To Cache or not to Cache: The 3G case

Jeffrey Erman Alexandre Gerber Mohammad T. Hajiaghayi
Dan Pei Subhabrata Sen Oliver Spatscheck
AT&T Labs - Research

ABSTRACT

Recent studies have shown that in the wired broadband world, caching of HTTP objects results in substantial savings in network resources. What about cellular networks? We examine the characteristics of HTTP traffic generated by millions of wireless users across one of the world's largest 3G cellular networks, and explore the potential of forward caching. We provide a simple cost model that third parties can easily use to determine the cost-benefit tradeoffs for their own cellular network settings. This is the first large scale caching analysis for cellular networks.

1. INTRODUCTION

Cellular networks have witnessed tremendous growth recently. For instance, one major US wireless carrier claimed to have experienced a growth of 5000% in its data traffic over 3 years, while a network equipment manufacturer [2] predicts mobile data traffic will grow at a compound annual growth rate (CAGR) of 108% between 2009-2014. Despite this rapid growth, fueled by the proliferation of smartphones, laptops with mobile data cards and new technologies improving the performance of cellular networks, there is still a rather limited understanding of the protocols, application mix and the characteristics of the traffic being carried. For wireline broadband traffic, recent studies have shown that the new killer app traffic is HTTP again [13, 10] and forward caching is a promising content delivery mechanism [10]. What about cellular networks?

Our flow level data set, collected over a 2-day period in March 2010 in a large US wireless carrier region covering multiple states and millions of subscribers, shows that HTTP traffic accounts for 82% of the average downstream traffic. HTTP being the killer protocol should not come as a surprise. Indeed, if HTTP has become the workhorse of various applications, ranging from video streaming to data downloads [13, 14] on broadband wireline access links, there are no reasons why it should be different for traffic generated by these same computers when they use 3G cards instead. HTTP dominating cellular data traffic naturally raises the question of the potential for HTTP forward caching. Proposed first in the 1990s for achieving improved client performance and reduced network cost, a forward cache is an HTTP cache is deployed within an Internet Service Provider's (ISP) network for caching all cacheable HTTP traffic accessed by its customers. In contrast to CDNs, a forward cache is deployed for the customers benefit and under the control of the ISP, rather than for the benefit of the content owner. Would that technology make sense for cellular networks?

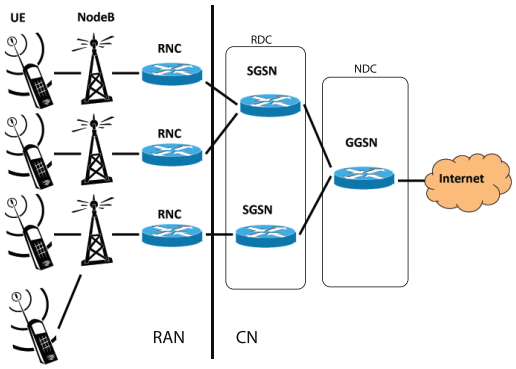

Figure 1: 3G Architecture

We first review the typical architecture of a 3G network. Figure 1 shows a typical Universal Mobile Telecommunication System (UMTS) data network architecture. Contrary to a typical wireline architecture that is relatively flat, cellular networks are highly centralized. A User Equipment (UE) goes through the Radio Access Network (RAN) first to the Node B, and then to the Radio Network Controller (RNC) to reach the core network (CN). The CN consists of Serving GPRS Support Nodes (SGSN) and Gateway GPRS Support Nodes (GGSN). The SGSN converts the mobile data into IP packets and send them to the GGSN through the GPRS Tunneling Protocol (GTP). The GGSN serves as the gateway between the cellular core network and the Internet. This means every IP packet sent to a UE has to go through a GGSN. Multiple SGSNs are stored in Regional Data Centers (RDC) and GGSNs are collocated in National Data Centers (NDC). This centralized architecture is ideal for forward caching. Therefore, it is worth studying the tradeoff between the network cost reduction and the additional cost of having the caches in cellular networks.

One might argue that improving the performance of the core network via caching won't make a significant difference to end-end latency, given the high latency of today's 3G RAN networks. This is a short term issue: the next generation of cellular networks (Long Term Evolution (LTE)), which are currently being deployed,

plan to provide RAN latencies under 10 msec [1].

Having made the case why it makes sense to study forward caching in 3G networks, we first characterize the properties of the cellular HTTP traffic amenable to caching in Section 2. We develop a cost model that considers different resource costs involved and computes the cost of using forward caching at different levels in a 3G network hierarchy (see Section 3). This can be used by network designers in performing the cost-benefit analysis of deploying forward caches in their network and determining the appropriate caching solution for their situation. Our results show:

- At the NDC level, the cache hit ratio for the overall population of UEs is 33%.
- Cache hit ratios increases as the UE population size increases. For sizes of $10K$ or more, different randomly selected UE populations of the same size exhibit significant and similar cache hit ratios.
- Using our caching cost model we show that in the regime where in-network caching leads to cost savings, caching at the RDC is the most beneficial.

2. TRAFFIC ANALYSIS

2.1 Data Collection

We collected HTTP request and response headers at the interface between the GGSNs and SGSNs in a large 3G wireless network in North America over a 2 day period in March 2010. During this period, we observed millions of UEs including laptops, smartphones, and regular cellphones making billions of HTTP requests. To preserve subscribers' privacy, we used a secure hash function (MD5) to hash the URL, host header and nonces into a request identifier.

Since traffic on the SGSN-GGSN interface is encapsulated using the GTP protocol, we were also able to directly extract from the encapsulation header, the NDC, GGSN, RDC and SGSN through which a particular HTTP request was served. Unfortunately, our collection method did not allow us to collect the size of the object returned for a particular request and we report our results in terms of requests only.

2.2 Forward Caching Background

We first introduce some forward caching background. When a HTTP request arrives from a client, the cache directs the request to the web server (called origin server) if the request indicates that the client wants a fresh copy of the requested object. Otherwise, the cache checks whether it has a local copy of the object. If not, the request is retrieved from the origin server. If yes, the cache checks if it is stale (TTL expired). If not stale, the cache serves the request locally from disk. If stale, the cache sends a "if-modified-since" request to the origin server, and serve the object locally if the origin server answers no or receive the object from the origin server if it has changed.

When the request is received from the origin server the cache will serve the object to the client. If the object indicates it is cacheable, the object is also written to the local disk.

2.3 Data Characterization

We next show some highlights characterizing the data used in this study.

As we do not have reliable object sizes, we have chosen to use just an unlimited cache size for all our evaluations to understand the maximum obtainable benefits. Experiments in this section were all conducted using data from Thursday March 25, 2010,15:00 GMT to Saturday March 27, 4:00 GMT, containing many billions of HTTP requests.

The first experiment looks at the cache hit ratio across all the requests. This is the same as if caching at the NDC level. At the end of the period, the hit ratio was stable at 33.4%. The amount of non-cacheable objects was 31.3% of the overall requests. If the non-cacheable requests are excluded, 48.7% of the cacheable objects are served from cache. While there is no previous work for 3G traffic, previous caching work on wirelines networks [11, 4] and our own [10] found cache hit ratios which range from 30%-49%. The majority of the earlier results when taking into account all the non-cacheable content are on the lower end of that range which is similar to our findings here.

We also investigated the number of requests per object which as expected follows a Zipf distribution. To compare this result to prior work we used the zipf R library [3] to fit a Zipf distribution to the data. The zipf distribution has an α of 0.88. [15] includes a comparative study of multiple papers [7, 12, 5, 8, 9, 6] which performed similar studies more than 10 years ago. These papers reported α values between 0.64 and 0.83 when object popularity is measured inside the network and close to 1 when measured on the end devices. The difference can possibly be explained by the fact that end devices cache popular object themselves (e.g. browser cache) and, therefore, less requests for these objects are visible in the network. Our distribution is a bit higher than the high end of the α range for in-network measurement. This difference could be due to changes in traffic composition in the years since those older studies were performed. It could also be due to the fact that 3G end devices have less resources then a desktop computer and, therefore, are somewhat less likely to cache all objects they might request in the future. On the other hand, the fact that the α is still not as close to 1 as prior end user studies have shown might

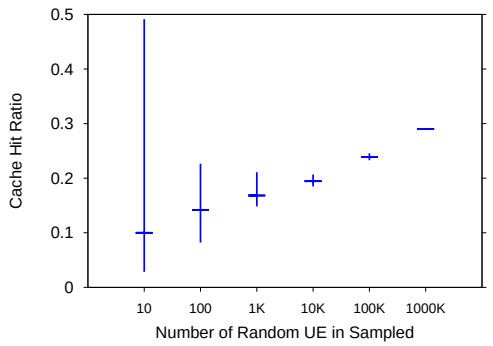

Figure 2: Cache hit ratio for different randomly chosen UE sizes.

indicate that even mobile end-devices already perform some level of local caching.

To evaluate the cache hit ratio as the number of UEs that use a cache increases, a set of experiments was conducted. The number of UEs chosen for an experiment was varied from 10 to 1 million increasing by an order of magnitude at each step. The UEs were chosen randomly from a list of several million UEs that were active during the evaluation period. For sample sizes below 10000, the experiment was repeated 50 times and all others repeated 10 times. Figure 2 shows the minimum, median, and maximum values obtained for the experiments.

The most interesting part of the result is how the cache hit ratios increase with consumer population size. As we were able to analyze data for such a large population, these results will actually allow us to guide future network designs. For example, if a 4G network is being planned and caching is being considered during the design phase either for cost or performance reasons, these numbers can guide the carrier to plan the correct number of aggregation points to allow for efficient caching.

Looking at the results, it is not surprising that the cache hit ratio has high variance for populations below 10000 subscribers. However, above that population size, it seems that caching is similarly beneficial to all randomly chosen populations of a given size. This would indicate that caches deployed for populations of 10000 or more will have predictable benefits. Not surprisingly the cache hit ratio increases as the population size increases, and we see the increasing trend even with 1 million UEs, suggesting additional caching gains even at this high level of population aggregation. There is however a diminishing returns trend with increasing UE size - that would be more apparent if the x-axis were on a linear instead of a logarithmic scale.

3. CACHING ANALYSIS

3.1 Caching Model

We model all wireline costs of delivering data traffic in a 3G network to SGSN, and exclude the radio network costs and the wireline cost from GGSN to UE from our analysis. This is reasonable since we do not consider changing the caching on the UE itself and as such the excluded costs is not impacted by any caching schema deployed within the included wireline network.

We number the different levels in the 3G network hierarchy, see Figure 1, in increasing order of depth with an NDC, GGSN, RDC, and SGSN at levels $1, 2, 3, 4$ respectively. Let e_i denote the number of instances/locations (e.g., NDCs, GGSNs, etc.) at level i. We shall consider caching at a single level in the 3G network hierarchy, with a forward cache deployed at every location in the selected level i ($1 \leq i \leq L = 4$). We also assume that a UE stays under the same SGSN from when an object is requested to its delivery time. We assume unlimited cache size and processing power per cache, but cost proportional to cache size and processing throughput.

Let n be the network-wide total number of requests from all UEs over the time interval of interest. For the forward cache at the j^{th} location at level i, we count the following: (i) requests arriving from UEs which are leaves of the subtree rooted at this location ($n_{i,j}$), (ii) number of these requests that are for cacheable objects but which require fetching the requested object from its origin server ($n_{i,j}^{src}$), (iii) requests for which the requested objects are served from the cache ($n_{i,j}^{cache}$), and (iv) requests for which the cache has a stale copy of the object (based on object validity timestamps) and therefore needs to send an if-modify-since request to the origin server to check if a new copy of the object needs to be downloaded ($n_{i,j}^{ifmod}$). Note that for some of the $n_{i,j}^{ifmod}$ requests, the source server will indicate that the cached version of the object is still valid, in which case the actual object will be served to the UE from the local cache. Finally let $n_{i,j}^{u}$ denote the total number of unique cacheable objects requested over the observation time interval. Let p denote the mean size (bytes) of a requested object, and q be the mean overhead (in bytes) associated with servicing an if-modified-since request.

We need the following cost metrics per byte of traffic. At the caching infrastructure, s: disk storage, c: CPU usage, d: disk bandwidth. Let $b_{i,i+1}$ and t respectively be the (i) bandwidth-mile cost per byte on the network path between 2 adjacent levels i and $(i+1), (1 \leq i \leq L-1)$ in the 3G network and (ii) the transit cost per byte that the 3G operator pays to its upstream provider network. Define $B_{l,m}$ to be the bandwidth-mile cost per byte on the network path between levels l and m ($1 \leq l < m \leq L$). Then, $B_{l,m} = \sum_{k=l}^{m-1} b_{k,k+1}$.

3.2 Cost Analysis: Caching at level i

The overhead of serving the requests using caching at level i is $O_i^{cache} = \sum_{j=1}^{e_i} O_{i,j}^{cache}$, where $O_{i,j}^{cache}$ is

the cost of serving requests arriving to the part of the 3G network being served by the j^{th} cache at level i. $O_{i,j}^{cache} = R_{i,j} + N_{i,j} + T_{i,j}$, where $R_{i,j}$ is the resource usage (disk storage, disk bandwidth and CPU usage) at the caching system, $N_{i,j}$ is the cost of serving the objects over the 3G network, and $T_{i,j}$ is the transit cost that the 3G operator pays to its upstream provider network. For ease of exposition, we define the corresponding network-wide cost components across all the caches at level i: $R_i = \sum_j R_{i,j}, N_i = \sum_j N_{i,j}, T_i = \sum_j T_{i,j}$. Then,

$$O_i^{cache} = R_i + N_i + T_i \quad (1)$$

We next compute each of these components.

Computing N_i: $N_{i,j}$ is the sum of the following:

1. The bandwidth-mile cost $(n_{i,j} - n_{i,j}^{cache}) * p * B_{1,i}$ on the network path between the NDC and the caching server, incurred when requested objects need to be fetched from the origin server, either because an object was not in the cache or because the cache had a stale copy that needed to be updated.

2. The additional bandwidth-mile cost $n_{i,j}^{ifmod} * q * B_{1,i}$ incurred by the if-modified-since requests (the cost of actual object download is already accounted for in 1 above) on the network path between the NDC and the caching server.

3. The bandwidth-mile cost $n_{i,j} * p * B_{i,L}$ incurred on the network path from the caching location down to the UEs[1] Every request contributes to this cost.

Let $V^i = \sum_j n_{i,j} * p.$ and $V_1^i = \sum_j (n_{i,j} - n_{i,j}^{cache}) * p + \sum_j n_{i,j}^{ifmod} * q$. Here V_1^i is the total traffic volume carried between the origin servers and the caches at level i. V^i is the total traffic corresponding to all the incoming requests to the network. Since this is independent of the caching level, we shall drop superscripts and use V to refer to this term in the remainder of the paper. Then

$$N_i = \sum_j N_{i,j} = V_1^i * B_{1,i} + V * B_{i,L} \quad (2)$$

Transit T_i: The transit cost $T_{i,j}$ for traffic between the NDC and the provider network is incurred for (i) objects that need to be fetched from the origin server, and (ii) for servicing if-modified-since requests. $T_{i,j} = ((n_{i,j} - n_{i,j}^{cache}) * p + n_{i,j}^{ifmod} * q) * t$. Then

$$T_i = \sum_j T_{i,j} = V_1^i * t \quad (3)$$

Computing R_i: $R_{i,j}$ is the sum of the following:

1. The storage overhead cost $n_{i,j}^u * p * s$ at the cache.

2. The disk bandwidth cost $(n_{i,j}^{cache} + n_{i,j}^{src}) * p * d$ at the caching system used for reading (for a cache hit) or writing (for a new cacheable object or replacing an old object with an updated version) to the disk.

3. The CPU and system bus overhead at the caching system $= n_{i,j} * p * c$.

Let $V_2^i = \sum_j n_{i,j}^u * p$, and $V_3^i = \sum_j (n_{i,j}^{cache} + n_{i,j}^{src}) * p$. V_2^i and V_3^i are the total volume of traffic stored at the level i caches and corresponding to requests for cacheable objects, respectively. Then

$$R_i = \sum_j R_{i,j} = V_2^i * s + V_3^i * d + V * c \quad (4)$$

3.3 Caching Benefits

In the absence of caching in the 3G network, each request incurs the cost of traversing the entire 3G hierarchy from the NDC to the UE and of transiting the 3G-upstream provider interface. As pointed out before, this only includes the wireline costs between the cache and SGSN as the network from SGSN to UE t is excluded. The total cost for serving the requests can be computed as

$$O^{nocache} = n*p*B_{1,L} + n*p*t = n*p(B_{1,L}+t) = V*(B_{1,L}+t) \quad (5)$$

and the caching benefit at level i is $O^{nocache} - O_i^{cache} = (V, V_1^i, V_2^i, V_3^i) * (-c + B_{1,i} + t, -t - B_{1,i}, -s, -d)^T$, (note that $B_{1,i} = -B_{i,L} + B_{1,L}$). This equation is basically the product of one traffic pattern vector (V, V_1^i, V_2^i, V_3^i) and one cost parameter vector. Table 1 shows the traffic pattern vector (normalized by V to keep the data confidentiality), computed for over a billion HTTP requests arriving over a 12 hour period to the large North American 3G provider described earlier. This realistic traffic pattern vector can be plugged into formulas to evaluate caching benefits at different levels for a different network where the traffic data are not readily available.

The formula for caching benefit at level i can be rearranged to the following form such that it can be seen that all cost parameters have linear impacts on the total cost when other parameters are fixed: $(-V, -V_2^i, -V_3^i, V - V_1^i, V - V_1^i) * (c, s, d, t, B_{1,i})^T = (-V, -V_2^i, -V_3^i, V - V_1^i, V - V_1^i) * (c, s, d, t, \sum_{k=1}^{i-1} b_{k,k+1})^T$, as shown in second last column of Table 1. It is also apparent that none of the parameters has dominating impact on the total cost because the constants of the factors are close to each other (range from 0.23 to 1).

3.4 Simplifying cost parameters

As the number of cost parameters is large, we now introduce a practical approach to simplifying them where it makes sense to do so. We note that, although different networks probably have different cost parameter values, our simplification approach below can apply to

[1]This only includes the wireline costs between the cache and the UE as the radio network cost is excluded.

Table 1: V_x^i normalized by V, caching benefits, and saving percentage

i	V_1^i	V_2^i	V_3^i	caching benefits	saving percentage
1	0.7037	0.2349	0.6890	$-c - 0.2349 * s - 0.6890 * d + 0.2973 * t$	$18.27 - 0.1641 * (c/b_{1,2})$
2	0.7151	0.2470	0.6890	$-c - 0.2480 * s - 0.6890 * d + 0.2849 * (t + b_{1,2})$	$17.53 - 0.1704 * (c/b_{1,2})$
3	0.7336	0.2719	0.6876	$-c - 0.2719 * s - 0.6876 * d + 0.2664 * (t + b_{1,2} + b_{2,3})$	$26.62 - 0.1726 * (c/b_{1,2})$
4	0.7639	0.3122	0.6859	$-c - 0.3122 * s - 0.6859 * d + 0.2361 * (t + b_{1,2} + b_{2,3} + b_{3,4})$	$23.61 - 0.1765 * (c/b_{1,2})$

other networks, and the parameters in our studied network are realistic.

Recall our goal is to evaluate the relative caching benefits at different levels i, which are essentially determined by the ratios of the cost parameters. We first assume a consistent cost unit.

We further assume that the computation cost $c = y$ cost units per Mbyte, and the network cost is z cost units per Mbyte per mile. We now show that practically all other parameters can be approximated by a constant times y or z.

We assume the storage cost $s = 1.3 * y$ cost units per MByte and the storage bandwidth cost $d = 1.3 * y$ cost units per MByte. This is a rough estimate in our context based on some commercial CPU/storage prices at some cloud computing providers. We assume $b_{1,2} = z$ and $b_{3,4} = z$ since the GGSN is physically located in the NDC and the SGSN is physically located in the RDC and we use 1 mile to approximate the distance. We specify the transit cost $t = 800 * z$ since 800 is a typical value for average route miles of traffic in North America. Similarly we specify $b_{2,3} = 500 * z$, assuming the typical NDC-RDC distance to be 500 miles - this is realistic as all US based providers have a very small number of NDCs in the US. Using these realistic assumptions we can reduce the independent cost parameters to just c and $b_{1,2}$. The cost saving percentages of caching at level i over the noncaching solution (cost=$V * (B_{1,L} + t) = 1302 * b_{1,2}$), after the above simplification, is shown in the last column of Table 1, for the traffic demand in our data.

3.5 Results and Sensitivity Test

The cost saving percentage is essentially a function of the caching level i and $c/b_{1,2}$, the ratio of the computation cost to the network bandwidth cost. We now briefly analyze how the caching saving percentage changes with these two parameters. A positive value represents savings whereas a negative value represents additional costs when caching is deployed. When the $c/b_{1,2}$ value is large, e.g., 1000, (i.e., computing cost is significantly higher than network bandwidth cost) caching solutions at all levels cost more than a non-caching solution. When $c/b_{1,2}$ decreases to respectively to 111.33, 102.87, 154.23, and 133.77, the savings correspondingly turn positive for the NDC, GGSN, RDC, and SGSN levels. When $c/b_{1,2}$ becomes very small, such as 0.01, the network cost is significantly higher than the computing cost, and the percentage savings become approximately 26.62%, 23.61%, 18.27%, 17.53% at the RDC, SGSN, NDC, GGSN levels, respectively. Generally speaking, when $c/b_{1,2}$ is small, caching at the RDC is most beneficial. Caching at RDC and SGSN is more beneficial than at GGSN and NDC, due to saving of the network cost $b_{2,3}$. RDC caching is more beneficial than SGSN caching because the former has a better cache hit ratio although it also has the additional cost of (relatively small) $b_{3,4}$. For similar reason, NDC caching is more beneficial than GGSN caching.

These results indicate that in the studied case, network costs need to be quite high before caching becomes beneficial from a financial perspective. However, a second benefit of caching is improved performance, e.g., reduced latency, for the user. We are currently evaluating the impact of caching on user experience.

4. CONCLUSION

We explored the potential of forward caching in 3G cellular networks by using traffic traces generated by millions of users from one of the world's largest 3G cellular networks. We found that the cache hit ratio is 33% when caching at NDCs, and as the population size grows from 10 to 1 million UEs, the cache hit ratio increases but exhibits diminishing additional benefits for larger populations. We developed a caching cost model that shows the tradeoffs between deploying forward caching at different levels in the 3G network hierarchy. By simulating the caching model at each network element on our large data set, we provide a set of parameters that can be used to calculate the benefit based on any network cost parameters. In our case study, we found caching at RDCs is the most beneficial with a 26.7% savings in cost. However, we also found that for a wide range of network cost parameters, caching is not that beneficial and, in fact, can cost significantly more than delivering the objects from the origin servers.

Building on this work, we see a number of opportunities for future research. One area we plan to explore is benefits of reduced latency achieved from caching.

5. REFERENCES

[1] 3GPP TR 25.913 V7.3.0: Requirements for Evolved E-UTRA and E-UTRAN. `http://www.arib.or.jp/IMT-2000/V720Mar09/5_Appendix/Rel7/25/25913-730.pdf`, 2006.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2009-2014. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html, February 2010.

[3] zipfR: user-friendly LNRE modelling in R. http://zipfR.R-Forge.R-project.org/, 2010.

[4] R. Aceres, F. Douglis, A. Feldmann, G. Glass, and M. Rabinovich. Web Proxy Caching: The Devil is in the Details. In *Workshop on Internet Server Performance*, 1998.

[5] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira. Characterizing reference locality in the www. In *PDIS*, pages 92–103, 1996.

[6] M. F. Arlitt, R. Friedrich, and T. Jin. Workload characterization of a web proxy in a cable modem environment. *SIGMETRICS Performance Evaluation Review*, 27(2):25–36, 1999.

[7] M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants. In *SIGMETRICS*, pages 126–137, 1996.

[8] P. Barford, A. Bestavros, A. Bradley, and M. Crovella. Changes in web client access patterns: Characteristics and caching implications. *World Wide Web*, 2(1-2):15–28, 1999.

[9] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM*, pages 126–134, 1999.

[10] J. Erman, A. Gerber, M. T. Hajiaghayi, D. Pei, and O. Spatscheck. Network-Aware Forward Caching. In *WWW'09*, Madrid, Spain, 2009.

[11] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. In *IEEE/ACM Transactions on Networking*, 1998.

[12] S. Glassman. A caching relay for the world wide web. *Computer Networks and ISDN Systems*, 27(2):165–173, 1994.

[13] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *IMC'09*, Chicago, USA, 2009.

[14] G. Maier, F. Schneider, and A. Feldmann. A First Look at Mobile Hand-held Device Traffic. In *PAM'10*, 2010.

[15] M. Rabinovich and O. Spatschek. *Web caching and replication*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.