A Real-time Architecture for Service Robots in USN Environments

Byoung-Wook Choi
Seoul National Univ. of Tech.
bwchoi@smnt.ac.kr

Dong-Gwan Shin
Robostar Co., Ltd.
shingun@robostar.co.kr

Jeong-Ho Park
Dasarobot Co., Ltd.
jhpark@dasarobot.co.kr

Soo-Yeong Yi
Seoul National Univ. of Tech.
suylee@smnt.ac.kr

Abstract – This paper implements a real-time software architecture of using dual kernel of standard Linux and real-time embedded Linux, Xenomai. It is used for the real-time control of service robots doing assisted-living service in Ubiquitous Sensor Network (USN) environments for the elderly people. This is achieved by monitoring their mental and physiological data with wireless sensor nodes. Sensed data in sensor nodes are routed back to a moving sink node by multi-hop communication. For the purpose of handling emergency situations, real-time processing is required both in routing algorithm from the sensor nodes to the moving sink node and processing collected data for invoking tasks. The real-time tasks are implemented with priorities to rapidly respond to urgent sensor data. In order to validate deterministic response of the proposed system, the performance measurement of the delay in handling sensing data transmission and performing trajectory control with feedback loop is evaluated with non real-time standard Linux.

Keywords – Service robot, real-time, USN, assisted living

1. Introduction

Recent advances in wireless communication, embedded systems and ubiquitous computing have enabled the development of small, low cost, low power, multi-functional sensor nodes. A USN is composed of a large number of sensor nodes, which consist of sensing, collecting, correlating, and aggregating data in parallel. The ability of the USN to sense phenomenon such as temperature, light, pressure, or movement make it a promising sensing technique for many applications. Therefore, USNs have been used in numerous applications including military applications, environmental monitoring, health applications, home automation and smart environment [1-4].

Due to the problem of an aging society, there are many proposals of using the USN for the senior citizens in hospitals and silver towns. As the population ages there will be an increasing demand on health care resources [5]. New and improved devices and applications are expected to emerge. UIUC suggested the I-Living System, which includes open-software and hardware to provide daily life support for the elderly people [6-7].

Realization of these USN applications requires wireless ad-hoc networking techniques. Since the sensor nodes are usually scattered and densely deployed in a sensor field, multi-hop communication is typically adopted by using neighbor nodes. Data collected in sensor nodes are routed back to a central entity. The central entity performs the functions of data collection, coordination and transmission to central controller, e.g., human interaction. Sometimes the central entity is referred to as a sink node. The sink node collects and transmits the data to the main controller for in-depth analysis such as monitoring, alarming, and forming database [8]. Furthermore, there are many ongoing researches to utilize sensor networks in advancing mobile service robots. A wireless sensor network can extend the sensory perception of people and robots far beyond their normal range. None of the studies reported related to the conduct of active services utilizing USNs. The mobile robots will be able to do more active service based on the data collected by sensor networks such as to respond to emergency situation rather than passive. In the sense of physical characteristics, our application is one of the examples of integrated sensor/actor node since the robot is capable of both sensing and acting. It is sometimes referred to as wireless sensor and actor network (WSAN) because it consists of a group of sensors and actors linked by wireless medium to perform distributed sensing and acting tasks. The design of WSANs featuring node mobility is investigated for control applications [9]. A mobile sensor platform for a mobile distributed sensor network for real-time target detection, recognition, and tracking was developed by Sandia National Laboratories [10].

In this paper, we deploy a sink node into a main robot controller then it becomes the moving sink node and it can be used for aggregating sensor data. The robot should act on its own sensor readings as well as the sink node’s data received from the networks. Therefore, coordinated behavior of the mobile robot is an important issue because the tasks that need to be performed evolve with time. For instance, in our application for the active assisted living, coordinated actions should be initiated on the event as soon as possible when the sensors are used to detect elderly people’s behavior, e.g., a fall [3, 9]. Thus a real-time control architecture is strongly required to do active service and achieve real-time communications to remote care providers [11]. In order to achieve real-time requirements, real-time Operating Systems (OSs) are needed. The real-time feature is provided by two open-source Linux extensions: RTAI [12] and Xenomai [13]. Even though both RTAI and Xenomai are worthy of consideration for our application, we developed a real-time software architecture, based on Xenomai, on a robot controller where a sink node is integrated. Xenomai
is a real-time development framework cooperating with the Linux kernel, in order to provide a pervasive, interface-agnostic, hard real-time support to user-space applications, seamlessly integrated into the GNU/Linux environment. Moreover, Xenomai is better structured and is available for a larger number of platforms and provides a set of emulation layers which may prove useful when porting large systems [13, 14]. In performance measurements on four different OSs, they revealed that the open-source software such as RTAI, Xenomai is suitable for hard real-time applications [14].

2. System Structure

A service system for the assisted-living in USN environments should provide sensor nodes to sense phenomenon, a sink node to aggregate data and send them to the main controller, a service robot to make active service to emergency situations and a remote monitoring system. And an efficient network routing algorithms from the sensor nodes to the sink node is required. Finally, a real-time software architecture to process data in order of the priority of tasks is needed.

In order to meet such requirements, Fig. 1 shows an overall structure to do active services for elderly people with service robots in USN environments. The proposed system consists of sensor fields, an integrated sink node and a remote monitoring server. The sensor nodes are usually scattered to sense phenomenon in sensor fields. The role of sensor nodes is to collect data and route data back to the sink node. The integrated sink node is a mobile robot in which a sink node is installed on it to perform the functions of data collecting. The robot should perform appropriate actions based on this collected data for example moving to the sensor node where the event is triggered and monitoring the behavior of elderly people by using a camera installed on the mobile robot. All actions are performed in the tasks scheduled by the real-time operating system. The robot controller also transmits collected information to a remote monitoring server via wireless LAN for doing other services such like alarming doctor or remote monitoring of the people.

![Fig. 1. Assisted-living system structure](image)

The sensor node consists of a control board called H-mote and a daughter sensor board [15]. The sensor node consists of two microprocessors; one is a micro-controller and the other is an RF chip. The processing unit of H-mote is an 8 MHz T1 MSP430 micro-controller. An open-source operating system, TinyOS, is used on this processor [16]. The RF chip is a CC2420 with 2.4 GHz IEEE 802.15.4 compliant RF transceiver. The H-mote provides connection to a sensor board and USB interface for integrating other controllers. The logical connection, however, is achieved by RS-232 interface. Therefore, we can use the same sensor control board for both sensor nodes and the sink node.

The sensor nodes are deployed in sensor fields to sense phenomenon. They transmit sensing data of physiological data or patient’s status to the sink nodes through a single-hop, or multi-hop. In general, a few sink nodes may be involved in collecting data, making decisions and taking actions. In this paper, a mobile sink node embedded on a main controller of the robot is used. It performs the functions of data collection and transmission data to the main controller. Sink mobility has attracted much research interest in recent years because it could improve network performance such as energy efficiency and throughput [17]. Therefore, wireless sensor networks with mobile sinks have many advantages over the static sensor networks.

3. Multi-hop network

This paper focuses on a type of WSAN that consists of many sensor nodes and an integrated sensor/actor node that is a sink node, where the collecting sensed data is handled periodically and transferred to the main controller which is in charge of acting on the data. The deployment diagram of the proposed system in the WSAN applications can be seen in Fig. 2. As it consists of a group of sensors and sensor/actor linked by wireless medium, radio frequency (RF), to perform distributed sensing and acting tasks, multi-hop networking techniques are required to transmit data from the sensor node to the sink node. As shown in Fig. 2, multi-hop, or ad-hoc, wireless networks use two or more wireless hops to convey information from a source to a destination. In this section, we present data transmission method of multi-hop and its data structure.

![Fig. 2. The deployment diagram of the proposed system.](image)

Although many protocols and algorithms have been proposed for traditional wireless ad-hoc networks, they are not well suited for the unique features and application requirements of sensor networks [1]. The sensor node used in this paper incorporates the CC2420 which supports the Zigbee protocol [18]. The Zigbee protocol has its very own protocol within the PHY/MAC class, hence preventing RF signal collision for reliable transmission between the nodes. Also, each node must reduce its power consumption for the convenience of maintenance, thus shortening the distance of data transmission range. Therefore, there is a limit for the wireless transmission of
the sensor node in range, and a single-hop transmission will not be efficient enough to transmit the data. To resolve this, there is a need for the multi-hop method, when the sink node tries to collect the data from the sensor node over a long distance. In this paper, we realize multi-hop technique through the Zigbee protocol for an efficient transfer of event information obtained in the sensor nodes.

In this application, sensor nodes are stationary whereas the actuator node, e.g., the mobile robot is mobile. And the sink node is installed in the mobile robot so it becomes a mobile sink node. Even though sink mobility can improve network performance such as energy efficiency and throughput; we need to reconfigure network route when the robot moves. An autonomous moving strategy for the mobile sinks in data-gathering applications is suggested in the previous result [17]. The movement of robot requires a quick and regular configuration of network routes, but such frequent configuration will reduce the transmission rate. Hence, an adequate time distribution depending on the speed of the robot is needed to prevent this. Along with such routing results, the position of the sensor node used by the user can be found, hence giving the robot access to the position where the event occurred.

4. Dual-kernel approach

In recent years, dramatic improvements in hardware computing power and free software quality have generated much interest in the possibility of using standard Linux for embedded real-time applications in robotics. Laboratory tests showed that standard Linux can be used for embedded real-time robotics [19]. Moreover, a new implementation of the scheduler has been provided in Linux kernel 2.6. However, the standard Linux is not yet suitable for hard real-time applications, as required in feedback control, especially in trajectory control in robotics. Therefore, we need to consider a way of adding to Linux the possibility of defining a few real-time tasks that are ensured to get control within a deterministic response time. This feature is provided by open-source Linux extension, Xenomai. In order to co-operate with Linux it is necessary that the underlying hardware be shared by Linux and additional component. This is achieved in Xenomai by using the adaptive domain environment for operating systems (Adeos) [20]. To implement this requirement, we have ported several patches for porting Linux, Adeos and Xenomai to the target board. We present a real-time control architecture based on Xenomai for assisted living service with the robot in USN environments.

The implementation to allow deterministic response times regardless of the actual Linux implementation is described in Fig. 3. Adeos is a resource virtualization layer available as a Linux kernel patch to run several operating systems on a single hardware platform. Adeos enables multiple entities called domains to exist simultaneously on the same machine. These domains do not necessarily see each other, but all of them see Adeos. We have ported two domains on the same hardware platform, one is the real-time embedded Linux, Xenomai 2.3 and the other is the non-real-time standard Linux 2.6.17. Kernel space application is handled first depending on the priority and then standard Linux kernel is scheduled with the lowest priority. This figure also illustrates the position of the Adeos layer in the Xenomai architecture. The Adeos interface is directly exposed to the Hardware Abstraction Layer (HAL) which underlies the Xenomai core. Therefore, most of the requests for Adeos services are issued from HAL, thus allowing predictable interrupt latencies in the lowest micro-second level range to Xenomai whatever activity Linux is undergoing.

As shown in Fig. 3, Xenomai allows running real-time applications named threads either strictly in kernel space, or within the address space of a Linux process. All threads managed by Xenomai are known from the real-time nucleus. Adeos ensures that events are dispatched in an orderly manner to the various client domains according to their respective priority. The events are incoming external interrupt, system calls issued by Linux applications and other system events triggered by the kernel code. Xenomai threads are running over the context of the highest priority domain but Linux kernel is considered as the lowest priority domain. Therefore, one can perform the real-time applications in the context of the Xenomai as well as standard Linux based application as the non real-time applications, hence insuring the convenience and extension of development.

5. A Real-time Architecture for the Sink node

In the proposed architecture, there are both the sink node and the main controller on the mobile service robot. The main roles of the robot are listed in the following:

- Sink Node: Gathers the data received from the sensor node and transmits it to the main controller
- Active Agent: Provides an active service of the robot itself for the user depending on the gathered data
- Gateway: Transmits the analyzed data to remote monitoring system via WLAN for further use

In order to perform such tasks, the service robot must be equipped with the proper computing devices. In this paper, the Intel/IBM compatible i386 single board computer (SBC) was chosen as the main controller for the robot, where the standard Linux and real-time Linux Xenomai are running together as described in Fig. 3. For the sink node installed on the mobile robot, the CC2420 is used to receive data from the sensor nodes and to transmit data to the main controller. The data transfer between the sink
node and the main controller is performed through RS-232 serial communication. The main controller also has a wireless communication module. Using the Ethernet port, the nodes are connected to the remote monitoring server which supervises all data transmitted. Within the WSANs, sensors detecting a phenomenon either transmit their readings to the actor nodes which process all incoming data and initiate appropriate action, or route back to the sink node which may issue action commands to actors. In this paper, the sink node gathers the data from the sensor nodes via wireless sensor network and analyzes it before issuing it to the actor node. The robot acts as the actor node in the proposed system so it processes all incoming data and performs active service. Also, the sink node needs to route the network frequently to provide a minimum energy data transmission in a multi-hop communication environment. In addition, the sink node must gather data from the information sent by sensor nodes, and then send the data to the actor in the order of importance in real-time.

The real-time sink node architecture can be seen in Fig. 4, fulfilling the requirements to fulfill active services in the WSANs. For performing the real-time data collecting and processing, the real-time software architecture based on Xenomai is ported on the SBC, the RT device driver of RT_COM is implemented to communicate the sink node through RS-232 interface, RT tasks in kernel space are designed to process incoming data and conduct real-time control with high priority, and finally both RT applications and NRT applications are implanted to do other functions in the user space with low priority. The data communication between tasks from the kernel space to the user space is achieved by RT FIFO (real-time FIFO mechanism) and message queue. In the sense of hardware, the sink node and sensor nodes have the same structure, and both nodes use TinyOS to efficiently manage limited resource. TinyOS is an appropriate operating system to provide an efficient performance within a limited hardware memory capacity, since it is an event-based status switching operating system. The sink node gathers the data from the sensor nodes via the Zigbee network protocol and transmits the data to the main controller with RS-232 interface.

6. Experiment and Performance Measurement

For performance measurement, we have implemented the proposed system. As WSAN, four sensor nodes that perform only sensing are deployed. The mobile robot shown in Fig. 3 is used as an actor on which a sink node is installed. This means that the sink node becomes a moving sink node because the sink node is fitted within the main controller. The robot has various sensors such as ultrasonic sensors and PSDs which are used for obstacle avoidance. In addition, a StarGazer is adopted to perform localization which is required to do active services such as autonomous navigation to the position of the elderly (for example finding the position of the elderly people by using the position of the issued sensor). Wireless AP (Access Point) for communicating with the remote monitoring system is installed as well.

Realization of the real-time control architecture for the assisted living services for the elderly in this paper requires wireless technique, robot control technique and real-time software implementation. Here we concentrated on the real-time performance for both handling data gathered from WSAN and performing trajectory control.

The implementation to allow deterministic response based on Xenomai is described in Fig. 4. For the real-time realization, Xenomai 2.3 has been applied to the Adeos architecture. The standard Linux of kernel 2.6.17 is ported to the main board for general application software as Root Domain, and the sink node uses CC2420 to route back data from the sensor nodes to the main controller. The data transfer between the sink node and the main controller is performed through RS-232 serial communication as described in Fig. 4. Therefore, we have realized the RT device driver of RT_COM to perform the real-time data collecting and processing.

The robot is mainly used as the actor so we need to implement real-time control software as well. The software modules needed to meet the requirement for the mobile robot are simply displayed as several processes in user space. In this paper, because the interface for the motor control board and localization sensor is serial communication, we can also use RT_COM for achieving motor control and handling position data. For other sensors, standard serial interface is used. The simplified overall implementation in the robot is described in Fig. 5. The general inter-task communication (ITC) mechanisms are used to achieve communication between RT tasks and NRT standard threads. The tasks requiring real-time performance, such like rt_motor, rt_local and rt_sensor, are implemented in kernel space with real-time priority. For other tasks related with general applications of the robot are implemented in user space as standard Linux threads, for instance sensor_thread and user processes.

All tables should be numbered consecutively and have a caption consisting of the table number and a brief title. This number should be used when referring to the table in the text. Tables should be inserted as part of the text as close as possible to its first reference.
Fig. 5. The realization of real-time software in the robot.

In the first round of tests, we focused on the difference in delay times of handling the sensing data between RT and NRT software environments. In this case, we can examine the real-time performance for aggregating sensor data even though those are gathered from the sensor nodes. Here, we focused on the control performance by driving the robot straight forward. Since the robot is an actor in this paper, the control performance is also important in performing the active services. The mobile robot will be controlled from (344, 100) (cm) to (344, 230) (cm), in Cartesian coordinates, while maintaining heading angle of the robot at 120 degrees. In robotics of mobile robots, position and heading angle are major control arguments. We note that the experiment here is different from the trajectory control described in Fig. 6, where feedback control is not involved. The feedback control loop should be performed with a control loop time of 300 ms to follow the reference command. The complicated algorithm, however, is not adopted since the purpose of the test is to verify the real-time characteristics not to show the performance of control algorithm. Although in [18], they evaluated the possibility of standard Linux for embedded real-time application in robotics and showed that a large variety of application could be implemented using Linux, the current version of Linux is not yet suitable for hard real-time applications, as required in feedback control. The trajectories and heading angles of the robot were also not correctly controlled, in case of NRT software environment as shown in Figs. 6(a) and 7(a). The results shown in Figs. 6(b) and 7(b) maintain the control goal regardless of the number of workload tasks even though the trajectory does follow the reference command exactly. The final control error results from the slippery floor and the poor resolution of localization sensors adopted in this paper. This is another problem that is beyond the scope of this paper and we will improve the control performance itself.

<table>
<thead>
<tr>
<th>No. of tasks</th>
<th>actual distance / reference distance (cm)</th>
<th>working time (ms)</th>
<th>actual time / reference time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>132/130</td>
<td>6,300</td>
<td>300/300</td>
</tr>
<tr>
<td>5</td>
<td>143/130</td>
<td>7,803</td>
<td>300–600/300</td>
</tr>
<tr>
<td>7</td>
<td>140/130</td>
<td>7,960</td>
<td>500–600/300</td>
</tr>
</tbody>
</table>

In Figs. 6-8, all results are reported according to the number of workload tasks, where symbols denote the number of workload tasks (X: no other task, □: 5 tasks, O: 7 tasks). Fig. 6 shows the robot trajectories in Cartesian space. Fig. 7 describes heading angle of the robot in the trajectory, and Fig. 8 indicates the performed control loop cycle. The control loop cycle time of Fig. 8(a) varies when workload increases. On the other hand, the control loop of Fig. 8(b) maintains at 300 ms. This result implies that the proposed architecture is feasible for hard real-time control.

Fig. 6. The trajectories of the robot in Cartesian space.

Fig. 7. Heading angles of the robot.

Fig. 8. The sampling times in control loop of the robot.

7. Conclusions

This paper proposes a real-time control software architecture for mobile robots in WSANs, to do health care service or assisted living service for the elderly person. The WSAN consists of a robot, a sink node fitted on the robot controller, and sensor nodes to sense phenomenon such like temperature, pressure or movement, etc. The robot becomes an actor for performing active services, and the sink node collecting the sensing data is the moving sink node. For facilitating the moving sink node, the network routing algorithm for multi-hop is conducted for the minimum energy cost by using the TinyOS. The real-time software architecture is based on a dual kernel
approach, where Xenomai and standard Linux are running together on top of the Adeos environment. In order to interface the sink node, the real-time serial device driver is also ported. For realization of the health care system, we introduced remote monitoring system without an explicit server. As we can use system calls under two kernels, the role of the gateway to the outside and the controller with real-time performance as well as the sink node, can be easily fulfilled.

In order to verify the feasibility and real-time performance, we concentrated on two features; sensor network data handling and control performance of the robot, with feedback loop. Based on the reported performance measure, we observe that the following facts:

- The performance of the proposed system, in the sense of the delay, is good in both handling sensing data transmission and performing trajectory control with feedback loop.
- Both real-time task and real-time serial device are required to satisfy real-time performance.
- Xenomai is better structured and is available for a larger number of platforms; the flexibility of the proposed system is applicable to both real-time and non real-time applications.
- The robot performs well for the moving sink node and the actor in WSANs.

Even though there are many research projects for sensor networks introduced in [1], we encourage more development in solutions to assist living for the elderly person with the mobile robot in WSANs as follows.
- Robot control problems including localization should be overcome.
- Non tactile sensors to sense the psychological data of the elderly person must be developed.
- The moving sink node has advantages and disadvantages. Re-routing requirement is one of the most important issue in this application.
- Position recognition, for instance RSSI, of the sensor nodes is essential for performing active services.

The more practical service which is implemented in the context of WSANs is needed to be a commercial solution, for example; visual tracking, voice chatting and rich database for clinical purpose.

Acknowledgements

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2007-013-D00034).

References


