PhD Forum: Not So Cooperative Caching

Xiaoyan Hu*, Jian Gong*
*School of Computer Science and Engineering, Southeast University, Nanjing, P. R. China
Email: {xyhu, jgong}@njnet.edu.cn

***Abstract*—This work proposes a scheme to promote autonomous and selfish NDN (Named Data Networking) peering domains to cooperate in caching, here dubbed *Not So Cooperative Caching* (NSCC). We consider a network comprised of selfish nodes; each is with a caching capability and an objective of reducing its *own* access cost by fetching data from local cache or from neighboring caches. The challenge is to determine what objects to cache at each node so as to induce low individual node access costs, and the realistic access "price" model which allows various access "prices" of different node pairs further complicates the decision making. NSCC attempts to identify mistreatment-free object placement to incur implicit cooperation even among these selfishly behaving domains, and to further identify Nash equilibrium object placement from mistreatment-free object placements so that no domain can unilaterally change its placement and benefit while the others keep theirs unchanged, and to improve the cooperation performance with respect to fairness So far, using a game-theoretic approach NSCC seeks a global object placement in which the individual node access costs are reduced as compared to that when they operate in isolation and achieves Nash equilibrium. Our preliminary experiments with IBR verified its effectiveness. And we discuss the specific issues of NSCC's implementation in NDN.**

I. Introduction

This work proposes a scheme to promote autonomous and selfish NDN (Named Data Networking) [1] peering domains to cooperate in caching. NDN domains tend to maintain caches to store popular objects so as to access them with minimum costs. But individual domain can hold only limited number of objects due to its storage space constraint. Since it is cheaper for them to access cached copies from peering domains than that from remote Internet, these peering domains have incentive to cooperate in determining what objects to cache at each domain (object placement decisions) and share data [2]. For example, as illustrated in Figure 1, three campus networks in Colorado State (CSU, UCB and UCD) are peering with each other and it is cheaper for them to access data from each other than from remote Internet (the weights on edges represent access costs between ends). However, as autonomous domains, even if they cooperate, each network aims at its *own* cost reduction. CSU is not specifically interested in cost reduction at UCB or UCD, but is rather concerned with decisions that affect its own cost. That said, they are willing to cooperate in servicing requests for copies, but they are not willing to constrain their caching decisions based on requests from other networks.

This work models the aforementioned "cooperative caching" as a non-cooperative game among these peering domains, dubbed *not so cooperative caching* (NSCC), to decide what objects to cache at each domain under the constraints of their cache sizes so as to minimize the access costs at individual domains. We realize that a *rational* domain will join NSCC group if and only if it gains more as compared to that under greedy local cache policy (GL) when operating in isolation (*mistreatment-free requirement*) in terms of cost reduction. So NSCC attempts to identify mistreatment-free object placement to at least cancel out the overhead in cooperation and to incur implicit cooperation even among these selfishly behaving domains. Moreover, NSCC attempts to identify pure Nash equilibrium object placement (EQ placement) from mistreatment-free object placements so that no domain can unilaterally change its placement and benefit while the others keep theirs unchanged, and further improve the cooperation performance with respect to fairness, i.e., identifying the EQ placement in which the gain of the domain with minimum gain is maximized. And the realistic access "price" model in which the access costs of different domain pairs are various further complicates the cache decision making. The present work attempts to find a fair EQ placement through Iterative Best Response (IBR) method and carefully arranging the order of domains in IBR procedure at each iteration. We also consider specific issues of implementing NSCC game in NDN.

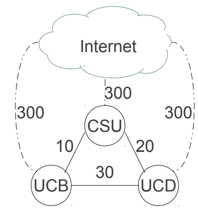

Fig. 1. NSCC group comprised of 3 peering campus networks.

II. Not So Cooperative Caching

NSCC is abstracted as follows. There is a set N of n domains peering with each other and a set M of m unit-sized objects. The access cost function is denoted as a $n \times (n+1)$ matrix D where $\forall i, j \in N, i \neq j$, d_{ij} and $d_{i,n+1}$ is the unit access costs from domain i to domain j and to remote Internet separately. And $t_l = d_{ii} < d_{ij} = d_{ji} \leq d_{i,n+1} = t_s$, i.e., it is cheaper for a domain to access data from local cache than from other peering domains, which is cheaper than from remote Internet; and the access costs between two domains are symmetric. Domain i is with cache size S_i objects and an access pattern (or users' demand) described by a rate vector r_i over M, $r_i = \{r_{i1}, \cdots, r_{im}\}$, where r_{ik} denotes the probability that domain i requests object k. We assume that demand estimates are given and can be obtained by statistics from users' access history or from suitable mechanisms for on-line content look-up, e.g. the distributed directory services in [3]. And the request set of i is $R_i = \{k \in M | r_{ik} > 0\}$ and $|R_i| > S_i$.

p_i denotes the placement of i , i.e., the set of objects stored

978-1-4799-1270-4/13/$31.00 ©2013 IEEE

at i; $p_i \subseteq R_i$ and $|p_i| = S_i$. $P = \{p_1, p_2, \cdots, p_n\}$ is referred to as a global placement and $P_{-i} = P - \{p_i\}$ denotes the *residual placement* of i under P. And $Q_{-i} = \cup_{p_j \in P_{-i}} p_j$ denotes the set of objects placed at all domains but i under residual placement P_{-i}. Let A_i be the set of placements available to i ($|A_i| = \binom{R_i}{S_i}$). Then the cost of i under P is defined as follow:

$$C_i(P) = \sum_{k \in p_i} r_{ik} t_l + \sum_{k \notin p_i, k \in Q_{-i}} r_{ik} d_{il(i,k)} + \sum_{k \notin p_i, k \notin Q_{-i}} r_{ik} t_s \quad (1)$$

where $d_{il(i,k)}$ is the access cost between i and the cheapest domain $l(i,k)$ that caches object k.

Definition 1: (Best Response) Given a residual placement P_{-i}, the best response for domain i is the placement $p_i \in A_i$ such that $C_i(P_{-i} + \{p_i\}) \leq C_i(P_{-i} + \{p_i^{'}\}), \forall p_i^{'} \in A_i, p_i^{'} \neq p_i$.

The best response at i is computed as follow: $g_{ik}(P_{-i})$ denotes the excess gain incurred by i from replicating object $k \in R_i$ under P_{-i} and is defined as follow:

$$g_{ik}(P_{-i}) = \begin{cases} r_{ik}(t_s - t_l) & \text{for } k \notin Q_{-i}, \\ r_{ik} \cdot (d_{il(i,k)} - t_l) & \text{for } k \in Q_{-i}. \end{cases} \quad (2)$$

Objects at domain i is sorted in descending order by $g_{ik}(P_{-i})$ and the S_i most valuable objects are selected to cache, which is the best response at i under P_{-i}.

Definition 2: (Stable Placement) A global placement P is stable if and only if it is composed of individual placements that are best responses.

Therefore stable placements are EQ placements of NSCC game in which no domain can unilaterally change its placement to increase its gain. Ragavendran et al. [4] proved that EQ placements are guaranteed to exist if the cost function forms an ultrametric, i.e., $\forall i, j, q \in N, d_{iq} \leq \max\{d_{ij}, d_{jq}\}$. Otherwise, it is NP-complete to determine the existence of EQ placement.

III. EQ Placements through Iterative Best Response

Definition 3: (Iterative Best Response (IBR)) Given an initial global placement $P^{(0)}$, start an iterative procedure where at iteration l the domains line up according to their order (predetermined) and performs:

1) domain i computes its best response $p_i^{(l)}$ to $P_{-i}^{(l,i-1)}$, after domain $i-1$ and before domain $i+1$;

2) $P^{(l,i)} = P_{-i}^{(l,i-1)} + \{p_i^{(l)}\}$.

$P^{(l,i-1)}$ is the global placement at iteration l (after $(i-1)$'s best response and prior to i's best response); $P_{-i}^{(l,i-1)}$ is the corresponding residual placement with respect to i. The IBR search stops and returns $P = P^{(t)}$ when at iteration t: $P^{(t)} = P^{(t-1)}, \forall i \in N$, i.e., when no domain can profit by re-placing.

We use IBR method to find stable placements. The convergence of IBR search to converge to EQ placement is in general hard to prove. Here we preliminarily use simulations to verify the convergence of IBR search. We give the following claims for EQ placements without proofs due to space limitation and the proofs would be given in the extended version of the paper.

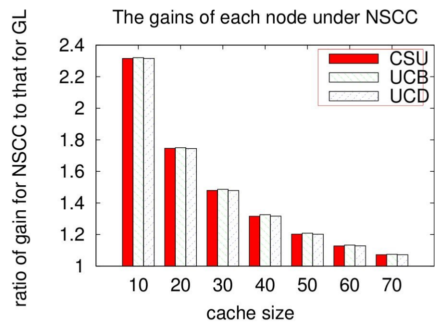

Fig. 2. The gain of each domain of NSCC game.

Claim 1: Each domain is mistreatment-free.

Claim 2: The price of anarchy of domain i for the NSCC game is upper bounded by t_s/d_i^{min}, and that of the NSCC group is upper bounded by t_s/d^{min} where $d_i^{min} = \min\{d_{ij} | \forall j \in N, j \neq i\}$ and $d^{min} = \min\{d_{ij} | \forall i, j \in N, j \neq i\}$.

We simulate the NSCC game in figure 1 and set $t_l = 0$, $t_s = 300$. There are 10000 objects in the system, the access patterns at 3 domains all follow Zipf-like distribution with exponent 0.6 and the domains are with equal cache sizes in terms of the number of objects. IBR starts from GL policy and the order of domains at each iteration is CSU, UCB, UCD. We use B_i, the ratio of the gain of node i under IBR to that under GL to measure the effectiveness of the game and the results are shown in figure 2. IBR converges to EQ placements after $1-2$ iterations for this game(at most 6 iterations in all experiments we have conducted) and as illustrated, each domain is mistreatment-free and gains more when cache sizes are smaller ($B_i > 2.31$ when each domain could hold 10 objects). Furthermore, the gains of domains are fairly close, i.e., it is fair among domains (the largest difference of B_i between different domains is less than 0.01). We also test the results when other 5 possible orders of domains are applied in IBR and the illustrated order gets the best fairness. We would explore how the order of domains based on their positions in topology would affect the fairness both in theory and through experiments in the future.

IV. Implementation Specific Issues

We plan to implement NSCC game with CCNX library [5] and run it on PlanetLab. NSCC game can be implemented either in a centralized way at certain third party or in a distributed way. If in a centralized way, all domains have to upload their access patterns to the third party which may incur considerable communication cost. Domains are not involved in the decision making and are notified of final placement. If in a distributed way, each domain has to notify others of its best response in each round of IBR and the communication cost is relatively smaller. Such information exchange needs to be implemented by NDN Interest and Data packets. If all domains are well-behaved, either way would be fine. Otherwise, we are not sure what kind of cases would tempt certain domain(s) to lie about their access pattern in the centralized way. Or in the distributed way, certain domain(s) may lie about or mislead other about their placements. Or certain domain(s) may fail or sometimes refuse to response others' requests.

Countermeasures for such issues should be developed. And we should decide the period of invoking the game to strike a balance between the overheads in information exchange and the benefit from adapting to demand dynamics, which may be challenging. Note that NDN naturally supports name-based multipath routing and thus routing should be manipulated such that data access requests would be sent to the cheapest data sources rather than all the sources.

V. Related Work

To the best of our knowledge, only a few recent works on game-theoretic aspect of cooperative caching. The work in [6], which serves as the seminal work on game-theoretic aspect of cooperative caching, studies selfish cooperative caching without consideration of storage limitation. Due to the limits on cache-capacity model, an important real-world restriction, the following works focus on the capacitated version [7], [8], [9]. They both consider distributed and capacitated selfish caching and follow the simplified access "price" model introduced in [10] where nodes are equidistant (equal access "price") from one to another and a special data source holds all objects. The work in [7] devises a cooperative caching strategy (TSLS) among selfish nodes such that Nash equilibrium object placement is obtained. The work in [8], [9] extends the work in [7] with node churn, i.e., random changes in the set of participating nodes in the group that may occur due to "join" and "leave" events, and studies corresponding game theoretic properties. Pollatos et al. [11] slightly extends the work in [7] to the case in which special data sources for different objects are at different distances. And Gopalakrishnan et al. [4] primarily focuses on the discussion of the existence of Nash equilibrium object placements in theory, does not devise a feasible algorithm to seek an object placement that enables selfish nodes to cooperate in caching, and not to mention experimental analysis. Rajahalme et al. [2] pioneers the exploration of the potential of data-oriented interdomain peering and data sharing in data-oriented networking, which precipitates the exploration of domains' incentive of "cooperative caching" in a more aggressive way. Our work differs from these above in that we extend the access cost model into a more realistic scenario which allows various access "prices" of different node pairs and we do not only focus on devising an algorithm to seek mistreatment-free EQ object placement, but also on the design of such a not so cooperative caching system in an NDN way in which users request content by names, information exchange among nodes uses NDN SYNC protocol [12] and a error checker component deals with misbehaving nodes.

VI. Conclusion & Future Work

This work proposes a scheme to promote autonomous and selfish NDN (Named Data Networking) peering domains to cooperate in caching, here dubbed *Not So Cooperative Caching (NSCC)*. We consider a network comprised of selfish nodes; each is with a caching capability and an objective of reducing its *own* access cost by fetching data from local cache or from neighboring caches. The challenge is to determine what objects to cache at each node (resulting in a global object placement) so as to induce low individual node access costs, and the realistic access "price" model which allows various access "prices" of different node pairs further complicates the decision making process.

The challenge is that NSCC attempts to identify mistreatment-free object placement to at least cancel out the overhead in cooperation and incur implicit cooperation even among these selfishly behaving domains. Moreover, NSCC attempts to identify pure Nash equilibrium object placement from mistreatment-free object placements so that no domain can unilaterally change its placement and benefit while the others keep theirs unchanged, and to further improve the cooperation performance with respect to fairness, i.e., identify the EQ placement in which the gain of the domain with minimum gain is maximized. So far, using a game-theoretic approach – Iterative Best Response (IBR), NSCC seeks a global object placement in which the access costs of individual selfish nodes would be reduced as compared to that when they operate in isolation. In IBR, each node makes its own placement decisions based on its local access pattern, the object placements at other nodes and the access "prices" from this node to other nodes. Our preliminary experiments with IBR verified its effectiveness. We also consider specific issues of implementing NSCC game in NDN.

Our next step is to refine the algorithm to the cases that we allow nodes in the group fail with some probability, which is common in the network environment and thus is further anchored in reality, and to define protocols for the detection of node failure and cheating and for information exchange among nodes, and finally to show how the scheme can be implemented in NDN which features routing by name, multipath routing and in-network caching capability.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. Braynard, "Networking named content," in *CoNEXT*, 2009.

[2] J. Rajahalme, M. Särelä, P. Nikander, and S. Tarkoma, "Incentive-compatible caching and peering in data-oriented networks," in *CoNEXT*, p. 62, 2008.

[3] M. V. Steen, F. J. Hauck, and A. S. Tanenbaum, "A model for worldwide tracking of distributed objects," 1996.

[4] R. Gopalakrishnan, D. Kanoulas, N. N. Karuturi, C. P. Rangan, R. Rajaraman, R. Sundaram, and R. Sundaram, "Cache me if you can: Capacitated selfish replication games.," in *LATIN*, pp. 420–432, 2012.

[5] "Ccnx project." http://www.ccnx.org/.

[6] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz, "Selfish caching in distributed systems: a game-theoretic analysis," in *Proc. 23rd annual ACM symposium on Principles of distributed computing*, pp. 21–30, ACM, 2004.

[7] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "Distributed selfish replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, pp. 1401–1413, Dec. 2006.

[8] E. Jaho, I. Koukoutsidis, I. Stavrakakis, and I. Jaho, "Cooperative content replication in networks with autonomous nodes," *Computer Communications*, December 2011.

[9] E. Jaho, I. Koukoutsidis, I. Stavrakakis, and I. Jaho, "Cooperative replication in content networks with nodes under churn," in *Proc. of the 7th international IFIP-TC6 networking conference*, pp. 457–469, 2008.

[10] A. Leff, J. L. Wolf, and P. S. Yu, "Replication algorithms in a remote caching architecture," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, pp. 1185–1204, Nov. 1993.

[11] G. G. Pollatos, O. A. Telelis, and V. Zissimopoulos, "On the social cost of distributed selfish content replication," in *Proc. of the 7th international IFIP-TC6 networking conference*, pp. 195–206, 2008.

[12] "Sync protocol." http://www.ccnx.org/releases/latest/doc/technical/.