Performance Monitoring and Control in Contention-Based Wireless Sensor Networks

Thomas Lindh #1, Ibrahim Orhan #2

# School of Technology and Health, KTH – Royal Institute of Technology
Marinens vag 30, 136 40 Haninge, Sweden

1 Thomas.Lindh@sth.kth.se
2 Ibrahim.Orhan@sth.kth.se

Abstract— This paper presents a method for performance monitoring and control in wireless body sensor networks based on measurement feedback. Test results using a prototype implementation of the method are also analyzed. The method has been evaluated for demanding healthcare related applications in wireless personal area networks.

I. INTRODUCTION

Wireless sensor networks are today being considered for a wide range of demanding applications. One example is wireless body sensor networks that enable continuous monitoring of patients’ vital signs parameters in everyday life situations. However, to transmit healthcare related parameters in wireless networks is also a challenge; especially if contention-based access is used. Recipients of data sent in wireless sensor networks need to know whether they can trust the information or not. To address this problem we have developed a performance meter that can measure the performance, and furthermore, to feed a performance control system with real-time measurement data. In Section II we put the results in the context of previous related work. Section III presents the approach and methods for monitoring, feedback and control. Section IV describes the implementation. Section V and Section VI contain the results from test cases.

II. RELATED WORK

Measurements, simulations and theoretical studies show that the loss ratio increases with the traffic load and number of sending nodes. Bianchi [1] has derived an analytical Markov chain model for saturated networks, further developed in [2] and extended to non-saturated networks in [3]. Channel errors e.g. due to external disturbances and obstacles in the environment, can of course increase the loss ratio further. Another related problem, studied in [4] is the reduced throughput in multi-hop networks, with one or more intermediate nodes between sender and receiver. Dunkels and Österlind [4] found that the implementation of packet copying in an intermediate forwarding node has significant impact on the throughput.

Performance in LR-WPAN has been analysed in several studies, often based on simulations ([5],[6]). A performance meter that keeps track of losses, inter-arrival jitter and throughput was presented at BSN 2008 [7]. Several papers have also addressed congestion and rate control in WLAN and LR-WPAN. CODA (congestion detection and avoidance in sensor networks) is a control scheme that uses an open-loop backpressure mechanism as well as a closed-loop control, where a sink node can regulate a source node’s sending rate by varying the rate of acknowledgements sent to the source [8]. CARA (collision-aware rate adaptation) uses the RTS packets in IEEE 802.11 as probes to determine whether losses are caused by collisions (related to CSMA/CA) or by channel errors [9].

III. PERFORMANCE MONITORING AND CONTROL

Fig. 1 shows a configuration with a set of sensor nodes (e.g. a combination of wearable sensors such as ECGs, accelerometers and pulse-oximeters, and fixed environment sensor nodes), a coordinator, one or several intermediate nodes with routing and forwarding capabilities. The application program, running in the coordinator, processes sensor data from the sources and sends the information along with an estimate of the information quality to the remote end-user application or presentation and storage. The information quality can be expressed in terms of e.g. the statistical uncertainty of estimated parameters and the highest frequency component in a signal to be recovered by the receiver.

![Fig. 1. A scenario where performance control is implemented in the coordinator and source nodes.](image-url)

The system presented in this paper consists of two main parts, a performance meter and a performance manager, described in more detail in the following sections. The performance monitoring and control capabilities can be implemented as an add-on capability to be used between applications running in the communicating endpoints, e.g. sensor nodes and a coordinator, and not link by link. The ambition has also been to minimize the traffic overhead and energy consumption. The
system is targeted to wireless sensor networks that use contention-based access, but can of course also be used in combination with contention-free access, such as guaranteed time slots. The applications, e.g. streaming data from accelerometers and ECGs, require certain levels of throughput and low loss ratio, however not necessarily zero.

The aim is, firstly, to provide quality estimates of the transmitted parameters, and secondly, to reuse this information and enable performance control that minimizes information loss and maintains the desired throughput. This closes the loop between measurements and control.

A. Performance Meter

The performance meter (presented in [7] and inspired by [10]) combines active and passive measurement techniques. It is based on so called monitoring blocks (Fig. 2). The accuracy and resolution of the measurement results is determined by the size of the monitoring block. The meter is implemented in the source nodes and the coordinator. It 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 measurement packets are inserted between blocks of ordinary data packets as seen in Fig. 2.

![Fig. 2 A monitoring block surrounded by two monitoring packets.](image)

They contain a sequence number, a timestamp and the cumulative number of packets and bytes transmitted from the sending node to the receiving node. 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. When a monitoring packet arrives, the receiving node will store a timestamp and the cumulative counter values of the number of received packets and bytes from the sending node.

The following metrics can be calculated and estimated based on the collected measurements: packet loss, inter-arrival jitter and throughput. More detailed information and test results are presented in [7].

B. Measurement-Based Performance Control

The performance manager, implemented in a coordinator node, bases its decisions on the feedback information it receives from the meter, in this case mainly packet loss and throughput. The meter delivers the performance updates for each incoming block of data packets, e.g. 100 packets, from a sensor to the coordinator. The output of the performance manager can e.g. be to increase or decrease the packet frequency, change the transmission power, enable or disable acknowledgment etc. In this study the control actions are limited to varying the packet frequency. Some examples of possible control algorithms are described in Section III-C.

In summary, the monitoring and control method has three main parameters, explained in the coming sections, that can be tuned: the size of the monitoring block (\(B\)); the number of previous monitoring blocks (\(B_n, B_{n-1}, B_{n-2}\) etc), and their relative weight, that the control algorithm is based on; and, the step size (\(\Delta t\)) that controls the time interval between transmitted packets (or packet frequency).

C. Feedback Control Algorithms

The output of the control algorithm, to decrease or increase the packet frequency, is based on performance data from the current and previous monitoring blocks. The loss ratio and throughput (received bits per second) for a number of the recently received monitoring blocks is kept in memory. The manager sends a request message to a sensor node to either reduce or increase the packet frequency by adding (or subtracting) \(\Delta t\) milliseconds to (or from) the time interval between the transmitted packets. The step size, \(\Delta t\), is determined by a weighted average of the performance feedback from the current monitoring block and \(m\) previous blocks.

Results from test cases with three simple control algorithms are presented in Section V; the first to protect performance of high priority nodes (Section V-A), the second where all nodes have the same priority (Section V-B), and finally a case where these algorithms are combined and priority is assigned dynamically to one node (Section V-C).

D. Test cases

The purpose of the first control algorithm (Section V-A) is to maintain throughput and minimize losses for a node with high priority, by punishing nodes with low priority. The algorithm works like this. The manager will keep the throughput between a maximum and minimum level. If the throughput drops below the minimum level, the performance manager tells the node to decrease the packet interval by \(\Delta t\) milliseconds. If the throughput rises above the maximum level, the performance manager increases the packet interval by \(\Delta t\) milliseconds.

When monitoring block \(B_n\) has arrived at the coordinator, a weighted average of the throughput and the loss ratio based on block \(B_m\), \(B_{n-1}\), and \(B_{n-2}\) are computed. If the loss ratio for the high-priority node is above the threshold, the coordinator instructs the low priority nodes to increase the packet interval by \(\Delta t\) milliseconds. This is repeated until the loss ratio for the prioritized node is below the threshold.

One reason for a drop in throughput is the contention-based access mechanism. The total time interval between two transmitted packets at the sender side is \(t_{\text{ACCESS}} + t_{\text{DELAY}} + t_{\text{TRANS}}\), where \(t_{\text{ACCESS}}\) is the waiting time due to the access method CSMA/CA, \(t_{\text{DELAY}}\) is the interval between the time the transmission of packet \(n\) is completed and the time the transmission of packet \(n+1\) begins, and \(t_{\text{TRANS}}\) is the frame transmission time (the frame size divided by the bit rate, 250kb/s in this case). \(t_{\text{DELAY}}\) is increased or reduced by the \(\Delta t\) control requests described above. The access time varies depending on how many nodes that are trying to transmit in the same channel. An increase in \(t_{\text{ACCESS}}\) will lower the packet frequency and throughput.

In the second case, where all nodes have the same priority (Section V-B), each node tries to maximize its throughput under the condition that the loss ratio is below a threshold. The third case (Section V-C) is a combination of the two previous ones. From the beginning both nodes have the same
priority. After a certain time, one of the nodes is assigned high priority and higher expected throughput.

IV. SYSTEM IMPLEMENTATION

The testbed used in this work consists of TmoteSky sensor nodes running TinyOS 2.1.0 programmed in nesC. The radio (CC2420) and link layer are compliant with IEEE 802.15.4 LR-WPAN in contention-based access mode. The software system consists of two parts, the performance meter and the performance manager. The meter stores performance data, as described in Section III-A, for each block of received packets (the monitoring block size). This monitoring data is used in two ways: firstly, to estimate the information quality of transmitted sensor data, and secondly, to feed the performance manager function with information for control decision. The performance meter is 60 lines of nesC code in the coordinator and 25 lines of code in a sensor node. The performance manager part is implemented as 65 lines in the coordinator and 5 lines of code in a sensor node.

V. TEST RESULTS

Fig. 3 shows a test scenario with two sensor nodes that are streaming ECG samples and accelerometer samples to the coordinator through a forwarding intermediate node. Results from two test cases with different control algorithms are presented in the following sections.

A. Control Scheme with Priority

The control algorithm in our test case means that one of the sensor nodes has high priority and the other one has low priority. The loss ratio threshold is computed as a weighted average of the three recent consecutive monitoring blocks and compared to the threshold 0.02. The required bit rate is 8kb/s, which corresponds to approximately 250Hz sampling rate per axis for a two-axis accelerometer or a 500Hz ECG.

Fig. 4 - Fig. 7 illustrate how the implemented algorithm works in practice. The high priority node starts from 10kb/s and slows down to the expected bit rate 8kb/s (Fig. 4). The second node is turned on shortly thereafter (t ≈ 80s) at a rate of nearly 16kb/s (Fig. 5). The received bit rate from the high priority node falls sharply (Fig. 4). The solid lines (blue) show the received bit rate measured at the coordinator. The dotted lines (red) represent the sending bit rate from the sensor node.

The loss ratio for the high priority node peaks at nearly 0.45 (Fig. 6) when the second node starts transmitting. The loss ratio for the low-priority nodes is shown in Fig. 7.

The performance manager reads the performance data provided by the meter for each block of incoming data packets. The monitoring block size is 100 packets in this test case. As soon as the manager detects the increased loss ratio for the high-priority node, it will instruct the other node to slow down. The low priority node will directly decrease the transmitting rate (Fig. 5), which results in lower loss ratio (Fig. 6) and higher throughput (Fig. 4) for the prioritized node. As the loss ratio approaches the threshold, the sending rate of the low priority node stabilizes around 3kb/s (Fig. 5). The performance manager strives to maintain the desired throughput (8kb/s) for the high-priority during the remaining part of the test, with an average loss ratio below the threshold.
Control Scheme without Priority

In this test case a priority scheme is not used. Both sensor nodes are controlled independently by the performance manager under the condition that the loss ratio is below a threshold. If the loss ratio is above the threshold, the sensor node will be instructed to decrease the sending rate (increase the packet interval by $\Delta t$ milliseconds). No expected throughput is specified.

Both sensor nodes start sending at 18kb/s as seen in Fig. 8 and Fig. 9. The high loss ratio for both nodes means that the performance manager will order both of them to slow down until the losses fall below the threshold. It can also be observed that the sensor node sometimes maintains the sending rate, even though the loss ratio is significantly higher than the threshold (Fig. 8 and Fig. 10). The explanation is that during heavy loss monitoring packets will also be lost, which delays the decision to decrease the packet frequency.

After a while, the first node’s throughput stabilizes around 3kb/s (Fig. 8) and around 3.5kb/s for the second node (Fig. 9). Since the control of the sensor nodes is independent of each other, the throughput will normally not be on the same level. One reason is different loss characteristics of the two channels; another may be different starting values. Each sensor tries to find its maximum bit rate without exceeding the loss ratio threshold.

At approximately $t=180s$ the manager has observed that the recent monitoring blocks are loss-free. The packet frequency is therefore increased for node 2 (Fig. 9). At $t=210s$, sensor node 2 stops transmitting (Fig. 9), which results in approximately zero packet loss for sensor node 1 (Fig. 10). The manager therefore tells the node to increase the packet frequency, up to around 10kb/s, where the loss threshold forces the node to slow down (Fig. 8).

Dynamic Priority Assignment

Fig. 11 and Fig 12 show a combination of the previous two control algorithms. Both nodes start at bit rate just below 15kb/s with the upper limit loss ratio 0.02. No node is given priority over the other. The throughput stabilizes between 4kb/s and 5kb/s. At $t=300$ seconds, one of the nodes (Fig. 11) is dynamically assigned high priority, while the other node has to be satisfied with what is left. The reason might be that a higher sampling rate is needed for a sensor.

The bit rate for the high-priority node rises to the required 8kb/s (Fig. 11) and the other sensor node backs off to around 2.5 kb/s (Fig. 12). The step response in Fig. 11 takes around 30s. This time period can be reduced either by allowing larger step sizes ($\Delta t$) or decreasing the interval between the monitoring packets.)
Fig. 11 The sensor node is assigned high priority at t=300s and raises the bit rate to 8kb/s.

Fig. 12 The sensor node is assigned low priority at t=300s and reduces the bit rate to 2.5kb/s.

D. Multi-Hop Cases

The bit rate from a sensor to a coordinator will to a large extent depend on the number of hops between the source and destination [4]. The maximum received throughput for the equipment in our testbed (Section IV) using maximum packet length (payload 112 byte) was 50kb/s for one hop, 35kb/s for two hops and 20kb/s for three hops. This is of course a crucial limitation for demanding applications.

VI. EVALUATION

This study shows that it is feasible to use the measurement method, based on monitoring blocks, for performance monitoring as well as for feedback control of the performance of applications in sensor networks. The result using elementary priority control algorithms was promising. The method has been implemented in a network with contention-based (CSMA/CA) access. It can of course also be used for the contention-based part of a super-frame in beacon mode in IEEE 802.15.4, where the contention-free part has guaranteed time-slots for the most demanding applications.

One observation is that avoiding packet loss in situations of buffer saturation by reducing the packet frequency is more straightforward than to handle packet loss due to collisions and possible channel errors.

The size of the monitoring block is an important parameter for the resolution of the performance metrics as well as for the responsiveness of the control function. It deserves a more detailed study. A problem related to control theory is the design of the control algorithm with respect to current and previous estimates of performance data.

To find out, in real-time, what capacity is available for a specified loss ratio, given that a second node transmits at 2kb/s, is another application of the control method.

VII. CONCLUSIONS

The presented control method and the prototype implementation can be used to provide quality of service control of applications in wireless sensor networks using contention-based access. A performance meter continuously feeds the performance control function with quality estimates of the transmitted sensor data. The tests results of the implemented algorithms were promising.

REFERENCES