ISSN:0975-9646

Shanmugavadivu et al. / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 1 (4) , 2010, 249-255

Caching Technique for Improving Data Retrieval Performance in Mobile Ad Hoc Networks

Mrs. K.Shanmugavadivu [1] and Dr M.Madheswaran[2]

[1] Professor / Department of MCA/ K.S.Rangasamy College of Technology / Tiruchengode -637215
mahesh_ksv@rediffmail.com

[2] Principal, Muthayammal Engineering College, Rasipuram.

Abstract

Mobile Ad Hoc Networks (MANETs) provide an attractive solution for networking in the situations where network infrastructure or service subscription is not available. Cooperative caching scheme can improve the accessibility of data objects. In this paper, we proposed a neighbor group data caching scheme called Neighbor Group Data Caching (NGDC) for improving data access efficiency in MANETs. The objective is to improve data availability and access efficiency by collaborating local resources of mobile nodes. cache resolution and cache management are the two problems of cooperative caching. To improve data availability and access efficiency, cooperative caching discovers data sources which induce less communication cost by utilizing neighbor group nodes. For cache management, cooperative caching increases the effective capacity of cooperative caches by minimizing caching duplications within the cooperation zone and accommodating more data varieties.We evaluate the performance of the Neighbor Group Data Caching by using NS2 and compare it with the existing schemes such as Neighbor caching and ZoneCooperative. The experimental results show that the cache hit ratio is increased by about 4%~40% and the average latency is reduced by about 5%~37% compared with other schemes.

Keywords:

Ad hoc, Cache placement, Cooperative caching, Cache Consistency, Cache Management

1. INTRODUCTION

A Mobile Ad hoc Network (MANET) is a collection of wireless mobile nodes forming a temporary network without the need for base stations or any other preexisting network infrastructure. In a peer-to-peer fashion, mobile nodes can communicate with each other by using wireless multihop communication. Due to its low cost, high flexibility, fast network establishment and self-reconfiguration, ad hoc networking has received much interest during the last ten years. However, without a fixed infrastructure, frequent path changes cause significant numbers of routing packets to discover new paths, leading to increased transmission latency over fixed networks. Each MANET node can serve as a router, and may move arbitrary and dynamically connected to form network depending on their positions and transmission range. The topology of the ad hoc network depends on the transmission power of the nodes and the location of the MNs, which may change with time. Caching techniques are an efficient solution for increasing the performance in message or data communication. The original idea of caching is that the data accessed by MHs has the properties of temporal and spatial locality. Higher temporal and spatial locality ensures that most accesses will go to the data that were accessed recently in the past and that reside in the cache. Therefore, caching frequently requested data can improve the performance of data communication. Data caching and prefetching techniques used in traditional wireless networks can be extended to be used in MANETs. In this paper , we investigate the use of caching and prefetching techniques for improving data accessibility and reducing latency in MANET environments. As mobile clients in ad hoc networks may have similar tasks and share common interest, cooperative caching, which allows the sharing and coordination of cached data among multiple clients, can be used to reduce the bandwidth and power consumption. Caching has been widely used in wired networks, such as the Internet, to increase the performance of web services [1-4] However, the existing cooperative caching schemes cannot be implemented directly in MANETs due to the host mobility and resource constraints that characterize these networks. As a result, new approaches have been proposed to tackle these challenges[5-10]. These approaches have been introduced to increase data accessibility and reduce query delay in MANETs. In [5], a cooperative cache-based data access scheme is proposed for ad hoc networks. The schemes presented in [9-10] are based on a specific routing protocol. The scheme in [9] used popularity, access cost, and coherency as criteria to replace cached data item when a mobile host's cache space is full. In [10], a transparent cache-based mechanism based on a new on-demand routing protocol called Dynamic Backup Routes Routing Protocol is proposed. The routing protocol and the cache mechanism allow the caching of data. This scheme allowed the cached data to be moved to a backup host in response to a link failure in order to guarantee data access. In [8], the implementation of an architecture similar to cooperative caching which defines two protocols to share and disseminate data among mobile hosts was proposed. However, the scheme focused on data dissemination in a single-hop rather than cooperative caching in a multi-hop environment. Another study utilized a novel architecture for database caching based on the separation of queries and responses [11]. The experimental results indicated that the scheme improved data accessibility by reducing response time in the presence of host mobility. Cooperative caching is an effective mechanism for increasing data accessibility in both wired and wireless networks. However, caching alone is not sufficient to guarantee high data accessibility and low communication latency in dynamic systems and with limited network resources. In this paper we propose an neighbor group data caching mechanism for MANETs. This paper provides the following contributions. First, we use a clustering architecture that allows localized and adaptive data caching and prefetching mechanism to increase data accessibility and reduce latency in the presence of host mobility. Second, we use a cache replacement policy to the cached data. Thus, eviction of data in the cache depends on a time to live parameter. Third, the proposed caching and prefetching architecture is flexible and does not rely on any specific routing protocol. The remainder of this article is organized as follows. Section 2 reviews the related works for cooperative caching schemes in mobile ad hoc networks. The proposed system architecture and the cooperative caching and prefetching strategies in presented in section 3. Section 4 presents the cache replacement policy and data consistency management. Section 5 presents the results of performance evaluation based on simulation experiments. Finally, Section 6 presents conclusions and future research work.

II. RELATED WORKS

A. CacheData and CachePath

In CacheData, the intermediate hosts, which are located along the path between the source host and the destination host, cache frequently accessed data items. In CacheData, the router node caches the data instead of the path when it finds that the data is frequently accessed. CacheData enforces another rule: A node does not cache the data if all requests for the data are from the same node. The CacheData approach needs extra space to save the data, it should be used prudently. In CachePath, the intermediate hosts record the routing path information of passing data. CachePath only records the data path when it is closer to the caching host than the data source. To handle cache consistency, CachePath and CacheData use a simple weak consistency model based on the time-to-live mechanism. In this model, a routing node considers a cached copy up-to-date if its TTL hasn't expired. If the TTL has expired, the node removes the map from its routing table (or removes the cached data). As a result, the routing node forwards future requests for this data to the data source. We optimize this model by allowing nodes to refresh a cached data item if a fresh copy of the same data passes by. If the fresh copy contains the same data but a newer TTL, the node updates only the cached data's TTL field. If the data center has updated the data item, the node replaces both the cached data item and its TTL with the fresh copy.

B. ZoneCooperative

The ZoneCooperative [11] scheme considers the progress of data discovery. In ZC, each client has a cache to store the frequently accessed data items. The data items in the cache satisfy not only the client's own requests but also the data requests passing through it from other clients. For a data miss in the local cache, the client first searches the data in its zone before forwarding the request to the next client that lies on a path towards server. Zone cooperative (ZC) caching scheme for data retrieval in mobile ad hoc networks. The ZC caching uses a simple weak consistency model based on the time-to-live (TTL), in which a client considers a cached copy up-to-date if its TTL has not expired. The client removes the cached data when the TTL expires. A client refreshes a cached data item and its TTL if a fresh copy of the same data passes by. However, the latency may become longer if the neighbors of intermediate nodes do not have a copy of the requested data object for the request.

C. GroupCaching

Group Caching (GC) allows each MH and its 1-hop neighbors form a group. The caching status is exchanged and maintained periodically in a group.In Group Caching, the caching space in MHs can be efficiently utilized and thus the redundancy of cached data is decreased and the average access latency is reduced. Although cooperative caching can provide the high accessability of data objects, the caching performance (cache hit ratio and average latency) can be reduced significantly due to the property of dynamic topology in MANETs. The group can store more data objects from the destinations than an MH because the group members are cooperative to cache the data objects.

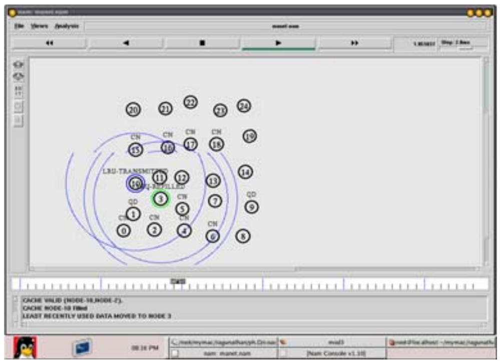

Fig. 1 Select a Node MH for Data Storage

Fig. 2 Movement of Data between neighbor node

III. PROPOSED NEIGHBOR GROUP DATA CACHING SCHEME

A. Network Model

The network consists of mobile hosts that form group. The network connectivity is maintained using a periodic Hello message that is exchanged among one-hop neighbors. other. The set of one hop neighbors of a client *MHi* is denoted by *MH*1 The combination of clients and transitive closure of their one hop neighbors forms a mobile ad hoc network. As clients can physically move, there is no guarantee that a neighbor at time t will remain in the cluster at later time $t + \tau$. The devices might be turned off or on at any time, so the set of live clients varies with time and has no fixed size.

B System Environment

The system environment is assumed to be an ad hoc network where MH access data items held as originals by other MHs. A MH that holds the original value of a data item is called data source/server/center. A data request initiated by a host is forwarded hop-by-hop along the routing path until it reaches the data source and then the data source sends back the requested data. Each MH maintains local cache in its hard disk. To reduce the bandwidth consumption and query latency, the number of hops between the data source/cache and the requester should be as small as possible. Most MHs, however, do not have sufficient cache storage and hence the caching strategy is to be devised efficiently. In this system environment, we also make the following assumptions:

– Unique identifier is assigned to each host in the system. The system has total of M hosts and MHi ($1 \leq i \leq M$) is a host identifier. Each host moves freely.
– The network is divided into several one hop non-overlapping groups where in each group
a node could be in one of two roles: MHi or ordinary node. MHi is a node that maintains information of different ordinary nodes in its group.

– We assign a unique data identifier to each data item located in the system. The set of all data items is denoted by $D = \{d1, d2, \ldots, dN\}$, where N is the total number of data items and dj $(1 \le j \le N)$ is a data identifier. Di denotes the actual data of the item with *id di* . Size of data item di is si (in bytes).
– Each MH has a cache space of C bytes.
– Each data item is periodically updated at data source. After a data item is updated, its cached copy (maintained on one or more hosts) may become invalid.

1) Caching Control Message :

In this caching scheme, design a caching control message to exchange the caching status in a group periodically. In experimental, every MHs exchange interval at every second. The caching control message contains the fields: { Cached data id, Timestamp, Remaining available cache space}. The caching control message is periodically sent by MHs. Each MH can maintain localized caching statues of one-hop neighbors for performing cache placement and replacement. Figure 1 illustrates the group in the view of MH D.

IV. CACHE REPLACEMENT POLICY AND DATA CONSISTENCY MANAGEMENT

In NGDC scheme, it proposes how and where to place the data object in a group member when an MH receives a data object from the destination. Based on the usage of caching control message, each MH knows the remaining available cache space of other MHs in a group and the IDs and timestamps of their cached data objects. First of all, when an MH receives a data object (called receiving MH), it caches the data object if the cache space is enough. Otherwise, the receiving MH checks the available cache spaces of its group members. If the available cache space of any group member is sufficient to store the data object, the receiving MH puts the data object to the group member randomly. Second, if the available cache space of every group member is not sufficient to cache the received object, the receiving MH lookups the group_table to see if there exists a group member that already caches the data object. If yes, the data object is not cached. If no, the receiving MH selects next neighbor MHs. The receiving MH checks the available cache spaces of next 1-hop neighbor members. If the available cache space of any next 1-hop neighbor member is sufficient to store the data object, the receiving MH puts the data object to the next 1-hop neighbor

TABLE I

THE SIMULATED PAREMETERS

Simulator	Network Simulator(NS2)[9]
Simulation time	6000 Seconds
Network size	1500m x 500m
Transmission range of MH	100m
Mobility model	Random waypoint
Speed of mobile host	1~10m/s randomly
Total of data item set	1000 data item
Average Query Rate	0.2 /Second
Hot data	20% of total data item set
Probability of query in hot data	80%
DataSize	10kBytes
Cache size	200kBytes, 400kBytes, 600kBytes,800kBytes, 1000kBytes,1200kBytes, 1400kBytes,
Compared Schemes	Cache Data[1], Zone Cooperative[2], Proposed Group Caching
Replacement policy	LRU

The process of data discovery performs the searches in the caching nodes for the requested object. In Neighbor Group Caching, when a requester (source) wants to retrieve a data object from the data source, it first checks its MHs to see if the data object exists locally. If yes, it returns the data object (cache hit) to the application. If no, it lookups its group_table for the data object, if yes, the requester redirects the data request to group member, and waits the replied data object (remote cache hit). If the requester can not find any cached record for the desired data object in the MHs and its one hop neighbor, it starts to execute the data discovery process in the next neighbor group member. Again if the requester can not find any cached record for the desired data object in the MHs and its one hop neighbor and its one hop neighbor, it starts to execute the data discovery process. Initially, the requester constructs a routing path to the destination and sends the data request to the next neighbor MH in order to reach the data source (destination). When the intermediate nodes receive a data request in the routing path, they lookup their self_table and group_table and its one hop neighbor of group member for the data request. The process of lookup first searches self_table and then searches the group_table and its neighbor group member. If the receiving MH can not find the record the request in its selft_table and group_table and its one neighbor group member, it forwards the request to the next MH on the routing path. If the destination (data source) receives the data request, it replies the data object via the routing path. When the intermediate node receives the pass-by data object, it performs the cache placement and replacement according to their self_table and group table. There are two schemes that can deal with the cache consistency problem: weak consistency and strong consistency. Under the weak consistency, a cached data object is associated with an attribute, TimeToLive (TTL). If the TTL time expires, the cached data object is removed. Under the strong consistency, if a cached data object is requested, the caching node first asks the data source to see if the cached data object is valid or not. Because of the energy concern and the constrain of wireless bandwidth, we prefer using the weak consistency in mobile ad hoc networks.

V. PERFORMANCE EVALUATION

The performance evaluation is shown in this section. The simulation model is given in Section 5.1. In SimpleCaching, only requester caches the replied data object for itself. All schemes use LRU as the cache replacement policy. The performance metrics is introduced in Section 4.3. Section 4.4 shows the results in performance evaluation.

A. The Simulation Model

The simulation is performed on NS2 with the CMU wireless extension. In our simulation, the AODV routing protocol [14] was tested as the underlying ad hoc routing algorithm. The simulation time is set 6000 seconds. The number of mobile hosts is set to 100 in a fixed area. We assume that the wireless bandwidth is 2MB/s and the radio range is 100m. There are totally 1000 data items distributed uniformly among all MHs. The number of hot data objects is set to 200 and all hot data objects are distributed uniformly among all MHs. The probability of queries for the hot data is set to 80%. The query rate of MHs is set to 0.2/second. In order to simulate the node join and leave operations, we set a join/leave rate. If the value of join/leave rate is 20, there will be ten MHs randomly joining and leaving the network every 20 seconds. If an MH joins or leaves the network, its content of cache will be cleared.We model the movement of nodes in a 1500m x 500m rectangle area. The moving pattern follows the random way point mobility model [15]. Initially, nodes are placed randomly in the area. Each node selects a random destination and moves toward the destination. After the node reaches its destination, it pauses for a random period of time and repeats this movement pattern. The detail of other simulation parameters is shown in **Table 1.**

B Performance Metrics

The Performance metrics are average hop count, cache hit ratio(include remote cache hit ratio in remote caching node), and average latency of data objects.

1) Average hop count :

The number of hop counts between the source and the destination or caching nodes.

2) Cache hit ratio:

T he combined cache hit ratio in the requester and its groupthe group members

3) Average latency:

The time interval between the time of generating a query in the requester and the time of receiving requested data object from the data source.

C. Simulation Results

1) Average hop count : We first measure the hop counts in all schemes. Table 2 shows the average hop count between the source and the destination when a requester wants to retrieve a data object. The destination can be the data source or intermediate caching nodes The simulation is run under different cache sizes and different join/leave rates. In all schemes when the cache size is large, the average hop count is reduced. In Neighbor Group Data Caching, the average hop count is the lowest because the Neighbor Group Data Caching improves the cache hit ratio and then reduce the average hop count.

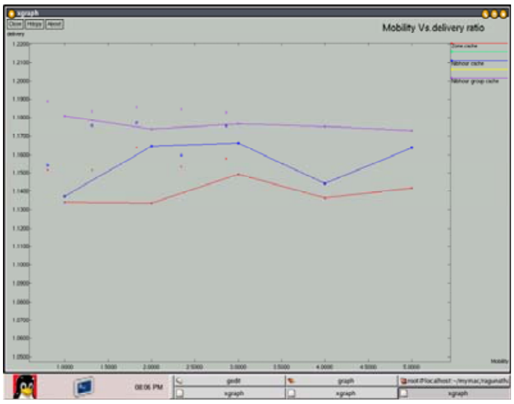

Fig. 3 Effect of delivery ratio on different node leave/Join rate

2) Cache Hit Ratio :

Figure 4 and figure 5 shows the cache hit ratios of MHs under different cache sizes and join/leave rates. The

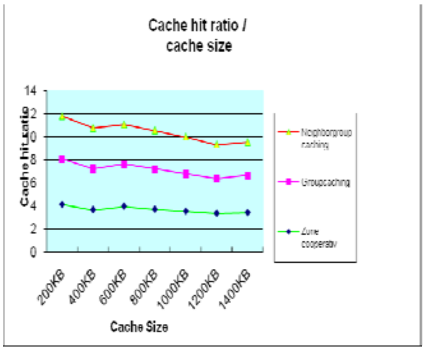

Fig. 4 Cache hit ratio under different node cache size

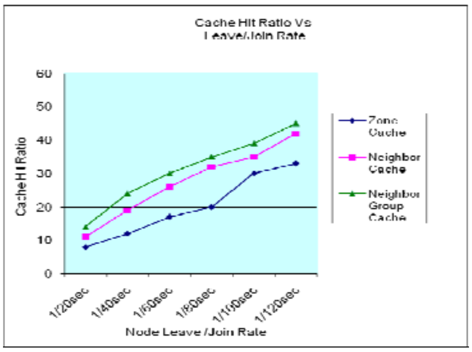

Fig. 5 Cache hit ratio under different node leave/join rate

measured cache hit ratio includes the cache hit (local cache hit) in the requester and cache hit in the other MHs (remote cache hit) except the data source. The cache size is set to 200KB, 400KB, 600KB, 800KB, 1000KB, 1200KB and 1400KB. The size of a data item is set to 10KB. The pair of source and destination nodes is randomly selected in the simulation. In general, the cache hit ratio increases while the cache size increases. Figure 2 shows the In Neighbor Group Data Caching has a higher cache hit ratio than others because both the MH and its group members can store data objects. These cached data objects Improve the cache hit ratio. Figure 3 shows the experimental results under the dynamic topology. In every 20, 40, 60, 80, 100, and 120 seconds, ten MHs are selected randomly for joining and leaving the network. When an MH leaves the network, it removes all cached data objects. When the MH joins the network, the content of its cache is set to empty. In Neighbor Group Data Caching shows the highest cache hit ratio because it utilizes all the available cache space of neighbors (group members). When an MH joins the network, its available cache space can be utilized by other MHs. Therefore, In Neighbor Group Data Caching, the cache hit ratio is higher than other schemes. In ZoneCooperative and Neighbor Cache schemes, there is no cooperative caching protocol among MHs. So the MH can not efficiently integrate their neighbor's cache space.

3) Average Latency :

Figure 6 and figure 7 shows the average latency under different cache sizes and different join/leave rates. We

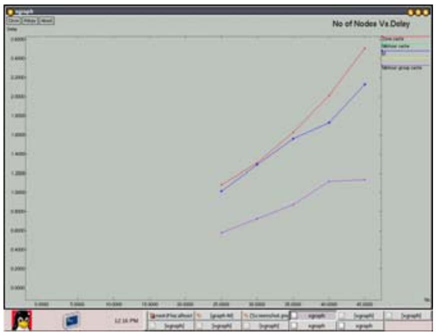

Fig. 6 Effect of average query latency on number of nodes

know that ZoneCooperative scheme has no cooperative protocol among the MHs. Therefore, when an MH receives a data request, it needs to send a request to its zone and waits for the response. As a result, it leads to the long latency if there is no cache record in a zone along the routing path. In GroupCaching, the MH and its one-hop neighbors form a group. If a data request is received, the MH can check its self_table and group_table immediately. No communication with its neighbors is needed to know the caching status in other group members.

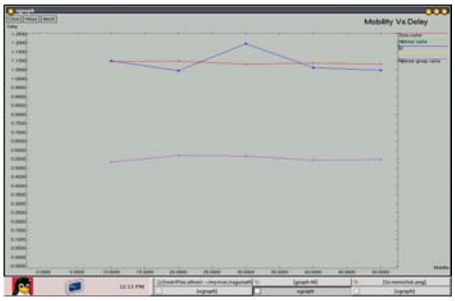

Fig. 7 Effect of average query latency on different node leave/Join rate

As a result, the average latency is reduced. Also, due to the placement and replacement algorithms executed in a group, all the MHs in a group member can cache more data objects and then reduce the redundancy of the cached data. Therefore, the average latency is reduced compared with other schemes.

VI. CONCLUSION

In this paper, we propose a Neighbor Group caching scheme (NGDC) for mobile ad hoc networks. MHs maintain the localized caching status among the group members. Therefore, the MHs can cooperative to store different data objects. Furthermore, if an MH has available cache space, it can be utilized by its neighbors as soon as it joins a group. It improves the cache hit ratio and reduces the average latency compared with existing schemes. In the future work, we will investigate the integration of broadcasting and cooperative caching.

REFERENCES

[1] V. Alex, “A good file system,” in *Proc of USENIX File System Workshop*, pp.1—12, May, 1992.
[2] A. Rousskov and D. Wessels, “Cache digests,” *Computer Networks and ISDN Systems*, vol.30, 1999, pp.2155-2168.
[3] J. Wang, “A survey of web caching schemes for the Internet,” *ACM SIGCOMM, Computer Communication Review*, pp.36—46.
[4] D. Wessels and K. Claffy, “ICP and the squid web cache,” *IEEE*, vol. 16, Jan. 1998, pp.345-357.
[5] G. Cao, L. Yin and C.R. Das, “Cooperative cache-based data access in ad hoc networks,” *IEEE Computer Society*, vol. 37, 2004, pp.32-39.
[6] T. Hara, “Replica allocation in ad hoc networks with periodic data update,” in *Proc. Int. Conf. on Mobile Data Management (MDM’02)*, Jan., 2002, pp.79-86.
[7] S. Lim, “A novel caching scheme for internet based mobile ad hoc networks,” *IEEE*, Oct. 2003, pp.38-43.
[8] M. Papadopoui and H. Schulzrinne, “Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices,” in *Proc. MobiHoc, ACM Press*, pp.117—127, 2001.
[9] F. Sailhan and V. Issarny, “Cooperative caching in ad hoc networks,” *Proc. of the 4th Int. Conf. on Mobile Data Management*. 2003, pp.13-28.
[10] Y-H. Wang, J. Chen, C-F. Chao and C-M. Lee “A transparent cachebased mechanism for mobile ad hoc networks,” *Proc. Of the Third Int. Conf. on Inform Tech and Applications (ICITA’05)*, vol.2, Jan. 2005, pp.305-310.
[11] H. Artail, H. Safa and S. Pierre, “Database Caching in MANETs Based on Separation of Queries and Responses,” in Proc. IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob 2005), Montréal, Canada.
[12] L. Yin and G. Cao, “Supporting Cooperative Caching in Ad Hoc Networks,” IEEE INFOCOM, pp. 2537-2547, March 2004.
[13] G. Cao, L. Yin and C. Das, “Cooperative Cache Based Data Access Framework for Ad Hoc Networks,” IEEE Computer, pp. 32-39, February 2004.
[14] Takahiro Hara, “Replica Allocation Methods in Ad Hoc Networks with Data Update,” Kluwer Journal of Mobile Networks and Applications,8(4), pp. 343-354, 2003.
[15] Chand, N. Joshi, R. C., and Misra, M., "Efficient Cooperative Caching in Ad Hoc Networks Communication System Software and Middleware," 2006. Comsware 2006. First International Conference on 08-12 Jan. 2006 Page(s):1 - 8

Authors

Shanmugavadivu. K received the B.ScDegree from Bharathiar university and MCA Degrees from Bharathidasan University and M.Phil (C.S) , M.Phil (C.S), Manonmaniam Sundaranar University. She is working as a Professor in Department of MCA at K.S.R College of Technology, Tiruchengode. Her area of interest Computer Networks, Network Protocols, Mobile Ad hoc Networks, Software Engineering. She is a member of ISTE.

M.Madheswaran received the BE Degree from Madurai Kamaraj University in 1990, ME Degree from Birla Institute of Technology, Mesra, Ranchi, India in 1992, both in Electronics and Communication Engineering. He obtained his PhD degreee in Electronics Engineering from the Institute of Technology, Banaras Hindu University, Varanasi, India, in 1999. At present he is a Principal of Muthayammal Engineering College, Rasipuram, India. He has authored over forty five research publications in international and national journals and conferences. His areas of interest are theoretical modeling and simulation of high-speed semiconductor devices for integrated optoelectronics application, Bio-optics and Bio-signal Processing. He was awarded the Young Scientist Fellowship (YSF) by the State Council for Science and Technology, TamilNadu, in 1994 and Senior Research Fellowship (SRF) by the Council of Scientific and Industrial Research (CSIR), Government of India in 1996. Also he has received YSF from SERC, Department of Science and Technology, Govt. of India. He is named in Marquis Who’s Who in Science and engineering in the year 2006. He is a life member of IETE, ISTE and IE (India) and also a senior member of IEEE.