Does the restart method work? Preliminary results on Efficiency improvements for interactions of web-agents

Michael Schroeder, Luke Boro
Dept. of Computing, City University, London, UK, msch@soi.city.ac.uk

ABSTRACT

With the wide adoption of web-agents, improving the efficiency of their interactions is of great importance. The restart method aims to reduce overall completion times of such interactions by using the distribution of completion times to determine a best timeout after which it is more promising to re-issue a request than to wait for completion.

In this paper, we review existing work on the restart method and self-similarity of internet traffic. We extend existing work by analysing the restart method for the four interaction phases connection, request, processing, reply. We obtain results for web-agents retrieving statically and dynamically generated webpages and discuss implications for agents based on middleware such as RPC, Corba, or mobile agents.

Regarding the connection phase, previous work by Huberman et al. found a lognormal distribution in UDP connections. In contrast to the simpler UDP, we focus on the more complex and more widely used TCP/IP connections, for which we obtain more complicated distributions, which lead nonetheless to improvements in 65% of our experiments. Furthermore, in contrast to previous work we relate document size to reply times.

1. INTRODUCTION

Interactions of web-based agents vary in time and are greatly influenced by hardware on both sides, the networks used, and the operating system's software used. As interaction completion times vary in time, they can be described as random variables. Knowledge about this random process is very useful to improve performance. Knowing the distribution an agent can come up with a maximum time it is willing to wait for a reply to its request. This choice has to be taken carefully, as the request will have to be re-issued to the same or another agent hopefully with a better response time. This performance improvement is known as restart method [8, 9, 12, 11, 3]. It is often encountered in practice when browsing the web: a web page takes ages to be retrieved. After some time users impatiently reload the page and it is promptly downloaded.

The principle underlying this example is general in nature. Given a client and a server, the client issues a request to the server. If the request is not served after a certain amount of time, the client reissues the request, possibly to another server. Using this strategy, the client expects an overall improvement, if the timeout period is set correctly.

How can this be achieved? First of all data needs to be collected to find out the distribution of completed requests. For certain distributions [3] a cutoff can be chosen, which reduces the mean and variance of the new distribution in comparison to the one, which does not employ the restart method. An example of a distribution, where the restart method works, is the lognormal distribution [11, 12]. It is asymmetrical and after a peak the distribution slowly tails off. Therefore performance improvements can be expected if requests are timed out after the peak.

Huberman et al. [8, 9, 12, 11] investigated the restart method for UDP-connections and retrieval of index web pages, where they found lognormal distributions in the data. While Huberman et al. are mainly interested in social processes influencing network traffic, our work aims to put the restart method to practice. This requires in particular a detailed analysis of the different phases of agent interactions, which is not considered by Huberman et al.

Basically, an interaction between agents can be broken down into four phases: (1) the client connects to the server, (2) the client issues the request, (3) the server processes the request, (4) the server sends the reply to the client. These four phases can have very different characteristics. Let us consider four scenarios:

(1) Static webpages. An information agent retrieves plain, static webpages, i.e. they were not generated dynamically by CGI. Phase 1 involves connection with TCP. Will this have a lognormal distribution as the UDP-connections mentioned above, especially bearing in mind that UDP is much simpler than TCP? In phase 2, the client submits the query. As the agent just requests a plain webpage only the filename needs to be transmitted and we can assume that this phase takes nearly constant time with very little variance. The same holds for answering the query, as the server simply locates the file to be transmitted in the next phase. This last phase, however, will have an interesting distribution, which we expect to be dominated by the network's capabilities and the size of the file.

(2) Dynamically generated HTML pages. Let us consider the same scenario involving dynamically generated web pages. Phase 1 and 4 are the same as in (1), but phase 2 can vary now from sending a few short parameters, which could be assumed constant, to uploading a large file, which will be dominated by the document size. Also, phase 3 becomes now an interesting random variable, as the dynamic generation of the webpage depends highly on the application run, ranging from a quick lookup in a database to long computation of results. In general, the distribution of dynamically generated pages cannot be estimated.

(3) Remote procedure calls and mobile agents. This scenario

is similar to (2). In particular, phase 2 and 4, involve the transmission of objects/agents and can be described by a distribution, which could significantly differ from the one governing phase 3.

With these scenarios defined, we set out to develop a benchmarking tool for performance improvements of web-agents. The tool has to address in particular a detailed analysis of TCP connection to websites, benchmarking of CGI servers, and distribution of internet traffic in general.

The paper is organized as follows. We briefly review TCP/IP and its methods to deal with congestion. We describe the restart method and review the log-normal distribution and self-similarity in internet traffic. Next, we review and criticize Huberman's work on the restart method. Then we set out for a detailed analysis of interactions of web-agents. We give an algorithm to estimate the best restart time and then investigate the distributions and restart method applied to the different phases of interactions of web-agents.

2. BACKGROUND

2.1 TCP/IP and UDP

The TCP/IP internet protocol suite is a collection of protocols that can be used to communicate between many different networks. These protocols are open protocols which means that they are not proprietary and not for profit. They are freely available and they can run on different hardware architectures and different operating systems [10]. This spirit of openness is one of the foundations of the internet. TCP/IP protocols are designed using layering principles in such a way that the protocols are independent of the various transmission technologies that may be used in different networks. There are four layers in the TCP/IP suite: (1) The lowest layer is called the link layer which corresponds to hardware such as a PC's network interface card. (2) Next comes the network layer which is where the internet protocol (IP) operates. This protocol is used to define IP datagrams that are sent through a network. IP provides connectionless, best effort packet delivery service. Connectionless means that the routes between hosts are not fixed i.e. each packet makes its own route through the network and they are often different. What characterize the best effort packet delivery service is that it does not guaranty delivery. (3) The TCP/IP protocol suite has two protocols in the next layer called the transport layer: the User Datagrams Protocol (UDP) for an unreliable connectionless delivery service and the Transmission Control Protocol (TCP) for reliable stream delivery. (4) The last layer is known as the application layer where services such as FTP and HTTP resides.

2.2 Congestion

The Internet is a worldwide collection of heterogeneous networks connected with routers and gateways that use TCP/IP for interconnections between them [10]. This complex structure creates one large virtual network. A router or gateway is a computer that connects two or more networks and forwards packets of data between them [2]. When traffic through a router is so intense that the router cannot process nor store all datagrams that arrive at it, a condition known as Internet congestion occurs.

According to Comer [5] datagrams can arrive at a router faster than the router can process them for two reasons: (1) they can be generated by a single very fast computer; (2) or they can be generated by a number of computers, which send their datagrams through the same router.

If the datagrams start arriving to the router faster that it can process them, the router stores the datagrams in its memory for processing at a later time. If this condition continues, the router will eventually exhaust the memory allocated for storing the datagrams and it will start discarding them. A router can warn the computer that originally sent the datagrams of this condition and request it to lower its transmission rate. This is achieved by sending what is called an ICMP source quench message. One such message is usually sent for each discarded datagram.

2.3 Improvements to Internet network performance

Several mechanisms in TCP have been developed to improve the problem of internet congestion:

(1) TCP provides a sliding window scheme for controlling end-to-end flow. The window indicates the size of the buffer that the receiver has presently available and this size is advertised in every acknowledgement [4].

(2) The Maximum Segment Size Option in TCP can affect efficiency in two ways: large segments can increase throughput, but too large segments can cause IP fragmentation which can result in the loss of some fragments. This would again decrease efficiency [16].

(3) Two further techniques can be implemented in TCP to help with congestion: Slow Start and Multiplicative Decrease. Both techniques use the congestion window limit to control traffic. Slow Start uses the congestion window limit to decrease traffic following the loss of a sent segment, while Multiplicative Decrease uses it to increase the traffic after the network has recovered from the congestion problem [5].

(4) Congestion can often be caused by retransmission. Therefore, the requirements for internet hosts published by the Network Working Group Force stipulate that "a host TCP must implement Khan's algorithm and Jacobson's algorithm for computing the retransmission timeout." By ensuring that these algorithms are always implemented, the occurrence of congestion due to retransmission is diminished [2].

(5) Another common reason for congestion is due to a network being overloaded with single-character messages created by a keyboard, a condition known as 'small-packet problem'. The solution to this is to prevent new TCP segments being sent when new outgoing data arrives from the user, if any previously transmitted data on the connection remains unacknowledged [13].

As a conclusion, let us summarize that TCP provides a number of mechanisms to deal with congestion. In contrast, this complexity and sophistication is not present in UDP.

2.4 Improving congestion based on the restart strategy

While the above methods are integrated into TCP, Lukose, Huberman, Maurer, Hogg [8, 9, 12, 11] and Chalasani, Jha, Shehory, and Sycara [3] describe applicable methods for efficiency improvements based on the restart method.

The idea is that it is often advantageous to restart a request, which is taking a long time to be served, rather than to keep waiting for the original request to be fulfilled. But how long should one wait? To answer this question one first needs to find out about the distribution of the time it takes to serve a request. As an example, [11] performed multiple connectivity tests from the US to the UK measuring UDP packet round trip-times, which follow quite closely a lognormal distribution. As an example consider Figure 1, which shows the distribution of roundtrip times of UDP packets from the UK to Japan and the right figure clearly shows the close match of the cumulative distribution of the real data to a lognormal distribution with mean 9.25 and variance of 0.5.

With the distribution known, a best cutoff time can be chosen. Figure 2 shows an algorithm to compute the best cutoff time based

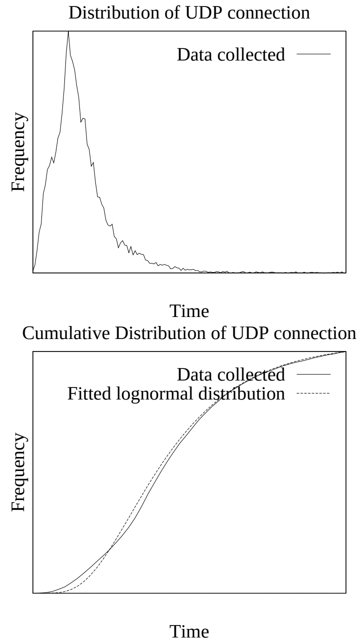

Figure 1: Distribution of round-trip times of UDP packets sent from the UK to Japan. The right shows the match of the cumulative distribution to the cumulative lognormal distribution with mean 9.25 and variance 0.5.

```
Given timings X = (x_1, ..., x_n)
for c = 1 to max(X)
    i = 0;
    Y = (0, ..., 0);
    for j = 1 to n
        y_i = y_i + min{c, x_j};
        if x_j < c then i++;
    μ_c = (1/j) Σ y_j;
return index c, where μ_c minimal
```

Figure 2: Computing the best cutoff time for a given test set.

on the existing data. For every possible restart time it derives from the existing data the distribution resulting from using the cutoff under consideration. The cutoff leading to the distribution with smallest mean is then selected as best cutoff. The complexity of this algorithm is $O(nm)$, where n is the size of the data and m is the number of possible cutoff values.

With a cutoff value chosen, the restart method can be employed and the question remains whether the theoretically best cutoff leads to improvements in practice. But before we answer this question let us review the results obtained in previous work.

2.5 Review of existing work

The motivation of Huberman et al. is not the improvement of low-level protocols but the study of social phenomena on the internet. Lukose and Huberman [11] argue e.g. that the lognormal distribution "reflects the dynamics of millions of users confronted with the dilemma of either consuming bandwidth greedily and thus creating congestion, or postponing access to remote information and paying the corresponding cost in utility". The authors are in particular interested in the question whether employing the restart method on a large scale still combats congestion [12]. And they show based on their theoretical model that an overall adoption of the restart method leads to an overall improvement [12].

However, [6] question the above approach by pointing out that although the internet is influenced by social processes, there is no evidence of these processes in the experiments Huberman et al. carried out. In particular, [6] point out that the experiments based on UDP reported in [9, 11] shows very low rates of packet loss and thus cannot serve as evidence for any congestion.

In another paper, [12] investigate the restart method when applied to web page retrieval. They show the distribution of 40,000 index-pages and argue that it fits a log-normal distribution and that the restart method is applicable. [15] confirms that bulk transfer traffic such as FTP, SMTP, and NNTP data can be fitted by a log-normal distribution, but also points out that this is an approximation as experiments show self-similarity in network traffic [7, 19, 14, 18], which means that the distributions are heavy-tailed approximating $x^{-\alpha}$ (rather than lognormal).

There is another aspect to the distribution for systems, in which client request services and servers reply. The distribution is a complex mixture of many different distributions. We can generalize the scenario: An agent requesting a service from another agent has to connect, transmit the request, the request has to be processed, and the reply sent back. These four phases can have very different distributions leading to quite different a distribution overall. Before we present our findings, we will give the relevant details on the important lognormal distribution and review the literature on self-similarity of web traffic.

3. DEFINITIONS AND STATISTICAL BACKGROUND

3.1 The lognormal distribution

Let X be a normally distributed random variable with mean μ and variance σ, for short $X \sim N(\mu,\sigma)$. The distribution function for a normally distributed variable is given as

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

and Φ denotes the cumulative distribution function, i.e $\Phi(x) = \int_{-\infty}^{x} \phi(x)dx$. The following relation holds between normally and lognormally distributed random variables: X is lognormally distributed, short $X \sim LN(\mu,\sigma)$ if $ln(X)$ is normally distributed. This implies that the cumulative distribution function of a lognormally distributed variable can be expressed in terms of Φ, namely as $\Phi(ln(x))$.

One of the reasons why both distributions are so important is due to the Central Limit Theorem, which relates any infinite number of random variables with the same distribution to the normal and lognormal distribution:

THEOREM 1. *Central limit theorem (e.g. [1]). Let X_i be random variables with the same distribution, then $\sum_{i=1}^{\infty} X_i \sim N(\mu,\sigma)$ and $\prod_{i=1}^{\infty} X_i \sim LN(\mu,\sigma)$.*

As a consequence, a finite number of random variables approximates the (log)normal distribution, when added up (multiplied). This profound result has an immediate application to networks such as the internet. Queuing network models with an independent queue lengths are often considered accurate descriptions for such systems [17].

As a simple example let us consider UDP, which we pointed out above is a simple and fast protocol with no complicated congestion

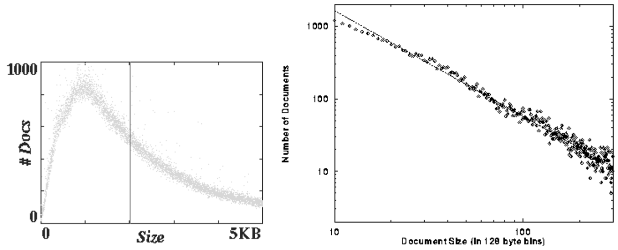

Figure 3: Left: Distribution of 2.6 million web documents [19] Right: Distribution of 500,000 web documents [7]

control integrated. According to Huberman et al. [9, 11] and as confirmed in Fig. 1 round-trip times of UDP packets follow a lognormal distribution. Does this distribution lend itself to improvements via the restart method? Chalasani, Jha, Shehory, and Sycara [3] prove a theorem establishing conditions when *not* to expect an improvement. These can be easily checked for the lognormal distribution by an example: With mean $\mu = 6.89$, variance $\sigma = 0.829$, and a cutoff point $b = 8$, we can check their condition that the cumulative distribution must be larger than the cutoff divided by cutoff and mean; in this case $\Phi((\frac{b-\mu}{\sigma}) = \Phi(\frac{8-6.89}{0.829}) = \Phi(1.34) = 0.9099 > \frac{8}{6.89+8} = 0.539$, hence according to [3], improvements are not ruled out.

To summarize, we have three reasons why lognormal is important in practice: (1) The CLT states that the multiplicative effect of any number of random vairables approximates the lognormal distribution. (2) For this reason it is a good model for queuing networks. (3) In practice, traffic such as UDP fits a lognormal distribution very well (see Fig. 1).

3.2 Self-similarity and heavy-tailed distributions

However, in practice there is evidence that network traffic is heavy-tailed [15, 19, 7, 14, 18], which means that the distribution approximates $x^{-\alpha}$ in the long-run. This implies that with infinitely small probability infinitely large values occur. A distribution is heavy-tailed if $P[X > x] \sim x^{-\alpha}$ as $x \to \infty$, where $0 < \alpha < 2$, i.e. the asymptotic shape of the distribution follows a power law. Heavy-tailed distributions are different from other distributions in that they have infinite variance if $\alpha < 2$ and also infinity mean if $\alpha < 1$. Intuitively, this means that the smaller α is, the bigger the area covered by the distributions tail becomes. Applied in practice, [19, 7] find that distributions (see Fig. 3) of web documents are heavy-tailed. Self-similarity means that infinitely large documents can occur with infinitely small probability. But as Paxson finds [15], bulk transfer traffic such as FTP, SMTP, and NNTP data, can still be best modeled by a lognormal distribution omitting the aspects of self-similarity.

4. EXPERIMENTS AND RESULTS

4.1 Benchmarking tool

Given the theoretical and practical background, we developed a benchmarking tool to clarify the restart method applied to web-agents by contributing answers to the following questions:

(1) There is a huge difference between the fast, non-reliable, and simple UDP and the slower, reliable, and more complex TCP protocol. While the simple UDP traffic fits a lognormal distribution neatly, it is an open question whether TCP will fit lognormal?

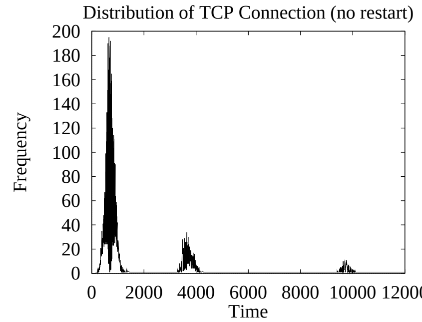

Figure 4: Distribution of 10.000 TCP connections made from UK to a site in France. On average connection 1720ms and most frequently 661ms. After applying logarithm to the data, mean and variance are 6.89 and 0.69, respectively.

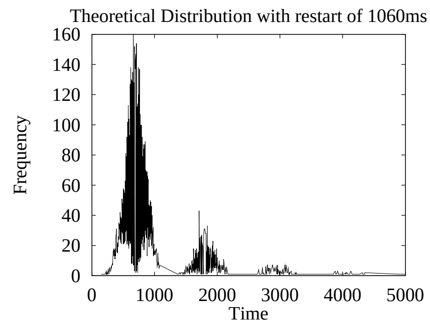

Figure 5: Based on data in Figure 4, the best theoretical restart time is after 1060 ms giving the distribution displayed with mean and variance reduced to 6.71 and 0.27

(2) While previous work treats an agent's request as a single distribution, we argue that this necessarily reflects a mix of four different distributions, as an agent's request is composed of four phases of connection, request, processing, reply. What do these distributions look like?

(3) The relation of a theoretically and practically best cutoff is not straightforward. If the restart method is employed, the agents have to compute the best cutoff. Given sample data, the algorithm in Fig. 2 computes the best cutoff to minimise the mean. Does this work in practice?

4.2 Results for TCP connections

Each complete connectivity test performed was composed of 10,000 connection trials to a website specified in that test. The total number of different websites contacted was 20. The number of complete tests conducted on each of the 20 sites ranged from 2 to 6. In total, 53 separate tests were carried out over the course of a two month period. Therefore, results relating to 53,0000 individual connection trials were collected to form the basis of the analysis presented.

The first of each of the series of tests to a particular website was a test in which the restart method was not applied. Fig. 4 shows the distribution which eventually tails off like a lognormal distribution, but which has three peaks, unlike the simpler distributions for UDP packets, which Huberman et al. [8, 9, 11] considered.

Next, we computed the best cutoff times using the first set of experiments. The algorithm in Fig. 2 basically computes the dis-

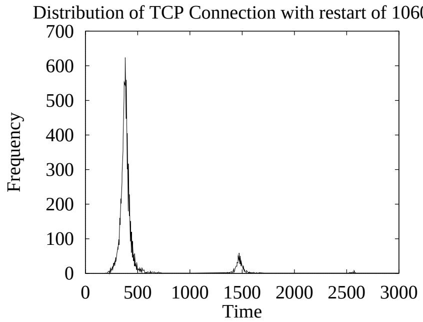

Figure 6: Distribution of 10.000 TCP connections made from UK to a site in France with a restart of 1060 ms. The average is down to 472 ms and mean and variance are 6.04 and 0.15, respectively. The restart method works better than predicted in fig. 5.

tributions for all possible cutoff times and selectes the one with minimal mean value. Fig. 5 shows the resulting distribution after applying the algorithm to the data in Fig. 4.

The second test to the website was always conducted using this recommended restart time as calculated by the algorithm in Fig. 2. Given a distribution with no restart as in Fig. 4 and the one resulting theoretically from the best cutoff as shown in Fig. 5, let us consider the practical verification as depicted in Fig. 6. For this example it shows clearly an even better behavior than theoretically predicted. After applying logarithm to all data, the mean and variance of the data without restart was 6.89 and 0.69, respectively, and predicted to go down to 6.71 and 0.27 for the best cutoff time. But in practice results were even significantly better at 6.04 and 0.15, respectively.

However, sometimes the theoretically best cutoff time did not lead to better connectivity in which case more tests were performed with different restart times in an attempt to obtain a better result than the one obtained in the very first test without restart. Below is a summary of the 53 separate tests to 20 websites that were completed:

(1) The tests conducted on 10 of the 20 websites resulted in improved connectivity from the first attempt (see Fig. 7). This means that when the second test to the website was run, using the restart method and the calculated theoretical restart time generated by the algorithm in Fig. 2, the test immediately showed improved connectivity.

(2) The tests conducted on 2 of the 20 websites eventually resulted in improved connectivity. This means that while the second test based on the theoretically best restart time did not show improved connectivity, when further tests were conducted based on restart times longer than the theoretical restart time, improved connectivity was achieved.

(3) In the case of tests conducted on another 2 of the 20 websites, the best connectivity results that could be achieved using the restart method were equal to the results of the first test, i.e. the test in which the restart method was not applied.

(4) Finally, in the case of tests conducted on 6 of the 20 websites, improved connectivity using the restart method could not be achieved given the number of tests that were run (i.e. a maximum of 6 complete tests to a particular site).

Another point worth noting is that a natural explanation for improved connectivity could be a different route taken for successive connections. Therefore we checked packet routes to all 20 sites on different occasions with the result that although routes may differ over longer time periods, successive connections always used the same route.

Site	Restart	Avg	%	Hops
insead.fr	0	1720		
	1060	472	27	11
uni-lepzig.de	0	305		
	70	91	30	18
tu-berlin.de	0	108		
	110	49	45	17
yahoo.co.uk	0	124		
	371	61	49	15
public.srce.hr	0	949		
	1000	646	68	15
www.bris.ac.uk	0	20		
	320	15	75	12
amazon.co.uk	0	236		
	1000	181	77	13
www.nagoya-u.ac.jp	0	660		
	1000	506	77	26
usyd.edu.au	0	718		
	811	610	85	23
ntu.edu.au	0	865		
	970	747	86	18
www.monash.edu.au	0	734		
	840	660	90	17
ucd.ie	0	33		
	90	31	94	15
ucla.edu	0	169		
	690	166	98	16

Figure 7: Summary of average TCP connections for 10.000 connections made from London, where restart led to improvements. Times are in ms.

4.3 Distribution of Index-page retrieval

Regarding web-agents dealing with static web pages, the above sections detailed improvements on phase 1 of the interaction, the TCP connection. As we argued in the introduction that request and processing (phase 2 and 3) are negligible for static web-pages, we can turn now to phase 4, the transmission of the document. Huberman et al. [12] considered 40.000 index pages and found a log-normal distribution. In contrast, [19, 7] found heavy-tailed distributions. But according to [15], the latter are best approximated by a log-normal distributions. However, regarding Huberman et al.'s findings there remains an open question. Document retrieval times are influenced by two factors: the network and the document size. While network queuing models give rise to log-normal distributions, the latter are heavy-tailed as depicted in Fig. 3 [19, 7]. However, Fig. 8 (left) shows both document sizes and retrieval times. A plot of one series divided by the other shows first linear behavior, which indicates that both function behaves the same. This does not hold for the tails. Nonetheless, for the restart method the initial peak in the curves are the most relevant parts and these show that document retrieval time is mostly influenced by document size.

4.4 Benchmarking a CGI-server

Having considered all four phases for static web-pages let us turn to dynamically generated pages. While phase 1 and 4 have the same distribution reported for the static case, phase 2 and 3, request and processing, can be significantly different. In fact, it

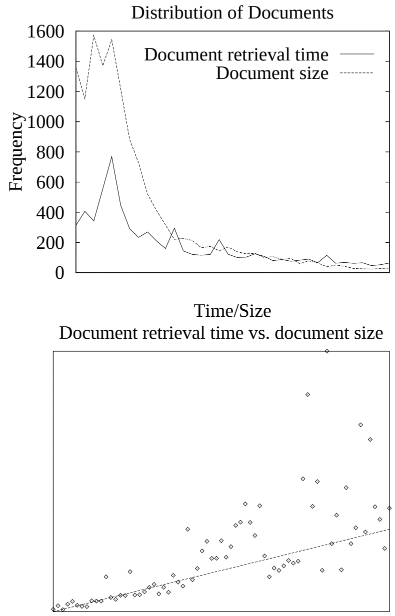

Figure 8: Left: Distribution of documents size and retrieval times for index-pages of ca. 15000 websites. Right: Retrieval time against document size is nearly linear and indicates the similar behaviour of the two curves on the left.

is not possible to make general statements about these as they are highly application dependent. A further problem concerns obtaining the data. So far we considered all data from the client's perspective, which causes problems when estimating the server's processing time. However, we can get this data indirectly. For static pages, we know the distribution of phases 2, 3, and 4. With 2 and 3 being negligible this amounts to knowing timings for phase 4. For dynamically generated pages, we can neglect phase 2 and measure phases 3 and 4. With 4 being known from the static case, we can infer timings for phase 3. Consider e.g. Fig. 9, which shows retrieval times for static and dynamic pages for three sites and which allows to infer processing time from the difference between static and dynamic pages (93, 33, and 130%, respectively).

The same method can be applied to the third scenario: remote procedure calls and mobile agents. In this scenario clients could infer processing time of servers, if servers would provide a standard operation, which requires no processing time and allows the client to collect data on result transmission. Alternatively, the server could return processing time directly as part of their replies. This would enable clients to construct an exact distribution of the server's processing time, which would enable the client to tune its restart method more accurately.

5. CONCLUSIONS

In this paper, we set out to answer three main questions: (1) With results applying the restart method to UDP connections around, are they transferable to TCP-connections? (2) If the restart method were to be integrated in protocols, it has to deal with four very different phases of interactions, namely connection, request, processing, and reply. What do time distributions for these phases for different types of agents look like? (3) Does the restart method work in practice? Does a theoretically predicted best cutoff value

Site	Byte/sec non-CGI	Byte/sec CGI	Diff	%
www.bris.ac.uk	11.5	5.6	5.9	93
www.jayde.com	9.8	7.4	2.4	33
www.princeton.edu	2.2	0.9	1.3	130

Figure 9: Benchmarking the processing of a CGI server by measuring time for page retrieval of CGI and non-CGI pages from the same site. The diff-column shows bytes/second needed for processing, i.e. serving the request excl. transmission.

lead to better results in practice?

Regarding (1), we compared UDP and TCP and reviewed the literature on congestion control and pinpointed the main difference: UDP is simple, unreliable , and fast; TCP is more complex, reliable, slower, and has a number of congestion control features built-in. UDP and TCP connection times behave differently. While the former follows the log-normal curve as put forward in [11] and verified in Fig. 1, TCP connections seem to consist of three - possibly lognormal - distributions peaking at different times.

Regarding (2), we broke down agent interactions into four phases and considered them for different scenarios: agents dealing with statically and dynamically generated webpages; the latter being similar to agents using remote procedure calls and mobile agents. For the statically generated pages, we considered the connection phase in details and argued that phase 2 and 3 are nearly constant. For the last phase of document retrieval we carried out an experiment with ca. 15,000 websites and we showed the relation of document retrieval time to document size behaving initially similarly. For dynamically generated webpages, we pointed out that distributions of times for phases 2,3 and 4 are highly application dependent, but showed a method how to indirectly obtain timings.

Regarding (3), we analyzed TCP connections and showed that the theoretical best cutoff as computed by the algorithm in Fig. 2 led to improvements in 65% of our experiments. In this context, it is worth noting the number of hops, which are a natural measure for the distance the traffic has to travel, are not relevant to improvements.

With these findings, we extend previous work such as [3], which dealt with theoretical aspects of the restart method and [8, 9, 12, 11], which focuses on the impact of collective behavior on internet traffic and which considers UDP connections and index-page retrieval. We reviewed this work and pointed to a critique [6], which argues that the UDP experiments carried out by the authors do not show any sign congestion. Different from this work we focused on the more widely used TCP-connection and compared theoretically predicted best cutoff to results, when put into practice. Furthermore, we broke down the interaction into phases, which can have very different distributions.

Future work, should address how to estimate the best cutoff time in theory and in practice and how to dynamically update the best cutoff time during runtime. The estimation of the best cutoff time should be based on different groups and profiles of destinations. To extend the restart method to remote procedure calls clients should get timings on the server's processing, which would enable them to build a server profile, which they can use to decide for the best timeout value.

Acknowledgement. We'd like to thank Julie McCann for valuable discussions and feedback, Jon Crowcroft for discussion and Ricardo Lopez for the document retrieval data.

6. REFERENCES

[1] J. Aitchison and J.A.C. Brown. *The Lognormal Distribution with special reference to its uses in economics*. Cambrdige University Press, 1969.

[2] R. Barden. Requirements for internet hosts – communication layers. Technical report, Internet Engineering Task Force, RFC: 1122, 1989.

[3] Prasad Chalasani, Somesh Jha, Onn Shehory, and Katia Sycara. Query restart strategies for web agents. In *Proceedings of Agents98*. AAAI Press, 1998.

[4] D. Clark. Window and acknowledgement strategy in tcp. Technical report, Computer Systems and Communication Group, RCF: 813, 1982.

[5] D. Comer. *InternetWorking With TCP/IP Volume I: Principles, Protocols, and Architecture*. Prentice Hall, 1995.

[6] J. Crowcroft, M. Luby, and V. Paxson. A response to lukose and huberman's article: Social dilemmas and internet congestion, 1998.

[7] Carlos R. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of www client-based traces". Technical Report TR-95-010, CS department of Boston University, 1995.

[8] B. Huberman, R. Lukose, and T. Hogg. An economic approach to hard computational problems. *Science*, 275:51–54, 1997.

[9] Bernardo A. Huberman and Rajan M. Lukose. Social dilemmas and internet congestion. *Science*, 277:535–537, July 25 1997.

[10] C. Hunt. *TCP/IP Network Administration*. O'Reilley, 1997.

[11] R. Lukose and B. Huberman. A methodology for managing risk in electronic transaction over the internet. In *Proceedings of International Conference on Computational Economics*, 1997.

[12] S. Maurer and B. Huberman. Restart strategies and internet congestion. Technical report, Palo Alto: Xerox Palo Alto Research Center, 1999.

[13] J. Nagle. Congestion control in ip/tcp internetworks. Technical report, Network Working Group, RFC: 896, 1984.

[14] Kihong Park, Gitae Kim, and Mark Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proceedings of the 4th Internation Conference on Network Protocols (ICNP96)*, 1996.

[15] V. Paxson. Empirically-derived analytic models of wide-area tcp connections. *IEEE/ACM Transactions on Networking*, 2(4), 1994.

[16] J. Postel. Tcp maximum segment size and relevant topics. Technical report, Network Working Group, RFC: 879, 1983.

[17] K.S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Wiley, 1982.

[18] W. Willinger, V. Paxson, and M.S. Taqqu. Self-similarity and heavy tails: Structural modeling of network traffic. In *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Birkhauser, 1998.

[19] A. Woodruff, P.M. Aoki, E. Brewer, P. Gauthier, and L.A. Rowe. An investigation of documents on the world world web. In *Proceedings of the 5th International Conference on the World Wide Web*, pages 963–980, Paris, France, 1995.