LMS Performance Issues: A Case Study of D2L

Sourish Roy Carey Williamson Rachel McLean
Department of Computer Science, University of Calgary
Calgary, AB, Canada T2N 1N4
(sourish.roy, cwill, rarmclea)@ucalgary.ca

Abstract

This paper presents a network traffic measurement study of Desire2Learn (D2L), which is the Learning Management System (LMS) used by the University of Calgary. Our study is motivated by anecdotal reports of sluggish D2L performance, particularly for file uploads. Using active and passive network measurements, we identify the root causes of the poor D2L performance. The main issues identified are: (1) excessive HTTP redirections in our university's D2L setup; (2) non-negligible network latency to the server hosting the D2L content; and (3) suboptimal TCP settings that limit end-to-end throughput. We discuss these issues, and identify potential solutions, including caching. Our discrete-event simulation results indicate that a simple Web proxy cache at our university could reduce D2L requests by 50-70%.

keywords: Learning Management System (LMS), Network traffic measurement, User-perceived performance, Web-based systems, TCP, Throughput, Response time

1 Introduction

Almost every university has a Learning Management System (LMS) to help support its educational mandate. LMS technology augments classroom learning with support for online learning. The concept of e-learning has been around for a long time, but it is only over the last decade that the deployment of Web-based LMS software has become ubiquitous. Prior to Internet deployment, most LMS solutions were closed systems within individual institutions.

An LMS is essential for managing the learning activities for courses online. It provides instructors with techniques to create and deliver content. It also allows them to check student participation and engagement in the course curriculum, and assess student performance on assignments, quizzes, and exams.

One popular LMS is Brightspace by D2L (Desire2Learn), which we refer to as "D2L" in this paper [7]. D2L was started by John Baker in 1999 as a system to manage courses and student learning. The company is headquartered in Kitchener, Ontario, Canada, and has more than 800 employees in Canada, Australia, Brazil, Singapore, UK, and USA.

In 2014, the University of Calgary selected D2L as its new LMS, as a replacement for Blackboard. Every instructor and student now has access to D2L as our official LMS. Thus, thousands of on-campus users are using D2L each day to create/view course content, record/watch lectures, and enter/view grades. These LMS activities generate a lot of network traffic, using thousands of TCP connections, from many IP addresses, and multiple heterogeneous devices. Knowledge of the D2L traffic patterns can help us understand its impact on the learning environment at the University of Calgary. Thus a workload characterization study of D2L LMS traffic is warranted [17].

Anecdotal reports from faculty/staff at the University of Calgary indicate that D2L is "slow". This problem has existed since 2014, but has not yet been resolved. One possible reason for the sluggishness is that D2L content is hosted remotely in Ontario, approximately 3200 km from Calgary. Indeed, our network measurements confirm that this is an important contributing factor. However, we also find other technical issues with the D2L configuration that hamper user-perceived performance. For example, file uploads for content producers (i.e., faculty/staff) are much slower than file downloads for content consumers (i.e., students).

An analysis of D2L traffic can help identify the reasons for its slow performance. In computer networking, traffic measurement and analysis are crucial to the design, operation, and maintenance of local and wide-area networks. This area of research [11] is used extensively in both academia and industry. By collecting network traffic measurement data, we can assess the usage of a network, and develop improved communication protocols for future networks.

The rest of this paper is organized as follows. Section 2 provides some background on network traffic measurement, and prior research literature. Section 3 describes the research methods used for our work. Section 4 presents the results from our study, focusing on HTTP redirection, network latency, and TCP throughput. Section 5 presents more recent measurement results to further investigate D2L performance. Section 6 concludes the paper.

2 Background & Related Work

Network traffic measurement provides a means to analyze network usage and understand network performance. Lots of prior research has characterized Internet traffic from the early 1990's [9] to the present day [11] (e.g., Web traffic [3, 6, 13, 15], peer-to-peer file sharing [4], YouTube [10, 12], online social networks [5], Netflix [1, 14]). Network traffic measurement studies such as these are useful not only for workload characterization, but also for network troubleshooting, protocol debugging, and performance evaluation.

In 1996, Arlitt *et al.* [3] characterized Web server workloads by analyzing access logs. These logs record requests for Web site URLs, including time of request, client IP address, content accessed, and document size. They used 6 different data sets in their study: three from universities, two from research organizations, and one from a commercial Internet provider. They identified common workload characteristics, such as Zipf-like object popularities and heavy-tailed file size distributions. Based on their findings, they proposed improved Web caching systems with frequency-based cache management policies.

In 2001, Almeida *et al.* [2] analyzed server log data for educational media servers at two major US universities (University of Wisconsin, and University of California, Berkeley). Their paper focused on the eTeach system and BIBS (Berkeley International Broadcasting System), which delivered high quality media content. Their study provides a benchmark against which future media server workloads can be compared.

Newton et al. [15] conducted a long-term Web traffic measurement study to see how this traffic has changed over time. They used the TCP/IP packet headers (1999-2012) in packet traces collected on the Internet link for the University of North Carolina. They performed an in-depth analysis of HTTP request sizes and responses, identifying growth in the size and complexity of Web pages, and increased use of cookies.

Our paper builds upon the methods and insights from these prior works, and focuses on the network performance of D2L, as an example of an LMS.

3 Methodology

Network traffic measurement can be classified [11, 20] based on the location of the network monitor (e.g., edge or core), the network analysis tools used (e.g., hardware or software), and the data collection mechanism (e.g., passive or active). In our work, we use a combination of passive and active measurement approaches.

In passive network measurements, data is gathered by passively listening to network traffic, without generating any additional traffic. In our study, we use specialized hardware to collect information about all campus-level traffic on our edge network. The monitor records information about the inbound/outbound network traffic passing through the university's edge routers. This collection takes place through a mirrored stream of all packet-level Internet traffic entering/leaving the University of Calgary network.

Our network monitor is a Dell server, which processes the mirrored traffic stream. It is equipped with two Intel Xeon E5-2690 CPUs (32 logical cores @2.9 GHz), 64 GB RAM, and 5.5 TB of local hard disk storage for the logs. The operating system (OS) on this server is CentOS 6.6 x64. The monitor utilizes an Endace DAG 8.1SX for capturing the traffic and filtering it. It was designed for 10 Gbps Ethernet, and uses several programmable functions in the hardware to boost the performance of packet processing. The primary use of the Endace DAG card is to split the incoming traffic into streams for processing by the Bro logging system.

Bro is an open-source framework for network analysis and security [8, 16]. In our work, the Bro logging system monitors all packet-level network activities, and produces connection-level logs summarizing all the traffic. Our primary interest is in the connection, HTTP, and SSL logs. The connection logs provide data regarding each observed connection, such as start time, end time, bytes transferred (inbound/outbound data), duration, and termination state. The HTTP log helps us identify the source/destination IPs, HTTP methods, hosts, URIs, referer URLs, and user agents. Finally, the SSL logs show us HTTPS connections, with fields like timestamps, TLS/SSL encryption methods, plus source/destination ports.

Bro collects and generates logs on an hourly basis, which we aggregate together to provide a semester-long view of D2L traffic. We collect and analyze data from the HTTP, SSL, and connection logs to produce the results reported in this paper.

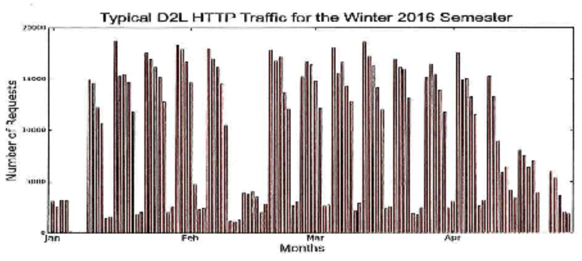
In addition to the Endace/Bro data collection described above, we also use Wireshark [21] to collect packet-level details on several D2L test sessions from our own desktop computers. Wireshark captures packets in real time, and displays them in a human-readable format. Using Wireshark, we can explore the details of D2L interactions for our own test sessions.

Unlike passive approaches, active measurements generate extra packets on the network. These can be used to measure the time taken to reach a target destination, the capacity available for a network path, or the response time for an application. Since this category of measurement generates additional traffic, we have performed active measurements judiciously, using basic active measurement tools like `ping` and `traceroute` that have minimal impact on the network.

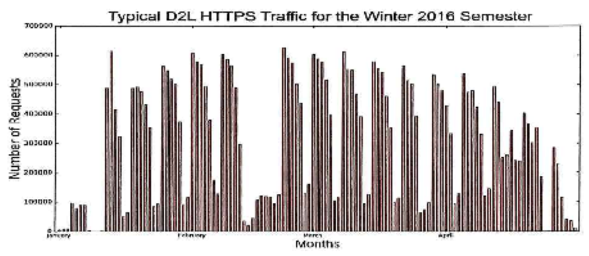
4 Measurement Results

4.1 D2L Traffic Overview

Figure 1 provides a high-level overview of the D2L traffic observed on our campus network during the Winter 2016 semester (January-April 2016). Note that D2L traffic occurs over both HTTP (for initiation/termination of D2L sessions) and HTTPS (for actual D2L interactions), and that there is a strong correlation between the two types of traffic.



(a) HTTP requests per day



(b) HTTPS requests per day

Figure 1: D2L Traffic Profile for Winter 2016

Our measurements allow us to quantify the traffic volume, data volume, response time, and throughput for all D2L users during the Winter 2016 semester. When lectures began on January 11, the D2L traffic increased. The D2L traffic pattern varies throughout the semester, with a dip during Reading Week break in February, and a sharp decline after final exams in April.

The D2L traffic shows strong daily and weekly patterns, with weekday traffic far exceeding that on weekends. On a typical weekday, we observe about 16,000 HTTP requests to D2L from the University of Calgary network, and about 500,000 HTTPS requests to D2L. The 30-fold difference between HTTP and HTTPS indicates that most D2L interactions occur via HTTPS. The HTTP traffic is primarily for session initiation/termination.

4.2 HTTP Redirection Issue

The first D2L performance issue that we have identified is related to how D2L is configured to operate within the University of Calgary IT infrastructure. In particular, session initiation involves user authentication. This step actually involves several HTTP redirections to the Central Authentication Service (CAS) at the University of Calgary. These interactions are complex, and add noticable latency to the D2L experience.

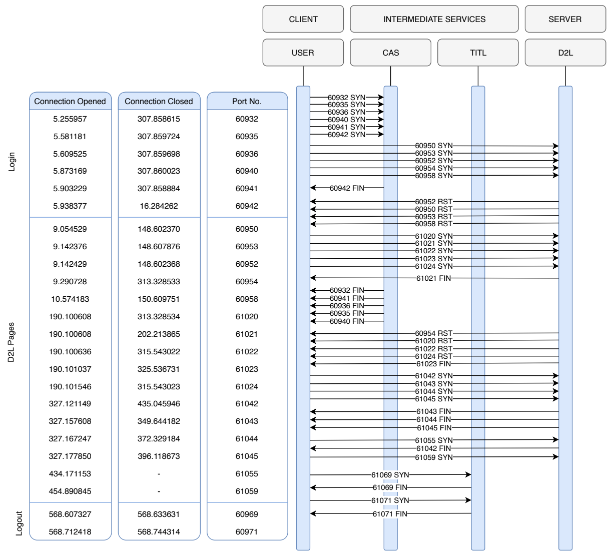
Figure 2 shows a schematic illustration of a D2L test session that we conducted. This session lasts about 10 minutes, and involves several steps. First, the initial attempt to contact D2L via HTTP is redirected to use HTTPS instead. Second, the request is redirected from D2L to CAS at the University of Calgary for user authentication. Third, the Web browser uses multiple TCP connections in parallel to load the CAS login page (i.e., CSS file, logo, background, Javascript). Fourth, once the user logs in successfully, another HTTPS redirection occurs to re-connect with D2L. The Web browser then launches multiple TCP connections to retrieve the different components of the D2L landing page, including colour template, university logo, menu buttons, and course home page. The user is now ready to begin their D2L session. In this test session, the user browses several course pages, viewing slides from the course content, and uploading and downloading a few files. Finally, the user logs out.

During logout, another series of HTTP redirections occur. The first of these is from D2L to an e-learning server hosted by the Taylor Institute for Teaching and Learning (TITL) at the University of Calgary. Next, the Web browser uses parallel TCP connections to load the different components of the session logout page. Finally, there is a superfluous HTTP redirection from the TITL server to itself, to change the URL from "`logout`" to "`logout/`".

4.3 Network Latency Issue

The second D2L performance problem relates to how far away the D2L server is from the University of Calgary. In particular, the network round-trip-time (RTT) is about 40 ms, which is non-negligible.

Figure 3 shows how an on-campus user accesses D2L. In this figure, the campus network is enclosed within a triangle, while the D2L hosted service in Ontario is



	Connection Opened	Connection Closed	Port No.
Login	5.255957	307.858615	60932
	5.581181	307.859724	60935
	5.609525	307.859698	60936
	5.873169	307.860023	60940
	5.903229	307.858884	60941
	5.938377	16.284262	60942
D2L Pages	9.054529	148.602370	60950
	9.142376	148.607876	60953
	9.142429	148.602368	60952
	9.290728	313.328533	60954
	10.574183	150.609751	60958
	190.100608	313.328534	61020
	190.100608	202.213865	61021
	190.100636	315.543022	61022
	190.101037	325.536731	61023
	190.101546	315.543023	61024
	327.121149	435.045946	61042
	327.157608	349.644182	61043
	327.167247	372.329184	61044
	327.177850	396.118673	61045
	434.171153	-	61055
	454.890845	-	61059
Logout	568.607327	568.633631	60969
	568.712418	568.744314	60971

Figure 2: Example of a D2L Session

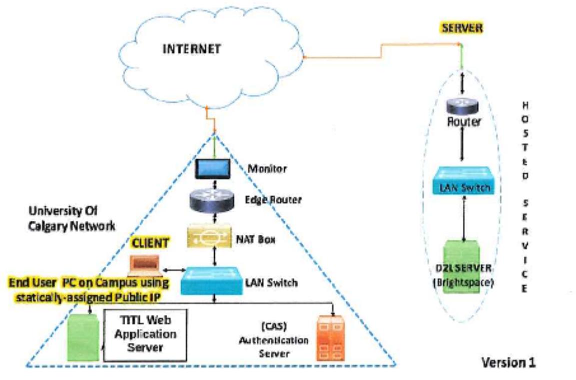

INTERNET
SERVER
Router
LAN Switch
D2LSERVER (Brightspace)
HOSTED SERVICE
Monitor
Edge Router
University Of Calgary Network
CLIENT
NAT Box
End User PC on Campus using statically-assigned Public IP
LAN Switch
TITL Web Application Server
(CAS) Authentication Server
Version 1

Figure 3: Network Path for D2L Users on Campus

indicated by the oval on the right. We are interested in characterizing the Internet path between the two.

Figure 4 shows the `traceroute` results, which indicate that the D2L hosted service (i.e., `desire2learn.ip4.torontointernetexchange.net`) is located at a data center in Toronto. The network path has 17 hops with a total RTT of 37 ms.

A recurring theme in our study is the adverse impact of network latency on user-perceived performance in D2L. The performance of D2L is affected by these high RTT values. Users spend time waiting for responses from a distant data center in Toronto. This hinders

```
$ traceroute d2l.ucalgary.ca
traceroute to d2l.ucalgary.ca (199.30.181.42), 30 hops max, 60 byte packets
 1  deptNFSgate (172.17.10.1)  0.233 ms  0.217 ms  0.302 ms
 2  * * *
 3  10.58.48.1 (10.58.48.1)  0.367 ms  0.370 ms  0.363 ms
 4  * * *
 5  10.16.18.1 (10.16.18.1)  0.433 ms  0.404 ms  0.401 ms
 6  10.16.18.4 (10.16.18.4)  0.302 ms  0.246 ms  0.237 ms
 7  10.16.17.1 (10.16.17.1)  0.438 ms  0.403 ms  0.432 ms
 8  10.59.226.26 (10.59.226.26)  0.334 ms  0.324 ms  0.333 ms
 9  h74.gpvpn.ucalgary.ca (136.159.199.74)  3.296 ms  3.333 ms  3.471 ms
10  h66-244-233-17.bigpipeinc.com (66.244.233.17)  0.744 ms  0.633 ms  0.624 ms
11  h208-118-103-166.bigpipeinc.com (208.118.103.166)  0.880 ms  0.869 ms  0.836 ms
12  clgr2rtr2.canarie.ca (199.212.24.66)  0.721 ms  0.755 ms  0.726 ms
13  wnpg1rtr2.canarie.ca (205.189.33.199)  36.400 ms  36.180 ms  36.307 ms
14  canariecds.ip4.torontointernetxchange.net (206.108.34.170)  36.543 ms  36.514 ms  36.368 ms
15  desire2learn.ip4.torontointernetxchange.net (206.108.34.184)  36.668 ms  36.511 ms  36.484 ms
16  * * *
17  ucalgary.desire2learn.com (199.30.181.42)  36.727 ms  36.810 ms  36.770 ms
```

Figure 4: Traceroute Results for `d2l.ucalgary.ca`

the responsiveness of the D2L Web site, particularly when multiple HTTP/HTTPS redirections occur. Furthermore, the network bandwidth is not well utilized during TCP slow start, and D2L performance suffers.

4.4 TCP Throughput Issue

The third problem with D2L at our university is the TCP throughput, which is an important factor that affects network application performance. Using our empirical measurement data, we calculated the Average Data Rate (ADR) for D2L data transfers, which is the size of a transferred file divided by the elapsed time duration. This metric indicates the average throughput for D2L connections, in bits per second (bps).

Figure 5 shows a Log-Log Complementary Distribution (LLCD) plot of the ADR from some of our empirical data. The average ADR is 500 Kbps, with some data points up to 5 Mbps for inbound connections. A much lower ADR is seen for outbound connections, with the average being 50 Kbps, and a maximum ADR of around 350 Kbps. Note that these throughput values represent only the average, and not the instantaneous throughput. Specifically, they are calculated from the byte counts and the durations reported in the connection logs, and the duration includes all the TCP connection handshaking, slow start effects, and any timeouts used for persistent connections.

There are two intriguing observations in Figure 5. First, the throughput values are very low (i.e., much lower than expected on CANARIE's fast national network). Second, the average throughput differs for uploads and downloads, almost by a factor of two.

To further investigate this issue, we conducted some D2L test sessions involving uploads and downloads for a single 3.2 MB data file. Table 1 summarizes the results of our experiment, which confirms the asymmetric performance for uploads and downloads. The highest throughput achieved for downloads was 14 Mbps, while that for uploads was 7 Mbps. These results were

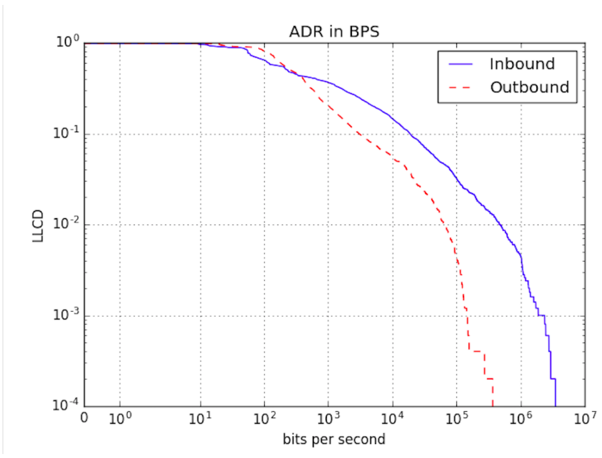


Figure 5: LLCD Plot of D2L Throughput

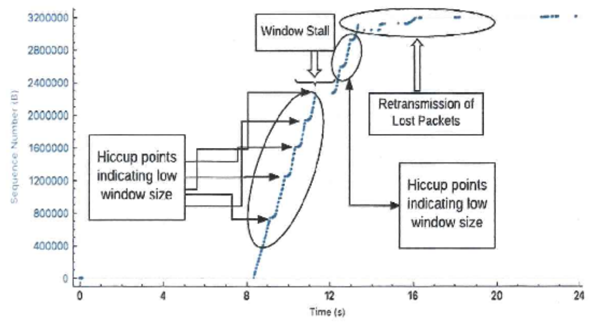


Figure 6: TCP Seqnum Plot for D2L File Upload

consistent across all test scenarios considered.

Table 1: TCP Throughput for D2L Transfers (3.2 MB)

Scenario	Device	OS	Download	Upload
On campus, wired	Desktop	Windows 8	14 Mbps	7 Mbps
On campus, wireless	Laptop	Mac OS X	14 Mbps	7 Mbps
Off campus, wireless	Laptop	Mac OS X	14 Mbps	7 Mbps

We used Wireshark and some active measurements to learn more about the TCP version and settings used by D2L data transfers. Wireshark provides information such as TCP options, maximum segment size (MSS), slow start, window size, sequence number analysis, and others. OS fingerprinting allows us to infer the operating system (Windows 2008 R2) and TCP version (Compound TCP [18, 19]) used by the D2L server, which is running Microsoft's Internet Information Server (IIS version 7.5).

Figure 7 illustrates the TCP receiver window size advertised by the D2L server during a file upload. The first observation is that the maximum advertised window size is 64 KB, which is the default socket buffer size for Compound TCP. This is a very small window size to use on networks with a large delay-bandwidth product, such as our scenario. The second observation is that the advertised window size fluctuates a lot, indicating that the server is slow in processing the arriving data packets. There are a half-dozen occurrences of small windows where the data transfer is inhibited. There is even a window stall event between 11 and 12 seconds, where the receiver window is almost zero (395 bytes, too small for the uploader to send another MSS).

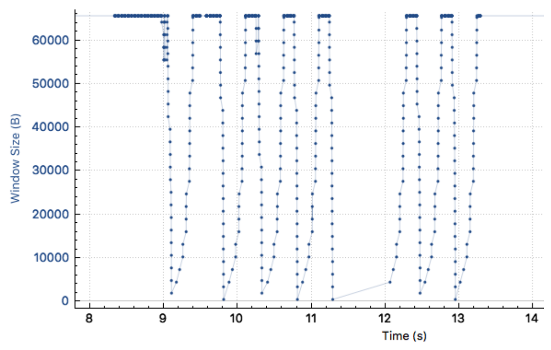


Figure 7: TCP Receive Window for D2L File Upload

These results demonstrate that the D2L data transfer performance is window-limited. Even if data transfers were perfect, with 64 KB of data exchanged every 40 ms, the maximum throughput would be 14 Mbps, which is what we observed in our download experiments. A larger window size of approximately 1 MB would be required to better exploit the network path between Calgary and Toronto.

Understanding why upload performance (7 Mbps) is worse than downloads (14 Mbps) requires even further investigation. To obtain insight into this problem, we conducted our own active measurement experiment on a D2L test session using special software called `mitmproxy`, which acts as a man-in-the-middle (mitm) proxy. This software intercepts the traffic between a client and a server, and can report all the HTTP/HTTPS traffic requests made by the user.

With `mitmproxy` in place, we can view the details of our D2L test sessions, including HTTP requests/responses, file names/sizes, and response times. These experiments showed that downloads use the GET method, while uploads use the POST method. However, the file uploads involve many POST requests, each with a small transfer size. Furthermore, D2L internally updates a file directory structure on uploads, as indicated by UpdateTreeBrowser in one of its URLs. In addition, there is an activity feed popup right after the new content is uploaded into D2L, as a notification for the user. These activities all increase the delay for D2L file uploads.

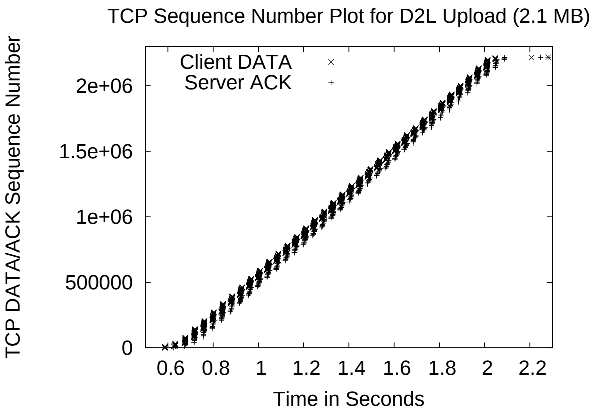


Figure 8: File Upload with D2L's IIS 10.0 Server

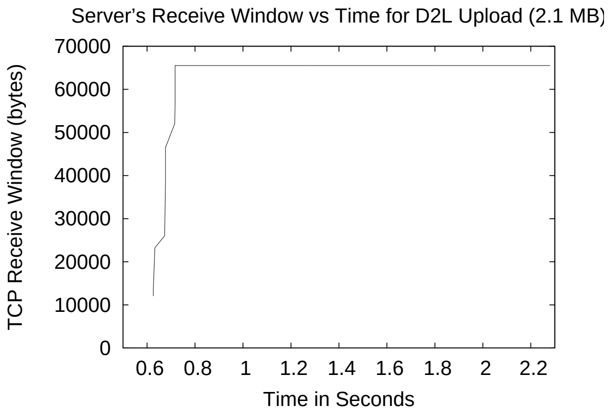


Figure 9: TCP Window during File Upload (IIS 10.0)

5 Additional Results

In May 2018, we repeated our D2L traffic measurements to see what, if anything, had changed from our earlier measurements in Winter 2016. Our additional results focus on D2L server performance, recent upgrades, and potential Web proxy cache performance.

5.1 D2L Server

The first observation from our May 2018 measurements is that the server-side infrastructure for D2L has changed since Winter 2016. In particular, the HTTP response headers now show that the D2L Web server is running IIS 10.0, rather than IIS 7.5. The server hardware and operating system have also been upgraded.

This upgrade has improved the performance for D2L file uploads, as shown in Figure 8. In this example of a 2.1 MB file upload, the TCP sequence number plot shows that the data transfer is completed in about 1.7 seconds, for an average throughput of 14 Mbps. Furthermore, Figure 9 shows that the server has no problem keeping up with the data, since the receiver advertised window for TCP is almost always 64 KB. To be specific, the initial advertised window is about 12 KB, but this window is dynamically resized by TCP until it reaches the maximum of 64 KB, and stays at that value for the rest of the transfer.

The good news here is that uploads are now just as fast as downloads; the bad news is that uploads and downloads are both still window-limited. The TCP sequence number plot in Figure 8 still shows bursts of 64 KB at a time, with an ensuing wait of about 40 ms for the window's worth of acknowledgements to return. As a result, the average throughput never exceeds 14 Mbps, despite the multi-Gbps network path between the client and the server. TCP window scaling would be required to better utilize this network path.

5.2 D2L User Interface

On May 4, 2018, our university launched a new version of D2L with an improved user interface. The intent of the new release was to improve the user experience, with a new design layout, easier navigation, and mobile-friendly content. We collected detailed measurements both before and after the new release, in order to understand the differences. Importantly, this "upgrade" was only to the user interface, and did not address any of the fundamental network performance issues identified earlier.

While the new design is aesthetically pleasing, there are two new problems with the D2L deployment at our university. One problem is that the D2L home page is much larger and more complicated than it was before. Another problem is that D2L seems to have made almost all content uncacheable. We comment on these two issues next.

Table 2 provides a comparison of D2L Web pages both before and after the user interface upgrade. These are examples of pages in a D2L browsing session for a Computer Science student registered in the CPSC 413 course on Complexity Theory. The most striking observation is that the D2L home page, which all students visit by default upon login, has approximately doubled in size (KB) and complexity (number of objects). This results in slower Web browsing performance than before the upgrade. The other example pages are comparable to or smaller than they were before.

Table 2: Comparison of D2L Web Page Complexity

Type of Web Page	Before Update		After Update	
	Objects	Size	Objects	Size
D2L Home Page	21	638 KB	45	1,239 KB
Course Home Page	27	466 KB	11	567 KB
Course Content Page	21	219 KB	17	173 KB
View Content Page	19	44 KB	16	52 KB

The second issue relates to content caching. Based on the analysis of HTTP response headers in Wireshark, most D2L content is now marked as uncacheable (i.e., "`private, max-age=0`"), unlike the previous version of D2L which allowed caching of static content (i.e., "`public, max-age=3600''`). Such content was typically cacheable for one hour (3600 seconds).

The underlying reason for making content uncacheable is not clear. We speculate that it is to enable full tracking of student learning activities, with all Web requests coming to the origin server. Another possibility is that it reflects new privacy regulations, so that no user data is stored without their knowledge. The downside of uncacheable content is that the content must be retrieved repeatedly from the origin server far away. The most glaring example in our experiments was the user profile image (21 KB), which was downloaded 61 times during a 10-minute browsing session.

As with our earlier work, we have shared these results with the University of Calgary Information Technologies (UCIT) team at our university. They are in touch with D2L to find better technical solutions for our D2L performance problems.

5.3 Web Caching

Our final result relates to the potential benefits of Web proxy caching to improve D2L performance at our university. We use trace-driven simulation for this analysis, with an arbitrarily chosen D2L workload trace from February 14, 2018. There were approximately 402,000 Web object transfers on that day.

Our discrete-event simulation models a simple Web proxy cache on campus, which stores and remembers previously seen static Web objects. Since the cache has a finite size, a cache replacement policy is used to manage the cache, and remove previous items when more space is needed to store a new item. We use this caching simulator to estimate the potential savings in D2L requests with a Web proxy cache.

We consider several different replacement policies to manage the cache. RAND (Random) removes a randomly-chosen object when more space is needed. FIFO (First In First Out) removes the oldest object in the cache. LRU (Least Recently Used) removes objects that have not been used recently. SIZE removes large objects, while keeping small objects. LFU (Least Frequently Used) removes objects that have not been used often. We consider two variants of the latter: In-Cache LFU (CLFU) only tracks the popularity of objects that are in the cache, resetting the popularity of an object to zero when it leaves the cache, while Global LFU (GLFU) always remembers the cumulative popularity of an object, whether it is still in the cache or not.

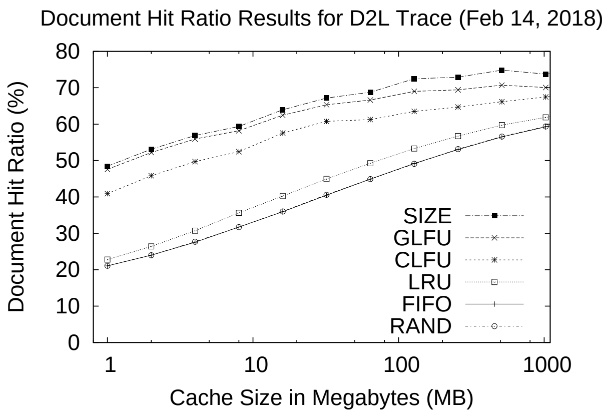


Figure 10: Simulation Results for Web Proxy Cache

Figure 10 shows the results from our Web proxy cache simulations. On this graph, the horizontal axis shows the cache size in Megabytes (MB) on a logarithmic scale, while the vertical axis shows the Document Hit Ratio (DHR) on a linear scale. Higher values of DHR are better, since they represent more objects being retrieved from the local proxy cache on campus, and fewer requests going all the way to the D2L server.

In general, the results in Figure 10 show very good potential for Web proxy caching. Even a modest size cache (10 MB) can provide a hit ratio of over 50%, reducing by half the number of requests to the origin server. With a cache size of 1 GB, the DHR would be approximately 70%.

Figure 10 also shows that there are notable performance differences among the replacement policies used to manage the cache. The best policy is SIZE, which focuses on caching a large number of small objects, such as the icons, sprites, and logos used on many of the D2L pages. The frequency-based policies also perform well, with GLFU always outperforming CLFU (as expected). The LRU policy is next best. FIFO and RAND are poor cache management policies, as expected.

Two caveats apply to these Web proxy caching results. First, these results assume that all D2L objects are static Web content, and eligible for caching. This assumption may not hold for dynamic content, or for uncacheable requests that D2L uses to track student learning behavior. Second, our caching study ignores the issue of end-to-end encryption, which D2L uses to protect user privacy. Any practical caching solution for D2L, such as a Web proxy cache or a CDN node, would have to deal with these two issues.

6 Conclusions

In this paper, we presented an empirical measurement study of the Desire2Learn (D2L) Learning Management System (LMS) adopted for use at the University of Calgary. The motivation for our study was to gain a better understanding of the system configuration, and its performance limitations.

While studying an LMS such as D2L is complex, there are three main technical issues that emerge from our study as root causes for the poor performance of D2L. The first issue is the excessive use of HTTP redirection at the University of Calgary to manage login/logout for D2L sessions. The second issue is the network RTT latency for D2L users in Calgary to access course content that is remotely hosted in Ontario. Finally, the TCP configuration on the D2L server has a maximum window size of 64 KB, which limits data transfer throughput.

The main conclusion from our study is that D2L is slow, and unnecessarily so. Fortunately, the observed performance problems are all fixable, as follows. First, we observed over one million HTTP redirects during the Winter 2016 term, which could be eliminated to minimize network round-trips and reduce server load. Second, there is a 40 ms RTT latency for University of Calgary users to access D2L content. Using a content delivery network (CDN), or placing a CDN node locally on campus, could greatly accelerate content delivery. Our simulation-based study suggests that a simple Web proxy cache, properly managed, could reduce D2L content requests by 50-70%. Finally, the TCP window size used by D2L is small, and does not scale dynamically based on the observed characteristics of the network path. An expanded TCP window size would solve this problem, improving throughput for both uploads and downloads. We hope that the insights from our study will improve future deployments of the D2L LMS, both at our university and elsewhere.

Acknowledgements

Financial support for this research was provided by NSERC. The authors thank D'Arcy Norman at TITL for his detailed knowledge of D2L, and the CATA 2018 reviewers for their feedback and suggestions on the paper.

References

[1] V. Adhikari, Y. Guo, F. Hao, V. Hilt, Z-L. Zhang, M. Varvello, and M. Steiner, "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs", *IEEE/ACM Transactions on Networking*, Vol. 23, No. 6, pp. 1984-1997, December 2015.

[2] J. Almeida, J. Krueger, D. Eager, and M. Vernon, "Analysis of Educational Media Server Workloads", *Proceedings of ACM NOSSDAV*, Port Jefferson, NY, January 2001.

[3] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", *IEEE/ACM Trans. on Networking*, Vol. 5, No. 5, pp. 631-645, Oct. 1997.

[4] N. Basher, A. Mahanti, A. Mahanti, C. Williamson, and M. Arlitt, "A Comparative Analysis of Web and Peer-fo-Peer Traffic", *Proceedings of WWW*, pp. 287-296, Beijing, China, April 2008.

[5] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing User Behavior in Online Social Networks", *Proceedings of ACM IMC*, pp. 49-62, Chicago, IL, November 2009.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications", *Proceedings of IEEE INFOCOM*, pp. 126-134, New York, NY, March 1999.

[7] Brightspace By D2L, The Brightspace Cloud Content Delivery Network. `https://community.brightspace.com/resources` (Aug 2017)

[8] Bro, The Bro Network Security Monitor, `https://www.bro.org` (Jan 2018)

[9] R. Caceres, P. Danzig, S. Jamin, and D. Mitzel, "Characteristics of Wide-area TCP/IP Conversations", *Proceedings of ACM SIGCOMM*, pp. 101-112, Zurich, Switzerland, August 1991.

[10] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User-Generated

Content Video System", *Proceedings of ACM IMC*, pp. 1-14, San Diego, CA, November 2007.

[11] M. Crovella and B. Krishnamurthy, *Internet Measurement: Infrastructure, Traffic and Applications*, John Wiley & Sons, 2006.

[12] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "YouTube Traffic: A View from the Edge", *Proceedings of ACM IMC*, pp. 15-28, San Diego, CA, November 2007.

[13] F. Hernandez-Campos, K. Jeffay, and F. Smith, "Tracking the Evolution of Web Traffic: 1995-2003", *Proceedings of IEEE MASCOTS*, pp. 16-25, Orlando, FL, October 2003.

[14] M. Laterman, M. Arlitt, and C. Williamson, "A Campus-Level View of Netflix and Twitch: Characterization and Performance Implications", *Proceedings of SCS SPECTS*, pp. 15-28, Seattle, WA, July 2017.

[15] B. Newton, K. Jeffay, and J. Aikat, "The Continued Evolution of Web Traffic", *Proceedings of IEEE MASCOTS*, pp. 80-89, San Francisco, CA, August 2013.

[16] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-time", *Computer Networks*, Vol. 31, No. 23, pp. 2435-2463, December 1999.

[17] S. Roy, *Characterizing D2L Usage at the U of C*, MSc Thesis, University of Calgary, August 2017.

[18] K. Tan and J. Song, "Compound TCP: A Scalable and TCP-friendly Congestion Control for High-speed Networks", *Proceedings of 4th International workshop on Protocols for Fast Long-Distance Networks*, Nara, Japan, February 2006.

[19] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A Compound TCP Approach for High-Speed and Long-Distance Networks", *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2009.

[20] C. Williamson, "Internet Traffic Measurement", *IEEE Internet Computing*, Vol. 5, No. 6, pp. 70-74, November/December 2001.

[21] Wireshark. `www.wireshark.org` (Jan 2018)