ESTIMATION OF THE PROBABILITY DENSITY FUNCTION OF END-TO-END DELAYS IN WIRELESS SENSOR NETWORKS
- TECHNICAL REPORT -

Ramon Serna Oliver
Technische Universität Kaiserslautern
Chair of Real-Time Systems
Kaiserslautern, Germany
serna_oliver@eit.uni-kl.de

January 2009

Abstract

The nature of Wireless Sensor Networks (WSN) prevents applying classic real-time methods unless restrictive assumptions are taken about the participating entities. Thus, for applicability in realistic deployments alternative methods capable of offering meaningful Quality of Service (QoS) based on realistic assumptions are needed.

This technical report presents an approach to estimate probabilistic timeliness guarantees of end-to-end message delivery delays in WSN. Each node computes at run-time local statistics about its message forwarding latency with low computational and memory requirements. The composition of this local information is used at run-time to construct a metric which estimates the probability density function (*pdf*) of the end-to-end latency of a path. This metric benefits adaptive QoS as it is constantly updated at run-time and reflects the actual network status. Simulation results underline the accuracy of the method.

Chapter 1

Introduction

1.1 Basic concepts

Wireless Sensor Networks (WSN) are formed by a set of resource constrained nodes communicating via hop-by-hop message forwarding and a small set of data sinks. Characteristics of WSN are of high variability among different application domains. For instance, size of the network and density of nodes are two parameters with a great variability, as well as the mobility of nodes and exposure to ambient phenomenon.
There are, however, a number of restrictions common in WSN deployments. These include the unreliability of their links [18] and the absence of stationary relays, as well as unknown network topology. These, together with strong energy constraints represents a major challenge to provide any sense of timeliness guarantees.

1.2 Challenges

Existing real-time methods targeting timeliness guarantees in different sets of wireless networks do not apply in the case of WSN. For instance, strategies based on resource reservations [12] would over-constrain the network capacity up to the point of loosing feasibility.
Classic methods based on hard real-time (HRT) demand deterministic behavior at each of the network layers, which is only possible under the assumption of ideal environments [6]. For example, at the MAC layer, bounded delays might

be achieved by means of periodic sensing of the medium and neighborhood synchronization, which is often not affordable in terms of energy. Similarly, at the routing level, global knowledge of the network topology (i.e. routing table) does not fit in the limited memory capacity of a node. On top of this, assumptions of perfect channel conditions are common and represent the grounds for many existing approaches. Unfortunately, WSN have proved to present a high bit error rate (BER) and due to their application domains, a higher exposure to environmental phenomenon than classic wireless networks (e.g. Wi-Fi).
Methods based on queuing theory answer whether a message can or cannot meet its deadline based on is service and/or inter-arrival times. However, offline estimations of this parameters are not accurate, hence need to be tuned at run-time.
The large number of limitations that prevent bounded end-to-end delays to be guaranteed in real WSN deployments suggests a different approach.

1.3 Introduction to the presented approach

In this technical report we present a probabilistic metric to evaluate end-to-end timeliness performance at run-time. This metric estimates the *probability density function* (*pdf*) of the end-to-end latency of any given routing path. The required computations demand little processing power as well as memory capacity and is continuously updated to reflect the change over time of the network status. Moreover, previous knowledge of the network status is not needed.
The accurate monitoring of the timeliness performance introduced by this metric can be used to calibrate off-line models or given to the application layer to receive feed-back about the network performance. We believe that the simulation results presented later in this paper show convincing arguments of its applicability.

1.4 Structure of this document

The rest of the paper is organized as follows: Section 2 explores the related work in this field. Section 3.1 describes the construction of the timeliness monitoring metric, which is later validated in Section 4.1. Finally, Section 5.1 concludes the paper.

Chapter 2

Related work

Ongoing research is carried out in this area at different levels.

2.1 Routing protocols

At the routing level, [5] and [9] assign velocities to messages which must be kept in order to fulfill their timeliness requirements. However, both assume static networks and nodes equipped with localization capabilities. In [13] delay guarantees are provided at the expense of limiting the length of routing paths.

2.2 MAC protocols

At the MAC level, [4] achieves hard real-time guarantees given an hexagonal topology of static nodes. This requirement is later relaxed in [16] although it still relies on static nodes. Besides, both papers are built on the assumptions of bounded network density and optimum communication conditions.
Much of the existing research is based on TDMA scheduling of neighboring nodes (e.g. [8]). Although valid results are obtained in controlled environments, the common restriction of this methods is the assumption of absence of network errors, which might cause retransmissions and jeopardize the transmission schedule.

2.3 Schedulability analysis

[1] approaches a sufficient schedulability condition to guarantee end-to-end delays in multi-hop WSN under specific assumptions on the message transmission times and channel transmission speeds as well as network density and path lengths.

2.4 Queuing theory

[10] makes use of queuing models to determine the expected transmission latency of messages on each hop. This translates into a traffic regulation mechanism to drop messages without expectations to meet their deadlines. Additionally, under unstable conditions, it estimates the PDF of the delay distribution with a Gaussian distribution, although little motivation is given to justify the choice.

Chapter 3

Probabilistic approach

3.1 Timeliness monitoring

The objective of timeliness monitoring is to build a metric which, at run-time, evaluates the timeliness performance of an end-to-end path. However, we argued before in this paper that an accurate end-to-end delay estimation (i.e. guarantee) is not feasible due to the principles of WSN. Therefore, we introduce probabilistic analysis, such that the metric reflects an estimated probability of the end-to-end latency of a path.
This metric can be used at different levels depending on the network strategy. A routing protocol can be set to propagate this information towards the sink and/or back to the source in order to adjust the routing decisions to the quality of the path. Similarly, an application can decide on the strategy to follow if the observed metric does not satisfy its timeliness requirements. And at a different level, path discovery strategies can benefit of this metric at the time to decide among a number of options to build a routing tree.
The focus of this paper is to validate and evaluate the accuracy of the metric to represent the current status of the network in terms of timeliness performance. Existing routing protocols can be adapted to make use of this metric and take routing decisions based on the probability of end-to-end delays.

3.1.1 Notation and definitions

A WSN can be represented as a graph $G(N, L)$ formed by a set of nodes N and a set of *single-hop links* L. Two nodes $n_i, n_j \in N$ are *directly connected* at a given

time if there is a link $l \in L$, $l = (n_i, n_j)$ such that n_i and n_j can send and receive messages from each other. $S \subset N$ is the subset of sinks. Sinks might outperform nodes with respect to resources and energy availability.
A *[routing] path* rp_{n_1,n_q} is a sequence of links $(n_1, n_2)..(n_{q-1}, n_q)$ such that each intermediate node in the path is directly connected to the next one, thus providing a *multi-hop link* between the first node (source) and the final node (destination). A path s is contained in another path p if all the links belonging to s also belong to p. In that case, we define s as a *segment* of p. The length of a path is equal to the number of links belonging to it (hence, $|rp_{n_1,n_q}| = q$).

3.1.2 Probabilistic estimation of end-to-end delays

Information regarding end-to-end timeliness performance is crucial for time sensitive applications. In this paper we target wide-area WSN such as those related to environment monitoring (e.g. fire detection, structural monitoring of buildings, etc.) with specific timeliness sensitive data acquisition (e.g. fire or intrusion alarms, structural damage, etc.).
Routing paths are built by the routing protocol to provide a link between the sensor node and the data collector (sink). In such scenarios, nodes will continuously produce data at a certain frequency, and hence the paths are likely to be re-used for a certain period of time. During the path building procedure, a routing protocol needs to decide among multiple paths based on predefined criteria. The probabilistic timeliness metric fits well in this procedure as it can originate paths with higher probability of performing well.
Moreover, while sending data through a path, the metric reflects the change over time of the timeliness performance, hence aiding on the decision of re-routing or not. Furthermore, the application can benefit of this information as, in fact, it reflects the probability of achieving its timeliness requirements.

Single-hop forwarding latency

In the simplest case, a path is composed of only one link (that is, $|rp| = 1$). In this case, given two hops n and s such that there exists a link $l = (n, s)$, we define D_n as the *random variable* (RV) which characterizes the transmission latency of a message from n to s and $p(D_n)$ as the *pdf* such that $p_n(\varepsilon) = P(D_n \leq \varepsilon)$; the probability that n introduces a delay of at most ε in forwarding the message to the next hop.
The transmission latency of a message δ is considered as the time between the

message entering a node t_{in} (either because the application layer sets a new message to be sent, or because the MAC layer receives a message which has to be forwarded), and the reception of an acknowledgement from the receiver t_{ack}. Notice that this calculation is pessimistic as it introduces the additional time to receive the acknowledgment. Depending on the MAC protocol this might be relatively small and can be neglected, but in any case it should be possible to estimate this time as a constant ρ. Hence,

$$\delta = t_{ack} - t_{in} - \rho \tag{3.1}$$

The above calculation is done at each hop every time a message goes through, thus an increasing sequence of values $\rho_0, .., \rho_k$, is generated at each node. Each of these values represents a sample of the random variable D for the given hop.
If the distribution of D was known, finding out the expected value and its probability or the probability of any given delay would be reduced to trivial. Unfortunately, at this point we can not extract any conclusion about the distribution of D. To proceed, we calculate the sample mean $\bar{x}$ and sample variance s^2 that characterize this distribution. For short paths (i.e. a few links), $\bar{x}$ should be a good estimator of the latency and s^2 provide a rough indicator of the link quality: hops with high variance may be experiencing a higher number of retransmissions.
We propose the *exponential weighted moving average* (*EWMA*) [3] as a mean to obtain this calculation with little memory utilization and low CPU overhead (Equation 3.2). A parameter α ($0 \leq \alpha \leq 1$) is set to weigh the actual measurements with respect to the past, hence smoothing the consequences of the past trends and possible aberrations.

$$\begin{aligned} \bar{x}_n^* &= \alpha\delta + (1-\alpha)\bar{x}_{n-1}^* \\ s_n^{2^*} &= \frac{\alpha}{2-\alpha} s_n^2 \end{aligned} \tag{3.2}$$

Equation 3.3 provides the sample variance s^2 with low memory requirements.

$$s_n^2 = \frac{n-1}{n} s_{n-1}^2 + \frac{n-1}{n}(x - \bar{x}_n)^2 \tag{3.3}$$

Equation 3.3 and Equation 3.2 are updated at each node every time a message is forwarded through it.

3.1.3 Averagin parameters

Estimation of end-to-end latency

The end-to-end latency of a path rp is also a RV, D_{rp}, formed by the composition of the delays of its intermediate links:

$$D_{rp} = \sum_{\forall (i,j) \in rp} D_{(i,j)} \tag{3.4}$$

and,

$$p_{D_{rp}}(\tau) = P(D_{rp} \leq \tau) \tag{3.5}$$

Then, assuming that the *pdf*s of the RVs have the same distribution, non-negative and mutually independent, it is possible to apply the *Central Limit Theorem* (*CLT*) [2] to characterize the *pdf* of the path as a normally distributed[1] RV, $D_{rp} \sim N(\mu_{D_{rp}}, s^2_{D_{rp}})$ [17]. The assumption of all RVs being mutually independent is taken as a premise at this point and discussed in Section 3.1.3.

$$\begin{aligned} \mu = \bar{x}_{D_{rp}} &= \sum_{\forall l \in rp} \bar{x}^*_{D_l} \\ \sigma^2 = s^2_{D_{rp}} &= \sum_{\forall l \in rp} s^{2\,*}_{nD_l} \end{aligned} \tag{3.6}$$

Therefore, under these circumstances the probability introduced in Equation 3.5 converges to

$$\begin{aligned} p_{D_{rp}}(\tau) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d} e^{\frac{y^2}{2}} dy \\ \tau &= \frac{D_{rp} - \mu_{D_{rp}}}{\sigma_{D_{rp}}} \end{aligned} \tag{3.7}$$

In this case, the expected value for the end-to-end latency of rp is $E(D_{rp}) = \mu_{D_{rp}}$.

Assumption of independence

The assumption of independence of the RV is necessary to apply the CLT. This might seem to be a strong assumption and indeed, it comes at a certain risk. Message latencies across a network might present dependencies under certain circumstances forcing events to happen in a non-independent way (i.e. $E[V_a, Vb] \neq$

[1] Although the CLT is commonly applied to large number of samples, an argumentation about good approximations for smaller sums of RVs is given in [11].

$E[V_a] \cdot E[V_b]]$). However, we initially took this assumption as a premise and expected that the spatial distribution of nodes and the typical low throughput of WSN would minimize this possibility.
After performing simulations, we believe that the dependencies are not of relevance in the general case and only in situations of high network saturation they might arise (e.g. packet dropping due to buffer overflows). The evaluation in Section 4.1 will show whether this assumption was appropriate and if not, how significant are its effects.

Chapter 4

Experimental results

4.1 Evaluation

To evaluate the estimation metric we performed extensive simulations with the simulation tool Omnet++ [15]. We chose WiseMAC [7] as an energy-efficient MAC protocol specially design for WSN.

4.1.1 Scenario

We simulated traffic messages from a sender node n to a sink s with the interference of cross-traffic coming from neighbor nodes as depicted in Figure 4.1. A common setup for each simulation run was chosen with variation in the length of the path and cross-traffic parameters:

- different experiments with path length: $|rp| = \{5, 10\}$,
- source node n sending periodic messages to sink with $T = 30s$,
- messages sent from n to s capture the aggregation of the estimated parameters at each intermediate link (Equation 3.6).

  a value $\alpha = 0.9$ is selected.
- the real end-to-end transmission latency experienced by each message is captured at s,
- each hop in the path has two neighbors, forwarding cross traffic to it following a Poisson distribution with parameter $\lambda = \{60s, 120s, 480s, 1200s\}$,

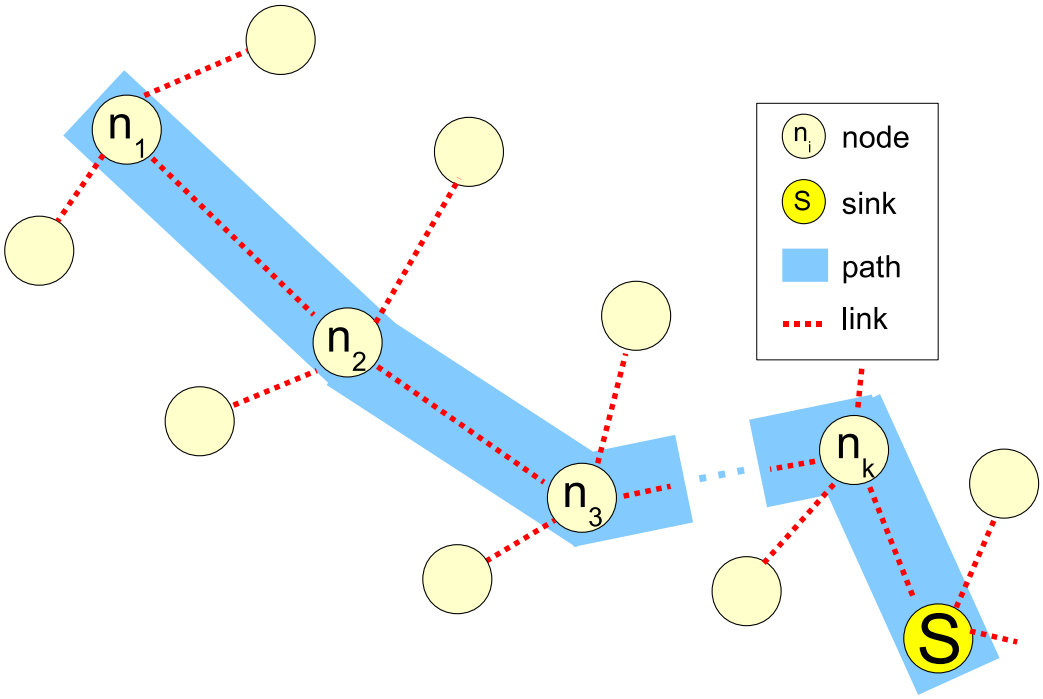

Figure 4.1: Scenario

- radial distance between nodes following a uniform distribution with range 8 to 20 meters,
- radio interface according to the specification of the RFM TR1100 radio transceiver.

Hence, the results of eight different simulation runs with the combination of parameters λ and $|rp|$ are presented. Each run simulated a period of 10 days. Notice that the process of building the routing path is not considered at this moment. The purpose of the simulations is to evaluate the validity of the method to obtain the *pdf* of the end-to-end latency distribution.

4.1.2 Evaluation criteria

To evaluate the application of the CLT with the assumptions of independence and of equally distributed RV at each intermediate hop, we analyze the real and estimated *pdf* of the routing path latency. Under static network conditions (i.e. no

transmitted message), $\bar{x}$ and s^2 would be constant at each hop. Therefore, a succession large enough of messages transmitted across the path would suffice to estimate the shape of its distribution function. This would give an estimation of the real distribution function of the end-to-end latency.
In such case, the real distribution could be compared to the Normal distribution with parameters μ and σ^2 (Equation 3.6) with the parameters captured by the messages (i.e. sum of $\bar{x}$ and s^2 at each intermediate hop).
Unfortunately, the construction of the metric invalidates this option. Each time that a message is forwarded by a hop, it recalculates its $\bar{x}$ and s^2. Thus, if a message is forwarded by any of the intermediate links of the path, it will produce a change in the final end-to-end parameters. In other words, the network conditions are varying every time that a message is being forwarded.
Therefore, for each message going through the path, we obtain a measured end-to-end delay (real) and a set of parameters μ and σ^2. However, with only one sample of the real distribution we cannot extract anything about its approximation to the Normal distribution, and as each real measurement has been taken at different instants of time, hence different network status, it is not possible to approximate the real distribution with the whole succession.
To overcome this problem we perform two complementary tests:
Test 1. Normalize each sample of the real distribution to the standard Normal distribution $N(0,1)$: If $X \sim N(\mu, \sigma^2)$, then $Z = \frac{X-\mu}{\sigma}$, where $Z \sim N(0,1)$. This way, instead of comparing each individual sample to a $N(\mu, \sigma)$ with different parameters, we can compare all samples to a $N(0,1)$. Thus, the expectation is that the distribution of the succession of normalized samples approximates a $N(0,1)$.
Test 2. Compare the number of "hits" of each interval determined by the distance σ from the center point (μ). This is known to be around 68%, 27%, 4.2% and 0.2% respectively for the intervals $I_1 = (-\sigma, \sigma)$, $I_2 = (-2\sigma, -\sigma) \cup (\sigma, 2\sigma)$, $I_3 = (-3\sigma, -2\sigma) \cup (2\sigma, 3\sigma)$ and $I_4 = (-\infty, -3\sigma) \cup (3\sigma, \infty)$. If the estimated distribution is accurate, the number of samples falling in each of these intervals should follow a similar proportion.

4.1.3 Simulation results

To analyze the simulation results, we first show two representative cases. For the experiment with $|rp| = 5$, Figure 4.2 shows the histogram and probability density function after normalization compared to the *pdf* of the standard Normal ($N(0,1)$). The depicted graphics have been cropped at the interval $(-4, 4)$.

At first sight, two questions arise: the difference between the two curves at the

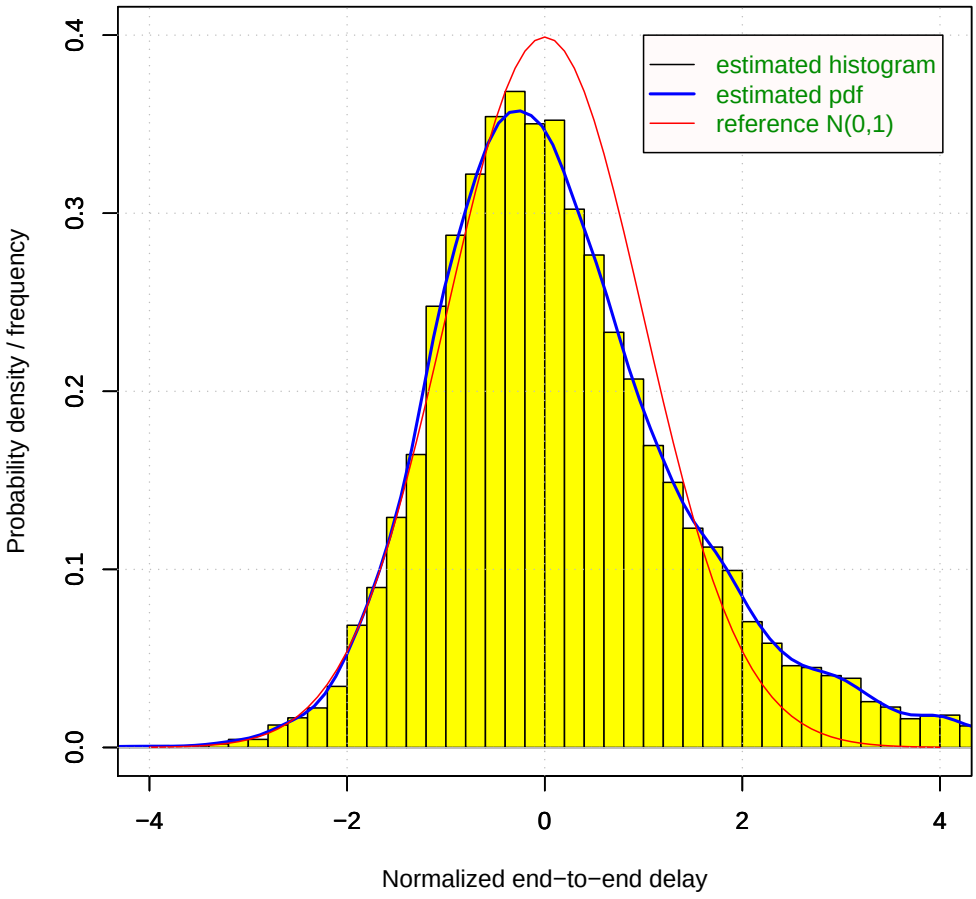

Figure 4.2: Normalized histogram and estimated pdf vs N(0,1), n=5

central point and the higher tail on the right side. Both effects are related to each other and can be explained by the nature of the experiment measurements. In fact, the values represented come from measured end-to-end delays. This necessarily introduces a tail effect, as there is a clear limit on the possible values from the left side (i.e. time delays cannot be negative) but none on the right side.
Looking at the range of absolute values, we see that with a mean sample value of

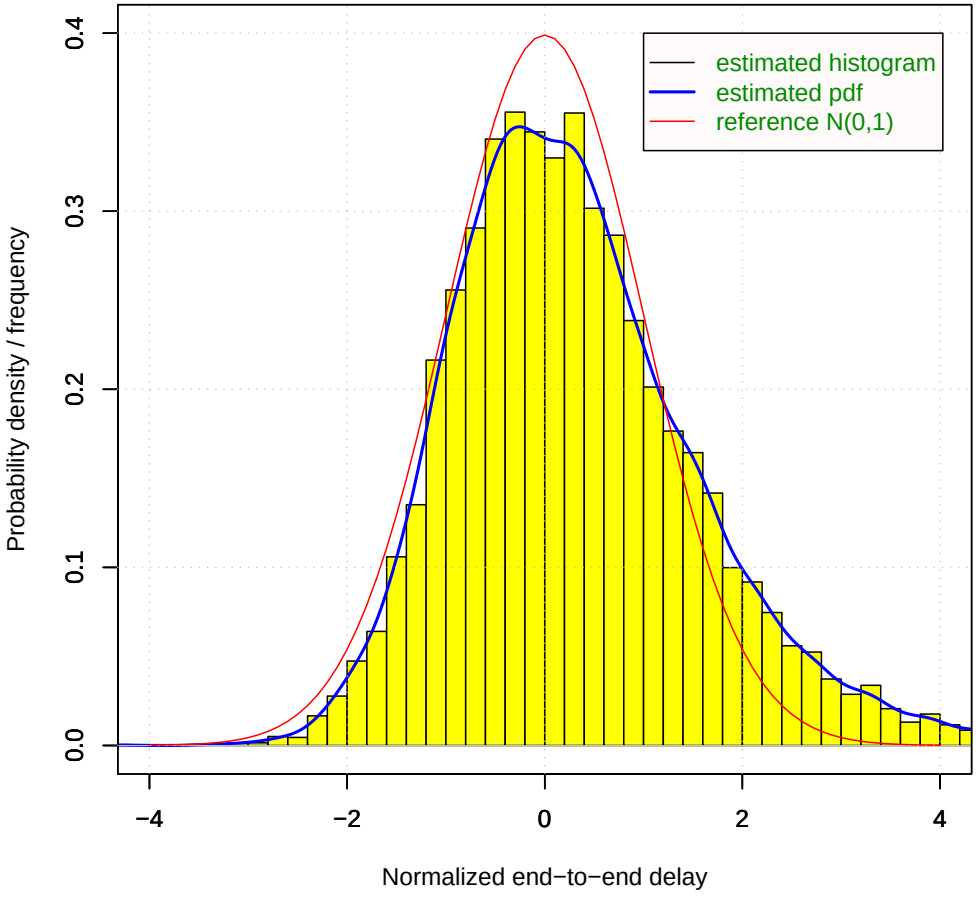

Figure 4.3: Normalized histogram and estimated pdf vs N(0,1), n=10

$6.5ms$ very few messages achieved a delay less or equal than $2ms$ and the distance between the minimum value and the mean is approximately of $5ms$. However, on the right side, this distance goes up to around $34ms$, with a maximum value close to $40ms$.

Notice that the α value performing the EWMA is responsible, in a certain way, of this effects. A lower α acts as a filter for higher sampled values and hence, reduces the tail on the right side. However, this would also affect the sample vari-

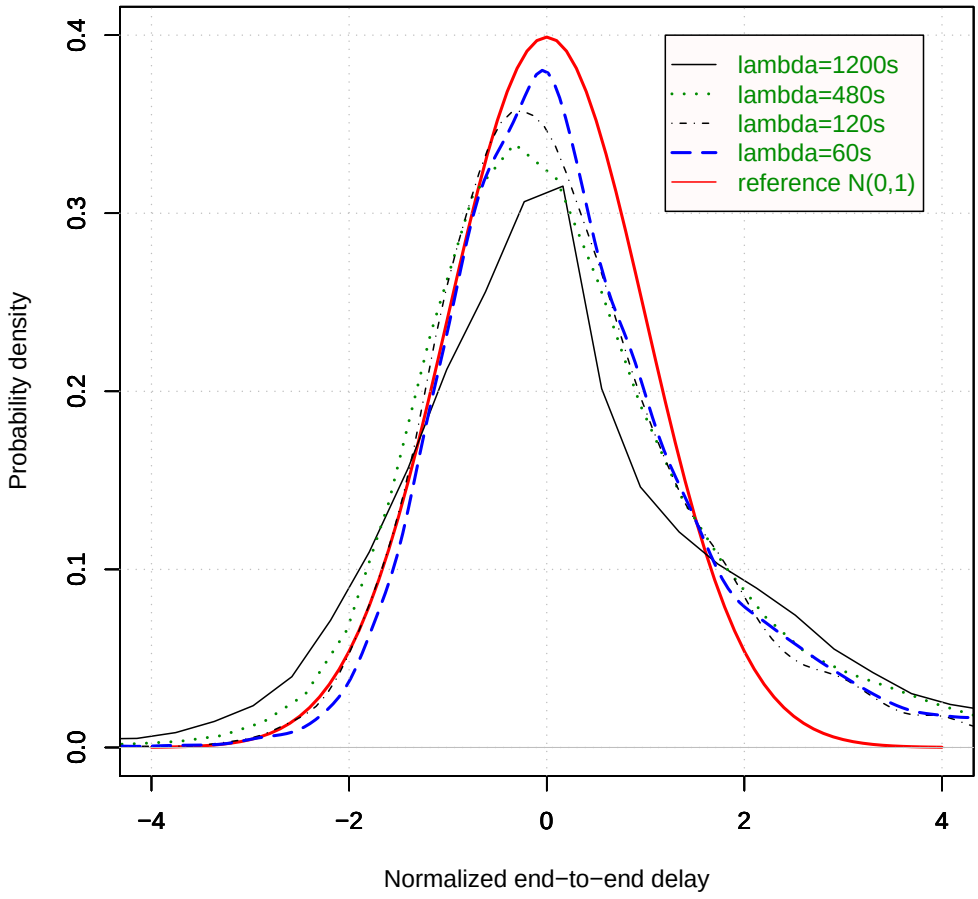

Figure 4.4: Estimated pdfs vs N(0,1) with path length 5

ance s^2 as values would get closer to each other. Thus, a side effect would be a distortion on the estimated distribution which would look thinner. On the other hand, higher values of α would reduce the smoothing effect of the EWMA and estimate a better value for the sample variance. This would definitely reflect on the peak of the estimated distribution, although, at the same time, produce a thicker distribution shape.

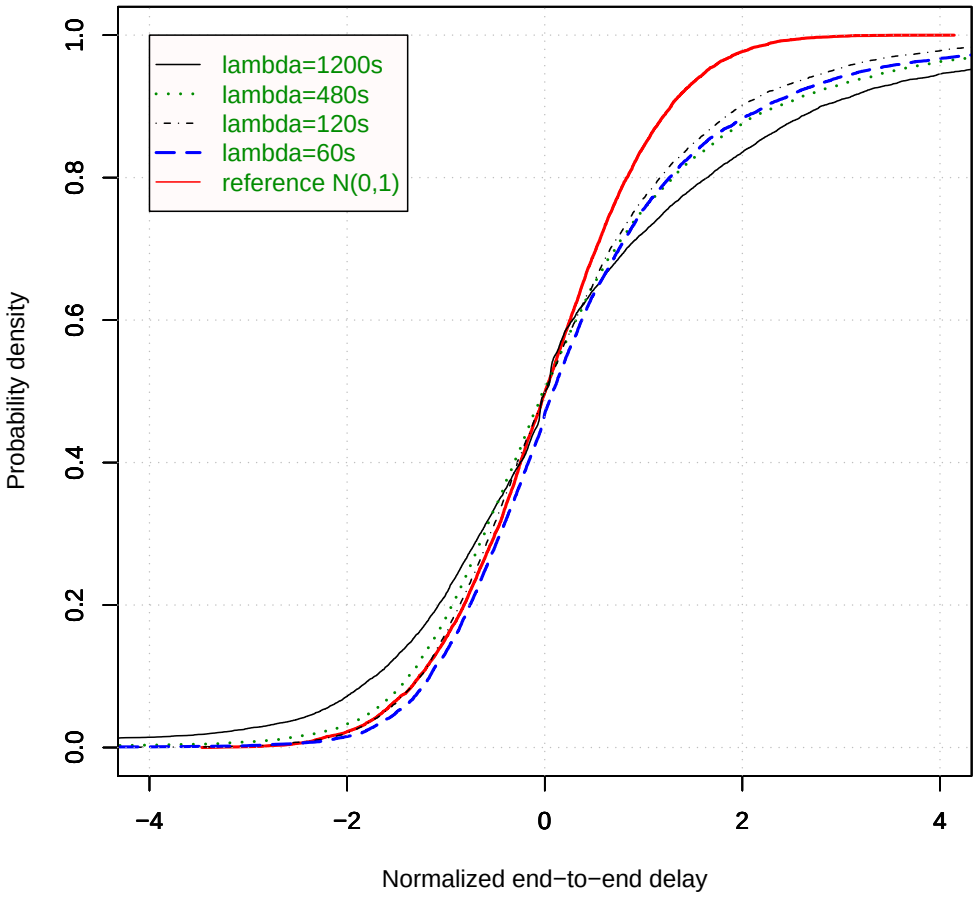

Figure 4.5: Estimated PDFs vs N(0,1) with path length 5

Figure 4.3 shows the same results for the case of $|rp = 10|$. This case does not differ much from the previous one, except for the fact that it is noticeable that the estimated *pdf* is slightly more centered than it was in the previous case. This again, is not an unexpected result as it was already presumed that longer paths would produce better estimated distributions. However, it is remarkable that even with paths as short as 5 hops it is possible to obtain relatively accurate results.
Figure 4.4 and Figure 4.5 show respectively the *pdf* and *probability distribution*

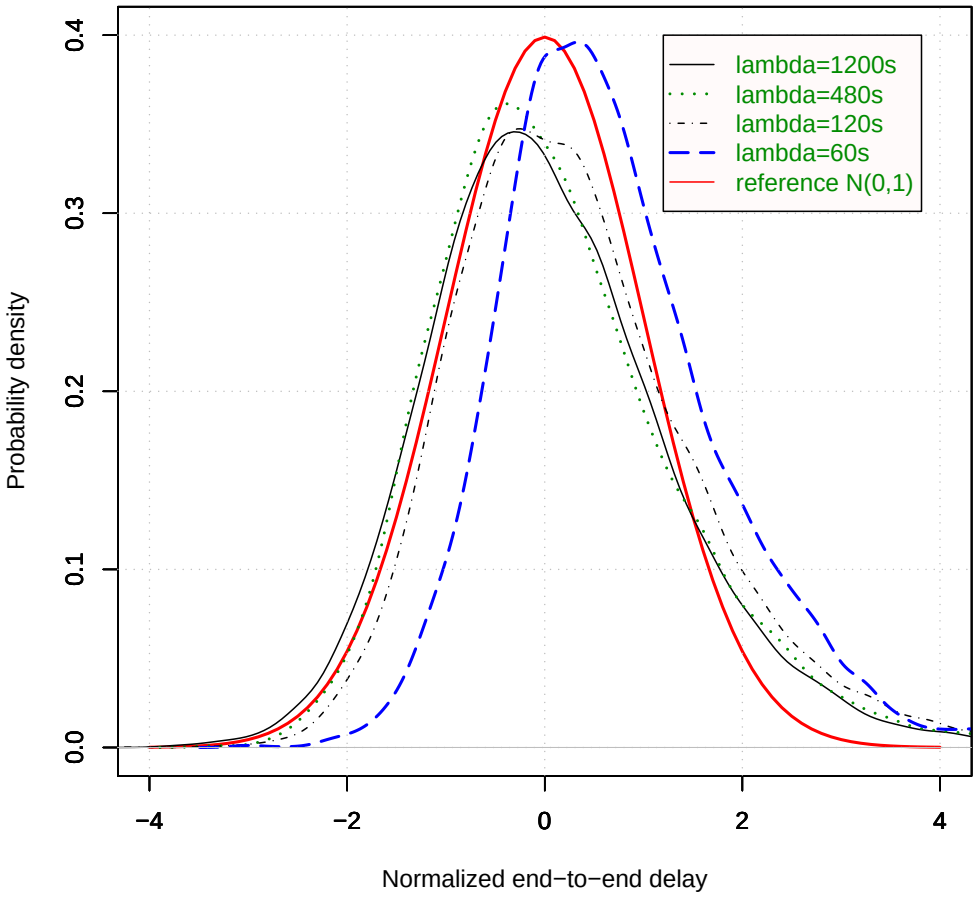

Figure 4.6: Estimated pdfs vs N(0,1) with path length 10

function (PDF) of all four cases with $|rp| = 5$ and variations in the cross traffic (λ parameter).
As can be observed, the accuracy increases proportionally to the cross-traffic parameters. This is due to the fact that the higher the amount of messages going through the network is, the more frequently intermediate nodes refresh their local estimations. In other words, if the traffic is too low, the estimated values at the arrival of a message loose accuracy by the time that the next message is received.

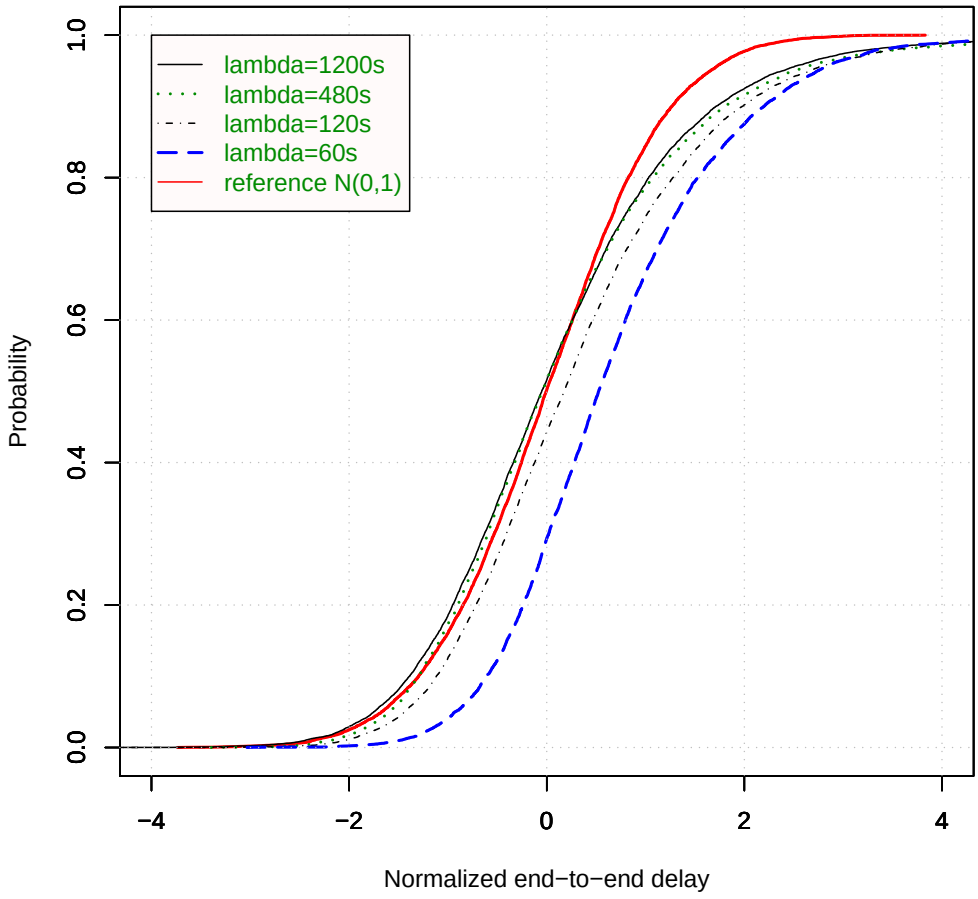

Figure 4.7: Estimated PDFs vs N(0,1) with path length 10

In Figure 4.5, the “lower peak” described before can be appreciated from the point of view of the estimated probability. The higher part of the curve is visibly bellow the reference curve, which means that the estimation becomes pessimistic (i.e. the method will predict a lower probability for delays above the expected end-to-end delay). However, the same does not happen, except for the case of very low traffic, in the lowest part of the curve. This means that the estimated probability for end-to-end delays bellow the expected value do not over-estimate the capacity of

the path.

Figures 4.6 and 4.7 repeat the same experiment with a path length $|rp| = 10$.

λ N(0,1)	I_1 (68%)	I_2 (27%)	I_3 (4.2%)	I_4 (0.2%)
60	62.2%	24.6%	7.1%	6.1%
120	61.1%	27.1%	7%	4.8%
480	53.3%	27%	8%	7.7%
1200	50.8%	25.6%	11.9%	11.8%

Table 4.1: Percentage of hits per σ interval with path length 5

λ N(0,1)	I_1 (68%)	I_2 (27%)	I_3 (4.2%)	I_4 (0.2%)
60	62.4%	24.9%	9.2%	3.5%
120	62%	27.1%	7.4%	3.6%
480	61.4%	28.5%	6.9%	3.2%
1200	60.7%	28.9%	7.6%	2.8%

Table 4.2: Percentage of hits per σ interval with path length 10

In this case, a general better fitting of the estimated curves, as suggested in Figure 4.3, is visible. It has been already argued that longer path are expected to produce more accurate results. However, the curve with $\lambda = 60$ draws the attention both for its accuracy with respect to the shape as well as for appearing to be shifted to the right. In Figure 4.7 this shift clearly shows a constant underestimation of the end-to-end delay (i.e. pessimistic predictions).

The explanation for this effect lies on the higher amount of missed acknowledgments for this experiment. When an acknowledgment is missed, the sender considers that the message was not received, and hence proceeds with its retransmission. However, the message was properly delivered and the receiver is ready to forward it further away. The result is that the calculated latency of the message at the sender node is notably worse than the real delay experienced by the message. Such phenomenon are expected to happen in WSN, and this result shows that measures must be taken to countermeasure its effects.

Table 4.1 and Table 4.2 present the results for the second test with the reference to the standard Normal in brackets. Again, the tail effect can be seen as the interval I_4 receives significantly more hits than expected. Similarly, interval I_1 reflects a lower percentage of hits, which agrees with the previous figures.

Chapter 5

conclusions

5.1 Final remarks

In this paper we presented a new approach to obtain a probabilistic timeliness performance metric in Wireless Sensor Networks. A statistic analysis of low computational requirements is applied at run-time on each node to analyze its message forwarding latency. This information is used to construct a metric which reflects the estimation of the probability density function of the end-to-end latency of a routing path. Simulations results for a set of different scenarios underline the accuracy of this method.
The paper motivates the use of probabilistic approaches instead of methods aiming at hard real-time by means of adding constraints and hence reducing its applicability.

5.2 Future work

Future work in this area includes the consideration of global energy consumption (i.e. energy-timeliness trade-offs), study of possible node adjustments to achieve local improvements of metric values (e.g. back-off exponents, size of preambles, etc) as well as the application of the probabilistic timeliness metric in network protocols.

Bibliography

[1] T. Abdelzaher, S. Prabh, and R. Kiran. On real-time capacity limits of multihop wireless sensor networks. In *Proceedings of the IEEE International Real-Time Symposium (RTSS)*, 2004.

[2] M. G. Bulmer. *Principles of Statistics*. 1979 edition, 1967.

[3] L. Burgstahler and M. Neubauer. New modification of the exponential moving average algorithm for bandwidth estimation. *ITC Specialist Seminar*, 2002.

[4] M. Caccamo, L. Zhang, L. Sha, and G. Buttazzo. An implicit prioritized access protocol for wireless sensor networks. In *23rd IEEE Real-Time Systems Symposium (RTSS)*, 2002.

[5] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher. Real-time power-aware routing in sensor networks. In *14th IEEE International Workshop on Quality of Service (IWQoS).*, June 2006.

[6] J.-D. Decotignie. Keynote: Real-time and wireless sensor networks: Why do we need another view at it? In *7th International Workshop on RealTime Networks (RTN) (abstract)*, pages 6–7, July 2008.

[7] A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Ninth IEEE Symposium on Computers and Communication (ISCC)*, June 2004.

[8] S. Gobriel, R. Cleric, and D. Mosse. Adaptations of TDMA scheduling for wireless sensor networks. In *7th International Workshop on Real-Time Networks (RTN)*, 2008.

[9] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. In *Proceedings of ICDCS'03*, May 2003.

[10] K. Karenos and V. Kalogeraki. Real-time trafc management in sensor networks. In *Proceedings of the 27th IEEE Real-Time Systems Symposium (RTSS)*, 2006.

[11] T. Korkmaz and M. Krunz. Bandwidth-delay constrained path selection under inaccurate state information. In *IEEE/ACM Transactions on Networking*, June 2003.

[12] L. Rizvanovic and G. Fohler. The MATRIX - a framework for real-time resource management for video streaming in networks of heterogenous devices. In *The In-*

ternational Conference on Consumer Electronics 2007, Las Vegas, USA, January 2007.

[13] A. Sahoo and P. Baronia. An energy efficient MAC in wireless sensor networks to provide delay guarantee. In *15th IEEE Workshop on Local and Metropolitan Area Networks. LANMAN*, June 2007.

[14] R. Serna Oliver and G. Fohler. Probabilistic routing for wireless sensor networks. In *Work-in-Progress Session, 29th IEEE Real-Time Systems Symposium*, December 2008.

[15] A. Varga. OMNeT++ discrete event simulation environment. url: http://www.omnetpp.org.

[16] T. Watteyne and I. Auge-Blum. Proposition of a hard real-time MAC protocol for wireless sensor networks. In *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE Computer Society, 2005.

[17] Y. Xiao, K. Thulasiraman, and G. Xue. Approximation and heuristic algorithms for delay constrained path selection under inaccurate state information. In *International Conference on Quality of Service in Heterogeneous Wired/Wireless Netowrks (QSHINE)*, 2004.

[18] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *ACM Conference on Mobile Systems, Applications, and Services (Mobisys)*, pages 125–138, June 2004.