A Simulation Study of the Effects of Multi-path Approaches in e-Commerce Applications

Paolo Romano, Francesco Quaglia and Bruno Ciciani
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy

Abstract

Response time is a key factor of any e-Commerce application, and a set of solutions have been proposed to provide low response time despite network congestions or failures. Being them mostly based on caching of Web objects and replication of DBMS managed data at the edges, or at intermediate points, of the Web infrastructure, they well fit the requirements of client requests only performing read access to (dynamic) application data. However, these solutions typically require any update request to be redirected to the origin DBMSs, which act as the masters within the replication scheme. Hence, update requests typically do not take advantage from data replication and related client proximity. In order to alleviate the effects of network congestions or failures, we have proposed a multi-path protocol that, depending on current network conditions, increases the likelihood for the update request to be processed along a responsive (e.g. a failure free) network path in between the client location and the origin DBMS sites. In this paper we present an extensive simulation study of the effects of such a multi-path approach on the client perceived response time. The study relies on both Brite generated network topologies and the NLANR graph. Also, well known realistic TCP models are used to capture the effects of network delays during both normal and anomalous (i.e. packet loss affected) operation mode. By the results, our multi-path approach increases the likelihood for the system to maintain an adequate level of service under a wide range of network operation modes, hence including anomalous ones, which is instead not achieved in case of the standard approach not leveraging path-diversity for handling update requests at the origin (master) sites.

Keywords: e-Commerce, Web Infrastructures, Path-Diversity, Application Delivery Networks, Performance Guarantees, Performance Study.

1 Introduction

The user's perceived response time and reliability are two of the main issues for differentiation among e-Commerce Web sites, since they directly determine the level of user's satisfaction while interacting with the e-Commerce application [6]. Hence they necessarily need to be taken into account in the process of engineering the underlying Web infrastructure in order not to incur the devastating phenomenon of excessive abandon rate from users. Specifically, as demonstrated in [20], the abandon rate from users reveals modest (i.e. under the 2%) if the response time is under the threshold value of 7 seconds. Instead, it dramatically increases, up to 70% in case of a few additional seconds of delay in the delivery of the output at the client side.

To limit such a phenomenon, which is actually detrimental to the business process supported by the e-Commerce site, a spectrum of solutions have been proposed in order to ensure application availability and timely delivery of contents to the end users [9, 10, 12, 13, 19]. A key approach for most of these solutions is the employment of both Web object caching techniques and also DBMS replication techniques, which can provide the benefits of overcoming network overloads (or failures) by increasing the proximity between clients and contents, thus allowing for enhanced response time and application availability.

However, even though some of these solutions deal with caching and replication of dynamic Web contents (e.g. [12]), they still rely on direct access to the origin (primary) DBMS in case of client requests altering the application state, such as product ordering. Therefore, increased proximity to the clients cannot address the level of service seen by the users issuing update requests. For these users, network overloads or failures can lead to an excessive penalty in the perceived response time, which might ultimately degrade the brand name of the e-Commerce Web site on the basis of the negative type of experience these users receive. Given that, as widely demonstrated by characterizations of the well know TCP-W e-Commerce benchmark [17], update requests broadly represent (at least) the 10% of client interactions, satisfaction of users issuing update requests is a relevant issue to address. Furthermore, update requests are usually submitted as the concluding step of a sequence of interactions (e.g. the final submission of a purchase order after a browsing session in an e-Shop), which is the most critical step as it might trigger the activation of, e.g., some transactional billing logic possibly spanning multiple

data centers, as in the common case of e-Commerce Web sites relying on third-parties for validation of electronic payments.

In order to cope with this issue, in a previous work [16] we have proposed a multi-path approach allowing an update request to be routed in parallel along multiple network paths (hence via different edge servers) towards the origin DBMSs. This is done in order to reduce the likelihood of experiencing network congestions or failures. At the same time, our proposal embeds lightweight mechanisms for allowing a single edge server, among the multiply involved ones, to timely process the update request and report the output to the client (this ensures application safety by guaranteing at-most once semantic for the update of application data).

In this paper we propose an extensive simulation study of the effects of such a multi-path approach, in order to assess its benefits in a wide variety of system settings. The evaluation is based on both Brite generated network topologies [7] and the NLANR graph [15], representative of connectivity among Internet autonomous systems. Also, we use the TCP model in [8] to simulate network latencies realistically, considering both the case of normal operation mode and run time anomalies associated with, e.g., packet losses. Actually, we simulate the case of Web infrastructures layered over public networks over the Internet, and also the case of Web infrastructures relying on (virtual) private interconnection between edge servers and back-end data centers hosting DBMSs. This allows the quantification of the benefits from our multi-path protocol when considering main scenarios for what concerns the organization of Web infrastructures currently offered by Application Service Providers (ASPs).

The remainder of this paper is organized as follows. In Section 2 we shortly overview the behavior of our multi-path protocol. The extensive simulation study is presented in Section 3. Assessments and conclusions are reported in Section 4.

2 Multi-path Protocol Overview

The multi-path protocol we have presented in [16] is tailored for e-Commerce applications hosted by Web infrastructures consisting of a set of edge servers and a set of autonomous back-end data centers (see Figure 1), which maintain different data sets via autonomous DBMSs. These are also referred to as Application Delivery Networks (ADNs). Note that data center autonomy allows our proposal to cope with the general case of, e.g., multiple parties involved within a same business process. Actually, the interconnection between edge servers and data centers can take place either through a (virtual) private network under the control of the ASPs owning the whole infrastructure, or through the Internet. The former case typically ensures more controlled communication latency among remote servers within the infrastructure, at least in normal network operation mode.

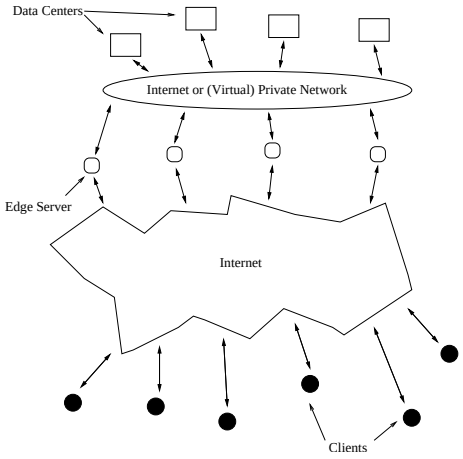

Figure 1. Target System Architecture.

The edge servers host the business logic for executing transactions against the data centers, which are responsible for guaranteeing the availability and consistency of the application data. The edge serves can perform caching of (dynamic) Web objects, and can also perform caching of application data residing at the data centers via secondary DBMSs residing at edge server locations [12]. This allows the edge servers to reply to read only requests from clients without the need for interaction with the back-end data centers. On the other hand, each time an update request is received from whichever client, the edge server needs to connect to the back-end data centers for the execution of a distributed (atomic) transaction manipulating the original copy of application data. Consistency of replicated application data maintained at secondary DBMSs is ensured via either asynchronous (lazy approach) or synchronous (eager approach) update mechanisms driven by the back-end data centers [11].

Our multi-path protocol tackles the negative effects of network congestion or failures on the handling of update requests from whichever client. Specifically, given that update requests require end-to-end interactions involving (far) back-end data centers, the current network conditions are a main factor affecting client perceived response time. To increase the likelihood for the response time to be under the threshold value leading to complete satisfaction of an interactive end-user (as mentioned, such a threshold value typically reveals on the order of 7 seconds [20]), our protocol lets the client application to perform parallel invocations of multiple edge servers along different-network paths. These servers, in their turn, connect to the back-end data centers (and set up a fresh transactional context) in parallel, again exploiting path diversity provided by the underlying network topology. When a data center receives the first connection request for a given client from whichever edge server, it waits for incoming connection requests from other edge servers for a short timeout period (on the order of few tens of milliseconds), and updates a data structure

keeping track of information related to the identities of the edge servers that requested connection within the timeout period. Afterwards, this information is returned to the edge servers requesting connection via the connection acknowledgment. Using an ordering relation on the identities of the edge servers, each edge server receiving the connection acknowledgments from the data centers is able to determine whether:

(A) It has promptly requested connection to each data center (i.e. within the timeout expiration, or has even triggered the timeout at the data center with its connection request), and

(B) In the ordering relation among server identifiers, it precedes any other edge server that promptly connected to the data centers.

If both conditions A and B are satisfied, then the edge server goes on executing the distributed transaction associated with the update request from the client. Overall, the transaction is executed by only one among the multiply contacted edge servers (hence ensuring at most-once semantic), which, depending on current network conditions, has been promptly reached by the client request and has been able to promptly connect to the back-end data centers involved in the transaction.

Actually, the timeout period at the data centers while collecting incoming connection requests from different edge servers, and before sending out any connection acknowledgment, has been introduced in order to address variance in the responsiveness in between an edge server and different data centers. Specifically, such a timeout allows including in the set of "good candidates" for transaction processing edge servers that are responsive towards all the data centers, even though there might be some other edge server more responsive in the connection to a given data center, but less responsive towards other data centers.

Further details on the mechanisms underlying the protocol can be found in [16]. Anyway, it is worth remarking that the protocol can be implemented on top of conventional technology (e.g. DBMS technology) by simply having the connection phase between the edge servers and the back-end data centers (and the related information update) supported via a proper wrapper.

3 Simulation Study

3.1 Network Model

As highlighted in a number of previous studies [4, 2], the effectiveness of any multi-path solution strongly depends on the actual disjointness among the simultaneously explored paths.

To determine how our proposal fares in different networks, we took an approach similar to the one used in [4]. In our experiments, we examined both Brite [7] generated topologies (in this case both flat and hierarchical topologies are considered, which we will refer to as, respectively, BRITE-f and BRITE-h) and the NLANR [15] graph, representative of connectivity among Internet autonomous systems at the latest available date, namely January 2000. As in [4], where a multi-path approach for video streaming applications is proposed, to assign the client, edge server and data center roles to a subset of the nodes in the topologies, we used a placement algorithm based on the connectivity degree of nodes:

- *Edge Servers:* To emulate edge server location in an ADN, we placed servers at the edge of a topology, where edges are defined as nodes with degree of two or three.

- *Data Centers:* To emulate data center location at the most connected part of a network, we place servers at the core nodes of the topology, which we define as nodes with the highest degree.

- *Clients:* To emulate client location at the furthest edge of a topology, clients were randomly chosen among those nodes having degree of one.

Obviously the ideal case would be to use a real edge server location graph from an ADN company, but such information is proprietary and not available, which is the reason why we chose to rely on this simple placement algorithm inspired by the one presented in [4] in the context of Content Delivery Networks (CDNs) based video-streaming delivery.

To generate realistic values for the network latencies perceived by the hosts participating in our protocol, under both normal and anomalous (e.g. congested) situations, the considered topologies were complemented by both mathematical models and publicly available empirical measurements of Internet latencies.

For what concerns the packet loss model across the topology links, we chose the widely adopted two-state Gilbert model parameterized by transition probabilities $\{p,q\}$ where p is the probability of going from no loss state to loss state, and q is the probability of going from loss to no loss. The Gilbert model is widely used to model bursty traffic for its simplicity and mathematical tractability. Like in several other studies, e.g., [4], we assumed for simplicity that faults over each link can be modelled as independent.

In order to accurately determine the message transfer time over TCP connections in presence of packet losses, we adopted the TCP analytical model in [8]. This model provides accurate estimations of TCP transfer times on the basis of (i) the expected number of packet losses, (ii) the number of TCP fragments to be sent (i.e. the message size in kilobytes) and (iii) the end-to-end RTT latency. The former TCP model parameter, i.e. the number of packet losses during the delivery of a message, is obtained directly by the topology simulator. The message size is randomly determined according to a heavy-tailed distribution, namely a Pareto, since a number of studies (e.g. [5]) have shown that

WWW traffic exhibits heavy-tailed message size distributions. The end-to-end RTT for each message transmission is derived by means of the RTT probability distribution shown in [1], that was empirically obtained at the light of the RTT measurements carried out between the NASA's Glenn Research Center Web Server and its clients. These RTTs are representative of end-to-end network latency between hosts communicating across the Internet. In order to correlate the length (in terms of number of hops) of a path in a topology with the corresponding end-to-end RTT value, we determine an RTT value for each link over which packets are transmitted according to the empirical end-to-end RTT distribution and scaling (dividing) such value by the average path length.

Note that in practice a strong correlation exists between a link RTT and the possible presence of packet losses over that link. In fact, the RTT values are comprehensive of router queueing delays, which are very likely to be high in case of packet losses (since these latter events are typically due precisely to the excessive growth of routers queues). In order to capture such a correlation in our simulator, in absence of packet losses we randomly pick the current link RTT from the first half of the empirical RTT distribution, namely the half collecting the lowest measured RTT values. Conversely, in presence of packet losses over a link, we randomly pick the current link RTT from the second half of the empirical RTT distribution.

3.2 Edge Server Selection Policies

In conventional Content and Application Delivery Networks, i.e. conventional Web infrastructures not leveraging path diversity, client requests are routed towards a single edge server over a single path and the selected edge server is typically the one on the shortest path to the client. This mechanism may be straightforwardly adopted in our proposal by selecting the closest edge servers to the client, or one may envision the development of more sophisticated policies taking into account specific topological information in order to achieve larger benefits from the multi-path approach.

To cope with a relatively wide spectrum of possibilities, we implemented the following three selection policies in our simulator:

- *Shortest Paths.* Simply choose the closest edge servers to the client, employing hop counts as distance metric. In the following, we will refer this selection policy to as SP.

- *Disjointness Ordered Paths.* Always select the edge server on the shortest path. Then choose the edge servers whose paths to the client have a minimum number of links in common with the shortest path. If more than one server has the same number of joint links with the shortest path, choose the one having minimum length (measured in hop counts). In the following, we will refer this selection policy to as DP.

- *Disjointness$\times$Length Ordered Paths.* Always select the edge server on the shortest path. Then choose the edge servers whose paths have the minimum values of the product between (i) the correlation with the shortest path and (ii) the additional length with respect to the shortest path. In other words, with this policy, if the path towards an edge server is highly disjoint from the shortest path, but such edge server is very far from the client, than this edge server will not be considered by the client as a good candidate for the parallel invocation scheme. In the following, we will refer this selection policy to as D$\times$LP.

3.3 Transactional Workload Model

For what concerns the transactional workload model used in the simulation, we rely on the so called "shopping workload", namely the reference transaction profile specified by TPC-W [17]. This benchmark is widely used for measuring the performance of e-Commerce systems, and relies on simulation of a breadth of activities of a business oriented transactional Web application. The shopping transaction profile is derived by TPC-W on the basis of the composition of two different customer profiles (also referred to as customer interactions) known as *browse* and *order*, respectively. The browse interaction involves browsing as well as querying activities, while the order interaction involves real update of data (e.g. loading shopping cards) at the data centers. The shopping transaction profile is based on a composition of 80% browse interactions and 20% order interactions. By the characterization of TPC-W performed in [14] we also have that the DBMS page reference pattern for such a mix of interaction is such that 96.6% of page references are in read only mode, and 3.4% of page references are in write mode.

3.4 System Settings

For what concerns the size of the data set maintained at each data center and other system settings, we again exploit the study in [14], where a global data set size of about 20 GB has been presented as a reasonable value for typical e-Commerce applications. In that study, the DBMS residing at the data center has 4 KB page size and is run on an IBM eServer xSeries 255 machine, with 4 CPUs (1.5 GHz), 8 GB of RAM storage, 12 IBM U320 disks (15000 RPM), running Windows 2000 Advanced Server. Also, the DBMS is placed on a 5-disk hardware RAID-0. For this data set size, the characterization of the shopping transaction profile presented in [14] gives rise to an average number of 35 referenced pages for each interaction. Resource consumption at the data centers while handling the interactions proper of the shopping transaction profile are explicitly simulated in our analysis on the basis of the benchmarking results in [14], obtained just for that type of hardware architecture.

We consider a whole Web infrastructure consisting of six back-end data centers and twenty edge servers. As shown in previous studies related to content delivery applications

Topology	#nodes	#edges	average path length between client and edge server			average path length between edge server and data center			average correlation ratio on the different used paths (client side)		
			SP	DP	D×LP	SP	DP	D×LP	SP	DP	D×LP
BRITE-f	5000	5000	9.1	9.3	9.1	8.9	8.9	8.9	0.53	0.46	0.52
BRITE-h	5000	5100	15.1	15.9	15.3	25.6	25.6	25.6	0.30	0.16	0.22
NLANR	6474	24467	3.0	3.1	3.0	2.3	2.3	2.3	0.42	0.35	0.41

Table 1. Summary of Topological Parameters.

[3, 4], the number of paths that is expected to maximize the benefits from a path-diversity protocol has been shown to be on the order of two. Hence we focus on the case of two edge servers contacted in parallel by the client. Fixed this setting, for the reader's convenience, we report in Table 1 a summary of the main parameters related to the different analyzed network topologies, together with information on the length and correlation of network paths for the different edge server selection policies (i.e. SP, DP, D×LP). These data have been obtained by considering clients spread in 500 different locations across the network.

In the simulation study we explicitly avoid to model caching of DBMS data at the edge servers. This choice derives from that, as outlined before, this type of caching requires explicit mechanisms for the maintenance of the consistency of replicated data [11], which might impact on the latency seen by the users. Hence, we exclude caching of DBMS data in order to avoid any interference due to these mechanisms while performing the evaluation of our multi-path protocol. At the same time, we gather statistical data by only considering the latency experienced by users really performing updates of application data, for which caching of DBMS data at the edge servers provides no advantage due to the fact that the corresponding requests are redirected to the origin data centers in order to manipulate the original data copy. This actually ensures fairness in the evaluation.

Finally, to capture network congestion/overload situations, we have fixed the packet loss Gilbert model parameter q at the value of 0.8 which corresponds to an expected burst loss length of 1.25 (studies [18] have shown that consecutive losses rarely last more than four packets and this value of q corresponds to the longest average path we are aware of). For what concerns the parameter p, we have considered two different values in the simulation study, selected as representative of interconnection between edge serves and data centers either via Internet or via a (virtual) private network under the control of the ASPs. In the former case, p was set to yield a moderate end-to-end loss rate of 5% for an average path length of 3 to 16 hops, depending on the topology. In the latter case, p was set to yield the extremely reduced end-to-end loss rate of 1% for the same average path lengths. The message size distribution has been obtained through a Pareto with α=1.5 and b=2.

3.5 Results

We report in Figure 2 and in Figure 3 the cumulative distribution function (CDF) of browser perceived response times for the two considered Brite topologies (flat and hierarchical) and the NLANR topology. In other words, we report on the Y-axis the experimentally evaluated probability for a browser to experience a response time lower than the corresponding value on the X-axis. The plots comparatively report the browser perceived response time CDF when adopting a baseline algorithm not employing path diversity and our multi-path protocol (with the three different policies for selecting the edge servers to be contacted in parallel by the client). For our protocol, we have also varied the value of the timeout used at the data centers during the connection phase in the interval between 0 and 500 milliseconds.

By the plots we get that the multi-path protocol provides remarkable benefits, in terms of increased system responsiveness. For the case of edge servers communicating with data centers via the Internet (see Figure 2), exploiting path-diversity in the BRITE-f topology allows achieving browser perceived response times less than 7 seconds (i.e. less than the maximum value complying with a reasonable expectation for an interactive end-user [20]) in at least the 80% of the cases, whereas the baseline protocol achieves response times less than 7 seconds in the 65% of the cases (it behaves slightly better that this percentage only for the DP edge server selection policy). Analogous considerations hold also for the results obtained with the NLANR topology, where the multi-path protocol achieves browser perceived response times less than 7 seconds again in the 80% of the cases, whereas the baseline protocol achieves response times less than 7 seconds in less then the 65% of the cases. Slightly reduced advantages are observed for the BRITE-h topology where, despite the relevant amount of path diversity between clients and edge servers (see Table 1), the hierarchical organization of the network topology does not favor disjointness in between the edge servers and the back-end data centers. Also, network paths between edge servers and data centers even result significatively longer than network paths between clients and edge servers, which, together with that reduced level of disjointness, additionally contributes to reduced effectiveness of the multi-path approach.

The results related to the case of communication be-

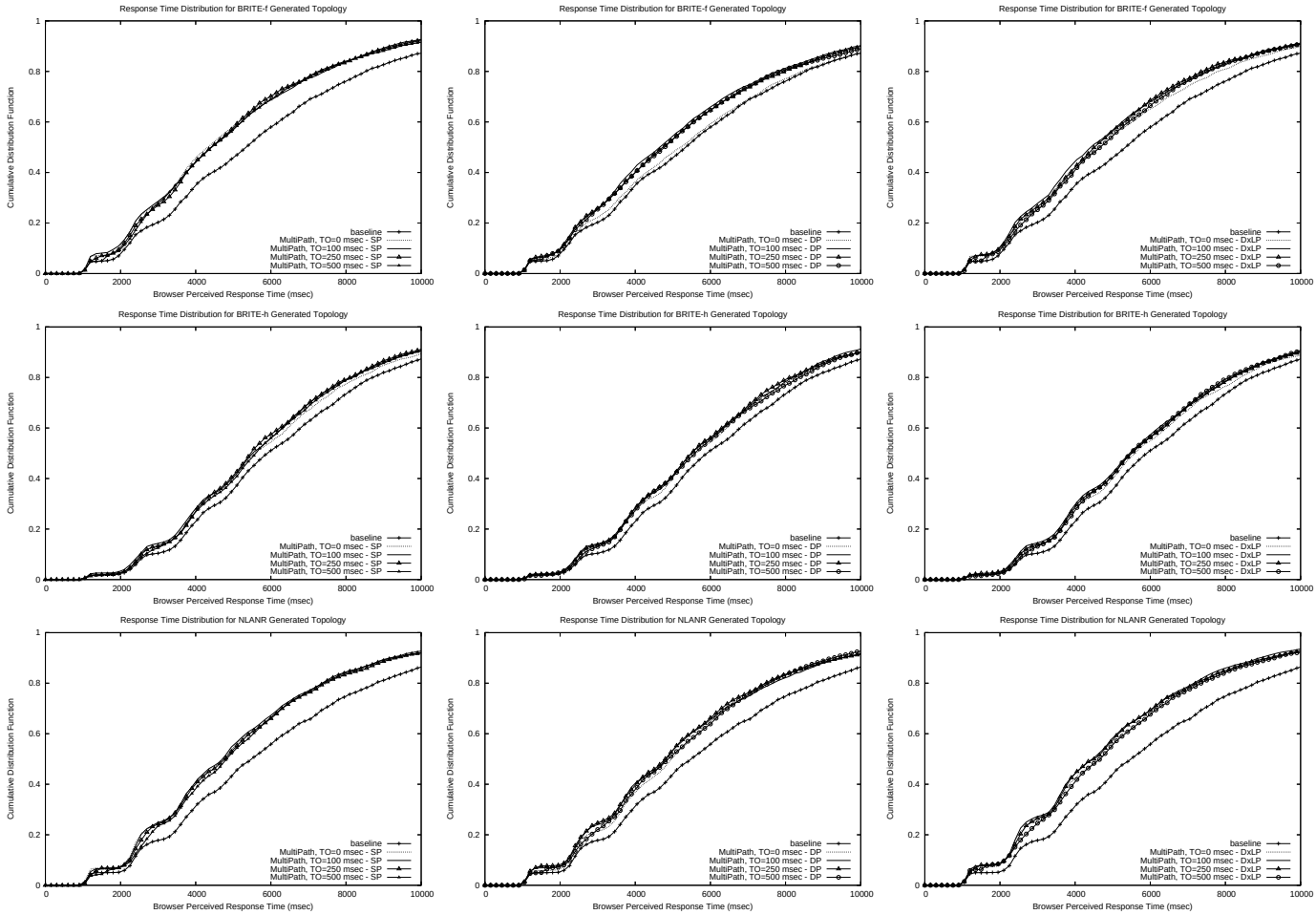

Figure 2. Browser Perceived Response Time CDF for the Case of Edge Servers and Data Centers Communicating via Internet.

tween edge servers and data centers via a (virtual) private network, see Figure 3, confirm the previous tendencies, with the only observation that, compared to the case of Internet based communication, this time we expect higher system responsiveness due to the more controlled network behavior at the side of the Web infrastructure (recall that for this configuration the parameter p has been set to obtain the extremely reduced packet loss percentage over a path of 1%). Hence, the advantages from the multi-path protocol need to be evaluated for response time on the order of the reasonable value of 3/4 seconds, which is guaranteed by the multi-path protocol in about the 90% of the cases. Instead, even in such a controlled network scenario, that response time is guaranteed by the baseline in the reduced percentage of the 80% of the cases.

Another important observation from the plots is that they show significant benefits from the multi-path protocol even in case of no exploitation of path correlation information in the selection of the edge servers to be contacted in parallel by the client. In fact, the benefits achieved by users employing the correlation unaware selection scheme, namely SP, are in practice identical to those achievable with the other selection policies. This is an interesting result that confirms the feasibility of the multi-path protocol also in environments where it is difficult or impossible to infer the path correlation of the underlying network topology.

The plots in Figure 4 and in Figure 5 provide a different perspective to quantify the benefits achievable through the multi-path approach. In these graphs we report the histograms of the percentage reduction in response time over the baseline for all the three considered network topologies and for the three edge server selection policies SP, DP and D$\times$LP. Such a data visualization highlights that there is a relevant percentage of clients that experiences a remarkable reduction in the perceived response time (evaluated as $\frac{Time_{baseline} - Time_{multi\_path}}{Time_{baseline}}$) when the multi-path approach

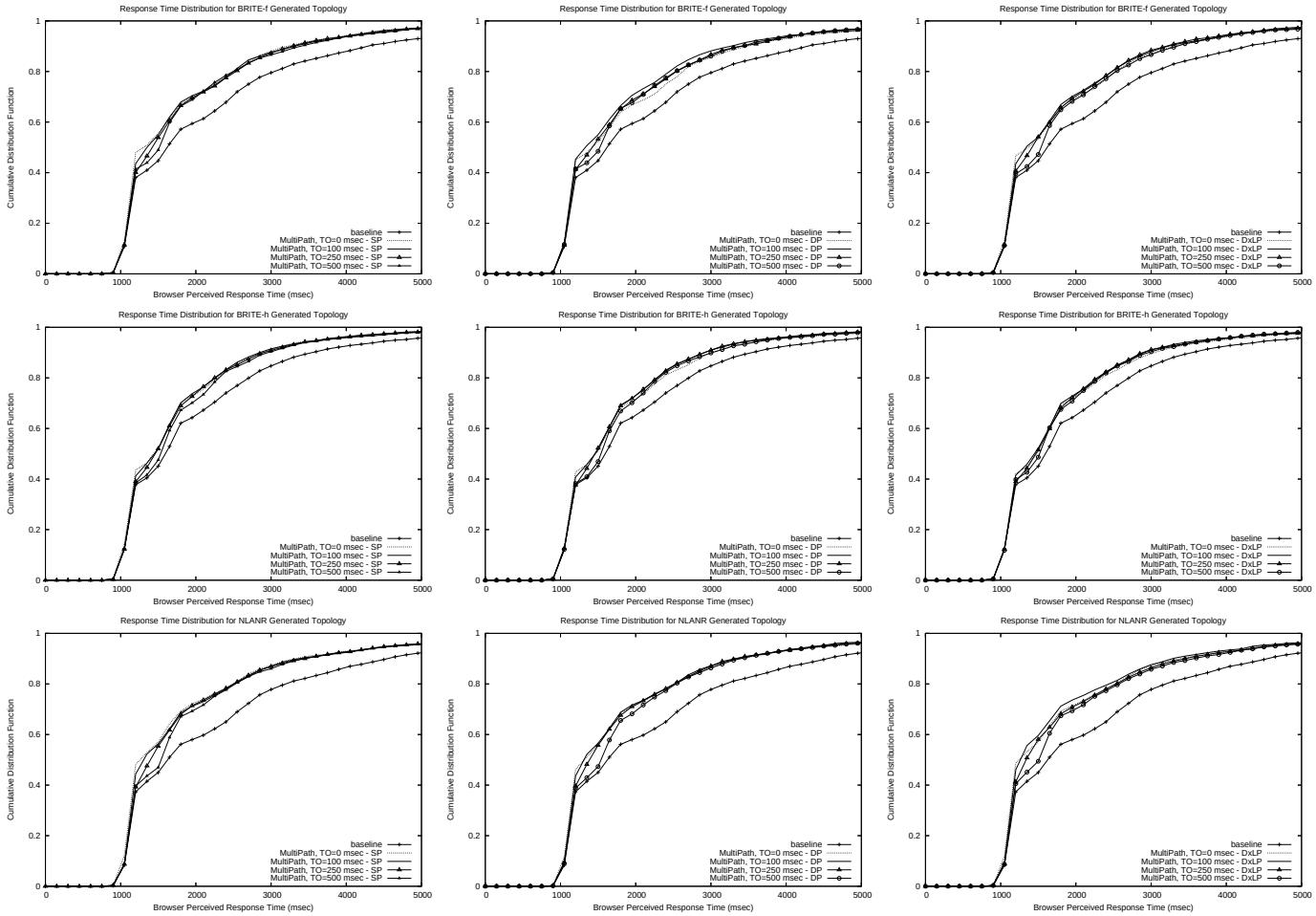

Figure 3. Browser Perceived Response Time CDF for the Case of Edge Servers and Data Centers Communicating via a (Virtual) Private Network.

is used. In all the topologies the percentage of clients that gets a response time reduction greater than (or equal to) 50% is at least the 50%, and the 25% of clients get at least a 70% reduction in the response time.

4 Assessments and Conclusions

In this paper we have shown via an extended simulation study how a multi-path approach can be an effective way to tackle network anomalies (such as congestion or failures). These can impact the user perceived response time in case of update requests that need access and manipulation of primary copies of e-Commerce application data residing at origin data centers. The simulation data clearly outline that multi-path does not provide benefits only in a limited number of system settings, instead its advantages span in a wide spectrum of system organizations ranging from, e.g., Internet to (virtual) private network interconnection at the server side. This points out how multi-path can be effectively employed in combination with any other technique optimizing the system run-time behavior. As a final note, to our knowledge this is the first study explicitly focused on evaluating multi-path approaches in the context of transactional (e.g. e-Commerce) applications.

References

[1] M. Allman, "A Web Server's View of the Transport Layer", *ACM Computer Communication Review*, vol.30, no.5, 2000.

[2] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek and R. Morris, "The case for resilient overlay networks", *Proceedings of HotOS VIII*, May 2001.

[3] J. Apostolopoulos and M.D. Trott, "Path diversity for enhanced media streaming", *IEEE Communications Magazine*, vol.42, no.8, pp.80–87, 2004.

[4] J. Apostolopoulos, T. Wong, W. Tan and S. Wee, "On Multiple Description Streaming with Content Delivery Networks", *Proceedings of IEEE INFOCOM*, June 2002.

[5] P. Barford, A. Bestavros, A. Bradley and M. Crovella, "Changes in Web Client Access Patterns - Characteristics and Caching Implications", *World Wide Web*, vol.2, no.1-2, pp.15–28, 1999.

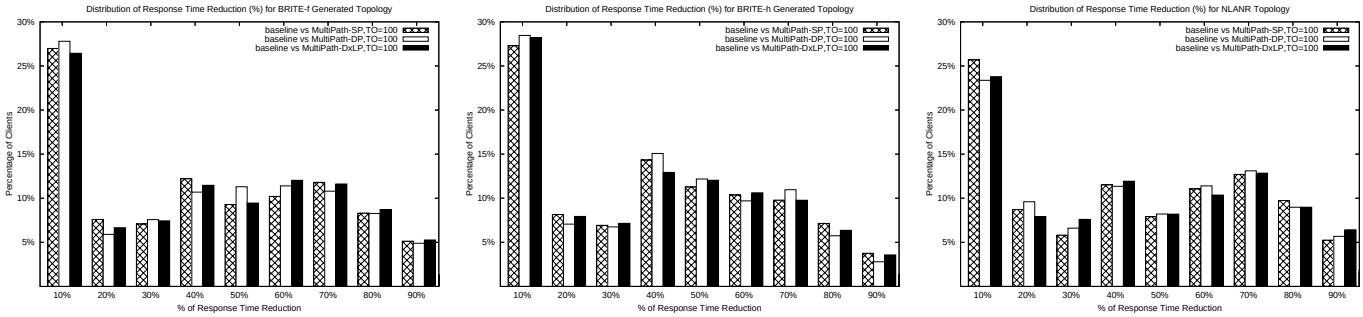

Figure 4. Distribution of Browser Perceived Response Response Time Reduction for the Case of Edge Servers and Data Centers Communicating via Internet.

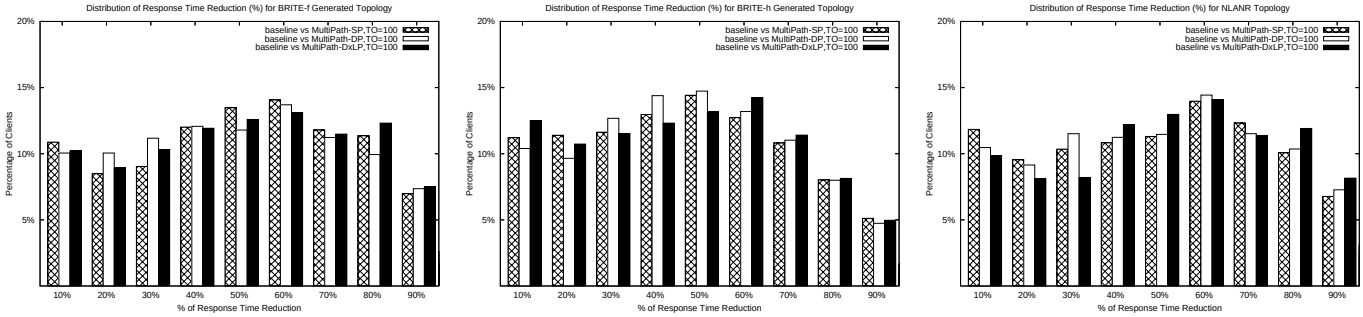

Figure 5. Distribution of Browser Perceived Response Response Time Reduction for the Case of Edge Servers and Data Centers Communicating via a (Virtual) Private Network.

[6] N. Bhatti, A. Bouch, and A. Kuchinsky, "Integrating User-perceived Quality into Web Server Design". *Proceedings of the 9th World-Wide Web Conference*, pp. 1-16, Amsterdam, The Netherlands, June 2000.

[7] Brite, Boston University Representative Internet Topology Generator, `http://www.cs.bu.edu/brite/`

[8] N. Cardwell, S. Savage, and T. Anderson, "Modeling the performance of short TCP connections", *Technical Report*, Computer Science Department, Washington University, Nov. 1998

[9] J. Challenger, P. Dantzig and A. Iyengar, "A Scalable and Highly Available System for Serv- ing Dynamic Data at Frequently Accessed Web Sites", *Proceedings of ACM/IEEE Supercomputing*, Orlando, Florida, November 1998.

[10] J. Challenger, A. Iyengar and P. Dantzig, "Scalable System for Consistently Caching Dy- namic Web Data", *Proceedings of the IEEE INFOCOM*, New York, March 1999.

[11] J. Gray, P. Helland, P. O'Neil and D. Shasha, "The dangers of replication and a solution", *Proceedings of the 1996 ACM SIGMOD international Conference on Management of Data*, Montreal, Quebec, Canada, pp.173–182, 1996.

[12] W.S. Li, W.P. Hsiung, D.V. Kalashnikov and R. Sion, "Issues and Evaluations of Caching Solutions for Web Application Acceleration", *Proceedings of the 28th VLDB Conference*, pp.1019–1030, Hong Kong, China, 2002.

[13] A. Iyengar and J. Challenger, "Improving Web Server Performance by Caching Dynamic Data", *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, 1997.

[14] F. Liu, Y. Zhao, W. Wang, D.J. Makaroff, "Database Server Workload Characterization in an e-Commerce Environment", *Proceedings of IEEE MASCOTS*, pp.475–483, 2004.

[15] NLANR, `http://www.nlanr.net/Routing/rawdata`

[16] P. Romano, F. Quaglia and B. Ciciani, "Design and Evaluation of a Parallel Edge Server Invocation Protocol for Transactional Applications over the Web", *Proceedings of the 6th IEEE Symposium on Applications and the Internet (SAINT)*, Phoenix, Arizona, USA, January 2006.

[17] Transaction Processing Performance Council (TPC), "TPC Benchmark^{TM} W", `http://www.tpc.org/tpcw`

[18] J. Wenyu and H. Schulzrinne, "Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality," *Proceedings of ACM NOSSDAV*, 2000.

[19] K. Yagoub, D. Florescu, V. Issarny and P. Valduriez, "Caching Strategies for Data Intensive Web Sites", *Proc. of the 26th VLDB Conference*, pp.188–199, Cairo, Egypt, 2000.

[20] Zona Research, `http://www.zonaresearch.com/`