Cache Management Strategy for CCN based on Content Popularity

César Bernardini, Thomas Silverston, Olivier Festor

▶ **To cite this version:**

César Bernardini, Thomas Silverston, Olivier Festor. Cache Management Strategy for CCN based on Content Popularity. 7th International Conference on Autonomous Infrastructure (AIMS), IFIP, Jun 2013, Barcelonne, Spain. pp.92-95, 10.1007/978-3-642-38998-6_12 . hal-00929736

HAL Id: hal-00929736

https://hal.inria.fr/hal-00929736

Submitted on 13 Jan 2014

Cache Management Strategy for CCN based on Content Popularity

César Bernardini[*,†], Thomas Silverston[*,†] and Olivier Festor[†]

[*]Université de Lorraine, LORIA, UMR 7503, France
[†]Inria Grand Est - Nancy, France
Email: {cesar.bernardini, thomas.silverston, olivier.festor}@inria.fr

Abstract. Content Centric Networking is a promising architecture for the Future Internet to deliver content at large-scale. It relies on named data and caching features which consists of storing content across the delivery path to serve forthcoming requests. As some content is more likely to be requested than other, caching only popular content may help to manage the cache of CCN nodes. In this paper, we present our new caching strategy adapted to CCN and based on the popularity of content. We show through simulation experiments that our strategy is able to cache less content while it still achieves a higher Cache Hit and outperforms existing default caching strategy in CCN.

1 Introduction

The Internet is currently mostly used for accessing content. Indeed in the 2000s, P2P traffic for file-sharing counted for about 80% of the overall Internet traffic. Nowadays, video streaming services such as Youtube represent the most important part of the Internet traffic. It is expected that the sum of all forms of video (TV, VoD and P2P) will be approximately 86% of global consumer traffic by 2016 [1].

While the Internet was designed for -and still focuses on- host-to-host communication (IP), users are only interested in actual content rather than source location. Hence, new Information-Centric Networking architectures (ICN) such as CCN [2] have been proposed. ICN architectures give high priority to efficient content distribution at large scale and have attracted considerable attention from the research community [3].

Content Centric Networking (CCN) is a network architecture based on named data where a packet address named content, not location. The notion of host as defined into IP does not exist anymore. In CCN, the content is not retrieved from a dedicated server, as it is the case for the current Internet. The premise is that content delivery can be enhanced by including per-node-caching as content traverses the network. Content is therefore replicated and located at different points of the network, increasing availability for incoming requests. An important feature for CCN is to manage the cache of nodes with caching strategies and replacement policies, which decide whether to cache and in case the cache is full, the element to be replaced respectively. It is therefore essential to design efficient cache management strategies adapted to CCN networks. Some content can be largely popular and requested by a large number of users while some other may have a more limited audience and stay confidential. Thus, we believe that the popularity of content should be take into account while designing efficient cache management strategy for CCN.

In this work, we present our new cache management strategy, Most Popular Content (MPC), designed for CCN. Instead of storing all the content at every node on the path, MPC caches only popular content. MPC caches less than CCN default strategy but still improves in-network caching performance while -at the same time- decreases resource consumption. In the paper, we first provide an overview of CCN, then we introduce MPC and present our simulation environment and performance evaluation of MPC.

2 Most Popular Caching Strategy

2.1 CCN Overview

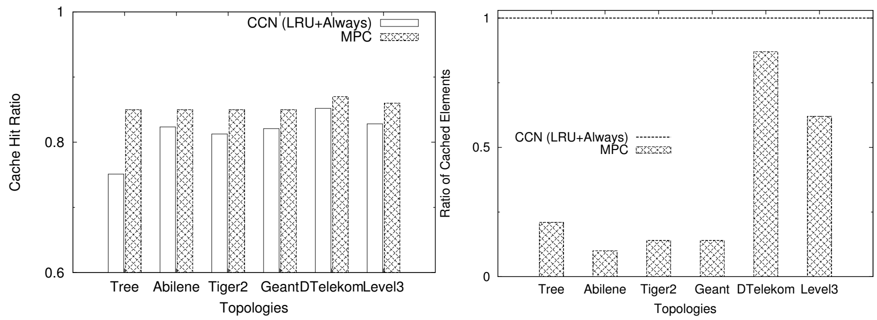
CCN architecture is mostly based on two primitives: *Interest* and *Data*. A consumer requests content by sending an *Interest* message in the network; any node hearing the request and having the data can issue a response with a *Data* message. The content is therefore transmitted to the consumer and every node on the delivery path can cache the data. With no clearly defined cache management strategy, the CCN default strategy always stores content at all nodes on the delivery path (*Always* strategy).

As this strategy could lead to replace unpopular by popular content, we argue that caching only popular content will allow to achieve high performances and it will at the same time save resources. Therefore, we design MPC -*Most Popular Content*- a new cache management strategy for CCN, where nodes cache only popular content.

2.2 Most Popular Caching Strategy

In MPC, every node counts locally the number of requests for each content name, and stores the pair *(Content Name; Popularity Count)* into a *Popularity Table*. Once a content name reaches locally a *Popularity Threshold*, the content name is tagged as popular and if the node holds the content it suggests its neighbor nodes to cache it through a new *Suggestion* primitive. These suggestion messages may be accepted or not, according to local policies such as resource availability. As the popularity of a content can decrease with time after the suggestion process, the *Popularity Count* is reinitialized according to a *Reset Value* in order to prevent flooding the same content to neighbors. MPC Strategy influences directly in CCN node requirements.

In addition to CCN cache space required, MPC needs an extra space to store the *Popularity Table*. For instance, to keep one million table entries means $1GB$ of RAM memory using $1023B$ per content name and $1B$ for the Popularity Count (we used fixed length for the name to simplify calculation).



(a) Cache Hit Ratio (b) Ratio of Cached Elements

Fig. 1: MPC vs. CCN (LRU/Always) over different topologies

3 Simulation Environment

In order to evaluate our new strategy MPC, we use ccnSim [4], a chunk-level CCN simulator, developed in C++ over the Omnet++ framework.

For our experiments, we use a large-scale Youtube-like catalog containing 10^8 video of $10MB$ each: approximately a catalog of $1PB$. The cache size is set to 10 GB. The popularity of files is modeled following a MZipf distribution function available in ccnSim [4]. Then, for each experiment, we randomly set one catalog and 8 requester nodes on the topologies. In the experiments, we ramdomly place one catalog and 8 requester nodes into the ccnSim-included topologies. We then perform 10 runs of the same simulation and provide the average value.

4 Preliminary Results

The performance of MPC is measured according to (i) the probability to obtain a *Cache Hit* all along the path; (ii) the number of caching operations over the number of issued messages (*Ratio of Cached Elements*); the proportion of traveled hops across the network with respects to the original path from content source to its destination. (*Stretch*); (iv) the ratio of different chunks stored in the caches (*Diversity*).

In the Figure 1a, our strategy *Cache Hit* is greater than 85% and at the same time higher than CCN. Even when CCN reaches its highest results with Level3 or DTelecom topologies, MPC still outperforms CCN. The *Ratio of Cached Elements* is presented in the Figure 1b. CCN's default strategy generates one operation per every issued message which means CCN Ratio of Cached Elements is always 1. Fig. 1b shows that our MPC strategy caches up to 5 times less content than CCN for the Tree, Abilene, Geant and Tiger topologies (approximately 20%), performing less caching operations and saving memory. With DTelecom and Level3, our strategy caches more content than other topologies but still outperform CCN (80% and 60% respectively). For this particular topology, this is due to the high connection degree of nodes.

Due to space constraints, we do not present the Figure for the Stretch metric or Diversity of content in cache. For the Stretch metric, MPC and CCN exhibit similar results: about 10% for all the topologies and both strategies are able to cache content close to requesters.

The CCN Diversity ranges from 28% to 35% for all the topologies and the MPC Diversity is much lower from 3% to 18%. Regarding Diversity, it was expected that MPC is less efficient than CCN since MPC has been designed in order to cache only popular content, limiting the diversity of the chunks in the cache of nodes.

5 Conclusion

In this work, we present MPC, our new cache management strategy for CCN networks. MPC strategy caches only popular content and reduces the cache load at each node.

Our simulation experiments show that MPC outperforms the CCN default caching strategy. MPC achieves a higher Cache Hit Ratio and still reduces drastically the number of replicas. By caching less data and improving the Cache Hit ratio, MPC improves network resources consumption.

6 Future Work

As future work, we expect to investigate the popularity concept in social environments. Nowadays, the Internet has become a social-oriented network, where users organize themselves into communities and share content among them. A group of popular users receive most of the attention and tend to act as opinion leaders. Thus based on this fact, we expect to privilege popular users by proactively spreading their content into caches. We advocate popular users generate content more likely to be requested by other users.

References

1. Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2011-2016. Technical report, Cisco, feb 2012.
2. V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM.
3. http://www.ccnx.org
4. D. Rossi and G. Rossini. Caching performance of content centric networksunder multi-path routing (and more). Technical report, Telecom ParisTech, 2011.