Topology Stability-based Clustering for Wireless Mesh Networks

Martin Krebs, André Stein, Mónica Alejandra Lora
Department of Computer Science, Informatik 4
RWTH Aachen University, Germany
{krebs, stein, lora}@cs.rwth-aachen.de

Abstract—**In the past, many clustering algorithms for ad-hoc networks have been proposed. Their main objective is to solve the scalability issue of ad-hoc networks by grouping nodes into clusters. The challenge in MANETs for those clustering algorithms is to cope with the high node mobility which affects the stability of the cluster structures. Wireless mesh networks consist of a static backbone and a number of mobile nodes. In the backbone of a wireless mesh network the topology is relatively static. However, topology changes occur due to frequent link losses and temporary link instability. Due to the static nature of the backbone, mobility-based approaches are not suitable in this case. In this paper, we state the important aspects for stable clustering in wireless mesh networks based on the investigation of a 45-node wireless mesh testbed. We analyze well-known clustering algorithms and their performance in a real world large-scale environment. Finally, we propose a new clustering algorithm called *Stable Link Clustering Algorithm (SLCA)*.**

I. Introduction

Wireless mesh networks (WMNs) [1] are a very popular technology for scenarios where the installation of cables is not possible or too expensive. Possible application scenarios are open community networks, provider networks or wireless campus networks. Because of their flat hierarchy wireless mesh networks have the same problems as MANETs where scalability is a big challenge when the number of nodes becomes large. Therefore, clustering is a widely accepted approach for MANETs to achieve scalability by dynamically grouping the network into sub-groups called clusters. Each cluster consists of one clusterhead, a number of cluster members and gateway nodes which connect two clusters.

Currently, there are a number of clustering schemes which are optimized for a certain type of wireless network and scenario [2]. Clustering schemes for MANETs are based on mobility aspects and are thus not suited well for wireless mesh networks with a static backbone. A clustering scheme for wireless mesh networks should take into account the heterogeneity of node types and most importantly identify a stable structure, i.e. reliable links and nodes. As this requirement cannot be achieved in one communication round, it is clear that such a clustering scheme can only increase its performance by a longer runtime. In this paper, we propose a new multi-hop clustering scheme called *Stable Link Clustering Algorithm (SLCA)* which takes into account the long-term stability of links and neighbor nodes. We also propose mechanisms to increase the clustering stability by reducing temporary topology fluctuations caused by link flapping.

The remainder of this paper is as follows: In Section II, we discuss existing and related works. In Section III we discuss modifications to existing clustering algorithms, followed by Section IV which presents our Stable Link Clustering Algorithm (SLCA). In Section V we discuss the results obtained from our testbed. Finally, Section VI comprises the conclusion.

II. Related Work

The first and most well known clustering schemes are Lowest-ID by Ephremides et al. [3] and Highest Connectivity by Parekh [4]. In the first scheme a node becomes clusterhead if it has the lowest identifier (ID), e.g. IP-address, in its radio range. The second scheme is similar, but uses instead of a static ID the current node degree, which changes over time because of changes of the neighbor connectivity. Although those approaches are quite old, the basic ideas are still used in recent works. Later Gerla and Tsai extended and compared those two approaches [5]. Those schemes are very popular in MANETs but are prone to topology stability issues and the ripple effect [2]. A so-called *ripple effect* happens when two clusterheads move in communication range where one clusterhead has to give up its role eventually. This change may result in a chain reaction and force other clusters into reorganization as well. Mitton et al. [6] proposed a density-based clustering scheme which takes into account the density of a node's neighborhood. In a more recent work, Xu and Wang [7] propose interesting topology stability approaches for mobile scenarios. However, for mesh networks new long-term stability metrics have to be defined to cope with link flapping caused by temporary interferences.

III. Clustering Algorithms

A. Model

In the following we present our model: Let $G = (V, E)$ be an undirected graph where V denotes the set of wireless mesh nodes with $|V| = n$ and E the set of wireless links. The open neighborhood of a vertex $v \in V$ is defined as $N(v) = \{u : u \neq v \wedge dist(u, v) \leq 1\}$. The (open) k-neighborhood of a vertex $v \in V$ is defined as $N_k(v) = \{u : u \neq v \wedge dist(u, v) \leq k\}$, correspondingly. The degree $deg(v) = |N(v)|$ is the number of edges incident to v. The k-degree $deg_k(v)$ of vertex v is defined as $|N_k(v)|$. The distance $dist(u, v)$ of two nodes u, v is defined as the minimum length of all paths from u to v. A set $D \subseteq V$ of vertices in a graph $G = (V, E)$ is called a *dominating set* if every vertex $v \in V$ is either an element of D (dominating node) or is adjacent to an element of D (dominated node). A subset $D \subseteq V$ is a *k-distance dominating set*, if for all nodes $v \in V$ holds that each node is either a dominating node or at least k-hops away from a dominating node. A dominating node is called *clusterhead*, a dominated node is called *cluster member*.

B. Investigated Clustering Schemes

We now extend the well-known clustering schemes Lowest-ID and highest connectivity to multi-hop clustering algorithms. Therefore, each node broadcasts a clustering message regularly (*Cluster_Message_Interval*) with a Time-To-Live of k hops. The status messages includes *score, hop count, k, originator-IP, message sequence number, validity time, status, number of 1-hop neighbors* and *selected clusterhead*. The score defines the degree which is announced by a node. Each node stores the k-neighbor information for three times the status message interval in its k-neighbor table. If an entry is not updated it is deleted from the neighbor table. The clustering schemes are event-driven based on changes in the k-neighborhood. A node selects a new clusterhead if its current clusterhead is no longer in the neighbor table or another node is better than the current one. In this paper, we use the k-distance dominating set variant for the clustering process. A node is either a clusterhead or has a clusterhead in the distance of k hops.

1) k-Lowest-ID: Each node selects the node with the lowest identifier (IP address) as its clusterhead. The ID is a static clustering criterion which does not change over time. With the Lowest-ID criterion, clustering can start immediately, because the ID is present on startup of a node.

2) k-Highest Connectivity: Each node announces its connectivity (degree) in its status message. In the next round, each node selects the node with the highest degree as clusterhead. This scheme needs one setup round in advance and is a dynamic criterion which changes over time due to topology changes and link flapping. When using the highest connectivity criterion for clusterhead selection, nodes in the center of the network are automatically preferred because of their higher connectivity degree and higher importance for the connectivity.

The link quality [8] is not a suitable metric to measure mid-term and long-term link stability. The performance of the link quality is dependent on the size of the measurement window. Whereas a too small window size reflects only a very short time, a too large window size does not consider current link quality changes. Therefore, we present a new metric called *connection rating* which is based on a penalty function which awards stable links and penalizes instable links. In the following, we present mechanisms to stabilize clustering algorithms by identifying stable structures in WMNs.

IV. The Stable Link Clustering Algorithm

We now propose our multi-hop clustering algorithm for wireless mesh networks called *Stable Link Clustering Algorithm (SLCA)*. It is based on a k-distance dominating set-based clustering scheme with highest connectivity and several

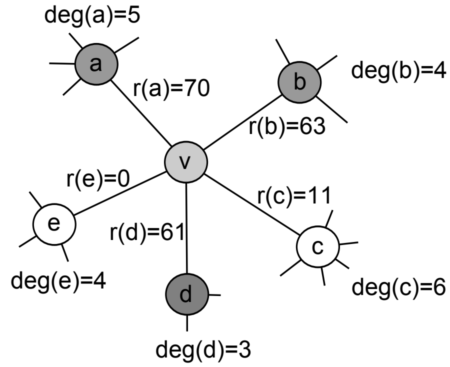

Fig. 1. Example topology

stability mechanisms. Each node broadcasts a clustering message regularly every *Cluster_Message_Interval* seconds with a Time-To-Live of k hops. Based on the received clustering messages, each node calculates for every neighbor node $v_j \in N_k(v)$ the connection rating $r(v_j) \in \{0, .., 100\}$. When a node receives a status message, the node calculates how many packets have been lost based on the status message packet sequence number. Let $loss_{t-1,t}$ be the number of lost status messages between two messages at points in time $t-1$ and t. The connection rating $r_t(v_j)$ of a neighbor node $v_j \in N_k(v)$ at time t is defined as follows:

$$r_t(v_j) = \begin{cases} r_{t-1}(v_j) + 1 & : \quad loss_{t-1,t} = 0 \\ r_{t-1}(v_j) - 2 \cdot loss_{t-1,t} & : \quad loss_{t-1,t} > 1 \end{cases} \tag{1}$$

The connection rating is a penalty function where a successfully received status message is awarded with one point and the loss of one or more packets is penalized with twice the number of lost packets. Although, counting successfully received packets is a similar idea to the link quality in OLSR with ETX-extension [8], the link quality is based on a limited-time window and only suited for short time predictions.

In the next step, each node executes the clustering algorithm based on the clusterhead candidate set $N_k^r(v) \subseteq N_k(v)$ which only takes into account nodes which are higher than a certain connection rating threshold.

The connection rating threshold is calculated based on the best connection rating in the neighbor table. Therefore, the function $f_{exp}(max_r)$ is applied to the highest rating max_r of all neighbor node ratings $r(v_1), ..., r(v_n)$. The result is a value which states the minimum required rating value a node must have to be included in the clusterhead candidate set. The idea of this function is that only the nodes with the highest rating, with respect to the best node, are considered in the clusterhead selection process. The following function calculates the required rating which a node must have to be included in the selection process.

$$f_{exp}(max_r) = \left\lfloor \left[1 - exp(-\frac{3 \cdot max_r}{100}) \right] \cdot max_r \right\rfloor \tag{2}$$

The idea of the exponential function in $f_{exp}(max_r)$ is that a large candidate set is generated if the rating of the best node is very small, e.g. at startup. If the rating of the best node is large, i.e. $max_r \rightarrow 100$, the candidate set is very small. For example, if the best node w has a rating of $r(w) = 100$, all other nodes must have a rating of at least 95 to be included in the stable node set. The exponential function is more suitable than a fixed value, because the difference is flexible for different rating values. The resulting candidate set is now:

$$N_k^r(v) = \{v_j \in N_k(v) | r(v_j) \geq f_{exp}(max_r)\} \quad (3)$$

The value $|N_k^r(v)|$ is used as new node degree, i. e. instead of using all nodes for calculating the node degree, the node degree is based on set of stable nodes. This reduces the temporary fluctuations of the broadcasted degree over time.

A. Candidate Flap and Loop Protection

In the next step, the clusterhead selection process with flap protection is applied. The candidate *flap protection* prevents that a node switches its clusterhead because of temporary link instabilities. Normally, a node changes its clusterhead if another node has a higher degree to ensure a small number of clusters. Now, a node is only allowed to switch to another clusterhead if $deg(CH_{new}) > deg(CH_{current}) + a$ where $deg(CH_{new})$ is the degree of the new candidate node CH_{new}, $deg(CH_{current})$ is the degree of the currently selected clusterhead and a is a positive integer value. Our testbed measurements show, that a value of $a = 2$ is a good trade-off to prevent unnecessary flapping. This mechanism only applies to changes to a better clusterhead. If the connection to the currently selected clusterhead is lost, a switch is allowed without restrictions.

In the next step, the *loop protection* prevents that a node oscillates between two clusterheads during consecutive election rounds which is caused by topology updates. It works as follows: when a node elects a clusterhead, it records a timestamp t_{prev} of the current time and the previous selected clusterhead CH_{prev}. During the next round, a change of the cluster head is only allowed if the new clusterhead candidate is not the old previous clusterhead CH_{prev} and the blocking time t_{loop} is expired. The blocking time t_{loop} during that a switch to the previous clusterhead is not allowed is set to six times the status message interval. When a switch is prevented, i.e. a node tries to elect the previously selected node, the blocking time t_{loop} is doubled.

B. Example

We now explain our algorithm by example but without flap and loop protection, see Fig. 1. Node v in the center of the figure has five neighbors, i. e. $N(v) = \{a, b, c, d, e\}$. Node a has the highest connection rating max_r with $r(a) = 70$. After applying Eq. 2 we get $f_{exp}(70) = 61.43$, which means that a neighbor node must have at least a connection rating of 61 or better to be included in the clusterhead candidate set. The clusterhead candidate set, based on Eq. 3, is now $N_k^r(v) = \{a, b, d\}$. The other two nodes c and e are not included in

Algorithm 1 Stable Link Clustering Algorithm (SLCA)

$CH_{curr} \leftarrow 0$ ▷ current cluster head
$CH_{prev} \leftarrow 0$ ▷ previous cluster head
$t_{prev} \leftarrow 0$ ▷ timestamp previous cluster head
$t_{loop} = 6 * Cluster\_Message\_Interval$
$a \geq 0$ ▷ parameter for candidate flap protection
while true **do**
    Send and receive clustering messages
    Calculate $r_t(v_j) \quad \forall v_j \in N_k(v)$
    $deg(v) \leftarrow |N_k^r(v)|$
    for all $v_j \in N_k^r(v)$ **do**
        ▷ candidate flap protection
        if $deg(v_j) > deg(CH_{curr}) + a$ **then**
            ▷ candidate loop protection
            if $v_j = CH_{prev}$ **and** $t_{prev} + t_{loop} > now()$ **then** ▷ Do not allow cluster head change
                $t_{loop} \leftarrow t_{loop} * 2$
            else ▷ Allow cluster head change
                $CH_{prev} \leftarrow CH_{curr}$
                $CH_{curr} \leftarrow v_j$
                $t_{prev} \leftarrow now()$
            end if
        end if
    end for
end while

the candidate set, because their connection rating is below 61. Thus, they are not considered as clusterhead candidates. The new degree for the black node v is now $deg(v) = 3$. Finally, node v selects node a as its new clusterhead, because a has the highest node degree of the nodes in $N_k^r(v)$.

V. Results

In this section we discuss the performance of SLCA and compare it with well-known clustering algorithms in our testbed. The algorithms have been implemented in C++ for the Linux Operating System on our wireless mesh routers. As routing daemon we use the popular OLSRD implementation in version 0.5.5 [8] with ETX configuration. For all measurements we use 45 static wireless mesh routers. Before we start to discuss the results, we present our wireless mesh testbed.

A. Testbed

The UMIC-Mesh testbed [9] is located at RWTH Aachen University, Germany. The testbed currently consists of 45 wireless mesh routers which are deployed at various offices inside three buildings of the Department of Computer Science. The department consists of one four- and two three-story buildings with one mesh router per office and in total at maximum three mesh routers per floor. The mesh routers are single board computers (SBC) based on the ALIX.2C2/3C2 (x86) board by PC Engines running on a minimal Ubuntu Linux. Each board has one 100 Mb/s Ethernet interface, two miniPCI slots, one RS232 serial port, and two USB ports. The routers have one 500 MHz AMD Geode LX800 CPU and

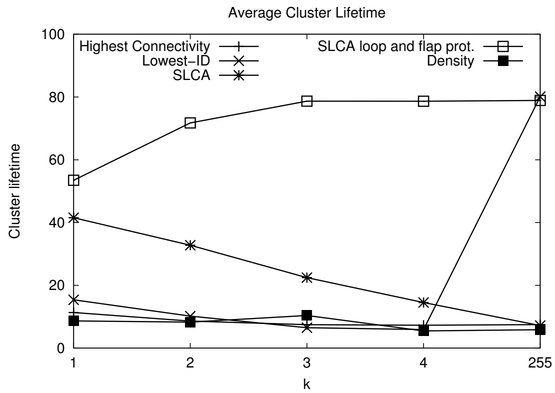

Fig. 2. Cluster lifetime

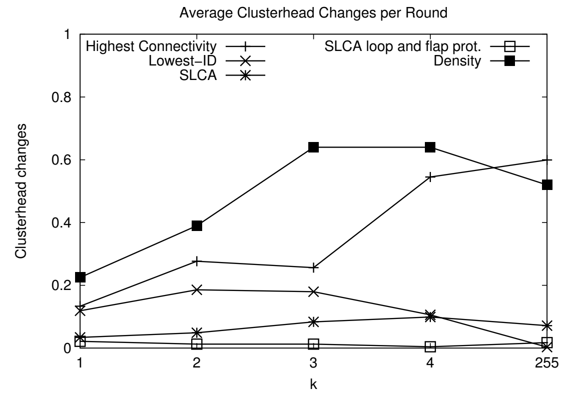

Fig. 3. Clusterhead changes per node per round

256 MB of DDR DRAM. Each router has two WLAN IEEE 802.11a/b/g interfaces which are based on Atheros AR5213 XR chips, and two omnidirectional antennas. The testbed topology is quite dense to achieve a high coverage over all floors. Most of the links are indoor links, but there are also many inter-building links.

B. Cluster Lifetime

In the first measurement, see Fig. 2, we investigate the average lifetime of a cluster. The cluster lifetime denotes the time from the point a node is elected as clusterhead until the point a node changes its status to normal node. Whereas in MANETs, the cluster lifetime is dependent on mobility issues, the cluster lifetime in WMNs depends on link stability. We use now 45 wireless mesh routers of our testbed and compare SLCA with the clustering algorithms Lowest-ID, Highest Connectivity and Density from [6] using different k distances. The total run time of all measurements is 700 clustering rounds. A clustering message is sent every 5 seconds. Thus, a neighbor node is kept in the neighbor table for 15 seconds and discarded if there is no further clustering message received.

The results show that Highest Connectivity and Density have the shortest cluster lifetime. SLCA without flap protection outperforms the existing clustering schemes, because the algorithm is based on a set with stable nodes only. When using flap and loop protection in SLCA, the cluster lifetime is increased even more. Both enhancements prevent successfully temporary flapping which leads to a larger overall cluster lifetime.

The measurements show, that the density-based approach does not work well in our testbed environment with temporary link flapping. This maybe the case, because the topology is very dense and a large fraction of links is instable. For our future work, a density clustering scheme, which operates on a stable link set, may be investigated in more detail.

C. Clusterhead Changes

We now investigate the average number of clusterhead changes of a node during one clustering interval, i.e. the time between sending two status messages. During this time each node receives status messages from its neighbor nodes and decides whether to change its clusterhead because of a better candidate than the one currently selected or not. Thus, the clustering scheme is event-driven and based on changes in the neighborhood. A low number of clusterhead changes denotes a stable clustering scheme whereas a high number of clusterhead changes denotes an instable scheme.

Fig. 3 shows that our proposed SLCA scheme causes the lowest number of clusterhead changes. Especially when using Density and Highest Connectivity clustering, a node changes its clusterhead very often. Both schemes are based on topology-related information like node degree. Lowest-ID performs better than both schemes regarding clusterhead changes, because the clustering criterion is static (ID) and is not affected by changes in connectivity. Our proposed SLCA scheme outperforms all other existing clustering schemes regarding clusterhead changes. However, SLCA with loop and flap protection have the best performance and cause only minimal cluster changes.

D. Number of clusters over time

In this measurement we investigate the number of clusters over time. Therefore we use 45 mesh routers and perform 700 clustering rounds with different k distances. In general, the number of clusters decrease with a higher k, because every clusterhead dominates a larger area. Highest Connectivity, see Fig. 5, generates fewer clusters than Lowest-ID, see Fig. 4, because the Highest Connectivity clustering scheme prefers nodes as clusterhead with a high node degree which cover a large number of neighbor nodes. However, both schemes have a very high fluctuation of clusters over time due to instable links. This results in many re-clustering situations.

Our proposed SLCA scheme generates more clusters than both previous schemes, because the number of clusterhead candidate nodes is smaller, as only nodes with the best connection rating are considered. From the point of stability, SLCA is more stable over time because of fewer re-clustering situations. By comparing SLCA, see Fig. 6, and SLCA with candidate loop and flap protection of 2, see Fig. 7, it can be

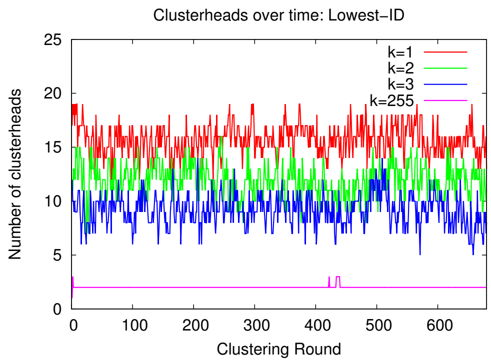

Fig. 4. Lowest-ID: clusterheads over time

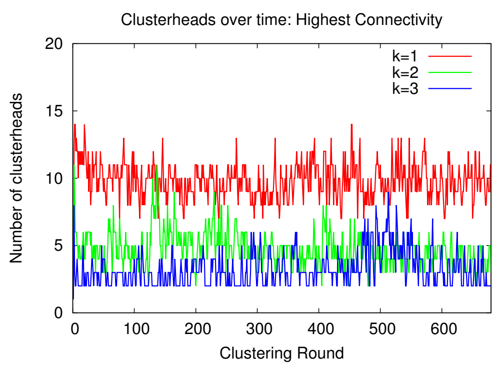

Fig. 5. Highest Connectivity: clusterheads over time

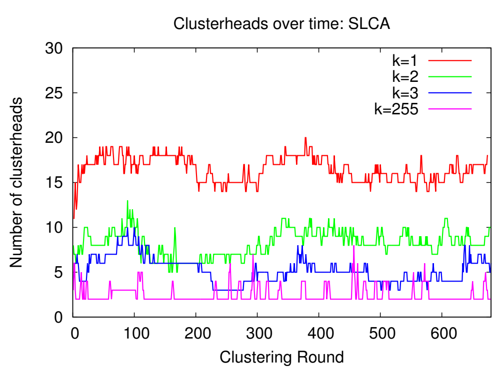

Fig. 6. SLCA: clusterheads over time

seen that the latter two optimizations have a large impact on the stability of the clustering scheme over time. There are still a few fluctuations in the number of clusterheads over time, especially in when using $k = 1$ and $k = 2$. Those issues cannot be optimized any further by our methods, because the links of a few nodes are too bad to achieve a stable cluster structure.

VI. Conclusion

In this paper we presented topology stability considerations for clustering in wireless mesh networks. We discussed several techniques to stabilize the clustering process to reduce

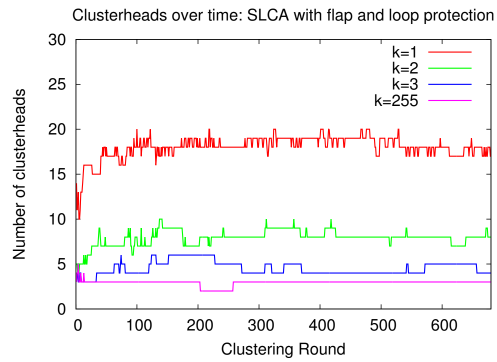

Fig. 7. SLCA with flap and loop protection: clusterheads over time

re-clustering situations which occur due to temporary link flapping. Furthermore, we presented a stable link clustering algorithm which is based on the Highest Connectivity criterion with topology stability optimizations. Finally, we analyzed our approach in our 45 node wireless mesh testbed and compared it with well-known existing clustering schemes. Our proposed approach outperforms those schemes regarding cluster lifetime and re-clustering situations over time.

Our results show, that two issues affect the stability of a clustering scheme: It is important to select a set of stable neighbor nodes, and furthermore it is necessary to prevent temporary flapping of the announced node degree, because an instable node degree may force other nodes in the neighborhood to change their clusterheads.

Acknowledgment

This research was funded in part by the DFG Cluster of Excellence on Ultra-high Speed Information and Communication (UMIC), German Research Foundation grant DFG EXC 89.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, March 2005.
[2] J. Yu and P. Chong, "A survey of clustering schemes for mobile ad hoc networks," *Communications Surveys & Tutorials, IEEE*, vol. 7, no. 1, pp. 32–48, Qtr. 2005.
[3] A. Ephremides, J. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, Januar 1987.
[4] A. K. Parekh, "Selecting Routers in Ad-hoc Wireless Networks," in *Proceedings SBT/IEEE Intl Telecommunications Symposium*, 1994, pp. 1–18.
[5] M. Gerla and J. T. chieh Tsai, "Multicluster, Mobile, Multimedia Radio Network," *Journal of Wireless Networks*, vol. 1, pp. 255–265, 1995.
[6] N. Mitton, A. Busson, and E. Fleury, "Self-Organization in Large Scale Ad Hoc Networks," in *Proc. The Third Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2004)*, Juni 2004, pp. 14–23.
[7] Y. Xu and W. Wang, "Topology Stability Analysis and Its Application in Hierarchical Mobile Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1546–1560, March 2009.
[8] "Hompage of Optimized Link State Routing Daemon (OLSRD)," 2010. [Online]. Available: http://www.olsr.org/
[9] UMIC-Mesh, "Hompage of the UMIC Wireless Mesh Testbed," 2010. [Online]. Available: http://www.umic-mesh.net/