Optimised local caching in cellular mobile networks

Åke Arvidsson a,⇑, Attila Mihály b, Lars Westberg a

aEricsson AB, SE-164 80 Stockholm, Sweden
bEricsson Kft., Irinyi J. u. 4-20, HU-1117 Budapest, Hungary

ABSTRACT

Motivated by the problem of increasing backhaul transmission costs in cellular mobile networks we examine the potential of highly distributed caching solutions. In particular we propose to overcome the problem of poor hit ratios in such solutions by forming caching domains which support pooling (the ability to fetch content from other caches when necessary) and equalisation (active transfers between caches to preload popular content). The characteristics of such a scheme are investigated by applying a new analytical model to a set of realistic examples. The model is verified against both computer based simulations and measurements in a real network, and it is found that the proposed scheme can reduce backhaul bandwidth requirements for an average cell inside a domain by a factor of 7 for web-like traffic and by a factor of 45 for video-like traffic, while the corresponding factors outside a domain amount to 41 and 997 respectively.

1. Introduction

Data traffic in mobile cellular networks is growing rapidly; in August 2010 it was noted [1] that mobile broadband accounts for 10% of the subscriptions, 50% of the volume and grows ten times faster than voice such that it nearly tripled last year. Similarly, in 2009 O2 (UK) reported that its mobile data traffic doubled every three months in 2009, TIM (Italy) announced a growth of 216% from mid-2008 to mid-2009 and AT&T (US) reported a growth of 5000% over the past 3 years [2]. Moreover, this growth is expected to continue for the foreseeable future; the Cisco Systems forecast for 2009–2014 [2] predicts a compound annual growth rate of 108% in general and 131% for video in particular, which means that about 66% of the mobile data traffic in 2014 will be video.

On the technology side, new radio technologies continue to provide higher bandwidths hence we have, e.g., WCDMA with HSPA, which offers 42 Mbps downlink (DL) and 12 Mbps uplink (UL), and LTE, which offers 160 Mbps DL and 50 Mbps UL [3]. The current challenge is, however, the backhaul where the cost of replacing typical present interfaces, e.g., E1s/T1s over copper or microwave, with new high speed interfaces, e.g., Ethernet over fibre, often is prohibitively high. These problems are particularly pronounced for sites which combine remote locations with heavy loads; a typical example is primary aggregation sites, aggregation level one in Fig. 1.

In this paper we will examine the extent to which caches can solve this dilemma. In Section 2 we discuss caching in a general context and highlight some problems which are particular to mobile networks whereas in Section 3 we describe a possible solution to these problems. The performance of our proposed solution is examined in Section 4 by means of mathematical models, computer simulations and real measurements. Finally we sum up our findings and outline some further work in Section 5. Two concluding appendices describe our mathematical models; Section A deals with cache performance calculations and Section B deals with content popularity distributions.

2. Preliminaries

A cache typically stores (copies of) popular content such as, e.g., web objects and video tracks and handles requests...
for such content by either, if the requested content is stored, sending it to the requesting user or, if the content is not stored, forwarding the request to a different provider such as, e.g., the original source or another cache, storing (a copy of) the response and then sending it to the requesting user.\(^1\) The first case is referred to as a cache "hit" and the second case is referred to as a cache "miss".

The cache hit ratio may be defined as the fraction of requests that result in hits and caches are said to be more efficient the higher this ratio. Caches can only store limited amounts of content hence one needs rules, also known as a caching strategies, to decide which content to insert or delete. The hit ratio of a cache depends not only on the cache itself (in terms of the caching strategy and the amount of memory) but also on the content (in terms of content sizes and relative popularities).

The caching strategy is the set of rules used to insert new content and delete old content such that the available memory is used as efficiently as possible. A simple example of such a strategy is to insert all new content as it is requested and, when necessary, to delete the least recently requested content. More details on this may be found in, e.g., [4].

The relative popularity characterises how often certain content is requested in relation to certain other content. A common model in this context is the Zipf model or variants thereof. For the rth most popular item (where r thus is a strictly positive integer), the request rate is proportional to \(r^{-1}\) in the pure Zipf model, to \(r^{-\beta}\) (where \(\beta\) is a strictly positive real) in the modified Zipf model and to \((r + k)^{-\beta}\) (where \(k\) is a strictly positive integer) in the Zipf–Mandelbrot model. In the literature we note that, e.g., a pure Zipf model is used in [5] to model web objects, a modified Zipf model is used in [6] to model on-demand video tracks and a Zipf–Mandelbrot model is used in [7] to model corporate video tracks. We also note that the heavy tails of these distributions imply considerable difficulties in measuring \(\beta\) and \(k\) and that, as discussed in e.g., [8], this may explain the large range of models and parameters suggested in the literature. In particular we note the relationship between log-normal distributions and Zipf distributions, cf. [9] and the discussion in [10].

Another important consequence of the heavy tails is the fact that the heavier the load on the cache, the higher the hit ratio. To see this, note that requests for content can be seen as samples from the popularity distribution and the more samples, the better the estimate. The linkage between high request rates and high hit ratios is also driven by the opportunity to deliver stored items many times before they become unpopular.

\(^1\) Content may also be stored in advance based on, e.g., expected popularity of certain types of content or business agreements with certain providers of content.
cle with a “C”) and which keeps track of all caches and all requests in its domain (in this case the entire network). Site A also has a gateway to the internet (green box with two arrows ”−”).

A request for content from a user at, e.g., site A11 in Fig. 2 will thus be handled by the cache at that site. If there is a hit the cache will deliver the content to the user and report the event to the controller. If there is a miss the cache will forward the request to the controller. The controller will then arrange a delivery and instruct the cache how to handle it. The delivery arrangement is based on lists of which content can be found where (the controller will know this as it handles all misses) and on current link loads (the controller and the cache may monitor link loads) while the caching instructions will be based on accurate popularity statistics (the controller knows all requests in the entire domain).

The controller may also instruct caches to proactively copy content between themselves. By means of these preloads the hit ratios of caches in “leaf” sites (e.g., aggregation level 0 in Fig. 1) can approach those of caches in more central sites (e.g., aggregation level N in Fig. 1). The transfers typically take place during off peak periods, e.g., at night, using gentle protocols, e.g., one that yields more to congestion than TCP and/or using low priority channels, e.g., worse than best effort in DiffServ. Potentially popular content missing in, e.g., cache A11 in Fig. 2 may thus be sent from caches A12, A1 or A depending on where it can be found and on current link loads.

Finally the controller may instruct caches at “transit sites” to intercept and cache passing traffic. A delivery to, e.g., site A11 in Fig. 2 may thus be cached at site A (if delivered from the internet), at site A1 (if delivered from the internet or from site A) and at site A11 (in all cases). In this way caches at more central sites will be more up-to-date and thus act as “backups” to those at less central sites. Such layering can be modified into complete hierarchical structures of primary, secondary, tertiary caches etc. with a common controller or, e.g., one controller per level. As an example, site A in Fig. 2 may belong to a first level, sites A1 and A2 may belong to a second level and sites A11, A12, A21 and A22 may belong to a third level. One can also use “overlapping” levels where caches belong to more than one level.

It is to be remarked that the total number of disks need not be much higher in distributed solutions than in central ones. The reason for this is that central solutions require higher bandwidth which often is achieved by trading off memory utilisation, cf. [11].

Our proposal forms a hierarchical and distributed structure of caches which share content and prefetch by equalisation with the aid of a central controller. With respect to web caching, hierarchical structures were first studied in [12], central entities in such structures in [13], prefetching in [14] and distributed architectures in [15]. With respect to cellular networks, many works, e.g., [16,17] consider performance enhancing proxies with or without explicit references to caching, and centralised caching as a complement to enhanced client caching is discussed in [18]. Thorough surveys of early works related to cooperative caching are found in, e.g., [19,20] and a very recent survey of works related to caching and prefetching is found in [21].

### 4. Performance

We will now examine the performance of our proposed solution. In Section 4.1 we introduce two content models referred to as “web” and “video” respectively, in Section 4.2 we obtain some results for these models and compare those results to real measurements, and in Section 4.3 we consider some general characteristics of our scheme and test our conclusions on data from a real network.

#### 4.1. Scenario

We consider Ω pieces of content subject to U editorial updates per day and A new arrivals per day. A cache is assumed to receive R requests per day and to store all content it has seen. Our proposed scheme is applied over a domain of C caches with nightly equalisations of the ι most popular items. Apart from the two extremes ι = 0 (no equalisation) and ι = Ω (full equalisation) we speak of “balanced” equalisation when the ratio η between night loads (cache equalisations to increase hit ratios) and day loads (cache misses resolved by pooling or other means) is set such that background traffic (equalisations) is maximised without disturbing foreground traffic (misses). A simple rule reads

$$\eta \approx \frac{\lambda - \lambda}{\lambda - 1}$$

where λ is the peak rate and λ is the mean rate. To see this, note that the numerator is the bandwidth available to background traffic while the denominator represents the bandwidth occupied by foreground traffic.

We measured the traffic per 15 min interval over three weeks in 318 cells of a real cellular network after which we for each cell computed the peak traffic, the average traffic and the ratio between peak and average. Ordering the cells by their peak-to-average ratios, Fig. 3 shows the the peak-to-average ratios λ/λ (black curve) and the corresponding average rates λ (grey curve). (The averages are renormalised to fit the scale of the diagram.) It is noted that there is a wide spread of peak-to-average ratios, from 860 to 1.8 (cf. the range of the black curve) and that cells with high peak-to-average ratios tend to have low average traffic and vice versa (cf. the opposite slopes of the two curves). With reference to Eq. (1) we thus note that η spans the range 0.8–859 with the highest values for the least loaded cells.

The first content model, which is supposed to represent web traffic, assumes that we have Ω = 1,600,000 web objects with relative popularities following the pure Zipf distribution (cf. p. 3) similar to [5,8]. The diagram to the left in Fig. 4 depicts the relative rate at which an object is requested vs. its popularity rank (rank one denotes the most popular object, rank two the second most popular object etc.). It is seen that the request rate drops with the popularity rank as a straight line with slope −1 in a log-log scale. Moreover, it is assumed that U = 32,000 web objects are updated per day and that the popularity of an updated object can be modelled as a sample from a modified Zipf distribution (cf. p. 3) with β = 0.075. The number of
updated objects is set to give an average life time of 50 days, as found in [22], and the popularity distribution is set to give a correlation of 0.10 between change rate and user popularity, a worst case of the results in [23]. We assume that new versions are requested at the same rate as old versions, and that old versions are not requested at all. Finally, it is assumed that \( A = 16,000 \) new web objects are added per day and that the initial popularity of a new object can be modelled as a sample from the same modified Zipf distribution. The number of new objects is set to give a weekly growth rate of 5–8%, as found in [24], and the popularity distribution is reused in absence of other, relevant information. The diagram to the right in Fig. 4 depicts the resulting popularity decay averaged over 1000 simulation experiments and it is seen that the rate at which a web object is requested drops over time. Note that the diagram refers to specific objects which means that the request rate in this sense drops to zero after an update.

The second content model, which is supposed to represent video traffic, assumes that we have 16,000 video tracks with relative popularities following the Zipf–Mandelbrot distribution (cf. p. 3) with \( \beta = 0.75 \) and \( k = 80 \) similar to [7]. The diagram to the left in Fig. 5 depicts the relative rate at which an object is requested vs. its popularity rank (as before rank one denotes the most popular object etc.). It is seen that on the order of the \( k \) most popular video tracks are approximately equally frequently requested while the remaining video tracks rapidly become less frequently requested. These characteristics are in perfect agreement with, e.g., [7]. Moreover, it is assumed that 16 new video tracks are added per day and that the initial popularity of a new track can be modelled as a sample from the same Zipf–Mandelbrot distribution. The diagram to the right in Fig. 5 depicts the resulting popularity decay averaged over 1000 simulation experiments, and it is seen that the popularity of a video track decays over time and that the decay accelerates over time. These characteristics are in perfect agreement with, e.g., [25].

The domain model considers \( C = 100 \) caches with \( R = 10,000 \) and \( R = 100 \) requests per cache for web and video respectively.

### 4.2. Results

Applying the analytical model in Sections A and B and a simulation model written from scratch in C++ using the results of Section B to the two content models and the domain model above, we get the results depicted in Figs. 6 and 7 and Table 1. In more detail, we consider full equalisation (\( \epsilon = 0 \)), balanced equalisation (with \( \epsilon \) set to render \( \eta = 1 \), cf. Eq. (A.20)) and no equalisation (\( \epsilon = 0 \)).

Fig. 6 shows cache hit ratios, cf. Eq. (A.9). It is seen that full equalisation (dashed) results in high hit ratios, no
equalisation (dotted) results in low hit ratios while balanced equalisation (dash-dotted) represents an intermediate case. It is also seen that pooling (solid) leads to very high hit ratios and we note that the agreement between analysis (curves) and simulations (crosses) is perfect.

Fig. 6. Hit ratios for web (left) and video (right). Curves and crosses indicate calculated results and simulated results respectively.

Fig. 7 shows the number of transferred objects during days (○) and nights (●), cf. Eqs. (A.7) and (A.8) respectively. It is seen that full equalisation (dashed) results in very low day time traffic and very high night time traffic, no equalisation (dotted) leads to high day time traffic and no night
time traffic while balanced equalisation (dash-dotted) makes day time traffic equal to night time traffic. Again we note that the agreement between analysis (curves) and simulations (crosses) is perfect.

Table 1 shows, for the three equalisation levels, the corresponding $\ell$-values, cf. Eq. (A.20), and the calculated steady state values for hit ratios, day traffic and night traffic, cf. Eqs. (A.19), (A.17) and (A.18) respectively. We see that the tabulated values and the curves in Figs. 6 and 7 are in perfect agreement.

Comparing the dynamics of the two content models in Figs. 6 and 7, we note that the curves for pools and full equalisation are about the same in both cases (steady state is reached almost immediately) while the curves for balanced equalisation and no equalisation converge differently (steady state is reached in about 35 days for the web case but in about 70 weeks for the video case). The duration of the transient period (with lower hit ratios and higher traffic loads) depends on the request rate $R$ (more requests shorten transients), the number of updates $U$ and arrivals $A$ (more updates and arrivals shorten transients) and on the popularity distribution (steeper slopes $\beta$ shorten transients while the plateau $k$ has little impact).

In most realistic cases, however, we expect networks to be “over-dimensioned” initially (to cope with growing traffic) hence we may assume that short initial peaks (as in the web case) do not pose a problem while long initial peaks (as in the video case) may have to be mitigated through initial, off-line or out-of-band preloading of content. Equalisations can also be made over the air between base stations with overlapping coverage.

Finally we compare our results against measurements of YouTube requests in a real network. The characteristics of YouTube traffic have been studied extensively, e.g., [25] and we refrain from repeating this but simply investigate the performance of our scheme when applied to the “raw data”.

The results are shown in Fig. 8 with hit ratios (left) and traffic volumes (right). It is immediately seen that the hit ratios in Fig. 8 are qualitatively the same as those in Fig. 6; full equalisation (dashed) results in high cache hit ratios, no equalisation (dotted) results in low cache hit ratios and pooling (solid) results in high domain hit ratios. Similarly, it is immediately seen that the traffic volumes in Fig. 8 are qualitatively the same as those in Fig. 7; “All” refers to full equalisation while “Rpt” refers to balanced equalisation in the sense that only those items that were requested more than once during the measurements were equalised, and we note that the volumes of All (Full) are larger than those of Rpt (Balanced) and that both curves drop with time. It is also noted that our measurement period is too short to reach steady state.

4.3. Characteristics

We now examine the steady state gains in terms of hit ratios and traffic reductions as functions of the bandwidth available for equalisation in the scenarios in Section 4.1 and some variants thereof.

- $R$ indicates a variant with double request rate and this case also represents different placements; in Fig. 2 we have that, e.g., A1 and A2 see three times the rates of A11, A12, A21 and A22 (their “own” requests plus the ones of their “subordinates”).
- $C$ indicates a variant with double number of caches and this case represents larger pools.
- $U, A$ indicate a variant with double updates and double arrivals and this case represents more dynamic content.
- $\Omega, U, A$ indicate a variant with double content number, double updates and double arrivals and this case really represents more content as double updates and double arrivals merely maintain constant dynamics.

The hit ratios, cf. Eq. (A.19), are shown in Fig. 9; solid curves indicate the original models and dashed curves indicate variants. In general we note, firstly, that the more

![Fig. 8. Hit ratios (left) and data transfers (right) of YouTube traffic in a real network.](image-url)
transmission resources available to equalisation, the higher the hit ratio and, secondly, that this applies to all variants (cf. the higher cache hit ratios obtained for higher background-to-foreground ratios). We also note that generous equalisations can be "saturated" (cf. the right part of the curves) and that tight equalisations give small returns (cf. the left part of the curves and Table 2).

The traffic reductions inside the domain (i.e. the reductions due to equalisation), cf. Eq. (A.21), are shown in Fig. 10; as before solid curves indicate the original models and dashed curves indicate variants. The general observations are the same, viz., firstly, the more transmission resources available to equalisation, the bigger the traffic reduction and, secondly, this applies to all variants (cf. the higher traffic reduction factors obtained for higher background-to-foreground ratios). Again we also note that generous equalisations can be "saturated" (cf. the right part of the curves) and that tight equalisations give small returns (cf. the left part of the curves and Table 3).

Finally we compare the different cases and note that:

- The high gain in $R$ is largely attributed to the fact that more requests at day time allow for more equalisations at night time while the total request rate in the pool, and thus the selection of objects to equalise, is the same in the $R$ case as in the $C$ case.
- The reduced gain in $\Omega, U, A$ and $U, A$ are about the same and largely an effect of the fact that there are more new objects. A further case (not shown) where we also scale $R$ gives results which are about the same as for the original case; i.e. the higher rate of requests allows for enough equalisations to make up for the higher numbers of updates and arrivals.

The traffic reductions outside the domain (i.e. the reductions due to pooling), cf. Eq. (A.21) with $C=1$ cache, are...
shown in Table 4. It is immediately noted that the gains are significant in all cases; the “miss ratios” in the pool (not shown) are very low, the order of magnitude is $10^{-2}$ for web and $10^{-3}$ for video, and this results in very high traffic reductions.

Applying our findings to the measurements depicted in Fig. 3 we find that 50% of the cells permit $\eta > 10$ in which case caching with equalisation would boost traffic reductions by a factor of 3 or more for web and 10 or more for video (compared to caching without equalisation) and 95% of the cells permit $\eta > 2$ in which case these factors amount to about 2 or more for web and 3 or more for video.

It is also noted that the “negative correlation” between peak-to-average ratios and average rates seen in Fig. 3 suggests a “positive correlation” between the need for a boost (low hit rates) and the ability to provide it through equalisation (high peak-to-mean ratio). To see this, first note that cells with low average traffic (small values of the grey curve in Fig. 3) can be expected to have low request rates and thus to exhibit poor hit ratios, and then note that these cells tend to have high peak-to-average ratios (large values of the black curve in Fig. 3) and thus to offer the greatest potential for equalisation.

The above hypothesis was tested by applying our two content models to the example in Fig. 3. We included all $C = 318$ cells in the domain and set the request rate at each cell proportional to its measured average rate, and such that the average request rates were kept fixed at $R = 10,000$ and $R = 100$ for web and video respectively. Finally we determined balanced equalisation levels for each cell by solving for the equalisation level $\xi$ that equals the traffic balance in Eq. (A.20) to the proposal in Eq. (1), using the peak-to-average ratios seen in the measurements.

The results are shown in Fig. 11 (with cells ordered by their measured average rates) and we see, firstly, that the hit ratios (left diagrams) are increased considerably by balanced equalisation (dash-dotted) compared to no equalisation (dotted) and, secondly, that the bandwidth savings (right diagrams) are increased by balanced equalisation (dash-dotted) compared to no equalisation (dotted). In particular, we note that equalisation almost equals the hit ratios in all cells except for the last few ones, with very low average rates, which become problematic and for which it is realised that more aggressive equalisation, and hence less pronounced traffic reductions, may be required. (Alternatively, these cells may be left as they are since their low average rates suggest that they are rather insignificant.)

Summing up, balanced equalisation in an average cell in Fig. 3 will, in the web case, increase the hit ratio from 0.62

<table>
<thead>
<tr>
<th>Model</th>
<th>–</th>
<th>$R$</th>
<th>$C$</th>
<th>$U, A$</th>
<th>$\Omega, U, A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>28</td>
<td>50</td>
<td>50</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>Video</td>
<td>627</td>
<td>1253</td>
<td>1253</td>
<td>314</td>
<td>314</td>
</tr>
</tbody>
</table>

Fig. 11. Results for the example in Fig. 3 web (top) and video (bottom); hit ratios (left) and traffic reduction factors (right). The cells are ordered by their average rates.
to and 0.85 and the traffic reduction factor from 3.00 to 7.33 and, in the video case, increase the hit ratio from 0.67 to 0.96 and the traffic reduction factor from 7.26 to 45.5. Moreover, pooling will reduce the traffic outside the domain by a factor of 41 and 997 for web and video respectively.

5. Conclusions and further work

We proposed a distributed caching scheme where caches are organised in domains with pooling (fetching content from other caches in the domain) and equalisation (redistributing popular content between caches in the domain). The scheme is particularly suitable in cellular mobile networks where increasing backhaul transmission costs is a growing problem. We also examined the characteristics of the scheme by applying a newly developed analytical model to two different content models and verified its correctness by comparing to computer based simulations and measurements in a real network. Applied to a real network, the results indicate that the traffic may be reduced inside the domain by a factor of 7 for web-like traffic and by a factor of 45 for video-like traffic, thanks to equalisation, and outside the domain by a factor of 41 for web-like traffic and by a factor of 997 for video-like traffic, thanks to pooling.

Future work of interest includes applying longer measured traces to more realistic caching algorithms and further investigations of “continuous equalisation” where controllers constantly maintain updated queues of redistribution candidates which are served in some order of priority as free bandwidth becomes available.

Appendix A. Cache performance

Consider a cache with content “sorted” in order of popularity such that entry \( r \) corresponds to the piece of content with popularity rank \( r \) and let a correct copy be called valid \((V)\) and expired or non-existent copy be called stale \((S)\). The status of an entry will change over time as a result of requests, equalisations, updates and arrivals. Fig. A.12 shows a daily cycle starting with an initial (reference) status and over the day it is seen how requests refresh \( r - 6 \) and \( r + 5 \), equalisations refresh \( r - 5 \) and \( r + 3 \), updates preclude \( r - 3 \) and \( r + 4 \) and arrivals preclude \( r - 4 \) and \( r + 2 \). In particular, it is seen how arrivals cause “right shifts” of the caching status as, e.g., the arrival of a new \( r - 4 \) causes the old \( r - 4 \) to become \( r - 5 \) and so on. The final status on day \( t \) will then be the initial (reference) status on day \( t + 1 \) and so on.

We now derive a state based model of a cache entry at the reference points. First consider \( r = 1 \) and let \( P_V^r(t) \) and \( P_S^r(t) \) denote the probabilities that at reference point \( t \) the entry of \( r \) is valid and stale respectively. Noting that

- a valid entry will stay valid if it is not subject to updates or arrivals and will become stale otherwise and that
- a stale entry will become valid if it is subject to requests or equalisation but not subject to updates or arrivals and will remain stale otherwise

and omitting the references to \( r \) we may write

\[
P_V^r(t + 1) = P_V^r(t)q_0q_4 + P_S^r(t)(p_R + p_E)q_0q_4, \quad (A.1)
\]

\[
P_S^r(t + 1) = P_V^r(t)[1 - q_0q_4] + P_S^r(t)[1 - (p_R + p_E)q_0q_4]. \quad (A.2)
\]

where \( p_R, p_E, p_U \) and \( p_A \) denote the probabilities that the content with rank \( r \) is subject to request, equalisation, update and arrival respectively,

\[
p_R = 1 - [1 - f_R(r)]^R,
\]

\[
p_E = 1 - [1 - f_E(r)]^{R(C - 1)}[1 - f_R(r)]^R,
\]

\[
p_U = 1 - [1 - f_U(r)]^U,
\]

\[
p_A = 1 - [1 - f_A(r)]^A,
\]

\[
q = 1 - p \quad \text{and} \quad f_R(r), f_E(r) \quad \text{and} \quad f_A(r) \quad \text{denote the probability density functions with respect to rank for requests, updates and arrivals respectively, cf. Section B.}
\]

To understand these expressions consider, e.g., the one for \( p_R \) and note that \( f_R(r) \) is the probability that content \( r \) is requested, \( 1 - f_R(r) \) is the probability that content \( r \) is not requested, \( [1 - f_R(r)]^R \) is the probability that content \( r \) is not requested in \( R \) attempts and \( 1 - [1 - f_R(r)]^R \) is the probability that content \( r \) is requested at least once in \( R \) attempts. Similar reasoning applies to the other expressions.

Next consider \( r > 1 \) and recall that arrivals cause old entries to be “exported” to lower ranks and new entries to be “imported” from higher ranks (cf. the “right shifts” in Fig. A.12). Noting that

- a valid entry will become stale if it is subject to an arrival or if an imported entry is stale and stay valid otherwise and that
- a stale entry will stay stale if it is subject to an arrival or if the imported entry is stale and become valid otherwise

and letting \( I_{V_r}^r(t) \) and \( I_{S_r}^r(t) \) denote the probabilities that at reference point \( t \) the imported entry to \( r \) is valid and stale respectively we may write

\[
P_V^r(t + 1) = q_A I_{V_r}^r(t), \quad (A.3)
\]

\[
P_S^r(t + 1) = q_A I_{S_r}^r(t) + p_A \quad (A.4)
\]
where $r$ again is omitted to improve readability.

Finally, consider any $r$ and let $\Pi_W^r(t)$ and $\Pi_S^r(t)$ denote the probabilities that at reference point $t$ the exported entry from $r$ is valid and stale respectively. Again omitting $r$ we may write

$$\Pi_W^r(t) = P_W(t)q_U + P_S(t)(p_k + p_E)q_U,$$

$$\Pi_S^r(t) = P_W(t)p_U + P_S(t)[1 - (p_k + p_E)q_U].$$

(4.5) to (4.6)

Entries imported to $r$ correspond to entries exported from $r'$; from Fig. A.12 it is seen that $r' < r$ and that the distance between $r$ and $r'$ depends on the number of arrivals in the range $1, \ldots, r - 1$. We refrain from a detailed calculation but simply approximate the distance by its expectation, $AF_{\alpha}(r)$ where $F_{\alpha}(r)$ is the cumulative distribution function with respect to rank for arrivals, cf. Section B. In more detail $r'$ must be an integer and we set $r' = \text{max}(r - \lfloor AF_{\alpha}(r) \rfloor, 1)$. Formally we thus write

$$\Pi_W^r(t) \approx \Pi_W^{\text{max}(r - \lfloor AF_{\alpha}(r) \rfloor, 1)}.$$  

(4.7)

$$\Pi_S^r(t) \approx \Pi_S^{\text{max}(r - \lfloor AF_{\alpha}(r) \rfloor, 1)}.$$  

(4.8)

The computational procedure is thus to step through $r = 1, \ldots, \Omega$, apply Equations (A.5) and (A.6) using the existing probabilities and then compute new probabilities from Equations (A.1) and (A.2), for $r = 1$, or (A.3) and (A.4), for $r > 1$.

We now readily obtain the number of cache misses $M$ as the expected number of request driven transitions from stale entries to valid entries

$$M(t) = \sum_{r=1}^{\Omega} P_S(t, r)p_k(r).$$

(4.9)

the number of equalisations $E$ as the expected number of equalisation driven transitions from stale entries to valid entries

$$E(t) = \sum_{r=1}^{\Omega} P_S(t, r)p_k(r).$$

(4.10)

where the upper index reflects the upper limit on equalisation. The cache hit ratio $h(t)$ may then be expressed as

$$h(t) = 1 - \frac{M(t)}{R},$$

(4.11)

i.e. one less the cache miss ratio, and the balance as

$$\eta(t) = \frac{E(t)}{M(t)}.$$  

(4.12)

Other metrics may also be obtained.

Of particular interest is the steady state solution which is obtained by taking the limits in Eqs. (A.1)–(A.6) as $t \to \infty$ and solving the resulting equalities with the help of normalisation constraints. Thus, from Eqs. (A.1) and (A.2) and $P_W + P_S = 1$ we get

$$P_W^\infty = 1 - P_S^\infty,$$

$$P_S^\infty = \frac{1 - q_A}{1 - (p_k + p_E)q_A},$$

(4.13) to (4.14)

similarly we get from Eqs. (A.3) and (A.4)

$$P_W^\infty = 1 - P_S^\infty,$$

$$P_S^\infty = 1 - q_A \Pi_W^\infty,$$

(4.15) to (4.16)

and from Eqs. (A.5) and (A.6)

$$\Pi_W^\infty = 1 - \Pi_S^\infty,$$

$$\Pi_S^\infty = 1 - P_W - P_S(p_k + p_E)q_U.$$  

(4.17)

we may express the steady state hit ratio as

$$h = 1 - \frac{M}{R},$$

(4.18)

the steady state balance $\eta$ as

$$\eta = \frac{E}{M},$$

(4.19)

and the steady state traffic reduction $\rho$ at balanced equalisation as

$$\rho = \frac{R}{M},$$

(4.20)

which says that out of the $R$ requests only the $M$ misses remain. Note that, with balanced equalisation Eq. (1), the $E = \eta M$ equalisations are “for free” as they use “idle gaps” in the traffic.

Appendix B. Zipfian distributions

The Zipf–Mandelbrot distribution with parameters $k$ and $\beta$ (which reduces the modified Zipf distribution if $k = 0$ and to the pure Zipf distribution if $k = 0$ and $\beta = 1$) states that $f(r)$, the relative request rate of the $r$th most popular item, is proportional to $(r + k)^{-\beta}$.

To obtain a closed form expression for the cumulative probability function $F(r)$ we consider the more tractable, continuous version [26] which states that the relative request rate $\phi(r)$ is proportional to $\int_{s=r}^{s+r} (s + k)^{-\beta} ds$.

$$F(r) = \int_{s=0}^{s=r} \frac{D}{(s + k)^{\beta}} \approx \Phi(r) = \int_{s=0}^{s=r} \frac{C}{(s + k)^{\beta}} ds$$

$$= \begin{cases} \frac{C}{\ln(1 + \beta)} \left( \ln(k + r + 1) - \ln(k + 1) \right), & \beta \neq 1, \\ \frac{C}{\ln(1 + \beta)} \ln(k + r + 1), & \beta = 1, \end{cases}$$

(4.21)

where the normalisation constants $D$ and $C$ are defined as

$$D = \left( \sum_{r=1}^{\Omega} \frac{1}{(r + k)^{\beta}} \right)^{-1}$$

$$C = \left( \int_{r=1}^{\Omega} \frac{1}{(r + k)^{\beta}} dr \right)^{-1}.$$
with $2$ denoting the total number of items. The continuous version also enables simple sampling from Zipfian distributions by the inverse transformation method; if $u$ is a uniformly distributed number in the interval $[0,1)$, then $r$,

$$
r = \begin{cases} 
\left(\frac{(k + 1)^{1-\beta} + u^{1-\beta}}{c} - k\right)^{\frac{1}{1-\beta}} & \beta \neq 1, \\
(k + 1)^{\beta/c} - k & \beta = 1
\end{cases}
$$

is a Zipf–Mandelbrot distributed rank with parameters $\beta$ and $k$.

References


Ake Arvidsson obtained his M.Sc. and Ph.D. degrees in Electrical Engineering from Lund University, Sweden, in 1982 and 1990 respectively. He has worked with several consultancy companies and held various academic positions in Sweden and Australia. In 1998 he joined Ericsson and since 2008 he has been with Ericsson Research. His current research interests include content delivery, quality of experience and mobility modelling.

Attila Mihály obtained his M.Sc. degree in Physics at the Babes-Bolyai University in Cluj-Napoca (1993), Romania. He then got his Ph.D. in Physics at the University of Debrecen, Hungary (1999). Since 1999, he has been working at Ericsson Research in Hungary. His current research interests include wireless communications, QoS, content delivery and caching.

Lars Westberg obtained his M.Sc. and Lic.D. degrees from the Royal Institute of Technology Stockholm, Sweden, in 1990 and 1993 respectively. In 1977 he joined Ericsson and since 1995 he has been with Ericsson Research. His current research interests include content delivery and cloud computing in mobile networks.