

A Novel Implementation of Light Weight Cooperative Cache Management Algorithm through CDN for Effective Minimization of Bandwidth Cost

Bandari Muniswamy¹ , Dr.N.Geethanjali²

Research Scholar, Dept. of Computer Science &Technology, Sri Krishnadevaraya University, Anantapuramu, A.P. India¹

Associate Professor, Dept. of Computer Science &Technology, Sri Krishnadevaraya University, Anantapuramu, A.P.India²

ABSTRACT: Wireless networks are increasingly applied to extend applications with QoS constraints. Two problems come up when dealing with traffic with QoS constraints. One is admission control, which consists of finding out whether it is possible to meet the requirements of a set of nodes. The other is finding an optimal scheduling policy to conform to the needs of all guests. Content distribution networks (CDNs) which function to deliver web objects (e.g., Text files, applications, music and television, and so on) have seen tremendous growth since its emergence. To minimize the retrieving delay experienced by a user with a request for a network object, caching strategies are often applied - contents are duplicated at the boundaries of the mesh, which is nearer to the user such that the network distance between the user and the object is cut down. In the present report, we are developing light-weight cooperative cache management algorithm aimed at maximizing the traffic volume served from cache and minimizing the bandwidth cost. As a canonical scenario, we concentrate on a bunch of distributed caches, either linked directly or via a parent node, and articulate the content placement problem as a linear program in order to benchmark the globally optimal performance. Under certain symmetry assumptions, the optimal solution of the linear program is exhibited to possess a rather simple construction. Also interesting in its own right, the optimal structure offers valuable guidance for the purpose of low-complexity cache management and replacement algorithms. A new improved routing technique is proposed, whose primary destination is to arrive at the energy consumption balance and prolonging wireless sensor network lifetime. We show that the public presentation of the suggested algorithm is guaranteed to be within a constant divisor of the globally optimal performance, with far more benign worst case ratios than in prior employment, even in asymmetric scenarios.

KEYWORDS: Cooperative caching, Content distribution networks, globally optimal performance.

I. INTRODUCTION

Wireless networks have been widely deployed for a mixture of uses. Among the many applications that benefit from wireless networks, those with quality of service (QoS) constraints are increasingly of interest. They include video streaming, VoIP, real-time monitoring, networked control, etc. One common characteristic of these applications is that they have some requirements on throughput, delay, and delivery ratio. Hence, most current network mechanisms, which simply offer “best effort” service, are not adequate for these applications. While at that place has been much research interest in providing QoS [1], there is a shortage of analytical studies and theoretical guarantees on the service that can be offered. A central difficulty is that it is important to specifically take into account a most important characteristic of wireless network, that is, the lossy channel. As more and more devices, such as cordless telephones, Bluetooth and Zigbee devices, are accessing the same unlicensed channel as wireless networks, packet loss can no longer be ignored. Each type of content is

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

related to a particular desired quality of avail but, broadly speaking, low delays between request and reception is beneficial for all types of contentedness.

The Network Communications layer describes the network features of desktop machines connected over the Internet or small wireless or sensor-based devices that are connected in an ad-hoc way. The active nature of peers poses challenges in the communication paradigm. The Overlay Nodes Management layer covers the management of peers, which include the discovery of peers and routing algorithms for optimization. The Features Management layer deals with the security, dependability, fault resiliency, and aggregated resource availability aspects of preserving the robustness of P2P systems. The Services Specific layer supports the underlying P2P infrastructure and the application-specific components through scheduling of parallel and computation-intensive tasks, content and file management. Metadata describes the content stored across the P2P peers and the fix information. The Application-level layer is concerned with tools, applications, and services that are implemented with specific functionalities on top of the underlying P2P overlay infrastructure.

A content distribution network (CDN) is a broadcast system that routes requests for content originating from end-users to caches that can service these queries; the CDN [2] then returns content using a net that connects such caches to end-users. The motivation behind such a scheme is that obtaining content from a stash that is near a user is likely to have a shorter delay than from one that is farther off, due to a smaller number of hops to be covered. Nevertheless, putting a heavy demand for a popular component of substance on the nearest cache might be counterproductive, as the link capacity between each cache and the end-users are finite. In contrast to [3], [4], [9], we focus on specific topologies, motivated by real system deployments, and demonstrate that the optimal result of the relaxed integer program has a bare construction. In summation, we utilize the insight into the optimal social organization to develop efficient distributed content placement algorithms with far more favorable performance ratios than the ones in [3], [4], [9]. When it comes to the invention of actual content placement and eviction algorithms, a further important distinction with earlier works in different contexts manifests itself.

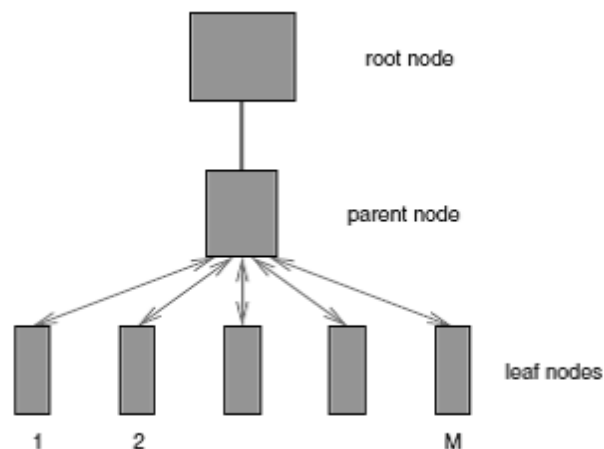


Fig. 1. Cache Cluster Overview.

In our setting, transferring content pro-actively into a peer cache carries a significant cost penalty, and it only makes sense to cache an item when it is actually requested. This restricts the role of cooperative caching algorithms proposed in different settings that ignore the bandwidth consumption when moving content around so as to get to the optimal position.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

II. RELATED WORK

In 1999 Napster [8] pioneered the idea of a peer-to-peer file sharing system supporting centralized file search facility. It was the first organization to pick out that requests for popular content need not be transmitted to a central server, but instead could be managed by many peers that hold the requested capacity. Such P2P file-sharing schemes are self-scaling in that as more peers join the system, they add to the aggregate download capability. Napster achieved this self-scaling behavior by utilizing a centralized search facility based on file lists provided by each peer; thus, it does not need much bandwidth for the centralized search. Such a system throws the issue of a single level of failure due to the centralized search mechanism. Nevertheless, a case filed by the Recording Industry Association of America (RIAA) tried to force Napster to close down the free-for-all file-sharing service for copyright digital music — literally, its killer application. All the same, this paradigm caught the imagination of platform providers and users alike. Gnutella [9–11] is a decentralized organization that administers both the search and download capabilities, building an overlay network of equals. It is the first system that makes usage of an unstructured P2P overlay network. An unstructured P2P system is composed of peers joining the net with some loose rules, without any prior knowledge of the regional anatomy. The network uses flooding as the mechanism to send queries across the overlay with a limited range. When a match receives the flood query, it transmits a list of all content matching the inquiry to the originating peer.

Tassioulas and Ephremides [17] have proposed a max weight scheduling policy and proved that it is throughput optimal. Neely [10] have further evaluated this policy and have shown that the policy achieves a constant average time lag. Shakkottai and Stolyar [14] have evaluated various scheduling policies to bear out a mix of real-time and non-real-time traffic. Johnson and Cox [7] have proposed a policy that purports to achieve both small packet delay and high user throughput. Duo and Bambos [4] have focused on the tradeoff between user fairness and organization performance and designed a policy for this determination. However, all these works lack a thorough theoretical study with provable performance guarantees. Raghunathan et al [12] and Shakkottai and Srikant [13] have developed analytical results on scheduling. Nevertheless, the destination of their works is to minimize the total number of expired packets among all users, which will necessarily be unfair to customers with poor channel qualities. Stolyar and Ramanan [16] aim at offering QoS guarantees on a per customer basis. Their approach offers asymptotic optimality only when the period is large. Kawata et al [8] have focused more on execution issues and enhancing QoS for the IEEE 802.11 mechanisms. Their simulations have been taken in a controlled environment where packet losses are infrequent. Other works [1] [2] have considered different performance metrics and modeling assumptions.

III. PEER-PEER OVERLAY NETWORKS

In this class, the overlay network assigns keys to data items and organizes its peers, into a graph that represents each data key to a peer-peer. This structured graph enables efficient discovery of data items using the given key. Nevertheless, in its simple form, this grade of systems does not support complex queries and it is necessary to store a copy or a pointer to each data object (or value) at the peer responsible for the data object's key. The QoS constraints of a client are described as follows: At the start of every period of length τ , where the length is measured in time-slots, each client $n \in \{1, 2, \dots, N\}$ generates one data packet. The parcels of all clients are to be given up to the AP within the next τ time slots before the close of the point.

If a bundle is not redeemed by the close of the point, it is marked as expired and removed from the system. Therefore, we can vouch that the delay of each delivered packet is less than τ . Further, client n requires a delivery ratio of at least q_n . That is, the proportion of expired packets cannot exceed $1 - q_n$. At long last, chewing over the nature of the unreliable wireless channels, client and accepts a channel reliability of p_n ; that is, the proportion of transmissions of the client n that are successfully driven home to the AP is p_n . This channel reliability reflects qualities of both uplink and downlink since a successful transmission includes the rescue of both the command message from the waiter and the data package from the specified node.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

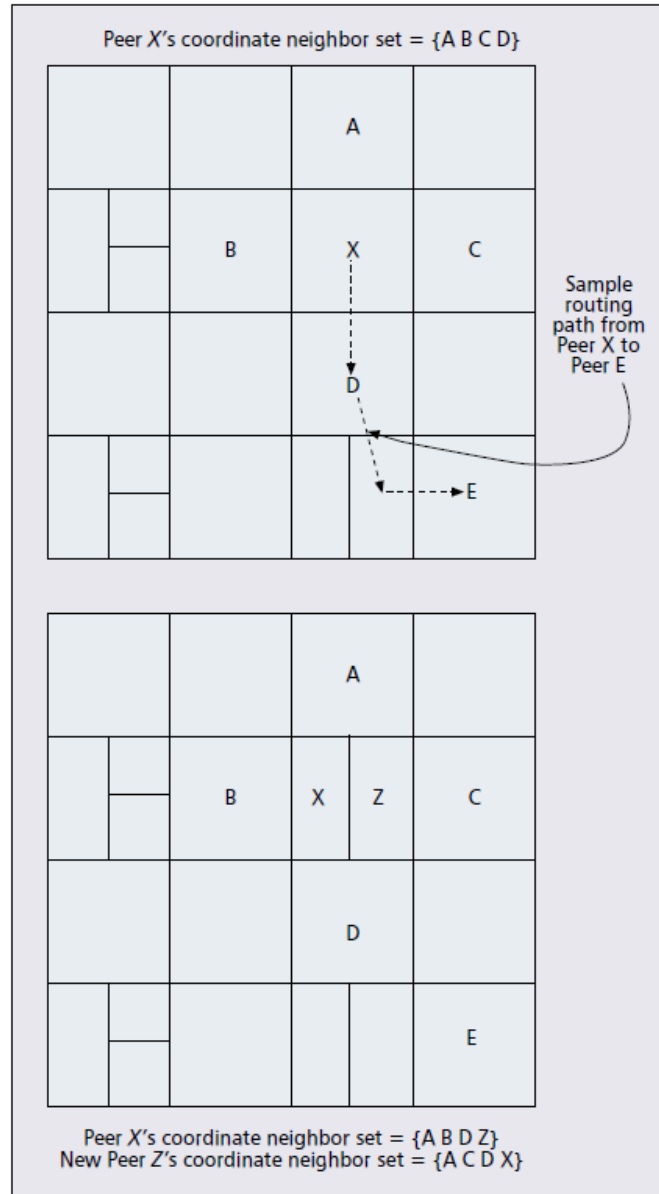


Fig.2 Example of 2-d space CAN before and after Peer Z joins

The architectural design is a virtual multi-dimensional Cartesian coordinate space on a multi-torus. This d -dimensional coordinate space is completely logical. The entire coordinate space is dynamically partitioned among all the peers (N number of peers) in the organization such that every peer possesses its individual, distinct zone within the overall distance. A CAN peer maintains a routing table that contains the IP address and virtual coordinate zone of each of its neighbors in the coordinate space. A CAN message includes the address coordinates. Using the neighbor coordinates, a peer route a message to its destination using a simple greedy forwarding to the neighbor peer that is nearest to the destination coordinates. CAN has a routing performance of $O(d \times N^{1/d})$ and its routing state is roughly $2 \times d$ bound. As shown in Fig.2,

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

which we adapted from the CAN paper [15], the virtual coordinate space used to store {**key**, **value**} pairs as follows: to store a pair {**K**, **V**}, **key K** is deterministically mapped onto a point P in the coordinate space using a uniform hash function. The lookup protocol retrieves an entry corresponding to key K, and any peer can use the same deterministic hash function to map K onto point P and then recover the corresponding value V from the point P. If the requesting peer or its immediate neighbors do not own the point P, the request must be routed through the CAN infrastructure until it hits the peer where P lays. A peer maintains the IP addresses of those peers that hold coordinate zones adjoining its zone. This circle of immediate neighbors in the coordinate space serves as a coordinate routing table that enables efficient routing between points in this quad.

IV. PROPOSED SYSTEM

Since our CDN model bears a resemblance to a switch, we are cheered by the Max-Weight scheduling algorithm that was established to be throughput optimal for a switch [8]. Unlike a switch, all the same, we suffer to take content placement, eviction and scheduling decisions at refresh instants. Imagine that at refresh instants we find a schedule such that:

$$X_{sd}^{c*}[k] = \arg \max_{\forall s,c,d} \sum_{X_{sd}^{c*}} q_s^c[k] C_{sd} X_{sd}^c[k]$$

We refer to the above optimization as Max Weight optimization independent of cache contents (MWI). Notice that the resolution does not specify which bits of content are to be evicted, and we could choose to simply evict a random subset of the cache contents that does not intervene with the agenda. Further, at the inter-refresh time instants, we ask a schedule with the proviso that it only incorporates content that is already present in the caches. We could again use a Max-Weight schedule, except that it must now be calculated subject to the presence of scheduled content (MWP). We refer to the policy that comprises of MWI at the refresh instants, and MWP at the inter-refresh instants as the Periodic Max-Weight scheduling algorithm (PMW).

Algorithm:

Select a node i and an item n . If an item n is currently not stored at node i , and it is higher utility than some item m that is currently stored at node i , then replace item m by item n .

$$\text{sub} \sum_{n=1}^N S_n x_{0n} \leq B_0$$

$$\sum_{n=1}^N S_n (P_n + (M - 1)q_n) \leq MB$$

$$P_n + x_{0n} \leq 1, n=1, \dots, N$$

$$q_n + x_{0n} \leq 1, n=1, \dots, N$$

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

where M denotes the number of ‘leaf’ nodes, $s_1; \dots; s_N$ denotes the collection of N content items, d_{in} denotes the demand for n -th content item in node i , c_{ij} denotes the cost associated with transferring content from node i to node j , x_{ijn} is a boolean indicates whether requests for the n -th item at node i are served from node j or not. The problem was simplified to a linear programming problem-back pack-type problem. The problem was further studied in two scenarios - intra level cache cooperation and inters level cache cooperation. In intra-level cache cooperation case, the knapsack-type problem was further simplified to a knapsack of size MB and $2N$ items of sizes

$$a_n = s_n ; a_{N+n} = (M - 1)s_n ; n = 1 \dots N.$$

The estimation of a largest debt first policy is to reckon the debt owed to each guest at the start of every period. The server then sets the priorities of clients according to their debts, clients with larger debt getting a higher precedence. Ties are broken by lexicographic order. In each time slot of the period, the client with the highest priority among those who have not yet succeeded in a transmission is scheduled to conduct. The sole difference between the two policies lies in the definitions of debts. The first policy, which we call the largest time-based debt first policy, tries to realize every client obtain a parcel of time at least as heavy as its implied work load. The next policy, which we call the largest weighted delivery debt first policy, approaches the QoS requirements more directly. It tries to get every client have a success rate higher than the desired delivery ratio, that is, q_n .

V. EVALUATION AND RESULTS

It is worth noting that the best between either full or zero return is often not a lot worse than the optimal content placement. Yet, neither full nor zero replication performs well across the entire range of α values, and it is important to set the level of return based on the steepness of the popularity distribution. In event of the Zipf-Mandelbrot distribution, one could in principle compute the optimal level of replication as function of α .

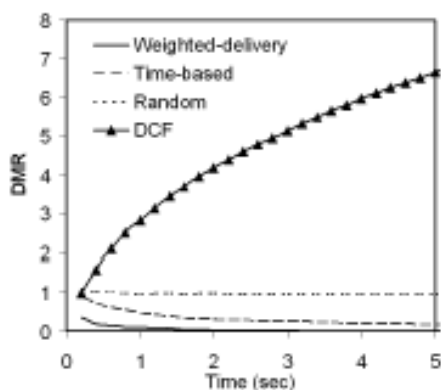
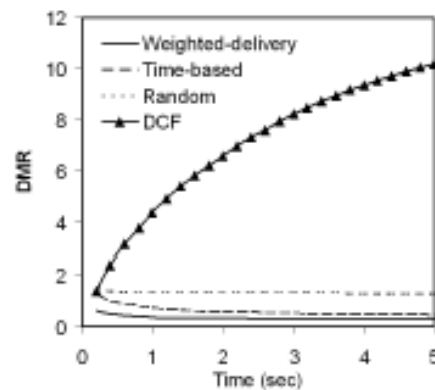


Fig.3.(a) Performance of a feasible set



(b) Performance of a infeasible set

Figures 4(a)–(c) show the results for cache sizes $B = 200$ GB, $B = 1$ TB, and $B = 2$ TB, respectively. Figure 4(d) shows similar results for a cache size $B = 1$ TB, where in addition the parent node has a cache of size $B_0 = 2$ TB. As the results demonstrate, the bandwidth costs markedly decrease with increasing values of α , reflecting the well-known fact that

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

caching effectiveness improves as the popularity distribution gets more outrageous. Even when $B = 200$ GB, which signifies that the collective cache space can have no more than 10% of the full content, caching reduces the bandwidth cost to a fraction of what they would have been in the absence of caching

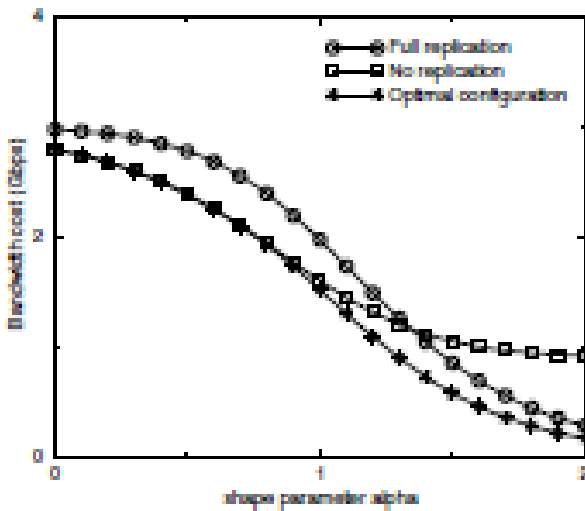
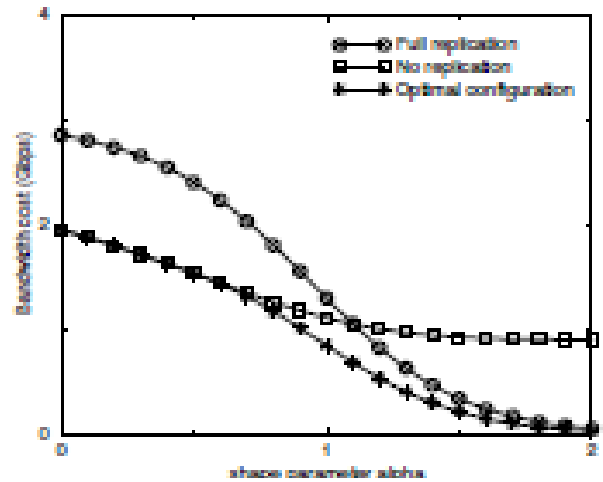
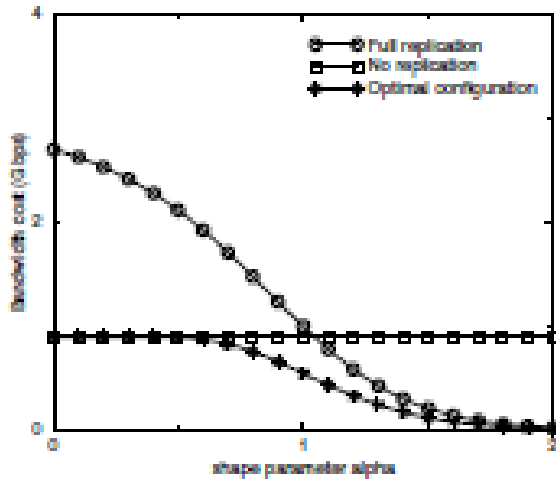


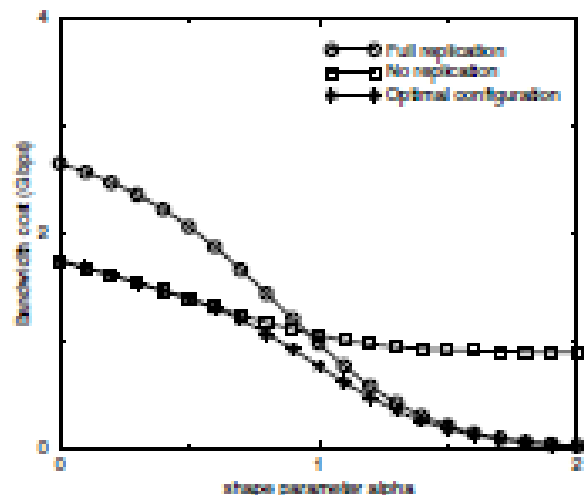
Fig.4. (a) Cache size $B = 200$ GB



(b) Cache size $B = 1$ TB



(c) Cache size $B = 2$ TB



(d) $B = 1$ TB, $B_0 = 2$ T

So far, we have assumed that the relative popularities of the various content items are known exactly. In practice, popularities are unlikely to be known with complete accuracy, and will require to be calculated. In society to study the impact of estimation errors, we carried an experiment where the popularities are no longer assumed to be cognized, but

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

estimated based on the actual received requests using geometrically smoothed average with some time constant T as a proxy for the length of the measurement window.

VI. CONCLUSION & FUTURE WORK

Wireless sensor networks available with limited source power through their life cycle. Since the shelling of the sensor node cannot be replaced or recharged, energy preservation is a crucial issue in the blueprint of the wireless sensor network infrastructure. We have developed lightweight cooperative cache management algorithms, which reach close to the globally optimum performance, with favorable worst-case ratios, even in asymmetric scenarios. Numerical solutions for typical popularity distributions demonstrate that the real performance is far more serious than the worst-case guarantees suggest.

We applied the abstraction of a switch to model the CDN, and our aim was to design algorithms that would be throughput optimum (stabilize the system) as well as yield short queues. Our main constraints were finite cache sizes and the periodicities with which content is refreshed in the caches. We showed how algorithms that engender large, negative Lyapunov drifts in the system are desirable, since such drifts budget short average queue lengths. The numerical results also illustrate the complex interaction between distributed cache cooperation and popularity estimation in the presence of dynamic content ingestion and aging. The public presentation of the proposed method depicts a serious improvement in the area of maximizing network lifetime using adaptive routing techniques.

Simulation results demonstrate the generality of the proposed fuzzy routing technique, so that the proposed routing technique could be utilized for any design framework. Future work includes streaming traffic with requests that have hard delay constraints, and which are dismissed if such a constraint cannot be satisfied. On that point are also exciting challenges and issues for applying P2P overlay networks in a hybrid wired and wireless web environment. We view the future of P2P overlay networks inexorably linked to the take up and subsequent commercial success of P2P overlay computing, personal area and ad-hoc networking, mobile location-based services, mirrored content delivery, and networked file-sharing, as models. We may also suppose that the prevalent problems of the Internet — controlling spam, maintaining directory services, multicasting content, and so on — have intuitive solutions from various P2P overlay network schemes. Nevertheless, in order to go ahead, the development community needs to see the applicability of several schemes for Structured and Unstructured P2P overlay network models. This study has been a minor endeavor to come up to this demand.

REFERENCES

- [1] C. Labovitz, D. McPherson, and S. Iekel-Johnson, “Atlas Internet Observatory 2009 annual report,” in *Proc. NANOG-47*, MI, Jun. 2009.
- [2] R. M. P. Raghavan, *Randomized Algorithms*. New York: Cambridge University Press, 1995.
- [3] P. Cao and S. Irani, “Cost-aware WWW proxy caching algorithms,” in *Proc. 1997 USENIX Symposium on Internet Technology and Systems*, Berkeley, CA, Dec. 1997.
- [4] K. Psounis and B. Prabhakar, “Efficient randomized web-cache replacement schemes using samples from past eviction times,” *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 441–455, Nov. 2002.
- [5] N. Laoutaris, O. Telelis, Orestis, V. Zissimopoulos, and I. Stavrakakis, “Distributed selfish replication,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 12, pp. 1401–1413, Dec. 2006.
- [6] S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithms for content distribution networks,” in *Proc. IEEE INFOCOM 2010*, San Diego, CA, Mar. 2010.
- [7] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
- [8] Napster, available at <http://www.napster.com/>
- [9] (2001) Gnutella development forum, the Gnutella v0.6 protocol, available at http://groups.yahoo.com/group/the_gdf/files/
- [10] (2002) Gnucleus, the Gnutella Web caching system, available: at <http://www.gnucleus.com/gwebcache>
- [11] (2002) Gnutella ultrapeers, available at http://rfcgnutella.sourceforge.net/developer/testing/Ultrapeers_1.0.html
- [12] F. Dabek et al., “Towards a Common API for Structured Peer-to-peer Overlays,” *Proc. 2nd Int’l. Wksp. Peer-to-Peer Systems (IPTPS 2003)*, Berkeley, California, USA, Feb. 20-21, 2003.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

- [13] B. Karp et al., "Spurring Adoption of DHTs with OpenHash, a Public DHT Service," Proc. 3rd Int'l. Wksp. Peer-to-Peer Systems (IPTPS 2004), Berkeley, California, USA, Feb. 26-27, 2004.
- [14] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," Proc. IPTPS, Cambridge, MA, USA, Feb. 2002, pp. 53–65.
- [15] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A Scalable and Dynamic Emulation of the Butterfly," Proc. ACM PODC 2002, Monterey, CA, USA, July 2002, pp. 183–92.
- [16] P. Sarkar, J.H. Hartman (2001). Hint-based cooperative caching. ACM Trans. Comp. Syst. 18, 387–419.
- [17] T. Silverston, O. Fourmaux (2007). Measuring P2P IPTV systems. In: Proc. ACM NOSSDAV '07.
- [18] M. van Steen, F.J. Hauck, A.S. Tanenbaum (1996). A model for worldwide tracking of distributed objects. In: Proc. TINA '96, 203–212.
- [19] C. Swamy (2004). Algorithms for the data placement problem. Preprint.
- [20] G.M. Voelker, E.J. Anderson, T. Kimbrel, M.J. Feeley, J.S. Chase, A.R. Karlin, H.M. Levy (1998). Implementing cooperative prefetching and caching in a globally-managed memory system. In: Proc. ACM SIGMETRICS '98.
- [21] National Institute of Standards and Technology (NIST), "Secure hash standard," U.S. Department of Commerce, National Technical Information Service FIPS 180-1, Apr. 1995.
- [22] F. Dabek *et al.*, "Wide-area Cooperative Storage with CFs," *Proc. 18th ACM Symp. Operating Systems Principles*, 2001, pp. 202–15.