Analysis of a Large Multimedia-Rich Web Portal for the Validation of Personal Delivery Networks

Jeroen van der Hooft†, Stefano Petrangeli†, Tim Wauters†, Rameez Rahman§,
Nico Verzijp§, Rafael Huysegems§, Tom Bostoen§ and Filip De Turck†
† Department of Information Technology, Ghent University - iMinds
§ Nokia - Bell Labs
E-mail: jeroen.vanderhooft@intec.ugent.be

***Abstract*—With the increasing popularity of multimedia-rich web portals, reducing latency has become more and more important. The current median web page load time is in the order of seconds, while research has shown that user waiting times must remain below two seconds to achieve optimal acceptance. In this paper, we analyzed a large dataset obtained from a major Belgian news provider, focusing on content popularity, user activity and user preference towards article news categories. Based on this analysis, we introduce the concept of personal delivery networks (PDNs), in which content is stored closer to the end user, at delivery caches in the edge of the core network or even in the access network. PDN nodes proactively prefetch and evict content on a per-user basis, opening opportunities for personalized low-latency delivery of multimedia-rich web applications. Initial results show that a PDN-based approach allows to significantly reduce the average latency.**

I. Introduction

While multimedia consumption used to be limited to home desktops, progress in industry and academia alike has allowed users to access high-quality multimedia entertainment on a broad range of devices, including TVs, PCs, tablets and smartphones. Content delivered over the Internet used to be mostly static, but nowadays consists mainly of interactive web applications, for which latency is the main performance bottleneck perceived by end users. As an example, the current median web page load time is in the order of seconds, while user waiting times must remain below two seconds to achieve optimal acceptance [1], [2]. Moreover, the move to multimedia-rich web applications also resulted in an increased dependency on over-the-top video streaming to deliver the video content to the end user. A recent study by Conviva showed that failure to deliver the video over the best-effort Internet in a timely manner, resulting in a high initial startup delay and video stalling, has a highly negative impact on the user's Quality of Experience (QoE) [3].

The situation has significantly improved through the use of content delivery networks (CDNs). CDNs have been around for over a decade now, allowing the content provider to relieve the origin server and bring his content closer to the end user. Although CDNs were typically used to cache static content only, commercial solutions such as Amazon Cloudfront [4] allow caching of semi-dynamic objects as well, increasing opportunities for dynamic content in web applications. Recently, telecommunications service providers have started to deploy their own telco CDNs, which deliver content from servers in the access/edge network closer to the end user than the delivery servers of a traditional CDN. This approach not only reduces latency, but also allows operators to implement their own content management operations in order to provide a better QoE. However, telco CDNs are currently mainly used for the delivery of premium video content, and not for the delivery of multimedia-rich web applications and web portals.

In this paper, focus is on personalized low-latency delivery of multimedia content through the concept of personal delivery networks (PDNs). In contrast with traditional CDNs, PDNs store content much closer to the end user, at delivery caches in the edge of the core network or even in the access network. They proactively prefetch and evict content to and from these caches on a per-user basis. An important aspect of a PDN is thus user profiling, to accurately determine a user's preference towards certain content, such as news pages in web portals or songs and videos in multimedia web applications. Using such approach not only results in a reduced latency, but also allows to increase the user's QoE through better network provisioning in the last mile. The main goal in this paper is to illustrate possible latency reductions with PDN, applying the concept on a relevant use case where the user's QoE can be enhanced by actively prefetching content.

The remainder of this paper is structured as follows. In Section II, we analyze an existing dataset from a major Belgian news website, as a step towards content and user profiling for web portals. In Section III, the concept of PDNs is presented in more detail, listing a number of key enablers and possible advantages. In Section IV, we discuss how the analyzed data can be used to intelligently place content closer to the user, moving from CDNs in the core network towards PDN delivery nodes. Results from a first evaluation are presented in Section V, characterizing the possible gains of a PDN-based approach in terms of latency reduction. Final conclusions are presented in Section VI, along with an outlook on future work.

II. Multimedia-Rich Web Portal Dataset

Deredactie.be[1] is one of the major news websites in Belgium. In recent years, its focus has shifted largely from simple text-based articles towards multimedia-rich news reports. It

[1] http://deredactie.be/cm/vrtnieuws

978-3-901882-89-0 @2017 IFIP

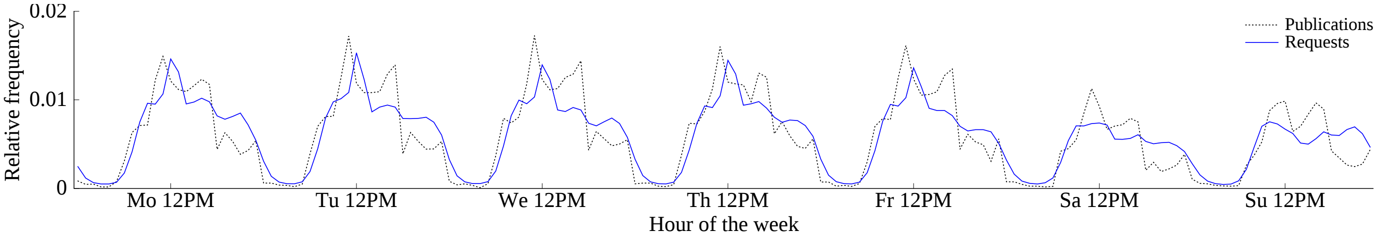

Figure 1. Relative frequency of article publications and article requests per hour of the week.

Attribute	Details
userId	User's cookie ID for deredactie.be
userIP	User's IPv4 address
userIPs	User's registered IPv4 addresses
userTime	User's local time
serverTime	Server's local time
requestURL	Requested URL
referURL	Referrer URL
userAgent	JSON containing user agent information
timeOnPage	Time spent on page

Table I
OVERVIEW OF COLLECTED INFORMATION [5].

is possible to watch recent news broadcasts on demand, and to view livestreams on events such as parliamentary debates and speeches. The news website is also integrated with other websites providing information on sports and culture, and links to a radio station portal[2]. Because of the high amount of multimedia-rich articles and active users, the web portal is an excellent use case for content prefetching.

From April 2015 till January 2016, all requests from the deredactie.be were actively monitored and logged by Van Canneyt et al. [5]. To this end, an embedded JavaScript was used to direct clients to a dedicated server hosted on iLab.t's infrastructure[3]. Clients requested a one-pixel image, while providing required information through a specific URL query. Table I gives an overview of all attributes, which were logged and stored as JSON objects. The user's cookie ID is used as a unique user identifier, although one most note that one user can use multiple operating systems and browsers, and can renew the cookie at any point in time. The requested URL can be parsed to retrieve the article ID on deredactie.be, and the referrer URL to find correlations between pages and analyze the article's popularity on social media. The user agent and the time spent on a page was logged as well, although this information is not used in this paper. In total, the dataset contains over 90 million article requests, issued by more than 12 million different users. Note that the dataset has been anonymized, hashing all IP addresses and cookie IDs.

For every article requested in this ten-month period, the corresponding XML file was retrieved from the website and parsed to extract information on the article's title, content, author etc. For every article, the website also assigns a certain category, including but not limited to Video, World, Belgium, Regional, Economy, Science, Opinion and Culture & Media. This categorization allows us to take an initial step towards user profiling, focusing on a user's preference towards certain article categories.

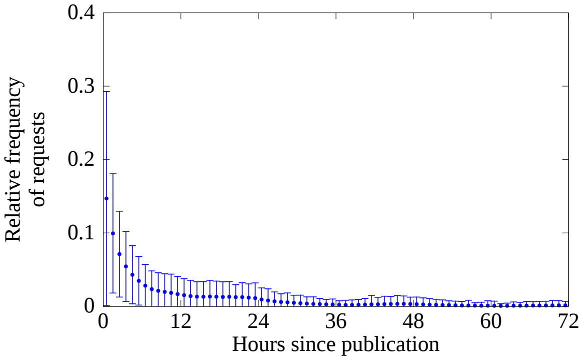

Figure 2. Relative frequency of article requests as a function of the hours since publication.

A. Article Publication & Consumption

Over a period of ten months, 26,006 Dutch annotated articles (i.e. articles with a defined title, category, author etc.) have been published by the news provider, resulting in an average of 85.0 articles per day. Note that this does not include articles which are updated on a daily basis, such as the weather forecast. As shown in Figure 1, most articles are published between 9AM and 5PM, yet several authors tend to be active until 1AM. A strong decrease of publications is observed between the hours of 12 to 1PM and 6 to 7PM, which correspond to lunch and dinner time respectively. Looking at article consumption, less than 10% of requests are issued between 12 and 7AM, while a significant peak is observed between 12 and 1PM on week days.

Since the focus in this paper is on improving the user experience for media-rich websites by bringing the content closer to the user, either by caching in the edge or access network or by actively pushing content to the user's browser, we are mainly interested in when published content is consumed by the user. For all annotated articles published in the ten-month period, Figure 2 shows the relative frequency of requests as a function of the hours since publication. For an average article, 50% of the total number of requests are issued within the first six hours of publication. The best-fitting line corresponds to a power-law distribution with $f(x) = 0.625 \times x^{-1.497}$, where x indicates the hours since publication. This is in line with research by Van Canneyt et al., who have shown that the interest in news content decreases rapidly over time [5].

[2] http://radioplus.be
[3] http://ilabt.iminds.be/iminds-virtualwall-overview

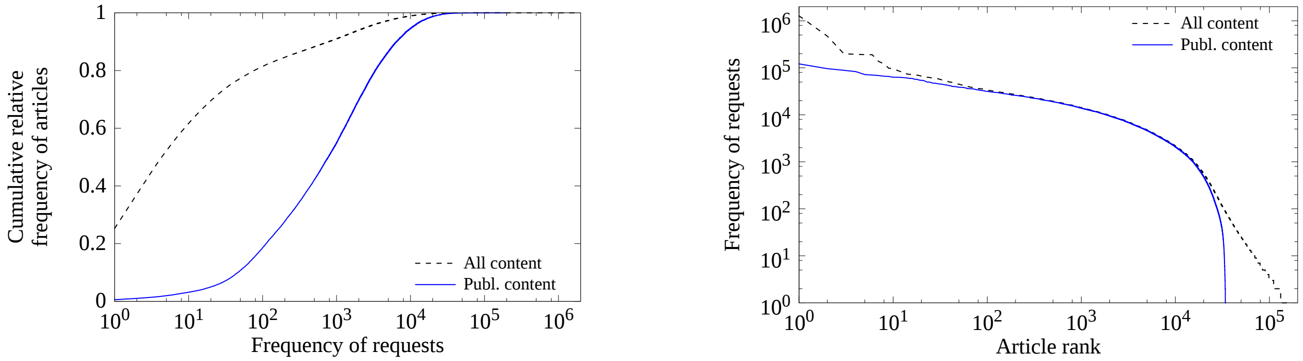

Figure 3. Cumulative frequency of article popularity (left) and article popularity in terms of article rank (right). Of the articles published within the considered ten-month period, 2.9% of articles is read less than 10 times, while the top 30.2% of articles is read more than 2000 times each.

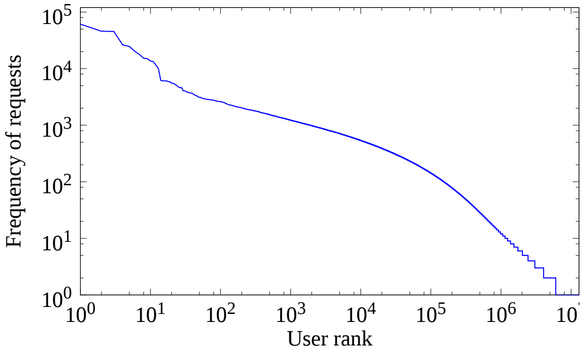

Figure 4. Requested number of articles in terms of user rank. Within the considered ten-month period, 90.3% of users read less than 10 articles, while the top 1.2% of users read more than 100 articles each.

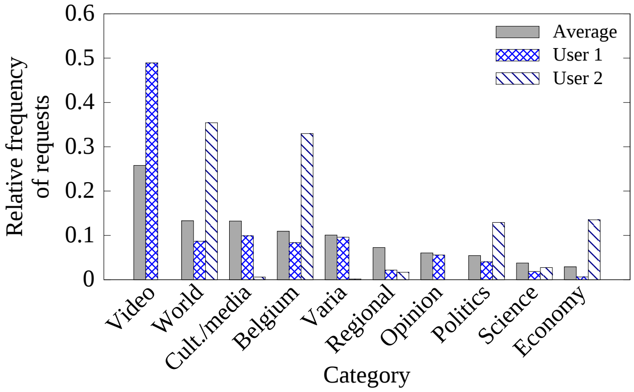

Figure 5. Relative frequency of article requests within a certain category, for the average user and for two individual users.

B. Article Popularity

Another interesting factor is the overall article popularity, i.e. the total number of requests an article receives within the considered ten-month period. As shown in Figure 3a for all 176,285 visited articles, 60.3% of articles are visited less than 10 times, while only 9.0% of articles are visited more than 1,000 times. The most visited article is the weather forecast, since it is updated on a daily basis and is thus requested multiple times by the same user. The number of requested articles is a factor 7 higher than the number of published articles within this period, which shows that a lot of older articles are still visited (e.g. through search engines). Looking at issued requests for articles published within the ten-month period only, results are considerably different. This time, only 2.9% of articles are visited less than 10 times, while 30.2% of articles are visited 2,000 times or more. Note that the article rank and popularity, illustrated in Figure 3b, largely coincide, indicating a straightforward relation between the popularity of content and whether or not the content was published before or during data collection.

C. User Activity

Figure 4 shows the cumulative relative frequency of user activity, measured as the total amount of requested articles. Within the considered ten-month period, 88.8% of users read less than 10 articles, while the top 0.5% of users read more than 1,000 articles each. Outliers are observed for the 10 most active users; closer examination revealed that these users are in fact crawlers (e.g. the Googlebot-Image crawler). However, one must be careful to interpret these results: user identification is based on the cookie ID, which can change over time by clearing the cache and is susceptible to user stitching, i.e. the impossibility of recognizing different users on different devices and browsers. We can conclude, however, that there is a high variability among users, indicating a potential benefit for a personalized approach for content delivery (e.g. more assigned storage space).

D. User Preferences

As an initial step towards user profiling, we analyzed users' preference for different categories of the news website. Figure 5 shows the overall user interest, along with the preferences of two individual users who exhibit very different navigation patterns. On average, the three most popular categories are Video, World and Culture & Media, with 25.8%, 13.3% and 13.2% of user requests respectively. Other categories such as Science and Economy, accounting for 3.8% and 2.9% respectively, are less popular. Looking at the first individual user, the Video category accounts for 48.9% of article requests. A closer look on the collected logs revealed that this user mainly uses the website to watch full news programs. Other popular categories include Culture & Media and Miscella-

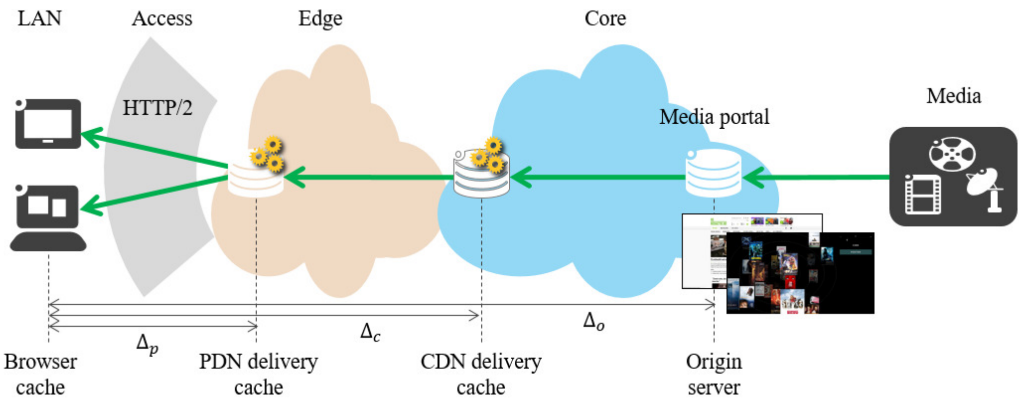

Figure 6. Personal Delivery Network architecture. Whereas traditional CDNs bring the content closer in the core network, PDN delivery nodes are located in the edge network or even in the access network.

neous, which often embed video within the text article. The second user however, has significantly different preferences. Not a single article in the Video category was requested, indicating a preference towards text-based articles. This is confirmed by a strong preference towards the categories World and Belgium, accounting for 35.4% and 33.0% of article requests respectively, which are mainly text-based. The same is true for the category Politics and Economy, which account for 12.9% and 13.5% of this user's requests respectively.

Based on these results, we conclude that different users are likely to consume the provided media in different ways, and to have very different preferences towards article categories. In the next section, we will explain in more detail how a user's preference towards one or multiple categories can be used to more intelligently place newly published articles towards the edge or the network, or even to push new content to the user's browser to anticipate future user requests and reduce the perceived latency.

III. Personal Delivery Network Architecture

CDNs have been used for quite some time now, relieving the origin server and bringing content closer to the end user. Recently, there has been a major shift towards edge computing, pushing the frontier of services away from centralized nodes to the logical extremes of the network [6]. One example for multimedia content delivery is the deployment of telco CDNs, which allow content providers to offer premium video streaming with a superior QoE. Building further upon this trend, we envision a per-user low-latency content delivery through the concept of a PDN. In contrast with traditional CDNs, PDNs store content much closer to the end user, at delivery nodes in the edge of the core network or even in the access network. This allows to further reduce latency and improve the user's QoE through better network provisioning in the last mile.

Figure 6 illustrates the approach of PDN. Origin servers are located anywhere in the backbone network, storing all multimedia content related to the provided services. CDN delivery nodes alleviate the load on these origin servers and significantly reduce the latency between client and server. The PDN takes this process one step further, proactively prefetching and evicting content to and from delivery caches located at the edge of the network. User profiling is used to determine a user's preference towards certain content, and allows a per-user caching strategy. Many techniques for user profiling have been proposed in literature, and can be applied towards intelligent content placement [7]. Content providers such as Netflix[4] use proprietary software to determine user preferences and suggestions, which can be taken into account as well. PDNs allows internet service providers to leverage their edge cloud for user-centric content delivery, further reducing the latency and increasing the user's QoE.

The concept of PDN can be further extended with storage capabilities at the user's home (e.g. at the home gateway) or at the client's device (e.g. browser cache). Recently developed techniques can be used to achieve the latter, such as the HTTP/2 protocol [8]. Among others, this successor of HTTP/1.1 offers a server push feature, allowing servers or proxies to actively push content along with previously requested content. Although a reduction of the page loading time up to 64% can be achieved in some cases, the gains of HTTP/2 are highly scenario dependent [9]. Using more intelligent web prediction algorithms, combined with HTTP/2's server push to transfer the content to the client before it is requested, can result in a further reduction of the page loading time and an improved QoE for the end user. This is especially true for multimedia-rich websites which rely on over-the-top video streaming to deliver their content: recent work by Wei et al. and Huysegems et al. has shown that HTTP/2 allows to significantly reduce the startup time and end-to-end delay in live video streaming scenarios [10], [11]. These techniques can be further extended, for instance by prefetching future video segments in video streaming sessions.

In the next section, we explain how results from the news website analysis are used to select content eligible for PDN storage, allowing us to perform an initial evaluation of the proposed concept.

[4] https://www.netflix.com

IV. Content Placement in PDNs

As previously mentioned in Section III, content prefetching in PDNs is performed in a user-centric way, taking into account user activity and preference towards certain types of content. To apply this procedure to the deredactie.be news website, we use the results of the dataset analysis to decide upon which articles should be prefetched by a PDN delivery node. To determine whether or not an article is of interest to a particular user, three methods are used:

- Absolute number of requests: prefetch the article if it belongs to one of n categories with the highest amount of views by the user;
- Relative number of requests (n): push the article if it belongs to one of n categories with the highest ratio of viewed and published articles. Users can have a distinct interest towards certain categories, but more articles are published for certain categories than for others. Dividing the number of article views within a category by the amount of articles pushed within this category, can allow to increase the consumption rate of prefetched content;
- Relative number of requests (θ): prefetch the article if the ratio of the relative amount of viewed and published articles is higher than a certain threshold θ. As an example, when 12% of articles are published within a certain category, and 24% of the user's requests are on articles within this category, this ratio equals 2.

Each of these methods results in a different amount of prefetched articles, depending on the number of articles published within the relevant news categories. Considering more article categories thus results in more prefetched content, and thus network traffic. In the next section, performance for the three methods is evaluated in a PDN-based scenario. Results are compared with a CDN-based approach, in which several relevant reactive cache replacement heuristics are considered: Low Inter-Reference Recency Set (LIRS), Adaptive Replacement Cache (ARC), Least Recently Used (LRU), Least Frequently Used (LFU), Random Replacement (RR) and, as a baseline, Bélády's Optimal Replacement Algorithm (OPT).

V. Evaluation and Discussion

To perform a basic evaluation of the concepts discussed above, we use the last two months of the collected request logs to replay CDN- and PDN-enabled network scenarios. To this end, a basic simulation setup was made in Java, running on a 2-core Intel i5-4200U 1.60GHz CPU, 8 GB RAM with Ubuntu 14.04 LTS installed. Users are either connected to a shared CDN cache, or to a PDN delivery node. The setup in Figure 6 is used as a reference, where the average latency between client and origin server Δ_o is set to $100\,\text{ms}$, latency between client and CDN delivery node Δ_c is set to $20\,\text{ms}$ and latency between client and PDN cache Δ_p is set to $5\,\text{ms}$. These numbers correspond to a typical wired network scenario; note that more latency will be introduced in mobile core networks, having an impact on possible latency reductions.

In a first set of experiments, we assume that every user is connected to a CDN cache, using the cache replacement heuristics discussed in Section IV. The cache hit ratio is evaluated as a function of the cache size, expressed as the number of articles it can hold. As shown in Figure 7, using the optimal algorithm with a cache size of 8 allows to handle 50% of incoming article requests. At a certain point in time, a limited amount of articles are highly popular, while others are not. Using a cache size higher than 128 pages results in a relatively low gain for the cache hit ratio. This is directly related to the fact that the cache can hold (almost) all articles published that day (85.0 on average), and the analysis in Figure 2 has shown that an article receives most of its requests within the first 24 hours after publication. Performance for the LFU heuristic is lower than for random replacement, since the most popular articles are cached for the whole two-month period; this issue can be solved through the use of a sliding window, so that only recently published articles will be considered. Results for LIRS and ARC are similar, clearly outperforming the more straightforward LRU replacement heuristic. For LIRS, cache hit rates of 93.2% and 96.1% are obtained for a cache size of 256 and 1024 respectively.

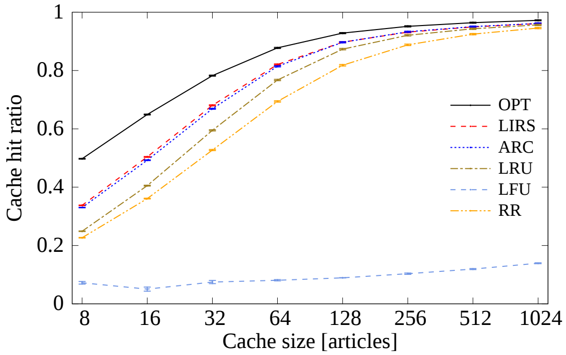

Figure 7. Cache hit ratio as a function of the cache size, for several cache replacement strategies.

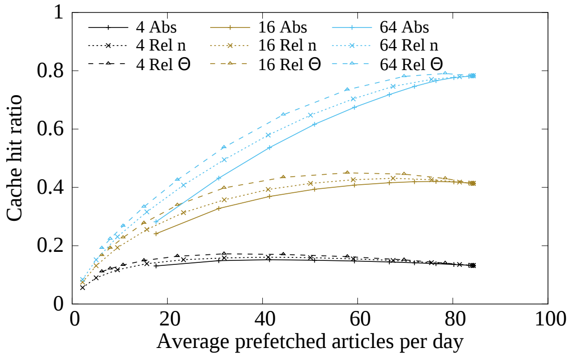

Figure 8. Cache hit ratio as a function of the average amount of prefetched articles, for multiple values of the cache size.

In a second set of experiments, we select a subset of 10,000 users which have visited the most articles within the ten-month period of data collection. These users each are assigned a certain amount of storage on a PDN delivery cache, where articles can be placed as soon as they are published. The first eight months of data are used for profile building, while the last two months are used to evaluate the cache hit ratio as a function of the cache size and the profiling metrics. For the first two methods described in Section IV, the parameter n is

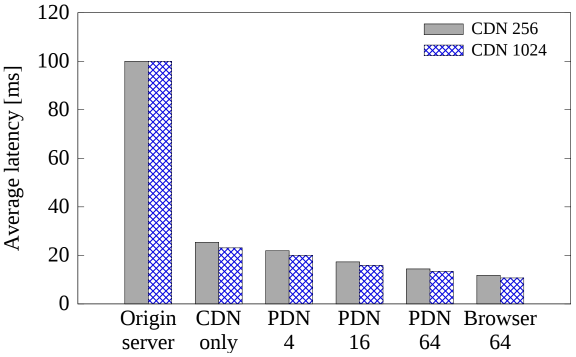

Figure 9. Average latency for different configurations, considering a cache size of 256 and 1024 for CDN, a per-user cache size of 4, 16 and 64 for PDN and a browser cache size of 64.

varied from 1 to 16. For $n = 1$, only articles published in the top category are prefetched, while for $n \to \infty$, all published articles are considered. As shown in Figure 8 for cache sizes of 4, 16 and 64, the relative metric outperforms the absolute one: fewer articles need to be prefetched by the PDN in order to achieve a similar cache hit ratio. For a cache size of 4 and 16, considering more categories has a negative impact on performance: relevant articles are wrongly removed in favor of less relevant articles, resulting in a lower cache hit ratio. For the third method, the parameter θ is varied from 0 to 2. For $\theta = 0$, all articles are considered, while for $\theta = 2$ only well-read article categories are considered. As shown in Figure 8, this metric outperforms the other two in terms of cache hit ratio versus transferred articles. Highest values for a cache size of 4, 16 and 64 are achieved for $\theta = 1.0$, $\theta = 0.6$ and $\theta = 0.2$ respectively, with respective hit rates of 17.2%, 45.0% and 79.0%. To limit the amount of articles transferred to the PDN, a value of $\theta = 1$ can be used for all configurations, resulting in an average of 31.9 articles prefetched per day and a cache hit ratio of 39.7% and 53.8% for a cache size of 16 and 64 respectively.

Figure 9, finally, shows results in terms of latency. When news pages are requested from the origin server, latency is always equal to Δ_o, or $100\,\mathrm{ms}$. Using a reactive LIRS-enabled CDN cache significantly reduces this delay, to an average of 25.4 and $23.1\,\mathrm{ms}$ for a cache size of 256 and 1024 respectively. Using a PDN allows to further reduce the average latency to values between $21.9\,\mathrm{ms}$ for a CDN cache size of 256 and a PDN cache of 4, and $13.4\,\mathrm{ms}$ for a CDN cache size of 1024 and a PDN cache size of 64. Results further improve when content stored by the PDN nodes is actively prefetched by or pushed to the client upon connecting to the news website. Assuming a negligible latency in this case, the proposed approach can result in an average latency as low as $10.7\,\mathrm{ms}$.

VI. Conclusions and Outlines for Future Work

In this paper, we analyzed a large dataset containing user requests from deredactie.be, a major Belgian news provider, focusing on content popularity, user activity and user preference towards certain news categories. Based on the analyzed data, the concept of personal delivery networks (PDNs) was introduced, which store content close to the end user, at delivery caches in the edge of the core network or even in the access network. PDN nodes can proactively prefetch and evict content on a per-user basis, opening opportunities for personalized low-latency delivery of multimedia-rich web applications. Applying this concept to the news website, we used three initial content placement techniques to decide whether or not newly pushed articles should be prefetched by the PDN, and showed that the proposed approach can result in a significant latency reduction.

Many possibilities exist to further expand the evaluated scenario. While our focus was on one major news website, multiple websites require more elaborate profiling logic, dynamically partitioning the PDN cache to hold content for each website. Different sources also provide different types of information, content and metadata, and tend to use different categories to label their articles; although the latter can be solved through the use of the IPTC media topic standard. Whereas user profiling in this paper is performed mainly through article categories and a user's preference towards them, extensions can focus on more elaborated web prediction algorithms using article attributes such as the title, author(s), time of publication, named entities etc. Other interesting research paths include the application of an online learning algorithm to cope with changing user activity, and an extension to the cache replacement algorithms to take into account different content sources such as HTML, CSS, images and video.

Acknowledgment

Jeroen van der Hooft is funded by grant of the Agency for Innovation by Science and Technology in Flanders (IWT). This research was performed partially within the iMinds PRO-FLOW project (150223).

References

[1] Google, "Global Site Speed Overview: How Fast Are Websites Around The World?" 2012. [Online]. Available: http://analytics.blogspot.be/2012/04/global-site-speed-overview-how-fast-are.html

[2] S. Egger, T. Hofeld, R. Schatz, and M. Fiedler, "Waiting Times in Quality of Experience for Web-Based Services," in *International Workshop on Quality of Multimedia Experience*, 2012.

[3] Conviva, "Viewer Experience Report," 2015.

[4] Amazon, "Amazon CloudFront Documentation," 2016. [Online]. Available: https://aws.amazon.com/documentation/cloudfront/

[5] S. Van Canneyt, P. Leroux, B. Dhoedt, and T. Demeester, "Towards a Data-Drive Online News Publishing Strategy [submitted]," 2016.

[6] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in *ACM Workshop on Mobile Big Data*, 2015, pp. 37–42.

[7] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web Data Extraction, Applications and Techniques: A Survey," *Computing Resource Repository*, vol. 70, pp. 301–323, 2014.

[8] M. Belshe, R. Peon, and M. Thomson, "Hypertext Transfer Protocol Version 2," RFC Editor, Tech. Rep. Internet-Draft, 2015. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-httpbis-http2/

[9] Y. Elkhatib, G. Tyson, and M. Welzl, "Can SPDY Really Make the Web Faster?" in *IEEE/IFIP Networking Conference*, 2014, pp. 1–9.

[10] S. Wei and V. Swaminathan, "Low Latency Live Video Streaming over HTTP 2.0," in *ACM Network and Operating System Support on Digital Audio and Video Workshop*, 2014, pp. 37:37–37:42.

[11] R. Huysegems, J. van der Hooft, T. Bostoen, P. Alface, S. Petrangeli, T. Wauters, and F. De Turck, "HTTP/2-Based Methods to Improve the Live Experience of Adaptive Streaming," in *ACM Multimedia Conference*, 2015, pp. 541–550.