International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 6, November-December 2014 **ISSN 2278-6856**

Comparison of Ant-Net, Honeybee and SRAAA Protocols in MANETs

[1]Capt. Dr. S. Santhosh Baboo, [2]V J Chakravarthy

[1]Associate Professor,D G Vaishnav College, Chennai, India

[2]Research Scholar, D G Vaishnav College, Chennai, India

Abstract

Mobile Ad Hoc networks (MANETs) is a collection of small randomly dispersed nodes that have the capability to move, establish wireless communication with each other and do computations and process operations. A network especially a wireless network strongly depends on the routing protocols to route the sensed data to the Base Station (BS) via some intermediate nodes. Due to the fast emergence of the wireless sensing, a lot of work has been done on the various categories of routing protocols of MANET like location-based, data-centric, hierarchical routing protocols etc. to measure the network performance. But recent studies are provided with the evidence that Quality-of-Service (QoS) routing can enhance the network performance by increasing the network utilization, compared to routing that is not sensitive to QoS requirements of traffic [1]. So in this paper, the focus is on evaluation and comparison of the network performance in the MANET having QoS routing. The comparison of the three QoS routing protocols AntNet, Honey Bee Technique and SRAAA (Secured Right Angled and Ant Search Hybrid Routing Protocol) on the basis of various performance metrics such as Bit Error Rate (BER) vs Signal-to-Noise Ratio (SNR), Average End-to-End Delay vs. BER, Packet Delivery Ratio vs BER, Energy Consumed vs BER, Network Lifetime vs Energy Consumption, Throughput vs BER and Throughput vs. SNR has been done in this research paper. On the basis of observed simulation results, it is concluded that the performance of SRAAA is better than the performance of the other two comparing protocols.

Index Terms.

Keywords:- WSNs, MANETs, Swarm Intelligence, Ant-Net Routing, Honeybee Routing, performance analysis.

1. INTRODUCTION

Swarm intelligence, as demonstrated by natural biological Ants and bees, has numerous powerful properties desirable in many engineering systems, such as network routing. In addition, new paradigms for designing autonomous and scalable systems may result from analytically understanding and extending the design principles and operations exhibited by intelligent biological Ants and Bees [2]. A key element of future design paradigms will be emergent intelligence – simple local interactions of autonomous Ant/Bee members, with simple primitives, giving rise to complex and intelligent global behavior. Communication network management is becoming increasingly difficult due to the increasing size, rapidly changing topology, and complexity of communication networks. A new class of algorithms, inspired by swarm intelligence, is currently being developed that can potentially solve numerous problems of modern communications networks. These algorithms rely on the interaction of a multitude of simultaneously interacting agents. A survey of such algorithms and their performance is presented here[3]. Modern communication networks are becoming increasingly diverse and heterogeneous. This is the consequence of the addition of an increasing array of devices and services, both wired and wireless. The need for seamless interaction of numerous heterogeneous network components represents a formidable challenge, especially for networks that have traditionally used centralized methods of network control. This is true for both packet-switched and virtual circuit networks, and the Internet, which is becoming an ever more complex collection of a diversity of subnets. The need to incorporate wireless and possibly ad-hoc networks into the existing wire-link infrastructure renders the requirement for efficient network routing even more demanding. Routing algorithms in modern networks must address numerous problems. Two of the usual performance metrics of a network are average throughput and delay. The interaction between routing and flow control affects how well these metrics are jointly optimized. The balance of delay and throughput is determined by the flow-control scheme – good routing results in a more favorable delay-throughput curve [2]. Quality of service (QoS) guarantee is another important performance measure. Here, a user might require a guaranteed allocation of bandwidth, a maximum delay, or a minimum hop-count. Such guarantees only make sense for virtual-circuit networks. This is because in applications that require logical connections there is demand for a minimum flow rate of data. This is unlike packet-switched types of service where best-effort routing is implemented. Although logical connections use static routing, the establishment of the connection is prone to the same problems that affect routing in the rest of the network. Current routing algorithms are not adequate to tackle the increasing complexity of such networks. Centralized algorithms have scalability problems; static algorithms have trouble keeping up-to-date with network changes; and other distributed and dynamic algorithms have oscillations and stability problems[1]. Swarm intelligence routing provides a promising alternative to these approaches. Swarm intelligence utilizes mobile software agents for network management. These agents are autonomous entities, both proactive and reactive, and

have the capability to adapt, cooperate and move intelligently from one location to the other in the communication network. Swarm intelligence, in particular, uses stigmergy (i.e. communication through the environment) for agent interaction. Swarm intelligence exhibits emergent behavior wherein simple interactions of autonomous agents, with simple primitives, give rise to a complex behavior that has not been specified explicitly[1].

2. ROUTING ALGORITHM IN MANET

Routing algorithms can be classified as static or dynamic, and centralized or distributed. Centralized algorithms are usually used in legacy routing systems and have problems with scalability and inordinate demand for managing decisions requiring human attention. Another drawback is the inability of the network to recover in case of failure at the central controlling station. Static routing assumes that network conditions are time-invariant. The method does not assess the network load when trying to find the shortest-path route. Maximizing throughput for a time varying load in a limited-capacity transmission line is an NP-complete problem. Adaptive routing schemes also have problems, including inconsistencies arising from node failures and potential oscillations that lead to circular paths and instability. Another problem with adaptive algorithms applied to ad-hoc networks arises when changes in the network occur too frequently to allow routing updates to propagate throughout all network nodes [1]. A network is called combinatorial stable if it changes sufficiently slowly for the routing updates to be propagated to all the nodes. Routing algorithms can also be classified as minimal or non-minimal. Minimal routing allows packets to follow only minimal cost paths, while non-minimal routing allows more flexibility in choosing the path by utilizing other heuristics. Minimal routing can further be subdivided into optimal routing and shortest-path routing. In the former, the objective is to optimize the mean flow of the entire network; while in shortest-path routing the goal is to find the minimum-cost path between two nodes [2]. Another class of routing algorithms is one where the routing scheme guarantees specified QoS requirements pertaining to delay and bandwidth. These algorithms are usually message based, i.e. they find a feasible path satisfying the QoS constraints based on an exchange of messages between the nodes. These algorithms have the tendency to temporarily overuse network resources until they find goal is to balance the load throughout all network resources without idleness and overloading.

3. SWARM INTELLIGENCE

Swarm Intelligence appears in biological swarms of certain insect species. It gives rise to complex and often intelligent behavior through complex interaction of thousands of autonomous swarm members. Interaction is based on primitive instincts with no supervision. The end result is accomplishment of very complex forms of social behavior and fulfillment of a number of optimization and other tasks. The main principle behind these interactions is called stigmergy, or communication through the environment [2]. An example is pheromone laying on trails followed by Swarms. Pheromone is a potent form of hormone that can be sensed by Swarms as they travel along trails. It attracts Swarms and therefore Swarms tend to follow trails that have high pheromone concentrations. This causes an autocatalytic reaction, i.e., one that is accelerated by itself. Swarms attracted by the pheromone will lay more of the same on the same trail, causing even more Swarms to be attracted. Another form of stigmergy alters the environment in such a manner as to promote further similar action by the agents. This process is dubbed task-related stigmergy [4]. An example is sand grain laying by termites when constructing nests. In the initial stages of construction, termites lay sand grains at random locations. This stimulates further laying by other members of the swarm, until a single heap of sand grains randomly reaches a critical mass that is larger than its neighboring heaps. At that point, most termites are attracted to that specific heap, thereby selecting that specific site for construction of their nest. Swarm intelligence boasts a number of advantages due to the use of mobile agents and stigmergy these are:

Scalability: Population of the agents can be adapted according to the network size. Scalability is also promoted by local and distributed agent interactions

Fault tolerance: Swarm intelligent processes do not rely on a centralized control mechanism. Therefore the loss of few nodes or links does not result in catastrophic failure, but rather leads to graceful, scalable degradation.

Adaptation: Agents can change, die or reproduce, according to network changes.

Speed: Changes in the network can be propagated very fast, in contrast with the Bellman-Ford algorithm

Modularity: Agents act independently of other network layers.

Autonomy: Little or no human supervision is required.

Parallelism: Agent's operations are inherently parallel.

These properties make swarm intelligence very attractive for ad-hoc wireless networks. They also render swarm intelligence suitable for a variety of other applications, apart from routing, including robotics and optimization.

4. ANTNET ROUTING IN MANETS

There are a number of developed swarm-based routing algorithms. The most celebrated one is AntNet, an adaptive agent-based routing algorithm that has outperformed the best-known routing algorithms on several packet-switched communications networks. The authors develop an extension of the AntNet algorithm with QoS guarantees, imposing certain restrictions on bandwidth and hop-count. In the AntNet algorithm, routing is determined by means of very complex interactions of forward and backward network exploration agents ("ants"). The idea behind this sub division of agents is to allow the backward ants to utilize the useful information gathered by the forward ants on their trip from source to destination. Based on this principle, no node routing updates are performed by the forward ants.

Their only purpose in life is to report network delay conditions to the backward ants, in the form of trip times between each network node. The backward ants inherit this raw data and use it to update the routing table of the nodes. An example of an AntNet routing table is in Table. The entries of the routing table are probabilities, and as such, must sum to 1 for each row of the network[4]. These probabilities serve a dual purpose:

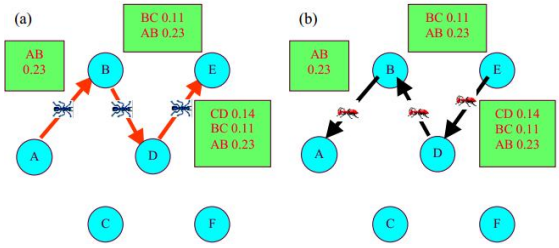
Table 1: Ant Net Routing table

Destination	Next Hop		
		B	**C**
	E	0.15	0.85
	F	0.75	0.25

(1) The exploration agents of the network use them to decide the next hop to a destination, randomly selecting among all candidates based on the routing table probabilities for a specific destination

(2) The data packets deterministically select the path with the highest probability for the next hop The sequence of actions in AntNet (see Fig.) is simple and intuitive:

1. Each network node launches forward ants to all destinations in regular time intervals.
2. The ant finds a path to the destination randomly based on the current routing tables.
3. The forward ant creates a stack, pushing in trip times for every node as that node is reached
4. When the destination is reached, the backward ant inherits the stack.
5. The backward ant pops the stack entries and follows the path in reverse.
6. The node tables of each visited node are updated based on the trip times



(a) Forward ant movement (b) Backward ant movement

Figure 1 Ant Movement

The update of the routing table is reminiscent of other actor-critic systems, where the raw information contained in the trip time is processed by the critic and then used to train the actor to manage the system more efficiently (see Fig.)

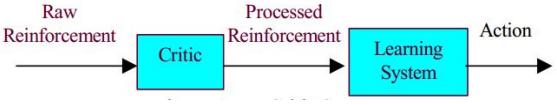


Figure 2 Actor – Critic System

As an intermediate quantity in the processing of the raw trip time information, we need a measure that takes on small values when the trip time is short relative to the mean and vice-versa. Except for the routing table, each node also possesses a table with records of the mean and variance of the trip time to every destination. A typical trip-time table is Table.

Table 2: Routing Table

Destination	Trip Times		
		B	**C**
	E	0.24	0.02
	F	0.18	0.01

The ratio of the variance to the mean, (σ μ), is used as a measure of the consistency of the trip times, and to accordingly alter the effect of the trip time on the routing table. Based on the value of r′, we determine the relative goodness of the trip time of an ant. Corresponding strategies of either decreasing or increasing the value of r' by a certain amount are then followed, based on setting the threshold for the good/bad trip time to 0.5, and selecting a threshold δ for the (σ μ) ratio (see Table)

Table 3: Processing Cases

	$r' < 0.5$	$r' > 0.5$
$\frac{\sigma}{\mu} > \delta$	$r' - a'(1 - e^{-a\frac{\sigma}{\mu}})$	$r' - a'(1 - e^{-a\frac{\sigma}{\mu}})$
$\frac{\sigma}{\mu} < \delta$	$r' - e^{-a\frac{\sigma}{\mu}}$	$r' + e^{-a\frac{\sigma}{\mu}}$

The principle behind these updates is that small values of r′ correspond to small values of T and vice versa. By way of example, and examining the case where the consistency is high and the time is good, we want the processed r′ to be even smaller. Doing so underscores the goodness of this trip time and its consistency. Therefore, an exponential quantity is subtracted. This quantity is the exponentially decaying function of the consistency ratio and achieves its highest value when the variance is very small [4]. The decay rate can be controlled through parameters a′ and a. Further positive or negative reinforcement of good or bad routes takes place next, via negative feedback. Any positive reinforcement of probability should be negatively proportional to current probabilities, and any negative reinforcement should be proportional to current probabilities. The effect of this is to prevent saturation to 0 or 1 of the routing table probabilities. The node that receives the positive reinforcement is the one from which the backward ant comes. This is the same node chosen by the forward ant as next-hop on the way to its destination. All the other neighbors of the current node need to be negatively reinforced to preserve the unit sum of all the next-hop probabilities.

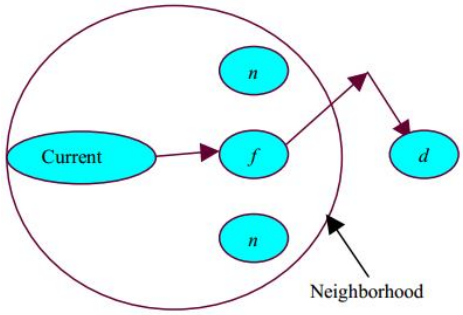


Figure 3 Current Node Neighborhood

The packets of the network then use these probabilities in a deterministic way, choosing as next hop the one with the highest probability [3].

Pseudo code:

Let C1 is the cognitive component C1 ϵ [0,1]
C2 is the social component C2 ϵ [0,1]
W is the inertia W ϵ [0,1]
P is the set of population of the node
D is the bit-string that represents the problem
N is the number of population
M is the number of dimension
Low is the low bound of the area for the node
High is the upper bound of the area for the node
vmax is the maximum change that each node can take during each iteration
i ϵ N ; i = 0,1,2,3,…,N , N = |P|
j ϵ M ; j = 0,1,2,3,…,M , M = |D|
parij is the current position of the node
plij is the previous best solution in the group
pvij is the velocity of the node in the bit-string
plfiti is the best local fitness of the population
pgj is the best position of the dimension
pfiti is the variable to store the fitness value of the node
gfit is the global fitness or the best solution

Main {
1 Initial the Node (parij), Local Best (plij), Node's Velocity (pvij), local fitness
(plfiti), Global Best (pgj), Node Fitness (pfiti) and Global Fitness (gfit) by
calling Sub-function Initialize()
2 While Termination Criterion not met do
3 Update the latest position by calling Sub-function Update()
4 Figure out the best solution by calling Sub-function Calc()
5 End While
}

Initialize () {
1 For each node i do
2 For each node j do
3 Initial the displacement of the node by using the formula that
generate random nodes in the areas provided (parij)
4 Initial the local best node (plij)
5 Initial the velocity of the node (pvij)
6 End for
7 End for
8 For each node i do
9 Initial the fitness value for each node by calling Sub-function Fitness(pari)
10 Initial the local fitness value (plfiti)
11 End for
12 Initial the global fitness (gfit)
13 For each node j do
14 Initial the gbest in that dimension (pgj)
15 End for
} Let xi is the variable to store the current flying position
sum is the variable to store the summation of the fitness function

Fitness (xi) {
1 For each node i do
2 Input the fitness function equation into the variable (sum)
3 End for
4 return the summation (sum)
}

Update () {
1 For each node i do
2 For each node j do
3 Update the place for each node using formula from (2.4.3)
4 End For
5 End For
6 If displacement of node is out of upper bound
7 Set node at the upper bound
8 Else if out of lower bound
9 Set node at the lower bound
10 End if
}

Calculate () {
1 For each node i do
2 Calling function Fitness and stored in node
3 If the fitness value is already min
4 Update to local fitness
5 For each node j do
6 Update the local best solution
7 End For
8 End if
9 End For
10 If local best better than the global one
11 Update to global fitness
12 End if
13 For each node i do
14 For each node j do
15 Update the node velocity by using formula (2.4.6)
16 End for
17 End for

18 Adjust the cognitive and social components using linear equation
}

5. ARTIFICIAL HONEY BEE COLONY

The Artificial Bee Colony algorithm (AHBC) is a population based algorithm that was introduced by Karaboga in 2005. It is inspired by the honey bee behavior of foraging. The algorithm consists or three components:
Employed bees,
Unemployed bees
Food sources.
The employed bees nodes search for the most profitable food source, depending of closeness to the hive and especially of how much food it consists. The algorithm is mostly used as an optimization algorithm, for example its first application was solving problems such as the job scheduling problem or traveling salesman. In nature, the bee foraging activity begins with the scouts [5]. They will scout the area around the hive to find food sources. Scouts can search to a radius of up to three kilometers from the hive. After finding the sources of food, they return to the hive and share the information. They do that by performing a dance (waggle dance). Depending on the length and other properties of the dance, they manage to share information on the food sources, like distance from the hive in coordination with the sun position and how much nectar it holds. From the dance area, other working bees nodes can select a food source to collect nectar from. The hive always has some scouts to keep looking for new food sources [6]. As in nature, the algorithm keeps the main elements. As developed in there are three kinds of bee nodes: workers, onlookers and scouts. Worker bee nodes are the ones who visit a food source that it was previously visited by itself, the onlookers are the ones who check the dance area to decide the food patch that they will visit and the scouts are the bee nodess that keep searching for new sources around the hive. The algorithm developed uses the half of the hive bees as worker bee nodes. Also every working bee node has its own food patch, so the number of worker bee nodes equals the number of food source nodes. After a working bee has depleted its source, it becomes an onlooker. The main steps of the algorithm developed are as follow:
•Initialize.
• REPEAT.
(a) Place the employed bee nodes on the food sources in the memory;
(b) Place the onlooker bee nodes on the food sources in the memory;
(c) Send the scouts to the search area for discovering new food sources.
• UNTIL (requirements are met).
At the initial stage a random number of food source nodes are selected and the bees determine their nectar amount. This bee nodes return to the hive and perform the dance for the onlookers. Secondly, after they shared the information, every employed bee node visits the food patches they previously visited, as that is in its memory, and choose new food sources in the neighborhood of the one they visit. If an onlooker chooses a food area depending on the information provided by the workers through the dance, it will visit the sources in its neighborhood. If a source is abandoned it will be replaced by a new one that was previously found by the scout bee nodes. In this developed mode, at each iteration the scout bee nodes find at least one new food source node. By food source node we describe an actual possible solution. The number of total food source nodes is actually the total number of possible solutions for the problem. At every iteration we memorize the best solution that was found until some criteria are met or the number of total iterations was reached. There are also other approaches to the algorithm steps, but they keep the main elements. For example in chapter and there is a distinction in the algorithm steps, dividing the steps into forward step and backward step of the bee activity. The algorithm can give a set of possible solutions to a certain problem by memorizing the last few best solutions[5].

Steps involved in Honey bee optimization:

Step 1: Initialization: Spray ne percentage of the populations into the solution space randomly, and then calculate their fitness values, which are called the nectar amounts, where ne represents the ratio of employed bees to the total population. Once these populations are positioned into the solution space, they are called the employed bees
Step 2: Move the Onlookers: Calculate the probability of selecting a food source by the equation (1), select a food source to move to by roulette wheel selection for every onlooker bees and then determine the nectar amounts of them. The movement of the onlookers follows the equation (2).
Step 3: Move the Scouts: If the fitness values of the employed bees do not be improved by a continuous predetermined number of iterations, which is called "$Limit$", those food sources are abandoned, and these employed bees become the scouts. The scouts are moved by the equation (3).
Step 4: Update the Best Food Source Found So Far: Memorize the best fitness value and the position, which are found by the bees.
Step 5: Termination Checking: Check if the amount of the iterations satisfies the termination condition. If the termination condition is satisfied, terminate the program and output the results; otherwise go back to the Step 2.

$$Pi = F(\theta i)/\sum Sk{=}1\ F(\theta k) \quad \text{Eqn (1)}$$

Where θi denotes the position of the $i\ th$ employed bee, S represents the number of employed bees, and Pi is the probability of selecting the $i\ th$ employed bee.

$$xij\ (t+1) = \theta ij + \phi(\theta ij\ (t) - \theta kj\ (t)) \quad \text{Eqn (2)}$$

Where xi denotes the position of the $i\ th$ onlooker bee, t denotes the iteration number, θk is the randomly chosen

employed bee, j represents the dimension of the solution and $\phi(\cdot)$ produces a series of random variable in the range $[-1, 1]$

$$\theta ij = \theta ijmin + r \cdot (\theta ijmax - \theta ijmin) \quad \text{Eqn (3)}$$

Where r is a random number and $r \in [0, 1]$.

Procedure of AHBC:

The ABC consists of four main phases:

Initialization Phase:

The food sources, whose population size is SN, are randomly generated by scout bees. Each food source, represented by xm is an input vector to the optimization problem, xm has D variables and D is the dimension of searching space of the objective function to be optimized.

Employed Bee Phase:

Employed bee flies to a food source and finds a new food source within the neighborhood of the food source. The higher quantity food source is memorized by the employed bees. The food source information stored by employed bee will be shared with onlooker bees

Onlooker Bee Phase:

Onlooker bees calculates the profitability of food sources by observing the waggle dance in the dance area and then select a higher food source randomly. After that onlooker bees carry out randomly search in the neighborhood of food source. The quantity of a food source is evaluated by its profitability and the profitability of all food sources

Scout Phase:

If the profitability of food source cannot be improved and the times of unchanged greater than the predetermined number of trials, which called "limit", the solutions will be abandoned by scout bees. Then, the new solutions are randomly searched by the scout bees

Pseudo Code:

```
1 Begin
2 Initialize the solution population ,
i = 1, . . . , SN
3 Evaluate population
4 cycle = 1
5 Repeat
6 Generate new solutions vmi for the
employed bees using (ii) and evaluate
them.
7 Keep the best solution between current
and candidate
8 Select the visited solution for onlooker
bees by their fitness
9 Generate new solutions vmi for the
Onlooker bees using (ii) and evaluate
them
10 Keep the best solution between current
and candidate
11 Determine if exist an abandoned food
Source and replace it using a scout bee
12 Save in memory the best solution so far
13 cycle = cycle + 1
14 Until cycle = M C N
```

6. Secured Right Angled and Ant Search Hybrid Routing Protocol (SRAAA):

The developed RAAA [6] protocol will go for RREQ/RREP, RREQ/SRREP and SRREQ/SRREP cycle with RABGR and ANT Search method using BNPS and NNPS technique

SRAAA Mechanism:

Our main focuses are to introduce SRAAA [8] to protect data transmission and to construct a secure routing protocol. Our SRAAA approach uses a hybrid of security mechanisms so that it satisfies the main security requirement and guarantees the discovery of a correct and secure route. The security mechanisms that the protocol uses are the hash function, Certificates, time synchronization and route discovery request. SRAAA works as a group and has Four stages, examined in turn in the remainder of this section:

Route Request Process:

1. Route Request message MD5 encryption by Destination
2. Send Encrypted Route Request Message with Symmetric key from Destination with unique ID (MAC Address)
3. Decrypting MD5 by source and check the route request

Detect and Eliminate the Attackers from The Routing table Process:

4. Detecting Attackers in the network by looking up duplicate requests
5. Removing those nodes from the routing table

Certificate Distribution to all the authenticated nodes:

6. Source Generates and distribute the Certificates to all the authenticated nodes

Packet Transfer Process:

7. Sending Packets to the proper destination with SES data encryption technique
8. Receiving packets and SES decryption by destination

Route Request Process:

The MD5 hash function is used to encrypt and update the Request data necessary for the routing process in order to secure the mutable data request, which in this case is the head direction and time to find a proper unique destination, whose information uses hash chains. SRAAA uses hash chains in order to secure the mutable data request of the head direction and Td, the maximum time to find a destination node, for any node in the network, including an intermediate node and the destination node, which when it receives the message can verify that the mutable data request has not been decremented by any attacker. SRAAA forms a hash chain by applying it one way. A hash function is the operation whereby a node creates an RREQ and a hash function repeatedly to begin. The MD5 Hash function functionality is explained briefly in next headings. Using these Request packets is being encrypted and sending to the source. The source node will have the symmetric key to decrypt this message to read the proper request message.

Detect and Eliminate the Attackers from The Routing table Process:

Using the above process source node could easily find the proper destination. And it could easily find the attackers by receiving duplicate requests. It would be the strongest way to find the attackers and eliminate from the network by removing these from routing table.

Certificate Distribution to all the authenticated nodes:

RAAA adopts the Source node generate Certificate approach because of its superiority in distributing keys and achieving integrity and non-repudiation. The system uses symmetric and public keys. The symmetric key is used to sign the certificate and the public key of all the nodes, while the public key is used to renew certificates that are issued by source node. All nodes must to have verified certificates. The public keys and the corresponding symmetric keys of all nodes are created by the source node, which also issue the public-key certificates of all nodes. Each node has its own public/Symmetric key pair. Public keys can be distributed to another node in the secure path stage, while Symmetric keys should be kept confidential to individual nodes. Each node in SRAAA approach receives exactly one certificate after securely authenticating its identity to the Source. Each node will hold its certificate in the Node Databases. The main structure of node certificates, it contains the identifier of the node, its public key, the name of the source issuing this certificate, the certificate issue and expiry dates, and the public key of the node. Finally, the contents of the certificate will be attached to the signature of the source node. All nodes in a network should maintain fresh certificates with the source node. At the secure path stage, nodes use their certificates to authenticate themselves to other nodes in the network.

Packet Transfer Process:

SRAAA approach is to use a SES algorithm to establish secure data between nodes. The Secure Path Stage is found in the first process and is based on the requirement for all nodes to have a secure path with other nodes before sending any route request packet. Any node receiving an RREQ from the source node or another node without a secure path should discard the request. In our approach, each node is given the system public key in order for any node to be able to send a Secure Path Request to another node the first time the certified public keys are exchanged. The authenticity of the certificate can be confirmed as the nodes have the system public key. The first objective of the SPS is the exchange of the certified public keys and their confirmation, while its second objective is to ensure the identity of the sender before acceptance of the RREQ. The SPS considers secure authentication node by node.

```
Algorithm: Pseudo-code for SRAAA mechanism
Source Broadcasts RREQ packet
Destination and Attackers Send RREP
If RREP packet received then
Decrepit the Reply message with MD5 and Symmetric key
and Mac Address
Find the Attackers
Eliminate from Routing Table
Find Multi path to the destination with RAAA
Sends data packets to destination in smart route
```

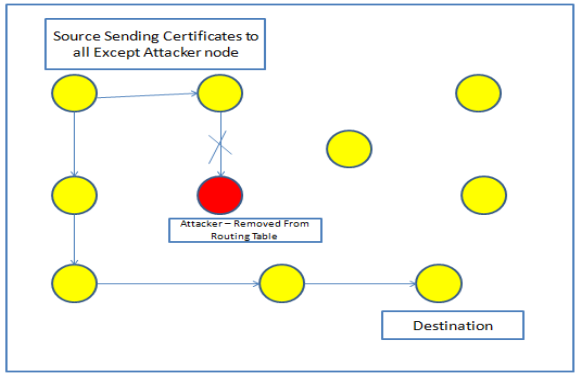


Figure 4 SRAAA Mechanism

7. QoS METRICS AND COMPARISON

Network Simulator supports different parameters for the measurement performance evaluation of the WSN under different routing protocols. The metrics used are Bit Error Rate (BER) vs. Signal-to-Noise Ratio (SNR), Average end-to-end delay vs. BER, Packet Delivery Ratio vs. BER, Energy Consumed vs. BER, Network Lifetime vs. Energy Consumption, Throughput vs. BER, and Throughput vs. SNR. The table of the metrics with their details, used to evaluate the performance of the routing protocols is given below in Table

Table 4: QoS Metrics

Sl.No	QoS Metrics	Definition
1	Bit Error Rate (BER) (measured in %age)	BER is number of bit errors divided by total number of transferred bits in a specified time interval
2	Packet Delivery Ratio (expressed in number of packets)	It is the time taken by a data packet to be transmitted across a network from source to destination
3	Packet Delivery Ratio (expressed in number of packets)	It is the ratio of total number of delivered packets successfully received by the sink node to the number of packets sent by all sensor nodes in the network
4	Energy Consumed (measured in KJ)	It is a measure of rate at which energy is dissipated by sensor nodes in a WSN within a specific time period
5	Network Lifetime (measured in minutes)	The lifetime of a WSN can be defined as the time elapsed until the first node dies, the last node dies, or a fraction of nodes dies
6	Average Network Throughput (measured in bps i.e. bits per seconds)	It is the average number of data packets successfully received by the sink node per unit time

Table 5: QoS Comparision

Metrics	SRAAA	Ant-Net	Honey Bee
BER (At 0 db SNR)	0.01046 %	0.11%	0.05805 %
Delay (At 0% BER)	0.5824 seconds	0.6165 seconds	0.5866 seconds
Delivery Ratio (At 50% BER)	5 packets	3 packets	4 packets
Energy Consumed (At 20% BER)	12.88 KJ	20.89 KJ	25.11 KJ
Network Lifetime (At 20 KJ Energy Consumption)	61600 minutes	37510 minutes	53000 minutes
Throughput (At 1% BER)	1104*10 ^3 bps	104*10^ 3 bps	156*10^ 3 bps
Throughput (At 8 db SNR)	25*10^4 bps	2*10^4 bps	2*10^3 bps

Simulation Results:

The below figure shows the graph of SRAAA with ANTNET and HONEYBEE protocols. The X-axis of the graph indicates the no of nodes and the Y-axis shows the throughput. As we can clearly observe from the graph, the throughput of SRAAA is very much better when compared with Honeybee and Ant-net. Thus in case of Throughput, SRAAA performs significantly well when compared to Honeybee and Ant-net.

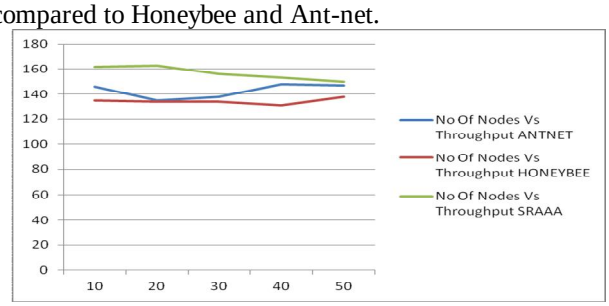


Figure 5 Number of Nodes Vs Throughput

In this below End to end delay metric, The received packets are much more in case of SRAAA than others. The packets dropped in SRAAA are very less (i.e. can be neglected) when compared with honeybee and Ant-Net. Thus, the Average end to end delay is more in SRAAA when compared to Ant-Net and Honeybee.

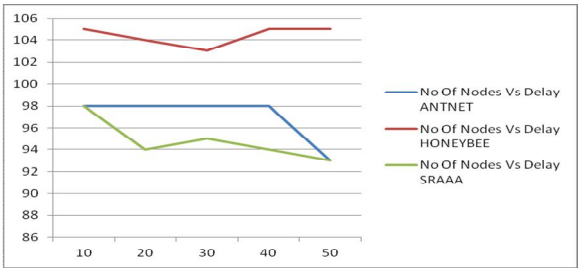


Figure 6 Number of Nodes Vs Delay

Figure shows the comparison chart of packet delivery probability for Direct Delivery Routing, Epidemic Routing and Spray and Wait Routing. From the chart it can be noticed that when 5 nodes are there at that time packet delivery probability given by all the three routing protocols are almost equal. Whereas in the case when total number of nodes are 10, 20, 30, 40 and 50, the SRAAA shows increment in packet delivery probability So, as the total number of nodes increase the possibilities to meet with the destination node in the Direct Delivery routing increases.

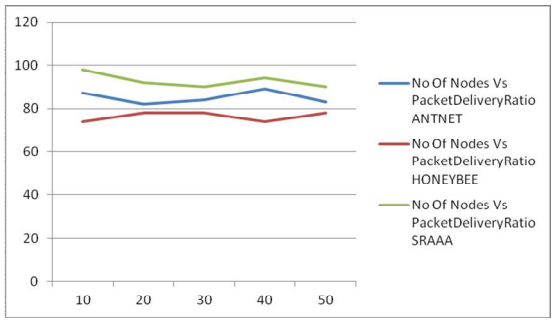


Figure 7 Number of Nodes Vs Packet Delivery Ratio

Data rate defined as that number of packet sends per second. Here we are plotting the graph between Throughput and data rate. Data rate of a connection is a measure of how data bits can travel from one end to the other in a measurable amount of nodes. Throughput of a connection is a measure of how much information data can be moved from one end to the other in a measurable amount of time. The Graph Shown in figure comprises the results of Throughput with Data rate, taking data rate along X-axis and Throughput SRAAA improves 8.87% of the Throughput in terms of Data rate when compared with all other protocols.

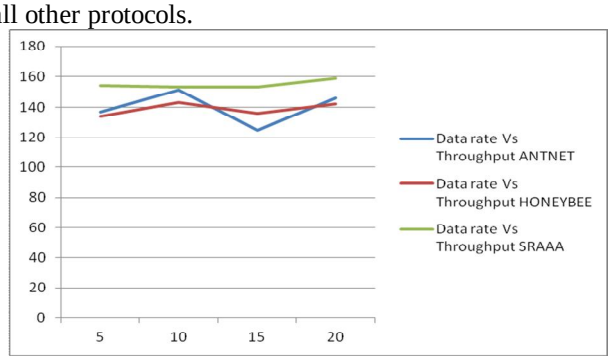


Figure 8 Data Rate Vs Throughput

The Graph Shown in figure comprises the results of Delay with Data rate, taking data rate along X-axis and Delay along Y-axis, we could conclude that our SRAAA protocol has low delay at low data rate as well as at high data rate when compare to all other protocols. SRAAA improves 14.22% of the Delay in terms of Data rate when compared with all other protocols.

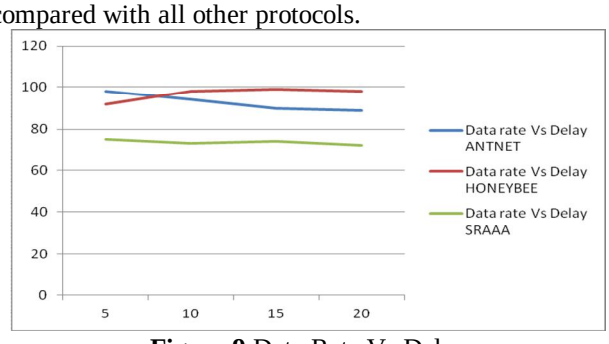


Figure 9 Data Rate Vs Delay

The Graph Shown in figure comprises the results of Packet Delivery Ratio with Data rate, taking data rate

along X-axis and PDR along Y-axis. This graph indicates that at low data rate the PDR of SRAAA is going to increase, but overall RAAA produce better Packet Delivery ratio as compare with others protocols. SRAAA improves 15.32% of the Packet delivery ratio in terms of Data rate when compared with all other protocols.

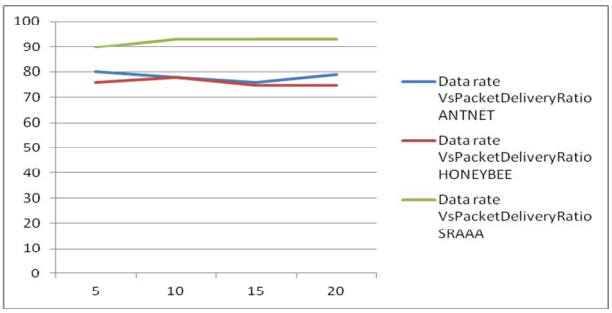


Figure 10 Data Rate Vs Packet Delivery Ratio

8. CONCLUSION

This paper concludes the results in two scenarios where number of nodes varies in these scenarios to observe the variation in the performance of three routing protocols Honey Bee, AntNet, and SRAAA of Quality of Service (QoS), deployed over MANETs to analyze their behavior with respect to QoS metrics. The study of these Wireless Sensor routing protocols shows that the SRAAA protocol is the best protocol as per the simulation results of this chapter because it shows almost the best performance for all the metrics. SRAAA shows the lowest BER as the SNR increases, the lowest Average End-to-End Delay as the BER increases, the highest Packet Delivery Ratio as the BER increases, the lowest Energy Consumption as the BER increases, the highest Network Lifetime as the Energy Consumption increases, the highest Throughput as the BER increases and again the highest Throughput as the SNR increases among the other two comparing routing protocols, in both the scenarios. And the REAR gives the worst performance for almost all the metrics against which the comparison has done, except against the Energy Consumption as the BER increases and the throughput as the SNR increases. And the Swarm Intelligence protocol gives the moderate results for almost all the metrics, but shows the worst results against the Energy Consumption as the BER increases and the Throughput as the SNR increases. So, it is clear that all the routing protocols show the better performance in the network having less number of sensor nodes as compared to the network having more number of sensor nodes. But, the SRAAA protocol also outperforms against each QoS metrics on comparing the results of the both the scenarios among the other two protocols.

References

[1] Deepak Verma, Amardeep Kaur "Performance Comparison of Qos Based Routing Protocols Mbrr, Rear and Speed for Wireless Sensor Networks", Internet Draft, 2013.

[2] Manisha Mehra, Sangeeta "A survey on Swarm Intelligence Techniques", Internet Draft, 2014.

[3] Krzysztof walkowiak "Ant Algorithm for Flow Assignment in Connection-Oriented Networks", internet draft, 2005.

[4] Kassabalidis, M.A. El-Sharkawi, R.J.Marks II, P. Arabshahi, A.A. Gray "Swarm Intelligence for Routing in Communication Networks", Internet Draft, 2005.

[5] D. Karaboga, B. Basturk, "On the performance of artificial bee colony (ABC) algorithm", Internet Draft, 2007.

[6] Sandeep Kumar, Vivek Kumar Sharma, Rajani Kumar "A Novel Hybrid Crossover based Artificial Bee Colony Algorithm for Optimization Problem", Internet Draft, 2013.

[7] Capt. Dr. S. Santhosh Baboo, V J Chakravarthy "New Framework for Routing in MANET Right Angled and Ant Search Protocol **(RAAA)",** International Journal of Engineering Sciences & Research Technology, July 2013.

[8] V J Chakravarthy, Capt. Dr. S. Santhosh Baboo "SRAAA – Secured Right Angled And Ant Search Hybrid Routing Protocol for MANETs", ijiris journal, volume 1, issue 4, october 2014.