Predicting Performance for Reading News Online from within a Web Browser Sandbox

by

Murad Kaplan

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

December 2011

APPROVED:

Professor Mark Claypool, Thesis Advisor

Professor Craig Wills, Thesis Reader and Department Head

To my father Mustafa (Rest in Peace), my mother Najah, my siblings Mohammed, Narges, Hussain, Malik, Kamal, and Razan, and finally to my beautiful fiance Sabria who make my WORLD.

Abstract

Measuring Internet performance for home users can provide useful information for improving network performance. Such measurements typically require users to install special software on their machines, a major impediment to use. To overcome this impediment, we designed and implemented several scripting techniques to predict Internet performance within the tightly constrained sandbox environment of a Web browser. Our techniques are integrated into a Web site project called How's My Network that provides performance predictions for common Internet activities, with this thesis concentrating on the performance of online news, social networks, and online shopping. We started our approach by characterizing news sites to understand their structures. After that, we designed models to predict the user's performance for reading news online. We then implement these models using Javascript and evaluate their results. We find out that news sites share common characteristics in their structures with outliers for some. Predicting the page load time according to number objects coming from dominant domain, the one providing the most number of objects, gives more accurate predictions than using total number of objects across all domains. The contributions of this work include the design of new approaches for predicting Web browser performance, and the implementation and evaluation of the effectiveness of our approach to predict Web browser performance.

Acknowledgements

Words fail me in expressing my gratitude to Prof. Mark Claypool. He has been my mentor, colleague and friend all at once. His confidence in me has boosted my morale to no limits. On countless occasions he has gone out of his way to help me with anything and everything. For all these, and more, thank you, Mark.

I would like to thank everyone at PEDS. Their comments, criticisms, suggestions and insights have helped a lot in shaping this thesis. I would like to mention my best colleague at WPI, Mihajlo Zeljkovic , for being there whenever I needed him. Thanks also to all my other friends here at WPI for their support and motivation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As Internet use has grown, so has the importance of understanding network performance for end-users. Network researchers care about network statistics, such as throughput and round-trip time, while end-users care about how effectively their network connection supports the applications they want to use. What is needed are techniques that provide researchers with low-level network data, while providing end-users with effective, easy-to-access methods of understanding their network performance.

Unfortunately, most current measurement tools and platforms have high impediments to use for the typical home user [15]. Dimes [9] and DipZoom [3] provide good data, but require users to install software on their machines to run measurements. PlanetLab [27] and Archipelago [20] are even more flexible in the data that can be gathered, but require participants to permanently contribute to the infrastructure. Tools such Speedtest [1] have low impediments to use by being available through a standard Web browser, but are not designed to inform network research. Other tools such as Gomez [6] go a step further by providing incentives (e.g. money) for use, but are closed in that the data returned is not available to the users or to researchers.

Moreover, current network tools and platforms do not provide incentives for use in that the information returned is not understandable or even relevant to most end-users for the applications they care about. For example, Speedtest provides ping times, download speeds and upload speeds to the nearest server. Most home users are unable to map such low-level performance data to the applications they care about, such as playing a network game, talking in a VoIP session, or reading news online. What is needed are performance tools that are targeted towards the end user, while also providing valuable, low-level data for network researchers.

The How's My Network (HMN) project seeks to provide a measurement platform with low impediments to use, by running within a Web browser without the need of additional installed software, while providing users with incentives in the form of meaningful performance data on the applications users care about. Through several Javascript techniques [33], users' browsers, upon visiting HMN, collect accurate, low-level network data that is mapped to the applications in which they have indicated interest.

This thesis presents the results of making HMN predictions for Web browsing, concentrating on the domain of online news. We focus on online news because reading news online via a Web browser is an increasingly important Internet activity. For example in countries such as Korea, more than half of the population reads news online [8]. Moreover, the number of people who read online news is likely to grow with new generations that get early exposure to the Internet. The Internet is already the main source of news for 16-24 year olds, and about 5 percent of all Internet visits are related to reading news online [11].

We characterize news Web sites and analyze browser download mechanisms to provide a basis for models to predict performance. Our Javascript techniques are used to implement the models, allowing us to compare the predicted performance to actual performance for a range of news sites. Our prediction techniques are applied to social networks and online shopping in order to test the generality of the methods deployed.

Evaluation shows that using the number of objects from the dominant domain, the one providing the most number of objects, improves prediction performance over using the total number of objects. Downloading objects in parallel, as is supported by all modern browsers, is also an improvement over downloading objects serially. About 40% of predictions made based on parallel downloads from the dominant domain have no noticeable error as far as users are concerned, and nearly 90% of all predictions are within one star on a five-star performance scale for news, social networks and shopping.

The rest of this thesis is organized as follows: Chapter 2 gives background and related work. Chapter 3 describes our approach to predict the performance of Web browsing and presents our experiments designed to evaluate the effectiveness of the performance predictions. Chapter 4 presents results and evaluations from the experiments. Chapter 5 shows how news performance prediction methods can be used on shopping and social sites. Finally, Chapter 6 summarizes our conclusions and presents possible future work.

Chapter 2

Background

Resent research has focused on Web application performance measurements. Initially, the focus was on Web characterization and how the design of a Web page can affect the end user's experience. Later, the focus shifted towards network measurement platforms and performance predictors. While these performance measurement tools focus on the server side, they give measurements that do not easily map to user experience. Also, to the best of our knowledge, there has been no research targeted online news performance. This chapter introduces online news and the HMN project and describes some related work in these areas in detail, and discusses how our work differs from each of them.

2.1 Web Characterization

Characterization of the structure of Web pages has been done since almost the beginning of the World Wide Web [28]. The Web Characterization Project has been a significant contributor of research in Web structural analysis [22][25], as the project conducts annual samples of the Web to analyze trends in size and content. Games and Silva presented a characterization of the Portuguese Web [18], with their results derived from a crawl by Tumba, a research and archival engine [13]. Boldi et al. studied the structural properties of the African Web by analyzing HTTP header fields and the contents of HTML pages [13]. Punpiti presented quantitative measurements and analyses of documents hosted under the `.th` domain [29]. Wills et al. performed a large scale Web crawl providing statistics regarding the outcome of download attempts and distribution of types and sizes of documents [21]. This previous work provides a general Web characterization, while our work characterizes news Web sites and uses this characterization to build a model for predicting performance for reading news online.

2.2 Network Measurement Platforms

A network measurement platform is a system of Internet nodes and software designed to gather network performance data. The platform receives and processes events from elements in the network. Events from servers and other critical resources can also be forwarded to a measurement platform [16]. Traditionally, Internet measurement has been done from points in the network infrastructure or from research labs and universities. However, such measurement paradigms increasingly exclude the performance from vantage points of home Internet users. Most current measurement tools and platforms have high impediments to use for the typical home user [15]. Dimes [9] and DipZoom [3] provide useful data, but require users to install software on their machines to run measurements. PlanetLab [27] and Archipelago [20] are flexible in the data that can be gathered, but require participants to permanently contribute to the infrastructure. Tools such Speedtest [1] have low impediments to use by being available through a standard Web browser, but are not designed to inform network research. Other tools such as Gomez [6] go a step further by providing incentives (e.g. money) for use, but are closed in that the data returned is not available to the users or to researchers. Moreover, current network tools and platforms do not provide incentives for use in that the information returned is not understandable or even relevant to most end-users for the applications they care about. For example, Speedtest provides ping times, download speeds and upload speeds to the nearest server. Most home users are unable to map such low-level performance data to the applications they care about, such as playing a network game, talking in a VoIP session, or reading news online. The user perspective is not only important to understand network performance, but also for researchers to track longitudinal trends as the Internet and its applications evolve. It is also important for researchers for next-generation network design and infrastructure building [5]. This thesis focuses on overcoming these challenging and builds part of a measurement platform by developing tools in Javascript, allowing easy access and measurement from users on home networks.

2.3 Performance Predictions

Internet performance prediction is made difficult by the sheer size and heterogeneity of the Internet. Thus, there are many studies for predicting Internet performance [26] [23] [32]. Predicting Internet performance can either be done by exhaustive simulation, which is time consuming and requires a lot of data, or by gathering sample data and building a model. Hamed et al. and Cheung et al. provide studies on Web services and application performance predictions, but neither focuses on simulating the user and instead their models focus on low level network measurements with no attention to user's perspective [19][14]. For Web browsing, our studies [33] showed that different browsers have different behaviors and overhead (e.g. Javascript speed, CSS rendering speed, and page load time). We use careful measurements of these behaviors in predicting user Web browsing performance.

2.4 Online News

In many countries, the Web pages of news broadcasters and newspapers play a large role in attracting news-related visits. More recently, newspaper Web sites have seen strong growth in the popularity of their content, with large newspapers reporting several million visitors to their pages per month, increasingly including readers from abroad, a radical shift from print newspapers [8]. News Web sites have specific challenges from a design perspective in terms of layout, usability and navigation. Regardless of the type of news they cover, news Web sites must display a significant amount of content on the home page. These newspaper Web sites are commonly built with grid-based design, the grid being a popular choice not only because of the sharp look it creates but also because of its effectiveness in managing and organizing a large amount of content [7].

Web browsing performance for reading news is reflected in the Quality of Experience (QoE) for the user. For a given Web content and layout, QoE is determined primarily by how fast the entire page is fully retrieved, and also by how fast the content page loads [24]. Web page size, measured in Kbytes, impacts the total time to download the Web page: the larger the page, the longer it takes to download. When measuring page size, however, there are many components of the page that contribute to that like image sizes, Javascript, applets, and banners [25]. When we design our models, we take into account the unique structure of news sites and how they are retrieved by Web browsers. We also choose the number of objects to download with sizes related to page size in order to give more accurate predictions.

2.5 How's my Network

Predicting Internet performance for home users can provide information to improve network performance. These predictions can also provide Internet applications with information on how effectively these applications are viewed and retrieved by users.

The How's My Network (HMN) project is a system that provides home users with accurate predictions of home network performance. Previous work showed that simple scripting techniques in the Web browser can measure the time needed to download Web objects [21]. The Javascript Document Object Model (DOM) allows scripts to dynamically insert a resource into the current document and receive an event notice when this resource is loaded. We use this feature to measure the download performance for an object from a Web server being tested in our work. Using the DOM method, the HMN project implements tools that predict the behaviors of different Internet applications, collecting data on these applications and interpreting these results to give users a prediction of their home network performance for using these applications. The tools are intended to function within the sandbox environment of a Web browser, enabling broad participation by users without their need to install any additional software. HMN supports several common Internet applications that users may use in their home networks. For each application, HMN provides the user with detailed information of their network performance for the particular application.

When a user goes to the HMN Web page, a list of Internet activities (Web browsing, VoIP, Video conferencing, etc..) are shown. For each activity, users can select a specific application. For example, a user choosing VoIP, can select Skype or Yahoo Messenger. For Web browsing, users can select News sites (e.g. CNN or NY-Times) or Social networks (e.g. Facebook or Twitter). After selection, users can start tests to see network performance for each chosen application. Performance is characterized using five point rating. Two explanations are given regarding application performance, text that includes measurements, such as estimated time to download a page; and low level performance information providing some insight into the way measurements are performed.

Chapter 3

Approach

In order to quickly provide information to users about the suitability of their networks for Web browsing, our goal is to predict the time it will take for a user browser to load a Web page based on a few measurements. A key aspect of this work is to be able to measure the performance of downloading an object from an arbitrary server on the Web. Our approach uses the following steps to provide predictions for page download performance:

- *Characterization:* We characterize news Web sites. We analyze their structures and compare them. We look at the objects types, sizes and from which domains these objects where retrieved. We also analyzed three popular Web browsers and study their mechanisms of rendering and downloading the site objects.

- *Building models:* After characterization, we use what we learned from the sites' structures and the browsers' behaviors to build models that can be used to predict Web browser performance. We built four different models to match the differences observed among the sites and Web browsers.

- *Testing and Evaluation:* We experimentally tested the models we designed with different browsers and different Web sites. We setup environments to control bandwidth and delay and compare the predicted performance times to actual times over a range of controlled network settings.

3.1 Characterization

To characterize news sites, we chose the four most visited news sites in the US, according to D-Lib Magazine [25]:

- CNN (www.cnn.com) (CNN)
- MSNBC (www.msnbc.msn.com) (MSN)
- The New York Times (www.nytimes.com) (NYT)
- The LA Times (www.latimes.com) (LAT)

According to Souders [31], factors that may affect the page load time are: number of objects, sizes of objects and page, and number and location of servers. As depicted in Figure 3.1, we looked at seven pages for each news site (home, sport, travel, health, world, politics, and article), chosen since they are in common to all.

For each site, page data was gathered by completely downloading each page using PageStats [17], a Firefox plug-in that automatically fetches Web pages, recording the URLs and object sizes. The URL provides the domain used to serve the object, including both the original domain and third-party servers that are often used for images or ads. The data was gathered in the time period of February 2011 to March 2011.

Figure 3.2 shows the distribution of object sizes for different levels of the four news sites chosen. The Y axis is the cumulative distribution of the objects, and

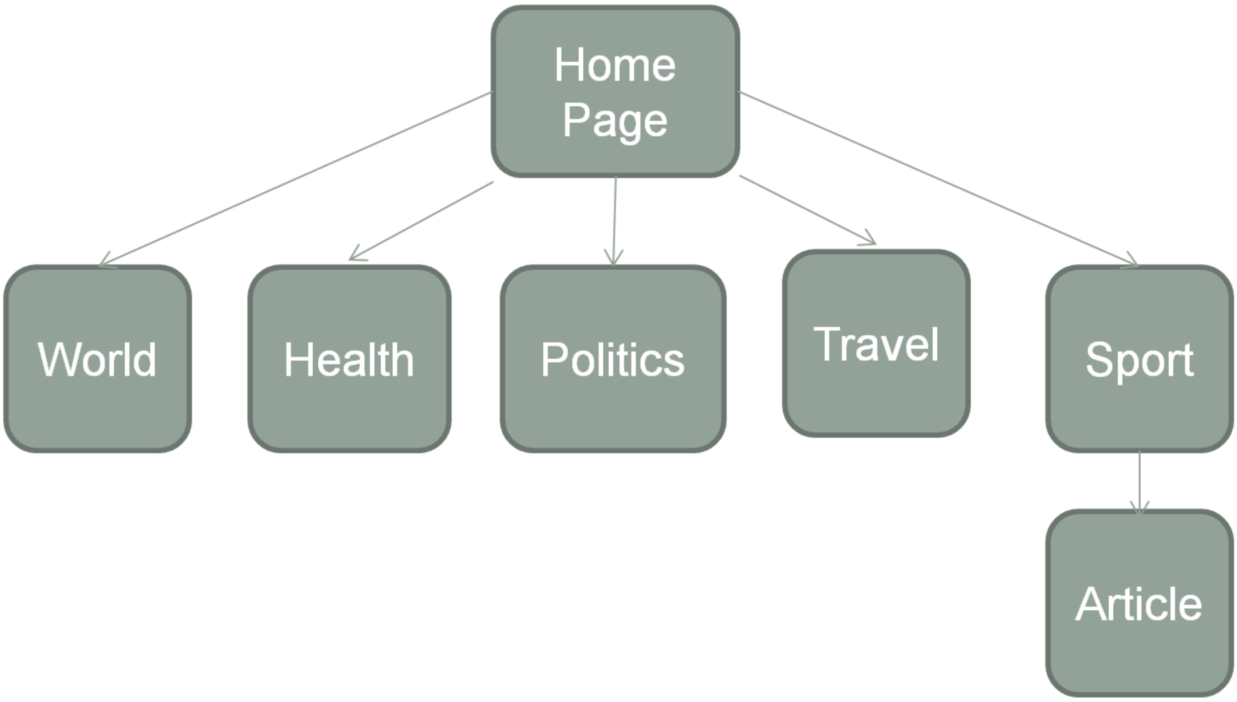


Figure 3.1: News levels of characterization

the X axis is the object sizes. The red line represents all four sites combined. The New York Times and MSNBC have higher median and average object sizes than the other news sites.

Figure 3.3 shows number of objects for each page in the four news sites. Except for the home page, the New York Times and the LA Times have the highest number of objects. On the other side, CNN has the lowest number of objects. Figure 3.4 shows the size for each page of the four news sites. The LA times has the largest page size, except for the politics page, compared with the other news sites.

Figure 3.5 shows the number of domains that each site retrieves objects from for each page. MSNBC has the highest number of domains, in general, among the other news sites. In addition, Figure 3.6 shows, for each page, the largest number of objects coming for this domain compared to the total number of objects in the

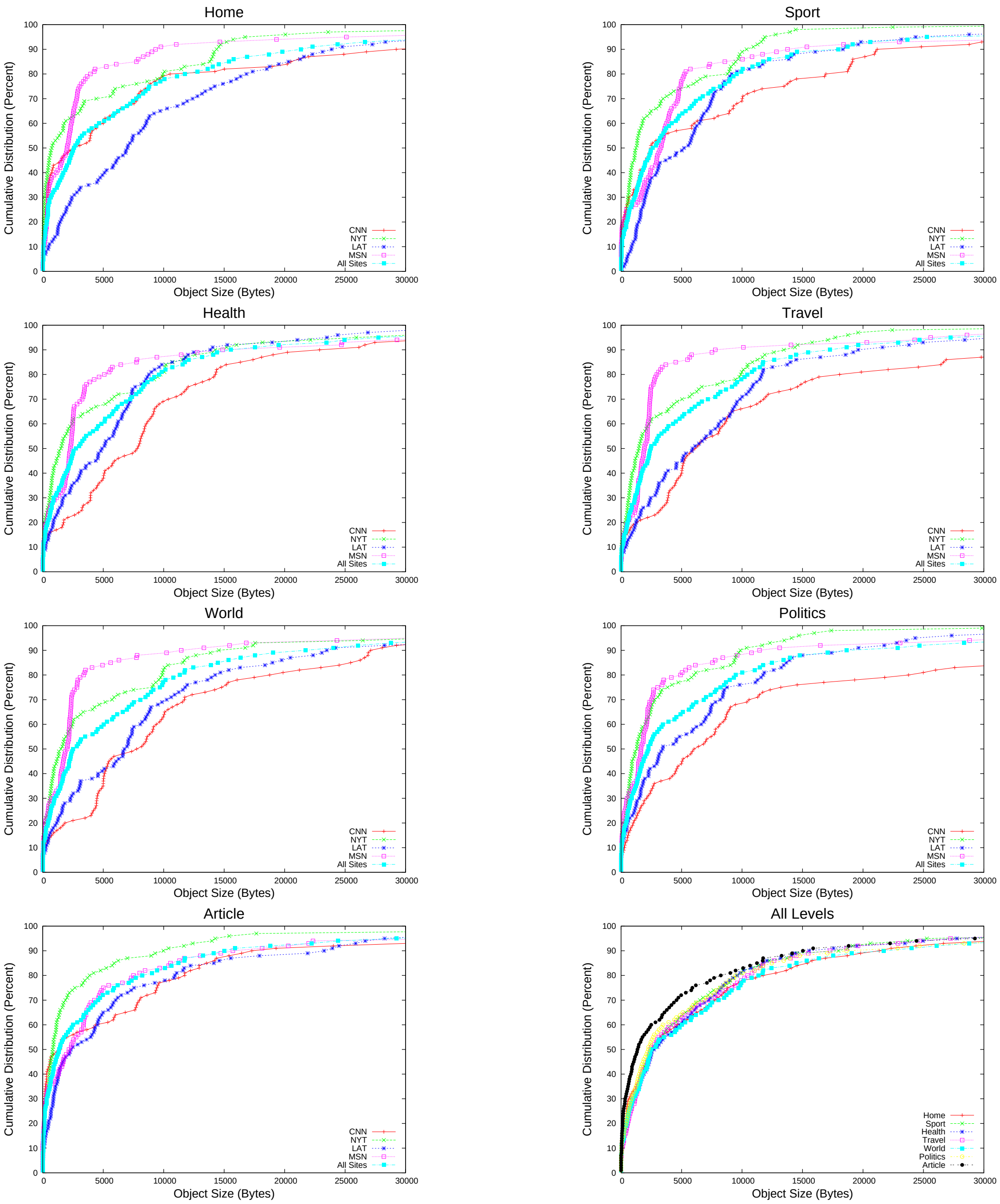


Figure 3.2: Distribution of object sizes for all levels of the four news sites

news site home page. The New York Times has the highest percentage and MSNBC has the lowest.

Figure 3.7 shows the average number of objects and average page size of each page of all news sites. The average number of objects for different pages is around 100

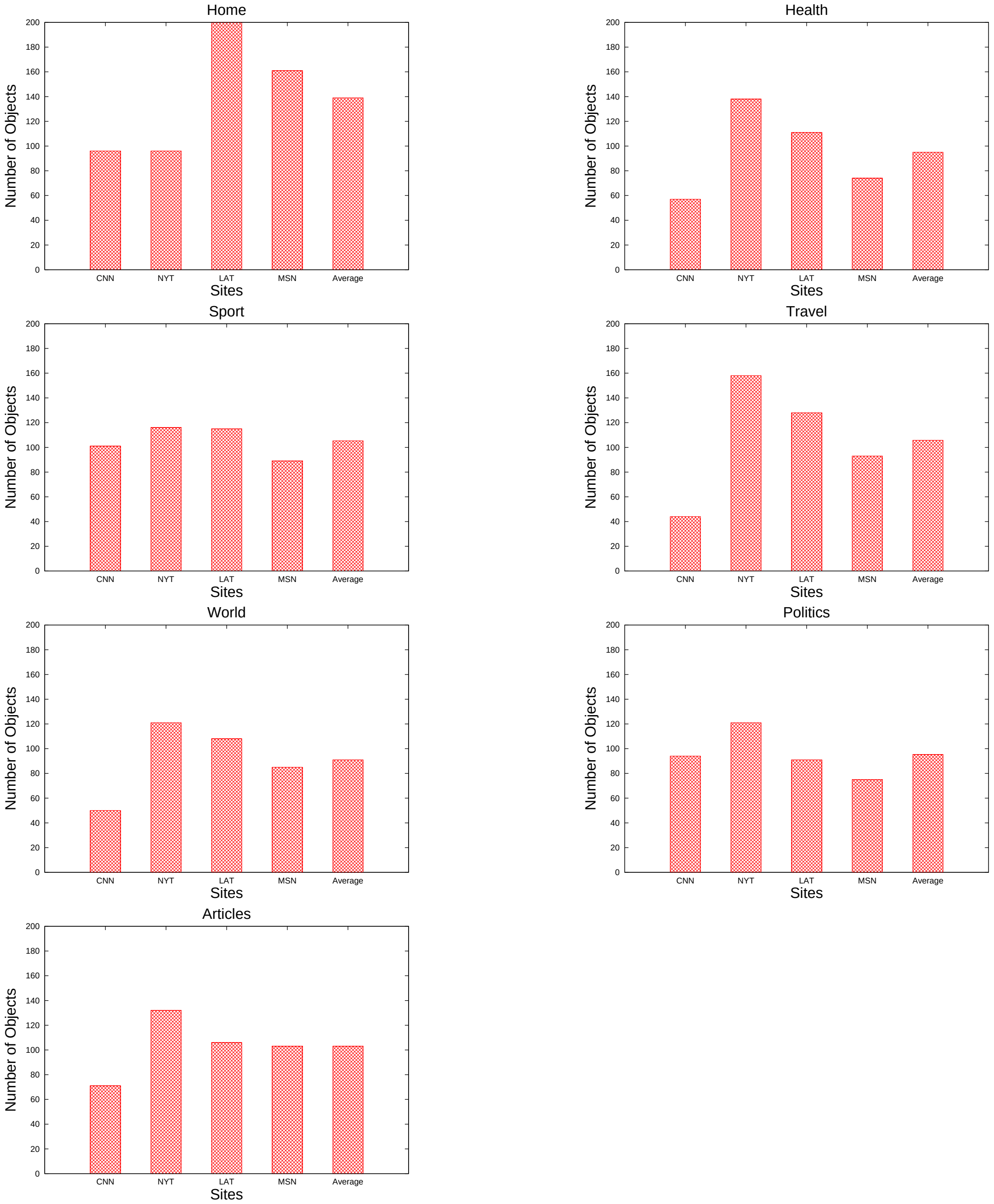


Figure 3.3: Number of objects for all levels for the four news sites

except for the home page. Similarly, the average page size is between 600 Kbytes and 700 Kbytes except the home page. Similarly, Figure 3.7 shows the average number of domains that each news site retrieves its object from for each page. It also shows the highest percentage of total number of objects among the domains

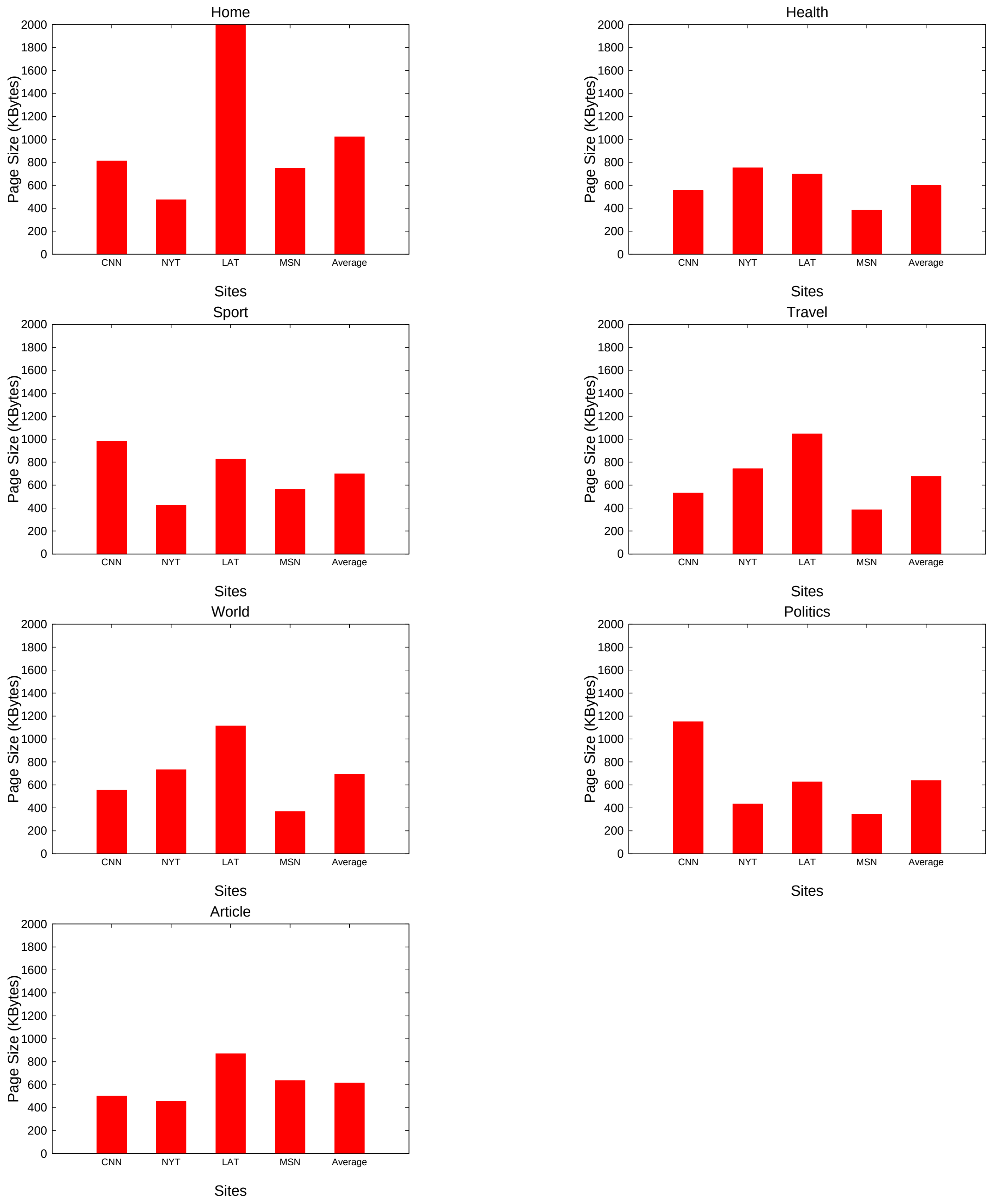


Figure 3.4: Page size for all levels for the four news sites

from which the news site retrieves its objects.

Table 3.1 shows summary results for each news home page, indicated in the first column. The second column is the total number of objects in the page. The next three columns are, in Kbytes, the average object size, median object size, and

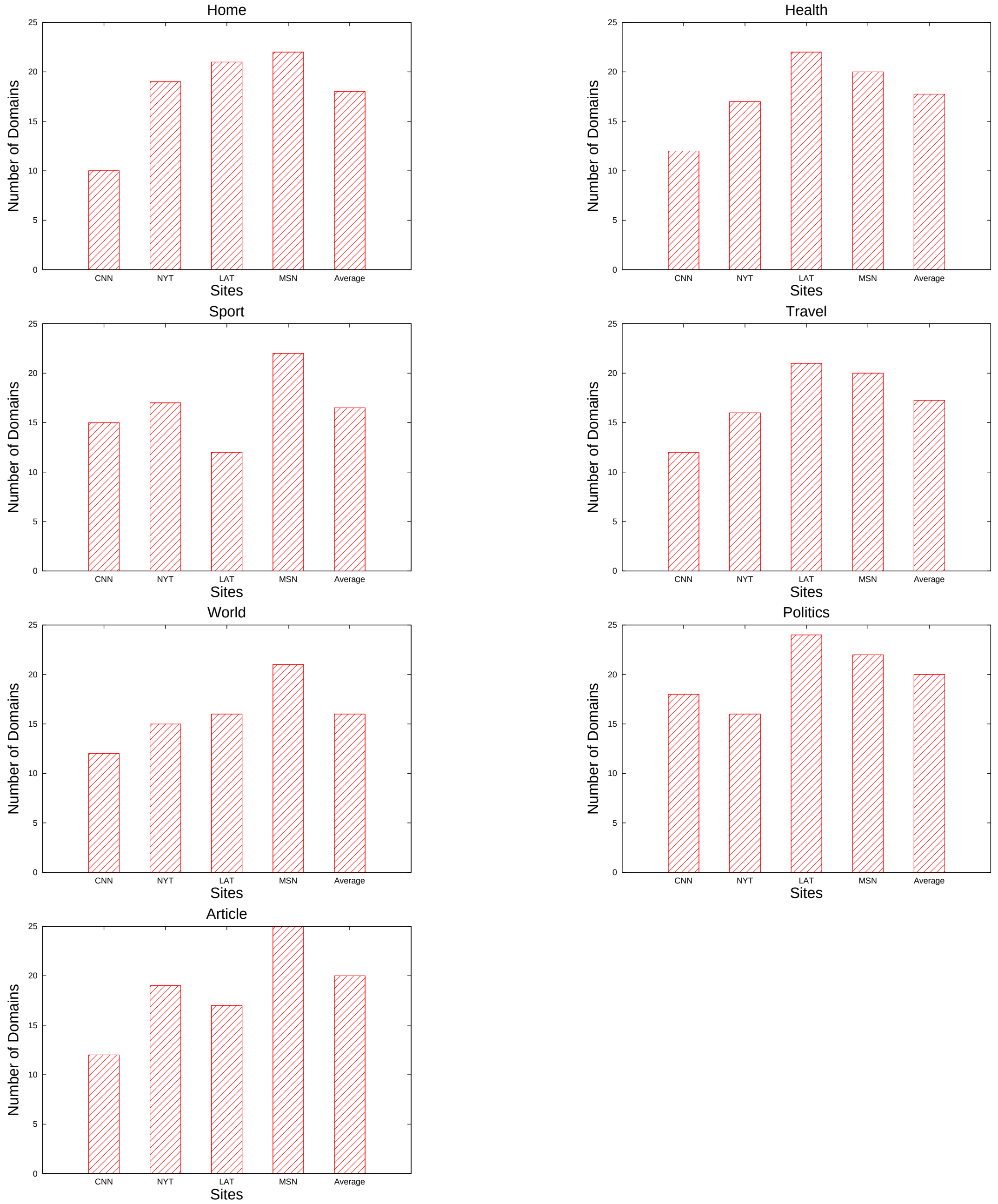


Figure 3.5: Number of different domains each news site retrieves objects from

maximum object size, respectively. The sixth column is the total size for all objects. The last two columns are the number of different domains used to serve the objects and the fraction of those objects that come from a single (or dominant) domain.

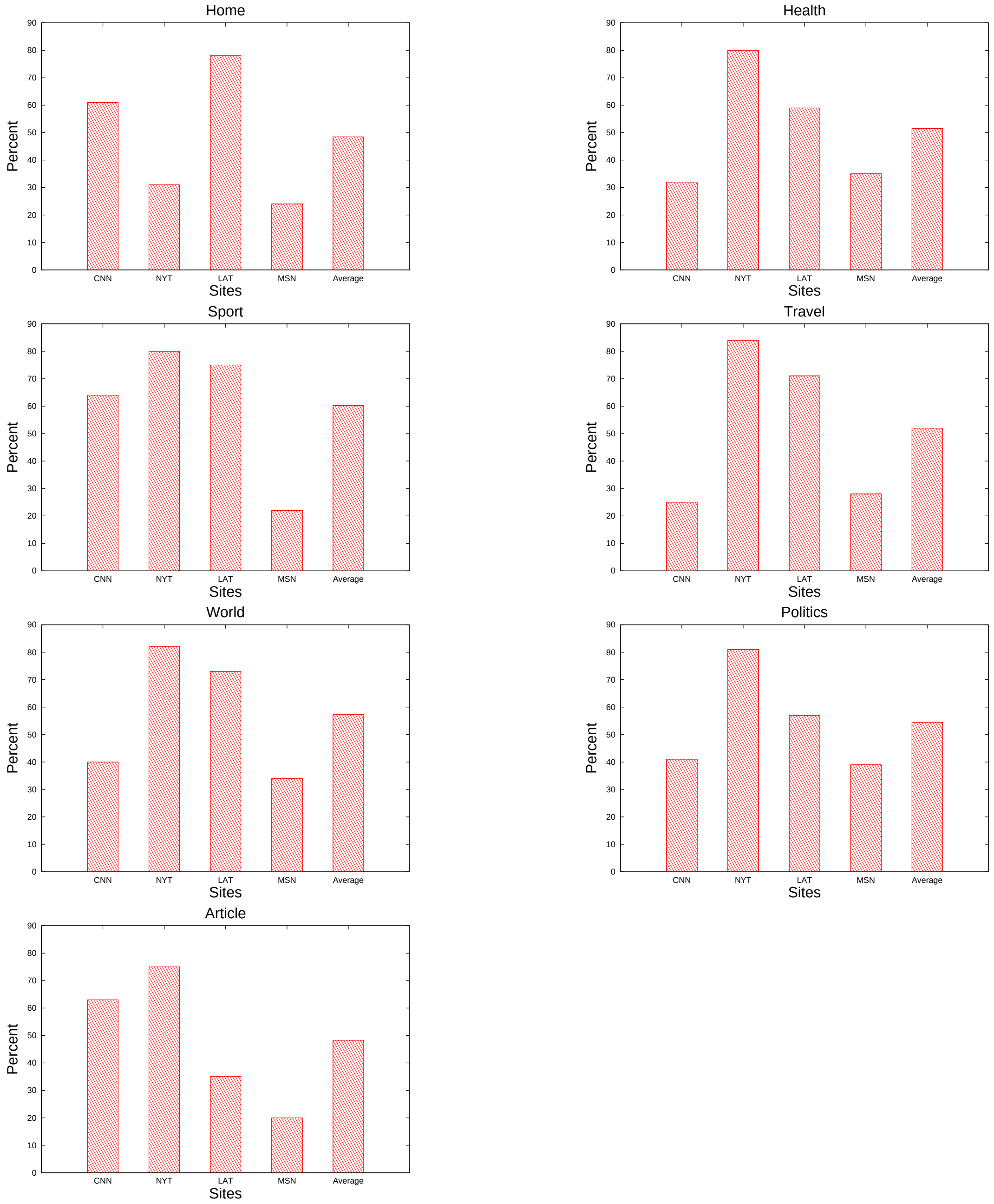


Figure 3.6: Highest percentage of objects come from a single (or dominant) domain

The number of objects varies across news sites, with the fewest (CNN and NYT) having about half those of the most (LAT). This proportion of 2 to 1 holds for the average and maximum sized objects, also, although not necessarily for the same sites. The median sizes vary considerably more, with the smallest median (NYT)

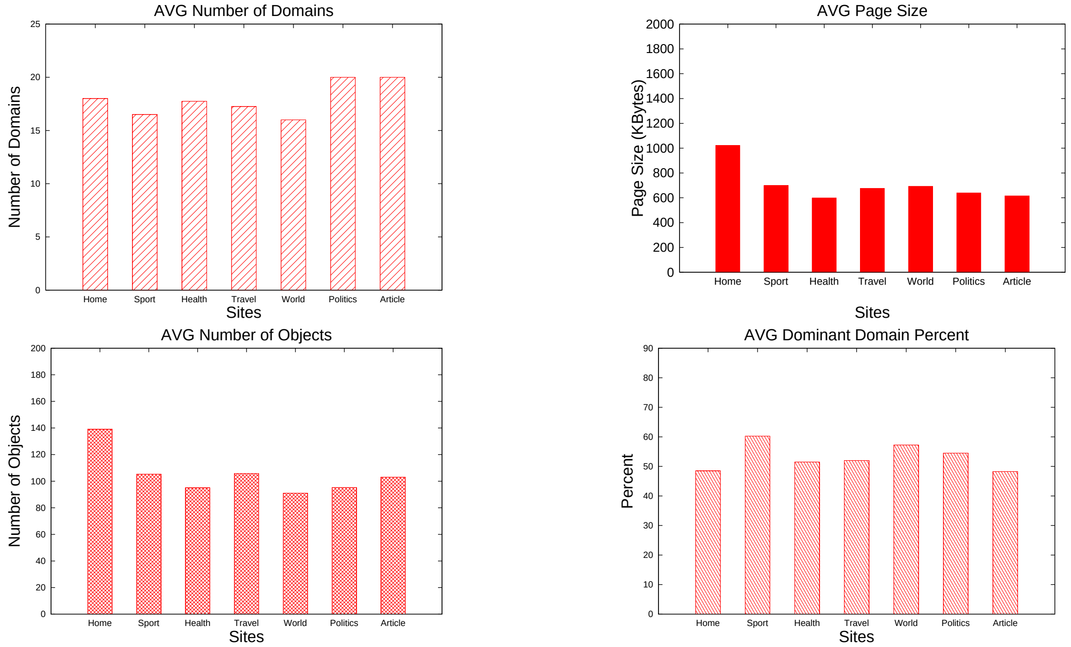


Figure 3.7: Summary results for news home pages characterization

Table 3.1: Characterization of news sites

News Site	Objects (Count)	Container Size (Kbytes)	Avg Size (Kbytes)	Median Size (Kbytes)	Max Size (Kbytes)	Total Size (Kbytes)	Domains (Count)	Dominant Domain (% Objects)
CNN	96	22.3	8.7	2.5	61.4	813.1	10	61%
NYT	96	3	5.1	0.7	54.7	474.5	19	31%
LAT	203	54.8	10.3	7.0	65.7	2049.5	21	78%
MSN	161	47.6	4.8	2.0	90.8	748.9	22	24%
Average	139	32	7.2	3.1	66	1021.5	18	49%

being only one-tenth the size of the largest median (LAT). The total sizes, follow a similar, but less dramatic, difference of about 4 to 1 for the LAT to the NYT. The number of domains varies from 10 to about 20, with about 25% to 75% of the objects being served from a single domain at the most.

We also looked at the browsers behaviors for retrieving news pages. Figure 3.8 shows a screen shot from Fiddler's timeline [4][1] for CNN home page for the three most popular Web browsers according to BrowserScope site[2] (Google Chrome v12, Internet Explorer (IE) v8, and Firefox v4). Fiddler colors objects according to their

[1] Fiddler is a Web debugging proxy used to log all HTTP(S) traffic between client and server.

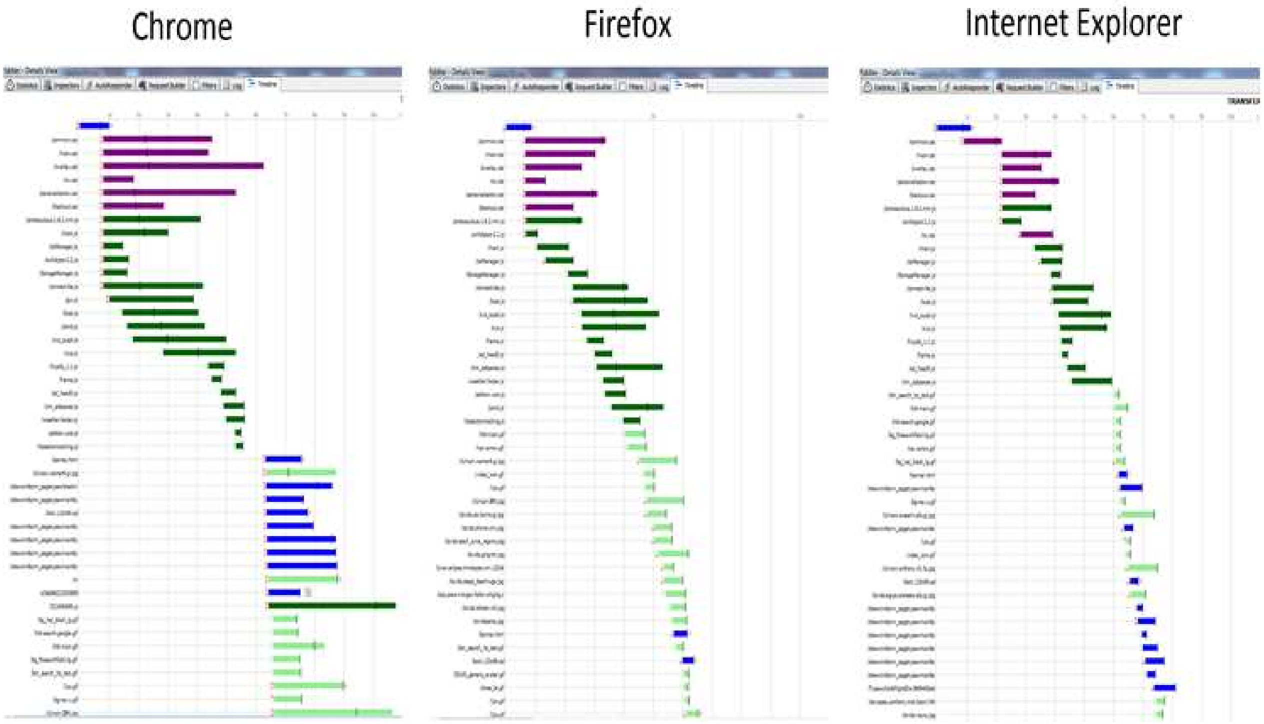


Figure 3.8: Fiddler timeline for Web browsers

types: light-green for images, dark-green for Javascript, purple for style sheet and blue otherwise. The left column shows the object's URL. The second object in the Internet Explorer timeline, a style sheet, has a conditional comment so it blocks the download of other objects in parallel. We can also see that Internet Explorer blocks the download of the images, light green, until all the Javascript files, dark green, have been downloaded.

The order and means by which objects are downloaded depends upon the Web browser used. The mechanisms used by the three Web browsers are provided in Table 3.2. The first column is the maximum number of connections that can be simultaneously opened per host. The second column is the maximum number of connections a browser opens across all hosts. The third column indicates whether or not the browser downloads scripts in parallel with other images in the page.

Table 3.2: Web browsers characteristics

Browser	Connection Per Host	Maximum Connections	DL Script Image	Style Sheet
Chrome	6	35	Yes	Yes
Internet Explorer	6	30	Yes	Yes
Firefox	6	35	No	No

The last column shows if the browser blocks objects from downloading if there is a comment in the style sheet.

All three browsers make up to 6 persistent connections per host, with about 30 connections maximum. Both Firefox and Chrome download objects while downloading scripts and style sheets, while IE blocks other downloads while downloading a script or downloading a style sheet with a comment.

3.2 Prediction Methods

The data on the structure of news sites and the mechanisms browsers use for download form the basis for methods that can be used to predict Web browser performance.

Preliminary investigation with Fiddler showed the container page (i.e. index.html) is always downloaded first, followed by embedded objects, sometimes downloaded serially, sometimes downloaded in parallel. Based on these observations, we explore two primary aspects of Web page retrieval that affect performance: 1) the total number of objects (T) versus the objects from the dominant domain (D), and 2) downloading Web objects serially (S) or in parallel (P).

For each page (home, sport, and article) of each news site we chose two objects from the object list we crawled using pagestats. The first object's size is equal to the container object's size. The second object's size is equal to the average size of all objects in the page. We downloaded each using our Javascript techniques developed for the HMN project [33]. We used the DOM method since it was shown to be the most accurate compared to other methods. In the DOM method, we download the object in the browser's `iframe` and measure the time it takes to be loaded in the browser using the `onload` event as shown in the following example:

```
[language=Javascript]
  var object_load_start = new Date().getTime();
  var URL = ”http://www.reuters.com/images/sprite-core.gif”;
  iframe_avgObject.onload = log(object_load_start, URL);

  function log(object_load_start, file_URL){
    var object_load_end = d.getTime();
    report = ((object_load_end - object_load_start) + ”ms\n”);
    el.getElementsByTagName('textarea')[0].value += report;
  }
```

Combinations of both aspects (T/D and S/P) provide four possible methods that can be used to predict Web browser performance (ST, SD, PT, PD) shown in Table 3.3. In the Formula column, T_c is the time to download the container, T_o is the time to download an average-size object, N_t is the number of total objects, N_d is the number of objects in the dominant domain and p is the number of downloads in parallel (6, as in Table 3.2 for all browsers tested). There is an associated "cost" in using each method to predict performance, defined here as the number of downloads. The serial methods download the container page plus one object while the parallel methods download the container page plus the number of objects downloaded in parallel.

Table 3.3: Methods for predicting performance for Web page download

Method	Formula	Cost
Serial-Total (ST)	$T = T_c + T_o \times \frac{N_t}{1}$	2
Serial-Dominant (SD)	$T = T_c + T_o \times \frac{N_d}{1}$	2
Parallel-Total (PT)	$T = T_c + T_o \times \frac{N_t}{p}$	$1 + p$
Parallel-Dominant (PD)	$T = T_c + T_o \times \frac{N_d}{p}$	$1 + p$

3.3 Experiment Setup

The effectiveness of our methods to predict Web browser performance was measured through a set of experiments as shown in Figure 3.9. A PC was setup to support modern browsers. The PC had a 2.93 GHz Intel Core i7 CPU, with 8 GBytes of RAM, running Windows 7 Service Pack 1 and Web browsers Chrome v12, Internet Explorer v8, and Firefox v4. The PC sent network traffic by way of a Linux PC acting as a router, configured with netem to provide a network environment of a typical residential ADSL link: 1 Mb/s download and 256 Kb/s upload with 100 milliseconds of round-trip time. For testing, our initial four news sites were expanded to the ten most popular news sites in the U.S. [25]: CNN (CNN), the Los Angeles Times (LAT), MSNBC (MSN), the New York Times (NYT), the Washington Post (WPT), ABC News (ABC), Reuters (REU), BBC (BBC), the Huffington Post (HPT), and the USA Today (USA). The tests were run on the weekends from May to June, 2011. The experiments measured the time to retrieve a Web page, from the initial request until the Javascript `onload` event was received. Also, during each run, the objects that represent the `index.html` files and the average size objects were downloaded and timed using the DOM technique described in Section 3.2. For each browser type, each Web site was measured five times, with the browser cache cleared between runs.

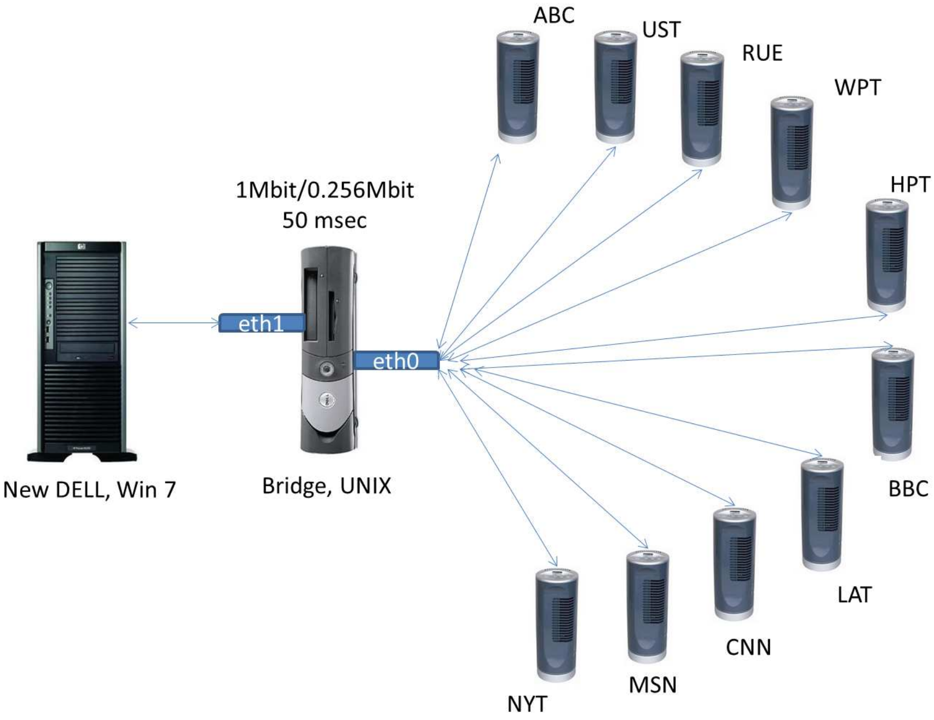


Figure 3.9: Experiment setup

Chapter 4

Results

In this chapter, we evaluate the four prediction methods and present performance data for each level and Web browser. We start by showing the actual time to load a news page in Web browser and then the predicted time for each method compared to the actual time. We then show the error percentage for each method according to a specific Web browser. We also average our results and give a summary at the end of the chapter.

Figure 4.1 shows, for each Web browser, the download time averaged across five measurements for each of the ten news pages across the three levels, with the standard deviation shown by error bars. The time needed to download each page varies across news sites and across browsers, with the largest difference from the New York Times in Chrome to the LA Times in Internet Explorer. These time differences are largely due to the differences in the number and sizes of objects. The times vary somewhat less, but still by as much as by 50%, across browsers to the same news site, with the largest difference to the USA Today for the home page. This performance difference depends on the type of objects the site retrieves and the browser functionality as in Table 3.2. Overall, the home and sport pages have similar page load times across browsers. While the article show that the times vary somewhat more, it reaches 60% across browsers to the same news site as shown in WPT.

To have a better a view of news load times, we show the distribution of times need to load news pages as shown in Figure 4.2. As we noticed in Figure 4.1, home and sport have less variance across browsers and up to 80% page load times are close, about 20 seconds. In articles, however, up to 30% of times are close, about 12 seconds.

Figure 4.3 show the average time difference in seconds for the four prediction methods among the three browsers for each level. The vertical axis is the difference between the measured download time and the predicted download time. Each row represents a news level while each column represents a Web browser. The graphs show that each methods prediction may vary according to the level of the news site, mostly because of the difference in structure between the levels. The graphs show that time differences are higher in Firefox while from the levels perspective; article has the highest outliers exceeding 30 seconds in many sites such as WPI and HPT. Overall, average on (last cluster on the right), D methods outperform the T methods for most runs.

Using the total number of objects tends to overestimate the prediction time as shown in Figure 4.4, while using the number of objects from the dominant domain gives predictions less than the actual load time (AT) for some news sites. For all levels, parallel methods give prediction errors less than serial methods.

Figure 4.5 shows the average time difference in seconds for the four prediction methods among the three browsers for each level. The vertical axis is the average difference between the measured download time and the predicted download time. The horizontal axis has two main clusters, with the left side showing the serial

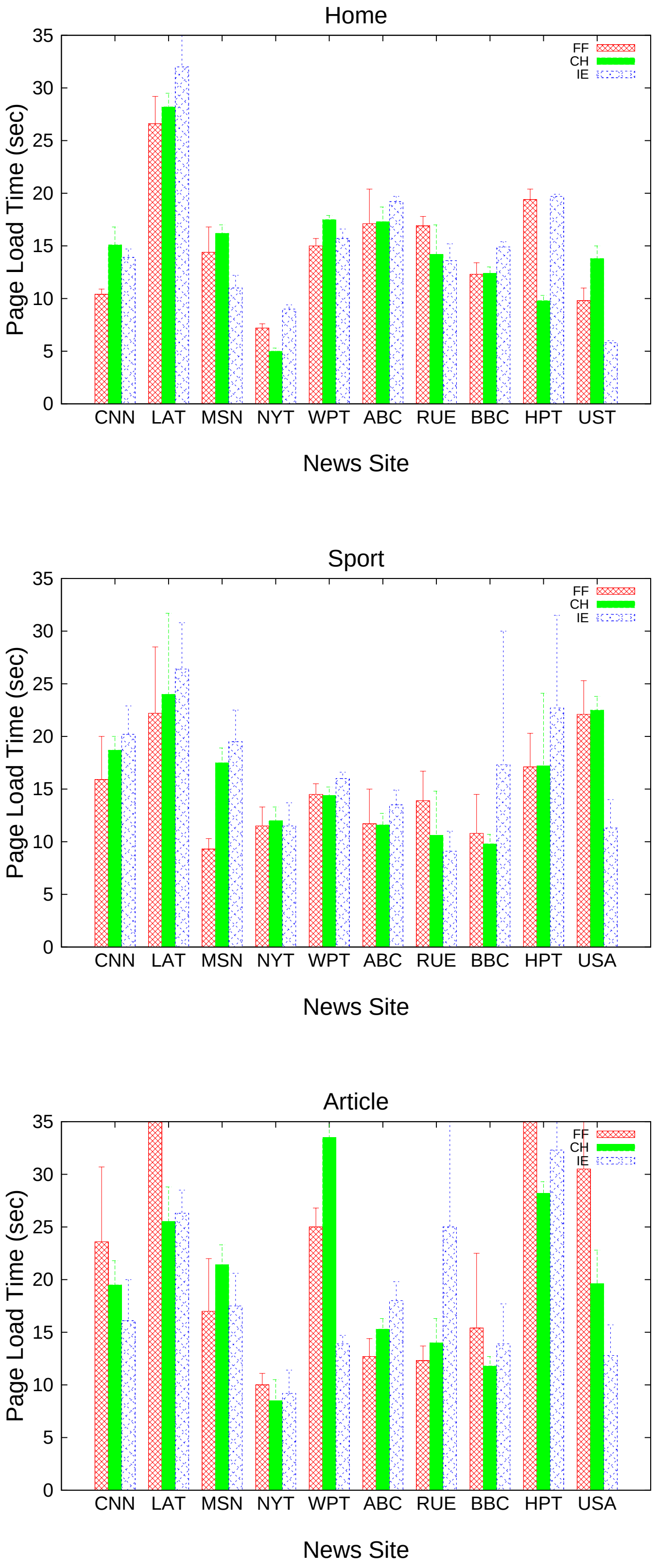
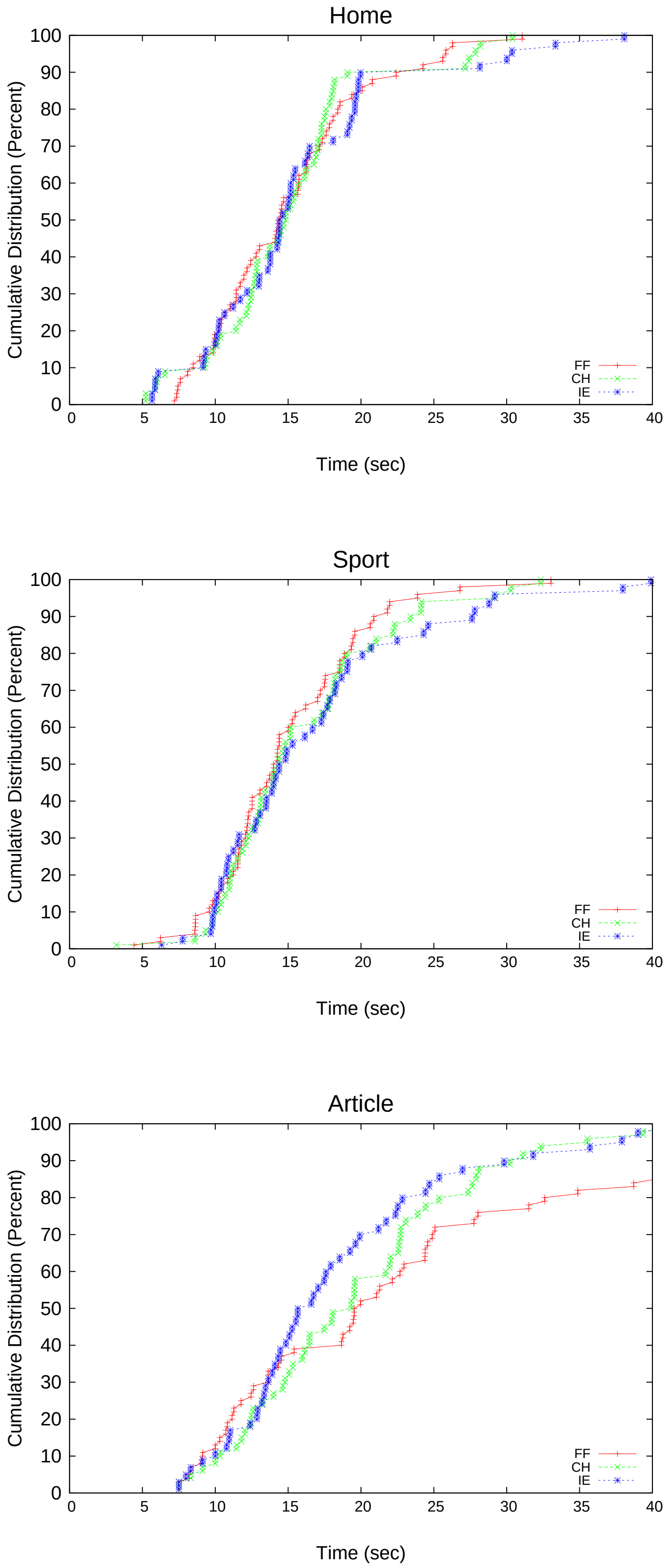


Figure 4.1: Average download times for news sites


Home

Sport

Article

Figure 4.2: Distribution of page download times for news sites

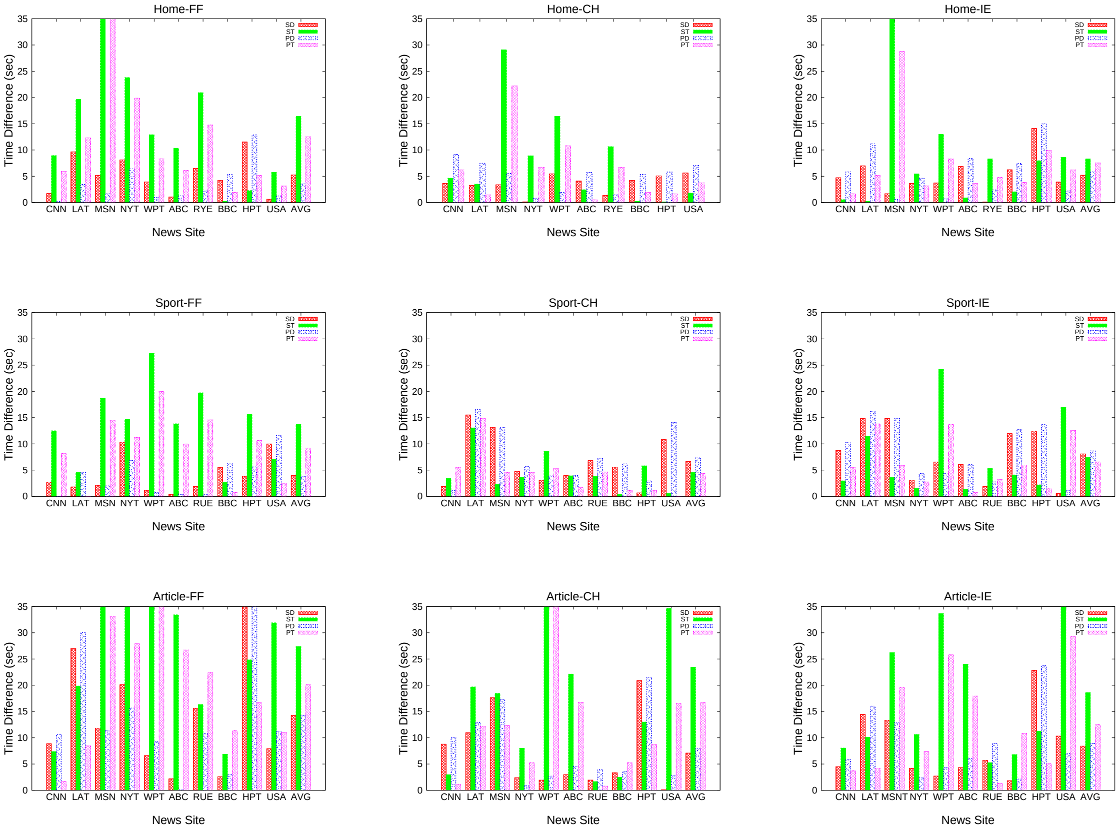


Figure 4.3: Difference in times between predicted and measured for each level and browser

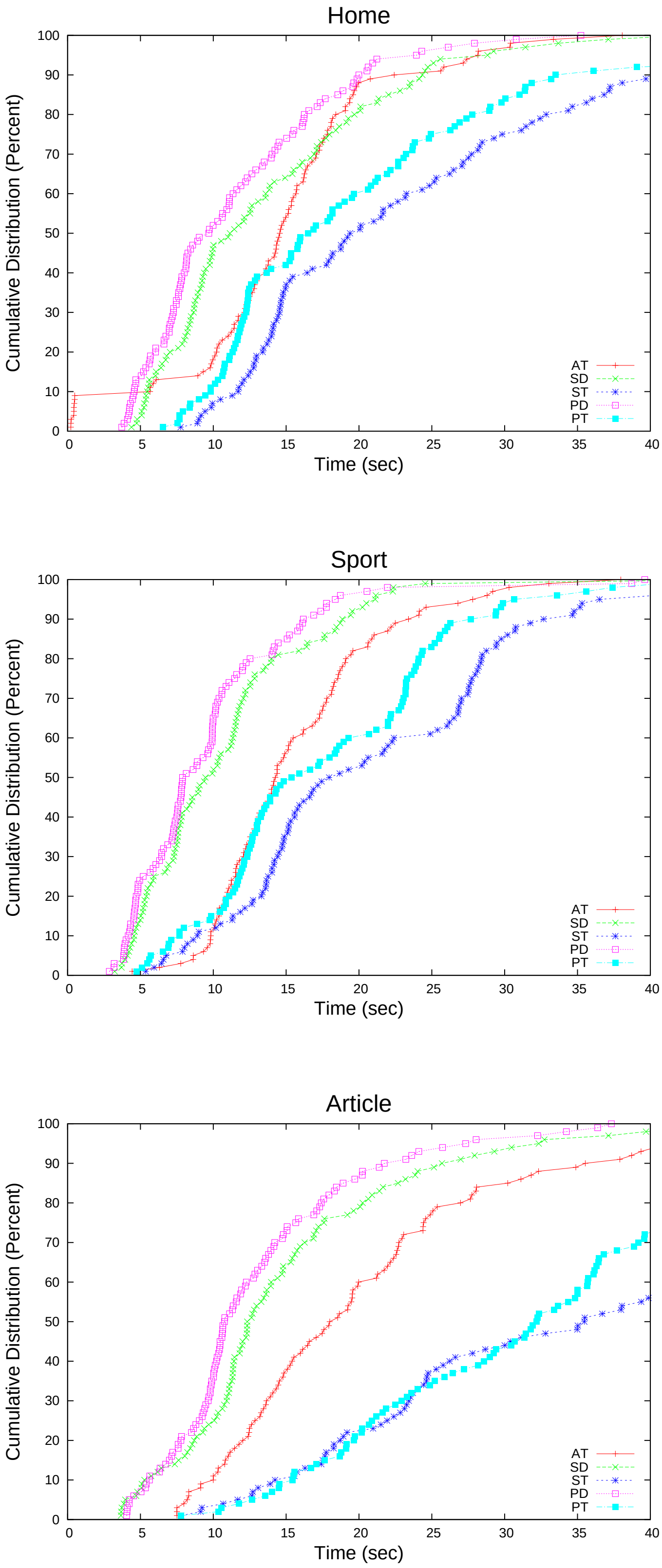


Figure 4.4: Distribution of prediction times and page download times for news pages

predictions and the right side showing the parallel predictions. Within each side, there is a separate cluster for measurements done on Firefox (FF), Chrome (CH) and Internet Explorer (IE). The bars depict the four different prediction methods: serial-dominant (SD), serial-total (ST), parallel-dominant (PD) and parallel-total (PT). For home and article pages, the SD bars are lower (better predictions) than the ST bars and the PD bars are lower than the PT bars for all browsers. These results indicate that using the number of objects served from the dominant domain provides better predictions than using the total number of objects.

We analyzed the container download time compared to the rest of the time calculated in the methods' equations in order to see the container's contribution to the prediction methods. Figure 4.6 shows the distribution of container download time and the rest of the time calculated by each method. For all levels, the container download has a time of 0.5 to 2 seconds. HMN could eliminate downloading the container and have a constant included for all methods, reducing the cost of downloading objects in a user's browser and time needed to complete the run.

While the measured time differences may be of interest for network researchers, the typical user may not notice the impact of an additional few seconds of page load time to their browsing experience. In order to provide incentives for users, we provide performance predictions intended to have more relevance. Both measured download and predicted times are mapped to a 5-point star scale, which is shown to the user in the HMN results page when the tests are complete. The determination of thresholds was informed by previous work on Web site response times [24] [30] where response times within 10 seconds are generally satisfactory while longer response times lead to more dissatisfaction. The mapping of performance to stars is shown in Table 4.1. Stars are only provided as in integer number of stars (e.g. no 1/2 star ratings).

The difference between the download time and the prediction time in stars pro-

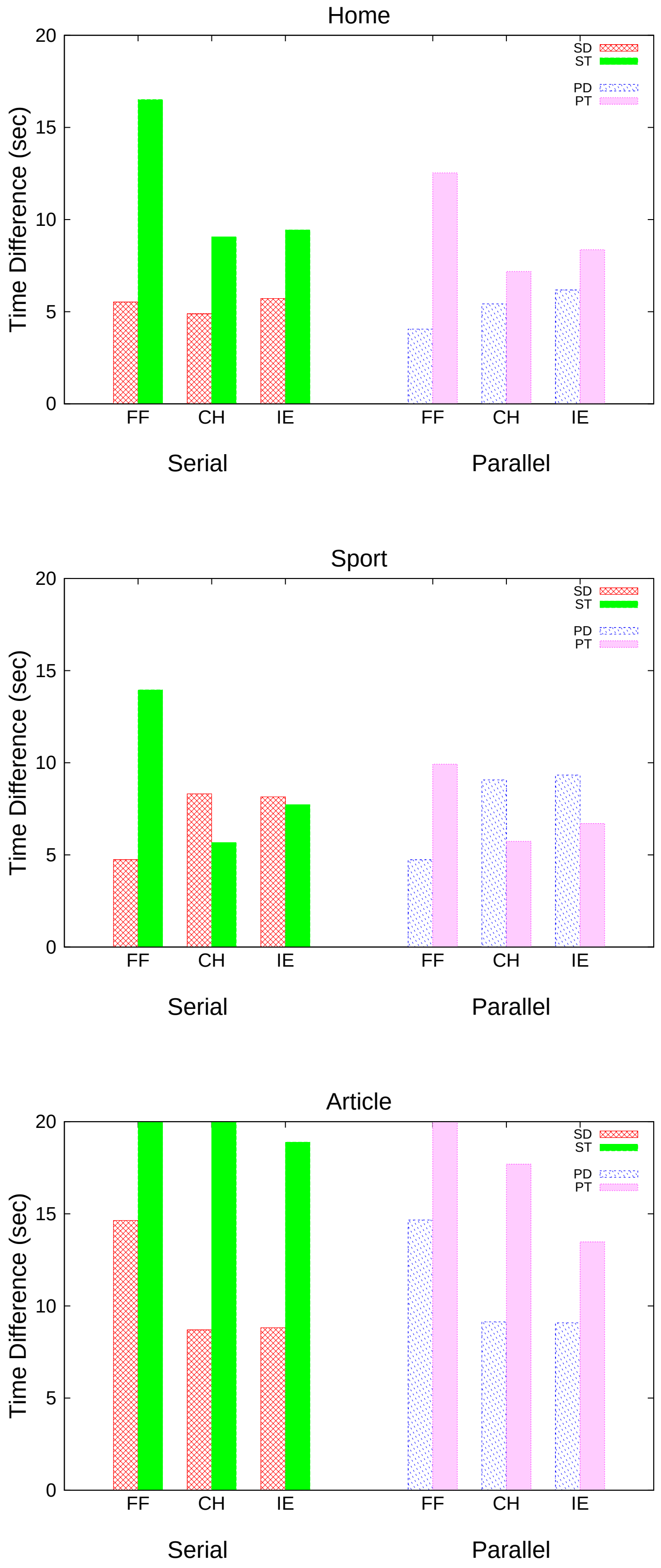


Figure 4.5: Average time difference between predicted and measured for news pages

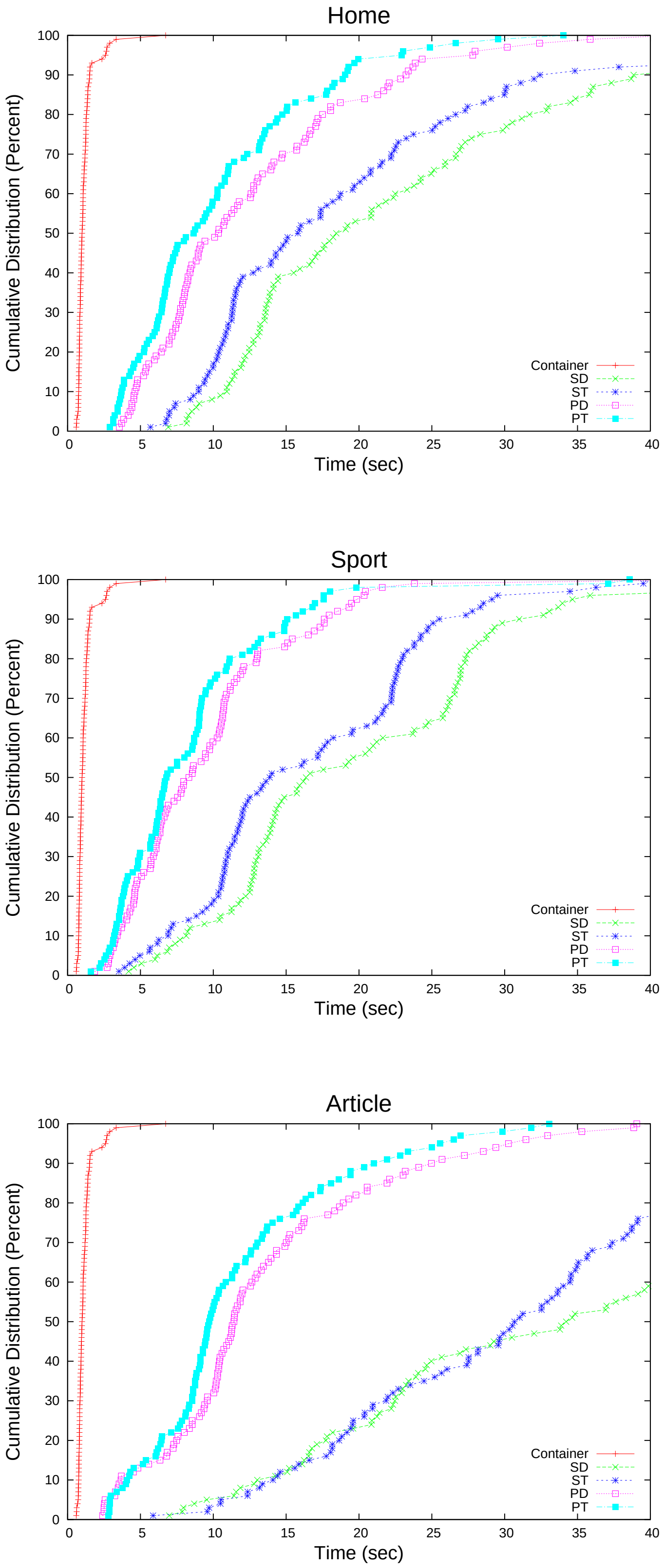


Figure 4.6: Distribution of trendline of container download time and prediction times for news sites

Table 4.1: Stars scale

Time (seconds):	[0-5]	(5-10]	(10-15]	(15-20]	(20+
Stars:	★★★★★	★★★★	★★★	★★	★

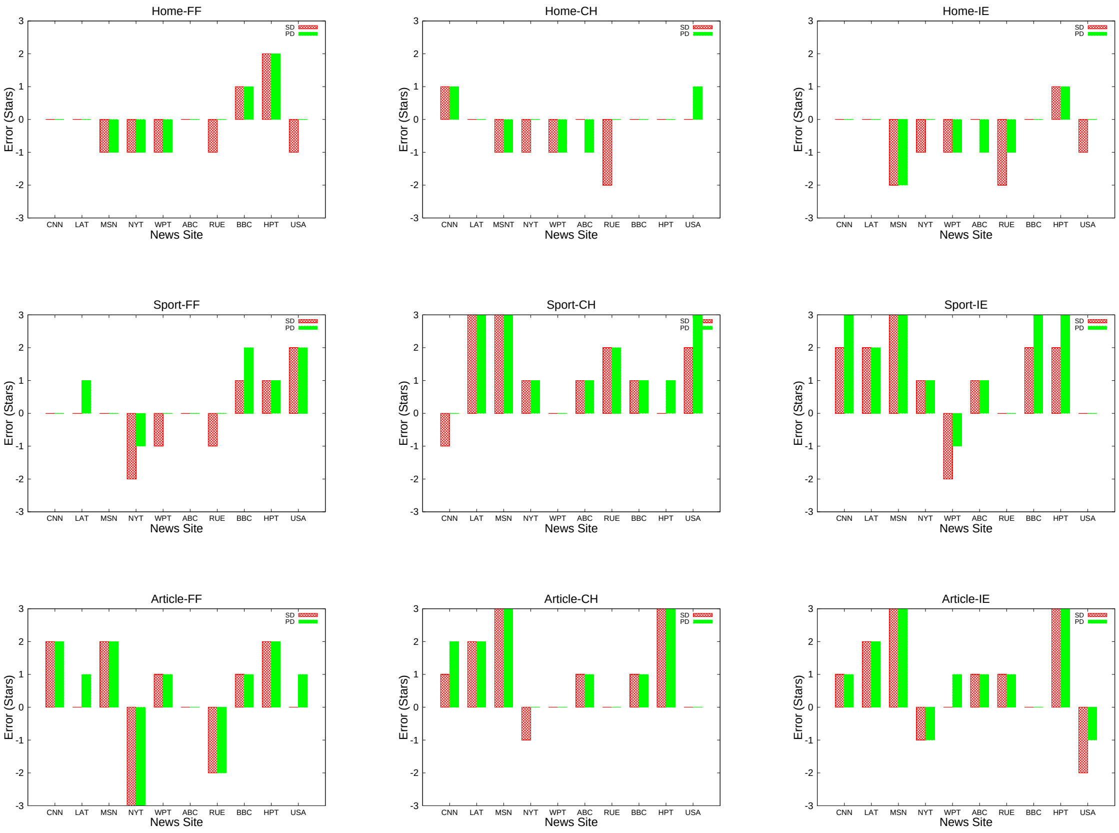


Figure 4.7: Prediction error for all levels of news sites across browsers

vides a prediction error. Figure 4.7 shows the error in stars for each level for each news Web site. The horizontal axis indicates the site while the vertical axis indicates the error, in stars. Errors less than zero indicate the prediction time was greater than the measured time. Each row represents a news level while each column represents a Web browser. Since using the number of objects from the dominant domain gives a better prediction than using the total number, for all subsequent graphs, only analysis of predictions using the number of objects in the dominant domain is shown.

It can be seen that our methods give the best prediction at the home level. In many cases, for home pages (CNN, LAT and ABC) both SD and PD provide accurate predictions that effectively predict browser performance with no error. In most other cases, the error is only 1 star and the only 2 star error is for HPT. Comparing PD to SD, in all cases PD does as well as SD, except for RUE and USA where SD is worse since it overestimates the download times in Firefox and Internet Explorer. For articles and sport pages, the methods effectively predict browser performance with no error, but we get higher error rates that reach 3 star error in some pages such as LAT and MSN.

To evaluate the effectiveness of the PD predictions compared to the SD predictions, Figure 4.8 shows the cumulative distribution functions (CDFs) of prediction errors for all news sites and browsers. The horizontal axis is the prediction error (in stars) and the vertical axis is the cumulative distribution. There are two trendlines, one for SD and one for PD.

The graphs indicate that for home, PD provides accurate predictions just over 40% of the time, while SD is somewhat worse at just under 30%. At the right of the distribution, for about 3% of the predictions, PD has nearly 3 stars in error, compared to only 0.5% for SD. The case for sport and article is different as the trendline shifts to the left. We believe this shift is related to the different structures of sport and article pages as shown in Section 3.1. We leave the analysis of this shift as future work that could focus in particular on sections and articles in news sites.

We also analyzed the effectiveness of the PD predictions across browsers and levels as shown in Figure 4.9. PD performs the best at the home level with 45% zero star error predictions. It also gives up to 43% of zero star error in Firefox while having the worst predictions in Chrome with about only 10% zero star error.

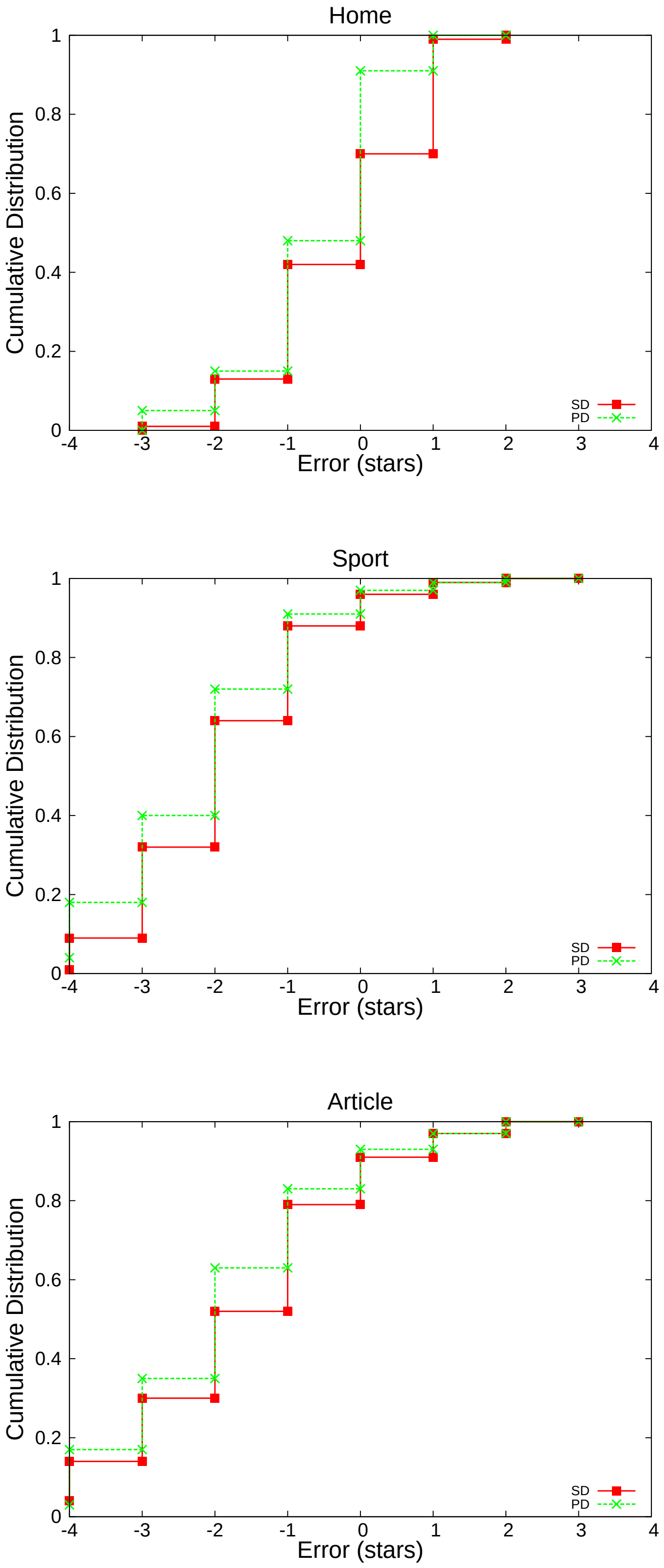


Figure 4.8: CDF of prediction errors for news sites across browsers

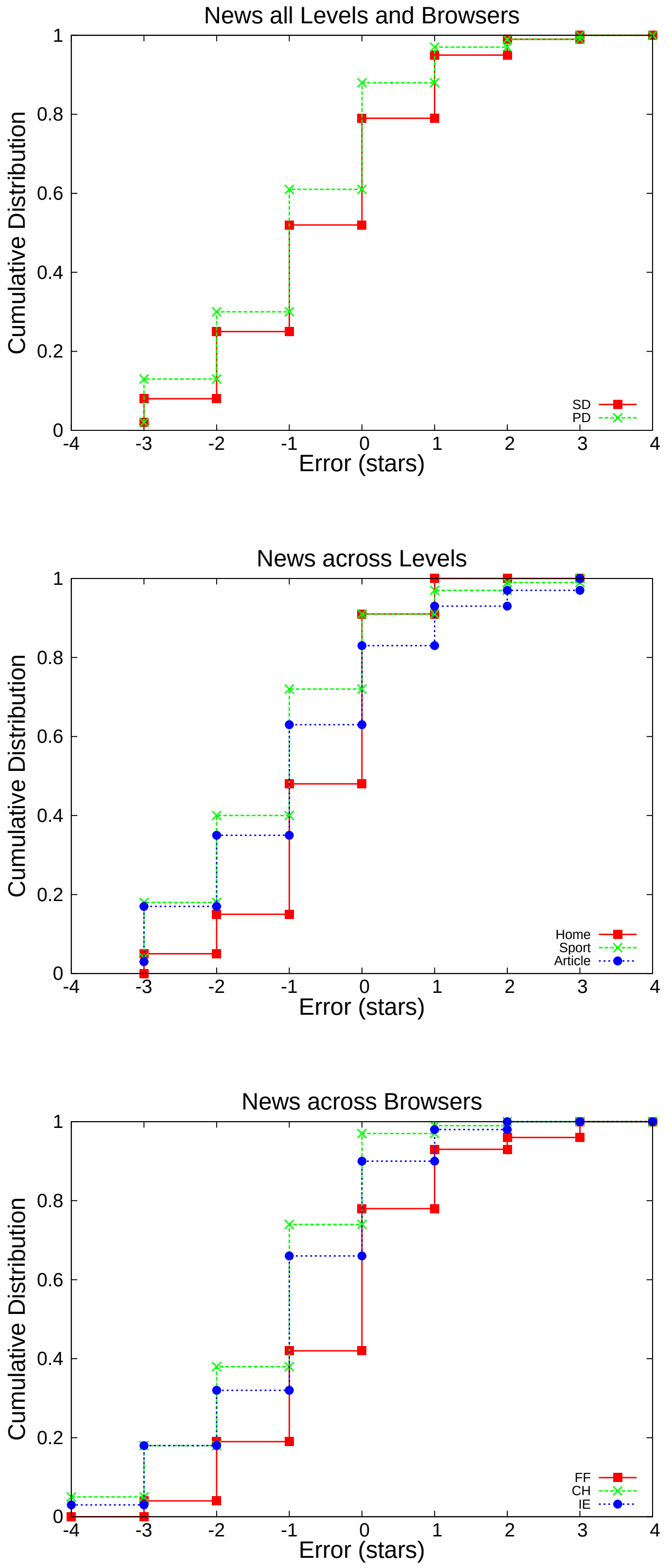


Figure 4.9: CDF of prediction errors for news sites

Chapter 5

Shopping and Social Sites

After vetting our prediction methods with online news, we applied our approach to social networks and online shopping. The five most popular sites were chosen for each category [12] [10] for characterization and evaluation. For shopping: Amazon (AZ), Ebay (EB), Bizrate (BZ), Giftag (GT), and SmartBargains (SB). For social networks: Facebook (FB), Twitter (TW), Myspace (MS), Friendster (FR), and Linkedin (LD). The analysis of the comparative distribution for PD and SD across all browsers is provided.

Figure 5.1 shows our characterization results for shopping sites. The page size is relatively small to news sites, about 200 Kbytes, with a maximum page size of 889 Kbytes for Amazon. While Giftag has the smallest page size of 215.8 Kbytes, it has the highest percentage of dominant domain objects with 93% and less number of domain which is only 4 domains. Table 5.1 shows a summary of the shopping characterization results. We can see that although SmartBargains has the highest number of objects, it has the smallest container page, average object, median object and maximum object sizes.

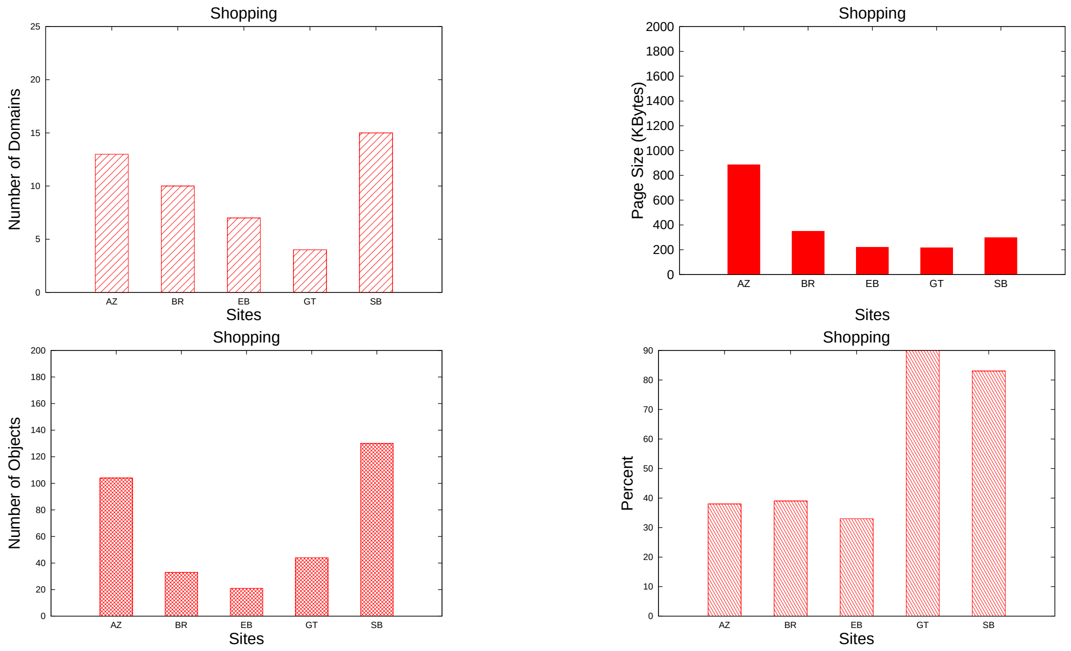


Figure 5.1: Summary results for shopping pages characterization

Table 5.1: Characterization of shopping sites

Shopping Site	Objects (Count)	Container Size (Kbytes)	Avg Size (Kbytes)	Median Size (Kbytes)	Max Size (Kbytes)	Total Size (Kbytes)	Domains (Count)	Dominant Domain (% Objects)
AZ	104	17.1	8.4	2.8	51.9	884.9	13	38%
BR	33	12.7	10.5	4.5	87	347.2	10	39%
EB	21	20.1	9.9	2.9	55.7	218.2	7	33%
GT	44	3.3	4.9	2.4	47.3	215.8	4	93%
SB	130	0.15	2.3	0.5	38.6	298	15	83%
Average	66.4	10.67	7.2	2.6	56.1	392.8	9.8	57%

We repeated the characterization with social sites as shown in Figure 5.2 and Table 5.2 . Social sites have close page size, about 400 Kbytes, compared to shopping and news sites. Also social page sizes are about double that of shopping sites except Amazon.

Figure 5.3 shows the error in stars for each level for each news Web site. The left column shows shopping in three browsers and the right columns shows social in three browsers. The methods give better predictions for shopping and social sites than for news with the worst prediction of two stars for both methods in social in Chrome. The two methods give accurate prediction for Amazon and Ebay at all

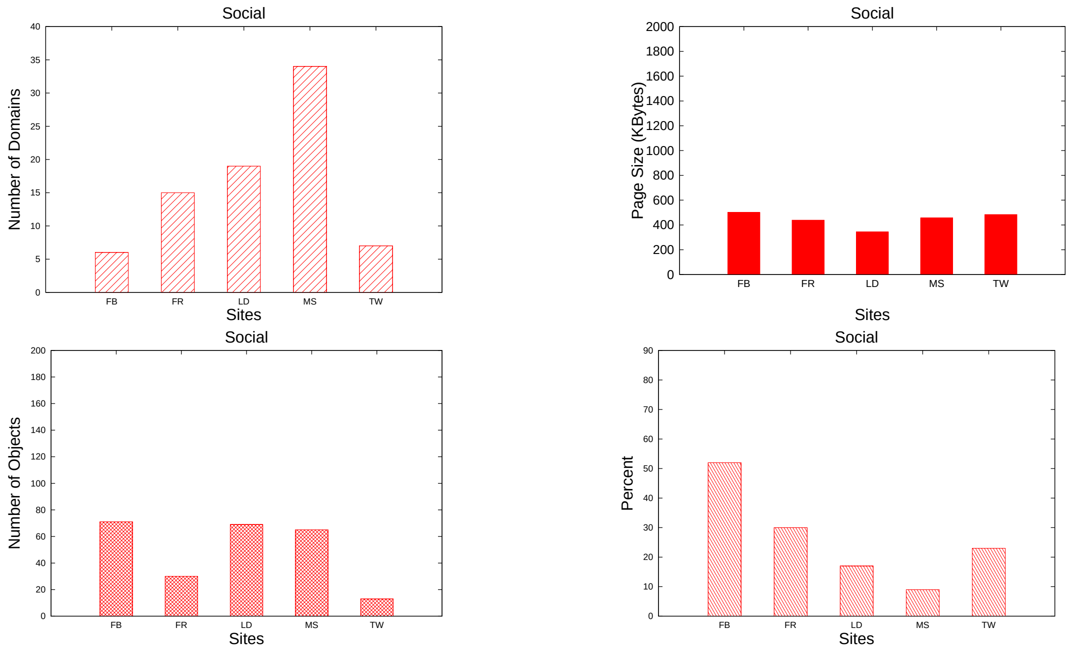


Figure 5.2: Summary results for social pages characterization

Table 5.2: Characterization of social sites

Social Site	Objects (Count)	Container Size (Kbytes)	Avg Size (Kbytes)	Median Size (Kbytes)	Max Size (Kbytes)	Total Size (Kbytes)	Domains (Count)	Dominant Domain (% Objects)
FB	71	7.1	4.6	2.7	15.9	500.5	6	52%
FR	30	0.2	14.6	5.2	159.7	438.7	15	30%
LD	69	2.7	4.9	1.8	83.9	343.7	19	17%
MS	65	22.1	7.0	0.6	112.0	456.4	34	9%
TW	13	10.1	37.2	10.1	303.9	484.0	7	23%
Average	49.6	8.44	13.7	4.8	135.1	444.7	16.2	26.2%

levels and browsers except Amazon in shopping using Chrome.

Figure 5.4 shows the CDFs of prediction errors for shopping and social sites for all three browsers, with the two trendlines representing PD and SD for the first two graphs. And three trendlines represent online news, social networks and online shopping in the lower graph. For shopping, PD shows an 80% accurate prediction while in social, PD has s 60% accurate prediction. In the last graph, PD is about as effective for social networking as it is for news, while PD provides even better predictions for online shopping. For online shopping, about 65% of the predictions are effectively perfect and no predictions are worse than 2 stars in error.

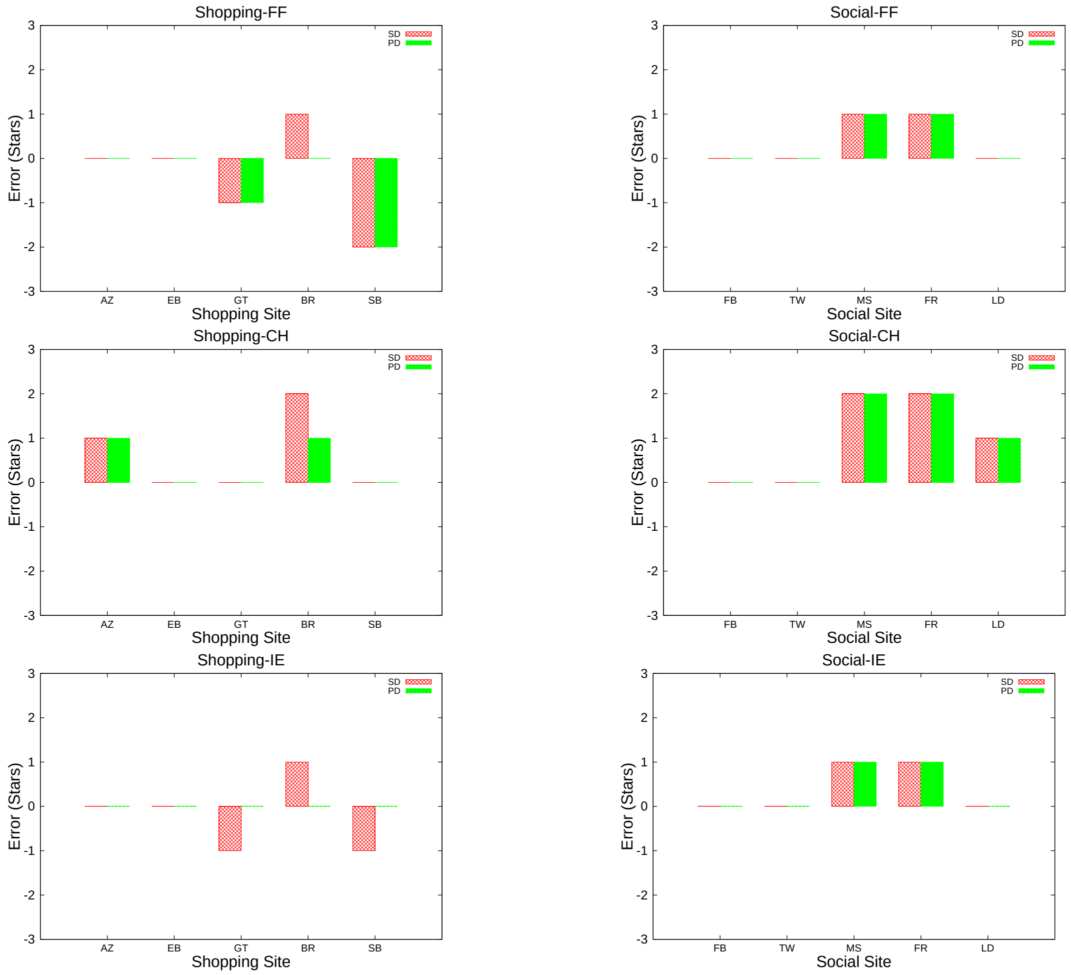


Figure 5.3: Prediction errors for shopping and social sites across browsers

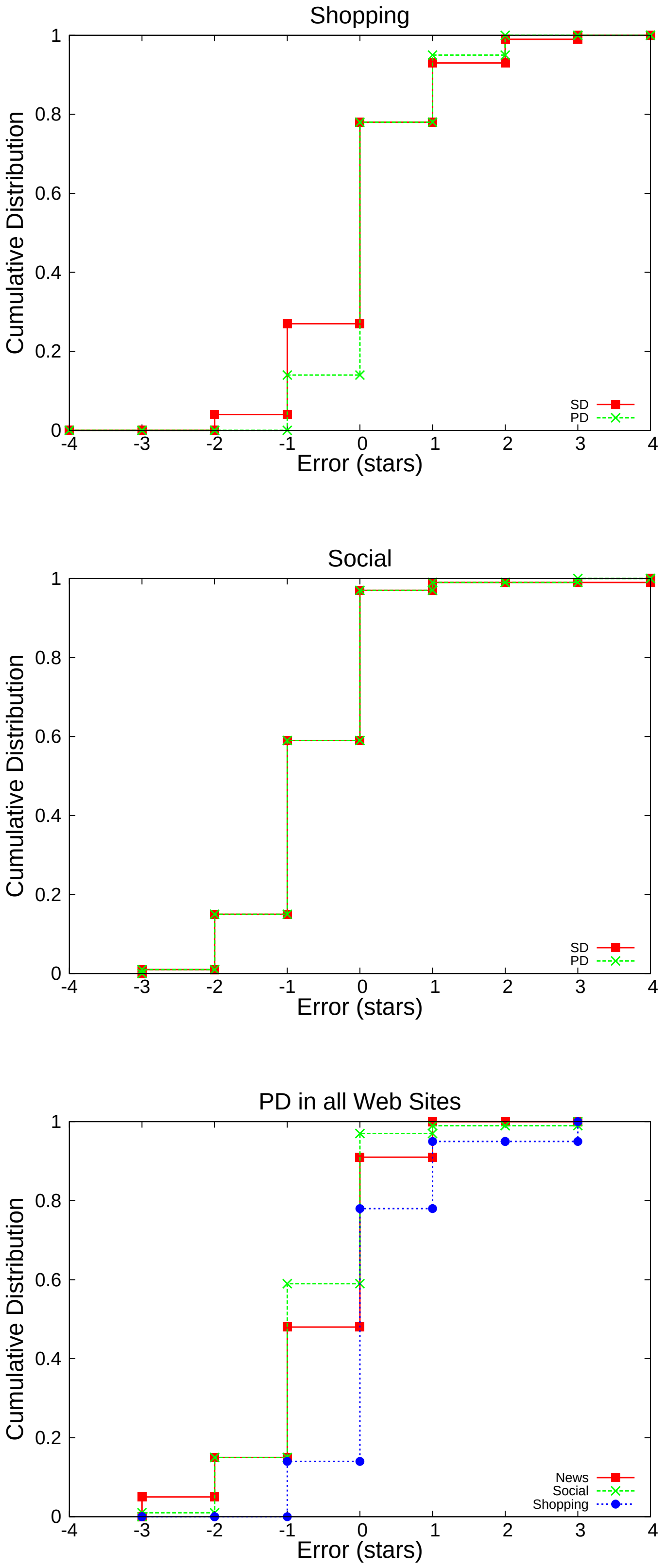


Figure 5.4: CDF of prediction errors for shopping and news across browsers

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Platforms that provide network measurements are needed by researchers to better understand and improve networks. End users can provide valuable network data if there are low impediments to participating in measurements studies and if there are incentives that encourage them to provide data.

The How's My Network (HMN) measurement project seeks to provide low impediments to user participation by enabling end-host measurements from within the constrained sandbox of a Web browser, thereby not needing the user to install any additional software. Incentives for users are provided in the form of meaningful feedback data for the applications they care about, such as predicting the number of stars a network can provide in support of Web browsing using a small number of object downloads.

This thesis presents an approach for predicting the performance for Web browsing from within a browser sandbox. The approach analyzes the structure of Web sites and download techniques for a Web browser, then proposes three methods for predicting performance based on the number of objects and method of download.

Detailed experimental results for online news find that using the number of objects from the dominant domain is better than using the total number of objects, being over twice as effective for some Web browsers. Assuming objects download in parallel rather than serially provides generally better predictions, with about 15% more accurate predictions for online news. Applying our methods to social networks and online shopping shows our approach works for other media too, with nearly half of all predictions having effectively no error.

6.2 Future Work

Building upon our methods to predict the performance of news, shopping, and social sites, our future work includes extensions to other Web sites such as online banking and blogs. We would first characterize the Web sites, follows by applying our models for performance prediction. This would help design better performance predication for these Web sites and the applications they include, providing results for typical home users and network researchers at the same time.

We also would like to develop our methods to include other Web site characteristics such as object types and to target Multimedia in online news. This will give Web designers a better understanding of how their Web sites are retrieved by the end users and what characteristics should be optimized to make Web sites lighter and faster.

Finally, we would like to continue the development of the HMN project to provide predictions of the performance for other Internet applications such as VoIP, online games, and video streaming. Additional metrics may include QoS, packet lose, and round trip-time. In all cases, the approach is to take measurements from within the browser sandbox, while providing incentives for participation in the form of meaningful feedback to the applications users care about.

Bibliography

[1] Broadband reports.com Speed Test. http://www.dslreports.com/stest/.

[2] BrowserScope. http://www.browserscope.org.

[3] Deep Internet Performance Zoom. http://dipzoom.case.edu/.

[4] Fiddler, Web Debugging Proxy . http://www.fiddler2.com/.

[5] Global Environment for Network Innovations. http://www.geni.net/.

[6] Gomez Peer Community. http://www.porivo.com/.

[7] Newspaper Website Design. http://www.ejordanweb.com/index.php/.

[8] OECD Examines the Future of News and the Internet. http://tinyurl.com/3n26e8j.

[9] The DIMES Project. http://www.netdimes.org/.

[10] Top 15 Most Popular Comparison Shopping Websites. http://www.ebizmba.com/articles/shopping-websites.

[11] What is Web Journalism. http://journalism.about.com/od/trends/a/whatwebjour.htm/.

[12] Social Networking Websites Review, 2011. http://social-networking-websites-review.toptenreviews.com.

[13] P. Boldi, B. Coldenotti, M. Santini, and S. Vigna. Structural Properties of the African Web. In *Proceedings of the 11th International World Wide Web Conference*, Honolulu, Hawaii, 2002.

[14] L. Cheung, L Golubchik, and F Sha. A Study of Web Services Performance Prediction: A Client's Perspective. In *Proceedings of the 2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, Washington, DC, USA, 2011.

[15] KC. Claffy, M. Crovella, T. Friedman, C. Shannon, and N. Spring. Community-Oriented Network Measurement Infrastructure (CONMI) Workshop Report. *SIGCOMM Comput. Commun. Rev.*, 36(2):41–48, 2006.

[16] M. Claypool, R. Kinicki, and C. Wills. User-centered network measurement. Technical Report WPI-CS-TR-07-08, Computer Science Department, Worcester Polytechnic Institute, 2007.

[17] S. DeDeo. Pagestats, May 2006. `http://www.cs.wpi.edu/~cew/pagestats/`.

[18] D. Games and M. Silva. The Case for Portuguese Web Search Engine. Technical Report DI/FCUL TR 03-03, Department of Informatics, University of Lisbon, 2004.

[19] O. Hamed and N. Kafri. Performance Prediction of Web Based Application Architectures Case Study: .NET vs. Java EE. *International Journal of Web Applications*, 1(3), 2009.

[20] Y. Hyun. Archipelago Meas. Infrastructure. In *CAIDA-WIDE Workshop*, San Diego, CA, 2006.

[21] B. Krishnamurthy and C. Wills. Privacy Diffusion on the Web: A Longitudinal Perspective. In *Proceedings of the World Wide Web Conference*, pages 541–550, Madrid, Spain, 2009.

[22] E. Neill. Web Characterization Project, 2002. `http://Wcp.oclc.org/`.

[23] T.S. Eugene Ng and Hui Zhang. Internet Network Distance with Coordinates-Based Approaches. In *Proceedings of IEEE INFOCOM*, New York, NY, 2002.

[24] J. Nielsen. Website Response Times, June 2009. `http://www.useit.com/alertbox/response-times.html`.

[25] E. O. Neill, B. Lavoie, and R. Bennett. Trends in the Evolution of the Public Web. *D-Lib Magazine*, 9(4), 2003.

[26] F. Papadopoulos and K. Psounis. Predicting the Performance of Internet-like Networks Using Scaled-down Replicas. In *ACM SIGMETRICS Performance Evaluation Review*, volume 35, New york, NY, 2007.

[27] L. Peterson, A. Bavier, M. Fiuczynski, and S. Muir. Experiences Building PlanetLab. In *USENIX Symposium on OSDI*, Seattle, WA, 2006.

[28] J. Pitkow. Summary of WWW Characterizations. In *Computer Networks and ISDN Systems*, volume 30, 1998.

[29] S. Punpiti. Measuring and Analysis of the Thai World Wide Web. In *Proceedings of the Asia Pacific Advance Network,*, pages 225–230, 2000.

[30] Peter J. Sevcik. Understanding How Users View Application Performance, July 2002. `http://www.netforecast.com/Articles/BCRC22PerformanceZones.pdf`.

[31] S. Souders. *High Performance Web Sites*. O'Reilly Media, 2007.

[32] C. Xing, M. Chen, and L. Yang. Predicting Available Bandwidth of Internet Path with Ultra Metric Space-based Approaches. In *Proceedings of Global Telecommunications Conference*, pages 1–6, Honolulu, HI, 2010.

[33] M. Zeljkovic, M. Kaplan, C. Wills, and M. Claypool. JavaScript and Flash Overhead in the Web Browser Sandbox. Technical Report WPI-CS-TR-10-14, Computer Science Department, Worcester Polytechnic Institute, 2010.