 10 ms, respectively. The installed flows are depicted as dashed arcs. The target traffic matrix is set to 11 Mbit/s for each flow resulting in an average link utilization of 66% at the bottleneck link.

The parking-lot scenario has the advantage of being already complex enough to show the effect of different round-trip times and hop counts but still being simple enough to provide fast simulations and an easy

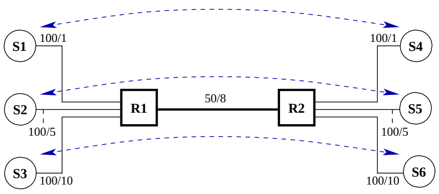

Figure 1: Topology of the bottleneck simulation model (link capacities in Mbit/s and propagation delay in ms).

interpretation of the results. All links between the source nodes S1 – S12 and the router nodes R1 – R5 are configured with a rate of 1 Gbit/s and a propagation delay of 1 ms so that there is no bottleneck there. The routers are connected via links with 50 Mbit/s. The links from R1 to R2 and R4 to R5 have a propagation delay of 8 ms, the remaining two links have a delay of 24 ms.

The specified traffic matrix for the parking-lot scenario is very simple: all flows should achieve throughput of 8.5 Mbit/s which results in a maximum average link load of 68 % (between R2 and R4). Although this traffic matrix is simple, it is nevertheless hard to find a source allocation: due to TCP's sensitivity to different round-trip times and packet losses the six-hop connections can not achieve the same rate as the shorter connections so that relatively more sources are needed in longer flows.

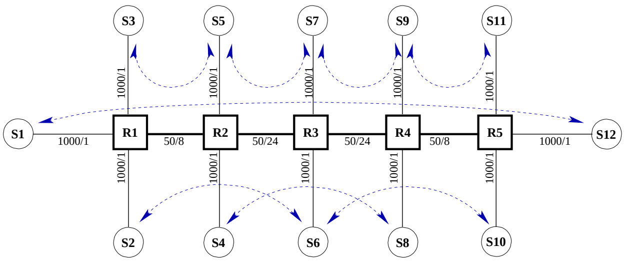

Figure 2: Topology of the parking lot simulation model (link capacities in Mbit/s and propagation delay in ms).

The B-WiN scenario (Fig. 3) is very complex, takes a long time to simulate and it is harder to understand the results. Nevertheless, it is a model of an early state of the network that connects all German universities and research institutes. The results of simulations with this model can be regarded as a test case to evaluate real network behavior. 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. The network is virtually fully meshed: all nodes have connections to all neighbor nodes which results in 110 flows. The routing was optimized by the providers in order to keep the traffic load below 70 % on all links except for those connecting the german part of the network with the USA (node "US/11"); these links were allowed to have a load around 90 %.

We used two variants of the traffic matrix to test the performance: the original, measured traffic matrix (B-WiN-1) and a second variant, where the throughput values where reduced so that the maximum value of the average link loads is limited to 70 % instead of 98.4 % (B-WiN-2). We expected a degradation in estimation accuracy for high loaded links (larger RTTs, different loss rates) since these effects are not covered in the estimation described in Section 2.1.

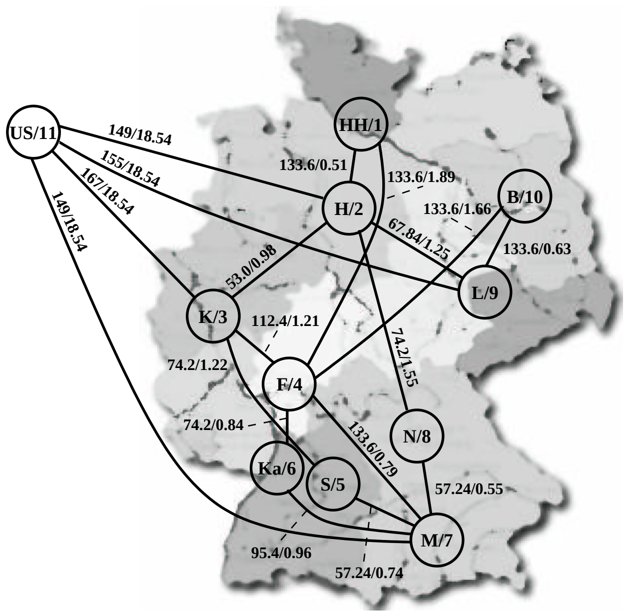

Figure 3: Topology of the B-WiN simulation model (link capacities in Mbit/s and propagation delay in ms).

The IP-routers are modeled as output buffered systems with FIFO queues and the time to find the route was neglected. All buffers where used in packet mode.

4 Simulation Results

The simulation scenarios can be set up with the findings of Section 2.1 and the measurements of the user behavior for the parameters average HTTP transfer volume b and off-time t_{off}. We use here values of $b \approx 60$ KB and $t_{off} \approx 40$ s for realistic settings, similar to the findings in [7]. These values are not usable in all cases: the scenario B-WiN-1 needs approximately a total of 120000 sources according to our estimations. The simulation of 7000 sources requires already 1 GB of RAM with our simulator and the simulation speed decreases with increasing number of sources. We limited the total number of sources to 7000 in this study. This can be achieved by either increasing the average HTTP transfer volume b or by decreasing the off-time t_{off}. Increasing the parameter b changes the characteristic of the traffic dramatically since TCP would operate more in congestion-avoidance phase than in slow-start phase. Reducing t_{off} on the other hand introduces smaller changes to the traffic dynamics since only the idle-phase of the source is affected. Nevertheless, we also vary the parameter b in this study in order to show that the proposed source allocation method is working for various parameter ranges and scenarios.

The simulation results for all scenarios are summarized in Table 1. The simulation scenario B-WiN-2 refers to the traffic matrix limited to produce an average link load of not more than 70% and the scenario B-WiN-1 uses the original, measured traffic matrix. Positive Δ-values represent larger quantities than estimated or given, negative values mean that the measured values are lower than the estimated or given quantities. The simulation time was chosen to be bigger or equal to $10 \cdot t_{off}$; some tests with longer simulations showed similar results. The performance was measured with three parameters: ΔTP_s: the sum of the differences between given and measured traffic matrix, ΔTP_{sa}: the sum of the absolute differences divided by the sum of TP, $\Delta tp_{ij,m}$: the maximum value of the absolute difference in throughput in all flows in percent of the given throughput value.

The granularity of integer numbers for the number of sources in the flow from node i to node j n_{ij} and the total number of sources n leads as a matter of principle to a coarser granularity of the maximum error $\Delta tp_{ij,m}$ when n decreases. Realistic parameters for i and t_{off} are $i \approx 6$ and $t_{off} \approx 40$. For these settings all data is transferred in the slow-start phase of TCP (neglecting losses). Increasing the HTTP volume to $i = 20$ means an increase of t_{on} so that already more than 80% of the data is transfered in congestion avoidance phase. Ignoring losses in congestion-avoidance produces certainly a larger error than in slow-start in the estimation so that the error increases (for a constant value of t_{off}).

All simulations results show the good performance of the proposed source allocation: ΔTP_{sum} is within $\pm 6.5\%$ error, $\Delta TP_{sum,abs}$ is less than 8.5% and $\Delta tp_{ij,max}$ is in the interval $[-11.9, 18.8]$. The performance of the algorithm is better for the reduced link rates in B-WiN-2 than in B-WiN-1 as expected.

5 Conclusion

We have considered the problem of calibrating the allocation of HTTP traffic sources for large simulation models of TCP/IP networks. The target was to minimize the difference between a prescribed and the measured traffic matrix. We have derived an approximate solution for the average source rate of the modeled HTTP/TCP sources and we have proposed a method to distribute the sources in order to meet the given traffic matrix. We have shown with three simulation scenarios and various simulations that the performance of the proposed method is reasonably good: in 22 of the 28 cases the maximum error was smaller than $\pm 10\%$. We can conclude that the proposed method for the source allocation solved the problem: the method performs good in all tested cases. Future work will focus on modeling and evaluation of the performance of the method for different loss rates.

References:

[1] Robert Morris, "Scalable TCP congestion control," in *IEEE INFOCOM 2000, Tel Aviv*, Mar. 2000, pp. 1176–1183.

[2] Sally Floyd and Vern Paxson, "Difficulties in simulating the internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, Aug. 2001.

[3] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson, "On the self-similar nature of ethernet traffic," Tech. Rep., Bellcore, Boston University, Mar. 1993.

[4] Mark E. Crovella and Azer Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Transactions*

Table 1: Simulation results. Buffer $B = 5000$ IP-packets, i: number of HTTP objects of size 10 KB, t_{off}: off-time [s].

Scenario	i	t_{off}	n	ΔTP_s [%]	ΔTP_{sa} [%]	$\Delta tp_{ij,m}$ [%]
bottleneck	6	40	5526	5.8	5.8	7.2
	10	40	3318	5.9	5.9	7.0
	15	40	2214	5.3	5.3	6.6
	20	40	1662	4.4	4.4	8.3
	6	10	1404	5.3	5.3	6.9
	10	10	840	4.0	4.0	6.3
	15	10	564	4.0	4.0	5.5
	20	10	424	3.8	3.8	6.0
parking-lot	10	40	6854	5.3	5.3	7.8
	20	40	3434	4.3	4.3	10.8
	6	20	5768	5.1	5.1	9.6
	10	20	3458	5.0	5.0	11.5
	15	20	2314	4.5	4.5	8.1
	20	20	1738	4.4	4.7	12.2
	6	10	2934	4.5	4.5	8.4
	10	10	1758	3.2	3.6	7.6
	15	10	1180	3.9	4.3	11.8
B-WiN-2	20	5	4450	2.6	5.0	9.9
	6	2	6104	3.2	3.9	8.5
	6	1	3241	0.8	3.8	5.4
	10	1	1944	-2.4	5.0	-5.4
	15	1	1318	-4.3	6.5	-7.2
	20	1	1000	-6.4	8.5	-11.9
B-WiN-1	20	5	4668	2.1	5.2	18.8
	6	2	6442	3.0	4.8	-5.1
	6	1	3447	0.7	4.7	-5.1
	10	1	2073	-2.3	6.1	-7.5
	15	1	1404	-4.2	7.3	-7.9

on networking, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[5] Hans-Peter Schwefel and Lester Lipsky, "Buffer size issues in the presence of self-similar traffic," in *3rd IFIP Workshop on Traffic Management and Design of ATM Networks*, London, Apr. 1999.

[6] Hans-Peter Schwefel and Lester Lipsky, "Performance results for analytic models of traffic in telecommunication systems, based on multiple on-off sources with self-similar behavior," *To be presented at the 16th International Teletraffic Congress*, June 1999.

[7] Hyoung-Kee Choi and John O. Limb, "A behavioral model of web traffic," in *Proceedings of the Seventh Annual International Conference on Network Protocols, Toronto, Canada*, Nov. 1999.

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

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

[10] Kai Below and Ulrich Killat, "On the configuration of simulations of large network models with HTTP/TCP sources," in *5th ITC Specialist Seminar, Internet Traffic Engineering and Traffic Management*, Würzburg, July 2002.

[11] Kai Below, Carsten Schwill, and Ulrich Killat, "Erhöhung des Nutzungsgrades eines ATM Netzes für den Wissenschaftsbereich (ERNANI),"

Tech. Rep., Dept. Communication Networks, Technical University Hamburg-Harburg, Sept. 2001, http://www.tu-harburg.de/et6/papers/documents/Below_Kai/Abschlussbericht-ERNANI.pdf.