



**HAL**  
open science

## Broadcast Strategies and Performance Evaluation of IEEE 802.15.4 in Wireless Body Area Networks WBAN

Wafa Badreddine, Claude Chaudet, Federico Petruzzi, Maria Potop-Butucaru

► **To cite this version:**

Wafa Badreddine, Claude Chaudet, Federico Petruzzi, Maria Potop-Butucaru. Broadcast Strategies and Performance Evaluation of IEEE 802.15.4 in Wireless Body Area Networks WBAN. [Research Report] LIP6 UMR 7606, UPMC Sorbonne Universités, France; UPMC - Paris 6 Sorbonne Universités. 2016. hal-01404848v1

**HAL Id: hal-01404848**

**<https://hal.science/hal-01404848v1>**

Submitted on 5 Dec 2016 (v1), last revised 6 Jun 2019 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Broadcast Strategies and Performance Evaluation of IEEE 802.15.4 in Wireless Body Area Networks WBAN

Wafa Badreddine<sup>1</sup>, Claude Chaudet<sup>2</sup>, Federico Petrucci<sup>1</sup>, and Maria Potop-Butucaru<sup>1</sup>

<sup>1</sup>UPMC Sorbonne Universites, LIP6-CNRS UMR 7606, France, LastName.FistName@lip6.fr

<sup>2</sup>Institut Mines-Telecom, Telecom ParisTech, CNRS LTCI UMR 5141, claude.chaudet@telecom-paristech.fr

The rapid advances in sensors and ultra-low power wireless communication has enabled a new generation of wireless sensor networks: Wireless Body Area Networks (WBAN). To the best of our knowledge the current paper is the first to address broadcast in WBAN. We first analyze several broadcast strategies inspired from the area of Delay Tolerant Networks (DTN). The proposed strategies are evaluated via the OMNET++ simulator that we enriched with realistic human body mobility models and channel models issued from the recent research on biomedical and health informatics. Contrary to the common expectation, our results show that existing research in DTN cannot be transposed without significant modifications in WBANs area. That is, existing broadcast strategies for DTNs do not perform well with human body mobility. However, our extensive simulations give valuable insights and directions for designing efficient broadcast in WBAN. Furthermore, we propose a novel broadcast strategy that outperforms the existing ones in terms of end-to-end delay, network coverage and energy consumption. Additionally, we performed investigations of independent interest related to the ability of all the studied strategies to ensure the total order delivery property when stressed with various packet rates. These investigations open new and challenging research directions.

*Index Terms*—Omnet++, WBAN, Broadcast, mobility model.

## I. INTRODUCTION

Wireless Body Area Networks (WBAN) open an interdisciplinary area within Wireless Sensor Networks (WSN) research, in which sensors are used to monitor, collect and transmit medical signs and other measurements of body parameters. The intelligent sensors can be integrated into clothes (wearable WBANs), or placed directly on or inside a body. If typical applications target personalized, predictive, preventive and participatory healthcare, WBANs also have interesting applications in military, security, sports and gaming fields. Healthcare workers, for instance, are really in demand of systems that permit a continuous monitoring of elderly people or patients to support them in their daily life. WBANs history is just at its beginning, and many news and improvements are expected in near future.

Body area networks differ from typical large-scale wireless sensor networks in many aspects: the size of the network is limited to a dozen of nodes, in-network mobility follows the body movements and the wireless channel has its specificities. Links have, in general, a very short range and a

quality that varies with the wearer's posture, but remains low in the general case. Indeed, the transmission power is kept low, which improves devices autonomy and reduces wearers electromagnetic exposition. Consequently, the effects of body absorption, reflections and interference cannot be neglected and it is difficult to maintain a direct link (one-hop) between a data collection point and all WBAN nodes. Multi-hop communication represent a viable alternative, but multi-hop communication protocols proposed in literature are not optimized for this specific mobility pattern. Adapted algorithms could easily take advantage of the particular connection changes pattern to limit control traffic, preserving bandwidth, energy and limiting radio emissions. Delay-Tolerant Networks protocols seem particularly relevant in this context, as they are designed to tackle the intermittent connectivity, thanks to their store-and-forward philosophy. However, they usually either suppose a perfectly known mobility as in interplanetary networks, or a fully random mobility. WBANs mobility exhibit a certain regularity that could be utilized, with a certain degree of randomness.

In this paper, we concentrate on the sole *broadcasting* problem. We analyze the behavior of various broadcast strategies adapted from DTN literature and propose an alternative strategy from the analysis of the strengths and weaknesses of the different approaches. We compare network coverage, completion delay and required transmissions of 9 different algorithms over real human body mobility traces.

The paper is organized as follow: Section II presents the broadcast problem. Section III introduces relevant related works and presents various broadcast strategies, including an original contribution, that we have compared through simulation in Section IV. Section IV first describes the limitations of the channel models used so far in OMNET++, then describes the realistic channel model we implemented and finally presents and analyzes the evaluation results from various broadcast strategies in terms of end-to-end delay, energy and node coverage.

## II. BACKGROUND

In the past decade, several routing protocols have been proposed for different types of multi-hop networks (ad-hoc, sensors, vehicular or DTN). Most of these protocols rely on the periodical update of a cost function, the algorithm seeking for the minimal cost paths across the network. The cost function can be based on any network-related parameter (delay, number

This work was funded by SMART-BAN project (Labex SMART) <http://www.smart-labex.fr>. En extended abstract of this work [2] has been published in ACM MSWIM 2015.

of hops, congestion, stability, QoS parameters, etc.), system-related parameter (e.g. battery level, temperature or available memory), or application level (e.g. security or measured data). The algorithm can rely on a local vision of the network, each node selects locally the path that exhibits the smallest cost to reach the destination, or it computes following a global optimization process that tries to minimize the total, network-wide, cost. Multiple metrics can be combined, generally through a linear combination (weighted sum).

In most cases, in-network mobility is handled by updating frequently the measured parameters and the cost, and by selecting dynamically the best path. Human body mobility, however, has a quasi-periodical behavior that could be exploited. Links appear and disappear quickly, in a quasi-regular pattern and learning this pattern, its regularities and its deviations, could influence the protocol design, reducing the frequency of cost updates. Such patterns were already utilized in the area of interplanetary networks (IPN), which led to standard proposals in the DTN field. However, on-body wireless channel is far less reliable and less predictable than the satellite-to-satellite link and mobility has a periodical component but is never perfectly regular. Therefore even though DTN paradigms are clearly of interest, adaptations are necessary.

### III. BROADCAST STRATEGIES

Based on the amount of knowledge that is available, the broadcast algorithms for DTN are divided in two big families [3]: dissemination (a.k.a. flooding) algorithms and forwarding (a.k.a. knowledge) algorithms. Flooding consists in disseminating every message to every node in the network. This strategy is adopted when no knowledge on the networks is available. Flooding maximizes the probability that a message is successfully transferred to its destination(s) by replicating multiple copies of the message, hoping that one will succeed in reaching its destination.

While providing near-optimal performance regarding delay, the primary limitation of these protocols is their energy consumption and their overhead due to excessive packet transmissions. Blind flooding algorithms generate numerous useless transmissions, while more intelligent algorithms usually rely on the knowledge of the nodes movement to predict spatio-temporal connectivity. The more precisely the mobility pattern is characterized, the more optimized the dissemination will be. However, acquiring this information also has a cost and there is a subtle balance to find between duplicate data packets and control messages.

#### A. Related works

Optimizing flooding in wireless multihop networks has been a constant concern over the past decades. In their seminal paper [15], Ni *et al.* the *broadcast storm* problem in mobile ad-hoc networks: blind flooding generates numerous transmissions all over the network in the same time frame, causing collisions, increased contention and redundant transmissions. Ni *et al.* analyze different strategies to alleviate this effect.

Each node can simply condition its retransmission to a constant probability, or take a forwarding decision based on the number of copies received during a certain time frame, on the distance between the source and the destination or on the location of the nodes if it is available. They also examine the effect of partitioning the network in 1-hops clusters, the cluster heads forming a dominating set among the network. The dominating sets-based approaches will be the source of numerous contributions, but is not really relevant in WBANs, as the size and diameter of the network remains very limited.

Many works have explored the probabilistic flooding approach, starting from Haas *et al.* [5], who identify the existence of a threshold on the forwarding probability below which probabilistic flooding fails. Sasson *et al.*, in [18] will push the analysis further using percolation theory to find the threshold that triggers such phase transition on random graphs. These works are not directly applicable to WBANs, as the mobility pattern is very different and non-homogeneous, but they show that the networks shape and dynamics have an influence on the optimal forwarding probability.

However, even with a probabilistic transmission, a node may receive multiple instances of the same message from various paths. Vahdat *et al.*, in [20], introduce *epidemic routing*. They suppose that a MANET is formed by multiple mobile clusters that eventually meet. Inside a cluster, they disseminate information through basic flooding and when two nodes belonging to different clusters meet, they exchange a *summary vector* containing the messages ID they already possess. They then exchange only the missing messages that will reach new clusters this way. Considering that the nodes were divided in connected groups without any guarantee from global connectivity was a first step towards delay-tolerant networks.

When connection is intermittent, unicast routing towards a single destination cannot be based simply on a classical shortest path algorithm like link-state routing. Indeed, the time required to collect all links information is too high compared to the links intermittency. That's why several papers such as [10], [7] propose and evaluate mobility-aware shortest path algorithms. Besides the chosen strategies, these contributions confirmed that the transmissions efficiency increases with the knowledge of the mobility pattern. In other words, knowing how nodes move allows to reduce the number of unnecessary transmissions and the delivery delay.

In [6], the authors compared various strategies for controlling flooding in delay tolerant networks. They compare basic probabilities approaches with time-to-live approaches in terms of number of hops or time stamps and with explicit notification when the destination has received the message. Further more, [19] proposes an interesting flooding-forwarding strategy, **Spray and Wait**. The spray phase (flooding) is as follows: for every message originating at a source node, copies are initially spread and forwarded by the source and possibly other nodes receiving a copy to distinct relays. In the wait phase (forwarding), if the destination is not found in the spraying phase, each node carrying a message copy performs a direct transmission (will forward the message only to its destination). A variant of this algorithm is *Spray*

and *Focus*. The forwarding requirement is to have some knowledge of the network, called in this article as "utility function" that represent how much the node is useful to reach the destination. The authors of [4] propose another strategy to optimize broadcasting in DTN: the *k-neighbor* broadcast scheme, in which a packet is retransmitted if and only if the number of neighbors present exceed a threshold,  $K$ , and if at least one of these neighbors did not receive the message yet. Even if the implementation is different, we find back here the concepts behind epidemic routing and the notion of not systematically transmitting a packet to reduce the number of transmission. The simulation-based evaluation shows a good performance and study the influence of the threshold on the number of neighbors,  $K$ . Yet, this threshold is the same all across the network.

In the past decade, several routing protocols have been proposed for WBANs that can be classified with respect to their aims in different categories [11], [12]: temperature based routing protocols, cross layer, cost-effective routing protocols, QoS-based routing protocols. Most of these proposals are not suitable when sensors are external (on the body and not in the body). DTN like solutions [1] and many other WBAN proposals (such as EDSR [13]) do not take into account the mobility of the human-body: during the transmissions of the message over the path already computed, disconnections can happen causing failures.

The authors of [17] study the case of unicast routing under the assumption that postural changes provoke network disconnections. They propose a store-and-forward approach and use a probabilistic proactive routing approach, defining stochastic links costs. They show through simulation and experiments that taking into account the particular mobility of the body improves transmission delay when compared to a traditional probabilistic algorithm for DTN. The same authors present in [16] an adaptation of their algorithm taking into account location-aware networks: a node forwards packets if this action results in bringing the data closer (physically) to the destination.

### B. Old and new broadcast strategies

This excerpt from the literature shows that there is room for optimizing the flooding procedure by taking into account the specific mobility pattern. When a node receives a flooded packet, it needs to take a decision whether to forward it or not, to which of its neighbors and when. However, acquiring the necessary information on the nodes mobility is not free. It either requires a precise nodes localization mechanism, which is typically achieved by measuring round-trip time of flights between couple of nodes in IR-UWB body area networks, or requires to rely on *hello messages* to learn connections/disconnection patterns. In both cases the improvement realized by the mobility pattern characterization could be lost by the necessary control traffic. That's why we wished to compare various strategies with different levels of knowledge, in a scenario that is representative of a real WBAN.

More specifically, we will compare the following strategies:

- *Flooding*: Nodes retransmit each received packet while its TTL (Time-To-Live) greater than 1.

- *Plain flooding*: Nodes retransmit the packet only the first time they receive it. So, each packet will be retransmitted only one time by each node.
- *Pruned flooding*: Nodes retransmit each packet to only  $K$  nodes chosen randomly.
- *Probabilistic flooding ( $P=0.5$ )*: Nodes decide to retransmit packets based on a probability  $P$ . For each received packet, nodes choose a random variable and compare it with  $P$ . For simulations, we set this probability to a constant value equal to 0.5.
- *Probabilistic flooding ( $P=P/2$ )*: Nodes decide to retransmit packets based on a probability  $P$  too. For each received packet, nodes choose a random variable and compare it with  $P$ . For this strategy,  $P$  decreases as the number of retransmissions increases. The first instance of the packet is retransmitted with a probability equal to 1 (initially  $P = 1$ , then with probability equal to 0.5 ( $P = P/2$ ) then 0.25, etc.
- *Tabu flooding*: For this strategy, extra information is added to the original message indicating for each node in the network if it has already received the message or not. Consequently each node retransmits only to the nodes not yet covered.
- *EBP (Efficient Broadcast Protocol)*: We adapted this strategy to WBAN context from [4]. In this protocol nodes decide to retransmit a packet under two conditions: Only when they are surrounded by at least  $K$  neighbors, and if at least one of these neighbors has not received the packet yet. While in [4] the threshold value  $K$  is fixed and uniform across the network, we adapt its value to the WBAN environment, setting a higher value for the gateway (*sink*) and a lower one for the peripheral nodes who are less likely to encounter uncovered nodes. In our implementation the  $K$  value is set to 3 for the gateway (chest), to 1 for the head, ankle and the wrist (peripheral nodes) and to 2 for the other nodes. To update neighbors' map, nodes relies on control message (*Hello message*), while updating the information concerning who did or not received the message relies on both control and data messages.
- *MBP (Mixed Broadcast Protocol)*: We proposed this strategy in [2] as a mix between the *dissemination-based* and *knowledge-based* approaches. The broadcast begins as a basic flooding algorithm (i.e *Flooding* strategy). When a node receives a message, it checks the number of hops  $NH$  this message has traveled since its emission by the *sink*, and compares it with a threshold on the number of hops  $\Delta$ :
  - As long as  $NH < \Delta$ , node forwards the packet, using simple flooding, simply transmitting the packet to all neighbors in its range provided that its TTL greater than 1.
  - When  $NH = \Delta$ , node waits during a time  $T$  to receive up to  $Q$  acknowledgments.
    - \* if it receives a number of acknowledgments greater or equal than  $Q$ , the node stops rebroadcasting the message.

- \* if it fails to receive  $Q$  acknowledgments, it simply rebroadcasts the packet.
- When  $NH > \Delta$ , node applies the same strategy as when  $NH = \Delta$ , but it sends in addition an acknowledgment to the node it received the packet from.
- **OptFlood (Optimized Flooding)**: Our novel strategy described below that outperforms the strategies adapted from DTN or WSN but also the MBP strategy we proposed in [2].

The simulation results we detail in Section IV allow us to come out with an interesting set of conclusions:

- Regarding the average end-to-end delay, i.e. the time required for a message to reach all the nodes, both dissemination (flooding-based approaches) and knowledge-based (EPB-like) protocols spend more time to cover the peripheral nodes (which represent about 20% of the network) compared to the rest of the network.
- When looking at the volume of control traffic, using an EPB-like protocol makes the number of unnecessary retransmissions increase, due to an increased volume of control packets in the network used mainly to discover neighbors.
- For *Flooding*, it is very important to set a good *TTL* value in order to control the retransmissions. However, it ensures a fast network coverage, indeed, it represents the best end-to-end delay for all postures.
- With *EBP*, using a relatively high threshold on the number of neighbors required to transmit ( $K$ ), works well only if the Hello-interval  $I$  is short enough to have a "realistic" representation of the network. Otherwise, with large Hello-interval only the last Hello message received is reliable, while the other neighbors saved in the neighbor-table could be obsolete.
- Knowledge-based algorithms include control messages such as neighborhood discovery messages. This approach does not work well in the WBAN scenario because it may happen that during the communication (exchange of control messages) to "take decision" the connections disappear. Moreover this increases further transmissions, receptions, and related problems such as collisions and interferences.
- For *MBP*, since it mixes flooding-based and knowledge-based approaches, a significant amelioration is noticed. Results show a good percentage of covered nodes (close to *Flooding* strategy) with 50% less transmissions and receptions comparing to *Flooding*. However, looking at the end-to-end delay, we noticed that although *MBP* shows a better delay than the other strategies, the difference with the end-to-end delay of *Flooding* strategy remains important.

These conclusions guided us to define a novel protocol *OptFlood (Optimized Flooding)* by taking into account the strengths and weaknesses of the basic strategy *Flooding*. The latter shows far better performances considering the coverage of the network and specially the end-to-end delay. Unfortunately, the important number of transmissions and receptions

makes the use of this strategy inadequate for WBAN environment where a low energy consumption is required. *Optimized Flooding* is a revised version of *Flooding* whose purpose is to keep the good end-to-end delay given by *Flooding* while lowering energy consumption with the simplest way and the minimum cost.

*Optimized Flooding* uses two variables: **CptLocal** and **cptGlobal**, described below:

- **cptGlobal** Each message carries *cptGlobal* that can be read and updated by all nodes in the network that receive this message.
- **CptLocal** is a local variable to each node. It can be read and modified by its owner. *cptLocal* is set to *cptGlobal* of the last received message. In other words, *cptLocal* is the local copy of *cptGlobal*.

Our algorithm proceeds as follow: When a node receives a message:

- For the first time: the node increments message' *cptGlobal*, updates its *cptLocal* and rebroadcasts the message.
- For the  $x \geq 2$  time:
  - The node verifies if it already incremented the *cptGlobal* in this copy of the message. In case not, the node increments the value of *cptGlobal*.
  - The node compares *cptGlobal* with *cptMax* which corresponds to the number of nodes in the network and to the maximum value that *cptGlobal* could reach.
    - \* If  $cptGlobal = cptMax$ , this means all nodes already received the message ie 100% of the network is covered so the node stops rebroadcasting this message and all the incoming copies of this message.
    - \* If  $cptGlobal < cptMax$ , this means that there are still not covered nodes. Then the node compares the value of *cptGlobal* with the value of *cptLocal*:
      - if  $cptGlobal \leq cptLocal$ : which means a message with a *cptGlobal* equal to the current *cptLocal* has already been broadcasted in the neighborhood. This copy of message is obsolete hence the node discards it.
      - if  $cptGlobal > cptLocal$ : node updates its *cptLocal* with the new value of *cptGlobal* and rebroadcasts the message in the neighborhood with the freshest value of *cptGlobal*.

#### IV. SIMULATION SETTINGS AND RESULTS

In order to test the algorithms described above (Section III-B) in a specific WBAN scenario, we implemented them under the Omnet++ simulator. Omnet++ includes a set of modules that specifically model the lower network layers of WSN and WBAN through the Mixim project [9]. It includes a set of propagation models, electronics and power consumption models and medium access control protocols. The MoBAN framework for Omnet++ adds mobility models for WBANs of 12 nodes in 4 different postures. Unfortunately studying in deep the MoBAN code, we discovered that it mainly

focuses on the mobility resulting from the change of position, rather than describing coherent and continuous movements. Besides, it models the movement of each node with respect to the centroid of the body and the signal attenuation between couples of nodes is approximated with a simple propagation formula that is not accurate enough to model low-power on-body transmission. It does not model absorption and reflection effects due to the body, alterations due to the presence of clothes and eventually interference from other technologies at the same frequency since 2.4 GHz is a crowded band.

### A. Channel model

We therefore decided to implement a realistic channel model published in [14] over the physical layer implementation provided by the Mixim framework. This channel model of an on-body 2.45 GHz channel between 7 nodes, that belong to the same WBAN, using small directional antennas modeled as if they were 1.5 cm away from the body. Nodes are assumed to be attached to the human body on the head, chest, upper arm, wrist, navel, thigh, and ankle.

The nodes positions are calculated in 7 postures: walking (walk), walking weakly (weak), running (run), sitting down (sit), wearing a jacket (wear), sleeping (sleep), and lying down (lie). these postures are represented on Fig. 1. Walk, weak, and run are variations of walking motions. Sit and lie are variations of up-and-down movement. Wear and sleep are relatively irregular postures and movements.

Channel attenuation is calculated between each couple of nodes for each of these positions as the average attenuation (in dB) and the standard deviation (in dBm). The model takes into account: the shadowing, reflection, diffraction, and scattering by body parts.

### B. Simulation Settings

Above the channel model described in the previous section, we used standard protocol implementations provided by the Mixim framework. In particular, we used, for the medium access control layer, the IEEE 802.15.4 implementation. The sensitivity levels, header length of the packets and other basic information and parameters are taken from the 802.15.4 standards.

Each data point is the average of 50 simulations run with different seeds. The simulations are performed with only one packet transmission from the gateway (i.e the sink). The transmission power is set at the minimum limit level  $-60$  dBm that allows an intermittent communication given the channel attenuation and the receiver sensitivity  $-100$  dBm. With this value of transmission power, we guarantee that at each time  $t$  of the simulation, we have a connected network and at the same time we ensure a limited energy consumption.

### C. Simulation results

To compare between the different broadcast strategies presented in the previous section III-B (<sup>1</sup>), we studied for each one three basic parameters:

<sup>1</sup>Note that we performed an additional study on the timer used in the MBP strategy in order to choose the one offering the best compromise. The results of this study are reported in VII-B

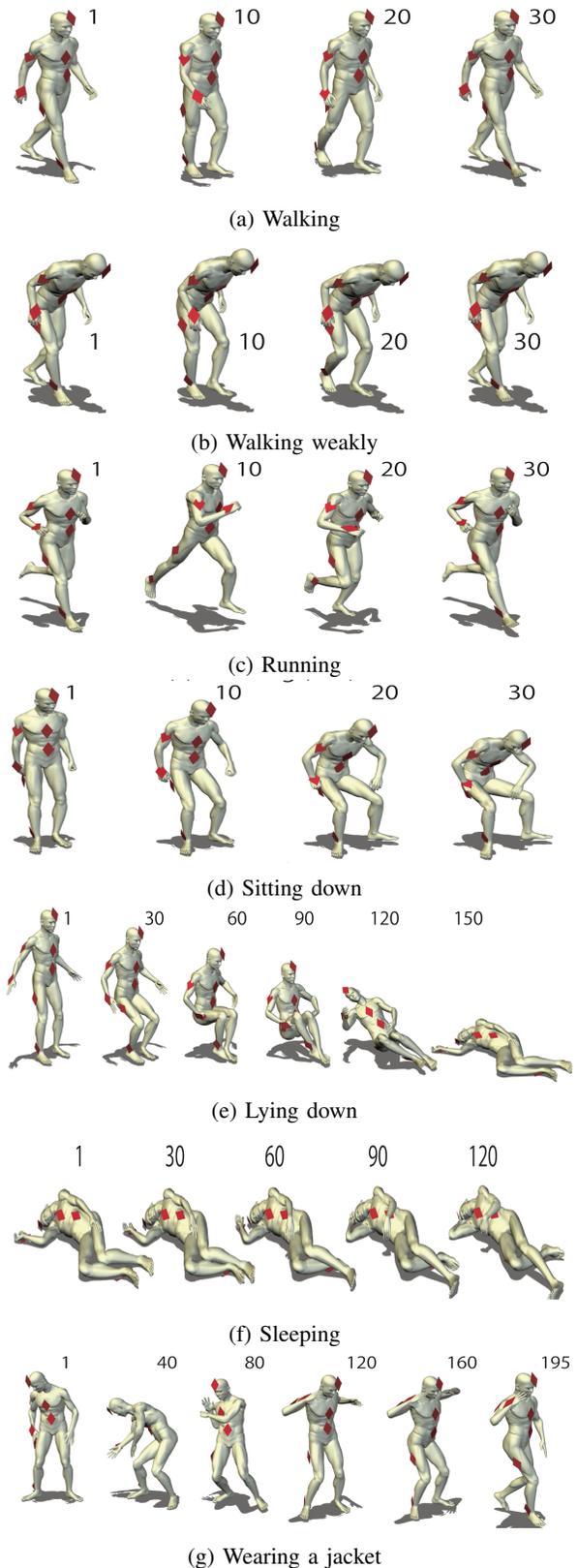


Fig. 1: Postures used in [14] to model the WBAN channel (Pictures source: [14])

- **Percentage of covered nodes:** Since our unique source is the sink which broadcasts a single message in the network, we therefore calculate the percentage of nodes that have received this message.
- **End-to-end delay:** The average End-to-end delay is the time a message takes to reach the destination (i.e. every node except the sink).
- **Number of transmissions and receptions:** This is a key parameter, it helps us to have an estimation of the energy consumption at each node and for each posture for all the strategies.

### 1) Percentage of covered nodes

We first compare the percentage of covered nodes after the broadcast of the message from the sink. Figures 2 and 3 show this percentage variations for each algorithm in function of the *TTL* "Time-To-Live" set for the packets. The choice of the *TTL* is justified by the fact that it is the only parameter in common between all the considered algorithms.

The first observation is about the value of the percentage of covered nodes when *TTL* is equal to one. As shown in figures 2 and 3, it is almost the same for all strategies and is equal to 51% except for *EBP*. This value shows that half of the network is covered after the first transmission by the sink. For these simulations, human body posture is *walk posture*. In this posture, the sink has at least three neighbors that can reach at the first time which explain the percentage of covered nodes with *TTL* = 1 (the other postures are shown in the figure 4). The slight decrease seen with *EBP* strategy is due to packet losses caused by collisions with control packets.

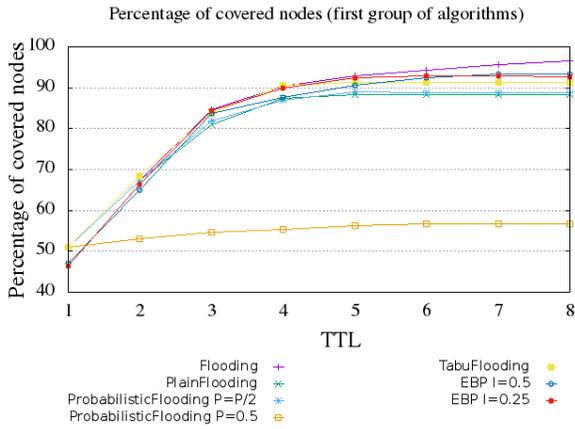


Fig. 2: Percentage of covered nodes (first group)

The strategy *Flooding* represents the reference of flooding-based strategies. With this strategy nodes continue to retransmit the packet while its *TTL* is greater than 1, regardless of the past. For this reason, and in absence of conditions on retransmissions, strategy *Flooding* shows the highest percentage of covered nodes. For *EBP*, *I* represents the interval between two successive control message *Hello message*. Each node send periodically a *Hello message* to discover the neighborhood since the decision to rebroadcast or not the message depends on the neighborhood state. *EBP* shows good results with a slight increase when  $I = 0.25$ . Indeed, with  $I = 0.5$ , the node

has a less realistic view of the neighborhood which could force it to decide to not rebroadcast the message.

A notable observation is that *EBP* stops automatically the broadcast when the each node's neighbors are covered, while the classical *Flooding* algorithm (Figure 2) has to wait until the *TTL* of all packets reaches 1.

*Plain Flooding* and *Tabu Flooding* coverage starts to increase with the *TTL* and stabilize rapidly. Both algorithms stop even when the network is not fully covered. In case of *Plain Flooding* a few number of transmissions is allowed, only once, which corresponds to the first reception of the message. For *Tabu Flooding*, nodes only rebroadcast messages, if it is intended for them, and only for uncovered node. So in case the destination is not in the sender's range, the message will be dropped by the other nodes. As it is very hard for these two algorithms to reach a full coverage, the notions of completion delay and the required amount of transmissions to cover the whole network are meaningless. Therefore these algorithms will be left out of the subsequent evaluations.

The *Probabilistic Flooding* with decreasing probability ( $P = P/2$ ) achieves a better coverage than the constant probability version with  $P = 0.5$ . In fact, with *Probabilistic Flooding*,  $P$  initially equal to 1, which means that the probability to rebroadcast a message for the first time is equal to 1 too, since the chosen random variable is necessarily less than  $P$ .

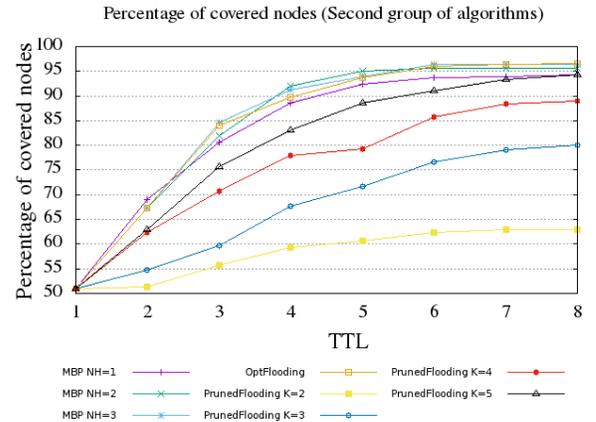


Fig. 3: Percentage of covered nodes (Second group)

For *Pruned Flooding* strategy,  $K$  represents the number of nodes, a node chooses randomly to forwards the message to. In order to study the influence of this parameter on the behavior of the strategy, we varied the value of  $K$  from 2 to 5. Figure3 shows that the number of nodes to choose randomly has a direct effect on the coverage percentage. With  $K = 2$  and  $K = 3$  it is almost impossible to cover all the nodes even with higher values of *TTL*. When  $K = 5$ , we have better results but still the lowest percentage comparing to the other strategies.

For *MBP*,  $NH$  represents the threshold on the total number of hops the message passed through. We therefore studied the influence of this parameter on *MBP* results. Figure3 shows a slight variation depending on  $NH$  parameter especially

between  $NH = 2$  and  $NH = 3$ . In general, *MBP* shows a good percentage of covered nodes, specially with low values of *TTL*, for example, with  $TTL = 3$ , the strategy covers between 80% and 85% (with different values of  $NH$ ) of the nodes.

The novel strategy *Optimized Flooding* has good results. For example, with  $TTL = 4$ , 90% of the network is covered. As this strategy is an adaptation of the strategy *Flooding*, the percentage of covered nodes values and even the variation in function of the *TTL* are almost the same.

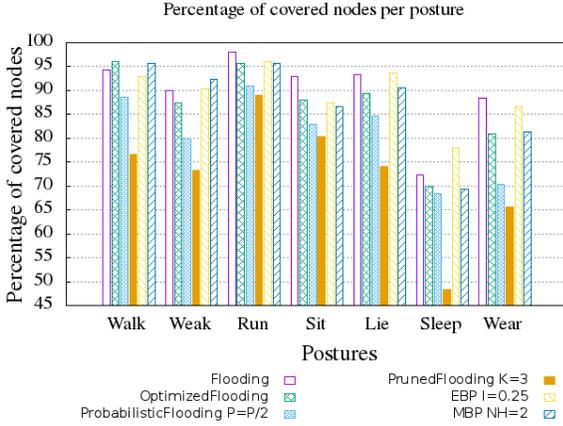


Fig. 4: Percentage of covered nodes per posture

Figure 4 represents the network coverage percentage in different postures.

The lowest percentage, independently of the posture, is with *Pruned Flooding* strategy, with  $K = 3$  a low number of nodes are designated to rebroadcast the message, knowing that a node doesn't rebroadcast a message destined to another node.

*Probabilistic Flooding* shows a better percentage than *Pruned Flooding* strategy but still the lowest. Both strategies have higher variations among the postures due to their random nature.

Strategy *Flooding* shows the highest percentage in almost all postures. However, *MBP*, *EBP* and *Optimized Flooding* show good percentage of covered nodes between 70% and 96%. For these strategies and for the percentage of covered nodes parameter, even if a variation in function of the postures is observed, it still less significant than for the two other studied parameters: End-To-End delay (IV-C2) and number of transmissions and receptions (IV-C3). It is important to point that the sleep posture is a specific case that represents a static posture where some nodes are hidden and remain so all along the simulation. Even so, *Optimized Flooding* reaches 70% of covered nodes.

## 2) Average End-To-End Delay

Figure 5 represents the average End-to-End delay, i.e. the average time required for a message sent from the sink to reach a node.

As expected, *Flooding* algorithm has the best performance due to the huge amount of message copies in the network.

The strategy *Probabilistic Flooding* ( $P=P/2$ ) shows a higher end-to-end delay due to the decreasing amount of retransmissions with respect to the flooding.

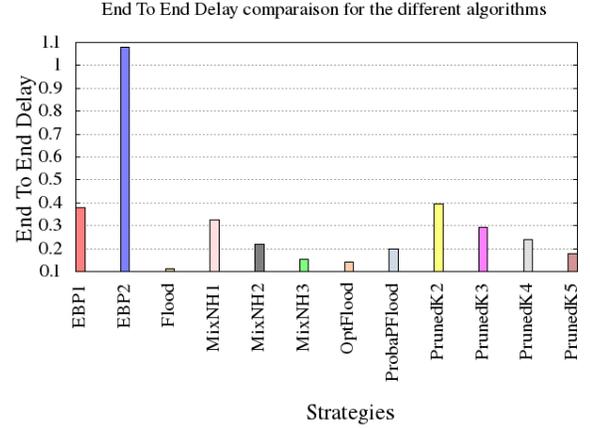


Fig. 5: Average End-to-End delay comparison between strategies

The end-to-end delay achieved by *EBP* is strictly dependent from the *Hello message* interval ( $I$ ). When  $I$  is short, for instance  $I = 0.25s$ , each time the connections in the model change, it is soon considered by nodes, resulting in a precise acquisition of the network state and a quick decision making.

*Pruned Flooding* end-to-end delay is also directly related to the value of  $K$  parameter: the higher is the value of  $K$ , the lower is the end-to-end delay. In particular in cases when  $K = 2$  and  $K = 3$ , the algorithm spends a lot of time trying to cover the whole network. This is due to the low number of nodes that receive the packet and are designated to retransmit.

*MBP* presents better performances in term of End-to-End delay than *EBP* and *Pruned Flooding*, even if, with this strategy, a node delays packet retransmission after  $T$  timer expiration. *MBP* End-to-End delay depends also on  $NH$  value. With  $NH = 3$ , *MBP* attempts to cover the nodes more quickly than for lower values, because, in this case, nodes use to rebroadcast the message faster based on simple flooding without delaying the retransmission.

Finally, *Optimized Flooding* has an end-to-end delay close to *Flooding* strategy and it is also the best End-To-End delay after *Flooding*. Our first goal is reached since the idea is to maintain the low End-To-End delay given by *Flooding*.

Figure 6 details the average End-to-End delay per node.

We can notice that all algorithms behave similarly: they spend more time to cover the peripheral nodes in the network (example: node at the ankle). Indeed, peripheral nodes are highly mobile and located at a greater distance from the sink than the other ones. As a result, even if an algorithm is able to cover the central part of the network in a very short time, it then has to wait for a connection opportunity with the last nodes.

*EBP* and *pruned Flooding* present the worst End-To-End delay. In case of *EBP*, nodes wait before retransmitting the message until all conditions are satisfied, i.e. the number of nodes in the neighborhood.

Both of *Flooding* and *Optimized Flooding* strategies have the lowest End-To-End delay for all the nodes. These results are especially important regarding the end-to-end delay of the

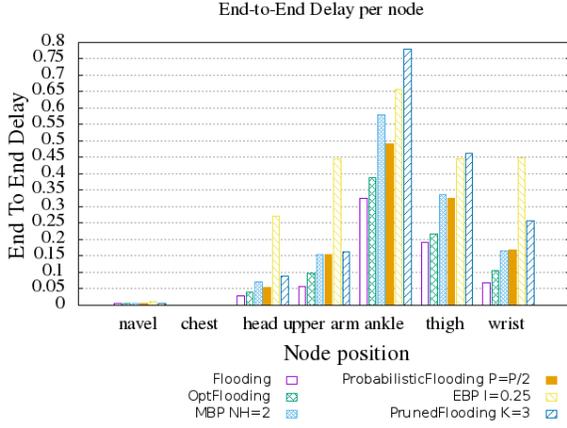


Fig. 6: Average End-to-End Delay per node

peripheral nodes. *Optimized Flooding* comes to deal moderately with this limit comparing to the other strategies including *MBP*.

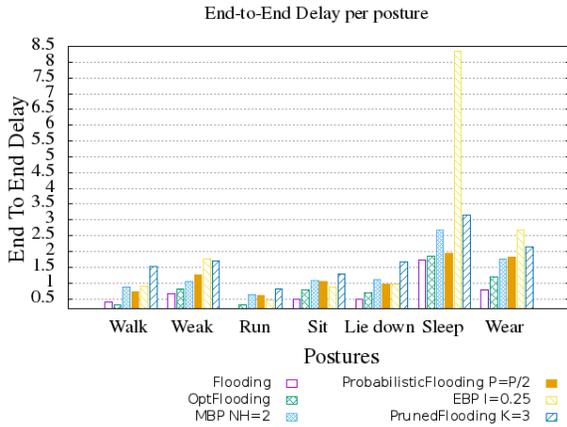


Fig. 7: Average End-to-End delay per posture

Figure 7 shows the average End-To-End delay per posture. *EBP* seems to perform better in the high mobility postures. Indeed, in *EBP*, because the number of neighbors is important, a peak is noticed in sleep position where some nodes are hidden and remain so all along the simulation, for that, nodes decide to not rebroadcast the message and keep it until all conditions are satisfied.

In other hand, *Flooding*, *Optimized Flooding* and *MBP* are the less affected by human body postures than *EBP* and *Pruned Flooding*. However, we notice better results (closest to *Flooding*) and less variations in function of the postures with *Optimized Flooding* than with *MBP*.

### 3) Number of transmissions and receptions

The number of transmissions and receptions is a key parameter for WBAN network. The number of transmissions reflects the channel load and also gives an indication about the amount of electro-magnetic energy that will be absorbed by the body. The number of receptions gives an indication on the energy consumption of the different nodes in the network and hence

on the devices autonomy, their capability to rely on a reduced size battery or even to harvest energy.

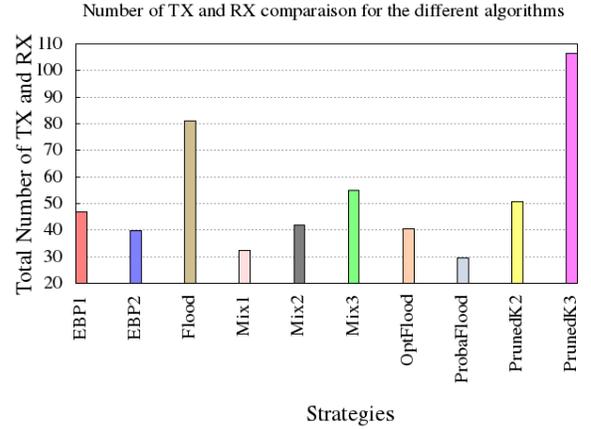


Fig. 8: Number of transmissions and receptions

Figure 8 compares the total number of transmissions and receptions for all the studied algorithms.

*Flooding* pays its good End-To-End delay and coverage performance as it exhibits the highest number which indicates a high energy consumption. It is the only algorithm that has no limitation on the number of transmissions besides the *TTL*.

The number of transmissions of *EBP* varies with *I*. In fact, *EBP* strongly relies on the transmission of *hello messages* that cannot be neglected, this causes collisions with data packets (especially with shorter interval *I*) leading to lots of dropped packets and to unnecessary retransmissions. However it is important to note here that these results only account for data packets transmissions and receptions.

The adaptive *Probabilistic Flooding* ( $P=P/2$ ) has a better performance than the classical *Flooding*, because the probability decrease limits the number of retransmissions as the packet travels in the network.

The energy consumed by *Pruned Flooding* depends on the parameter *K*: while for the end-to-end delay and coverage, the algorithm performs well with high values of *K*, in this case a visible reduction in number of transmissions and receptions is observed with lower values of *K*. For each retransmission, the node duplicates the packet as much as the number of nodes to choose randomly i.e as much as *K* value.

*MBP* does not suffer from the same issue, even if control messages such as acknowledgments are also necessary. In fact, acknowledgments are only sent when a packet is effectively forwarded, the control traffic is directly related to the network activity. *MBP* performance therefore mostly depends on the *NH* parameter and lowering this value results in a better energy efficiency.

Finally, with the strategy *Optimized Flooding*, a significant decrease in the number of transmissions and receptions is observed: 50% compared to *Flooding*. It's an important result. Remember that the unique limitation of *Flooding* strategy is related to the number of transmissions and receptions in other words the energy consumptions.

Figure 9 shows the total number of transmissions and

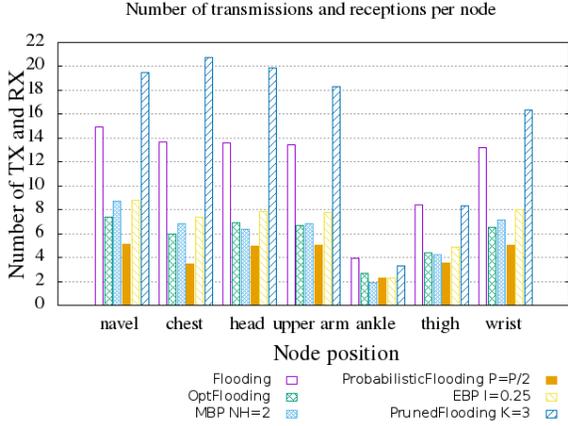


Fig. 9: Number of transmissions and receptions per node

receptions for each node.

We observe that nodes at the center of the network (navel and chest) have the highest number of transmissions and receptions, their position allows them to communicate with more nodes. On the contrary the node on the ankle is the one with the lowest number of transmissions and receptions because standing in the periphery has few occasions to communicate with the rest of the network which explain again the highest End-To-End delay for this node. Besides, when the message reaches this node (node at the ankle), its *TTL* has already been decremented multiple times and it generally has no uncovered neighbor as far as knowledge-based algorithms are concerned.

*Flooding* and *Pruned Flooding* present the highest number for all nodes. Nodes continue to rebroadcast messages while *TTL* is greater than 1. It is worst with *Pruned Flooding* where each message is duplicated three times (these results are for  $K = 3$ ).

*Probabilistic Flooding* ( $P=P/2$ ) performs better than *Flooding* and *Pruned Flooding*. Indeed, the central nodes happen to transmit more than the others, but their neighbors will limit their transmission probability immediately, resulting in a lower number of receptions. This algorithm therefore better distributes the consumption across the network.

One of the main weaknesses of the flooding algorithms that their performance strongly depends on the *TTL*. Increasing the *TTL* lets the total amount of transmissions and receptions rise very quickly. This issue is partially solved in the adaptive *Probabilistic Flooding* ( $P=P/2$ ): the probability decrease works as an automatic brake, drastically reducing the transmission probability.

For *MBP* and *EBP* the total number of transmissions and receptions is 50% less than *Flooding* and *Pruned Flooding*. Indeed, *MBP* uses the information from the acknowledgement to control and stop the transmissions, while *EBP* uses information included in the control messages (*Hello message*).

Even if, for our strategy *Optimized Flooding*, no control message or acknowledgement are exchanged between nodes, we observe a significant decrease on the total number of transmissions and receptions per node. The decision to stop the retransmission is related to the validity of the information

contained in each copy of the message. This information is used to stop sending the message if the network is fully covered as well as eliminate the back and forth of the same copy between two nodes, especially between highly connected nodes.

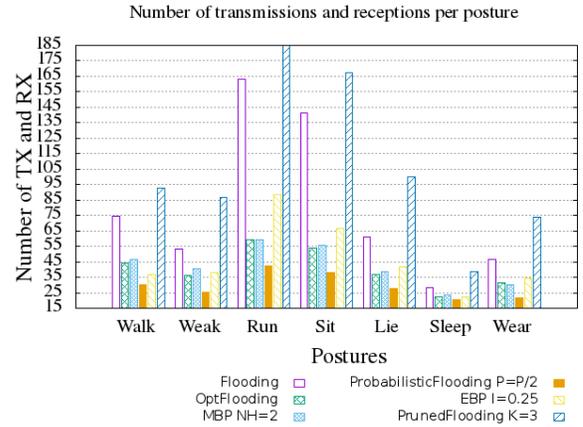


Fig. 10: Number of transmissions and receptions per posture

Figure 10 represents the number of transmissions and receptions for each posture. It's important to study each strategy behavior in function of the wearer posture.

We can notice, on this graph, for all strategies, the highest number is for *Run* and *Sit* postures. In fact, *Run* posture represents the highest mobility, even with frequent disconnections, nodes are able to meet more frequent to exchange messages. In *Sit* posture, nodes are closer to each other so they are able to exchange packets with a minimum loss.

*Sleep* posture presents the lowest number of transmissions and receptions for all strategies. At a first glance, it seems an advantage for this posture but reconsidering the observations related to the percentage of covered nodes 4 and to the end-to-end delay 7, this is rather related to the high disconnections between nodes.

The two strategies *MBP* and *Optimized Flooding* are less independent from the postures (ie less variations). Still, *Optimized Flooding* presents a lower number of transmissions and receptions than *MBP* except for some postures which could be explained by the fact that *Optimized Flooding* covers more nodes as shown in figure 4.

## V. TOTAL ORDER RESILIENCE

In this section we push further our study in order to detect the capacity of the strategies described to be reliable and to ensure the total order message delivery. That is, messages sent by the sink in a specific order should be received by each node in the system in that specific order.

We stress strategies with transmission rate ranging from 1 to 1000 packets per second broadcasted from the sink to the other nodes. We fixed the MAC buffer capacity to  $100^2$ .

We study two parameters:

<sup>2</sup>Additional simulations for various MAC buffer capacities are presented in Section VII-A.

- 1) Percentage of covered nodes: we calculate the number of messages received at each node than we deduce the percentage.
- 2) Percentage of desequencing: the percentage of messages received in the wrong order.

a) *Percentage of covered Nodes*: Figure 11 presents the percentage of covered nodes in function of packets number per second.

Contrary to the expectations, all studied strategies behave similarly: when the packets rate goes to 1000 packets per second the percentage of covered nodes almost linearly decreases to 10%. At 100 packets per second, percentage of covered nodes barely exceeds 50%.

For *Flooding* and *MBP* percentage stagnates until 10 packets per second than the curve starts decrease. However, For *Optimized Flooding* and *Probabilistic Flooding*, percentage remains constant until 20 packets/s

*PrunedFlooding* presents the lowest percentage. Other strategies results are quite close at the beginning and starting from certain point curves overlap and converge to the same point.

b) *Percentage of Desequencing*: Figure 12 presents the percentage of desequencing.

Contrary to results about the percentage of covered nodes, from curves, three phases can be seen:

- At the beginning, all strategies present 0% of desequencing. At this point, strategies are able to handle more than one packet in the network.
- Then, from certain point (depending on the strategy), this percentage increases linearly. Here, referring to Figure 11, percentage of covered nodes decreases due to collisions and packets loss. Sequencing is no longer ensured.
- Finally, from certain point, percentage decreases to converge to 0% again. Also, referring to figure 11, this is due to the fact that few packets are received.

*MBP* strategy presents the highest percentage of desequencing starting from 2 packets/s.

Percentage of desequencing increases starting from 10 packets/s for *Flooding* and *Pruned Flooding*, and from 20 packets/s for *Optimized Flooding* and *Probabilistic Flooding*.

*PrunedFlooding* low percentage of desequencing is due to a low percentage of covered nodes.

*Optimized Flooding* strategy presents the lowest percentage of desequencing compared to *Flooding* and *MBP* strategies.

## VI. CONCLUSION AND FUTURE WORKS

In this paper we evaluated through simulation the performance of several DTN-inspired broadcast strategies in a WBAN context. Our simulations, realized with the Omnet++ simulator, the Mixim framework and a WBAN channel model proposed in the literature and issued from realistic data-sets, allowed us to compare flooding-like strategies that forward packets blindly and differ mostly by how their stopping criterion with a representative knowledge-based algorithm, EBP, which relies on the knowledge of the neighborhood of each node and its evolutions. The simulations realized over a 7 nodes network in 7 types of movements allow a fine

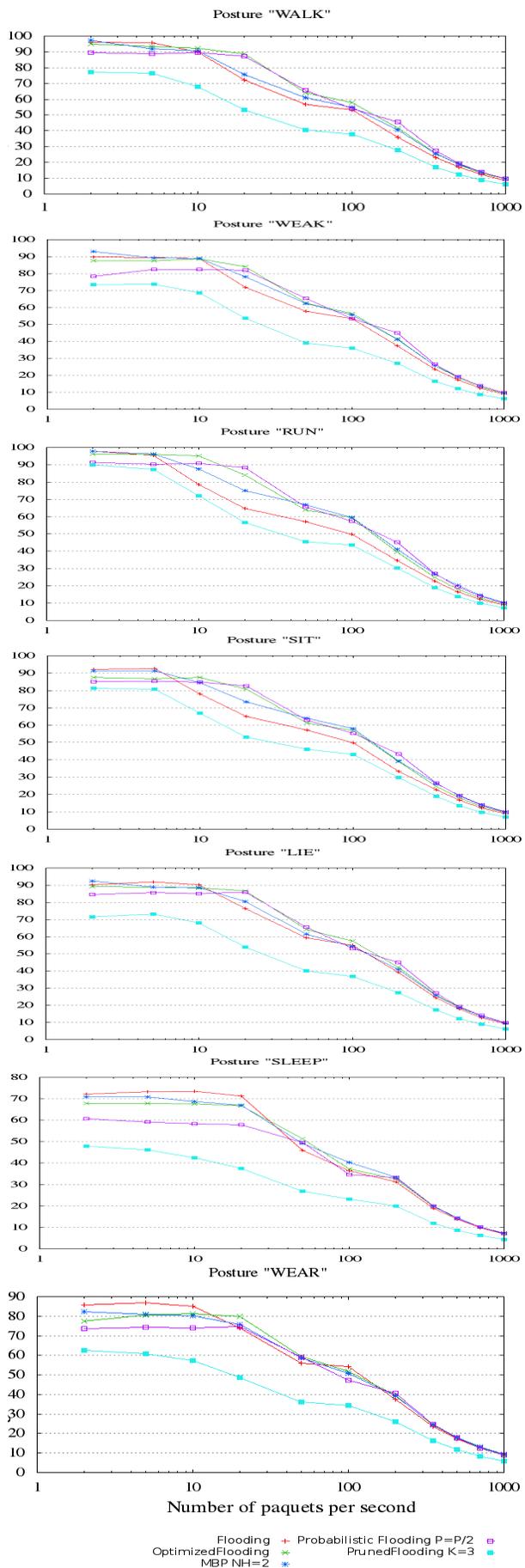


Fig. 11: Percentage of covered nodes per posture

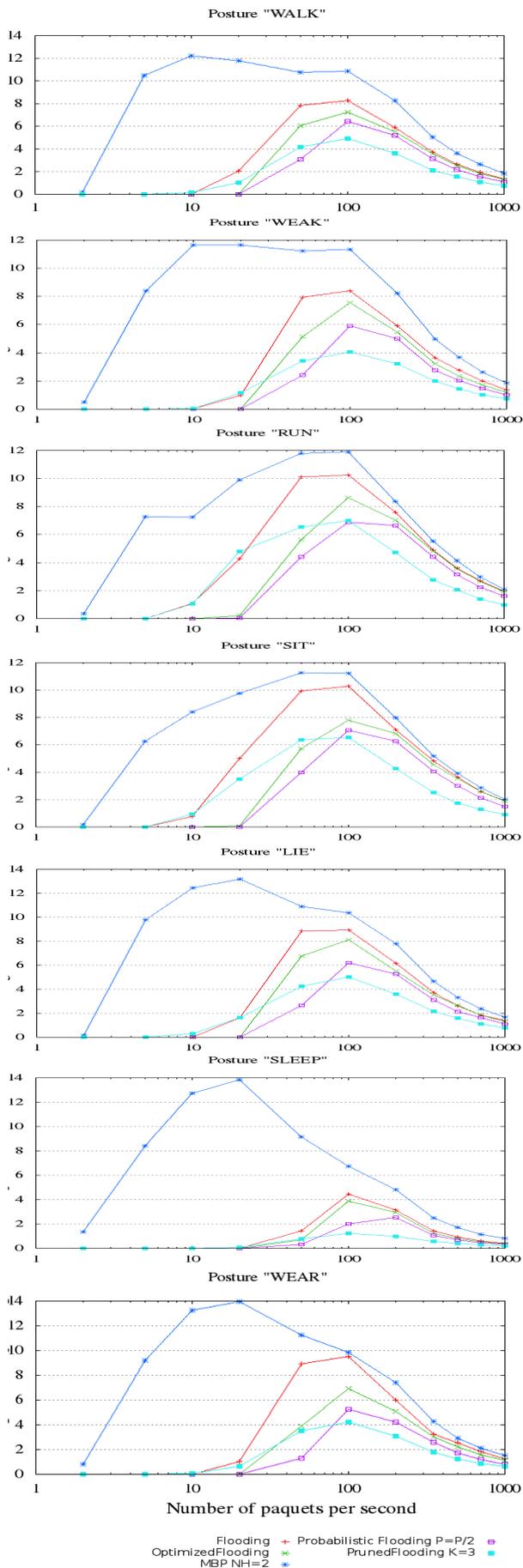


Fig. 12: Percentage of Desequencing per posture

characterization of the compromise that exists between the capacity to flood the whole network quickly and the cost induced by this performance. Simulations also allowed us to identify some less intuitive behaviors: for all strategies, most of the time is spent trying to reach leaf nodes, which makes us think that the key lies in adaptive algorithms that are able to mix different strategies.

We described a novel protocols that relies on such an adaptive approach: *MBP*, for Mixed Broadcast Protocol that applies a more aggressive strategy in the center of the network, where connections are more stable, and becomes more cautious at the border of the network, where a blind transmission has a good chance of success. This protocol has been reported in the extended abstract of this work [2]. Additionally we proposed the *Optimized Flooding* protocol that improves the performances of *MBP*. Furthermore, we propose a preliminary study related to the total order reliability of the existing strategies. This study advocates for further investigations on this specific direction.

As future works, we will adapt this work to the context of IR-UWB transmissions, which is a promising technology for WBAN. Another future work would be a detailed study of existing channel models [21] and a comparison with the channel model. Furthermore, another interesting future direction would be to consider collisions among multiple WBANs [22], [8].

## VII. APPENDIX: ADDITIONAL SIMULATIONS

### A. Impact of MAC queue length

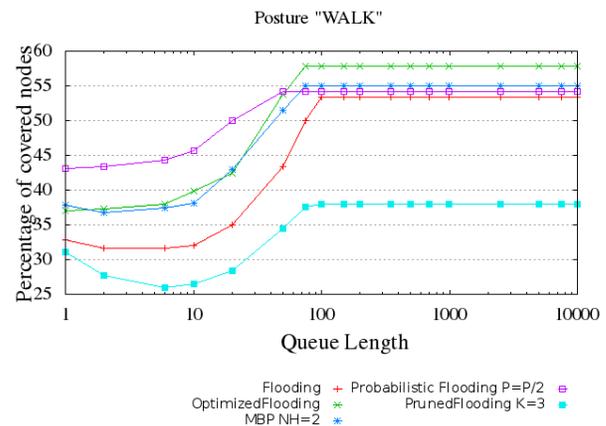


Fig. 13: Percentage of covered nodes for different queue length

We run simulations with 100 packets per second and we vary MAC buffer capacity from 1 to 10000 packets. Figures 13 and 14 show respectively the percentage of covered nodes and the percentage of desequencing function of the MAC queue length. *Flooding* shows less resilience to high transmission rate. *Probabilistic Flooding* is less affected by queue length in MAC layer. Our novel strategy stabilizes to the highest highest percentage of covered nodes. *MBP* shows again a high percentage of desequencing.

We can conclude that MAC queue length influences our strategies performance while its value is lower than the transmission rate. For a queue length superior than the transmission

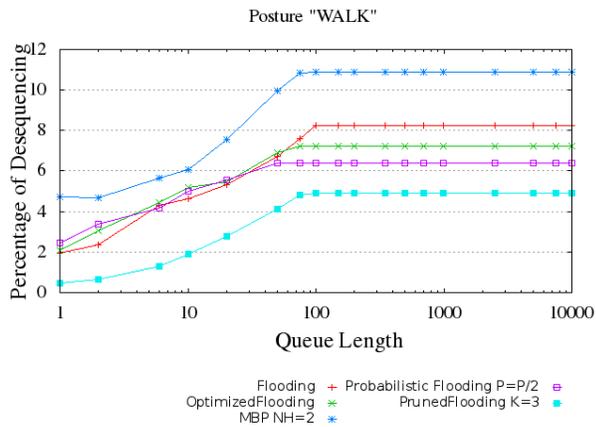


Fig. 14: Percentage of Desequencing for different queue length

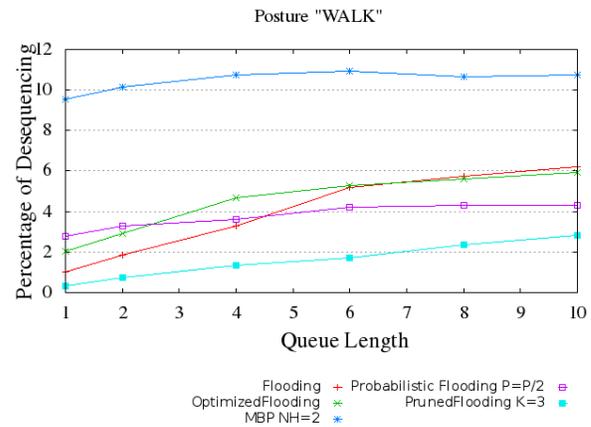


Fig. 16: Percentage of Desequencing for different queue length

rate, no variation is observed in the percentage of covered nodes nor in the percentage of desequencing.

In the following we fix the transmission rate at 10 packets per second (the rate usually used in the biomedical applications) and a variation of the queue length from 1 to 10. The simulations results are proposed in Figures 16 and 15.

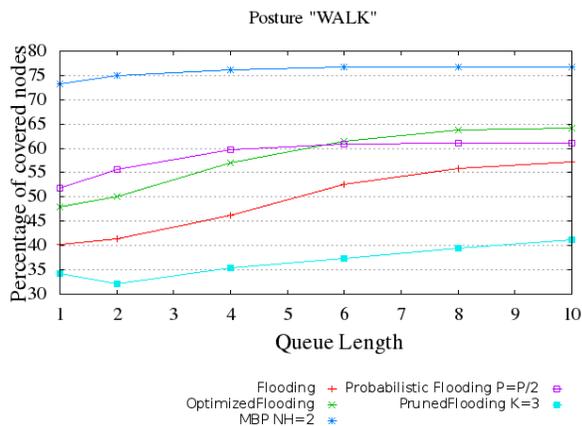


Fig. 15: Percentage of covered nodes for different queue length

### B. MBP: Timer study

In the following we focus on the timer used for MBP strategy. The objective of this study is to have a closer inside on the MBP parameters (i.e. timer) impact on the overall performances of this strategy. Recall that in MBP strategy, starting from a certain threshold on the number of hops ( $NH$ ), a node delays message broadcasting after a timer expiration. Thus, we will examine the influence of this timer on MBP strategy performances. For simulations, we considered  $NH$  equal to 2. We also varied timer's value to obtain 15 different values from 0.005 to 1.000 second.

Figure 17 shows the percentage of covered nodes for each timer value per posture.

For all postures, the obtained percentages for each timer value are extremely close: the average difference is around 2.5%.

We also noted that with the increase of the timer value, the simulation results are closer to each other.

The first four timer values (the smallest ones) show better percentage for covered nodes. Short timer pushes nodes to broadcast more often the message in the network which increases the probability of receiving the message by the neighboring nodes.

Figure 18 shows the average end-to-end delay for each timer value per posture. It should be noted the performance of MBP strategy. With the increase of the timer value, the MBP strategy spends more time to cover all nodes in the network. The best end-to-end delay is obtained with timer's value equal to 0.005 second.

Figure 19 shows the number of transmissions and receptions for each timer per posture. Contrary to the end-to-end delay parameter, we obtain a descending curve. This is due to the fact that when the timer is longer, nodes spend more time waiting before rebroadcasting the message. Obviously, this has a direct impact on the average end-to-end delay. However, short timers impact the energy consumption (see Figure 19).

## REFERENCES

- [1] N. T. Antonio Manuel Ortiz, Nedal Ababneh and J. Morrison. Adaptive routing for multihop IEEE 802.15.6 wireless body area networks. In *20th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2012, Split, Croatia, September 11-13, 2012*, pages 1–5, 2012.
- [2] W. Badreddine, C. Chaudet, F. Petrucci, and M. Potop-Butucaru. Broadcast strategies in wireless body area networks. pages 83–90, 2015.
- [3] J. K. S. Evan P. C. Jones, Lily Li and P. A. S. Ward. Practical routing in delay-tolerant networks. *IEEE Trans. Mob. Comput.*, 6(8):943–959, 2007.
- [4] A. Goundan, E. Coe, and C. Raghavendra. Efficient broadcasting in delay tolerant networks. In *IEEE Global Telecommunications Conference (GLOBECOM)*, New Orleans, LA, USA, Nov. 2008.
- [5] Z. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)*, New York, USA, June 2002.
- [6] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer. Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks. In *4th IFIP-TC6 international conference on Networking Technologies, Services, and Protocols (Networking)*, Waterloo, Canada, May 2005.
- [7] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *2004 conference on Applications, technologies, architectures, and protocols for computer communications (Sigcomm '04)*, Oct. 2004.

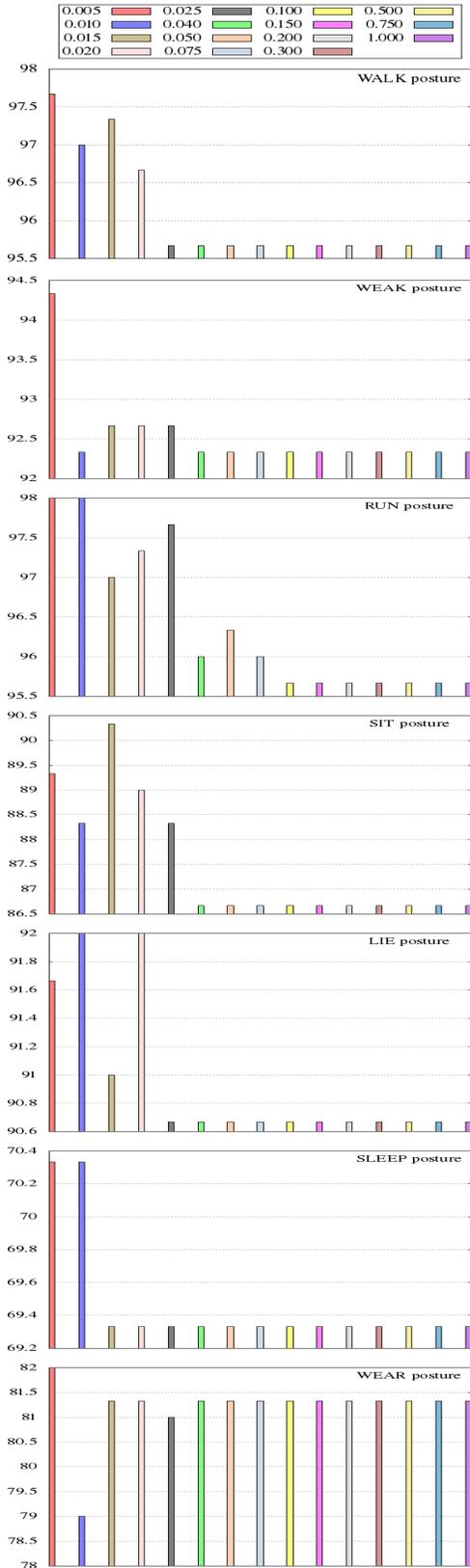


Fig. 17: Percentage of covered nodes per posture for each timer

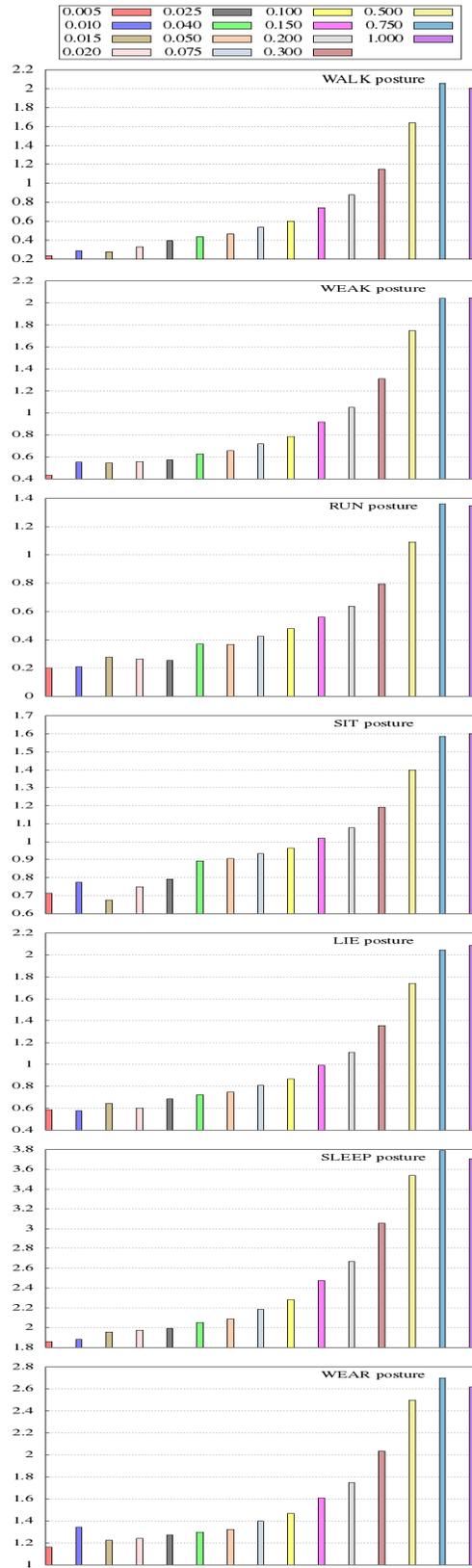


Fig. 18: Average End-To-End Delay per posture for each timer

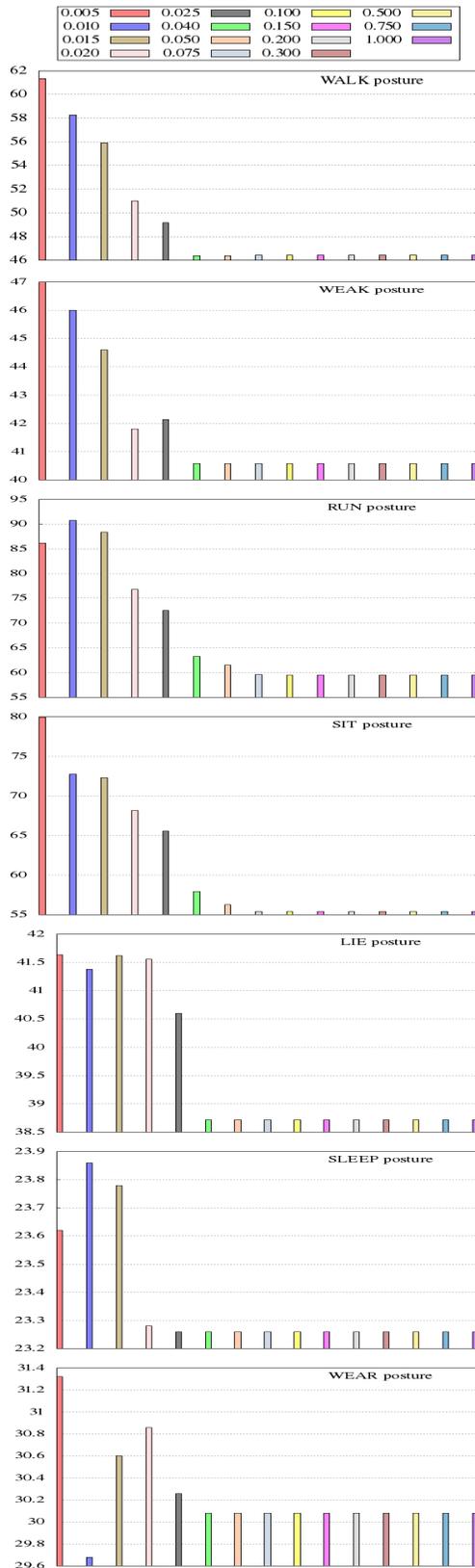


Fig. 19: Number of transmissions and receptions per posture for each timer

- [8] B. K.-P. Koh and P.-Y. Kong. Performance study on zigbee-based wireless personal area networks for real-time health monitoring. volume vol. 28, pages 537–540, Aug 2006.
- [9] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. K. Haneveld, T. Parker, O. Visser, H. S. Lichte, and S. Valentin. Simulating wireless and mobile networks in omnet++ – the mixim vision. In *1st International Workshop on OMNeT++ (hosted by SIMUTools 2008)*, Marseill, France, Mar. 2008.
- [10] S. Merugu, M. H. Ammar, and E. W. Zegura. Routing in space and time in networks with predictable mobility. Technical Report GIT-CC-04-07, Georgia Institute of Technology, 2004.
- [11] S. Movassaghi, M. Abolhasan, and J. Lipman. A review of routing protocols in wireless body area networks. *JNW*, 8(3):559–575, 2013.
- [12] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour. Wireless body area networks: A survey. *IEEE Communications Surveys and Tutorials*, 16(3):1658–1686, 2014.
- [13] Ms.Venkateswari.R and D. Rani.S. Design of an energy efficient and delay tolerant routing protocol for wireless body area network. In *International Journal on Computer Science and Engineering (IJCSSE)*.
- [14] J.-i. Naganawa, K. Wangchuk, M. Kim, T. Aoyagi, and J.-i. Takada. Simulation-based scenario-specific channel modeling for wban cooperative transmission schemes. *IEEE Journal of Biomedical and Health Informatics*, PP(99), May 2014.
- [15] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom)*, Seattle, WA, USA, Aug. 1999.
- [16] M. Quwaider and S. Biswas. On-body packet routing algorithms for body sensor networks. In *First International Conference on Networks and Communications (NETCOM '09)*, Chennai, India, Dec. 2009.
- [17] M. Quwaider and S. Biswas. Probabilistic routing in on-body sensor networks with postural disconnections. In *7th ACM international symposium on Mobility management and wireless access (MobiWAC '09)*, Tenerife, Spain, Oct. 2009.
- [18] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *Wireless Communications and Networking (WCNC)*, New Orleans, LA, USA, Mar. 2003.
- [19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, WDTN '05, pages 252–259, 2005.
- [20] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, Apr. 2000.
- [21] H. Viittala, M. Hamalainen, J. Iinatti, and A. Tapparugsanagorn. Different experimental wban channel models and ieee802.15.6 models: Comparison and effects. In *Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on*, pages 1–5, Nov 2009.
- [22] T. R. Youssef Iraqi and P. Kong. Prevention of collisions among two wireless personal area networks. In *7th IFIP Wireless and Mobile Networking Conference, WMNC 2014, Vilamoura, Portugal, May 20-22, 2014. pages 1-7.*