Clustering Based on Trust of a Node in Mobile Ad–Hoc NETworks

Pallavi Khatri[1,*], Shashikala Tapaswi[2], and Udai P. Verma[3]

[1] ITM University, Gwalior, India
[2] ABV – IIITM, Gwalior
[3] JU, Gwalior
pallavi.khatri.csit@itmuniverse.in

Abstract. The nodes in Mobile ad hoc networks join and leave the networks dynamically. At some point of time there is a possibility of enormous increase in the size of the network. Handling nodes in big network may put a burden on network management schemes and may introduce delays in the network. Dividing big networks in small groups called clusters may prove to be a good solution for handling them in a better and efficient manner. As MANET (Mobile Ad hoc networks) are self organized, the challenge of achieving security is critical. Evolving and managing trust relationships among the nodes in the network are important to carry efficient transmissions. This work proposes a trust based clustering algorithm which forms a cluster of trusted nodes only. Criteria used to select the nodes are the trust value of a node, weight of a node and its residual energy. A trusted cluster gives a better performance in terms of increase in throughput of the network which is well supported by the results produced by this approach.

Keywords: MANET, Security, Trust Value, Cluster.

1 Introduction

As the want for quicker communication with no geographical barriers is arising day by day, it's turning into tough job to deploy infrastructure network all over. All the devices of next generation demand a lot of mobility, less reliability on infrastructure and have reached the majority over the world. Best answer to those demands would be an infrastructure less ad hoc network where the nodes become a part of and leave the network dynamically. Nodes in ad hoc network are susceptible to attacks or could also be easily compromised. To achieve a definite level of security the necessity of recording information of nodes within the network is needed. Once the network is tiny, this task appears to be possible however just in case of huge networks this becomes tough. Increasing number of nodes within the network will be divided in smaller manageable groups referred to as clusters resulting in a hierarchical data structure of the network. Clusters are little groups consisting of a central watching

[*] Corresponding author.

Sabu M. Thampi et al. (Eds.): SSCC 2013, CCIS 377, pp. 150–161, 2013.

node referred to as a Cluster Head *(CH)* and its Cluster Members *(CM)* .The choice of clusters formation is usually arbitrary and does not take security in to account. A malicious node in the network may claim to become a cluster head and may attract the nodes to join its cluster. This will create a compromised cluster and will hamper the network efficiency with attacks and data loss. This work tries to bring in the concept of trust value of a node before choosing it either as a cluster head or a cluster member. This approach relies on trust value *(TVi)*, weight *(Wi)*, which is the number of neighboring nodes of the claiming node, and residual energy *(REi)* of a node *i* to elect it as Cluster Head *(CH)*. *CH* then selects the cluster members based on their trust value. Using trust value as a parameter for election of cluster head and clusters members gives better performance than many other approaches already proposed. It gives a new dimension to the formation of trusted clusters. Proposed scheme also uses a key management scheme for generation of a group key *(GK)* generated by a *CH* using its public key and its *ID* (Identity) and is distributed among the cluster members. This *GK* is used for encrypting and decrypting the data during transmission.

Rest of the paper is organized in following subsections. Section 2 discusses the various clustering schemes used in MANET and their limitations. In section 3 the details of the proposed protocol, approach, and notations are given. Simulation environment used for this study is detailed in section 4. The results obtained and discussions along with concluding remarks are done in section 5.

2 Related Work

In recent years lot of work has been done in the area clustered mobile ad hoc networks. Many clustering algorithms have been proposed in the past to efficiently divide the network in to small groups with aim of maintaining the efficiency of the network. In this section the works related to clustering the MANET have been discussed.

2.1 Clustering Schemes in MANET

Clustering using trust as a metric has been a very broad area of research in recent years. This section briefs about few strategies used for clustering in mobile ad hoc networks. Work in [1] proposes an on demand Weight Clustering Algorithm (WCA) which elects a cluster head based on the weight of a node. Security of the network is not taken care in this approach it simply concentrates on cluster formation. Leader election algorithm [2, 3] does not have any mechanism to detect the malicious nature of a node in the network. Scheme in [4] puts forth a Vice Cluster Head in Cluster Based Routing Protocol (VCH – CBRP) which is an extended version of CBRP (Cluster Based Routing Protocol). In this approach when a *CH* becomes idle, a vice node in the network declares itself as a *CH* provided it has a bidirectional link to one or more neighbors. This approach may fail when a node declared as vice moves out of the range of *CH*. Highest degree algorithm proposed in [5] is based on the number of its neighboring nodes which is defined as the degree of a node. A node having highest degree becomes a cluster head. Another *ID* (Identity) based algorithm [6] assigns a unique *ID* to each node and a node with lowest *ID* is chosen as cluster head. In [7] authors propose a clustering scheme based on the real distance between the nodes. This distance is measured on the basis of received signal strength of a message. This approach selects the most stable node but may not work efficiently because there is hardly any node in ad hoc network which is stable. Work in [8] proposed a cluster based trust aware routing protocol which protects the transmitted packets from the malicious node. This approach keeps a check on malicious node and if one is detected it is isolated from the network. Becheler et al. [9] uses a concept of threshold cryptography where a *CA* (Central Authority) is required to distribute the fragments of key to all the nodes in the cluster. This approach may be time consuming and will waste bandwidth of network and energy of a node when a new node tries to join the cluster. Approach in [10] proposes a clustering algorithm based on trust which overcomes the drawbacks of [9] but fails to detail the implementation of firewall in pure ad hoc networks. In [11] author proposed a self organized public key management system for fully self organized ad hoc networks. Each node here maintains a certificate repository before claiming to use the system but the drawback with this approach is that it assumes trust to be transitive which is not always true.

3 Proposed Protocol Details

Existing clustering algorithms discussed in section 2 concentrate on clustering the network by creating small groups but fail to take care of trust among the nodes while electing a *CH*. There are cases when a malicious node advertises itself to take care of it trust of node should be known. Many strategies check for the residual energy of all nodes while forming clusters but it will waste bandwidth of the network. Rather, checking residual energy for only those nodes which claim to be *CH* would be beneficial. Primary goal of this work is to propose and develop a trust based clustering algorithm which will help to enhance the performance of mobile ad hoc networks. Proposed work uses Trust Value, Residual Energy and weight of a node to elect it as a *CH*.

Trust Value of a node *i* can be calculated using equation (1):

$$TVi = f(Traffic\ statistics\ of\ a\ node) \tag{1}$$

Traffic statistics of a node used are number of packets dropped by a node, number of packets forwarded to wrong destination, number of false routing messages generated by a node and number of replay packets generated by a node.

Weight of a node which is the total number of 1 – hop neighbors it has can be evaluated using equation (2):

$$(Wi) = \Sigma\ all\ 1 - hop\ neighbors\ of\ i \tag{2}$$

Each node in the network is equipped with an energy model. Every node consumes energy during every transmission and reception done through it *(E_{loss})*. Energy is also consumed in switching on the transmitter and receiver *(E_{TR})* at a node and handling the overheads of the network *(E_O)*. Every node a continuously monitor its energy and logs it every Δs seconds.

Assuming d as the distance between two nodes, the total energy (E_T) consumed in transmitting a packet of size m to a distance d is:

$$E_T(m,d)=E_{TR}*m+E_{loss}*m*d^2 \tag{3}$$

Energy consumed in receiving a packet is:

$$E_R(m)=E_{TR}*m \tag{4}$$

Combining equation (3) and (4) we get the total energy consumed in one transaction, this is given as:

$$E_{TOTAL}=E_T(m,d)+E_R(m)+E_O \tag{5}$$

Residual energy of a node can be calculated using equation 5 as:

$$RE_{new} = RE_{old} - E_{TOTAL} \tag{6}$$

RE_{old} is the Residual energy of a node till previous transaction. Using equation (1), (2) and (6) a cluster head which is eligible to send a claim() packet can be decided.

$$Eligible\ (CH) = max\ (TVi \cdot Wi \cdot REi) \tag{7}$$

And operation is applied among these parameters to choose the CH.

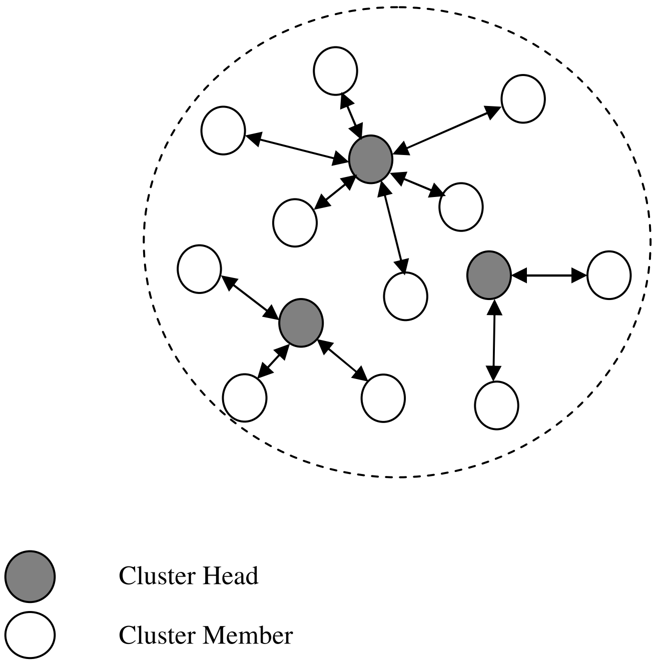


Fig. 1. Clustering in network

Following assumptions are made before implementing the protocol:

1. All nodes in the network maintain the complete information of their 1 – hop neighbors. This is done by periodically broadcasting the hello packets.
2. A classical routing algorithm AODV [12] is running in the network.
3. Every node has a self generated key pair consisting of a secret key *(SK)* and a public key *(PK)*
4. Each cluster consists of a Cluster Head *(CH)* and its 1 – hop Cluster Members *(CM).*
5. All the nodes in a cluster once it is formed are trusted.

Figure 1 shows the architecture of a clustered network which is obtained during this work. It consists of Cluster heads and Cluster members and bidirectional links joining them.

Proposed approach uses trust as a basic metric for making a cluster. Trust is the degree of belief that one node has on another. It is assumed that a TMS (Trust Management System) [13] is running at every node. Each node uses this TMS to evaluate its trust value and stores it in its routing table along with the trust values of all its 1 – hop neighbors. Nodes share this information in the network. Trust Management system allows forming three different trust relationships among the nodes in the network and categorizing the nodes on the basis of their trust values as given in Table 1:

Table 1. Trust Relationships

Nodes trust relationship	Trust Value
Fully Trusted (FT)	$TV_{max} = 1$
Partially Trusted (PT)	$TV_{avg} = 0.5$
Not Trusted (NT)	$TV_{min} = 0$

Trust of a node is evaluated based on the traffic statistics of a node. When a new node *y* enters the network, it sends a *join()* massage to its nearest Cluster Head. *CH* checks for its *TV* and if $TV_y >= TV_{threshold}$ it is allowed to be a member of the cluster. At this stage the trust value assigned to *y* is TV_{min} and eventually with every successful transaction done by y this value is incremented.

3.1 Cluster Head Election

Cluster control architecture used in this work is one hop clustering. For formation of a cluster the routing table of all the nodes which has recorded the details of its 1 – hop neighbors are read. Based on this weight of every node *i*, *(Wi)* is calculated which is the total number of 1 – hop neighbors of a node. A node having maximum weight claims to be a cluster head and broadcasts *claim()* to all its neighbors. Every neighbor receiving this claim*()* message will check the *TV* of the member who claims to be a cluster head in their surroundings and their respective routing tables and if *TV* >= $TV_{threshold}$, then the residual energy of the claiming node is evaluated. If this energy is found to be at some satisfactory level then the claiming node is declared as a cluster head.

Algorithm Cluster Head Election

```
N - Total number of nodes in the network
x, n - a node in the network
Wn - weight of a node n
TVn - Trust value of node n
CH - $i^{th}$ cluster head
nnode - neighboring node
TVnnode - trust value of neighboring node
CM - cluster member
RT - Routing table
REx - Residual energy of x

 1. For every node n ϵ N, Calculate $W_n$
 2. For a node x ϵ N
    If $W_x$ = max ($W_1$, $W_2$, .........$W_n$)
    Then node x broadcasts claim() message
 3. For all 1 - hop neighbors receiving claim()
    Check $TV_x$ in RT of nnode of x.
 4.  If $TV_x$ >= $TV_{threshold}$ Then check $RE_x$
 5. If $RE_x$ is >> Min energy required for  transmission
 6. Then set a FT relationship between node   x   and
    evaluating nnode
 7. Else if  $TV_x$ = TV threshold and $RE_x$ = min(energy)
    set a PT relationship between node x and evaluating
    node.
 8. Else set a NT relationship between x and evaluating
    node.
 9. all nodes having FT or PT relation with claiming
    node x jointly declare x as a cluster head (CH)
10. This election is broadcasted in the network.
```

3.2 Cluster Member Selection

The 1 – hop nodes which receive *claim()* from elected *CH* send the *join()* message to the elected cluster head. The *CH* before allowing a node to be its cluster member checks for their *TV* and then allows it to be a *CM*.

Algorithm election Cluster member
NNODE: set of all 1 hop neighbors

```
For every node having a FT or PT relation with
elected CH
send join() message to respective CH
for every nnode ϵ NNODE sending join()
check $TV_{nnode}$ in RT(CH) and
    RT(NNODE - nnode)
if $TV_{nnode}$ >= $TV_{threshold}$ then
   nnode = CM (CH).
```

3.3 Cluster Group Key Generation

All the Cluster Members of a cluster self generate their public keys and submit it to their respective Cluster Heads. Using these keys the *CH* computes a group key *(GK)* is distributed among all the cluster members of a specific cluster forming a key agreement zone between the Cluster head and its Cluster Members. All the cluster members contribute towards the computation of the *GK*. This *GK* is used for encryption / decryption of message exchanges within a group.

$$Group\ Key\ (GK) = f[PK(CMn)] \tag{8}$$

$where\ n - all\ 1\ \ hop\ neighbors\ \ of\ CH$

Algorithm Cluster Group Key Generation.

```
For all CM ϵ CH
Generate a key pair (SK,PK)
Submit the public keys PK of all nnode to CH
$CH_i$ generates a group key ($GK_i$) for $i^{th}$ cluster
Distribute $GK_i$ in all CM of $i^{th}$ cluster.
```

3.4 Recomputation of Group Key

A group key needs to be recomputed in the following cases.

1. Non Trusted *nnode* detected: Trust relationships among the *CH*s and *CM*s are periodically checked and till anode is having a *FT* or *PT* relationship with its *CH* it is allowed to be a part of this cluster. When a *CH* detects a non trusted neighbor it broadcasts this in the cluster and is excluded from the cluster. At this stage a new *GK* is computed and redistributed in the cluster.

2. Mobility / Death of a *nnode*: A node apart from being malicious can also move out of the RF (Radio Frequency) range of the cluster head due to mobility or may have died because of exhausted battery. A *CH* if not gets any hello packet from a specific *CM* after a specific interval tries to search for the *CM* through *nnode*. If *CM* is found it tries to establish a link with this node, else considering it as moved

to another cluster or dead its respective *RT* (Routing Table) entries are removed from *CH*s Routing table. At this stage a new *GK* is computed.

3. New *nnode* sends a *join()* : when a new *nnode* tries to join the cluster , sends its key to the *CH*. *CH* then computes a new *GK* and distributes it in the group.

3.5 Certificate Computation for *nnode*

Once all the cluster head have been elected they establish a session key among themselves using their key pair for inter cluster communication. Intra cluster communication takes place using Cluster Group key. *CH*s also generate a certificate for their respective *CM*s and broadcast it to all *CH*s or pass it them whenever required. The certificate *CERT* is computes as:

$$CERT^{CH}_{nnode} = SK_{CH}(PK_{CH}, PK_{nnode}, ID_{CH}, ID_{nnode}, validity) \quad (9)$$

$CERT^{CH}_{nnode}$ – certificate computed by *CH* for nnonde.

SK_{CH} – Secret Key of *CH*

PK_{CH} – Public Key of CH

PK_{nnode} –Public key of nnode

ID_{CH} – Identity of CH

ID_{nnode} – ID of nnode

validity – Validity period of Certificate

4 Simulation Environment

Simulation of the proposed protocol has been done on NS – 2.34 (Network Simulator) [14] and the results are produced and analyzed using tracegraph 2.02 [15] analyzer with the parameters given in Table 2.

Table 2. Simulation Parameters

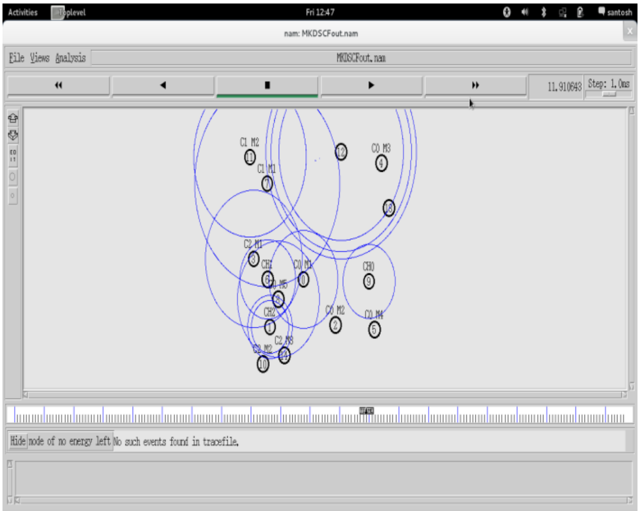
Simulation Parameter	Value
Channel Type	Wireless Channel
MAC Type	802.11
Number of nodes	15
Routing Protocol	AODV
Dimension	500*500 m
Simulation Time	20s
Data Interval	0.5 s

Initial weight and initial energy of every node in the network is calculated. Using TMS in [13] the initial trust values of all the nods is evaluated and nodes are categorized as fully trusted (FT), partially trusted (PT) or non trusted (NT). Various initial node parameters are given in Table 3.

Table 3. Node Parameters

Node Number	Corresponding Weight	Initial Energy	Initial TV
0	5	64	0
1	4	192	1
2	5	110	0.5
3	2	127	0
4	1	154	1
5	3	131	0.5
6	4	56	0.5
7	3	90	0
8	4	31	0
9	5	168	1
10	1	82	0
11	2	87	0
12	3	166	0.5
13	1	150	1
14	1	127	1

Figure 2 shows the network scenario after all the clusters were formed.



No such events found in tracefile.

Fig. 2. Network Scenario

Applying algorithms for *CH* and *CM* four clusters were formed with *CH*s being node 1, 9, 6 and 12. The cluster members list is as given in Table 4.

Table 4. Cluster Details

Cluster Head	Cluster Members
Node 1	3,10,14
Node 6	7,11
Node 9	0,2,4,5,8
Node 12	13

5 Results and Conclusion

The proposed protocol is being compared with classical AODV protocol running for the same simulation environment and following results were obtained which prove the benefit of clustering in the network. At the same time the results also support the trust criteria used in forming the clusters by improving the network parameters.

Figure 3 gives the initial parameters of the nodes in the network which are used in deciding the first Cluster head of the network and gives node 1, 6, 9 and 12 as the initial *CH*s of the system.

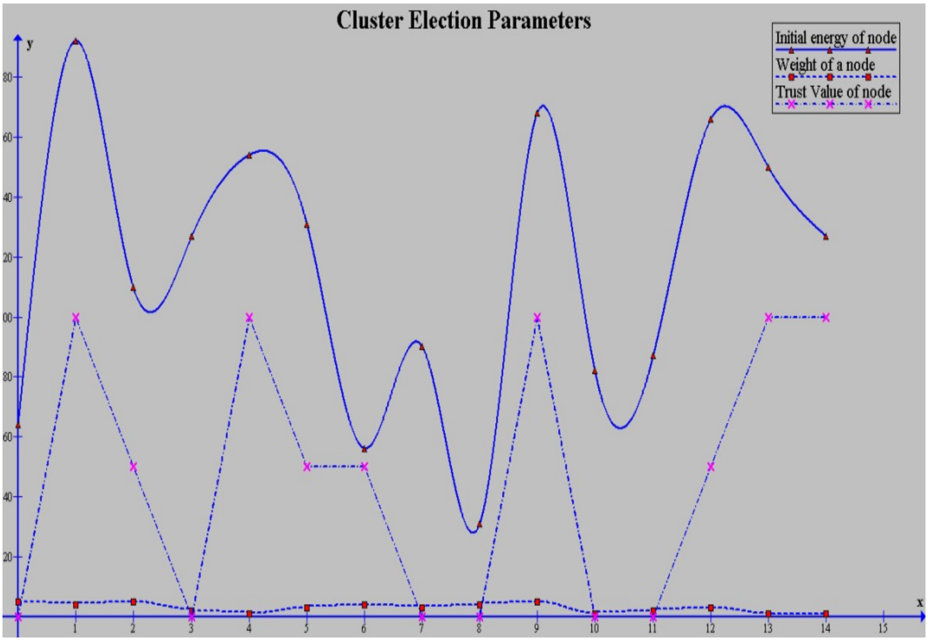


Fig. 3. Cluster Election Parameters

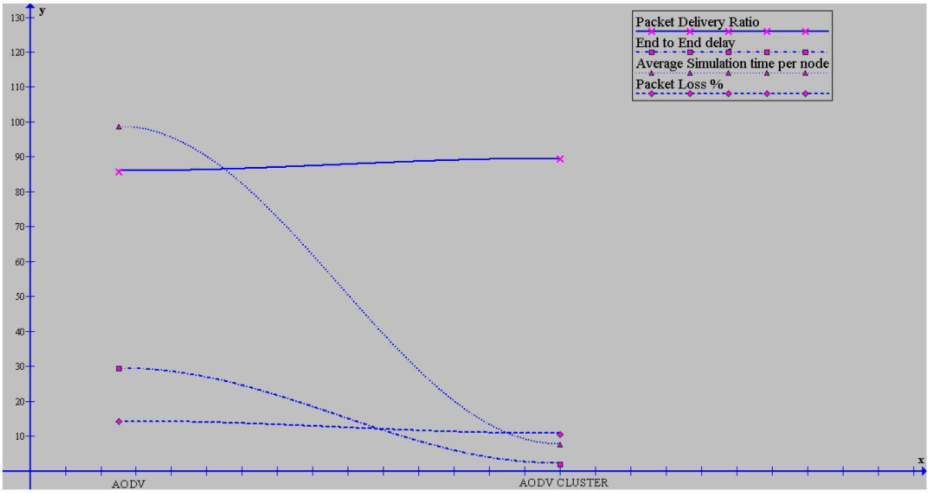


Fig. 4. Performance Metrics

Figure 4 show the improvement in the network parameters when trust based clustering is used. It clearly shows that average end to end delay of the network is decreased as the group to which a node transmits data is small. Average simulation time at every node also decreases. The trust incorporated in the network reduces the packet drop percentage in the network and as a result the packet delivery ratio of the network improves when trust based clustering scheme is used.

References

[1] Chatterjee, M., Das, S.K., Turgut, D.: An on-demand weighted clustering algorithm (WCA) for ad hoc networks. In: Proc. of IEEE GLOBECOM, San Francisco, pp. 1697–1701 (2000)

[2] Malpani, N., Welch, J., Vaidya, N.: Leader Election Algorithms for Mobile Ad Hoc Networks. In: Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, MA (2000)

[3] Vasudevan, S., Decleene, B., Immerman, N., Kurose, J., Towsley, D.: Leader Election Algorithms for Wireless Ad Hoc Networks. In: Proc. of DARPA Information Survivability Conference and Exposition (2003)

[4] Yassein, M.B., Hijazi, N.: Improvement on Cluster Based Routing Protocol By Using Vice Cluster Head. In: Fourth International Conference on Next Generation Mobile Applications, Services and Technologies (2010)

[5] Gerla, M., Tsai, J.T.C.: Multicluster, mobile, multimedia radio network. Wireless Networks 1(3), 255–265 (1995)

[6] Baker, D.J., Ephremides, A.: The architectural organization of a mobile radio network via a distributed algorithm. IEEE Transactions on Communications, 1694–1701 (1981)

[7] Er, I.I., Seah, W.K.G.: Mobility-based D-hop Clustering Algorithm for Mobile Ad hoc Networks. In: IEEE WCNC, Atlanta, USA (2004)

[8] Safa, H., Artail, H., Tabet, D.: A cluster based trust-aware routing protocol for mobile ad hoc networks. Springer Science Business Media, LLC (2009)

[9] Bechler, M., Hof, H.-J., Kraft, D., Pahlke, F., Wolf, L.: A Cluster- Based Security Architecture for Ad Hoc Networks. In: Proc. of IEEE INFOCOM (2004)

[10] Rachedi, et al.: Trust and mobility based clustering algorithm for secure ad hoc networks. In: Proc. of ICSNC 2006 (2006) ISBN: 0-7695-2699-3

[11] Hubaux, J.P., Buttyan, L., Capkun, S.: The Quest for Security in Mobile Ad Hoc Networks. In: Proc. of ACM Symposium on Mobile Ad Hoc Networking and Computing, pp. 146–155 (2001)

[12] Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proc. of the Mobile Computing Systems and Applications, pp. 90–100

[13] Khatri, P., Tapaswi, S., Verma, U.P.: Trust evaluation in wireless ad hoc networks using fuzzy system. In: Potdar, V., Mukhopadhyay, D. (eds.) CUBE 2012, pp. 779–783 (2012)

[14] NS-2 simulation tool home page (2000), http://www.isi.edu/nsnam/ns/

[15] Malek, J.: Trace graph - Network Simulator NS-2 trace files analyser (2003), http://www.tracegraph.com/