

Dominating Set Theory based Semantic Overlay Networks for Efficient and Resilient Content Distribution

J. Amutharaj

Arulmigu Kalasalingam College of Engineering/Department of CSE, Srivilliputhur, India.

Email: amutharaj@yahoo.com

S. Radhakrishnan

Arulmigu Kalasalingam College of Engineering/Department of CSE, Srivilliputhur, India.

Email: srk@akce.ac.in

Abstract— Recently overlay networks have emerged as an efficient and flexible method for content distribution. An overlay network is a network running on top of another network, usually the Internet. This network is formed by a subset of underlying physical nodes. The connections between the overlay nodes are provided by overlay links, each of which is usually composed of one or more physical link. These networks are employed in many settings to provide logical communication infrastructure over an existing communication networks. The main objective of the overlay network is to reduce routing path lengths, stretched by the overlay routing process. In the existing solutions developed, a kind of fixed infrastructure in the form of excessive message exchange is necessary to guarantee good overlay properties.

The scope of our effort is to construct an overlay network based on Dominating Set Theory to optimize the number of nodes for large data transfer. Fast Replica algorithm is applied to reduce the content transfer time for replicating the content within the semantic network. A dynamic parallel access scheme is introduced to download a file from different peers in parallel from the Semantic Overlay Network (SON), where the end users can access the members of the SON at the same time, fetching different portions of that file from different peers and reassembling them locally. That is, the load is dynamically shared among all the peers. To eliminate the need for retransmission requests from the end users, an enhanced digital fountain with Tornado codes is applied. Decoding algorithm at the receiver will reconstruct the original content. In this no feedback mechanisms are needed to ensure reliable delivery. This paper analyzes the performance of sequential unicast, multiple unicast and fast replica with tornado content distribution strategies in terms of content replication time and delivery ratio. This paper also analyzes the impact of dominating set theory for the construction of semantic overlay networks.

Index Terms—Semantic Overlay Networks, Dominating Set Theory, Multicasting, Fast Replica, Content Distribution, Replication Time, Delivery Ratio.

I. INTRODUCTION

Peer-to-peer networks are emerging as a significant vehicle for providing distributed services (e.g. search, content integration and administration) both on the Internet [1] and in enterprises. Content Delivery Networks (CDN's) based on a large-scale distributed network of sites located closer to the edges of the Internet are used for efficient delivery of digital content including software packages and multimedia content. Locating content in decentralized peer-to-peer system is a challenging problem. Ensuring the availability of content on the Internet is expensive and only few options are available. They use premium content hosting services, build and manage their own content distribution infrastructures, or contract with Content Delivery Networks [2].

The main goal of the CDN's architecture is to minimize the network impact in the critical path of content delivery as well as to overcome the overload problem that is a serious threat for busy sites serving popular contents. For typical web documents served via CDN, there is no need for active replication of the original content at the edge servers. For large documents, software packages and media files, it is desirable to replicate these files at edge servers in advance. For large files it is a challenging, resource-intensive problem [3], e.g. Media files can require significant bandwidth and download time due to their large sizes.

In order to offload popular servers and improve end-user experience, copies of popular content are often stored in different locations. With mirror site replication, documents from a primary site are proactively replicated at secondary sites. When a copy of the same document exists at multiple servers, choosing the server that provides the best response time is not trivial and the resulting performance can dramatically vary depending on the server selected [4,5].

Instead of downloading the entire document from one server, a user downloads different parts of the same

document from each of the servers in parallel. Once all the parts of the document are received, the user reconstructs the original document by reassembling the different parts. There are several advantages while using a dynamic parallel access. First, as the block size is small, a dynamic parallel access can easily adapt to the changing network/server conditions. Second, as the client is using several connections to different servers, a parallel access is more resilient to congestion and failure in the network/server. Third, the server selection process is eliminated since clients connect to all available servers with a document copy. Fourth, the throughput seen by the client will increase. There is an overhead incurred when opening multiple connections and extra traffic generated to perform block request.

Laurent Massoulie [16] proposed an algorithm called the localizer which reduces network load, helps to evenly balance the number of neighbors of each node in overlay, sharing the load and improving the resilience to random node failures or disconnections. Vasileios et al. [17] proposed a set of mechanisms designed to detect and repair errors in the data stream, by utilizing the highly redundant connectivity in overlay networks, also most nodes can effectively detect corrupted data streams even in the presence of multiple tampering nodes. Ying Zhu et al. [18] evolved a distributed algorithm that uses the strategy of progressively and adaptively evolving the overlay topology over time toward high-quality topologies, with respect to end-to-end throughput of data dissemination. Bo Han and Weijia Jia [19] insisted a novel distribution algorithm called Area algorithm for CDS formation in wireless ad hoc networks with time and message. Natarajan Meghanathan [20] demonstrated that it is possible to determine the sequence of stable connected dominating sets provided the complete knowledge of the future topology changes is given in detail.

Trac N. Nguyen [21] showed that the minimum D-Hop connected dominating set problem is NP-complete for planar unit disk graphs with maximum degree 4, by comparing the performance of several known 1-hop heuristics. Jason et al. and D.T. Huynh [22] showed that the problem of adapting minimum connected D-Hop Dominating sets to local topology changes in wireless Ad Hoc networks is NP-complete by proposing a heuristic algorithm. Jie Wu et al. [23] proposed a notion of an extended Dominating set where each node in an Ad Hoc network is covered by either a Dominating neighbor or several 2-Hop Dominating neighbors.

Yam in li et al. [24] proposed an efficient, distributed and localized algorithm for finding an almost connected Dominating set of small size on wireless Ad Hoc networks, which finds a CDS faster provided the size of the found CDS is comparatively smaller. Amutharaj and Radhakrishnan [30] designed a dominating set theory based semantic overlay network for efficient content distribution. They analyzed the performance of sequential unicast, multiple unicast and fast replica content distribution schemes in terms of content distribution time.

They also estimated the encoding and decoding times for tornado codes.

II. DOMINATING SET THEORY BASED SEMANTIC OVERLAY NETWORKS

An overlay network is a virtual network of nodes and logical links that is built on top of existing network with the purpose to implement some new network services that is not available in the existing network. In Semantic Overlay Networks (SON's), nodes with semantically similar content are "clustered" together and a node may be a member of more than one cluster. Queries are routed to the appropriate SON's increasing the chances that matching files will be found quickly and reducing the search load on nodes that have unrelated content. SON's can significantly improve query performance and at the same time allow users to decide what content to put in their computers and with whom to connect.

We propose that the clustering of nodes of the SON can be done by forming the dominating set. Dominating Sets [19, 20, 21, and 22] have been used successfully in routing in MANETs [25, 26, and 27] and sensor networks [28, 29]. The rule based on dominating set can optimize the number of nodes in the SON by finding the sub graph of the original network. A dominating set is a set of nodes such that any node in the network is a neighbor of some element in the set [12]. In particular, by $G = (V, E)$ we denote a graph, where V is the set of nodes and E is the set of edges. Two nodes u and v are adjacent vertices if there is an edge $\{u, v\} \in E$. The neighborhood $N(v)$ of node v is the set of nodes adjacent to v . A dominating set of G is a subset V' of V such that every node $u \in V \setminus V'$ is adjacent to at least one node $v \in V'$. The nodes selected in dominating set are connected and their number is significantly reduced.

In a peer-to-peer system, nodes typically connect to a small set of random nodes and queries are propagated along these connections. Such query flooding tends to be very expensive. We propose that node connections be influenced by content, so that for example, nodes having many multimedia files will connect to other similar nodes. Thus semantically related nodes form a SON [6]. Peer-to-peer systems offer the potential for low cost sharing of information, autonomy and privacy. However, query processing in current peer-to-peer systems is very inefficient and does not scale well.

This inefficiency arises because most peer-to-peer systems create a random overlay network where queries are blindly forwarded from node to node. As an alternative, there have been proposals for rigid peer-to-peer systems that place content at nodes based on hash functions, thus making it easier to locate content later on. Although such schemes provide good performance for point queries they are not as effective for approximate, range or text queries. Furthermore in general, nodes may not be willing to accept neither arbitrary content nor arbitrary connections from other nodes [7].

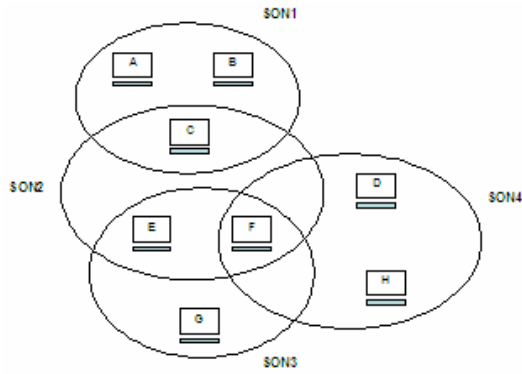


Figure 1. Semantic Overlay Networks

“Figure. 1,” shows a Semantic Overlay Networks where nodes with semantically similar content are “clustered” together and a node may be a member of more than one cluster. From the “Figure. 1,” the nodes are clustered as follows.

- Nodes A, B and C are clustered together (SON1)
- Nodes C, E and F are clustered together (SON2)
- Nodes E, F and G are clustered together (SON3)
- Nodes F, D and H are clustered together (SON4)

The predicted transfer time of a content of size ‘S’ can be obtained by:

$$P_{TT} = E_{(TS)} + E_{(LSS)} + E_{(LCA)} \quad (1)$$

Where $E_{(TS)}$, is the mean request waiting time (the average time that a request spends in the socket’s arrival queue of a server before it is handled by a thread), $E_{(LSS)}$ is the transfer time spent during the slow start phase at the beginning of the download and $E_{(LCA)}$ is the transfer time spent during the congestion avoidance phase.

A node joining the system, first floods the network with requests for the hierarchy. Then, the node runs a content classifier based on the hierarchy obtained on all its contents. Then, a node classifier assigns the node to specific SON’s. This can be done again by flooding the network until nodes in that SON are found. When the nodes issue a query, first it classifies it and sends it to the appropriate SON. After the query is sent to the appropriate SON, nodes within the SON find matches by using some propagation mechanism such as flooding or super peers. In this paper dominating set is used to form the SON and the algorithm used for content distribution in the SON is the Fast Replica Algorithm, which provides an efficient and reliable replication of large files in the Internet environment [8].

III. FAST REPLICA ALGORITHM

Files are replicated from many hosts in the system based upon which nodes download these files. Whole file replication is simple to implement and has a low state cost. It must only maintain the state proportional to the number of replicas. The cost of replicating the entire file in one operation can be cumbersome in both space and

time, particularly for systems that support applications with large objects such as audio, video, and software packages. Block level replication divides each file objects into an ordered sequence of fixed size.

This has a benefit of naming the individual parts of an object independently. A block-level system may download different parts of an object simultaneously from different nodes and it reduces the overall download time. Since the unit of replication is small and fixed, the cost to replicate an individual block can be small and distributed among many peers.

A novel algorithm called fast replica is proposed for an efficient and reliable replication of large files in the SON [8]. There are few basic ideas that are exploited in fast replica. In order to replicate a large file among ‘n’ nodes, the original file is partitioned into ‘n’ sub files of equal size and each sub file is transferred to a different node in the group. After that, each node propagates its sub file to the remaining nodes in the group. Thus instead of the typical replication of an entire file to ‘n’ nodes by using ‘n’ internet paths connecting the original node to the replication group, fast replica exploits $n \times n$ diverse internet paths within the replication group where each path is used for transferring $1/n$ th of the file.

We analyze the performance of the fast replica in the small and its potential performance benefits, and outline the configurations and network conditions when fast replica may be inefficient. Then using the fast replica in the small as the induction step, we design a scalable fast replica algorithm, which can be used for replication of large files to a large number of nodes.

A. Distribution step

As shown in “Figure. 2,” the originator node N_0 opens n concurrent network connections to nodes $\{N_1 \dots N_n\}$, and sends to each recipient node N_i ($1 \leq i \leq n$) the following;

- A distribution list of nodes $R = \{N_1, \dots, N_n\}$ to which sub file F_i has to be sent on the next step.
- Sub file F_i .

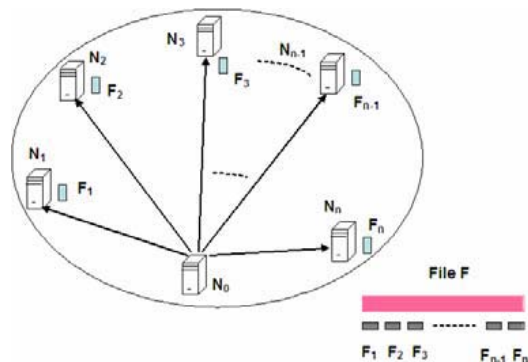


Figure 2. Distribution Step in Fast Replica

B. Collection step

After receiving file F_i , node N_i opens n-1 concurrent network connections to remaining nodes in the group and

send sub file F_i to them as shown in “Figure. 3,” for node N_1 .

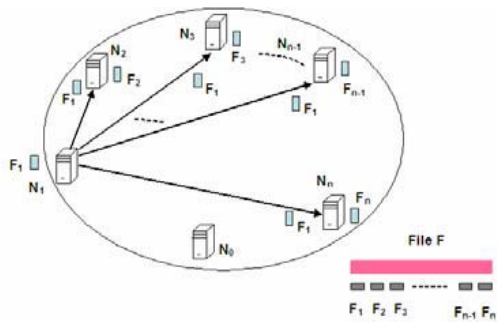


Figure 3. Collection Step in Fast Replica

C. Resilience step

We extend the use of fast replica algorithm to deal with node failures. For example, if node N_1 fails during either transfer, then this event may impact all nodes N_2, \dots, N_n in the group because each node depends on node N_1 to receive sub file f_1 . In the described scenario as shown in Figure. 4, node N_i is acting as a recipient node in the replication set. If a node fails when it acts as the origin node N_i , this failure impacts all of the replication groups in the replication sub tree rooted in node N_i .

The reliable fast replica algorithm proposed efficiently deals with node failures by making the local repair decision within the particular group of nodes. It keeps the main structure of the fast replica algorithm practically unchanged while adding the desired property of resilience to node failure.

In reliable fast replica, the nodes of each group are exchanging the heartbeat messages with their origin node. The heartbeat messages from nodes to their origin node are augmented with additional information on the corresponding algorithm. Once the content is distributed in the SON, the receiver has to recollect all the content from the SON in a parallel manner. A receiver can reconstruct the original source data once it receives a sufficient number of packets by decoding. For this encoding and decoding we used Tornado codes, in which retransmission is not needed.

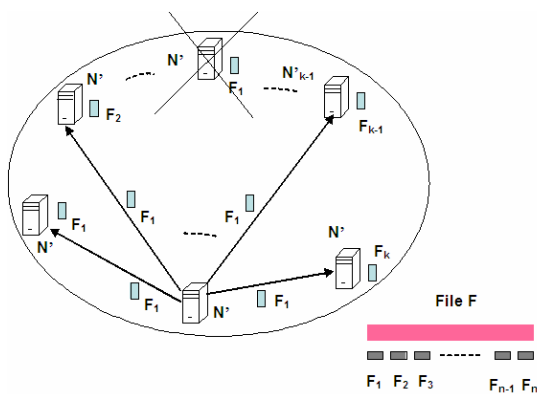


Figure 4. Fast Replica – Resilience step

IV. TORNADO CODES

The receiver can run the Tornado decoding algorithm in real-time as the encoding packets arrive, and reconstruct the original file as soon as it determines that sufficiently many packets have arrived. The basic principle behind the use of erasure codes is that the original source data, in the form of a sequence of ‘k’ encoded packets, along with additional redundant packets, are transmitted by the sender and the redundant data can be used to recover lost source data at the receivers.

The following points need to be considered when determining the size of the blocks requested.

- The number of blocks chosen should be much larger than the number of mirror servers that are accessed in parallel.
- Each block should be small enough to provide a fine granularity of striping and ensure that the transfer of the last block requested from each server terminates at about the same time.
- Each block should be sufficiently large enough to keep the inter block idle time small compared to the download time of a block.
- The document requested via parallel access must be sufficiently large.

V. EXPERIMENTAL RESULTS

A. Performance metrics for fast replica

Let Time denote the transfer time of file F from the original node N_0 to node N_i as measured at node N_i . Two performance metrics: average and maximum replication times are considered. In idealistic setting all the nodes and links are homogeneous, and let each node can support n network connections to other nodes at B bytes/sec. Then,

$$\text{Time}_{\text{distribution}} = \text{Size}(F) / (nxB) \tag{2}$$

$$\text{Time}_{\text{collection}} = \text{Size}(F) / (nxB) \tag{3}$$

For Fast Replica

$$\begin{aligned} \text{Time}_{FR} &= \text{Time}_{\text{distribution}} + \text{Time}_{\text{collection}} \\ &= 2 \times \text{Size}(F) / (nxB) \end{aligned} \tag{4}$$

For Multiple Unicast

$$\text{Time}_{MU} = \text{Size}(F) / B \tag{5}$$

$$\begin{aligned} \text{Replication Time Speedup} &= \text{Time}_{MU} / \text{Time}_{FR} \\ &= n / 2 \end{aligned} \tag{6}$$

In this work, the following distribution schemes are compared.

- *Fast Replica* - files are replicated from many hosts in the system based upon which nodes download these files.

- *Sequential Unicast* - approximates distribution via IP multicast, measures transfer time of entire file from the source to each recipient independently.
- *Multiple Unicast* - simultaneously transfers the entire file to all the recipient nodes by using concurrent connections.

In the experimental outcome, fast replica significantly outperforms Multiple Unicast especially for larger data transfer. For configuration of 8 nodes and a speedup value of 4, the performance benefits are

- 13 (max) times for a 1.5 MB file
- 6.5 (max) times for a 36 MB file

B. Performance of different content distribution schemes

We experimented with 12 different size files; 100 KB, 750 KB, 1.5 MB, 3 MB, 4.5 MB, 6 MB, 7.5 MB, 9 MB, 36 MB, 54 MB, 72 MB, 128 MB and 8 nodes. “Figure. 5,” shows the content replication time measured by different, individual recipient nodes for different file sizes of 100 KB, 750 KB, 1.5 MB, 3 MB, 4.5 MB, 6 MB, 7.5 MB, 9 MB, 36 MB, 54 MB, 72 MB, 128 MB when 8 nodes are in the replication set. High variability of replication time under Multiple and Sequential Multicast is identified. File replication time under Fast Replica across different file sizes in an 8 node replication set are much more stable and predictable. Hence, Fast replica with tornado encoding/decoding outperforms sequential multicast, multiple unicast content distribution schemes. This is depicted in Figure. 5.

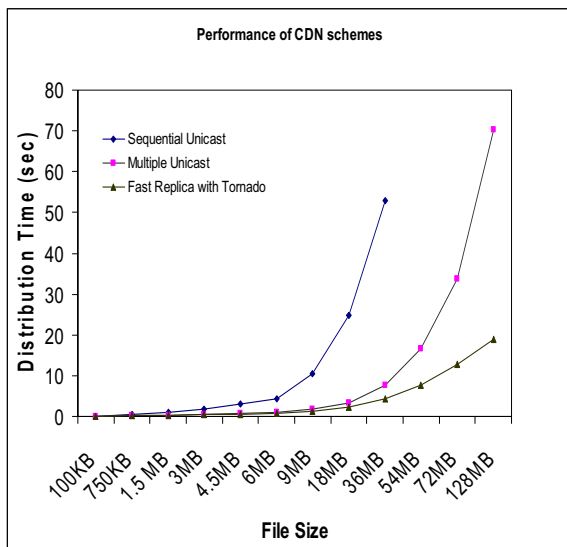


Figure 5. Content Distribution Times for various schemes

C. Performance metrics for Tornado codes

Given input parameters K and N, tornado codes are typically designed to take a set of K packets and produce a set of L redundant packets for a total of $N = CK = K + L$ encoding packets all of a fixed length P. C refers to the stretch factor of the erasure code ($C=N/K$).

Tornado codes produce an N packet encoding from a K packet source. To reconstruct the source data, it is necessary to recover ϵK of the N encoding packets, where $\epsilon > 1$. Encoding and decoding times in the idealistic setting is shown in the Table 1.

$$\text{Encoding time} = (K + L) \ln (1/ \epsilon) P \tag{7}$$

$$\text{Decoding time} = (K + L) \ln (1/ \epsilon) P \tag{8}$$

Table 1 Tornado encoding / decoding timings

Size	Encoding (sec.)	Decoding (sec.)
100KB	0.09	0.14
750KB	0.25	0.25
1.5MB	0.32	0.36
3MB	0.46	0.58
4.5MB	0.59	0.79
6MB	0.76	0.96
7.5MB	1.27	1.32
9MB	2.25	2.49
36 MB	4.32	4.36
54 MB	7.56	7.42
72 MB	12.83	11.83
128 MB	18.94	16.47

D. Analysis on the impact of Dominating Set based SON

By the implementation of dominating set for the clustering of nodes in the SON, the number of nodes for content replication is reduced to 60 percentages of the total number of nodes in the SON. Although the numbers of nodes are reduced, there will not be any change in the redundancy because of the proposed tornado codes used for encoding and decoding of the contents.

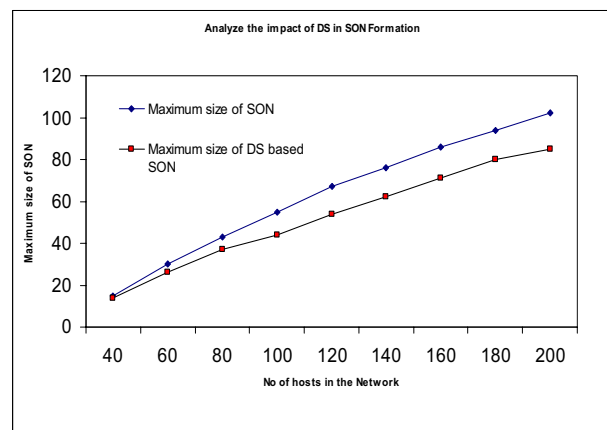


Figure 6. Analyze the impact of DS in SON Formation

Applying Dominating Set for SON formation yields reduction in content replication time, since number of replication is reduced to 60 percentage of total replication. This is depicted in Figure. 6. Even though there will be a reduction in replication time, the content collection time will slightly increase than the normal content collection time.

E. Performance analysis on congestion

Congestion on any of the n paths from origin node to recipient nodes impact both multiple unicast and sequential multicast. Fast replica uses any of these paths for transferring only 1/nth of the file.

F. Analyze the performance of various content distribution schemes during node failure:

The worst case delivery ratio of the different schemes like fast replica with Tornado, multiple unicast and sequential unicast as the number of simultaneous node failures on the overlay network has been analyzed and its performance is shown in “Figure. 7”. The data delivery ratio is defined as the ratio of the number of data packets successfully received by the node to the number of data packets sent by the source.

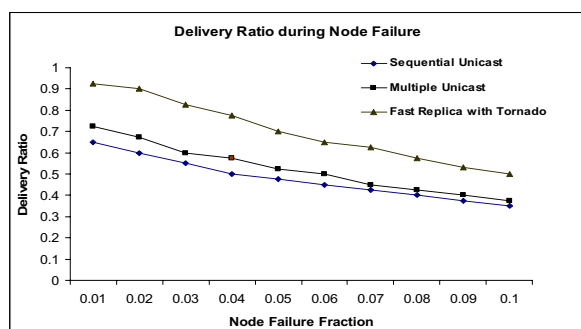


Figure 7. Analyze the Performance of CDN during node failures.

From the analysis it was observed that the delivery ratio of the fast replica with Tornado scheme degrades gracefully with increase in the number of simultaneous node failure while transmission of packets. The data delivery ratio for the best-effort protocol falls significantly with increase in the change rate of the overlay.

VI CONCLUSION

An algorithm called fast replica is applied for efficient and reliable replication of large files by creating Semantic Overlay Networks based on Dominating Set. Fast Replica partitions an original file into a set of sub files and uses diverse Internet paths among the receiving nodes to propagate the sub files within the replication set in order to speed up the overall download time for the original content. Scalability can be achieved using the algorithm by clustering the nodes in a set of replication groups, and by arranging efficient group communications among them, i.e. by building the semantic overlay network based on dominating set. An approach to parallel access to multiple sites using Tornado codes for reliable transfer demonstrates the trade off for the need of feedback, the bandwidth requirements, the download time and the memory requirements. We also analyze the performance of sequential unicast, multiple unicast with fast replica. We found that file replication time under Fast Replica across different file sizes in an 8 node replication set are

much more stable and predictable. The performance of sequential unicast, multiple unicast and fast replica with Tornado is analyzed in terms of content distribution time and delivery ratio. We also analyzed the impact of dominating set theory for the clustering of nodes in the SON.

REFERENCES

- [1] I. Clarke, O. Sandberg, B. Wiley, and T.W Hong, “A distributed anonymous information storage and retrieval system”, proceedings of the workshop on Design issues in Anonymity and Unobservability, Berkeley, 2000.
- [2] Akamai Technologies, Inc “http://www.akamai.com.
- [3] J.W. Byers, M. Luby and M. Mitzenmacher, “Accessing multiple mirror sites in parallel: using Tornado codes to speed up downloads”, proceedings IEEE INFOCOM, Mar. 1999.
- [4] Z. Fei, S. Bhattacharjee, E. W. Zegura and M. H. Ammar, “A novel server selection technique for improving the response time of a replicated service”, proceedings in IEEE INFOCOM, vol. 2. San Francisco, CA, March 1998.
- [5] M. Sayal, Y. Breitbart, P. Scheuermann, and R. Vingralek, “Selection algorithm for replicated web servers,” presented at the workshop Internet server performance, Madison, June 1998.
- [6] Arturo Crespo, Hector Garcia Moilna, “Semantic Overlay networks for P2P systems”, Google Technologies Inc., Stanford University, January 2003.
- [7] Thorsten Strufe, Dietrich Reschke, “Efficient Content Distribution in semi-decentralized peer-to-peer Networks”, proceedings of 8th Netties conference, Technische Universitat Ilmenau, Sep.-Oct. 2002.
- [8] L. Cherksova, J. Kee, “Fast Replica: Efficient Large file Distribution within Content Delivery Networks”, Proc. Of the 4th SENIX symposium on Internet Technologies, March 2002.
- [9] Pablo Rodriques and Emst W. Biersack, “Dynamic Parallel Access to Replicated Content in the Internet”, IEEE / ACM Transactions on Networking, vol.10 No.4, Aug. 2002.
- [10] J. Byers, M. Luby, M. Mitzenmacher, A. Rege., “A Digital Fountain approach to reliable distribution of bulk data”, Proc. of ACM SIGCOMM, 1998.
- [11] Mohammad Malli, Chadi Barakat, Walid Dabbous, “An Efficient Approach for Content Delivery in Overlay Networks”, Project Planet, INRIA-Sophia Antipolis, France, 2005.
- [12] Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic, “Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks”. IEEE transactions on Parallel and Distributed Systems, Vol. 13, Jan. 2002.
- [13] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris, “Resilient Overlay Networks”, Proc. of ACM Symp. on Operating Systems Principles, Oct. 2001.
- [14] Balachander Krishnamurthy, Craig Wills, and Yin Zhang, “On the Use and Performance of Content Distribution Networks”, ACM Signal Communication Internet Measurement Workshop 2001.
- [15] Shaikh. A, Tewari. R, and Agrawal. M, “On the effectiveness of DNS-based server selection”, Proc. of the IEEE Infocom 2001 conference, April 2001.
- [16] Laurent Massoulie, Anne-Marie Kermarrec, Ayalvadi J. Ganesh, “Network Awareness and Failure Resilience in Self -Organising Overlay Networks”, Proc. of the

- 22nd, International Symposium on Reliable Distributed Systems, 2003.
- [17] Vasileios Pappas, Beichuan Zhang, Andreas Terzis, Lixia Zhang, "Fault-Tolerant Data Delivery for Multicast Overlay Networks", Proc. of the 24th International Conference on Distributed Computing Systems, 2004.
- [18] Ying Zhu, Jiang Guo, Baochu Li, "oEvolve: Toward Evolutionary Overlay topologies for High-Bandwidth Data Dissemination", IEEE Journal on Selected Areas in Communications, vol 22, No. 7, Sep 2004.
- [19] Bo Han and Weijia Jia, "Design and Analysis of Connected Dominating Set Formation for Topology Control in Wireless Adhoc Networks", Proc. of 14th International Conference on Computer Communication and Networks, Oct 2005.
- [20] Meghanathan, N., "An Algorithm to Determine the Sequence of Stable Connected Dominating sets in Mobile Adhoc Networks", International Conference on Internet and Web Applications and Services, Feb, 2006.
- [21] Nguyen, T.N., Huynh, D.T., "Connected D-Hop Dominating Sets in Mobile Adhoc Networks", 4th International Symposium on Modeling and Optimization in mobile Adhoc and Wireless Networks, April 2006.
- [22] Lu, K., Bolla, J.A., Huynh, D.T., "Adapting connected d-hop Dominating Sets to Topology changes in Wireless Adhoc Networks", Proc. of 25th IEEE International Performance, Computing and Communication Conference, April 2006.
- [23] Jie Wu, Mihaela Cardei, Fei Dai, Shuhui Yang, "Extended Dominating Set and its Applications in Adhoc Networks using Cooperative communication", IEEE Transactions on Parallel and Distributed Systems, No. 8, Vol 17, Aug 2006.
- [24] Yamin Li, Shietung Peng, Warming Chu, "An Efficient Algorithm for Finding an Almost Connected Dominating Set of Small size on Wireless Adhoc Networks", IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Oct 2006.
- [25] Jie Wu "Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links", IEEE Transactions on parallel and distributed systems, vol. 13, no. 9, September 2002.
- [26] Tamaghna Acharya Rajarshi Roy "Distributed Algorithm for Power Aware Minimum Connected Dominating Set for Routing in Wireless Ad Hoc Network", Proceedings of the International Conference on Parallel Processing Workshops (IEEE - ICPPW'05), 2005
- [27] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks" in the Proc. of Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7-14, Aug. 1999.
- [28] L. Gewali, K. Mohammad, and M. Tin, "Interference Aware Dominating Set for Sensor Network", in the Proc. of the Third International Conference on Information Technology : New Generations (ITNG '06), 2006, IEEE – Computer Society.
- [29] Chi Ma, Y. Yang, and Z. Zhang, "Constructing Battery-Aware Virtual Backbone in Sensor Networks", in the Proc. of the International Conference on Parallel Processing (IEEE – ICPP '05), IEEE Computer Society.
- [30] Amutharaj, J., and Radhakrishnan, S., "Dominating Set based Semantic Overlay Networks for Efficient Content Distribution" proceedings in IEEE ICSCN 2007, vol. 1. Madras Institute of Technology, Anna University, Chennai, Feb 2007.

J. Amutharaj received his Bachelor of Engineering Degree from Manonmaniam Sundarnar University, Tirunelveli and Master of Engineering from Madurai Kamaraj University, Madurai. He is currently doing his doctoral program from Anna University, Chennai. He is a member of CSI, ISTE and Network Technology Group of TIFAC-CORE in Network Engineering. He has presented twelve research papers in National Conferences and four papers in International Conferences in Network Engineering.

S. Radhakrishnan received his Master of Technology and Doctorate in Philosophy from Banaras Hindu University, Banaras. He is the Director (Admissions and Students Affairs) and Head of Computer Science and Engineering Department in Arulmigu Kalasalingam College of Engineering, Srivilliputhur. He is a member of ISTE. He is a Principal Investigator of Research Promotion Scheme (RPS) project funded by Department of Science and Technology, Government of India. He is the Project Director of Network Technology Group of TIFAC-CORE in Network Engineering. This prestigious project is funded by TIFAC, Department of Science and Technology, Government of India and Cisco Systems. He is currently guiding twelve Research Scholars in the areas of Network Engineering, Sensor Networks, Optical Networks, Wireless Networks and Bio-medical Instrumentation.