A Black-Box Approach to Latency and Throughput Analysis

Daniel Brahneborg
Infoflex Connect AB
Stockholm, Sweden
daniel.brahneborg@infoflexconnect.se

Wasif Afzal, Adnan Čaušević
Software Testing Laboratory, Mälardalen University
Västerås, Sweden
{wasif.afzal,adnan.causevic}@mdh.se

***Abstract*—To enable fast and reliable delivery of mobile text messages (SMS), special bidirectional protocols are often used. Measuring the achieved throughput and involved latency is however non-trivial, due to the complexity of these protocols. Modifying an existing system would incur too much of a risk, so instead a new tool was created to analyse the log files containing information about this traffic in a black-box fashion. When the produced raw data was converted into graphs, they gave new insights into the behaviour of both the protocols and the remote systems involved.**

***Keywords*-log file analysis; latency; black-box**

I. Introduction and Problem Description

Mobile text messages, usually called SMS (short for "Short Message Service"), have some interesting characteristics. In the simplest use case, the SMS goes from the sender's device (usually a mobile phone) to the mobile network operator, and then on to the recipient's device. Quite often, it must however also go from one network operator to another, and sometimes either the sender or the recipient is a client running an application on Internet, e.g. for meeting reminders or when voting on TV shows. As there is a cost associated with sending an SMS, the sender must have an account with the recipients network operator, making it possible for the operator to charge the sender the correct amount. To avoid a combinatorial explosion of these accounts, the client with the Internet application does not need an account with every possible operator, but instead has a single account with an "SMS broker". This SMS broker then maintains accounts with the operators, enabling traffic between them all.

Without going into too much detail about the protocols, lets define some common terminology:

Node: Used to refer to both clients, operators and the zero or more SMS brokers between them.

Request: A data packet containing an SMS message, including the sender, recipient and message body, sent in the direction towards the recipient.

Response: A data packet going in the opposite direction to the request, containing a message id. This message id is used for delivery reports and troubleshooting.

PDU: Short for Protocol Data Unit (PDU), and is used to refer to both requests and responses.

Latency: The time elapsed between a request and its corresponding response.

Throughput: The number of SMS messages received by one node and forwarded on to the next node, per some specified time unit.

While the SMS brokers reduce the total number of accounts needed, they also make the combined network traffic more difficult to analyse. Some of these brokers specialise in operators in a particular region, decreasing the total number of accounts further but increasing the number of nodes each message must pass to reach its destination. How the messages are handled by each broker varies, e.g. they may store the messages on disk to avoid losing data and thereby also revenue in case there is a failure, or wait for acknowledgment from the next node before they respond back to the previous node.

With this context in mind, we want to know more about the latency and throughput achieved in a production system of an SMS broker. As the latency is dependent on so many factors, including remote systems with proprietary architectures, we argue that finding this measurement must be done in a black-box manner. We opted for using the available log files, as that would also let us determine the throughput, without having to modify the core product and risk introducing bugs.

II. Case Study: Enterprise Messaging Gateway

We have examined the log files from Enterprise Messaging Gateway (EMG), an application from Infoflex Connect AB, used by many SMS brokers. EMG produces log files containing an entry for each transferred PDU. As sliding windows can be used to increase throughput when latency is high, and the responses can arrive in any order, the request and the response can be quite far apart in the log file.

III. Solution Proposal

The workflow for the proposed solution is shown in Fig. 1. EMG generates log files with timestamped entries, and the new tool ELFA (EMG Log File Analyser) reads these files and calculates the requested statistics. This information can then be loaded into R Studio[1] to generate graphs and perform more advanced calculations on the raw data.

[1] `https://www.rstudio.com`

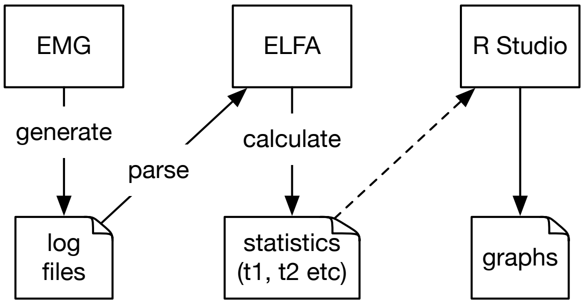

Figure 1. Our solution proposal workflow.

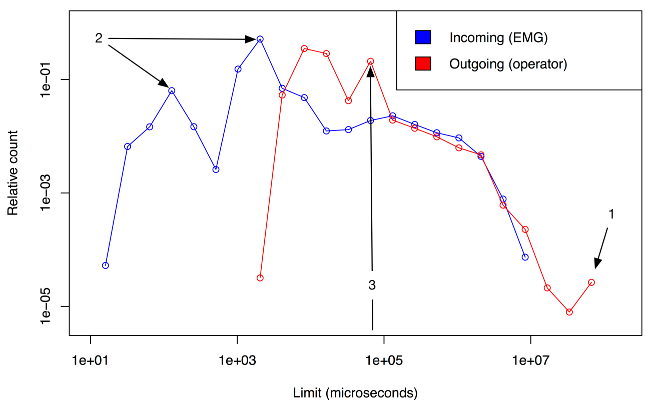

Figure 2. Latency between EMG (incoming) and an operator (outgoing).

IV. Preliminary Results

ELFA was used on log files for several operators, containing information on 4 months of SMS traffic in Sweden. The output for all the examined operators were similar.

A. Latency

The ratio of request and response pairs which complete within a certain time for one of the operators are shown in Fig. 2. The X-axis is the time limit in μs, and the Y-axis is the relative number of pairs, in both cases on a logarithmic scale. The larger ratio that each dot represents, the higher up it is. The blue line shows the response times for incoming traffic to EMG, and the red line the response time for outgoing traffic to the operator. We can see that the operator sometimes responds very quickly, shown by the red line starting close to 1e+03 μs (1 ms).

The right end of the red line, with an unusually large number of messages with a latency of more than 30 seconds, indicate that something problematic happened. This data point is marked with "1". Defining the exact conditions for what constitutes such outliers, while minimising false positives, is left as future work.

For incoming traffic, there are two clear peaks in the diagram. A closer investigation reveals that the operator sends "keep-alive" PDUs, which is a special type used on otherwise idle connections, to verify that the remote side is still alive. These requests require less processing by EMG than regular messages, and this difference is reflected in these two peaks on the blue line, marked with "2", being so far apart. Other factors may lead to similar peaks, but if so, they are too small or too close to be visible in this diagram.

Similar peaks can be seen for the outgoing traffic as well, but here the "keep-alives" correspond to the peak to the right. This suggests that the software on the operator side has a different architecture than EMG. Their support department confirmed an explicit delay of 50 ms for these responses, in agreement with this peak, in Fig. 2 marked with "3".

Mosberger and Jin [1] created the tool `httperf` for testing HTTP traffic. In one of their examples, they got the reply rates "min 139.8 avg 148.3 max 150.3" (as requests per second), which suggests that most of the requests get a reply at the maximum speed, while a smaller ratio are slower. The same type of distribution is seen here.

B. Throughput

Another part of the ELFA output, not included here, showed that the throughput never went above about 50 messages per second for this operator. Being allowed to send messages at a higher speed costs more, so this limit is important when negotiating terms with both SMS brokers and operators.

V. Conclusions & Future Work

Understanding the network usage is important for maximizing the cost/benefit ratio for network intensive applications. By data mining existing log files, values for both latency and throughput could be calculated. The results revealed several new patterns and distributions in network traffic when seen from the client's perspective.

In this paper, ELFA is used to calculate latency and throughput of different communicating systems. ELFA can also be used as part of the system testing to verify quality requirements, e.g. "at least x% of the time the system should send at least y messages per second". Additionally, once outliers have been properly defined, ELFA could also be used for automatic on-line reporting of anomalies.

Acknowledgments

This work was sponsored by The Knowledge Foundation industrial PhD school ITS ESS-H and Infoflex Connect AB. Wasif Afzal is partly supported by The Knowledge Foundation through grant 20160139 (TestMine) and 20130085 (TOCSYC).

References

[1] D. Mosberger and T. Jin, "httperf - A Tool for Measuring Web Server Performance," *SIGMETRICS Performance Evaluation Review*, vol. 26, no. 3, pp. 31–37, 1998.