doi:10.20944/preprints202007.0550.v1

Article

A Bidirectional Adaptive Multihop Routing Algorithm for Wireless Body Area Networks

Abdelrahman Miky [1], Mohamed Saleh [2], Bassem Mokhtar [3] and M. R. M. Rizk [4]

[1] Depart. of Medical Equipment Technology, Pharos University, Egypt; abdelrahman.miky@pua.edu.eg
[2] Department of Electrical Engineering, Pharos University, Egypt; mohamed.saleh@pua.edu.eg
[3] Deparment of Electrical Engineering, Alexandria University, Egypt; bmokhtar@alexu.edu.eg
[4] Deparment of Electrical Engineering, Alexandria University, Egypt; mrmrizk@ieee.org
* Correspondence: mohamed.saleh@pua.edu.eg

Version July 22, 2020 submitted to Sensors

Abstract: Wireless Body Area Networks are composed of sensor nodes that may be implanted in the body or worn on it. A node is composed of a sensing unit, a processor and a radio unit. One of the nodes, the sink, acts as a gateway between the body area network and other networks such as the Internet. We propose a routing protocol that constructs paths between nodes such that the final network topology is a tree rooted at the sink. The protocol's aim is to increase network lifetime and reliability, and to adapt to network conditions dynamically. Moreover, the protocol enables communications between nodes and sink both in the upstream direction, from nodes to sink, and in the downstream direction from sink to nodes. When the network tree is constructed, a node chooses its parent, i.e., next hop to sink, by using one of various criteria. Namely, these are the number of hops between parent and sink, energy level of parent, received signal strength from parent, number of current parent's children, and a fuzzy logic function that combines multiple criteria. Moreover, as time progresses the tree structure may dynamically change to adapt to conditions such as the near-depletion of a routing node's energy. Simulation results show improvements in network lifetime and energy consumption over the older version of the protocol.

Keywords: Wireless Body Area Networks, Adaptive Routing, Two-way Communication in BANs, Routing protocol in BAN, Fuzzy logic

---

1. Introduction

In Wireless Body Area Networks (WBANs) [1–3], nodes implanted in the body or placed on it sense data that represent some physiological characteristics, such as body temperature or blood pressure. These data usually need to be collected remotely for some purpose such as patient monitoring or health studies. One node, the sink, therefore acts as a gateway to forward traffic from its WBAN to a remote network. Several topologies for WBANs have been proposed [4–6] ranging from star topologies with the sink at the center of the star, to tree or mesh topologies. Whenever possible, multihop communication over these topologies is favored for two reasons [7]. First, it saves energy by allowing nodes to reduce their transmission range and hence their transmission power. Second, it enables the expansion of the network's area to cover the whole body even while using small resource-limited nodes such as implanted sensors, which also suffer from high levels of signal attenuation from body tissue [8,9].

Many routing protocols have been proposed for WBANs [10], where end-to-end multihop routes are constructed while targeting some objective. Some of these objectives are specific to WBANs such as avoiding high temperature rises in nodes due to high traffic loads [11]. Other objectives are general such as minimizing delay. The objective of a particular routing protocol determines some criteria for the route selection process. For instance, if the protocol aims at minimizing packet delay when a node communicates with the sink, then routes will be selected based on the number of hops to sink. Moreover, routing protocols for WBANs need to take into account the fact that nodes are limited in energy and computation resources.

The Adaptive Multihop Routing (AMR) protocol [12] is designed specifically for WBANs and supports multiple route selection criteria. The selection is based on the value of a metric that is evaluated for each possible route. Comparing values of metrics, the protocol chooses the route that satisfies the required criteria. The defined metrics and the route selection process are detailed in Section 2. Routes constructed by the AMR protocol form a tree topology that is rooted at the network's sink node. Traffic travels upward following tree branches till it reaches the sink. So, all parent nodes, i.e., all nodes except leaf nodes, act as relaying nodes that not only send their data but also forward data from their children.

The protocol is adaptive due to two design aspects. First, by supporting multiple route selection criteria, the protocol's objective can be varied to suit the network. For instance, as objective, we can choose between minimizing delay or increasing network lifetime. Moreover, the protocol uses fuzzy logic to combine several selection criteria and hence provide a compromise between various objectives. Second, it enables dynamic changes in network topology where a parent node whose stored energy is near depletion may save energy by not forwarding traffic from its children which become "orphaned". Orphan nodes are able to rejoin the network by choosing a new parent.

By modifying the design of the AMR protocol, we were able to improve its performance and extend its functionality. The first modification is to change the route selection process by adding a new metric that a node uses when joining the network. This metric is the number of current children of the potential parent node. In more detail, a node, when joining the network's tree, chooses the parent with the least number of children. This is to prevent a situation where some nodes are overloaded with children and others have few or no children. Overloaded nodes suffer premature "death" which deteriorates network lifetime. The second modification is related to the sink, where the original protocol gives priority to the sink to become parent even against the logic of the route selection criterion. For instance if the criterion is reliable communication, then a node should choose as parent the node with the highest received signal strength even if it is not the sink. The third and final modification is to extend the protocol by adding support for bidirectional flow of data. The original protocol, as many other WBAN routing protocols, supports a unidirectional flow of data; upstream from nodes to sink. This is the prevalent direction since this is how sensing data are collected. However, some data need to travel downstream from sink to nodes such as network management data, e.g., configuration commands.

In Section 2, we present our design of the protocol. We start by introducing the design of the original protocol as a Finite State Machine (FSM). We follow that by an analysis of this design based on careful inspection of the protocol's specification and simulation results. Finally, we describe our modifications and additions to the original design. Section 3 includes both the analysis and simulation results of our proposed protocol. Results are discussed in order to provide a better understanding the protocol's behavior. Finally, the paper is concluded in Section 4.

2. Our Protocol Design

In the original protocol [12], the main objective is to construct a tree topology. The sink is the root of tree at level 0. Level 1 includes all nodes that are one hop from root, i.e., the root's direct children. In general, if a node is at level i (i hops from root) then all of its direct children are at level $i + 1$.

2.1. Original Design

Tree construction in the original protocol follows the following steps:

- The node configured as sink sends a broadcast `Hello` message.
- When a node receives the sink's `Hello`, it replies by a `Join` message.

- When the sink receives a `join` message from node i, it replies to i by an `Accept` message. Node i then updates its routing table to list the sink as its parent.
- When a node becomes the sink's direct child, it sends a broadcast `Hello` message.
- Nodes that receive one or more `Hello` messages that are not from sink, wait for an h-wait time period. The value of h-wait is a protocol parameter that is configured in a timer.
- The function of the h-wait time period is to enable a node to potentially receive more than one `Hello` message. The node then uses a metric to decide which `Hello` sender to choose as potential parent.
- When the h-wait period has passed, the node sends a `Join` message to the parent it has chosen. Then, the node waits for an a-wait time period expecting to receive an `Accept` message.
- When the node receives an `Accept` message, it modifies its routing table by listing the message's sender as parent. Now the node is part of the tree being constructed.
- When a node joins the tree at some level, it sends broadcast `Hello` messages advertising its willingness to become parent. Thus, the tree continues to be constructed.

The metric that a node uses to decide which `Hello` sender to choose as potential parent is one of the following four metrics [12]:

- NoH: The node chooses the parent with the least Number of Hops away from root.
- RSSI: The node chooses the parent with the highest Received Signal Strength Indicator.
- BEL: The node chooses the parent with the highest Battery Energy Level.
- FLF: The node uses a Fuzzy Logic Function of the three previous metrics.

A node that has joined the network will be able to send data packets to the sink by sending them to its parent. The parent will, in turn, forward packets to its own parent, and so on till packets reach the sink. So, a node needs to store only the address of its parent in its routing table. On the other hand, a parent does not store the addresses of its child nodes. The original protocol does not thus support communication downstream from sink to nodes. It only supports communication in the upstream direction. Moreover, a `Leave` message is defined that may be sent from a child node to its parent in the tree. The message is sent when the energy level of the parent is lower than some threshold. After sending this message the child will stop sending and forwarding messages to its parent. The objective is to save energy and thus prolong the parent's life. The original protocol was tested in hardware using actual sensor nodes and results were collected about network lifetime, Packet Delivery Ration (PDR), average number of transmissions per packet delivered, and total remaining energy in network.

2.2. Our Analysis

We studied the operation of the AMR protocol and ran ns-2 [13] simulations in order to analyze its performance and were able to find the following shortcomings. Firstly, when a node hears a `Hello` message that is sent by sink, it tries to join the sink without waiting to receive other `Hello` messages. However, when it hears a `Hello` message from a node that is not the sink, it waits for time period h-wait to receive more `Hello` messages. It will then choose a parent from one of the `Hello` senders. This logic favors the sink over other nodes even when the parent choice metric dictates otherwise. Of course, at the start, the sink will always be chosen as parent since it will be the only node broadcasting `Hello` messages. This is necessary for "bootstrapping" the network. However, after some nodes have joined the sink, they will be broadcasting `Hello` messages. At that point, nodes hearing `Hello` messages should wait for time period h-wait even if they hear a `Hello` from sink.

Figure 1 depicts an example network topology; the sink is placed at the ankle and other nodes are placed over the body within the transmission range of sink. If nodes use the NoH criterion to choose their parent, then all nodes should choose the sink. However if nodes use the RSSI criterion and all nodes have the same transmission power, then nodes C and D should choose E or F as parent, and nodes E and F should choose the sink S as parent. and so on.

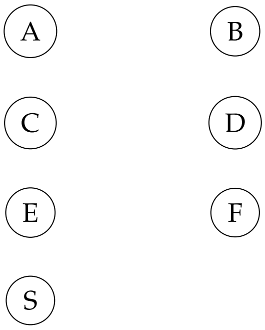

Figure 1. Example topology where the sink S is at the ankle.

Secondly, it is desirable that parent nodes in the network tree are more or less equally loaded by network traffic. This is because if a node is much more loaded than others, its energy will be depleted prematurely, which well negatively affect the network's lifetime. For nodes to be equally loaded by traffic, the number of children should not vary greatly from one parent node to another, since parent nodes forward traffic received from their children. The original protocol does not attempt to provide an even distribution of child nodes to parent nodes.

Thirdly, the original protocol states that a child node should send a `Leave` message to its parent, whenever the energy level of parent drops below some threshold. However, it is not clear from the protocol's description how the child will know the value of the parent's energy. This is emphasized by the fact that network traffic is sent only in the upstream direction, from child to parent. In this situation, the parent needs to broadcast information about its energy level, either periodically or only when the energy level drops below the configured threshold. Then, child nodes will be able to send the `Leave` message in ample time. This way parent nodes will consume valuable energy and create more network traffic than needed for collecting sensor data.

Fourthly and finally, the original protocol does not account for the need to send data in the downstream direction from sink to nodes. Whereas upstream traffic is needed for collecting data from sensors, downstream messages are necessary to perform network management tasks such as modifying a node's configuration, or sending a command to a node.

2.3. Modified Design–First Version

In a previous work [14], we presented Modified AMR (MAMR). Its finite state machine model is depicted in Figure 2. The initial sate is called "start", and we have the following states:

- w-hlo: In this state, the node waits for "h-wait" time units to collect `Hello` messages, then it sends a `Join` message to one of the `Hello` senders and goes into state "w-acc".
- w-acc: In this state, the node waits for an `Accept` message for "a-wait" time units. If it receives an `Accept` message it switches to the "data" state, if not, it resends a `Join` message.
- data: In this state, the node sends its own data to its parent and forwards data from its children. If the node's energy drops below a preset threshold (low-E event) it sends a broadcast `Leave` message and goes into the "send" state. Also, in this state, the node is ready to receive `Join` messages and send `Accept` messages.
- send: In this sate, the node only sends its own collected data. It does not forward data.
- 2-acc: The node goes into this state upon receiving a `Leave` message from its parent. This state is similar to the "w-acc" state in that the node tries to join a parent to be able to re-enter the "data" state.

The MAMR protocol is different than the original one in three aspects. The first is that we treat the `Hello` message received from sink no different than a `Hello` received from any other node. This difference is depicted in Figure 2, where the dashed lines are the transitions of the original protocol

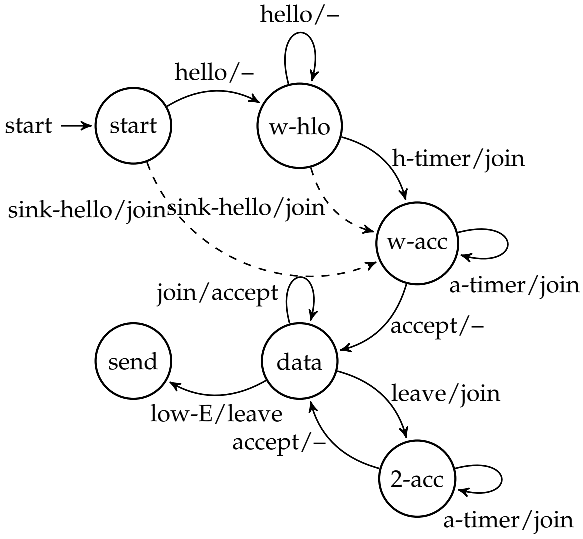

Figure 2. MAMR protocol's Finite state machine.

(AMR). In AMR, once a `Hello` message is received from sink, a `Join` message is sent right away to sink without continuing the "h-wait" time period.

The second modification is that we add a new metric that nodes use to choose their parent. This is the Count of Children (CoC) metric. A node that uses this metric will choose as parent the `Hello` sender that has the least number of children. The objective is to make the number of *direct* children more or less equal between parent nodes, so that the energy depletion rate is almost constant for all parent nodes. The network lifetime is therefore increased by preventing early "death" (total depletion of energy) that happens to overloaded nodes. Consequently, we modified the fuzzy logic function to include all metrics, namely NoH, RSSI, BEL, and CoC. Table 1 is the original one from the AMR protocol, it has three inputs: Number of hops, residual energy in node, and RSSI. The output is the connection metric. The design of our fuzzy logic function is illustrated in Table 2 where the connection metric is an input along with the number of children. The output of Table 2 is the one we use as the new fuzzy metric.

Table 1. Original Fuzzy Logic Control for AMR Protocol.

AMR Tradtional Fuzzy Table				
	Residual Energy			
Number Of Hops	Low	Medium	High	RSSI
near	**Bad**	**Bad**	**Median**	Poor
near	**Bad**	**Median**	**Good**	Average
near	**Median**	**Good**	**Good**	Good
Medium	**Bad**	**Bad**	**Bad**	Poor
Medium	**Bad**	**Median**	**Median**	Average
Medium	**Median**	**Median**	**Good**	Good
Far	**Bad**	**Bad**	**Bad**	Poor
Far	**Bad**	**Median**	**Median**	Average
Far	**Median**	**Median**	**Median**	Good

The third modification is that the `Leave` message is sent from a parent node, as opposed to being sent from a child node in the original protocol. In the MAMR protocol, a parent node broadcasts a `Leave` message, we note here that a parent does not know the addresses of its children. When a node receives this message and if it is a child to the sender, it will delete its parent's address from the routing table and go to the "2-acc" state. It will also, send a `Join` message to one of the senders of the `Hello` messages that it previously collected. It chooses this sender to be its new parent according to the same metric that it used for its previous parent (the one that sent a `Leave` message), except that it now removes the previous

Table 2. New Fuzzy Logic Control.

AMR Modified Fuzzy Table			
Connection metric	Number Of Child		
	Small	Average	Large
Bad	**Near Median**	**Too Bad**	**Too Bad**
Median	**Near Median**	**Near Median**	**Too Bad**
Good	**Very Good**	**Very Good**	**Near Median**

parent from the list of senders. The MAMR protocol gives two options for the decision to send a `Leave` message. In MAMR-E, a `Leave` message is sent when the parent's energy drops below a particular level. In MAMR-T, a `Leave` message is sent when the parent's *time till death* t_d drops below a particular value. The value of t_d is estimated from the rate of decrease of energy:

$$t_d = E_{curr} \cdot \frac{\Delta t}{\Delta E} \tag{1}$$

In Equation 1 above, E_{curr} is the current energy level, and Δt is the time it takes for energy to drop by a value of ΔE.

2.4. Modified Design–Second Version

Our second version of the protocol, presented hereafter, is called Two-way AMR (TAMR). We modify MAMR to support bidirectional flow of data, i.e., upstream flow from nodes to sink and downstream flow from sink to nodes. In MAMR, once the tree is constructed, each node will store only the address of its parent in its routing table. This is sufficient for one-way (uni-directional) operation since all traffic travels to the sink. Each node needs to know only what is the next tree node on the way to the sink. When extending the MAMR protocol to design the TAMR protocol three issues need to be handled:

- The structure of the routing table in each node and how the table is filled and updated.
- The structure of protocol messages.
- The forwarding of data packets in the network tree.

The first modification to the MAMR protocol concerns routing tables. Now a routing table needs to have two fields (columns): Final destination and next hop. In case of upward traffic, the final destination is always the sink and the next hop is the node's parent. So, upward traffic needs only a single entry in the table. In case of downward traffic, the source is always the sink, but the final destination may be any node in the network tree. As a result, there may be multiple routing table entries for downward traffic. Due to the tree topology of the network, the routing table of a parent node will have only entries where the destination is a descendant of this parent. Descendants of a parent are all the nodes of the subtree rooted at this parent. The tree topology means that nodes that are not descendants of a parent cannot be reached from this parent. So, the routing table in a node has $N_d + 1$ entries, where N_d is the number of descendants of this node and one more entry is needed for upward traffic.

In other words, a parent node needs to know which nodes are its descendants, and, for each descendant, the parent needs to know the address of the next hop on the route to this descendant. In order to convey this information to nodes, special routing packets could be used, but for WBANs, this will consume precious energy, and possibly cause collisions with other packets. Hence, a degradation of network performance may result. So, we decide to use data packets to convey this routing information by modifying the packet structure. In the MAMR protocol, the packet contains only the source address since, in upward traffic, the destination is always the sink. For TAMR, we add a field for the destination address. We note here that this is a network layer header, so source and destination addresses are end-to-end, i.e., they do not change as the packet travels from node to node. This is in contrast to addresses in the MAC layer header such as the ones used in the IEEE 802.15.6. standard [4], which are called recipient ID and sender ID. Those are hop-by-hop addresses and change as the packet travels through the network. The sender ID is address of the current sender of the packet not of the source of the packet. Also, the recipient ID is the address of the current intended receiver of the packet not of the packet's final destination. We make use of this fact to fill the routing tables of nodes in a cross-layer fashion.

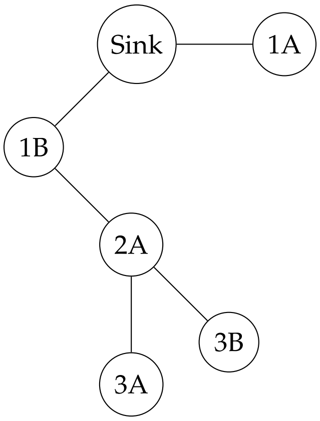

Figure 3. Example of a tree constructed by TAMR protocol.

When a node, e.g, node 3B in Figure 3 sends a data packet in the upward direction, i.e., to sink, it knows from the tree construction phase that its parent is node 2A. Therefore, the header of the MAC layer will have recipient ID = 2A and sender ID = 3B. The header of the network layer, however, will have destination address = sink and source address = 3B. Of course, this implies that we are using MAC addresses in the network layer which is a feature of a cross-layer approach. When the packet is received by node 2A, it examines both headers of the MAC and network layers and can thus know the following information:

- Node 3B is its descendant: This is known from the source address of the network layer header.
- Node 3B is its direct child: This is known from the sender ID of the MAC layer header.

Node 2A will then update its routing table by inserting an entry that indicates that Node 3B is the next hop to node 3B, i.e., 3B is a dirsct child. In this step, in cross-layer interaction, a node reads information from both the MAC header and network layer header to update its routing table.

Now node 2A forwards the packet by sending it to its parent; node 1B. The packet sent to 1B has a MAC layer header with recipient ID = 1B and sender ID = 2A. The header of the network layer, will have the same destination and source addresses as the packet sent by 3B, i.e., sink and 3B, respectively. When node 1B receives the packet, it examines both headers of the MAC and network layers and can thus know the following information:

- Node 3B is its descendant: This is known from the source address of the network layer header.
- Node 2A is its direct child and it is the next hop to node 3B: This is known from the sender ID of the MAC layer header.

Node 1B updates its routing table accordingly, and finally, when the packet arrives at sink, the sink will know that node 1B is the next hop to node 3B. We can thus conclude that a single upward-directed data packet updates routing tables for all nodes on the route to sink. In other words, an entry is added to these routing tables where the entry's final destination is the source address of the packet (from the network layer header) and the entry's next hop is sender ID (from the MAC layer header). So, for the routing table of a node to be fully updated, each of its descendants should send at least a single data packet. The sink's routing table will be fully updated when all nodes in the network have sent at least a single data packet.

Similar to the MAMR protocol, the TAMR protocol gives two options for the decision to send a `Leave` message. In TAMR-E, a `Leave` message is sent when the parent's energy drops below a particular level. In TAMR-T, a `Leave` message is sent when the parent's *time till death* t_d drops below a particular value. The value of t_d is estimated from 1.

3. Analysis and Simulation

Network Simulator 2 (NS-2) was used to simulate the behavior of the original protocol (AMR), and our two versions: MAMR and TAMR. Two scenarios are run where the network consists of a sink and 13 nodes. In the first scenario, the sink is placed at the ankle and in the second one, the sink is placed at the waist.

3.1. Analysis of Operation

Ideally, all nodes should be able to join the network and send data. However, some factors affect the operation of the protocol and may cause degradation in performance. Namely, these are: Collisions, processing delays, values for timers, channel parameters, and number of retransmission retries. In our simulation scenarios, we use the non-becon, no superframe CSMA/CA mode of operation of the IEEE 802.15.6 protocol. A collision at the MAC layer causes retransmissions up to a particular number of retries. Packet loss may therefore result. Also, a node takes some time to respond to an incoming message. This processing delay may cause the some variation in network behavior. For instance, a node, after joining the tree, broadcasts a `Hello` message, after some delay. The variation in this delay between different nodes affects the final topology of the tree, since some nodes may miss receiving the `Hello` message due to the expiration of their h-timer. This is also an example of how the values of timers (the h-timer in this case) affect node behavior. Also, channel parameters are dynamic in WBANs and depend on body posture. Communications may be disrupted, and path loss changes with time. Finally, due to collisions and the limited number of `Join` retransmission retries, some nodes may not be able to join the network. This happens when the `Join` or `Accept` messages are lost or excessively delayed.

3.2. Simulation Results

In the first simulation scenario, the sink is located at the ankle of one leg, and the other 13 nodes are placed the head (2 nodes), shoulders (2 nodes), arms (2 nodes each), waist (2 nodes), and legs (2 nodes at one leg and one node at the other). In the second scenario, one of the waist nodes become sink and the ankle sink becomes a regular node.

- Number of transmissions per delivered packet.
- Network lifetime, computed as the time that passes till the first node "death" in the network. Death here means total energy depletion. We assume an initial energy of 2 Joules (J) [12].
- Normalized residual energy averaged over all nodes.

Each parameter is computed for the protocol versions: AMR, MAMR-E, and TAMR-E. For each protocol version, all parent choice metrics are used, namely, FLF, NoH, BEL, RSSI, and CoC. The CoC metric is the one that we added in MAMR as explained in Section 2.3. Figures 4, 5, and 6 present simulation results.

3.3. Results Discussion

- Network lifetime: In Figure 4, we note that MAMR greatly improves network lifetime due to the modified behavior of the `Leave` message. Also, TAMR provides a performance comparable to MAMR. This is because TAMR does not use special routing packets but uses data packets to deduce routing information.
- Residual energy: Figure 5 depicts the normalized residual energy averaged over all nodes, after 1000 sent data packets per node. We note that MAMR provides better performance by preventing the situation where the energy of some network nodes are being sharply depleted. The dynamic behavior of the `Leave` message during network operation, shifts the network traffic between

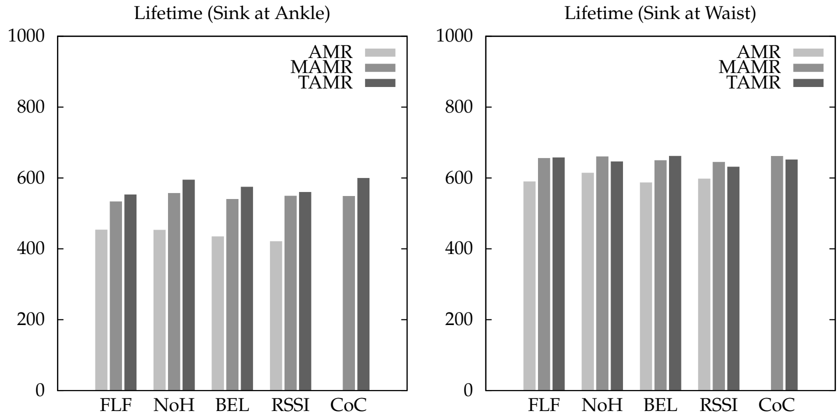

Figure 4. Simulation Results for network lifetime.

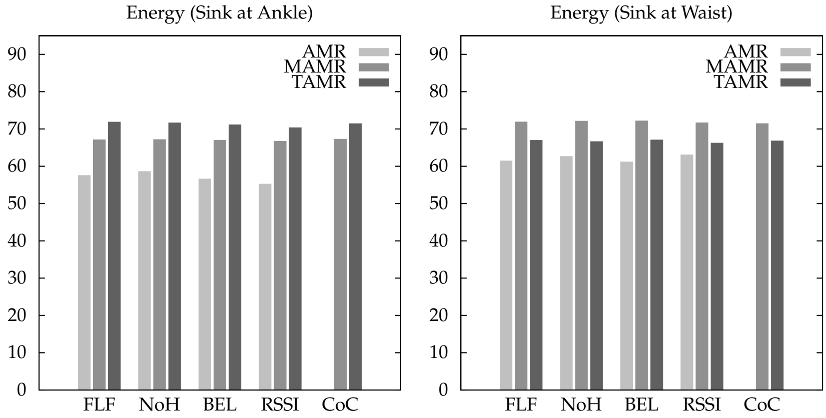

Figure 5. Simulation Results for normalized residual energy.

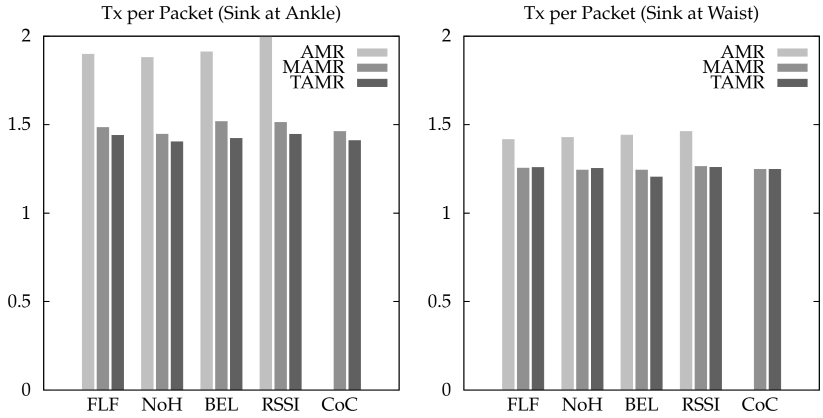

Figure 6. Simulation Results for number of transmissions.

nodes so that the energy depletion rate is almost even across all nodes. Again, TAMR provides performance that is not significantly different than that of MAMR.

- Number of transmissions per delivered packet: Finally, In Figure 6 we show the average number of transmissions per delivered packet for 1000 packets sent per node. Of course, when the sink is at the ankle this number is larger than when the sink at the waist, since in general network end-to-end paths will be longer. Also in MAMR the number is smaller. In fact, this shows a drawback in MAMR, since, due to processing delays, a node may still send a packet or more to its parent even after the parent has sent a `Leave` message. This type of packets will not be forwarded by the parent. Here also we note that TAMR has comparable performance to MAMR.

4. Conclusion and Future Work

We presented a routing protocol for WBANs that enables communication in both upstream direction, from nodes to sink, and downstream direction from sink to nodes. Most previous protocols in WBANs focus on the upstream direction since it is the direction of data collected by sensors to be sent to a base station or the cloud. This is the dominant direction in the network. However, we may need to send data in the downstream direction to send configuration parameters or other commands to sensors. To this end, our protocol enables a node in the network's tree to store information about its children, in addition to information about its parent. Simulation results show that the protocol leads to increased network lifetime. We intend to extend the this work by investigating the performance of the protocol over the IEEE 802.15.6 MAC and physical layer standard when using super frames with and without beacons.

1. Latré, B.; Braem, B.; Moerman, I.; Blondia, C.; Demeester, P. A Survey on Wireless Body Area Networks. *Wirel. Netw.* **2011**, *17*, 1–18.
2. Movassaghi, S.; Abolhasan, M.; Lipman, J.; Smith, D.; Jamalipour, A. Wireless Body Area Networks: A Survey. *IEEE Communications Surveys & Tutorials* **2014**, *16*.
3. Ullahn, S.; others. A Comprehensive Survey of Wireless Body Area Networks. *Journal of Medical Systems* **2012**, *36*.
4. IEEE Computer Society. *IEEE Std 802.15.6-2012 standard for local and metropolitan area networks part 15.6: Wireless body area networks*; IEEE Computer Society, 2012.
5. Rostampour, A.; Moghim, N.; Kaedi, M. A New Energy-Efficient Topology for Wireless Body Area Networks. *Journal of Medical Signals and Sensors* **2017**, *7*.
6. Society, I.C. *IEEE Std 802.15.4-2011, IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*; IEEE Computer Society, 2011.
7. Deepak, K.; Babu, A. Improving energy efficiency of incremental relay based cooperative communications in wireless body area networks. *International Journal of Communication Systems* **2015**, *28*, 91–111.
8. Cotton, S.L.; D'Errico, R.; Oestges, C. A review of radio channel models for body centric communications. *Radio Science* **2014**.
9. Huq, M.A.; Iftikhar, M.; Chilamkurti, N. Behavior of IEEE 802.15.4 Channel Models on Implant Body Area Network. Quality, Reliability, Security and Robustness in Heterogeneous Networks. Springer International Publishing, 2017.
10. Bhanumathi, V.; Sangeetha, C.P. A guide for the selection of routing protocols in WBAN for healthcare applications. *Human Centric Computing and Information Sciences* **2017**, *7*.
11. Bag, A.; Bassiouni, M.A. Hotspot Preventing Routing Algorithm for Delay-Sensitive Biomedical Sensor Networks. 2007 IEEE International Conference on Portable Information Devices, 2007.
12. Ortiz, A.M.; Ababneh, N.; Timmons, N.; Morrison, J. Adaptive routing for multihop IEEE 802.15.6 Wireless Body Area Networks. Proceedings of the 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2012.
13. The ns Manual. https://www.isi.edu/nsnam/ns/ns-documentation.html.

14. Miky, A.; Saleh, M.; Mokhtar, B.; Rizk, M.R.M. Adaptive dynamic routing for IEEE 802.15.6 wireless body area networks. 2018 35th National Radio Science Conference (NRSC), 2018.