

Determining the Sources of Delay in a Distributed Learning Environment

Alan Ruddle, Collin Allison, Peter Lindsay, David Mckechan

► **To cite this version:**

Alan Ruddle, Collin Allison, Peter Lindsay, David Mckechan. Determining the Sources of Delay in a Distributed Learning Environment. 1st LEGE-WG International Workshop on Educational Models for GRID Based Services, Lausanne, Switzerland, 2002, Lausanne, Switzerland. pp.12, 2002. <hal-00190451>

HAL Id: hal-00190451

<https://telearn.archives-ouvertes.fr/hal-00190451>

Submitted on 23 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Determining the Sources of Delay in a Distributed Learning Environment*

A. Ruddle, C. Allison, P. Lindsay, D. McKechnan
School of Computer Science
University of St Andrews
St Andrews, KY16 9SS
{alanr, ca, plindsay, dm}@dcs.st-andrews.ac.uk

Abstract

In the context of LeGE-WG we expect to see the Open Grid Services Architecture (OGSA) being used as a guiding framework for the future deployment of Distributed Learning Environment (DLEs). OGSA is totally service-oriented and it has been known for sometime that Quality of Service (QoS) is key to the success of DLEs. Delay in particular, as experienced by the end user, is one of the key QoS parameters for a DLE. This paper describes techniques for identifying sources of such delay, and a model for its analysis. Four major sources of delay are distinguished: Server, Client, Network and Protocol. The analysis techniques are illustrated in a case study of the traffic associated with a set of operational DLEs serving six Universities over a period of several months. This paper contributes towards the use of OGSA for e-learning by providing a detailed understanding of the QoS requirements for an OGSA compliant e-learning service.

Key Words: Quality of Service, QoS, Distributed Learning Environments, DLEs, e-Learning, LeGE-WG, Open Grid Services Architecture, TAGS, web-based learning, TCP, http, Network Delay.

1. INTRODUCTION

It is important from the perspective of good educational practice that online learning should be *interactive, responsive* and *engaging*. A number of studies [1, 2] have shown that the delay between a users action, such as clicking on a user interface element or entering text and receiving a response from the system, is a critical QoS issue. Slow responses can quickly dissuade teachers and learners alike from investing their time in the use of DLE services. In this paper a methodology for assessing the factors that contribute to the delay experienced by users is presented and the implications that flow from the results obtained by its application are discussed. TAGS [3] is a framework for the research, development and deployment of DLEs which differs from conventional online learning packages in that it is QoS aware [4]. Recently reports of some users experiencing significant delays while using TAGS prompted us to return to work in analysing the source of delay in network service previously published in [5].

2. THE DELAY COMPONENT MODEL

* This is a revised and shortened version of a paper which originally appeared in the proceedings of PGNet 2001.

The notion of a CURL, the Closure over the full resolution of a URL, was introduced in [5]. It is precisely the period between a user clicking on a URL or HTML action button and the results being rendered on the user's display. The original model identified three major components of delay: the server, the network and the client. The results from that analysis showed the network delay to be relatively insignificant and that a surprisingly large amount of delay depended on the platform/browser combination. Accordingly, that exercise did not carry out a detailed analysis of network behaviour, but that is now addressed in the revised model described in this paper, which places a stronger focus on the information that can be gleaned from the transport level packet headers.

3. THE TAGS CASE STUDY

TAGS educational resources have certain characteristics that are of interest from a QoS perspective. They tend to be highly interactive and generate a two-way flow of information which stands in contrast to the simple client pull model of static web pages. The dynamic generation of these pages on the server means that caching is of limited use in reducing latency. Much of the time YAGS users have high bandwidth access to the Internet and high bandwidth pathways to the TAGS server. For a smaller proportion of their time, when the system is being accessed from home or whilst away from the host institution, smaller bandwidths may be available. There is therefore a wide variation in the network conditions that users experience due to the mobility of the user base.

3.1 Traffic Measurements and Analysis

Two techniques were used to collect data on TAGS usage. Firstly the http server logs were analysed to determine the distribution of transfer sizes. This information was used to select a range of file sizes associated with data from the second source, TCPDump packet level traces. These traces were post-processed to determine the levels of congestion and the distributions of network RTTs. The following types of transfers were selected to be analysed in detail:

- A 27KB static text/html file which corresponds to a TAGS help file.
- A 10 KB form of the sort used to enter the marks for a small tutorial group.
- A dynamically generated 7 KB table of the sort used to view information about a tutorial group.
- A 27KB table dynamically generated from a number of small files on the server. This type of table is used to view information about all the students in a module.
- A larger module overview table, dynamically generated from a database.

The next four sections describe techniques used to identify the client, network, protocol and server limitations.

4. CLIENT LIMITATION

Absolute client limitation can be decomposed into two elements: one at the start of the connection (the resolve time); and one at the end of the connection (the presentation time). Consider the time taken to resolve the URL and form the HTTP GET request at the start of the connection. To determine upper and lower bounds on the clients contribution to this delay four measurement points are required; the time the user clicks on the page (C), the transmission of the clients TCP SYN segment (S), the receipt of the server's SYN/ACK segment and the transmission of the HTTP GET request (G). A lower bound on the resolve time is given by:

$$\text{Resolve Time} = (S-C) + (G-A)$$

An upper bound on the resolve time is given by G-C (although this measurement includes network delay the request may have been being formed during this time). Next, consider the presentation time. The upper bound is given by the time to display the web page from the arrival of the first data. The lower is the time required to display a page once all the data has been received.

A small harness web application that consists of a server script, which dynamically generates a multi-framed set of web pages and records measurements in a server side data repository was developed. Links to the page being measured are included in a presentation frame. When a user clicks on one of these links the system time is read using the JavaScript onClick() method and temporarily stored in a data frame. When the new page finishes loading the onLoad() Javascript method is used to record the time in the data frame. These pairs of readings are periodically downloaded to the server pending analysis. Taken together each pair defines the duration of a CURL. To obtain measurements of the checkpoints intermediary to the CURL's end points passive packet level monitoring of the client was utilised.

<i>F</i> <i>i</i> <i>l</i> <i>e</i>	<i>L</i> <i>i</i> <i>n</i> <i>u</i> <i>x</i>	<i>W</i> <i>i</i> <i>n</i> <i>d</i> <i>o</i> <i>w</i> <i>s</i>	<i>W</i> <i>i</i> <i>n</i> <i>d</i> <i>o</i> <i>w</i> <i>s</i>	<i>W</i> <i>i</i> <i>n</i> <i>d</i> <i>o</i> <i>w</i> <i>s</i>
<i>T</i> <i>y</i> <i>p</i> <i>e</i>	<i>N</i> <i>a</i> <i>v</i> <i>i</i> <i>g</i> <i>a</i> <i>t</i> <i>o</i> <i>r</i> 4 . 7	2 0 0 0 <i>N</i> <i>a</i> <i>v</i> <i>i</i> <i>g</i> <i>a</i> <i>t</i> <i>o</i> <i>r</i> 4 . 7	2 0 0 0 <i>N</i> <i>a</i> <i>v</i> <i>i</i> <i>g</i> <i>a</i> <i>t</i> <i>o</i> <i>r</i> 6	2 0 0 0 <i>I</i> <i>E</i> 5
<i>S</i> <i>t</i> <i>a</i> <i>t</i> <i>i</i> <i>c</i>	0 . 0 9 5	0 . 0 3 7	0 . 1 9 1	0 . 0 5 2
<i>F</i> <i>i</i> <i>l</i> <i>e</i>				
<i>S</i> <i>m</i> <i>a</i> <i>l</i> <i>l</i>	0 . 0 9 4	0 . 3 6 8	0 . 3 1 7	0 . 0 9 8
<i>F</i> <i>o</i> <i>r</i> <i>m</i>				
<i>S</i> <i>m</i> <i>a</i> <i>l</i> <i>l</i>	0 . 0 8 1	0 . 1 0 1	0 . 2 5 1	0 . 1 3 1

T a b l e				
L a r g e	0 . 1 0 9	0 . 4 5 9	0 . 4 1 9	0 . 1 9 2
T a b l e				

TABLE 1: Absolute Client Limitation in Seconds

The results presented in Table 1 show that the absolute client limitation is a small proportion of the delay experienced by users when a page is dynamically generated.

5. NETWORK LIMITATION

A connection is network limited when the fair rate at which data is delivered by the network is not fast enough to keep up with the rate at which data is produced by the server or consumed by the client. The absolute size of the delay that can be attributed to the network may be estimated if two variables are known:

- the network Round Trip Time (**RTT**)
- the probability **P** of a dropped packet

It is then possible to calculate the limitation imposed by the network from the traffic statistics gathered using passive monitoring by applying the model derived and validated in [6-8]. Throughput, as a function of loss probability is given by the TCP fair equation (see Fig. 1). **P** is the probability of loss, **MSS** is the maximum segment size and **C** is a constant term.

$$T = \frac{C * MSS}{RTT * \sqrt{p}}$$

FIGURE 1: The TCP Fair Equation

In this case study network measurement was used to estimate a value for **C** of 1.079. Using 95% confidence intervals the upper interval is at 1.086 and the lower at 1.073. The delay in round trip times **D** is given by dividing **S**, the size of the transfer in segments, by **W** the average window size. **W** is obtained by removing the RTT term from the throughput equation: **D = S/W**.

Table 2 gives the expected network component of delay, which covers the range of transfer sizes shown in Table 1 and for the levels of congestion found in our analysis of TAGS traffic. The results shown are for the latency attributable to the network in RTTs. They can be scaled for a particular RTT by multiplying the figure in the table by the RTT. An MSS value of 1460 bytes is assumed.

If it is assumed that delays in the region of 20 ms are perceptible to the user then it can be concluded that for a path with a RTT of 10ms the network delay becomes important for small files at levels of congestion around 10%, for medium sized files at levels of congestion between 0.1 and 1% and for large files whenever congestion is present.

C	1	2	5	1
o	0	6	1	.
n				5
g	K	K	K	
e	b	b	b	M

stion %	ytes	ytes	ytes	bytes
0.01	1	1	1	10
0.10	1	1	1	31
1.00	1	2	4	95
10.00	2	6	11	302
20.00	3	8	15	426

TABLE 2: Expected Network Limitation in RTTs

If it is further assumed that delays above ten seconds will have a strong negative impact upon the browsing experience then the network component is only likely to be significant, when the RTT is in the order of 200ms and levels of congestion are above 1% for medium to large files. This scenario would affect only a small proportion of TAGs traffic. Although the network delay will be large enough to be perceptible for a significant proportion of TAGS traffic it is not large enough to have a strong negative impact upon the users experience.

6. PROTOCOL LIMITATION

In practice a number of factors mean that TCP does not reach the throughput implied by the fair equation. In the absence of server and client limitation the difference can be attributed to protocol limitation. These factors are:

Upon the detection of loss either Slow Start or Fast Retransmit may occur, depending upon whether duplicate acknowledgements or a Retransmit Timeout (RTO) signalled the loss. Close to 50% of the occasions when a packet is dropped an RTO is required to recover[9].

The offered window size may limit the throughput achieved when congestion is low and a large amount of data needs to be transferred.

TCP was designed for bulk transfers, where the steady state behaviour of TCP dominates and the transient effects of Start up are marginal. For short transfers TCP's start up transients will significantly reduce the throughput achieved.

The effect of Window Limitation and the Slow Start algorithms was evaluated. Window size is used in preference to throughput, as it is independent of the RTT, consequently the comparisons made hold across a range of round trip times. The effect is evaluated by comparing the Average Window size for a connection, with what the average would have been if the Congestion Avoidance algorithm controlled all data flow. This second average we term the optimum window size.

This optimum window size is the minimum of the file size and the Fair Window, because in the case where the file size is smaller than the Fair Window, file size limits the number of packets that could be transmitted in the first window. The fair window size can be obtained from the TCP Fair equation by removing the RTT term from the right hand side.

Average window size is simply the mean size of the utilised window size, which may be obtained by dividing the size of the transfer by the number of rounds that it takes to complete. The ratio between the Average and the Optimum Window size is the metric used to determine the influence of Slow Start and Window limitation:

$$\text{Ratio} = \text{Average} / \min(\text{File Size}, \text{Fair Window})$$

This ratio has been generated for loss probabilities ranging from 0.01% to 20% and for transfer sizes ranging from one segment to 100,000 and plotted in Figure 2. The X-axis is the log of the transfer size in packets. The Y-axis is the log of the Fair Window size. The Z-axis is the ratio described above. Figure 3 shows the results for a run with the offered window size set to 65700 bytes or 45 1460 byte segments.

Reading the graphs from left to right it can be observed that for connections that can complete within a single segment a ratio of average to fair of one is achieved. However, as the transfer size increases the ratio declines exponentially. This reflects the static initialisation of Slow Start resulting in the connection taking multiple RTTs to complete, if the initial window size had been initialised to the fair window size data transfer would have been completed in one window. Once the file size becomes smaller than the fair window size the ratio begins to climb towards one. This reflects the fact that for larger connections the congestion window is able to open sufficiently for bandwidth utilisation to tend towards the fair value. There is however a peak, where the average window size of the connection exceeds the fair window size. This is caused by Slow Start's exponential increase allowing the congestion window to open beyond the fair window size.

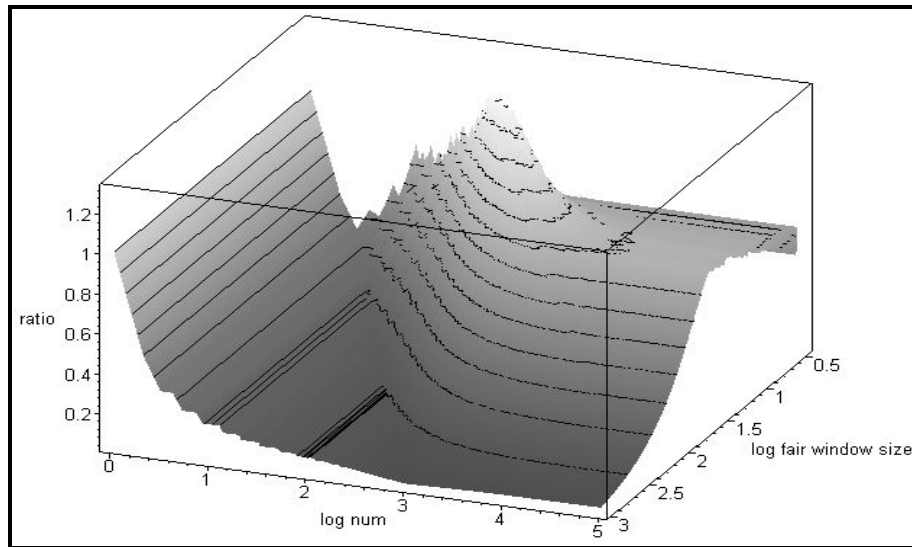


FIGURE 2: Ratio of Average to Fair Window Sizes

(Congestion range 0.01% to 20%, File Size range 1 to 100,000 segments, 45 Segment offered window size)

For connections that last beyond the initial Slow Start stage, window size again reduces until the oscillations centre on the fair window size. Thus for long connections where the Slow Start transients are amortised over a long Congestion Avoidance stage the ratio tends to one. Reading the graphs from top to bottom: where congestion is high and the Fair Window size is small, the trough where the bandwidth is under utilised is narrow and shallow; as congestion decreases and the Fair Window size increases the trough deepens and becomes more prolonged.

7. SERVER LIMITATION

Absolute server limitation is the time taken for the server to process an HTTP GET request and start delivering data. It may be determined by reading packet level traces captured at the server and is the difference between the arrival time of the get request and the departure time of the of the first data packet. For the TAGS traffic under study this amounts to a few milliseconds. The rate at which the web server receives data for transmission may be the determining factor in deciding the length of the data transfer phase for a web application. This rate has been determined experimentally for the representative selection of file types.

Whether a connection is server limited can be verified by determining the relationship between the window size utilised during a connection and comparing it with the congestion and advertised window. If the utilised window is smaller than the minimum of the congestion window and the advertised window then at that point in the connections lifetime it can be said to be server limited.

The advertised window can be obtained directly from TCP packet headers. The evolution of TCP's congestion window can be calculated as TCP's Slow Start and Congestion Avoidance algorithms are well known and packet losses can be detected by the retransmission of dropped packets. Examples of connections with and without server limitation are shown in Figure 3. Once the existence of server limitation is established it can be quantified. This was achieved by taking readings of the times the first and last data packets were transmitted for each connection. The size of each transfer is known so the rate can be calculated as quantity of data over the data transfer time

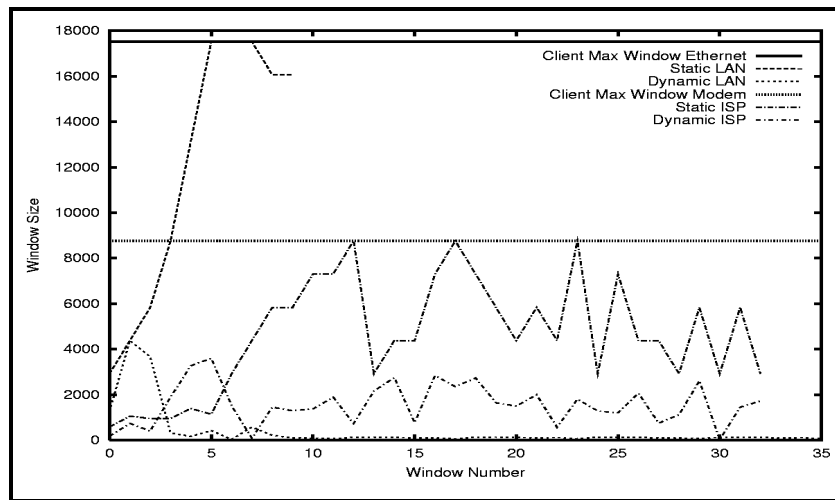


FIGURE 3: Utilised Window Sizes

Measurements for TAGS traffic are presented in Table 3. The first column shows the average absolute limitation for each transfer type. The second and third columns show the transfer delay and the data transfer rate. The fourth column shows whether the connections were server limited throughout their lifetimes.

It is interesting to note that despite the low RTTs and absence of congestion the transfer of static text files did not show server limitation. This is in contrast to the transfers that draw their data from a CGI process.

<i>F</i>				
----------	--	--	--	--

Determining the Sources of Delay in a Distributed Learning Environment

e T y p e				
S t a t i c F i l e				
S m a l l T a b l e				
S m a l l F o r m				
L a r g e T a b l e				
L a r g e T a b				

1				
e				
2				

TABLE 3: Server Limitation

It would be expected that the server limitation rate would remain constant over a range of RTTs and congestion regimes. For TCP as congestion increases and the RTT goes up the rate at which it can carry data decreases. Having experimentally determined a server limitation rate for TAGs pages it is useful to test the hypothesis that server rate would remain unchanged across a range of network conditions. A number of transfers were undertaken using a 56 Kb/s modem connected to a popular free ISP. The results are shown in table 4. There is little change in the server rates for the dynamic files confirming our hypothesis that the connections are predominantly server limited. There is a significant decrease in the rate for the static file transfer which can be explained by an increase in the significance of the network limitation.

Determining the Sources of Delay in a Distributed Learning Environment

File Type	Mean		Rate (K/bits)
Statistical File	3 . 1 7 2		6 7 . 8
Small Table	2 . 9 3 3		2 0 . 7
Small File for	2 . 8 5 2		2 9 . 3

m			
L	1		3
a	2		2
r	.		.
g	5		5
e	5		
	7		
T			
a			
b			
l			
e			
1			
L	4		5
a	1		.
r	.		2
g	6		
e	3		
	9		
T			
a			
b			
l			
e			
2			

TABLE 4: 56 Kb/s Modem and ISP Data Transfer

8. CONCLUSION

We have described the characteristics of DLEs constructed using the TAGS framework and noted that: i) they are highly interactive distributed applications, and ii) that delay, as experienced by the user, is a key QoS parameter. A structured timing model of the delay has been presented, which facilitates an analysis of the proportion of delay that can be attributed to the network, transport protocol, client and server.

Our results indicate that for dynamically generated files server limitation is the most important factor. In the case of the larger dynamically generated tables, the server limitation is in the order of 10 seconds. The contribution of client limitation is in the order of 100s of milliseconds. For network conditions with congestion less than 10% and RTTs of less than 100ms the combined value for protocol and network limitation will be less than ten round trip times or one second. However, for statically generated pages a different picture emerges. For paths with a significant RTT, and relatively low levels of loss, protocol limitation adds significantly to the latency experienced by the user. This suggests the need to address the mismatch between Web traffic and TCP's congestion

control algorithms. For certain file types, such as those containing large tables, client side limitation is also significant; in our example in the region of one third of a second. So, even if measures are taken to reduce the extent of server limitation for a significant proportion of connections network and protocol limitation remain significant.

REFERENCES

- [1] Bouch, A., N. Bhatti, and A.J. Kuchinsky. (2000) *Quality is in the eye of the beholder: Meeting users' requirements for Internet Quality of Service*. in *CHI'2000*. .
- [2] Ramsay, J., A. Barbesi, and J. Preece, (1998) *A psychological investigation of long retrieval times on the World Wide Web*. *Interacting with Computers*. **10**(10): p. 77-86.
- [3] Allison, C., et al. (2000) *The TAGS Framework for Web-Based Learning Environments*. in *Web Based Learning Environments*. Portugal: University of Porto, FEUP Editions.
- [4] Allison, C., et al., (2001) *An Holistic View of Quality of Service*. *Interactive Learning Environments*, **9**(1).
- [5] Allison, C., M. Bramley, and J. Serrano. (1998) *The World Wide Wait: Where Does the Time Go?* in *Engineering Systems and Software for the Next Decade*. Vasteras, Sweden: IEEE Press.
- [6] Mahdavi, J. and S. Floyd, (1997) *TCP-Friendly unicast rate-based flow control*, N.s.t.e.e.-i.m. list, Editor. .
- [7] Mathis, M., J. Semke, and J. Mahdavi., (1997) *The macroscopic behaviour of the TCP Congestion Avoidance algorithm*. *Computer Communication Review*, (27).
- [8] Padhye, J., et al. (1998) *Modeling TCP throughput: A simple model and its emperical validation*. in *ACM SIGCOMM 98*.
- [9] Balakrishnan, H., et al. (1998) *TCP behaviour of a busy Internet server: Analysis and improvements*. in *IEEE Infocomm 98*.