TPR: Dead End Aware Table less Position based Routing Scheme for Low Power Data-Centric Wireless Sensor Networks

S. A. Madani, D. Weber, S. Mahlknecht,
Institute of Computer Technology, Vienna University of Technology, Austria
{madani, weber, mahlknecht}@ict.tuwien.ac.at

Abstract— **In this paper we present a table less position based routing scheme for low power data centric wireless sensor networks. TPR is localized, uses greedy forwarding approach, and does not rely on neighborhood information. These characteristics reduces the communication overhead (no neighborhood information exchange), makes the protocol highly scalable (no routing tables are maintained and beacons are not exchanged when a node leaves or enters the networks), and performs better in mobile environments (as the next hop is non-deterministic and is computed at real time). It also deals with dead end problems by a recovery strategy in a distributed and localized way. TPR is implemented in the OMNET++ based discrete event simulation environment PAWiS. The results show that TPR provides guaranteed delivery, extended network life time, and a mechanism to route on the basis of end-to-end delay and/or energy consumption.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are characterized by restricted energy, processing power, and memory. New routing techniques which are energy efficient, memory efficient, and less computational complex are required. The routing protocols for wireless networks can be categorized into geographic and non-geographic [1]. Geographic routing protocols route the packets based on geographic location of the source, the next hop, and the destination. Position of the nodes can be obtained from low power GPS (Global Positioning System) receivers or relative coordinates can be found using techniques like incoming signal strength [2]. Position based routing schemes are highly scalable, exhibit robust behavior against frequent topological changes [3], can reduce/avoid communication and processing overhead caused by neighborhood information exchange, and can minimize memory usage by not maintaining routing tables.

Two main components of position based routing are *location services* and *forwarding strategy* [4]. Location services translate the identity of a node into its geographic position. In the many-to-one communication paradigm, the participating nodes must know the position of the sink node. Forwarding strategies can be based on minimizing the number of hop counts, geographic distance, delay and/or, energy consumption. In distance based greedy forwarding, a node always forwards the packet to the node with lesser Euclidean distance towards the sink node. If a message arrives at a node, which is shorter in distance than all its neighbor nodes and is not with in the communication range of the sink node, the node is known as *dead-end* (concave node) e.g., node F in Figure 1 and the associated strategy to recover from dead end problem is called recovery strategy. This paper deals with a dead end aware

Table less Position based Routing (TPR) scheme for data centric low power WSNs. The development of TPR is driven by a Wireless Container Monitoring System project [5] with WSNs. Each sensor node comprises a GPS, GSM/UMTS, and short range communication system. The availability of positions of nodes drives our motivation to develop a low power position based routing scheme for similar class of applications.

TPR is initiated by the sink node. The sink node informs all the other nodes regarding its current position by a broadcast message at fixed time intervals. This broadcast is also used to compute accumulative costs towards the sink node by the other nodes. As the nodes may be mobile (e.g., if containers are on a running train), each node save it's position obtained from GPS receiver at the same time intervals at which the sink node saves its position without strict time synchronization requirements. Now these positions are used as relative positions until the updated position of the sink node is broadcasted again. This is in contrast with other position based routing strategies which assume position of destination to be known.

Any node which wants to send/relay data packet sends it blindly without forwarder's address. All the nodes within the forwarding area (section IV, B) *compete* to become relay nodes for the transit packet based on *total cost* towards the sink node. The winning node *suppress* the other competing nodes wit adaptive transmit power. If a node is a dead-end, it re-routes the data packet based on the *recovery strategy* (section IV, E).

The paper is organized as follows. In Section II, an overview of the related work is given. Section III discusses TPR in detail. Simulation of the protocol is discussed in Section IV. The paper is concluded by highlighting future work and open research issues in Section V.

II. RELATED WORK

Many routing protocols have been proposed in the recent years [1-5]. Based on experimental studies [6], it is verified that reactive and proactive routing layers including AODV [7], DSDV [8] or DSR [9] that do no use location information and are based on exchange of routing information are not scalable. The overheads incurred by maintaining routing tables in such schemes due to mobility and topological changes is non-linear in networks while localized position based routing algorithms only need accurate neighborhood information and information about the sink node to provide scalable solutions [6]. In some cases like TPR, even the neighborhood information is not required. Most Forward within Radius (MFR) [10], forwards packets based on the notion of *progress*. Given a destination *D*, Source *S* forwards the packet to the neighbor which is nearest

to D and is within the transmission radius R of node S (node "A" in Figure 1). Nearest Forward within Radius (NFR) [11], transmits a packet to the nearest neighbor of the S which is in direction of the D (node "B" in Figure 1). NFR reduces the number of collisions (by lowering the transmit power to restrict signal propagation to a confined area) as opposed to MFR where the number of hop counts is reduced. Compass Routing (CR) [12], forwards packets to the next node that forms the smallest angle between the line connecting S and D (node "A" in Figure 1). CR may create loops [2] under special conditions. Randomized CR algorithm, [13] a variant of CR, tries to avoid loops by taking randomized routing decisions (node "A" or node "B" in Figure 1). GEographic DIstance Routing (GEDIR) [2] can be differentiated from other greedy algorithms in situations when the sending node itself is a local minimum (dead-end). For example in Figure 1, assume source node F wants to send a message m to destination node D, and F and D are not in the transmission range of one another. In this case F is a *dead-end*, but according to GEDIR, node F will still forward m to node B, in hope that B may have another neighbor which is closest to the destination D. GEDIR is proved to be loop free algorithm, although local loops may get created which can be dealt with by dropping the message by limited memorization. Greedy Perimeter Stateless Routing (GPSR) [14] is a routing strategy based on combining two different forwarding strategies: greedy forwarding and right hand rule. GPSR uses greedy forwarding approach by using only neighborhood information. When greedy forwarding becomes impossible (if a packet arrives at a dead end), than the forwarding strategy is switched to the right hand rule, where the packet is forwarded along the FACE of the planar graph [13]. The basic assumption of considering a planar graph can be computational complex when considering nodes in a 3 coordinate system (having differences in altitudes as well). GPSR results in large number of beacon messages to maintain routing table entries [1] which results in communication and processing overheads. GPSR performs well in dense networks where the average degree is greater than 20 but performance deteriorates with a decrease in the density of networks [15]. Distance Routing Effect Algorithm for Mobility (DREAM) [16] is designed keeping in view the node mobility and distances between nodes. The routing table in each node maintains network-wide location information. Each node sends its location information to every other node with help of control packets. Though it is claimed in [16] that DREAM is a loop free routing algorithm, it is shown in [2] that it is not loop free with a counter example. Location Aided Routing (LAR) [17] is a routing strategy for mobile ad hoc networks (MANETs) where location information is used to enhance the performance by reducing flooding overhead protocols where flooding is used as the basic mechanism for route discovery (e.g. reactive routing protocols). LAR and DREAM reduce the communication overhead to some extent by the use of directional flooding but still paths are maintained between two communication pairs. The enhancement achieved is to restrict the flooding to a particular zone as done in DREAM. All the

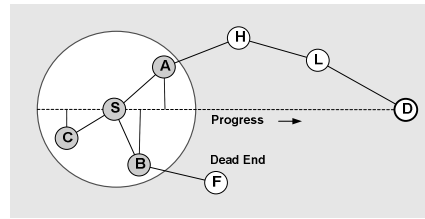routing schemes discussed above are based on maintaining routing tables as opposed to TPR.

Figure 1: Position Based Routing Strategies

The work that is most related to TPR is Implicit Geographic Routing (IGF) [18], Contention Based Routing (CBR) [3], and Blind Geographic Routing (BGR) [1]. The basic idea of IGF is to *non-deterministically route packets by allowing next hop candidates to "compete" in the forwarding process*". IGF is a combined routing-MAC scheme which uses the 802.11 DCF MAC scheme. The forwarding zone is defined by an angle (30 degrees in this case) with a line connecting the source and destination. It also introduces energy aware and distance aware metrics to facilitate routing decisions. The energy related metric is based on local information only and would experience problems discussed in Section IV A. Communication failures in case of absence of nodes in the forwarding zone are identified but not explained. In CBR, the routing scheme is divided into the *selection* process and the *suppression* (area based suppression and active suppression) process. In the selection process, the next hop is determined by means of contention while in the suppression process if a node is selected as a forwarding node; other nodes within the same transmission radius are suppressed from being selected. The timers used for contention can be based on progress. The dead end problem is not solved in the paper and it is mentioned that one of the existing schemes can be used as a recovery strategy. BGR, in principle a variant of IGF and CBR, forwards the packets in a greedy manner and is table-less. It focuses on minimizing duplicate packets. Our scheme is different from the aforementioned schemes in the definition of the Forwarding Area (FA), the recovery strategy, and by considering the network wide energy aware cost towards the sink node (which is considered by none of the schemes). A distributed algorithm is presented in [15] for the dead end problem in location based routing but requires neighborhood tables to be maintained.

III. PROTOCOL DESCRIPTION

The protocol is divided into different components which are discussed as follows:

A. Location Service and Setup Phase

All the participating nodes in the network update their current location after time interval T, and use this location information until the next location update. The value of T can be set based on application requirements e.g., for containers transported by trains, the update can be done less frequently as the relative position of the containers do not change while on the railway. The sink node initiates the setup by sending a broadcast message (m_{setup}). m_{setup} contains the current location

of the node, the minimum and the maximum State Of Charge (SOC_{min} and SOC_{max}), and the cost to the sink (C_{sink}) which is set to zero by the sink node. Each node N_i computes its own state of charge (SOC) which is given by Q/Q_{max} where Q is the remaining battery capacity and Q_{max} is the initial battery capacity. The remaining battery capacity can be computed by linear discharge model or the Rakhmatov battery model [19]. Initially N_i initializes its SOC_{min} and SOC_{max} to SOC. N_i may receive different SOC ranges from multiple sources but stores only the maximum and the minimum SOC. If node's own SOC is less or greater than the received SOC_{min} or SOC_{max} respectively, it updates m_{setup} before forwarding it. Ni also stores the location of the sink node and the accumulative cost towards the sink node (Section IV, part C). At the end of each setup, every node knows the network wide minimum and maximum SOC, sink location, and C_{sink} (without neighborhood or next hop information). Whenever the sink issues a new setup message m_{setup} it tags it with a different identifier Id_{setup}. Each node receiving m_{setup} checks the message for setup updates. Whenever a node updates its setup information it forwards m_{setup}. A setup update occurs when Id_{setup} changes, when the received cost is lower than the saved one, or when SOC_{min} or SOC_{max} changes. Whenever Id_{setup} changes the node drops all setup information and reinitializes its setup state based on most recently received data.

This cost is than used in response timers (part C) which are used by nodes to become the relay nodes for any transit traffic. The intuition to use the accumulative cost towards the sink node is as follow: Assume that node S intends to send a packet to node D as shown in Figure 2. The percentage values represent the remaining battery capacity e.g., 90% for node A. Now if the decision is based only on the remaining battery capacity of the node and not the accumulative cost towards the sink node, node A will become the relay node based on high energy reserve (low cost). The message will than reach the destination D via node B and C which are low energy nodes and would soon result in depleted nodes. The network life time (the time of the depletion of the first node is considered to be the network life time) can be increased by using the accumulative cost towards the sink node which would have resulted in the selection of node E. This principle also applies to other parameters like delay or congestion.
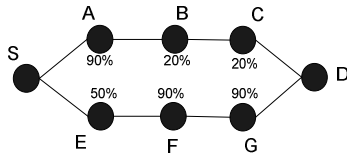

Figure 2: Routing decision based on accumulative cost vs. local cost

B. Forwarding Area, Contention, and Suppression

The FA is defined by an angle α drawn at a line connecting the sender node S and destination node D as well as the transmission range of the S. The angle α can be selected based on the density of the network i.e. the denser the network, the smaller the angle, e.g., [18] considers 30 degrees. All the nodes that are within the transmission range R and within an angle α are considered to be in the FA. For example in Figure 4 dark nodes are considered to be in the FA if S is the sender node. Any of the dark nodes which become relay node for the transit packet would send an ACK message back to S. This ACK message is also used to *suppress* the other nodes which are competing to become relay nodes. The transmission range covered by the ACK message is R_x which is the maximum possible distance between two nodes with in the FA and is given by ,

$$R_x = \max\left(R, R\sqrt{2(1 - \cos(\alpha))}\right) \qquad (1)$$

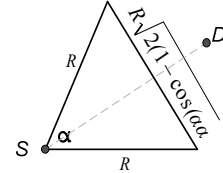The second term in the equation is given by the law of cosines as shown in Figure 3 .

Figure 3: Law of Cosines

Assuming $\alpha = 90°$, the transmission range for data packets would be R (inner circle in Figure 4) while that for ACK messages would be $R_x = \sqrt{2}R$. Alternatively to reduce the transmission range and hence the interference range as in [11], the original message can be sent with a range of $R/\sqrt{2}$ and the ACK can be sent back with a transmission range of R (for $\alpha = 90°$). The maximum value of α can be restricted to 90 degrees which would make sure that all the nodes selected in FA would result in positive progress towards the destination. Thus adaptive transmission power control along with positions of the source node, relay node, and the sink node are used to define the FA.
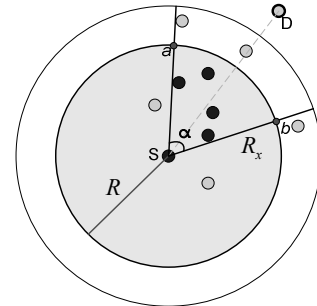

Figure 4: Forwarding Area

C. Response Timers

A response timer (T_r) is used by nodes which are competing to become relay nodes in the FA. T_r is given by,

$$T_r = T_{max}\left(\eta C_d + (1 - \eta)C_e\right) \qquad (2)$$

T_{max} is the maximum possible value of the T_r (maximum delay) defined by the user where $0 \leq \{\eta C_d + (1 - \eta)C_e\} \leq 1$ as well as $0 <= \{\eta, C_d, C_e\} <= 1$. T_{max} can have significant impact on the delay and is dependent on node density [3, 1]. C_d is the time attributed to T_r based on *positive progress* that is made towards the sink node. The higher the progress made by

the competing node, the lesser is the time added to T_r. C_d is given by

$$C_d = 1 - \frac{\overline{SN_i}}{R} \qquad (3)$$

$\overline{SN_i}$ is the Euclidean distance between sender node S and any competing node N_i. Assuming $\overline{SN_i} = R$ would result in maximum possible progress and hence attributing no time to T_r. C_e is the time attributed to T_r based on the total energy cost towards the sink node. The higher the cost to reach the destination, the higher is the time added to the T_r. C_e is given by,

$$C_e = \frac{(1 - SOC_{scaled})^\sigma + C_{min}}{HC} \qquad (4)$$

where SOC_{scaled} is SOC of the node scaled in the range SOC_{max} and SOC_{min}. The use of SOC_{scaled} is to resist performance degradation caused because of the lower differences between SOC values i.e. the lower the differences between SOC values, the lower is the time attributed to T_r and hence, the higher the number of duplicate packets. Additionally, by using a scaled SOC, T_r will not degrade to large timeouts (resulting in higher delays) caused by comparatively more drained batteries. C_{min} is the minimum cost received by competing nodes in the setup phase or as piggybacked information in ACKs. Motivated from [21], the cost based on remaining battery capacity is varied in linear, quadratic, and cubic fashion. The cutoff values which are set empirically are selected from [21]. For $Q \geq 80\%, \sigma = 3$, for $20\% \leq Q < 80\%, \sigma = 2$ while for $Q < 20\%, \sigma = 1$. This make sure that nodes with less energy would increase their cost non-linearly and hence attribute more time to T_r. HC is the hop count along the minimum cost path that is stored by each node N_i. Every node saves the value obtained from equation 4 to attribute time to T_r while forwarding the accumulative cost (cost obtained before dividing it by HC), so that values of C_e are normalized between 0 and 1 as required by the equation 2. It should be noted that the time attributed by C_e would not result in energy optimal routing but energy aware routing as the cost propagates slowly between two consecutive setup phases. Also the mobility of nodes would affect the total cost based routing decisions adversely.

D. Routing Strategy

Assume that in Figure 5, sender S wants to send a message m to destination D. S will blindly forward m and only nodes C and B (being in FA) would compete to become the forwarding nodes. Both node C and node B would start their response timers. As soon as a timer of one of them expires, lets say node B, it would send back an ACK message to S. Now node C would also hear the ACK message which carries the message ID of m and thus would cancel its timer. In this way, the node whose timers expire first wins to become the forwarding node. It should be noted that nodes S after sending m also starts its ACK timer whose value is T_{max} plus some additional random

time (to account for unprecedented delays e.g. caused by the MAC layer). If the ACK timer expires, a recovery strategy is initiated which is discussed in the following section.

E. Dead-End Problem and Recovery Strategy

A node which is nearest to the sink node in comparison to all its neighbors and not within the transmission range R of the sink is a *dead end* because only those nodes are selected in FA which would result in positive progress. We call these nodes as *dead* nodes while the others as *live* nodes. In Figure 5, assume that node F has either a self generated packet or transit packet to forward to the destination node D. Node F will forward the message m, and other nodes if any in the FA will process the message. At the same time, it will start the ACK timer. If the ACK timer expires, before ACK arrives, node F assumes that there is no neighbor node with positive progress towards the destination node D. Node F will declare itself a *dead* node and broadcast the message $m_{reverse}$ with a field indicating reverse message. Being a *reverse* message, all the neighbor nodes which are out of FA will compete in a similar way to relay the message. In this case node B will get hold of message $m_{reverse,}$ convert it to a message m and would forward the message as in normal routine. As node F has already declared itself *dead,* it would not compete to become the relay node, and hence the whole process will be repeated until message reaches node S and ultimately to D via node C and E. When node F hears a message from a *live* node, which is nearer to the destination than itself, it would convert itself into a *live* node and would take part in competing to become a relay node as long as it is a *live* node.
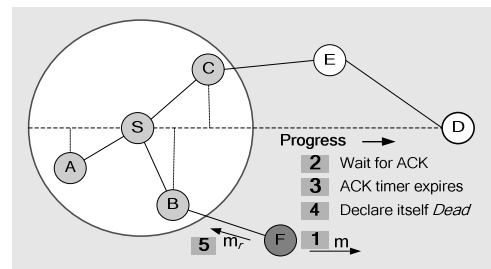

Figure 5 : Dead end problem and recovery strategy

IV. SIMULATION AND RESULTS

TPR is simulated in the OMNet ++ [23] based discrete event simulation framework PAWiS [24] which is specialized to simulate low power systems. An unobstructed unit disc graph is considered for simulation. A first order radio model to transmit and receive messages is adopted from [20]. The amount of energy consumed to transmit a packet is given by E_{TX} with

$$E_{TX} = E_{electronics} \times k + E_{amplifier} \times k \times R^2 \qquad (5)$$

where $E_{electronics}$ is the energy consumed to run the transceiver circuitry, k is the size of the packet in bits, $E_{amplifier}$ is the energy consumed by the amplifier to achieve an acceptable SNR and R is the transmission range. The energy consumed to receive a packet E_{RX} is given by

$$E_{RX} = E_{electronics} \times k . \qquad (6)$$

As in [20], $E_{electronics}$ = 50nJ/bit, $E_{amplifier}$ = 100pJ/bit/m^2, and k = 512 bits and 100 bits for data packet and ACK packets respectively. $E_{electronics}$ for receiving and transmission is assumed equal. The MAC layer is implemented as a simple CSMA scheme. As the communication is done hop by hop, and end to end reliability is not concerned, the transport layer is not implemented. The application layer is abstracted by probabilistic period sampling (if the change in value is greater than 20 %, a data packet is sent to the sink). T_{max} is set to 45 milliseconds as done in [3]. Decreasing the value of T_{max} would result in increased number of collisions while increasing it would result in increased delays. The value of T_{max} needs to be dynamically set based on node density.

Figure 6 shows simulation results for 25 nodes uniformly distributed over area of 200 x 200 m^2. Although the remaining energy of nodes with η = 1 is higher but the energy consumption is not uniformly distributed which is clear from certain peaks in Figure 6. It also shows that the network life (depletion of the first node is considered as network life) of the network for η = 0 (energy aware) is greater than for η = 1 (progress aware). The encircled area shows that node 19 has exhausted all its energy for η = 1 and hence resulting in a partially disconnected network.
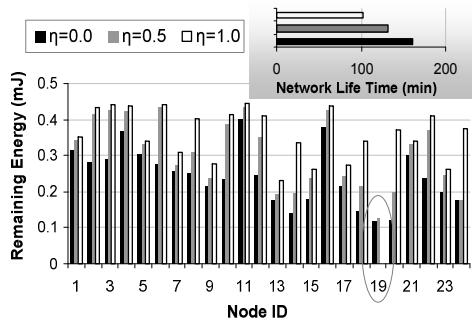

Figure 6 : Remaining energy and network life time

Figure 7 shows the end to end delay for different values of η. The results show that lower end to end delay for η =1 and it keep increasing for lower values of η. A few peaks encircled are because of the initiation of recovery schemes. The recovery scheme may get initiated even if there is a path available because it may not receive the packet correctly or is busy in processing self/other packets. The similar peaks may be for any η as shown in Figure 7 **.**
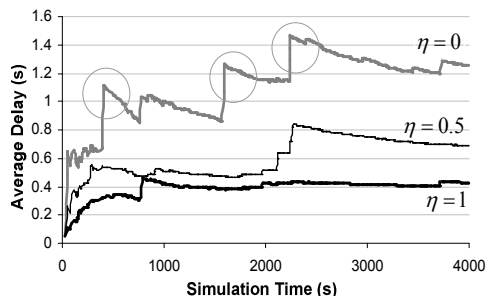

Figure 7 : Average end to end delay for different values of η

Figure 8 shows the delivery ratio for different values of η .It is clear from the figure that the protocol promises guaranteed

delivery. A delivery ratio of higher than 1 indicates packet duplicates. For η = 1, the delivery ratio is near to the optimal value. As we decrease the value η, the delivery ratio increases. The reason for this is that for lower values of η, e.g. η = 0, the protocols always try to balance the energy consumption across the network. Once the equilibrium is attained, and the remaining energy of nodes are almost at the same level, the difference between the timers of contending nodes also becomes very small, and the transit packet is processed before it receives the ACK message.
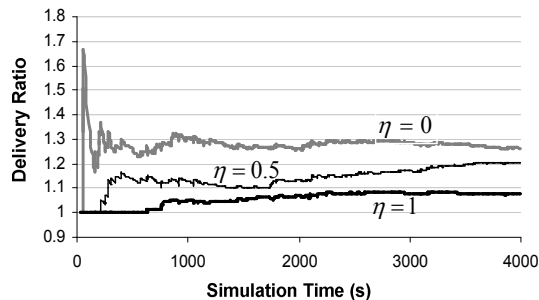

Figure 8 : Delivery ratio of different values of η

Figure 9 shows the average end to end delay for static sink against a mobile sink. A higher delay and a few abrupt changes indicate some potential forwarding nodes becoming dead nodes when the sink nodes move out of their transmission range.
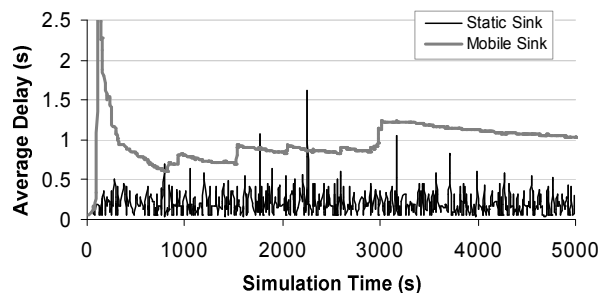

Figure 9 : Average end to end delay for static vs. mobile sink

Figure 10 shows the delivery ratio of a static sink against a mobile sink in the simulation topology. It is clear in the figure that the delivery of a message is guaranteed even if the sink node is moving. Higher delivery ratio for the mobile sink is because of firing of reverse messages forced due to mobility.
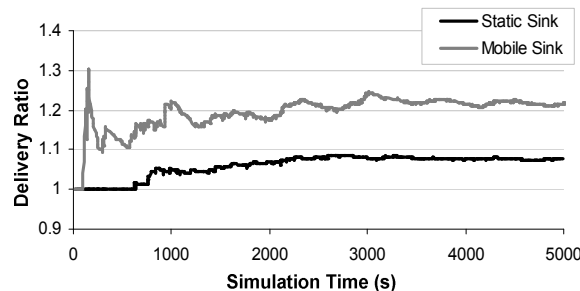

Figure 10 : Delivery ratio for static vs. mobile sink

The average end to end delay values in order of seconds do not reflect the absolute values. These values can be minimized to greater extent by using optimized value of T_{max}. The delivery

ratio of 1 indicates guaranteed delivery while delivery ratio of greater than 1 indicates packet duplicates. The only reason for packet duplicates is either the competing nodes have almost the same energy level or they provide the more or less the same progress towards the destination. The usage of η between the range of 0 and 1 and not the both extremes will eliminate the problem to some extent (by reducing the probability to have both the value in the same range).

V. Conclusion and future work

We have shown that TPR is well suited for low power wireless networks of embedded devices. The promising features include reduced message overhead, guaranteed delivery, and an adjustable delay-energy aware metric which can be adapted to extend the network life time or decrease the end to end delay (or draw a line in-between). TPR is highly scalable as no routing tables are maintained. Setting up response timers is a challenging issue specifically when the competing nodes provide same progress and/or same remaining energy based cost and requires further research. Future work also includes an adaptation of the currently used forwarding acknowledgement scheme to a CTS/RTS like approach (as done in [3]) to reduce the data packet duplicates at the cost of increasing control packet duplicates. Additionally a method to dynamically control weight parameters (progress or energy aware) during network lifetime is currently under development. We also intend to use obstructed graph models in future to see the robustness of protocol in such environments. Further testing and extensive simulation to compare TPR with similar class of protocols with detailed modeling of MAC layer and Radio will be done. Extensive simulations and performance evaluation would be followed by implementation of TPR on a real sensor node.

References

[1]   Witt, M.; Turau, V., BGR: blind geographic routing for sensor networks, Third International Workshop on Intelligent Solutions in Embedded Systems, 2005.

[2]   I. Stojmenovic and Xu Lin, "Loop-free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks," IEEE Trans. Parallel Dist. Sys., vol. 12, no. 10, 2001.

[3]   H. Fueßler, J. Widmer, M. Mauve, and H. Hartenstein. A Novel Forwarding Paradigm for Position-Based Routing (with Implicit addressing). In IEEE Computer Comm. Workshop (CCW 2003), 2003.

[4]   Martin Mauve, Jörg Widmer, and Hannes Hartenstein. A survey on position-based routing in mobile ad-hoc networks. IEEE Network Magazine, November 2001.

[5]   S. Mahlknecht, S. Madani, "On architecture of low power wireless sensor networks for container tracking and monitoring application", fifth IEEE international conference on industrial informatics, July 2007.

[6]   I. Stojmenovic, "Position-based routing in ad hoc networks," IEEE Commun. Magazine, vol. 40, pp. 128--134, July 2002.

[7]   C. Perkins and E. Royer, "Ad hoc On demand distance vector routing", Proc. 2nd IEEE Wksp Mobile Comp. Sys. & Apps., Feb 1999.

[8]   C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers", Comp. Commun. Rev., Oct. 1994.

[9]   D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks", Mobile Computing, Kluwer, 1996.

[10] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," IEEE Transactions on Communications, vol. 32, no. 3, pp. 246–257, March 1984.

[11] Ting-Chao Hou; Victor Li, "Transmission Range Control in Multihop Packet Radio Networks", IEEE Transactions on Communications, Vol.34, Iss.1, Jan 1986.

[12] E. Kranakis, H. Singh and J. Urrutia, Compass routing on geometric networks, Proc. 11th Canadian Conference on Computational Geometry, Vancouver, August, 1999.

[13] F. Ara´ujo, "Position-based distributed hash tables," Ph.D. dissertation, University of Lisbon, Lisbon, Portugal, 2005.

[14] Karp, B. and Kung, H. T. 2000. GPSR: greedy perimeter stateless routing for wireless networks. 6th Annual international Conference on Mobile Computing and Networking, August 2000.

[15] Zou, L.; Lu, M.; Xiong, Z., A Distributed Algorithm for the Dead End Problem of Location Based Routing in Sensor Networks, IEEE Transactions on Vehicular Technology, Vol.54, Iss.4, July 2005.

[16] Basagni, S., Chlamtac, I., Syrotiuk, V. R., and Woodward, B. A. 1998. A distance routing effect algorithm for mobility (DREAM). 4th Annual ACM/IEEE international Conference on Mobile Computing and Networking, October 25 - 30, 1998.

[17] Ko, Y. and Vaidya, N. H. 2000. Location-aided routing (LAR) in mobile ad hoc networks. Wirel. Netw. 6, 4 , Jul. 2000.

[18] B. Blum, T. He, S. Son, and J. Stankovic. IGF: A state-free robust communication protocol for wireless sensor networks. Technical Report CS-2003-11, Department of Computer Science, University of Virginia, 2003.

[19] Rakhmatov, D. N. and Vrudhula, S. B. "An analytical high-level battery model for use in energy management of portable electronic systems". IEEE/ACM international Conference on Computer-Aided Design, 2001.

[20] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. Hawaii International Conference on System Sciences Jan. 2000.

[21] Mahlknecht, S., Madani, S.A., Roetzer, M., "Energy Aware Distance Vector Routing Scheme for Data Centric Low Power Wireless Sensor Networks", 5th IEEE International conference on industrial informatics, 2006.

[22] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In Proc. IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002.

[23] www.omnetpp.org

[24] D. Weber, J. Glaser, S. Mahlknecht, "Discrete event simulation framework for power aware wireless sensor networks", in proceedings of fifth IEEE international conference on industrial informatics, July 2007.