Could In-Network Caching Benefit Information-Centric Networking?

Sen Wang, Jun Bi, Jianping Wu, Zhaogeng Li, Wei Zhang, Xu Yang

Department of Computer Science, Tsinghua University, Beijing, China

Network Research Center, Tsinghua University, Beijing, China

Tsinghua National Laboratory for Information Science and Technology (TNList)

{wangsen, lizhaogeng, zw, yangxu}@netarchlab.tsinghua.edu.cn; {junbi, jianping}@cernet.edu.cn

ABSTRACT

Information-Centric Networking is (ICN) [1] gaining increasingly concerns, as an important direction of the Future Internet Architecture research. Although In-network caching is considered as one of the most significant properties of ICN, the cache policy for ICN is still little explored. In this paper, we formulate the in-network caching problem of ICN into Mixed-Integer Linear Programming problem. We also propose a novel cache policy named LB (Least Benefit) with taking into account the distance factor and a new forwarding scheme with shallow flooding (FSF for short) to improve the performance further,. Our simulation results show that with in-networking caching, the average hops of the ICN network can be reduced significantly by nearly 50% with simple cache policy like LFU and with some simple improvement such as LB and FSF the average hop can be reduced further.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: *Distributed networks*

General Terms

Design, experimentation, performance.

Keywords

Information-Centric Networking; Caching; Future Internet;

1. INTRODUCTION

As a Future Internet Architecture proposal, ICN intends to motivate the architectural transition from today's host-centric Internet architecture to information-centric. In this research area, many approaches have been proposed such as PSIRP, NetInf, PURSUIT, CCN, DONA and NDN [1]. Although In-network caching is considered as one of the most significant properties of ICN, the cache policy for ICN is still explored little so far. In [2], preliminary evaluation on network performance improvement by random autonomous caching is exhibited with very simple topology and scenario.

With the aim of being independent of any specific approaches aforementioned, in this paper we extract the ICN structuring architectural properties and assumptions as the following. We refer to the content identifier as CID (content ID) which names a content object uniquely. The content transmission mechanism adopts the Pub/Sub communication paradigm which is embedded in most of the approaches aforementioned. Within the Pub/Sub paradigm, a request is sent to get a specific content object with the CID specified in its packet. The request is routed to the original content object without awareness of any cached copy in routers by some routing mechanism (e.g. OSPF). We make this assumption because of considering that knowing current caching state of the whole network would imposes the routing system significant burden in terms of maintaining extra states. Along the path of a request, any router caching a copy of the desired content object could respond to this request with its copy. We assume the content will go back to the requester along the same path which means an accurate symmetric routing. While forwarding a content object, an intermediate router can decide freely whether to cache the content object according to its own cache policy.

In this paper, we consider the benefits of deployment of ICN and study the impact of cache policies on the overall network performance of ICN. In this paper, the optimal cache for ICN is explored. We formulate the in-network caching problem of ICN into Linear Programming problem (Integer Programming, more accurately). A novel cache policy referred as LB (least benefit) is proposed which takes into count the benefit of a cache hit instead of simply counting the hit number as LFU does. Several series of simulations are conducted over simple topology and practical ISP topology to evaluate the proposed cache policy and forwarding strategy. We use synthetic traffic, which is generated from content-based traffic model deriving from traffic study of Web caching.

The rest of the paper is organized as follows. Section 2 presents the Linear Programming formulation of the in-network caching problem of ICN. In Section 3, a novel

AINTEC '11, November 09 - 11 2011, Bangkok, Thailand

cache policy named LB is proposed and In Section 4 a novel forwarding scheme with shallow flooding is presented. Section 5 gives the simulation results about various cache policies and novel forwarding scheme. We conclude in Section 7.

2. CACHING IN ICN

For the performance of ICN, the main concern in our opinion is how to cache content generated from its customers or imported from exterior network optimally so that the network resource consumed by the transfer of these content or/and the transmission latency was minimized. Considering the static case, the in-network caching problem can be described as the following. Given an ICN network, the request rate from each router to each content object, the storage capacity of each router, a set of initial content objects, an assignment of resident routers of these content objects and a routing path from each router to each content object, the goal is to find a feasible assignment of caching copies of each content object to routers in order to minimize the overall resource consumption of the network. We formulate this problem into a MILP problem as follows.

Graph construction: The ICN network is represented as an undirected graph $G = (V, E)$; V is the set of nodes in the network; E is the set of edges in the network; $b: V \rightarrow R^{+}$ is a function. $b(v)$ denotes the storage capacity of the node v; C is the set of initial content objects. A content object is denoted by a tuple (l, m), where l is the resident node of the content and m is the size. For any content object $c \in C$, function $l(c)$ and $m(c)$ return the resident node and the size respectively.

Traffic demand: $q_{v,c}$ is the ratio between the number of requests initiated from node v for content object c and the number of all the requests. Obviously, we have $\sum_{c \in C} \sum_{v \in V} q_{v,c} = 1$.

Routing path: We assume that there is only one routing path between any pair of nodes. $p_{u,v}$ denotes the single path between node u and node v and the function $u(p_{u,v})$ gives the length of this path. The k-th node in the path from node u is denoted by the return value of the function $g(p_{u,v}, k)$ for any $0 \leq k \leq u(p_{u,v})$.

An assignment of in-network caching: We use a series of binary variables $\delta_{c,v} \in \{0, 1\}$ to describe the caching state. If node v has cached the content object c in its storage, the $\delta_{c,v}$ has the value 1, otherwise 0.

For a request, the network resource consumed is estimated as the product of the number of the hops traversed to get the first copy of the desired content object and the size of corresponding content object. Then the overall objective function is as follows.

$$\min \sum_{v \in V} \sum_{c \in C} q_{v,c} * m(c) * \min_{0 \leq i < u(p_{v,l(c)}),\ \delta_{c,g(p_{v,l(c)},i)} != 0} \left(\delta_{c,g(p_{v,l(c)},i)} * i,\ u(p_{v,l(c)})\right) \quad (1)$$

$\min_{0 \leq i < u(p_{u,v}),\ \delta_{c,g(p_{v,l(c)},i)} != 0} \left(\delta_{c,g(p_{v,l(c)},i)} * i, u(p_{v,l(c)})\right)$ is the minimal number of hops from v to any copy of content object c along the routing path $P_{v,l(c)}$, including the original content object resided in node $l(c)$. The objective function (1) is subject to the storage capacity constraint of any node v:

$$\sum_{c \in C} \delta_{c,v} * m(c) \leq b(v), \forall v \in V \quad (2)$$

Note that the *min* function is not a linear function, so we involve a series of additional continuous variables $\mu_{v,c}$ and binary variables $\sigma_{v,c,i}$ for every node v and every content object c to linearize the objective function. The binary variables $\sigma_{v,c,i} \in \{0,1\}$ indicate whether the $\delta_{c,g(p_{v,l(c)},i)} * i$ is the minimal hop number for any i subject to $0 \leq i < u(p_{v,l(c)})$. The $\sigma_{v,c,u(p_{v,l(c)})}$ represents the original content object. These variables are subject to the following constraint:

$$\sum_{i=0}^{i \leq u(p_{v,l(c)})} \sigma_{v,c,i} = 1, \forall v \in V, \forall c \in C \quad (3)$$

Besides, we have the following constraints for the variables $\mu_{v,c}$:

$$\delta_{c,g(p_{v,l(c)},i)} \geq \sigma_{v,c,i}, \forall v \in V, \forall c \in C, 0 \leq i < u(p_{v,l(c)}) \quad (4)$$

$$\mu_{v,c} \geq \delta_{c,g(p_{v,l(c)},i)} * i - (\max_{v \in V, c \in C} u(p_{v,l(c)})) * (1 - \sigma_{v,c,i}), \forall v \in V, \forall c \in C, 0 \leq i < u(p_{v,l(c)}) \quad (5)$$

$$\mu_{v,c} \geq u(p_{v,l(c)}) - (\max_{v \in V, c \in C} u(p_{v,l(c)})) * \left(1 - \sigma_{v,c,u(p_{v,l(c)})}\right), \forall v \in V, \forall c \in C \quad (6)$$

Inequality (4) indicates that when $\sigma_{v,c,i}$ has the value 1, the $\delta_{c,g(p_{v,l(c)},i)}$ can not have the value 0. Inequality (5) associated with (6) assures that $\mu_{v,c}$ is at least larger than one of $\delta_{c,g(p_{v,l(c)},i)} * i$ for any i subject to $0 \leq i < u(p_{v,l(c)})$ or the path length $u(p_{v,l(c)})$. Finally, we have the ultimate objective function (7) subject to (2)(3)(4)(5)(6).

$$\min \sum_{v \in V} \sum_{c \in C} q_{v,c} * m(c) * \mu_{v,c} \quad (7)$$

3. CACHING POLICY

We expect to improve the LFU by taking into account the distance from the hit cache to the resident node of the original content object. Intuitively, we give each hit a weigh which is equal to the hop gains by caching the content object locally. We refer to the new cache policy as LB (Least Benefit).

To implement LB, each node needs to know the hop gain from caching some content object locally, for which we introduce an additional field to the ICN protocol, namely:

The *HopCount* is in the content packet and specifies the current hop count the packet has traversed from the resident node of this copy of the content object.

With this additional field, every intermediate node can know the hop reduction for any content object it has encountered if caching it locally. When a hit happens, the current overall benefit of the hit content object would be increased by the current hop reduction. When a replacement is needed for an incoming new content object, the content object with least benefit will be evicted out.

4. FORWARDING WITH FLOODING

In the section, we discuss the impact of forwarding on the network performance. The study above is based on the traditional forwarding mechanism as the internet does, namely, single path. We expect that with the ability of caching more intelligent forwarding should be endowed in ICN.

We found that when a request is forwarding to the original content object, the "search area" is bound to the single path from the requesting node to the residential node of desired content object. We expect to improve the cache hit rate by enlarging the "search area". Here we introduce a bit more intelligent forwarding scheme inspired by P2P forwarding, which is named Forwarding with Shallow Flooding (FSF for short). The FSF scheme is simple as Figure 2 shows. When a request is received by a node, the request will be flood to all its other interfaces with a specific flooding depth, while forwarding the request to the corresponding original content object according to its forwarding table. When multiple content objects are found, only the first coming content object is forwarding to its requester.

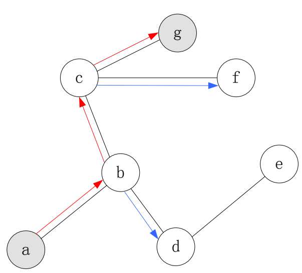


Figure 1. Forwarding with One-step Flooding

5. EVALUATION

In this section, we present our simulation results of various caching policies applied in ICN. The main evaluating metric is the average hops of all the requests to get corresponding content objects. We implement a content-based transport protocol based on Pub/Sub communication model of ICN and various caching policies on open source network simulation software NS3.

5.1 Traffic Model for Evaluation

We argue that designed around content, the ICN features should be studied around contend-based request model. We expect the ICN traffic model has the same features as those of web traffic model which prior studies have exhibited. [3] shows that the request rate of HTTP object follows the Zipf-like distribution, which means the request probability of the ith most popular object is proportional to $\frac{1}{i^{\alpha}}$ and independent reference model is sufficient for qualitative analysis [3]. In this paper, the Zipf–Mandelbrot (Also known as the Pareto-Zipf) distribution is used as the traffic model for simulation. The request rate distribution is represented as follows.

$$\mathbf{P(i)} = \frac{\Omega}{(i+q)^{\alpha}} \quad \mathbf{(8)}$$

The two parameters α and q in the equation above are the shape parameter and shift parameter respectively and Ω is the normalizing constant.

5.2 Evaluation without FSF

For simplicity, we fix the size of all the content objects to 1 KB in all our simulations. We distribute 10,000 content objects randomly to all nodes. For every node, the request distribution follows the popularity distribution of (8). Each node has a cache with the size of 500 content objects. The parameters of traffic pattern, namely α and q, are fixed to 0.7 and 0.7 which are reasonable for real traffic of web request. The arrival of request follows the Poisson Process, which means that the time interval of two adjacent requests is exponentially distributed. Unless some change of these parameters is specifically stated, all the following experiments are conducted under the parameter values above.

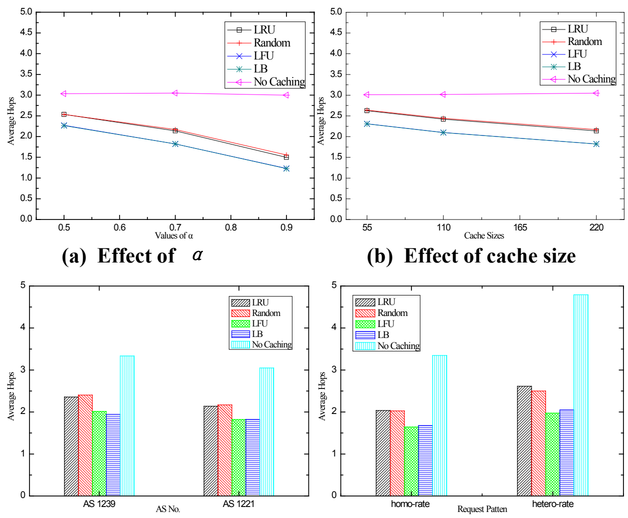
A series of simulations using practical ISP topology is conducted to evaluate cache policies. First we use the PoP topology of ISP with AS No. 1221 provided by [4] for studying the effect of parameter α and cache size on *average hops*. The simulation results are shown in Figure 2 (a) and Figure 2 (b).

Figure 2 (a) show that the parameter α does have significant impact on *average hops*. While α increase from 0.5 to 0.9, the *average hops* drops from 2.26 to 1.23 and the hop reduction rate rises from 25.4% to 59.2% for LB. Figure 2 (b) shows the effect of cache size. The aggregate cache size of all the nodes varies from 55% of the total size of all the content objects to 220%. The average hops descents when the cache size ascents, but the slope descents accordingly, which means simply increasing cache size cannot decrease the two metrics efficiently.

With α and cache size to be fixed to 0.7 and 110% respectively, the LB gains 30.5% reduction in *average hops* compared to that in the case of no cache and has almost no obvious difference with LFU. It seems that LB does not perform obviously better than LFU and we cannot improve the performance by simply making use of the distance factor.

We explored the effect of different ISP topologies on the performance of various cache policies by simulating under the PoP topology of another ISP with AS No. 1239, which has 78 nodes and 84 edges, larger that AS 1221 with 44 nodes and 44 edges. The Figure 2 (c) shows that the average hops of the five cache policies for the *Pop topologies* of the two ISPs. In this figure, it can be observed that different topologies result in almost no difference in average hop reductions, which are 40.3% and 41.7% for AS 1221 and AS 1239 respectively with the cache policy of LB.

We also studied the effect of heterogeneous request rate among nodes. In all the former simulation, each node request content objects with the same mean value of request intervals, namely one second. In contrast, the request rates of nodes range from 10 per second to 1 in this simulation. The simulation is conducted under the PoP topology of AS 1239. Figure 2 (d) shows the corresponding results, which suggests that in the setting of heterogeneous request rates, more average hops can be achieved, arising from 49.9% to 57.3%.



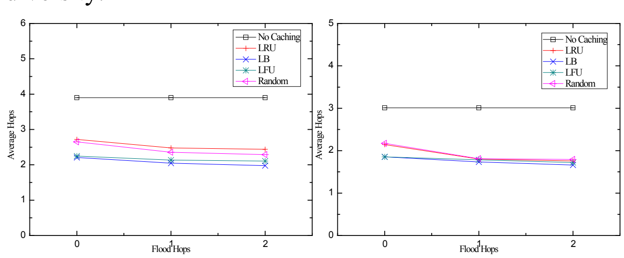
(a) Effect of α **(b) Effect of cache size**

(c) Effect of topology (d) Effect of request pattern

Figure 2. Simulation results without FSF

5.3 Evaluation with FSF

In this subsection we evaluate the FSF with various cache policies. Two series of simulations were conducted under a 6×6 mesh topology and the PoP topology of AS 1221 respectively. The results are shown in Figure 3. The FSF can further decrease the average hops by 6.3% with 2 hops flooding for the LB and LB is better than LFU obviously while the flooding hops increasing. We speculate that it is due to the fact that LB takes into account the distance factor which gives the objects cached in adjacent nodes diversity.



(a) On Mesh Topology (b) On ISP Topology

Figure 3. Simulation results with FSF

6. RELATED WORK

While ICN is gaining increasingly concerns, some research effort is made to address its specific mechanism including caching. [2] focuses on the packet-level caching in ICN and tests the caching policy of random autonomous caching with flow-based testing traffic model under simple network topology. Considering the Information-centric nature of ICN, this paper studies the issue of in-network caching of ICN from the perspective of content object.

In the research area of CDN, the problem of how to replicate object optimally in terms of reducing the latency of the request and bandwidth consumption is studied extensively, including both mathematic and practical efforts. These researches are most relevant to our work in terms of LP formulation. [5] studies the object replication problem with respect to minimize the average hop count to get the object as this paper does. It assumes some mechanism to get the object from the nearest AS (corresponding to the node in this paper). In contrast, the caching issue of ICN discussed in this paper does not assume any path given by routing system to the nearest copy, considering the massive scale of content objects and the network itself which disables the assumption of some kind of global information about current caching state.

7. CONCLUSION

We formulate the in-network caching problem of ICN into Mixed-Integer Linear Programming problem. The proposed cache policy LB (Least Benefit) performs better than LFU when the proposed forwarding scheme FSF is involved too and reduces the average hops further by 6.3%. Our simulation results show that with in-networking caching, the average hops of the ICN network can be reduced significantly by nearly 50% and with some simple improvement such as LB and FSF the average hop can be reduced further.

8. REFERENCES

[1] Bengt Ahlgren, Christian Dannewitz etc. A Survey of Information-Centric Networking (Draft), http://drops.dagstuhl.de/opus/volltexte/2011/2941/pdf/10492.KutscherDirk.Paper.2941.pdf.

[2] S. Arianfar, P. Nikander, and J. Ott, “On content-centric router design and implications,” ACM ReArch 2010, November 30, 2010, Philadelphia, USA.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and Zipf-like distributions: Evidence and implications,” IEEE INFOCOM, 1999.

[4] Rocketfuel, "An ISP Topology Mapping Engine," http://www.cs.washington.edu/research/networking/rocketfuel/

[5] J. Kangasharju, J. Roberts, and K.W. Ross, “Object replication strategies in content distribution networks,” Computer Communications, vol. 25, no. 4, 2002, pp. 376-383.