Directed Link Utilization with Mahalle+

Gerry Siegemund, Volker Turau
Institute of Telematics
Hamburg University of Technology
Email: {gerry.siegemund, turau}@tu-harburg.de

Stefan Lohs, Jörg Nolte
Distributed Systems \ Operating System Group
Brandenburg University of Technology Cottbus
Email: {slohs, jon}@informatik.tu-cottbus.de

***Abstract*—Self-stabilization provides non-masking fault tolerance in distributed systems. Self-stabilizing algorithms (SSA) are defined on the assumption that either the system's topology is fixed over time or topology changes are isolated events occurring at a very low rate. These assumptions are not valid in wireless sensor networks (WSNs) where link qualities change rapidly. Therefore, neighborhood management protocols (NMP) are used to ensure the stability of the network topology for a longer time period. Furthermore, symmetrical links between nodes are more desirable than unsymmetrical ones, therefore, unidirectional links are often omitted. This paper presents an augmentation of the NMP Mahalle$^+$ to transparently utilize certain unidirectional links to increase the performance of SSAs running on top of Mahalle$^+$.**

I. INTRODUCTION

Self-stabilizing distributed systems are guaranteed to converge to a desired state or behavior in finite time, regardless of the initial state. Convergence is guaranteed, i.e., after the system is affected by transient faults of unknown scale or nature, it will return to the desired behavior. Hence, self-stabilization is a powerful approach for non-masking fault-tolerance. The actions of each individual node of a self-stabilizing system lead to a global behavior possibly not known to each entity. A node can only evaluate its local view, i.e., its own state and that of its neighbors. Alterations of the neighborhood relation may force nodes to invoke rules leading to updates of their states possibly triggering more rule executions in the entire network. Ultimately, an algorithm running on top of a vigorously changing topology will not converge to a stable state. This situation occurs in WSNs where links frequently disintegrate while others are established regularly.

The goal of a NMP is to maintain a neighborhood relation such that the resulting topology is stable and fulfills certain criteria (first of all the connectedness of the induced graph). These protocols often store histories of link quality parameters leading to a considerable memory consumption. Since memory is a limited resource in WSNs each node maintains only few neighbors, resulting in new challenges. Agility and stability are two essential abilities of neighborhood protocols. Agility ensures that the protocol adjusts quickly to new or failing nodes (or links) respectively. On the other hand, transient faults, like burst errors, need to be ignored to keep the topology stable.

Application algorithms running on top of any NMP are influenced by the performance of it. The more stable the produced topology the better the overall performance. Many SSAs use the assumption of bidirectional links, i.e., each communication channel between two nodes always works in both ways. This assumption does not hold, according to [5] unsymmetrical links can make over half of all connections in a network. It has to be noted that the definition of a unidirectional link varies greatly. One can argue that any missed message in either direction can qualify for unidirectional communication. While others argue that 50 % message loss in either way might be expectable.

Our definition is as follows, if the link quality (defined by a link quality estimator) in either direction is below a threshold (Q) and in the other direction it is above, then the connection is not symmetrical. Q may be adjusted for application needs. Section III will offer more details on this matter.

On the other hand, self-stabilizing algorithms are not restricted to symmetrical links. In [6] Masuzawa et al. developed a unidirectional maximal independent set algorithm, nevertheless claiming that this tasked proved to be a difficult one. In addition, [1] defines certain bounds for self-stabilization in unidirectional networks, mainly for the vertex coloring problem.

Mahalle$^+$ [8] naturally favors symmetric links when building its neighbor list, but as long as there is space in the neighborhood list, also unidirectional links will be recorded. (Since WSNs are not synchronized from a node's perspective all connections are unsymmetrical at one point.) An application running on top of Mahalle$^+$ might only utilize the symmetrical links.

The contribution of this paper is an augmentation of the NMP Mahalle$^+$, to transparently utilize directed links to increase the number of available symmetric links for applications running on top of Mahalle$^+$.

II. RELATED WORK

Herman [3] introduced the first model for the usage of self-stabilizing algorithms in WSNs. A fixed topology with message loss and corruption is initiated, hence, there is no necessity for a NMP. His main contribution is the cached sensornet transformation (CST) where each node maintains a copy of the state of each neighbor (with respect to the fixed topology). Nodes periodically broadcast their state. They only execute an action when an uncorrupted message from each neighbor, since its last action, was received. Under the assumption that each message is received with a fixed probability and that message transmissions are independent events the system will eventually reach a legitimate state with probability 1 (see also [9]).

The intuitive way to set up a NMP is to add node ids of presumable neighbors into a list. Due to the restrictions of sensor nodes (e.g., low memory) the list will mostly have a constant size. Should the list be full either no new neighbor can be added or neighbors have to be removed first. Many compared NMPs, as well as the naive implementation, will not further update the list once it is full. Woo uses the following replacement schemes to keep the list of neighbors up to date: First in first out, least-recently heard, and frequency [10].

The *Link Estimation Exchange Protocol* (LEEP) [2] from TinyOS, uses a fixed neighborhood table size of 10. Every node sends out its current knowledge about all its neighbors. The eviction policy for a full neighborhood table always replaces the node with the least quality value of LEEPs link quality estimator. Hence, in dense networks, when there exist more than 10 neighbors with a high quality value, link changes occur permanently. The quality value and LEEPs broadcasting data can be used to determine if a node is a symmetrical one or not.

None of the above mentioned protocols utilize unidirectional links and we are not familiar with any NMP which takes care of such a case. In [4, pp.106-108] a neihborhood discovery algorithm for the Unidirectional Link Triangle Routing (ULTR) protocol is introduced. The basic idea of the routing algorithm is similar to our approach while the neighborhood discovery can not be understood as a NMP.

III. MAHALLE+ AND FORWARDED MESSAGES

This section illustrates the neighborhood algorithm Mahalle$^+$. It forms a stable network topology on top of a physical network despite transient faults. Mahalle$^+$ uses a link quality estimator, two lists, and a set of eviction rules to gather a number of good neighbors. The basic state diagram of Mahalle$^+$ is depicted in Figure 1.

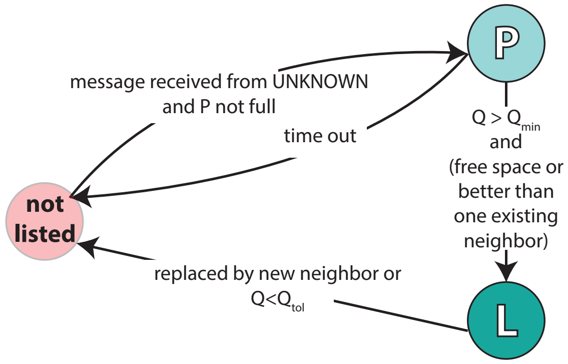

Fig. 1: Basic state diagram of the Mahalle$^+$ NMP (neighbor list L and preparation list P)

For a node to enter the preparation list P simply a message containing Mahalle$^+$ maintenance information needs to be received. Concurrently received messages stimulate the assessment of information about a neighboring node.

A	L	**A**	C		
	F	L	**A**	C	
	B	**A**	I		
	E	F	G	H	I

Fig. 2: Mahalle$^+$ example neighborhood view of a single Node A.

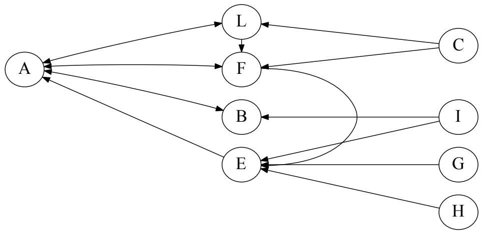

Fig. 3: Local view of a Node A

The link quality estimator values for a neighboring node need to exceed a certain threshold Q_{min} and stay above another threshold Q_{tol}, for a node to be at all considered as an actual neighbor. Any link estimator can be used, while we decided to use the very potent Holistic Packet Statistic (HoPS) [7]. HoPS produces four different estimations which all aid the channel evaluation.

If a nodes link quality exceeds Q_{min} it can be put onto the neighborlist L. L's size ($|L|$) is restricted, therefore, certain replacement policies are in place to keep the best possible neighbors [8].

The originating id of all messages travailing through the neighborhood layer is screened. Messages from unknown senders will not be passed to upper layers. Figure 2 shows a typical example Mahalle$^+$ neighborhood table of a Node A. The graphical representation of the local view of Node A can be seen in Figure 3. Note that the arrows point out the travel direction of messages.

In this example Node A and E are not symmetrical neighbors because Node E choose not to add Node A to its neighbor list, possibly due to a unidirectional communication channel. Mahalle$^+$ provides two-hop neighborhood information, therefore, Node A can logically compute that communication between Node A and Node E is possible, even with a unidirectional channel, if Node F forwards all messages from Node A.

Only slight modifications to Mahalle$^+$s implementation were necessary to find the relay node and change the message format to introduce the forwarding of messages. Firstly, a new message-type was introduced for *forwarding* and *forwarded* messages. All nodes that are supposed to forward a message are added at the end of such a Mahalle$^+$ maintenance message, including a field for the number of forwarding nodes. If the additional information does not fit into a maintenance message, then one, until now, unused flag in the message is set to

Tab. I: Mahalle$^+$ new message types

Message Type	Flag 1	Forwarding Ids
FORWARD	0	Number of and Forwarding IDs follow; containing own ID: forward message
FORWARD	1	Every node which receives this message may forward it
FORWARDED	*	Message was forwarded; has one extra field for preliminary sender ID

indicate that every node receiving this message shall forward it (Table I).

Mahalle$^+$ sends maintenance data in regular time intervals, it attaches its data either to application messages or sends out the information by itself. Forwarded messages need not to be acknowledged. They are just forwarded once by the respective node, which also handles the neighborhood data attached.

Nodes receiving relayed messages evaluate the data differently than regular messages. First, it might happen that a Node V accepts another Node F as a neighbor through relayed data, but the direct link between both nodes improves (due to some positive change in the environment). The forwarding of data is then just useless overhead. Therefore, Node F demanding the forwarding of its messages, should stop doing so. Furthermore, if messages, forwarded or not, are handled the same way, then the Node V will attach the information that Node F is a neighbor of V to the maintenance messages. If Node F receives such a message it computes that V and F are symmetric neighbors, therefore, it will stop to ask for relaying of its maintenance messages. That again will cause the *pseudo symmetric* link between both nodes to break, and causing a restart of the forwarding procedure. This would have the effect of destabilizing the whole topology, introducing a number of unwanted link changes.

Hence, a node in the neighbor list is either treated as a *direct* neighbor or as a *pseudo direct* neighbor. Nodes receiving both, direct messages of a node and forwarded messages of the same node treat them as two different nodes (direct: F_d, pseudo direct: F_p). Both nodes build up their link quality and both can be added to the preparation list (P). If F_d advances from P to L the maintenance message will include the id of F_d, hence, node F will stop asking for forwarding of its messages. That means that, the quality value of F_d will drop until the information is cleared from P. Should F_d advance to L then there will be no mention of this in the maintenance messages, i.e., the message flow of forwarded messages will continue.

The basic functionality of Mahalle$^+$ is not compromised by this approach. The eviction algorithm may choose to keep pseudo symmetrical neighbors or not. Their link quality is influenced by two connections (sender → relay → receiver), possibly resulting in increased message loss, this might make them less desirable as neighbors.

IV. EVALUATION

The evaluation was accomplished using simulations with OMNeT++ and the MiXiM framework, path-loss models and log-normal shadowing are used to influence the channel behavior.

For our approach to have any effect on the system, the build topology, and the dependent application we first assessed the occurrence of such possible pseudo symmetric communication patterns. In several simulations over differed topologies (quadratic, random, or line) we evaluated the number of symmetrical and pseudo symmetrical links in the emerged topology. Figure 4 shows these findings for different neighborhood table sizes and for the following test setup: A grid topology with 50 to 200 nodes, a varying neighborhood table size $|L|$ of 5 to 10 and an average degree of 11. The bottom bars represent the pseudo symmetrical neighbors and the top bars the symmetrical once respectively. The x-axis states the number links on average over all nodes found and utilized by Mahalle$^+$.

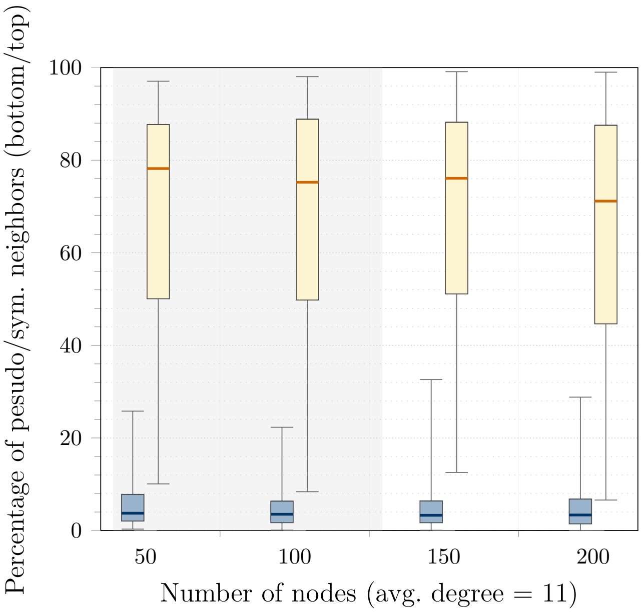

Fig. 4: Symmetric and possible pseudo symmetric neighbors; different neighborhood list sizes; without forwarding messages

We have found, that the higher the number of possible neighbors, i.e, the size of the neighbor table, the higher the number of pseudo symmetric neighbors. Even though, on average less than ten percent of all neighbors of a node are pseudo symmetrical, as can be seen in Figure 4, with 200 nodes and a maximum of 2000 edges, that still concerns roughly 100 connections. Occasionally , the number of pseudo symmetric nodes can even reach as high as forty percent.

After applying our approach of forwarding beneficial messages an increase of symmetrical neighbors, as depicted in Figure 5, can be found. With increasing degree our approach yields no increase in the average utilization of symmetric links, because the availability of symmetric links is already very high, i.e., with a degree much higher than the possible neighbors per node the number of symmetrical neighbors is sufficient.

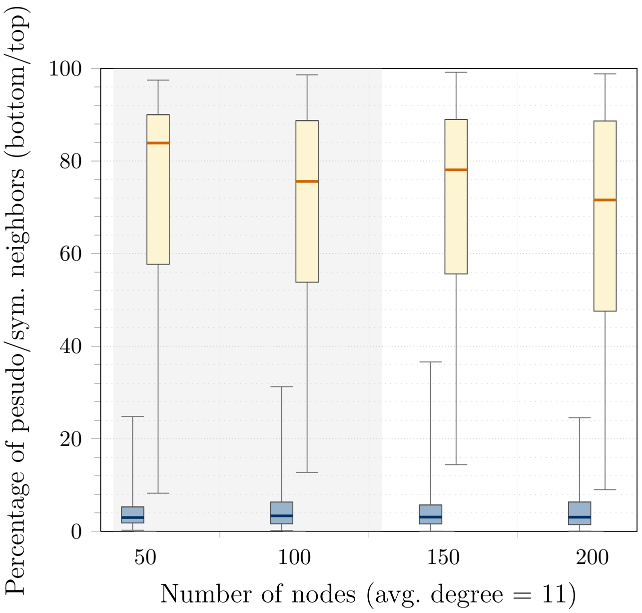

Fig. 5: Symmetric and possible pseudo symmetric neighbors; different neighborhood list sizes; with forwarding messages

Nevertheless, our approach increases the symmetrical links to be used by an application running on top of Mahalle$^+$, i.e., symmetrical plus pseudo symmetrical links. Therefore, e.g., tree construction can benefit. Short cuts trough pseudo symmetrical neighbors can shorten the overall tree depth. This, on the other hand, is very topology dependent, in simulation we found that this happens rarely, but the outcomes where never worse than with the original approach.

The presented figures only show the overall improvement of the build topology, which are quantified as a small increase of the average symmetrical links. In special cases, on the other hand, our approach might help to overcome bigger problems. Consider the node setup in Figure 6, with Mahalle$^+$ the network will not be symmetrically connected, with our augmentation it will be. (Note: The directed arrows point out the chosen neighbors, the message flow is vice versa; unlike Fig. 3).

Node 1 chooses Node 2 as a neighbor, i.e., it can receive messages from it. Node 3 chooses Node 2 and it is symmetrically connected to Node 1. Since there is no symmetrical connection between Sub-network I and II upper layer algorithms relying on these connections will not work correctly. Node 1 can compute that Node 2 did not choose it as a neighbor, possibly due to a directed channel, and that Node 3 can act as a relay Node. Again, Node 1 hears Node 2 and Node 2 hears Node 3, even though the relay node 3 does not hear Node 2, it will forward the messages from Node 1. The green arrow in the figure implicates the pseudo symmetrical connection between Node 1 and 2. In the same way also nodes with naturally few neighbors, like nodes on the outside or edge of a network, can benefit from our augmentation.

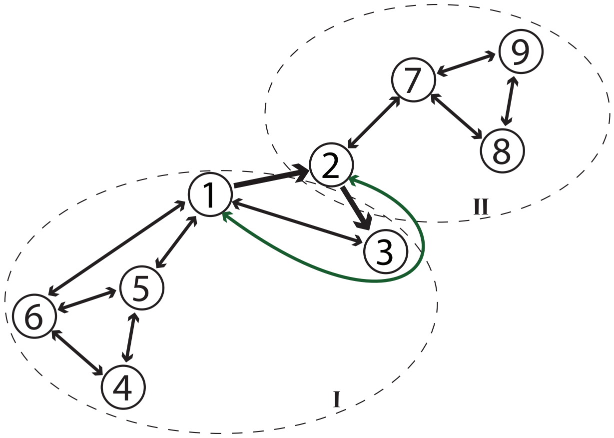

Fig. 6: Pseudo symmetric connection, Node 3 as a relay node

CONCLUSION

We presented an augmentation to the Mahalle$^+$ neighborhood protocol, to utilize a group of unidirectional links to increase the usefulness of Mahalle$^+$ for applications. Our approach does not interfere with the self-stabilizing abilities of Mahalle$^+$. Especially for networks with a low to medium average degree our approach is beneficial.

ACKNOWLEDGMENTS

This research was funded by the Deutsche Forschungsgemeinschaft (DFG), contract number TU 221/6-1.

REFERENCES

[1] Bernard, S., Devismes, S., Potop-Butucaru, M.G., Tixeuil, S.: Bounds for self-stabilization in unidirectional networks. arXiv preprint arXiv:0805.0851 (2008)
[2] Gnawali, O.: The link estimation exchange protocol (LEEP) (2007), TinyOS Extension Proposal (TEP)
[3] Herman, T.: Models of self-stabilization and sensor networks. In: Distributed Computing - IWDC 2003, LNCS, vol. 2918, pp. 205–214. Springer (2003)
[4] Karnapke, R.: Unidirectional Links in Wireless Sensor Networks. Ph.D. thesis, Universitätsbibliothek (2012)
[5] Lohs, S., Karnapke, R., Nolte, J.: Link stability in a wireless sensor network–an experimental study. In: Sensor Systems and Software, pp. 146–161. Springer (2012)
[6] Masuzawa, T., Tixeuil, S.: Stabilizing maximal independent set in unidirectional networks is hard. arXiv preprint arXiv:0903.3106 (2009)
[7] Renner, C., Ernst, S., Weyer, C., Turau, V.: Prediction accuracy of link-quality estimators. In: Proc. 8th Europ. Conf. on Wireless Sensor Networks (EWSN) (2011)
[8] Siegemund, G., Turau, V., Lohs, S., Karnapke, R., Nolte, J.: Agile and stable topology control for wireless sensor networks
[9] Turau, V., Weyer, C.: Fault tolerance in wireless sensor networks through self-stabilization. Int. J. of Com. Networks & Distribyuted Systems 2(1), 78–98 (2009)
[10] Woo, A.: A holistic approach to multihop routing in sensor networks. Ph.D. thesis, Berkeley, CA, USA (2004)