Trusted Destination Sequenced Distance Vector Routing Protocol for Mobile Ad-hoc Network

Mohd Zamir Arif Gaurav Shrivastava

Student, Department of CSE., R.K.D.F., Bhopal

ABSTRACT

A Mobile Ad hoc Network (MANET) is a self-organized system comprised of mobile wireless nodes with peer relationships. MANETs can operate without fixed infrastructure and can survive rapid changes in the network topology.

Due to multi-hop routing and absence of any trusted third party in open environment, MANETs are vulnerable to attacks by malicious nodes or unwanted packet forwarding through UDSDV (Un-trust Destination Sequence distance vector routing). In order to decrease the unwanted data flooding and routing misbehaviour from malicious nodes or UDSDV node, we introduce the concept of trust based destination sequence distance vector routing that is TDSDV module, if we apply TDSDV routing and same time UDSDV node presence in the network so TDSDV node protect through unwanted packet flooding of the network and increases network performance.

In this paper we proposed TDSDV trust based destination sequence distance vector routing and analyze the behaviour on the network parameter like throughput, packet delivery ratio, end-to-end delay, routing overhead and energy consume via mobile nodes in all three cases DSDV, UDSDV and TDSDV routing.

Keywords:

Routing Load, average end-to-end delay, packet delivery fraction, TCP, UDP, and DSDV, UDSDV, TDSDV.

1. INTRODUCTION

The wireless nature and inherent features of mobile ad hoc networks make them vulnerable to a wide variety of attacks by misbehaving nodes. Such attacks range from passive eavesdropping, where a node tries to obtain unauthorized access to data destined for another node, to active interference where malicious nodes hinder network performance by not obeying globally acceptable rules. For instance, a node can behave maliciously by not forwarding packets on behalf of other peer nodes. However, when a node exhibits malicious behaviour it is not always because it intends to do so. A node may also misbehave because it is overloaded, broken, compromised or congested in addition to intentionally being selfish or malicious [1,2,3]. An overloaded node lacks the CPU cycles to attend its local and/or network tasks, which leads it to drop packets owing to its inability to process them. A broken node has a software or hardware fault that prevents it from performing its network duties properly. A compromised node may be victim of an attack that degrades its data forwarding capabilities. A congested node receives more packets than the bandwidth available to it allows it to send, its buffer fills and eventually it has to drop incoming packets. A selfish node is unwilling to use its resources such as battery life, bandwidth or processing power to forward packets on behalf of other nodes. A malicious node drops packets or generates additional packets solely to disrupt the network performance and prevent other nodes from accessing any network services (a denial of service attack). Both selfish and malicious nodes expect, however, other nodes to forward packets on their behalf in spite of their own misbehaviour.

2. RELATED WORK

In [4] Zhao et al have reviewed the existing approaches of available bandwidth estimation. They presented the efforts and challenges in estimation of bandwidth. Also, they proposed a model for finding available bandwidth with improved accuracy of sensing based bandwidth estimation as well as prediction of available bandwidth.

In [5] Gui et al have defined routing optimality with the usage of different metrics like path length, energy consumption and energy aware load balancing within the hosts. Along with they have proposed a methodology for self-healing and optimizing routing (SHORT) technique for MANET. SHORT increases performance with regard to bandwidth and latency. They classified SHORT into two categories such as Path-Aware SHORT and Energy-Aware SHORT.

The QAMNet [6] approach extends existing ODMRP routing by introducing traffic prioritization, distributed resource probing and admission control mechanisms to provide QoS multicasting. For available bandwidth estimation, it used the same method given in SWAN [7] where the threshold rate for real-time flows is computed and the available bandwidth estimated as the deference between the threshold rate of real-time traffic and the current rate of real-time traffic. It is very difficult to estimate the threshold rate accurately because the threshold rate may change dynamically depending on traffic pattern [7]. The value of threshold rate should be

Manuscript received August 5, 2013
Manuscript revised August 20, 2013

chosen in a sensible way: Choosing a value that is too high results in a poor performance of real- time flows, and choosing a value that is too low results in the denial of real-time flows for which the available resource would have sufficed.

The localization methods are also distinguished by their form of computation, “centralized” or “decentralized”. For example, MDS-MAP [8] is a centralized localization that calculates the relative positions of all the nodes based on connectivity information by Multidimensional Scaling (MDS). Similarly, DWMDS (Dynamic Weighted MDS) [9] uses movement constraints in addition to the connectivity information, and estimates the trajectories of mobile nodes. TRACKIE [10] first estimates mobile nodes that were likely to move between landmarks straight. Based on their estimated trajectories, it estimates the trajectories of the other nodes.

In decentralized methods, the position of each node is computed by the node itself or cooperation with the other nodes. For example, APIT [11] assumes a set of triangles formed by landmarks, checks whether a node is located inside or outside of each triangle, and estimates its location. Amorphous [12] and REP [13] assume that location information is sent through multi-hop relay from landmarks, and each node estimates its positions based on hop counts from landmarks. In particular, REP first detects holes in an isotropic sensor network, and then estimates the distance between nodes accurately considering the holes. In MCL [14], each mobile node manages its Area of Presence (AoP) and refines its AoP whenever it encounters a landmark. In UPL [15], each mobile node estimates its AoP accurately based on AoP received from its neighbouring nodes and obstacle information.

3. PROBLEM STATMENT

Our aim to protect mobile ad-hoc network via apply trusted mechanism in Mobile ad-hoc network. That time we use proactive destination sequence with destination vector routing in MANET. All above procedure done through network simulator and analyze result through PDF (packet delivery ratio), routing load, throughput, network density bases.

4. PROPOSED WORK

In our simulation we create trusted network with TDSDV protocol and measure the parameter of bandwidth, residual energy and data rate, throughput etc. here we describe each steps one by one.

4.1 TDSDV Routing

Trusted Destination Sequenced Distance Vector (TDSDV) Routing Protocol for MANET is a proactive secured routing protocol. It gains some of the inherent qualities of the distance vector algorithm. In such kind of proactive routing protocols, each node repeatedly maintains state-of-the-art routes to every other node in the network. Routing information at regular intervals transmitted throughout the network in order to preserve routing table stability. When the route discovery process is initiated, the two state-of-the art estimations such as bandwidth and variance residual energy will be calculated using (1) and (2). That information is mainly used to determine the path from the source node to the destination node. The routing table is updated at every node by discovering the variation in routing knowledge about all the existing destinations with the number of nodes to the destination.

When the attacker tries to impersonate as intermediate node our TDSDV protocol will recognize the intruder using Intruder Detection Methodology, and redirect the path to the destination. In addition, to offer loop freedom our protocol TDSDV uses succession count, which is offered, by the destination node. However, when a route has already existed before traffic arrives, transmission takes place without any delay. Else, traffic packets must wait in queue till the node gets routing information equivalent to its destination. In case of highly dynamic network topology, the proactive schemes need a noteworthy quantity of resources to maintain routing information up-to-date and reliable.

4.2 Intruder Detection Methodology (IDM)

After calculating the path in which packets are to be routed, the source node will forward certain number packets to the next hop (node). The number of packets thus sent to the first hop will be set as threshold value. Thus obtained threshold value will be verified at every node in the path before despatching the packets. And if any of the nodes in the path has got different value other than that of threshold value then they are treated as Intruder and the path is rediscovered with the new threshold value and discarding the intruder node. Once again the above process is repeated till such time it reaches the destination node.

When the non-availability of a route to the next node, the node instantly updates the succession count and broadcasts the knowledge to its neighbours. When a node gets routing knowledge then it verifies in its routing table. If it does not have such entry into the routing table then updates the routing table with routing information it has obtained. If the node finds that it has already had an entry into its routing table then it compares the succession count of the received information with the routing table entry and updates the information. If it has succession count that is less than that of the received one then it rejects the information with the least succession count. Suppose both the succession counts are one and the same then the node keeps the information that has the shortest route or the least number of hops to that destination.

4.3 Proposed Algorithm For UDSDV And TDSDV Node

Here we proposed an algorithm for finding Un-trusted node using TDSDV routing, general scenario for ad hoc network as well as concept of TDSDV module. Firstly we discuss about DSDV discovery and normal ad hoc scenario.

4.3.1 Algorithm for DSDV Routing Discovery and Scenario Generation

```
Mobile node = N;   // Number of mobile nodes
Sender node = S;     // sub set of N
Receiver Node = R;          //sub Set of N
Start simulation time = $t_0$
Set routing protocol = DSDV;
Set MAC = 802.11           // for Wireless Communication
Set radio range = rr;     //initialize radio range
RREQ_B(S, R, rr)
        {
If ((rr<=250) && (next hop >0))
{
   Compute route ()
        {
         rtable->insert(table->rt_nexthop); // nexthop to RREQ source
         rtable1->insert (rtable1->rt_nexthop); // nexthop to RREQ destination
          If (dest==true)

        {send ack to source node with rtable1;
        Sender generate sequence number for data sending;
            Data_packet_send (s_no, nexthop, type)
        }
        else      {
           destination not found;
                  }
        }
        }
        Else {destination un-reachable;
             }
  }
```

4.3.2 Algorithm for UDSDV Routing and Scenario Generation

```
Mobile node = N;   // Number of mobile nodes
Sender node = S;     // sub set of N
Receiver Node = R;          //sub Set of N
Set Un-trusted Node = U    // Node work Un-trusted mode that also subset of N;
Node U set routing = UDSDV // work through Un-trusted routing;
U node flood unwanted or junk of packet to all neighbour nodes
If ((rr<=250) && (next hop >0))
 {
   Check weakness of the radio range node;
   If (any node receives unwanted flood packet)
          {Node congested and not works properly}
  }
E lse {destination un-reachable;       }
```

4.3.3 TDSDV (Trusted Destination Sequence Vector Routing) Algorithm also IDM

```
Create node =TDSDV; // Trusted as a IDM
Set routing = DSDV;
If ((node in radio range) && (next hop !=Null)
 {Capture load(all_node)
 Create normal_profile();
 Create abnormal_table();
 If   ((load   <   =   max_limit)   &&(new_profile ==normal_profile()))
          {        No any attack;  }
         Else {     Attack in network;
                   If (new_attack == abnormal_table())
                           { Block the infected node ;
                           }
                   Else {
                   Insert Value into abnormal_table ();
                   Find_attack_info ();
                        }
}
     Else {"node out of range or destination unreachable"
          }
Find_attack_info (node_number, pkt_type, time)
         {         Captute infection type;
                   Infect percentage;
                   Port_number;
         }
```

5. SIMULATION ENVIRONMENT

The simulator we have used to simulate the ad-hoc routing protocols in is the Network Simulator 2 (ns) [16] from Berkeley. To simulate the mobile wireless radio environment we have used a mobility extension to ns that is developed by the CMU Monarch project at Carnegie Mellon University.

5.1 Network Animator (NAM)

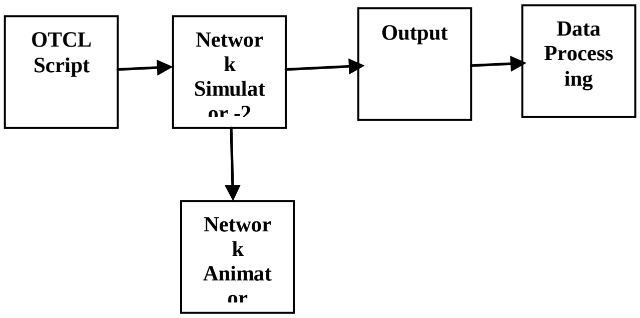


Figure 1: Network simulator 2

NAM is a very good visualization tool that visualizes the packets as they propagate through the network. An overview of how a simulation is done in ns is shown in Figure 1

5.2 Simulation Parameter

We get Simulator Parameter like Number of nodes, Dimension, Routing protocol, transport layer protocol, application layer data and maximum speed of mobile nodes etc. According to below table 1 we simulate our mobile ad-hoc network.

Table 1 Simulation parameter

Number of nodes	40
Dimension of simulated area	800×600
Routing Protocol	DSDV, UDSDV, TDSDV
Simulation time (seconds)	30,100
Transport Layer	TCP ,FTP
Traffic type	CBR
Packet size (bytes)	1000
Number of traffic connections	10
Maximum Speed (m/s)	Random

5.3 Performance Parameter

This section presents the performance parameters used to evaluate the proposed Location Tracking technique case Traffic Analysis. The main performance parameters are Routing message overhead, average end to end delay, and throughput. Under each main performance parameters, there are secondary performance parameters which affect it or depend on it.

5.3.1. Routing Load

The total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each transmission of the packet or each hop counts as one transmission.

5.3.2. Average End to End Delay

This includes all the possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times.

It is calculated as the total summation of the division of total end to end delay (Dt) by the number of packets delivered (Npd) divided by the number of nodes (Nn) as in Eq.(1)

$$\frac{\sum \frac{Dt}{Npd}}{\mathrm{Nn}}$$

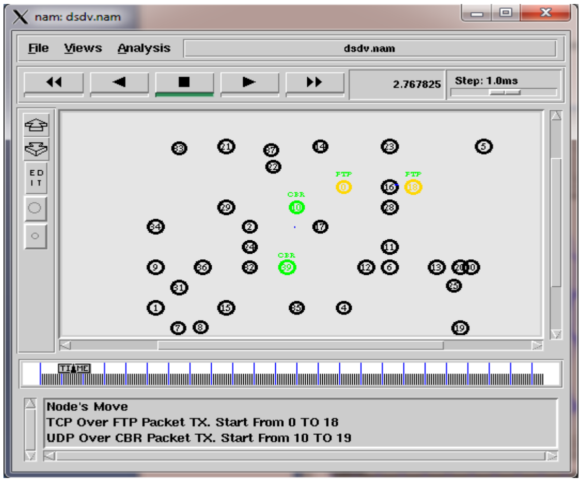
5.3.3 Packet Dropped:

The routers might fail to deliver or drop some packets or data if they arrive when their buffer are already full. Some, none, or all the packets or data might be dropped, depending on the state of the network, and it is impossible to determine what will happen in advance.

5.4 Nam visualization

The simulation described in this project was tested using the ns-2 test-bed that allows users to create arbitrary network topologies [16]. By changing the logical topology of the network, ns-2 users can conduct tests in an ad hoc network without having to physically move the nodes. Ns-2 controls the test scenarios through a wired interface, while the ad hoc nodes communicate through a wireless interface.

The topology shown in Figure 2 is used 40 mobile nodes to show how the node senses the neighbour nodes and sends data to destination through shortest path.



Node's Move
TCP Over FTP Packet TX. Start From 0 TO 18
UDP Over CBR Packet TX. Start From 10 TO 19

Figure 2: A sample topology generated by ns-2 forty Node Case

The overall goal of the simulation experiments is to measure the accuracy and robustness of our Trust based routing and intrusion prevention scheme for wireless mobile ad hoc networks while continuing to successfully deliver data packets to their destinations. To measure this ability, a variety of workloads were applied to the simulated network, including node movement, data traffic patterns, node density and varying percentages of malicious nodes.

Our simulation test bed in ns-2 simulator [40] is based on a movement space with 40 mobile nodes. IEEE 802.11 MAC layer is used with carrier sense and back-off mechanisms and the transport layer used is User Datagram Protocol (UDP) and transport control protocol (TCP). Nodes move according to the random waypoint mobility model. Assuming that the mobility of the ad-hoc networks is inversely proportional to the pause time, we have simulated the mobility by use of pause time. The longer the pause time, the less the mobility.

5.5 Throughput Analysis DSDV, UDSDV and TDSDV Case

In mobile ad-hoc wireless networks, such as packet radio, throughput or network throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain wireless network node. The throughput is usually measured in bits per second (bps), and sometimes in data packets per second or data packets per time slot.

In our simulation our throughput measure and get the result in DSDV normal routing time and TDSDV trust routing time throughput value is similar, but UDSDV un-trust DSDV routing time throughput value degraded after 25th seconds that means receiving percentage decrease.

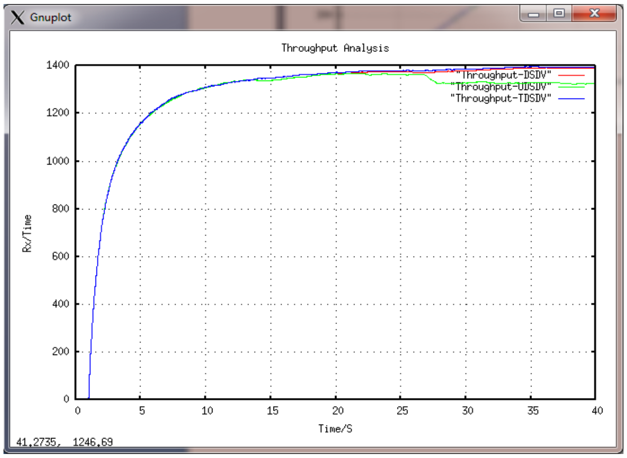


Figure 3 Throughput Analyses DSDV, UDSDV and TDSDV Time

5.6 Gnuplot for Routing Overhead Analysis

Routing Overhead means total number of routing packet spread on the network out of actual data transmission packet that value if lower that means our routing performance is better and conclude maximum network utilize through actual data transmission not through routing packet. In our simulation we create three different scenario DSDV time, UDSDV and TDSDV time, normal DSDV routing and TDSDV time routing load is nearly same and minimum but the case of un-trust UDSDV case routing overhead is very higher nearly 82000 routing packets.

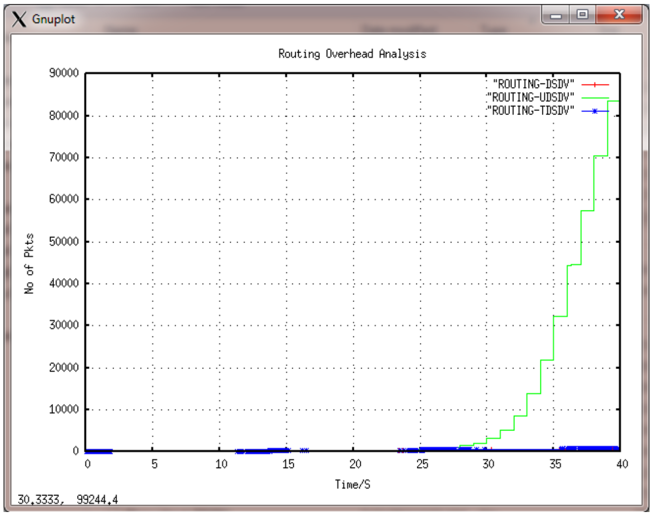


Figure 4 Routing Overhead Analysis

5.7 Gnuplot for Packet Delivery Fraction

In this simulation forty mobile nodes are created and calculate packet delivery ratio, packet delivery ratio is a ratio between packets receives by the authentic receiver from genuine packets sends by sender at current time. If packet delivery ratio is higher that means performance is best, here in this result if UDSDV node p resent in network that time packet delivery ration is 90% that conclude node UDSDV node misbehave and decrease the performance of the network nearly 7% . Result also shows DSDV time and TDSDV time both gives same packet delivery ratio that concludes 100 percent recovery through TDSDV node.

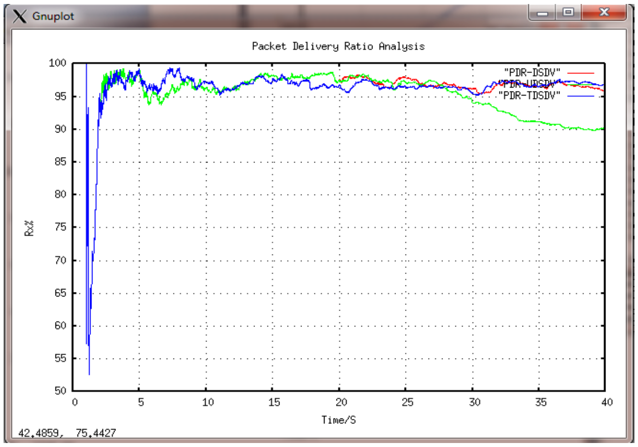


Figure 5 Packet Delivery Ratio Analyses

5.8 Untruth Time Infection Percentage Analysis (UDSDV)

Here we analyze infection percentage spreading on our network, basically un-trusted node enter on network at 1st second and send bunch of packet to our network, if any node receive that infected un-trust bunch of packet so that mobile node infected through attacker activity. In our simulation at 24th second network infect via UDSDV activity. And result shows nearly 10% network infected via unwanted packet and this produce congestion on the network so that the network performance very degraded.

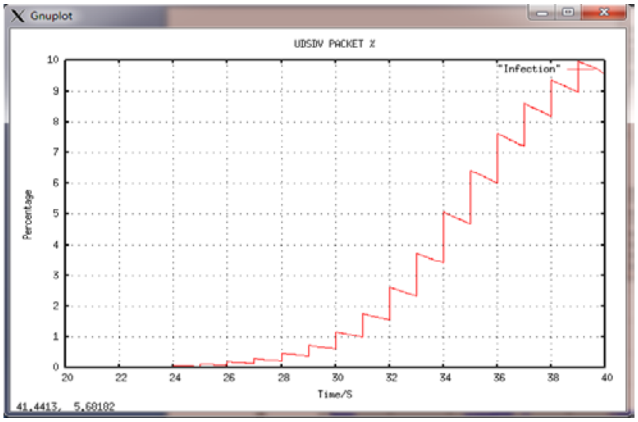


Figure 6 Un-trust Time Infection Percentage Analysis (UDSDV)

6. CONCLUSION

In this dissertation we designed and developed a Trust based destination sequenced routing (TDSDV) routing protocol which meets the requirements of QoS such as improved throughput with better packet delivery ratio and reduced end-to-end delay and reduced no of drop in packets given in table. Additionally, we provide a secure route maintenance mechanism by involving threshold in terms of packets.

We perform number of test in ns-2 simulator and analyze the result we get the summery result according to test simulation in normal DSDV routing time total number of packet transmitted by the genuine sender is 3724 but in case of UDSDV routing time node (un-trust) inter on the network so that packet transmission only 3666 that means 2% transmission decreases. But if we set one node as TDSDV so transmission percentage increases as compare UDSDV time that result concludes 6% data delivery increases. Other side also PDF packet delivery fraction analysis if UDSDV routing case on to the network so 90.15% PDF. And TDSDV gives better the PDF it is 96.6%. we also analyze routing overhead in normal DSDV routing case only 0.13% of routing load but UDSDV routing present so routing overhead is increases and routing load as 12.65%. That means very–very routing over head increases it gives poor performance of the network, finally we conclude our result TDSDV (Trusted destination distance vector routing) 99.9% data recover. And IDS time only 0.13% routing overhead.

Table 2 Overall Summery Analyses DSDV, UDSDV and TDSDV

Overall ANALYSIS				
PARAMETER		DSDV	UDSDV	TDSDV
SEND	=	3724	3666	3707
RECV	=	3570	3305	3583
ROUTINGPKTS	=	470	41824	483
PDF	=	95.86	90.15	96.65
NRL	=	0.13	12.65	0.13
Average e-e delay(ms)	=	393.97	432.69	391.77
No. of dropped data (packets)	=	151	358	120
No. of dropped data (bytes)	=	99868	199432	73496

REFERENCES

[1] Dr. S. Santhosh Baboo1 and S. Ramesh "Secured-destination Sequenced Distance Vector (SSDV) Routing Protocol for Mobile Ad-hoc Networks" International Journal of Computer Science and Telecommunications [Volume 2, Issue 8, November 2011]

[2] H. Yang, J. Shu, X. Meng, and S. Lu, “SCAN: Self-organized network-layer security in mobile ad hoc networks,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, February 2006, pp. 261-273.

[3] P. Karn, “MACA – a new channel access method for packet radio,” in proceedings. ARRL/CRRL Amateur Radio Computer Networking Conference, September 1990.

[4] Haitao Zhao, Jibo Wei, Shan Wang and Yong Xi, “Available Bandwidth Estimation and Prediction in Ad hoc Networks”, Wireless Networks, Vol.14, pp. 29–46, 2008.

[5] Chao Gui & Mohapatra, “A Framework for Self-healing and Optimizing Routing Techniques for Mobile Ad hoc Networks”, Wireless Networks, Vol.14 No.1, pp.29-46, 2008.

[6] H. Tebbe, and A. Kassler, “QAMNet: Providing Quality of Service to Ad-hoc Multicast Enabled Networks”, 1st International Symposium on Wireless Pervasive Computing (ISWPC), Thailand, 2006.

[7] G. S. Ahn, A. T. Campbell, A. Veres and L.H. Sun, "SWAN: Service Differentiation in Stateless Wireless Ad hoc Networks", In Proceedings IEEE INFOCOM, 2002.

[8] Y. Shang, W. Rml, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. IEEE Transaction on Parallel and Distributed Systems, 15(11):961–974, 2004.

[9] J. M. Cabero, F. D. la Torre, A. Sanchez, and I. Arizaga. Indoor people tracking based on dynamic weighted multidimensional scaling. In Proc. of MSWiM 2007, pages 328–335, 2007.

[10] S. Fujii, A. Uchiyama, T. Umedu, H. Yamaguchi, and T. Higashino. An off-line algorithm to estimate trajectories of mobile nodes using ad-hoc communication. In Proc. Of PerCom 2008, pages 117–124, 2008

[11] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Rang e-free localization schemes for large scale sensor networks. In Proc. of MobiCom 2003, pages 81–95, 2003.

[12] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In Proc. of IPSN 2003, pages 333–348, 2003.

[13] M. Li and Y. Liu. Rendered path: range-free localization in anisotropic sensor networks with holes. In Proc. Of MobiCom 2007, pages 51–62, 2007.

[14] P. Mohapatra, J. Li, and C. Gui, “QoS in mobile ad hoc networks,” IEEE Wireless Commun. Mag. (Special Issue on QoS in Next-Generation Wireless Multimedia Communications Systems), pp. 44–52, 2003.

[15] A. Uchiyama, S. Fujii, K. Maeda, T. Umedu, H. Yamaguchi, and T. Higashino. Ad-hoc localization in urban district. In Proc. of INFOCOM 2007 Mini-Symposium, pages 2306– 2310, 2007.

[16] The Network Simulator – ns-2 http://www.isi.edu/nsnam/ns/