MS2006
Proceedings of the International Conference on Modeling and Simulation 2006
28 – 30 August 2006, Konya, TURKEY
Paper No B101

MODELLING AND SIMULATION OF WIRELESS SENSOR AND MOBILE AD HOC NETWORKS

Kayhan Erciyes, Orhan Dagdeviren, Deniz Cokuslu
Izmir Institute of Technology
Izmir Institute of Technology, Gulbahce Koyu, Urla Izmir
kayhanerciyes@iyte.edu.tr, orhandagdeviren@iyte.edu.tr,
denizcokuslu@iyte.edu.tr

ABSTRACT

Mobile ad hoc Networks do not have any fixed infrastructure and consist of wireless mobile nodes that perform various data communication tasks. Wireless Sensor Networks are a special type of ad hoc Networks which are highly distributed networks of small, lightweight wireless nodes, deployed in large numbers to monitor the environment or a system. In this paper we provide a study of modeling of Mobile Ad hoc (MANET) and Wireless Sensor Networks (WSN) by graph theoretic structures and simulating with *ns2* simulator. We provide the simulation results of our approaches on MANETs.

Keywords: Mobile ad hoc Network, Wireless Sensor Network Dominating Set, Minimum Spanning Tree, *ns2*

INTRODUCTION

A mobile ad-hoc network (MANET) is a self-configuring network of mobile hosts connected by wireless links which forms an arbitrary topology. The hosts are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger internet. Minimal configuration and quick deployment make ad hoc networks suitable for emergency situations like natural or human-induced disasters, military conflicts, emergency medical situations. An important way to support efficient communication between nodes of a MANET is to develop a wireless mobile backbone architecture. Nodes in a MANET are powered by batteries only. Therefore, amount of communication should be minimized to avoid a premature drop out of a node from the network. A wireless sensor network (WSN) is a network made of many small computers, which are employed in the processing of sensor data. The uses for WSNs are many and varied. They could be used in industry to monitor dangerous/hermitically sealed environments. They could be deployed in wilderness areas, where they would remain for many years (monitoring some environmental variable) without the need to recharge/replace their power supplies. WSNs, are also energy-constrained.

In mathematics and computer science, graph theory has for its subject matter the properties of graphs. A graph is a pair $G=(V, E)$, where G is a set of vertices, and E is a set of pairs of vertices or edges. An undirected graph is a graph in which the nodes are connected by undirected arcs. MANETs and WSNs can be modeled by an undirected graph. The hosts of MANETs and WSNs can be represented by vertices and the transmission channels can be represented by edges. By this representation, we can use graph theoretical structures to maintain a mobile backbone architecture and clustering. Two main methods of this process are the constructions of a dominating set and a minimum spanning tree on a graph.

In this paper, we aim to show the modeling techniques and simulation environments for MANETs. We firstly define dominating set and minimum spanning tree concepts in the background section. In the Modeling of MANETs section, we describe our approaches based on dominating set and minimum spanning tree for clustering and backbone formation for MANETs. Simulation of MANETs section summarizes simulation environments for MANETs and *ns2* environment is explained with our studies. Also, in this section, we provide the results of our algorithms.

BACKGROUND

Every vertex in G is either in S or adjacent to a vertex in S. Dominating sets can be classified into three main classes, Connected Dominating Sets (CDS), Weakly Connected Dominating Sets (WCDS) and Independent Dominating Sets(IDS)[1] as described below and also shown in Fig.1.

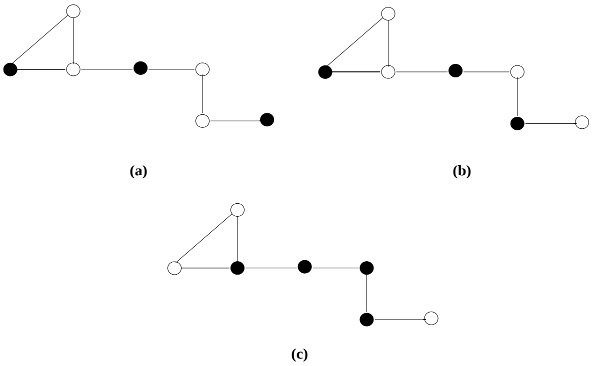

Fig.1. (a) ICDS (b)WCDS (c)CDS

- Independent Dominating Sets: IDS is a dominating set S of a graph G in which there are no adjacent vertices. Fig.1.a shows a sample independent dominating set where black nodes show cluster heads.
- Weakly Connected Dominating Sets: A weakly induced subgraph S_w is a subset S of a graph G that contains the vertices of S their neighbors and all edges of the original graph G with at least one endpoint in S. A subset S is a weakly-

connected dominating set, if *S* is dominating and S_w is connected[8]. Black nodes in Fig.1.b show a WCDS example.

- Connected Dominating Sets: A connected dominating set (CDS) is a subset *S* of a graph *G* such that *S* forms a dominating set and *S* is connected. Fig.1.c shows a sample CDS.

A graph G is connected if there is a path between any distinct *V*. A graph $G_s = (G_s, E_s)$ is a spanning subgraph of *G=(V,E)* if $V_s = V$. A spanning tree of a graph is an undirected connected acyclic spanning subgraph. Intuitively, a spanning tree for a graph is a subgraph that has the minimum number of edges for maintaining connectivity[5]. A graph and its minimum spanning tree is shown in Fig.2.

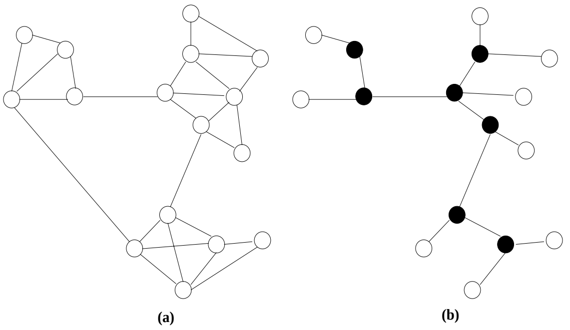

Fig.2. (a)A Graph (b)Its Minimum Spanning Tree

MODELING OF MANETS

Dominating Set based Modeling

By using IDS, one can guarantee that there are no adjacent cluster heads in the entire graph. This minimizes the number of dummy clusters in the network. Although IDS are suitable for constructing optimum sized dominating sets, they have some defficiencies such as lack of direct communication between cluster heads. In order to obtain the connectivity between cluster heads, WCDSs can be used to construct clusters. CDSs have many advantages in network applications such as ease of broadcasting and constructing virtual backbones[2], however, when we try to obtain a connected dominating set, we may have undesirable number of clusterheads. So in constructing connected dominating sets, our primary problem is the minimum connected dominating set decision problem.

Dominating Set Based Algorithm[3] finds a minimal connected dominating set in a MANET. We developed our algorithm based on Wu's CDS Algorithm[4] but we add some extra heuristics. First, we determined some situations that a node cannot change its color after the first phase. We also consider the degree of a node when marking it. This is due to the fact that a node with a higher degree should have a better chance of being in CDS as it has more neighbors than a node with a lower degree.

We assume that each node has a unique *node_id* and knows its adjacent neighbors. Each node has a *color* indicating whether the node is in the dominating set or not. The *color* is set to BLACK if the node is in the dominating set, or WHITE if the node is not in the dominating set. Color GRAY is used to indicate that the node is marked after the first phase, but it will change its color after the second phase as either WHITE or BLACK. Every node in the network performs the same local algorithm periodically. We obtained the resulting connected dominating set in Fig.3 by using our algorithm. In the first phase of our algorithm, there are two conditions. At the end of the first phase, nodes 6, 8, 9 and 10 determine their colors permanently.

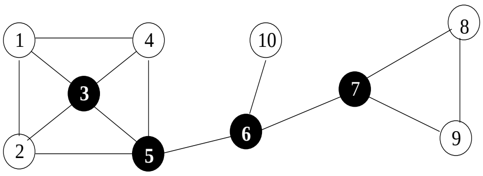

Fig.3. An Example Operation of Dominating Set Based Algorithm

Minimum Spanning Tree based Modeling

Merging Clustering Algorithm(MCA) [6] finds clusters in a MANET by merging the clusters to form higher level clusters as mentioned in Gallagher, Humblet, Spira's algorithm [7]. However, we focus on the clustering operation by discarding minimum spanning tree. This reduces the message complexity. The second contribution is to use upper and lower bound heuristics for clustering operation which results balanced number of nodes in the cluster formed.

We assume that each node has distinct *node_id*. Moreover, each node knows its *cluster_leader_id, cluster_id* and *cluster_level*. *Cluster_level* is identified by the number of the nodes in a cluster. Leader node is the node with maximum *cluster_id*. *Cluster_leader_id* is identified by the *node_id* of the leader node in a cluster. *Cluster_leader_id* is equal to the *cluster_id*. The local algorithm consists of sending messages over adjoining links, waiting for incoming messages and processing messages.

Assume the graph model of the mobile network in Fig.4. *K* parameter is given as 3. The cluster formation scheme is applied. The clusters are shown in Fig.4.

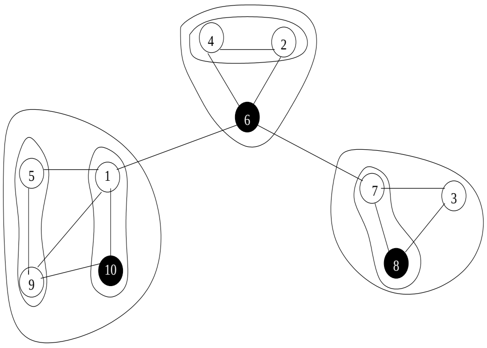

Fig.4. An Example Operation of MCA

The main idea of Backbone Formation Algorithm is to maintain a directed ring architecture by constructing a minimum spanning tree between clusterheads and classifying clusterheads into *BACKBONE* or *LEAF* nodes, periodically. To maintain these structures, each clusterhead broadcasts a *Leader_Info* message by flooding. In this phase, cluster member nodes are acting as routers to transmit *Leader_Info* messages. Algorithm has two modes of operation; a hop-based backbone formation scheme and a position-based backbone formation scheme. In hop-based backbone formation scheme, minimum number of hops between clusterheads are taken into consideration in the minimum spanning tree construction. Minimum hop counts can be obtained during the flooding scheme. For highly mobile scenarios, an agreement between clusterheads must be maintained to guarantee the consistent hop information. In position-based backbone formation scheme, positions of clusterheads are used to construct the minimum spanning tree. If each node knows its velocity and the direction of velocity, these information can be appended with a timestamp to the *Leader_Info* message to construct a better minimum spanning tree. But in this mode, nodes must be equipped with a position tracker like a GPS receiver.

A balanced clustered MANET with its clusterheads and minimum spanning tree is shown in Fig.5. *BACKBONE* clusterheads are filled with black and *LEAF* clusterheads are filled with white.

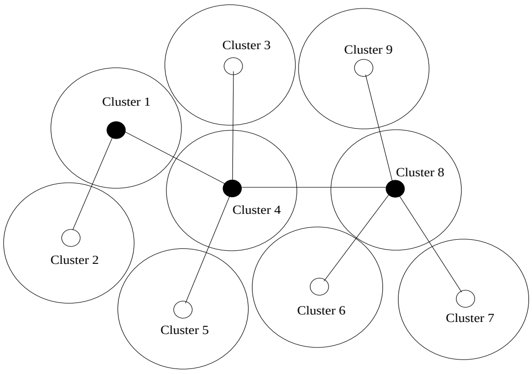

Fig.5 A clustered MANET and its minimum spaning tree.

SIMULATION OF MANETS

Simulation Environments

There are several different simulation programs that can be used for the simulation of MANETs. *QualNet*, *ns2*, *GloMoSim*, and *OPNET* are programs used to simulate MANETs. We choose *ns2* simulator for several reasons. Firstly, *ns2* is probably the most commonly used software of the four. The source code can be downloaded, is free of charge, and compiled on different platforms, e.g. Unix and Windows. Many wireless extensions have been contributed from the UCB Daedelus, the CMU Monarch projects and Sun Microsystems. In *ns2*, it is possible to alter and write your own code to make it more suitable for your own scenarios. [9]

The wireless model essentially consists of the *MobileNode* class at the core, with additional supporting features that allows simulations of multi-hop ad-hoc networks, wireless LANs etc. The *MobileNode* object is a split object. The *C++* class *MobileNode* is derived from parent class *Node*. *MobileNode* is the basic *ns Node* object with added functionalities like movement, ability to transmit and receive on a channel that allows it to be used to create mobile, wireless simulation environments. The class *MobileNode* is derived from the base class *Node*. *MobileNode* is a split object. The mobility features including node movement, periodic position updates, maintaining topology boundary etc. are implemented in *C++* while plumbing of network components within *MobileNode* itself (like classifiers, *dmux* , *LL, Mac*, *Channel* etc) have been implemented in *Otcl.* The network stack for a mobile node consists of a link layer(*LL*), an ARP module connected to LL, an interface priority queue(*IFq*), a mac layer(*MAC*), a network interface(*netIF*), all connected to the channel [10]. These network components are created and plumbed together in *OTcl*. To create and configure *nn* number of mobile nodes with network stack, *Otcl* codes are used :

```
$ns_ node-config
   -adhocRouting $opt(adhocRouting)
   -llType $opt(ll)
   -macType $opt(mac)
   -ifqType $opt(ifq)
   -ifqLen $opt(ifqlen)
   -antType $opt(ant)
   -propInstance [new $opt(prop)]
   -phyType $opt(netif)
   -channel [new $opt(chan)]
   -topoInstance $topo
   -wiredRouting OFF
   -agentTrace ON
   -routerTrace OFF
   -macTrace OFF
   for { set j 0 } { $j < $opt(nn)} {incr j} {
      set node_($j) [ $ns_ node ]
   }
```

We create 10 to 100 nodes for our simulations. The protocol stack is adjusted to *IEEE 802.11* standards. We use Dynamic Source Distance Vector routing algorithm for earlier simulations but we are working to implement our cluster-based routing protocol. Upper transport protocol, we implemented our clustering algorithm and backbone formation algorithm.

The mobile node is designed to move in a three dimensional topology. However the third dimension (Z) is not used. That is the mobile node is assumed to move always on a flat terrain with Z always equal to 0. Thus the mobile node has X, Y, Z(=0) coordinates that is continually adjusted as the node moves. The node movement and is defined in a separate file for convinience. Movement file can be generated using CMU's movement generator. By this generator, one can set number of nodes, pause time, maximum speed, minimum speed, simulation time, maximum X and Y coordinates[10]. The executable code is given below:

```
./setdest -n <num_of_nodes> -p <pausetime> -s <maxspeed> -t <simtime> -x <maxx> -y <maxy>
```

In our simulations, we generate different size of flat surfaces to create small, medium and large distances between nodes. Small, Medium and Large surfaces vary between 310m * 310m to 400m* 400m, 410m * 410m to 500m * 500m, 515m * 515m to 650m * 650m respectively. Random movements are generated for each simulation. Low, medium and high mobility scenarios are generated and respective node speeds are limited between 1.0m/s to 5.0m/s, 5.0m/s to 10.0m/s, 10.0m/s to 20.0m/s.

Simulation Results

Fig.6 displays the run-time results of the merging clustering algorithm ranging from 10 to 100 nodes. Run-time values are increased linearly when we increase the total number of MANETs from 10 to 100 node. Linear increase of the total message number can be seen in Fig.7. For a MANET with 40 nodes, K heuristic is chosen from 3 to 7 to measure the cluster quality as shown in Fig.8. Coefficient of variations from K=3 to K=7 are respectively , 0.53, 0.24, 0.13, 0.64 and 0.63. When *K* is equal to 5, maximum balanced clusters occurred with a coefficient of variation of 0.13.

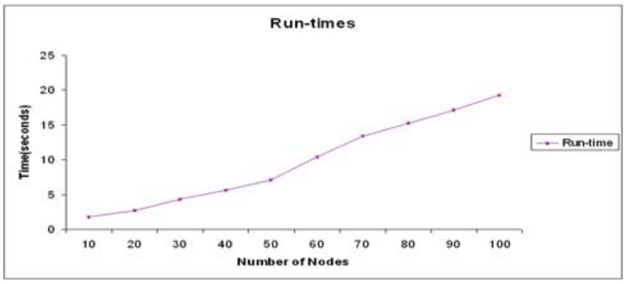

Fig.6. Run-time results for MCA

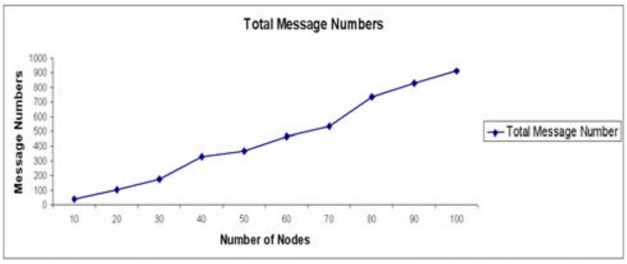

Fig.7. Total Message Numbers

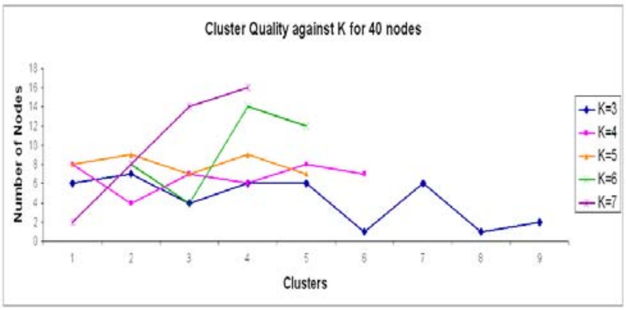

Fig.8. Cluster Quality against K for 40 nodes

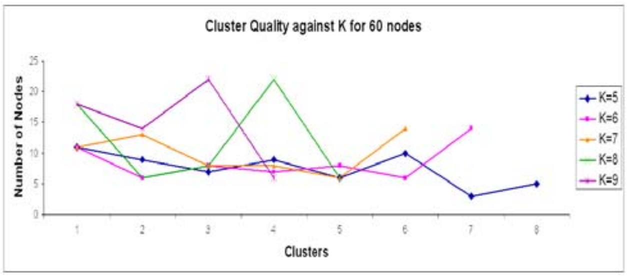

Fig.9. Cluster Quality against K for 60 nodes

Also we can state that total number of clusters are decreased when *K* heuristic is increased as shown in from Fig.8. Cluster Quality against *K* heuristic which is chosen from 5 to 9 for a MANET with 60 nodes is shown in Fig.9. Coefficient of variations from *K*=5 to *K*=9 are respectively, 0.36, 0.34, 0.32, 0.62, 0.46. *K*=7 is chosen for this scenario. In Fig.10, *K* heuristic is fixed to 5 for a MANET with 40 nodes and cluster quality with respect to mobility parameter is measured by selecting from low to high mobility. Coefficient of variation from low to high mobility parameter is respectively, 0.18, 0.37 and 0.25. The MANET with low mobility results in more balanced clusters than MANET with high mobility due to rapid change of network topology. Total Surface area of the MANET is an important parameter which effects the distance between nodes and connectivity. In Fig.11, cluster quality with respect to total surface area is shown.

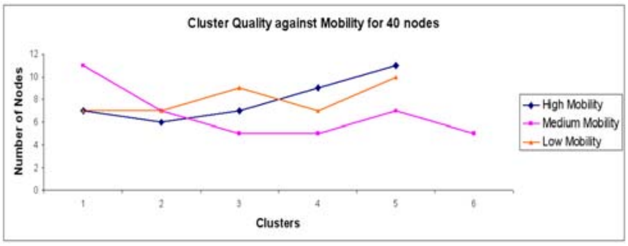

Fig.10. Cluster Quality against Mobility for 40 nodes

Size of surface areas are 340m * 340m, 440m * 440m and 560m * 560m. Coefficient of variation values are respectively, 0.24, 0.25 and 0.28. As we increase the surface area from 340m * 340m to 560m * 560m, clusters become more balanced. Also, cluster node counts are decreased and total number of clusters in MANET are increased.

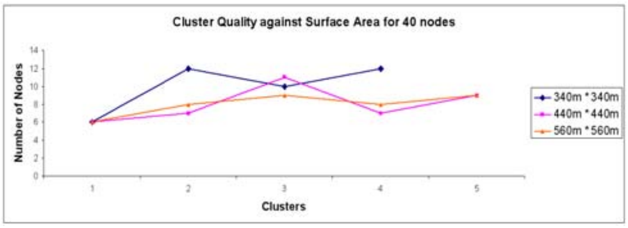

Fig.11. Cluster Quality against Surface Area

To measure the round-trip delay for backbone formation algorithm the total number of nodes are varied between 10 to 100 in Fig.14. Round-trip delay times increase linearly from 20ms to 60ms approximately as shown in Fig.15.

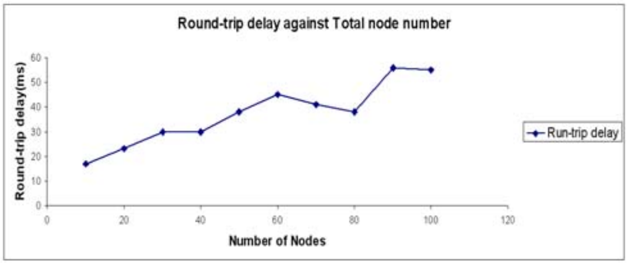

Fig.12. Round-trip delay against Total node number

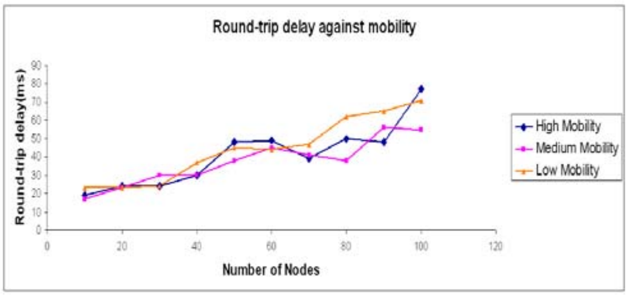

Fig.13. Round-trip delay against Mobility

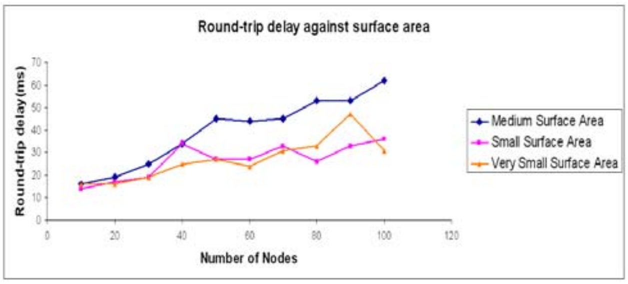

Fig.14. Round-trip delay against surface area

CONCLUSIONS

We showed the dominating set based and minimum spanning tree based modeling techniques for MANETs. We described our approaches for these techniques. Merging clustering algorithm, backbone formation and dominating set based algorithm proposes solutions for clustering and backbone formation problem in MANETs and WSNs. The *ns2* simulation environment is explained with some basic functionalities. Lastly, we provided the simulation results of our approaches and conclude that *ns2* is a suitable simulation tool to test various functionalities of MANETs and WSNs .

REFERENCES

1. Haynes, T. W., Hedetniemi, S. T., Slater, P. J., *Domination in graphs, Advanced Topics*, Marcel Dekker, Inc., 1998.
2. Stojmenovic, I., Seddigh, M., Zunic, J.,*Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks*, IEEE Transactions on Paralel and Distributed Systems, Vol.13, pp. 14-25, 2002.
3. Cokuslu, D., Erciyes, K., Dagdeviren, O., *A Dominating Set Based Clustering Algorithm for Mobile Ad hoc Networks*, to be published in the Proc. of ICCS, 2006.
4. Wu, J., Li, H., *A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks*, Springer Science and Business Media B.V., Formerly Kluwer Academic Publishers B.V., 2001.
5. Grimaldi, R. P., *Discrete and Combinatorial Mathematics, An Applied Introduction*, Addison Wesley Longman, Inc., 1999.
6. Dagdeviren, O., Erciyes, K., Cokuslu, D., *Merging Clustering Algorithms*, to be published in the Proc. of ICCSA, 2006.
7. Gallagher, R. G., Humblet, P. A. and Spira, P. M., *A Distributed Algorithm for Minimum-Weight Spanning Trees*, ACM Transactions on Programming Languages and Systems 5, pp. 66-77., 1983.
8. Chen, Y. P., Liestman, A. L., Liu, J., *Clustering Algorithms for Ad Hoc Wireless Networks*, Nova Science Publishers, 2004.
9. Halvardsson M., Lindberg P., *Reliable Group Communication in Military Ad hoc Network*, Technical Report, Vaxjo University, 2004.
10. *Ns2 documentation*, www.isi.edu/nsnam/ns/ns-documentation