Real-time routing and retry strategies for low-latency 802.15.4 control networks

Koen Holtman
Philips Research
High Tech Campus 34
5656AE Eindhoven, the Netherlands
+31 40 27 91461
Koen.Holtman@philips.com

Peter van der Stok
Philips Research
High Tech Campus 34
5656AE Eindhoven, the Netherlands
+31 40 27 49657
Peter.van.der.Stok@philips.com

ABSTRACT

In many applications of wireless control networks, the latency of message delivery is an important consideration. In a lighting control network where a light switch sends a wireless message to a lamp, a worst case end-to-end latency of 200 ms or better is desired, so that the working of the switch feels 'immediate' to the end user. This paper studies the probability that latency deadlines of a few hundred ms are exceeded. We use a 802.15.4 test network, located in a real-life office environment, to evaluate and compare the effects of several re-try and re-routing strategies and different MAC parameter settings. Testing under realistic conditions, in an office environment when people are present, is important to accurately determine worst case latency performance as experienced by end users. At night, without any people in the building, performance is much better than during the day. In order to accurately observe the effect of different strategies, test runs lasting at least a week are needed. We find that retrying message delivery via a single delivery route is sub-optimal. Keeping a set of two or more candidate routes for subsequent re-tries greatly improves worst-case latency. We show that the use of time slotting and energy saving strategies is not necessarily incompatible with the goal of optimizing for human-observable latency.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design]:** Wireless communication, C.2.2 **[Network Protocols]**: Routing protocols

General Terms

Measurement, Performance, Reliability, Experimentation, Human Factors.

Keywords

Control networks, wireless sensor networks, 802.15.4, latency, retry strategies, routing strategies, energy saving.

1. INTRODUCTION

A wireless control network differs from the more commonly considered wireless sensor network (WSN) because it incorporates actuators in addition to sensors. Whereas in most WSN designs, the object is to get all sensing data forwarded to a central (logging) location for later analysis, in a wireless control network the object is to control actuators based on data from close-by sensors. Home and office control systems are important applications of wireless control networks. The low cost of wireless sensor nodes, and their ability to run on batteries, (or even on energy harvested from the environment), makes it feasible to equip rooms with many sensors, which can be used to increase both comfort and energy efficiency.

The reliability and latency of a wireless building control system must be just as good as that of the wired system that it replaces. Consider the case where a lamp in a room is wirelessly connected to a switch on the wall. In such a setup, it is not the average end-to-end latency that will be important for the quality perception of the user, but the worst-case latency. If the latency is too high too often, the system will break the ‘immediacy’ mental model that the user has for light switches, which negatively impacts the user experience and ultimately the user acceptance. Our design goal is to minimize the probability that 200 ms latency is exceeded. As a rule of thumb in user interface design, 200 ms latency is still short enough that the user can accept the relation between stimulus and response as ‘immediate’.

Contrary to most publications we are less concerned with the maximum achievable throughput, average-case latency, or scalability of routing algorithms. We expect that most communications in building control will not need more than 2 hops, with 4 hops being exceptional. We therefore focus on quality improvement of the 1-hop and 2-hop cases, which represent the bulk of the wireless control communication in the building.

We concentrate on the effects of multipath fading which lead to unexpected link failures of very good links during the day time. Effects coming from interference caused by the high density of nodes are not investigated. Therefore, the measurements concentrate on a relatively low number of nodes reflecting the node and message density we expect for building control

2. MEASUREMENTS IN LITERATURE

This paper is motivated by measured communication behaviour in buildings with occupants. In the literature measurements are described related to: Point to point communication, Sharing the medium between multiple senders, and Routing over a large dynamic network. This section summarizes the main published results as introduction to our routing suggestions to improve the probability that packets arrive in time (within their deadline).

A naïve model of transmission by an omni-directional antenna in empty space, states that the strength of the signal decreases with the square of the distance [9]. Unfortunately, space is not empty but is populated with reflectors and absorbers of the RF signal. Reflections from surfaces can meet the original signal with a slightly different phase thus reducing or completely removing the signal. Report [9] shows that most simple propagation models used in simulations do not correctly represent the transmission as measured in situ. Measurements of the communication quality between a single sender and a receiver indicate that there are usually three regions: (1) a clear region with little or no reception losses, (2) a transitional region where packet loss ranges from a few percent to complete loss unrelated to the transmission distance, and (3) a region where almost no packets are received.

Over time the link quality fluctuates as well. These phenomena are reported in [1][2]. The clear region for IEEE 802.15.4 [3] is reported to be on average between 3 and 15 meters in [5][6][8], where in some cases the signal had to penetrate office walls. In [7] it is observed that the height of the sender has a significant impact on packet yield. Papers [1], [4], [5] and [7] show that RSSI is not a good indicator for success rate. Link quality Indicator (LQI) is seen to perform better in [7]. From the papers we learn that the transmission range depends on the direction, that channels are not necessarily symmetric, and that channel strengths change over time, and with the height of node. Assuming that the point to point transmission is in the clear region, dependent on the load and the number of load generating senders, the medium throughput around a given node still suffers from multi-sender interference. In [4], confirmed by [6] it is shown that the total throughput obtained with one sender is halved when four senders try to occupy the channel. The lowered throughput can come from the many retries, the larger back-off times associated with retries, or the loss of packets.

The behaviour of the links over time is discussed in [10]. Conclusions are that the number of packets needed for successful transmission is a better quality indication than Reception Rate (RR). Another conclusion is that at a given time a stable link is the best link to use and that failure over a stable link is accompanied by failures over the unreliable links as well. In [2] and [12] it is shown that link quality and not hop count should be at the basis of path selection. In [12] also the unpredictable stability over the day is shown. In contrast to earlier papers, the measurements in [13] suggest that RSSI is a good indicator of link quality. The authors conclude that the new chip technology of the CC2420 improved the usability of the RSSI value. In [14] it is shown that irregularity of the radio signal has a high impact on the routing efficiency. In [11] and [13] an overview of wireless communication link-failure over time is presented.

The notion of real-time deadline and delay is not frequently cited in the context of routing. The authors of [15] show that no guarantees on end-to-end delays can be given but that we are confronted with an end-to-end delay probability distribution. This notion has led to the development of the SPEED protocol where the probability of meeting a deadline was recalculated during the progress of the packet over the multi-hop path [16]. The SWR protocol [17] uses multiple paths routing while removing packets which will not meet their deadline. Another probabilistic real-time routing protocol is presented in [18], where the forwarding time is estimated for each hop during transmission.

In this paper we investigate link performance over at least a week and try different routing and packet retry schemes to improve the transmission success rate within the deadline. This paper extends existing work by:

- Observations over periods of at least a week,
- Testing links that are in the clear region,
- Observation in an office building during working hours,
- Concentrating on one-hop and two-hop routes.

3. TEST NETWORK

Building on our earlier work [19][21], we constructed a test network with 8 network nodes in an office building, and did extensive measurements on latency. Walls are made of plasterboard, but the offices contain many large metal filing cabinets, creating a strong multi-path environment. Node locations for the tests are shown in figure 1. Each node 1-7 is a sensor node, sending messages with (arbitrary) sensing data to node 0, at random times with an average rate of 2.7 messages per second per sensor node. This high message rate is unrealistic for a control network in real-life situations, but it does ensure that we can gather sufficient statistics in a test run lasting about a week. As shown in figure 1, the nodes 0-4 are all in the same office. We located them behind obstacles in this office in such a way that there is no direct line of sight between them – this means that multipath cancellation (Rician or Rayleigh fading) can cause link failure even inside this office.

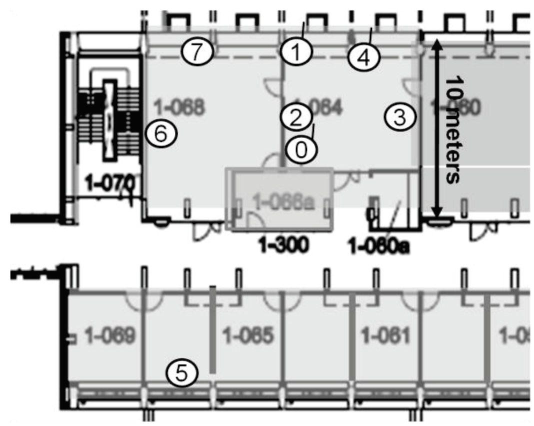

Figure 1. Sensor nodes 1-7 send messages to node 0.

The nodes use the 802.15.4 protocol in the 2.4 Ghz band, and are implemented using Jennic JN5139-Z01-M00/M001 wireless modules [20] with custom software implementing the delivery algorithm in figure 2. Nodes 0,1, and 4 have whip antennas, the other nodes have small ceramic antennas. All nodes can communicate directly with node 0, although the link quality between node 5 and node 0 proved to be lower than between the other nodes and node 0.

The messages are fixed-length and very small. Including protocol overheads, an 802.15.4 packet carrying a message has a length of 27 bytes. We use the 802.15.4 MAC uni-cast mechanism with CSMA/CA and acknowledgments, with each node doing up to SMRT ('single MAC invocation retries') packet sending re-tries whenever it finds a clear channel. The parameter SMRT is set to 4 (the 802.15.4 MAC default) in most of our measurements. In an outer loop, the MAC is invoked every WT ('wait time') milliseconds until an end-to-end acknowledge message (sent by node 0 via the reverse route) is received. Reception time of the end-to-end ack determines the latency of the message. The parameter WT is set to 40 ms in most of the measurements.

Routes can contain multiple hops. If a node receives a message that it should forward, according to the routing instructions in the message, the node will invoke IEEE_802.15.4_MAC_unicast_with_ack() once to do the forwarding. If this fails, the routing node will discard the message. The original node will do a retry, possibly via another router, when its timer t2 runs out.

```
deliver_message(m) {
    start timer t1 counting up from 0 ms;
  do {
    pick a delivery route r;
    format message m with route r into packet p;
    start timer t2 counting down from WT ms;
    IEEE_802.15.4_MAC_unicast_with_ack(p);
    wait_until( (t2 < 0) ||
        (end-to-end acknowledgement received from MAC) );
  } while(no end-to-end acknowledgement received);
   measured_message_latency = t1;}

IEEE_802.15.4_MAC_unicast_with_ack(p) {
  Try to detect a clear channel, with exponential backoff;
   // we use MaxCSMABackoffs=4 and minBE=1 so this
   // detection phase takes between 1 and 19 ms.
  if(no clear channel found) return;
  for(i=0; i<SMRT; i++) {
   use radio to send packet p; //takes ~2 ms
   use radio to listen for MAC acknowledgement
          packet; // takes ~1 ms
   if(valid MAC acknowledgement packet heard) break;
 } }
```

Figure 2. Message delivery algorithm used by nodes.

Several variants of the algorithm in figure 2 are possible. For example, if the MAC fails to deliver a message with acknowledgment, the outer loop might retry immediately, rather than waiting for t2 to run out. We have not (yet) tested this variant – it creates a larger risk of packet storms in the system, but the speedup might be worth-while.

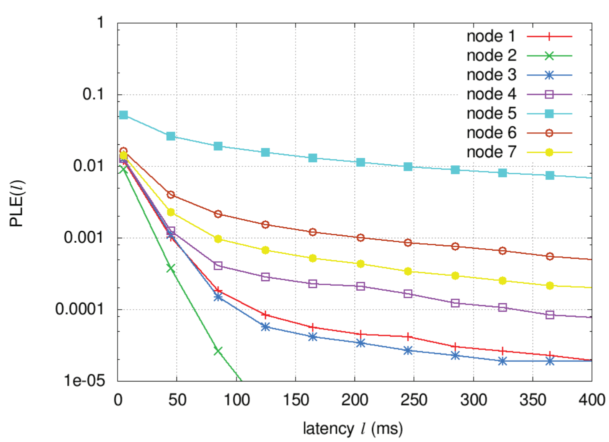

Figure 3. PLE(l) for all nodes with 1-hop route to node 0.

4. SINGLE HOP LATENCY

We consider the latency of routing along a single hop: we do a test where all sensor nodes always choose the direct route to node 0 to deliver their message with WT=40 ms and SMRT=4. In order to get sufficient statistics about the infrequent high latencies, we run the test network for many days. This creates test logs containing millions of message latency measurements per node. To interpret the test run data, with a focus on the question of how the end user will experience the latency, we found it instructive to define the metric PLE(l): the probability that a latency l is exceeded by a message. We compute it as follows from a test run:

<u>PLE(l) for node n</u> =

(number of messages sent during office hours by node n with measured_message_latency >=l) / (total number of messages sent during office hours by node n).

Because of our interest in human-observable latency, the PLE calculation does not consider messages sent outside of office hours – this is discussed in more detail in [19]. Figure 3 plots PLE(l) for all sensor nodes, showing large differences for the nodes. Node 2, closest to node 0, has the best (lowest) curve. Node 5, furthest from node 0, has the worst. For the other nodes, there is no strong correlation between curve position and the distance of the node n to node 0.

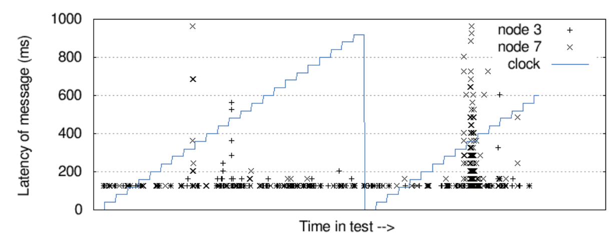

Figure 4. Messages with latency higher than 125 ms.

Figure 4 shows how high latencies occur over time, for nodes 3 and 7. The X axis is time over a 38 hour section of the run taken for figure 3, the saw-tooth line plots a 24-hour clock. The high latencies occur almost exclusively during office hours. Node 7 experienced latencies higher than 1000 ms off the Y axis scale. We see that a node can have 'good days' where a given latency deadline is never, or almost never, exceeded, and 'bad days' where it is exceeded often within a few hours. Test runs have to be long because every node needs to have a chance to experience its 'bad days' at the rate that they happen for that node. The unpredictability of the number of 'bad days' in a week, and the possibility of long term changes in the environment, makes us cautious in comparing curves from different test runs. To measure the differences between alternative message delivery strategies, we use a single long test run in which the nodes switch to a random different strategy every 10 minutes. Our testing approach extends earlier work, where conclusions were drawn on the basis of shorter run durations under cleaner conditions.

5. MAC INVOCATION INTERVAL WT

With WT=40 ms, a failing link makes the node invoke the MAC, every 40 ms, sending at most SMRT=4 packets. So if a link is failing, WT=40 ms generally leads to a PHY packet sending rate of 100 packets per second, until success. This is an excessively high rate which may lead to packet storms. In [21], we used a 5-node test-bed (with SMRT=4) to investigate the effect of varying WT. Some indicative results from [21] are shown in figure 5. For node 4, far away from node 0, we see an improvement when lowering WT. This is conform expectation, because the time between retries is smaller and consequently the probability of meeting the deadline for a given number of retries is higher.

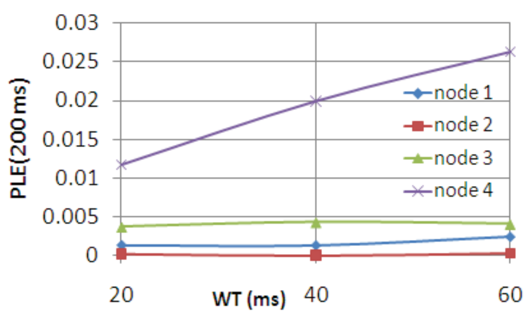

Figure 5. Effect of WT on PLE(200ms) for 4 nodes on different test bed, with 1-hop routing to node 0.

However, for the other nodes, closer to node 0, we see no difference that is statistically significant for a one-week test run. Apparently, with values of 66, 100, and 200 PHY packets per second, we are well past the point of increasing returns for these short hops. This can be explained as follows. First, if a link suffers from significant path fading for a duration much larger than the packet size, sending packets faster will not help. Second, the positive effect of sending more packets might be counterbalanced by the negative effects of longer medium occupation. Third, while the MAC is waiting for a clear channel, its 802.15.4 radio will not be able to receive an end-to-end acknowledge message directed to it. Instead, the MAC will interpret the attempt to deliver the message as a busy channel, causing it to wait. Finding out which of these effects, if any, is dominant needs more study.

In a similar test run on the 8-node test-bed of figure 1, we again saw no significant difference, for PLE(l) with l>100 ms, between WT=20 and WT=40. For all test results in sections 6-9 we therefore use WT=40 ms and SMRT=4.

6. IMPROVEMENT BY ADDING MORE CANDIDATE ROUTES

Except for nodes 5 and 6, the curves in figure 3 are satisfactory, from the standpoint of human-observable latency. Nevertheless, an optimization that makes the curves go down more steeply, increases the quality of the user experience. As we discussed more extensively in [19], a lot of the high latencies in figure 3 can be attributed to multipath fading of the single link. Performance can be improved if the sensor node maintains a list of multiple candidate routes, and re-tries along other routes if attempts along the preferred route fail. Figure 6 shows the performance of the test network with an improved routing scheme. Each node first tries the direct route to node 0 twice. If that does not work, routing via node 1 is tried, and if that does not work routing via node 2. If that still fails, the direct route is tried twice again, and so on. Values of PLE(200 ms) are greatly improved (reduced) except for node 2, which goes from extremely good to merely good.

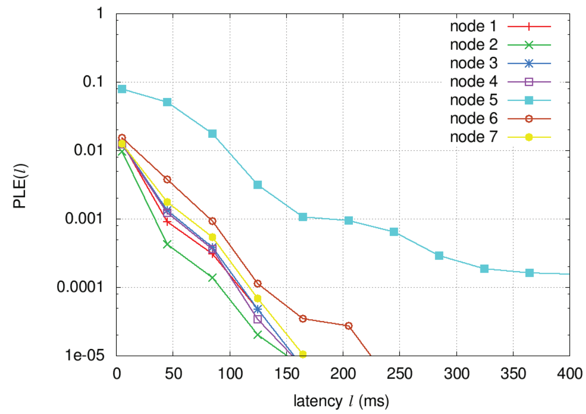

Figure 6. PLE(l) for all nodes using 3 candidate routes.

7. TWO-HOP LATENCY

We now turn to the study of two-hop routes. We measure the PLE for message delivery along each of the all possible 1-hop and 2-hop routes to node 0. In this test run we decrease the message sending rate in the system to 1.4 messages per node per second, to avoid a situation where system self-interference becomes a dominant factor in determining PLE. Each route is tested individually: the node keeps trying the route under test until the message is delivered. Over the course of 2 working days, each route was tested by sending 12500 messages over it on average. This number was sufficient to determine PLE(l) up to l=165 ms with reasonable statistical accuracy. A longer test run might show a different picture with PLE scores that are sometimes lower, because nodes have a bigger chance of experiencing a 'bad day'.

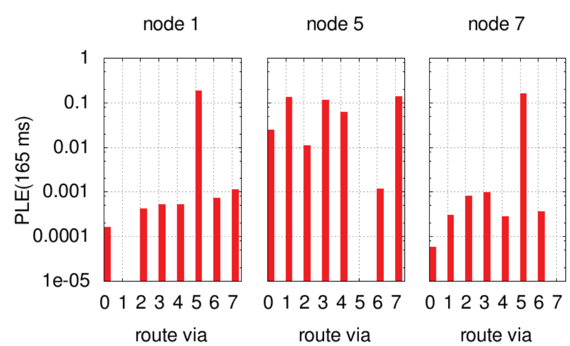

Figure 7. PLE(165 ms) for all 1-hop and 2-hop routes starting at nodes 1, 5 and 7.

Figure 7 shows some representative measurement results. As in figure 3, we see a spread in route quality. For all nodes but node 5, the 1-hop route directly to node 0 (the bar labelled 0 in the graphs) out-performs all 2-hop routes by a significant margin.

8. RSSI USAGE

As it takes a long time to measure PLE curves accurately, an obvious question is whether PLE could be predicted based on RSSI given the contradictory results from the literature presented in section 2. In the same test run we also measured the average RSSI value of each link. Also we devised a measure to combine the RSSI values of the two links involved in a two hop path (called the route RSSI). In figure 8 we plot the relation between PLE(165 ms) and the route RSSI. The route RSSI for a 2-hop route is computed by multiplying the RSSI values of the two hops. We also tried addition and taking the minimum, but found that multiplication gave the best result in terms of PLE correlation. Looking at the 2-hop routes only, we see that a very low RSSI, RSSI <20, is a good predictor for a bad route. For RSSI>20 however, there is no longer any visible correlation between RSSI and route quality. Apparently, in this region, the link budget along the route is so high that, with the number of retries we do, it stops being the dominant mechanism in determining PLE. Therefore, if we are to find the best 2-hop routes with a high probability, we have to measure their actual PLE, though an RSSI cut-off at 20 can be used to reduce the number of routes to be measured.

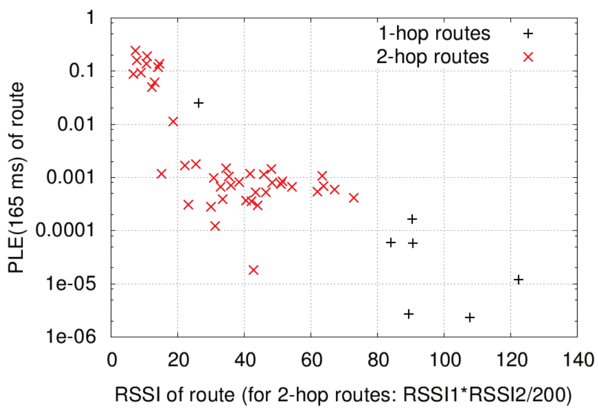

Figure 8. PLE(165 ms) versus route RSSI in test network.

Another technique to save resources in selecting routes is to base selection on PLE(*l*) measurements with *l*<165 ms, which can be measured more quickly. We have observed that the PLE curves hardly cross in the region from 85 to 165. Therefore using PLE(85 ms) instead of PLE(165 ms) could lead to nearly as good route selection results, while measurement times are shorter.

9. NUMBER OF CANDIDATE ROUTES

We now turn to the question of how many candidate routes a node should maintain in order to achieve the best possible PLE. As more candidate routes are added, especially routes with an individual PLE much worse than the PLE of the best candidate route, we can expect diminishing returns, or even a worsening of the PLE.

First, we run a one-week test comparing the performance of having *c* candidate routes in a node, with *c* in the range 1-4. The *c* candidate routes used by a node are always the *c* best (lowest PLE) routes identified in the test of section 7. A node first tries the best candidate route twice, and then tries the other candidate routes in from best-to-worst order. If the message is still not delivered, it tries the best route twice again, etc. The result of this test is that having multiple candidate routes outperforms having only one candidate route. However, after sending 100.000 test messages for each *c* for each node, we find no statistically significant difference in the PLE(200 ms) for 2, 3, and 4 candidate routes. In practical terms, having 2 candidate routes is as good as having 4 in this test.

In a second one-week test, we studied the performance of multiple candidate routes if all candidate routes are 'long' 2-hop routes. We eliminated the direct route to node 0, and the route via node 2 which has one very short hop, as candidates, and ran the test again with the remaining best routes. Ignoring inter-node interference, the results are therefore somewhat indicative of the PLE that can be expected in a larger test network, for nodes that are too far away from node 0 to reach it with a single hop. Figure 9 shows the test results. For node 5, we see a clear improvement in PLE(200 ms) when more candidate routes are added. For all other nodes, these is an improvement when going from 1 candidate route to multiple routes, but again no statistically significant differences for 2, 3, and 4 candidate routes. Figure 10 shows averages for the test results in figure 9.

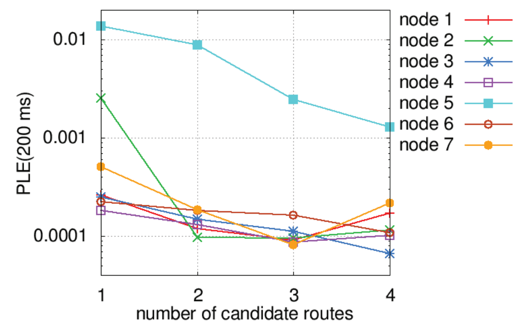

Figure 9. PLE(200 ms) for 'long' 2-hop routes only.

We conclude that in a dense network, for example the network of figure 1 but with node 5 excluded, keeping 2 candidate routes will be sufficient, and this can keep routing tables small. A more general networking solution should however have the ability to maintain more than 2 candidate routes, to optimize the PLE for relatively isolated nodes like node 5.

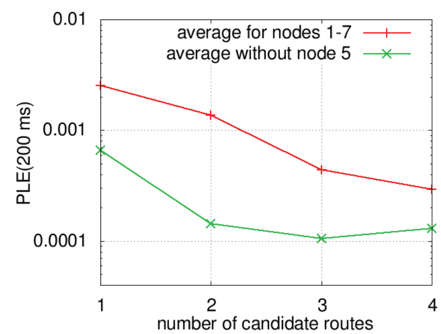

Figure 10. Average PLE(200 ms) for 'long' 2-hop routes.

For routing via long 2-hop routes only, figure 9 shows a best-case PLE(200 ms) of about 0.0001, which is high compared to the PLE(200 ms) values found in the test of figure 3, where direct routing to node 0 is allowed. Apparently, another factor than multipath fading is dominant in determining the value of the best case PLE(200 ms). It might be possible to lower the influence of this factor by re-tuning the system, for example by changing WT, decreasing the message sending rate per node (which lowers system self-interference), or changing the MAC parameters. This is a topic for further study.

10. PLE IN TIME SLOTTED NETWORKS

The use of a time-slotted 802.15.4 MAC, instead of CSMA, can be beneficial to latency in networks that need to handle high traffic loads [22], because it avoids collisions and hidden node problems. In time slotted networks, a sensor node is constrained in the opportunities it has to use the channel for retries: so the expected PLE curves are different from the ones shown above, where nodes will use the channel with a very high duty cycle when retrying. Furthermore, time slotted networks get more efficient at carrying high traffic loads if slots are kept small, with less reserved time in a slot for MAC retries [22], so it is preferable in time slotted networks to run with a low setting for the MAC retry parameter, a low SMRT. It is interesting to study what happens to PLE if a test network with the algorithm of figure 2 is run with a lower SMRT, approximating a time slotted network.

Experiments were done on the test network shown in figure 11. This test network used the same node hardware as in figure 1, but was built in a different location in the same building. (The location of figure 1 had been refurbished, removing most metal closets, preventing us from reproducing the setup of figure 1.)

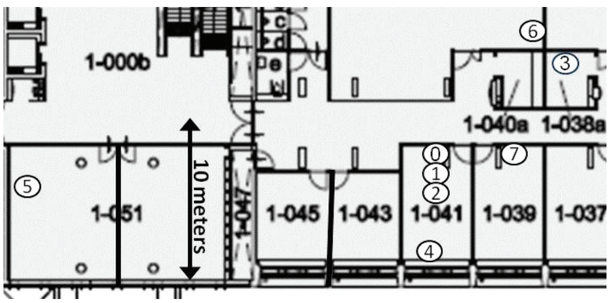

Figure 11. New test network. Sensor nodes 1-7 send messages to node 0.

Figure 12 shows the effect of different SMRT values on the PLE, with WT=40 ms, during a 3-week test. Each node does all delivery attempts over a 1-hop route to node 0, and node 0 always uses SMRT=4 when sending back its end-to-end acknowledge packet.

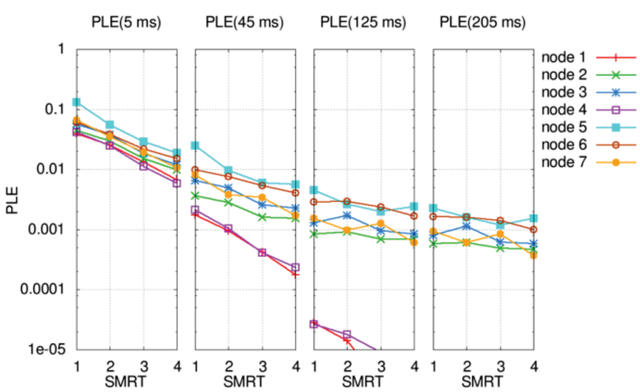

Figure 12. PLE values when routing directly to node 0, for different SMRT values (=number of MAC retries done in each delivery attempt).

Leftmost in figure 12, the PLE(5 ms) value can also be interpreted as the probability that the message is not delivered after a single MAC invocation. As expected, with a larger SMRT the probability of non-delivery is lower. Less intuitively, the graphs for PLE(125 ms) and PLE(205 ms) do not show any statistically significant effect of SMRT on the probability of delivering the message within the deadline. So, under the low traffic loads in our test network, when we go from SMRT=4 down to SMRT=1, operating on the channel like a time-slotted protocol with a 40 ms cycle time, PLE(205 ms) is hardly affected: the lower node channel occupancy associated with time slotting does not have a negative effect. For a higher network workload, lowering SMRT in a time slotted network is expected to improve worst-case latency [22]: a lower SMRT means shorter slots, so each node actually has more frequent opportunities to retry and empty its re-send queues. Overall, these results indicate that moving from CSMA to time-slotting with a fast cycle time is not incompatible with optimising for human-observable PLE.

Looking overall at the performance of the network nodes, we see that PLE(205 ms) for node 1 and 4 are so low that they are off the charts. All other nodes have a PLE(205 ms) around 0.0001, with much less spread than in the measurements done on the first test network (figure 3). The bad performance of node 2 in this test is somewhat surprising, given its location. In later tests with this network, node 2 performed better, so apparently it was just very 'unlucky' in the 3 weeks of this test.

11. ENERGY SCAVENGING SENSOR NODES

Given the positive result of the previous section, showing that a lower channel access rate is not incompatible with optimising PLE, the topic of energy scavenging nodes comes in scope. An energy scavenging sensor node is not powered by a battery or mains power connection, but extracts the energy it needs from its environment [23]. Consider an energy scavenging sensor node that uses a small solar cell to (re)charge a capacitor. Whenever a message needs to be sent, the radio will have to be used intelligently and sparingly, to minimise the probability of non delivery before the capacitor runs out of charge. It is not always possible or economical to put a very large capacitor in a low duty cycle energy scavenging sensor, to store a large reserve of energy just in case. Larger capacitors have larger leak currents: they require larger energy scavenging mechanisms just to keep them charged fully. Thus, when considering if an energy scavenging node using a capacitor of a certain size can be an acceptable control network product to an end user, first of all, we have to consider the probability that the node will not deliver the control message at all.

We used the testbed of figure 11 to study this probability. We ran a 3-week test, with all sensor nodes trying to send directly to node 0 (using no alternate routes), using different waiting times WT between sends, and computing from the test logs the probability of non-delivery after N MAC invocations. We use SMRT=1, to keep the energy used per MAC invocation as low as possible. This also allows us to refer to one MAC invocation as being 'one send' in the graphs and discussion below. Following the algorithm in figure 2, our sensor nodes use carrier sense to try to detect a clear channel before sending a message, whereas a typical energy scavenging node will omit the carrier sense as it uses a lot of energy. The test results, in figure 13, are therefore most representative of an energy scavenging network with a very low channel duty cycle.

Figure 13 clearly shows that reliability can be improved not just by sending more often, but also by waiting longer between sends. The nodes 1, 2, and 4, all very close to node 0, performed very well in this test run, with a failure probability so low that it is off the charts as N grows larger. The effect of increasing WT on reliability is large: if latency is not a concern (e.g. for a type of sensor node where the user does not notice or care about a long latency) the best strategy for the node is to use a WT of 500 ms or even larger. However, for many types of energy scavenging nodes we would like to optimise both the probability of delivery and the PLE(200 ms).

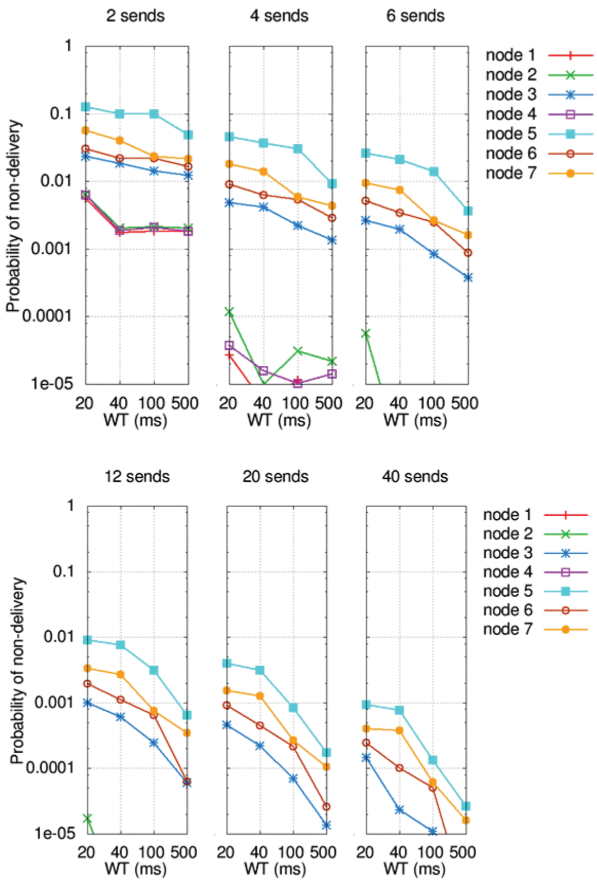

Figure 13. Probability of message non-delivery within N sends by an energy scavenging node, with different waiting times WT between sends.

12. ENERGY SCAVENGING NODES AND MULTIPLE CANDIDATE ROUTES

The results in sections 6 and 9 predict that an energy scavenging node might benefit from using multiple candidate routes when trying to deliver its message. We tested this prediction, with up to 3 candidate routes, as follows. We moved node 5 in figure 11 to office 1-041, locating it to the right of node 4, and then programmed node 1 and 5 to act as routers, with SMRT=4. All other nodes are set up to behave as energy scavenging nodes: we set SMRT=1, and configured them to use up to 3 routes: the direct route to node 0, the route via 1 to 0, and the route via 5 to 0. The nodes use WT=120 ms when trying via 1 route, WT=60 ms when trying via 2 routes, and WT=40 ms when trying via 3 routes – so each route is always tried once every 120 ms. Test results are shown in figure 14.

Leftmost in figure 14, when 3 sends are done, the effects of having multiple candidate routes are somewhat mixed. More significantly however, with 6 or more sends the use of multiple candidate routes significantly improves the overall network performance. For 6 sends, we have a satisfying result in that the probability of non-delivery is lower than 0.001 for all nodes, something that was not achieved after 6 sends in figure 13, even not with WT=500 ms. The use of multiple candidate routes also optimises PLE. In this test, when using 3 candidate routes with WT=40 ms, 6 sends are done within 200 ms, so with 3 candidate routes we obtained a PLE(205 ms)<0.001 for all nodes. It is clear that using multiple candidate routes is a very useful technique to increase the reliability of energy scavenging sensor nodes, especially if low latency is desired too.

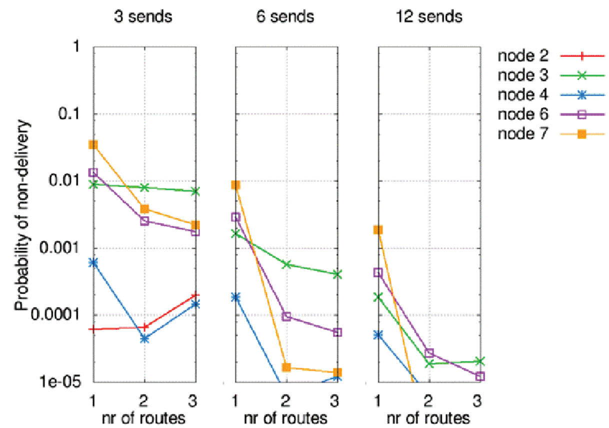

Figure 14. Probability of message non-delivery with different numbers of candidate routes and SMRT=1.

13. CONCLUSIONS

To make 802.15.4 control networks acceptable as a replacement for wired control, we must look into the way that end users experience the latency characteristics of the network. Based on this consideration, we have introduced a latency measure PLE, and optimized PLE(200 ms) based on measurements done during working hours in an office environment. Test runs lasting at least a week were done with most links in the clear region. The test results indicate that each node should maintain a list of multiple candidate routes that it can use to deliver a message. We have shown in [21] that candidate route information can be created and stored in advance. Only infrequent updates are necessary to adapt to changes that occur over timescales of weeks.

Compared to most WSN test networks in literature, our test network has a higher node density, as we expect multiple wireless sensors and actuators per room. This leads to a system where, if the retry strategy in the protocol is designed well, low link quality is no longer a dominant cause of latency deadlines being exceeded. Instead, for PLE(200 ms), fluctuating multi-path cancellation becomes a dominant cause [21], which can be eliminated by using multiple candidate routes. When this cause is eliminated, the presence or absence of a clear-region 1-hop link to the destination becomes a major determinant. We have some tentative evidence that system self-interference will become a significant determinant at some point above an average rate of 40 packets per second in the network, not counting MAC acknowledge packets. This high average packet rate is unrealistic in a small building control network, but might be approached in a large control network that needs to use multi-hop routes often to deliver messages. It is likely that the introduction of an exponential back-off in the invoking the MAC will improve worst-case latency under higher network loads.

While our test network uses CSMA, our measurements predict that time slotted networks with a short cycle time are not necessarily at a disadvantage when it comes to achieving a good PLE(200 ms).

Finally we have looked at the problems faced by energy scavenging nodes based on capacitors, that can only do a few retries before their energy runs out. We have measured for such nodes, across the parameter space of capacitor size and retry speed, the expected reliability of message delivery over a single hop fixed route. We then show that the technique of using multiple candidate routes can be very useful to optimize the reliability of energy scavenging nodes, while simultaneously achieving a good PLE(200 ms).

There are still many unanswered questions related to the PLE(200 ms) quality metric. In particular, the problem of predicting or optimizing PLE under high network loads has not yet been addressed.

14. REFERENCES

[1] J. Zhao, and R. Govindan, *Understanding Packet Delivery Performance in Dense Wireless Sensor Networks*, SenSys 2003.

[2] A. Woo, T Tong, D. Culler, *Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks*, SenSys, 2003.

[3] IEEE Computer society, part 15.4, *Wireless medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless personal Area networks (LRWPAN),* Institute of Electrical and Electronics Engineers, Inc, 2003.

[4] J-S. Lee, *An Experiment on Performance Study of IEEE 802.15.4 Wireless Networks, ETFA 20*05.

[5] O. Hyncica, P. Kacz, P. Fiedler, Z. Bradac, P. Kucera, and R. Vrba, *The Zigbee Experience,* ISCCSP 2006.

[6] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, Performance Study of IEEE 802.15.4 Using Measurements and Simulations, WCNC 2006.

[7] M.M. Holland, R.G Aures, W.B. Heinzelman, Experimental Investigation of Radio Performance in Wireless Sensor Networks,2nd IEEE workshop on wireless mesh networks, 2006.

[8] M. Zuniga, and B. KrishnamaChari, *Analyzing the Transitional Region in Low-Power Wireless Links*, SECON, 2004.

[9] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan. and C. Elliott, *Experimental evaluation of wireless simulation assumptions*, Dartmouth Computer Science Technical report, TR2004-507, June 2004.

[10] A. Cerpa, J.L. Wong, M. Potkonjak, D. Estrin, *Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing*, MobiHoc, 2005.

[11] A. Meier, *Safety-Critical Wireless Sensor Networks*, PhD thesis 18451, ETHZ, 2009.

[12] D.S.J. de Couto, D. Aguayo, B.A. Chambers, R. Morris, *Performance of Multihop Wireless Networks: Shortest Path is Not Enough,* ACM SigComm communications review, Vol 33, No 1, 2003.

[13] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, *Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks*, Technical Report SING-06-00, 2006.

[14] G. Zhou, T. He, S. Krishnamurthy, and J.A. Stankovic, *Impact of Radio Irregularity on Wireless Sensor Networks,* MobiSys '04, June 2004.

[15] Y. Wang, M.C. Vuran, S. Goddard, *Cross-layer analysis of the End-to-end Delay Distribution in Wireless Sensor Networks*, 30th IEEE RTSS, 2009.

[16] T. He, J.A. Stankovic, C. Lu, T. Abdelzaher, *"SPEED: a stateless protocol for real-time communication in sensor networks,* ICDCS-23, 2003.

[17] M. Soyturk, D.T. Altilar, *Reliable Real-Time Data Acquisition for Rapidly Deployable Mission-Critical Wireless Sensor Networks*, IEEE INFOCOM, 2008

[18] R. S. Oliver, G Fohler, *Probabilistic Routing for Wireless Sensor Networks*, 29th IEEE Real-RTSS WiP 2008.

[19] K. Holtman. *Long-duration Reliability Tests of Low Power Wireless Sensing and Control Links in an Office Environment*. In: ARCS 2010 Workshop Proceedings of the 23th Int. Conf. on Architecture of Computing Systems 2010, pp 259-258.

[20] http://www.jennic.com/

[21] K. Holtman; P. van der Stok. *Routing recommendations for low-latency 802.15.4 control networks.* In: ARCS 2011 Workshop of the 24th Int. Conf. on Architecture of Computing Systems 2011, pp 285-290.

[22] P. Jurcik, R. Severino, A. Koubaa, M. Alves, E. Tovar, "Dimensioning and Worst-case Analysis of Cluster-Tree Sensor Networks", ACM Transactions on Sensor Networks, Volume 7, Issue 2, Article 14, August 2010.

[23] H. Raisigel, G. Chabanis, I Ressejac, M. Trouillon. Autonomous Wireless Sensor Node for Building Climate Conditioning Application, Sensor Technologies and Applications (SENSORCOMM) 2010.