MANET: An Alternative Approach to Reduce Flooding by Propagating Neighborhood Information

Balakrishnan C[1], Sudarsun S[2], Bharathi Mani[3], Srinivasa Ragavan[4]
{radusdirect@yahoo.com}

ABSTRACT

Mobile Ad Hoc Networks (MANETs) exemplify a complex distributed network, which is characterized by the lack of any infrastructure. The lack of infrastructure though on one hand purports many significant advantages over the infrastructure-based networks, these networks have additional constraints that conventional networks do not have. For example, the connection establishment is costly in terms of time and resource where the network is mostly affected by connection request *flooding.* The proposed approach presents a way to reduce flooding in MANETs. Flooding is dictated by the propagation of connection-request packets from the source to its neighborhood nodes. The proposed architecture embarks on the concept of sharing neighborhood information. The proposed approach focuses on exposing its neighborhood *peer* to another node that is referred to as its *friend-node*, which had requested/forwarded connection request. If there is a high probability for the friend node to communicate through the exposed routes, this could improve the efficacy of bandwidth utilization by reducing flooding, as the routes have been acquired, without any broadcasts. *Friendship* between nodes is quantized based on empirical computations and heuristic algorithms. The nodes store the neighborhood information in their cache that is periodically verified for consistency. Inconsistent routes are erased rather than being updated after a *record-validity* period. The vicinity information is tracked based on a -- *I'm alive* signal to other nodes. These broadcasts are limited to a hop count of *one* and executed when the network activity is feeble.

Keywords: MANET, flooding, cache, connection-request, sharing, friend-node

1. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are formed dynamically by an autonomous system of nodes that are connected via wireless links without using the existing network infrastructure. The nodes in an ad hoc network can communicate with any other node that resides within its transmission range. For communicating beyond this transmission range, the nodes use intermediate routes to relay the message hop by hop. Hence, these networks are also referred to as "*Multi-Hop Wireless Ad Hoc Networks*" [1]. To cope with the self-organizing, continuously evolving dynamic nature, peer-to-peer communication environment of the MANET various parameters of the conventional routing protocols require a change. The two major classes of routing protocols that provide solutions to the routing problems in MANETs are *Proactive Routing* and *Reactive Routing Protocols* [1].

The principal objective of a routing protocol is efficient discovery and establishment of a route between the source and the destination so that there can be a timely and efficient delivery of information between them. A *Locating S*ervice is used to locate the receiver inside the network. It dynamically maps the logical address of the receiver to its current location in the network. Once the receiver is located, *routing and forwarding algorithms* are used to route the information through the MANET. The routing is done using one-hop transmission service provided by the enabling technologies to construct an end-to-end (reliable) delivery services, from sender to one or more receivers. A number of features are expected to be supported by the routing protocols which include parameters like minimal control & processing overhead, loop freedom & prevention, efficient dynamic topology establishment and maintenance, scalability, support for unidirectional links, security & reliability and support for Quality of Service.

2. BACKGROUND

The routing protocols used in MANETs fall into two major categories quoted below.

- Proactive Routing Protocols
- Reactive Routing Protocols

The proactive routing protocols are also referred to as *Table-Driven* protocols [11]. The proactive protocols possess routing tables for maintaining the routing information to the various nodes in the network. When required, a route is looked up from the repository of routes maintained, and the destination is contacted. Each node uses flooding approach to broadcast its location [5], which when received by the other nodes, is stored as the route to the broadcasting node. A node broadcast when the topology of the network changes. To reduce the overhead, the frequency at which a node posts its updates can be varied along with the area over which the messages are delivered. Since proactive routing protocols use routing tables for storing all the nodes' location information, this approach demands huge memory spaces. The updates also when frequently posted might instigate congestion at various links of the network. On the other hand, because routing information is constantly propagated and maintained in table-driven routing protocols, a route to every other node in the ad-hoc network is always available, regardless of whether or not it is needed. Therefore, there is practically zero delay in fetching a stored route. As a result, the proactive approach provides a better quality of service.

Reactive routing protocols perform route discovery on a lazy approach. They don't maintain a routing table as in the case of proactive routing protocols but they invoke a route discovery mechanism on demand. In other words, only when a node needs to send information to any other peer, is

the route found out by the protocol. They are hence called as *On-Demand* routing protocols [1].

In general, on-demand routing protocols are more efficient than proactive routing protocols. The on demand protocols don't have the overhead of maintaining the huge routing tables, but at the same time have to propagate broadcast messages for connection request. When a node using an on-demand protocol desires a route to a new destination, it will have to wait until such a route can be discovered. This latency has the tendency to make the protocol unsuitable for real-time communications. In comparison with the proactive routing protocols, the on-demand routing protocols can much easily adapt to the highly dynamic nature of the MANET.

In addition to the proactive and reactive classes of protocols, there are other protocols for routing and forwarding that are classified as *Hybrid* routing protocols which combine the quality and performance strengths of proactive and reactive protocols.

3. CACHING & TIMEOUTS

In order for the destination node to know the location of the destination or the receiver they have to acquire the route through the process of flooding. Flooding involves broadcasting of a packet to all the nodes of the network requesting the route of the destination node [4]. The nodes either respond with a reply back to the sender if in case, the current receiving node of the packet is the destination, or otherwise they forward the packet to other nodes. The destination node responds to the sender with the connection acknowledgement. The path traced by the acknowledge packet is remembered in all the forwarding hosts as the route from sender to destination. Once the path is set in routing tables of hops en route, further packets from sender to receiver is guided based on the forwarders' routing table.

On one hand though the process of flooding helps the sender to dynamically obtain the location of the destination and the route over which information could be transmitted, it unnecessarily augments the load on the network as all the nodes in the network participate in the process of flooding [2]. If flooding is carried out frequently then it may altogether lead to the instability of the network. The direct implication of this observation is that flooding should be kept as infrequent as possible.

One standard way to reduce the flooding mechanism is to provide the nodes with a small *cache* where the routes could be stored for future reference. The continuously changing characteristic of the ad hoc network environment poses further a problem, when routes are stored in a cache. There is always a possibility for the destination nodes to move from their place to another or even switch off. The cache reflects a static value and therefore in order to keep the data in the cache consistent they must be updated as frequently as possible. To account for these the routes in cache must be updated and validated periodically. The direct implication of this is that broadcast will be done frequently, which is to be avoided at all costs.

A new parameter referred to as the *timeout* period [4] is introduced to alleviate the problems arising. The timeout period is maintained for every route of the destination stored in the cache. This parameter reflects the lifetime of a route. The moment the timeout value expires the route in cache is deleted. Though still broadcasts are made, the frequency of broadcasts is reduced. The value of the timeout period reflects the frequency at which flooding occurs and if it is chosen to be a large number there is still a possibility of the route to have become invalid before the expiry of the cache. Therefore, the timeout value has to be prudently chosen. A tradeoff has therefore to be struck between the consistency of information and the frequency of broadcasts made.

4. NEIGHBOURHOOD

The suggested protocol falls under the category of reactive protocols. The nodes in this approach obtain the routes only when demand arises. The nodes use the common flooding approach to acquire the routes. Though flooding is used at the initial phases, it is decreased gradually. The most common method of using a cache with a fixed timeout period for each route is also used. The nodes are equipped with a small cache to save the routes and a timeout value is chosen. The selection of the timeout value is done as in case of the conventional networks, keeping in mind the other constraints of the network. However, the variation in the approach comes from the fact that the expiry of the timeout period does not trigger an update. The routes of the destination in the cache are rather erased after the timeout period. The nodes may then have to use flooding again to regain the routes, but in order to avoid that routes are shared between the nodes based on some criteria.

The primary focus of the protocol is on sharing information about the neighborhood of a peer with yet another node in the network. The neighborhood of a peer here reflects the contacts of the node in question with that of other nodes in the network. In other words, the neighborhood reflects the entries of routes in the cache. This information regarding the contacts a node has with other nodes in the vicinity is stored in tables or any other suitable data structure that is compatible with the protocol being adopted. The sharing of neighborhood information is not a mandatory task rather it is done at the discretion of the nodes concerned. The given approach intends to minimize the flooding requests that are needed to acquire the same information in the absence of the sharing mechanism.

The sharing of information occurs mainly between the nodes that are in direct contact with one another. Though the same can be carried out between nodes that are connected through a series

of a finite number of intermediate nodes, factors such as the power levels of the nodes could be questioned to decide whether such a sharing could improve the efficiency or not. Since the sharing of information is at the discretion of the nodes in contact, they can decide whether the process has to be carried out or not. The process affects only the nodes that are communicating and any series of finite set of intermediate nodes that connect the two nodes, if such a process is to be triggered between nodes connected by multi-hop links. The nodes can also take into account parameters such as the current load on the route connecting the nodes, the current load on the two nodes and the power levels required to carry out the operation before initiating the sharing of information.

4.1 Intimacy Factor

The process of sharing of neighborhood information occurs when the receiving node decides that the node that started the communication between the two is ready for accepting the information. This decision could be made based on a parameter called the *intimacy factor*. The intimacy factor reflects the level of trust between the two nodes that are communicating. A threshold level of intimacy factor could be defined called as IF_{THRES}, which could then be compared against the intimacy factor, calculated between two communicating nodes to determine, when exactly to commence the sharing of neighborhood information. If the intimacy factor calculated is greater than IF_{THRES} then the receiving node can make a request to the sender enquiring its acceptance to the information about the routes to nearby nodes. This request is optional and the receiver does so at its discretion

After the receiver ensures that the node that initiated the communication is ready to receive the neighborhood information, it posts a request to the sender. The sender can accept or reject the request. It can take into account the load on the link, the load on it, and its power level before posting to the receiver its consent. This can ensure that the sharing of routes doesn't exhaust the limited resources available. After the transmission of the sender's consent to the request, the sharing of the information or routes begins. The receiver shares a percentage of its cache entries with its friend node, depending upon the power levels and other such criteria. The sender then comes to know of the locations of various destinations close to the receiver. There may be a good possibility for the sender to send messages to these destinations, in which case the flooding process required for acquiring the same, have been eliminated.

4.2 Modeling

In a MANET, the presented approach could be modeled in the following way.

Total number of nodes in the network = $\boldsymbol{T_n}$
Total number of nodes in cache = $\boldsymbol{K_n}$
Unknown nodes = $\boldsymbol{U_n}$

The network is considered to have T_n number of nodes. The initiator of communication or the sender is assumed to have knowledge of routes of certain number of nodes in the network. The sender is unaware of the route of the other nodes, of which some may be near the receiver, with which the sender is currently communicating. The receiver is assumed to have a similar knowledge of routes of various nodes in the network. The set of routes in the receiver's cache need not be disjoint in comparison with the contents of the cache in the source's node; although the greater, the dissimilarity in the contents of the cache would imply a greater efficiency in the working of the protocol.

Route Gain Ratio (RGR) = (contents of sender's cache) ~ (contents of receiver's cache)

$$RGR \propto \eta$$

Where η, is the efficiency of the protocol

After the receiving the routes of the nodes in the neighborhood of the receiver, these are stored in the cache of the sender. The basic understanding is that, given that the sender has contacted the receiver, it has a good probability to communicate with the nodes nearby the receiver. Since the approach is reactive protocol oriented, new routes have to be acquired before the transmission of information to the other destinations. Calculating the probability that the sender communicates with any of the unknown nodes or nodes for which it does not have the location, a clear understanding of the working efficiency of the protocol can be obtained.

Number of nodes (given) : $\boldsymbol{T_n}$
Probability that an unknown node is contacted by the sender :$\boldsymbol{P_u}$

The approach will prove to be efficient only if the sender can utilize the information obtained from the receiver before it expires.

Time available for the sender for utilizing the routes : $\boldsymbol{T_{out}}$
Assuming the average time spent per node as,
Average time spent in communicating with a node : $\boldsymbol{T_{avg}}$
Total number of calls possible before routes expire : $\boldsymbol{T_{out} / T_{avg} = T_c}$
Total number of unknown nodes : $\boldsymbol{U_n}$ (nodes whose route are unknown to sender)
Probty. that an unknown node is contacted : $\boldsymbol{P_u}$

$$\boldsymbol{P_u = (U_n \, C \, T_{calls}) / (T_n \, C \, U_n)}$$

When T_n is large, P_n tends to be very small. The efficiency of the protocol increases only when the unknown nodes contacted for a subset of the nodes in the neighborhood of the receiver. In other words, the maximum efficiency is gained only when the unknown node contacted is one of those exposed by the receiver to the sender during the sharing of neighborhood information.

Let number of nodes exposed = E_n
Probty. that a node exposed is contacted : P_e

$$P_e = (E_n C T_c) / (U_n C E_n)$$

Probability that the node contacted forms a subset of the nodes exposed **: $P = P_c * P_e$**

If the probability that the node contacted is from the set of nodes whose routes have been exposed by the receiver, then the protocols succeeds in eliminating the flooding requests which otherwise would have been required to contact the unknown nodes. Considering the MANET environment to consist of a large number of nodes n and the probability P_u being small, Poisson distribution could used to model the situation as following.

Total number of nodes = $n = T_n$
Probability that an exposed node is communicated = P
Let x be the number of exposed nodes contacted by the sender.
Then, $nP = \lambda$

The set of routes that are exposed are only valid until the timeout period, after which they are deleted from the cache. The quantity of maximum concern here is the number of exposed nodes that are contacted.

Probty. that x nodes are contacted = $P(X = x)$

$$P(X = x) = (e^{-\lambda} \lambda^x) / x!$$

$$P(X = x) = (e^{-np} \lambda^x) / x!$$

$$P(X = x) = (e^{-n(Pc*Pe)} \lambda^x) / x!$$

Where $P_c = (U_c C T_c) / (T_c C U_n)$
$P_e = (E_n C T_c) / (U_n C E_n)$

Total exposed nodes contacted : $T_e = P*E_n$

The higher the value of T_e, the lesser the broadcasts required for getting the routes for the unknown nodes. The probability that no exposed node is contacted is given by $P(X = 0)$.

$$P(X = 0) = e^{-n(Pc*Pe)}$$

$$Pc * Pe > 0 \text{ and always a finite quantity}$$

$$P(X = 0) = e^{-n(Pc*Pe)} > 0$$

4.3 Increasing the Probability

The probability of contacting an exposed node is therefore never zero. To improve the probability and decrease further the flooding process that are carried out, the value of $P(X=x)$ must be closer to unity. To increase the number of exposed nodes contacted there exists two possible approaches, one by improving the value of E_n and the other wherein P is increased. Boosting the value of E_n is not under the control of the designer. E_n signifies the number of exposed nodes and is directly dependent on the neighborhood of the receiver that exposes the routes of the nodes to the sender. The value of E_n depends on the topology of the network, the density of the network and the mobility of the nodes in the network. Although E_n is strictly not under the control of the network designer, the value of E_n can be enhanced considerably by increasing the number of nodes exposed. In general, the receiver might then be expected to expose routes of the direct contacts it has, to the sender. In order to escalate further the probability of contacting an exposed node, it can augment the sample space of the nodes exposed. In other words, it can expose more nodes. This involves the receiver exposing nodes that are connected to it even through multi-hop links. The different nodes can be exposed one by one based on priorities assigned to them according to the distance of the exposed node from the receiver. The sender may wish to stop the transaction at any time in the middle by issuing an "*I'm satiated"* message. The receiver on receiving the message stops sending the routes.

The second method of increasing the probability P to improve the value of T_e proves to be more feasible. In order to amplify the value of P the number of nodes that can be contacted before the exposed routes become invalid, can be boosted. This implies that the timeout period should be increased. If timeout value is enhanced then it can have two impacts on the network. The first impact is one, which would lead to lesser number of flooding, due to less frequent updates and a higher value of probability of contacting an exposed node. The second would promote a chance for the data or the routes to be corrupted between the timeout periods. As a consequence of this, a tradeoff has to be struck between consistency of data and the reduction of flooding requests.

4.4 Cases

The total number of messages that are transferred between the sender and receiver depends on the amount of information shared between them. It also depends on the number of intermediate nodes that are present between the sender and the receiver. However, considering the fact that the sharing of information only affects the sender, receiver and the finite number of intermediate nodes, if any is present; it can be concluded that the number of messages processed and transferred would be less than in case of flooding. This relies on the fact that the latter process involves all the nodes of the MANET environment. Therefore, even if the sharing process is a slightly prolonged one the process does not have any impact on the other nodes of the network, which still remain free for communication.

Also in the suggested approach, the flooding requests are minimized to a hop count of one. The flooding is initially limited to the immediate circle of nodes remaining in the coverage region of the node that is broadcasting the flooding request. Of these nodes, if any has the route to the destination in

its cache it posts a message to the sender of the broadcast request, replying that the destination is its friend. At this point, the sender might establish the connection with the responding node, rather than the destination nodes itself. This intermediate node then routes the messages sent to it, to the intended destination.

In the worst case, the immediate circle of nodes in the coverage region of the source node might not have the route to the destination. Under such circumstances, the source node can rebroadcast the message with a hop count that can be found using an algorithm to increase the depth of flooding exponentially. When the flooding is done again with a greater hop count than used in he previous broadcast, the request is posted to more number of nodes. The process is repeated with a more number of nodes covered during each time.

The connection to the intended destination is then broken into different connections that involve nodes in direct contacts. Therefore, the source sends the message to the node with which it maintains the direct contact and which knows the route to the destination. This node then contacts with the destination or with the other node in the set of nodes connecting the source with the destination. The focus of the sender of the information is then on passing the information packet only to the node in direct contact. The task of routing the packet to the intended destination is then vested with the intermediate node that receives the packet. This responsibility then shifts from intermediate node to another one, if multiple nodes are present between the link connecting the sender and the receiver, as in case of any other multi-hop link. The last node meets with the responsibility, when it receives the packet and transmits it to the intended receiver through a direct link.

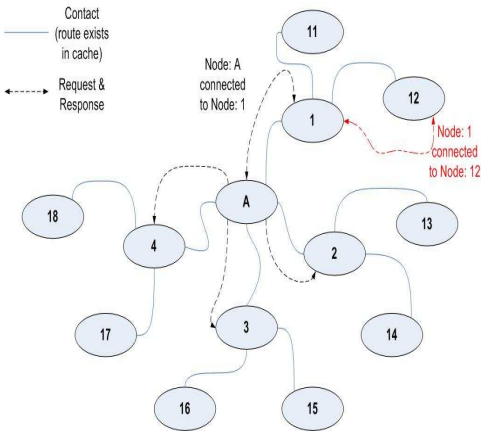

Figure 1

4.5 Security

The fact that the suggested protocol directs a message to the friends of the destination, and establishes the contacts with the same rather than the destination itself, poses a serious security problem [4]. The friends of the destination node that route the message could easily view the message and if is a malicious hacker could even tamper the message. This could be alleviated though with encryption of the data in the message packets. By encrypting the data part of the message packet and leaving the header of the packet encryption free, the intermediate nodes could always route the message, but the option of seeing or tampering stays invalid. An encryption scheme such as *Public Key Infrastructure (PKI)* could be employed to secure the packet transmitted. All the routing nodes, and also those in the network could know the public key; however, the private key to decrypt the data in the packet could be secured inside the receiver or the intended destination. Any other security system that is more efficient than this could also be deployed if it's found suitable to the working environment.

5. FRIEND & STRANGER NODES

In general, when two nodes start communicating with each other the sender or the initiator of the communication is moved to the *stranger node* state with respect to the receiver. As the communication proceeds, the intimacy factor is augmented based on some well-defined method. After the intimacy factor crosses the threshold value, the stranger node moves to the *friend node* state again with respect to the receiving node. This transformation between the states indicates that the receiver now is starting to trust the sender and share some information regarding the routes of nodes in its vicinity. The change of state triggers the sharing of routes, which is initiated by the receiver at the end of the ongoing transactions. The speed of this state change is a very important parameter in the design of the protocol. The faster the change, the earlier the sender or the initiator obtains the neighborhood information. This also has the consequence of a malicious node being able to *quickly* get the location of various destinations and launch an attack on the network. After the state change, the receiver is identified as being ready to receive the request for sharing the information regarding nearby nodes. The nodes that are acquired from the receiver are stored in the cache with a timeout period. Like any ordinary route that is stored in the cache after the expiry of the timeout period as per the norms of the protocol the routes are cleared.

The method of shifting the state of a source node or the initiator of a communication, from stranger node to friend node could be based either on some empirical or heuristic algorithms. Empirically this could be done by maintaining a track of the messages transmitted between the nodes concerned or calculating the time during which the communication persists. It should also be noted that when the time of communication is taken into account, the factor could affect the sharing process. In fact, it could bring down the efficiency of the protocol as the time to make use of the routes acquired is reduced. A balance therefore must be found between the two parameters. On the other hand, if the factor is based on the messages

transmitted, a counter must be maintained by the receiver to count the packets received. In the aforementioned situation, the counter value could be directly used as the intimacy factor or could be weighted by any suitable constant to give the intimacy factor values.

Let the number of packets transmitted by the stranger node to receiver be P_t

$P_t \propto k^*$ *Intimacy Factor*,
where k is some constant

There also remains a good chance for the routes exposed to be already known to the sender. Under such circumstance, if possible the sender tries to correct the information that is maintained in the cache of the receiver. The sender then posts a "*Gratis Reply"* to the receiver. This informs the receiver the route, which was declared corrupt, and the new route that has to replace the corrupted one. A comparison is therefore required at the sender's side when it's receiving the exposed nodes' routes to ensure that the routes are correct. If during the comparison process the sender or the friend node to the receiver, identifies a route that is already known to it but is different from the one exposed by the receiver, it has to be able to discriminate between the right and the faulty route. The faulty one need not always be a wrong route, but can be an old route for which a newer version exists. To facilitate the required comparison an extra argument is used, which is referred to as the "*Earmark"*. The earmark is calculated by subtracting the time at which a route was discovered from a standard reference time used across the network. A mechanism can be used to either accept a standard reference or to communicate a chosen reference across nodes whichever proves feasible.

6. ILLUSTRATION

The protocol succeeds in reducing the flooding requests required by reducing the initial broadcast made to a hop count of one. If the destination is still not found, the protocol increases the depth of flooding further to locate the destination. The protocol scales well for networks of all sizes.

Considering a scenario where a node *A* in the MANET wishes to communicate with the node *12*, it sends a broadcast reply to the nodes in its coverage region. This is represented in figure 2. The immediate circle of nodes around the node *A* consist of nodes *1, 2, 3 & 4*.

In the considered situation, of these nodes only nodes *2 & 4* have some contacts with other nodes of the network. Node *2* has a route to the node *14* in its cache, and similarly node *4* has a route to node *18* in its cache. When nodes *1, 2, 3 & 4* receive the route request packet from node *A*, each node checks in its cache for the intended destination. If the route to the destination node is found in the cache of any of these nodes, then they reply back to the sender of the route request, with the message that the destination is the friend of the replying node. On the other hand, if the destination is unknown to all the

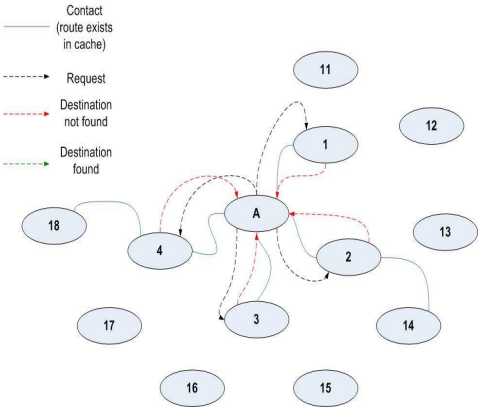

Figure 2

nodes, which is the case depicted above in figure 2; the nodes reply back with the message that the destination was not found. The source then can choose to increase the depth of flooding using some exponential algorithm, and try to locate the destination. This is presented in figure 3 below.

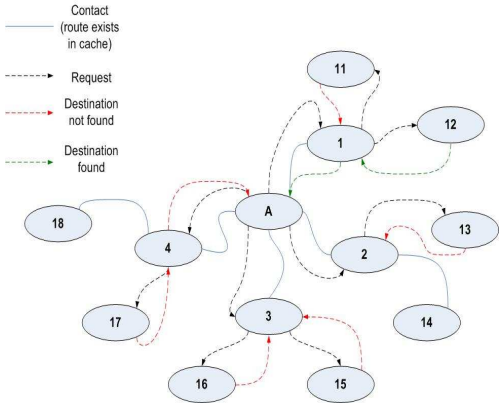

Figure 3

The nodes 1, 2, 3 & 4 now extend the flooding to their coverage region. The nodes 11 through 18 then receive the message. Since, the node *12* is the intended destination it posts a reply back to the node *1*, instead of the sender conveying the fact that it is the intended destination. The node *1* then replies to the source node *A*. The node *1* also in the process learns that the destination node *12* and also node *11* in its coverage region are in its neighborhood. Now the connection that the source node tries to establish is only between itself and the node *1*, which then once again attempts to connect to node *12* the actual destination. The packets are then transmitted from the source node *A* to the node *1*, and from there to the final destination node *12*.

A similar scenario where the node is further away from the source is shown below in figure 4. In the situation shown below the destination is node *21*, which is connected to node *12*. The source node is still considered to be node *A*. After the previous broadcast *A,* and assuming that a timeout period has not passed between the last broadcast and this

broadcast, the node *A* is assumed to have the routes to nodes *1, 2, 3 & 4*. In addition, the nodes *1, 2, 3 & 4* would have the routes to all their neighbors in their cache.

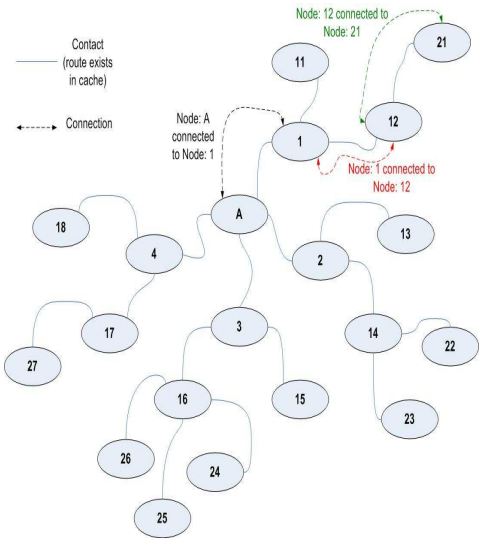

Figure 4

In addition to the above assumptions, two more assumptions are made. One, that the node *12* meanwhile established a contact with node *21* and the other, that node *1* had contacted node *12* in such a way so as to prove itself as the friend of node *12*. With the second assumption, it could be observed that since node *1* had proved be the friend of node *12*, it would have obtained a portion of the cache of node *12* as part of the neighborhood sharing. This shared information would have contained the route to node *21*. Now with all these factors in effect, when node *A* posts a broadcast reply with a hop count *1*, node *1* replies back to *A* with the message that node *21* is accessible by it. Node *A* now establishes a contact with node *12*. Then node *12* in turn connects to node *21*. The connection is now established in parts between node *A* the source or the sender with the node *21*, the receiver or the destination. The packets would then be sent by node *A* to node *12*, and then from node *12* to node *21*. This way, the messages reach the destination. The most important factor of importance is the security to be maintained as the messages are given to different nodes, in such a way that they are considered the intended destination.

7. CONCLUSION

The efficiency of the protocol has varying degrees of dependency with many parameters of the network, which includes the timeout period, the intimacy factor and others. The choice of these parameters if could be brought under the designer's control should be prudently made. The state change of a node from stranger to friend based on intimacy factor is to be probed further. Since the protocol doesn't place any constraints on the number of nodes in the network, the protocol can scale to networks of any size. The calculation of the intimacy factor remains still a topic under study. Similarly, the strategy to increase the depth of flooding each time when the destination is not found is under further research.

8. REFERENCES

[1] Imrich Chlamtac, Marco Conti, Jennifer J.-N.Liu, *Mobile ad hoc networking: imperatives and challenges*

[2] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva, *A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols*

[3] Yalin Everen Sagduyu, Anthony Ephremides, *Energy-efficient MAC in ad-hoc networks inspired by collision resolution concepts*

[4] Srdjan Krco, Frank Fitzek, Bernard Hunt, *WWRF/WG4/Ad-Hoc Networking – subgroup white paper version 1.2.1*

[5] Ioannis C. Avramopoulos, Hisashi Kobayashi, *Guaranteed Delivery Flooding Protocols for Mobile Ad Hoc Networks*

[6] V. Shurbanov, J.Redi, *Energy-Efficient Flooding in Mobile Ad-Hoc Networks*

[7] Yunjung Yi, Mario Gerla, Taek Jin Kwon, *Efficient Flooding in Ad Hoc Networks with On-demand (Passive) Clustering*

[8] Yunjung Yi, Mario Gerla, Taek Jin Kwon, *Efficient Flooding in Ad hoc Networks: a Comparative Performance Study*

[9] Eszter Kail, Gábor Németh, Zoltán Richárd Turnányi, *Throughput of Ideally Routed Wireless Ad Hoc Networks*

[10] Ronald L. Graham, Donald E. Knuth, Oren Patashnik, *Concrete Mathematics, Addison Wesley Publishing Company, Pub., 1989*

[11] Prince Samar, Zygmunt J. Haas, *Strategies for Broadcasting Updates for Proactive Routing Protocols in Mobile Ad hoc Networks*

[12] Yoav Sasson, David Cavin, André Schiper, *Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks*

ABOUT THE AUTHORS

[1]**Balakrishnan C** is a pre-final B.Tech Inf Tech student from the S.R.M Engineering College

[2]**Sudarsun S** is a Senior Scientist working for Checktronix India Pvt Ltd, Chennai. He has his master's degree in Computer Science from IIT Madras and bachelor's degree in Electronics and Instrumentation Engg from Madras University with a gold medal.

[3]**Bharathi Mani** is a pre-final B.Tech Inf Tech student from the S.R.M Engineering College

[4]**Srinivasa Raghavan** is a sophomore B.Tech Inf Tech student from the S.R.M Engineering College