Evaluating the Impact of Different Document Types on the Performance of Web Cache Replacement Schemes*

Christoph Lindemann and Oliver P. Waldhorst
University of Dortmund
Department of Computer Science
August-Schmidt-Straße 12
44227 Dortmund, Germany
http://www4.cs.uni-dortmund.de/~Lindemann/

Abstract

*In this paper, we present a comprehensive performance study of Least Recently Used and Least Frequently Used with Dynamic Aging as traditional replacement schemes as well as for the newly proposed schemes Greedy Dual Size and Greedy Dual *. The goal of our study constitutes the understanding how these replacement schemes deal with different web document types. Using trace-driven simulation, we present curves plotting the hit rate and byte hit rate broken down for image, HTML, multi media, and application documents. The presented results show for the first workload that under the packet cost model Greedy Dual * outperforms the other schemes both in terms of hit rate and byte hit rate for image, HTML, and multi media documents. However, the advantages of Greedy Dual * diminish when the workload contains more distinct multi media documents and a larger number of requests to multi media documents.*

1. Introduction

Current and previous web request streams contain only a small percentage of requests to multi media documents. The percentage of requests to application documents like Postscript and PDF has already increased substantially in recent years. Due to the rapidly increasing popularity of digital audio (i.e., MP3) and video (i.e., MPEG) documents and the sustained growth of application documents in the web, we conjecture that in future workloads the percentage of requests to such documents will be substantially larger than in current request streams seen at a caching proxy. This change in workload characteristics will hold for both institutional caching proxies and proxies residing in a backbone network. Thus, it is important to investigate the impact of web document types on the performance of web cache replacement schemes.

A comprehensive characterization of previous web workloads was given by Arlitt and Williamson [2]. A recent survey article on performance characteristics of the web provided by Crovella [5] explains why many of the characteristics of web workloads (e.g., document sizes and document popularity) possess high variability. The temporal locality in web workloads has been subject to two recent papers. Jin and Bestavros investigated temporal locality in web cache request streams [7]. Eager, Mahanti and Williamson investigated the impact of temporal locality on proxy cache performance [9]. They also provided a detailed workload characterization for hierarchies of caching proxies [10]. They observed that in several workloads measured in 1998 HTML and image documents account for over 95% of all requests.

The optimization of cache replacement schemes is important because the growth rate of web content (i.e., multi media documents) is much higher than anticipated growth of memory sizes for future web caches [8]. Furthermore, recent studies (see e.g. [3]) have shown hit rate and byte hit rate grow in a log-like fashion as a function of size of the web cache. Cao and Irani introduced the web cache replacement scheme Greedy Dual Size (GDS [4]) that takes into account document sizes and a user defined cost function. They proved that GDS is on-line optimal with respect to this cost function. Jin and Bestavros introduced the web cache replacement scheme Greedy Dual * (GD *) as an improvement to GDS [8]. They compared the performance of this newly proposed replacement scheme with traditional schemes as Least Recently Used (LRU), Least Frequently Used with Dynamic Aging (LFU-DA), and with the size-aware scheme GDS [8]. Arlitt, Friedrich, and Jin provided a comparative performance study of six web cache replacement schemes among which are LRU, LFU-DA, and Greedy Dual Size [1]. They also observed an extreme

* This work was supported by the German Research Network Association (DFN-Verein) with funds of the Federal Ministry of Education and Research of Germany (BMBF)

non-uniformity in popularity of web requests seen at caching proxies. All these previous performance studies consider the overall request stream rather than requests to individual document types for analyzing the performance of replacement schemes.

In this paper, we present comprehensive performance studies for LRU and LFU-DA as traditional replacement schemes as well as newly proposed schemes GDS and GD* broken down to image, HTML, multi media and application documents. The performance results are derived by trace-driven simulation. The goal of our study constitutes the understanding how these replacement schemes deal with different web document types. This understanding is important for the effective design of web cache replacement schemes under changing workload characteristics. We consider two traces recently collected in upper-level caching proxies at DFN [6] and at RTP [11].

As novel results, the breakdown into document types illustrates that for the DFN trace under the constant cost model GD* is clearly superior to the other schemes in terms of hit rate for image and HTML documents while it performs worst of all considered schemes for multi media documents. Under the packet cost model, GD* is clearly superior to the other schemes in terms of hit rate for image, HTML and application documents. Furthermore, under the packet cost model GD* outperforms the other schemes both in terms of hit rates and byte hit rates for image, HTML, and multi media documents. For the RTP trace, the comparative performance study yields the same results for overall hit rates and byte hit rates as for the DFN trace. However, the advantages of GD* broken down for individual document types diminish or even vanish at all. For example, for image, HTML, and application documents under constant cost the advantage of GD* over the other schemes with respect to hit rate is considerably smaller than for the DFN trace. In terms of byte hit rate, under packet cost GDS outperforms GD* for HTML, multi media, and application documents. This is due to the significantly different workload characteristics. In fact, the RTP trace contains not only significantly higher percentages of distinct multi media documents and requests to multi media documents than the DFN trace, but also significantly different characteristics for document popularity and temporal correlation.

This paper is organized as follows. Section 2 provides a comprehensive characterization of the workloads derived from the considered traces. To make the paper self-contained, the four web cache replacement schemes under investigation are recalled in Section 3. In Sections 4, we present a comparative performance study of the considered web cache replacement schemes using the trace data. Finally, concluding remarks are given.

2. Workload Characterization

To characterize the performance of caching proxies, we consider two different traces. The first trace was recorded at the National Laboratory for Applied Network Research (NLANR [11]) cache site at Research Triangle Park, North Carolina in February 2001. A second trace was recorded in July 2001 in the German research network by DFN [6]. These traces are referred to as RTP and DFN, respectively. Both traces were collected at a primary-level proxy cache in the core network.

Preprocessing the traces, we exclude uncacheable documents by commonly known heuristics, e.g. by looking for string “cgi” or “?” in the requested URL. From the remaining requests, we consider responses with HTTP status codes 200 (OK), 203 (Non Authoritative Information), 206 (Partial Content), 300 (Multiple Choices), 301 (Moved Permanently), 302 (Found), and 304 (Not Modified) as cacheable according to [1], [4], [7]. Table 1 summarizes the properties of the traces after preprocessing. We break down the request stream of documents according to their content type as specified in the HTTP header. If no content type entry is specified, we guess the document type using the file extension. We distinguish between four main classes of web documents: Text documents (e.g., .html, .htm), image documents (e.g., .gif, .jpeg), multi media documents (e.g., .mp3, .ram, .mpeg, .mov), and application documents (e.g., .ps, .pdf, .zip). Text files (e.g. .tex, .java) are added to the class of HTML documents. Table 2 and 3 show a breakdown of the traces by document type. We observe that in current workloads HTML and image documents together account for about 95% of documents seen and of requests received. Multi media and application account for a small fraction of about 5% of unique documents and request, but for a significant fraction of over 40% of trace size and requested bytes. This observation has also been reported in [10] for a number of other proxy traces.

Tables 4 and 5 show a breakdown of the statistical properties of document and transfer sizes for the different document types. As observed in [10], we find that mean and median transfer sizes are largest for multi media documents. A new observation constitutes that the class of application documents shows quite large mean values for document and transfer sizes, while median sizes are very

	DFN	RTP
Date	27.06. - 03.07.01	09.02. - 14.02.01
Distinct Documents	2,987,565	2,227,339
Overall Size (GB)	39.54	34.09
Total Requests	6,718,210	4,144,009
Requested Data (GB)	80.32	86.54

Table 1. Properties of DFN and RTP trace

	Images	HTML	Multi Media	Application	Other
% of Distinct Documents	72.72	21.98	0.23	4.83	0.24
% of Overall Size	34.70	21.80	13.50	29.20	0.80
% of Total Requests	76.27	20.08	0.14	3.24	0.27
% of Requested Data	30.86	21.21	12.18	34.82	0.93

Table 2. DFN Trace: Workload characteristics broken down into document types

	Images	HTML	Multi Media	Application	Other
% of Distinct Documents	73.09	19.27	0.41	4.85	2.38
% of Overall Size	31.55	18.97	17.67	29.71	2.10
% of Total Requests	74.29	18.05	0.33	4.84	2.49
% of Requested Data	19.71	44.19	11.65	21.92	2.53

Table 3. RTP Trace: Workload characteristics broken down into document types

small. This can be explained by the broad spectrum of different document types covered by the Mime types starting with the prefix *application/*.

A key property for the performance of web caching constitutes temporal locality in the request stream. Temporal locality can be quantified by the relationship between the probability of an access to a web document and the time passed since the last access to this document. As discussed in [7], [8], temporal locality in the request stream is caused by two different sources: The popularity of web documents and the temporal correlation in the request stream. A popular web document is seen often in a request stream. Therefore, popular documents are referenced more often in a short time interval than less popular documents. Temporal correlation takes into account the time between two successive references to the same document. A hot web document is requested several times in a short interval whereas the average document is referenced just a few times. Temporal locality can be characterized by two parameters. The first parameter, denoted as the popularity index α describes the distribution of popularity among the individual documents. The number of requests N to a web document is proportional to its popularity rank ρ to the power of $-\alpha$, that is: $N \sim \rho^{-\alpha}$. The popularity index α can be determined the slope of the log/log scale plot for the number of references to a web document as function of its popularity rank. The second parameter, denoted as β, measures the temporal correlation between two successive references to the same web document. The probability P that a document is requested again after n requests is proportional to n to the power of $-\beta$, that is: $P \sim n^{-\beta}$, for equally popular documents. The parameter β can be determined by plotting the reference count as a function of references made between two successive references to the same document for equally popular documents. For a detailed discussion of the two sources of temporal locality, we refer to [8].

For the DFN and RTP traces, the calculated values for α and β with respect to the different document type are shown in Tables 4 and 5. Large values of α show that there are some extremely popular image documents, whereas smaller values show that requests are more evenly distributed among text documents and most evenly among multi media and application documents. The slope β of the distribution of short-term temporal correlation shows the inverse trend. That is there is a high correlation between two successive requests to a multi media or application document, whereas successive requests to images are nearly uncorrelated. The impact of the characteristics of the different document types on the performance of web cache replacement schemes is shown in the performance curves presented in Section 4.

3. Considered Cache Replacement Schemes

In traditional memory systems object sizes (i.e., a cache line or a memory page) and miss penalties (delay for bringing an object into the cache) are constant. The salient feature of web caching lies in the high variability of both the cost for bringing in new web documents and the size of such documents. In this paper, we present a comparative performance study for Least Recently Used, Least Frequently Used with Dynamic Aging, and two size-aware replacement schemes Greedy Dual Size and Greedy Dual *, which have been recently proposed.

In [8], two cost models for web cache replacement schemes have been introduced. In the constant cost model, the cost of document retrieval is fixed. The packet cost model assumes that the number of TCP packets transmitted determines the cost of document retrieval.

	Images	HTML	Multi Media	Application	Other
Mean of Document Size (KB)	6.623	13.762	825.283	83.929	45.242
Median of Document Size (KB)	2.320	3.212	36.118	0.737	0.737
CoV of Document Size	4.207	112.502	3.161	11.128	8.229
Mean of Transfer Size (KB)	5.073	13.242	1114.010	134.859	42.028
Median of Transfer Size (KB)	1.758	3.384	45.457	0.883	3.834
CoV of Transfer Size	4.106	81.677	3.379	7.771	9.416
Slope of Popularity Distribution α	0.653	0.536	0.396	0.348	0.847
Degree of Temporal Correlations β	0.521	0.600	0.794	0.819	0.697

Table 4. DFN Trace: Breakdown of document sizes and temporal locality

	Images	HTML	Multi Media	Application	Other
Mean of Document Size (KB)	6.927	15.803	692.368	98.328	14.161
Median of Document Size (KB)	2.349	3.273	37.977	0.820	0.316
CoV of Document Size	5.084	10.597	3.248	13.556	45.311
Mean of Transfer Size (KB)	5.810	53.603	781.552	99.072	22.410
Median of Transfer Size (KB)	1.948	5.719	48.068	1.071	2.937
CoV of Transfer Size	5.774	7.907	3.645	12.081	28.064
Slope of Popularity Distribution α	0.524	0.527	0.455	0.400	0.649
Degree of Temporal Correlations β	0.616	0.774	0.823	0.883	0.712

Table 5. RTP Trace: Breakdown of document sizes and temporal locality

The constant cost model is the model of choice for institutional proxy caches, which mainly aim at reducing end user latency by optimizing the hit rate. The packet cost model is appropriate for backbone proxy caches aiming at reducing network traffic by optimizing the byte hit rate.

Least Recently Used (LRU [2]) is a recency-based policy. It is based on the assumption that a recently referenced document will be referenced again in near future. Therefore, on replacement LRU removes the document from cache that has not been referenced for the longest period of time. LRU is the most widely used cache replacement scheme. Because LRU considers a fixed cost and size of documents, LRU does not discriminate large documents and thus optimizes the byte hit rate. The good performance of LRU is due to the exploitation of locality of reference in the document request stream. The disadvantage of LRU lies in neglecting the variability in cost and size of web documents. Furthermore, LRU does not take into account frequency information in the request stream.

Least Frequently Used with Dynamic Aging (LFU-DA [2]) is a frequency-based policy that also takes into account the recency information under a fixed cost and fixed size assumption. In LFU, a decision to evict a document from cache is made by the number of references made to that document. The reference count for all documents in cache is kept and the document with smallest reference count is evicted. LFU-DA extends LFU by a dynamic aging algorithm in order to avoid cache pollution. LFU-DA keeps a cache age, which is set to the reference count of the last evicted document. When putting a new document into cache or referencing an old one, the cache age is added to the documents reference count. It has been shown that LFU-DA achieves high byte hit rates.

Greedy Dual Size (GDS [4]) proposed by Cao and Irani considers variability in cost and size of web documents by choosing the victim for replacement based on the ratio between the cost and size of documents. GDS associates a value H with each web document p in the cache. When document p is brought initially into the cache or is referenced while already in cache, $H(p)$ is set to $c(p)/s(p)$. Here $s(p)$ is the document size and $c(p)$ is a cost function describing the cost of bringing p into the cache. When a document has to be replaced, the victim $\hat{p}$ with $\hat{H}_{\min} := \min\{H(p)\}$ is chosen among all documents resident in the cache. Subsequently, all H values are reduced by $\hat{H}_{\min}$ [4]. However, similar to LRU, the disadvantage of GDS lies in not taking into account frequency information in the request stream.

Greedy Dual * (GD* [7], [8]) proposed by Jin and Bestavros captures both popularity and temporal correlation in a web document reference stream. The frequency in the formula for the base value $H'(p)$ captures long-term popularity. Temporal correlation is taken into account by the rate of aging controlled by the parameter β. GD* sets the values of H for a document p to $H'(p) = \left(f(p) \cdot c(p)/s(p)\right)^{-\beta}$ where $f(p)$ is the reference

count of the document. The parameter β characterizes the temporal correlation between successive references to a certain document observed in the workload as recalled in Section 2. The novel feature of GD* is that $f(p)$ and β can be calculated in an on-line fashion, which makes the algorithm adaptive to these workload characteristics.

GDS and GD* describe families of algorithms. The optimized performance measure (i.e. hit rate or byte hit rate) of a specific implementation depends on the definition of the cost function $c(p)$. In this paper we examine two variants of GDS and GD*. The first applies the constant cost model by setting cost function to $c(p) = 1$. We refer to the resulting algorithms as GDS(1) and GD*(1), respectively. The second variant applies the packet cost model by setting the cost function to the number of TCP packets needed to transmit document p, i.e., $c(p) = 2 + s(p)/536$. These replacement schemes are denoted GDS(packets) and GD*(packets), respectively.

4. Comparative Performance Study

4.1. The Simulation Model

To investigate impact of the different document types on the performance of the replacement schemes LRU, LFA-DA, GDS(1), GD*(1), GDS(packets), and GD*(packets) we developed as simulation model of a single caching proxy. We use this simulation model to determine hit rates and byte hit rates broken down for each of the considered document types. During the processing of the traces, the simulator keeps track of the number of requests and the number of hits for individual document types. Subsequently, hit rate and byte hit rate are determined individually for each type. That is, the hit rate on images is calculated as the ratio between the number of hits on images and the number of requested images. Besides recording the data needed to determine the hit rates and byte hit rates, the simulator keeps track of the number of cached documents and document sizes for individual document types. These quantities are used to calculate the fraction of cache documents and the fraction of cached bytes used by the individual document types.

The performance measures hit rate and byte hit rate are significantly influenced by initial misses for an empty cache and by document modifications. To avoid cold start misses, we use 10% of the total requests recorded in a trace to fill the cache. Following [1], we distinguish between document modifications and interrupted document transfers. Therefore, the simulator keeps track of the document size for each individual document recorded in the traces. If the size changes by less than 5% between two successive requests, we assume that the document has been modified and count the request as a miss. Otherwise, we assume that the client has interrupted the document transfer. Note that this treatment of document modifications is different to [7], [8]. In these previous works, each change of the document size is considered as a document modification. This assumption results in higher modification rates especially for large multi media and application documents for which users are likely to interrupt transfers due to large transfer times.

4.2. Adaptability of Greedy Dual*

In a first experiment, we evaluate the ability of the GD* replacement scheme to adapt to the actual workload seen at the proxy cache. Under the constant cost model, the optimal case constitutes that for each document type (i.e., images, HTML, multi media, and application) the fraction of cached documents is equal to the fraction of requests to this document type in the request stream. Figure 1 plots the fraction of cached documents (left) and cached bytes (right) as a function of requests for each document type and GD*(1) and LRU, respectively. As workload the DFN trace is considered. The cache size is set to 1 GByte.

Figure 1 shows that for GD*(1) the fraction of cached bytes for each document type is nearly constant. These fractions are close to the corresponding fraction of requests and close to the corresponding fractions of requested documents for each type. Opposed to that, for LRU the fractions of cached bytes for each class are highly variable and quite different to the fraction of requested documents of this type. For example, the fraction of images is smaller than 76% and the fraction of application documents is substantially larger than 15%. Similar results have been observed for the RTP trace. These observations explain why GD*(1) achieves high hit rates: GD*(1) does not waste space of the web cache by keeping large multi media and application documents that will not be requested again in the near future. Opposed to GD*(1), LRU keeps the count of documents for each class close to the according fraction of requests. Thus, LRU is able to deliver even large documents, achieving high byte hit rates on the cost of lower hit rates.

4.3. Performance Results for DFN Trace

In a second experiment, we provide a comparative study for the web replacement schemes LRU, LFU-DA, GDS(1), and GD*(1) for the DFN trace. Other replacement schemes are not considered, since in [4] it has been shown that GDS outperforms these schemes. As performance measures, hit rate and byte hit rate are considered. In Figure 2 and 3, we plot the hit rate (left) and byte hit rate (right) for increasing cache sizes. Cache sizes are chosen from about 0.05% to about 40% of overall trace size mentioned in Table 1. In the following,

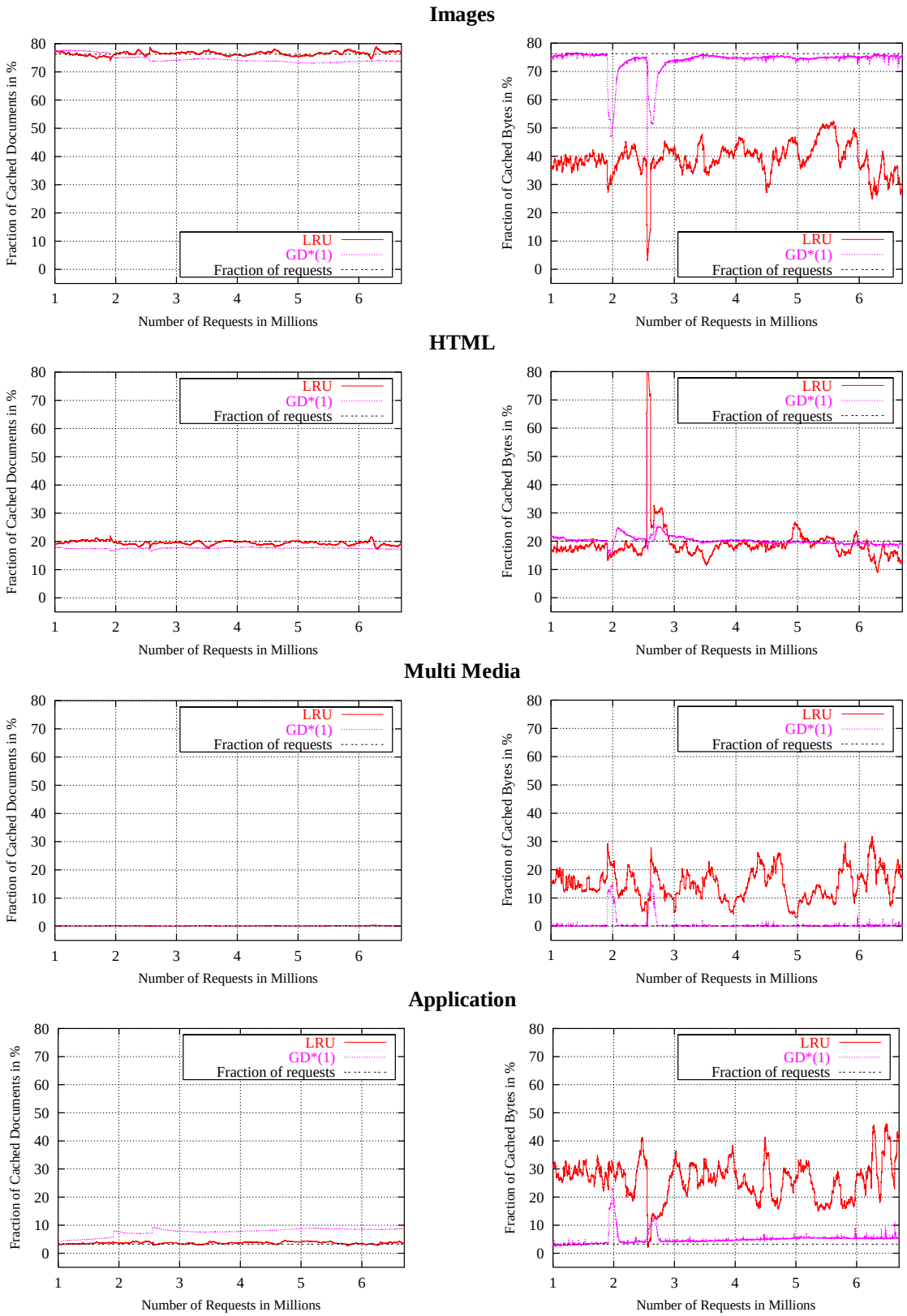

Figure 1. Occupation of web cache by the different document types. Left: fraction of cached documents; right: fraction of cached bytes

we relate our observations to the results of [8], where GD* has been introduced.

Consistent with [8], we observe that frequency based replacement schemes outperform recency-based schemes in terms of hit rates. As shown in Figure 2, GD*(1) outperforms GDS(1) and LFU-DA outperforms LRU in terms of hit rate for the document types images, HTML, and application. The breakdown into document types shows that this is most obvious for images and application documents while there is only a small advantage for HTML documents. For multi media documents, LFU-DA achieves the best hit rates closely followed by LRU. Moreover, GD*(1) performs worse than GDS(1), because the ratio between document reference count and document size, which gets very small for infrequent accessed multi media documents, becomes even smaller when taken to the power of $1/\beta$.

Consistent with [8], we observe that in terms of hit rate LRU and LFU-DA perform worse than GDS(1) and GD*(1). The breakdown into document types in Figure 2 shows that this observation is significant for HTML and image documents while there are only small advantages for application documents. This observation can be explained by the fact that LRU and LFU-DA do not take into account document sizes. For large multi media documents, the size-awareness of GDS(1) and GD*(1) leads to significantly lower hit rates and byte hit rates. Thus, opposed to [8] we do not observe that GD*(1) stays competitive with LRU and LFU-DA in terms of byte hit rate. As shown in Figure 2, for image, HTML and application documents the byte hit rate achieved by GD*(1) stays competitive to LRU and LFU-DA. However, for multi media documents GD*(1) performs significantly worse in terms of byte hit rate than LRU and LFU-DA. Since the byte hit rate for multi media documents dominate the overall byte hit rate, this observation leads to a poor byte hit rate for GD*(1). This inconsistency with [8] can be explained by the different treatment of document modifications as explained in Section 4.1.

In a third experiment, we study the performance of GD* and GDS for DFN trace under packet cost model. Figure 3 compares GD*(packets) and GDS(packets) with LRU and LFU-DA. Consistent with [8], we observe that GD*(packets) outperforms LRU, LFU-DA and GDS(packets) both in terms of hit and byte hit rates. Opposed to the constant cost model of GD*(1), GD*(packets) does not discriminate large documents. The breakdown into document types shows that GD*(packets) has clear advantages in terms of hit rate over the other schemes for images, HTML and application documents. Furthermore, GD*(packets) achieves significant higher byte hit rates than LRU, LFU-DA, and GDS(packets) for images, HTML, and multi media documents. Comparing Figures 2 and 3, GD*(packets) achieves lower hit rates than GD*(1) for image and application documents but considerably higher byte hit rates for HTML, multi media, and application documents. For multi media documents GD*(packets) clearly outperforms GD*(1) both for hit rate and byte hit rate.

4.4. Performance Results for RTP Trace

Recall from Section 2 that the RTP trace has significantly different characteristics than the DFN trace. In fact, the RTP trace contains a significantly higher percentage of distinct multi media documents and percentage of requests to multi media documents (i.e., 0.41% versus 0.23% and 0.33% versus 0.14%). Moreover, the RTP trace contains a smaller percentage of requested data to image and application documents than the DFN trace (i.e., 19.7% versus 30.8% and 21.9% versus 34.8%, respectively) and a higher percentage of requests to HTML documents (i.e., 44.2% versus 21.2%). Comparing Tables 4 and 5 show significant differences between the DFN trace and the RTP trace in the coefficient of variation (COV) of the size of HTML documents. Due to space limitations, we omit the performance curves and just give a summary of the results observed.

Under the constant cost model, i.e., GD*(1) and GDS(1), the RTP trace yields in a comparative performance study both for hit rate and byte hit rate, the same results as the DFN trace. That is for images, HTML and application documents GD*(1) closely followed by GDS(1) outperform LFU-DA and LRU in terms of hit rate. For multi media documents LFU-DA and LRU clearly perform better than GDS(1) and GD*(1) both for hit rate and byte hit rate. However, the main difference of the RTP trace to the DFN trace lies in the scale of the y-axis. That is for the RTP trace hit rates up to 0.5 are achieved for image and application documents. For all document types byte hit rates up to 0.3 are achieved in the RTP trace.

Under the packet cost model, i.e., GD*(packets) and GDS(packets), the RTP trace yields in a comparative performance study regarding hit rate for image, HTML, and application documents the same results as the DFN trace. However, for each of these three document types the advantage of GD*(packets) over the other schemes is smaller than in the DFN trace. Moreover, in the RTP trace GD*(packets) achieves for multi media documents a slightly lower hit rate than GDS(packets), LRU and LFU-DA. In terms of byte hit rate, GD*(packets) does not perform better than GDS(packets) for HTML, multi media and application documents. Overall, hit rates up to 0.5 are achieved for image, HTML, and application documents. For all document types byte hit rates up to 0.4 are achieved.

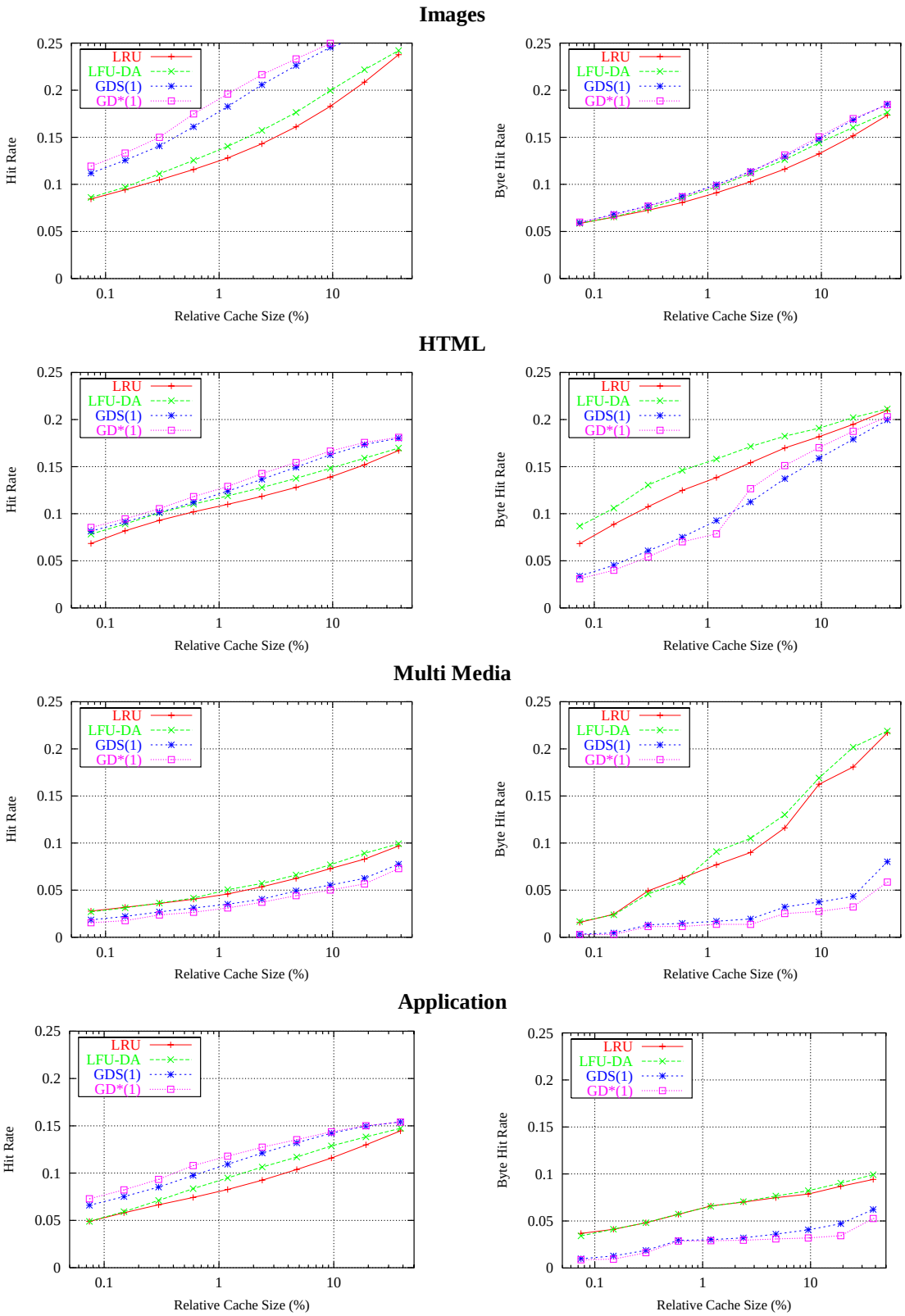

Figure 2. DFN trace: Breakdown of hit rates for different document types under constant cost model. Left: hit rate; right: byte hit rate

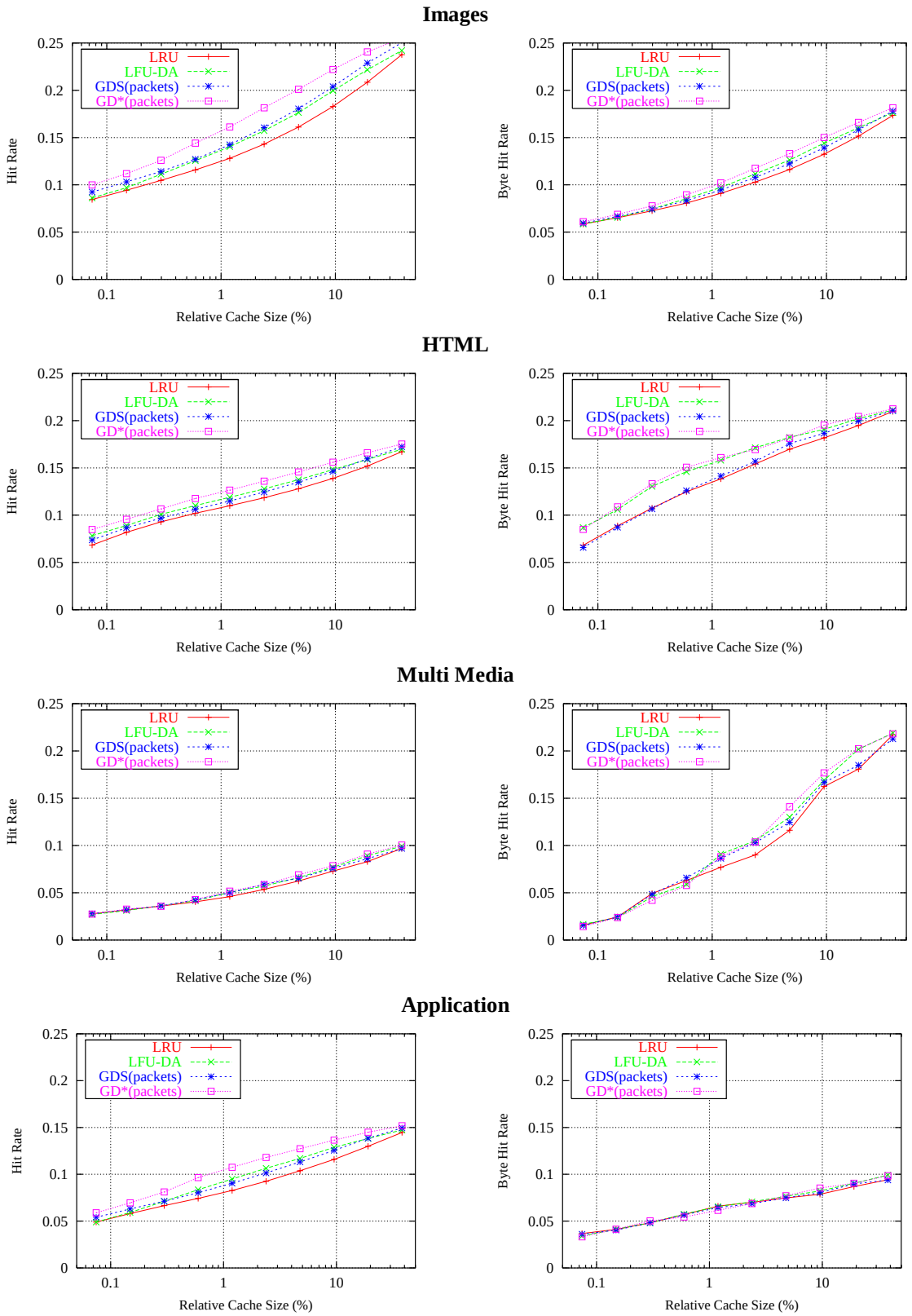

Figure 3. DFN trace: Breakdown of hit rates for different document types under packet cost model. Left: hit rate; right: byte hit rate

Recalling Tables 4 and 5, these different performance characteristics of GD*(packets) for the RTP trace can be explained as follows. GD*(packets) suffers from the small slope a of the popularity distribution in the RTP trace. This leads to many equally popular documents, introducing false frequency decisions. The slopes β of the distribution of temporal correlation for HTML, multi media, and application documents are much bigger than the overall slope of the distribution of temporal correlation, which is dominated by the slope of image documents. This causes additional errors in replacement decisions performed by GD*(packets), leading to low byte hit rates for HTML, multi media, and application documents.

Conclusion

In this paper, we presented comprehensive performance studies for LRU and LFU-DA as traditional replacement schemes as well as newly proposed schemes GDS and GD*. Opposed to previous studies, we presented curves plotting hit rate and byte hit rates broken down to images, HTML, multi media, and application documents in order to understand how web cache replacement schemes deal with different document types.

An investigation of the adaptability of GD*(1) evidently shows that GD*(1) does not waste cache space by keeping large multi media documents that are likely not to be referenced in the near future. This observation explains why GD*(1) almost always achieves considerably higher hit rate than other replacement schemes. In further experiments, we present comparative performance studies for GD*, GDS, LFU-DA, and LRU under the constant cost model and under the packet cost model, respectively.

For the overall hit and byte hit rates, all but one of our results are consistent with [8] both for the DFN trace and the RTP trace. However, for hit and byte hit rates broken down for individual document types, we observe significant differences to [8]. For the DFN trace, GD*(1) is clearly superior to the other schemes in terms of hit rate for image and HTML documents. GD*(packets) outperforms the other schemes both in terms of hit rate and byte hit rate for image, HTML, and multi media documents. For the RTP trace, the advantages of GD* broken down for individual document types diminish or even vanish at all because the RTP trace contains not only significantly higher percentages of distinct multi media documents and requests to multi media documents than the DFN trace, but also significantly different characteristics for document popularity and temporal correlation.

References

[1] M. F. Arlitt, R. Friedrich, and T. Jin, Performance Evaluation of Web Proxy Cache Replacement Policies. *Performance Evaluation* **39**, 149-164, 2000.

[2] M. F. Arlitt and C. Williamson, Internet Web Servers, Workload Characterization and Performance Implications. *IEEE/ACM Trans. on Networking* **5**, 631-645, 1997.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, Web Caching and Zipf-like Distributions: Evidence and Implications. *Proc. 21st Annual Conf. of the IEEE Computer and Communication Societies*, (*INFOCOM 99*) New York, NY, 126-134, 1999.

[4] P. Cao and S. Irani, Cost-Aware WWW Proxy Caching Algorithms, *Proc. 1st USENIX Symp. on Internet Technologies and Systems*, Monterey, CA, 193-206, 1997.

[5] M. Crovella, Performance Characteristics of the World Wide Web, in: G. Haring, C. Lindemann, M. Reiser (Eds.) *Performance Evaluation: Origins and Directions*, *LNCS Vol. 1769*, 219-232, Springer, 2000.

[6] C. Grimm, H. Pralle and J. Vöckler, The DFN Cache Mesh, http://www.cache.dfn.de/

[7] S. Jin and A. Bestavros, Temporal Locality in Web Request Streams: Sources, Characteristics, and Caching Implications. *Technical Report 1999-014*, CS Department, Boston University, 1999.

[8] S. Jin and A. Bestavros, Greedy Dual* Web Caching Algorithm: Exploiting the Two Sources of Temporal Locality in Web Request Streams. *Computer Communications Special Issue on 5th Web Caching and Content Delivery Workshop*, 22, 174-183, 2000.

[9] A. Mahanti, D. Eager, and C. Williamson, Temporal Locality and its Impact on Web Proxy Cache Performance. *Performance Evaluation* **42**, 187-203, 2000.

[10] A. Mahanti, C. Williamson and D. Eager, Workload Characterization of a Web Proxy Caching Hierarchy, *IEEE Network Magazine* **14**(3), 16-23, 2000

[11] NLANR, National Laboratory for Applied Network Research, http://www.nlanr.net/