A Fault Tolerant Framework to Detect Routing Failures in Air Pollution Monitoring MANET Using 2ACK

B.Kameswara Rao[1], A.S.N.Chakravarthy[2]

Abstract — This paper proposes routing fault detection in MANETs using 2ACK scheme. Routing protocols for MANETs are designed based on the assumption that all participating nodes are fully cooperative. However, due to the open structure and scarcely available battery-based energy, node fault may exist. In the existing system, there is a possibility that when a sender chooses an intermediate link to send some message to a destination, the intermediate link may pose problems such as, the intermediate node may not forward the packets to destination, it may take very long time to send packets or it may modify the contents of the packet. In MANETs, as there is no retransmission of packets once it is sent, care must be taken not to lose packets. We have analysed and evaluated a technique, termed 2ACK scheme to detect and mitigate the effect of such routing fault in MANETs environment. It is based on a simple 2-hop acknowledgment packet that is sent back by the receiver of the next-hop link. 2ACK transmission takes place for only a fraction of data packets, but not for all. Such a selective acknowledgment is intended to reduce the additional routing overhead caused by the 2ACK scheme. Our contribution in this paper is that, we have embedded some security aspects with 2ACK to check confidentiality of the message by verifying the original hash code with the hash code generated at the destination. If 2ACK is not received within the wait time or the hash code of the message is changed then the node to next hop link of sender is declared as the misbehaving link. We simulated the routing fault detection using 2ACK scheme to test the operation scheme in terms of performance parameters.

Keywords— 2ACK, MANETs, routing fault, selfish node

I. Introduction

A mobile ad hoc network (MANET) is a collection of mobile nodes (hosts) which communicate with each other via wireless links either directly or relying on other nodes as routers. The operation of MANETs does not depend on pre-existing infrastructure or base stations. Network nodes in MANETs are free to move randomly. Therefore, the network topology of a MANETs may change rapidly and unpredictably. All network activities such as discovering the topology and delivering data packets have to be executed by the nodes themselves either individually or collectively. Depending on its application, the structure of a MANET may vary from a small, static network that is highly power-constrained to a large-scale, mobile, highly dynamic network.

There are two types of MANETs: closed and open [1]. In a closed MANET, all mobile nodes cooperate with each other towards a common goal, such as emergency search/rescue or military and law enforcement operations. In an open MANET, different mobile nodes with different goals share their resources in order to ensure global connectivity. However, some resources are consumed quickly as the nodes participate in the network functions. For in-stance, battery power is considered to be most important in a mobile environment. An individual mobile node may attempt to benefit from other nodes, but refuse to share its own resources. Such nodes are called selfish nodes or misbehaving nodes and their behaviour is termed as selfish-ness or fault. One of the major sources of energy consumption in the mobile nodes of MANETs is wireless transmission. A selfish node may refuse to forward data packets for other nodes in order to conserve its own energy [2], [3].

In MANETs, routing fault can severely degrade the performance at the routing layer. Specifically, nodes may participate in the route discovery and maintenance processes but refuse to forward data packets. How do we detect such fault? How to make such detection process more efficient and accurate. We analysed the 2ACK technique [4] to detect such misbehaving nodes or links. Routes containing such nodes will be eliminated from consideration. The source node will be able to choose an appropriate route to send its data. The 2ACK scheme is a network-layer technique to detect misbehaving links and to mitigate their effects. The 2ACK scheme detects fault through the use of a new type of acknowledgment packet, termed 2ACK. A 2ACK packet is assigned a fixed route of two hops (three nodes) in the opposite direction of the data traffic route. In this work, we provide security features to 2ACK, where confidentiality of the message is checked by verifying the original hash code with the hash code generated at the destination.

The rest of the paper is organized as follows. Section 2 discusses related work in this area. Section 3 describes the proposed work. Section 4 presents the simulation procedure, performance parameters and the results of the pro-posed work. Finally, we conclude in Section 5.

II. Outcome of the Parallel Research Works

The security problem and the fault problem of wire-less networks including MANETs have been studied by many researchers. Various techniques have been proposed to prevent selfishness in MANETs. Some of the related works are as follows.

The work given in [5] explains detection of malicious nodes by the destination node, isolation of malicious nodes by discarding the path and prevention data packets by using dispersion techniques.

The work given in [4] describes the performance degradation caused by selfish (misbehaving) nodes in MANETs. They have proposed and evaluated a technique, termed 2ACK, to detect and mitigate the effect of such routing fault.

The work given in [6] presents cooperative, distributed intrusion detection architecture for MANETs that is intended to address some challenges. The architecture is organized as a dynamic hierarchy in which data acquisition occurs at the leaves, with intrusion detection data being incrementally aggregated, reduced, analysed, and correlated as it flows upward towards the root.

The work given in [7] explains the problem of identification of misbehaving nodes and refusing to forward packets to a destination. They have proposed a reactive identification mechanism that does not rely on continuous overhearing or intensive acknowledgment techniques, but is only activated in the event of performance degradation.

The work given in [8] proposes a general solution to packet dropping fault in mobile ad hoc networks. The solution allows monitoring, detecting, and isolating the droppers.

The work given in [9] proposes signal strength based routing for wireless ad hoc networks. It uses signal strengths on the multi hop to identify stable route from source to destination in an ad hoc networks. A stable route helps to reduce control packets overhead during route maintenance and avoids route interruptions. Some of the related work is given [10], [11], [12].

III. Proposed Novel Framework

The proposed system is used to detect the fault routing using 2ACK and also check the confidentiality of the data message in MANETs environment. Here, we used a scheme called 2ACK scheme, where the destination node of the next hop link will send back a 2 hop acknowledgement called 2ACK to indicate that the data packet has been received successfully. The proposed work (2ACK with confidentiality) is as follows.

- If the 2ACK time is less than the wait time and the original message contents are not altered at the inter-mediate node then, a message is given to sender that the link is working properly.
- If the 2ACK time is more than the wait time and the original message contents are not altered at the intermediate node, then a message is given to sender that the link is misbehaving.
- If the 2ACK time is more than the wait time and the original message contents are altered at the intermediate node, then message is given to sender that the link is misbehaving and confidentiality is lost.
- If the 2ACK time is less than the wait time and the original message contents are altered at the intermediate node then, a message is given to sender that the link is working properly and confidentiality is lost.

At destination, a hash code will be generated and compared with the sender's hash code to check the confidentiality of message. Hence, if the link is misbehaving, sender to transmit messages will not use it in future and loss of packets can be avoided.

This section presents system model, and functioning scheme.

A. System Model

In the existing system, there is a possibility that when a sender chooses an intermediate link to send some message to destination, the intermediate link may give problems such as the intermediate node may not forward the packets to destination, it may take very long time to send packets or it may modify the contents of the packet. In MANETs, as there is no retransmission of packets once it is sent, hence care is to be taken that packets are not lost.

Noting that a misbehaving node can either be the sender or the receiver of the next-hop link, we have focused on the problem of detecting misbehaving links instead of misbehaving nodes using 2ACK scheme. In the next-hop link, a misbehaving sender or a misbehaving receiver has a similar adverse effect on the data packet. It will not be forwarded further. The result is that this link will be tagged. Our approach is used to discuss the significantly simplification of the routing detection mechanism and also checking the confidentiality of the message in MANETs environment.

Figure1 shows the system model of the proposed work. The various modules in the system model are as follows..

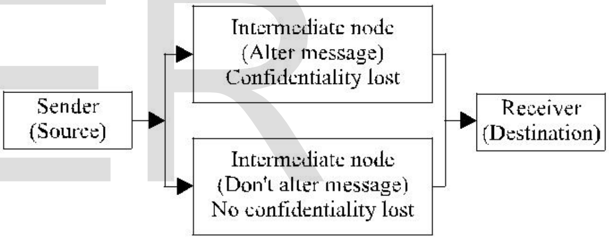

Fig. 1.System model.

- **Module 1**: Sender module (Source node). The task of this module is to read the message and then divide the message into packets of 48 bytes in length, send the packet to receiver through the intermediate node and receive acknowledgement from the receiver node through the intermediate node. After sending every packet the "Cpkts" counter is incremented by 1. 2ACK time is compared with the wait time. If 2ACK is less than wait time, "Cmiss" counter is incremented by 1. The ratio of "Cmiss" to "Cpkts" is com-pared with the "Rmiss" (a threshold ratio). If it is less than "Rmiss", link is working properly otherwise misbehaving.

- **Module 2**: Intermediate module (Intermediate node). The task of this module is to receive packet from sender, alter/don't alter the message and send it to destination. Get 2ACK packet from the receiver and send 2ACK packet to sender.

- **Module 3**: Receiver module (Destination node). The task of this module is to receive message from the intermediate node, take out destination name and hash code and decode it. Compare the hash code of source node and destination node for security purpose. Send 2ACK to source through the intermediate node

B. Functioning of Scheme

- **Algorithm of 2ACK Scheme**

We have used the triplet of N1 → N2 → N3 as an example to illustrate 2ACK's pseudo code. Where N1 is assumed as the source node, N2 is the intermediate node and N3 is the destination node. Note that such codes run on each of the sender/receiver of the 2ACK packets.

Nomenclature:{**Cpkts**= the number of the message pack-ets sent, **Cmiss** = the number of the 2ACK packets missed, **d** = the acknowledgement ratio. **WT**= waiting time, i.e., the maximum time allotted to receive 2ACK packet}

```
At node N1
while(true) do
    • Read the destination address;
    • Read the message;
    • Find the length of the message.
      Cmiss=0, Cpkts=0, WT=20 ms, d=0.2,

      2ACK Time=Current Time (Acknowledgement ac-
      cepted time) – Start Time.

      while(length>48 bytes) do Take
          out 48 message packet;
          Length = length – 48;

          Encode message using hash function;
          Send message along with the hash key;
          Cpkts++ ;
          Receive 2ACK packet;

          if(2ACK time>WT) then
              Cmiss++ ;
          end
      end

      if(length<48 bytes) then

          Encode message using hash function;
          Send message along with the hash key;
          Cpkts++;
          Receive 2ACK packet;

          if(2ACK time>WT) then
              Cmiss++;
          end
      end
end

At node N2
while(true) do
  Read message from source N1 if
  (Alter) then

      Add dummy bytes of characters;
      Process it and forward to destination N3;
      Receive 2ACK from N3 and send it to N1;

  else if (Do not Alter) then

      Process it and forward to destination N3;
      Receive 2ACK from N3 and send it to N1;

  end

end

At node N3
while(true) do
  Read message from N2;

  Take out destination name and hash code;
  Decode the message;

  Send 2ACK packet to N2;

end

At N1 and N3 parallel

while(true) do

  if((Cmiss/Cpkts)>d and (hash code of source msg) !=
  (hash code of destination msg)) then

      Link is misbehaving and the confidentiality is
      lost;

  end

  if((Cmiss/Cpkts)<d and (hash code of source msg) !=
  (hash code of destination msg)) then

      Link is working properly and the confidentiality is
      lost;

  end

  if((Cmiss/Cpkts)>d and (hash code of source msg)=
  (hash code of destination msg)) then

      Link is misbehaving;

  end

  if((Cmiss/Cpkts)<d and (hash code of source msg)=
  (hash code of destination msg)) then

      Link is working properly;

  end

end
```

IV. Results and Discussion

We conducted simulation of the proposed scheme by using C programming language. The proposed scheme has been simulated in various network scenarios. Simulations are carried out extensively with random number for 100 iterations. This section presents the simulation model, simulation procedure and results and discussions.

4.1. Simulation Model

Our simulation model consists of *N* number of nodes. The nodes are selected randomly in MANETs environment. The first node is always assumed as the source node and the last node is assumed as the destination node. Remaining nodes are assumed as the intermediate nodes (e.g., *N* = 70 nodes, in that first, i.e., N1 is assumed as source node and last, i.e., N70 is assumed as the destination node and N2 to N69 are assumed as the intermediate nodes). We have used some of the functions in our simulation model.

- **Pm** – the fraction of nodes that are misbehaving. The misbehaving nodes are selected among all network nodes randomly;

- **Rmiss**– the threshold to determine the allowable ratio of the total number of 2ACK packets missed to the total number of data packets sent;

- **R2ack** – the acknowledgement ratio, the fraction ofdata packets that are acknowledged with 2ACK pack-ets (maintained at the 2ACK sender).

4.2. Simulation Procedure

To illustrate some of the results of simulation, we have considered the following environment variables as follows: N = 10 to 90 for different cases, Pm = 0, 0.1, 0.2, 0.3, 0.4, WT = 20 ms and R2ack = 0.05, 0.2, 0.5, and 1.

Begin

1) Randomly generate number of nodes N.

2) Compute the acknowledgement time in the absence of misbehaving nodes.

3) Compute for the selected parameter for different values of Pm ranging from 0 to 0.4 and find the number of misbehaving nodes.

4) Wait for some delay and the compute the same parameter for different R2ack values ranging from 0.05 to 1.

5) Apply the proposed scheme.

6) Compute the performance parameters.

7) Generate the graphs.

End

4.3. Performance Parameters

We have used the following parameters to measure the performance of the 2ACK scheme in MANETs.

- **Packet delivery ratio (PDR)** – the ratio of the number of packets received at the destination and the number of packets sent by the source.

- **Routing overhead (RO)** – the ratio of the amount of routing related transmissions (such as fault report, 2ACK etc) to the amount of data transmissions. The amount is in bytes. Both forwarded and transmitted packets are counted.

- **2ACK time** – it measures the time required to receive the 2ACK packet from destination node to source node during the absence of misbehaving nodes.

- **2ACK time1** – it measures the time required to receive the 2ACK packet from destination node to source node during the presence of some misbehaving nodes.

- **Throughput** – it measures the overall performance of the 2ACK scheme with respect to the fault ratio.

4.4. Results and Discussion

Figure 2 shows the packet delivery ratio versus fault ratio. The varied Pm from 0 (all of the nodes are well behaved) to 0.4 (40% of the nodes are misbehave). We have observed that most packets were delivered when Pm = 0 (no misbehaving nodes). The packet delivery ratio decreases as Pm increases. The 2ACK scheme delivered over 90% of the data packets even when Pm = 0.4. The acknowledgment ratio R2ack was set to 0.05, 0.2, 0.5 and 1 respectively. We can see R2ack does not appreciably affect the PDR performance of the 2ACK scheme.

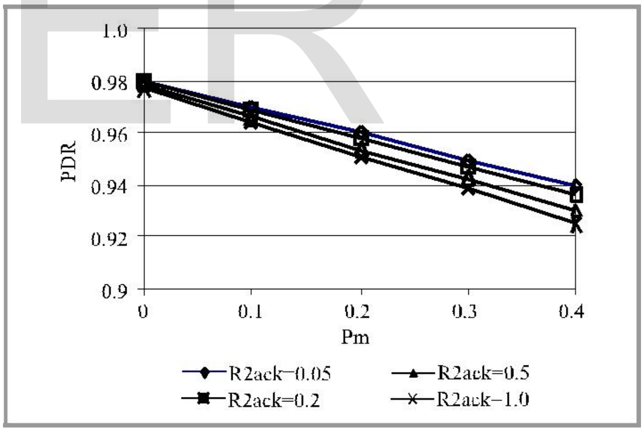

Fig. 2.Packet delivery ratio (PDR) versus fault ratio (Pm).

Figure 3 shows the routing overhead (RO) of the 2ACK scheme with different acknowledgment ratios, R2ack. We varied Pm from 0 (all of the nodes are well behaved) to 0.4 (40% of the nodes are misbehave). Here, we com-pare routing overhead of the 2ACK scheme with different R2ack values. Overhead of the 2ACK scheme is highest when R2ack = 1. This is due to the large number of the 2ACK packets transmitted in the network.

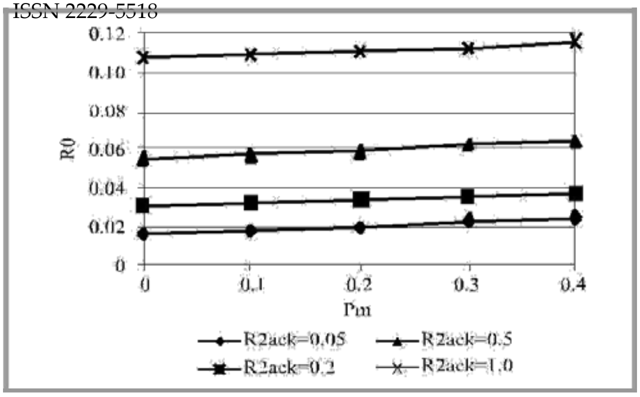

Fig. 3. Routing overhead (RO) versus fault ratio (Pm).

As the value of R2ack decreases, the routing overhead reduces dramatically. Therefore, R2ack in the 2ACK scheme provides an effective “knob” to tune the routing overhead.

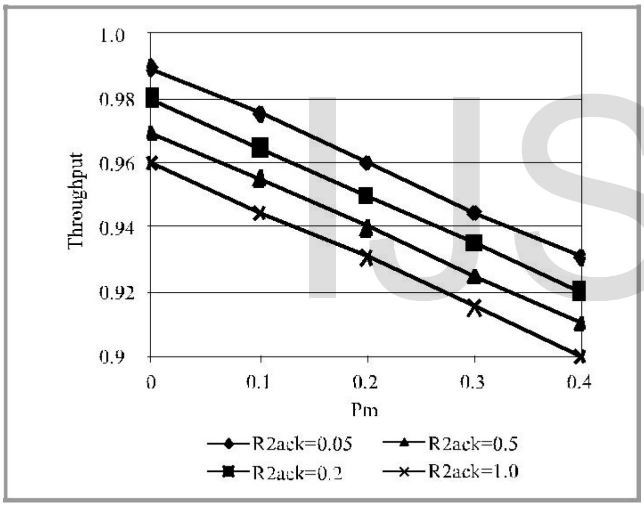

Fig. 4.Throughput versus fault ratio (Pm).

Figure 4 shows the relative throughput of the 2ACK scheme with different acknowledgment ratios, R2ack. We varied Pm from 0 (all of the nodes are well behaved) to 0.4 (40% of the nodes are misbehave). Here, we compare throughput of the 2ACK scheme with different R2ack values as well as with the different fault ratios values. Throughput will be high when the fault ratio is 0

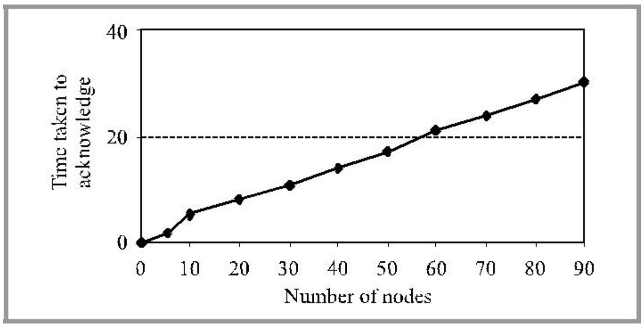

Fig. 5.Number of nodes versus time taken to acknowledge.

Routing Fault Detection in MANETs Using 2ACK (no misbehaving nodes) and R2ack is 0.05 (5 2ACK has to be sent for every 100 packets). The throughput decreases as Pm increases or R2ack increases. For instance, when Pm = 0.4 and R2ack = 1, the 2ACK scheme is able to sup-port a relative throughput of 90%.

Figure 5 shows the number of the nodes increases, the 2ACK time will also increases in MANET environment. The number of nodes are randomly selected and wait time is set for 20 ms. The time is calculated for the expected 2ACK packet. If received within 20 ms, it is called a successful 2ACK. If not it called as lost 2ACK.

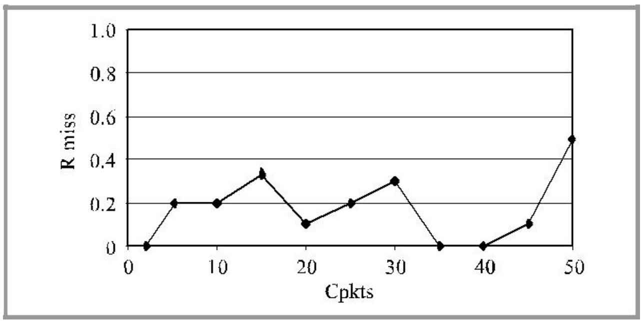

Fig. 6. 2ACK miss ratio (Rmiss) versus number of packets sent.

Figure 6 shows the graph of 2ACK miss ratio (Rmiss) versus number of packets sent (Cpkts). Cmiss depends upon the 2ACK time which varies on the number of misbehaving nodes. Hence, the graph varies drastically.

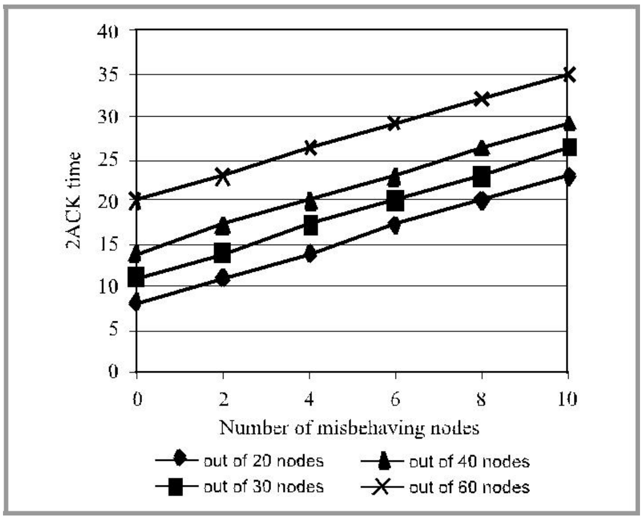

Fig. 7.Number of misbehaving nodes versus 2ACK time.

Figure 7 shows the graph of 2ACK time with respect to the number of misbehaving nodes. As the number of misbehaving nodes increases, the time taken to receive the 2ACK packet will also increases gradually.

V. Conclusion

Mobile ad hoc networks have been an area for active research over the past few years, due to their potentially widespread application in military and civilian communications. Such a network is highly dependent on the co-operation of all its members to perform networking functions. This makes it highly vulnerable to selfish nodes or fault nodes. In this paper, we have investigated the performance degradation caused by such selfish (misbehaving) nodes in MANETs. We have analysed and evaluated a technique, termed 2ACK, to detect and mitigate the effect of such routing fault. Extensive analysis of the 2ACK scheme has been performed to evaluate its performance. We have embedded some security aspects with 2ACK to check confidentiality of the message by verifying the original hash code with the hash code generated at the destination. Our simulation results show that the 2ACK scheme maintains up to 91% packet delivery ratio even when there are 40% misbehaving nodes in the MANETs that we have studied. The regular DSR scheme could only offer a packet delivery ratio of 40%. The false alarm rate and routing overhead of the 2ACK scheme are investigated as well. One advantage of the 2ACK scheme is its flexibility to control overhead with the use of the R2ack parameter.

References

[1] E. Lorenzini, “Cooperation”, in Proc. Sust. Coop. Multi-Hop Wirel. Netw., 2007 [Online]. Available: http://www.research.microsoft.com/enus/um/people/ratul/.../ nsdi2005-catch.pdf

[2] G. F. Mariasy, P. Georgiadis, D. Flitzanis, and K. Mandalas, “Cooperation enforcement schemes for MANETs: a survey: re-search articles”, in Wirel. Commun. Mobile Comput., vol. 6, iss. 3, pp. 319–332, 2006.

[3] L. Tamilselvan and V. Sankaranarayanan, “Prevention of co-operative black hole attack in MANET”, J. Netw., vol. 3, no. 5, pp. 13–20, 2008.

[4] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, “An acknowledgment-based approach for the detection of routing mis-behavior in MANETs”, IEEE Trans. Mob. Comput., vol. 6, no. 5, pp. 536–550, 2007.

[5] S. Dhanalakshmi and, M. Rajaram, “A reliable and secure frame-work for detection and isolation of malicious nodes in MANET”, Int. J. Comp. Sci. Netw. Secur., vol. 8, no. 10, pp. 184–190, 2008.

[6] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Tal-pade, C. Ko, R. Balupari, C.-Y. Tseng, T. Bowen, K. Levitt, and J. Rowe, “A general cooperative intrusion detection architecture for MANETs”, in Proc. 3rd IEEE Int. Inform. Assur. Worksh., College Park, USA, 2005, pp. 57–70.

[7] W. Kozma Jr. and L. Lazos, “Reactive identification of fault in ad hoc networks based on random audits”, 2008 [Online]. Avail-able: http://www.ieeexplore.ieee.org/iel5/4557722/4557723/ 04557810.pdf

[8] D. Djenouri, O. Mahmoudi, M. Bouamama, D. Llewellyn-Jones, and M. Merabti, “On securing MANET routing protocol against control packet dropping” [Online]. Available: http://www.cscjournals.org/ Journals/IJCSS/Volume2/Issue1/IJCSS-24.pdf

[9] S. R. Biradar, K. Sharma, S. K. Sarkar, and Puttamadappa C., “Signal strengths based stable route for wireless ad-hoc networks” in Proc. Int. Worksh. Conf. Stat. Phys. Appr. Multi-Discip. Probl., Guwahati, India, 2008 [Online]. Available: http://www.iitg.ac.in/ statphys/files/abs cn 05.pdf

[10] K.-W. Chin, J. Judge, A. Williams, and R. Kermode, “Implementa-tion experience with MANET routing protocols”, ACM SIGCOMM Comp. Commun. Rev., vol. 32, no. 5, pp. 49–59, 2002.

[11] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Sri-vastava, “Coverage problems in wireless ad-hoc sensor networks” [Online]. Available: http://www.ece.rice.edu/~ 1/papers/Infocom coverage 01.pdf

[12] K. Singh, A. Nedos, and S. Clarke, “Distributed computing and networking”, in Proc. 8th Int. Conf. ICDCN 2006, Guwahati, India, 2006, Berlin Heidelberg: Springer [Online]. Available: http://www.springerlink.com/index/82r23149410vr703.pdf.

Author Profile

B.KameswaraRao, Research scholar from JNTUK, Kakinada from the department of computer science and engineering. He is working as faculty member in department of computer science and Engineering from VizagInstitue of Technology, India. He is a member of ISTE. He published papers in various National /International journals. His areas of interest Web Technology Applications, Wireless Sensor Networks, computer Networks, Spatial Databases, Computer Applications in Geo Science.

Dr. A. S .N. Chakravarthy, Currently is working as Professor in Dept. of Computer Science and Engineering in JNTUK UCEV . He received Ph.D. in Computer Science & Engineering from AcharyaNagarjuna University, Guntur, India in 2011. He is having 14 years of teaching experiance .He has published papers in various National / International journals and conferences. He is a member of IEEE, IEI, ISTE, IETE,CSI, ISCA and reviewer for various International Journals. His research areas include Data Security,Cyber Security, Cloud Privacy and Digital Forensics