社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－UBI－8（37）
2005／6／9

Effective Information Gathering of Predictable Agent System over Wireless Sensor Networks.

Young-Min Ji*, Kang-Ho Woon**, Sung-Dong Kim**, Dong-Hyun Chae* and Sun-Shin An*,

* Department of Electronics and Computer Engineering, Korea University, Korea,
yangsamy@dsys.korea.ac.kr, hsunhwa@dsys.korea.ac.kr, sunshin@dsys.korea.ac.kr
** Ubiquitous Computing Research Center, Korea Electronics Technology Institute,
khwon@keti.re.kr, sdkim@keti.re.kr

ABSTRACT

In general, wireless sensor networks contain a lot of sensor nodes which used to gather the sensing information and sent it to other nodes. Actually, energy is most of problem so that we need to research on the sensor networks. In this case, we consider about the efficiency agent system is prediction of sensing information. The predictable agent system is simple and has several energy saving features. We implemented this Agent System over TinyOS[1] and discuss the results of some execution runs of this implementation.

1. Introduction

Recently, Ubiquitous Computing technology that means smart space concept is applied to the infrastructure. To organize smart space, it is necessary to have the interface which relates computing space with the physical space information. sensor network is the real-world information acquisition infra-system to transmit the information to the computing space, is the system that giving sensing and information processing ability at each network node, and is the distributed processing system using Radio Frequency.

Wireless Sensor networks need trade-off between reliability of information and low power consumption characteristic by itself. Furthermore, sensor network needs local processing inside node to minimize power consumption by RF transmission and multi-hop. Actually, the energy to transmit 1KB data in 100m distance away other node is consumed 3J and that energy has ability that the processor of 100 MIPS ability executes processing in 30 billionth[2].

Wireless sensor networks consists of battery, sensor and tiny device. This low-cost devices are deployed in the field. TinyOS has some features, limited transmission, limited power, multi-hop routing. Special quality of processor, sensor and communication hardware can greatly contributed to wireless sensor network. Specially, these features improve environment monitoring field and Home networking field. Therefore various information is collected by sensor network. Sensor device consisted of various sensor. Context-aware in ubiquitous computing, context-awareness is the concept that express and processes composition information.

This paper predicts sensing information, and we compares current sensing information with threshold information, and then we predict the correct situation. This predictable agent system improve the power consumption and context awareness.

1) This work was partially supported by CUCN (National Center of Excellence in Ubiquitous Computing and Networking).

2. Predictable Agent System

2.1 Sensor Network Node Architecture

Information is the important concept in wireless sensor networks. Predictable Agent decides transmission using threshold information, sensing information, irregular information and predicting information for the power efficiency. Commonly sensor nodes have 3 component as following.

- Processing Component : Component that achieve data processing function to transmit sensing information that is passed from sensor component into other network node through communication component.

- Sensing Component : Sensor of light, temperature, humidity, shock, acceleration etc. usually can be used variously according to application. These sensors have created state information that more than two is various being used as complex.

- Communication Component : It is other Network Node and RF interface for information exchange. Generally, RF interface uses an ISM band of

900MHz or 2.4GHz substitution.

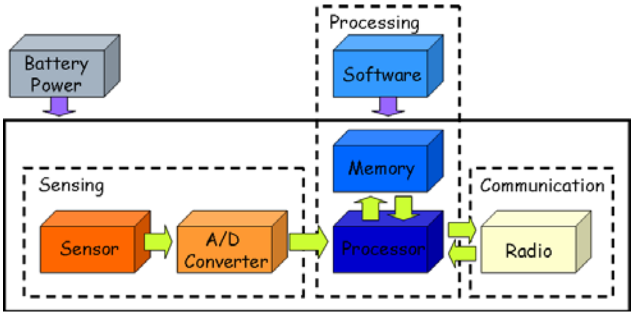

Figure 1 : Sensor Network Node Architecture

We consider efficient information transmission using sensing component and processing component among 3 component with figure 1. And is going to do to judge using sensing information designing agent system on the application layer with figure 2.

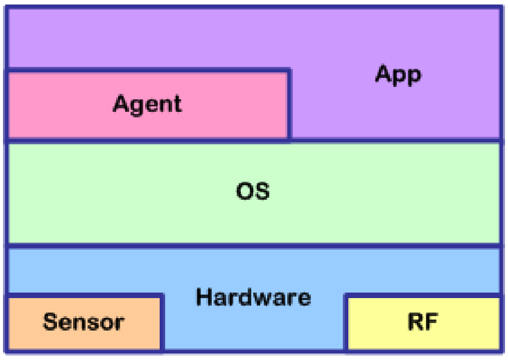

Figure 2 : Sensor Network Node Layer

2.2 Sensor Network Protocol Stack

Generally, a sensor network is composed of Layer. Layer comes for OSI 7 Layer of the same structure, too. As of the each layer, it is composed Application, Transport, Network, Data Link, Physical. And a consideration element is composed of Power Management Plan, Mobility Management Plan, Task Management Plan[3].

- Power Management Plan : The capability, lifetime, and performance of the sensors are all constrained by energy. The sensors should be able to be active for a reasonably long time without recharging the battery because maintenance is expensive.

- Mobility Management Plan : There is no standard for Mobility management in these types of sensor networks. The lack of centralized control point makes location management for sensor networks a very challenging task. For wireless sensor networks, the constraint of energy consumption is the main bottleneck for providing such service.

- Task Management Plan : The task management plane balances and schedules the sensing tasks given to a specific region. Not all sensor nodes in that region are required to perform the sensing task at the same time. As a result, some sensor nodes perform the task more than the others depending on their power level.

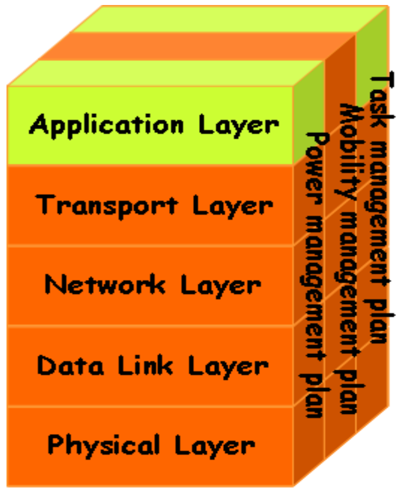

Figure 3 : Sensor Network Protocol Stack

The management planes are needed, so that sensor nodes can work together in a power efficient way, route data in a mobile sensor network, and share resources between sensor nodes. From the whole sensor network standpoint, it is more efficient if sensor nodes can collaborate with each other, so the lifetime of the sensor networks can be prolonged. We will Consider power management plan and task management plan 2 branch on an application in this paper.

2.3 Sensor Network Architecture

Usually, cluster are node set that is composed by phenomenon that is aimed or sensor nodes adjoining geographically to draw sensing information. To compose cluster, one node is used as cluster head (head) and this manages or control many nodes in cluster[4].

Network architecture that compose in this treatise has structure of 3 hierarchies of cluster base that adjoin physically. First tier composes cluster that is consisted of sensor with figure 2 and second tier and third tier compose cluster as conceptional by each network node and head node.

One head node clyster cluster, and tens ~ hundreds nodes exist in cluster. Nodes are threading sensor more than each one. Ubiquitous Computing environment uses Context-aware technology basically. Context-aware is technology intelligence enemy who can express and passes information of current physical space to

Computing space and computer make suitable judgment recognizing this[5]. Use many sensors to express situation information in sensor network. Many sensors can express one situation as complex.

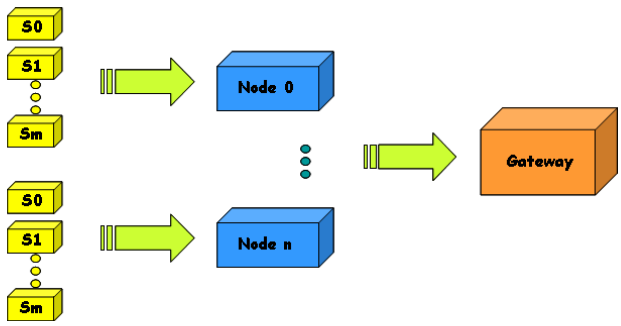

Figure 4 : Sensor Network Architecture

The most important technological issue is electric power problem in sensor network. Optimize the data amount that is transmitted by RF to minimize power consumption in sensor network. Then, can bring efficiency in electric power consumption if run RF data transmission when peculiarity of sensor network situation information is happened.

2.4 Predictable Agent System and Context Aware

Each sensor nodes have sensing component in sensor network. These sensing components collect information and transmit the information. However, energy efficiency is dropped more gradually if informations that is transmitted from sensing component are not information about all worth situations and this informations become much. It is efficient if have distinguished data that this sensing components acquire little more effectively. We are going to divide efficient a sensing information using estimate Agent that predict a sensing information that acquire forward using past a sensing information and use the estimate result and pass information.

The next a sensing information is generally predicted as an exponential average of the measured lengths of previous a sensing information. Let ν be the scale of the nth sensing informations, and let γ_{n+1} be our predicted information for the next sensing information. Then, for α, $0 \leq \alpha \leq 1$, define

$$\gamma_{n+1} = \alpha\nu_n + (1-\alpha)\gamma_n \quad (1)$$

This formula defines an exponential average. The information of γ_n contains our most recent information. stores the past history in our prediction. The parameter α controls the relative weight of recent and past history in our prediction. If α = 0, then $\gamma_{n+1} = \gamma_n$, and recent history has no effect if α = 1, then $\gamma_{n+1} = \nu_n$, and only the most recent sensing information matters. More commonly, α = 1/2, so recent history and past history are equally weighted. The initial ν_0 can be defined as a constant or as an overall system average. To understand the behavior of the exponential average, we can expand the formula for γ_{n+1} by substituting for γ_n, to find

$$\gamma_{n+1} = \alpha\nu_n + (1-\alpha)\alpha\gamma_{n-1} + \cdots + (1-\alpha)^i\alpha\gamma_{n-i} + \cdots + (1-\alpha)^{n+1}\gamma_0 \quad (2)$$

Since both α and $(1-\alpha)$ are less than or equal to 1, each successive term has less weight than its predecessor[6].

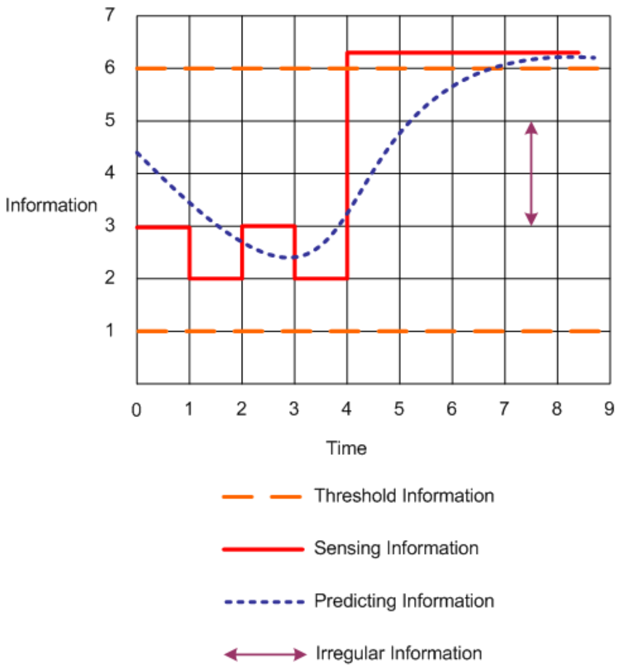

Figure 5 : Prediction of the Information

Assume that figure 5 deploy Predictable Agent System to actuality bridge and is graph that predict the state. If see Threshold Information, it displays that this information is important because is dangerous when progress is more than 6 on the Richter scale. Display dangerous situation if greaten more than actuality Sensing Information this Threshold Information. Predicting Information uses present Sensing Information, predict future states. If Predicting Information is big more than Threshold Information, can inform that danger states. And Irregular Information is thing to predict that information changes suddenly. Because range of fluctuation of information is big displays that

Dest Addr (2 Bytes)	Handler ID (1 Byte)	Group ID (1 Byte)	Msg Len (1 Byte)	Source Addr (2 Bytes)	Counter (2 Bytes)	Channel (2 Bytes)	Priority (2 Bytes)	Reading (2 Bytes)

Figure 7 : Msg Format

state of bridge is very uncertain. If use Predictable Agent with upside, can predict state of bridge and detect correct state.

3. Environment

In this section, we provide a brief overview of the mote hardware architecture, the TinyOS system, and NesC code for mote-based Predictable Agent System.

3.1 TinyOS

TinyOS is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools – all of which can be used as-is or be further refined for a custom application. TinyOS's event-driven execution model enables fine-grained power management yet allows the scheduling flexibility made necessary by the unpredictable nature of wireless communication and physical world interfaces.

3.2 MicaZ

The MICAz is a 2.4GHz, IEEE 802.15.4 compliant, Mote module used for enabling low-power, wireless, sensor networks. The MICAz Mote[7][8] features several new capabilities that enhance the overall functionality of wireless sensor networking products.

- IEEE 802.15.4 / ZigBee compliant RF transceiver
- 2.4 to 2.4835 GHz, a globally compatible ISM band
- Direct sequence spread spectrum radio which is resistant to RF interference and provides inherent data security
- 250 kbps data rate

Figure 6 : MicaZ mote sensor node

3.3 NesC Code for predictable agent system

Figure 7 is message format that consider predictation. Message has a priority flied that is message type. If priority value of variable is 0, this message displays very important message. and priority value increases, priority becomes low. Figure 8 is source code that embody actuality message. there are two types of enum variable. first is a environment variable. second is priority variable. And message has five field. sourceMoteID is a source addresss, lastSampleNumber is sensing counts, channel is access channels, priority is a type of message, data is sensing data field[9].

```
enum {
  ALPHA = 1/4,
  PERIODIC = 100,
  THRESHOLD = 600,
  IRREGULAR = 100
};

enum {
  MSG_PRIORITY_HIGH = 0,
  MSG_PRIORITY_MID = 1,
  MSG_PRIORITY_LOW = 2,
  MSG_PRIORITY_PERIODIC = 3
};

struct PredictableAgentMsg
{
    uint16_t sourceMoteID;
    uint16_t lastSampleNumber;
    uint16_t channel;
    uint16_t priority;
    uint16_t data;
};
```

Figure 8 : Msg Header

```
atomic {
  predicting_info = 0;
  threshold_info = THRESHOLD;
  irregular_info = IRREGULAR;
  pastSensing_info = 0;
  periodicCount = PERIODIC;
  readingNumber = 0;
  currentMsg = 0;
}
```

Figure 9 : Predictable agent system's initialization Code.

```
async event result_t ADC.dataReady(uint16_t data) {
  struct PredictableAgentMsg *pack;
  atomic {
    pack = (struct PredictableAgentMsg *)msg[currentMsg].data;
    predicting_info = ALPHA*predicting_info + (1-ALPHA)*data;

    if (data > threshold_info) {
      pack->priority = MSG_PRIORITY_HIGH;
      pack->data = data;
      pastSensing_info = data;
      periodicCount = PERIODIC;
      post dataTask();
    }
    else if ((data - pastSensing_info) > irregular_info) {
      pack->priority = MSG_PRIORITY_MID;
      pack->data = data;
      pastSensing_info = data;
      periodicCount = PERIODIC;
      post dataTask();
    }
    else if (predicting_info > threshold_info) {
      pack->priority = MSG_PRIORITY_LOW;
      pack->data = data;
      periodicCount = PERIODIC;
      post dataTask();
    }
    else if (periodicCount <= 1) {
      pack->priority = MSG_PRIORITY_PERIODIC;
      pack->data = data;
      periodicCount = PERIODIC;
      post dataTask();
    }
    periodicCount--;
  }
  if (data > 0x0200)
    call Leds.redOn();
  else
    call Leds.redOff();

  return SUCCESS;
}
```

Figure 10 : Main Process NesC Code

Figure 10 is Predictable agent system's main process code. Is process that do not transmit if decide each message type by priority and condition does not fit. periodicCount variable. transmits present state in schedule cycle.

4. Discussion

We have implemented the above mote predictable agent systems over TinyOS and embedded our implementation on MicaZ motes. We used the implemented agent, with t around 260 seconds, to measure the packet transmission

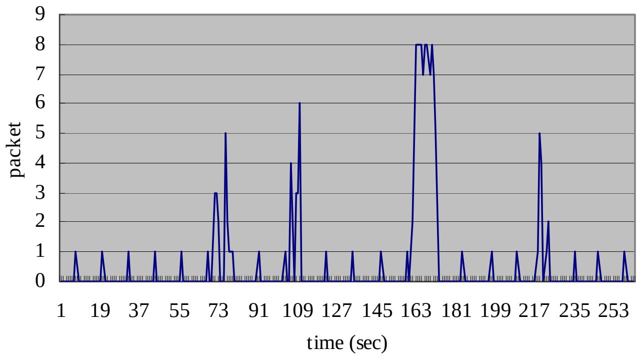

Figure 11 : Implementation Result

Figure 11 is value that measure the packet transmission rate by time. this shows that important information is transmitted by high transfer rate. There are three high rate terms at 73, 109 and 217 seconds. These have priority of MSG_PRIORITY_MID or MSG_PRIORITY_LOW. And the most highest term at 163 seconds which priority is MSG_PRIORITY_HIGH, implies that critical events occurred.

5. Conclusion

The Implementation of predictable agent system on TinyOS with the MicaZ platform represents a promising advance and enhances the power-consumption and provides the context-aware in the wireless sensor networks. However, if surrounding situation changes frequently, performance drops preferably. This predictable agent system secures performance improvement if surrounding environment does not change frequently.

Possible future development work includes enhancement of predictable agent system on TinyOS with the addition of powerful prediction and the incoporation of secure independent environment. And is going to develop by agent system that support the localization and support the programmability[10][11].

REFERENCES

[1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Pister, System Architecture directions for network sensors, ASPLOS 2000.

[2] Sohrabi, K., Gao, J, Ailawadhi V. Pottie G.J., "Protocols for self-organization of a wireless sensor network", IEEE personal communications, 2000

[3] A Protocol Suite for Wireless Sensor and Actor Networks (invited). Ian F. Akyildiz, Ismail H. Kasimoglu, Georgia Institute of Technology,. Atlanta, GA, USA 19-22 September 2004

[4] Yang Yu, Bhaskar Krishnamachari, Viktor K.Prasnna, "Issues in designing middlware for wireless sensor networks", IEEE Network, January 2004.

[5] Jukka Riekki, Jouni Huhtinen, Pekka Ala-Siuru, Pettteri Alahuhta, Jouni Kaartinen, Juha Roning, "Genie of the Net: Context-aware Information Management"

[6] Operationg System Concepts Sixth Edition. SilBersCharz, Galvin, Gagne

[7] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for ”"smart dust”". In Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking, pages 271 – -278, 1999.

[8] TinyOS. http://today.cs.berkeley.edu/tos/.

[9] http://www.tinyos.net/tinyos-1.x/doc/tutorial/lesson6.html

[10] Boulis A., Han C., and Srivastara M., Design and implementation of a framework for efficient and programmable sensor networks, in Proc. MobiSys 2003, San Francisco, CA, May 6-8 2003.

[11] Boulis A., Srivastava M.B., A framework for efficient and programmable sensor networks, in Proc. OPENARCH 2002, New York, June 2002.