Performance Analysis and improvement of ZigBee routing protocol

Bilel Nefzi, Ye-Qiong Song

PERFORMANCE ANALYSIS AND IMPROVEMENT OF ZIGBEE ROUTING PROTOCOL

Bilel NEFZI * **Ye-Qiong Song** **

* *LORIA-Nancy University, Villers Les Nancy, France*
** *LORIA-INPL, Villers Les Nancy, France*

Abstract: ZigBee is a recent wireless standard based on IEEE 802.15.4 for Personal Area Networks. Its use in Wireless Sensor Networks arouses many interests. In this paper, a performance analysis and an improvement of ZigBee routing protocol are carried out. ZigBee routing protocol uses a modified AODV by default and Hierarchical Tree Routing as last resort. Firstly, these two algorithms are compared in terms of delay performance and energy consumption. The results showed that Hierarchical Tree Routing provides shorter average end to end delay but performs poorly in terms of energy consumption. So for supporting real time communication, it is desirable to freely choose one or another according to the type of traffic (real-time and non real-time). Secondly, Hierarchical Tree Routing algorithm is slightly modified to provide shorter delays than the original one.

Keywords: ZigBee, IEEE 802.15.4, Hierarchichal Tree Routing protocol, analysis, NS2, end to end delay, energy consumption.

1. INTRODUCTION

ZigBee (Alliance, 2006) is a wireless "standard" of ZigBee alliance based on IEEE 802.15.4 standard (IEEE-TG15.4, 2003) for Personal Area Networks. It defines the network and application layers on the top of physical and data link layers normalized in IEEE 802.15.4. ZigBee stack offers a wireless communication solution coupled with low cost, low energy consumption characteristics. It can be used in consumer electronics, industrial controls, PC peripherals, toys and games, etc. However, one of the potential applications of this standard is in Wireless Sensor Networks (WSN). In fact, IEEE 802.15.4 is designed to achieve a very low power consumption through several optimizations in Physical layer and Medium Access Control (MAC) sub-layer like the use of low duty cycles. The network layer uses a modified AODV (Ad Hoc On Demand Distance Vector) by default and Hierarchical Tree Routing (HTR) as last resort.

Recent research works in WSN have focused on Quality of Service (QoS) support to improve the reliability and performance under severe energy constraints. The improvement of QoS can be tackled in any layer. For instance several research work has been carried out on improving real time support in MAC sub-layer using GTS (Guaranteed Time Slot) mechanism of IEEE 802.15.4 (Francomme *et al.*, 2006). This improves only real time QoS in single hop networks. In network layer, which provides end to end real time QoS in multi hop networks, this is done by adding and improving the QoS support to the routing algorithm. However, before doing that we need to analyze the performance of the existing routing algorithms. It is clear that our aim in long term is to provide real time support in ZigBee Routing Protocol (ZRP).

This paper presents firstly a performance study of ZRP. It consists in analyzing both HTR and AODV and comparing their performance in terms

of end to end delay and energy consumption. Secondly, it describes the enhancements made to the HTR algorithm to provide shorter end to end delays.

AODV performance analysis has been a subject of many articles in the last few years. Das and al. (Das *et al.*, 2000), for example, made a performance comparison between AODV and Dynamic Source Routing (*The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks*, 1998) (DSR). In (Cano and Manzoni, 2000), the authors presented a comparison study of some Mobile Ad Hoc routing protocols in terms of energy consumption. For HTR, (Koubaa *et al.*, 2006*b*) tackled modeling and Worst-Case dimensioning of cluster tree WSNs using network calculus theory (Leboudec and Thiran, 2001). The authors were interested in dimensioning a cluster tree WSN : How to find the delay bounds of traffic in a cluster tree topology giving certain network parameters and a minimum service guarantee (using GTS mechanism defined in IEEE 802.15.4 for example).

The motivation for proposing this analysis is to have a better idea of routing in the ZigBee stack. We believe that this is a good starting point for providing QoS support to ZigBee standard. Actually, the results showed that the HTR algorithm is very interesting for supporting real time communications. So, a first work is made to improve its end to end delay.

The rest of the paper is organized as following. Section 2 gives an overview of ZigBee network layer. Section 3 presents the simulation study and performance evaluation of ZRP. Section 4 presents the new HTR algorithm and simulation results related to it. Section 5 concludes the paper

2. ZRP OVERVIEW

The ZigBee network layer supports star, tree and mesh topology. Since our study is focused on the performance of HTR topology, only this one will be described. In tree networks, a master device called ZigBee coordinator is responsible for starting the network and for choosing certain key network parameters. The network is then extended using ZigBee routers. End devices can join the network through an association to either the ZigBee coordinator or the ZigBee router. The data and control messages forwarding follow a hierarchical routing strategy. Cluster Tree networks may employ beacon-oriented communication as described in the IEEE 802.15.4 specification.

When the tree address allocation is enabled, the network addresses are assigned using a distributed address allocation scheme that is designed to provide every potential parent with a finite sub-block of network addresses to distribute to its children. During network establishment, the ZigBee coordinator determines the maximum number of children per parent (Cm) and the maximum number of ZigBee routers (Rm) between these children. In addition, every node has an attribute called depth which is the minimum number of hopes to reach the ZigBee coordinator using only parent child links. The ZigBee coordinator has a depth of 0 and determines the maximum depth of the network (Lm). A function called $Cskip(d)$ (equation 1) is used after that to calculate the size of the address sub-bloc being distributed by each parent located at depth d.

$$Cskip(d) = \begin{cases} 1 + Cm \cdot (Lm - d - 1), \text{ if Rm=1} \\ \dfrac{1 + Cm - Rm - Cm \cdot Rm^{Lm-d-1}}{1 - Rm} \\ \text{otherwise} \end{cases}, \tag{1}$$

Network address distribution is as follows. The coordinator has always the address 0. For router-capable child devices, the address assignment uses the value of $Cskip(d)$ as an offset : if the node is the first served, its address is 1 greater than its parent address. Otherwise, the addresses are separated from each other by $Cskip(d)$. For simple end devices, network addresses are assigned in a sequential manner using the following rule :

$$A_n = A_{parent} + Cskip(d) \cdot Rm + n \tag{2}$$

Here $1 \leq n \leq (Cm - Rm)$ and A_{parent} represents the address of the parent.

Routing rules.

If The node has a routing table and there is a routing table entry for the destination
then Use it
else if There is a room for another entry
then Try a route discovery
else Route along the tree using HTR

The route discovery uses a modified version of AODV. The only difference is in the cost of a link. The link cost $C\{l\}$ for a link l is a function with values in the interval $[0 \dots 7]$ defined as :

$$C\{l\} = min(7, round(\frac{1}{p_l^4})) \tag{3}$$

Here p_l denotes the probability of packet delivery on the link l. We shall note that the standard permits to report a constant value of 7 for link cost (thus we come back to the standard version of AODV).

AODV is a reactive routing protocol; the network is silent until a connection is needed. At that point the network node that needs a connection broadcasts a request to its neighbours who forward this message to theirs and record the node that

they heard it from. When a node receives such a message and already has a route to the desired node, it sends a message backwards through the reverse route to the requesting node. The needy node then begins using the route that has the least number of hops through other nodes.

The hierarchical routing is based on some comparison rules: for a ZigBee Router with address A at depth d, if the following logical expression is true, then a destination device with address D is a descendant.

$$A < D < A + Cskip(d-1) \tag{4}$$

If it is the case, the Next Hop is given by the following rule:

- $N = D$ for ZigBee end devices where $D > A + Rm \cdot Cskip(d)$
- $N = A + 1 + \lfloor \frac{D-(A+1)}{Cskip(d)} \rfloor \times Cskip(d)$, otherwise

(5)

If the expression is not true, the destination is not a descendant and the message should be routed through A's parent.

Example. This example (figure 1) illustrates the whole routing mechanism. In figure 1, numbers denote node addresses and numbers put in square brackets represent the parent address of a node. We suppose that the maximum number of entries in the routing table is equal to 3. Here, node 4 needs to send data to node 10 and so looks at its routing table (figure 2). Three cases can occur. In the first one, it finds an entry for node 10 (see figure 2(a)). So the Next Hop is known and transmission can start. In the second case, it doesn't find an entry for node 10 but there is a free entry (see figure 2(b)). So node 4 starts a route discovery using AODV protocol. In the last case, it doesn't find an entry for node 10 and the routing table is full. In this case, it uses HTR algorithm for data transfer. The route is constructed as follows: node 4 sends data to its parent (node 1) since node 10 is not a descendant (because node 4 knows the sub-block of network addresses of all its descendants). Node 1 sends data to its parent too (node 0) for the same reason. Node 10 is a descendant of node 0, so this one uses the Next Hop formula (equation 5) to find the successor which is in our case node 2. Node 2 determines its successor using the same technique which is node 5. Finally, node 5 transmits data to node 10 directly since it is its child.

3. PERFORMANCE STUDY

In this section, we describe firstly the simulation setup. Then we report the analysis of simulation results.

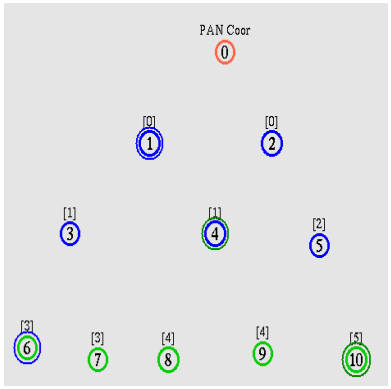


Fig. 1. ZBR example

Destination	Next Hop
0	2
3	1
10	5

(a)

Destination	Next Hop
0	2
3	1

(b)

Destination	Next Hop
0	2
3	1
6	1

(c)

Fig. 2. Example of some routing table states of node 4

3.1 Simulation description and scenarios

We have developed the ZigBee HTR algorithm under NS-2 simulator (ns, n.d.) on the top of the existing IEEE 802.15.4 implementation. The model is then used to study, mainly, its delay and power consumption behaviors. A comparison with the AODV routing algorithm is also made. We consider a WSN in a surface of 80 m x 80 m with one PAN coordinator and 84 ZigBee router nodes (figure 3). Since simple end devices do not participate in the routing procedure, we don't make use of them in simulation. We consider a fully connected network where every node hears just its immediate neighbors (one hop neighbors). It has a maximum depth (Lm) of 6, a maximum number of children (Cm) per parent of 5 and a maximum number of routers among children (Rm) of 5. The network uses beacon enabled mode to ensure a global synchronization. For this the ZigBee coordinator and some ZigBee routers emit periodically Beacon frames. In fact, the simulation model is configured to avoid all possible Beacon collisions. The Beacon scheduling mechanism defined by ZigBee is not implemented in the simulator. The *Beacon Order* (BO) and the *Superframe Order* (SO) are equal to 3. The MAC sublayer uses slotted version of CSMA/CA as channel access protocol. The traffic generated by source nodes is Poisson distributed with a mean inter-

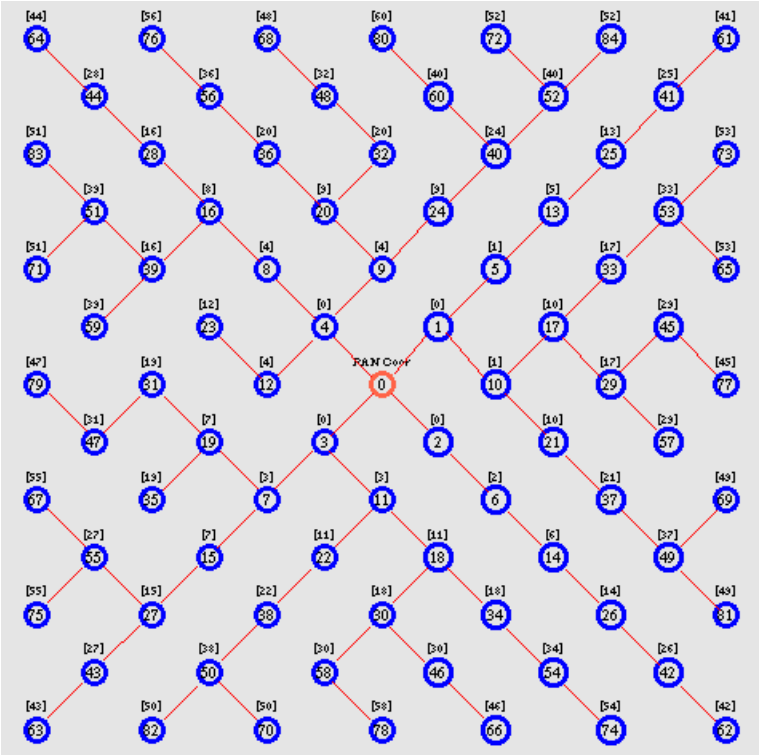


Fig. 3. Nodes deployment

arrival rate equal to 0.2s (or a throughput equal to 3.08 Kbit/s) which represents the normal traffic load for a sensor node (Koubaa *et al.*, 2006*a*). The case of a heavy load will be studied in 3.2.4. The data frame size at the MAC level is equal to 77 bytes. The data transmission rate provided by IEEE 802.15.4 is equal to 250kbit/s. The data transfer session is 500 seconds long. The case of a shorter one will be discussed in 3.2.3.

Five scenarios are considered. The first one considers data transmission between all nodes to the PAN Coordinator. In the second one, node 61 (upper right) is chosen as a sink. The third one considers data transmission between two random nodes. The forth one analyzes the case of heavy load. In the last scenario, the worst case of HTR and AODV is studied and a comparison between them is made.

Finally, the energy consumption is analyzed to show the cost of each algorithm with regard to energy.

3.2 Simulation results

In what follows we give simulation results of the five scenarios and energy consumption analysis.

3.2.1. Case that the sink is the PAN coordinator

Figure 4 shows average delay of AODV and HTR calculated as a function of nodes depth.

For nodes localised in the first depth, the average delay is the same, which is expected because the Route Discovery delay in AODV is very short. We shall remember here that HTR doesn't send any type of requests. For other depths, HTR Algorithm is more powerful than AODV, mainly because HTR route is usually the best route to the PAN Coordinator and AODV must discover it before sending data.

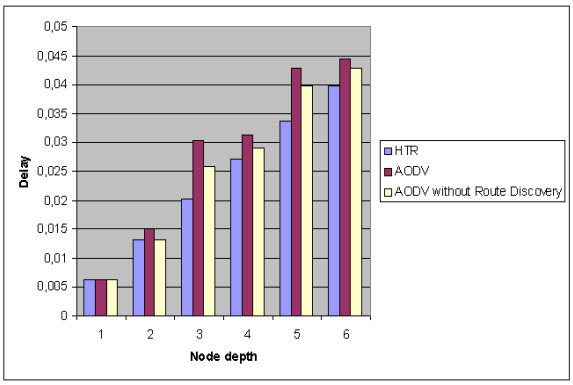


Fig. 4. Average delays as a function of nodes depth

To compare the effect of routes taken by both algorithms, we have eliminated the delay caused by route discovery process in AODV. This is also illustrated in Figure 4. In the first two depths, AODV and HTR obtain the same results which means that they choose the same path to node 0. For the other depths, HTR algorithm performs better than AODV algorithm. We have observed that both algorithms don't choose the same path

to destination and even worse AODV doesn't choose always the best route. The analysis of trace files showed that this is caused by some extra delays in Route Request messages transmission. In fact, whenever a node receives a route request from two other nodes, it broadcasts only the first arrived. In some cases, the route request that follows the best route arrives later and then, it will not be chosen. The extra delays are usually caused by beacon transmission. Since beacons have the highest priority, routers that emit them queue route request messages before transmitting them which adds an extra delay.

3.2.2. Case that the sink is the node 61 Table 1 shows the average delay of both HTR and AODV algorithms when the sink is at node 61 (upper right). AODV's average delay is slightly higher than HTR's one. This result shows that even if the sink isn't located in the center of the network, HTR performs better in comparison with AODV. How ever, the low variance value of HTR compared to AODV's one means that some nodes don't follow the best route to node 61, specially those who are around it because normally they lead to a shorter end to end delay in comparison with far nodes. So, the high AODV's end to end delays are likely to be caused by Route Discovery process. In the next scenario, we will try to confirm this conclusion.

The end to end delay results led us to ask the following question: if HTR performs better, why we use it in the last order (ref. section 2)? In the next simulations, we will try to find the answer.

Table 1. Case of the sink is node 61 : statistical results

	HTR	AODV
Average Delay (ms)	66	67
Variance	15	27

3.2.3. Case of random source and destination In this series of simulations, two nodes are chosen randomly and traffic goes between them. 100 simulations are done; the results are illustrated in Table 2. Again, in average, HTR outperforms AODV in terms of transfer delay. In addition, the shorter the session time, the higher the AODV's delay. The results of HTR are invariant with respect to session time. However, the delay variance is higher in AODV routing algorithm, which means that the dispersion around the average is higher too. These results are somehow unexpected, because AODV chooses in most of times the best route from a given source and destination while HTR algorithm always send data over the tree. So, again, the delay caused by route discovery is eliminated to show the impact of this process.

Table 2. Case of randomly chosen nodes : statistical results

	Average Delay (ms)	Variance
HTR	49	17
AODV with 10s tr. session time	102	290
AODV with 500s tr. session time	56	37
AODV without R. Disc. delay	47	24

The results (table 2) show that the average delay is better for AODV. So we can conclude that the route discovery in AODV is a real handicap even for a long data transfer session (500 seconds in the simulation).

3.2.4. Case of heavy load In this series of simulations, we have re-run the first 3 scenarios but with a mean inter-arrival rate equal to $0.05s$ which gives a data throughput of $12.32\ Kbit/s$. Tables 3, 4 and 5 show the simulation results when the sink is in the PAN coordinator, the node 61 and for a random choice of nodes.

We notice that for heavy traffic load, AODV's end to end delay become very high whereas HTR's end to end delay remained almost the same (little bit higher than normal load). This huge difference when AODV protocol is used is caused by collisions in Route Discovery messages. In fact, the higher the load, the higher the collision probability. So, collisions in Route Discovery messages lead to an extra discovery delay. HTR protocol, how ever, does not need to send route discovery messages. So the impact of collisions will be negligible.

Table 3. Case of a heavy load and the sink is node 0 : statistical results

	Average Delay (ms)	Variance
HTR normal *ld*	28	10
AODV normal *ld*	35	14
HTR heavy *ld*	33	12
AODV heavy *ld*	143	460

Table 4. Case of a heavy load and the sink is node 61 : statistical results

	Average Delay (ms)	Variance
HTR	72	19
AODV	91	14

Table 5. Case of a heavy load and random choice of nodes : statistical results

	Average Delay (ms)	Variance
HTR	57	20
AODV	113	305

3.2.5. Worst case In this scenario, an analysis of the worst case of both algorithms is made. The HTR worst case is when two nodes are close to each other and the traffic transmitted along the tree goes too far away before returning. An example is illustrated by the figure 5(a); data goes from node 64 to node 76. The AODV worst case is when source and destination nodes are too far away from each other which lead to a long route discovery delay. The figure 5(b) illustrates such a case; data goes from node 62 to node 64. Table 6 illustrates the obtained results. For the HTR's worst case, AODV performs better since the nodes are close to each other and route discovery process doesn't take too much time. This result could justify the use of AODV in the first place in ZigBee routing protocol. For AODV's worst case, and as expected, HTR performs better even if we know that the route along the tree could not be the best route. On the one hand Route Discovery mechanism costs high. On the other hand, the route discovered may not be the best route to destination which is the case in our example.

Table 6. Worst case results (in seconds)

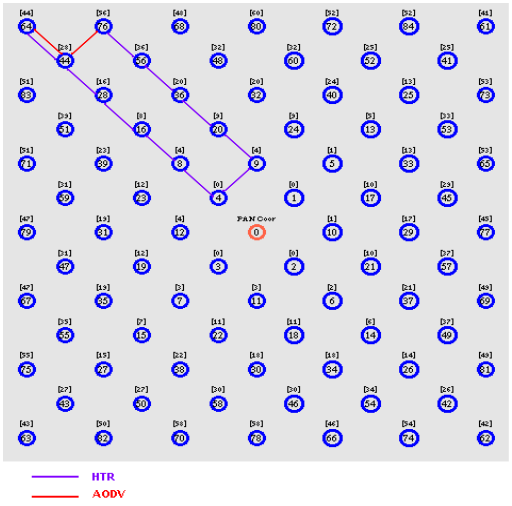
	HTR	AODV
HTR's worst case	72	15
AODV's worst case	85	95

3.2.6. Energy consumption analysis Here we consider nodes with limited energy power. We used NS2 energy model. $initialEnergy$, $rxPower$ and $txPower$ are respectively set to 1, 0.3 and 0.3. We choose two nodes randomly and traffic goes between these nodes. Two observations come out of the simulations.

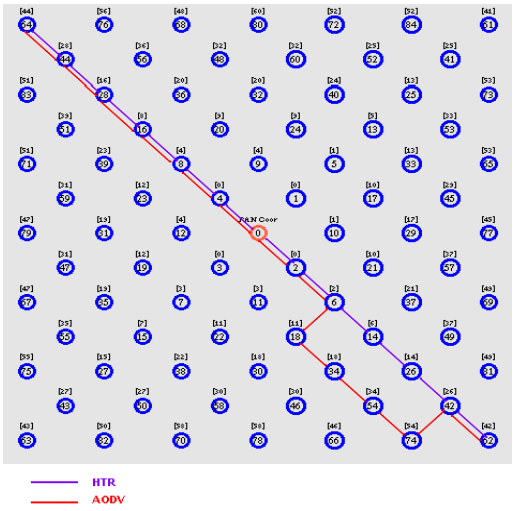
First, for HTR algorithm, the traffic stops when the first node along the tree dies. the main reason for this is because HTR is a static routing algorithm. This phenomenon is catastrophic for HTR because if a node dies, all its child can't transmit data anymore. AODV, however, continue to operate whenever at least one path exists between the sender and the receiver.

Second, nodes tend to die faster in HTR. Table 7 shows death times of the first node for both algorithms. Here traffic goes between node 67 and 62. The analysis of trace files showed that HTR protocol has a bigger throughput which gives a higher packet transmitted number and so a higher energy consumption.

The main difference between both protocols is that the energy consumption with AODV is more uniformly distributed over the network because HTR uses always the same parent child links whereas AODV discover the route before sending data.



(a)



(b)

Fig. 5. HTR's (a) and AODV's (b) worst case

Table 7. Death time of first node (in seconds)

	HTR	AODV
Death time (s)	55,300576	60,193088

Considering that energy is a critical parameter in WSN, the use of AODV by default is a justified choice despite the good performance of HTR algorithm.

4. IMPROVEMENT OF THE HTR ALGORITHM

In what follows we give the modified algorithm in the first paragraph, an illustration example in the second one and the results of the simulations in the last paragraph.

4.1 Description of the modified algorithm

The idea of the improvement is to use the neighbors table of a node in routing decisions. Using the neighbor's Tree address, a node is able to check if a destination is one of its neighbor's descendant. If it is the case, data is forwarded to that neighbor thus leading to shorter paths.

Let us consider a packet that has D as destination and arrives to node A that has d as depth. Let $V(x)$ denotes the neighbor list of node x and $d(x)$ the depth of node x. The modified HTR algorithm (M-HTR) is as follows :

If D is a descendant of node A
then Use rule given by equation 5 to find the Next Hop

else For all N in $V(A)$
if D is a descendant of N (using rule given by equation 6)
then $Next\text{-}Hop = N$.

else (not a descendant, not a neighbor and not a descendant of any of A's neighbors) send the packet to the father.

If there are multiple neighbors that fit the rule (D is a descendant of more than one neighbor), choose the one with the highest depth.

$$N < D < N + Cskip(d(N) - 1) \qquad (6)$$

4.2 Illustration example

Let us consider the network illustrated in figure 6 where node 4 sends data to node 10 (the same network example used in section 2).

For the basic HTR algorithm, the path is 4→ 1→ 0→ 2→ 5→ 10 (see the example in section 2 for route construction). So, the route (red line in figure 6) is 5 hops long.
For the M-HTR algorithm, the path is constructed as follows: Node 4 finds that node 10 is not a descendant. So it tests if it is a descendant of one of its neighbors (nodes 1, 2 and 5). Indeed, it finds that it is a descendant of nodes 2 and 5. Since the latter has the highest depth, it is chosen and data is sent to it. Node 5 just sends data to node 10 since it is its child. So, the path (purple line in figure 6) is 2 hops long.

4.3 Simulation results

We have developed the M-HTR under NS-2 simulator. We used the same network and parameters as in section 3. 100 simulations are ran and in each one two random nodes are chosen and traffic goes between these nodes. Average results of the comparison between the basic HTR and M-HTR are shown in table 8.
As it is expected, M-HTR performs better in terms of end to end delay and the variance is little bit higher. We found that for 100 simulations, 21 paths are improved and the rest remains the same. This makes a 21% enhancement score. The average delay of these 21 simulations is given in table 9. The high delays obtained in these 21 simulations (larger than the average) for the Basic HTR algorithms show that most of the improved paths are Worst Cases. M-HTR succeeds to eliminate these worst cases and to achieve a low end to end delay. Moreover, M-HTR improves the basic algorithm but never degrade it : the number of hops for M-HTR is in the worst case equal to the Basic algorithm.

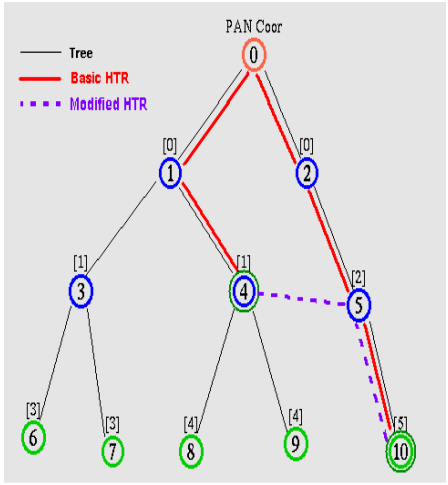


Fig. 6. Illustration example of the basic and the modified versions of HTR

Table 8. Basic and Modified HTR algorithms.Case of randomly chosen nodes : statistical results

	Basic HTR	M-HTR
Average Delay (s)	49	45
Variance	17	18

Table 9. Basic and Modified HTR algorithms. Improved paths : statistical results

	Basic HTR	M-HTR
Average Delay (s)	55	36

5. CONCLUSION AND FUTURE WORK

We have analyzed the delay and energy consumption performance of ZBR. It has been shown that HTR provides shorter average end to end delay but performs poorly in terms of energy consumption in the way that it is not well distributed over

the whole network. Considering that energy is a critical parameter in WSN, the use of AODV by default is a justified choice. The simulation is also ran with 40 nodes (not reported in this paper) and the results remain the same.

However, the good delay performance of HTR led us to think about improving it to support real time applications. In fact, the worst case and energy consumption analysis showed that HTR has a great potential of improvements. So, a first work is presented in this paper to ameliorate it. M-HTR reduced end to end delay and succeeded to remove worst cases.

We think that for supporting real time communication, it is highly desirable to freely choose a combination of both algorithms according to the type of traffic (real-time and non real-time). So, further work must be done to ameliorate HTR protocol. We can easily make it more dynamic to avoid high energy consumption and to add real time characteristics.
Besides, GTS channel access mechanism is not used in our simulations. In our future work, we will try to obtain a more general analysis of ZRP using all access mechanisms implemented in IEEE 802.15.4.

REFERENCES

Alliance, ZigBee (2006). Zigbee document 053474r13.

Cano, Juan Carlos and Pietro Manzoni (2000). A performance comparison of energy consumption for mobile ad hoc network routing protocols. *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000. Proceedings. 8th International Symposium on* pp. 57–64.

Das, Samir R, Charles E Perkins and Elizabeth M Royer (2000). Performance comparison of two on-demand routing protocols for adhoc networks. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* **1**, 3–12.

Francomme, J., G. Mercier and T. Val (2006). A simple method for guaranteed deadline of periodic messages in 802.15.4 cluster cells for automation control applications. *11th IEEE International Conference on Emerging Technologies and Factory Automation*.

IEEE-TG15.4 (2003). Part 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for low-rate Wireless Personal Area Networks (LR-WPANs). *IEEE standard for information technology*.

Koubaa, Anis, Mario Alves and Eduardo Tovar (2006*a*). i-game: An implicit gts allocation mechanism in ieee 802.15.4 for time-sensitive wireless sensor networks. In: *ECRTS '06: Proceedings of the 18th Euromicro Conference on Real-Time Systems*. IEEE Computer Society. Washington, DC, USA. pp. 183–192.

Koubaa, Anis, Mario Alves and Eduardo Tovar (2006*b*). Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. *27th IEEE International Real-Time Systems Symposium, 2006. RTSS '06.*

Leboudec, J.-Y and P. Thiran (2001). A theory of deterministic queuing systems for the internet. *Lecture Notes in Computer Science(LNCS)*.

ns (n.d.). The Network Simulator NS-2. http://www.isi.edu/nsnam/ns/.

The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (1998). *http://www3.ietf.org/proceedings/98dec/I-D/draft-ietf-manet-dsr-00.txt*.