EMC: The Effective Multi-path Caching Scheme for Named Data Networking

Hao Wu, Jun Li, Yi Wang, Bin Liu
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, China
wu-h11@mails.tsinghua.edu.cn, lijunreed@mail.tsinghua.edu.cn, wy@ieee.org, liub@tsinghua.edu.cn

***Abstract*—The Named Data Networking (NDN) is proposed recently as a promising paradigm for the future Internet due to its built-in caching and name-based routing for efficient content distribution. For the time being, the research on NDN caching is still a preliminary topic, especially for the scenario of an ISP with multiple gateways. For more in-depth excavation, we have studied the effective intra-ISP caching under multiple gateways and multi-path routing in this paper. With the primary objective of reducing the inter-ISP traffic, we develop a popularity-based coordinated caching strategy named the Effective Multi-path Caching scheme (EMC), which substantially saves more than 50% inter-ISP traffic and more than 30% content access latency. Through evaluation, we observe that EMC significantly outperforms the widely used Leaving Copies Everywhere (LCE) scheme and Leaving Copies with Probability (LCProb) scheme in terms of reducing both the inter-ISP traffic as well as the content access latency. Extensive simulation results demonstrate that our proposed caching scheme is effective, scalable and light-weight.**

I. Introduction

With the popularization of the content-oriented applications like Facebook, LinkedIn and Youtube etc., Internet is getting much more concerned for its current inefficient operation mode, i.e., gives its priority to identify the end host devices and then accesses content, resulting in the emerging of the Content-centric Networking (CCN). Named Data Networking (NDN) [1], as an instance of the materialization, is recently proposed as a promising architecture for the future Internet, featured by its built-in caching capability and name-based routing [2] [3].

In this paper, we pay more attention to the systematic in-network caching. In NDN, each router is equipped with a certain amount of memory to cache contents [2] [3]. This is quite different from the IP network where caches are only located in dedicated severs at the edges. Basically, NDN runs requester-driven communication model, i.e., a client will first send out an *interest packet* for the desired content, and then a server returns the content within a *data packet*. By caching the popular contents in routers, the corresponding requests will need no longer to traverse the whole Internet to the end host server(s), but are served by closer NDN routers on the en-route path(s). Therefore a considerable amount of redundant traffic load will be potentially saved in the midway and thus the network transmission efficiency is correspondingly improved.

This work is supported by 863 project (2013AA013502), NSFC (61073171), Tsinghua University Initiative Scientific Research Program (20121080068), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20100002110051), NSFC (71071145).

Given the multi-dimensional constraints, such as the card space, cost and the system performance in a router, the practically equipped memory storage is relatively limited, comparing with the exponentially increasing contents over the Internet. So how to achieve maximal benefits by caching the content replicas effectively under a limited memory becomes a challenging research issue. Although the caching schemes in IP-based networks have already been studied for years, a number of exclusive mechanisms and architectures make the caching design in NDN a brandnew research topic, while the existing caching mechanisms (like the web caching) cannot be simply applied in NDN. We focus on the study of the intra-ISP caching in NDN and take the reduction of inter-ISP traffic as our primary objective, for the reason that the bandwidth of internal links in ISPs are usually surplus, while the inter-ISP links are bearing heavy traffic load and always regarded as the bottleneck of the Internet. Furthermore, taking the user experience into account, we regard the reduction of content access latency as an additional important metric.

Since most of the real-life ISPs have more than one gateway to interconnect with peer-ISPs and there may be multiple content sources of a certain file, we develop our caching model within ISPs under multiple gateways and multi-path routing in this work. We present a popularity-based coordinated caching strategy named as the Effective Multi-path Caching scheme (EMC), which helps routers make online caching decisions. Taking advantage of the global popularity information, EMC then adjusts the cache placement among local routers in order to serve as many requests as possible. In general network settings, more than 50% inter-ISP traffic and 30% content access latency can be saved. We estimate the performance of EMC under real networks while a variety of impact factors, including cache capacity, request pattern, content population and network topology, are compositively considered. Also, the preliminary cost is evaluated. Extensive simulation results show that EMC can markedly improve the caching performance on reducing the inter-ISP traffic and the content access latency with acceptable overheads.

The rest of the paper is organized as follows: Section II introduces our caching model. Section III presents the popularity-based coordinated caching scheme–EMC. The evaluation platform, impact factors and simulation results are discussed in Section IV. The related works are summarized in Section V. Finally, Section VI concludes the paper.

978-1-4673-5775-3/13/$31.00 ©2013 IEEE

II. System Model and Problem Statement

Since NDN is a newly proposed future network, we first sketch out the NDN architecture under multi-path routing in this section. Then, combined with our optimized objective, we discuss the caching model within a single ISP with multiple gateways.

A. An overview of NDN architecture

As a content-oriented network, NDN is characterized as the in-network caching and the name-based routing. Specifically, in order to facilitate content transmission, a particular object is supposed to be divided into several fixed chunks, which are regarded as the basic units transported in NDN. And differing from the dedicated caching severs deployed in the IP-based networks, each router in NDN is equipped with a certain amount of memory to cache the passing contents, serving the subsequent requests without the need of forwarding them to the source servers.

To be content-wise, NDN abandons the traditional IP address which infers nothing with the essence of the content but assigns each kind of contents with a globally unique *content identifier* (e.g. the value of a hash function with respect to its URL), through which the dispersed chunks of a specified kind of contents can be recognized and retrieved in a multi-path fashion.

B. The caching model within an ISP with multiple gateways and multi-path routing

In this subsection, combining with the content requesting and fetching processes, we first introduce the en-route caching pattern in the context of NDN. Then, we illustrate our intra-ISP caching model through a typical case study, while the multi-path routing mechanism is highlighted.

When requesting a particular content object, the host sends an *interest packet* attached with the *content identifier* to its connected router. Then the router checks its local cache. If hit, the router should return the corresponding replica to the host and no longer forward the *interest packet* anymore. Otherwise, the router forwards the *interest packet* to the next router. In the same manner, the *interest packet* is forwarded along the routing path towards the content source. It should be emphasized that the routing paths of the requests/chunks are determined by parsing the components of the *content identifier* and may be different even between a same departure/destination pair of routers in NDN. As soon as the request is served by either a cache or the source, the replica will travel downwards to the requester and each en-route router can make decision whether to cache it, under the local caching scheme.

Let's consider the model illustrated in Figure 1(a). From the practical perspective, multiple gateways are taken into account. ISP_A accommodates 10 content routers. Therein, R_1 and R_2 are gateways to interconnect with other ISPs, either provider-ISPs or peer-ISPs. R_3 and R_4 are core routers and the others are access routers which are responsible for receiving the content requests from their end users or customer networks.

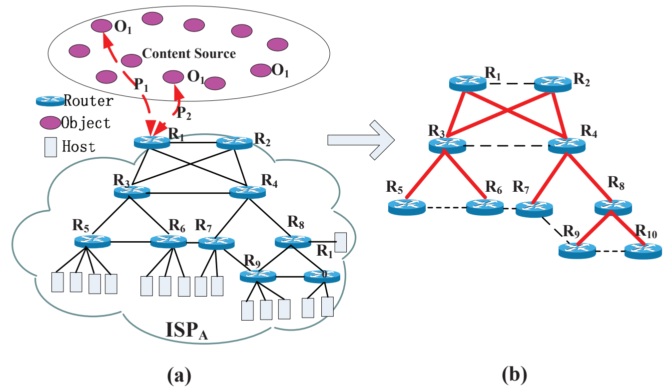


Fig. 1. An ISP with multiple gateways and multi-path routing

When an access router receives an *interest packet* for a content object (say O_1), the router forwards it towards the content source guided by the *content identifier*. As there are multiple locations that hold O_1, there are multiple routing paths from the requester to the content originators. In Figure 1(a), P_1 and P_2 are both the routing paths from gateway R_1 towards the content source of O_1. Actually, for all the content objects outside the ISP*, the routing paths from the requester to the content originators within the ISP range are summarized to the paths from the requester to the ISP gateways. Clearly, all the content replicas retrieved from sources outside will pass through the ISP gateways and can be cached at some inner routers on the paths to the requesters, with duplicated content objects being removed.

We extract those paths involved in every available content's name-based routing and then obtain the final caching infrastructure. Without loss of generality, we assume that all the kinds of contents are requested at every access router in ISP_A. Then, we extract a multi-tree-like caching topology as depicted in Figure 1(b). Actually, our caching scheme can be deployed in any caching topology.

III. EMC: The Effective Multi-Path Caching Scheme

With the objective of reducing the inter-ISP traffic, we present a popularity-based intra-ISP caching strategy, named the Effective Multi-path Caching scheme (EMC) under the environment of multi-gateway and multi-path routing. Since the coming request statistics are not priori-known and the fact that the request rates are generally stable during a fine time size, our caching scheme is triggered at every preset interval to determine the replica placement based on the most recent history of request statistics.

EMC consists of three major stages as illustrated in Algorithm 1, including the initialization, the request spread & aggregation and the cache distribution processes. Table I summarizes the notations used in EMC.

A. The initialization of EMC

Following the discussion in Section II-(B), we first extract the intra-ISP caching topology *M* from the input router-level

*Since our objective is to reducing the inter-ISP traffic, we focus on the content objects outside the ISP.

TABLE I
THE NOTATIONS OF EMC

Notations	Descriptions
M	the intra-ISP caching topology extracted from the input router-level ISP network
$Gateway$	the set of gateway routers of the ISP
U	the set of access routers of the ISP
R.$TotalSpace$	the storage capability of router R's subnet
R.RV	stands for *Request Vector*, a local vector recording each object's request rate at R
R.$LOWTHRESHOLD$	the minimum request rate of currently cacheable objects at R
R.$MAXSEND$	the size of the request profile spread by R to be aggregated by the upriver router
O.LOW	a boolean value: TRUE indicates content O is locally unpopular at current router
R.$CachedTable$	a local table recording the already cached objects of router R
R.$CacheMark$	a local table recording the objects to be cached when passing router R

Algorithm 1 EMC

1. **Initialization**:
   i. Extract the caching topology M through aggregating all the paths involved in the name-based routings of the available content items
   ii. Calculate the value of *TotalSpace* at each router in M
2. Each access router collects the request profile in *RV* during the most recent interval
3. REQUEST_SPREAD_&_AGGREGATION(M,$Gateway, U$)
4. CACHE_DISTRIBUTION(M,$Gateway$)

ISP network by aggregating all the paths involved in the name-based routings of the available content items.

For an access router, it keeps the multiple routing paths for different contents towards the gateways, while each intermediate router contained in these paths keeps the associated paths for the content request aggregation and the cache distribution. Meanwhile, each intermediate router calculates the value of its local *TotalSpace* by summating the total cache capacity of the routers that regard it as their subsequent en-route router towards the gateway. Thus, for a router R_i, if we delineate a subnet S_i covering all the subordinate routers of R_i in *M*, we can conclude that R_i is the gateway of subnet S_i and R_i's *TotalSpace* indicates the storage capability of S_i.

We have sharply cut down the content profile by screening out the unpopular objects during request collection in order to dramatically reduce the communication and storage overheads. Specifically, we involve a *LOWTHRESHOLD* value at every router, which is the minimum value in term of the request rate of the currently cacheable objects.

B. The request spread & aggregation

All the access routers monitor the content requests periodically and record every object's request rate in a local vector called *RV*. However, due to the Pareto principle (which is also known as 80/20 principle), there is no need to convey the whole *RV* from an access router to its upriver router. Dealing with the sporadic requests will never get a better cache layout but incur more overheads. Instead, a router generates a request profile containing an appropriate amount ($MAXSEND$ in Algorithm 2) of most popular objects and sends it out. This amount can be estimated by the number of currently cached items in the network.

When EMC is triggered, as shown in Algorithm 2, all the access routers first spread their request profiles collected in the latest interval towards the gateways. As mentioned, in a given topology, the individual *Forwarding Path* for each content object should be extracted by parsing its *content identifier*. For an object O_k, an access router gets the *Forwarding Path* of O_k towards every gateway, and spreads O_k's request rate along its *Forwarding Path*s.

As for request aggregation, the intermediate routers on the *Forwarding Path*s (including gateways) aggregate all the request rates belonging to the same item sent from their subordinate nodes. Then, the en-route routers will check the request rate of each item. If a certain object's request rate is less than the value of local *LOWTHRESHOLD*, a *LOW* flag will be set to the object item indicating it is locally unpopular, which deals with the skewed content access at different access routers to make the most of the restricted cache room.

Finally, we come to a request-aligning process. That is, each en-route router spreads the *content identifer*s of the objects with *LOW* flag among the *Forwarding Path*s to the gateways. As a result, the aggregated item of O_k in the upriver routers is to be set the *LOW* flag as long as anyone of its components is attached with a *LOW* flag. So the *LOW* flag propagates in an infectious process.

Algorithm 2 REQUEST_SPREAD_&_AGGREGATION (M,$Gateway, U$)

/ scan each access router */*
for $i \leftarrow 1$ to LENGTH(U) **do**
  / scan each gateway which current access router is towards */*
  for $j \leftarrow 1$ to LENGTH($Gateway$) **do**
    / scan each content object in current access router's Request Vector */*
    for $k \leftarrow$1 to $MAXSEND$ **do**
      $OID \leftarrow U[i].RV[k].Oid$
      $ORate \leftarrow U[i].RV[k].Rate$
      Get the $ForwardingPath$ from $U[i]$ to $Gateway[j]$ through parsing $U[i].RV[k].ContentIdentifier$
      Add the corresponding request rate of O_{OID} in RV of every router among current $ForwardingPath$ by $ORate$
    end for
  end for
end for
Request-Aligning process:
i. Each router in M checks local RV and sets LOW flag to the items whose request rate is less than local $LOWTHRESHOLD$
ii. The router records the *content identifier*s of items with LOW in a vector and spreads it among the $ForwardingPath$s
iii. All the routers receiving the vector set LOW flag to those local involved items

C. The cache distribution

Illustrated in Algorithm 3, during the stage of cache distribution, each router makes caching decision from gateways downwards to access routers. At the very beginning, the router sorts the aggregated *RV* in descending order in term of the number of request rates.

Algorithm 3 CACHE_DISTRIBUTION(M,$Gateway$)

```
/* scan each gateway router */
for i ← 1 to LENGTH(Gateway) do
    Put all Gateway[i]'s sub-routers into ChildSet
    /* scan each sub-router of Gateway[i] */
    for s ← 1 to LENGTH(ChildSet) do
        CurrentRouter ← ChildSet[s]
        Sort CurrentRouter.RV in descending order in term of the
        number of request rates
        /* scan each content object in current access router's Request
        Vector */
        for k ←1 to LENGTH(CurrentRouter.RV) do
            OID ← CurrentRouter.RV[k].Oid
            ORate ← CurrentRouter.RV[k].Rate
            if CurrentRouter.RV[k].LOW = TRUE then
                CurrentRouter.RV[k].LOW ←FALSE
                Delta ← CurrentRouter.TotalSpace
                move CurrentRouter.RV[k] to
              CurrentRouter.RV[k + Delta]
                continue
            else if O_OID exists in CurrentRouter.CachedTable then
                Update the corresponding item of O_OID in
              CurrentRouter.CachedTable with current time
            else if There is enough room in CurrentRouter's cache to
            accommodate object O_OID then
                Create an item of O_OID in CurrentRouter.CacheMark
              and cache it as soon as the replica arrives
            else
                continue
            end if
            if CurrentRouter is a gateway of M then
                Delete O_OID among other gateway routers in M
            end if
            Delete O_OID among all the subordinate nodes of
            CurrentRouter
        end for
    end for
    /* execute the distribution process among the ChildSet of
    Gateway[i] recursively */
    CACHE_DISTRIBUTION(M,ChildSet)
end for
```

Then, it fetches the records from the top of its ordered *RV* and in turn, till the local cache capacity is filled to full. Special emphasis should be laid on the objects with *LOW* flags: those items should be moved *TotalSpace* positions backwards in the ordered *RV* in order to further reduce the communication overhead, for they are off-balance requested among the access routers.

If a record of a certain object has existed in the router's local *CachedTable* which records the already cached objects, then the corresponding item is updated with the current time to make it fresher. Otherwise, if the available cache space is large enough to accommodate the object, EMC creates an new corresponding item in the local *CacheMark* Table, which means the replica will be cached as soon as it passes the router.

In both cases above, the corresponding items of that object are removed from the *RV*s and caches of all the subordinate nodes of current router, in order to thoroughly eliminate caching redundancy. Besides, if the current router is a gateway of M, the items are also removed from other gateway nodes in M, because every gateway router is reachable for all the intra-ISP routers so that there is no need to cache a same object more than once among gateways.

D. A case study: The cache placement under EMC

We explain how EMC places the replicas through a case study. We consider the intra-ISP caching model introduced in Figure 1 as an example.

Given a set of caches and request rates, the problem is to decide which objects should be stored in the caching network and where to cache them, to satisfy our main objective of reducing the inter-ISP traffic. The objective can be interpreted as enhancing the intra-ISP cache hit rate by the in-network caching, with the capacity constraint of each cache. The data of request rates at each access router are given in Table II, where the three columns (a)∼(c) correspond to different sets of access statistics respectively. In each column, *O*=$\{o_1, o_2 ... o_N\}$ denotes a set of cacheable content objects and *#R* stands for the number of each access router. Provided that each content router in the topology could cache **one** content object, the replica placement under EMC for each access pattern is illustrated in Figure 2, respectively.

The EMC's outcome shows that generally the object with a higher aggregated request rate is cached near to the gateway routers. Besides, we observe that a low request rate of a globally popular object at an access router tends to draw the caching location closer to the network edge, once the request rate is lower than the value of local *LOWTHRESHOLD*. For instance, dataset (b) in Table II is with biased access rate of O_1 at the access routers, compared with dataset (a). If we reduce the access rate at $R_7 \sim R_{10}$ down to 12 or less (11 in dataset (b)), which is lower than the access rate of the object that drops within the cacheable region, the cache layout is then changed as illustrated in Figure 2(a) and (b). In the same manner, replica placement with the input of dataset (c) is consistent to the cache layout in Figure 2(c). Though the most popular objects $O_1 \sim O_3$ are biasedly requested among access routers, their request rate at each node is larger than the request threshold (10 in the case) and therefore their replica placement can still cover all the requests. O_3 is drawn downwards just because its aggregated request rate at R_4 is less than that of O_5.

IV. Performance Evaluation

In this section, we evaluate the performance of EMC in the real-life networks, while a variety of impact factors are considered. To show the practical feasibility of EMC, preliminary cost estimation is given as well.

A. Evaluation setup

To fully understand the performance of the proposed scheme, we build our exclusive evaluation platform. We briefly introduce the settings here.

1) Input data: The arrival pattern of content requests at each access router follows Poisson arrival $P_n(t) = (\lambda t)^n e^{-\lambda t}/n!$. Considering each router may cover different amounts of user populations, the average request rate λ at an access router follows the uniform distribution *U*(3,000, 7,000). As suggested in [4], the popularity of objects is governed by Zipf distribution, *f(i)* = $i^{-\alpha}/\sum_1^N k^{-\alpha}(0.5 \leqslant \alpha \leqslant 1.0)$, where

TABLE II
THE REQUEST RATE AT EACH ACCESS ROUTER

(a)

#R	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
5	40	25	20	16	13	10	8	6
6	40	25	20	16	13	10	8	6
7	40	25	20	16	13	10	8	6
8	40	25	20	16	13	10	8	6
9	40	25	20	16	13	10	8	6
10	40	25	20	16	13	10	8	6

(b)

#R	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
5	98	25	20	16	13	10	8	6
6	98	25	20	16	13	10	8	6
7	11	25	20	16	13	10	8	6
8	11	25	20	16	13	10	8	6
9	11	25	20	16	13	10	8	6
10	11	25	20	16	13	10	8	6

(c)

#R	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
5	100	75	40	30	13	10	8	6
6	100	75	40	30	13	10	8	6
7	10	10	10	9	13	10	8	6
8	10	10	10	9	13	10	8	6
9	10	10	10	9	13	10	8	6
10	10	10	10	9	13	10	8	6

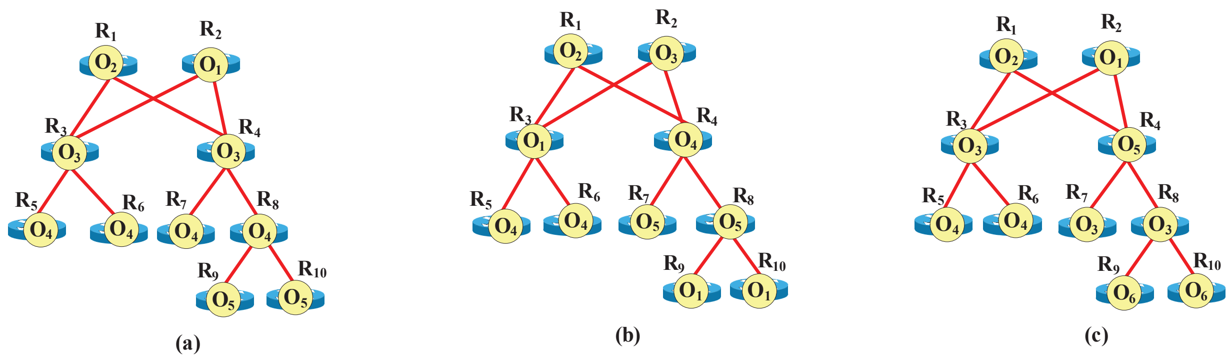


Fig. 2. The cache placement under EMC

N is the total number of requested content types and α is the skewness factor.

2) Network topology: We involve three real-life network topologies in the evaluation, as summarized in Table III. Comparatively, AS701 is a tier-1 ISP, while AS59 and AS224 are stub ISPs.

TABLE III
THE NETWORK TOPOLOGY

Network	AS number	Gateways	Nodes
Wisconsin University	AS59	2	41
UUNet Alternet	AS701	3	75
UNINETT	AS224	2	208

3) Metrics and compared schemes: Since NDN is a newly proposed infrastructure of future network and none of the same works aiming at caching in ISPs with multiple gateways and multi-path routing has been done yet, we take Leaving Copies Everywhere (LCE) [5] which is applied in the most influential article for NDN [2] and Leaving Copies with Probability (LCProb) as the baseline schemes for comparison. Both the schemes are widely used in most hierarchical caches due to their high performance and ease of implementation. In LCE scheme, once the requested object is sent back along the delivery path, each en-route router is supposed to replicate the copy of the object. And LCProb is quite similar to LCE except that the retrieved replicas are selectively cached by probability.

Based on the main optimized objective in our work, it is quite reasonable to regard the *Ratio of Reduced Traffic* (*RRT*) as the primary metric to quantificationally estimate the performance. Besides, taking the user experience into account, we use the *Ratio of Reduced Access Latency* (*RRAL*) as another important metric in our evaluation as well.

4) The default setting: We have conducted numerous simulations under all the available combinations of parameters and selected the most representative configuration as the default setting. We focus on the routing topology of AS701 in most of our simulations. The skewness parameter α of Zipf distribution followed by the object popularity is set to 0.8. We set the default population of contents to $N = 2,000$, and each object's size is assumed to 1, for the ease of illustration. The cache capacity at a single router is described as the *relative cache capacity* which is the proportion of a router's absolute capacity in the total size of all the content objects. The default value is 1% (That is, the absolute cache capacity at a router is $N \times 1\% = 20$.).

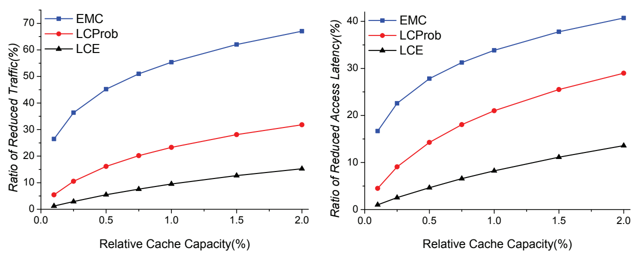
B. Evaluation results

We show the evaluation results with the impacts of *relative cache capacity*, request pattern, content population and network topology respectively.

1) Relative cache capacity: We first discuss the impact of the *relative cache capacity* in the range from 0.1% to 2%.

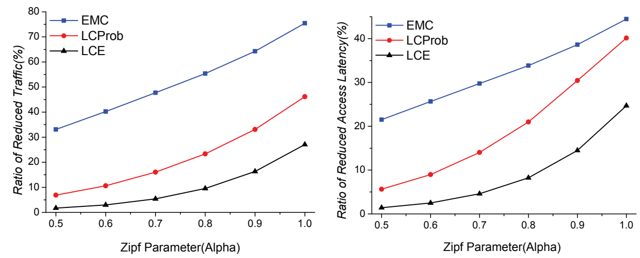
As shown in Figure 3, the performances of the three schemes have improved monotonically with the increasing of *relative cache capacity*, for the more room in routers, the more contents can be cached. Obviously, the proposed EMC significantly outperforms the baseline schemes in both *RRT* and *RRAL*.

2) Request pattern: The request pattern is governed by the Zipf skewness parameter α which indicates the concentration degree. When α approaches to 1, a few objects may cover the majority of requests; while when the value of α is near to 0, it indicates the object popularity is almost homogeneous.



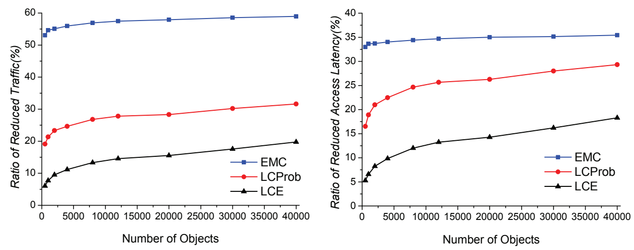
(a) *RRT* (b) *RRAL*

Fig. 3. Caching performance vs. Relative cache capacity



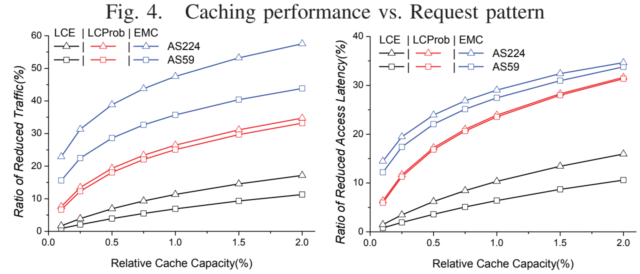
(a) *RRT* (b) *RRAL*

Fig. 4. Caching performance vs. Request pattern



(a) *RRT* (b) *RRAL*

Fig. 5. Caching performance vs. Content population



(a) *RRT* (b) *RRAL*

Fig. 6. Caching performance vs. Network topology

In Figure 4, we investigate the impact of α across the range from 0.5 to 1.0 which implies the request pattern is getting concentrated. It is not surprising that *RRT* and *RRAL* are steadily increasing as α increases, because the cached objects account for a relatively higher proportion of the requests under a larger α. In comparison with LCE and LCProb, EMC consistently provides the preferable performance over all the range of the Zipf parameter α.

3) Content population: We then conduct experiments to examine the impact of the increase of content population on the effectiveness and scalability of EMC.

As is shown in Figure 5, when the *relative cache capacity* is fixed to 1%, we observe that EMC still achieves a much better performance than the two widely-used baseline schemes when dealing with the increasing content population. Besides, EMC shows great effectiveness and scalability since both the metrics tend to converge as the number of objects increases. As the order of contents population in the Internet is up to 10^8, a stable performance gain of EMC can be expected if deployed in large-scale networks.

4) Network topology: We have also exploited the other two router-level topologies with different scales (AS224 and AS59 shown in Table III) respectively. Because AS224 has much more nodes than other networks, we enlarge the number of content objects to 4,000 in the simulation. Figure 6 illustrates that all the involved schemes are insensitive to the topology change, which ensures the scalability and the ease of deployment of EMC.

C. Measurement of overheads

To prove EMC's practical feasibility, we have not only deduced the expressions of overheads of EMC, but also conducted preliminary experiments to measure the cost in terms of the storage use, the communication overhead and the execution time. Since LCE and LCProb do not have the routers cooperate with each other in making caching decisions, there are no storage and communication cost.

1) Storage: We roughly deduce the expression of storage cost of EMC as $StoCost = 5.467 + 4.125N'$ bytes, where *N'* is the number of the objects involved in the object access profile. The storage cost of each router in AS701 is 1490 bytes.

2) Communication: Since the network AS701 can hold about 360 distinct object items (2,000 objects considered) under Zipf access pattern, each access router sends the statistics profile including 400 objects (the value of *MAXSEND*) for the request aggregation process in EMC. As such, the average communication consumption for each router is around 16KB during a triggered interval.

3) Execution time: Table IV shows the actual execution time of EMC and LCProb measured from all the three network topologies, while LCE is even more time-consuming than LCProb because the replicas are cached in all the passing routers. EMC executes periodically according to the acquired object access profile, while LCProb has to dispose every individual object timely upon the object's arrival. So it is reasonable that EMC can achieve a preferable execution time.

TABLE IV
THE EXECUTION TIME

Network	Nodes	EMC(ms)	LCProb(ms)	Objects
Wisconsin University	41	83	187.4	2000
UUNet Alternet	75	387.9	551.3	2000
UNINETT	208	813.55	1622.8	2000

V. Related Work

As a promising future network architecture, NDN has become a quite hot research topic nowadays. More and more researchers pay their attention to the related areas and some of their works have led to great success, such as [6] [7] in NDN name lookup and the thorough study on the PIT structure in

NDN conducted in [8]. Comparatively, the caching design is a more difficult task due to the special caching mechanisms and architectures in NDN. To address the challenge incurred by the explosive growth of contents, the in-network caching tends to be an inherent essence for the future networks (e.g. [1] [2] [3] [9]).

Though the caching schemes in traditional networks have been extensively studied (e.g. [10] [11] [12]), caching in content-oriented networks is still a new research area where rare works have been done yet. In recent studies on content-centric caching, [13] converted the collaborative in-network caching to linear programming problems with the guide of traffic engineering. In [14], Dai et al. decoupled the caching design into three tiers and solved them respectively, on a real-world IPTV system. And D.Rossi et al. [15] proposed a novel simulation study of the caching performance in CCN. All of them assume the named content in the content-oriented networks can in principle take any routing path in the network. That is, it is possible to fetch content replicas from any node of the hierarchical network topology. However, as suggested in NDN [1], the replicas should be cached in routers on the en-route path to further reduce the redundant traffic. According to this mechanism, only the routers along the routing path can cache formerly requested objects for future reuse. The similar idea has been studied in [16].

The algorithms proposed in the most recent works [17] [18] can achieve a much closer overall performance to the results of theoretically optimal solutions. But they just focused on the ISPs with single gateway and single-path routing. Due to the suggestion promoted by NDN [1] that the multi-path routing should be available for a particular named content to enhance performance and the fact that most of the real-life ISPs have several gateways to interconnect with peer-ISPs and provider-ISPs, we establish and study the intra-ISP caching model with multi-gateway and multi-path routing mechanisms.

In this work, we develop a popularity-based coordinated caching scheme which is supposed to make caching decisions on the fly. To our best knowledge, none of the same works has been done yet.

VI. Conclusion

In this paper, we have modeled the effective NDN caching system and developed a popularity-based coordinated caching scheme named EMC for ISPs with multiple getaways and multi-path routing, aiming at reducing the inter-ISP traffic and the average content access latency. We thoroughly evaluate the performance of EMC while a variety of impact factors and the overhead is considered in real networks. Extensive experimental results demonstrate that EMC significantly improves the caching performance on reducing the inter-ISP traffic and the content access latency with acceptable overhead. In addition, our proposed scheme retains distinct stability and scalability under a wide range of configurations.

References

[1] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named data networking (ndn) project," 2010.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2063176.2063204

[3] C. Dannewitz, "Netinf: An information-centric design for the future internet," in *Proceedings 3rd GI/ITG KuVS Workshop on The Future Internet, 2009*, may 2009, pp. 1 –3.

[4] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 6, pp. 1447 –1460, dec. 2008.

[5] N. Laoutaris, S. Syntila, and I. Stavrakakis, "Meta algorithms for hierarchical web caches," in *Performance, Computing, and Communications, 2004 IEEE International Conference on*, 2004, pp. 445 – 452.

[6] Y. Wang, K. He, H. Dai, W. Meng, J. Jiang, B. Liu, and Y. Chen, "Scalable name lookup in ndn using effective name component encoding," in *Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems*, ser. ICDCS '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 688–697. [Online]. Available: http://dx.doi.org/10.1109/ICDCS.2012.35

[7] Y. Wang, H. Dai, J. Jiang, K. He, W. Meng, and B. Liu, "Parallel name lookup for named data networking," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, dec. 2011, pp. 1 –5.

[8] H. Dai, B. Liu, Y. Chen, and Y. Wang, "On pending interest table in named data networking," in *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, ser. ANCS '12. New York, NY, USA: ACM, 2012, pp. 211–222. [Online]. Available: http://doi.acm.org/10.1145/2396556.2396600

[9] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '07. New York, NY, USA: ACM, 2007, pp. 181–192. [Online]. Available: http://doi.acm.org/10.1145/1282380.1282402

[10] L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary cache: a scalable wide-area web cache sharing protocol," *Networking, IEEE/ACM Transactions on*, vol. 8, no. 3, pp. 281 –293, jun 2000.

[11] O. Saleh and M. Hefeeda, "Modeling and caching of peer-to-peer traffic," in *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, nov. 2006, pp. 249 –258.

[12] M. Abrams, C. R. Standridge, G. Abdulla, E. A. Fox, and S. Williams, "Removal policies in network caches for world-wide web documents," in *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '96. New York, NY, USA: ACM, 1996, pp. 293–305. [Online]. Available: http://doi.acm.org/10.1145/248156.248182

[13] H. Xie, G. Shi, and P. Wang, "Tecc: Towards collaborative in-network caching guided by traffic engineering," in *INFOCOM, 2012 Proceedings IEEE*, march 2012, pp. 2546 –2550.

[14] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, "Collaborative hierarchical caching with dynamic request routing for massive content distribution," in *INFOCOM, 2012 Proceedings IEEE*, march 2012, pp. 2444 –2452.

[15] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing," in *Telecom ParisTech, Technical Report*, 2011.

[16] S. Bhattacharjee, K. Calvert, and E. Zegura, "Self-organizing wide-area network caches," in *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, mar-2 apr 1998, pp. 600 –608 vol.2.

[17] J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, X. Wang, Y. Zhang, and L. Dong, "Popularity-driven coordinated caching in named data networking," in *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, ser. ANCS '12. New York, NY, USA: ACM, 2012, pp. 15–26. [Online]. Available: http://doi.acm.org/10.1145/2396556.2396561

[18] J. Li, H. Wu, B. Liu, and J. Lu, "Effective caching schemes for minimizing inter-isp traffic in named data networking," in *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, dec. 2012, pp. 580 –587.