# CHAPTER 4

# SOURCE INITIATED ENERGY EFFICIENT SCHEME FOR MOBILE AD HOC NETWORKS

Mobile Ad hoc Network (MANET) is an infrastructure-less multi-hop network where each node communicates with other nodes directly or indirectly through intermediate nodes. Thus, all nodes in a MANET basically function as mobile routers participating in some routing protocol required for deciding and maintaining the routes. Since MANETs are infrastructure-less, self-organizing, rapidly deployable wireless networks, they are highly suitable for applications involving special outdoor events, communications in regions with no wireless infrastructure, emergencies and natural disasters, and military operations Perkins (2001).

In MANETs, mobile nodes are organized in a random manner. These sensor nodes are mainly deployed in physical environment to gather data and deliver the data to the destination wherever requires. Mobile nodes collect the route information through overhearing and store this information in route caches through Dynamic Source Routing (DSR) protocol. When the route cache freshness is absent, it leads to the stale route information resulting in pollution caches. If the node overhears the packet to another node, node's energy consumption occurs unnecessarily. The main goal of this research work is to reduce the effect of overhearing and to avoid the stale route problems while improving the energy efficiency using the Source Initiated Energy Efficient (SIEE) algorithm. Due to the lack of route cache update, the stale route entry and overhearing is originated among the network. In order to improve route cache performance five mechanisms are developed in DSR by using the algorithm SIEE. Simulation results shows that the proposed algorithm achieves better performance than the existing methods.

## 4.1 DYNAMIC SOURCE ROUTING (DSR) PROTOCOL AND THE EFFECTS OF OVERHEARING

Dynamic Source Routing protocol is a simple and efficient routing protocol designed especially for use in multi-hop, Wireless Ad hoc Networks of mobile nodes. DSR allows network to be completely self-organizing and self- configuring, without the need for any existing infrastructure or administration (Narayan et al., 2004). DSR gathers the route information through overhearing. Overhearing improves the routing efficiency in DSR by eavesdropping other communications to gather route information but it spends a significant amount of energy.

## 4.2 PROBLEM OF STALE ROUTE IN SOURCE ROUTING

According to Lim. S. et al. (2005), when the link errors or RERR are not propagated to the nodes, the route caches often contain stale route information for an extended period of time. In addition to this, the erased stale routes are possibly un-erased due to in-flight data packets carrying the stale routes. If a node has an invalid route in its route cache or receives a Route Reply that contains an invalid route, it would attempt to transmit a number of data packets without success while consuming energy.

The design choices for route cache in Dynamic Source Routing protocol concludes that there must be a mechanism, such as cache timeout, that efficiently expels stale route information. The main cause of the stale route problem is node mobility. It is unconditional overhearing that dramatically exaggerates the problem. This is because DSR generates more than one RREP packets for a route discovery to offer alternative routes in addition to the primary route to the source. While the primary route is checked for its validity during data communication between the source and the destination, alternative routes may remain in route cache unchecked even after they become stale. This is the case not only for the nodes along the alternative routes, but also for all their neighbours because of unconditional overhearing.

## 4.3 IMPLEMENTATION OF SOURCE INITIATED ENERGY EFFICIENT (SIEE) ALGORITHM

The mechanisms used to avoid stale route problems and to achieve minimum energy consumption are as follows:

Dynamic Source Routing aggressively uses route caching. Using source routing, it is possible to cache every overheard route without causing loops. Any forwarding node caches any source route in a packet; it forwards the packets for possible future use. The destination node replies to all requests. Thus the source node learns many exchange routes to the destination nodes that are cached. Swap routes are useful in case the primary route breaks. If any intermediate node on a route learns routes to the source and destination as well as other intermediate nodes on that route. A large amount of routing information is gathered and cached with just a single query reply cycle. So these cached routes may be used in replying to subsequent route queries. The reply from caches provides dual performance advantages. Further, it reduces route discovery latency and without replies from caches the route query flood will reach all nodes in the network. Cached replies satisfy the query flood early, thus saving on routing overheads.

When there is no way to remove stale cache entries then route replies may carry stale routes. Attempted data transmissions using stale routes incur overheads and generate additional error packets and can potentially pollute other caches when a packet with a stale route is forwarded or snooped on. In the following, there are three problems are identified with the DSR protocol that are the root cause of the stale cache problem.

**Case i**

The link break and route errors are not propagated to all caches that has an entry with the broken link then the RERR (Route error) is unicast only to the source whose data packet is accountable for identifying the link crack via a link layer feedback. Only limited number of caches are cleaned is considered. The failure information is propagated by piggybacking it onto the subsequent Route Requests from the source. If the Route Requests may not be propagated network-wide, many caches may remain unclean.

**Case ii**

Till now there is no mechanism proposed to expire stale routes. If not fresh, stale cache entries will stay forever in the cache.

**Case iii**

There is no way to determine the freshness of any route information For example, even after a stale cache entry is erased by a route error, a subsequent “in-flight” data packet carrying the same stale route can put that entry right back in. This problem is mixed up by liberal use of snooping. Stale routes are chosen up by any other node overhearing any transmission. Thus, cache “pollution” can propagate fairly quickly.

## 4.4 PROPOSED APPROACHES

### 4.4.1 Maximum Error Declaration

The proposed approach is based on the idea that bad news should be propagated “fast and wide”. In case if we want to increase the speed and we need to the extent of error propagation, so the route errors are now transmitted as broadcast packets at the MAC (Medium Access Control) layer. First, the node that determines the link breakage, broadcasts the route error packet containing the broken link information. Once receiving a route error, a node updates its route cache so that all source routes containing the broken link are shortened at the point of failure.

A node receiving a RERR (Route Error propagates), it further states that only there exists a cached route containing the broken link and that route was used before in the packets forwarded by the node. Note that using this scheme route errors reach all the sources in a tree fashion starting from the point of failure. In effect, route error information is efficiently disseminated to all the nodes that forwarded packets along the broken route and to the neighbours of such nodes that may have acquired the broken route through snooping.

### 4.4.2 Clock-based Expiration of Route

Recall that link breakage is detected only by a link layer feedback, when an attempted data transmission fails. Thus loss of a route will go undetected if there is no attempt to use this route. A more proactive clock-based approach will be able to fresh up such routes. A clock based approach is based on the hypothesis that routes are only valid for a specific amount of time T (Timeout period) from their last use. Each node in a cached route now has an associated timestamp of last use.  This timestamp is updated each time the cached route or part thereof is “seen” in a unicast packet being forwarded by the node.  The main portions of cached routes unused in the past interval are pruned.  The advantage of this approach depends critically on the proper selection of the timeout period a very small value for the timeout may cause many unnecessary route invalidations, while a very large value may defeat the purpose of this technique.  Although well-chosen static values can be obtained for a given network, a single timeout for all the nodes may not be appropriate in all scenarios and for all network sizes.  Therefore, a dynamic mechanism is desirable as it allows each node to choose timeout values independently, based on its observed route stability.

The proposed technique heuristic approach is used for adaptive selection of timeouts locally at each node based on the average route lifetime and the time between the links break seen by the node.  When a cached route breaks due to link breakage or upon receipt of a route error, the lifetime of the broken route is computed as the time elapsed since it was last entered in the cache.  Average route lifetime is obtained using the lifetimes of all broken routes in the past.  The time of the latest link breakage seen by a node is also maintained.

When route breaks occur uniformly in time, the average route lifetime itself provides a good estimate.  However, when many route breaks occur in short bursts with a large separation in time, the average route lifetime does not accurately predict during the periods of no route breaks.  The value of route life time _T is computed periodically and is used to terminate the stale entries from the cache.  In the experiments, every half a second the route cache is computed and then checked for stale entries.

### 4.4.3 Unconstructive Caches

In order to improve error handling in DSR, caching of negative information has already been recommended. In order to make use of this way, every node caches the broken links seen recently via the link layer feedback or route error packets. Within an interval of creating this entry, if a node is to forward a packet with a source route containing the broken link, (i) the packet is fallen and (ii) a route error packet is produced. In addition, the negative cache is always checked for broken links before adding a new entry in the route cache. Essentially, route cache and negative cache are mutually exclusive with respect to the links present in them. This prevents the cache pollution problem.

### 4.4.4 Various Steps of SIEE Algorithm

The concept of SIEE algorithm is explained below in the text box. If any link break (LB) occurs, the Route Error (RERR) message is transmitted and route cache will be updated between source and destination node. If the expected packet arrival time ($T_{out}$) reaches the limit, the routes are validated and determined. Otherwise the negative caches may occur and to determine cached pollutions as well as to initiate cache freshness. Conditional overhearing (CO) indicates that nodes are operating in promiscuous mode. In this mode, network connectivity is diagnosed. In randomized overhearing (RO), node scan only header of the packets. Maximum energy is determined during No overhearing (NO). The total energy consumption includes energy spent on conditional overhearing ($E_{CO}$), no overhearing ($E_{NO}$) and randomized overhearing ($E_{RO}$).

```
/* Source Initiated Energy Efficient Algorithm*/
Input: Conditional overhearing, no overhearing and randomized overhearing
of nodes.
Output: Calculating total energy consumption and updating route cache.
{
if (LB =1)  RERR is transmitted and Route cache is updated
}
if (Tout = 1 ) Valid routes are determined
else if ( Negative caches )
{
  Cache Pollutions are determined & cache freshness is initiated.

If( CO = 1 )
{
Nodes operating in promiscuous mode.
}
else ( RO = 1)
{
   Node scan only the header of the data
}
if( NO = 1) Maximum Energy Savings are determined.
}
{
Total Energy Consumption = $E_{CO}$+$E_{NO}$ + $E_{RO}$
else
route cache is updated }
```

## 4.5 PERFORMANCE ANALYSIS

NS2 tool is used to simulate the proposed algorithm. In the simulation, 50 mobile nodes move in a 1000 meter x 1000 meter square region for 50 seconds simulation time. All nodes have the same transmission range of 100 meters. The simulated traffic is Constant Bit Rate (CBR). The simulation settings and parameters are summarized in Table 4.1.

**Table 4.1 Simulation settings and parameters of SIEE**

| Channel Type | Channel/Wireless Channel |
|---|---|
| Radio-Propagation Model | Propagation / Two Ray Ground |
| Antenna Type | Antenna / Omni Directional Antenna |
| Interface Queue Type | Priqueue |
| Max Packet in Interface priority queue (IFq) | 250 |
| Network Interface Type | Phy/Wirelessphy |
| Mac | Mac/802.11 |
| Number Of Mobile Nodes | 50 |
| Routing Protocol | DSR |
| Simulation End Time | 50 (Sec) |
| Node Speed | 30(Meters/Sec) |

### 4.5.1 Performance Metrics

The performance was evaluated based on the following metrics.

**Control Overhead:** The control overhead is defined as the total number of routing control packets normalized by the total number of received data packets.

**End-to-end Delay:** The end-to-end-delay is averaged over all surviving data packets from the sources to the destinations.

**Packet Delivery Ratio:** It is the ratio of the number .of packets received successfully and the total number of packets transmitted. The percentage scale is used. The proposed algorithm is compared with DBEE-CLA and RANDOMCAST in presence of overhearing environment. The simulation results are presented below.

Figure 4.1 shows the results of average end-to-end delay for varying the nodes from 20 to 100. From the results, it shows that SIEE scheme has slightly lower delay than the RANDOMCAST and DBEE-CLA scheme because of authentication routes.

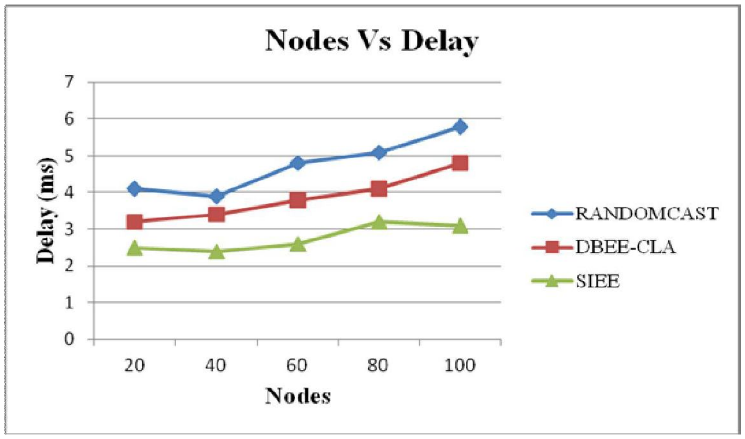

**Figure 4.1 Nodes vs. Delay**

Figure 4.2 presents the energy consumption. The comparison of energy consumption for SIEE, DBEE-CLA and RANDOMCAST is shown. It is clearly seen that energy consumed by SIEE is less compared to RANDOMCAST and DBEE-CLA.

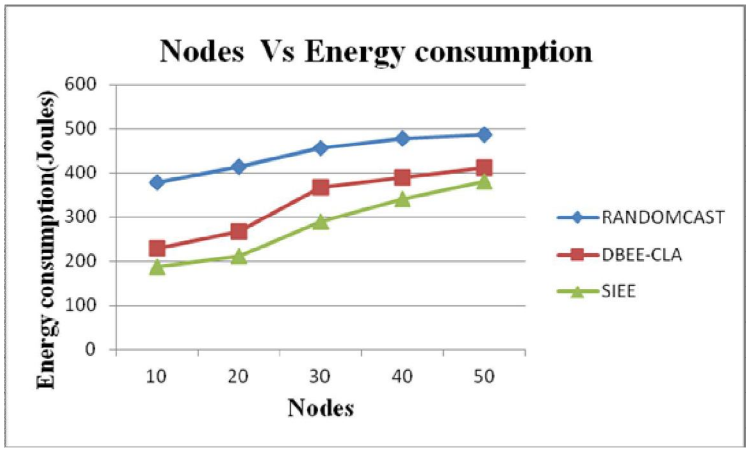

**Figure 4.2 Nodes vs. Total energy consumption**

Figure 4.3 presents the comparison of overhead. It is clearly shown that the overhead of SIEE has low overhead than the RANDOMCAST and DBEE-CLA.

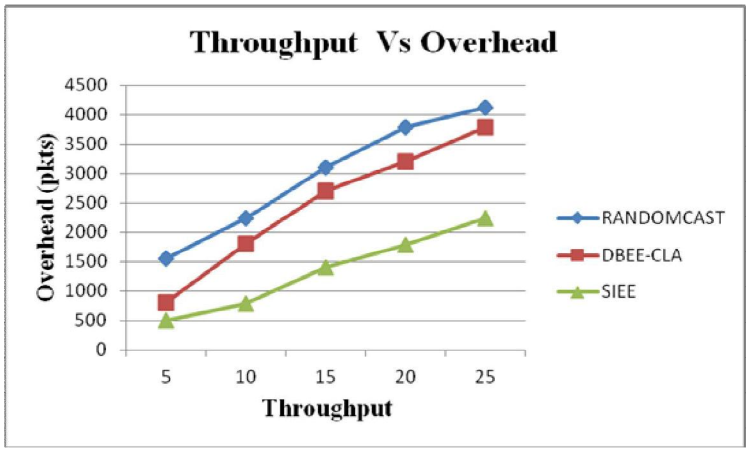

**Figure 4.3 Throughput vs. Overhead**

Figure 4.4 shows the results of Mobility vs. Delay. From the results, we can see that SIEE scheme has slightly lower delay than the RANDOMCAST and DBEE-CLA scheme because of authentication routes.

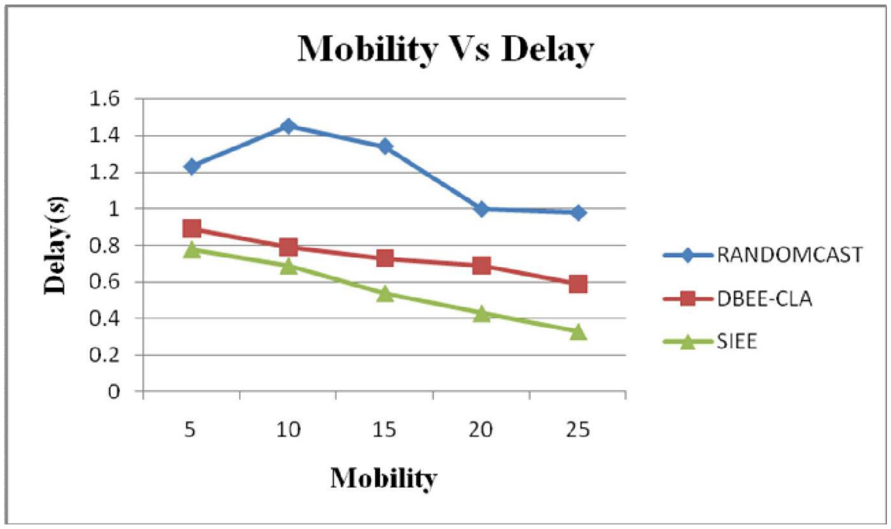

**Figure 4.4 Mobility vs. Delay**

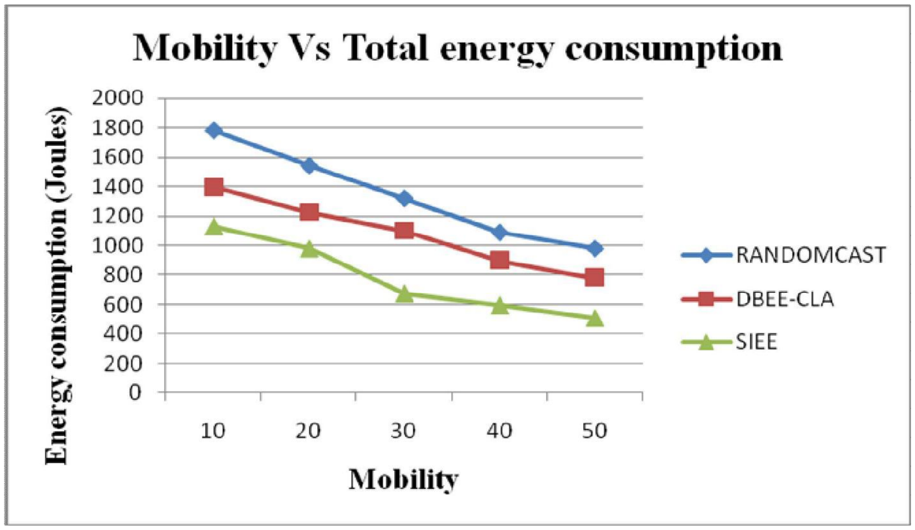

**Figure 4.5 Mobility vs. Total energy consumption**

Figure 4.5 presents the comparison of total energy consumption while varying the mobility from 10 to 50. It is clearly shown that the energy consumption of SIEE has low overhead than the RANDOMCAST and DBEE-CLA.

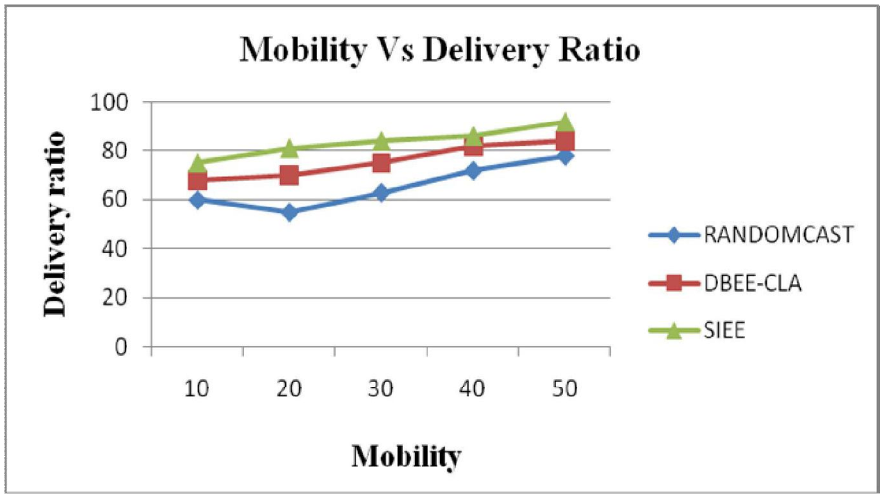

**Figure 4.6 Mobility vs. Delivery ratio**

Figure 4.6 shows the results of average packet delivery ratio for the mobility 10, 20…50 for the 100 nodes scenario. Clearly our SIEE scheme achieves more delivery ratio than the RANDOMCAST and DBEE-CLA scheme since it has both reliability and security features.

## 4.6 SUMMARY

The performance of SIEE is analyzed over DBEE-CLA, RANDOMCAST. SIEE is a three phase scheme consisting of maximum error detection, clock based expiration of routes and unconstructive caches. Using SIEE the routing and packet forwarding behaviors of the nodes is monitored in each hop. The proposed scheme totally avoids the link error, detects the expiration time of links and negative caches to improve the cache freshness. By simulation results, it is observed that the SIEE achieves more packet delivery ratio while attaining less delay and overhead, less energy consumption compared to DBEE-CLA and RANDOMCAST.