3-2008

HARMONI: Context-aware filtering of sensor data for continuous remote health monitoring

Iqbal MOHOMED

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Maria EBLING

William JEROME

HARMONI: Context-aware Filtering of Sensor Data for Continuous Remote Health Monitoring

Iqbal Mohomed*, Archan Misra+, Maria Ebling+, William Jerome+
University of Toronto, Toronto, ON, Canada
+IBM T. J. Watson Research Center, Hawthorne, NY, USA
iq@cs.toronto.edu {archan, ebling, wfj}@us.ibm.com

Abstract

A promising architecture for remote healthcare monitoring involves the use of a pervasive device (such as a cellular phone), which aggregates data from multiple body-worn medical sensors and transmits the data to the backend. Unfortunately, the volume of data generated by increasingly sophisticated continuously-active sensors can overwhelm the resources on the mobile device. We propose imbuing the mobile device with the intelligence to perform context-aware filtering of sensor data streams in order to reduce transmissions in cases where the observed data corresponds to the norm expected by the system in a given context. To investigate the efficacy of this technique, we implemented the HARMONI middleware on a mobile device, and used it to collect real sensor data from users. Our experiments demonstrate that context-aware filtering can reduce the uplink bandwidth requirements of the system by up to 72%.

1. Introduction

Remote medical monitoring, particularly of people with chronic diseases such as congestive heart failure or Alzheimer's, is widely viewed as a critical technological innovation for reducing healthcare delivery costs and improving the responsiveness of medical care. Our focus in this paper is on improving the operational efficiency of the so-called 3-tier remote monitoring architecture [1, 2], where an open, standards-based personal gateway device (such as a PDA or smart phone) is interposed between the body-worn biomedical sensors and the remote backend infrastructure. The gateway device connects to the sensors via a short-range Body Area Network (BAN) technology (e.g., Bluetooth™ or ZigBee™), and uses a second wide-area (WAN) interface (e.g., GPRS or 802.11) to transport the collected sensor data to the backend infrastructure.

Existing prototypes conforming to the 3-tier architecture are geared towards low-intensity or episodically used sensors (e.g., weight scales or glucose readings); accordingly, they employ a relatively unsophisticated usage model, where the mobile device merely relays the sporadic sensor data (without any local processing) back to the remote sensor for either real-time or offline analysis. To significantly reduce the higher volumes of sensor data that will arise from the use of increasingly sophisticated, continually-transmitting sensors (e.g., ECG or EMG), we propose a technique called *context-aware filtering*, in which the relaying mobile device dynamically modifies its processing logic based on changes in the user's context. This technique is particularly beneficial for applications that require fine-grain transmission of sensor data only on those 'rare' occasions when the readings deviate from the *normal* or *expected* pattern.

To evaluate the context-aware filtering technique, we have implemented the HARMONI (Healthcare-oriented Adaptive Remote Monitoring) middleware. HARMONI includes a light-weight event engine that runs on the mobile device, and processes incoming sensor data streams using rules that are appropriate for the current context. Whenever the user's context changes, the rules that are in effect on the event engine are updated.

Using the HARMONI prototype, we collected live traces of the heart rate of users as their context changed and applied various context-driven rules to these traces. Our experiments suggest that appropriate context-aware filtering can significantly reduce the volume of transmitted medical data, and can be as important, if not more, than just compression on the raw data stream.

In the sections that follow, we present the context-aware filtering technique (sec. 2), describe the HARMONI middleware (sec. 3), provide some results from our evaluation (sec. 4) and conclusions (sec 5.).

2. Context-Aware Filtering

In many sensor environments, a large fraction of the collected data conforms to *expected* patterns, and data transmissions are only necessary for communicating *rare* or *unexpected* events. However, the use of pervasive devices to remotely monitor medical data from diverse individuals introduces a new feature, *variable context*, that is typically not associated with other sensing applications for rare-event detection. As an example, sensors used for detecting the rare condition of radioactive fallout [3] or network intrusions [4] have fixed processing logic on the sensors; these sensors typically match their sensed data against known signatures to determine if the detected event is interesting enough to warrant transmission to the backend. In contrast to these scenarios, medical monitoring is often highly situation-dependent for two distinct reasons:

a) individuals change their behavior and activity multiple times throughout the day, which directly affects the expected patterns for many medical streams (e.g., heart rate)

b) the set of medical patterns or events of interest also vary based on factors such as changes in the user's environment, evolving medical history or currently prescribed medication.

Based on these observations, we propose the *context-aware data filtering* technique, in which the relaying mobile device dynamically modifies its processing logic based on changes in the user's context. The term 'filtering' refers to a variety of operations that may be locally applied to one or more sensor streams before their transmission to the backend, such as elimination of one or more samples from a data stream (e.g., remove heart rate readings falling between 70-90), statistical summarization of a sequence of sensor data samples (e.g., compute average of heart rate readings over 10 minutes) or transformation of a set of sensor data samples (e.g., transmit the high-frequency components of an EKG signal).

3. HARMONI Middleware Overview

To investigate context-aware filtering, we designed and implemented the HARMONI middleware for remote health monitoring. Figure 1 shows the overall functional architecture of HARMONI, including the division of functionality between the pervasive mobile device and the backend server. At the mobile end, the core component is the event processor, also known as the Event Engine, which processes the data readings from the different sensors, and applies a variety of filters and transformations to generate higher-layer events from the underlying sensor streams.

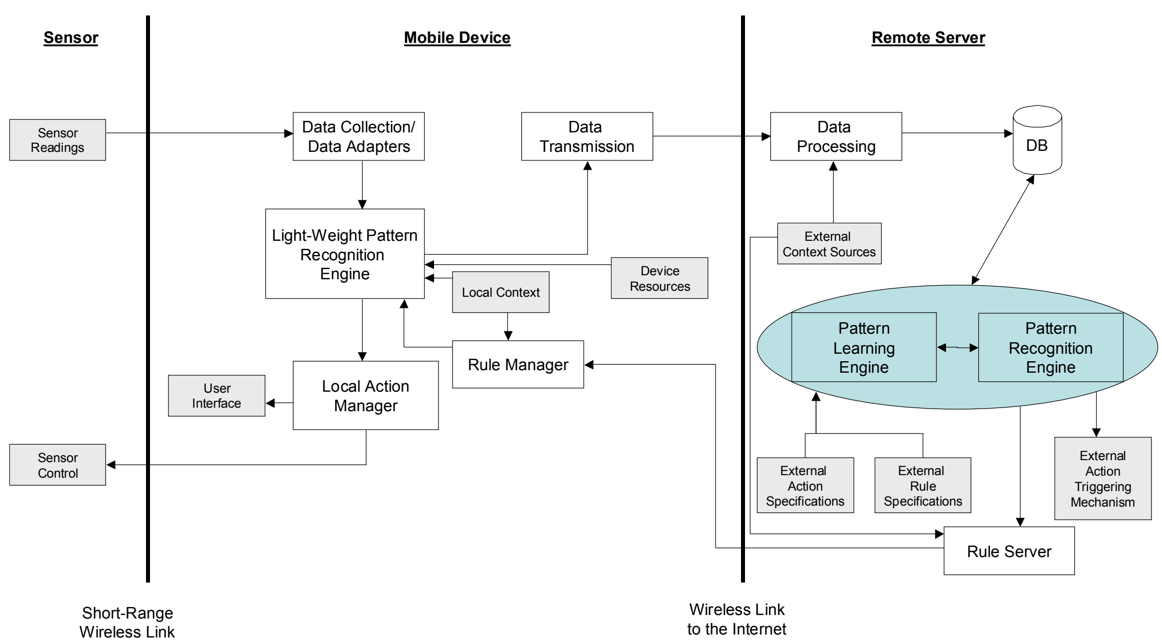

Figure 1: Functional Architecture of HARMONI . Unshaded boxes consist of the software components that we implemented. The gray boxes represent components external to HARMONI. The boxes in the blue ellipse are server-side components that enable rule personalization, a future goal of HARMONI.

Other components of HARMONI are described next:

- The Data Adaptation component on the mobile device is responsible for interfacing to the individual sensors and converting their proprietary data formats into a standard sensor stream format that may be processed by the Event Engine.
- The Rule Manager on the mobile device and the Rule Server on the backend interact with each other to dynamically alter the processing rules that are currently used by the Event Engine, in response to appropriate changes in context.
- The Data Transmission component on the mobile device is responsible for compressing the output events generated by the Rule Engine, and subsequently transmitting them to the backend infrastructure, where the events are decompressed and consumed by the Data Processing layer.
- The Local Action Manager component is responsible for manipulating settings on the local device. Examples of local actions include the generation of audible alarms (e.g., when irregular heart rates are detected on a heart patient) or the setting of a reminder on the device's calendar (e.g., issuing an 'exercise' reminder for 6 pm, when a patient's data suggests that they have not walked the required number of steps in the morning).

3.1. HARMONI Event Engine

The HARMONI Event Engine is responsible for processing the continuous streams of sensor events, arriving via the Data Adaptation component, based on a set of configured rules. Not only can the set of rules that are active on the event engine change, new rules can be uploaded by the backend (the Rule Server component) to the client device (via the Rule Manager component). Each rule consists of two parts: (a) Predicate, which describes the specific conditions (e.g., attribute ranges of the sensor samples) that trigger the application of the rule, and (b) the Action, which describes the actual processing performed on the underlying data. Processing includes transmitting data or aggregate values to the backend (via the data transmission component) as well as triggering actions on the local device (by sending messages to the Local Action Manager component).

3.2. Implementation Details

We implemented HARMONI on the Nokia 770 Internet Tablet, which is a Linux-based wireless PDA with two radios - an 802.11b/g interface and Bluetooth 1.2. For our experiments, we used the Nonin 4100 sensor, which provides Pulse Oximetry (SpO2, defined as the amount of oxygen saturation in the blood) and heart rate (pulse) data. The sensor transmitted data to the PDA via Bluetooth while the HARMONI middleware relayed data to a backend server via 802.11b.

4. Evaluation

We conducted a study in which HARMONI was used to record the heart rates of four different individuals. All participants were adult males between the ages of 26 and 40. None of the participants had any serious medical condition that was previously known, but they varied in their level of physical fitness.

Users participated in hour-long sessions, with no more than one session per day for any user. For the entire duration of the session, a Nonin pulse oximeter was attached to the participant's non-dominant hand. We collected only the heart rate data from the sensor stream. In the first half of the session, we asked the participant to engage in their regular office work. For all four individuals, this mainly consisted of using the computer and telephone, and reading paper documents. In the second half of the session, we asked participants to engage in strenuous exercise. The precise form of the workout was left to the participant's discretion, but we requested that they attempt to engage in what they felt was their usual aerobic workout. Participants reported engaging in a variety of activities, such as running, jogging, jumping-jacks and abdominal crunches. Thus, each session effectively involves a "context-switch" at the mid-point, from the relatively sedentary "office" context to the "gym" context characterized by vigorous activity.

Out of the four participants, one user participated in three sessions, one participated in two sessions, and the remaining two users participated in a single session each. Thus, we obtained a total of seven distinct traces from our study.

4.1. Results

Once we collected sensor traces, we replayed them under different scenarios of interest. The replay mechanism acted as a stub for the data collection system in that data events were sent to the event engine based on the recorded traces instead of being extracted directly from the transmissions of the heart rate sensor. This trace-driven approach is necessary for making meaningful comparisons because uncontrollable physiological and environmental variations make it impossible to get the exact same data stream from two different sessions.

Figure 2 shows the amount of data transmitted over the network for a representative user in four different "context" conditions. In the first condition ("None"), no specific context is assumed, and the rule that is effective on the event-engine transmits all sensor values. In the second condition, it is assumed that the user is in the "office" context for the entire duration of the run. The rule that is in effect in this context expects the user's heart rate to be in the range of 50 and 90 beats per minute (bpm). If 10 consecutive sensor samples all stay within this 50-90 bounds, the rule engine transmits a single average of these 10 values. Any values outside this bound are transmitted individually. In the third condition, the "gym" context is assumed to apply for the entire duration of the session. The rule that is effective in this scenario is similar to the previous one except that the expected heart rate range is between 90 and 170 bpm. The fourth condition reflects the correct operation of HARMONI, in that it reflects the use of the "office" context for the first half of the session (i.e., the rule indicates a normal bound of 50-90), followed by a switch to the "gym" context for the latter half (i.e, the normal bound in the rule changes to 90-170).

From the figure, we see that appropriate context-filtering results in a significant reduction in the amount of data transmitted, with a 72% reduction when the data is transmitted uncompressed and a 51% reduction when data is compressed using an LZ-78 compressor. Another significant observation is that the total amount of compressed data that is transmitted without context-aware filtering is 34% greater than that transmitted with context-filtering without compression. Thus, we conclude that **context-aware filtering can lead to a significant reduction in the amount of data that must be transmitted by a mobile device**. Moreover, the result demonstrates that the use of context-aware filtering (without any compression) can have more impact (result in lower traffic of 19.59 KB) than the application of compression directly on the raw data stream (29.64 KB).

Figure 2 also illustrates that the use of an incorrect context state (and thus an inappropriate filtering rule) can impose a performance penalty in terms of higher data volume. For example, if the "office" context is assumed throughout (i.e., an incorrect context applies for ~50% of the total session duration), the total data transmitted is higher (by ~48% when compression is absent, and by ~17% when compression is applied) than when the context is correctly switched mid-stream. We conclude that **filtering based on the incorrect context can lead to reduced data savings.**

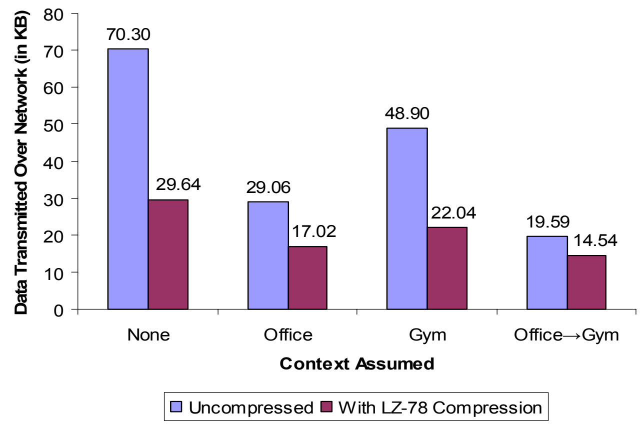

Figure 2: Relative Volume of Data Transmission for a Specific User Session under 4 different Event-Processing Rules.

5. Conclusions

In this paper, we presented context-aware filtering, a technique to reduce the data transmission requirements of continuous long-term remote health monitoring applications that utilize a mobile device to relay data-streams generated by wearable sensors to a remote server. Context-aware filtering allows the mobile device to dynamically apply situation-specific processing logic intended to reduce the size of the data stream. To evaluate our technique, we developed the HARMONI middleware, which enables context-aware filtering through a rule-based event processing engine, where rules are updated based on changes in context. Using our HARMONI prototype, we conducted a study in which we monitored the heart rate of users in different contexts. In this study, context-aware filtering reduced the amount of data transmitted by the mobile device by up to 72%.

6. References

[1] D. Husemann; C. Narayanaswami and M. Nidd, P*ersonal Mobile Hub*, 8th International Symposium on Wearable Computers (ISWC 2004), Oct 2004, Arlington, VA, USA

[2] E. Lubrin, E. Lawrence and K.F. Navarro, *MoteCare: An Adaptive Smart BAN Health Monitoring System*, 24th IASTED International Conference on Biomedical Engineering, Feb 2006.

[3] S. M. Brennan, A. B. Maccabe, A, M. Mielke and D. C. Torney, *Radiation Detection with Distributed Sensor Networks,* IEEE Computer, Aug 2004.

[4] A. Da Silva, et al, *Decentralized intrusion detection in wireless sensor networks*, 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile networks, Oct 2005.