國立交通大學

網路工程研究所

碩 士 論 文

在無線感測網路中針對分散式多重資料源與多重目的地資料傳輸的行動代理人模型

A Mobile Agent Model for Distributed Multi-Source, Multi-Destination Data Transmission in a Wireless Sensor Network

研 究 生：林育萱

指導教授：曾煜棋　教授

易志偉　教授

中 華 民 國　九 十 八　年 六 月

在無線感測網路中針對分散式多重資料源與多重目的地資料傳輸的行動代理人模型

A Mobile Agent Model for Distributed Multi-Source, Multi-Destination Data Transmission in a Wireless Sensor Network Cooperative Compound Document Editing System

研 究 生：林育萱　　Student：Yu-Hsuan　Lin

指導教授：曾煜棋　　Advisor：Prof.　Yu-Chee Tseng

易志偉　　Prof.　Chih-Wei Yi

國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

A Mobile Agent Model for Distributed Multi-Source, Multi-Destination Data Transmission in a Wireless Sensor Network

Student: **Yu-Hsuan Lin**

Advisor: **Prof. Yu-Chee Tseng**

Prof. Chih-Wei Yi

Department of Computer Science

National Chiao-Tung University

1001 Ta Hsueh Road, Hsinchu, Taiwan 300, R.O.C.

在無線感測網路中針對分散式多重資料源與多重目的地資料傳輸的行動代理人模型

學生：林育萱　　　　　　　　　　　　　　　　指導教授：曾煜棋 教授
易志偉 教授

國立交通大學網路工程研究所碩士班

摘　　　要

在無線感測網路中，資料傳輸型態可以分成一對一、多對一、一對多以及多對多這幾種傳輸型態。在本篇論文中，我們探討了無線感測網路中，多對多的資料傳輸型態。在這種多對多的傳輸中，有許多目地點，而每個目的點都有一個目標函式，而此目標函式需要得到某些資料源的感測資料，以進行目標函式的運算。當感測資料在網路中傳輸時，我們透過資料群播與聚集的技術來同時且有效地計算這些目標函式。

在本篇論文中，我們針對分散式網內群播及聚集技術，提出了一個行動代理人模型。一個行動代理人存在於一個邏輯角色中，其能自動地指示該點的群播及聚集運算，透過此模型，我們能減少網路中資料傳輸的花費。而為了達到最佳化目的，每個邏輯角色在網路中會進行移動、分裂成多個邏輯角色、或者和其他邏輯角色進行結合。而行動代理人模型是採分散式的方式運作在網路中。透過模擬的結果，可以顯示出行動代理人模型在無線感測網路中多對多資料傳輸的顯著效能。

關鍵字：資料聚集、多對多資料傳輸、資料群播、無線通訊、無線感測網路。

A Mobile Agent Model for Distributed Multi-Source, Multi-Destination Data Transmission in a Wireless Sensor Network

Student: Yu-Hsuan Lin Advisor: Prof. Yu-Chee Tseng
Prof. Chih-Wei Yi

Department of Computer Science
National Chiao-Tung University

ABSTRACT

Communication patterns in a wireless sensor network (WSN) can be classified as *one-to-one*, *one-to-many*, *many-to-one*, and *many-to-many*. In this paper, we consider the most general many-to-many communication in a WSN. In such many-to-many communications, each destination has an objective function, which requires inputs from multiple source nodes, and there are multiple destinations each with its own defined objective function. In order to compute these objective functions efficiently and simultaneously, sensing data may be multicast and aggregated while being transmitted in the WSN. We propose a *mobile agent model* for distributed in-network multicast and aggregation to reduce the total induced communication cost, where a mobile agent in a logical role which can conduct multicast/aggregation operations autonomously. A logical role can also migrate itself around, split into multiple roles, or merge with other roles for optimization purposes. The scheme works in a distributed manner. Extensive simulations are conducted to verify our results.

Keywords: aggregation, many-to-many communication, multicast, wireless communication, wireless sensor network.

誌謝

首先我要誠摯地感謝我的指導教授曾煜棋老師，在這兩年的研究生涯中給予了我很多的指導以及幫助，在老師的指導下，我學到了許多做研究的方法與態度，以及論文寫作方面的技巧，也因此讓我能順利地完成了這篇論文。同時我也要感謝共同指導教授易志偉老師給予的指導與建議，以及口試委員沈建中教授、許健平、呂學士教授在口試中所給予的寶貴意見，讓這篇論文更加完整。

此外，我也由衷地感謝同組的葉倫武學長，在這兩年的學習中，當我遇到困難時能適時地給予我指導及幫助。最後要感謝 HSCC 實驗室全體的成員，不論是在研究上或者其他方面都讓我受益良多，我很珍惜和大家相處這兩年的點點滴滴。

最後，感謝我的父母及家人，在我做任何決定時都給予支持與關懷，使我能夠無虞地完成我的學業。

在此向大家獻上我誠摯地謝意跟祝福

林 育 萱 謹識於

國立交通大學網路工程研究所碩士班

中華民國九十八年六月

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid progress of wireless communication and embedded MEMS technologies has made *wireless sensor networks* (WSNs) possible. A WSN normally consists of many inexpensive wireless nodes, each capable of collecting, processing, storing environmental information, and communicating with neighboring nodes. Many WSN applications have been developed, such as emergency guiding [1][4], object tracking [3][10], and smart living space [5][8].

Communication patterns in a WSN can be classified as *one-to-one*, *one-to-many*, *many-to-one*, and *many-to-many*. For example, issuing queries to multiple sensor nodes is one-to-many communication and collecting data from multiple nodes is many-to-one communication (or convergecast). In this paper, we consider the most general many-to-many communication, where there are multiple sources and multiple destinations. Each destination has its own objective function, which requires inputs from multiple sources. Different destinations have different objective functions requiring inputs from different sources. To minimize the incurred communication overheads, relay nodes may need to conduct multicast and aggregation on sensing data at proper places. This problem is quite challenging because when and how to multicast/aggregate sensing data has big impact on performance. Fig. 1.1(a) shows a smart surveillance scenario with several rotatable cameras, light sensors, sound sensors, and pressure sensors. Camera C_1 will be activated if the objective function $F_{C_1} = (L_1|L_2)\&(S_1)\&(P_1|P_2|P_3)$ is true, C_2 will be activated if $F_{C_2} = (L_1|L_3|L_5)\&(S_2)\&(P_2|P_3|P_4)$ is true, and C_3 will be activated if $F_{C_3} = (L_2|L_4)\&(S_1|S_3)\&(P_1|P_3|P_5)$ is true(here we assume that a sensor will return a TRUE value if it detects some changes). where a sensor will be triggered and return a TRUE value if it detects some changes. We are interested in efficiently transmitting the required sensing data to each camera to properly activate them. Fig. 1.1(b) shows a scheduling which requires 38 transmissions, while Fig. 1.1(c) shows an alternative which only requires 26 transmissions.

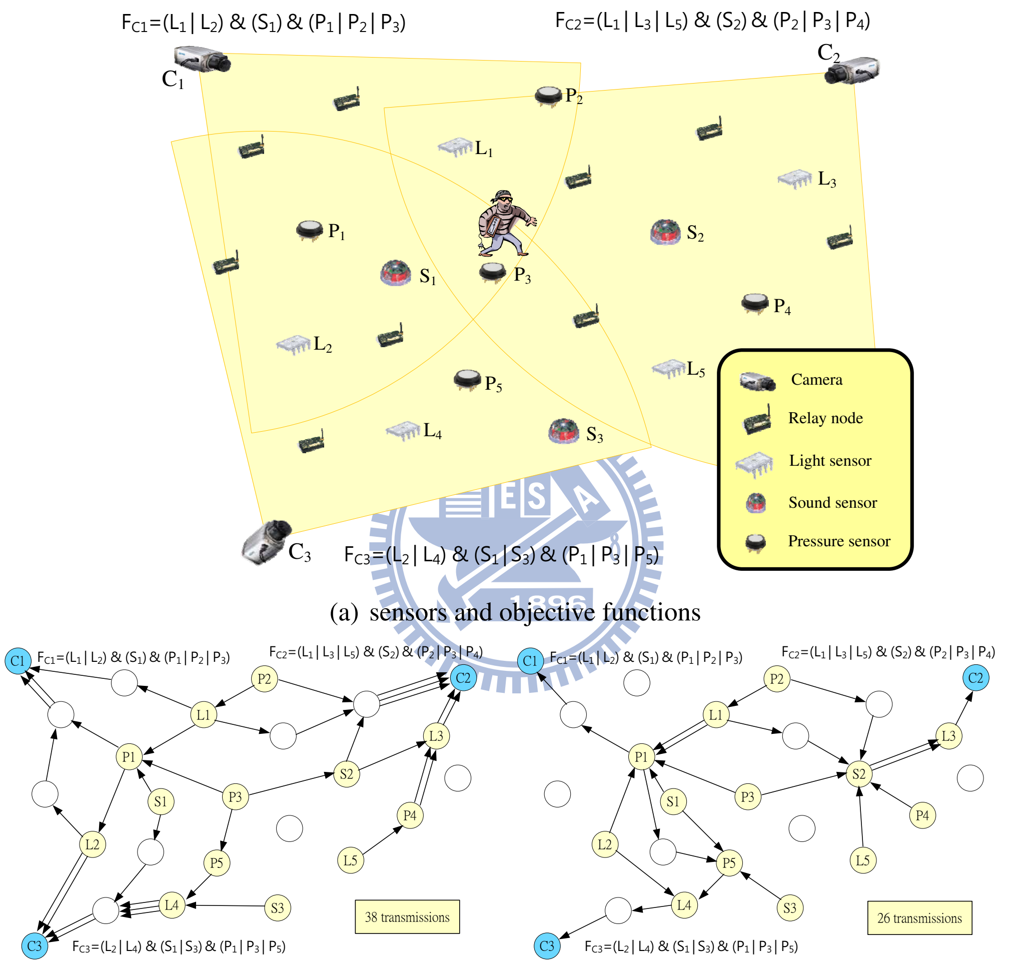
Several works [6][7][9] propose some schemes can be used for many-to-many communications. In reference [6], the authors propose a communication architecture to support multiple mobile users in a wireless sensor network. Each mobile user needs different kinds of sensing data in a wireless sensor network. All static sensor nodes form a tree topology rooted at the sink and each mobile user connects to the nearest static sensor nodes. All sensor nodes report data to the sink and the sink disseminates the necessary data to a set of static sensor nodes those are the nearest mobile users. It is not energy-efficient because all nodes must report sensing data to the sink. This scheme can not be adopted to our scenario because it needs a central sink to collect all sensing data. The work [9] proposes an aggregated hierarchical multicast algorithm to reduce the amount of data within many-to-many communication scheme. The work only focuses on how the data packets can be efficiently aggregated while providing bounds on delay jitter. Hence, the work can not be used for many-to-many communication in a wireless sensor network. The work [7] proposes a many-to-many aggregation algorithm in a wireless sensor network. According to the predefined aggregation function, the algorithm can obtain the transmission plan for each network edge. However, this work does not describe how to obtain the routing path from each source node to sink node. Hence, in our work, we propose a distributed algorithm to find the routing paths from each source to each sink to minimize the network cost according to the aggregation functions.

Other works focus on one-to-many or many-to-one communications in wireless sensor network. MVsink [2] proposes four many-to-one algorithms to reduce the network cost by incrementally building aggregation trees in the network. However, those algorithms can not support generalized aggregation functions. Since the work just considers about many-to-one communications, there are still many duplicate aggregation requirements to waste network resources when it extends to many-to-many communications.

Our proposed mobile agent model takes a view of logical plane in such many-to-many communications. We call a mobile agent as a logical role which can be created, terminated, moved and merged in the network. There are four types of logical roles in this model, such as aggregation role, multicast role, destination role, and source role.

At first, a destination role, requiring some sensing data from source role, processes its objective function into a virtual aggregation tree structure which is composed of those logical roles. Then, the destination role floods an OBJ_REQUEST message with the information of its objective function. The required source roles or suitable existing aggregation and multicast role will return OBJ_REPLY messages. According to the return messages, the destination role can reduce the aggregation roles in the virtual aggregation tree. Then the destination role creates the aggregation roles and multicast roles in the network and binds them to proper node. Those logical multicast and aggregation roles make a rule to route and aggregate the sensing data in the network. In the final step, the destination role triggers those logical roles to start updating. Those logical roles will move or merge themselves to the better candidate nodes dynamically. Consequently, the routing path will gradually change to reduce the communication cost.

The rest of this paper is organized as follows. Section 2 formulates our problem. Section 3 presents our proposed algorithm. Simulation results are given in Section 4. Finally, Section 5 concludes this paper.



(a) sensors and objective functions

(b) A simple scheduling

(c) A more efficient scheduling

Figure 1.1: A sensor-triggered surveillance scenario involving many-to-many communication.

Chapter 2

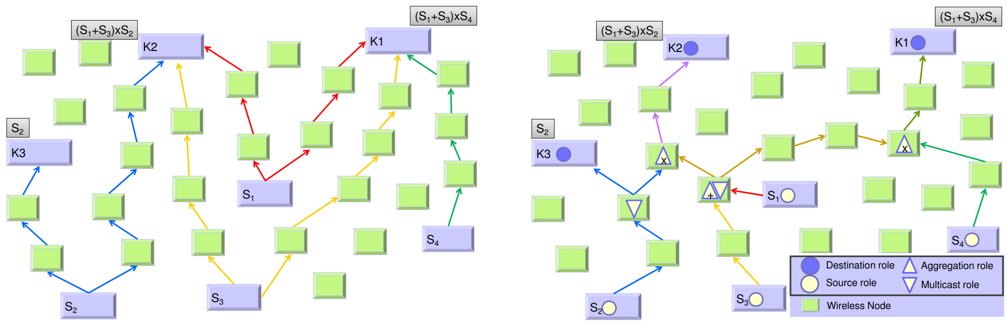
Problem Statement

We are given a WSN, on which many-to-many communication needs to be executed. There are n source nodes $S_1, ..., S_n$ and m destination nodes $K_1, ..., K_m$. Each destination node $K_i, i = 1 \ldots m$, has an objective function F_i, which requires inputs from some source nodes to compute its value. We assume that these functions contain operands connected by simple operators such as $+$, $-$, $\times$, $/$, $|$, &, min, max, sum, etc. Whenever needed, parentheses are used to clarify precedence. We assume that sources all generate data with the same period for the destinations to compute their functions. Our goal is to minimize the amount of data transmissions for such many-to-many communication.

Chapter 3

Mobile Agent Model for Many-to-Many Communication

We propose a *mobile agent model* to solve the many-to-many communication problem. A mobile agent is a *logical role*. We define four logical roles. A logical role can be created, moved, merged, and terminated under our mobile agent model. Mobile agents work like an overlay network to multicast/aggregation sensing data to reduce communication overheads. For one destination, the overlay network is like a logical tree, and for multiple destinations, it looks like a mesh. We show how mobile agents autonomously migrate themselves around to gradually reduce the induced data transmission costs. We also address how to add and delete an objective function in the network. Fig. 3.1 shows an example of a many-to-many communication, where K_1, K_2, and K_3 are three destination nodes. Fig. 3.1(a) shows a general data transmission with shortest path manner for the objective functions. Fig. 3.1(b) shows another result of data transmission under mobile agent model. We can see that aggregation roles and multicast roles navigate data flows and reduce the number of data transmissions in the network.



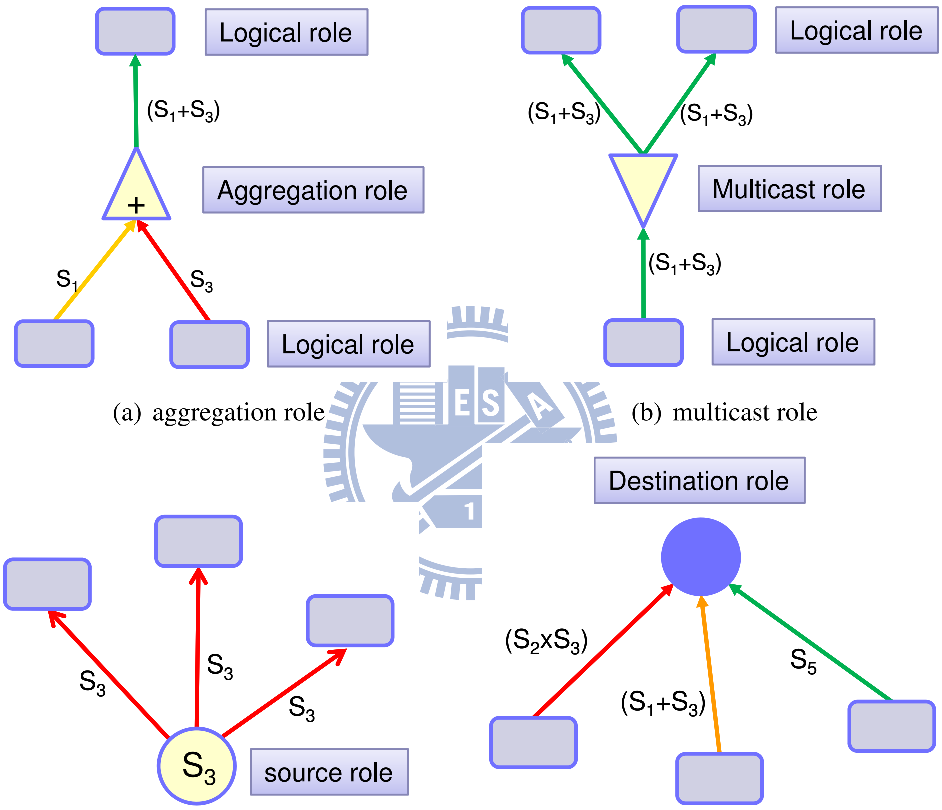
(a) A general shotest path manner (27 transmissions)

(b) Under mobile agent model (15 transmissions)

Figure 3.1: An overview example

Table 3.1: Logical roles.

	Aggregation Role	Multicast Role	Destination Role	Source Role
Movable	Yes	Yes	No	NO
Symbol	△	▽	●	○



(a) aggregation role

(b) multicast role

(c) source role

(d) destination role

Figure 3.2: Logical roles

3.1 Logical Roles

There are four types of logical roles in our model. A logical role should bind itself to a physical sensor node. However, a physical sensor node may be bound by multiple logical roles.

- Aggregation Role: An aggregation role takes inputs from multiple logical roles and combines them into one aggregated result as output to another logical role. The combination is through an aggregation operator. An aggregation role is movable.

- Multicast Role: A multicast role takes inputs from one logical role and duplicates it to multiple logical roles. A multicast role is also movable.

- Source Role: A source role has no input data but can transmit, when being subscribed by other logical roles, its sensing data to those logical roles. We assume that it will report its sensing data periodically. A source role is not movable.

- Destination Role: A destination role takes inputs from one or multiple logical roles to compute its objective function. A destination role is not movable. A new destination roles can be added at any time by specifying its objective function. After finishing, an existing destination role can be deleted.

Fig. 3.2 plots some examples of these logical roles. Note that two logical roles may be connected by a direct link or by a multi-hop route. A source role is triggered when it collects a piece of sensing data. A role of any other type is triggered when it receives all its required inputs.

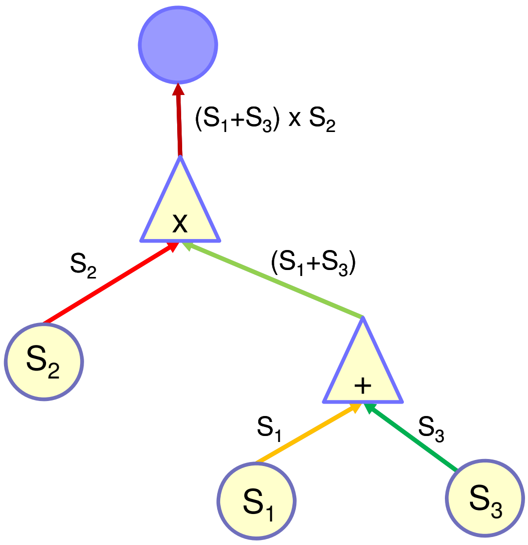


Figure 3.3: A logical aggregation tree in a logical scope, where the objective function is $F = (S_1 + S_3) \times S_2$.

3.2 Logical Aggregation Tree

The computation of an objective function can be regarded as a logical aggregation tree with sensing data being relayed and aggregated on their way to the destination role. In this mobile agent model, we decompose an objective function into multiple aggregation operations each as an aggregation role. As a result, for a destination role's point of view, there is a logical aggregation tree rooted at itself. Fig. 3.3 shows a example of a logical aggregation tree. The root is a destination role, all leaves are source roles, and the branch points are aggregation roles. Therefore, the destination role can compute the objective function when all required data from its upstream roles arrives.

3.3 Logical Aggregation/Multicast Mesh

In our mobile agent model, there are multiple logical aggregation trees in the network since there are multiple objective functions. The requested inputs of objective functions may overlap. Therefore, the logical aggregation trees may cross or share a subtree of each other in the network. Fig. 3.1(b) shows an example of three logical aggregation trees in the network. Those logical roles compose an aggregation/multicast mesh network in the logical scope. Each logical role only has to maintain the relationship between its input roles and output roles. For the reason, the functionalities of the mesh network won't be changed if we just move the logical roles without modifying the connections in the mesh network.

3.4 Maintenance of Logical Roles

In mobile agent model, we converts objective requirements into logical roles and then disseminates the logical roles among the network in a distributed manner. Each logical role will individually operate a updating process according to its type of logical roles. We define a sub-header for data communications between logical roles. Fig. 3.4 shows the structure of the sub-header. The $SrcID$ and $DestID$ are the pair of node ID and role ID ($NodeID, RoleID$) of a sender role and a receiver role. $TotalHops$ is the hop-count between the sender and receiver. $CurHops$ is the current hop-count from the sender. In a data transmission between two logical roles, a node which is on the routing path of the transmission and its 1-hop neighbor nodes will hear the data packet. According to the sub-header, these nodes can update its routes to the sender and receiver.

Type	SrcID	DestID	TotalHops	CurHops

Figure 3.4: Sub-header for data communications under mobile agent model.

Type	AggrID	Inputs num	In_1ID	cost_1	…	In_nID	cost_n

(a) AGGR_QUERY

Type	NodeID	Inputs num	In_1ID	…	In_mID	cost

(b) Type1 AGGR_REPLY

Type	AggrID	cost

(c) Type2 AGGR_REPLY

Type	McstID	Outputs num	Out_1ID	cost_1	…	Out_nID	cost_n

(d) MCST_QUERY

Type	NodeID	Outputs num	Out_1ID	…	Out_nID	cost

(e) MCST_REPLY

Figure 3.5: Control messages

3.4.1 Aggregation Role update

Aggregation roles trigger a updating procedure periodically after it is activated. For the race condition issue in the distributed system, an aggregation role requires to lock its input logical roles and the output logical role before it starts the updating process. First of all, the aggregation role A broadcasts an AGGR_QUERY message to its K-hop neighbors. The K-hop neighbors examine whether it can be a candidate node for the updating or not. A node C check which inputs of A it can connect. C find a maximum set $I_c \subseteq I$ to satisfy Eq. (3.1) where I is the inputs of A and $D(i, j)$ is the hop-count number between i and j. If C can find a I_C, C will become a type1 candidate node and send back a AGGR_REPLY message. According to the AGGR_REPLY messages, A determines which inputs should be aggregated on which type1 candidate nodes.

$$\sum_{j=1}^{n} D(i_{c_j}, C) + D(C, A) < \sum_{j=1}^{n} D(i_{c_j}, A), I_c = \{i_{c_1}, i_{c_2}, ..., i_{c_n}\} \qquad (3.1)$$

However, there might be an existing aggregation role which does the same aggregation operation and is near A. Hence, the existing aggregation role, $A_{existing}$, become a type2 candidate if it satisfies Eq. (3.2).

$$D(A_{existing}, A) < \sum_{i \in I} D(i, A) \qquad (3.2)$$

Type1 candidates are the better nodes to perform the current aggregation role. Hence, the current aggregation role can be moved or split to the candidate nodes to reduce the transmission cost. Nevertheless, a type2 candidate is an existing aggregation role which already performs the same aggregation operation in the network. That means it is worth to merge the two existing aggregation roles to a single aggregation role.

Both type1 and type2 candidates will transmit an AGGR_REPLY message to the current updating aggregation role. Then the aggregation role determines which type is better to process updating. For type1 updating, the aggregation role informs its input roles, output role and the candidate nodes to establish the routing relationship and then move or split itself to the candidate nodes. For type2 updating, the aggregation role informs its output role and the candidate role first. Then the current aggregation role gets destroyed and the candidate role takes a job to serve the output of current aggregation role. Since the candidate role has to serve two output logical roles after merging, a multicast role will be created to serve the two output logical roles. Fig. 3.6 shows an example of type2 updating.

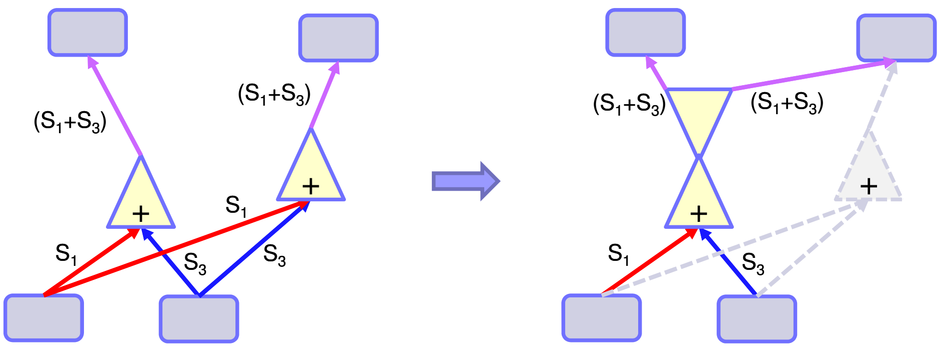


Figure 3.6: Type2 updating of the aggregation role .

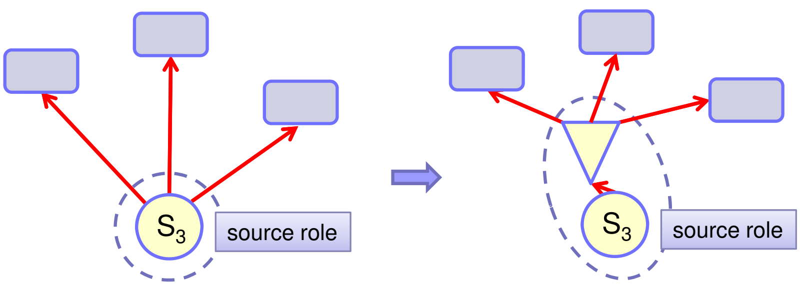


Figure 3.7: A source role creates a new multicast role on the current node.

Algorithm 1 Procedure $AggregationRoleUpdate()$ executed by an aggregation role periodically

Lock other corresponding roles
Broadcast AGGR_QUERY message to its K-hop neighbor nodes
Find the best updating candidates by AGGR_REPLY messages
if $Type2$ updating **then**
    merge this aggregation role to the candidate aggregation role
    and then terminate this role
else if $Type1$ updating **then**
    if require split **then**
        split the current aggregation role into multiple sub-aggregation roles
        move those roles to the candidate nodes
    else
        move current role to the candidate node
    end if
else
    (no candidates) sleep for a while
end if

3.4.2 Source Role update

A source role can serve more than one logical role. However, setting up multicast roles to relay the raw sensing data takes a chance to reduce the transmission costs. Therefore, a source role will periodically create a new multicast role to cope with its output logical roles if there are more than one logical role associated with it.

3.4.3 Multicast Role update

A multicast role processes an update procedure periodically. The updating procedure of a multicast role is similar to an aggregation role. Conversely, a multicast role, M, updates itself in the reverse direction of an aggregation role. First, M broadcasts MCST_QUERY message to its K-hop neighbors. A node which is received the message will verify routing costs to the outputs of M. A node, C, becomes a candidate node if it find a set O_c which satisfies the Eq. (3.3), where O is the outputs of M.

$$\sum_{j=1}^{n} D(o_{c_j}, C) + D(C, A) < \sum_{j=1}^{n} D(o_{c_j}, A), O_c = \{o_{c_1}, o_{c_2}, ..., o_{c_n}\}, O_c \subseteq O \quad (3.3)$$

After the candidate nodes return a MCST_REPLY message, M chooses the best candidate nodes to perform the current multicast role. Finally, M move or split inself to those candidate nodes. The new position of the multicast role makes a profit in the network since the transmission number of data packets is decreased.

Algorithm 2 Procedure $MulticastRoleUpdate()$ executed by a multicast role periodically

```
Lock other corresponding roles
Broadcast MCST_QUERY message to its K-hop neighbor nodes
Find the best updating candidates by MCST_REPLY messages
if receive MCST_REPLY message then
    if require split then
        split the current multicast role into multiple multicast roles
        move those roles to the candidate nodes
    else
        move current role to the candidate node
    end if
else
    (no candidates) sleep for a while
end if
```

3.5 Addition of an Objective Function

A destination role will be created when a physical node has an objective function in the network. At first, a destination role, K, floods an OBJ_REQUEST message with its objective function. Once a logical role can provide a part of the objective function, the logical role will send back an OBJ_REPLY contained which subset of objective function it can serve and the hop-count cost to the destination role. As a result, not only the requested source roles but also existing aggregation and multicast roles will inform K. K constructs a virtual logical aggregation

Algorithm 3 Procedure $AddObjectiveFuntion()$ executed by a destination role

```
Flood a OBJ_REQUEST message with the objective function
Construct a virtual logical aggregation tree according to the objective function
while not timeout do
    if receive OBJ_REPLY messages then
        Update the hop-count cost in the virtual tree structure
    end if
end while
Compute the minimal hop-count cost
Trim off the redundant branch points in the virtual tree
Create the aggregation roles for branch points in the network
Activate the aggregation roles
```

tree with the precedence of the objective function. According to OBJ_REPLY messages, K records the expected positions of leaves or branch points and the minimum hop-count cost from K to them. Subsequently, K computes the hop-count cost of each branch point. The hop-count cost of a branch point is the total cost of its sub-layer. Only the minimum cost will be kept if there is already a cost record by an OBJ_REPLY message. Fig. 3.8 shows an example of how to compute the cost of branch points. If the cost of a branch point is less than the total cost of the leaf points in the subtree, we trim off the subtree in the virtual tree. At the time, the remaining branch points in the virtual logical aggregation tree are the aggregation roles which K requires to create in the current network. Once the aggregation roles are created, the backbone of the logical aggregation tree is really constructed in the network. Finally, the destination role can obtain the objective result when the requested sensing data route along the logical aggregation tree in the network.

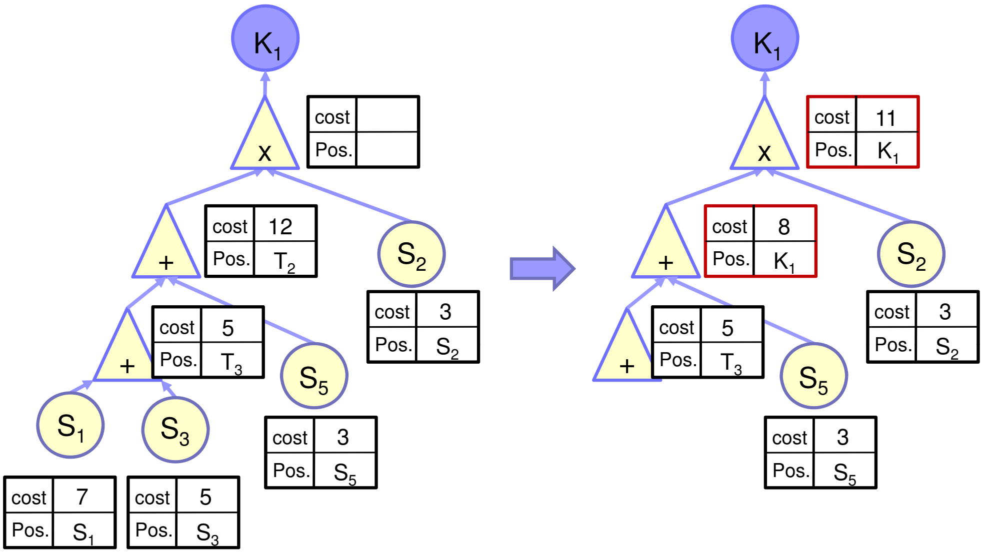


Figure 3.8: An example of how to compute the cost of branch points in addition process.

3.6 Deletion of an Objective Function

For deleting an existing objective function, the destination role has to broadcast an OBJ_CANCEL message to its associated logical roles in the network. The way to traverse the associated logical roles is forwarding the OBJ_CANCEL message in the reverse direction of the data packets until the current logical roles are multicast roles or source roles. An aggregation role will terminate itself and forward the message to next logical roles when it receives a OBJ_CANCEL message. A multicast role requires to inform its input role and another output role to link directly and then terminate itself without forwarding the OBJ_CANCEL message. However, when a source role receives an OBJ_CANCEL message, it only has to cancel the output to the forwarder of the OBJ_CANCEL message. Fig. 3.9 shows an example of deleting an objective function.

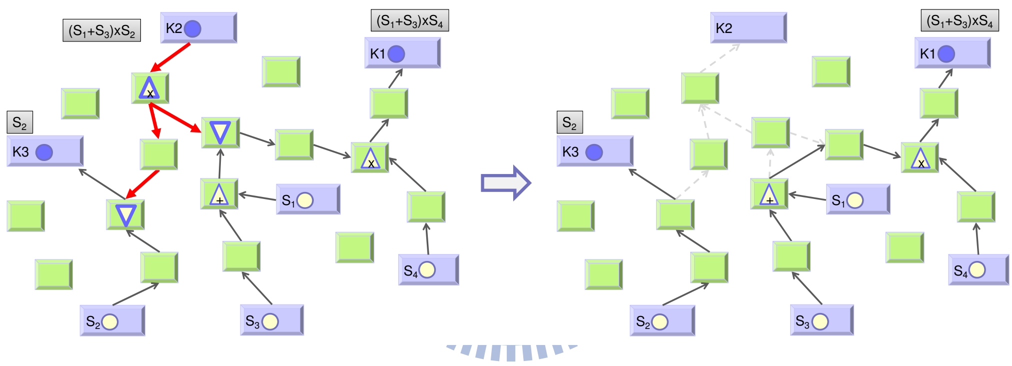


Figure 3.9: An example of deleting an objective function.

Chapter 4

Simulation Results

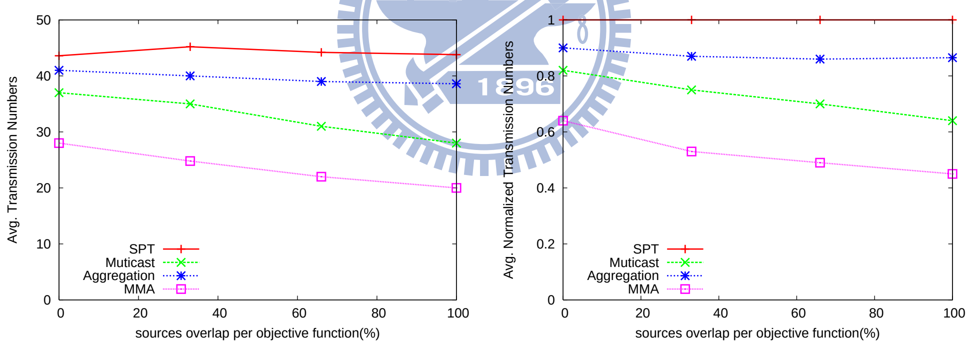
In this chapter, we address the simulation results comparing the performance of our solution against a shortest path mechanism, a simple multicast mechanism and a simple aggregation mechanism. We evaluate the relation between performance and the overlap of objective functions. We also increase the objective function in the size and numbers.

4.1 simulation environment

We setup 60 nodes randomly deployed in 500×500 $unit^2$ in the simulations. The transmission radius of each node is 50 units. In shortest path mechanism, the transmission number for a pair of a destination and a source is the least hop-count between the destination and the source. We sum up all the cost of the pairs in the network as total transmission numbers. In multicast mechanism, for each requested source and its subscriber, we construct a multicast tree based on a minimum spanning tree. The total transmission numbers is the total cost of the multicast trees. In aggregation mechanism, for each destination and its requiring source, we construct an aggregation tree based on a minimum spanning tree. As well as multicast mechanism, the total transmission numbers is the total cost of the aggregation trees in the network.

4.2 Overlap of Objective Function

First, we evaluate the overlap of objective functions. Fig. 4.1 shows the simulation results. In the figure, there are 3 objective functions in the network. Each of them requiring 6 source for their inputs. Since the more overlap of the objective functions have the more opportunities to share the sensing data, the transmission number decreases while we increase the overlap ratio. The SPT mechanism is case that transmits data packets individually without any aggregation data or multicast relay techniques. Therefore, we normalize the transmission numbers to the value of SPT. The aggregation mechanism brings just a little outperformance since there is precedence of operation in the objective functions. That means, it can not aggregation data at some aggregation node in its aggregation tree so the mechanism just reduce the transmission numbers slightly.

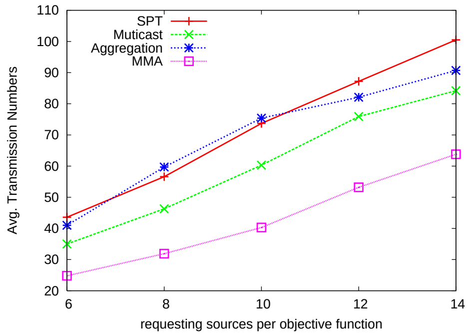


(a) Avg. Transmission Number (b) Normalized to SPT

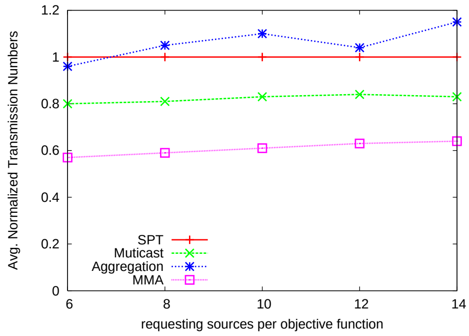
Figure 4.1: Overlap of Objective Functions

4.3 Increase Objective Function

In the section, we increase the objective function in numbers and size of inputs. Fig. 4.2 and Fig. 4.3 show the simulation results. The results are similar in both figures. Our solution takes the more efficient than multicast mechanism. However, the simple aggregation mechanism suffers from the many-to-many communication with objective functions. In some cases, SPT even surpasses the aggregation mechanism.

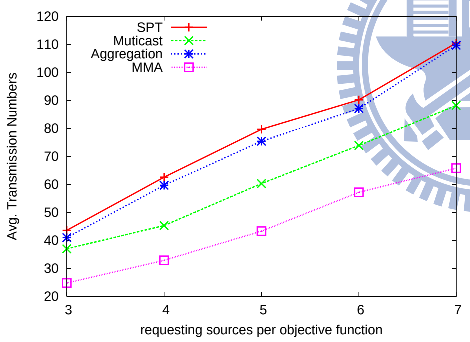


(a) Avg. Transmission Number

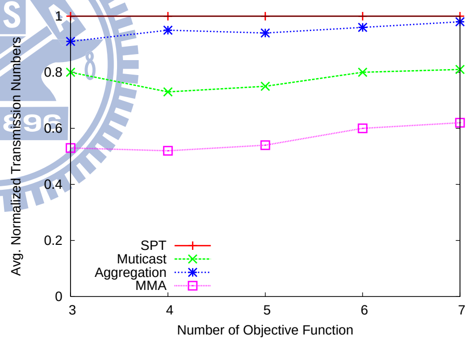


(b) Normalized to SPT

Figure 4.2: Increase the size of each objective function



(a) Avg. Transmission Number



(b) Normalized to SPT

Figure 4.3: Increase the number of objective functions

Chapter 5

Conclusions

This paper addressed the problem of many-to-many communications in a wireless sensor network. We propose a mobile agent model for distributed multi-source and multi-destination data transmission which takes a logical view to provide a routing schema to satisfy the objective requirements in the many-to-many communication. We decompose the objective requirements into several single aggregation operations in the network. Thus, it is easy to share the joint aggregation requirements. The simulation results reveal the outperformance of the model in the many-to-many communication. Moreover, Mobile agent model can be easily adapted to many-to-one or one-to-many communication schema since it concerns about not only the aggregation but also the multicast techniques.

Bibliography

[1] M. A. Batalin, G. S. Sukhatme, and M. Hattig. Mobile robot navigation using a sensor network. In *Proc. of IEEE Int'l Conference on Robotics and Automation*, 2004.

[2] L. L. Fernandes and A. L. Murphy. Mvsink: Incrementally building in-network aggregation trees. In *Proc. of European Conference on Wireless Sensor Networks*, 2009.

[3] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng. Efficient in-network moving object tracking in wireless sensor networks. *IEEE Trans. on Mobile Computing*, 5(8):1044–56, 2006.

[4] M.-S. Pan, C.-H. Tsai, and Y.-C. Tseng. Emergency guiding and monitoring applications in indoor 3d environments by wireless sensor networks. *International Journal of Sensor Networks (IJSNet)*, 1(1/2):2–10, 2006.

[5] M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin, and Y.-C. Tseng. A WSN-based intelligent light control system considering user activities and profiles. *IEEE Sensors*, 8(10):1710–1721, 2008.

[6] S. Park, E. Lee, Y. Choi, F. Yu, and S.-H. Kim. Communication architecture to support multiple mobile users in wireless sensor networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2008.

[7] A. Silberstein and J. Yang. Many-to-many aggregation for sensor networks. In *Proc. of IEEE Int'l Conference on Data Engineering (ICDE)*, 2007.

[8] X. Wang, J. S. Dong, C. Chin, S. Hettiarachchi, and D. Zhang. Semantic space: An infrastructure for smart spaces. *IEEE Pervasive Computing*, 3(3):32–39, 2004.

[9] T. Wolf and S. Y. Choi. Aggregated hierarchical multicast – a many-to-many communication paradigm using programmable networks. *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 33(3):358–369, 2003.

[10] Y. Xu, J. Winter, and W.-C. Lee. Prediction-based strategies for energy saving in object tracking sensor networks. In *Proc. of IEEE Int'l Conference on Mobile Data Management*, 2004.