ANALYTICAL MODEL OF HIERARCHICAL CACHE OPTIMIZATION IN THE NETWORK WITH UNBALANCED TRAFFIC

Lev B. Sofman
Frontline Communications
2221 Lakeside Blvd
Richardson, TX, USA
lev.sofman@ftr.com

ABSTRACT

Caching is important tool for reducing Internet access latency, network traffic and cost in content delivery networks (CDN). In case of hierarchical networks with several levels of caching (e.g., IPTV network), the problem is where and how much cache memory should be allocated in order to achieve maximum cost effectiveness. In this paper, we continue study of hierarchical cache optimization and present a model of hierarchical cache optimization for CDN with two levels of hierarchy and unbalanced traffic. This model depends on several basic parameters: traffic throughput for each low level node, cache effectiveness as a function of memory size, and cost parameters. Some reasonable assumptions about network cost structure and cache effectiveness function allow us to obtain an analytically optimal solution of the problem. We then analyze the factors that impact this solution.

Keywords: cache, traffic, network, cost, optimization.

1 INTRODUCTION

Many CDN networks have a hierarchical structure. For example, in IPTV network in metro area, at the top of the hierarchy is a Video Hub office (VHO) where all video contents are stored in video servers. VHO is connected to several Intermediate Offices (IO), every IO is connected to several Central Offices (CO), every CO is connected to several Access Nodes (AN), e.g. Digital Subscriber Line Access Multiplexers (DSLAMs), and every AN/DSLAM is connected to a number of subscribers.

In an IPTV network, Video on Demand and other video services generate large amount of unicast traffic from VHO to subscribers and, therefore, require additional BW/equipment resource in the network. To reduce this traffic (and overall network cost), part of video content (most popular titles) may be stored in caches closer to subscribers, e.g., in DSLAM, in service switches at COs, or in service routers at IO (Fig.1).

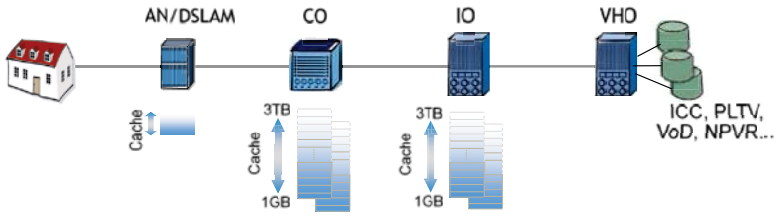

Figure 1: Hierarchical caching in IPTV network.

There is a tradeoff between cost savings for capacity on transmission links and costs for the caches (Krogfoss, Sofman, and Agrawal 2008). On the one hand, cache memory has a cost, but on another hand, caching of the content allows decreasing amount of traffic from upstream levels of this hierarchy and,

therefore decreasing of network cost. Overall, to find an optimal (in terms of network cost) location and amount of cache memory is a complex optimization problem.

In (Krogfoss, Sofman, and Agrawal 2008; Krogfoss, Sofman, and Agrawal 2009; Sofman and Krogfoss 2009) we assumed that network topology is a symmetrical tree and each low level node has the same amount of traffic (balanced traffic case). In this paper, we consider CDN network with unbalanced traffic. We solve the optimization problem for this model using analytical methods.

In the next Section, we review previous results on caching for multimedia services. In Section 3, we introduce concepts of cache effectiveness function and incremental (L, U)-cache that will be used in the study. In Section 4, we describe assumptions about cost parameters and cache effectiveness function that are used in our analytical model. Notations and mathematical formulation of the problem are presented in Section 5. We then describe methodology of solution and prove some important relationship between optimal cache parameters in Section 6. Sensitivity study in Section 7 investigates an impact of cache cost and the shape of cache effectiveness function on optimal cache size on every level of network. Section 8 summarizes results and contains some concluding remarks regarding importance of hierarchical caching for network cost optimization.

2 RELATED WORK

Content caching is one of the most effective solutions to improve performance and cost effectiveness of multimedia services. Reviews of caching techniques for multimedia services can be found in a number of publications (Ghandeharizadeh and Shayandeh 2007; Ghandeharizadeh et al. 2006; Chen et al. 2003; Cobarzan and Boszormenyi 2007; Lee and Park 2005). In particular, various caching techniques for different hit ratio metrics were discussed in (Ghandeharizadeh and Shayandeh 2007).

In (Krogfoss, Sofman, and Agrawal 2008; Krogfoss, Sofman, and Agrawal 2009), a hierarchical logical topology composed of different levels (e.g. AN/DSLAM, CO, IO, and VHO)) was considered, and network cost optimization was performed by choosing the best level where to put each content. One approach to solve this problem was presented in multi-level multi-service cache optimization model and implemented in internal optimization tool. In this model, network dimensioning was performed on very detail level: for a given topology (number of ANs/DSLAMs, COs and IOs), traffic throughput (video service characteristics), configuration and cost parameters, equipment dimensioning on every level of hierarchy was done and optimal cache architecture was found using some heuristic algorithm.

In another approach, a simplified analytical model for cache optimization was presented. The total network cost in this model was the sum of transport cost and memory cost, where the transport cost on every link was proportional to the rate of traffic that traverses the link and the memory cost was a linear function of the memory size. In this model, a minimum of the total network cost assumed to be at a stable point (i.e., where derivative of network cost is equal to zero). This model depends on basic parameters: traffic, topology, cache memory limitations, and cost parameters, e.g. memory cost per GB and unit network cost per unit of traffic on each level of hierarchy. Some reasonable assumptions allowed us receiving analytically optimal solution of the problem and analyze the factors that impact this solution.

In (Sofman and Krogfoss 2009), we refined the model (Krogfoss, Sofman, and Agrawal 2008; Krogfoss, Sofman, and Agrawal 2009), by considering so-called boundary cases (i.e., cases without cache or with maximum cache size) at each level of hierarchy. Network cost minimum assumed to be at a stable point or at a boundary point. As we demonstrated in (Sofman and Krogfoss 2009), boundary cases are important for understanding why in some cases optimal solution does not require caching at some levels, or requires using maximum allowed cache size on another levels. In the same time, boundary cases introduce significant complexity to the problem because they generate a large number of cases to be considered. In particular, it was demonstrated that optimal cache architecture may use caching at any combination of network hierarchy – DSLAM, CO and IO – depending on traffic, topology, and cost parameters, and therefore hierarchical caching is an important option for cost saving.

In Sofman, Krogfoss, and Agrawal (2008), the concept of content cacheability was introduced and a fast algorithm that uses cacheability to partition optimally a cache between several video services with different

traffic characteristics and content sizes was presented. The goal of the optimization was to serve maximum (in terms of bandwidth) amount of subscribers' requests subject to constraints on cache memory and throughput.

Caching algorithm is an important factor defining hit ratio and cache effectiveness. To make effective use of caching, a decision has to be made as to which documents should be evicted from the cache in case of cache saturation. Overview of various caching algorithms was presented in Balamash and Krunz (2004). Least Recently Used (LRU), Least Frequently Used (LFU), First in First out (FIFO) algorithms, various modifications and combinations of those algorithms are typically considered. Cache logic consisting of a modified LRU caching algorithm combined with one of three considered cache collaboration strategies (hierarchical, borrowing, and federated) were studied in Vleeschauwer and Robinson (2011).

3 CASHE EFFECTIVENESS FUNCTION AND INCREMENTAL (*L*,*U*)-CACHE

When we consider traffic in CDN it is important to distinguish between cacheable and non-cacheable content. Cacheable content is static information that does not change very often and can be cached. Non-cacheable content is dynamic information that changes regularly or for each user request and serves no purpose if it were cached. For example, linear TV or web pages that return the results of a search are non-cacheable, because their contents are unique almost all the time. We can define a portion of all traffic that corresponds to cacheable contents as P_{cache}.

Cache effectiveness is defined as the ratio of traffic (in terms of bit rate at busy hour) that is served from the cache to the total amount of requested traffic. Cache effectiveness is closely related to the concept of hit ratio which is define as a ratio of number of items served from the cache to the total number of requested items. Cache effectiveness depends on size and throughput of the cache, statistical characteristics of the traffic (e.g., popularity distribution of different titles, their bit rate, size in memory) and on caching algorithm. Under other equal conditions (the same statistical characteristics of the traffic and the same caching algorithm) cache effectiveness $H(m)$ is a function of memory size m. Cache effectiveness function increases with m; $H(0) = 0$, and $H(m)$ tends to P_{cache} when m increases (Fig. 2).

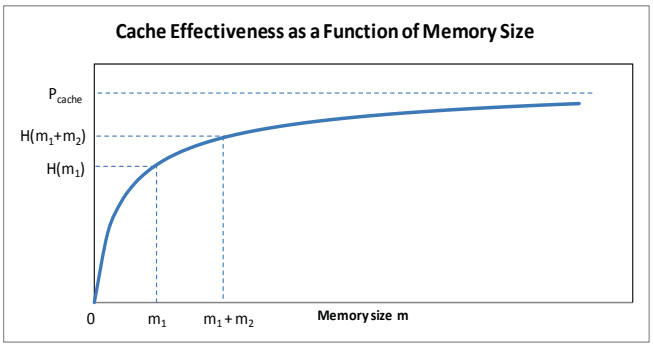

Figure 2: Cache Effectiveness function.

Note that because of effect of diminishing return, increase of memory size after some point makes insignificant impact on traffic and equipment cost reduction, and a total network cost (cache cost + equipment cost) will start to grow almost linearly with memory size. Therefore, there should be an optimal memory size where total network cost attains its minimum.

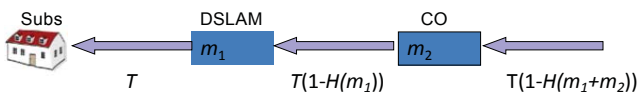

Figure 3: Cumulative effect of hierarchical caching.

In the hierarchical CDN, we can use a concept of "virtual" cache, introduced in Vleeschauwer and Robinson (2011). The "virtual" cache in any node of the tree is the actual cache at the node augmented by the cache

sizes in a node deeper (towards the leaves) in the tree, so called cumulative effect. A content residing in the "virtual" cache of the node (i.e., in the node itself or in a node further down the tree) reduces the traffic on the feeder link towards that node (and on all links towards the root of the tree). Fig. 2 and 3 illustrate a cumulative effect of hierarchical caching. If we allocate cache of size m_1 at DSLAM and cache of size m_2 at CO, part of the traffic will be served from the caches and amount of traffic between CO and DSLAM will be reduced by T x $H(m_1)$ comparing with original traffic demand T from subscribers. "Virtual" cache at CO includes an actual cache of size m_2 at CO augmented by cache of size m_1 at DSLAM. Upstream from DSLAM, traffic reduction will be equal to $TH(m_1)$; upstream from CO, traffic reduction will be equal to $TH(m_1 + m_2)$. Incremental traffic reduction at CO is equal to $T(H(m_1 + m_2) - H(m_1))$ and incremental cache effectiveness at CO is equal to $H(m_1 + m_2) - H(m_1)$.

In this example, cache at CO has two memory characteristics: lower bound $L = m_1$ and upper bound $U = m_1 + m_2$, and we will call it incremental (L, U)-cache, or simply (L, U)-cache. Note that incremental cache effectiveness depends not only on memory size $m_2 = U - L$, but on low and upper bound characteristics L and U. For the same memory size = $U - L$, incremental cache effectiveness, $H(U) - H(L)$, decreases with increase of L (effect of diminishing return).

In the following sections, the concept of (L, U)-cache will be used in our model of hierarchical cache optimization with unbalanced traffic to calculate traffic reduction on higher level of hierarchy. To visualize (L, U)-cache the following simplified example can be used (Fig. 4). When cache algorithm receives a request for content from a subscriber it makes a cache maintenance decision: to add a new content to the cache, to evict old content from the cache (if necessary), etc. Assume, that in our example the cache algorithm should maintain not one but two caches, of sizes L and U, $L < U$. Those items that will be cached in the cache of size U but not in cache of size L should belong to incremental (L, U)-cache. In simple terms, those items that are popular enough to be cached in cache U of larger size, but not popular enough to be cached in cache L of smaller size belong to (L, U)-cache.

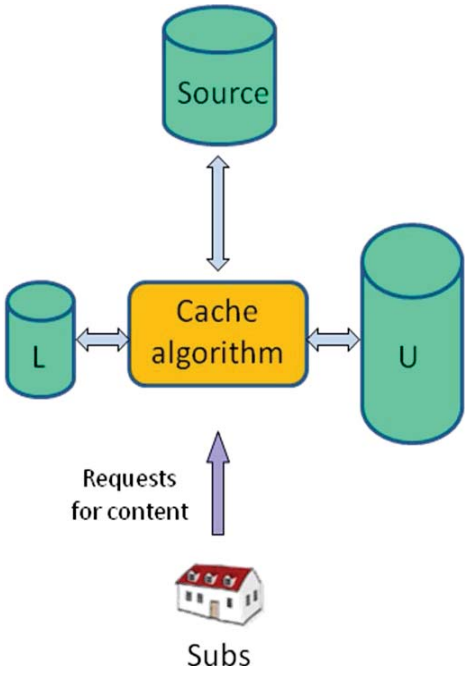

Figure 4: Explanation of (L, U)-cache.

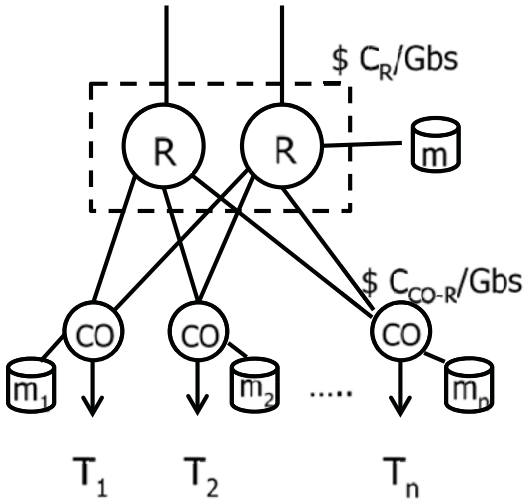

Figure 5: Hierarchical CDN with unbalanced traffic.

4 ASSUMPTIONS

In this study, we consider a CDN with two levels of hierarchy: several central offices (CO) on the low level are dual-homed to a couple of routers on the high level (Fig. 5). Each CO has its own amount of traffic demand (T_1, T_2, …). Every link between CO and the router is dimensioned for the whole amount of traffic

(protected mode) or for the half of the traffic (unprotected mode). As in (Krogfoss, Sofman, and Agrawal 2008), we assume that equipment cost is proportional to amount of traffic that traverses the equipment. More specifically, we estimate the equipment cost based on amount of traffic received by this equipment from upstream levels of network hierarchy (upstream traffic) and on amount of traffic send by this equipment downstream (downstream traffic). In order to calculate a total equipment cost, we define unit cost C_{CO} of network for the traffic between subscribers and CO level, unit cost $C_{CO\text{-}R}$ for the traffic between CO and router levels, and unit cost C_R for the traffic upstream from the router level. There is a cache at each CO as well as on router level; we assume that cost of the cache is linear function of its size (see more details in the next section). Total network cost includes equipment and cache cost. Our goal is to find optimal sizes of caches at each CO and at the router level that minimizes the network cost.

We assume that all COs have the same demographical profile in terms of traffic distribution and content popularity; in particular, all COs have the same cache effectiveness function $H(m)$.

Usually the size of cache memory occupied by one title is small compared with the size of the whole cache, so we can neglect a granularity factor and assume that function $H(m)$ is smooth, i.e. has a continuous derivative $H'(\mathrm{m})$ and this derivative decreases (effect of diminishing return). This condition implies that the function $H(m)$ is strictly concave – see Fig. 6. Function $H'(m)$ in some ideal case (when cache algorithm has complete information about statistical characteristics of the traffic) is equivalent to popularity curve, which, indeed, decreases as a function of the title's rank.

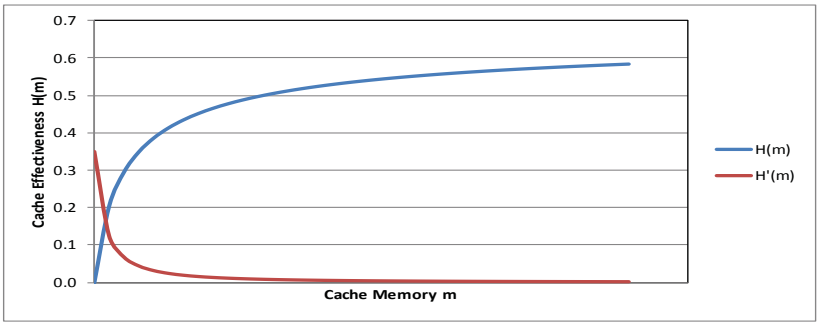

Figure 6: Cache effectiveness function and its derivative.

5 NOTATION AND MATHEMATICAL FORMULATION

5.1 Notations

N – a total number of COs in the network
T_i – traffic demand for i-th CO (Gbs), i = 1,…,N
P – protection factor: $P = 2$ in case of protected mode on CO-R link, otherwise, $P = 1$
M_i – maximum cache size for i-th CO (GB), $i = 1,\ldots,N$ – optional parameter
Trp_i – maximum cache throughput for i-th CO (Gbs), $i = 1,\ldots,N$ – optional parameter
M – maximum cache size for router (GB) – optional parameter

Cost parameters:

C_{CO} – network cost per unit of traffic on the link downstream from CO ($/Gbs)
$C_{CO\text{-}R}$ – network cost per unit of traffic on the link between CO and router ($/Gbs)
C_R – network cost per unit of traffic on the link upstream from router ($/Gbs)
CS_{CO} – memory cost per unit of storage at CO level ($/GB)
CF_{CO} – fix memory cost at CO level ($)
CS_R – memory cost per unit of storage at the router level ($/GB)
CF_R – fix memory cost at the router level ($)
$H(\mathrm{m})$ – cache effectiveness function

Decision variables:

m_i – cache memory size for i-th CO (GB), $i = 1,\ldots,N$
L and U – low and upper level of incremental (L, U)-cache at the router level (GB)

5.2 Mathematical Formulation

A total network cost, *NtwkCost*, may be calculated as

$$NtwkCost(m_1, m_2, ..., m_N, L, U) = C_{CO} \sum_{i=1}^{N} T_i + \sum_{i=1}^{N} C(m_i) + PC_{CO-R} \sum_{i=1}^{N} T_i (1 - H(m_i)) + C([L,U]) + C_R \sum_{i=1}^{N} T_i (1 - H(m_i, [L,U])) \quad (1)$$

where:

$C(m_i)$ is a cost of cache m_i at i-th CO:
$C(m_i) = CF_{CO} + CS_{CO}\ m_i$, if $m_i > 0$,
$C(m_i) = 0$, if $m_i = 0$, $i = 1,\ldots, N$

$C([L,U])$ is a cost of cache at the router (in the following, we assume that $L \le U$):
$C([L,U]) = CF_R + CS_R\ (U - L)$, if $L < U$;
$C([L,U]) = 0$, if $L = U$

$H(m_i, [L,U])$ is a factor of traffic T_i reduction upstream from the router level, it can be calculated as

$$H(m_i, [L,U]) = \begin{cases} H(m_i) + H(U) - H(L), & \text{if } m_i < L \\ H(U), & \text{if } L \le m_i < U \\ H(m_i), & \text{if } U \le m_i \end{cases} \quad (2)$$

The rationale for such calculation of the function $H(m_i, [L, U])$ is illustrated on Fig. 7. If, for example, $m_i < L$ (Fig.7 (a)), traffic reduction on the router level includes two components: traffic reduction $H(m_i)$ caused by cache at CO, and traffic reduction $H(U) - H(L)$ caused by cache at the router. In two other cases presented on Fig.7 (b) and Fig.7 (c), traffic reduction on the router level is defined by "virtual" cache of size U and m_i, correspondingly.

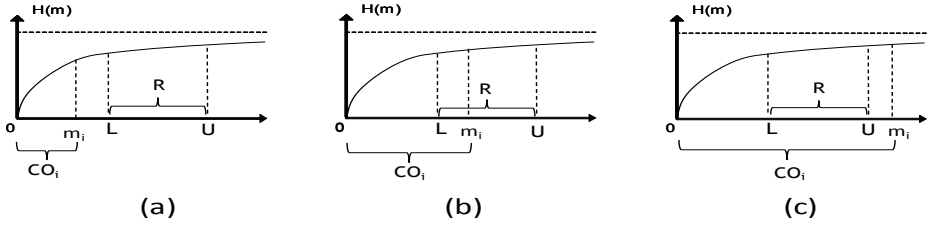

Figure 7: Calculation of traffic reduction upstream from the router level.

The goal is to minimize total network cost subject to constraints on cache memory size:

$$NtwkCost(m_1, m_2, ..., m_N, L, U) \to \min \quad (3)$$

such that $0 \le m_i \le M_i$,

$$T_i H(m_i) \le Trp_i,\ i = 1,2,...,N \quad (4)$$

and $0 \le L \le U$

Constraint optimization problem (3-4) has the following constraint factors: maximum size of memory M_i, maximum throughput Trp_i, parameters L and U. For fixed values of L and U, network cost (1) is a function of N variables $\vec{m} = (m_1,...,m_N)$ on a polyhedral defined by equation (4).

6 METHODOLOGY OF SOLUTION

Total network cost (1) can be partitioned into several components:

$$NtwkCost\,(m_1,m_2,...,m_N,L,U) = \sum_{i=1}^{N} NC(m_i,L,U) + C([L,U]) \tag{5}$$

where

$$NC(m_i,L,U) = C_{CO}T_i + C(m_i) + PC_{CO-R}T_i(1-H(m_i)) + C_RT_i(1-H(m_i,[L,U])) \tag{6}$$

is a cost component corresponding to i-th CO (i = 1, 2,…, N) and $C([L,U])$ is a cost of cache at the router. Note that each i-th cost component (6) depends only on one memory parameter m_i, memory size in the i-th CO. Therefore, in order to minimize (5) for a fixed values of L and U we can minimize each component (6) by m_i. Let m_i^o be an optimal value, i.e. $NC(m_i^o,L,U) = \min\{NC(m_i,L,U): 0 \le m_i \le \widetilde{M}_i\}$, where $\widetilde{M}_i = \min\{M_i, H^{-1}(Trp_i/T_i)\}$. We will use the following result from calculus.

Theorem. Let $f(x)$ be a lower semi-continuous function on an interval $[a, b]$, and has derivative $f'(x)$ on (a, b). Then one of the following should take place (Fig. 8):

(a) $f(x)$ attains its minimum at $x = a$
(b) $f(x)$ attains its minimum at $x = x_{\min}$, $a < x_{\min} < b$, and $f'(x_{\min}) = 0$ (i.e. $x_{\min}$ is a stable point)
(c) $f(x)$ attains its minimum at $x = b$

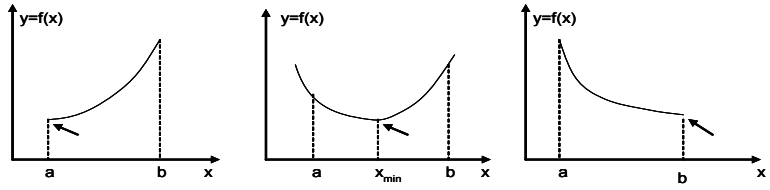

Figure 8: Minimum of function on segment.

Applying this theorem to functions $NC(m_i, L, U)$ (15) we can deduce that for every i = 1, 2,…, N the optimal value of m_i^o is either

- equal to one of the boundary values, 0 or $\widetilde{M}_i$, or
- one of the stable point(s) m_i^{st} whose value can be estimated from equation:

$$\frac{\partial(NC(m_i,L,U)}{\partial m_i} = 0 \tag{7}$$

Equations (2) and (7) allow us estimating a derivative of function H at a stable point:

$$H'(m_i) = CS_{CO}/(PC_{CO-R}T_i),\ \text{if } L \le m_i < U \tag{8}$$

$$H'(m_i) = CS_{CO}/((PC_{CO-R} + C_R)T_i),\ \text{if } m_i < L \text{ or } U \le m_i \tag{9}$$

From (8) and (9), we can calculate corresponding values of 1st and 2nd stable points, m_i^{st1} and m_i^{st2}. Depending on relationship between m_i^{st1}, m_i^{st2}, L and U, it can be one, two or no stable points at all.

As a result, using analytical approach, for a given values of *L* and *U,* we can find optimal values $(m_1^o, m_2^o, ..., m_N^o)$ that minimize the network cost (3) subject to constraints (4).

If $(m_1^o, m_2^o, ..., m_N^o, L^o, U^0)$ is an optimal solution of the problem (3-4) that uses incremental cache, i.e. $L^o < U^o$, then concavity of function *H* implies the following two statements regarding parameters L^o and U^o:

1. L^o should be equal to one of the optimal values for CO memories m_i^o, $i = 1,2,..., N$.
2. U^o should be equal to one of the following values:

- optimal values for CO memories m_i^o, $i = 1,2,.., N$,
- stable points, i.e. such values of *U* that

$$\frac{\partial NtwkCost(m_1^o, m_2^o, ..., m_N^o, L^o, U)}{\partial U} = 0$$

- boundary values, 0 or $U_{\max} = \min(\widetilde{U}, L^o + M)$, where $\widetilde{U}$ is defined from equation:

$$H'(\widetilde{U}) = CS_R / (C_R \sum_{i=1}^{N} T_i)$$

Using these observations, we can define a finite set of (*L*, *U*) pairs (candidate set) among which the optimal (L^o, U^0) pair can be found. We can calculate the optimal solution for each feasible (*L*, *U*) pair from the candidate set and then select the best solution over all such pairs.

7 SENSITIVITY STUDY

As a reference case, we consider a network with the following parameters:

$N = 4$ – total number of COs in the network with the following traffic demands: $T_1 = 50$ Gbs; $T_2 = 100$ Gbs; $T_3 = 200$ Gbs; $T_4 = 300$ Gbs.

$P = 1$, i.e. unprotected CO-R link.

$C_{CO\text{-}R}$ = \$6.5/Mbs– unit network cost on CO-R link.

C_R = \$4/Mbs– unit network cost on the link upstream from router.

$CS_{CO} = CS_R$ = \$22/GB – cost of memory per unit of storage at CO and at the router levels.

$CF_{CO} = CF_R$ = \$5,000 – fix memory cost at CO and at the router levels.

M = 5,000 GB – maximum memory size per CO and per router.

Cache effectiveness function $H(m)$ was modeled in a such a way that its derivative, $H'(m)$, represents a continuous version of Zipf-Mandelbrot distribution

$$H'(m) = \frac{K}{(m+q)^{\alpha}},$$

truncated on interval [0, M_{ZM}], where 0< M_{ZM} $\leq \infty$. Here, Zipf-Mandelbrot power parameter $\alpha > 0$ characterizes cache effectiveness steepness at 0, shift parameter $q > 0$, and *K* is normalization coefficient:

$$\int_0^{M_{ZM}} H'(m)dm = P_{cache}$$

For the reference case, we chose $q = 1$, M_{ZM} = 1,000,000, $P_{cache} = 1$ and variated ZM power parameter α. Larger α correspond to more steep cache effectiveness curve and higher content reuse.

In the following, we consider two cases:

(a) Unconstrained case, when there is no limit on CO cache throughput and

(b) Constrained case, when CO cache throughput should not exceed some threshold.
First, we investigate a sensitivity of cache parameters, i.e. optimal CO cache sizes as well as optimal parameters (L, U) of incremental cache at the router, to a ZM power parameter α (Fig. 9). We can make the following observations:

1. CO cache size became small when power parameter α is very small or very large. Indeed, if power parameter α is very small, cache effectiveness function becomes very flat, amount of traffic offload becomes smaller for the same memory size, i.e. caching becomes less effective because it cannot justify the cost of the memory. If, on the other hand, power parameter α is very large, cache effectiveness function becomes very steep, amount of traffic offload becomes larger for the same memory size, i.e. caching becomes very effective and optimal point will be obtained at a smaller memory size.

2. For unconstrained case, the more traffic per CO the more optimal size of cache at CO. E.g., in our reference scenario, traffic $T_1 \le T_2 \le T_3 \le T_4$, and corresponding optimal cache sizes m_1^o at CO-1, m_2^o at CO-2, m_3^o at CO-3 and m_4^o at CO-4 follow the same pattern $(m_1^o \le m_2^o \le m_3^o \le m_4^o)$ for each power parameter α. Indeed, in unconstrained case, the optimal value of cache size is defined by stable points of (8) and (9), and the values of these stable points increase with increase of traffic. For constrained case, however, this rule does not hold

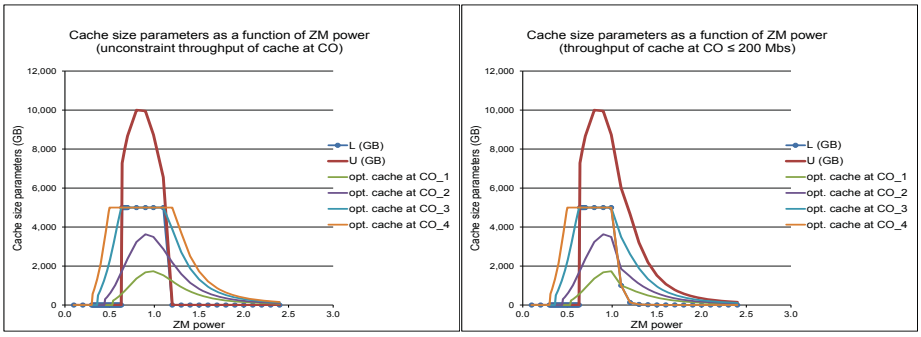

Figure 9: Sensitivity of cache parameters to ZM power.

any more as soon as traffic throughput became a limiting factor for a particular CO ($m_4^o < m_1^o$ when $\alpha > 1.1$).

3. There is a range of values of a power parameter α when cache at the router is utilized for the optimal solution; outside of this range we do not use cache at the router, i.e. $U = L = 0$. The size of this range depends on whether we have unconstrained or constrained case.

Next, we investigate a sensitivity of cache parameters to the cost of memory at CO (Fig. 10). In all following scenarios we assume that power parameter $\alpha = 0.9$. The following observation can be made:

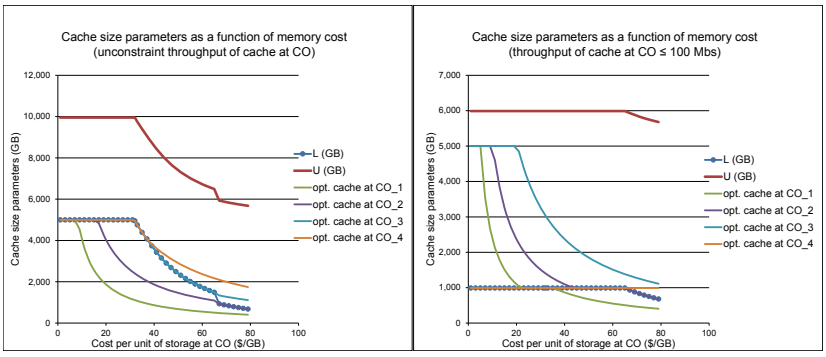

Figure 10: Sensitivity of cache parameters to memory cost at CO.

1. With increase of CO memory cost, optimal memory size at CO is constant or decreases.

2. For unconstrained case, the more traffic per CO the more optimal size of cache at CO, until cache size became equal to a maximum allowed memory amount per CO (5,000 GB in our case).

3. When CO memory cost is small (below some threshold), optimal CO memory size is equal to a maximum allowed memory amount per CO (5,000 GB in our case). With increase of CO memory cost, optimal memory per CO begin to decrease. The threshold depends on CO and increases with CO traffic.

4. Optimal parameters (L^o, U^o) of the cache at the router follow the observations made in Section 5:

4.1. L^o is equal to one of the optimal values for CO memories, $m_i^o, i = 1,2,3,4$; in this particular scenario, as cost of memory increases, L^o "jumps" from m_4^o to m_3^o to m_2^o .

4.2. When CO memory cost is below some threshold (in our case, below ~$32/GB), $U^o = U_{\max}$. When CO memory cost is above the threshold, U^o start decreasing, but a total memory at the router, $U^o - L^o$, continue to be equal to a maximum allowed memory at the router (5,000 GB).

5. For a constraint case, traffic throughput became a limiting factor for CO-4, that is why optimal cache size for CO-4, m_4^o , is fixed and for some values of CO memory cost is less than optimal cache size for CO-1, CO-2 and CO-3.

5.1. Parameter L^o of the cache at the router follows the observations made in Section 5: L^o is equal to one of the optimal values for CO memories $m_i^o, i = 1,2,3,4$; in this particular scenario, as cost of memory increases, L^o "jumps" from m_4^o to m_2^o .

5.2. A total memory at the router, $U^o - L^o$, is equal to a maximum allowed memory at the router (5,000 GB).

Next, we investigate a sensitivity of cache parameters to the cost of memory at the router (Fig. 11). The following observation can be made:

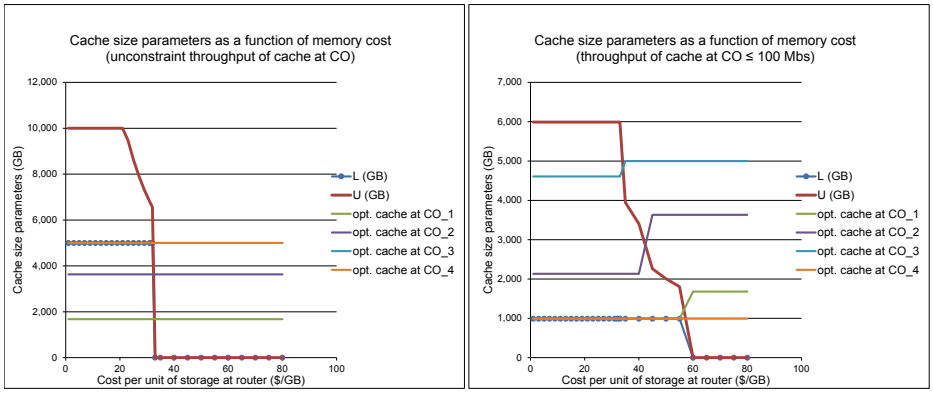

Figure 11: Sensitivity of cache parameters to memory cost at the router.

1. With increase of router memory cost, optimal memory size at the router, $U^o - L^o$, is constant or decreases.

2. For unconstrained case, cache at the router is utilized for optimal solution if memory cost at the router is below some threshold (~$32/GB). For this range of memory cost, parameter L^o of the cache at the router

follows the observations made in Section 5: L^o is equal to the optimal values of memory at CO-3 and CO-4: $L^o = m_3^o = m_4^o$

3. If memory cost at the router is below some threshold (~$23/GB), an optimal memory at the router, $U^o - L^o$, is equal to a maximum allowed memory (5,000 GB). With further increase of memory cost at the router, U^o (and size of the memory at the router) decreases to zero.

4. Optimal cache size per CO in this scenario does not depend on cost of memory at the router. Optimal size of cache at CO-3 and CO-4 are limited by maximum allowed cache size at CO: $m_3^o = m_4^o = 5{,}000\ \text{GB}$. As for optimal cache at CO-1 and CO-2, the rule about direct dependency of cache size and traffic holds: the more traffic per CO the more optimal size of cache at CO, i.e. $m_3^o = m_4^o > m_2^o > m_1^o$, where m_2^o and m_1^o corresponds to stable points (8) or (9).

5. For constrained case, traffic throughput became a limiting factor for CO-4, that is why optimal cache size for CO-4, m_4^o, is fixed and is less than optimal cache size for CO-1, CO-2 and (for some values of cost of memory at the router) for CO-3.

6. Parameter L^o of the cache at the router follows the observations made in Section 5: if memory cost at the router is below some threshold (~$60/GB), L^o is equal to the optimal values for CO-4 memory: $L^o = m_4^o$. With further increase of memory cost at the router, L^o decreases to zero.

7. If memory cost at the router is below some threshold (~$34/GB), an optimal memory at the router, $U^o - L^o$, is equal to a maximum allowed memory (5,000 GB). With further increase of memory cost at the router, U^o (and size of the memory at the router) decreases to zero.

8. As memory cost at the router increases, at some moment each of optimal cache at CO-1, CO-2 and CO-3 make a "jump" to the higher level, from the 1st stable point (8) to the 2nd stable point (9); this "jump" is correlated with sharp decrease of U^o. For example, optimal cache at CO-3, m_3^o, is equal to ~4,607 GB for the memory cost below ~$34/GB and m_3^o is equal to 5,000 GB for memory cost above $34/GB. At the same time, U^o decreases from ~5,991 GB (at $34/GB) to ~3,948 GB (at $35/GB).

8 CONCLUSION

In this paper we presented an analytical model for hierarchical cache optimization in CDN with two levels of hierarchy, COs on low level and routers on high level. We assume that all COs have different traffic demands (unbalanced traffic). The model uses several key parameters – number of COs, traffic per each CO, cache effectiveness function, unit memory and unit network cost – to calculate an optimal cache size in each CO and at the router. A concepts of cache effectiveness function and incremental (*L*, *U*)-cache at the router have been introduced. We presented analytical solution for optimal cache parameters and investigated relationship between these parameters. In particular, we demonstrated that for optimal network configuration of caches at CO and router levels:

1. Optimal size of cache at each CO may be calculated analytically for a given (*L*, *U*) pair.

2. A lower bound incremental cache characteristic *L* should be equal to one of the optimal cache sizes at CO.

3. There is a finite number of (*L*, *U*) pairs that are "candidates" for optimal solution of the problem. We can calculate optimal solution (as described in 1) for each (*L*, *U*) pair from candidate set and then select the best solution over all such pairs.

Sensitivity studies allowed estimating an impact of cache effectiveness function and unit cost of cache on optimal cache configuration and demonstrated that for many typical scenarios optimal cache configuration includes cache from one (CO) or both (CO and router) levels of network hierarchy.

ACKNOWLEDGMENTS

The author would like to thank Bill Krogfoss of Bell Labs for valuable technical discussions and useful suggestions.

REFERENCES

Balamash, A., and M. Krunz. 2004. "An overview of web caching replacement algorithms". In *Communications Surveys & Tutorials, IEEE* vol. 6, Issue: 2, pp. 44 – 56.

Chen, H., H. Jin, J. Sun, X. Liao, and D. Deng. 2003. "A new proxy caching scheme for parallel video servers". In *Proceedings of the International Conference on Computer Networks and Mobile Computing,* pp. 438 – 441.

Cobarzan, C., and L. Boszormenyi. 2007. "Further Developments of a Dynamic Distributed Video Proxy-Cache System". In *Proceedings of the 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based*, pp. 349 – 357.

Ghandeharizadeh, S., T. Helmi, T. Jung, S. Kapadia, and S. Shayandeh. 2006. "An Evaluation of Two Policies for Placement of Continuous Media in Multi-hop Wireless Networks". In *Proceedings of the Twelfth International Conference on Distributed Multimedia Systems*, pp. 1-13.

Ghandeharizadeh, S., and S. Shayandeh. 2007. "Greedy Cache Management Technique for mobile Devices". In *Proceedings of the IEEE 23rd International Conference on Data Engineering*, pp. 39 – 48.

Krogfoss, B., L. Sofman, and A. Agrawal. 2008. "Caching architecture and optimization strategies for IPTV networks". *Bell Lab Technical Journal* vol. 13, 3, pp. 13-28.

Krogfoss, B., L. Sofman, and A. Agrawal. 2009. "Hierarchical cache optimization in IPTV networks". In *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp.1-10.

Lee, J.P., and S.H. Park. 2005. "A cache management policy in proxy server for an efficient multimedia streaming service". In *Proceedings of the Ninth International Symposium on Consumer Electronics*, pp. 64 – 68.

Semi-continuity [Online]. Available: https://en.wikipedia.org/wiki/Semi-continuity.

Sofman, L., B. Krogfoss, and A. Agrawal. 2008. "Optimal Cache Partitioning in IPTV Network". In *Proceedings of the 11th Communications and Networking Simulation Symposium*, pp.79-84.

Sofman, L., and B. Krogfoss. 2009. "Analytical Model for Hierarchical Cache Optimization in IPTV Network". In *Proceedings of the IEEE Transactions on Broadcasting*, Vol. 55, 1, pp. 62-70.

Vleeschauwer, D., and D. Robinson. 2011. "Optimum Caching Strategies for a Telco CDN". *Bell Labs Technical Journal* vol. 16(2), pp. 115–132.

Zipf-Mandelbrot Law [Online]. Available: http://en.wikipedia.org/wiki/Zipf_Mandelbrot_law.

AUTHOR BIOGRAPHY

LEV B. SOFMAN is a Senior Network Architect in Frontier Communications. His areas of interest include network/traffic modeling and cost/performance analysis and optimization. Before joining Frontier Communications in October 2015, he worked with Alcetel-Lucent/Bell Labs from 2001 to 2015. Lev Sofman received his Ph.D. degrees in Mathematics from Moscow University, Russia, and from the Institute of Information Transmission Problems, Russian Academy of Sciences. His e-mail is lev.sofman@ftr.com.