Proceedings of the 2006 IEEE
International Conference on Information Acquisition
August 20 - 23, 2006, Weihai, Shandong, China

A Mobile Agents-Based Real-time Mechanism for Wireless Sensor Network Access on the Internet

Jieyin Bai
Shenyang Institue of Automation, CAS
Graduate School of the Chinese Academy of Sciences
Shenyang 110016
bai.jieyin@sia.cn

Chuanzhi Zang, Tianran Wang and Haibin Yu
Shenyang Institue of Automation, CAS
Chinese Academy of Sciences
Shenyang 110016

Abstract **- Due to the variety of applications and their importance, wireless sensor networks (WSNs) will need to be connected to the Internet. Some approaches have been proposed to connect wireless sensor networks to the existing TCP/IP networks, such as the application-level gateways or overlay networks. However, most existing approaches have to consume network bandwidth and node energy to maintain the static network structure which is neither scalable nor reliable. In this paper, we describe the Mobile agent Based Real-time (MBR) mechanism which use of the mobile software agent (MSA) paradigm to design a dynamic infrastructure for WSNs access on the Internet. We present a agent migration protocol based on *reinforcement learning* method to reduce the query delay and improve the total performance.**

Index Terms – Wireless sensor network, internet, mobile agent.

I. INTRODUCTION

Wireless sensor networks (WSNs) may consist of thousands to millions of tiny sensor nodes, with limited computational and communication capabilities. When networked together, these unattended devices can provide high-resolution information about sensed phenomena. Sensor nodes may transmit their readings to a gateway, which is responsible for forwarding the data from the sensor network to the remote base-station or database server through the Internet. Any researchers around the world could access the database server or perform remote interactions [1][2][3].

For connection between WSNs and the Internet, some crucial issues may need to be considered:

Power consumption. As a microelectronic device, the wireless sensor node can only be equipped with a limited power source. One limitation of the gateway-approach is that, a lot of data has to be routed from and back to the gateways, therefore, the high network load will make the nodes nearby the gateways exhaust their energy quickly and the whole network's lifetime will be shortened.

We plotted the average power consumption of sensor nodes in WSNs of a certain scale, as shown in Fig.1. This figure shows that the average energy consumption of the gateways is much more than the average energy consumption of their one-hop neighbors and all sensor nodes.

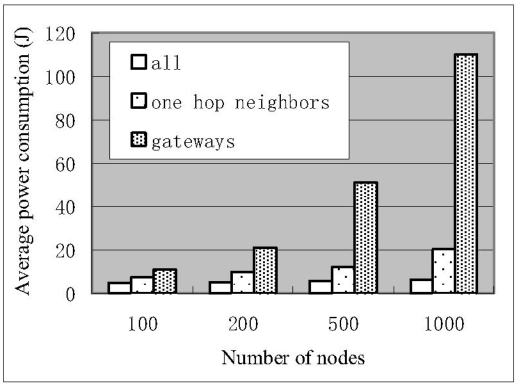

Fig.1. Average power consumption Versus number of sensor nodes.

Scalable. Marco Z etc recommend the use of overlay IP networks as an approach for integration between WSNs and the Internet. The IP network is constructed on top of underlying WSNs using a flooded-query approach (Directed Diffusion) or a directed-query approach (ACQUIRE). We consider that, to create an overlay IP network, it will be necessary to deploy WSNs with some more capable nodes with IP address. Moreover, the mapping between overlay network and WSNs is required. Such overlay solution may make the WSNs more expensive and complex.

Robustness. The task of connecting WSNs to the existing networks also brings with it another challenge: the nodes in unattended large-scale sensor networks are likely to be battery powered, with limited recharging capabilities. So the sensor nodes are prone to failures and the topology of a sensor network may change frequently. Some sensor nodes (e.g. the gateways, the nodes nearby the gateways, or the overlay network nodes) may fail or be blocked due to lack of power, and all nodes may have physical damage or environmental interference. The failure of gateway or few key sensor nodes on the virtual path of overlay network may affect the overall connection between the WSNs and the Internet. The re-building of overlay network will consume bandwidth and generate high delay.

The remainder of the paper is organized as follows. In Section II, we present a review of related work in integrating WSNs with the Internet. We formally define the mobile software agent-based infrastructure in Section III. Finally, the paper is concluded in Section VII.

1-4244-0529-7/06/$20.00 ©2006 IEEE

II. RELATED WORK

Most WSNs may have to be connected to an external network, through which the sensor reports can reach the final users and controlling messages can reach the sensor networks. We now focus on some approaches to unifying WSNs with the Internet in more detail and the mobile agent technology:

A. TCP/IP in WSNs

The ubiquity of TCP/IP has made it the de-facto standard protocol suite for most existing networking such as the Internet. It is envisioned that any network wishing to be connected to the Internet may need to address the question of how it will interface with the standard TCP/IP.

One of the feasible methods is to run TCP/IP in the wireless sensor network, which make it possible to directly connect the wireless sensor network with an external network infrastructure, without proxies or middleware [4][5]. In a TCP/IP sensor network, sensor data could be sent using the best-effort transport protocol UDP, and the reliable byte-stream transport protocol TCP would be used for administrative tasks. Reference [5] presented a number of optimization mechanisms to enable the use of TCP/IP for wireless sensor networks: application overlay routing, spatial IP address assignment, shared context header compression and distributed TCP caching (DTC). There are still, however, a number of problems that needs to be solved before TCP/IP can be a viable protocol for WSNs.

B. IP network overlaying WSNs

Another method is to construct an overlay IP network on top of underlying wireless sensor network. The method is similar to the application overlay routing which use link local IP broadcast that providing a direct mapping between the overlay network and the underlying wireless sensor networks.

Reference [1] used Directed Diffusion or ACQUIRE to build an overlay IP network. The technical challenge in this method is to construct a virtual tunnel that is abstracted from the multiple hops between nearby IP-addressable sensor nodes. If the IP-traffic inside the wireless sensor network is high, multiple paths would be constructed in order to load balance the power consumption in the non-IP nodes. DD is a data-centric communication paradigm that fits the multiple-paths-building task well. First, the *interest* of sink node will be distributed through the network. The interest description could be described as:

type: IP-addressable //detect IP-addressable nodes
interval: 0.1s //send message every 0.1s
duration: 5s //send total 50 messages

As the interest is propagated, the node set up gradients from the source back to the sink. Upon reception, the sources that have IP address will reply the interest. The reply description could be described as:

type: IP-addressable //IP-addressable sensor node
address: IP-address // IP-address of replying node

This reply message is sent back through the interest's gradient path. After the sink node received the reply, it must reinforce the most efficient paths (for this example, the IP tunnel). Because the sink node must refresh and reinforce the paths periodically, the amount of energy consumed by the network is high.

C. WSNs overlaying IP network

In this approach, WSNs packets are encapsulated into IP packets at the gateway and then directed from the gateway to any users on the Internet. This application-level overlay of WSNs over the Internet allows the remote users to query and interact with the WSNs directly via the transparent gateway [6].

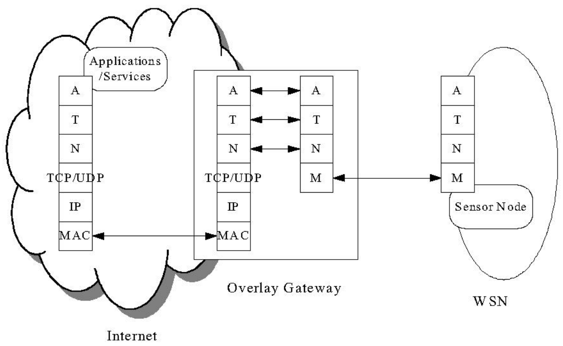

Fig.2. Operation of the WSN-overlaying-IP gateway

(A=application layer, T=transport layer, N=network layer, M=MAC layer)

D. Mobile Agent

There are two types of mobile agents being discussed in wireless sensor networks: *hardware agent* and *software agent*. Lang T etc [7] propose SENMA, a new network architecture for low power and large scale sensor networks. Mobile agents in SENMA are hardware units, like manned/unmanned aerial vehicles, ground vehicles equipped with sophisticated terminals and power generators, or specially designed light nodes that can hop around in the network.

In comparison with hardware agent, software agent is a special kind of software with dedicated instruction and data memory. Mobile software agent (MSA) can migrate from node to node, interact with the environment, and perform data processing autonomously. When a MSA roams through the network, the original dispatcher, which may be a mobile host, can disconnect from the network. The connection to the network is only required when launching mobile agent and fetching results from them. MSA have been used for many yeas on the Internet. Some middleware systems that utilize agents include Agent Tcl, Java Aglets, Ara, and Mole. They have been successfully used in data mining, e-commerce, and network management applications.

The MSA systems have been shown to be dynamic, intelligent and fault tolerant [8][9]. Due to these characters of mobile software agent and the inadequacies of overlay network approach to unifying WSNs with the Internet, we propose a protocol-independent MSA-based mechanism.

The common communicational model of WSNs is the sink/sensor model (Fig.3). In sink/sensor model, the sink node (e.g, the overlay gateway) sends requests to the sensor nodes by single-hop or multi-hop communication and then waits for the replications. This communicating model is "silly", since each node must keep at least one link alive to its neighbor after the initialization and the high-level nodes may have to maintain more complicated information. Many protocols of WSNs based on the common model have been shown to be **power aware**, however, these one-to-many and many-to-one data-flow patterns may not be appropriate for WSNs access on the Internet.

First, large amounts of data are moved around the network and the network traffic can be extremely heavy. This is especially unfit for real-time applications and low-bandwidth wireless connections due to the heavy collision. We consider, different users on the Internet may want to query the real-time sensor readings in different ways. Therefore, there will be a lot of redundant sensor data to be transmitted to the gateways.

Second, the performance of the network is seriously affected by the reliability and the scale of WSNs. If the multi-hop path between the sink node and the remote sensor nodes go down which may happen frequently in WSNs, both the sink node and the sensor nodes have to find alternative path or re-build routing tables until the connection is recovered.

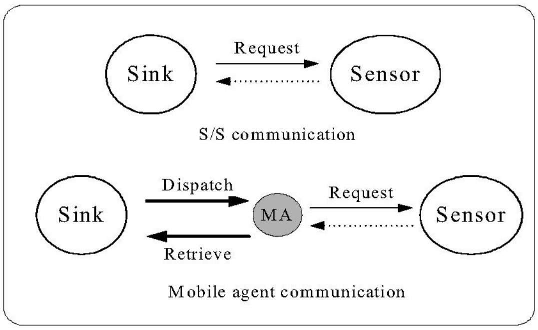

Fig.3. the sink/sensor model and the MSA model

III. MOBILE AGENT-BASED INFRASTRUCTURE

In this paper, we design an infrastructure using MSA, which adopts a new computation paradigm: data stay at the local site, while the integration process (code) is moved to the data sites.

The Internet user issues an active query agent through the *processing element* (the gateway node). The query agent migrates step by step through a sequence of nodes via the network. The node carrying the query integrates updates received from all its neighbor nodes within a look-ahead of *D* hops. After the query agent has received all updates from *D* hops neighbors and utilized these information to answer as much of the complex query as possible, it will migrate to next node. Until the query agent completely resolve the query, the agent will carry the final result to the originating user.

Our infrastructure has some similarities in spirit to ACQUIRE [10]. We use a new algorithm based on reinforcement learning to reduce the processing delay and improve the total performance.

A. Basic Model

In our infrastructure, users insert a query agent with complete task and integration process into the network through the gateway node. Instead of passing large amounts of raw data over the network, only the query agent with small size is sent.

We can define the agent as an entity of the following five attributes: **Identification**, **Query**, **Data**, **Itinerary** and **Gain**. These attributes are defined as follows.

- **Identification** is in the format of 3-tuple (*a*, *b*, *c*), where *a* indicates the identification number of its dispatcher and *b* the serial number assigned by its dispatcher and *c* the level of agent (master or worker).
- **Query** is the implementation of the agent algorithm, including query and trajectory. It defines the *coverage* parameter to describe the look-ahead hops of query agent. In our infrastructure, *coverage* is not a constant. It is modified continually to achieve maximum diffusion gain.
- **Data** is the query agent's data buffer which carries the results.
- **Itinerary** includes strategy of migration and information about the passed nodes
- **Gain** is the query agent's calculating buffer which carries the last two step query diffusion gain of each neighbor node. The definition of diffusion gain will be discussed later.

B. Query Diffusion Algorithm

Before the description of the algorithm based on the *reinforcement learning* method, we provide two definitions:

Definition 1: **Data Quality - Q.** Generally, the more sensor nodes visited, the higher the accuracy achieved by using any reasonable data fusion algorithm [8]. So the **Data Quality** of the result of current step query can be approximated as the function of the numbers of neighbor nodes within *N* hops. If sensor nodes are uniformly distributed, the numbers of neighbor nodes within *N* hops will be the function of *N*.

$$Q = f(N) \tag{1}$$

Definition 2: **Diffusion Gain - G.** As the coverage area increase (we call that query diffusion) both the communication costs and data quality increase.

$$G_s = W_q Q_s - W_c C_s \tag{2}$$

where s is the step count of the query agent propagation, Q_s is the value of the data quality of current step, C_s is the value of the total costs including processing delay and energy consumption of current step, W_q and W_c expresses the weights of the data quality and the costs respectively.

There are three phases of the query diffusion: **GAIN**, **BALANCE** and **DECLINE**. When we first inject the query agent into the network, the initial *coverage* is set to one that means single-hop diffusion. When the query agent arrives the second node, *coverage* will increase automatically. After the query agent has received all data from neighbor nodes in the coverage area, it will calculate the Diffusion Gain.

Our algorithm use the *Monte Carlo methods* of reinforcement learning to setup the *coverage*. A general every-visit Monte Carlo method suitable for nonstationary environments is:

$$V(s_t) \leftarrow V(s_t)+a[R_t-V(s_t)] \quad (3)$$

where $V(s_t)$ is the estimative object, R_t is the actual return following time t and a is a constant step-size parameter. Based on this method, we design a self-adaptive algorithm which is described as:

- If $G_s>0$ and $G_s>G_{s-1}$, the query diffusion is in the **GAIN** phase and *coverage* keeps increasing;
- If $G_s>0$ and $G_s<=G_{s-1}$, the query diffusion is in the **BALANCE** phase and *coverage* stops increasing;
- If $G_s <= 0$, the query diffusion is in the **DECLINE** phase and *coverage* starts decreasing.

C. Migration Strategy

In ACQUIRE, the query agent propagates through the sensor network and follows a random (or guided) path until it is completely resolved [10]. Then the query agent will return back to the gateway node along the reverse path. We present a heuristic strategy for the query agent routing.

After the query agent has received all data from neighbor nodes in the coverage area, the individual Diffusion Gain can be calculated based on the data quality and processing delay of each neighbor node. Let $G = \{G_1, G_2, G_3 \ldots G_N\}$ be the N gains of neighbor nodes, the query agent can choose the one with maximum gain as next node to forward. If none of the neighbor nodes in coverage area has matched data, the query agent will choose a node randomly.

IV. SIMULATION

In our unlimited energy simulation experiments, the network was randomly generated with the constraint that the graph be fully connected and the number of gateways was set up dependent on the scale of WSN. The radio speed (1 Mbps) and the energy dissipation (600mW in transmit mode and 200mW in receive mode) were chosen based on real data. As the common scenario, we used DD as routing protocol for the Network layer and 802.11 for the MAC layer. DD was also in charge of building the paths from sensor network to the gateways. For simplification, we employed one gateway every 300 nodes.

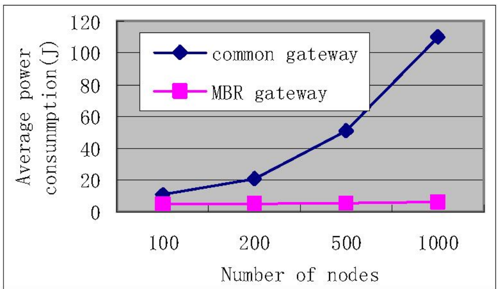

Fig.4. The MBR gateway use the energy at a slower rate and thus are able to remain operational for a longer period of time.

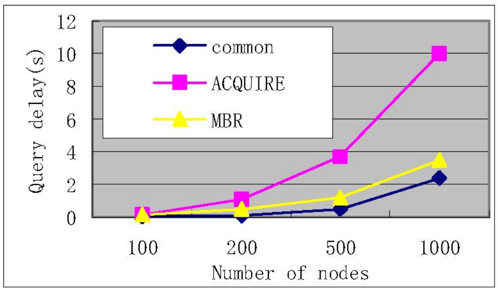

Fig.5. The MBR's query delay is higher than the common sink/sensor model and much lower than ACQUIRE.

V. CONCLUSIONS

In this paper, we have proposed a novel mechanism based on mobile software agent for WSN access on Internet. The key feature of our mechanism is the usage of *reinforcement learning* method to reduce the processing delay and improve the total performance.

The experiments show that, by applying the MBR mechanism, more reliable interaction between WSN and Internet can be achieved. Compared to other mechanism based on mobile agent for the moment (such as ACQUIRE), the MBR mechanism shows lower delay and lower energy consumption that is more applicable for unifying WSNs with the Internet. We believe the MBR is likely to perform on complex, one-shot queries which are more general in the Internet environment.

We intend to research the migration strategy further in the next phases of the study. The heuristic strategy performs better than the random choice on the real-time query, but may run into the local optimization. Now we focus on finding a solution to the problem by the *Ant Colony* algorithm.

REFERENCES

[1] Marco Zuniga Z, et al, "Integrating Future Large-scale Wireless Sensor Networks with the Internet", USC Computer Science Technical Report CS 03-792, 2003.

[2] Philippe Bonnet, et al, "Towards Sensor Database Systems", In Proceedings of the Second International Conference on Mobile Data Management. Hong Kong, January 2001.

[3] Ian F. Akyildiz, et al, "A Survey on Sensor Networks", IEEE Communications Magazine August 2002 102.

[4] S. Mishra R. Sridharan, R. Sridhar. "A robust header compression technique for wireless ad hoc networks". In *MobiHoc 2003*, 2003.

[5] A. Dunkels, J. Alonso, and T. Voigt, "Making TCP/IP Viable for Wireless Sensor Networks," in Proc. EWSN 2004, January 2004.

[6] Hui Dai, Richard Han, "Unifying Micro Sensor Networks with the Internet via Overlay Networking". In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks. 571-572.

[7] Lang Tong, et al, "Sensor networks with mobile agents". In Proceedings of. IEEE MILCOM'03, Boston, MA, October 2003.

[8] Hairong Qi, et al, "Multiresolution Data Integration Using Mobile Agents in Distributed Sensor Networks", IEEE Transactions on Systems, Man, and Cybernetics—PART C: Applications and Reviews, Vol. 31, NO. 3, august 2001.

[9] Wang Yan, et al, "A Mobile Agent based System for Distributed Database Access on the Internet", Communication Technology Proceedings, International Conference on, Volume: 2, P:1587 -1590 vol.2. 2000.

[10] N. Sadagopan, et al, “The ACQUIRE mechanism for efficient querying in sensor networks”, First IEEE International Workshop on Sensor Network Protocols and Applications, May 2003, Anchorage, AK, USA.