CoMon: A System Architecture for Improving Caching in CCN

Hani Salah, Benjamin Schiller, and Thorsten Strufe

TU Darmstadt, Germany
{hsalah, schiller, strufe}@cs.tu-darmstadt.de

***Abstract*—Content-Centric Networking (CCN) promises to yield large efficiency gains for Internet content distribution. Its autonomous cache management, however, raises doubts about achieving the intended goals optimally. A coordinated cache management, based on timely usage information, will help to fully leverage the cache efficiency. In this poster we introduce *CoMon*, a system architecture that implements Coordinated caching based on Monitoring of content usage and its stability. CoMon aims at improving CCN caching with *low monitoring and communication overheads*.**

I. Introduction

The use of the Internet today is dominated by content distribution and retrieval applications. These applications generate massive and ever-increasing volumes of traffic. This situation will soon prevent the current Internet infrastructure from satisfying user demands. To address this problem, Content-Centric Networking (CCN) [1] proposes a radical shift from the traditional host-centric (i.e. end-to-end) communication model to a content-centric model.

A key feature of CCN is its *on-path caching* strategy, i.e. contents are cached at intermediate routers on the path from the content provider to the consumer. This strategy has shown the ability to reduce redundant network traffic as well as content retrieval delays. In this strategy, however, the caching routers work *autonomously*, which results in unnecessary cache redundancy and cache-ignorant routing decisions.

We propose *CoMon*, a system architecture that utilizes timely monitoring of content usage and stability information to coordinate caching at the autonomous system (AS) level. CoMon aims at achieving high cache efficiency and simultaneously dropping the monitoring and communication overheads incurred by prior monitoring-based CCN caching solutions like [2]–[4].

In order to achieve the aforementioned goals, CoMon performs monitoring at only a small fraction of routers. It selects them at *strategic* locations. It also utilizes these routers only to *deflect* request packets off their original paths towards local AS caches, whenever a content match is found. CoMon employs a *coordinator device* for processing the monitored data, calculating cache assignments, and sharing valid cache settings with the monitoring routers. Our preliminary evaluation shows the feasibility of CoMon's design.

II. Overview of CoMon

CoMon's system architecture, shown in Fig. 1, includes three components, which we describe briefly here:

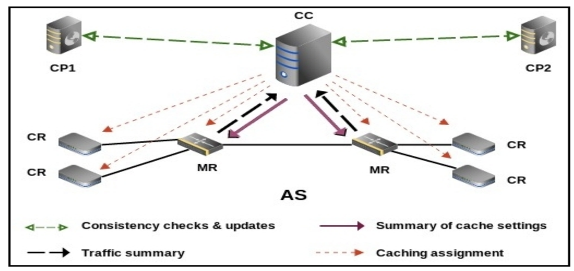

Fig. 1: CoMon architecture

1) *Content Coordinator (CC):* Each autonomous system (AS) hosts a device that processes monitored data, calculates cache assignments, checks the freshness of cached objects and updates staled objects. In actual implementation, the CC could be centralized or distributed.
2) *CCN Routers (CRs):* These are similar to regular CCN routers [1], but the caching approach is modified.
3) *Monitoring Routers (MRs):* In addition to the functions of CRs, the MRs monitor forwarded request packets at a *configurable* level of detail, upload summaries of their observations to the CC, and perform packet deflections. The CC can also query the MRs for finer-grained statistics, when needed.

In CoMon, all routers within the AS have routes to each others. CoMon ranks all AS routers according to a chosen graph centrality metric (e.g. betweenness centrality, degree centrality, routing centrality, group betweenness centrality), and selects the top-K central routers (or group) as MRs.

The aforementioned components work together as follows:

- The CC divides the name-space of named data objects into sections and maps each name section to one or multiple routers. The CC shares the *section-to-router mappings* and any updates on them with the MRs only. Consequently, cached objects are *addressed* by mapping their names to the corresponding routers.
- The CC employs a *cache consistency strategy* (e.g. cache invalidation or invalidation contracts) to check the freshness of cached objects and to update stale objects.

Based on the performed checks, the CC classifies objects belonging to different content providers (CPs) into distinct *content stability classes* (*CSC*s), e.g. very dynamic, dynamic, or static. The CC uses the CSCs to determine frequencies of future consistency checks.

- The CC calculates *cache assignments* by taking as inputs the popularity of objects and their CSCs. The output includes objects to be cached as well as their caching durations (i.e. ages). The CC performs these calculations either *periodically* or in response to events observed in the monitored data.
- The CC shares summaries of cache assignments only with the MRs. The MRs in turn update their routing tables accordingly. Routing information of cached objects include addresses of the caching routers (as CCN names), object names, and exit faces.
- Request packets are forwarded towards the corresponding CPs till they encounter the *first* MR in the path. If that MR has no routing information to a matching cached objects, the original path is preserved. Otherwise, that MR will write the address of the caching router to a newly introduced *location field*. It so *deflects* the packet off the original path towards the caching router. The next routers also forward the packet towards the identified location.

III. Preliminary Results

To evaluate the feasibility of CoMon, we performed a simulation study by implementing the basic CCN functionality in GTNA [5], a general routing analysis framework. We use real AS topologies from [6]. In the following, we use *Traffic Centrality (TC)* as a measure to rank the nodes (i.e. routers) by the number of request packets they capture.

We aim to answer the following questions: (*i*) What is the fraction of request packets that can be captured by top-K TC routers?, (*ii*) how stable are top-K TC routers for different simulation settings?, (*iii*) how accurate are the estimations of top-K TC routers for top-N popular objects?, and (*iv*) how much do graph-theoretic centrality metrics correlate to TC?

We applied the following settings in our simulations: total number of objects is 100 times the network size; the objects are distributed randomly in the network; each node issues 50 requests in round-robin order; the popularity of objects follows a Zipf distribution ($\alpha \in \{0.7, 0.85, 1.2, 1.8\}$); cache sizes $\in \{0, 1000, 10000, 100000\}$; cache replacement with LRU and LFU.

We summarize the results as follows:

- To answer question (*i*), we ranked the routers by their TC values and calculated the total unique request packets captured by top-K TC routers. The top-10% TC routers, for simulations applying caches of sizes 0 and 100000, were enough to capture more than 90% and 80% of overall packets, respectively.
- As for question (*ii*), we performed six simulations applying very different settings. In particular, we used different combinations of caches sizes and α values. We then calculated the correlations among their top-K TC routers. The results we achieved show a low correlation for the top-10% TC routers. However, the correlation for the top-25% TC routers was about 97% for simulations with no caches and 80% for simulations with caches of size 10000.

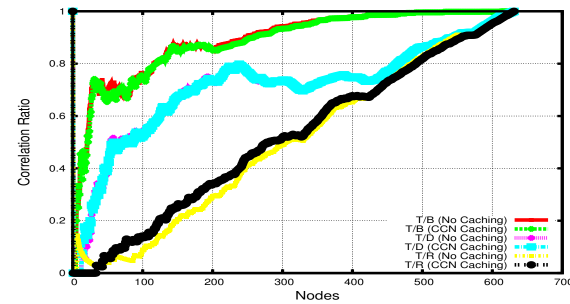

Fig. 2: TC correlations to BC, DC, and random placements

- Exemplary results for question (*iii*) are the estimations of the top-5% and the top-25% TC routers for top-N popular objects (for N $\in$ {5, 10, 20, 50}). The accuracy of all these estimations was above 80%, and reached about 99% with the top-25% TC routers with large cache sizes.
- Fig. 2 shows the correlations of top-K TC routers to random placement (T/R), degree centrality (T/D), and betweenness centrality (T/B). The best correlation in all the simulations was to BC. The cache size only had a slight impact on the correlations.

To conclude, the results above suggest that the proposed solution is feasible: There is always a small fraction of routers whose union of observations (in terms of the number of captured packets) are very close to the overall observations. Moreover, those routers are able to estimate most popular objects with high accuracy.

IV. Ongoing and Future Work

We are currently working on reducing duplicate monitoring data, designing more accurate monitor placement strategies, and implementing CoMon and evaluating it with real traces.

We next plan to extend CoMon and employ it in the detection of request flooding attacks. We also plan to consider a multi-AS scenario where ASes maintain content-level peering agreements to leverage each others' cache contents.

References

[1] V. Jacobson *et al.*, "Networking named content," in *Proceedings of ACM CoNEXT*, 2009.

[2] D. Saucez *et al.*, "Minimizing bandwidth on peering links with deflection in named data networking," 2012.

[3] C. Bernardini *et al.*, "Towards popularity-based caching in content centric networks," in *RESCOM 2012*, 2012.

[4] X. N. Nguyen *et al.*, "Efficient caching in content-centric networks using openflow," in *INFOCOM 2013 Student Workshop*, 2013.

[5] B. Schiller and T. Strufe, "Gtna 2.0 - a framework for rapid prototyping and evaluation of routing algorithms," in *Proceedings of SummerSim*, 2013.

[6] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *Proceedings of ACM SIGCOMM*, 2002.