3. CACHE ASSIGNMENT ALGORITHM

To solve the problem of assigning partitions originating from different egress routers to caches, we break it down into subproblems which assign partitions originating from the same egress router to caches in order to maximize the overall hit ratio while satisfying the constraint of the worst path stretch. These subproblems could be mapped into the MultiCommodity Facility Location (MCL) problem which is NP-hard. We propose a heuristic algorithm named *AssignCache* to solve it, by the following two means: 1) Assign exclusively a set of caches to a partition; 2) Limit the volume of requests which are directly sent to the egress router, while trying to find as many cache sets exclusively for partitions as possible. The least served ratio (*lsr* for short) is imposed on the cache sets found, which denotes the least request proportion of a partition should be served by caches. We first calculate the sum of the request proportions from ingress routers that a router r can serve for a specific egress router e, denoted by $CS_{r,e}$. Here, "can serve" means the path stretch imposed by using the router as cache does not exceed the path-stretch constraint. Function *served_ratio* in the following algorithm sums the request proportions from ingress routers in *served_set*.

Algorithm AssignCache

for all r in R
    calculate $CS_{r,e}$
Sort r in R in decreasing order of $CS_{r,e}$
selected_routers =∅
count = 0
for all r in R
    current_set = ∅
    served_set = ∅
    if r not in selected_routers
        current_set = current_set ∪ $\{r\}$
        selected_routers = selected_routers ∪ $\{r\}$
        served_set = served_set ∪ $can\_server(r)$
        while $served\_ratio$(served_set) < lsr
            $U = R -$ selected_routers
            $N = I -$ served_set
            Find a router $r' in\ U$, which can serve the largest and non-zero proportion of requests from ingress routers in N, If cannot, exit()
            current_set = current_set ∪ $\{r'\}$
            selected_routers = selected_routers ∪ $\{r'\}$
            served_set = served_set ∪ $can\_server(r')$
        count=count+1
        cache_sets[count]=current_set

4. EVALUATION WITH P2P TRAFFIC

We evaluate CPHR with a trace of P2P traffic which is provided by [5]. Basically, the original collected data are the distribution information of a large population of file replicas, which is attained from two-month passive measurement study of the Gnutella file sharing network. Like what the authors in [5] do, we assume that all the file replicas observed in an AS were downloaded from peers outside the AS sometime in the past. Here, we choose the file replicas observed in the network of AT&T. We map the IP address of the owner of each file replica into geographical location by Geo-IP database and choose the nearest PoP as its ingress point. As Figure 2 shows, the scattered red points are the file replicas (Only 1% of the total 1,026,081 replicas are randomly selected and shown in the figure), and the red bars on PoPs are the resulting request number from them. We assume the PoP at Los Angeles is connected to the single egress router from which all the file replicas originate. The green lines are the links of the shortest paths from all the PoPs to the egress point, along which the en-route caching mode is effective. We set the worst path stretch to 20ms which is roughly equal to the propagation latency from Los Angeles to New York, and set *lsr* to 0.7, which means at least 0.7 of the requests of a partition from all the ingresses are sent to assigned caches. With Algorithm *AssignCache*, 11 nodes are found for 11 partitions, which are plotted in Figure 2.

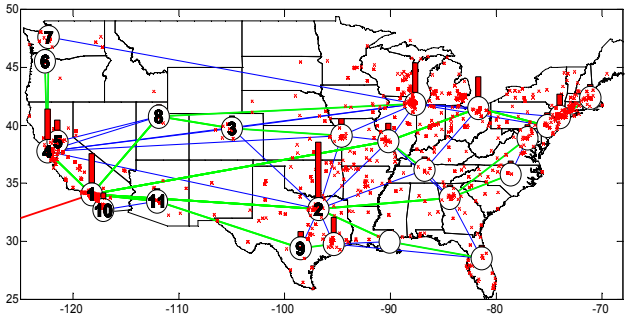

Figure 2. Evaluation scenario with a P2P trace of AT&T

The preliminary result of the trace-driven simulation is presented in Figure 3. CPHR can increase the overall byte hit ratio signifycantly for both cache replacement policy LRU and LFU. Setting cache size to 160G, CPHR can increase the overall byte hit ratio by 105.7% with LRU.

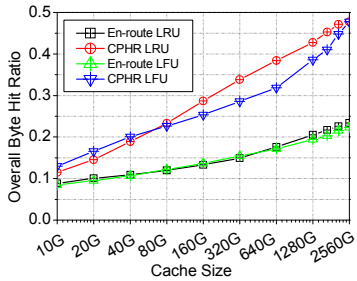

Figure 3. Overall byte hit ratio vs cache size

5. ACKNOWLEDGMENTS

This research is supported by the National High-tech R&D Program ("863" Program) of China (No.2013AA0l0605) and the National Science Foundation of China (No.61073172). Jun Bi is the corresponding author.

6. REFERENCES

[1] J. Choi, J. Han, E. Cho, T. T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," IEEE Communications Magazine, pp. 121–127, 2011.

[2] C. Fricker, P. Robert, J. Roberts and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," In Proc. of IEEE INFOCOM NOMEN, 2012.

[3] D. Perino, and M. Varvello, "A Reality Check for Content Centric Networking," Proceedings of the ACM SIGCOMM workshop on Information-centric networking, 2011.

[4] Z. Li, and G. Simon, "Time-Shifted TV in Content Centric Networks: the Case for Cooperative In-Network Caching," 2011 IEEE International Conference on Communications (ICC), 2011.

[5] Mohamed Hefeeda, and Behrooz Noorizadeh, "On the benefits of cooperative proxy caching for peer-to-peer traffic," IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 7, pp. 998–1010, 2010.

[6] L. Zhang et al. Named Data Networking (NDN) Project, 2010.http://named-data.net/ndn-proj.pdf.