Multi-Objective In-Network Caching Strategies

Liang Wang[†]
[†]Department of Computer Science, University of Helsinki, Finland

***Abstract*—The fast growth of user-generated content puts a significant burden on the current network infrastructure. By implementing caching functionality on network infrastructure, we can effectively eliminate redundant traffic, reduce delay and achieve other goals. In my thesis, we utilize optimization model and design of heuristic to study and develop novel caching strategies with multiple objectives for the networked caches. We evaluate their performance in the realistic settings.**

I. Introduction

Current Internet is used for disseminating all kinds of data, mainly multi-media content. Huge amount of user-generated content is produced and distributed via popular Internet services (e.g. Facebook and Youtube) every day. As a result, the caused traffic keeps increasing year by year. This fast growth in traffic puts a significant burden on the current infrastructure. Users suffer from severe network congestion and large delay.

However, [2] shows most of the content are consumed many times once they are generated, and their popularity follows zipf distribution. Further investigation also shows a large portion of traffic is redundant. These findings imply that using cache can significantly reduce the network traffic. The cache is usually installed at the network edge to serve users' requests. Another incentive of using cache is traffic cost grows much faster than the price of storage.

In Information-Centric Networks (ICN), content is accessed by name and routers are equipped with caches. These two features make it possible to share the content among different flows. Therefore ICN provides an ideal way to migrate caching functionality into network infrastructure. However, designing effective caching algorithm to manage networked caches is challenging. Modeling and evaluation are also difficult, and they are still the missing parts in the current research.

II. In-Network Caching Model

A caching strategy is a distributed algorithm running on each of the routers in a network. It manages a group of networked caches by letting each individual router make its own caching decision. A good caching strategy should maximize the utilization of network storage to meet various user-related and network-related objectives, e.g. eliminating redundant traffic, reducing delay, increasing availability and so on. A caching strategy consists of three polices as follows:

1) Admission policy: which content to cache?
2) Replacement policy: which content to evict?
3) Cooperation policy: where to cache the content in a network of collaborative caches?

In our model, we assume an ISP runs a network of M routers. Each router i is installed wtih a cache of size C_i. N files are from the item set $I = \{f_1, f_2, \cdots, f_N\}$, and the corresponding file size are $s_1, s_2, \cdots, s_N$. The popularity of file f_i is p_i and it follows zipf distribution. To share the content across the flows, the packet contains the content name. Clients keep requesting data from the servers. The servers may not be in the same ISP network. Distance (in number of hops) between cache i and the client is h_i. At time t, content distribution in the network is $Y_t = [Y_{i,j}]$ showing if file f_i is stored in cache j. Figure 1 summarizes the research model.

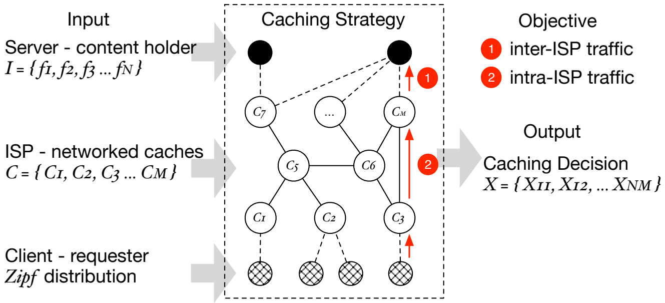


Fig. 1: In-Network Caching Model

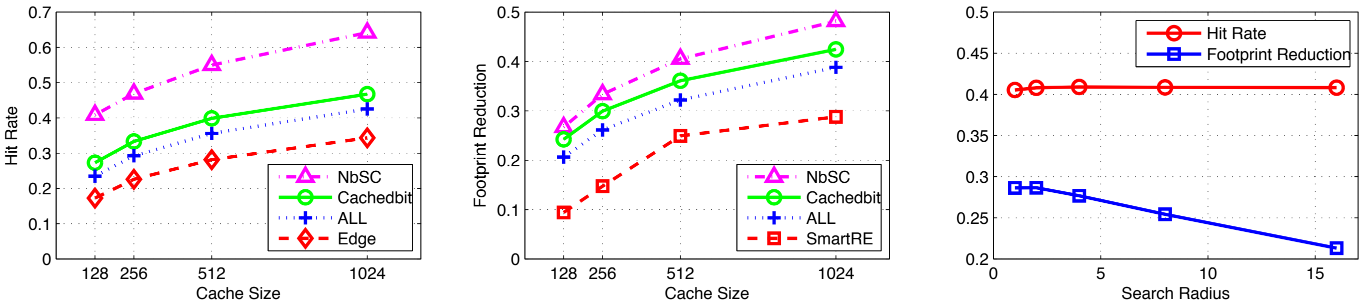
The ISP tries to reduce both outgoing traffic to other ISPs (inter-ISP traffic) and the traffic within its own network (intra-ISP traffic) by utilizing the network storage. The first objective can be achieved by maximizing byte hit rate of the network, and the second one can be achieved by minimizing the traffic footprint. The traffic footprint is defined as the product of traffic volume and the distance from the content source to the client. Further, footprint reduction is defined as the percent of reduction in footprint when caching is used. Higher reduction means in-network caching strategy can more effectively reduce the intra-ISP traffic. Let's assume that at time t the user requests file f_k, and f_k is stored at cache R_{hit}. Let S_C denote the set of routers that are located on the path from R_{hit} to the user. The optimal caching policy can be formulated as follows:

$$\max \sum_{i=1}^{N} s_i p_i \sum_{j=1}^{M} \frac{(M - h_j) X_{i,j}}{M} \tag{1}$$

$$\sum_{i=1}^{N} Y_{i,j} s_i X_{i,j} + s_k X_{k,j} (X_{k,j} - Y_{k,j}) \leq C_j \quad \forall C_j \in S_C \tag{2}$$

$$\sum_{j=1}^{M} X_{i,j} \leq K \quad \forall i \in \{1, 2, 3 ... N\} \tag{3}$$

where $X_{i,j} \in \{0, 1\}$ is our decision variable yielding value 1 if f_i will be cached in cache j, and 0 otherwise. The solution to



(a) Hit Rate

(b) Footprint Reduction

(c) NbSC with Different Search Radius

Fig. 2: Evaluation of Three Caching Strategies on Sprint Network

the above Linear Integer Programming Problem shows which item to be replaced at each cache in S_C for maximizing network's byte hit rate and footprint reduction simultaneously. Constraint 2 represents the cache capacity constraints while (3) represents our restriction that a file can only be stored at K caches ($K \leq M$). However, optimization model can only provide a reference upper bound of the system performance. In reality, the size of the problem makes it impractical to implement on the actual production network. We have to design distributed heuristic to adopt real-life complexity.

III. Evaluation

We already designed several different caching strategies and evaluated them on the realistic network topologies [3], [4]. We also compared them to other solutions such as edge-cache and SmartRE [1]. For the details of the algorithms and the experiment settings, please refer to [4].

In this paper, we only present the evaluation of three strategies (*ALL*, *Cachedbit* and *NbSC*) on Sprint network. *ALL* is the basic strategy, its admission policy simply caches everything and replacement policy uses LRU. *Cachedbit* extends *ALL* by probabilistically caching the packets passing by a router. *NbSC* extends *Cachedbit* by adding collaboration policy, it tries to search the missing content from the neighborhood.

Figure 2a shows hit rate as a function of cache size. As the cache size increases, the hit rates of all strategies increase. *Cachedbit* is better than *ALL*, and the improvement shows the importance of good admission policy. *NbSC* is the best of all. Since it uses the same admission policy as that of *Cachedbit*, the difference between both illustrates how collaboration policy boosts the hit rate. All the caching strategies, even the simplest one, are better than the edge-cache solution.

Figure 2b shows footprint reduction as a function of cache size. The reduction increases as the cache size becomes bigger. The ranking of the three strategies is the same as that in figure 2a. *NbSC* is the best and can significantly reduce the intra-ISP traffic. We compared our strategies to one of the prominent Redundancy-Elimination solution – SmartRE [1]. The results showed caching strategies outperform SmartRE significantly.

Searching from neighborhood accounts for the good performance of *NbSC*. Therefore, we also investigated how search radius impacts *NbSC*'s caching performance. Figure 2c shows hit rate and footprint reduction of *NbSC* as a function of different search radius. The results show as we increase the search radius, hit rate remains at the same level, but the footprint reduction deteriorates quickly. It implies small search radius is sufficient to save both inter- and intra-ISP traffic. Too big radius will hurt intra-ISP traffic because the requests may be redirected many times in the network.

IV. Future Direction

In current model, a file is considered as an indivisible unit and is either cached completely or not. However, this assumption does not hold for multi-media files. For example, users may only watch part of a video. Considering the size of videos, caching complete files apparently degrades utilization of the cache. However, bringing in partial caching will significantly increase the model complexity and difficulties in designing heuristic. As a future direction, we plan to first investigate the user behavior on viewing the parts of the video, then come up with a model for the popularity distribution of the parts within a video file. Based on this model, we will design low-complexity partial-caching strategy and compare it with the optimal solution of the modified problem in Section II.

Another direction is using graph-theoretical approaches to improve the performance of in-network caching. Because the network of routers is inherently a graph, many properties (e.g. degree, betweenness, centrality and etc.) can potentially influence the design of caching strategies, especially for cooperation policy. We plan to design more effective cooperative caching by exploiting graph structural properties, and study how these properties impact caching performance.

References

[1] A. Anand, V. Sekar, and A. Akella. SmartRE: an architecture for coordinated network-wide redundancy elimination. In *ACM SIGCOMM*, 2009.

[2] L. Breslau, P. Cue, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *IEEE INFOCOM*, 1999.

[3] W. Wong, M. Magalhães and J. Kangasharju. Content Routers: Fetching Data on Network Path. *IEEE ICC*, 2011.

[4] W. Wong, L. Wang, and J. Kangasharju. Neighborhood Search and Admission Control in Cooperative Caching Networks. *IEEE GLOBECOM*, 2012.