CHAPTER 2

CACHING STRATEGIES FOR MOBILE AD HOC NETWORK

2.1 INTRODUCTION

Now-a-days, wireless ad hoc networks have obtained enormous growth due to the potential applications on the battlefield, in disaster recovery, and outdoor assemblies. The ad hoc networks have many challenges like packet loss, low throughput and high delay, because they do not have any fixed infrastructure. The node mobility and resource constraints are major blocking factors in MANETs. Hence, the techniques designed for wired networks are not applicable to the above mentioned applications.

Data access and dissemination in a mobile ad hoc network (MANET) is a challenging issue due to frequent disconnections. In spite of many research contributions available for data caching in wireless ad hoc networks, there is a need to provide efficient data dissemination and information access among the highly mobile nodes. This research aims in the reduction of the query delay, making the number of hops between the data centre node and the requester node as few as possible, for providing effective data dissemination among the group members of the MANET, also improves the performance of the network and efficiency of data accessibility. Many challenges are encountered in the deployment of traditional caching techniques, namely cache data, cache path and hybrid cache, due to node mobility.
The enhanced caching techniques for a Comprehensive Cache Management scheme in mobile ad hoc networks introduce the Path Pre-Fetching Technique (PPF), which deals with efficient prefetched paths. The Integrated Query Caching Technique (IQCT) upgrades the caching efficiency in a mobile network environment. When a node wants to cache a data item, but the cache is full, and then the Hit count based cache Replacement Technique (HCRT) replaces the least frequently asked data item in the cache memory. However, Enhanced Hit count based Replacement Technique (EHCRT) effectively replaces the data in the cache memory by taking into account data item size, order and its hit count value. An Extended Adaptive Time-To-Live (Ex-ATTL) technique maintains the consistency between the cached data at the clients and the original data stored at the server. In this chapter an elaborate study on the various caching strategies have been done and based on the discussion a need for enhanced techniques has evolved which is dealt briefly and finally the chapter ends with the discussion on the caching schemes.

2.2 CACHING STRATEGIES

Data caching is a concept of storing the copy of frequently asked data in neighbor node or an intermediate node for rapid access. Once the data is stored in the local cache, the future requests fetch the data from the local cache than its original source. This will reduce the average access time of the data and effectively improves the system performance. Data caching methods are broadly classified into two types, which are data caching in wired networks and data caching in wireless networks.

The World Wide Web can be considered as a large distributed information system that provides access to shared data objects. The popularity of the Internet and World Wide Web continues to grow since the most popular applications running on the Internet and World Wide Web has an exponential
growth in size, which results in network congestion and server overloading. The rapid growth of the WWW could be attributed to the fact that at least till now, its usage has been quite inexpensive, and accessing information is faster than by using any other means. Also, the WWW has documents that appeal to a wide range of interests, such as business, entertainment, news, education, research, sports, stock market, travel, shopping, weather forecasting, maps, multimedia, etc. Although the Internet backbone capacity increases by 60 percent per year, the demand for bandwidth is likely to outstrip the supply in the foreseeable future, as more and more information services are moved onto the Web. If some kind of solution is not undertaken for the problems caused by its rapidly increasing growth, the WWW would become too congested, and its entire appeal would eventually be lost.

One way to meet this challenge is to scale the network and server bandwidth to keep up with the client demand, which is an expensive strategy. An alternative is caching, which reduces the network bandwidth and server load, by migrating the server files closer to those clients that use the files. Web caching has been recognized as one of the effective schemes, to alleviate the service bottleneck and reduce the network traffic, thereby minimizing the user access latency. Caching popular objects at locations close to the clients has been recognized as one of the effective solutions to alleviate Web service bottlenecks, and to reduce traffic over the internet and improve the scalability of the WWW system.

An extensive research had been done on WWW data caching, but a lot of research has to be carried out in caching in mobile ad hoc networks. In mobile ad hoc network, an effective cache management should consider the following issues:

- The technique to discover the data effectively and send the same to the requested node with the minimum response time is required.
The data to be cached in the system to improve the system performance is the next consideration, to develop the caching technique in MANET.

If the cache memory is not having the space for new data item, then there is a possibility of replacing the existing cache in the mobile node, which requires designing the effective replacement policy.

The effective cache consistency scheme is required to propagate the updations to all cached nodes, and ensuring that stale data items are not present in the network.

Most of the previous research in ad hoc networks focuses on the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. Although routing is an important issue in ad hoc networks, other issues such as information (data) access are also very important, since the ultimate goal of using ad hoc networks is to provide information access to mobile nodes. The limited communication resources and node mobility makes data dissemination and sharing a challenging task in MANET.

Some nodes in the ad hoc network such as the wireless LAN or cellular networks may have wireless interfaces without any wireless infrastructure. In ad hoc networks, a data request is forwarded hop by hop until it reaches the data center and then the data center sends the requested data back. Various routing algorithms have been designed to route messages in ad hoc networks. To reduce the bandwidth consumption and the query delay, the number of hops between the data center and the requester should be as small as possible. Although routing protocols can be used to achieve this goal, there is a limitation on how much they can achieve. The problem of the optimal placement of caches to reduce the overall cost of accessing data is motivated by the following two defining characteristics of ad hoc networks.
Firstly, ad hoc networks are multi-hop networks without a central base station. Thus, access to remote information typically occurs via multi-hop routing, which can greatly benefit from caching to reduce access latency. Secondly, the network is generally resource constrained, in terms of channel bandwidth or battery power in the nodes. Caching helps in reducing communication, which results in savings in bandwidth as well as battery energy.

The data accessibility in ad hoc networks is lower than that in the conventional fixed networks. One of the effective and popular methods to improve the performance of data access in MANET is to use caching, i.e., to cache frequently accessed data objects at the data source node and a group of caching nodes. Hence the other mobile users can access the cached data objects nearby with reduced traffic overhead and query latency. There are two types of caching approaches. The first technique employs a centralized server to deliver information to the network nodes as and when they require it. The second technique is placing a copy of the server content onto all network nodes. With caching, the data access latency is reduced, since some data access requests can be satisfied from the local cache, thereby obviating the need for data transmission over the scarce wireless links. When cache techniques are used, data consistency issues must be addressed to ensure that clients see only valid states of the data or at least do not unknowingly access the data that are stale, according to the rules of the consistency model. Classical cache invalidation strategies are not suitable for mobile environments, due to frequent disconnections and mobility of the clients. The cache consistency, cache management and invalidation are the attractive approaches for mobile environments. Caching frequently accessed data items in the local cache of the node is the efficient method to improve the performance in mobile computing systems. The caching method reduces the data access latency because some of the data access requests are satisfied from the local cache, which in turn avoids the need for the data transmission over
the scarce wireless links. The existing caching techniques are discussed below:

The Data accessibility depends on the TTL value. When the TTL is small, pre-fetching provides high data accessibility (Denko et al 2008). When the TTL is large, most of the cached items remain valid during the simulation, so some of the invalid items need to be prefetched, which reduces the effect of pre-fetching. Tian & Denko (2007) reveals that the integration of pre-fetching and caching improves the system performance when the TTL is small. There is another pre-fetching algorithm (Naveen & Awasthi 2012) where the pre-fetching rules are defined in such a way that when there occurs a cache miss, the prefetch set for the cache missed item are generated which reduces query latency.

The Data-path caching scheme is considered as an effective scheme to reduce data request delays without causing a large number of data clones. But still the data pre-fetching method has its limitations like memory space, query latency and link failure. To overcome this, the proposed path pre-fetching algorithm is working along with DSR protocol and the cached path is broadcasted to the one hop neighbor which enhances the efficiency of the system.

The traditional caching method works in a way that, when a data item is requested, the request searches for the response in the local cache first. If the requested data item is not present in the local cache, then the request is passed to data centre node. This simple cache scheme works well until the network connection through data centre node is reliable.

The working of cache data involves the verification of the data requests by intermediate nodes. When the intermediate node finds the data item that is frequently requested, then those data are cached by the
intermediate nodes depending on the conservative rule. The conservative rule says, when all request for data item come from the same node, the intermediate node can not cache the data item. When requested data item is available in intermediate node’s local cache, then the requested data item is fetched from the intermediate node’s local cache rather than fetching from faraway data centre node. The drawback of cache data algorithm is that, the data item cached in the local cache of the intermediate node will occupy large cache memory space (Cao et al 2004).

In the Cache Path algorithm, the nodes cache the path in nearest caching node instead of faraway data centre node and further request are redirected through the cached path (Cao et al 2004). When compared to cache data algorithm, cache path algorithm saves cache memory space. The drawback is that, the cached path may become stale because of link failures due to node mobility.

In the hybrid cache algorithm, the drawbacks of the above algorithms are resolved. The Cache data algorithm is used, when data item size and TTL value is less. Cache path algorithm is used when TTL value is high, ie, path is valid for long period of time and the distance between request node and destination node is high. The drawback of the hybrid cache is that, if the data which is cached by intermediate node is not on the forwarding path of the request to the data centre node, then the cached information can not be shared (Yin & Cao, 2006).

Hassan Artail et al, 2008 has created a Cooperative and Adaptive Caching System (COACS) for mobile ad hoc network. Even though this cooperative caching system minimizes delay and maximizes the cached data in the mobile ad hoc network, without inducing excessively large traffic at the nodes, for a mobile distributed system like COACS, nodes can play either as a query directory (QD) to cache queries of the requested nodes or as a cache
node (CN) to cache data (responses to queries), deciding which QD to send a request to and the next QD to forward it to (if the first one does not return a hit), is a crucial issue. To reach one QD, a request is forwarded from one end of the network to another, and it is forwarded back across the network, to reach a second QD. This increased the search process time and the query delay and decreased the cache hit ratio. The drawback is that the queries are stored in QD and the data are cached in the CN separately. This limitation is overcome in the proposed system two components Integrated Query Caching Technique (IQCT), where Query Node (QN) has two components, Query Indexer (QI) and Local Cache (LC) where, the Query Indexer (QI) of the query node stores the query without any actual data. The Local Cache (LC) of the query node stores the frequently requested data.

In the group caching method (Shobha & Rajanikanth, 2009), each and every node along with its one hop neighbor form a group within a transmission range by sending “Hello” message. One of the nodes act as master in a group, the communication speed increases when the master node in every group communicates with its members through one hop routing. Alternatively, delay is also decreased. The drawback is that, there is a lot of energy consumption when all nodes in the group maintain cache status by sending “Hello” message.

The limitations of the above existing caching techniques give rise to the need for the introduction of an enhanced query caching technique to provide better cache efficiency and improvement in the data accessibility and data dissemination.

The next component of Comprehensive Cache Management scheme is cache replacement. The cache replacement process involves two steps: first, if some of the cached data item is no longer in use, then these items will be detached to make space for the newly arrived data item. The data items that
have expired their Time-To-Live (TTL) value can be removed as the data becomes stale and cannot be used. So, periodical checks can be done to delete the data items with an expired TTL. If there is still not enough cache space after all the obsolete items are removed, the cache replacement will go to the second step, which is that one or more cached data items will be taken out from the cache space.

The LRU (Least Recently Used) replacement policy is widely used for cache replacement in mobile environment. This algorithm keeps the recently used data items in its local cache, and removes the least recently used one (Cao, 2003). The data which are not used for a longer period of time will be evicted first from the local cache.

The SXO (Size x Order) primarily focuses on two parameters for cache replacement (Cao et al 2004). Data with large size are the better choice for replacement since they occupy more space on cache memory. Replacing the large data item makes more room for new data item. The second parameter is data item order. The SXO scheme focuses on these two parameters for cache replacement. The drawback is that even after considering the data item size and data item order there is less cache hit ratio.

The LUV cache replacement policy is based on the Least Utility Value (LUV). The LUV is calculated based on the access probability of the data item ($A_i$), data item size ($S_i$), coherency ($TTL_i$) and distance ($\delta_i$) between the request node and data provider node (Chand et al 2006). The Equation 2.1 is used for calculating the LUV for a data item ($d_i$).

$$LUV = A_i \cdot TTL_i \cdot \delta_i / S_i$$  \hspace{1cm} (2.1)

The Energy Efficient Cooperative Cache Replacement Problem (ECORP) is an energy efficient cache replacement policy used in ad hoc
networks (Li et al, 2007). Here, the replacement decision is done based on the energy utilization for each data access.

The LRU-Min is similar to the LRU in implementation, but this algorithm considers the data size during the replacement. LRU-Min tries to minimize the number of documents to be replaced (Denko & Tian 2008). In this technique, the data is arranged on the basis of access time. The replacement happens only when the data is greater than the incoming data with lowest access time. Consider the data item with the size X which needs to be cached, and this algorithm searches the least recently used data item with size greater than the X. This reduces the number of replacements in the network.

The limitations of above existing replacement techniques has given the need for the development of Hit count based Cache Replacement Technique (HCRT) and Enhanced Hit count based Cache Replacement Technique (EHCRT) to provide better data replacement and to improve data accessibility.

The Comprehensive Cache Management scheme includes the cache invalidation scheme as the next component. The (Time –To-Live) TTL is the heart of the temporal dependent invalidation scheme. Cache Path and Cache Data handle cache consistency, using a simple weak consistency model, based on the TTL mechanism (Acharya et al 1995). In this model, a routing node considers a cached copy up-to-date, if its TTL has not expired. If the TTL expires, 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.

In the Adaptive TTL based cache invalidation scheme (Tang et al, 2007) for Hybrid Cache and Cooperative Cache schemes, if the TTL of a
particular data item expires, some cached data can be invalidated. Usually, the node removes such invalid data from the cache. To save space, when a cached data item expires, it is removed from the cache, while its id is kept in the “invalid” state as an indication of the node’s interest. While performing data forwarding, if a mobile node finds an invalid copy of the data in its cache, it deletes the old copy, and stores the new copy for future use. If an expired path or data item has not been refreshed for the duration of its original TTL time, it is removed from the cache. The drawback is that, the Adaptive TTL based cache invalidation scheme increases the computing overhead of the cache node when multiple query arises. The limitations of above existing cache invalidation scheme has given the need for the development of Extended Adaptive Time-To-Live (Ex-ATTL) scheme to avoid stale data and to reduce computing overhead of the cache node.

Though there are various caching schemes available for providing data accessibility and data dissemination in mobile ad hoc network, there is a need to provide Comprehensive Cache Management scheme with enhanced caching techniques for efficient data accessibility and data dissemination in mobile ad hoc networks. With this as an objective, a new Path Pre-fetching (PPF) technique is implemented for improving the data accessibility in MANET. Secondly, an Integrated Query Caching Technique (IQCT) for efficient cache memory utilization is introduced. As the third step, a new Hit count based cache replacement technique (HCRT) and Enhanced hit count based cache replacement technique (EHCRT) is deployed for effective replacement of data and for providing end-to-end reliable communication. Finally, in order to maintain cache consistency, Extended Adaptive Time-To-Live (Ex-ATTL) scheme is developed as an effective invalidation technique for avoiding the access of stale data in the mobile ad hoc network.
2.3 ENHANCED CACHING TECHNIQUES PROPOSED FOR DATA DISSEMINATION AND ACCESSIBILITY IN MOBILE AD HOC NETWORKS

The Path Pre-Fetching (PPF) technique uses a pre-fetch buffer for storing the path which is pre-fetched from the requested node to perform path caching. Since the entries are timed out after TTL value, the shortest path is broadcasted to the one-hop neighbor for pre-fetching only when the number of hops is large. The PPF is integrated with Dynamic Source Routing (DSR) protocol in such a way that it allows the caching of multiple routes for a single destination with nodes adding complete source-destination paths and hence minimizes the cache memory utilization.

An Integrated Query Caching Technique (IQCT) is developed with the concept of selecting the query node in a mobile environment, and storing the queries in the query indexer without the data. Information about the available Query nodes is then broadcasted across the mobile network to inform all the requesting mobile nodes about the number of QNs and where to send their queries to get the data from the provider nodes (PN) where queries are cached. Forwarding pointers are used to link QNs and Request Nodes (RN). If the requested query is not available in the QI, it stores the query and passes its request to another QI, which is in the nearest distance. Here the hit count value is checked and if it is found to be greater than the predefined threshold value then a copy of the requested data item is cached in the local cache memory for future request and the requested data is retrieved from the local cache of the query node itself instead of fetching the data from faraway provider node. When the hit count value is less than the threshold value the data is fetched from the provider node. This technique reduces the query delay because data is fetched from the nearest query node and hence cache memory is utilized in a better manner.
A new Hit count based Cache Replacement Technique (HCRT) suggests a suitable subset of data item for expulsion from the cache, when a node wants a data to be cached, but the cache is found full. The hit count based cache replacement technique checks for the free space in the cache memory. If the free space is enough to hold the data it accepts, otherwise it removes the least hit count valued data item from the cache memory to save the newly arrived data item. The above replacement technique is improved by Enhanced hit count based replacement technique (EHCRT), introducing additional two parameters, the size and order of the data item along with the hit count value for better cache replacement. The robustness of the proposed hit count based cache replacement schemes are investigated under various cache sizes. For varying cache sizes the proposed techniques produces higher cache hit ratio which gives better assurance for end to end reliable communication and improved network performance.

An Extended Adaptive Time-To-Live technique (Ex-ATTL) is deployed, in which each and every node maintains the cache invalidation hash table. This hash contains data item as its key and TTL of the data item as its value. It stores the caching time $T_c$, when successful copy of the data item is created in a mobile node. First the verification of the cache invalidation hash table for the current request is carried out where the technique checks for the requested data item in node’s local cache. If local cache contains the requested data item, then the request node simply uses its local cache. Otherwise it recalculates the new TTL using Equation 2.2 and repeats the first step and retrieves the data item from nearest cache node or from the data source. The NewTTL value is calculated subtracting $T_c$ from request generation time $T_{req}$ as shown in Equation 2.2.

$$\text{NewTTL} = T_{req} - T_c \quad (2.2)$$
The new TTL with the corresponding data item is appended and sent to the client node as response and also updated in its table only for the particular data item. After successful copy of the data item is created in localized cache or in any intermediate cache node for future requests, $T_c$ is cached from the cache node’s timer and updates this $T_c$ in its cache invalidation hash table. Finally, the cache node broadcasts the new cached data items id, TTL and its $T_c$ only to one hop neighbors. This technique solves the critical issue of data becoming stale and supports valid data accessing using clear TTL calculation mechanism.

2.4 CONCLUSION

An extensive study on the caching strategies for mobile ad hoc network has been carried out and identified the need for a Comprehensive Cache Management (CCM) scheme for providing path pre-fetching, achieving cache efficiency, cache replacement and cache consistency maintenance. This has lead to the introduction of new enhanced techniques for cache management. With this motivation, the path pre-fetching technique for efficient data accessibility is introduced in the next chapter.