Performance Control in Wireless Sensor Networks

Thomas Lindh
School of Technology and Health
KTH
Sweden
Thomas.Lindh@sth.kth.se

Ibrahim Orhan
School of Technology and Health
KTH
Sweden
Ibrahim.Orhan@sth.kth.se

Abstract— **This paper presents an implementation of a method for performance control in wireless body sensor networks based on measurement feedback, especially targeted for demanding healthcare applications.**

Keywords-body wireless body sensor networks, performance monitoring, performance control, healthcare.

I. INTRODUCTION

Recipients and users of healthcare related parameters transmitted in wireless sensor networks need to know whether they can trust the information or not. The quality of the received information is decisive for a future deployment of wireless sensors for medical and healthcare applications. To address this problem we have developed a method for performance management in wireless sensor networks. 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 and Section V contains the results from a test case. The final sections summarize the findings and discuss future work. The main contribution of this work is a method for performance management that can be used by demanding healthcare applications in wireless sensor networks.

II. RELATED WORK

Wireless communication, and in particular wireless body sensor networks, has obvious advantages in healthcare applications. To continuously monitor a patient's vital signs parameters in real time in everyday life situations, is only one example of the new possibilities that wireless communication offers. However, to transmit healthcare related parameters in wireless networks is also a challenge, especially in multiple access wireless networking.

Low-rate Wireless Personal Area Networks (IEEE 802.15.4) uses CSMA/CA (carrier sense multiple access) for contention-based access. 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 analyzed 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 ([8],[9]).

Our approach is to use feedback quality information from a monitoring method as input to a performance manager that can control the sending rate of the sensor nodes according to a control algorithm, e.g. minimizing losses and maintaining throughput for prioritized sensor nodes.

III. PERFORMANCE MONITORING AND CONTROL

The main idea behind the system presented in this paper is that a coordinator node has the capability to control the performance of sensor data from source nodes, based on feedback information from a performance meter (see e.g. the test scenario in Fig. 2, Section V).

A. Aim and Applications

The performance monitoring and control method is 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 also be used in combination with contention-free access, e.g. guaranteed time slots.

The applications, e.g. streaming data from accelerometers and ECGs, require certain levels of throughput and low, however not necessarily zero, loss ratio. 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 requested throughput. This closes the loop between monitoring and control. We have found no need to develop a separate generic transport protocol for these applications.

B. *Performance Control*

The performance manager is implemented in a coordinator node. It receives information from a performance meter regarding packet loss, inter-arrival jitter and throughput for the ongoing communication between a coordinator and the connected sensor nodes. The meter delivers performance measurements for each incoming block of data packets, e.g. 100 packets, from a sensor to the coordinator. The size of these monitoring blocks determines the resolution and accuracy of the performance estimates, and the cost in terms of overhead and energy for the method. The block size also sets a limit for the response time for control actions, the time it takes for the manager to react to performance reports.

The performance control is primarily based on feedback information on packet loss and throughput from the respective source node. In this paper we focus on a case with two levels of priority. High priority means that the required throughput (received bits per second) should be maintained and the packet loss ratio be minimized (below a threshold) for the prioritized nodes, possibly at the expense of nodes with low priority. If the loss ratio for the high priority node is above the threshold, the manager will instruct the low priority sensors, to decrease their transmission rate step by step until the loss ratio for the high priority nodes is below the threshold. If the loss ratio still is above the threshold, the sending rate of the high priority nodes will be decreased as well, and eventually be turned off if the loss ratio remains too high.

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 should be based on, and the step size (Δt) that controls the time interval between transmitted packets (or packet frequency).

C. *Feedback from a Performance Meter*

The performance meter (presented in [7] and inspired by [10] and [11]) that feeds the performance manager with information, combines active and passive measurement techniques. A light-weight performance meter is implemented in the source nodes and the coordinator. A 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.

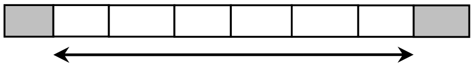

Figure 1. A monitoring block surrounded by two monitoring packets.

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. 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.

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.C, 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.

A. *Control Algorithm*

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) Δt milliseconds to (or from) the time interval between the transmitted packets. The step size, Δt, is determined by a weighted average of the performance feedback from the current monitoring block and m previous blocks.

The purpose of the control algorithm is to maintain the throughput between a maximum and minimum level, and minimize losses for a node with high priority by punishing nodes with low priority. The algorithm works like this.

The performance manager will keep the throughput of the high priority node between the maximum and minimum level. If the throughput drops below the minimum level, the performance manager asks the node to decrease the packet interval by Δt milliseconds. If the throughput rises above the maximum level, the performance manager asks the node to increase the packet interval by Δt milliseconds. When monitoring block B_n has arrived at the coordinator, a weighted average of the throughput and loss ratio based on block B_n, B_{n-1} and B_{n-2} are computed. If the average loss ratio for the high priority node is above the threshold, the coordinator instructs the low priority nodes to increase the packet interval by Δt milliseconds. This is repeated until the loss ratio for the prioritized node is below the threshold.

V. A Test Case

A. A Sensor Network with an Intermediate Node

Fig. 2 shows a test scenario with two sensor nodes that are streaming ECG samples and accelerometer samples to the coordinator through a forwarding intermediate node. The implementation details are described in Section IV.

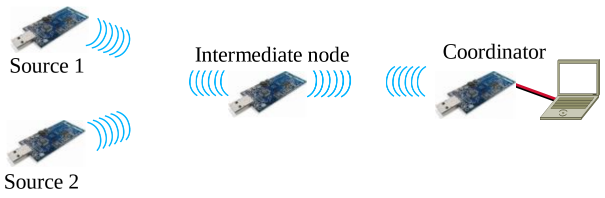

Figure 2. Two source nodes that are sending sensor data to a coordinator via an intermediate forwarding node. Source node 1 has high priority and source node 2 has low priority.

B. Control Scheme with Priority

The control algorithm in our test case means that sensor node1 has high priority and the other one has low priority. The loss ratio 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 per axis for a two-axis accelerometer or a 500Hz ECG.

Fig. 3 - Fig. 6 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. 3). The second node is turned on shortly thereafter (t≈80s) at a rate of nearly 16kb/s (Fig. 4). The received bit rate at the coordinator from the high priority node falls sharply (Fig. 3). 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. 5) when the second node starts transmitting. 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. 4), which results in lower loss ratio (Fig. 5) and increased throughput (Fig. 3) for the prioritized node. As the loss ratio drops and approaches the threshold, the sending rate of the low priority node stabilizes around 3kb/s (Fig. 4). 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.

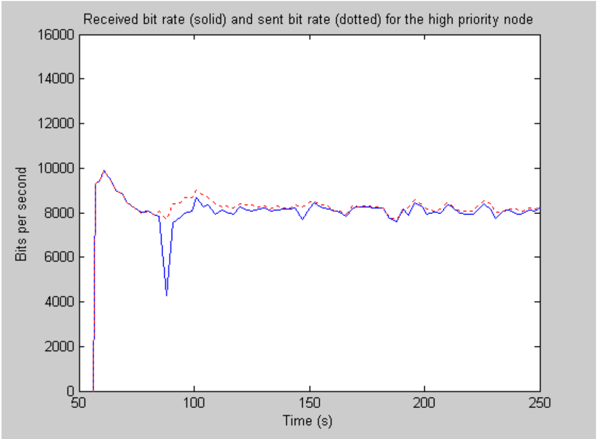

Figure 3. Throughput for the high priority node.

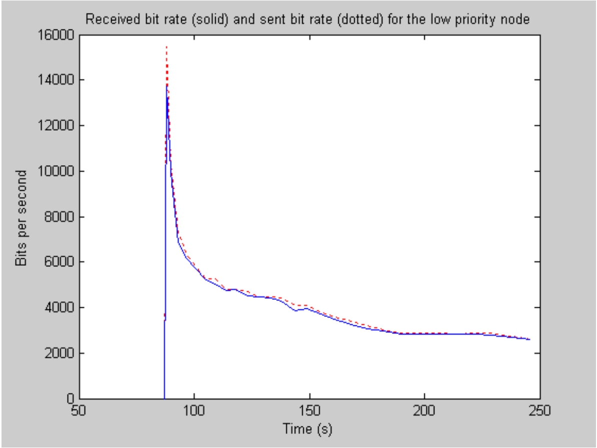

Figure 4. Throughput for the low priority node.

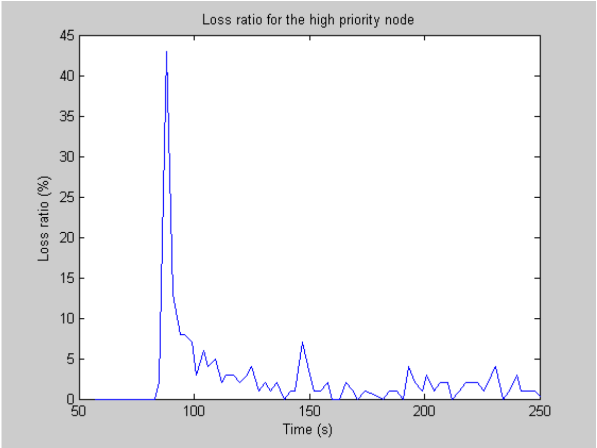

Figure 5. Loss ratio for the high priority node.

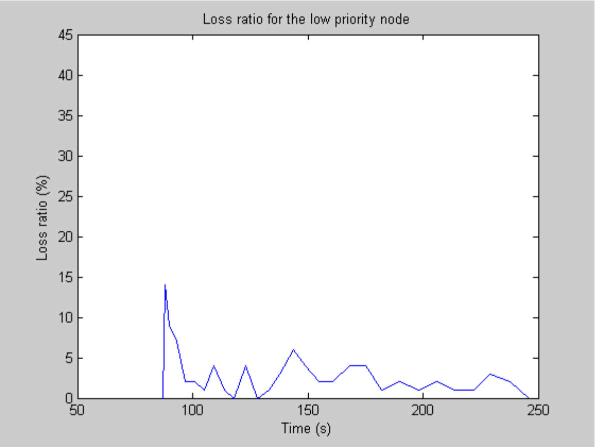

Figure 6. Loss ratio for the low-priority node.

VI. Results and Discussion

The implemented prototype shows that it is feasible to control the quality of service for applications in sensor networks with contention-based multiple access, based on feedback performance information. The control algorithm for priority can be used to protect sensors, e.g. streaming data from ECGs and accelerometers, to maintain a required throughput with a loss ratio below a threshold.

This project has also raised a number of interesting problems and challenges that can be addressed in more detail in future work. The performance monitoring and control method is designed with three main parameters that can be tuned for optimal results. Firstly, the size of the monitoring block that determines the resolution of the performance metrics as well as for the response time for the control actions. Secondly, the number of previous monitoring blocks (B_n, B_{n-1}, B_{n-2} etc), and their relative weight, that the control algorithm should be based on. The performance measurement results (mainly packet loss in this case), are calculated per each received monitoring block. To which degree the control method should be able to rapidly adapt to changes or mainly rely on an average of previous information, is decided by these parameters. Thirdly, the step size (Δt) that controls the time interval between transmitted packets (or packet frequency). The step size determines the response time but also the stability of the system. A more systematic study of these aspects related to control theory is part of future work

The control method is implemented for contention-based access network services and in a topology that is not limited to a single star topology. It can also be used in combination with contention-free access in star networks with a single hop between sensor nodes and the coordinator. Guaranteed time slots (GTS) in IEEE 802.15.4 can be used for the most critical application, and the remaining capacity shared among other sensor nodes according to one of the control algorithms presented in the paper.

Finally, a more detailed case study where healthcare and environment parameters are monitored and controlled on the application level is included in plans for future work.

VII. Conclusions

The presented control method implementation can be used to provide quality of service capabilities for applications in wireless sensor networks using contention-based access. A performance meter feeds the performance control function continuously with quality estimates of the transmitted sensor data. The test results using the control algorithm for priority were promising.

VIII. References

[1] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function”, IEEE JSAC, Volume 18, No 3, March 2000.

[2] Hai L. Vu, “Collision Probability in Saturated IEEE 802.11 Networks”, Australian Telecommunication Networks and Applications Conference, Australia, 2006.

[3] K. Duffy, D. Malone and D.J. Leith, “Modeling the 802.11 Distributed Coordination Function in Non-saturated Conditions”, Communications Letters, IEEE, Volume 9, Issue 8, pp. 715–717, Aug 2005.

[4] F. Österlind and A. Dunkels, “Approaching the Maximum 802.15.4 Multi-hop Throughput”, HotEmnets, Virginia, June 2008.

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

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

[7] I. Orhan, A. Gonga and T. Lindh, “An End-to-End Performance Meter for Applications in Wireless Body Sensor Networks”, BSN 2008, Hongkong, June 2008.

[8] C.Y. Wan, S.B. Eisenman, A.T. Campbell, “CODA: congestion detection and avoidance in sensor networks”, 1st conference on embedded networked sensor systems, Los Angeles, 2003.

[9] J. Kim, S. Kim, S, Choi and D. Qiao, “CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs”, INFOCOM, 2006.

[10] 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.

[11] M. Brenning, B. Olander, I. Orhan, J. Wennberg and Thomas Lindh: “NeTraWeb: a Web-Based Traffic Flow Performance Meter”, SNCNW2006, Luleå, Sweden, 2006.