A Performance Monitoring Method for Wireless Sensor Networks

Thomas Lindh
School of Technology and Health
Royal Institute of Technology, KTH
Thomas.Lindh@sth.kth.se

Ibrahim Orhan
School of Technology and Health
Royal Institute of Technology, KTH
Ibrahim.Orhan@sth.kth.se

António Gonga
School of Technology and Health
Royal Institute of Technology, KTH
Gonga@kth.se

ABSTRACT

This paper presents a monitoring method and its implementation as a light-weight end-to-end performance meter for quality-demanding applications in wireless sensor networks. The use of performance feedback information for control and management is also considered. The method is evaluated in a wireless sensor network testbed for healthcare applications.

Categories and Subject Descriptors

C.2 [**Computer-communication networks**]:

General Terms

Measurement, Performance.

Keywords

Wireless sensor networks, monitoring, measurements, performance meter.

1. INTRODUCTION

Wireless sensor networks are rapidly becoming a common infrastructure for a number of quality-demanding applications. It is crucial that estimates of e.g. medical and health-related parameters, transmitted via wireless body sensor networks, can be trusted. Correct information about the network performance parameters is necessary to be able to determine the uncertainty of the estimated vital sign parameters. This paper presents a measurement method that provides online transmission quality feedback to the applications. Performance measurements and monitoring in body sensor networks have been studied previously, e.g. in [1] and [2].

2. AN END-TO-END MONITORING METHOD

The method is based on the principle of end-to-end monitoring between the nodes where the applications run. A demanding application should be able to include the monitoring function as an add-on service. It is also important to minimize extra energy consumption.

2.1 Applications

Some typical applications that can use the proposed monitoring method are: in-home care, special housing and assistant living, hospital care, wellness and disease monitoring before and after hospital care, emergency care, fitness, sports and athlete training. Today, there exist a number of different sensors that can be attached to fixed or mobile wireless nodes, e.g. pulse-oximeters, ECGs, EEGs, blood pressure meters, glucose meters, thermometers, EMGs, bio-chemical sensors, accelerometers, gyroscopes and magnetometers, microphones, cameras, IR detectors, RFID tags etc. The monitoring method has been implemented in nesC and Java for an IEEE 802.15.4 low-power platform. Healthcare applications running in cell phones or PDAs communicating with wireless sensor nodes on the one hand, and back-end servers on the other hand, can also use the proposed method.

2.2 The Performance Meter

The approach is to combine active and passive techniques, inspired by the results from measurements in wireline networks in [3] and [4]. A light-weight performance meter is implemented in each node. The meter consists of two counters that keep track of the number of sent and received packets and bytes, and a function that can insert monitoring packets. These dedicated measurement packets are inserted between blocks of ordinary data packets as seen in Fig. 1. They contain a sequence number, a timestamp and the cumulative number of packets and bytes transmitted from the sending node to the receiving node.

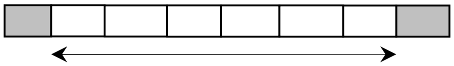

Figure 1. A monitoring block surrounded by two monitoring packets.

The interval between the monitoring packets, i.e. the size of the monitoring block, can be expressed in number of packets or a time interval, constant or varying randomly around a mean value. When a monitoring packet arrives, the receiving node stores a timestamp and the current cumulative counter values of the number of received packets and bytes from the sending node. Observe that for *k* sending nodes, the receiving node maintains *k* number of separate monitoring functions, one for each sending node.

Synchronization of the clocks in the participating nodes is not required. The local timestamps are used to calculate the inter-sending and inter-arrival times between pairs of monitoring packets. The inter-arrival jitter can then be calculated in a similar way as for RTP timestamps [5]. This means that arrival time variation is estimated based on the monitoring packets, which represent samples of the data packets' inter-arrival variation. Packet loss, on the other hand, is measured passively and directly using the counters.

Conference'04, Month 1–2, 2004, City, State, Country.

2.3 Performance Metrics

The following metrics can be calculated and estimated based on the collected measurements described in the previous subsection.

- Packet loss ratio: long-term and per monitoring block.
- The length of loss and loss-free periods defined as the number of consecutive monitoring blocks with or without losses. Can be expressed in time unit, number of blocks, and number of packets and bytes.
- Inter-arrival jitter, *J, is* defined as $J = (r_n-r_{n-1})-(s_n-s_{n-1})$, where *s* is the sending time and *r* is the receiving time. The monitoring packets provide samples of this delay variation metric, which means that the uncertainty of the estimated statistics (mean value, median, percentiles etc) is determined by the number of samples and the variance of the delay variation process.
- Throughput between sender and receiver: long-term average and also per monitoring block. The resolution of the peak rate is determined by the ratio between monitoring packets and ordinary data packets. This can also be seen as a measure of utilized capacity.

3. MEASUREMENTS

3.1 The Testbed and Measurement Scenarios

The testbed consists of various sensors attached to the IEEE 802.15.4 [6] compliant Tmote Sky motes [7]. Another Tmote Sky mote, connected to a laptop, acts as a base station. Fig. 2 and Fig. 3 show two different measurement scenarios. In the first case the sensor nodes are attached to the base station in a star topology.

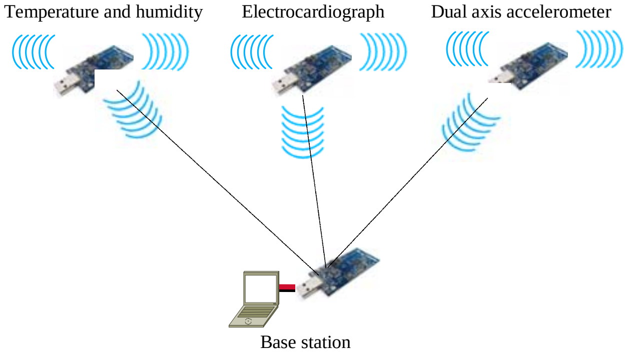

Figure 2. The measurement scenario with three sensor nodes and a base station.

In the second case (Fig. 3) a sensor node that is two hops away from the base station will be compared to being directly connected. An intermediate node forwards the packets from the sensor node to the base station. The buffer size is 20 packets. In both scenarios samples of sensor data are sent from the sensor node to the base station.

3.1.1 The Temperature and Humidity Sensors

The onboard temperature and humidity sensors are sampled twice a second and the collected samples are sent immediately to the base station. Monitoring packets are inserted between blocks of 100 data packets (each packet has 6 byte payload).

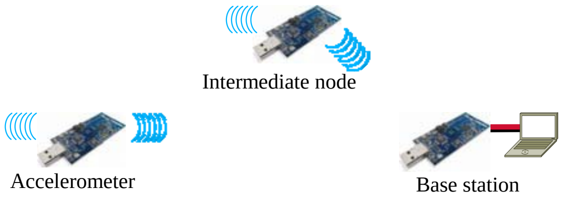

Figure 3. The measurement scenario with two hops between the sensor and base station.

3.1.2 The Temperature and Humidity Sensors

The onboard temperature and humidity sensors are sampled twice a second and the collected samples are sent immediately to the base station. Monitoring packets are inserted between blocks of 100 data packets (each packet has 6 byte payload).

3.1.3 The ECG Sensor

The mote reads samples at 200 Hz from the ADC-12 (analog-to-digital converters, 12 bits resolution), that is connected to an ECG. Samples are collected during five seconds. The radio, previously switched off, is now turned on, and the samples are sent to the base station. The procedure is then repeated. In total 13 samples are sent in every packet. The idea is to have the radio turned off as long as possible, sending several samples in each packet, in order to minimize the power consumption. In this case the mote transmits 77 packets back-to-back every five second. A monitoring packet is inserted between blocks of approximately 100 ordinary data packets

3.1.4 The Dual-Axis Accelerometer Sensor

A multi-sensor board (SBT80 from Easysen [8]) with a dual-axis accelerometer sensor is connected to two ADCs, one ADC to each axis. The node samples the accelerometer sensor at 100Hz and sends all samples to the base station. The radio is only turned on when sending the samples. The mote sends 20 packets per second, where each packet carries 10 samples, five samples from each axis. A monitoring packet is sent after every 200 data packets, i.e. with around 10 seconds' interval.

3.2 The Meter and Monitoring Packet Implementation

The performance meter is programmed in nesC for TinyOS 2.0. The sensor nodes read samples from the sensors (ECG, accelerometer and temp/humidity), assemble the samples and send them in packets to the base station. The number of bytes and packets are counted. The cumulative number of bytes and packet and a timestamp are inserted into a monitoring packet, which is sent after every *n* ordinary data packet.

A monitoring packet is 17 bytes long and includes the following fields: a start flag, a timestamp when packet is sent, type, a sequence number, number of packets sent, number of bytes sent, and a stop flag. The flags enable the base station to distinguish a monitoring packet from ordinary data packets. The sequence number identifies and keeps track of monitoring packets. The packet and byte fields contain the cumulative number of bytes and packets sent. Finally, the type field enables measuring several sensor data flows from the same node.

Each time the base station receives a data packet, it updates the number of bytes and packets for each sensor. The base station uses the source field in the CC2420 radio header to distinguish the packets from different sources. As soon as the base station receives a monitoring packet, it stores a timestamp and the updated counter values of the number and bytes and packets received from the specific sensor mote. A Java application on the laptop processes the data and creates two files for each sensor.

The file with data for the sending node (to the left in Fig. 4) contains the same information as carried in the monitoring packet: the sequence number, the timestamp, the total number of bytes sent, and the total number of packets sent. The file with measurement data from the receiving node contains the corresponding information at the base station side (to the right in Fig. 4.

```
70 623766 312400 14200     70 629597 312224 14192
71 632514 316800 14400     71 638347 316624 14392
72 641339 321200 14600     72 647176 321024 14592
73 650109 325600 14800     73 655949 325424 14792
```

Sender   Receiver

Figure 4. Measurement data from the sender and the receiver nodes. Columns from left to right: sequence number, timestamp (ms), cumulative number of bytes and packets.

4. RESULTS

In this section we show some results using the performance meter from the two scenarios in Fig. 2 and Fig. 3.

4.1 Loss Periods and Loss-Free Periods

The loss ratio per monitoring block during the measurement period for the accelerometer data is illustrated in Fig. 5. The distinct loss events in the beginning of the measurement period are caused by interferences in the radio communication. Table 1 shows the loss ratio per monitoring block and mean length of loss periods and loss-free periods for the three wireless links.

TABLE 1: PACKET LOSS BETWEEN SENSOR NODES AND BASE STATION

	Acc–Base stn	ECG–Base stn	Temp-Base stn
Mean loss ratio	3.8%	0.2%	0.6%
Max loss ratio	93.5%	4.0%	10.0%
Min loss ratio	0.0%	0.0%	0.0%
Loss period mean length (s)	37s	6s	11s
Loss-free period mean length(s)	15s	40s	165s

The loss ratio is between 0.8 and 0.9 during the three loss periods (Fig. 5). The length of the loss-periods (consecutive monitoring blocks that contain at least one lost packet) is shown in Fig. 6.

4.2 Inter-Arrival Delay Variation

Table 2 shows some statistics of the inter-arrival delay variation (jitter) for the scenario in Fig. 3 (in Section 3) two hops between the sensor and the base station compared to only one hop. The sensor node transmits 20 packets per second. The radio communication is not exposed to severe disturbances in this case. The time plots in Fig. 7 and Fig. 8 (on next page) show that the inter-arrival jitter is several times higher with an intermediate node between the sensor and the base station then without a middle node. The packet loss for sensing data via two hops instead of one hop is also much higher. The high levels of inter-arrival jitter and packet loss in the two-hop case is due to the receiving and forwarding performance of the intermediate node.

TABLE 2: INTER-ARRIVAL JITTER (MS)

Inter-arrival jitter (ms)	One hop	Two hops
Maximum	13	59
Minimum	0	0
Standard deviation	4.0	12.5

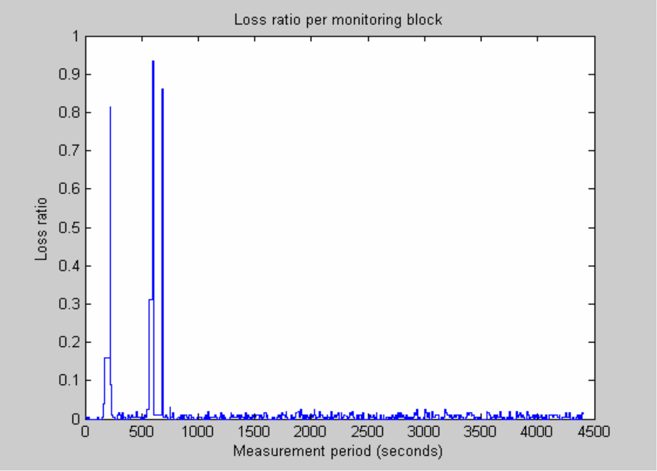

Figure 5. The loss ratio per monitoring block for accelerometer data.

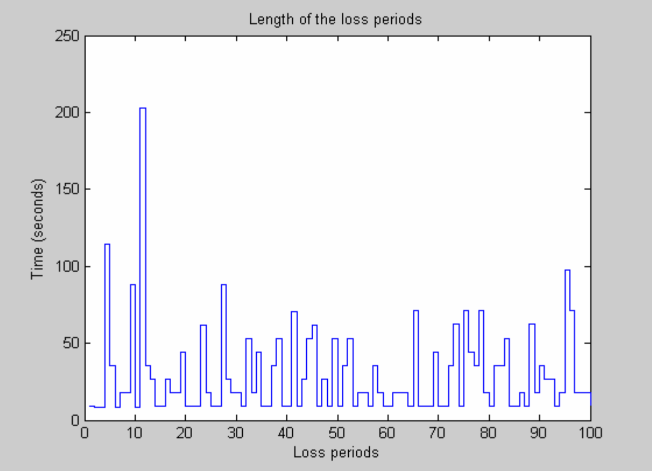

Figure 6. The length of loss periods during the measurement period in seconds on the y-axis and loss period number on the x-axis.

5. DISCUSSION

5.1 Uncertainty in Parameter Estimations

The result from the performance meter can be used to calculate the statistical uncertainty of the parameter estimates based on samples of sensor data. Table 3 shows some effects on estimation of e.g. vital sign parameters due to network performance deterioration. A loss period from one of the loss bursts in Fig. 6 and Fig.

7 is chosen as an illustration. The entire loss period consists of 20 monitoring blocks and lasts for 200 seconds.

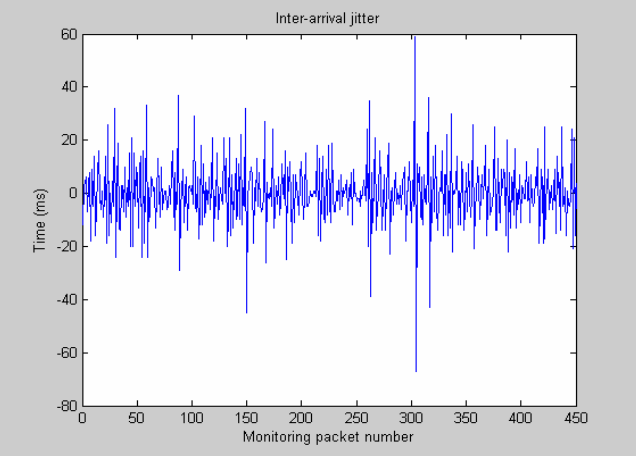

Figure 7. Inter-arrival jitter for the two-hop case.

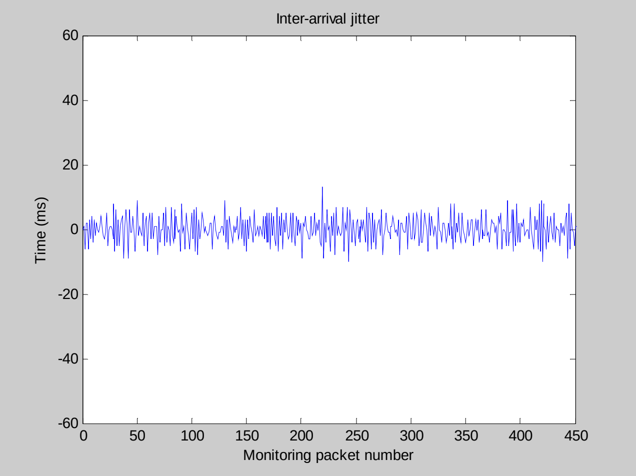

Figure 8. Inter-arrival jitter for the one-hop case.

Normally a monitoring block is 10 seconds, but in some cases a block can be several times longer, e.g. the second, third and fifth row in Table 3. The reason is that monitoring packets as well as data packets may disappear before they arrive at the destination node during a loss period. If one or several monitoring packets in a row are lost, the original monitoring blocks are merged into a larger block. Row 5 in Table 3 is a concatenation of 8 original blocks, where 7 monitoring packets were lost.

The loss ratio stretches from 0.5 % to 93.5 %. The deteriorated quality of a mean value estimate during the loss period is shown in the third column. We assume the standard deviation to be 4 units and a confidence level of 0.99 for the mean value estimate with the same resolution as the monitoring block size (10 seconds). The resulting confidence interval for an ideal communication channel without losses will be 0.92. A loss ratio of 31.3% (second row) leads to a confidence interval of 1.11, and to 3.62 for a loss ratio of 93.5% (during 40 seconds). For a sampling rate of 100 Hz the highest frequency component that can be recovered by the receiver on an ideal channel is 50 Hz (the sampling theorem). In this case the actual highest frequency component in the received signal is as low as 3 Hz during the 36 seconds long period with a loss ratio of 93.5%. Formula 1 below is used for calculating the confidence interval (*d*), where *z*=2.58 (for 0.99 confidence level), the estimated standard deviation *s*=4, and *n* is the number of samples from measurement data).

$$n = \frac{z^2 \cdot s^2}{(d/2)^2} \tag{1}$$

TABLE 3: UNCERTAINTY IN PARAMETER ESTIMATION IN LOSS PERIODS

Monitoring block duration (s)	Loss ratio	Conf. interval (0.99 level, stdev=4)	Highest frequency component in received signal
10 s	2.5%	0.93	50 Hz
20 s	31.3%	1.11	35 Hz
40 s	93.5%	3.62	3 Hz
10 s	1.0%	0.93	50 Hz
80 s	86.1%	2.47	18 Hz
10 s	3.0%	0.94	50 Hz
10s	0.5%	0.93	50 Hz
10	0.5%	0.93	50 Hz
10	1.0%	0.93	50 Hz

6. CONCLUSIONS

This paper presents a monitoring method and its implementation as a light-weight end-to-end performance meter for quality-demanding applications in wireless sensor networks. The measurement tests have shown promising results. The performance meter results are used to calculate the increased uncertainty in parameter estimates due to network performance deterioration during loss periods. Feedback information as input for real-time control has also been considered. Future work will focus on further improvement of the performance of the meter, analysis of traffic in wireless sensor networks, and to close the loop between monitoring and control based on the feedback information from the meter.

7. REFERENCES

[1] D. Cavalcanti et al.: "Performance Analysis of 802.14.4 and 802.11e for Body Sensor Network applications", BSN, Aachen. 2007.

[2] N. Golmie et al: "Performance analysis of low rate wireless technologies for medical applications" Computer Communications, June 2005.

[3] T. Lindh and N. Brownlee: "Integrating Active Methods and Flow Meters - an implementation using NeTraMet", Passive and Active Measurement workshop (PAM2003), San Diego, April 2003.

[4] M. Brenning, B. Olander, I. Orhan, J. Wennberg, T. Lindh: "NeTraWeb: a Web-Based Traffic Flow Performance Meter", SNCNW2006, Luleå, Sweden, 2006.

[5] "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, H. Schultzrinne et al., July 2003.

[6] IEEE Standard 802.15.4 - 2006.

[7] Tmote Sky – IEEE 802.15.4 compliant sensor module from Sentilla (previously Moteiv).

[8] STB80 – Multi-Modality Sensor Board for TelosB Mote.