

© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

Published in IEEE Communication Letter

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6459497>

Cooperative Caching Scheme for Content Oriented Networking

Ran Liu, Hao Yin, Xuejun Cai, Guangxi Zhu, Li Yu, Qilin Fan, and Jie Xu

Abstract— In this letter, we propose a novel cooperative caching scheme for the next-generation Internet — content oriented networking (CON), trying to minimize the content access delay for mobile users. We formulate the caching problem as a mixed integer programming model and propose a heuristic solution based on Lagrangian relaxation. Simulation results show that this scheme can greatly reduce content access delay.

Index Terms— Mobile Internet, content oriented networking, content delivery, cooperative caching, Lagrangian relaxation.

I. INTRODUCTION AND MOTIVATION

DURING the past five years, mobile Internet develops rapidly, which drastically changes the way user access web services, especially to the content services, while traditional Internet based on end-to-end principle is not well suited for mobile content services since it is originally designed for wired communications under the assumption of continuous source-to-destination path, short round trip time (RTT) and low data loss rate [1]. To this end, there has been a renewed interest in "clean-slate" design for the next-generation Internet, namely content oriented networking (CON) [2], [3], [4], striving to fundamentally improve the efficiency of mobile content services. One of its key ideas is in-network storage utilization. Taking advantage of rapid cost-reduction of semiconductor memory, it is affordable to equip routers with large storage space (we call them cache-routers hereafter), which enables the use of content caching as a basic network capability rather than as an external application-layer service as currently implemented in the Internet (e.g., CDN [5]).

Fig. 1 illustrates an abstract view of the CON architecture, in which cache-routers connect with each other and then construct a flat content network. Each cache-router caches some contents within its storage capacity under a certain policy, and each content has been cached in at least one cache-router. A mobile terminal accesses to CON through its local cache-router (i.e., the closest cache-router to the mobile

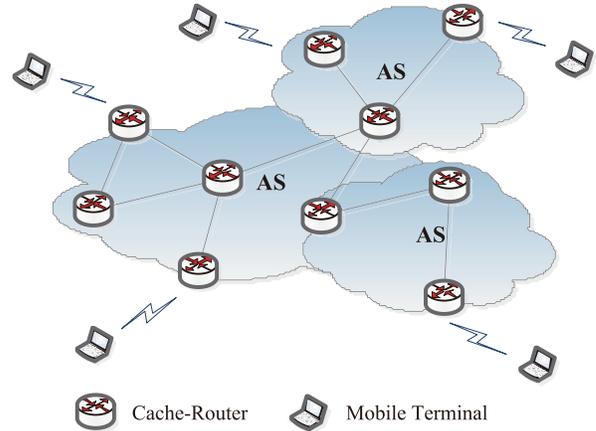


Fig. 1. Content oriented networking.

terminal). When the mobile terminal generates a request for a specific content, the request will be routed through CON, and when the content is found, the local cache-router will retrieve it from the nearest cache-router which has already cached the content and then serves the mobile terminal.

As we know, service delay is one of the most critical performance metrics for mobile services. First, power provision of mobile terminals is limited, and longer delay would lower power utilization efficacy [6]. Second, mobile users are less tolerant of service delay [7]. Last but not least, longer delay would directly lead to declined revenue of service providers [8]. In spite of the in-network storage characteristic of CON, mobile users may still suffer considerable delays due to high dynamics of mobile requests and user mobility [9], and the well studied caching policies such as least recently used (LRU) and first in first out (FIFO) replacement policies at individual cache-router would still not meet the delay requirement [10], [11]. To this end, we propose a Cooperative Cache-Router Caching (CCRC) scheme in this letter to further reduce the overall content access delay. In the next section, we formally formulate the problem as a mixed integer programming model. Section III provides the details of the cache scheme, with the simulation results on a carefully set scenario given in Section IV. At last, Section V concludes this letter.

II. PROBLEM FORMULATION

We model the CON topology to be a complete directed graph $G = (V, E)$, where V is the set of cache-routers and $E = (i, j : i, j \in V)$ is the set of links between cache-routers. The transmission delay on the link $i, j \in E$ is denoted by d_{ij} . Each cache-router $i \in V$ is associated with

Manuscript received July 28, 2012. The associate editor coordinating the review of this letter and approving it for publication was H.-P. Schwefel.

This work was funded by the Project 2011CB302601, 2012CB315800 supported by National Basic Research Program of China and the Project 61231010, 60972016, 60736012, 61222213, 61170290 supported by National Science Foundation of China

R. Liu, G. X. Zhu, and L. Yu are with the Dept. of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, PR China (e-mail: flashboy84@gmail.com, gxzhu@126.com, lyuyxt@sina.com.cn).

H. Yin, and Q. L. Fan are with the Dept. of Computer Science and Technology, Tsinghua University, Beijing, PR China (e-mail: h-yin@mail.tsinghua.edu.cn, amyfanqilin@gmail.com).

X.J. Cai is with the Ericsson China R&D Institute, Beijing, PR China (e-mail: xuejun.cai@ericsson.com).

J. Xu is with the School of Information Management, Wuhan University, Wuhan, PR China (e-mail: xuj@whu.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2013.020513.121680

three parameters: d_i , A_i^{store} , A_i^{serve} , where d_i denotes the processing delay of a client request, A_i^{store} means its storage capacity and A_i^{serve} presents its serving ability. In this model, A_i^{serve} is explicitly defined as session capacity (the maximum number of concurrent transmission sessions a cache-router could support) of cache-router i . The set of multimedia content to be cached and delivered to mobile clients is denoted by $C = \{1, 2, \dots, m\}$, where each content $c \in C$ has a capacity requirement denoted by b^c and a requesting demand in certain periods of time (e.g., 10 minutes) denoted by r_i^c . Due to the dynamic nature of the mobile environment, r_i^c would not stay unchanged. In real-world environment the value of r_i^c can be predicted using predictive analytics such as linear regression and Bayesian forecasting. Since prediction method is not the topic we want to discuss, we simply suppose we have already known the value of r_i^c in each time period.

It is notable that this letter considers a content placement problem in the viewpoint of mobile clients, trying to minimize the overall serving delay. To describe the model better, two decision variables that need to be calculated periodically are introduced:

$$x_{ij}^c = \begin{cases} 0 & \text{if cache-router } i \in V \text{ retrieves content } c \in C \text{ from} \\ & \text{cache-router } j \in V \text{ in a given update period,} \\ 1 & \text{otherwise,} \end{cases}$$

$$y_i^c = \begin{cases} 0 & \text{if cache-router } i \in V \text{ caches content } c \in C \text{ in a} \\ & \text{given update period,} \\ 1 & \text{otherwise.} \end{cases}$$

This model can be formulated as a mixed integer programming problem \mathcal{P} :

Minimize:

$$\sum_{i \in V} \sum_{j \in V} r_i^c ((d_{ij} + d_{ji})x_{ij}^c + d_j y_j^c) \quad (1)$$

Subject to

$$\sum_{c \in C} b^c y_i^c \leq A_i^{store}, \quad \forall i, c \quad (2)$$

$$\sum_{c \in C} \sum_{i \in V} r_i^c x_{ij}^c \leq A_j^{serve}, \quad \forall j, c \quad (3)$$

$$x_{ij}^c \leq y_j^c, \quad \forall i, j, c \quad (4)$$

$$\sum_{j \in V} x_{ij}^c = 1, \quad \forall i, c \quad (5)$$

$$r_i^c \in \{w | w \geq 0 \cap w \in \mathbb{Z}\}, \quad \forall i, c \quad (6)$$

$$x_{ij}^c \in \{0, 1\}, \quad \forall i, j, c \quad (7)$$

$$y_i^c \in \{0, 1\}, \quad \forall i, c \quad (8)$$

In this model, constraint (2) and (3) imply the constraints related to cache capacity and session ability of cache-routers, respectively. Constraint (4) ensures that a single cache-router could only request content from the cache-router which has already cached the content. Constraint (5) states that in one cache period, if a cache-router does not have the requested content, it could only request the content from one other cache-router, which would significantly reduce the complexity of data scheduling. Finally, constraint (6) ensures content request numbers are non-negative while constraint (7) and (8) impose binary restrictions on the two decision variables.

III. COOPERATIVE CACHING

In this section, we first simplify the problem by using Lagrangian relaxation and then propose a heuristic optimization algorithm based on subgradient.

A. Lagrangian Relaxation

Proposition 1. Problem \mathcal{P} is NP-hard.

Proof. We prove this proposition by restriction. Considering the special case of the integer programming problem with $C = c$ for all $i \in V$. When we drop the constraint (2) (3) and (6), and relax constraint (5) to be $\sum_{j \in V} x_{ij}^c \geq 1, \forall i, c$, the original problem \mathcal{P} can be simplified as a new problem \mathcal{RP} :

Minimize:

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \sum_{j \in V} f_j y_j \quad (9)$$

Subject to

$$\sum_{j \in V} x_{ij}^c \geq 1, \quad \forall i, c \quad (10)$$

(4), (7), (8)

where $c_{ij} = r_i(d_{ij} + d_{ji})$ and $f_j = \sum_{i \in V} r_i d_i$. Note that \mathcal{RP} is a classic Uncapacitated Location Problems (ULP), which is already known to be NP-hard. As a result, problem \mathcal{P} is NP-hard. \square

Considering the complexity of problem \mathcal{P} , it is unpractical to solve it directly. In this letter, we use Lagrangian relaxation to simplify problem \mathcal{P} . Since constraint (4) consists of two decision variables, we relax it in a Lagrangian fashion by associating Lagrange multiplier λ_{ij}^c to it. Consequently, we obtain the following relaxed problem, denoted by $\mathcal{LP}(\lambda)$:

Minimize:

$$\sum_{i \in V} \sum_{j \in V} r_i^c ((d_{ij} + d_{ji})x_{ij}^c + d_j y_j^c) + \sum_{i \in V} \sum_{j \in V} \lambda_{ij}^c (x_{ij}^c - y_j^c) \quad (11)$$

Subject to

(2), (3), (5), (6), (7), (8)

We divide problem $\mathcal{LP}(\lambda)$ into two subproblems: $\mathcal{SP1}(\lambda)$ and $\mathcal{SP2}(\lambda)$, where $\mathcal{SP1}(\lambda)$ is denoted by

Minimize:

$$\sum_{i \in V} \sum_{j \in V} (r_i^c (d_{ij} + d_{ji}) + \lambda_{ij}^c) x_{ij}^c \quad (12)$$

Subject to

(3), (5), (6), (7)

and $\mathcal{SP2}(\lambda)$ is denoted by

Minimize:

$$\sum_{i \in V} \sum_{j \in V} (r_i^c d_j - \lambda_{ij}^c) y_j^c \quad (13)$$

Subject to

(2), (6), (8)

Subproblem $\mathcal{SP1}(\lambda)$ can be further decomposed into $|V|$ subproblems in that $\mathcal{SP1}(\lambda) = \sum_{j \in V} \mathcal{SP1}^j(\lambda)$, where $\mathcal{SP1}^j(\lambda)$ is denoted by

Minimize:

$$\sum_{i \in V} (r_i^c (d_{ij} + d_{ji}) + \lambda_{ij}^c) x_{ij}^c \quad (14)$$

Subject to

(3), (5), (6), (7)

The solution of subproblem $\mathcal{SP1}^j(\lambda)$ is given by setting $x_{i^*j}^c = 1$ for cache-router i^* with minimum value of $r_i^c (d_{ij} + d_{ji}) + \lambda_{ij}^c$ and $x_{ij}^c = 0$ for all $i \in V \setminus \{i^*\}$

Subproblem $\mathcal{SP2}(\lambda)$ is a typical 0-1 programming problem. As a result, we can solve it in polynomial time by using simplex method.

B. Subgradient Optimization Algorithm

Getting the optimal solution to subproblem $\mathcal{SP1}(\lambda)$ and $\mathcal{SP2}(\lambda)$, we leverage subgradient optimization algorithm to get the approximate optimal solution to problem \mathcal{P} . The complete algorithm is described as follows:

- 1) Start with an initial multiplier λ^1 . Set the incumbent lower bound as $LB = 0$ and the incumbent upper bound as $UB = \infty$. Set $p = 1$.
- 2) Repeat the following steps until $\frac{UB-LB}{UB} < \epsilon$ or the maximum number of iterations has been reached.
 - Solve subproblem $\mathcal{SP1}(\lambda^p)$ and $\mathcal{SP2}(\lambda^p)$ respectively, get the optimal solutions (denoted by x^* and y^*) and results of the objective functions (denoted by $O(\mathcal{SP1}(\lambda^p))$ and $O(\mathcal{SP2}(\lambda^p))$), and calculate the result of the objective function of problem $\mathcal{LP}(\lambda^p)$ (denoted by $O(\mathcal{LP}(\lambda^p))$, where $O(\mathcal{LP}(\lambda^p))=O(\mathcal{SP1}(\lambda^p))+O(\mathcal{SP2}(\lambda^p))$). If $O(\mathcal{LP}(\lambda^p)) > LB$, set $LB = O(\mathcal{LP}(\lambda^p))$.
 - Put y^* into problem \mathcal{P} , reduce it to be an assignment problem, figure it out and get its upper bound UB .
 - Update Lagrange multiplier as follows: $\lambda^{p+1} = \max(0, \lambda^p + g^p s^p)$, where g^p and s^p represent subgradient vector and step-length at iteration p respectively. g^p is defined as $(g^p)_{ij}^c = x_{ij}^c - y_j^c$ and s^p is calculated as $s^p = \beta^p \frac{UB - O(\mathcal{LP}(\lambda^p))}{\|g^p\|^2}$, where $0 \leq \beta^p \leq 2$.
- 3) Set $p \leftarrow p + 1$.

IV. SIMULATION AND EVALUATION

A. Simulation Setup

Network model

We set up a rectangular simulation scenario and leverage GT-ITM [12] to construct the topology of cache-routers using Waxman 1 model. The probability that cache-router i directly connects with cache-router j is set to be $0.8d^{-\frac{D_{ij}}{0.9L}}$, where D_{ij} denotes the geographical distance between i and j , and L denotes the maximum distance between any two cache-routers. If two cache-routers connect directly with each other, the transmission delay between them will be distributed uniformly between $\max(0, \frac{1000D}{L} - 100)ms$ and $(\frac{1000D}{L} + 100)ms$. Also, the processing delay of any cache-router is distributed uniformly between $0ms$ and $100ms$.

User behavior

We use a random path to simulate users' motion path. A random starting point and a destination are selected for a mobile terminal at the beginning. The mobile terminal moves from the starting point to the destination at pre-set speed. When the mobile terminal reaches the destination, a new destination will be randomly chosen at once. This process repeats continuously until the simulation ends. The mobile user keeps on accessing particular streaming content when moving until the playback ends and then starts to request another streaming content. All contents of our simulation have the same size, and their popularity follows a Zipf distribution.

Comparable schemes

We further implement LRU (removing the content least recently used in the cache-router) and FIFO (removing the

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Session capacity of cache-routers	2000
Number of cache-routers	50
Number of multimedia content	100
Capacity requirement of multimedia content	100MB
Session duration of multimedia content	30mins
Zipf factor	0.85
ϵ	0.002
The maximum number of iterations	50
Cache size of cache-router	[0.5-5]GB
Number of mobile terminals	[10000-100000]
User speed	[0-10]m/s
Cache update cycle	10mins
Simulation duration	10hours
Area of simulation scenarios	15000*15000m ²

content first cached in the cache-router) scheme in the same environment. In these schemes, when a cache miss happens, the cache-router would get the content from the nearest cache-router which caches the content and is not overloaded.

Table I summarizes the experimental parameters that we use.

B. Performance Evaluation

In the performance evaluation, *average content access delay* has been taken as the primary performance metric, since the main target of our study is to reduce the service delay in mobile environment. We leverage three parameters, cache size of cache-router, number of users and move speed, to measure it.

Fig. 2 depicts the average content access delay as a function of cache size. Number of users and move speed are fixed to 10000 and $1m/s$ respectively. We change the cache size from $0.5GB$ to $5GB$. Obviously, since larger cache size leads to a higher hit rate, as cache size increases, content access delays of all three schemes decrease. Meanwhile, when cache size is small or large enough, the performances are very close, however, CCRC still outperforms LRU and FIFO on the whole. This is mainly because when cache size is quite small, what can be cached is limited, and the characteristics of all schemes are not fully shown; when cache size is large enough, most contents have already been cached, the differences among the three schemes make little impact on the cache hit rate. Owing to the fact that CCRC globally takes transmission and processing delay into consideration but LRU and FIFO decide what to be cached just by local user behavior, CCRC has better delay performance than the other two.

Fig. 3 shows the average content access delay as a function of the number of users. Cache size and move speed are set to be $4GB$ and $1m/s$ respectively, and then we change the number of users from 10000 to 100000. From Fig. 3, we can get two conclusions: (1) at the initial stage, content access delays of all three schemes increase rapidly as the number of users increases, and then the speed slows down when the number of users reaches 30,000; (2) CCRC performs the best, LRU shows intermediate performance, and FIFO is the worst. The reason for (1) is that at the beginning, the diversity of user requests gradually appears as the number of users increases, leading to decreasing cache hit rate; when the number of users

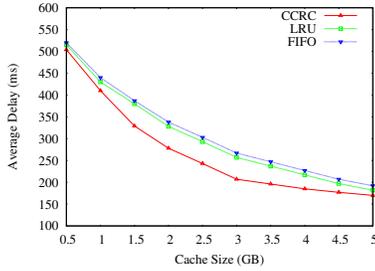


Fig. 2. Effect of cache size.

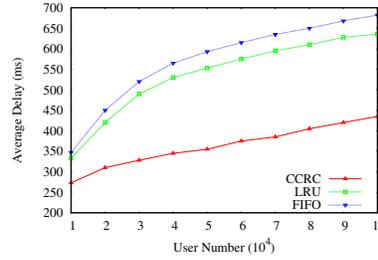


Fig. 3. Effect of users number.

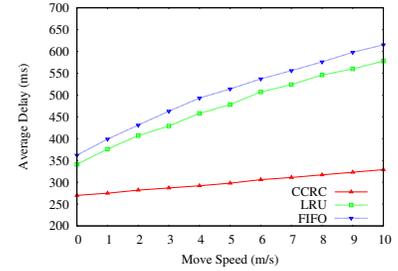


Fig. 4. Effect of move speed.

further increases, the impact of request diversity disappears. The reason for (2) is that CCRC holistically takes content popularity into consideration, however LRU just considers local popularity in one cache-router, and FIFO does not consider it at all. Intuitively, CCRC and LRU have higher cache hit rate. When a cache miss happens, CCRC ensures that a nearby cache-router could provide the content, but LRU could not.

Fig. 4 illustrates the average content access delay as a function of move speed. In this experiment, we fix cache size and number of users to be $4GB$ and 10000 respectively and increase move speed from $0m/s$ to $10m/s$. Apparently, with the increasing of move speed, the handover frequency among different cache-routers gets higher, which increases the number of user requests per unit time and reduces delay performance of all three schemes. Similarly, since CCRC holistically takes content popularity into account, its performance degradation is slower than LRU and FIFO. As showed in Fig. 4, for the highest speed of $10m/s$, the reduction of average delay by using CCRC scheme is largest within the considered parameter range, and the average delay of CCRC is 34% and 48% lower than LRU and FIFO, respectively.

We also measure on average how many multimedia contents will a cache-router update in each period. We set cache size, number of users and move speed to be $4GB$, 10000 and $1m/s$, and the results are 6.93, 5.17 and 4.95 when adopting CCRC, LRU and FIFO, respectively. Although CCRC leads to a higher content exchange possibility between cache-routers, the absolute differences among the three schemes are very small.

V. CONCLUSION

In this letter, focusing on the content placement problem for the next-generation Internet, we proposed a novel cooperative caching scheme based on Lagrangian relaxation. Trying to minimize content access delay for mobile users, a series of simulation experiments have been conducted to evaluate the delay performance of the proposed scheme. The results show that the proposed scheme can significantly improve the delay

performance compared to existing algorithms (i.e., LRU and FIFO) under various cache sizes, number of users and move speed.

As for future work, we intend to develop detailed analytical models. Furthermore, we will study more assess parameters, including cache size and maximize number of users to ensure acceptable delay. In order to improve caching decisions, We are interested in the possibility using path prediction algorithms to detect regular user movement patterns.

REFERENCES

- [1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. 2003 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 27–34.
- [2] T. Kaponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proc. 2007 ACM SIGCOMM*, pp. 181–92.
- [3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, and N. H. Briggs, "Networking named content," in *Proc. 2009 USENIX CoNEXT*, pp. 1–12.
- [4] J. Choi, J. Han, E. Cho, T. T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," *IEEE Commun. Mag.*, vol. 49, no. 3, pp. 121–127, Mar. 2011.
- [5] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: a platform for high-performance Internet applications," *Operating Systems Review*, pp. 2–19, 2010.
- [6] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. 2010 USENIX ATC*.
- [7] V. Roto and A. Oulasvirta, "Need for non-visual feedback with long response times in mobile HCI," in *Proc. 2005 International World Wide Web Conference*, pp. 775–781.
- [8] IDC, "Determining the return on investment of web application acceleration managed services," http://www.akamai.com/dl/whitepapers/IDC_ROI_Whitepaper_Oct2009.pdf, 2009.
- [9] C. H. Lee and D.Y. Eun, "On the forwarding performance under heterogeneous contact dynamics in mobile opportunistic networks," *IEEE Trans. Mobile Computing*, 2012.
- [10] J. Choi, J. Han, E. Cho, T. T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," *IEEE Commun. Mag.*, pp. 121–127, 2011.
- [11] G. Carofoglio, V. Gehlen, and D. Perino, "Experimental evaluation of memory management in content-centric networking," in *Proc. 2011 IEEE ICC*, pp. 1–6.
- [12] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. 1996 Joint Conference of the IEEE Computer Communications*, pp. 594–602.