Adaptive Routing in Wireless Sensor Networks for Fire Fighting

Chunlei An, Yunqi Luo, Andreas Timm-Giel

Adaptive Routing in Wireless Sensor Networks for Fire Fighting

Chunlei An, Yunqi Luo, and Andreas Timm-Giel

Institute of Communication Networks, Technical University of Hamburg,
Schwarzenbergstr. 95E, 21073 Hamburg, Germany
{chunlei.an,yunqi.luo,timm-giel}@tuhh.de
http://www.tuhh.de/et6/

Abstract. Fire fighters often work in dangerous and dynamic environments, which results in frequent change of network topologies and routing requirements. While the existing routing protocols are not able to cope with such a changeable environment, this paper proposes a self adaptive hybrid routing algorithm. This routing algorithm can switch between the proactive routing algorithm and reactive routing algorithm for each node pair automatically. An analytical model is created to describe the routing switch decision making algorithm. This model is based on a set of the cost functions. A numerical example shows the necessity of switching routing algorithms to reduce the overall control message overhead.

Keywords: Sensor Networks, Adaptive Routing, Hybrid Routing, Fire Fighting

1 Introduction

Wireless sensor networks play an increasingly relevant role in emergency and rescue scenarios. Nowadays fire fighters use different equipment for different functionalities. Each fire fighter needs one communication unit to keep contact with each other. This type of communication can be disturbed in noisy environments. Furthermore, each fire fighter also needs to carry a "dead man" alarm, which generates acoustic alarms when the fire fighter becomes incapacitated. One severe shortcoming of such a device is the limited alarming range. This means that only fire fighters who are close enough to hear the alarm can be informed about it. And it is also not reliable in noisy environments. In some cases the fire fighters have to risk their own safety for checking certain surroundings. This can happen when a fire fighter wants to open the door of a close room. Currently the fire fighters need to take off one of the gloves, and put the back of the hand close to the door for estimating the inner room temperature. This may be dangerous if the outside temperature is already high, or the fire fighter touches the door accidentally.

The GloveNet project [1] is funded by the German Federal Ministry of Eduction and Research (BMBF), and is targeting to solve the aforementioned problems. The main concept of this project is to explore the possibility of building

a WSN using intelligent gloves, which have compact sensor modules integrated. This module should provide alternatives to the functionalities mentioned before, so that the fire fighters can be better protected.

2 Problem Statement

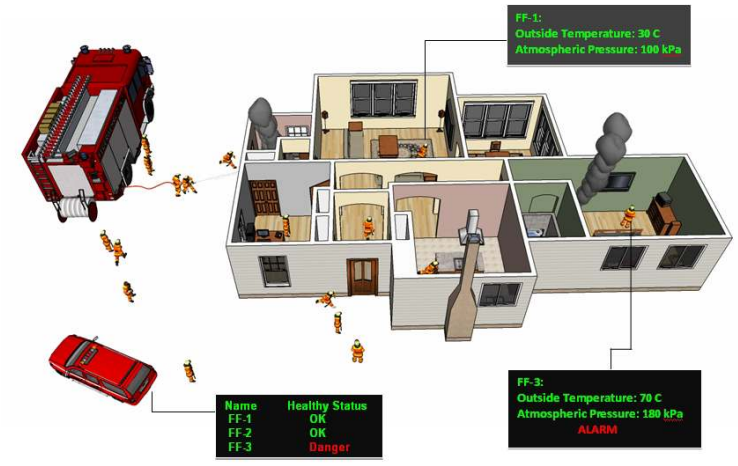

Fig. 1: Fire fighting scenario

In the fire fighter scenario in Fig. 1, there are several fire fighters in the rescue environment. The red car in Fig. 1 is present the commander for the hole fire fighter group, so all the information has to be transfer to the commander, which is also the sink node in Fig. 2. In the fire fighter group, they divided to two parts. One part of the group members runs into the building to rescue people and fight with the fire directly. The other part of the group members work outside of the building, preparing the water pipe and preparing themselves to go inside of the building to take place the group member who is tired. For the group members who work inside of the building, they have high mobility and also the link quality is not stable when the fire fighters run through from room to room. The group member who work outside of the building, their mobility is relatively low, and the link quality is higher. So the nodes in the left cloud in Fig. 2 presents the group of fire fighters who in operation inside of building. The other group is shown as the nodes in the middle cloud in Fig. 2.

Different routing schemes may fit different environments. For instance in Fig. 2, due to the frequent change of network topology, a reactive routing algorithm is preferred for the mobile nodes. On the contrary, a proactive routing scheme may be suitable for the static nodes and the sink node. Thereby a hybrid routing algorithm is expected. Moreover, tasks of the fire fighters can change, so when

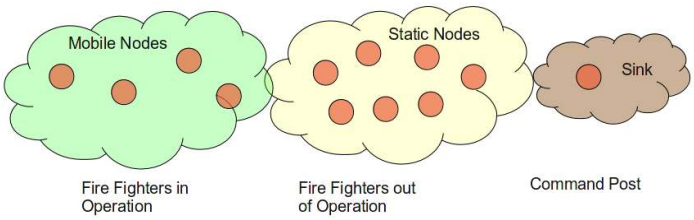

Fig. 2: Abstract network structure of the fire fighting scenario

some of the static nodes become part of the mobile nodes, their routing scheme should also change accordingly. This requires the adaptivity from the routing algorithm.

3 State of the Art

Researches towards adaptive and hybrid routing algorithms has been carried out in the recent years.

Figueiredo et al. present a hybrid and adaptive algorithm for routing in WSNs, called Multi-MAF,that adapts its behavior autonomously in response to the variation of network conditions [3]. In particular, the proposed algorithm applies both reactive and proactive strategies for routing infrastructure creation, and uses an event-detection estimation model to change between the strategies to save energy.

In [4] the authors propose a Programmable Routing Framework (PRF) that promotes the adaptability in routing services for WSNs. This framework includes a universal routing service and an automatic deployment service making use of different tunable parameters and programmable components. To change from one routing method to another, the proposed programmable routing framework must be reconfigured (by an operator), this means, it is not able to adjust its routing strategy according to the environmental change automatically.

The ChaMeLeon routing protocol [7] is a hybrid and adaptive routing protocol operating within a defined disaster area denoted as the Critical Area (CA). The main concept behind ChaMeLeon is the adaptability of its routing mechanisms towards changes in the physical and logical state of a MANET, e.g, the rescuers joining or leaving the network. ChaMeLeon adapts its routing behavior according to changes in the network size within a pre-defined CA. For small networks, ChaMeLeon routes data proactively using the Optimized Link State Routing (OLSR) protocol whereas for larger networks it utilizes the reactive Ad Hoc On Demand Distance Vector (AODV) Routing protocol so that overall routing performance is improved.

Another hybrid routing protocol called Adaptive Hybrid Domain Routing (AHDR) is proposed in [8]. AHDR organizes nodes within a 2-hop neighborhood into logical groups called Domains. Each domain has a Domain Lead. The

proactive routing scheme disseminates Domain topology information through the network with the help of Bridge Nodes – a subset of nodes that have links to nearby Domain Leads. The reactive routing scheme is used when a source AHDR node does not have a known route to a required destination. This scheme uses only a small subset of the network nodes carry the network routing messages through the network which reduces the AHDR overhead.

In [6] a hybrid routing protocol called Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network Protocol (APTEEN) is proposed, which allows comprehensive information retrieval. This protocol divides the nodes insides the network into different clusters. Different Code Division Multiple Access (CDMA) code is applied in each cluster to avoid inter-cluster interference, and inside each cluster the access to medium is controlled by the Time Division Multiple Access (TDMA) scheme. Furthermore, APTEEN combines the best features of proactive and reactive networks by creating a Hybrid network with that sends data periodically, as well as responds to sudden changes in attribute values. Performance evaluation shows that APTEEN outperforms existing protocols in terms of energy consumption and longevity of the network.

Protocols like PRF, ChaMeLeon, and AHDR are capable to adapt to different network communication situations, but require a thorough switch a routing algorithm in the whole network. Moreover, ChaMeLeon and AHDR are not designed for working on resource constraint devices. The routing protocol APTEEN is designed for resource constraint WSNs. Although it has the keyword adaptive included in its name, no support to adaptability has been explicitly described in the protocol. Due to this reason, it is not considered as an adaptive routing protocol here. Moreover, none of these routing protocols have been optimized for energy efficiency. Taking the project requirements and the literature study into account, a new routing protocol needs to be developed, which then can be combined with positioning for further improvement of the energy efficiency.

4 Self Adaptive Routing Algorithm

The new routing protocol in design should be more flexible to the change of the network topology, as well as to the data traffic characteristics. Considering the fact the different nodes, which locate at different part of the network, may have totally different environments, hence have different requirements to routing algorithms (as discussed in section 2). Therefore, routing algorithms can be chosen on node pair base. In other words, each individual inside the network is allowed to execute more than one routing algorithm simultaneously. For instance, node A can communicate with node B in a proactive manner, if they both agree that the link in between is stable. Meanwhile node A may set up a connection with node C using a reactive routing algorithm.

Two important techniques: dynamic neighbor update and mobility detection are investigated, in order to get up to date link status. The proposed self adaptive hybrid routing algorithm is then explained with an example afterwards.

4.1 Dynamic Neighbor Update

Dynamic neighbor update means that each node is aware of its immediate one-hop-neighbors at all times. To achieve this, all nodes are periodically sending out beacons. Based on the reception of these beacons, each node maintains a list of its direct neighbors.

Once a node detects a beacon from a previously unknown node, the receiving node will add the sending node to its own neighbor list. An entry in this dynamically created list contains the neighbor's address, the Received Signal Strength Indicator (RSSI) of the last received beacon, and a Time To Live (TTL) integer. The RSSI value is used for the mobility detection and the TTL value determines the lifetime of the connection as follows.

To detect the loss of a connection, a timer has been implemented, which is started periodically. Each time the timer expires, every entry of the neighbor list will be processed. First the TTL value will be decreased by one. If the TTL value is now equal to zero, the processing node will assume the connection to this node to be lost. It will therefore delete this entry from the neighbor list.

Every time a node receives the beacon of an already known neighbor it will search the according entry in the neighbor list and reset the TTL value to the default value. This will prevent this neighbor from timing out. Based on the above described method of maintaining a neighbor list, three parameters are considered critical for the duration of a connection: the TTL value, the amount of time it takes for the TTL timer to fire and the beacon sending frequency. These values have to be tuned so that a lost connection is detected as fast as possible, yet a few lost beacons should not result in a dropped connection.

4.2 Mobility Detection

Mobility Detection means that one node can detect if itself is moving or that other nodes are moving relatively to it. In this paper a method based on RSSI is implemented and tested. This method tracks the RSSI value of the nodes in the immediate neighborhood. This information is used to decide which nodes are moving relatively to the currently tracking node.

RSSI based Mobility Detection To detect if a neighbor is moving either towards or away from a node, the node uses the information from the neighbor list. It works in conjunction with the above described procedure. On reception of a packet the receiving node will check its neighbor list for the entry of the sender. If the sender is known, the RSSI value of the new packet will be compared to the previously saved value. Otherwise, it will be added to the list.

In the case that the RSSI value has decreased more than the specified threshold value, the neighbor will be assumed to be moving away. The TTL value for this neighbor will then be reduced, which effectively implies that the connection times out twice as fast. It has been chosen to halve the TTL value, but this has only been chosen for testing the concept and the value can probably be optimized further.

The parameters that influence the speed of a node movement detection by method are the frequency of sent beacons and the threshold value for the RSSI.

If the beacon frequency is too high, it could theoretically happen that the difference between any two consecutively measured RSSI values are always lower than the threshold, even if the node is moving. Yet this has not been observed in the simulations.

This method has been proven to work quite nicely in TinyOS Simulation (TOSSIM) [2] environment. The reduction of the connection timeout then reduced the packet loss in simulation scenarios with moving nodes by about 10%.

4.3 Self Adaptive Hybrid Routing

An example scenario (Figure 3) illustrates the situation where a self adaptive hybrid routing protocol can be applied. This network includes a proactive sub-network, which is composed of six nodes (n1 - n6). With the help of the techniques described in section 4.1 and 4.2, each individual within this sub-network identifies its direct adjacent neighbors as in a static status. Hereby proactive routing protocol, e.g., OLSR, is utilized for inter node communication. Other nodes are supposed to be moving arbitrarily, so they use a reactive routing algorithm to exchange information, say AODV.

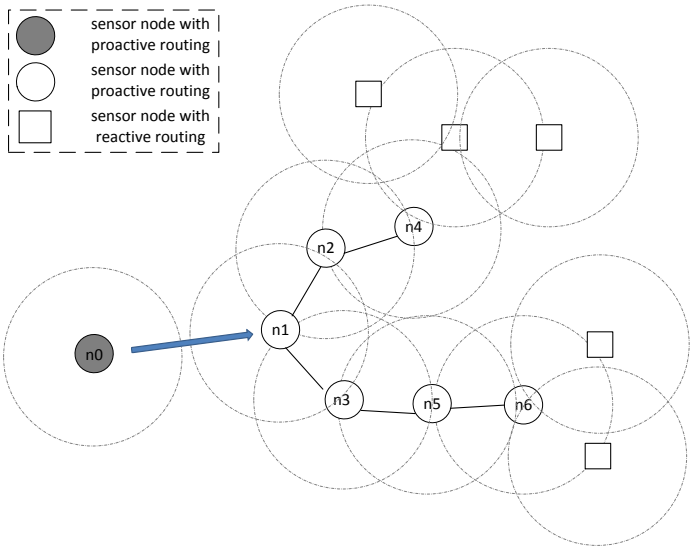

Fig. 3: Example Network Contains Nodes Using Both Proactive and Reactive Routing

The node n0 is now approaching n1, and is getting stabilized to the proactive sub-network. Thus according to the aforementioned techniques, n0 and n1 can take each other as a stable neighbor, and set up a proactive connectivity. Here arises the question: does this decision make sense?

The primary idea of switching between different routing algorithms is to improve the overall routing performance. In the fire fighting scenario, this means to improve the transmission efficiency, i.e., to reduce the amount of control overheads. A algorithm switching decision should be made only if this criteria is satisfied.

5 Analytical Model

Section 4 shows the necessity of making a reasonable routing algorithm switching decision. In the following subsections a preliminary analytical model is created with the aim to describe the decision making logic.

5.1 Routing Algorithm Switch Decision Making

A set of cost functions (Equation (1) - (5)) are defined to represent the basis of routing algorithm switch decision. The objective is to minimize the overall cost for a time period t. The cost is composed of two parts: the extra cost for switching to another routing scheme $Cost_{Switch}$ and the cost of using a routing scheme $Cost_{Algorithm}$ (Equation (2)). Equation (3) gives the definition of $Cost_{Switch}$. This cost is solely dependent on the number of overhead messages, whose output is scaled to the range $0 \leq Cost_{Switch} \leq 1$ by the correspondent scaling function f_s^{Switch}. The result is further weighted by a weighting factor α, which lies in the range $0 \leq \alpha \leq 1$. $Cost_{Algorithm}$ is defined in Equation (4). This cost depends on the current routing scheme over time period t. $f_s^{Algorithm}$ and β are the scaling function and the weighting factor, accordingly. The sum of α and β should be 1. Depending on different routing scheme, the value of α and β could be different.

As discussed in [10], different scaling functions can be chosen depending on different criterion value range. In this work it is assumed that the values of all the parameters are limited by their respective minimums and maximums. Therefore the general form of scaling functions f_s^{Switch} and $f_s^{Algorithm}$ are represented by linear functions given in Equation (5), where a and b are constants that are determined by the respective minimum and maximum.

$$Objective : min(Cost) \tag{1}$$

$$Cost = \alpha Cost_{Switch} + \beta \int_t Cost_{Algorithm} tdt \tag{2}$$

$$Cost_{Switch} = f_s^{Switch}(Overhead_{Switch}^{CtrMsg}) \tag{3}$$

$$Cost_{Algorithm} = f_s^{Algorithm}(Overhead_{Algorithm}^{CtrMsg}) \tag{4}$$

$$f_s(x) = ax + b; \qquad x_{min} < x < x_{max} \tag{5}$$

From Equation (1) - (5) it can be seen that the most important thing to decide the decision making cost is to determine $Overhead_{Switch}^{CtrMsg}$ and $Overhead_{Algorithm}^{CtrMsg}$.

In [9] control traffic overhead of different MANET routing protocols are studied. Routing protocols are classified as proactive and reactive routing protocols. The study shows that the control message overhead of different routing protocols is influenced by both network topology and the data traffic. A model is created (Equation (6) - (9)) to show the computation of the number of control messages under different circumstances.

$$Reactive\ Fixed: \qquad N_{rf} = \lambda O_r N^2 + h_r N \tag{6}$$

$$Reactive\ with\ Mobility: \qquad N_{rm} = O_r \mu \alpha L N^2 \tag{7}$$

$$Proactive\ Fixed: \qquad N_{pf} = h_p N + O_p t_p N^2 \tag{8}$$

$$Proactive\ with\ Mobility: \qquad N_{pm} = O_p \mu A N_p N^2 \tag{9}$$

Equation (6) shows the computation of the number of control messages N_{rf} when using a reactive routing algorithm, and all nodes are static. While Equation (7) describes how to calculate the amount of control messages caused by mobility. Similarly, Equation (8) - (9) are for proactive routing protocols.

The meaning of the variables is given in Table 1 - 2.

Table 1: Network and Traffic Parameters

(a) Network parameters

Network parameters	
N	number of nodes
μ	link breakage rate (mobility)
L	average length of a route

(b) Data traffic parameters

Traffic parameters	
λ	route creation rate per node
α	number of active routes per node (activity)

As discussed in Section 4, a switch of routing algorithm only affects the proactive sub-network. Therefore, $Overhead_{Switch}^{CtrMsg}$ solely depends on the characteristics of the proactive routing protocol. For instance, in OLSR only a node's Multipoint Relays (MPRs) are responsible to rebroadcast the according Topology Control (TC) messages, i.e., the number of control messages is equal to the number of this node's MPR N_{MPR} (see Equation 10). Selection of MPR

Table 2: Routing Protocol Parameters

(a) Proactive routing parameters

Proactive protocol parameters	
h_p	hello rate
t_p	topology broadcast rate
O_p	broadcast optimization factor
AN_p	active next hops ratio

(b) Reactive routing parameters

Reactive protocol parameters	
h_r	hello rate (0 when possible)
O_r	route request optimization factor

is closely related to the proactive sub-network's topology, which is difficult to estimate.

$$Overhead_{Switch}^{CtrMsg} = N_{MPR} \tag{10}$$

$Overhead_{Algorithm}^{CtrMsg}$ refers to the overall control message overhead, which consists of the control messages generated both when the node is static and mobile. Therefore, it can be expressed as in Equation (11)

$$\begin{gathered} Overhead_{Algorithm}^{CtrMsg} = N_{pf} + N_{pm} \\ or \\ Overhead_{Algorithm}^{CtrMsg} = N_{rf} + N_{rm} \end{gathered} \tag{11}$$

5.2 Numerical Example

In this section an example is given to demonstrate how should a algorithm switching decision be made.

Table 3: Parameter settings

Parameter settings			
λ	$a/60$	O_r	4
N	50	h_r	0
μ	1	a	1,1.2,1.4,...
L	2	h_p	0.5
O_p	0.13	t_p	0.25

Table 3 lists all the parameters and the according values. These values are taken from one of the simulation scenarios in [9].

Fig. 4 shows the relation between the traffic activity a and the routing algorithm cost $Cost_{Algorithm}$. In this case no scaling function and weighting factor

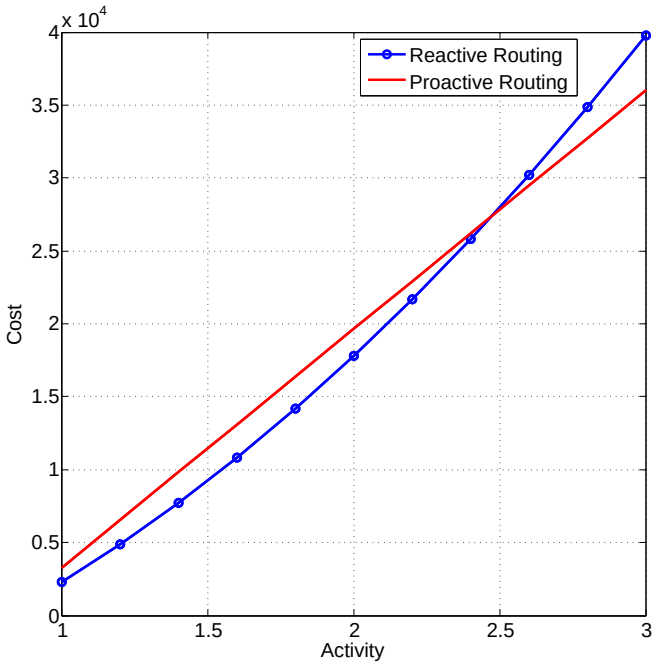

Fig. 4: Cost versus Activity

are used, therefore $Cost_{Algorithm}$ is identical to $Overhead^{CtrMsg}_{Algorithm}$. Results show that the proactive routing algorithm generates more control messages when the traffic activity is below 2.477 (the cross point of the two curves). The reactive routing protocol generates more control message overhead as soon as a goes above this value.

This figure also implies that for any given value of traffic activity a no algorithm switch should be done, if the control overhead caused by switch, $Cost_{Switch}$, is bigger than the gap between the two curves.

6 Conclusion and Future Work

6.1 Conclusion

It is shown in the previous sections, for some real WSN application scenarios, a routing scheme which supports both proactive and reactive routing is needed. This paper proposes a self adaptive hybrid routing algorithm, which can automatically switch between the proactive routing and reactive routing based on the current situation or that of the near future. A analytical model, which is based on a set of cost functions, is established to describe the decision making algorithm. The total cost consists of two parts, one part is the cost for switching routing algorithm, and another part is the cost for using a routing scheme. An

algorithm switch won't be performed, unless the overall cost after switching is going to be reduced.

For the future work, the proposed routing algorithm is to be evaluated in simulations.

6.2 Future Work

As mentioned in section 5.1, the control message overhead of performing a algorithm switch solely depends on the number of TC messages, which is needed to propagate the link state change to the whole proactive sub-network. This number is difficult to estimate, since as the RFC [5] describes, each node inside the proactive sub-network knows only the size of the sub-network, its direct neighbors and its according MPRs. The whole procedure can be eased, if each individual proactive node knows the overall amount of MPRs inside the network, which is exactly the number of broadcasting needed to disseminate the same TC message through the whole sub-network. It should be noticed that depending on their positions, different nodes may have different views to the whole sub-network, hence different sets of MPRs.

The RFC of OLSR [5] also indicates that the selection of MPRs is not optimized. The original idea is to guarantee that each node inside the proactive network is covered by at least one MPR. This leaves room to further reduce the number of MPRs, hence to reduce the required number of TC message transmissions. Researches have been carried out in this area, such as in [11] the authors propose a cooperative MPR selection algorithm. This is to be investigated in the future.

References

1. Glovenet project, `http://www.mrc-bremen.de/glovenet`
2. Tinyos tutorial: Tossim, `http://docs.tinyos.net/index.php/TOSSIM`
3. Figueiredo, S., C.M., Nakamura, F., E., Loureiro, F., A.A.: A hybrid adaptive routing algorithm for Event-Driven wireless sensor networks. In: Sensors. vol. 9, pp. 7287–7307 (Sep 2009)
4. He, Y., Raghavendra, C.S., Berson, S., Braden, B.: A programmable routing framework for autonomic sensor networks. In: PROC. AUTONOMIC COMPUTING WORKSHOP, FIFTH ANNUAL INTERNATIONAL WORKSHOP ON ACTIVE MIDDLEWARE SERVICES (AMS'03. pp. 60–68 (2003)
5. Jacquet, P.: Optimized link state routing protocol (OLSR). http://tools.ietf.org/html/rfc3626, `http://tools.ietf.org/html/rfc3626`
6. Manjeshwar, A., Agrawal, D.: APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. Parallel and Distributed Processing Symposium., Proceedings International, IPDPS pp. 195 –202 (2002)
7. Ramrekha, T., Panaousis, E., G.Millar, C.Politis: A hybrid and adaptive routing protocol for Emergency Situations . Request for Comments, Internet Engineering Task Force, IETF (Feb 2010)

8. Reza Ghanadan: Adaptive hybrid domain routing (AHDR) (2010), http://tools.ietf.org/html/draft-ghanadan-manet-ahdr-00
9. Viennot, L., Jacquet, P., Clausen, T.H.: Analyzing control traffic overhead versus mobility and data traffic activity in mobile Ad-Hoc network protocols. Wirel. Netw. 10(4), 447455 (Jul 2004), http://dx.doi.org/10.1023/B:WINE.0000028548.44719.fe
10. Wenning, B.: Context-Based Routing in Dynamic Networks. Vieweg and Teubner, 2010 edn. (Aug 2010)
11. Yamada, K., Itokawa, T., Kitasuka, T., Aritsugi, M.: Cooperative MPR selection to reduce topology control packets in OLSR. In: TENCON 2010 - 2010 IEEE Region 10 Conference. pp. 293 –298 (Nov 2010)