

A Performance Comparison of Virtual Backbone Formation Algorithms for Wireless Mesh Networks

Guillermo Ibáñez^(1,2), Eva Manzanedo⁽¹⁾, Juan A. Carral⁽²⁾, Antonio García⁽²⁾, José Manuel Arco⁽²⁾

⁽¹⁾Dpto. de Ingeniería Telemática. Universidad Carlos III de Madrid, Spain
eva.manzanedo.saenz@accenture.com

⁽²⁾Dpto de Automática. Universidad de Alcalá, Madrid, Spain
{gibanez, jac, antonio, jmarco}@aut.uah.es

Abstract: Currently wireless networks are dominant by star topology paradigm. Its natural the evolution is towards wireless mesh multi-hop networks. This article compares the performance of several algorithms for virtual backbone formation in ad hoc mesh networks both theoretically and through simulations. Firstly, an overview of the algorithms is given. Next, the results of the algorithm simulations made with the program Dominating Set Simulation Suite (DSSS) are described and interpreted. We have been extended the simulator to simulate the Mobile Backbone Network Topology Synthesis Algorithm. The results show that this algorithm has the best combination of performance characteristics among the compared algorithms.

Keywords: Ad hoc, algorithms, connected dominating set (CDS), Dominating Set Simulation Suite, mesh, routing

1. Introduction

Wireless mesh networks help solve wireless networks' dependency on fixed network infrastructure. An interest in wireless mesh networks is demonstrated by the existence of several proprietary solutions and the creation of workgroups for its standardization at wireless LAN, PAN and MAN level (respectively IEEE 802.11s, 802.15.5 and 802.16/802.16j). One of the main problems of multi-hop wireless networks is the lack of scalability of the routing protocols, which limits the maximum number of nodes. One solution for improving scalability by reducing routing complexity is the formation of a virtual *backbone* in the radio network, so that only the nodes that compose the backbone perform routing and forwarding functions. Among the advantages of using a virtual backbone for routing in the radio network we can highlight the following: The routing function in the network becomes concentrated within a small subset of nodes or sub network. The routing load is then limited to a subgroup of nodes and, unless changes in topology affect this subset it is not necessary to recalculate the routing tables. In proactive routing only the nodes in the backbone need to maintain routing information. In reactive routing, the search space of a route is reduced to the set of nodes in the backbone. The cost of disseminating routing information is also reduced. The use of hierarchical routing, reduces these costs in terms of bandwidth and power

consumption. It increases network scalability due to the reduction of the stored routing information through node reduction and hierarchy and the reduction of required processing, since not all the nodes of the network perform routing functions. It also reduces radio interference. Only a small subset of nodes participates in the communication path, for example [1].

This article compares the performance of several well known algorithms for backbone construction both theoretically and through simulation using the Dominating Set Simulation Suite (DSSS) [2]. We present the results of simulations of the following algorithms: MBN-TSA, tree, mesh, span, Minimum Connected Dominating Set (MCDS), marking process (rules 1&2 and rule k) and k -connected dominant set (CDS) k -CDS (k -gossip, KCC, CBCC-I and CBCC-II). Whilst the k -CDS focuses on reliability of the backbone, the rest of algorithms focus on obtaining minimum size of the backbone. The main parameters compared through simulation are: backbone size, number of backbone neighbours, average shortest path through backbone, number of state changes, locality and tolerance to node failure.

Similar previous works have been published recently. [3] and [4] compare different backbone formation algorithms for wireless sensor networks. The study specially the algorithm impact on per node energy consumption. Our work focus on generic wireless networks and we do not consider these parameter, however we study more generic algorithms. [5] and [6] propose new algorithms for constructing and maintaining dynamic backbone. They compare these new algorithms with some previous ones, nonetheless our works consider more algorithms.

The article is organized as follows: in Section II we introduce Mesh Networks; in section III Connected Dominating Sets are described. Section IV contains a summary description of the backbone formation algorithms compared, providing more detail of the MBN-TSA algorithm. Section V describes the evaluation of algorithms and the simulations performed. Section VI contains the conclusions and next, the acknowledgements.

2. Mesh Networks

The concept of the wireless mesh network (WMN) appeared years ago as a practical way to implement Mobile Ad-hoc networks (MANET) in applications that do not require mobility of all nodes. This new type of radio network is not considered to be an isolated network type, but rather an extension of the fixed networks with which they are interconnected, to extend their reach simply and effectively [7][8]. They are self configuring and dynamically self organizing networks, such that the nodes set up and maintain connectivity among them through the network mesh. In order to optimize the operation of the WMNs it is necessary to modify or develop new protocols, from the physical layer to the application layer [9].

WMNs consist of two types of nodes: routers and clients. Mesh routers can be implemented as dedicated equipment or on general purpose equipment (for example a PC). Mesh clients may also perform routing functions but they do not act as gateways. Examples of clients include PCs, PDAs, and IP phones. The dominant architecture in WMNs is called the Backbone Infrastructure (see figure 1). Mesh routers form an infrastructure or backbone to provide client connectivity. Some routers may also act as gateways, providing Internet connectivity to the mesh nodes, or integrating them with other radio networks. In other architectures like the hybrid architecture known as WMN client, mesh clients may perform routing and mesh routers may or may not exist.

The main characteristics of a WMN are the following:

- Multi-hop. WMNs are wireless multiple hop networks. Unlike conventional 802.11 networks, the paths in the wireless network have more than one hop. This multi-hop feature is needed to extend the network coverage without reduction of communication bandwidth and also to provide connectivity between users not directly attached.
- Ad-hoc configuration and deployment. Due to their flexible architecture, facility of deployment and configuration, fault tolerant capabilities and mesh connectivity, the WMNs do not require a very large initial investment and can be deployed gradually according to need.
- Limited mobility. Clients may or may not move but routers stay fixed.
- Power consumption constraints. Power requirements depend on the type of node. Routers usually do not have strict requirements on power consumption, whereas clients would normally need power efficient protocols.
- Compatibility and interoperability with existing radio networks. As an example, a WMN based on 802.11 technologies is compatible with 802.11 standards, i.e. with conventional WiFi clients. But WMNs also need to be compatible with other radio networks such as WiMAX, and ZigBee. The standardization of WMNs is currently undergoing at the IEEE group 802.11s. Other groups in charge of the standardization of mesh networks exist for other technologies. The IEEE 802.15.5 works on mesh networks based on Wireless Personal Area Networks (WPAN). Also a group named Mesh Ad-hoc Committee investigates the advantages of mesh networks using WiMAX (IEEE 802.16).

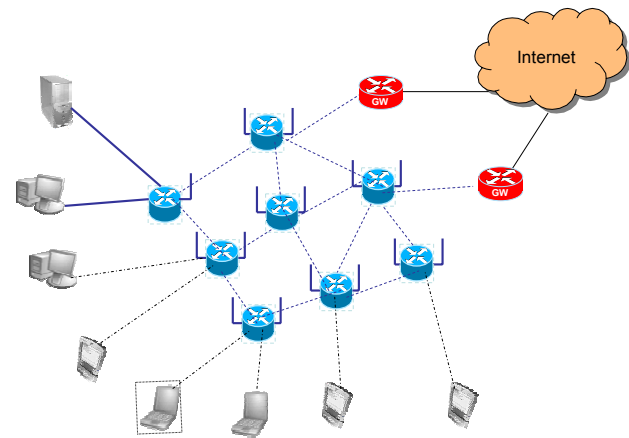


Figure 1. WMN (backbone infrastructure).

3. Connected Dominating Sets

Both ad hoc and mesh wireless networks can be represented by a *unit disk graph* [10] where every host (*vertex*) is associated with a disk centered at this vertex and a radius determined by the transmission range. In this context, two hosts are neighbours if they are covered by each other disk. In wireless networks some links may be unidirectional due to differences in the transmission power of hosts so a general disk graph with both unidirectional and bidirectional link is proposed to represent a wireless network [11].

3.1. Graph Theory

Non directed graphs are those that only have bidirectional links, in which each line represents a connection in both directions. A *dominant set* is a subset of the graph vertices such that every node that do not belongs to this subset is the neighbour of at least one node that belongs to the subset. *Directed graphs* are those in which unidirectional connections do exist. A unidirectional connection is represented by the ordered pair of vertices (u, v) indicating that a connection exists from u to v . In this case one says that u dominates v and that v is an *absorbent* of u [11].

Finally, the *degree* of a vertex of a graph is defined as the number of nodes with which it is connected directly, i.e. its neighbours.

3.2. Classification of CDS formation algorithms

Algorithms for CDS formation can be divided into two categories: centralized and distributed. Centralized algorithms rely on network wide knowledge and/or coordination. They normally produce smaller CDS than decentralized algorithms, but its high maintenance cost. An example is the MCDS algorithm explained below [12].

Decentralized (localized) algorithms only depend on neighbourhood information [13], [14], [15].

Decentralized algorithms can also be classified into Size-efficient (the CDSs built have constant efficiency rate but bigger convergence time and must operate as a sequence that

requires synchronization. An example is the algorithm described in [13]) or Time-efficient (the CDSs built by this type of algorithms do not get a constant efficiency rate, but converge in constant time. Examples are the algorithms presented in [11] and [16])

3.3. CDS based routing

Routing based on CDS is of a hierarchical nature; only the nodes that belong to the virtual backbone carry out the functions of information capture, mobility management, routing and packet forwarding. Examples of this routing type are Destination Sequenced Distance Vector (DSDV) protocol, Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distant Vector (AODV).

4. Backbone Formation Algorithms

In this section we provide a brief review of the algorithms for the formation of virtual backbones in radio networks that we have compared. The first six algorithms were initially designed for ad hoc networks, while the last two have been recently specifically proposed for wireless mesh networks.

4.1. Tree algorithms

These algorithms, described in [13] are based on the distributed formation of a maximum independent set (MIS). An *independent set* (IS) of a graph $G(V, E)$ is a subset in which no pair of nodes is adjacent. An MIS is an IS in which any other node is adjacent to a node that belongs to the MIS. It can be proved that any MIS is a Dominant Set (DS), and that any independent DS constitutes an MIS. So the formation of an MIS is a first step for the CDS formation. Additionally, the size of an MIS of an undirected graph is at the most 5 times greater than the size of the minimum dominant set (MDS).

4.2. Minimum CDS

The next two algorithms were proposed in [12]. The first algorithm builds a CDS from the formation of a tree T , initiated at the node of greatest degree. In every step, a node of T is chosen and connections to T are added from this node to all of its neighbours not included in T . At the end of the process, a spanning tree T is built and all the nodes that are not leaf nodes constitute the CDS. The complexity of this algorithm resides in selecting a good candidate node in each step. A modified version of the algorithm selects a pair of adjacent nodes. In this way in each step a node or a pair of nodes is chosen, depending on which option provides a greater number of coloured nodes. The second algorithm, works in two phases: first, a dominant set is formed and then the components of the DS are connected. At the beginning all the nodes are coloured as white. Whenever a new node is included in the dominant set, it is coloured as black and its adjacent nodes, dominated by it, are coloured as grey. In each step the node that provides the maximum reduction in the number of elements (either white nodes or component connected black nodes) is chosen. In the second phase pairs of black components are interconnected by selecting a chain of two nodes, until only a black component remains. The CDS is formed by the set of black nodes which are members of the connected component.

4.3. Span

The Span algorithm, proposed in [15], uses a similar approach to the marking process. Its main objective is to form a CDS (named *coordinator* nodes) so that the other nodes can switch to an energy saving mode without affecting the routing capacity of the network. A node is selected as a *coordinator* if it has two neighbours not directly connected or connected via one or two coordinators. Before becoming a coordinator, the node must wait for a certain time. This waiting period is calculated from its level of energy and the topology of its two-hop neighbour nodes. This delay can be seen as a priority level that causes that the nodes with smaller delay to have more possibilities of becoming coordinators. A modified version of the algorithm, based on a 3-hop neighborhood knowledge, is implemented in the simulator.

4.4. Mesh algorithm

The distributed backbone formation by means of the Link Cluster Algorithm (LCA), also known as the *mesh* algorithm, was proposed in [14]. The LCA algorithm is executed in two steps: cluster formation and union of clusters.

4.5. Marking Process

The marking process described in [16], is an algorithm, in which the nodes only interact with their neighbours to form a CDS. The resulting CDS of the marking process does not generally match the MCDS. There are many ways to reduce the size of the CDS generated by the marking process. One of these is mentioned in [17].

4.6. k-Connected k-Dominating Set (KCDS)

Three algorithms are proposed in [18] for the formation of a backbone from a *k-connected k-dominating* set. Unlike previously described algorithms whose primary target was to form the minimum CDS, its primary focus is to improve the tolerance to failure. In a wireless ad hoc network this last aspect is very important because a node can easily fail due to battery exhaustion or a connection can vanish temporarily during the movement of a node.

The first algorithm (*k-Gossip*) extends the Gossip based probabilistic approach [19].

The second algorithm (*k-coverage* condition) extends the coverage condition shown in [14], if a node is to be removed from the backbone, all its neighbours must be *k*-connected with each other via higher priority nodes.

Finally, a new hybrid algorithm called colour based *k*-CDS construction (CBKC) is proposed. It first makes a random partition of the network into several sub networks of different colours, and then a traditional CDS algorithm is applied to them. This algorithm is applied in [18] to the *k*-coverage condition in two ways.

4.7. MBN-TSA algorithm

This distributed algorithm defined in [20] for wireless mesh networks is based on the concept of a Mobile Backbone Network Topology Synthesis Algorithm (MBN-TSA). This architecture, initially introduced for wireless ad hoc networks in [21] and [22], consists of a hierarchical packet routing

architecture. Synchronization between network nodes is not required; each node is free to maintain its own clock. The schema is shown in the figure 2.

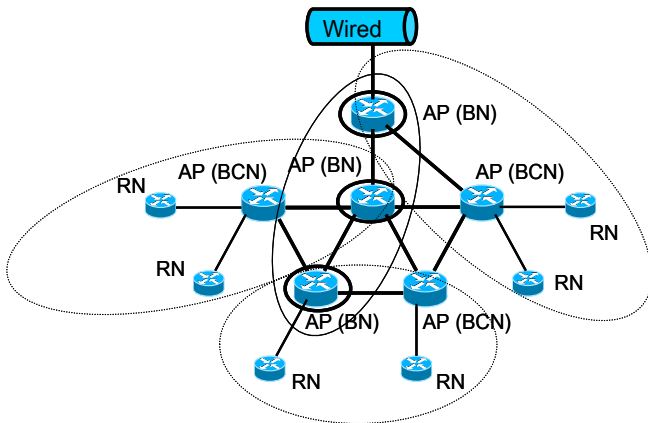


Figure 2. MBN Architecture of a mesh network [19].

The algorithm presented/displayed in [20] assumes that all nodes are equipped with a single radio and that they operate in the same frequency band.

5. Performance evaluation of the backbone formation algorithms

We have performed a comparative simulations of the algorithms described above. The parameters usually considered to measure the efficiency of CDS construction algorithm are the following:

- *Efficiency rate*. The ratio of the CDS size (number of nodes) calculated by the algorithm and the minimum CDS size of the graph.
 - *Computational complexity*. The complexity of the processing required at each node.
 - *Time complexity*. The number of iterations required by the algorithm.
 - *Communication overhead*. The size and number of interchanged messages during the CDS construction process.
- The objective of CDS construction algorithms is a low efficiency rate (small size of CDS) combined with low communication overhead and low computational complexity for backbone formation and maintenance.

5.1. Simulation test bed

In the simulations, we compare the size of the backbone generated by each of the algorithms, the average number of neighbours of a backbone node, the mean shortest path distance between two network nodes through the backbone and the backbone tolerance to a backbone node failure (i.e. whether the backbone remains connected and whether the rest of the network nodes still have access to a backbone node). We have also considered the influence of topological changes in the network, such as switching on or switching off of a backbone node and the distance to which these changes propagate in the network (the locality property).

To perform the simulations we have used the Dominating Set Simulation Suit (DSSS) tool available at [2]. We extended DSSS to simulate the MBN-TSA algorithm and included it in the comparisons. The simulations have been performed in batch mode.

We use two different network scenarios to limit the time of the simulation runs in some simulation types, (i) n nodes (ranging from 100 to 1000 in steps of 100), randomly placed in a 1500 x 1500 unit region and a transmission range of 300 units and, (ii) n nodes (ranging from 100 to 300 in steps of 25), randomly placed in a 1000 x 1000 unit region and a transmission range of 250 units. In any case, the networks simulated are static and fully connected (there are no isolated nodes), all the nodes have routing capabilities and all the links are considered bidirectional (non directed graphs).

Figure 3 shows a sample random network of 100 nodes deployed over a 1500 x 1500 unit area.

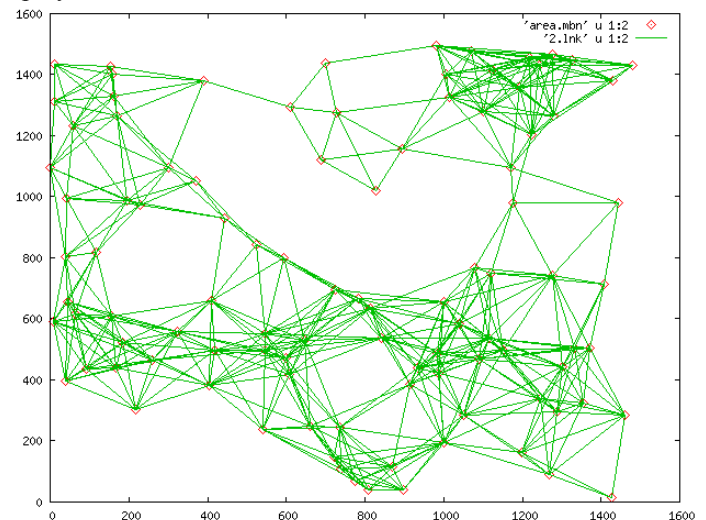


Figure 3. Sample network generated by DSSS

We present simulations results of the following algorithms: MBN-TSA, Tree, Mesh, Span, MCDS, Marking process (rules 1&2 and rule k) and k -CDS (k -gossip, KCC, CBCC-I and CBCC-II). We used the level based approach for cluster head selection in both tree and span. We use restricted implementations of the marking process (rules 1&2 and rule k). Finally we chose a value of $k = 2$ for the k -CDS algorithms ($pk = 50\%$ in k -gossip).

5.2. Simulation results

The results presented below are mean values obtained from simulation runs of up to 500 or 1000 randomly generated static radio networks. We guarantee a confidence interval below 10% or 1% with a 90% confidence level.

5.2.1. Backbone size

Figures 4 and 5 show the backbone sizes generated by the algorithms versus number of network nodes. The best performing algorithm is MCDS because it is centralized and obtains a good approximation to the minimum CDS. Next, MBN-TSA and Tree algorithms obtain good results and keep

fairly constant the backbone size when the number of nodes increases.

Figure 6 shows the comparison of the four KCDS algorithms with the MBN algorithm. These algorithms aim to a different main objective than the above mentioned algorithms. KCDS looks for some redundancy in the backbone to obtain higher tolerance to failures and more flexible routing, whilst the previous algorithms look for the smallest size backbone possible.

If the connectivity objective of simulated networks is $k=2$, then any pair of network nodes must be connected by at least two disjoint paths (i.e. not sharing any node). This is the reason that the backbone size is more than two times that required for MBN-TSA. CBCC-2 algorithm produces the smallest backbone size of this group in the low density networks and KCC in the more dense networks. CBCC-2, KCC and CBCC-1 keep the backbone size fairly constant when the number of nodes increases. The backbone size with k-gossip algorithm increases with increasing network size.

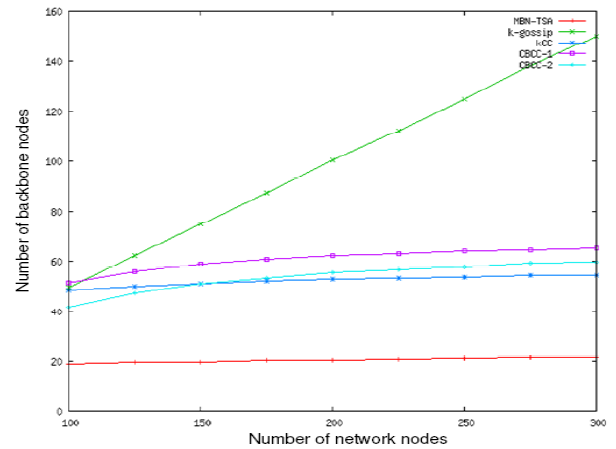


Figure 6. Backbone size MBN-TSA vs. KCDS

5.2.2. Average number of backbone neighbours

The average number of neighbours of any network node that belong to the backbone has also been simulated. As expected, the algorithms with the least neighbour nodes belonging to the backbone are MCDS, Tree and MBN-TSA (as seen in 5.3.1, they produce smaller backbones). For MCDS, the average number of neighbours is 2. Tree algorithm varies within 2,5 and 3,5 while MBN-TSA algorithm varies within 3 and 4. The algorithms with the highest number of neighbours belonging to backbone are Marking Process alone or with Rules 1 and 2. With Rules 1 and 2 the number of neighbours varies from 4 to 16. With Rule 1 only varies from 6 to 84 neighbours. Among the KCDS algorithms, the algorithm that generates the highest number of neighbours is the k-gossip.

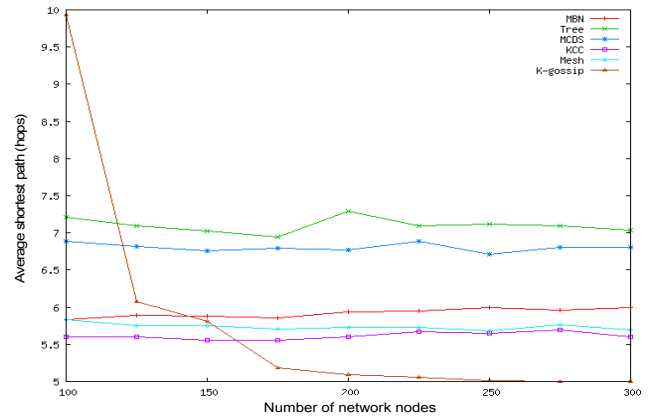


Figure 7. Average shortest path (i)

5.2.3. Average shortest path

The average shortest paths distance through the backbone has been compared. It is measured in terms number of hops and correlates with the backbone size and structure. Figs 8 and 9 show the results. Algorithms like MCDS and MBN-TSA present a higher minimum shortest path than other algorithms which build larger backbones, such as Span and Rules 1 and 2. Algorithms with lower shortest path are the Marking process with Rule 1 and 2 and the Span algorithm.

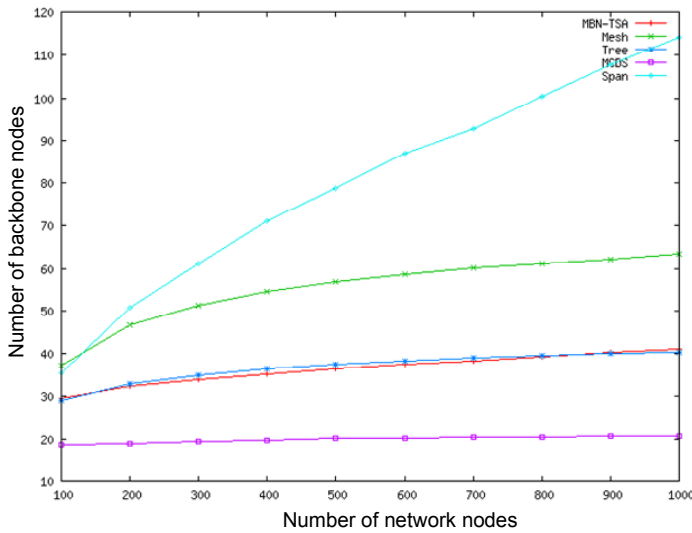


Figure 4. Comparison of backbone size (i).

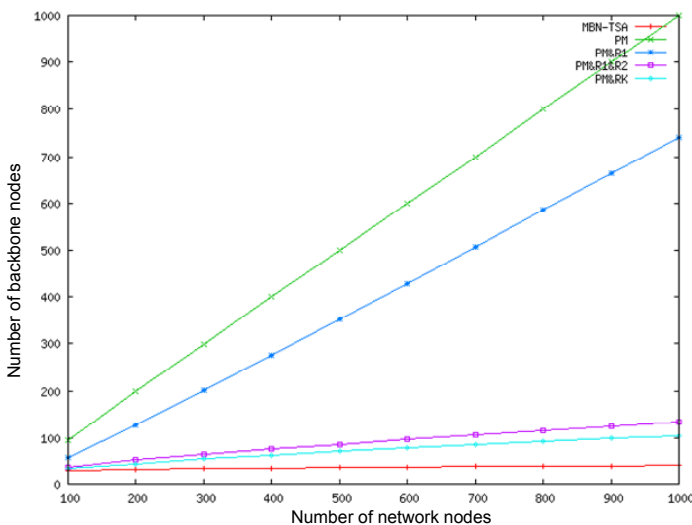


Figure 5. Comparison of backbone size (ii).

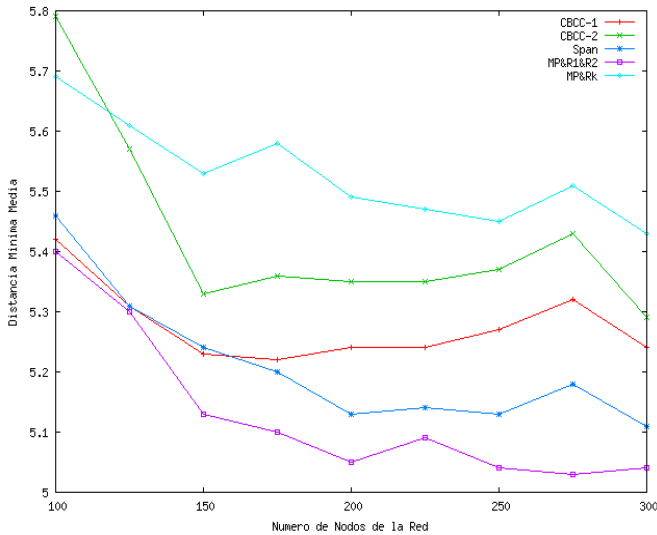


Figure 8. Average shortest path (ii).

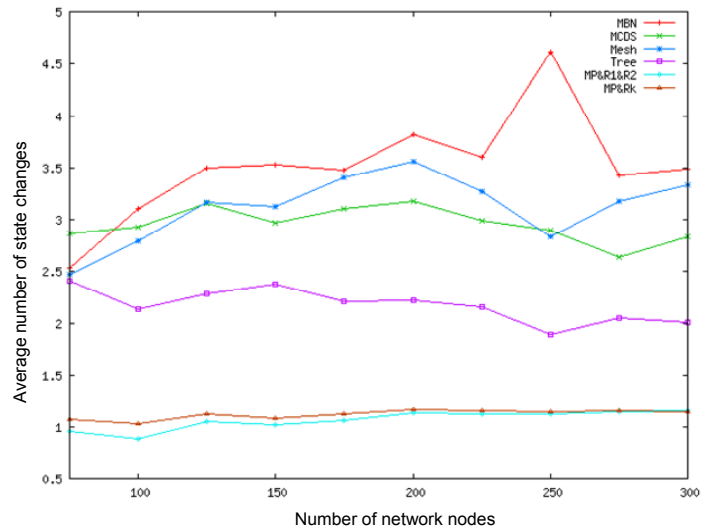


Figure 10. Average number of nodes changing state after switch on of a node.

5.2.4. Power on/off of a node

The topology of WMNs and MANETs may change even in the absence of mobility. A node may switch off for some time to save energy. A new node may join the network. In this section we evaluate the changes which occurred in the backbone by comparison of the backbone before and after the topological changes, counting the nodes affected by the change, i.e. the nodes belonging to the backbone that leave the backbone and vice versa. Only the MBN, MCDS, Mesh, Tree, Rules 1 and 2 and Rule k algorithms are compared. Figure 9 shows the comparison for the power on of a new node. MBN-TSA algorithm exhibits the highest number of state changes, increasing with the number of nodes (2,5 to 4) . MP&R1&R2 have the lowest values, approx. 1 node change. Fig. 10 shows the state changes after switch off of a node. Algorithm behaviour is similar to the power on case: MBN-TSA exhibits the highest number of state changes.

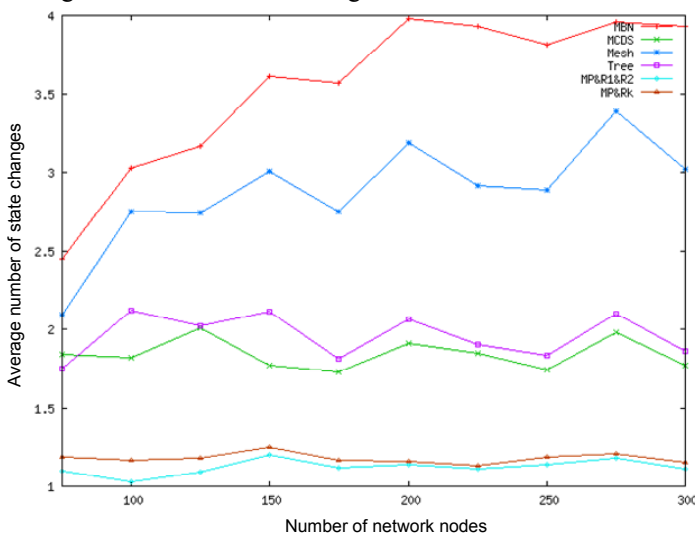


Figure 9. Average number of nodes changing state after switch on of a node.

5.2.5. Tolerance to a backbone node failure

Here we compare the resiliency of algorithms to a backbone node failure. The connectivity of all nodes after a backbone node failure is checked, and that the nodes previously connected through the failed node can connect via other backbone nodes without waiting for the rebuilding of the backbone. This is closely related with the redundancy of the generated backbone and the existence of multiple paths between nodes. Figures 11 and 12 shows the percentage of backbone failures after a single backbone node failure. MCDS and Span exhibits the greatest failure percentage.

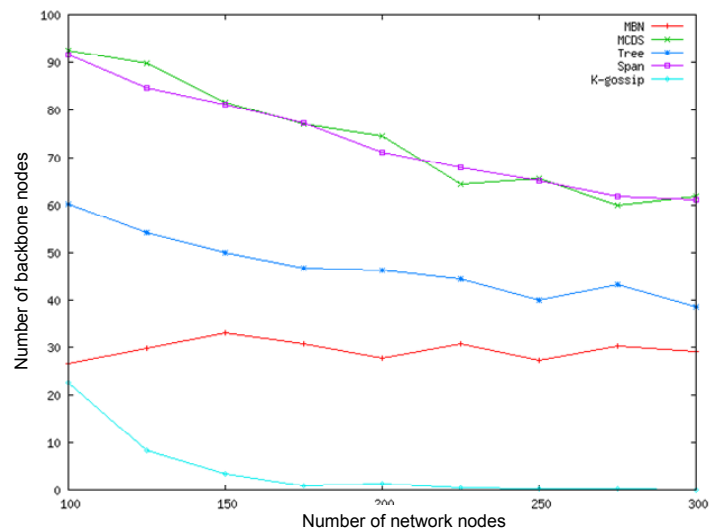


Figure 11. Comparison of percentage of backbone failure (i).

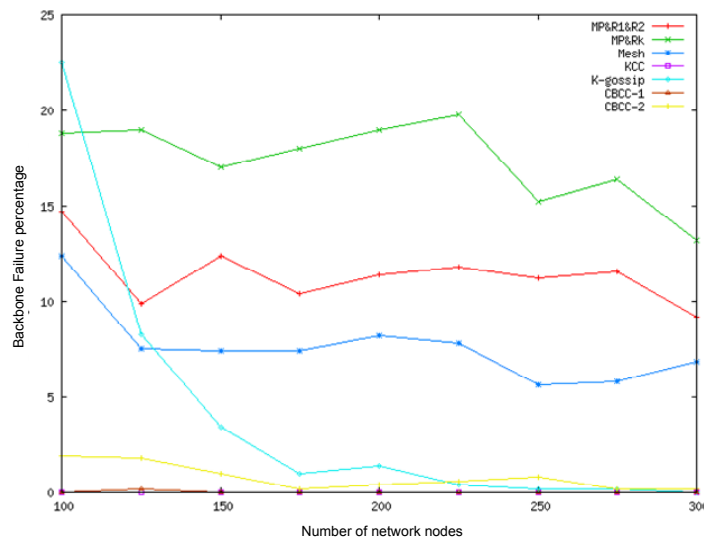


Figure 12. Percentage of backbone failure (ii).

Algorithm	Mean	1 hop (%)	2 hops (%)	3 hops (%)
MBN-TSA	4,28	29,21	65,42	83,64
MCDS	2,41	22,82	49,79	68,88
Mesh	3,20	30,00	67,19	90,63
Tree	1,97	28,43	64,47	88,32
MP R1&R2	1,17	58,97	100,00	100,00
MP Rk	1,21	58,68	100,00	100,00

Table 1. Propagation of topology changes (locality).

5.2.6. Locality

Table 1 shows the results for the locality property. It shows the distance (hops) to which a failure propagates changes and the accumulated percentage of node changes covered by a 1-hop, 2-hops and 3-hops distance.

5.3. Results analysis

We now perform a comparison of the algorithms. We use the results obtained from the simulations made with DSSS and the theoretical limits of operation of each algorithm extracted from the algorithm's documentation. The algorithms compared are: MBN-TSA, MCDS, Mesh, Tree, Span, Marking process with rules 1 & 2, and rule k , and KCDS algorithms. The ideal case happens when the limits for backbone size, computational complexity, the communication cost communication and the time complexity are constant and as low as possible. The more an algorithm approaches this ideal case, the more scalable it is. This means that the network size may increase without affecting negatively the algorithm's performance.

The algorithm which provides the best behaviour is MBN-TSA, since its limits are constant, which makes it highly scalable. The operation of the other algorithms can be classified according to the parameter considered the most important. Thus, when the objective is to obtain a constant backbone size, even at the expense of increased network resources consumption, it is advisable to chose algorithms like MCDS,

tree, KCC and CBCC. When the key point is to keep the processing capability required in the nodes constant, or when the consumption of network resources is to be minimized, the k -gossip algorithm is the best option. However it is necessary to remind that the backbone size generated by k -gossip increases with the number of nodes and that, being a probabilistic algorithm, it does not always generate a backbone KCDS. Finally, if we focus on the time complexity either, k -gossip and rules 1, 2 and k converge in a constant number of rounds. In networks with a high level of reliability, it is more advisable to use algorithms like MCDS and tree that create smaller backbones sizes and therefore reduce the load in the network due to routing. In networks in which there are more frequent failures it is better to use the KCDS algorithms, which provide path redundancy in the backbone. The density of nodes of the network should a key factor to choose a KCDS algorithm or an alternative, since the success rates of both options vary according to this parameter. The algorithms with a good locality property, like the marking process with rules 1 and 2, and rule k , may be suitable for networks in which the topological changes are frequent. Whenever a node is switched off or switched on, it is necessary to recalculate the backbone. If these changes affect only a few nodes next to the affected node, the computational load and communication overhead will be smaller. In table 2, we summarise the best and worse performing algorithm according to several parameters.

A detailed comparison of several algorithms for wireless backbone formation has been performed through simulation, adapting and extending the DSSS simulator with the simulation of the MBN-TSA algorithm and additional simulations. The MBN-TSA algorithm offers the best overall operation among the compared. On one hand, it exhibits constant time complexity, backbone size and communication cost, even if the density of nodes of the network increases. Additionally, the size of the backbone generated is relatively small and its tolerance to node failures is acceptable. Its greatest weakness is the locality property. The Algorithm selection can be seen as a trade off between backbone size and reliability. In the case of unreliable networks, algorithms such as k -CDS compensate using a greater backbone size in order to provide greater redundancy and ensure connectivity.

Parameter	Best performance	Worst performance
Backbone size	MCDS, tree, MBN-TSA	k -gossip, PM&R1&R2, span
Average number of backbone neighbours	MCDS, tree, MBN-TSA	k -gossip, PM&R1&R2, span
Average shortest path through BB	PM&R1&R2, span, k -gossip	Tree, MCDS, MBN-TSA
Number of state changes	PM&R1&R2, PM&Rk	MBN-TSA, mesh, MCDS
Locality	PM&R1&R2, PM&Rk	MCDS, MBN-TSA, mesh
Tolerance to one node failure	KCC, CBCC-1, CBCC-2, k -gossip	MCDS, span, tree

Table 2. Comparative table of the algorithms simulated in DSSS

6. Conclusions

A detailed comparison of several algorithms for wireless backbone formation has been performed through simulation, adapting and extending the DSSS simulator with the simulation of the MBN-TSA algorithm and additional simulations. The MBN-TSA algorithm offers the best overall operation among the compared.

7. Acknowledgements

This work has been partially supported by the Spanish Ministerio de Educación y Ciencia through project CAPITAL (TEC2004-05622-C04-03/TCM) and by Comunidad de Madrid E-Magerit (S-0505/TIC/000251) project. Thanks to Matthew Hutton who revised the manuscript.

References

- [1]. V. Barghavan B. Das. "Routing in ad-hoc networks using minimum connected dominating sets," IEEE International Conference on Communications (ICC), Montreal, Quebec, Canada, 8-12 June 1997, pp 376-380.
- [2]. WRSS. Fei Dai. June 2001. Available at (Internet): <<http://sourceforge.net/projects/wrss>> , (March 2006)
- [3]. S. Basagni, M. Mastrogiovanni, and C. Petrioli, "A Performance Comparison of Protocols for Clustering and Backbone Formation in Large Scale Ad Hoc Networks", in Proceedings of IEEE MASS 2004.
- [4]. S. Basagni, M. Mastrogiovanni, and A. Panconesi, "Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison," IEEE Transactions on Parallel Distributed Systems 2006.
- [5]. Y. Wang, W. Wang, and X. Li, "Distributed low-cost backbone formation for wireless ad hoc networks," in Proceedings of ACM MobiHoc 2005,
- [6]. H. Ju, and I. Rubin, "Performance analysis and enhancement for backbone based wireless mobile ad hoc networks", in Proceedings of IEEE BroadNets 2005.
- [7]. Ari, N. Jethami, A. Rangnekar, S. Natarajan, "Performance Analysis and Comparison of Ad Hoc Routing Protocols", CMSC 691T, MOBILE COMPUTING, PROJECT REPORT, University of Maryland Baltimore County. May 2000.
- [8]. R. Bruno, M. Conti y E. Gregori, "Mesh Networks: Commodity Multihop Ad Hoc Networks", IEEE Communications Magazine, pp. 123-131. March 2005.
- [9]. F. Akyildiz, X. Wang y Weilin Wang, "Wireless Mesh Networks: a Survey", Computer Networks, Elsevier, vol. 47, pp. 445-487. January 2005.
- [10]. B. N. Clark, C. J. Colbourn, and D.S. Johnson, "Unit Disk Graphs", Discrete Math., vol. 86, pp. 165-177, 1990.
- [11]. F. Dai y J. Wu, "An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 15, no. 10, pp. 908-920. October 2004.
- [12]. S. Guha y S. Khuller, "Approximation Algorithms for Connected Dominating Sets", Algorithmica, vol. 20, n. 4, pp. 374-387. April 1998.
- [13]. Khaled M. Alzoubi, Peng-Jun Wan y Ophir Frieder, "Distributed Heuristics for Connected Dominating Sets in Wireless Ad Hoc Networks", Journal of Communications and Networks, vol. 4, pp. 22-29. March 2002.
- [14]. D. Baker y A. Ephremides, "The Design and Simulation of a Mobile Radio Network with Distributed Control", IEEE Journal on Selected Areas in Communications, vol. SAC-2, no. 1. January 1984.
- [15]. B. Chen, K. Jamieson, H. Balakrishnan, y R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", ACM Wireless Networks J., vol. 8, no.5, pp. 481-494. September 2002.
- [16]. J. Wu y H. Li, "On calculating Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks", Proceedings Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm., pp. 7-14. 1999.
- [17]. J. Wu, "Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links", IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 9, pp. 866-881. September 2002.
- [18]. F. Dai y J. Wu, "On Constructing k-Connected K-Dominating Set in Wireless Networks", Proceedings of the 19th International Parallel and Distributed Processing Symposium (IPDPS 2005). April 2005.
- [19]. X. Hou y D. Tipper, "Gossip-based Sleep Protocol (GSP) for energy efficient routing in wireless ad hoc networks", Proceedings of IEEE Wireless Communications and Networking Conference. March 2004.
- [20]. H. Ju, I. Rubin, "Mesh Topology Construction for Interconnected Wireless LANs", Proc. IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON). September 2005.
- [21]. H. Ju, I. Rubin, K. Ni y C. Wu, "A Distributed Mobile Backbone Formation Algorithm for Wireless Ad Hoc Networks", Proceedings of IEEE International Conference on Broadband Networks (BroadNets), pp. 661-670. 2004
- [22]. H. Ju, I. Rubin, A. Behzad, R. Zhang, X. Huang, Y. Liu y R. Khalaf, "Ad Hoc Wireless Networks with Mobile Backbones", Proceedings of IEEE International Symposium on Personal, Indoor and Radio Communications (PIMRC), vol. 1, pp. 566-573. 2004.
- [23]. H. Ju, I. Rubin, "Backbone Topology Synthesis for Multi-Radio Meshed Wireless LANs", Proceedings IEEE Infocom. 2006.