# SECURE AND EFFICIENT DATA TRANSMISSION IN WIRELESS SENSOR NETWORK USING AUGMENTED TREE-BASED ROUTING PROTOCOL

**N.V.CHINNASAMY, DR. A.SENTHILKUMAR**

**Abstract** - Wireless sensor networks (WSNs) in different types of applications are used, such as fire monitoring, urban sensing, and perimeter surveillance. The recent year wireless sensor networks are broadly used in surveillance tasks, environmental control, tracking, monitoring and controlling etc. The wireless sensor networks need very secure communication in wake of them being in open field and being based on broadcasting technology. Different protocols or algorithms are designed to improve the energy of wireless sensor network. In this paper we discuss about the Most of proposed routing protocols do not operate efficiently with networks of more than a few hundred nodes. In this propose method is an augmented tree-based address space structure and a hierarchical multi-path routing protocol, referred to as Augmented Tree-based Routing (ATR), which utilizes such a structure in order to solve the scalability problem and to gain good flexibility against node failure, mobility and link congestion. Simulation results and performance comparisons with existing protocols substantiate the effectiveness of the ATR.

**Keywords:**-Networking, routing, Wireless Sensor Networks, Secure and Efficient Data Transmission, Cluster Based Wireless Sensor Networks, Dynamic addressing, distributed hash table.

I. **Introduction:** Wireless Sensor Networks consist of numerous autonomous sensor nodes devices are spatially distributed to sense and monitor various changes of the environment surrounding us. Such devices are also capable to communicate in wireless sensor networks and that can also sense, monitor, transmit, receive or process numerous data like pressure, temperature, sound, motion, humidity etc. The following section discussed about various sensor deployment environments and previously deployed schemes in the proposed domain.

The data transmission in WSNs can be done in two ways: (i) centralized (ii) decentralized. Centralized means such data processing and transfer can be carried out through the medium of a base station in WSNs [3]. In case of distributed or clustered wireless sensor environments, every cluster has obtained a high-configuration node called a cluster-head (CH). A sensor node of one cluster can only communicate with the other clusters sensor node by taking the first, permission of the respective cluster. In this function of cluster head to aggregate all the data sent by sensor nodes present in its neighborhood. Cluster-heads sent all the data to the master storage known as base station (BS).

In this paper schemes are hierarchically organized exploiting a tree structure for the address space management and routing. Routing protocols based on such addressing schemes can determine low performance and poor resilience to node failure/mobility [11]. In this problem of the incomplete information embedded in the tree-based addressing scheme, we propose to augment the tree structure by storing additional information in the node routing tables. the underlying neighbour discovering procedure. The main advantage of the proposed routing protocol, referred to as Augmented Tree based Routing (ATR) protocol, lies in the richer topology knowledge that allows one to resort to multi-path routing.

**II. Related Work :** Several researchers have studied problems related to data aggregation in WSNs.

**A. Data Aggregation in a Trusted Environment :** The Tiny Aggregation Service (TAG) to compute aggregates,such as Count and Average, using tree-based aggregation algorithms were proposed in [3]. Tree based aggregation algorithms to compute an order-statistic (i.e., quantile) have been proposed in [9].To address the communication loss problem in tree-based algorithms an aggregation framework called synopsis diffusion is designed in [9], which computes Count and Sum using a ring topology. Very similar algorithms are independently proposed in [3]. These works use duplicate-insensitive algorithms for computing aggregates based on [9]‟s algorithm for counting distinct elements in a multi-set.

**B. Secure Aggregation Techniques :** Several secure aggregation algorithms have been proposed assuming that the BS is the only aggregator node in the network [10]–[12]. These works did not consider in-network aggregation. Only recently, the research community has been paying attention to the security issues of hierarchical aggregation.

**1. Tree Based Data Aggregation :** Tree-based data aggregation approach builds an aggregation tree. This tree is a minimum spanning tree, sink node as root node and leaves consider as source node. In this technique data is transferred from leaves node to sink node and aggregation is done by parent nodes.

**2. Centralized Data Aggregation :** Data is gathering at centre node in centralized data aggregation technique. For this process it takes the help of shortest path using a multi-hop wireless protocol. The sensor nodes send the data packets to a centre node, which is the powerful node. The leader aggregates the data which can be queried. Each intermediate node has to send the data packets addressed to leader from the child nodes. So a large number of messages have to be transmitted for a query in the best case equal to the sum of external path lengths for each node.

**III - Attacks In Wireless Sensor Network Routing:** In this section, first some basic types of attacks that can be launched in WSNs. A more specific look on WSN network layer attacks will be taken.

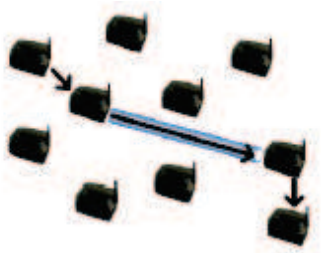
A. **General types of WSN attacks:** Attacks on WSNs can be classified into one or more of the following categories.

**Outsider vs. Insider attack:** in an outsider attack, a malicious node harms the WSN without being part of it. In contrast, in an insider attack the malicious node harms the WSN as (authorized) participant of the WSN.

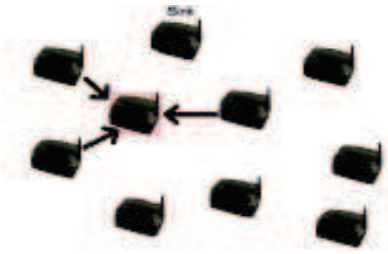
**Physical vs. Remote attack:** in a physical attack an adversary physically accesses the sensor node that should be harmed by tampering or destroying the sensor's hardware. In contrast, a remote attack is implemented from a (large) distance, e.g. by emitting a high-energy signal to interrupt the communication.

**Passive vs. Active attack:** in a passive attack an adversary just eavesdrops or monitors the communication within the WSN. In contrast, in an active attack the adversary directly influences the communication in the WSN by modifying, fabricating or suppressing data packets.

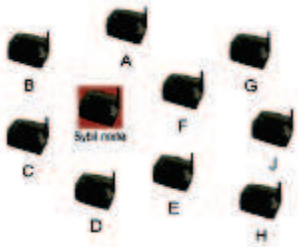
**Laptop-class vs. Mote-class attack:** a mote-class attack is an attack against a WSN that is implemented from a mote, i.e. the attacking device is of same type of hardware as the sensor nodes that should be attacked. In contrast, in a laptop-class attack, the adversary utilizes a device which is superior to the sensor nodes that should be attacked in terms of computational power and transmission power.



**Wormhole attack**



**Sinkhole attack**




**Sybil attack**

**Figure 1. WSNs Attacks**

**B. Attacks on the network layer**

There are several attacks that can be launched against the network layer in WSNs. Most of the attacks on the network layer can be classified in one of the following categories (see Figure 1):

**Information disclosure:** disclosure of routing information by passive or active participation in the WSN.

**Physical attack:** unauthorized access to sensor node through physical intervention.

**Energy exhaustion:** intentional waste of energy resources by adversaries, e.g. by requesting unnecessary routes.

**Denial of service:** flooding of the network with unnecessary routing requests.

**Spoofed, altered or replayed routing information:** changing the routing behavior by spoofing, altering or replaying routing information.

**Routing table overflow:** flooding of the routing table by creating multiple non-existing routes to make the routing algorithm collapse.

**HELLO flood attack:** intentionally inject bogus HELLO messages to remote nodes to confuse the routing protocol.

**Sybil attack:** creating a large number of pseudonymous entities to gain a greater influence on the network.

**Sinkhole/Blackhole attack:** trying to obtain all network packets in a certain network area by "looking attractive" to surrounding nodes.

**Wormhole attack:** making two nodes believe that they are neighbors by tunneling packets using a low latency link, though, in reality, they are far away from each other.

**Selective forwarding:** forwarding only certain packets in the network to save resources, i.e. behaving selfishly.

**IV. Preliminaries To Atr Protocol:** The proposed protocol can be represented as a binary tree of **L+1** levels, where L is the number of bits used for an address (Fig. 2).

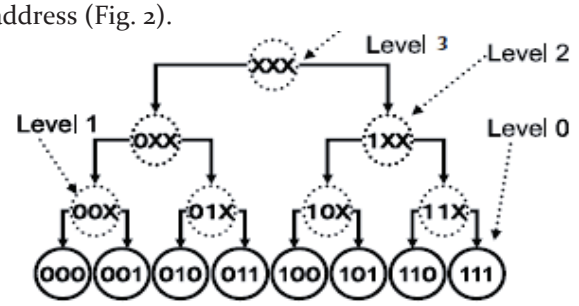



**Figure 2 – Address Space structure**

The leafs represent the network addresses, whereas the physical connections are represented by dotted lines and they do not necessarily correspond to the branches of the address space tree. In such structure, a level-k subtree is a set of nodes sharing an address prefix of (L-k) bits: a level-0 subtree is a leaf node, where [0xx] is a level-2 subtree containing the four addresses [000], [001], [010] and [011]. A level-k sibling of a given address is defined as the subtree that shares the same immediate parent of the level-k subtree of the considered address. Such an address allocation structure, exploited in the Dynamic Address RouTing (DART) protocol [12], assures that nodes, whose routing addresses share the same prefix, form a connected sub-graph in the network topology. In particular, the longer the shared address prefix between two nodes, the shorter the expected routing distance in the network topology. However, this treebased structure has a low fault-tolerance, since it exists only one path between a node and a sub-set of destinations, i.e. a sibling. The failure of a next hop breaks the connectivity of the network, leaving the destination set disconnected from the node. Another major weakness is that this structure suffers from traffic congestion. This is due to the availability of a unique next hop as a gateway for a whole sibling, i.e. a set of destination nodes, which can be constituted by many nodes.

To overcome such drawbacks, in this paper we propose to augment the tree structure by adding redundant paths for the packet forwarding. More specifically, unlike DART protocol in which each node maintains only one possible next hop toward the final destination (defining a single path along the tree structure of the address space), in ATR each node maintains and explores all the possible paths through its neighbours to reach the final destination. This is equivalent to use an augmented tree structure to perform forwarding, which slightly increases cost.

| | | |
|---|---|---|
| 01000010 | 01000010 | 01000010 |
| 10001111 | 10001101 | 10001111 |
| 00011010 | 00011010 | 00011010 |
| 00101010 | 00101010 | 00101010 |
| 01110111 | 01000010 | 01110111 |
| 01001011 | 01001010 | 01001011 |
| 11111101 | 11100000 | 11111101 |
| 01001110 | 01001110 | 01001110 |

**Physical DART logical ATR logical Adyacency Matrix Adyacency Matrix Adyacency Matrix**

**Figure 3 – Adjacency Matrix for 8 Nodes network**

In Fig. 3 shows the overlay graphs associated with the different path discovery resultsfor a full mesh network with four nodes. the paths from each node towards two destinations, say node '2' and '4'. The graphs evidence the presence of multiple paths towards the same destination in ATR. Moreover, they show that hierarchical single-path routing protocol does not always provide the shortest path, neither when the network is very simple.

To understand the potentiality of the proposed method, in Fig. 2 we have represented the adjacency matrixes associated with the physical and overlay graphs referring to a network with eight nodes. These matrixes differ for their numbers of '1' (communication links). The first one on the left represents the physical graph, i.e. the graph in which the edge eij is present if a physical communication link is available between the nodes i and j. The lack of ten edges ('1') in the DART matrix, with respect to the physical and ATR ones, evidences the inability of shortest-path routing protocols, as DART, to build a complete topological view of the network. As regards the routing issues, Fig. 3 shows the overlay graphs associated with the different path discovery results (DART and ATR) for a full mesh network with four nodes.

The graphs show the paths from each node towards two destinations, say node '2' and '4'. The graphs evidence the presence of multiple paths towards the same destination in ATR. Moreover, they show that hierarchical single-path routing protocol does not always provide the shortest path, neither when the network is very simple.

**V - Augmented Tree-Based Routing Protocol:** We differentiate four primary processes in ATR protocol. The Path Discovery Process updates the routing table of each node with the routing update sent by neighbour nodes in the hello packets.

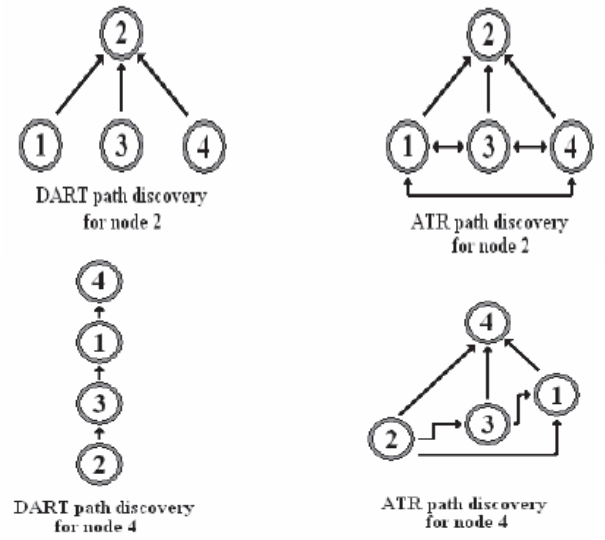



**Figure 4 – Graphs referring to path discovery process**

The Packet Forwarding Process singles out the right path to route the packets towards the destination. The Address Allocation Process selects a network address that reflects the node topological position inside the network. Finally, the Address Lookup Process provides the mapping between the unique node identifier used by higher levels and the transient network address used by the Path Discovery and Packet Forwarding processes.

**A. Path detection Process:** The proposed path discovery process is a multi-path version of the proactive shortest-path distance-vector one adopted by DART. It takes advantage of the augmented tree structure previously introduced to reduce the effects of node mobility and wireless propagation instability, without increasing the routing overhead. The Path Discovery Process uses the locally broadcasted hello packets exchanged by neighbour nodes to build up and update the routing tables. ATR differs from DART in the number of entries to reach each sibling stored in the routing tables. DART stores a unique entry for each sibling, the shortest one that the node could find. In this way, if the shortest path is unavailable
due to mobility, congestion or wireless propagation instability, the node could immediately route the data packets along one of the multiple available paths.

**Table 1 – Routing Update of node 1 (address[000])**

| Level | Sibling | NID | Cost | RouteLog |
|---|---|---|---|---|
| 0 | 001 | 3 | 1 | 001 |
| 1 | 01X | 2 | 1 | 010 |
| 2 | 1XX | 1 | 1 | 100 |

Table 1 shows an example of the routing update broadcasted by the node '3' with network address [000] in the network of Fig. 4. This routing update advices neighbor nodes only about which destination siblings the sending node could forward packets to, but it does not give information concerning the specific path the packets will be forwarded along. The routing updates need to store only binary information: "There is no route" or "There is at least one route", so that the routing overhead is the same for both in DART and ATR.
adopt a simple hop count metric. The routeLog is used by the loop avoidance mechanism to discard a route updating, already received. If a node does not receive any hello packets from a neighbor in a certain number of update periods, the expired

**Table 2 – DART Routing table of node 4 (address[001])**

| Level | Sibling | nextHop | ID | cost |
|---|---|---|---|---|
| 0 | 000 | 000 | 1 | 1 |
| 1 | 01X | 010 | 2 | 1 |
| 2 | 1XX | 100 | 3 | 1 |

Let's describe how a node updates its routing table. Suppose that the node '4' with network address [001] (Fig. 3) receives the routing update from node '1', as shown in Table 1. First, node '4' adds an entry for the sibling the address '000' belongs to, i.e. the level-0 sibling [000], with a one-hop cost and node '1' as next hop. Then it looks if the neighbour could act as forwarder for the higher level-k siblings, inspecting if the corresponding cost of the routing update has a finite value. In this example the node '4' adds node '000' as forwarder towards the level-1 sibling [01X] and the level-2 sibling [1XX], both with cost 2.

**Table 3 – ATR routing table of node 4 (address[001])**

| Level | Sibling | nextHop | ID | Cost |
|---|---|---|---|---|
| 0 | 000 | 000 | 1 | 1 |
| 1 | 01X | 000 | 1 | 2 |
| | | 010 | 3 | 1 |
| 2 | 1XX | 000 | 1 | 2 |
| | | 010 | 2 | 2 |
| | | 100 | 2 | 1 |

In Table 2 we report the routing table of node '4', for the network of Fig. 4, built by DART protocol, while Table 3 shows the same routing table built by ATR protocol. Also when the network size is very small, only four nodes, the ATR Path Discovery Process can take advantages of multiple neighbours in order to forward packets, thanks to its multipath approach

and its augmented tree-based address-space structure.

**B. Packet Forwarding Process:** The ATR multi-path routing exhibits temporal diversity, i.e. the Path Discovery Process performs a pre-emptive route discovery before the occurrence of route errors. Moreover, ATR could be easily extended to split a data transfer on multiple paths in the spatial domain, to reduce congestion effects and end-to-end delay. Let us describe the proposed Packet Forwarding Process. According to Table 3, if node '4' with network address [001] must forward a data packet to a node with network address [010], it first looks the entries related to the sibling the destination network address belongs to, i.e. the level-1 sibling [01X]. In this case there are two entries in the routing table, so node '4' will pick up the one exhibiting the least hop count metric, i.e. the node [010]. Otherwise, if there are no entries for the level-1 sibling, node '4' will expand its search to higher sibling, i.e. level-2 sibling [1XX].

we take advantage of multi-path defining a cross layer solution to handle with link failures. If a node detects a link failure after the forwarding of a data packet, namely if it does not receive the acknowledgement, the previously used next hop is invalidated. Then the data packet will be reforwarded using a different path already discovered by the Path Discovery Process. Evidently this leads to higher delays in packet delivery, however it is often more convenient to wait a little more instead of wasting the resources used up to here in packet forwarding [2]. The use of this link-breakage detection technique is another difference of the proposed approach with respect to DART.

**C. Address allotment Protocol:** ATR protocol makes use of the same address allocation, a distribution stateful approaches is based on multiple disjoint tables. when a node joins a network and selects an address, it keeps also the control over a subset of the address space, i.e. a sibling. Nodes exchange information about the utilized addresses and perform both the network-merging event detection and the partition one by locally broadcasting the hello packets. Here, we point out only our following change to address allocation process utilized by DART protocol, which allows to solve the following issue present in the original address selection procedure. When a node joins a network, it must choose a neighbour to get a valid network address. DART protocol suggests to choose the neighbour with the largest unoccupied address space, i.e. the highest free level-k sibling, to balance the routing table size among nodes.

This procedure, as shown by simulation results, never converges also for small networks since the routing table of the selected neighbour could be not update, and, therefore, the joining node could pick up an invalid address. Our proposal solves this issue by using as metric for the neighbour selection both the free address-space criteria and the node identifier. Moreover, if an invalid address is acquired from the first selected neighbour, the ATR address selection procedure scrolls the set of neighbours until a valid address will be obtained.

**D. Address Lookup Process:** The Address Lookup Process is built upon a DHT, our proposal makes use of caching techniques to reduce the delay and the overhead due to the procedure of looking up a network address of a node starting from its identifier. We want to underline that the proposed caching technique could be also used to provide fault tolerance to the whole process. Moreover, we investigate the issue of finding a good hash function, i.e. a hash one that balances the lookup traffic among nodes, with respect to the adopted address allocation procedure.

**VI. Simulation Results:** We present a numerical performance analysis of the proposed routing protocol, resorting to ns-2 network simulator. The duration of simulation experiment is set to 750 seconds, while the sizes of the scenario areas are chosen to keep the node density equal to 64 nodes/Km2. This value corresponds to a mean node connectivity degree of 12, which is a reasonable value to avoid the presence of isolated nodes. To generate mobile topologies, we have adopted the Random Way-Point as mobility model. Since proactive routing protocols are not suitable for networks with very high levels of mobility, the speed values are uniformly taken in the [0.5m/s; 5m/s] range, and the pause times uniformly taken in [0s; 100s]. Our comparison does not include the network address lookup layer, which is replaced with a global lookup table available to all nodes.

However in order to develop the proposed address lookup process, and in particular to adopt effective caching techniques, we have extensively investigate the overhead due to the network address lookup and update functions, and here we report a subset of the simulation results.

**A. Path Discover Process:** As already explained, the proposed multi-path approach has no effect on the routing overhead: both the size and the rate of the hello packets are the same for DART and ATR protocols. Instead, the node memory requirements are not the same: ATR protocol requires that nodes store all the available paths towards each sibling.

In this subsection, we evaluate the memory requirements of ATR and compare them with the ones of DART, in terms of routing table size. We have run a set of trials to measure the average number of routing tables entries of all participating nodes. As shown in Fig. 5, ATR exhibits stronger

memory requirements. However, as the number of nodes grows, the number of routing table entries saturates: this confirms that the proposed augmented tree-based address space structure scales satisfactorily.

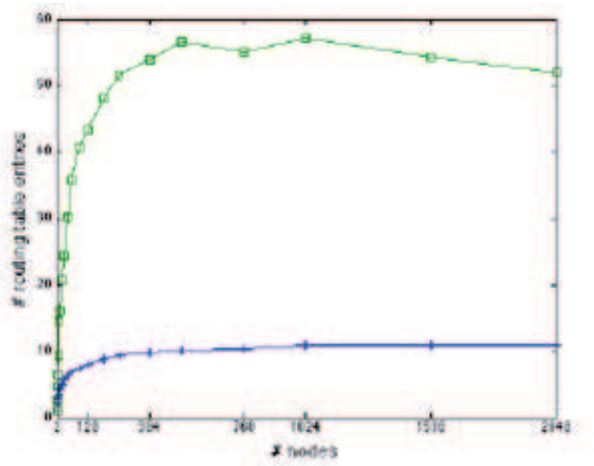


**Figure 5 – Memory requirements Comparison**

**B. Packet Forwarding Process**

Neither DART nor ATR were designed to optimize the throughput. In fact, their main requirement is to achieve scalability. Moreover, they are lacking in optimization work behind more widespread protocols. we are interested in comparing ATR not only with DART, but also with other two popular routing protocols, AODV and DSR . Let us underline that ATR does not adopt spatial diversity multi-path routing, so the comparison with shortestpath protocols makes sense.

In comparing these protocols, various types of Metrics:packet delivery ratio, path stretch measured in number of hops, and routing overhead. The data traffic is modeled as CBR flows over UDP protocol. We do not adopt the TCP as transport protocol to avoid the effects of elasticity of TCP flow control on routing performances. The data pattern is the most common one in simulation for ad hoc networks: the Random Traffic Model. The global load offered is kept constant at 250KB/S, in order to avoid running out of capacity due to multi-hop approach. The node load offered scales as O(1/n) to simulate sustainable data traffic, taking in account the routing overhead. Each flow has a start- and end- time uniformly picked in [450s, 720s], in order to achieve the address allocation convergence before data forwarding can be performed, and to guarantee a 30 seconds cold-down period to complete data packets delivery.

The packet delivery ratio describes the loss rate that will be seen by upper layers protocols. Fig. 6 (and the following in this subsection) shows this metric normalized to the packet delivery ratio of the ATR protocol. The experimental results show that the ATR scales always better than DART. Regarding to reactive protocols, when the number of nodes is relatively small, they perform well. However, the performances of ATR remain comparable with respect to AODV and DSR ones also in such situation. Differently, when the number of nodes grows, the reactive protocols lose their initial performance advantage and ATR outperforms. This trend is earlier manifested by DSR that exhibits the worst performance.

Fig. 7 shows the mean path stretch normalized to the one of ATR protocol. In absence of congestion, the path stretch measures the ability of a routing protocol to efficiently use network resources by selecting the shortest path to reach the destination.

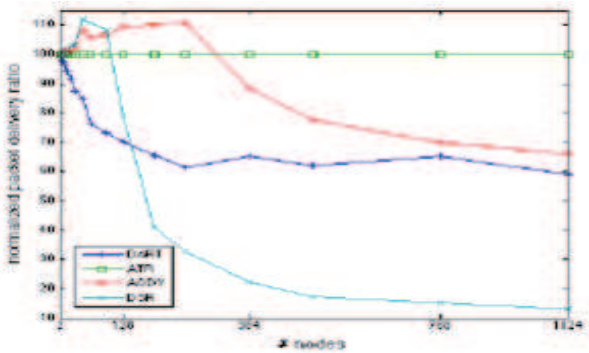


**Figure 6 – Normalized Packet delivery ratio**

The ATR multi-path property leads to longer paths with respect to shortest-path protocols, When the number of node grows, ATR delivers more packets with respect to the other protocols; then it is reasonable to assume that the packets delivered by ATR, and missed by the other ones, experience longer paths. This is also confirmed by comparing DART and ATR performances for small networks. When both the protocols reach the same packet delivery ratio values, ATR is able to find paths shorter of about 65% with respect to DART induced ones.

The last metric, the mean routing overhead normalized to the one of ATR protocol, measures the ability of a protocol of working well in congested or low-bandwidth environments.

Let us note that, for DSR and AODV simulations, we count routing packets sent over multiple hops as a singletransmission, as usually done, also if DART and ATR use only locally broadcasted routing packets.

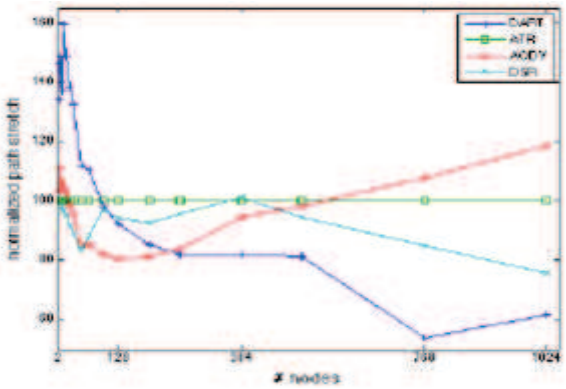



**Figure 7 – Normalized path Stretch**

Fig.8 points out that the routing overhead of DART and ATR are perfectly comparable, that is the packet delivery performance gain of ATR is obtained with no additional overhead, with the exception of memory requirements. when the number of nodes grows, the hierarchical routing overhead becomes perfectly comparable with the reactive one, wasting the aim of reactive protocols.

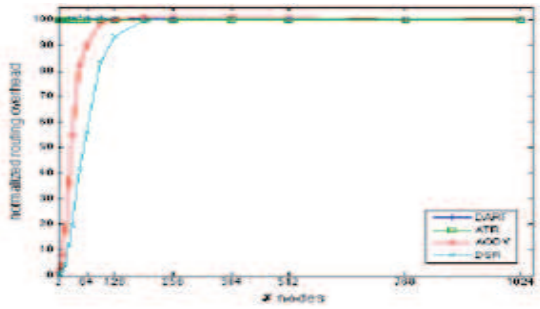



**Figure 8- Normalized routing overhead**

**C. Address Allocation Process**: To evaluate the improvements of our Address Allocation Process with respect to DART one, we set up a set of experiments with static topologies, no data traffic, simulation time equal to 450 seconds and nodes uniformly distributed. We measure the average times of last duplicate address and last invalid address event, for all the participating nodes. the DART address allocation procedure never converges when the number of nodes is mode than 32.

**D. Address Lookup Process:** To analyze the overhead due to the address lookup and update functions, we have chosen two metrics: the mean rate of network address updates and the mean number of network address bits that have been changed. The former measures the temporal locality of the network addresses; the latter measures the spatial locality of the network address, and should be used to define the caching techniques.

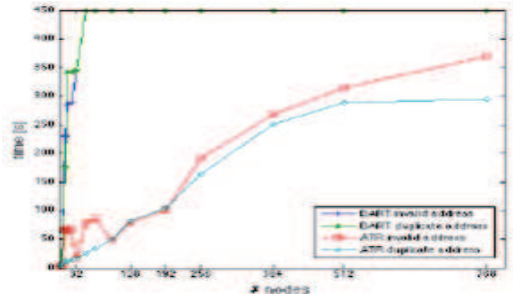



**Figure 9 – Address Allocation Process Convergence**

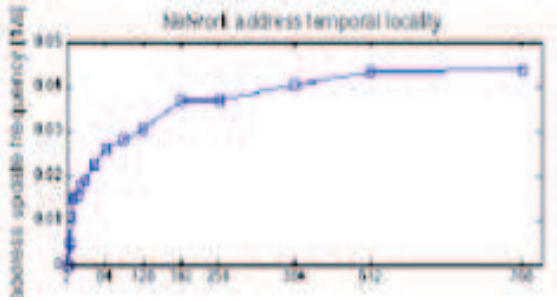



**Figure 10 – Address Lookup Process Metrics**

The mean address-update rate is measured in updating per second and its trend initially grows more then linearly. The maximum value is in the order of an updating every 20 seconds that is almost an order of magnitude higher than the time period necessary to route a data packet. Regards to the spatial locality metric, the high number of changed bits suggests adopting distributed caching techniques.

**VII. Conclusion:** This paper presents a secure data transmission for cluster-based WSNs (CWSNs), where the clusters are formed dynamically and periodically the security issues of in-network aggregation algorithms to compute aggregates such as predicate Count and Sum. The paper proposes a hierarchical multi-path routing protocol, referred to as Augmented Tree-based Routing (ATR) protocol, which exploits a new augmented tree-based address space structure, in order to solve the scalability problem and to gain good resilience against node failure/mobility and link congestion/instability in MANETs. Simulation results and performance comparisons with existing protocols substantiate the effectiveness of the ATR.

The Future work will focuses on developing new routing algorithms for routing the data from the source to the sink. This approach should confront with the difficulties of topology construction, data routing, loss tolerance by including several optimization techniques that further decrease message costs and improve tolerance to failure and loss.

**References:**


1. X. Hong, K. Xu and M. Gerla. "Scalable routing protocols for mobile ad hoc networks Network". IEEE Network, vol. 16, no. 4, 2002,
2. V.Kaladevi,Sharmila Devi, Detour Distance Energy of Some Graphs; Mathematical Sciences international Research Journal ISSN 2278 – 8697 Vol 3 Issue 2 (2014), Pg 677-681
3. J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu and J. A. Jetcheva. "A performance comparison of multi-hop wireless ad hoc network routing protocols". United States,1998.
4. Ishfaq A. Ganaie, Jitender Rattan, Ajay Kumar Mittal, V.K. Kukreja, Simulation of Packed Bed of Porous Particles; Mathematical Sciences International Research Journal ISSN 2278 – 8697 Vol 3 Issue 1 (2014), Pg 134-141
5. Y. Tseng, S. Ni, Y. Chen and J. Sheu. "The broadcast storm problem in a mobile ad hoc network". Wireless Networks, vol. 8, no. 2, 2002.
6. I.Chlamtac, M. Conti and J. Liu. "Mobile ad hoc networking:imperatives and challenges". Ad Hoc Networks, vol. 1, no. 1, 2003.
7. D.Anuradha,D.Balaji, Heuristic Algorithm for Finding More-for-Less; Mathematical Sciences International Research Journal ISSN 2278 – 8697 Vol 3 Issue 1 (2014), Pg 129-133
8. I.Akyildiz, X. Wang and W. Wang. "Wireless mesh networks: a survey". Computer Networks, vol. 47, no. 4, 2005, pp. 445-487.
9. M. Gerla, X. Hong and G. Pei. "Landmark routing in ad hoc networks with mobile ackbones". J. of Parallel and Distributed Computing, 2003.
10. J. Eriksson, M. Faloutsos and S. Krishnamurthy. "Peernet: Pushing peer-2-peer down the stack". In Proc. of IPTPS, 2003.
11. R.Mageswari, K. Srinivasa Rao, K. Sivakumar, Protein Similarity/Dissimilarity Using Contact Map; Mathematical Sciences international Research Journal ISSN 2278 – 8697 Vol 3 Issue 2 (2014), Pg 672-676
12. Perkins and E. Royer. "Ad hoc on-demand distance vector routing".2nd IEEE Workshop on Mobile Computing Systems and Applications,
13. New Orleans, LA, United States, 1999,
14. B. Johnson and D. A. Maltz. "Dynamic source routing in ad hoc wireless networks". In Mobile Computing, vol. 353, 1996, pp. 153-181.
15. J. Eriksson, M. Faloutsos and S. Krishnamurthy. "DART: Dynamic Address RouTing for Scalable Ad Hoc and Mesh Networks". IEEEACM Transactions on Networking, vol.15, no. 1, 2007, pp.119-132.
16. Manoj Solanki, Ramakant Bhardwaj, Arvind Bhore, Some Common Fixed Point theorems in 2-Metric Space, for Rational Expression; Mathematical Sciences international Research Journal ISSN 2278 – 8697 Vol 4 Issue 1 (2015), Pg 55-61
17. S. Nesargi and R. Prakash. "MANETconf: Configuration of hosts in a mobile ad hoc network". In Proc. of IEEE INFOCOM, 2002.
18. H. Zhou, L. M. Ni and M. W. Mutka, "Prophet address allocation for large scale MANETs,". Ad Hoc Networks, 2003.
19. M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea and A. Rowstron."Virtual ring routing: network routing inspired by DHTs",, 2007.
20. M. Caleffi, G. Ferraiuolo, and L. Paura. "On reliability of dynamic addressing routing protocols in mobile ad hoc networks. In proc ofWRECOM '07, Rome, Italy, 2007.
21. J. Yoon, M. Liu and B. Noble. "Random waypoint considered harmful".
22. Meenakshi.K, Hanumesha.A.G, Semigraphs And Fermats Theorem; Mathematical Sciences International Research Journal : ISSN 2278-8697Volume 4 Issue 2 (2015), Pg 398-400
23. In Proceedings of IEEE INFOCOM 2003,
24. G. Holland and N. Vaidya. "Analysis of TCP performance over mobile ad hoc networks" 1999.


N.V. Chinnasamy, Research Scholar, Periyar University, Salem, Tamil Nadu.
Dr. A. Senthil Kumar
Assistant Professor, Dept. of Computer Science, Arignar Anna Govt. Arts College, Namakkal, Tamil Nadu.