VIDEO SURVEILLANCE OVER WIRELESS NETWORKS

YANG ZOU

M.Phil

The Hong Kong Polytechnic University

2015

The Hong Kong Polytechnic University

Department of Computing

Video Surveillance over Wireless Networks

Yang Zou

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Philosophy

July 2013

CERTIFICATE of ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________________(Signature)

_________Yang Zou_______(Student Name)

ABSTRACT

Recent advances in wireless communication and portable devices have led to rapid development of wireless applications. Wireless video surveillance is one of the important topics. Wireless video surveillance can be deployed flexibly to cover different surveillance areas. However, it is still challenging to construct a scalable wireless video surveillance system. First of all, wireless networks are often associated with the scarcity of communication bandwidth. Thus, video transmission in wireless environment must be very efficient in order to avoid network congestion. Secondly, the asynchronous communication model and dynamic topologies exacerbate the unreliable wireless connections. As a result, it is difficult to adopt traditional server-client streaming services in wireless video surveillance systems. Thirdly, user mobility issues bring additional challenges to the video transmission process.

There are two key problems in construction of a scalable video surveillance system on top of wireless networks: in-network processing in the asynchronous wireless system and wireless video transmission. In this thesis, we investigate the challenging issues in developing video surveillance systems over wireless networks and propose a solution to address the issues. Our research will be divided into three parts as following.

In the first part, we study the Publish/Subscribe (Pub/Sub) paradigm to review its decoupling characteristics in time, space and synchronization. We utilize these characteristics and propose a design of Pub/Sub based video surveillance system on

top of interconnected wireless mesh networks (WMNs) and wireless sensor networks (WSNs). In the proposed framework, mobile users submit their subscriptions on particular events, i.e. traffic jam in Nathan Road. Subscriptions will be registered to the system and only related video content will be published to mobile users. This novel design provides users with a new approach to interact with wireless camera nodes. The proposed Pub/Sub middleware can help achieve both efficient use of wireless network resource and efficient detection of related events.

In the second part, we introduce a design of heterogeneous networks, which consists of Wireless Mesh Networks (WMNs) and Wireless Sensor Networks (WSNs), to support the Pub/Sub. In this part, we study the packets and information translation between WSNs and WMNs, and propose a specific strategy to interconnect these two types of networks. The event detection is realized by utilizing wireless sensors. Mesh nodes are in charge of subscription/event mapping and video content transmission. A prototype system is implemented with testing results to validate the effectiveness of our design.

In the third part, we focus on wireless mesh networks and study the problem of delay constrained Pub/Sub in order to optimize the in-network processing for real-time video subscription and publishing. We propose algorithms to customize the Pub/Sub middleware to support video surveillance for real-time events. Simulation and testing are conducted to evaluate the performance of our proposed solutions.

PUBLICATIONS

Yang Zou, Jiannong Cao, Hejun Wu. TraffiCast: Real-time Pub/Sub based video surveillance system over interconnected WMNs and WSNs, *Distributed Computing in Sensor Systems Workshops (DCOSSW 2010)*, June 2010, Santa Barbara, California, USA.

Yang Zou, Jiannong Cao, Jun Zhang. Pub-Eye: The delay constrained Pub/Sub for large scale wireless video surveillance, *Wireless Communications and Networking Conference (WCNC 2011)*, March 2011, Cancun, Quintana Roo, Mexico.

Jiannong Cao, Chisheng Zhang, Jun Zhang, Yueming Deng, Xin Xiao, **Yang Zou**, Miao Xiong, Jie Zhou, Gang Yao, Wei Feng, Liang Yang and Yao Yu ,SHAWK: Platform for Secure Integration of Heterogeneous Advanced Wireless Networks, *(AINA-2012)*, March 2012, Fukuoka, Japan .

Yang Zou, Jiannong Cao, et al., “Efficient Subscription Maintenance for Mobile Subscribers in Large Scale Wireless Video Surveillance”, Submitted for publication.

ACKNOWLEDGEMENTS

Firstly, I would like to express my deep gratefulness to my supervisor, Prof. Jiannong Cao, for his patient and systematic guidance, illuminating advices and encouragement in my research work and the writing up of this confirmation thesis.

Then, I would give my gratitude to Dr. Weigang Wu, Dr. Weiping Zhu, Dr. Jun Zhang and Dr. Yuan Zheng for their kind help in my research work and project work.

I also want to thank my fellow mates Mr. Wei Feng, Mr. Chisheng Zhang, Mr. Xin Xiao, Mr. Gang Yao, Mr. Yueming Deng, Mr. Haitao Chen, Miss. Miao Xiong, and Miss. Jie Zhou for their kind helps to my research and project works in one way or another.

Most Importantly, I would like to thank my wife Yan Cai. Her constant support and patience help me to enjoy my MPhil study. Finally, I would like to express my heart-felt gratitude to my parents, whose confidence in me is what drives me forward to continue on the road ahead.

TABLE OF CONTENTS

Abstract i

Publications iii

Acknowledgements iv

Table of Contents v

List of Figures viii

Chapter 1 Introduction 1

1.1 Motivations and Challenging Issues of Wireless Video Surveillance 1

1.2 Our Contributions 6

1.2.1 Pub/Sub Based Framework for Wireless Video Surveillance 8

1.2.2 Interconnected WSNs and WMNs to Support Wireless Video Surveillance........ 9

1.2.3 Delay Constrained Pub/Sub for Large Scale Real-time Video Subscription 10

1.3 Outline of the Thesis 10

Chapter 2 Literature Review 13

2.1 Wireless Video Surveillance Systems 13

2.1.1 Single-tier Architecture Based Systems 14

2.1.2 Multi-tier Architecture Based Systems 17

2.2 Work Distribution Mode in Pub/Sub Systems 20

2.2.1 Distributed Broadcast Based 21

2.2.2 Distributed Multicast Based 21

2.2.3 Improved Multicast Based 22

Chapter 3 Pub/Sub Framework Design for Wireless Video Surveillance 23

3.1 The Pub/Sub Paradigm 23
3.2 System Architecture of Pub/Sub Based Wireless Video Surveillance 26
3.3 System Components.. 28
3.4 Main Functions 29
3.4.1 Subscription 29
3.4.2 Notification 31
3.4.3 Unsubscribing 33
3.4.4 Mobile Subscriber Handoff.. 33
3.5 Summary 34

Chapter 4 Interconnected WSNs and WMSs for Pub/Sub Video Surveillance...35

4.1 System Design 35
4.1.1 Challenging Issues 35
4.1.2 Strategy of WSN-WMN Interconnection 38
4.1.3 Event Subscription and Detection.. 40
4.2 System Implementation 41
4.2.1 WSN Setup and the Interconnection Gateway.................................... 41
4.2.2 WMN Construction for Video Transmission....................................... 43
4.2.3 Pub/Sub on Top of the Heterogeneous Networks 46
4.3 Deployment and Field Testing.. 48
4.4 Security Extension 52
4.4.1 TLS Based Secure Video Transmission.. 52
4.4.2 Deployment.. 53
4.5 Summary 54

Chapter 5 Delay Constrained Pub/Sub for Video Surveillance........57

5.1 The Broker Selection Problem.. 58

5.2 Design of Heuristic Algorithms ... 62
5.3 Performance Evaluation ... 65
5.3.1 Fix K Algorithm ... 66
5.3.2 Dynamic K Algorithm ... 71
5.4 Testing ... 76
5.4.1 Multi-hop Delay Estimation ... 76
5.4.2 Field Testing Setup and Configuration ... 77
5.4.3 Field Testing Results ... 79
5.5 Conclusions ... 80
Chapter 6 Conclusions and Future Works ... 83
6.1 Conclusions ... 83
6.2 Future Works ... 84
References ... 87

LIST OF FIGURES

Figure 1 Evolution of video surveillance systems 2

Figure 2 Future wireless automated video surveillance systems 3

Figure 3 Challenging issues in wireless video surveillance 4

Figure 4 An outline of the contributions in this thesis 7

Figure 5 The Pub/Sub basic structure 8

Figure 6 The video surveillance system on top of WMNs and WSNs 9

Figure 7 Fire-tide hotpot for video surveillance 16

Figure 8 Example of multi-tier structure 17

Figure 9 SensEye system architecture 18

Figure 10 PTZ camera for multi-scale surveillance 19

Figure 11 Space decoupling 24

Figure 12 Time decoupling 24

Figure 13 Synchronization decoupling 25

Figure 14 Architecture of the Pub/Sub based wireless video surveillance 27

Figure 15 Algorithm for Subscribe 30

Figure 16 Algorithm for Notify 32

Figure 17 Unsubscribe Algorithm 33

Figure 18 Handling mobile subscribers 34

Figure 19 Single-gateway solution between WSNs and WMNs 38

Figure 20 Communications between WSNs and WMNs 39

Figure 21 Street model of our ITS platform 42

Figure 22 Adapted model car with MicaZ mote. 42

Figure 23 T902 mesh node and video capturing hardware 43

Figure 24 The messaging stucture of the client console 44

Figure 25 Communication process for the video client console 44

Figure 26 Client GUI that enables users to subscribe events 45

Figure 27 Broker selection procedure 45

Figure 28 Start video streaming 46

Figure 29 Camera control 46

Figure 30 Modified HNA message 47

Figure 31 Subscription registration and dissemination 47

Figure 32 Event notification and video publishing 48

Figure 33 Client data rate while receiving video stream from surveillance system ..51

Figure 34 The deployment of surveillance system for traffic monitoring 51

Figure 35 TLS session establishment for video transmission 54

Figure 36 The broker selection algorithm. 63

Figure 37 Density of the forwarded filter changes according to the location 64

Figure 38 Broker selection among the whole broker set 68

Figure 39 Comparison with self-selection algorithm 70

Figure 40 Comparison with d-SPR 71

Figure 41 Comparison between *Fix-K algorithm* and *Dynamic-K* 75

Figure 42 Field testing deployment 78

Figure 43 Field testing results 80

CHAPTER 1 INTRODUCTION

Recent advances in wireless communication and portable devices have led to rapid development of wireless applications. Wireless video surveillance is among the thriving examples. However, the underlying wireless environment raises new challenging issues on the development of video surveillance systems. We address the new challenges and propose a design of Publish/Subscribe (Pub/Sub) based video surveillance system on top of wireless heterogeneous networks. The proposed system supports automatic event detection and event-driven video content transmission with the optimization of the wireless network resources.

In this chapter, we first give the motivations and discuss challenging issues of wireless video surveillance. Then, we summarize our contributions and outline the contents and structures of the thesis.

1.1 Motivations and Challenging Issues of Wireless Video Surveillance

Wireless video surveillance is considered as one of the killer applications of wireless networks. However, to design and deploy a large scale video surveillance system in a wireless network environment is an open problem due to the shared wireless medium and asynchronous communication model. To start with, we look into the history of video surveillance system development so we can get a clear vision about the features required in future wireless video surveillance systems.

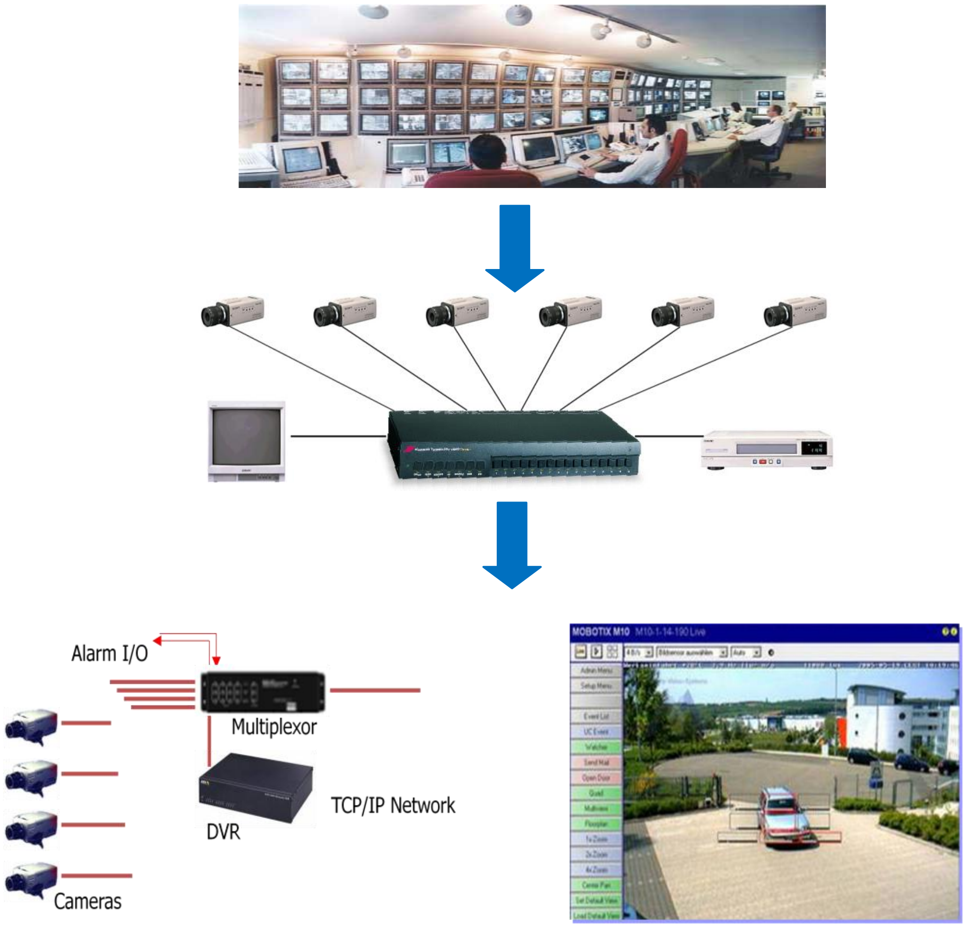


Figure 1 Evolution of video surveillance systems

The evolution of video surveillance can be categorized into three generations. The first generation of video surveillance systems are known as CCTV based systems. The CCTV video surveillance systems use analog video cameras to capture video stream and use analog cables to transmit captured video stream to the central server. Analog tapes are used for storing the video data. Human operators are required to detect events and control the cameras. The second generation of video surveillance systems also use analog video cameras to capture video stream. But for stream transmission, digital cables are used to replace analog cables. Human operators are no longer requisite for the surveillance systems, as motion detection techniques are applied to the second generation video surveillance systems. The current third generation of video surveillance systems utilize complete digital solution on video capturing and transmission. Compared to previous generations, the third generation of video surveillance systems not only support automatic event detection but also enable camera nodes to cooperate with each other.

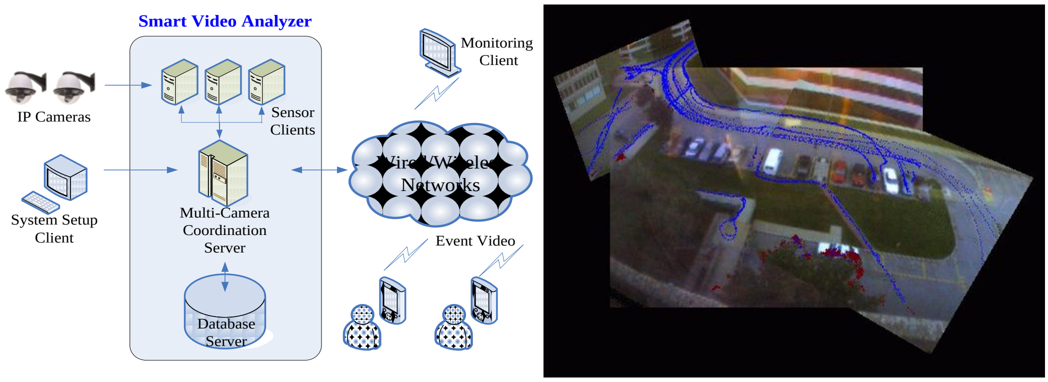


Figure 2 Future wireless automated video surveillance systems

Wireless video surveillance systems can provide a dynamic and scalable solution for camera network deployment [55]. Compared to the traditional surveillance systems, the surveillance system with the wireless backbone can be more scalable in size and function. Therefore, the emerging new wireless surveillance systems is not only expected to bring the advantage in flexibility and scalability but also required to inherit the existing features. However, the existing framework of surveillance systems, especially the networking part, is incompatible to the wireless video surveillance and cannot be adopted directly. For example, if camera nodes transmit video content constantly without any filtering, not only the surveillance is inefficient but also the network bandwidth is wasted for transmitting non-related video streams.

To design the wireless video surveillance systems, we summarize the following areas as the key challenging issues (shown in Figure 3).

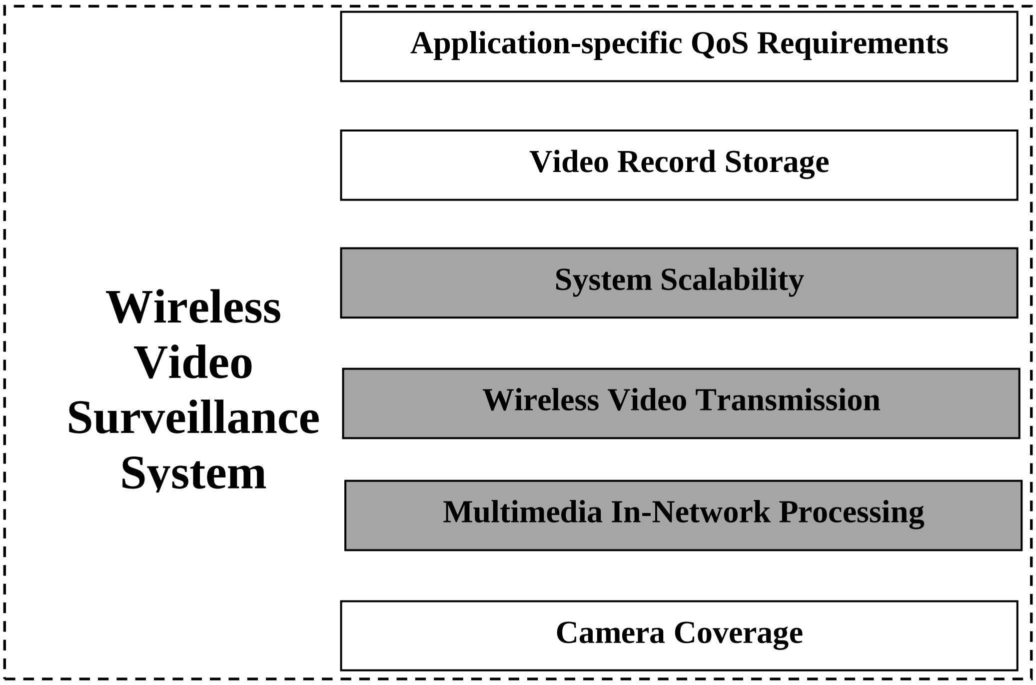


Figure 3 Challenging issues in wireless video surveillance

Camera Coverage*:* Video cameras have larger sensing range and are sensitive to direction of acquisition (directional). Video sensors can capture images only when there is unobstructed line of sight between the event and the sensor. Hence, coverage models developed for traditional wireless sensor networks are not sufficient for pre-deployment planning of a video surveillance system in the wireless mesh networks. In [39][56], a preliminary investigation of the coverage problem for video sensor networks is conducted. The concept of sensing range is replaced with the camera's field of view, i.e., the maximum volume visible from the camera. Detection algorithms designed for traditional sensor networks do not perform well with video sensors in terms of coverage preservation of the monitored area.

In-network processing: Video surveillance needs to apply multimedia in-network processing algorithms on the raw data extracted from the environment. This requires architectures for collaborative, distributed, and resource-constrained processing that allow for filtering and extraction of semantically relevant information at the edge of the sensor network. This may increase the system scalability by reducing the transmission of redundant information, merging data originated from multiple views, on different media, and with multiple resolutions. For example, in security applications, information from unimportant scenes can be compressed to a simple scalar value or even not be transmitted; while in environmental applications, distributed filtering techniques can create a time-elapsed image. Hence, it is necessary to develop application-independent architectures to flexibly perform in-network processing of the multimedia content gathered from the environment.

Wireless Video Transmission*:* Data transmission is always a key challenge in the real-time streaming system. In the new generation of video surveillance, wireless backbone is used to replace the cables. However, it also brings following difficulties [11][22]:

1. Transmission bandwidth and transmission power are scarce resources in wireless environment;
2. The high loss percentage of wireless links requires sophisticated techniques for channel encoding that often increase transmission delays.

System Scalability*:* The system scalability can be defined as both size and functionality scale of the system. A video surveillance system is also required to support cooperative detection with large number of camera nodes. Besides, unlike traditional wireless multimedia sensor applications, video surveillance system needs to support several heterogeneous and independent applications with different requirements. Thus, it is necessary to develop flexible, hierarchical architectures that can accommodate the requirements of all these applications in the same infrastructure.

Application-specific QoS Requirements*:* The wide variety of applications envisaged on surveillance system will have different requirements. Multimedia data includes snapshot and streaming multimedia content. Snapshot-type multimedia data contains event triggered observations obtained in a short time period. Streaming multimedia content is generated over longer time periods and requires sustained information delivery. Hence, a complete framework is required to deliver QoS and consider application-specific requirements.

Multimedia Data Storage*:* The video surveillance records are frequently used to provide forensic records, therefore all the captured video content should be stored to the database for future use. The traditional way of storing the video data is to transmit all the video record to a central database. However, in WMNs the transmission of all the video record to one centralized DB server is a loaded burden for the whole network.

1.2 **Our Contributions**

In our research, we focus on the in-network processing and the video transmission in surveillance systems. Figure 4 outlines our contributions. We propose a

Publish/Subscribe (Pub/Sub) based video surveillance system over the interconnected wireless mesh networks (WMNs) and wireless sensor networks (WSNs). A customized Pub/Sub paradigm is applied to provide scalable in-network processing for real-time event detection and notify users with the captured video stream. The underlying interconnected WMN and WSN heterogeneous architecture is proposed to better support event detection and video transmission.

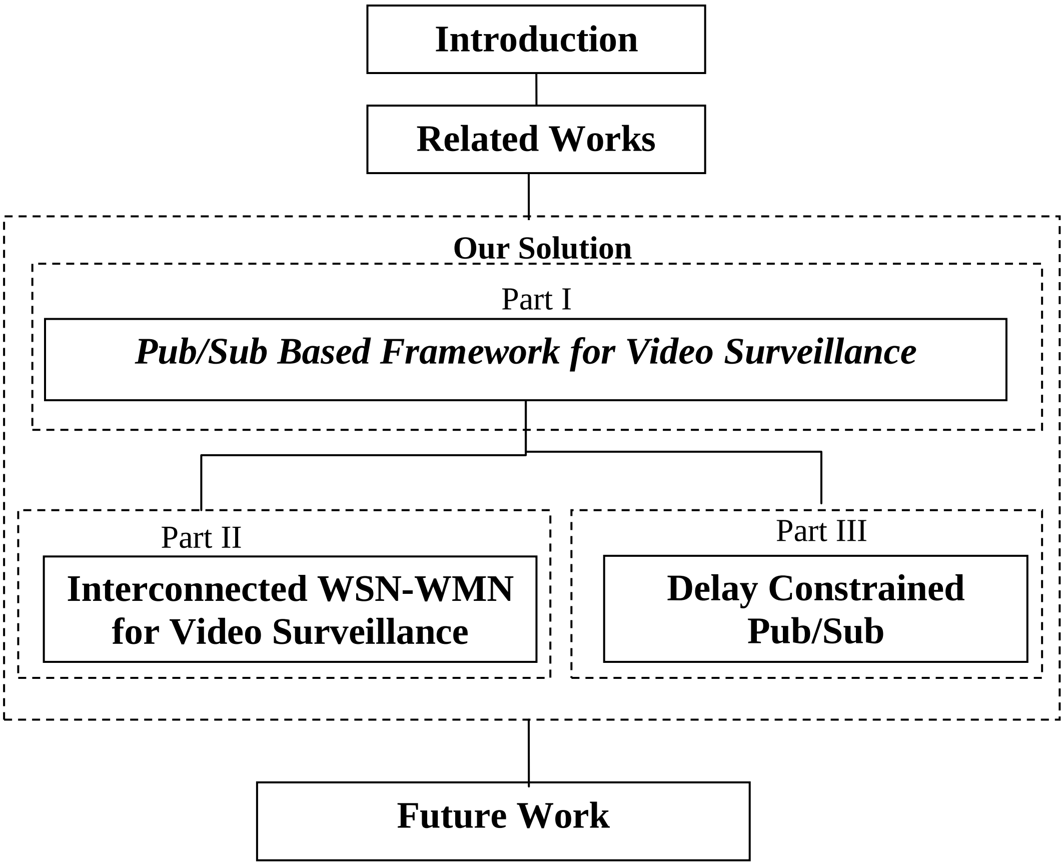


Figure 4 An outline of the contributions in this thesis

1.2.1 Pub/Sub Based Framework for Wireless Video Surveillance

In this part of our research, we propose a complete Pub/Sub based framework for video surveillance application on top of wireless networks. The publish/subscribe paradigm (Figure 5) is well received as the tool for building loosely coupled information distribution systems. The paradigm is popular for its decoupling characteristics in time, space and synchronization. There are three variants of Pub/Sub system in the existing works, namely topic based, content based and type based. The content based Pub/Sub is widely used for its fine granularity of event description, which can help achieve good expressiveness of user's interests.

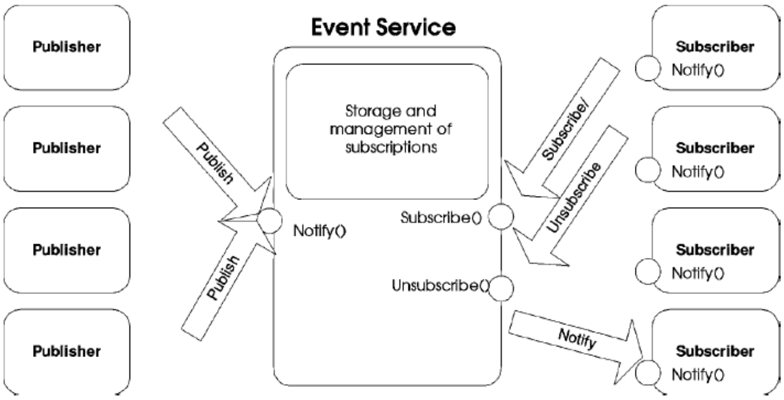


Figure 5 The Pub/Sub basic structure

In our research, we propose a content based Pub/Sub framework to improve the efficiency of video retrieval. Our solution can achieve both efficient use of network resources and efficient detection of events of interests. In the content based Pub/Sub, contents are structured in pairs of <attribute, value>. Subscribers can specify event of interests over content attributes as subscriptions, which will be registered as *filters* at the brokers. In the video surveillance application scenario, event notification no longer consists of text notification only. It is presented in a rich content form including video stream as well. The captured video stream will be associated with subscribed events and be forwarded to mobile subscribers.

1.2.2 Interconnected WSNs and WMNs to Support Wireless Video Surveillance

When designing our wireless video surveillance system, we not only require high performance network connections for real-time video stream transmission, but also require automatic detection of subscribed events. We propose a framework with support of interactions between wireless sensor networks for event detection and wireless mesh networks for video transmission. The interconnection of these two types of networks provides new features in wireless video surveillance systems and enables new services, but it also brings new challenges to system development.

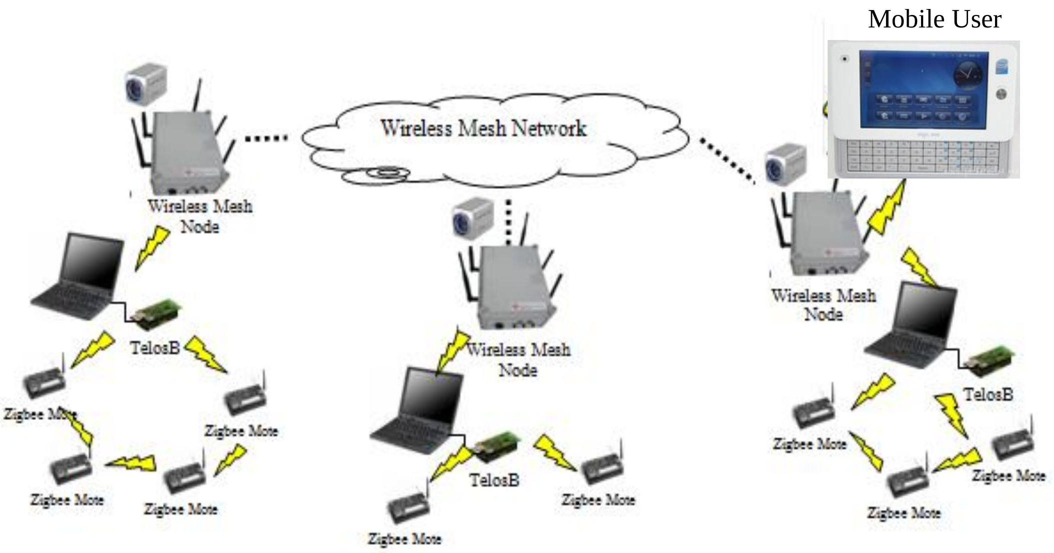


Figure 6 The video surveillance system on top of WMNs and WSNs

1.2.3 Delay Constrained Pub/Sub for Large Scale Real-time Video Subscription

The constraints on events publishing for surveillance purpose, including video stream, are surely much stricter than those in traditional event notification systems. The most important constraint is "real-time" requirement. As a consequence, time latency becomes a critical constraint in our design of the Pub/Sub middleware for real-time video surveillance. Here, time latency is defined as the time from the event occurs to the time that subscriber receives the video of subscription. In the rest part of this thesis, this time latency will be stated as delay for convenience. We propose two heuristic algorithms for managing all the subscriptions to form an efficient filter mapping structure, which can disseminate event notification to interested subscribers with the delay constraint. Simulation is conducted to evaluate the performance of the proposed solution. We also share our experience of real-world implementation of the Pub/Sub based video surveillance system.

1.3 Outline of the Thesis

With the considerations all above, we target at designing a scalable content based Pub/Sub system to support real-time event driven video surveillance over the interconnected WMN and WSN networks. As far as we know, we are the first to use Pub/Sub paradigm to support real-time video streaming service, and we are the first to jointly take delay constraint and scalability issue into consideration for designing Pub/Sub system.

The remainder of this thesis is organized as follows. In Chapter 2, we review the related work about existing video surveillance systems and Pub/Sub works. In Chapter 3, a Pub/Sub based system framework is proposed. In Chapter 4, we focus on the construction of underlying WSNs-WMNs heterogeneous networks to realize the Pub/Sub based wireless video surveillance system. We also look into the security issues and propose a preliminary solution to ensure the confidentiality of published video content. In Chapter 5, we study the problem of delay constrained Pub/Sub. Then, simulations and field testing are included to show the effectiveness of our proposed subscription dissemination algorithms.

CHAPTER 2 LITERATURE REVIEW

In this chapter we review the related works. We start with existing wireless surveillance system works in the first subsection. As introduced previously, the Pub/Sub paradigm will be applied to provide event based surveillance. We will study related works about Pub/Sub in the second subsection.

2.1 Wireless Video Surveillance Systems

In the early days, video surveillance systems basically rely on security guards to monitor the screens. IBM Research's "People Vision" [44] project was the first to introduce the concept of Smart Surveillance, and the application of automated analysis of surveillance video to reduce the tedious, time-consuming task of viewing video feeds from a large number of security cameras. The video surveillance has various sub-applications like "Access control in special areas", "Person-specific identification in certain scenes", "Crowd flux statistics and congestion analysis", "Anomaly detection and alarming" and so on. Each type of the surveillance applications actually has different requirement on both the camera hardware and the network data rate.

Multi-tier architecture for video surveillance is scalable, but it is too complex and is not necessary for wireless mesh. In the proposed system, we will introduce an alternative way to extend the system scalability in terms of functionality without increase the system complexity. With the homogenous architecture, the video surveillance technologies are evolving in isolation; for example, face recognition technology constrains the subject to be in front of the camera while motion detection on video streams does not require identity tracking.

Comparing single-tier and multi-tier architecture, there are disadvantages on both of them. For the single-tier model, functionality is not scalable. On the other side, for the multi-tier model, interactions between different tiers are complicated and will increase message overhead. We will classify the existing video surveillance systems into two categories, single-tier and multi-tier, in the following subsections.

2.1.1 Single-tier Architecture Based Systems

Most existing wireless video surveillance system are based on a flat, homogenous architecture in which every camera node has the same physical capabilities and can only interact with neighboring cameras.

W4 [19] is a real time visual surveillance system for detecting and tracking multiple people, and monitoring their activities in an outdoor environment. It operates on monocular gray-scale video imagery, or on video imagery from an infrared camera. W4 employs a combination of shape analysis and tracking to locate people and their parts (head, hands, feet, and torso) and to create models of people's appearance so that they can be tracked through interactions such as occlusions. It can determine whether a foreground region contains multiple people and can segment the region into its constituent people and track them. W4 can also determine whether people are carrying objects, and can segment objects from their silhouettes, and construct appearance models for them so they can be identified in subsequent frames. W4 can recognize events between people and objects, such as depositing an object, exchanging bags, or removing an object. It runs at 25 Hz for 320×240 resolution images on a 400 MHz dual-Pentium II PC.

CMU's DARPA VSAM [24] project, is to develop an end-to-end, multi-camera surveillance system that allows a single human operator to monitor activities in a cluttered environment using a distributed network of active video wireless sensors. In realistic surveillance scenarios, it is impossible for a single sensor to see all areas at once, or to visually track a moving object for a long period of time. A promising solution to this problem is to use a network of video sensors to cooperatively monitor all objects within an extended area and seamlessly track individual objects.

It proposes system-level algorithms that fuse sensor data, task sensors to perform autonomous cooperative behaviors, and display results to the operator in a comprehensible form. It also has built a test-bed which consists of multiple cameras distributed over an area of roughly 0.4 km on the campus of CMU.

Deep Vision Network In [36], the author proposes "Deep Vision network", which performs operations including object detection or classification, image segmentation, and motion analysis through a network of low-end MICA motes equipped with low resolution cameras. Information such as the presence of an intruder, the number of visitors in a scene or the probability of presence of a human in the monitored area is obtained by collecting the results of these operations. Deep Vision provides a querying interface to the user in the form of declarative queries. Each operation is represented as an attribute that can be executed through an appropriate query. In this way, low-level operations and processing are encapsulated in a high-level querying interface that enables simple interaction with the video network.

Fire-tide Hotpot is an example of existing commercial WMNs based video surveillance systems. It uses homogenous architecture, in which every camera node has the same physical capabilities. Figure 7 shows the sample deployment of the video surveillance system from Fire-tide. Fire-tide is a developer of wireless multi-service mesh technology, and Axis Communications, a company working on network video solutions, have announced a strategic partnership to deliver high-quality video over wireless mesh networks, which are being used by a number of cities to conduct wireless video surveillance.

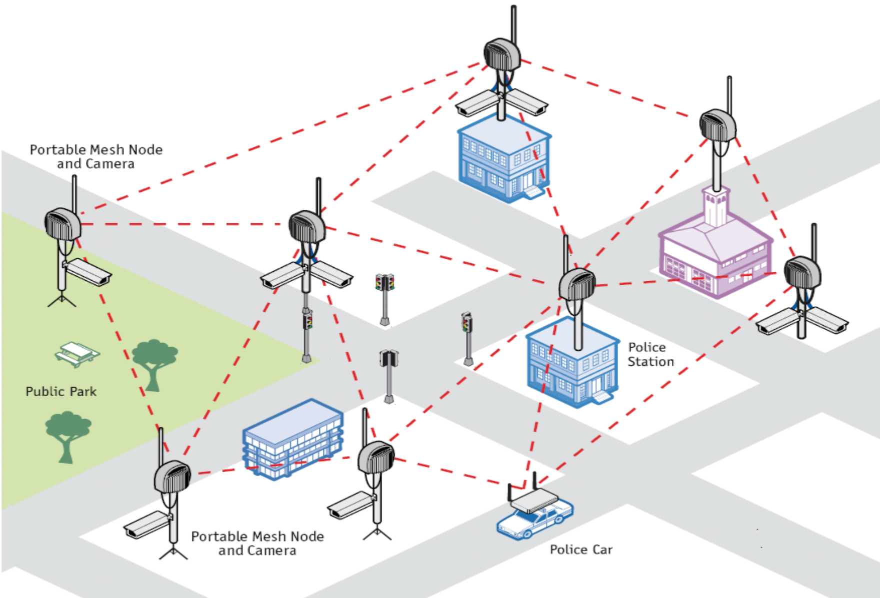


Figure 7 Fire-tide hotpot for video surveillance

2.1.2 Multi-tier Architecture Based Systems

In [25], a multi-tier architecture is advocated for video sensor networks for surveillance applications, with the lower tier constituted of low-resolution imaging sensors, and the higher tier composed of high-end pan-tilt-zoom cameras. It is argued, and shown by means of experiments, that such architecture offers considerable advantages with respect to the single-tier architecture.

The multi-tier architecture also results in the multi-scale property of the surveillance system. The goal of a multi-scale surveillance system is to acquire information about objects in the monitored space at several scales in a unified framework.

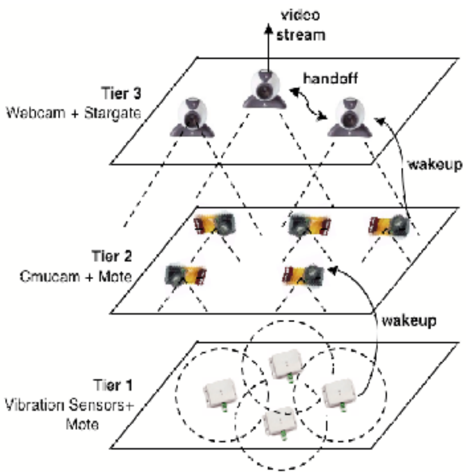


Figure 8 Example of multi-tier structure

SensEye [25] focuses on multi-tier networks where detection, recognition and tracking be performed on different nodes and cameras to achieve object detection, object recognition and object tracking.

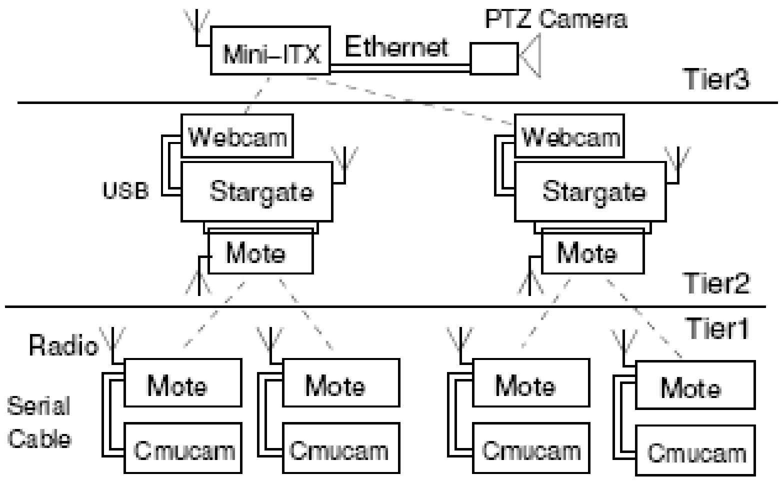


Figure 9 SensEye system architecture

SensEye assumes three-tier architecture. The lowest tier in SensEye comprises of Mote nodes equipped with 900MHz radios and low-fidelity Cyclops or CMUcam camera sensors. The second SensEye tier comprises of Stargate nodes, which equipped with web-cams. Each Stargate is equipped with an embedded 400MHz XScale processor that runs Linux and a web-cam that can capture higher fidelity images than Tier 1 cameras. Each Tier 2 node also consists of two radios—a 802.11 radio that is used by Stargate nodes to communicate with each other, and a 900MHz radio that is used to communicate with Motes in Tier 1. The third tier of SensEye contains a sparse deployment of high-resolution pan-tilt-zoom cameras connected to embedded PCs. The camera sensors at this tier are flexible and can be utilized to fill small gaps in coverage provided by Tier 2 and to provide additional redundancy for tasks such as localization. Nodes in each tier and across tiers are assumed to communicate using their wireless radios in ad-hoc mode; no base-stations are assumed in this environment. The radio interface at each tier is assumed to be individually duty-cycled to meet application requirements of latency and lifetime constraint at each node. Consequently, the application tasks need to be designed carefully since radios on the nodes (and nodes themselves) are not “always-on”.

IBM’s “Multi-scale Tracking for Smart Surveillance” [41] basically uses two groups of cameras.

1. The static cameras cover the complete scene of interest and provide a global view;
2. The pan tilt zoom (PTZ) cameras are meant to obtain detailed or fine-scale information about objects of interest in the scene.

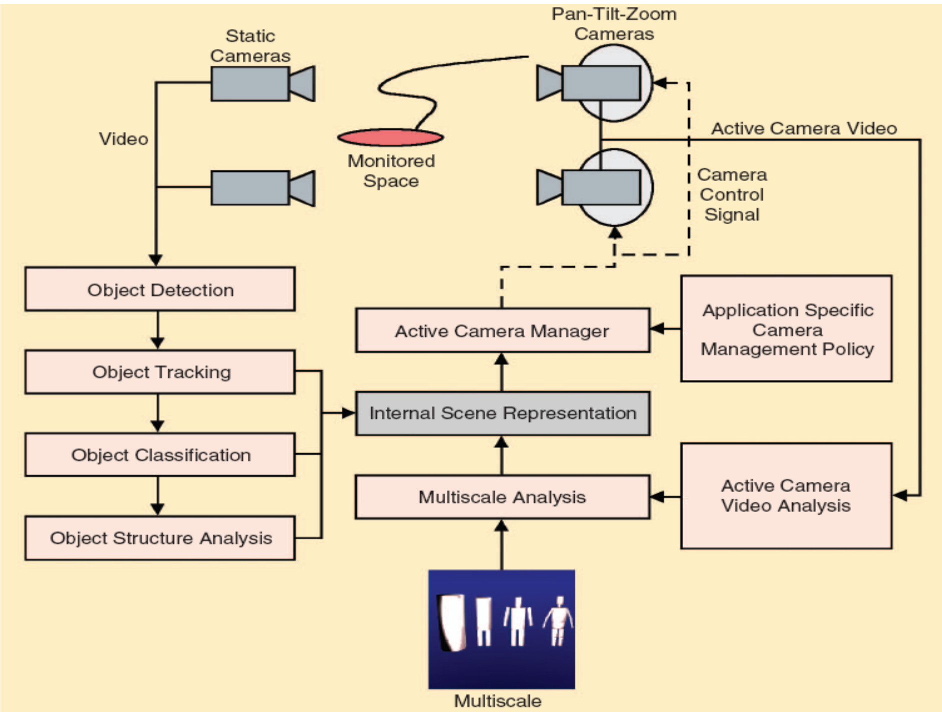


Figure 10 PTZ camera for multi-scale surveillance

Fixed camera images can be used to extract additional information about the objects at a coarse level, like object class (person, car, truck) or object attributes (position of a person’s head, velocity of the car, etc.). The coarse-scale information is used as a basis to “focus the attention of the PTZ cameras”.

Information from the PTZ cameras is then used to perform fine-scale analysis. For example, if the PTZ camera is directed towards a person, the fine-scale analysis could include face detection. The information from the coarse-scale and fine-scale analyses is combined in the internal scene representation.

2.2 Work Distribution Mode in Pub/Sub Systems

Publish/Subscribe paradigm is widely received as the tools for building loosely coupled information distribution systems. The paradigm is popular for its decoupling characteristics in time, space and synchronization. Three typical variants of Pub/Sub system can be found in the big family, namely topic based, content based and type based. The content based Pub/Sub is widely used for its fine granularity of event description, which can help achieve good expressiveness of the user’s interests.

In previous content based Pub/Sub systems, the major focuses are often the tradeoff between system scalability and the expressiveness of the event. As the centralized control must be discarded to achieve scalability, the centralized solutions will not be discussed.

Among the previous distributed content based Pub/Sub systems, there are three basic strategies for work distribution [18], namely “distributed broadcast” “distributed multicast” and “advertise and subscribe”. We further analyze the scalability of these three ways of work distribution in this subsection.

2.2.1 Distributed Broadcast Based

In "distributed broadcast", the subscription is only collected by the local event broker. When an event notification is received from the event publisher, the EB first looks up its local filters and then broadcast the event to all the other EBs to see whether there are any other interested subscribers. The cost of relaying the subscription in "distributed multicast" can be very small but the cost for broadcasting the event is very expensive and delay of delivering the event to the subscriber is very long. One example of broadcast based system is SIFT Grid [45].

2.2.2 Distributed Multicast Based

The "multicast based" works in an opposite way. In distributed multicast Pub/Sub, event is only forwarded to brokers that have subscripted for the very event. By using distributed multicast, the event notification relaying overhead can be reduced to a very low level. But, the system will pay extra price for the subscription relay and maintenance. In Siena [7], a new subscription is only forwarded to other brokers if the value range in this subscription covered existing filters registered at the brokers (E.g. "airport = any" covers the subscriptions "airport = Hong Kong"). In such way, the cost of subscription relay can be reduced and the propagation of subscription forms a scalable content filtering tree over the networks. However, construction of the content filtering tree is still too expensive to maintain.

2.2.3 Improved Multicast Based

The “advertise and subscribe” is the evolution of the “multicast based” alternative and is first introduced by the Siena [7]. In this approach, the event publisher uses advertisement to declare possible events that will be published and forwards the advertisement over the networks. The advertisement can then guide brokers to forward the subscription received. When a subscription is distributed from subscribers, it is only forwarded along the path of nodes that have the advertisement that can cover the subscription. In this way, the subscription forwarding cost is further reduced, compared to the previous two types of method. Moreover, computation of matching the subscription to the published event is also more effective. However, forwarding of the advertisement brings extra cost.

Although the advertisement can greatly improve the efficiency of setting up the content matching tree, it requires the knowledge of the publisher distribution, which is not always unavailable in practical applications. Thus, we are trying to design a new distributed algorithm to guide the brokers propagating the subscriptions.

Previous research works on Pub/Sub also figure out that the delay constraint is very important for some real time service. However, these works focus mostly on the QoS for notification publishing procedure. In our research, we consider not only the notification forwarding QoS procedure, but also the delay for mapping the interested group with the notification.

CHAPTER 3 PUB/SUB FRAMEWORK DESIGN FOR WIRELESS VIDEO SURVEILLANCE

In this chapter, we introduce our Pub/Sub based framework to support video surveillance in wireless networks. The proposed design enables users to subscribe events of interests and retrieve the real-time captured video content. We first study the Pub/Sub paradigm. Then we propose our Pub/Sub based system architecture in Section 3.2 and clarify the system components and main functions in Section 3.3 and Section 3.4.

3.1 The Pub/Sub Paradigm

In the wireless asynchronous systems, the interaction models applied have to remove the dependencies between information producers and receivers. The publish/subscribe [12] interaction scheme is able to provide a loosely coupled form of interactions required in large scale settings. Subscribers have the ability to express their interests in an event or events of certain patterns, and are subsequently notified of any events generated by a publisher, which matches their registered interests. An event is asynchronously propagated to all subscribers that registered interest in it. The decoupling that the event service provides between publishers and subscribers can be decomposed into the following three dimensions:

Space decoupling characteristics: The interacting parties do not need to communicate directly. Publishers publish events through an event service and subscribers get these events indirectly through the event service. Publishers do not hold the receiver list nor do they know how many of subscribers will be involved. On the subscriber side, subscribers do not have the list of publishers in order to get the events directly.

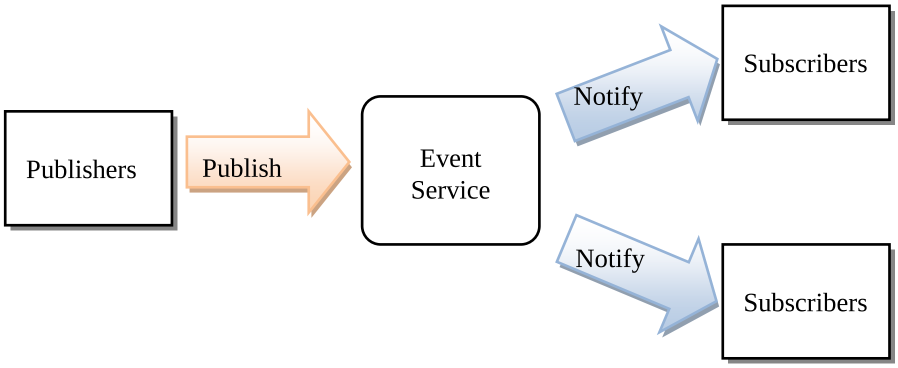


Figure 11 Space decoupling

Time decoupling: The interacting parties do not need to be actively participating in the interaction at the same time. In particular, publishers can publish events while subscribers are disconnected from the networks. On the other hand, subscribers can get notified about the occurrence of events while the original publishers of the events are disconnected.

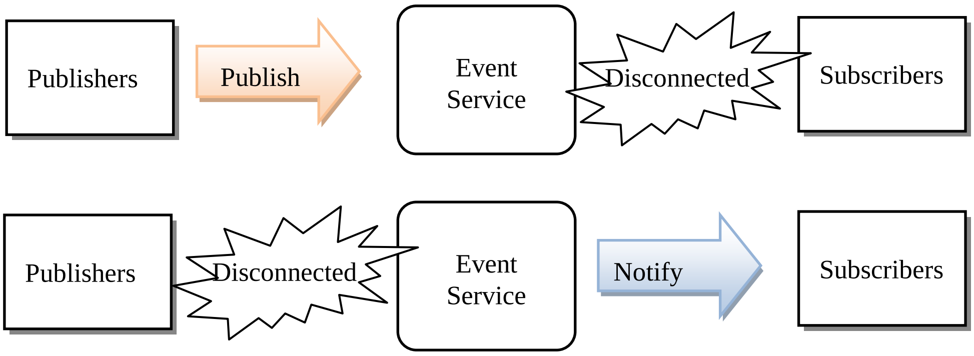


Figure 12 Time decoupling

Synchronization decoupling: Publishers are not blocked while producing events, and subscribers can be asynchronously notified (through a callback) of the occurrence of an event while performing concurrent activities. The production and consumption of events do not happen in the main flow of control of publishers or subscribers. Therefore, the event service does not happen in a synchronous manner.

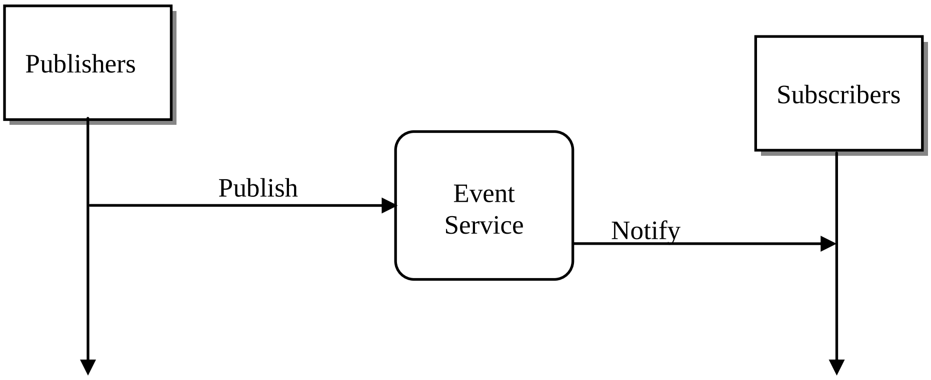


Figure 13 Synchronization decoupling

We take advantages of the Pub/Sub paradigm to improve efficiency of video retrieval which can achieve both efficient use of wireless network resources and efficient detection of events. The detailed design of the Pub/Sub middleware will be discussed in Section 3.2.

3.2 System Architecture of Pub/Sub Based Wireless Video Surveillance

The proposed system architecture of Pub/Sub based wireless video surveillance is shown in Figure 14. In this framework, we utilize the decoupling characteristics of Pub/Sub in time, space and synchronization to assist the in-network processing in the asynchronous wireless environment. The middleware is running on wireless mesh nodes (as brokers and publishers) and mobile devices (as mobile subscribers). Mobile Users can subscribe events of interests from any place at any time. On the other hand, video stream transmission is driven by users' subscriptions. This feature greatly improves the efficiency of video transmission.

From the aspect of the subscribers, as shown in Figure 14, mobile users submit event subscriptions (attribute set) to brokers through the Pub/Sub middleware. Event subscriptions will be registered and disseminated among brokers. Brokers then register the event subscriptions to form a filter and update information of the subscribers (route information to the subscribers).

From the aspect of the publishers, the wireless sensor nodes in the bottom layer periodically collect data by using various sensors, i.e. using light sensor to detect passing through vehicles. The collected raw sensor data cannot be used to publish event notifications to subscribers. Instead, the sensor data will be first reported to wireless mesh nodes. The wireless mesh nodes, each of those is equipped with video cameras and acting as broker and video publisher, will then translate the raw data into structured attribute set. The parsed sensing information will be mapped to the attributes filters (subscriptions) maintained on the brokers. According to the mapping result, Pub/Sub middleware will decide whether to start the video streaming service. The middleware will also return a list of matched subscribers together with the route information for propagating the published events (including video stream).

We will introduce system components of the proposed framework in Section 3.3. The Section 3.4 presents the main functions provided by our system. We will discuss the construction of interconnected WSNs and WMNs in Chapter 4. And a customized delay constraint Pub/Sub design will be introduced in Chapter 5.

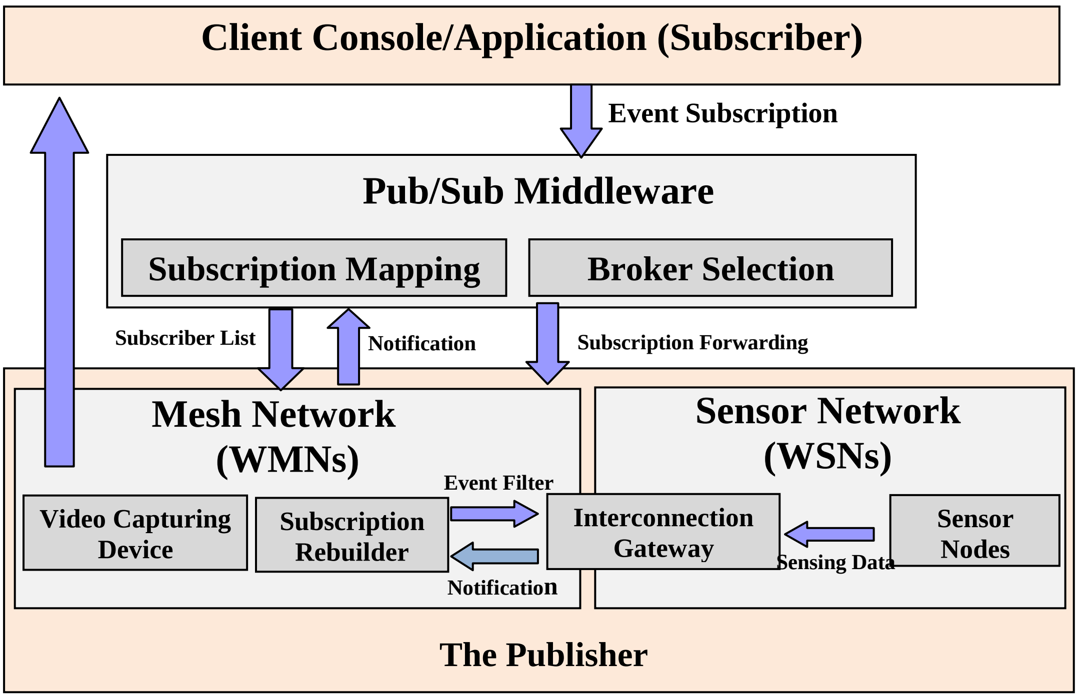


Figure 14 Architecture of the Pub/Sub based wireless video surveillance

3.3 System Components

The Pub/Sub based video surveillance System consists of three types of interconnected components: Event publisher (EP), Event Broker (EB) and Mobile Subscriber (MS).

EPs are the publishers that publish the rich content event notification to the brokers. Rich content event notification is defined as notification that not only has text notification but also contains video stream. The text notification is a set of attribute-value pairs, just the same as it in the traditional Pub/Sub system. The text notification is used to map subscriptions and events. When an event notification reaches the matched filter, the reverse path will be set for transmitting the video stream contents.

EBs are distributed over the whole network and together form the Pub/Sub overlay for event/subscription matching and delivery. When a subscription is received, the EB will forward it over the overlay. The registered and forwarded subscriptions will be further formed into a content filtering tree over the broker overlay [18]. When an event notification is received, the content filtering tree will map it to the filter and forward it to subscriber(s) if matched.

The MS registers subscriptions to EBs. The matched event notification will be forwarded to the MS. MS can disconnect from one broker and connect to another broker while roaming in the network.

The system generally works as follows.

a. The EP (camera node) uses text notification (containing pairs of <attribute, value>) to find interested group.
b. When an event is detected, the event publisher first publishes the

notification to the nearest EB, and the EB forwards the notification to EBs to find interested group.

c. The EB that has the matched filter for the received notification will send a reply and set a reverse path to the EP.
d. EP then multicasts the video streams to the interested group (MS).

We will introduce the detail strategies of our system in the following Chapter 5. After that, we will formulate the problem of scalable content-based Pub/Sub with delay constraint and give a heuristic algorithm to solve the problem.

3.4 Main Functions

In this part we will explain the detailed system functions provided by our Pub/Sub design for subscription forwarding and matching.

3.4.1 Subscription

Here we describe the “Subscribe” function. The subscribe function includes registration of a subscription and forwarding of the subscription in the broker overlay. Figure 15 shows the algorithm on the broker side for “Subscribe” function.

```
On broker B receives subscriptions f
Compare f with the root subscription filters f'
        //the root subscription f' defines most generic filter covering all others
        //The subscriptions already covered will not be forwarded
        If f' includes f
                Register the f on B
                Stop forwarding f
        //the subscriptions defines new notification

        Else
                Replace the root subcription filter f' with f
        //prepare to forward the new subscriptions
        //check if the subscription filtering tree already exsits
        If B belongs to the existing filtering tree T
                Forward f along the T
        Else
        //the algorithm to select broker will be introduced in the later section
                Select the next hop broker B'
                Forward f to B'
                Add B' to construct the filtering tree T
```

Figure 15 Algorithm for Subscribe

When subscriptions are first propagated from MS to directly connected broker, the broker will use the similar strategy with Siena [7] to decide whether to forward it. The EB first checks the maintained content filtering structure (usually is a tree) to see if the received subscriptions contain newly defined notifications that are not yet covered by existing filters (E.g. “airport = *any*” covers the subscription with filter “airport = *Hong Kong*”). If a new selectable notification is detected, the subscription will become the root subscription and be propagated to other brokers belonging to that filtering tree.

If a broker itself already belongs to one connected content filtering tree that calculated previously, the broker will just forward the subscription along the set of nodes that form the content filtering tree. If a broker does not connected to any existing constructed filtering structure, it will use a localized algorithm to decide how to propagate the subscription. The algorithm will be discussed in later part.

3.4.2 Notification

Here we introduce the “Notify” function of our Pub/Sub. The notify function includes event notification and video multicasting. The major difference in the notification procedure is that, our notification contains both event notification and video content of the event. Thus our notification is not from brokers to subscribers. In our notification procedure, brokers only match event notifications to subscriptions, and return a list of matched subscribers to publishers. Publishers then will publish the video streams directly to subscribers. Figure 16 shows the algorithm on the broker side for realizing the notify function.

When an initial broker firstly receives a new notification from an event publisher, it instantly forwards the subscription within several hops. The number of hops will be discussed in Chapter 5. When neighboring brokers receive the notification, they will look up the filtering structure maintained by them. In case a matched subscription/filter is detected by a broker, the broker will send a reply to the initial broker of that notification and set up a reverse path back to the initial broker. The reverse path connects the initial broker to the matched content filtering tree. When the initial broker has received all the replies, it will transmit the video stream along the connected matched content filtering tree.

```
On broker B receives notification E
If E is received from publisher
        //Check if there is root subscription f' registered on B
        If f' = NULL
        //the broker B is on the subscription filtering tree T
                Forward the E to the neighboring brokers with the lifetime K
( hop)
        If f'!= NULL
        //B is on the subscription filtering tree T
                Compare E with the root subscription filters f'
                //check the notification is matched
                If f' includes E
                        Set reverse path to the publisher
                        Forward subscriber list to the publisher
                //the subscriptions is not subscribed by any subscribers
                Else
                        Discard E
Else
// the notification is received from other broker
        If f' = NULL
        //the broker B is not on the subscription filtering tree T
                Set notification lifetime K = K-1
                Forward the E to the neighboring brokers with the lifetime K
(hop)
        If f'!= NULL
        //B is on the subscription filtering tree T
                Compare E with the root subscription filters f'
                //check the notification is matched
                If f' includes E
                        Set reverse path to the publisher
                        Forward subscriber list to the publisher
                //the subscriptions is not subscribed by any subscribers
                Else
                        Discard E
```

Figure 16 Algorithm for Notify

3.4.3 Unsubscribing

"Unsubscribe" function cancels previous subscriptions, but it is not exactly the inverse of subscription. It is more complex to handle and sometimes more expensive in terms of communication. The reason is that a single unsubscription needs to support cancellation of more than one previous subscriptions. Following Figure 17 shows our algorithm to handle "Unsubscribe".

```
On broker B receives Unsubscription f
//Check if there is root subscription f' registered on B
If f' = NULL
//the broker B is on the subscription filtering tree T
        Forward the unsubscription f to the neighboring brokers
If f'!= NULL
//B is on the subscription filtering tree T
        Compare f with the root subscription filters f'
        //check the notification is matched
        If f' includes f
                Deregister f on B
        //the unsubscriptions is the root subscription
        Else
                Deregister f' on B
                Recalculate the new root subscription f'' on B
                Forward f'' along the T
```

Figure 17 Unsubscribe Algorithm

3.4.4 Mobile Subscriber Handoff

Mobile Subscriber (MS) is connected to one EB and can freely roam among different brokers. Subscribers may move during an event delivery process, thus a handover operation is required to handle the event content forwarding to the new location of the subscribers.

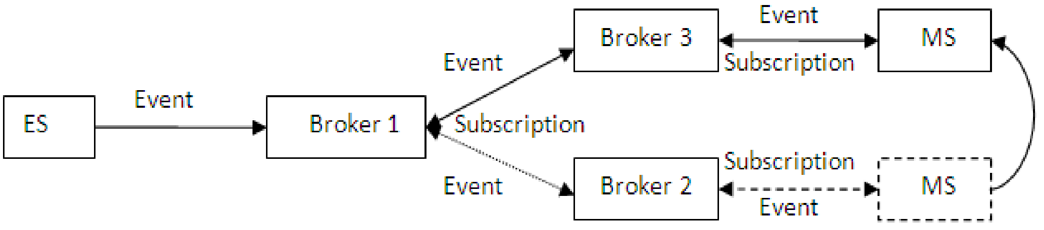


Figure 18 Handling mobile subscribers

Figure 18 shows a sample event matching procedure with a mobile subscriber. When a subscriber hands off from one broker to another, it first resubmits its subscription to the new broker. The new broker receives the subscription and updates the routing information from the subscriber to other brokers.

3.5 Summary

We propose a novel design of Pub/Sub system for real-time video surveillance service. We target at designing a scalable content based Pub/Sub system to support real-time video surveillance in large scale wireless mesh networks. The proposed system is one part of our HAWK project [9]. As far as we know, we are the first one that use Pub/Sub paradigm to support real-time video streaming service. We have implemented a prototype system for the proposed Pub/Sub based video surveillance, and have collected results for field testing.

CHAPTER 4 INTERCONNECTED WSNS AND WMSS FOR PUB/SUB VIDEO SURVEILLANCE

In this chapter, we propose a complete video surveillance system design on top of the interaction between wireless sensor networks (WSNs) for event subscription and wireless mesh networks (WMNs) for video transmission. The interconnection of these two types of networks provides new features in wireless video surveillance systems and enables new services, but it also brings new challenges for the system development. We discuss the challenging issues and propose our interconnection strategy. Prototype system is implemented and testing results are collected to show that our solution can fulfill the video surveillance requirements in the wireless network environment.

4.1 System Design

4.1.1 Challenging Issues

We first look into the challenging issues for interconnecting WSNs and WMNs. Then, we introduce our strategy for constructing a heterogeneous network composed of these two types of networks.

	Sensor Nodes	Mesh Nodes
Size	Small or tiny	Larger
Power	Small battery or energy harvesting	External power source (DC power)
Price	A few dollars or less	Expensive (hundreds of dollars)
Processing capability	Low	Much more powerful
Packet size	Small (bytes)	Much larger (Kbytes)
Data rate/Bandwidth	Low (kbps)	High (Several Mbps)
Interface Card	Single	Multiple (multi-radio, multi-channel)
Identification scheme	Simple ID based	IP address
Routing	None or simple routing protocol	Support complex routing protocol

Table 1 Comparison between WSN nodes and WMN nodes

With all these differences, to interconnect these two types of networks, we must address the following challenging issues [5].

1) **Different wireless technologies**: A communication link cannot be set up using two different wireless technologies. At least one device or group of devices should be able to receive and send both sensor packets and mesh packets. Note that, although direct links cannot be set up, the two different networks may interfere with each other as they possibly work at the same frequency. (ZigBee and 802.11b/g both work at the 2.4GHz frequency).

2) **Different packet formats**: Packet sizes and synchronization strategies are quite different. These differences make the interconnection more complex. In WSNs, the packet size is often about over ten bytes, while WMNs are using Kbytes packets.

3) **Different addressing schemes and routing strategies**: Although both WSNs and WMNs can form routing paths using self-organized routing protocols, route calculation is based on different metrics. Also, WMNs support IPv4 addressing, but WSNs only use simplified addressing schemes.

4) **Different information granularity**: WSNs are data centric and are usually designed for metadata collection, while WMNs are often service oriented and are designed for processing and transmitting more comprehensive content. Thus content translation is necessary for the interworking between WSNs and WMNs.

5) **Security issues**: Due to limited calculation capacity of WSNs, the security technologies used in WMNs interconnected cannot be directly ported to the sensor motes. E.g. pre-shared keys or certificate based encryption techniques [47]. Thus, sensor networks are more vulnerable to attacks.

The proposed strategy for interconnecting WSNs and WMNs will address challenging issues motioned above. And we will discuss more about the content translation for the Pub/Sub between WSNs and WMNs in Section 4.1.3.

4.1.2 Strategy of WSN-WMN Interconnection

The traditional solution for interconnecting two different types of networks is to utilize a gateway device. A gateway is a multi-interface network node, which has necessary protocols installed to communicate with both types of networks. In case of WSN and WMN networks, a simple gateway strategy is depicted in Figure 19. In this scenario, all sensors propagate their values towards a central gateway device, which acts as a sink for the sensor network. The gateway devices stores all received sensor values in a database. With a single gateway for interconnection, the system is not scalable, and the gateway node may also become a bottle neck node.

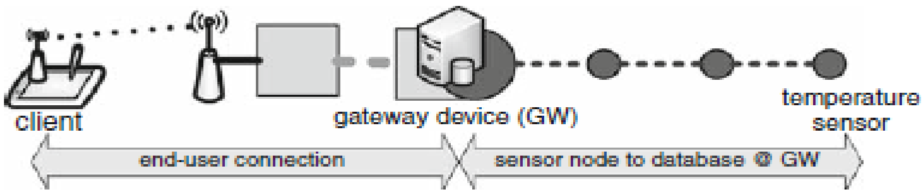


Figure 19 Single-gateway solution between WSNs and WMNs

Our strategy of interconnecting WSNs and WMNs is using a multi-gateway structure to construct a scalable heterogeneous network. This strategy is also introduced in [5]. But in our system, we do not require any kind of packet format translation.

We use wireless mesh nodes to form the Pub/Sub broker overlay network for mobile client to register the subscriptions. We use TelosB mote as gateway node, and every TelosB mote will connect with a batch of MicaZ motes to form a small cluster for event detection. Every small cluster is used for sensing data in one particular area. Every TelosB gateway node is attached to a mobile wireless device (notebook), which is a client connecting to the wireless mesh networks.

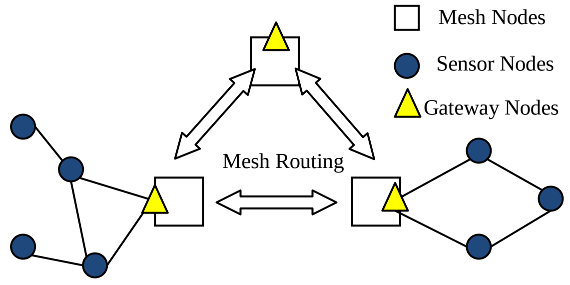


Figure 20 Communications between WSNs and WMNs

With such network hierarchy (Figure 20), the WSN is invisible to the upper layer mesh nodes. Every sensor cluster can be addressed with an IP address of the mobile device that the TelosB gateway attached. The underlying WSN does not require to directly access to the WMN nodes and all the WSN packets within a cluster are collected by the TelosB gateway nodes. The TelosB gateway can interact with other TelosB gateway nodes through the mesh route path by using socket communication.

Location ID	Block A	Block B	Block C	….
IP Address	192.168.20.0	192.168.21.0	192.168.22.0	….

Table 2 The Location-IP mapping table maintained at gateway nodes

In our system, we use IP address to associate the location information to the WSN sensing area and the camera nodes. We further build a location-IP mapping table (Table 2), which is maintained by every mesh node, to help associate the underlying WSN small cluster with the IP address of the gateway node. The mapping table will be automatically generated when a gateway node is detected, and the mapping information is shared by all mesh nodes.

4.1.3 Event Subscription and Detection

Publishers are distributed in both WSNs and WMNs. The differences of processing and transmission capability of WSNs and WMNs make it difficult for these two types of networks to “understand” each other. Some existing systems introduce solutions to support full translation between WSNs and the WMNs. This means that WMN nodes can directly access WSN nodes. However, this kind of full translation is very costly and usually makes the system less scalable. In our Pub/Sub based video surveillance system, we focus on the translation of the user subscribed events of interests. For example, a user can subscribe video content for the “traffic jam occurred in block C”. Without any translation, the WSN cannot understand the event not mentioning to help detect the traffic jam.

Our solution is to register the content translation rules in the rule engine (also compiler [26]) installed at the gateway nodes. In our system, we define the rule sets for translating the events subscribed in WMN into the primitive events that can be detected by the WSNs.

We have predefined translation rules for three types of events on the gateway node. The mesh subscriber is required to subscribe with event type and event location information. The following example describes the way of translating content.

Assuming one surveillance user subscribes the event of “traffic jam” in “block C” to one mesh broker.

1) By using the predefined rules, we translate the "traffic jam" into the following event presentation.

```
Event SimpleEvent {
    int event_Type =Jam_id ;
} where {
    number_of_cars > 3 and
    avg_pass_time > 10}
```

2) We use the location-IP mapping table to find the interconnection gateway, which is connecting the sensor clusters for block C.
3) The translated event subscription will be registered at the gateway node.
4) The gateway nodes map the subscriptions with the collected sensing data from the underlying sensor motes.
5) When a subscription is matched, the camera node associates with "block C" will be informed to publish the video to the subscriber.

4.2 System Implementation

In this section, we introduce the implemented video surveillance system over our intelligent transportation system (ITS) platform.

4.2.1 WSN Setup and the Interconnection Gateway

The proposed surveillance system uses our intelligent transportation system (ITS) platform as the underlying sensor network for event detection. We deploy sensors on the street model. The sensing data is collected and transmitted by using the Crossbow MicaZ motes. We use our self developed model cars (Figure 22) to invoke events. The model cars are also equipped with MicaZ motes. The sensor deployed on the road can help the model car to learn its location information and the information of the nearby cars. The model cars are also adapted to be able to receive remote control signal sent from the sensor motes deployed on the road side.

We use TelosB mote as the gateway node. Every TelosB mote is attached to notebook computer which is connected to the wireless mesh network. The road-side MicaZ motes transmit the data to the TelosB gateway node for further subscription mapping.

Figure 21 Street model of our ITS platform

Figure 22 Adapted model car with MicaZ mote.

4.2.2 WMN Construction for Video Transmission

We use our self developed T902 wireless mesh nodes as the surveillance nodes to capture and transmit video stream with the Pub/Sub middleware. The T902 also performs as a broker. Each T902 node has three wireless network interface cards (NICs) including one 802.11b/g NIC for client access and two 802.11a NICs for backbone transmission. Each surveillance node is connected with TelosB mote to capture events with the help of underlying wireless sensor networks.

The WMN part consists of following two components, video capture and compression module, and client console for video retrieval.

Video capture and compression module (Figure 23): We use Hikvision Compression/Decoding Card to produce H.264 video streaming data with CIF image format and 15fps frame rate. The Hikvision DVR card provides API for supporting video retrieval callback by network users, and we can use the surveillance node as a video server. The server side code is implemented using python.

Client console: We have implemented the GUI and the video retrieval module on the mobile handheld for user to access the captured video content (Figure 26). Figure 24 shows the messaging structure of the client console. And Figure 25 shows the procedure of communication establishment for client to retrieve the video.

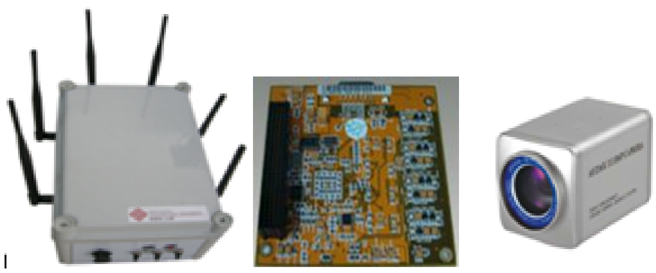


Figure 23 T902 mesh node and video capturing hardware

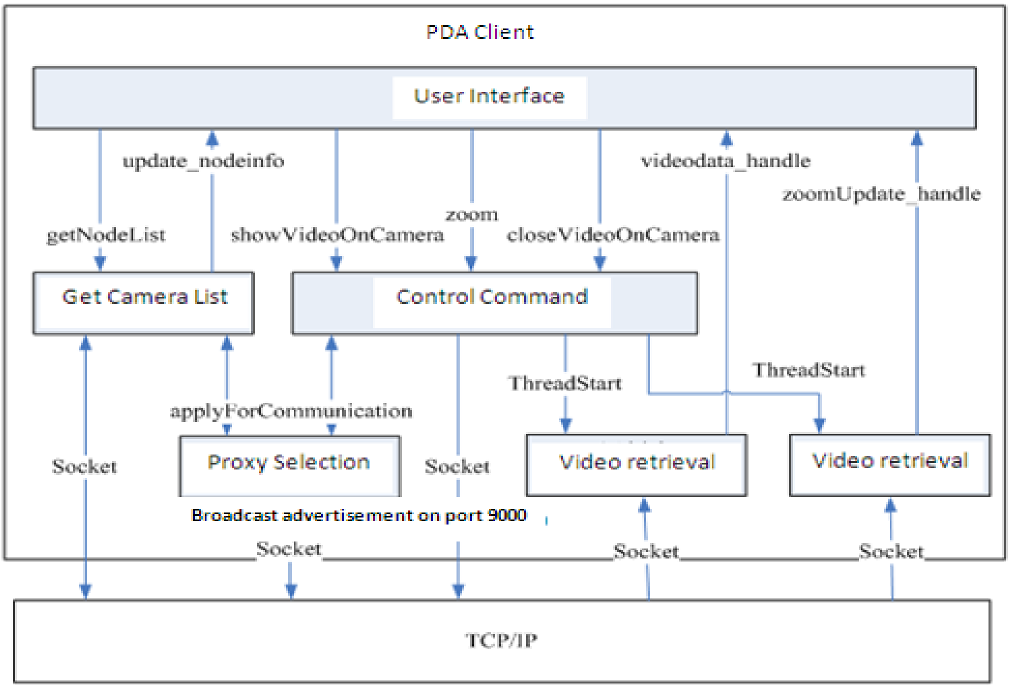


Figure 24 Messaging stucture of the client console

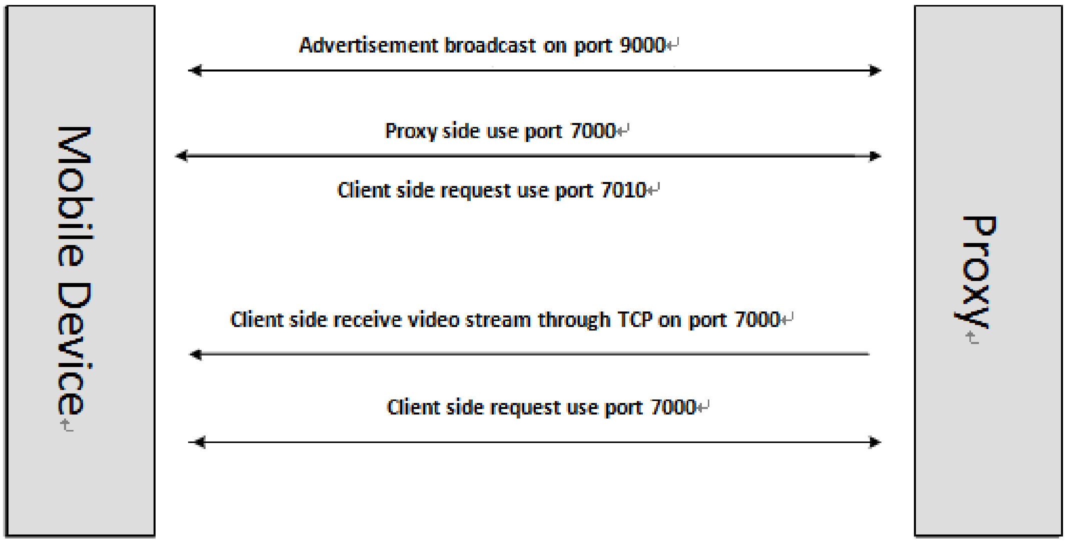


Figure 25 Communication process for the video client console

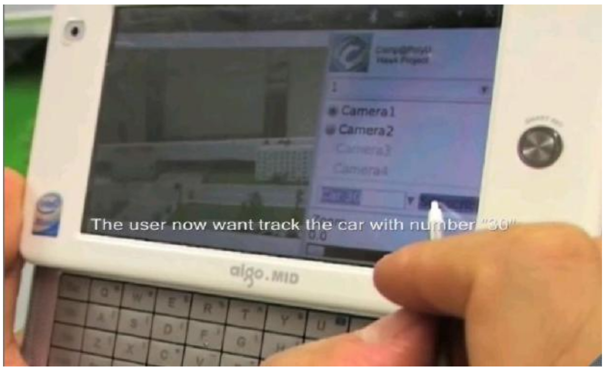


Figure 26 Client GUI that enables users to subscribe events

Client console has three main functional features: broker selection, video retrieval and camera control.

1. Broker selection: The client first broadcasts messages to find a broker to handle. After the connection is set up between the client and the broker, camera list will be updated on the client console. User can then manually select a camera to view.

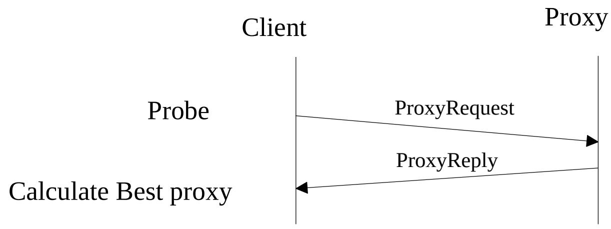


Figure 27 Broker selection procedure

2. Video retrieval: User can click camera ID on the camera list to view the real-time captured video content. The message passing procedure is shown in Figure 28.

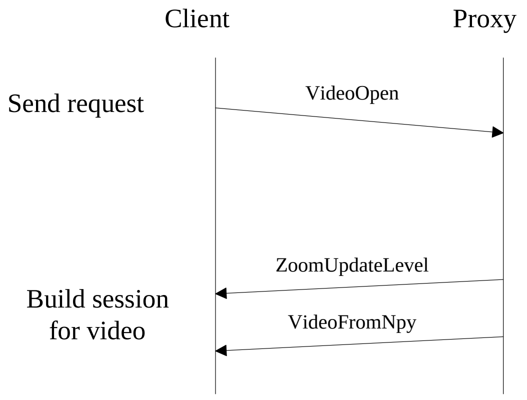


Figure 28 Start video streaming

3. Camera control: The camera control command is first sent to the broker that the client connecting with and will then be redirected to the selected camera node.

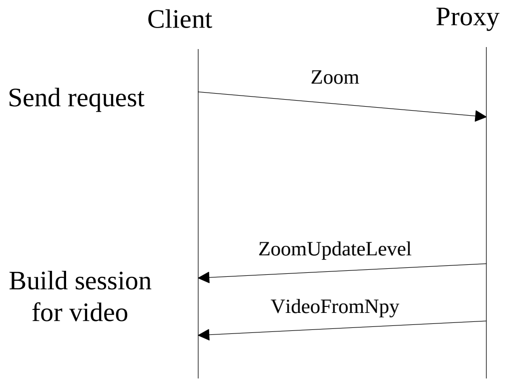


Figure 29 Camera control

4.2.3 Pub/Sub on Top of the Heterogeneous Networks

We have implemented a cross layer middleware for subscription registration and event matching. In current stage, our Pub/Sub middleware disseminates the subscriptions to every broker to form the content filtering structure.

In our prototype system, we configure every mesh router as a broker. OLSRd is a well known application layer daemon program for implementing the OLSR routing protocol used in wireless ad-hoc networks. In OLSR protocol, the HNA message is a predefined message type to handle routing updates for mobile clients. We modify the OLSRd and the HNA message type to integrate subscription registration. The modified OSLRd can handle the mobile user handoff to support mobile subscribers in the wireless network environment.

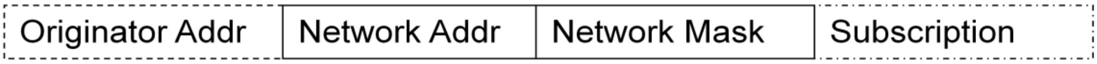


Figure 30 Modified HNA message

After the modification, the HNA message not only carries routing information of the client but also the predefined event subscription.

The subscription registration procedure is triggered by a MAC layer inverse-ARP like protocol. When a subscriber first connects to a T902 node, a modified inverse-ARP request will be sent to the brokers carrying with the IP information and the subscriptions. The subscription registration and forwarding procedure is shown in Figure 31.

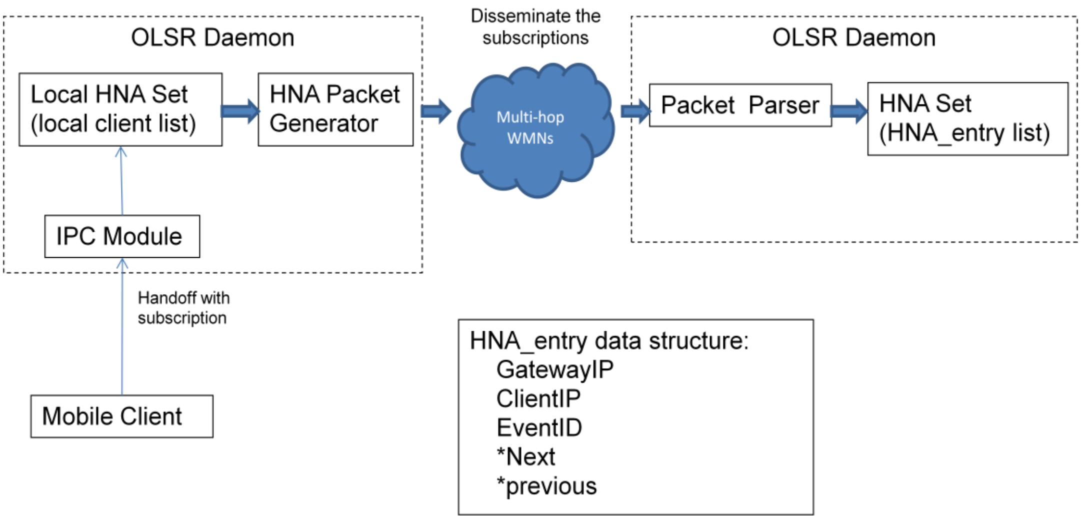


Figure 31 Subscription registration and dissemination

Subscriptions will be maintained in all the brokers. The mesh camera nodes do the event/subscription matching locally and then forward the video content to the client. Figure 32 shows the event detection and the publishing procedure.

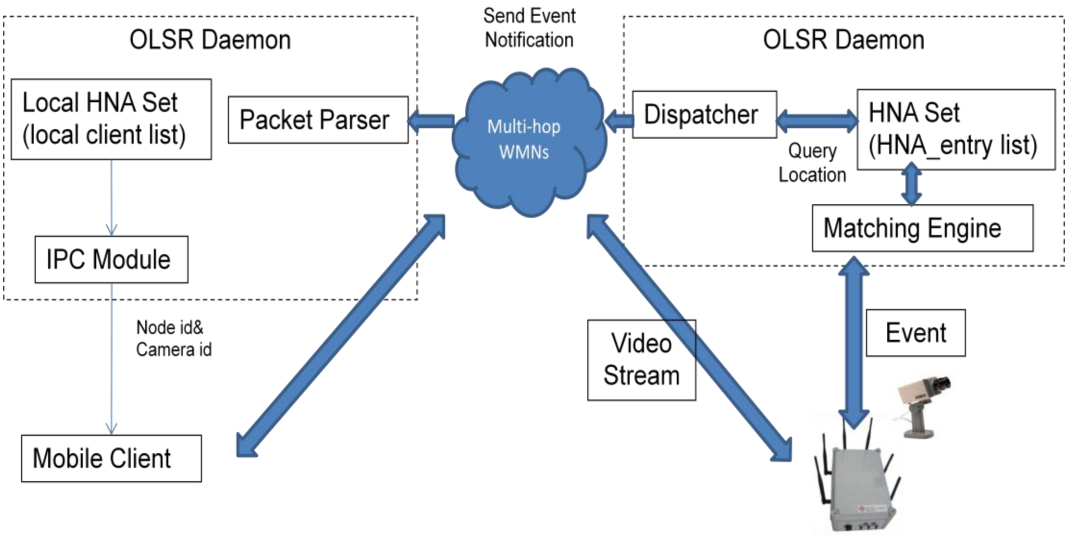


Figure 32 Event notification and video publishing

Events will be detected by wireless sensor networks and be matched at the TelosB gateway nodes.

4.3 **Deployment and Field Testing**

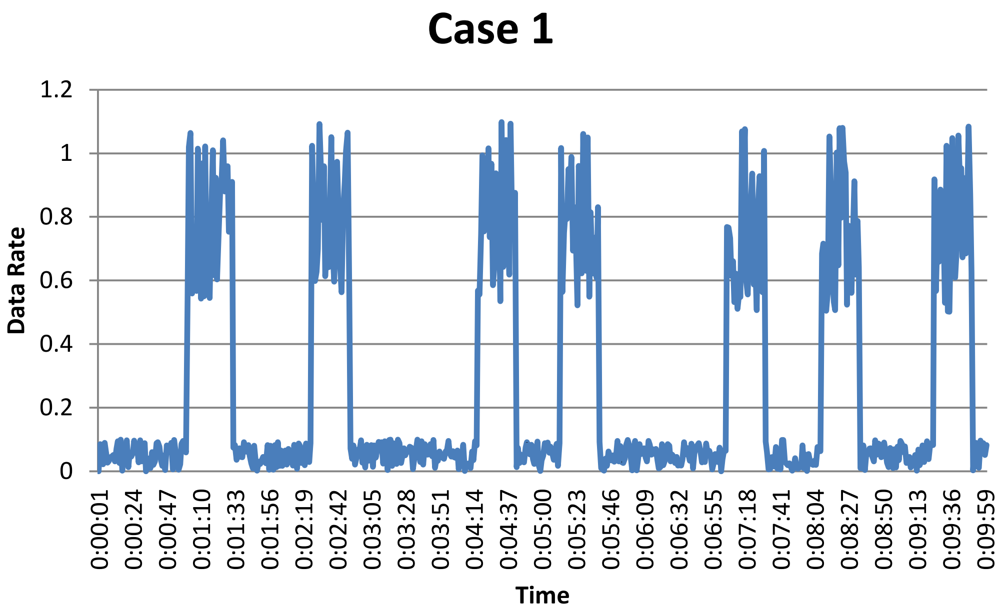
We deployed 8 T-902 nodes (Figure 34) to form a multi-hop wireless mesh network in one of our campus buildings. As the deployment is in-door, the per-hop distance is short with the average per-hop distance of 7 meters. We collected some experiment results from the deployed system. We measured the data rate for video stream transmission and find that the peak transmission rate can reach 1Mbps. As our wireless backbone can support much higher data rate, the video stream can be smoothly retrieved on the client side. Table 3 shows the performance of our wireless mesh network for video transmission.

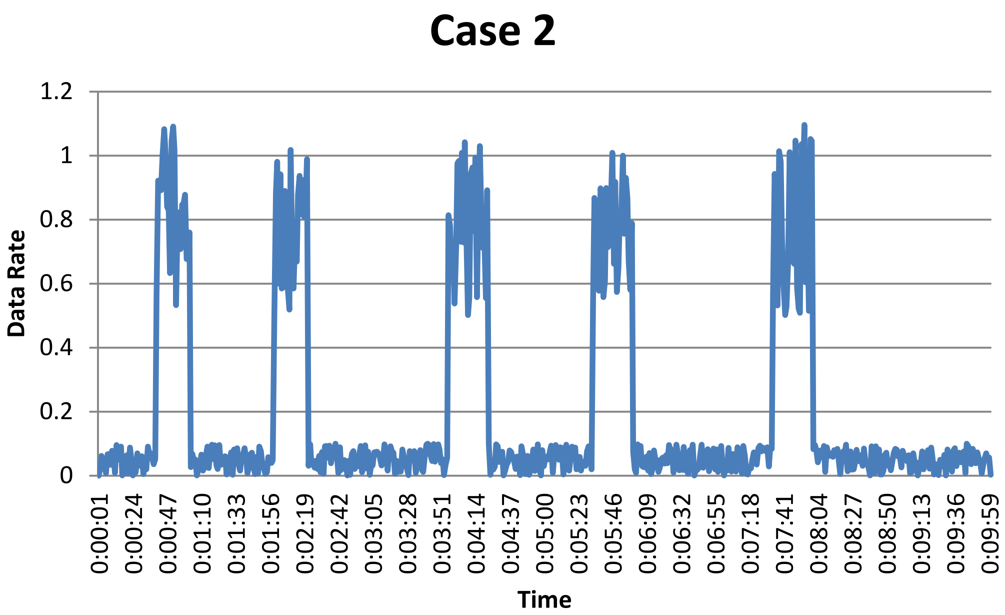
Average Per-hop distance	10meters
Per-hop Transmission Delay (average)	12 ms
Network Diameter (in hops)	3
Video Stream Data Rate (maximum)	1 Mbps
Subscription propagation delay (3 hops)	50 ms

Table 3 Key performance of Trafficast system

In the field-testing, we set up one mobile device as the subscriber. Several model cars with car ID configured in the MicaZ mote were set to move randomly on our ITS platform. The clients subscribed events through the Pub/Sub middleware and received the published video content from the video surveillance system. Subscriber can randomly subscribe a simple event, which consists of two attributes, namely car ID and area ID. We set up three rounds of simple test cases, in which the client subscribed the appearance of a particular car (car ID) in a specified area (area ID). The localization of model car was captured by the sensor. The following figures show the result of three random test cases. We captured data rate from the WiFi network interface of mobile devices to observe the users' wireless resource usage.

In our solution, the video frames will only be captured and transmitted when the detected car ID and area ID matches the subscribed event. Compared to the constant video stream retrieval, the Pub/Sub paradigm greatly saves the wireless bandwidth. The system only publishes and transmits video stream to the mobile user when the subscribed event is detected.


Case 1
Data Rate
1.2
1
0.8
0.6
0.4
0.2
0
0:00:01
0:00:24
0:00:47
0:01:10
0:01:33
0:01:56
0:02:19
0:02:42
0:03:05
0:03:28
0:03:51
0:04:14
0:04:37
0:05:00
0:05:23
0:05:46
0:06:09
0:06:32
0:06:55
0:07:18
0:07:41
0:08:04
0:08:27
0:08:50
0:09:13
0:09:36
0:09:59
Time


Case 2
Data Rate
1.2
1
0.8
0.6
0.4
0.2
0
0:00:01
0:00:24
0:00:47
0:01:10
0:01:33
0:01:56
0:02:19
0:02:42
0:03:05
0:03:28
0:03:51
0:04:14
0:04:37
0:05:00
0:05:23
0:05:46
0:06:09
0:06:32
0:06:55
0:07:18
0:07:41
0:08:04
0:08:27
0:08:50
0:09:13
0:09:36
0:09:59
Time

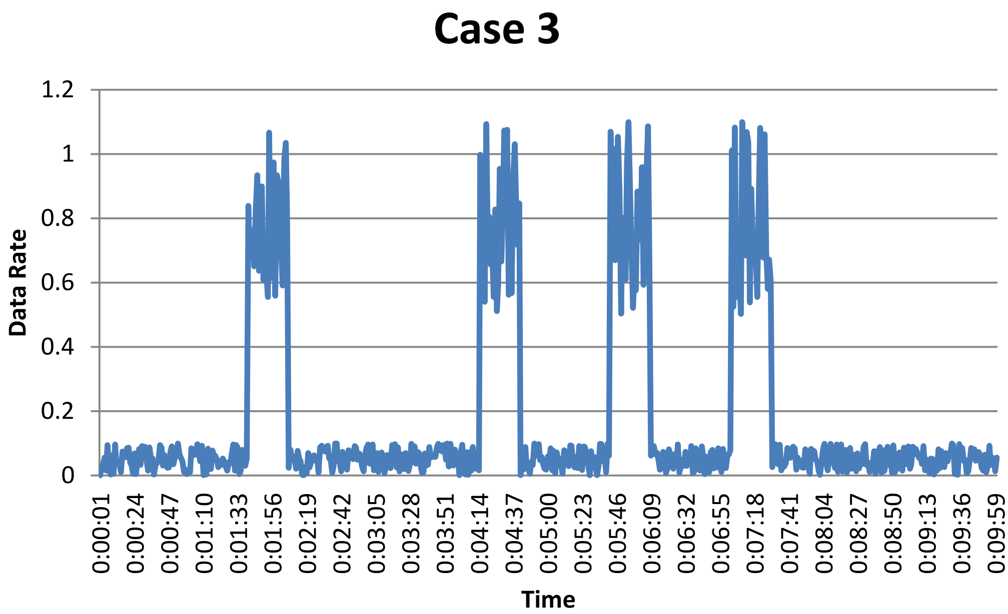


Figure 33 Client data rate while receiving video stream from surveillance system

Figure 34 The deployment of surveillance system for traffic monitoring

We also tested the subscriber handoff function. When a subscriber subscribed to an event, the subscription reached all brokers within 50ms. When the subscriber switched from one broker (T902 node) to another broker (T902 node), the handoff procedure took less than 200ms to update the routing information on the most distant T902 node. The demonstration is available in [33].

4.4 Security Extension

The video surveillance service is highly confidential and should be only available when the user is authorized from the system. The video content together with the user subscribed event should be protected in the venerable wireless environment.

We give a preliminary solution to provide the secure video transmission in the wireless environment (transport layer).We assume an honest-but-curious model for publishers, brokers and subscribers, as in [48]. This means that the entities follow the protocol, but may be curious to find out information by analyzing the messages that are exchanged. For example, a broker may try to read the content of an event or try to learn the filtering constrains of subscribers. Subscribers may want to read the events delivered to other subscribers. We also assume that a passive attacker outside the Pub/Sub system be able to listen on the communication and invade the privacy of the participants. Due to the shared medium characteristic, our work focuses on the protection of the published video content.

4.4.1 TLS Based Secure Video Transmission

To protect the publisher/subscriber communication and prevent the retrieved video streaming data being eavesdropped and tampered, an efficient way is to provide mutual authentication between the security context transfer source and the target. The TLS protocol allows client/server applications to communicate across a network in a way designed to prevent Man-in-the-middle attack. The work flow of our solution is as flowing.

1. A TLS client and a TLS-enabled server negotiate a connection (with state) by applying a handshaking procedure. During this handshake, the client and the server agree on various parameters used to establish the connection's security.
2. The handshake begins when the client connects to the server, requesting a secure connection, and presents a list of supported CipherSuites (ciphers and hash functions).
3. From this list, the server picks the strongest cipher and hash function that it supports, and notifies the client of the decision.
4. The server sends back its identification in the form of a digital certificate. The certificate usually contains server name, trusted certificate authority (CA), and server's public encryption key.
5. The client can optionally contact the server that issued the certificate (the trusted CA as above), and confirm whether the certificate is valid before proceeding.

4.4.2 Deployment

We have 3 different TLS sessions to keep protecting the published video content. Session 1 and Session 2 are used for interactions between brokers and subscribers. Session 3 is used for establishing a secure tunnel between different subscribers (optional). The open source GNUTLS kit for the TLS session establishment is used to provide the TLS sessions.

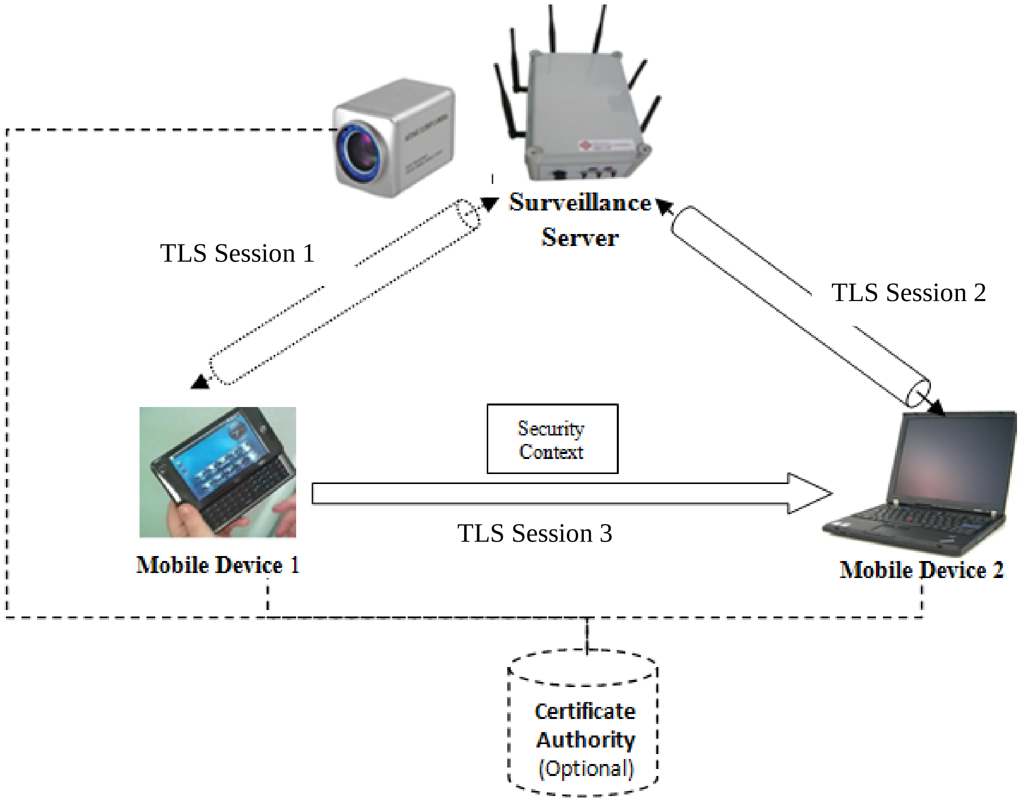


Figure 35 TLS session establishment for video transmission

4.5 **Summary**

In this chapter, we propose a novel system that utilizes Pub/Sub middleware to provide efficient video surveillance service over the interconnection of WMNs and WSNs. In order to apply the Pub/Sub paradigm to the proposed system, we investigate the issues of interconnecting WMNs with WSNs and then propose a practical solution. The proposed surveillance system considers the content translation between WMNs and WSNs to enable the Pub/Sub service between these two types of networks. We have already implemented a real system on our intelligent transportation system platform for traffic monitoring.

CHAPTER 5 DELAY CONSTRAINED PUB/SUB FOR VIDEO SURVEILLANCE

As the scale of wireless networks getting larger, the Pub/Sub system has to handle massive number of subscriptions. The content filtering structure, which is usually a tree, becomes very difficult to maintain. Thus, scalability is always an important consideration in designing the content based Pub/Sub system. The propagation and distribution of subscriptions/filters is the key to this scalability issue.

As introduced in Chapter 1, we target to design a scalable content based Pub/Sub system to support real-time video surveillance in large scale wireless networks. In this chapter, we investigate the problem of subscription dissemination in scalable delay constrained Pub/Sub on top of the wireless mesh/ad-hoc networks. Comparison among different Pub/Sub work distribution modes (in Chapter 2) shows that, the more brokers involved in the content filtering structure, the less time we spend on finding interested subscriber groups. This tradeoff must be carefully considered. Our objective is to use as few brokers as possible to manage the subscriptions/filters and meanwhile control the total time latency to be less than a certain threshold to support real-time video surveillance applications. We prove that the problem can be reduced to a minimum connected K-dominating set problem, which has been proved to be NP-complete. Two heuristic algorithms are then proposed to solve the problem, followed by a performance evaluation.

Unlike the Siena [7], our solution does not use advertisement or any other guidance for brokers to propagate subscriptions. Brokers use localized algorithm for propagating subscriptions, and Pub/Sub delay is controlled within the constraint. The video surveillance service is quite different from the traditional event notification service. As far as we know, we are the first to use Pub/Sub paradigm to support real-time video streaming service. Also, we are the first to jointly take delay constraint and scalability issue into consideration for designing Pub/Sub system.

We have implemented a prototype system for the proposed Pub/Sub based video surveillance and conducted field testing. The performance evaluation and testing results demonstrate that our solution works well. The remainder of this chapter is organized as follows. In Section 5.1, we give our problem formulation. A heuristic algorithm for subscription dissemination is proposed in Section 5.2. The Section 5.3 is the performance evaluation, followed by the field testing in Section 5.4.

5.1 The Broker Selection Problem

In this subsection, we give the problem formulation and introduce one heuristic algorithm to solve the problem.

In our case, total delay of event publishing is the period from the time that event occurs to the time that the last interested subscriber receives the event video content. Unlike the previous Pub/Sub works that only consider QoS issues for notification transmission, our work takes time latency, that caused by finding interested subscriber group, into consideration as well. Thus, in our delay model, total delay is further divided into subscription matching delay and notification transmission delay.

The subscription matching delay is caused when mapping notifications to registered filters (from the subscriptions). This part of delay is close related to the selection of the brokers that stores the filters and forms the content filtering structure. On the other hand, the transmission delay is used for transmitting the event to the subscriber from the event publisher.

The matching delay for interested group searching can be decided by the selected broker set *B*. We use $T_{match}(B)$ to represent matching delay. We use $\mathrm{T}_{\mathrm{transmit}}$ to represent the transmission delay. Let T_{delay} be the total delay and we have:

$$T_{delay=} T_{match}(B) + T_{transmit}$$

The following Tables 4 is the notation used in our problem formulation

B	The selected broker set to forward the subscriptions
Δ	The total delay constraint. The total delay is the period starting from the time the event occurs to the time that the last interested subscriber receives the event video
D	The upper bound of the transmission delay between any pair of nodes in the network
T	The upper bound of the transmission delay between any pair of neighboring brokers (one hop)
$T_{match}(B)$	The time required to match the notification and find all subscribers
$T_{transmit}$	$T_{transmit}$ is calculated from the time that first video frame sent out by camera node till this frame reaches to the last subscriber

Table 4 List of notation used in the problem formulation

Our problem can be formulated as: given a set of brokers and a set of subscriptions, the objective is to select the minimum set *B* of brokers to propagate and store the subscriptions and meanwhile subject to a delay constraint.

Given the constraint on total delay is δ. We formalize our problem as:

Min $|B|$.

s.t. $T_{match}(B) + T_{transmit} <= \delta$. (1)

We further make following assumptions.

a. We assume a uniform broker distribution.
b. The end-to-end transmission delay in the networks has an upper bound. The maximum delay in the network is D.
c. We assume the 1-hop transmission time between every two neighboring brokers is equal or less than t.
d. The network size is known to every broker (size here refers to the total number of nodes).

Based on the assumptions above, we then need to manage the matching delay to control the total delay. According to our observation on the relationship between the $T_{match}(B)$ and the $T_{transmit}$, we can get the following lemma and theorem.

Lemma 1: For every event publisher, if the content filter from all interested subscribers can be detected within K-hops, the $T_{match}(B)$ *then can be bounded within* $2K*t$ (in case flooding is used for looking for the filter).

Proof: As all matched filters can be found within K hops, we can get the time $K*t$

for finding the notification within K hops distance. Another Kt is used for all matched broker to set the reverse path back to the publisher. Thus, the total T_{match} can be bounded within $2K*t$. (End of proof)

Theorem 1: Given delay constraint δ, given the network end-to-end delay upper bound D as well as the one hop transmission delay t. If each notification can reach all matched filters within K hops, where $K \leq (\delta - D)/2t$, the total delay can be guaranteed to be less than the constraint δ.

Proof: $T_{match} + D \leq \delta$

$$=> T_{match} \leq \delta - D.$$

Based on the delay model and lemma 1, we can get

$$2K*t \leq \delta - D.$$

$$=> K \leq (\delta - D)/2t.$$

Thus, we can get a max K to select the minimum set of brokers which guarantee that the total Pub/Sub delay does not exceed the constraint. (End of proof)

The problem then can be reduced to the minimum connected K-dominating set problem, which has been proved to be an NP-Complete problem [4]. This problem is defined as follows:

Given a connected undirected graph G. Then, in graph G, to find a set of connected nodes S. The set S should have the property that every node in G is at most k edges away from at least one node in S and the size of S is minimized.

In our problem, we want to find the smallest set of brokers to forward the subscriptions. The advantage of finding this set is the maintained subscriptions filtering structure is simple even the whole network grows to a very large size. And meanwhile the total Pub/Sub delay is controlled within a threshold.

5.2 Design of Heuristic Algorithms

We consider the maximum end-to-end delay *D* of the network as the constant value for calculation of *K* (fixed K), we also can extend the algorithm by adjusting the value *K* according to the distance to the subscriber (dynamic K).

Construction of the minimum connected *K*-dominating set (*K*-CDS) is quite time consuming. Most of the existing algorithms for constructing the K-dominating set usually use several round to get the final set. As our major contribution is not on giving a best algorithm for finding the minimum K-CDS, we just use one light-weight single round algorithm for finding the set. The main idea is that when a broker needs to forward subscriptions, it will follow our algorithm selecting a subset of its one-hop neighbors to forward. By doing this, one connected *K*-dominating set will be incrementally constructed to propagate the subscriptions.

A brief outline of the broker selection algorithm run on each broker is shown in Figure 36. The *K* is pre-calculated and shared by every broker in the network. Whenever a broker receives a new subscription that needs to be forwarded, it will use the *K* to select the neighboring nodes.

```
K=floor((δ – D)/2t);
if received(subscription)
    if subscription defines new notification
        if subscription is from a subscriber
        //the broker is the initiate broker
         if can find (K+1)-hop neighbors
                select subset of1-hop neighbors;
                forward (subscription);
         else
         //the broker is a selected broker
             if can find non-shared (K+1)-hop neighbors
                    select subset of 1-hop neighbors;
                    forward(subscription);
    else
        do nothing
```

Figure 36 The broker selection algorithm.

Here, we make an additional assumption that every broker maintains a node set information of all brokers within *K* hop distance and exchange this set with its one hop neighbors. When a new subscription first registered at an initial broker, it will be forwarded and stored following the steps below.

a. The initial broker exchange its K-hop information with its one hop neighbors thus the broker can calculate the set of (K+1)-hop neighbors.
b. Then, the initial broker select a minimum subset of its 1-hop neighbor to K-dominating all its (K+1)-hop neighbors.
c. The selected neighboring broker will receive the subscription from the

initial broker.

d. The selected broker then can calculates its (K+1)-hop neighbors, which is not shared with the origin broker that forwards the subscription.

e. The broker then select a minimum subset of its 1-hop neighbors to K-dominating all its (K+1)-hop neighbors

f. In case the broker cannot find any (K+1)-hop neighbors that not shared with the origin broker, the broker will stop forwarding the subscription.

We propose an alternative solution which can adjust the value *K* according to the distance from the broker to the subscriber. In the dynamic *K* heuristic to disseminate the subscription copies over the broker overlay networks according to the distance to the subscribers. When the distance increases, the $T_{transmit}$ will increase correspondingly and the brokers have to adjust T_{match} to meet the delay requirement. The region far away from the subscriber will have higher subscription density, vice versa. The density of the subscription distribution is controlled under theorem 1. We use a function of distance *F* to estimate the $T_{transmit}$.

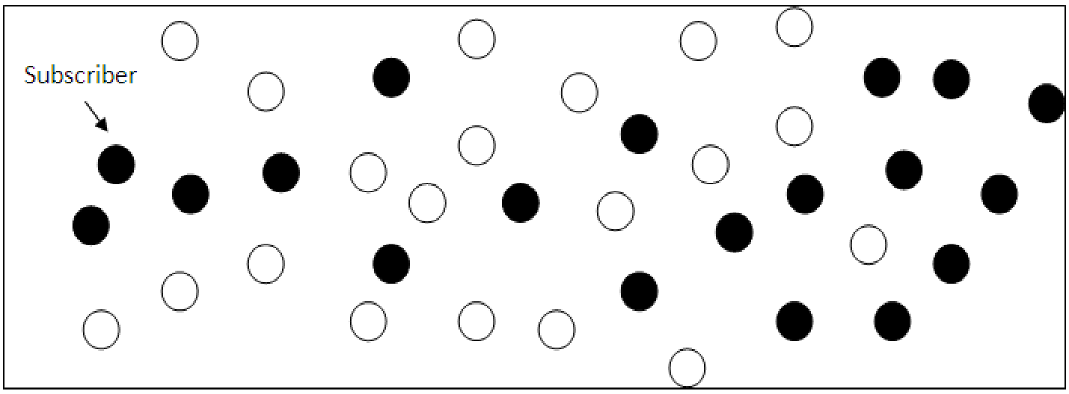


Figure 37 Density of the forwarded filter changes according to the location

Figure 37 above shows the scenario of *Dynamic-K*. The dark circles are the brokers with the subscription. The subscription density is adaptive according to the distance to the subscriber. Unlike the *Fix-k*, the *K* cannot be pre-calculated as different brokers have different *K* according to its distance to the subscriber. In *Dynamic-K*, every broker uses the following two steps to select neighbor to forward the subscription.

The distance from a event broker *j* (EB_j) to subscriber *i* (MS_i) is D_{ij}, and we can estimate the transmission time from EB_j to MS_i is as following

$$T_{transmit}' = F(D_{ij}).$$

$$\Rightarrow T_{match}' < \delta - F(D_{ij}) .$$

Suppose at EB_j, all subscription can be found within K_j hops, then we can get

$$2 K_j t <= \delta - F(D_{ij}).$$

$$\Rightarrow K_j <= (\delta - F(D_{ij}))/2t.$$

For construction of the minimum connected *K*-dominating set (*K*-CDS), we use the similar steps as *Fix K* to find the set.

5.3 **Performance Evaluation**

We use simulation to evaluate the efficiency of broker selection of our proposed algorithm for subscription dissemination. We first compare the selected broker set size between our *Fix-k* and the existing K-dominating set based network backbone selection algorithms.

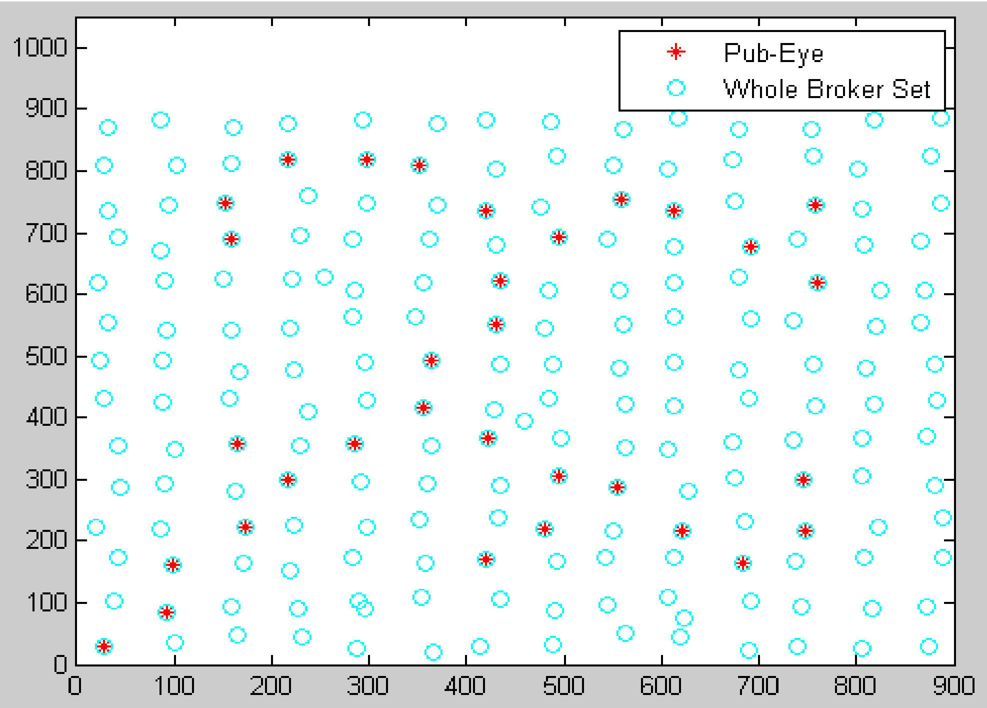
5.3.1 Fix K Algorithm

We conducted simulation using the Matlab. In order to compare with the existing works, we configured the whole network as a 900m $\times$ 900m 2D region to deploy brokers. We set up a random disk graph with each broker covering 100m communication range. The brokers within the 100m range will be treated as the one hop neighbor.

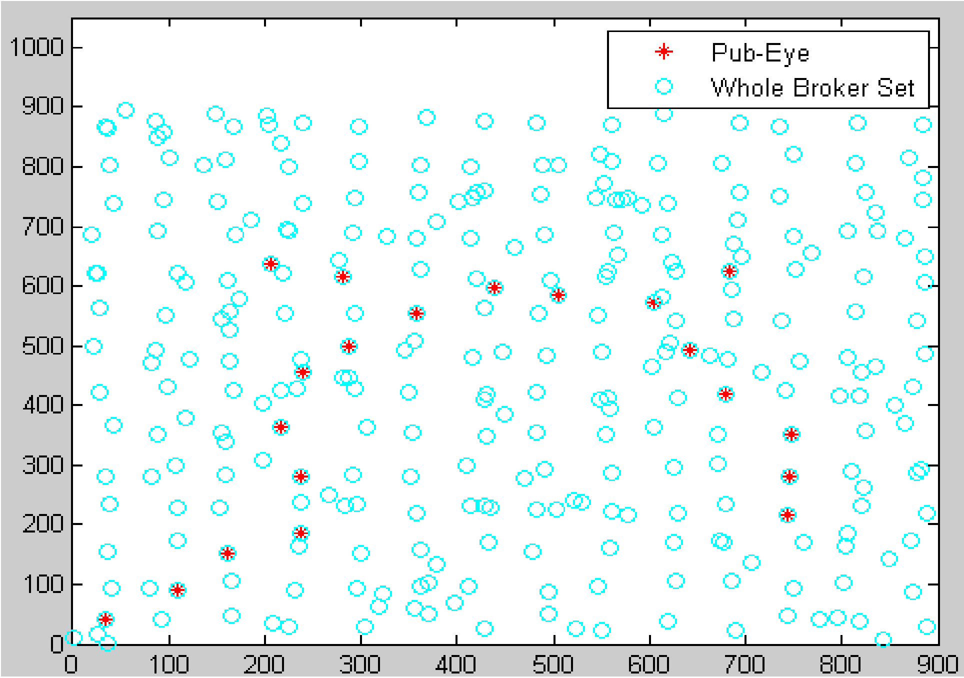
The value of K can be set by giving the network end-to-end delay upper-bound and the total delay constraint. According to our real implementation and testing, the 1-hop end-to-end delay can drift form less than 1ms to around 25ms, thus we set the one hop delay as 25ms to do the simulation. Note that the network transmission delay cannot just be simplified as No. of hops multiply the 1-hop delay. The multi-hop end-to-end delay can be affected by several factors, such as the queuing delay and the wireless interference, etc. We then tune the value of delay constraint δ to see the effects.

To start the subscription propagation, we randomly selected one broker (initial broker) to initiate a subscription to propagate to get the final size of selected brokers which stores the filter and forms the filtering overlay. We set the network end-to-end delay upper bound fixed to *D=200ms* and the one hop transmission delay *t*=25ms. The results are shown in the following Figure 38. The size of the network (N) varies from 200 brokers to 400 brokers.

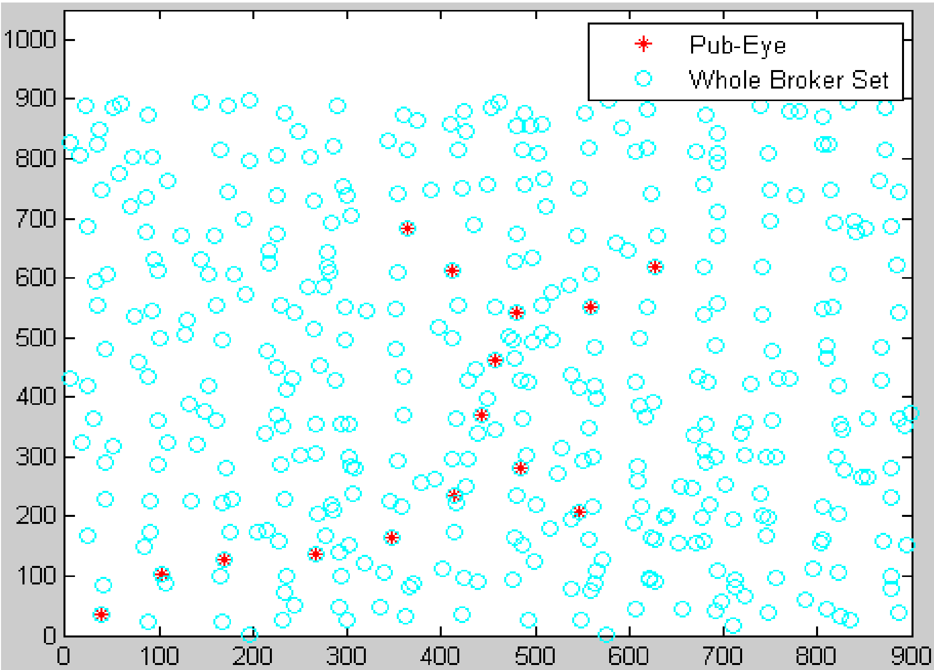
We adjust the value of the delay requirement δ to calculate different k. In Figure 38, the initial broker is set to the left-bottom corner to make it easy to view. The results show that we can select a very small portion from the entire broker set to propagate and store the subscription/filter.



(a) 200 nodes with the $K = 2$



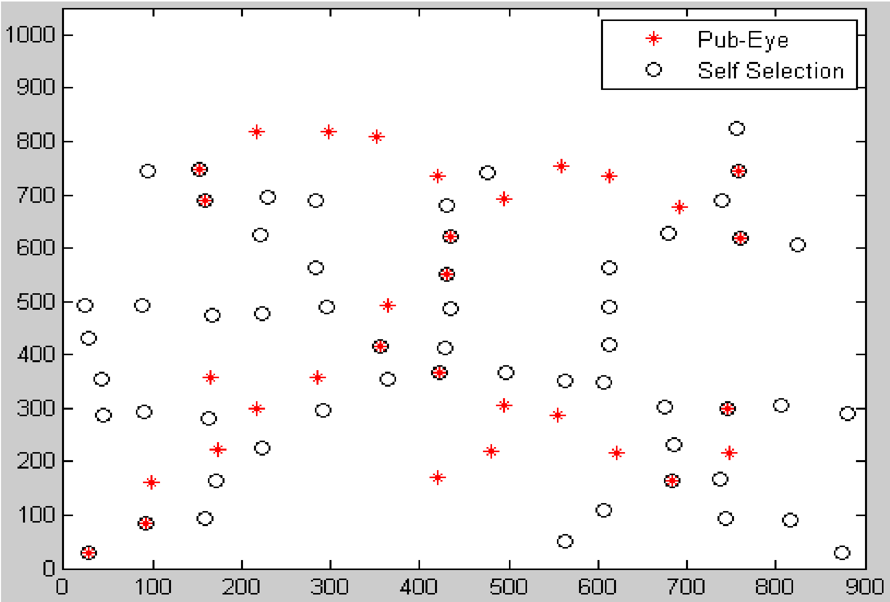
(b) 300 nodes with the *K* = 3



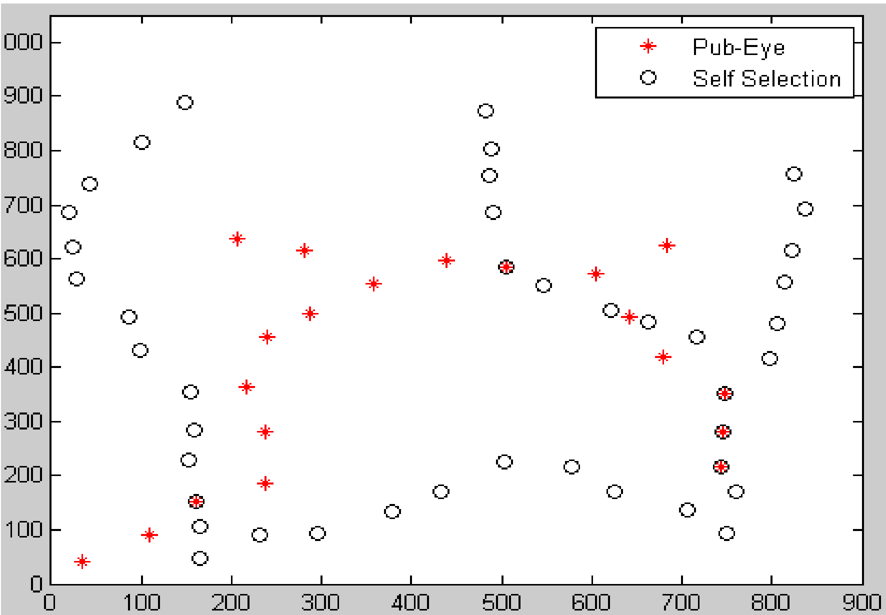
(c) 400 nodes with the K = 4

Figure 38 Broker selection among the whole broker set

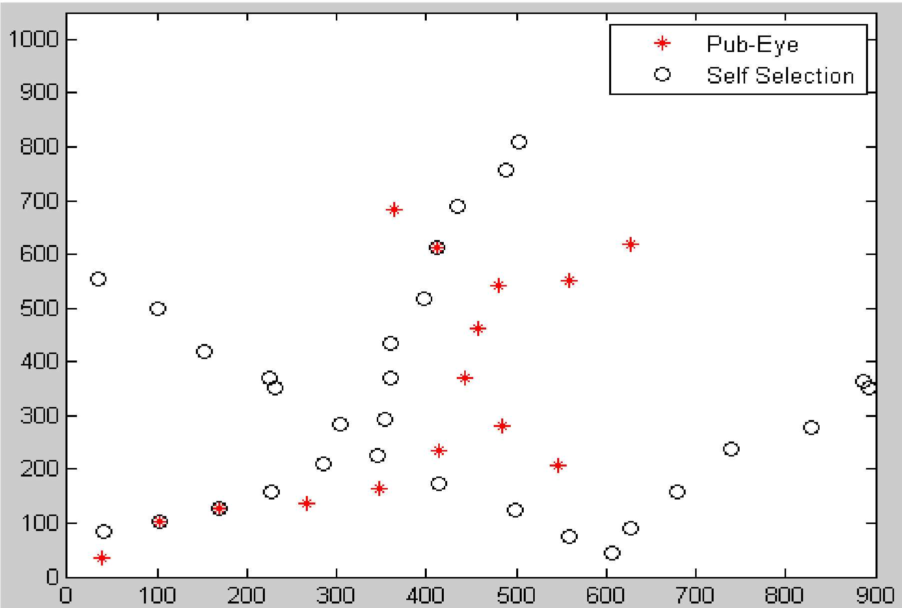
We then compare our distributed algorithm with another random self-selection based heuristic algorithm in Figure 39. In the self-selection algorithm, every broker uses a small probability ($1/K^2$) to self-select to be the forwarding node. After that, a connection algorithm (Steiner tree) is used to connect all the self-selected brokers to form forwarding path with local repairmen to ensure the coverage. We use the percentage out of the total brokers to measure the size of selected brokers.



(a) 200 nodes with the $K = 2$



(b) 300 nodes with the K = 3



(c) 400 nodes with the K = 4

Figure 39 Comparison with self-selection algorithm

The simulation result in Table 5 shows that our algorithm selects a very small portion of the brokers to forward the subscriptions. Our algorithm selects 50% fewer brokers than the self-selection algorithm in all cases. With the increase of the network size N, the benefit of our algorithm becomes more significant. Figure 40 gives examples to compare the selected brokers set between our algorithm and the self selection heuristic.

		Pub-Eye	Self Selection
N=200	K = 2	15.5%	32.5%
	K = 3	12%	23%
	K = 4	9%	15%
N=300	K = 2	11%	23%
	K = 3	7%	13%
	K = 4	4%	9%
N=400	K = 2	7.3%	20%
	K = 3	5.5%	10%
	K = 4	4%	8%

Table 5 Comparison of selected brokers between Pub-Eye and Self Selection

We also compared our algorithm with the existing *K*-CDS construction algorithms d-SPR proposed by Dhar [13] in a 400 nodes network (shown in Figure 40). The d-SPR algorithm is to select a connected d-dominating set to form backbones of the ad hoc wireless network. The d-SPR uses at least 2 rounds to find the set.

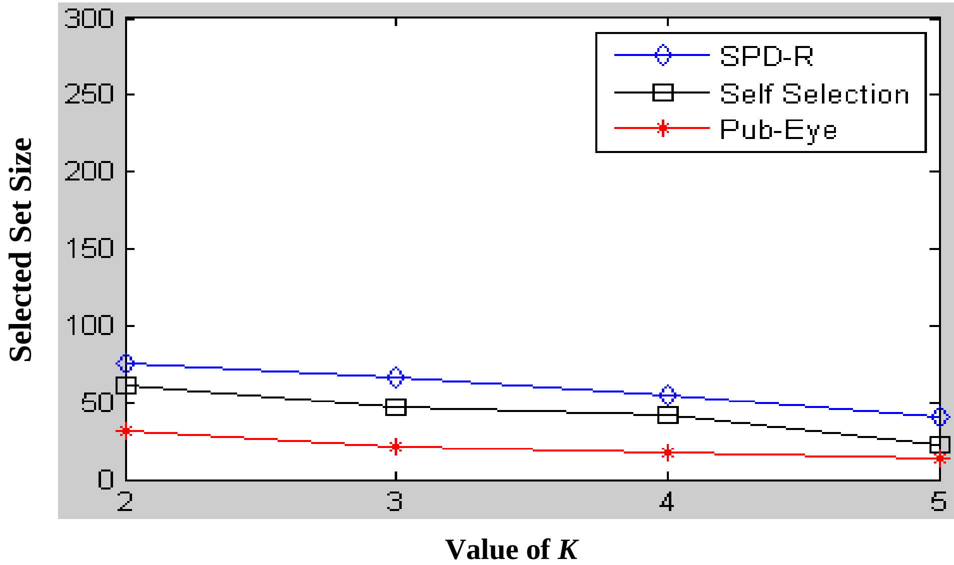
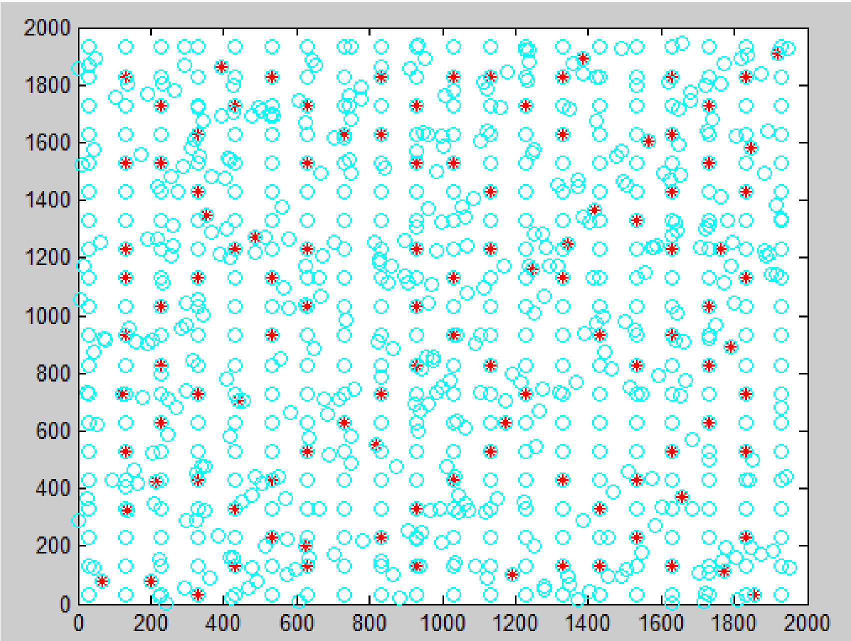


Figure 40 Comparison with d-SPR

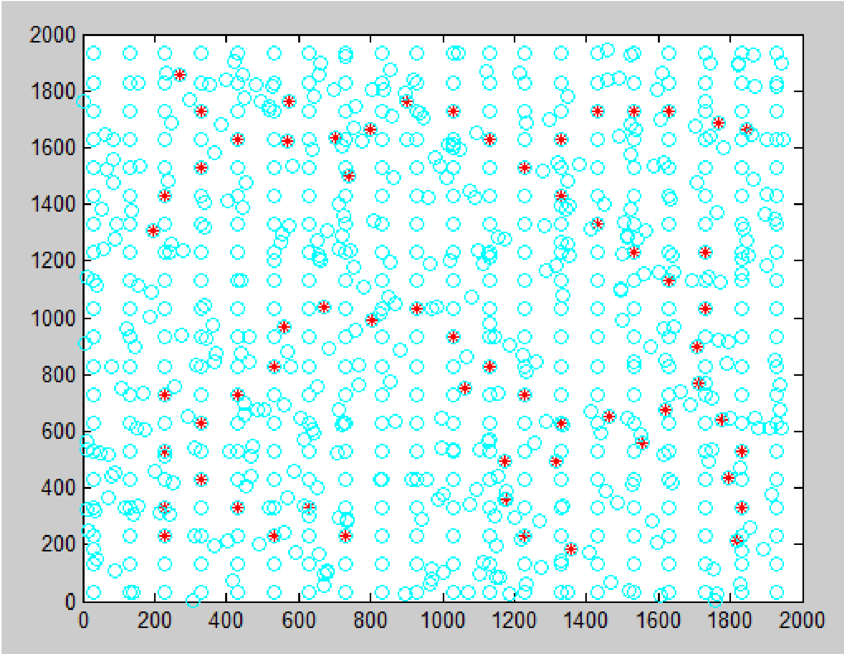
The result shows that our algorithm selected a smaller set of brokers than the d-SPR algorithm and self selection heuristic. Moreover, our algorithm runs only one round for node selection, which is less than the rounds required by d-SPR.

5.3.2 Dynamic K Algorithm

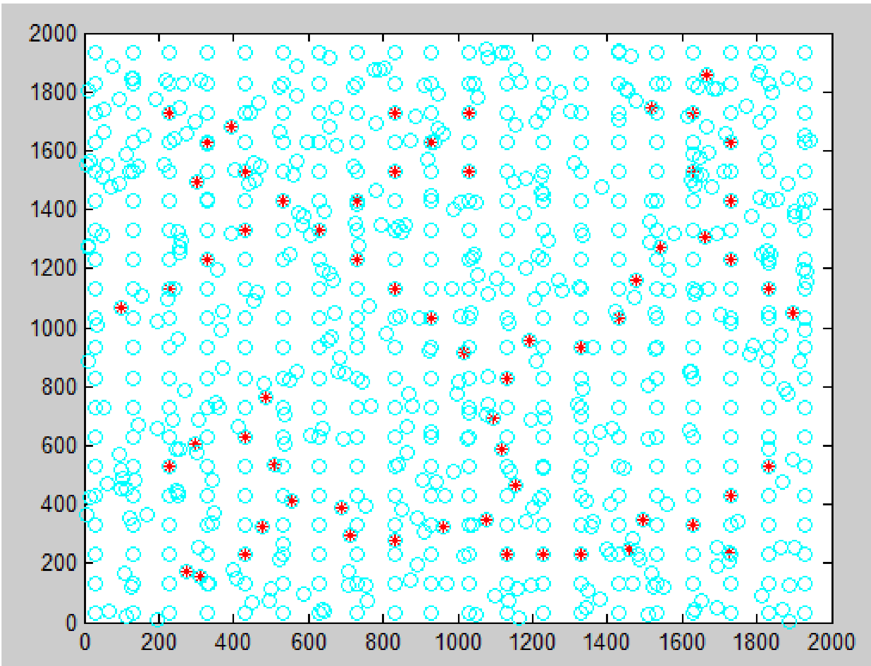
After comparing the performance of *Fix K* algorithm, we also conduct simulation for the dynamic algorithm. In *dynamic K*, the only difference is that calculation of *K* is based on the distance between the subscriber and the broker received the subscriptions. Similar to the field testing, we here use the one hop delay boundary and the hop distance between the subscriber and the broker to estimate the transmission delay. We compare the size of selected broker set using *dynamic K* with the set of using *Fix K* algorithm. To better compare the performance of *Fix-k* and *dynamic-k*, we re-configure the simulation settings to define a larger network size. We use 800 nodes uniformly deployed to form random disk graph in a 2000*2000 region. The transmission range between any pair is same to the previous simulation. We set the subscriber to locate at the center of the 2D region, and again we set the one hop delay boundary t to 25ms. As the setting of the network scale increases, we configure this network to have a relatively large delay boundary as 500ms. Then, we select different value of delay constraint δ to see the compare the selection efficiency between our two algorithms.



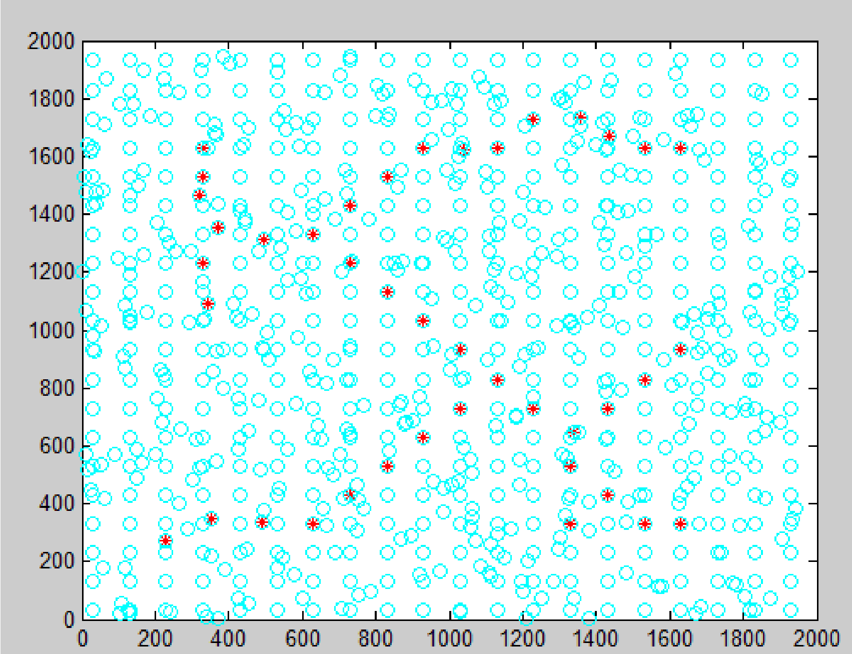
Fix-K with the δ = 550 ms



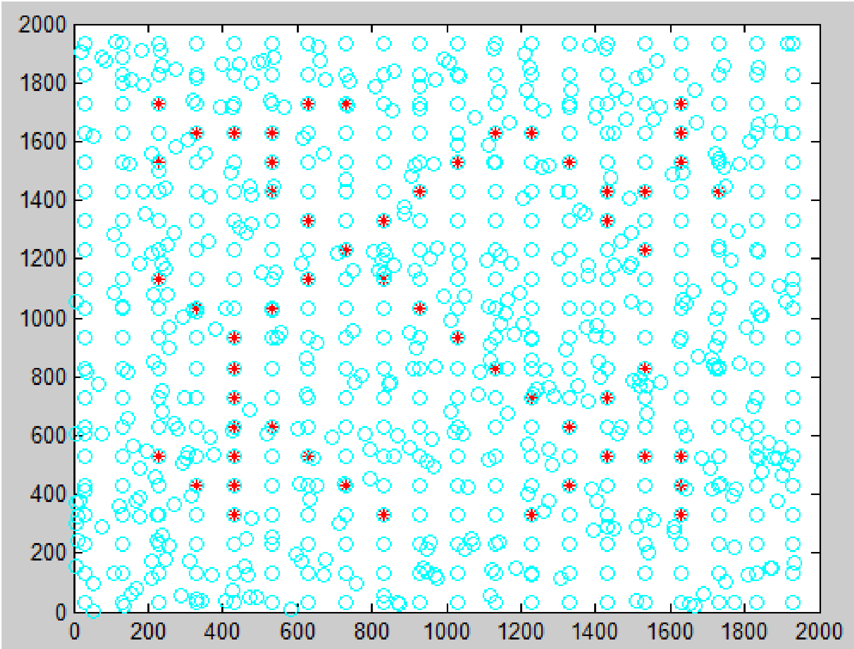
Dynamic-K with the δ = 550 ms



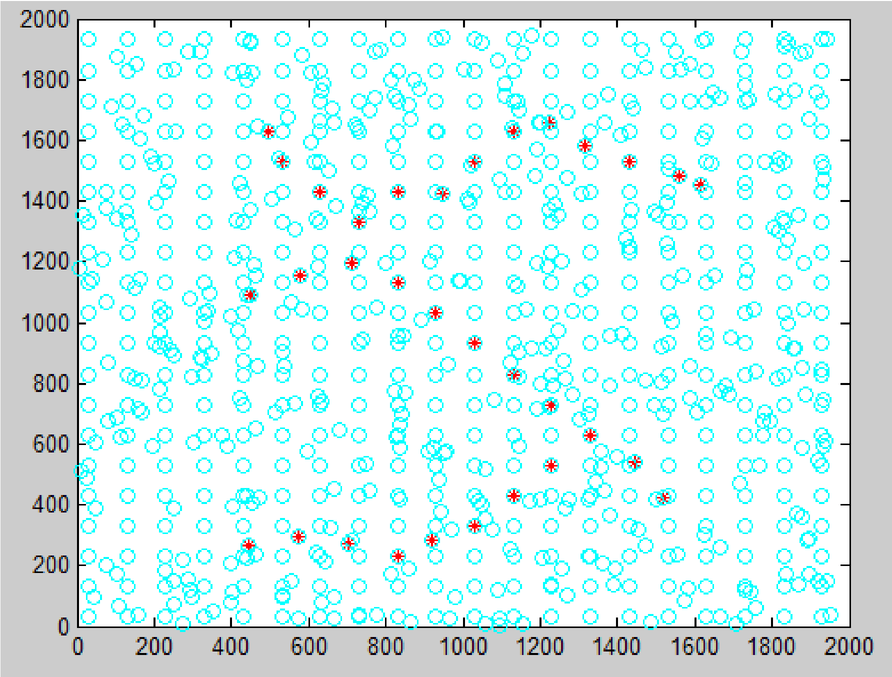
Fix-K with the δ = 600 ms



Dynamic-K with the δ = 600 ms



Fix-K with the δ = 700 ms



Dynamic-K with the δ = 700 ms

Figure 41 Comparison between *Fix-K algorithm* and *Dynamic-K*

		Fixed-K	Dynamic-K
δ =350	*D* = 200	4.5%	3.2%
	D = 250	7%	3.5%
	D = 300	9%	5.2%
δ =400	*D* = 200	3.5%	2.6%
	D = 250	4.5%	3.2%
	D = 300	7%	3.5%
δ =500	*D* = 200	3%	2.5%
	D = 250	3.5%	2.5%
	D = 300	7%	3.5%

Table 6 Selected Broker Set Size Comparison between *Fixed-K algorithm* and *Dynamic-K*

From Figures 41 and Table 6 shown above, the *Dynamic-K* out-performs the *Fixed K* in all cases. We also note that, the difference between these two algorithms is very significant when the value of network delay boundary *D* is close to the constraint δ.

As the *Fixed-K* always calculate the necessary value of *K* according to the network delay boundary, the calculated K will be backward-looking compared to the *Dynamic-K* in case the network delay boundary is large.

5.4 Testing

In order to test whether or not the selected *K*-dominating set can guarantee the delay bound, we conducted field testing to collect real experiment results.

5.4.1 Multi-hop Delay Estimation

To have a better view before introducing the field testing, we first give a brief analysis on the delay estimation in the multi-hop wireless mesh/ad-hoc networks. The wireless medium is shared and scarce, therefore ad hoc networks require an efficient MAC protocol. Since ad hoc networks lack infrastructure and centralized control, the MAC protocols for ad hoc networks should be distributed and can hardly be controlled. The delay of wireless ad hoc networks depend on the number of nodes, transmission range of the nodes, traffic pattern and the behavior of the MAC protocol.

We define the delay in our testing as the time taken by a packet to reach the destination after it has generated at the source. Meanwhile, according to the network model, the packet size scales with throughput. Under these assumptions the delay is simply proportional to the average number of hops between a source destination pair. In this field testing we will consider the extra delay caused by queuing. Besides, the packet size is assumed to be constant. The delay is defined as time taken by a packet to reach the destination after its arrival/generation at the source, there would be queuing delays at the source and intermediate nodes.

In [17], the authors use diffusion approximation method for solving open G/G/1 queuing networks. In [17] a random access MAC model is assumed and the average end-to-end delay in a multi-hop wireless network is given. The network consists of n + 1 nodes and each node is assumed to have an equal transmission range. The traffic model for the network may be described as each node in the network could be a source, destination and/or relay of packets. Each node generates packets with rate λ packets/sec. For the estimation of the queuing delay, [17] uses the queuing model based on the packet arrival rate λ and utilization factor ρ. In this model, the D_i can be calculated by the packet arrival rate and the utilization factor at node i.

$$D(n) = \bar{s} \times \frac{\rho}{\lambda_i(1 - \widehat{\rho)}}$$

The author also proves that For the random access MAC model the average end-to-end delay in a multihop wireless network, denoted by *D(n).* There exists

$$D(n) = \bar{s} \times \overline{D\iota}$$

Here *s* denotes the expected number of hops between source and destination and D_i is the average delay on node *i*. In our field testing, we will configure the queuing delay on each node by using this model.

5.4.2 Field Testing Setup and Configuration

We set up the testing field as shown in the following Figure 42. Ten laptops were included in the field testing to test the delay of the Pub/Sub. Three entities were included in the testing

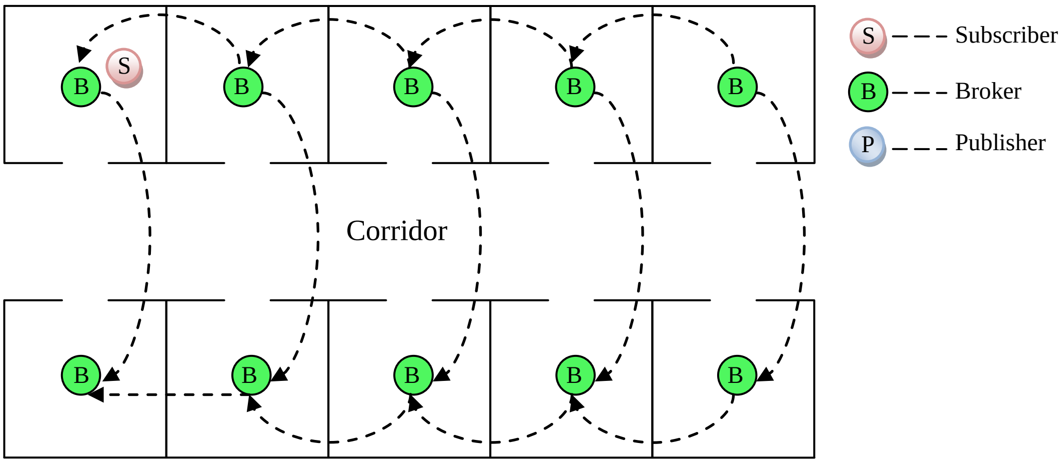


Figure 42 Field testing deployment

"S" represents the subscriber, who initiates the subscription forwarding through the broker overlay.

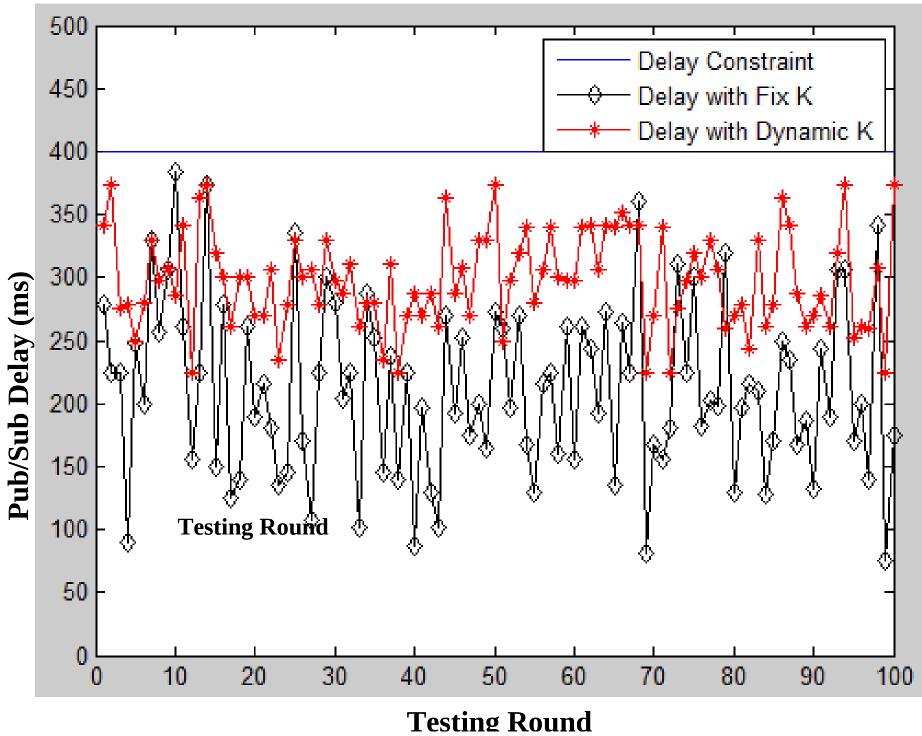
"B" stands for the broker and the topology of the broker overlay is set up with the maximum hop distance of 5 hops.

"P" is the publisher, which publishes the event and forwards the notification to the broker overlay for subscription matching.

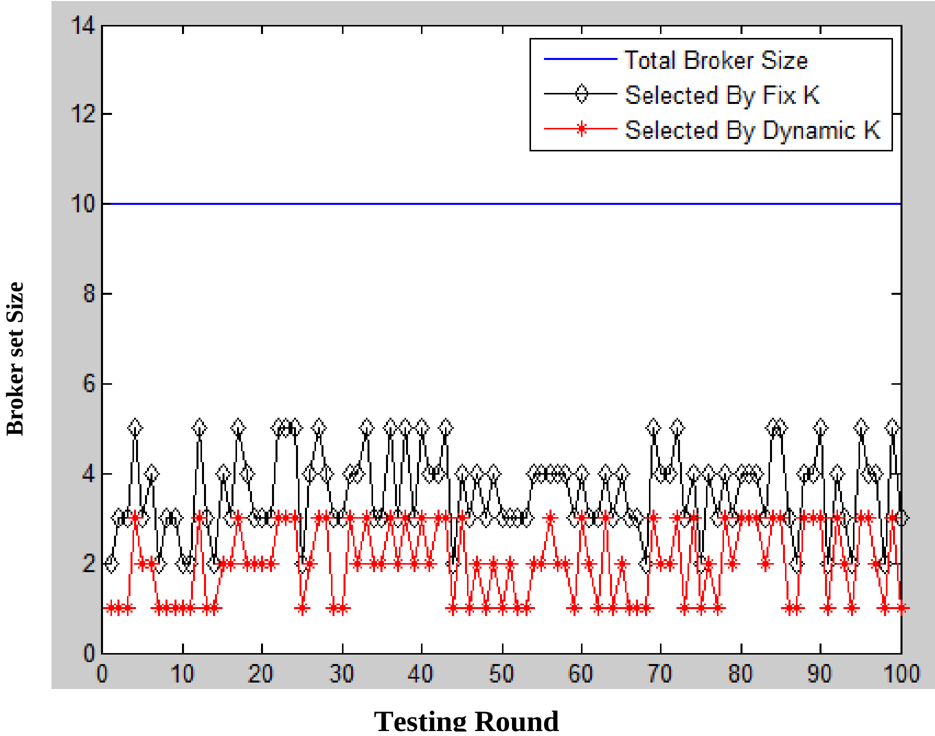
For each round of testing, one pair of subscriber and publisher were included. The location of the subscriber and publisher were randomly selected. We conducted testing for several rounds. Every round, a random position of subscriber was set. After that, we used our algorithm to select broker set to form the content filtering structure. We then randomly picked up one broker to initiate an event publishing process. We measured the time from the point that the event notification generated to the point that the subscriber received the event notification. We also set a random queuing delay in every relaying broker node.

5.4.3 Field Testing Results

We collected 100 rounds of testing results and the following figures showed the performance of the proposed algorithm. According to the results, all the collected delay time is less than the predefined delay boundary 400ms. We also look at the selected broker set size to see the relationship of the size of selected broker and the Pub/Sub delay. As shown in Figure 43, the *Fix K* often have a larger size of the selected broker set to maintains the subscription, which helps to achieve a smaller total delay.



(a) The Pub/Sub delay



(b) The broker set size

Figure 43 Field testing results

5.5 Conclusions

In order to apply the Pub/Sub paradigm to real-time video surveillance application scenario, we have investigated the problem of delay constrained content based Pub/Sub and proposed algorithms for disseminating the subscriptions to form the scalable distributed content filtering structure. The proposed system is the first work that jointly considers the delay constraint and scalability issues in distributed content based Pub/Sub system. We have implemented a prototype system and already integrated proposed features into the system.

In the future, we will further explore the real-time composite events publish/subscribe.

CHAPTER 6 CONCLUSIONS AND FUTURE WORKS

In this chapter, we summarize our works and outline the directions for future research. In this thesis, a complete design of Publish/Subscribe (Pub/Sub) based video surveillance system on top of interconnected WMNs and WSNs is proposed. The proposed system is scalable and easy to deploy. It also provides user a new way to interact with video surveillance cameras. We then study the problem of delay constrained Pub/Sub to support real-time event subscription. We propose a customized Pub/Sub middleware to support video surveillance for real-time events. The proposed Pub/Sub middleware can achieve both efficient use of wireless network resource and efficient detection of events.

6.1 Conclusions

In Chapter 3, we propose a novel Pub/Sub based frame work to provide efficient video surveillance service over wireless networks. In the framework, we utilize the decoupling characteristics of Pub/Sub in time, space and synchronization to assist the in-network processing in the asynchronous wireless environment. The video streaming services are driven by users' subscriptions. This feature greatly improves the efficiency of video transmission.

In order to apply the Pub/Sub paradigm to the proposed system, we investigate the issues of interconnecting WMNs with WSNs and then propose a practical solution in Chapter 4. The proposed surveillance system considers the content translation between WMNs and WSNs to enable the Pub/Sub service between these two types of networks. We have already implemented a prototype system on our intelligent transportation system platform for traffic monitoring. In the last part of Chapter 4, we introduce a preliminary solution as the security extension of the system. We use TLS protocol to provide transport layer secure tunnel for video transmission in the vulnerable wireless environment. The TLS session establishment has already been integrated into the prototype system.

In Chapter 5, we consider the real-time characteristic of the video surveillance application scenario. In order to apply the loosely coupled Pub/Sub paradigm to the real-time video surveillance, we have investigated the problem of delay constrained content based Pub/Sub and propose the algorithm for disseminating the subscriptions to form the scalable distributed content filtering structure. The proposed system is the first Pub/Sub system that jointly considers the delay constraint and scalability issues in distributed content based Pub/Sub system. We have implemented a real-world system and already put some proposed features into the system.

6.2 Future Works

In Chapter 4, our secure Pub/Sub work is still on-going. We will try to adapt the attribute based encryption scheme to provide the secure event subscription in the Pub/Sub system.

In Chapter 5, we focus on the subscription of the atomic event. However, in the real environment, the event can last for a certain period of time. We will further look into the design of Pub/Sub system for real-time continuous events. Moreover, in current design, we use the maximum transmission time to calculate the parameter K. The fix value of K is not optimal for the subscription forwarding. We are working on the algorithm using dynamic value of K to help select the brokers for subscription forwarding and maintenance.

REFERENCES

[1] D.Agrafiotis, T-K Chiew, "Seamless Wireless Networking for Video Surveillance Applications"

[2] K. Akkaya, M. Younis, "A survey of routing protocols in wireless sensor networks, Ad Hoc Network" (Elsevier) 3 (3) (2005) 325–349.

[3] K. Akkaya, M. Younis, An energy-aware QoS routing protocol for wireless sensor networks, in: Proc. of Intl. Conf. on Distributed Computing Systems Workshops (ICSDSW), Washington, DC, 2003

[4] K. M. Alzoubi, P.-J.Wan, and O. Frieder. "Distributed heuristics for connected dominating sets in wireless ad hoc networks". *Journal of Communications and Networks*, 4(1):22–29, Mar. 2002.

[5] S. Bouckaert , E. Poorter, B. Latré, J. Hoebeke, I. Moerman and P. Demeester "Strategies and Challenges for Interconnecting Wireless Mesh and wireless Sensor Networks", *Journal of Wireless Personal Communications*, Volume 53, Number 3,May, 2010

[6] J. Cao, K. Xie, W. Wu, et al., "HAWK: Real-world Implementation of High-performance Heterogeneous Wireless Network for Internet Access", *1st ICDCS International Workshop on Next Generation Network Architectures (NGNA'09),* June 22-26, 2009 Montreal, Canada

[7] A.Carzaniga, D.Rosenblum, and A.Wolf, "Design and evaluation of a wide-area event notification service". *ACM Trans. Computing. System. 19,* 3 (Aug.), 332–383, 2001.

[8] M.Castro, P.Druschel, A.-M.Kermarrec, and A.Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure". *IEEE J. Sel. Areas Commun. 20*, 8 (Oct.), 1489–1499. 2002.

[9] B. Chandramouli, J. Yang, P. K. Agarwal, A. Yu, and Y. Zheng. "ProSem: Scalable wide-area publish/subscribe". In *SIGMOD*, 2008.

[10] RT Collins, AJ Lipton, H Fujiyoshi and T Kanade, "Algorithms for cooperative multisensor surveillance", Proceedings of the IEEE, 2001

[11] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio,multi-hop wireless mesh networks," Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)

[12] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, June 2003

[13] W. Feng, J. Walpole, W. Feng, and C. Pu, "Moving towards massively scalable video-based sensor networks," Proceedings of the Workshop on New Visions for Large-Scale Networks: Research and Applications

[14] W. Feng, "Dissecting the video sensing landscape", Proc. of the ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV) 2005.

[15] E. Felemban, C.-G. Lee, E. Ekici, "MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks", IEEE Trans. Mobile Computing5 (6) (2006) 738–754

[16] A Hampapur, L Brown, J Connell, A Ekin, N Haas, M, “Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking”, Signal Processing Magazine, IEEE, 2005

[17] A Hampapur, LM Brown, J Connell, M Lu, H Merkl, “Multi-scale tracking for smart video surveillance “, IEEE Transactions on Signal Processing, 2005

[18] Y. Huang, H.Garcia-Molina. “Publish/subscribe in a mobile environment”. *In Proceedings of MobiDE* . 27–34, 2001.

[19] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: real-time surveillance of people and their activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp 809–830, 2000.

[20] T. He, J.A. Stankovic, C. Lu, T.F. Abdelzaher, “A spatiotemporal communication protocol for wireless sensor” networks, IEEE Trans. Parallel Distr. Syst. 16 (10) (2005) 995–1006.

[21] Zhihai He; Dapeng Wu, “Performance Analysis of Wireless Video Sensors in Video Surveillance”, Global Telecommunications Conference 2005. GLOBECOM '05

[22] F. Hu, S. Kumar, “Multimedia query with QoS considerations for wireless sensor networks in telemedicine”, Proc. of Society of Photo-Optical Instrumentation Engineers Intl. Conf. on Internet Multimedia Management Systems 2003.

[23] J. Koshy, R. Pandey, “Synthesizing scalable runtime environments for sensor networks”, Proceeding of the ACM Conf. on Embedded Networked Sensor Systems (SenSys), San Diego, CA, November 2005.

[24] T Kanade, R Collins, A Lipton, P Burt, L Wixson "Advances in cooperative multi-sensor video surveillance", Proc. DARPA Image Understanding Workshop, 1998

[25] P. Kulkarni, D. Ganesan, P. Shenoy, Q. Lu, "SensEye: a multi-tier camera sensor network", Proceedings of ACM Multimedia, Singapore, November 2005.

[26] S. Lai, J. Cao, and Y. Zheng. Psware: A publish / subscribe middleware supporting composite event in wireless sensor network. In *Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–6, March 2009.

[27] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV '98),* pp. 8–14, Princeton, NJ, USA, October 1998

[28] F Licandro, G Schembra, "Wireless mesh networks to support video surveillance: architecture, protocol, and implementation", *EURASIP Journal on Wireless Communications and Networking*, 2007

[29] B. Liang and Z.J. Haas. "Virtual Backbone Generation and Maintenance in Ad Hoc Network Mobility Management". *Infocom,* 2000

[30] F Licandro, G Schembra, "Wireless mesh networks to support video surveillance: architecture, protocol, and implementation", EURASIP Journal on Wireless Communications and Networking, 2007

[31] F Licandro, Alfio Lombardo and Giovanni Schembra "Applying Multipath Routing to a Video Surveillance System Deployed over a Wireless Mesh Network", WMuNeP'2006

[32] P. Mahonen, "Wireless video surveillance: system concepts", International Conference on Image Analysis and Processing, 1999.

[33] http://wmn.comp.polyu.edu.hk/hawk/doku.php?id=demonstrations

[34] http://www.ietf.org/rfc/rfc3626.txt

[35] W. Rao , L. Chen , A. Fu, "On Efficient Content Matching in Distributed Pub/Sub Systems", *Infocom* 2009

[36] M. Rahimi, S. Ahmadian, D. Zats, J. Garcia, M. Srivastava, D. Estrin, "Deep vision: Experiments in exploiting vision in wireless sensor networks", submitted for publication.

[37] T.Raty, L. Lehikoinen, "Scalable Video Transmission for a Surveillance System", Fifth International Conference on Information Technology: New Generations, ITNG 2008

[38] A Senior, S Pankanti and A Hampapur, L Brown, YL Tian; "Enabling Video Privacy through Computer Vision" SECURITY & PRIVACY, 2005

[39] S. Soro, W.B. Heinzelman, "On the coverage problem in video-based wireless sensor networks", Proceeding of the IEEE Intl. Conf. on Broadband Communications, Networks, and Systems (BroadNets), Boston, MA, USA, October 2005

[40] S. Sutor, F. Matusek and R. Reda, “WSSU: Hign Performance Wireless Self-Contained, Surveillance Unit: an Ad Hoc Video Surveillance System”, Telecommunications, 2008. AICT '08

[41] Ying-li Tian, Lisa Brown, Arun Hampapur, Max Lu, Andrew Senior and Chiao-fe Shu, “IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework”

[42] J. Wang, Chen Qimei, Zhang De and Bi Houjie, “Embedded Wireless Video Surveillance System for Vehicle” ITS Telecommunications Proceedings, 2006

[43] L. Savidge, H. Lee, H. Aghajan, A. Goldsmith, “QoS-based geographic routing for event-driven image sensor networks,”, Proc. of IEEE/CreateNet Intl. Workshop on Broadband Advanced Sensor Networks (BaseNets), Boston, MA, October 2005.

[44] PeopleVision Project, IBM Research,http://www.research.ibm.com/peoplevision/

[45] T.W. Yan and H. Garcia-Molina, “The SIFT information dissemination system”, *ACM Transactions on Database Systems* 24(4) (1999) 529–565

[46] S. Dhar, M. Q. Rieck, S. Pai and E. J. Kim. “Distributed Routing Schemes for Ad Hoc Networks Using d-SPR Sets”, *Microproces-sors and Microsystems, Special Issue on Resource Management in Wireless and Ad Hoc Mobile Networks*, Volume 28, Issue 8, October 2004, Pages 427-437

[47] Avancha, S., Undercoffer, J., Joshi, A., & Pinkston, J. ,“Security for wireless sensor networks”. In Wireless sensor networks, pp. 253–275. Norwell, MA, USA: Kluwer Academic Publishers, 2004.

[48] Srivatsa, Mudhakar, and Ling Liu. "Secure event dissemination in publish-subscribe networks." Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on. IEEE, 2007

[49] Raiciu, Costin, and David S. Rosenblum. "Enabling confidentiality in content-based publish/subscribe infrastructures." Securecomm and Workshops, 2006. IEEE, 2006.

[50] Ion, Mihaela, Giovanni Russello, and Bruno Crispo. "Supporting publication and subscription confidentiality in Pub/Sub networks." Security and Privacy in Communication Networks. Springer Berlin Heidelberg, 2010. 272-289.

[51] C. Wang, A. Carzaniga, D. Evans, and A. L. Wolf. "Security issues and requirements for internet-scale publish-subscribe systems". In Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35), 2002

[52] R. Ostrovsky, A. Sahai, and B. Waters. "Attribute-based encryption with nonmonotonic access structures". In Proceedings of the 14th ACM conference on Computer and communications security, page 203. ACM, 2007

[53] C. Dong, G. Russello, and N. Dulay. Shared and searchable encrypted data for untrusted servers. In Proceedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security, pages 127-143, Berlin, Heidelberg, 2008. Springer-Verlag

[54] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In IEEE Symposium on Security and Privacy, pages 321-334. Citeseer, 2007

[55] Chen, M., González, S., Cao, H., Zhang, Y., & Vuong, S. T. (2013). Enabling low bit-rate and reliable video surveillance over practical wireless sensor network. The Journal of Supercomputing, 65(1), page287-300, 2013

[56] Yildiz, Enes, et al. "Optimal Camera Placement for Providing Angular Coverage in Wireless Video Sensor Networks." , 2013.

[57] Jafarpour H, Hore B, Mehrotra S, et al. CCD: A Distributed Publish/Subscribe Framework for Rich Content Formats[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(5): 844-852.