國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

實現及實測在無線網狀網路中廣播演算法

Realizing and Benchmarking Broadcast Algorithms in Wireless Mesh Networks

研 究 生：陶錫泓

指導教授：林盈達 教授

中 華 民 國 九 十 七 年 七 月

實現及實測在無線網狀網路中廣播演算法

Realizing and Benchmarking Broadcast Algorithms in Wireless Mesh Networks

研 究 生：陶錫泓　　　　Student: Shi-Hung Tao

指導教授：林盈達　　　　Advisor: Dr. Ying-Dar Lin

國立交通大學

網路工程研究所

碩士論文

A Thesis
Submitted to Institutes of Network Engineering
College of Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
In

Computer Science and Engineering

June 2008
Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

國立交通大學
研究所碩士班

論文口試委員會審定書

本校 網路工程 研究所 陶錫泓 君

所提論文:

實現及實測在無線網狀網路中廣播演算法

Realizing and Benchmarking Broadcast Algorithms in Wireless Mesh Networks

合於碩士資格水準、業經本委員會評審認可。

口試委員：

指導教授：

所　　長：

中 華 民 國 九十七 年 七 月 一 日

Institute of Network and Engineering
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the thesis prepared by Shi Hung Tao. entitled Realizing and Benchmarking Broadcast Algorithms in Wireless Mesh Networks

and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Master of Science.

Committee Members:

Thesis Advisor:

Director:

Date: 2008-07-01

國　立　交　通　大　學

博碩士論文全文電子檔著作權授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學網路工程研究所， 九十六學年度第二學期取得碩士學位之論文。

論文題目：實現及實測在無線網狀網路中廣播演算法

指導教授：林盈達

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：	
本校及台灣聯合大學系統區域網路	■ 立即公開
校外網際網路	■ 立即公開

■ 全文電子檔送交國家圖書館

授 權 人：　陶　錫　泓

親筆簽名： 陶錫泓

中華民國　　九　十　七　年　九　月　一　日

國　立　交　通　大　學

博碩士紙本論文著作權授權書

（提供授權人裝訂於全文電子檔授權書之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學網路工程研究所， 九十六學年度第二學期取得碩士學位之論文。

論文題目：實現及實測在無線網狀網路中廣播演算法

指導教授：林盈達

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：____________________，請將論文延至____年____月____日再公開。

授 權 人： 陶 錫 泓

親筆簽名： 陶錫泓

中華民國　　九　十　七　年　九　月　一　日

國家圖書館
博碩士論文電子檔案上網授權書

ID:GT009556541

本授權書所授權之學位論文，為本人於國立交通大學網路工程研究所， 九十六學年度第二學期取得碩士學位之論文。

論文題目：實現及實測在無線網狀網路中廣播演算法

指導教授：林盈達

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

■ 論文全文上載網路公開之範圍及時間：2008.8.29 公開

※ 讀者基於非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人： 陶 錫 泓

親筆簽名： 陶錫泓

民國 九 十 七 年 九 月 一 日

實現及實測在無線網狀網路中廣播演算法

研究生：陶錫泓　　　　　　　　　　指導教授：林盈達

國立交通大學網路工程研究所

摘　　要

氾濫式廣播機制已經被證實在多節點跳躍無線網路上造成嚴重的廣播風暴問題，而此問題在無線網狀網路中變得更為嚴重。由於無線網狀網路介接有線區域網路，使得區域網路的廣播封包流入無線網路，造成更嚴重的碰撞機率。此外，無線網狀網路使用連結層廣播傳遞許多網路控制、路由、拓樸維持協定。因此，其對於廣播可靠性的要求也更高。已經有許多廣播演算法以提高有效性及可靠性而被提出，但鮮少在真實系統上被驗證。在本文中，我們研究五種具有代表性的廣播演算法，包括基於機率、基於延遲、使用鄰居資訊等方法。我們討論這些演算法在實作上的問題，並在真實系統上實現它們。此外，針對不同的拓樸及封包大小，透過實驗比較各演算法的可靠度、轉送機率與效益。不同於以往模擬結果中氾濫式廣播機制較為可靠，我們的研究結果顯示由於碰撞機率較輕微，Self-Pruning演算法可提供最可靠的廣播。另外，當同時考量可靠度與轉送機率時，Domain-Pruning演算法提供最佳的廣播效益。最後，以機率為基礎的演算法則因為過低的可靠性，不論在何種情況下皆不建議被使用。

關鍵字： 廣播風暴、廣播演算法、無線網狀網路

Realizing and Benchmarking Broadcast Algorithms in Wireless Mesh Networks

Student: Shi-Hung Tao **Advisor: Dr. Ying-Dar Lin**

Institute of Network Engineering
National Chiao-Tung University

Abstract

Broadcasting by flooding has been proved to cause broadcast storm problems in the multi-hops wireless networks, and it becomes more serious in a wireless mesh network (WMN). Due to bridging the wired LAN in WMN, the amount of broadcast traffic increases and the collision probability raises. In addition, the broadcast reliability is more important in WMN where the protocols of network-controlling, routing and topology maintaining are directly designed with layer-2 broadcast. Many algorithms have been developed for efficient and reliable broadcasting, though those approaches are seldom verified in the real world. In this work, we study five representative broadcast algorithms including the probability-based algorithms, delay-based algorithms and the algorithms using neighbor information. We discuss the common and algorithm-specific implementation issues, and implement them on the real-world testbed. The reliability, forwarding ratios and efficiencies are compared through experiments under different topologies and packet lengths. Different from the simulation results, in which the flooding approach performs better than others, our study shows that the self-pruning algorithm resulting in the best reliability due to its lighter collision probability. In addition, the domain-pruning algorithm always performs the best efficiency over others when taking both the reliability and forwarding ratios into consideration. Finally, the probability-based algorithms are not suggested in any situations due to its worst reliability.

Keywords: Broadcast storm, Broadcast algorithm, WMNs

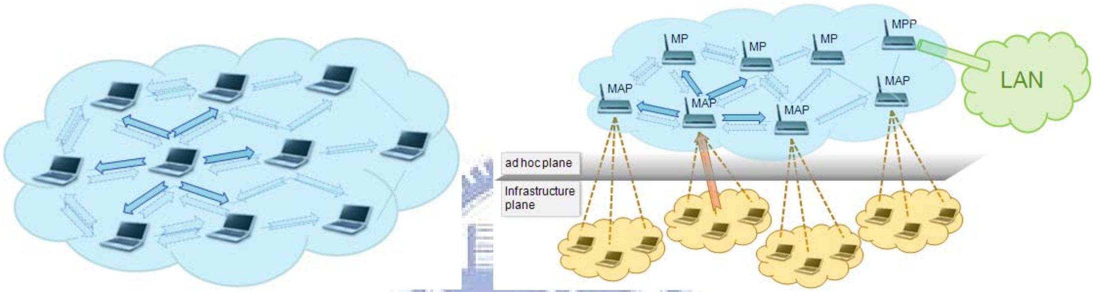
Contents

List of Figures

List of Tables

Chapter 1 Introduction

In wireless ad hoc networks, the straightforward broadcast mechanism transmits broadcasting messages to all its neighbors. This simple flooding [1] has been proved to cause two serious issues well-known as the broadcast storm problem [2]. First, it generates too many unnecessary transmissions due to the overlapped coverage by several nodes, as shown in Figure 1(a). Second, it raises higher collision probability since many nodes forward broadcast messages in a short period. Consequently, the reliability of broadcasting would degrade [3] in a mobile ad hoc network.



(a) MANET (b) WMN

Figure 1 Broadcast problems in the wireless ad hoc network and WMN

The broadcast storm problem becomes more serious in wireless mesh networks. As shown in Figure 1(b), a wireless mesh network (WMN) is a multi-hop wireless structure composed of both wireless ad hoc networks and infrastructure-based wireless networks. The roles in the ad hoc plane contain Mesh Point (MP), Mesh Access Point (MAP), and Mesh Portal Point (MPP). The end stations which are associated to MAPs build the infrastructure plane. The hybrid network structure would cause higher collision probability especially when both the ad hoc plane and infrastructure plane in Figure 1(b) share the *same* channel. In addition, the existing broadcast-based network-controlling protocols such as ARP [4] and STP [5] flood broadcasting messages from wired LAN bridged by the ad hoc plane into WMN, and thus increase the amount of broadcasting messages. Furthermore, the broadcast mechanism is more important in WMN where routing and topology maintenance are directly designed with layer-2 broadcast messages [6]. As a result, the broadcast storm problem in WMN becomes more frequent and its impact is more serious.

Many solutions have been developed for efficient and reliable broadcasting in wireless ad hoc networks. We classify most solutions into 8 categories in Table 1 and brief their ideas and advantages as follows. First, the naïve idea is to retransmit the broadcasting messages to all its neighbors either at all times [1] or under some predefined probability [2]. The advantage is the simplicity to implement, though it results in the broadcast storm problem. The second idea is to *delay* the decision of retransmission for a short time. A node could decide whether to retransmit a broadcasting message according to the phenomena observed in the short delay. The phenomena include the density of neighbors [7], the amount of received duplication [2,8], and the signal strength [9,10]. The advantage is that each node works independently without information exchange between neighbors. The last idea is to utilize the *neighbor* information within 2-hop. With the help of neighbor information, a node could prune unnecessary broadcasting messages by itself [11] or reduce the redundancy in a partial network topology to perform more actively for efficient and reliable broadcasting [3,13-20].

Table 1 Categories of broadcasting algorithms

Idea	**Design Unit**	**Method**	**Papers**
Naïve	Single	Flooding	Ho et al, 1999 [1]
		Probabilistic	Ni et al, 1999 [2]
Delay	Single	Dynamic Probabilistic	Zhang and Agrawal, 2002 [7]
		Counting-based	Ni et al, 1999 [2], Mohammed et al, 2005 [8]
		Distance Sensing	Li et al, 2006 [9], Li et al, 2007 [10]
Neighbor Information	Single	Self Pruning	Peng and Lu, 2000 [11]
	Cluster	Self Pruning / Domain Pruning	Lim and Kim, 2000 [12], Lim and Kim, 2001 [13], Lou and Wu, 2002 [14], Shen et al, 2007 [15],
		Neighbor Union / Gateway Selection	Wu and Li, 1999 [16], Qayyum et al, 2000 [17], Francois and David, 2006 [18], Keshavarz et al, 2007 [19], Hasegawa et al, 2007 [20], Lou and Wu, 2007 [3]

To compare various algorithms, Williams and Camp [21] evaluated 5 algorithms with simulation and pointed out that the algorithms using neighbor information are preferred over other algorithms. Talmai et al. also simulated 6 categories of algorithms [22], and showed that, in a failure-prone scenario, incorrect update of neighbor information would mislead neighbor information-based algorithms. Also, they concluded the redundancy of broadcasting messages is necessary for ensuring the reliability.

All the above broadcasting algorithms originally designed for wireless ad hoc networks need to be reconsidered in WMN. The wireless ad hoc network can be taken as a subset of WMN in terms of topology creation and routing construction, but the ad hoc plane of a WMN usually has *managed* topologies [23] and sufficient power. In other words, the nodes inside the WMN have *less* neighbors, *lower* mobility, *shorter* path [24] and more *critical* broadcast messages [6]. Thus, WMNs need a protocol with higher delivery ratio but could ignore the problems of power consumption and the efficiency in the high mobility scenario.

In this work, we study five representative broadcast algorithms: Dynamic Probabilistic Broadcast [7], Efficient Counting Broadcast [8], Scalable Broadcast Algorithm [11], Domain Pruning Algorithm [14], and Wu and Li Algorithm [16]. The first two algorithms have minimum cost of resources. The remainders are the representatives of each category using neighbor information. They are selected because of the outstanding performance in simulations [22]. Also, we discuss their properties and behaviors in WMN by taking into account both the efficiency and reliability, and implement and benchmark their performance on the real world testbeds. The reason we conduct the experiments on real world platforms instead of simulations is that the simulation results might be much different from the implementation results, especially in the multi-hop wireless networks. With insufficient *details* of transmission mechanism [25], link stability [26] and transmission reliability [26-28], the wireless simulations result in quite different throughput [27,28] and delay [28]. In the multi-hop environment, differences would be *aggregated* over multiple wireless links. Moreover, experiments on implementations could reveal resource consumption and computation complexity which are also vital in realizing these algorithms.

The organization of this work is as follows. Chapter 2 briefs the studied algorithms. Chapter 3 describes the implementation model and lists the solutions to various implementation issues. The experiment results and their lessons learned are presented in chapter 4. Finally, chapter 5 concludes the work and points out future works.

Chapter 2 The Five Selected Broadcast Algorithms

Here we overview the chosen algorithms implemented in this work. Then the selecting criteria and reasons are explained.

Selected algorithms

Simple Flooding: This is the straightforward solution to support broadcasting in wireless environments. It starts with a source node broadcasting a packet to all neighbors. Each neighbor in turn forwards the packet one time and the process continues until all reachable nodes having retransmitted the broadcasting packet. Due to its simplicity, the method is used by many broadcasting implementations as the last resort. However, it has been shown in [2] that the flooding approach leads to the broadcast storm problem including high contention, collision and redundancy, and results in low coverage and long latency.

Dynamic Probabilistic Algorithm: Zhang and Agrawal proposed the dynamic probabilistic algorithm [7] combining the probabilistic scheme and the density of neighbor nodes. The original probabilistic scheme is similar to the simple flooding, except that a node only forwards a message with a fixed probability P. Obviously, the scheme saves network resources in dense networks, but it performs worse in sparse networks. The dynamic probabilistic algorithm adjusts the value of P by considering both the density of neighbor nodes and the number of heard rebroadcasts. The retransmissions probability P is lowered whenever a node is placed in a dense area, while it is raised for a sparse area.

Efficient Counting Broadcast (ECB): Aminu et al [8] proposed the ECB algorithm that also combines the probabilistic scheme and a counter counting the number of heard rebroadcasts for each received broadcast. Instead of taking the network density into consideration in the dynamic probabilistic algorithm, ECB uses a fixed probability *P*. The probability takes effect only when the counter is under a predefined threshold. Otherwise, the forwarding is cancelled due to too many heard retransmissions of the same broadcast issued by its neighbors.

Scalable Broadcast Algorithm (SBA): The main idea of the deterministic self-pruning scalable broadcasting algorithm proposed by Peng and Lu [11] is that a node does not need to forward the packet already received by neighboring nodes. It requires that all nodes have the information of their neighbors within a 2-hop radius. This algorithm works in two steps: neighbor information discovery and packet forwarding. First, neighbor information discovery is achieved via broadcasting a HELLO announcement in which the list of one-hop neighbors is embedded. After collecting the HELLO messages from all its neighbors, the 2-hop topology information is built. Second, whenever a node *r* receives a broadcast *m* from its neighbor node *t*, by looking for its own neighborhood set, node *r* can determine whether to schedule a retransmission. The retransmission first delays a random backoff to avoid collision with its neighbors. The backoff scheme is based on the density of neighbors; thus, nodes with the most neighbors usually broadcast before the others.

Domain Pruning (DP): The domain pruning algorithm also uses 2-hop neighbor information for routing decisions [13]. Unlike SBA, however, DP requires a broadcasting node *r* proactively choosing a smallest set of 1-hop neighbors as its forwarding nodes. The set of forwarding nodes must cover all its 1-hop $N(r)$ and 2-hop neighbors $N(N(r))$. The DP algorithm also assumes that 1-hop broadcasting is reliable. Thus, when receiving a broadcast message from neighbor *t*, the node selects a smallest set from $N(r)$ as forwarding nodes to enclose all its 2-hop away nodes which are not covered by the broadcasting of *t* (i.e., $N(N(r))-N(r)-N(t)$). The original algorithm piggybacks the list of forwarding nodes in the broadcast message. As a result, it breaks the wireless standard and is incompatible with other nodes not supporting this algorithm.

Wu and Li Algorithm: Wu and Li proposed a connected-dominating-set-based algorithm to calculate a set of forward nodes that from a connected dominating set [16]. The concept of Wu and Li algorithm is to combine self-pruning and domain-pruning mechanisms which use 2-hop neighbor information too. Instead of choosing the forwarding nodes on demand, the algorithm statically constructs a local connected dominating set and selects the gateway nodes by neighbor union. A node is marked itself as a gateway if it has two 1-hop neighbors that are not direct connected. In addition, it also uses pruning rules to reduce even further the set of gateway nodes.

Justifying the selection

Besides the simple flooding method, Dynamic Probabilistic Algorithm and Efficient Counting Broadcast are good representatives for the probabilistic and counting-based approaches, respectively. Moreover, both of them combine the probabilistic and counting-based concepts to reduce redundancy, but they differ in the way to determine the rebroadcast probability. As the self-pruning mechanism, we choose Scalable Broadcast Algorithm for its efficient use of neighbor information and good simulation results. Domain Pruning Algorithm and Wu and Li Algorithm are chosen for similar reasons but as the representatives of the gateway selection mechanism.

Chapter 3 System Design and Algorithm Implementation

This section first presents the system architecture of our design, and then describes the generic and algorithm-specific implementation issues are described in turn.

3.1 System Architecture

We adopt IEEE 802.11s [6] as the wireless mesh environment. The IEEE 802.11s amendment that defines a wireless LAN mesh using IEEE 802.11 MAC/PHY layers is one of the most active standards and has increasing commercial spaces.

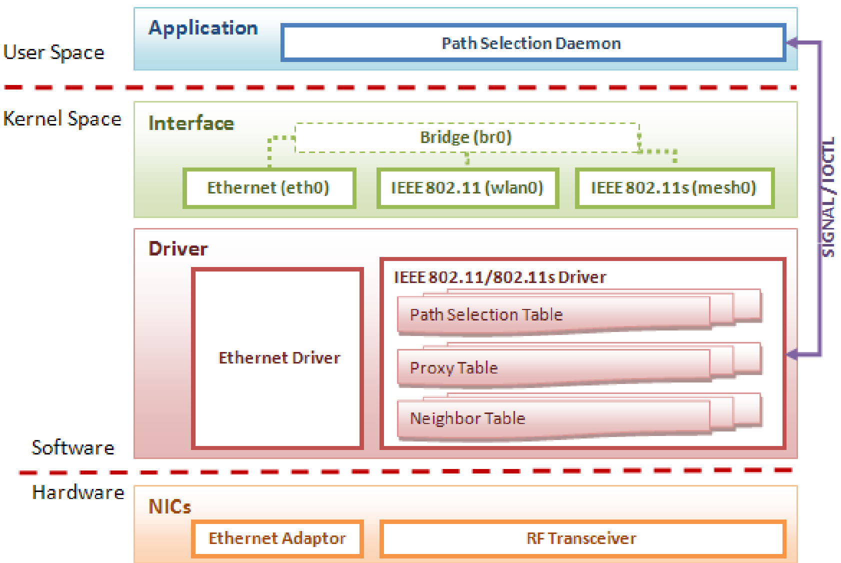


Figure 2 System Architecture

Our developing platform runs an embedded Linux (version 2.4.18). The system architecture of our design is depicted in Figure 2. In the driver layer, the IEEE 802.11/802.11s driver not only implements the functions of IEEE 802.11 specification but also supports the mesh services defined in the IEEE 802.11s amendment. Within the driver, the *neighbor table* records the neighbor information to maintain the mesh topology, and the *proxy table* and *path selection table* are the routing tables for non-mesh nodes (e.g., STAs) and mesh points (e.g., MPs), respectively. In the interface layer, two interfaces, wlan0 and mesh0, are multiplexed to serve both IEEE 802.11 networks and IEEE 802.11s networks concurrently on a single physical wireless adaptor. To bridge IEEE 802.3 traffic, an additional Ethernet adaptor can also present and co-work with the wireless adaptor on the same platform by the help of a virtual bridging interface, br0. In the user space, a Linux daemon program, called *PathSelection*, implements the mesh routing algorithm and updates those 3 tables residing in the driver.

Figure 3 shows the packet processing flows designed in our system. The main idea is to create a common framework where the implementation of each broadcasting algorithm is independent of other mesh functions. A data frame is first retrieved from the frame queue after receiving from the hardware receiver (Rx) and validating its sequence number and Time-To-Live field (TTL). Then, the unicast data and multicast

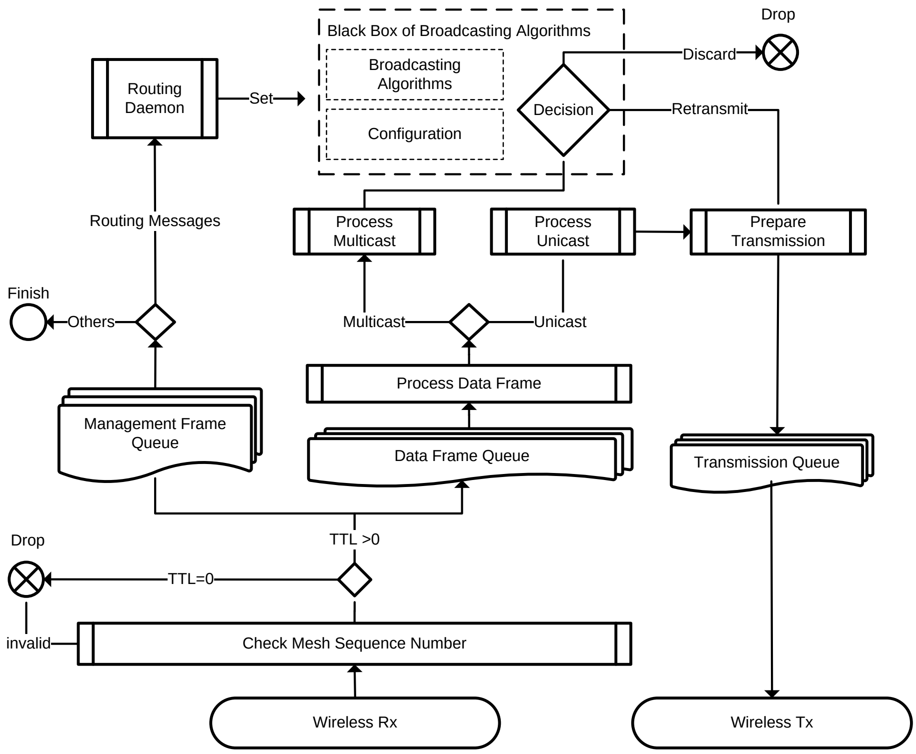


Figure 3 Packet processing flowchart

data are processed separately. For a unicast data frame, it would be sent to the transmission queue if required to be forwarded. On the other hand, a multicast data frame is fed into the *black-box of broadcasting algorithm* where our studied broadcasting algorithms are implemented. The black-box determines whether to forward the broadcasting message or not. If forwarding is required, the frame waits in the transmission queue for being transmitted to its neighbors. Then, the hardware transmitter (Tx) takes response for transmission. Besides, the necessary parameters of the black box, like the 2-hop neighbor information, are supported by the routing daemon after which processes the management frames. Obviously, this design model is flexible enough to support various broadcast algorithms.

3.2 Generic Implementation Issues

Three implementation issues are common to some algorithms and discussed in this subsection. First, a validation mechanism is devised to detect the duplicated frames caused by broadcasting. Second, a method to collect the 2-hop neighbor information is described, which is required by SBA, DP, and Wu and Li algorithms. Last, to be compatible with the wireless standard, a simple mechanism is designed to carry the gateway information to the receivers and to replace piggybacking originally implemented by some algorithms like the DP.

Duplicate packet validation

In wireless mesh networks, an MP blindly forwards a broadcast message to its neighbors, which might result in endless flooding due to the loop structure in a mesh topology. To avoid infinite rebroadcasting loop, IEEE 802.11s embeds a frame with the *Mesh-Sequence-Number* (MSEQ) field, a unique sequence number. A relaying node uses the tuple <*source MAC, MSEQ*> as a unique signature to detect duplicated frames. Therefore, a buffering method like the *Check Mesh Sequence Number* module in Figure 3 has to be implemented to store and check the tuple.

2-hop neighbor information collection

To collect the 2-hop neighbor information, a one-hop control message, called HELLO message, is introduced. A HELLO message is a broadcast frame with TTL = 1. Each mesh node periodically advertises a HELLO message to show the aliveness to its neighbors. The message piggybacks the neighbor list. Initially, only the list of

1-hop neighbors is announced and learned. After collecting all HELLO messages from its 1-hop neighbors, a mesh node can complete the list of 2-hop neighbors.

Gateway notification

Some algorithms piggyback on-demand information within its transmitted data frames like the gateway information of DP to a data frame, which results in breaking the wireless standards. As a result, the tainted data frame becomes meaningless for a node not supporting the algorithm. In our implementations, like the collection of 2-hop neighbors, the gateway information is also embedded within the HELLO messages. The solution is simple and efficient. First, it is not necessary to select different gateways and to piggyback them in a data frame on-demand, since the mobility in mesh networks is related lower. Second, the solution is compatible with wireless standards, because a mesh node without supporting a specific algorithm can completely ignore the HELLO message. Last, it reduces the computing time and media consumption by cancelling the attachments on every broadcast message.

3.3 Implementation of Each Algorithm

Each broadcasting algorithm is implemented in the black box as shown in Figure 3. After analyzing the executing flows of our studied algorithms, an implementation framework with three common running phases is designed, and they are: the periodical task, the observation phase, and the determination phase. First, the periodical task initializes algorithmic parameters like the probability value for Dynamic Probabilistic Algorithm, prepares the neighboring information, and periodically exchanges the HELLO messages. With the execution of periodical task, a mesh node could thus collect 2-hop neighbor information or construct its connected dominating set (CDS). Second, the observation phase actually processes a broadcast frame. It observes the phenomena when receiving a broadcast frame, adjusts the algorithmic parameters like the delay timer used in ECB and SBA, and queues a packet when required. Finally, the determination phase uses the observed phenomena to determine whether to forward the broadcast message. For each algorithm, the actions for each phase are presented in Table 2.

Table 2 Three phases of each algorithm

Algorithms	Periodical task	Observation phase	Determination phase
Dynamic Probabilistic Algorithm	Initiate the prob. value	Count the same frame for fine-tuning the prob. value	Prob. value
ECB	N/A	1. Queue the broadcast frame 2. Set delay timer 3. Count the same frame	Counting number & prob. value
SBA	Retrieve neighbor info.	1. Queue the broadcast frame 2. Initiate the remaining list 3. Set delay timer 4. Update the remaining list	Non-empty remaining list
DP	1. Retrieve neighbor info. 2. Assign the gateways for each neighbor	N/A	Gateway of the sender
Wu and Li	1. Retrieve neighbor info. 2. Construct local-CDS 3. Decide gateway property	N/A	Gateway property

In the periodical task, each algorithm could prepare the information for its mechanism; for example, the dynamic probabilistic algorithm initiates the probability value whenever associating or disassociating a neighbor. Besides, the SBA, DP and Wu and Li algorithms exchange the HELLO messages to retrieve the neighbor information. In particular, the a node using DP algorithm assigns its neighbors as gateways in the HELLO messages, and a node using Wu and Li algorithm would decide whether itself is a gateway according to the local-CDS. During the observation phase, the dynamic probabilistic algorithm counts the number of total received broadcasts in a time slice for fine-tuning the probability value. The ECB and SBA algorithms store the broadcast, set a timer for re-transmission decision later, and do their observing mechanism during this moment. The remaining two algorithms do not require this phase and can enter the determination phase directly. Finally, to determine whether to forward a broadcast in the determination phase, each algorithm check the specific parameters listed in Table 2.

To summarize, there are three lessons learned from the implementation. The first lesson is the wireless media resource occupied by the HELLO message. Because the HELLO message is a broadcasting-type control frame, its transmitting data rate is constrained by the common data rate of all associated neighbors. Hence, the message shares more resource than a unicast data frame. Fortunately, the HELLO message is issued by the periodical task, which can be arranged with a long execution period to reduce the side-effect. The second lesson is the precision of a timer required by the delay-based approaches such as ECB and SBA in the driver level (firmware solution). The minimum interval, 10 milliseconds as default, is constrained by the kernel. Comparing with the interval of a lengthy data frame (i.e., 1573 microseconds to transmit 1500 bytes Ethernet payload at 11Mbps in IEEE 802.11 b mode), this interval is too long in the wireless world. Although the value is adjustable by kernel re-compiling, the side-effect such as the cost of polling is also considerable. Therefore, a hardware solution is acceptable when implementing the observation phase. The last lesson is the buffer required by the delay-based algorithms during their observation phase. Apparently, the longer delay interval is, the more buffer is required. Besides, the buffer requirement is also proportional to the broadcast traffic rate. As a result, to prevent data loss due to insufficient buffer, the development of ECB and SBA should consider both of delay interval and expected broadcast traffic rate.

Chapter 4 Evaluation

4.1 Evaluation and Logging Mechanism

In order to conduct experiments with different parameter settings and different running conditions, a generic evaluation and logging mechanism is devised. The mechanism defines three roles in a WMN: sender, receiver, and logger. A sender takes response of initializing experiments, generating various experimental broadcast patterns, and collecting the experiment results. A logger is a program located on a mesh node where the studied broadcasting algorithms are implemented. During a mesh node suffering a series of experiments, a logger records the statistics from the running algorithm, and finally reports a sender the result. Last, a receiver simulates an end point, and the reliability from the user's view can thus be collected.

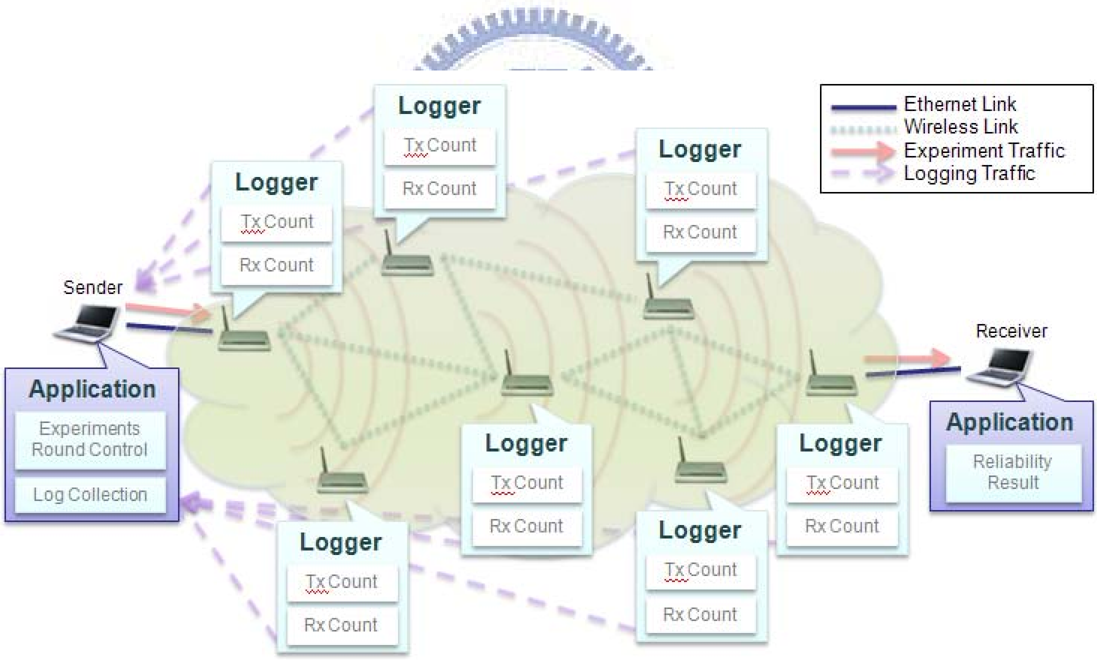


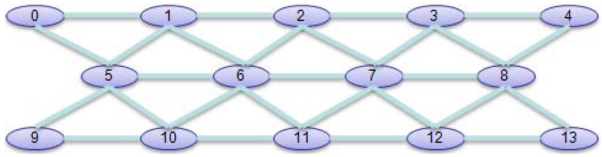
Figure 4 Evaluation and logging mechanisms

Figure 4 depicts the execution of our evaluation and logging mechanism. When starting a new turn of an experiment, an *experiments round control* (ERC) module in the sender transmits a *STARTUP* command to all nodes to install the algorithmic parameters and reset the execution of the *periodical task* module mentioned in last subsection. Then, different broadcasting patterns are issued by the ERC module. In the end of an experiment, the ERC module transmits a *FINISH* command to each node. When receiving the ending signal, the logger modules on all nodes transmit their statistics to the *log collection* module in the sender and reset the database for the next experiment.

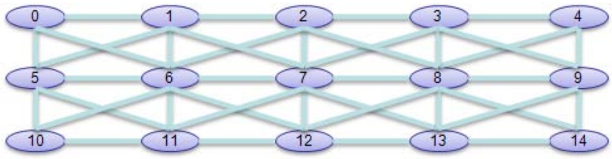
4.2 Benchmarking Environments

The experiment mesh node, Realtek *RTL8186*, is a commercial system-on-a-chip, embedded with an Ethernet and single-radio 802.11b/g controller, and a 180 MHz 32-bit MIPS processor. All experiments were conducted on the IEEE 802.11s-based wireless mesh environment where deployed less than 30 fixed-locations mesh nodes and used one common channel to transmit both data and control messages. Each evaluation ran 5000 broadcast frames, and there is a 100 milliseconds interval between each frame. In addition, each broadcast frame is transmitted with 1 Mbps as the commonest off-the-shelf WLAN solutions.

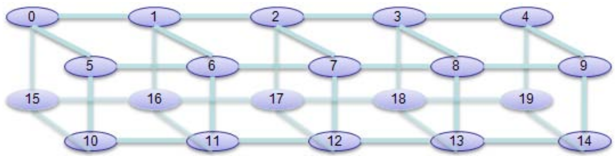
As shown in Figure 5, we defined three general scenarios for evaluation: the Triangle Mesh, 2D Fully Connected Cube Mesh (2D Mesh), and 3D Cube Mesh (3D Mesh) in which the Packet Error Rates (PER) are observed as 9.04%, 7.04% and 9.01% respectively during our testing. The main idea of first two is the general connected matrixes structure that can help us to observe the behavior of each algorithm, and the last topology shows the situation when a mesh is deployed in a building.



(a) Triangle Mesh, PER = 9.04%



(b) 2D Fully Connected Cube Mesh, PER = 7.04%



(c) 3D Cube Mesh, PER = 9.01%

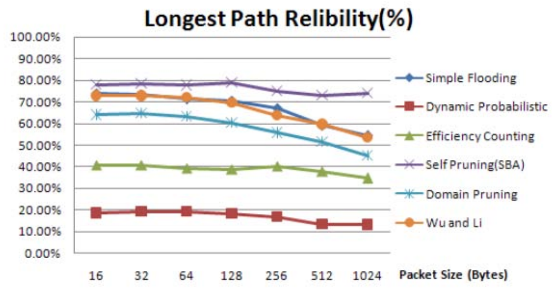
Figure 5 Evaluation scenarios

We defined four metrics to evaluate the studied broadcast algorithms: the Longest Path Reliability, Average Reliability, Forwarding Ratio, and Broadcasting Efficiency. First, the longest path reliability measures the reliability at the mesh node that is placed farthest away from the broadcast sender. It simulates the service-discovering situation where the client issues a broadcast to search a service; for example, a DHCP client asks the DHCP service by broadcasting a DHCP discovery message. Hence, the longest path reliability reveals the availability of broadcasting-based service. Second, the average reliability measures the mean reliability of all mesh nodes. It simulates the case where a server advertises important service information; for example, a gateway node in an IEEE 802.11s mesh proactively announces its existence. Therefore, the average reliability is the reliability of such service. Third, the forwarding ratio measures the re-transmission ratio of a received broadcast frame. Obviously, a higher ratio represents more usage of wireless media resource. Last, we defined the broadcasting efficiency, which is an index to express the contribution of each forwarding frame to the total reliability.

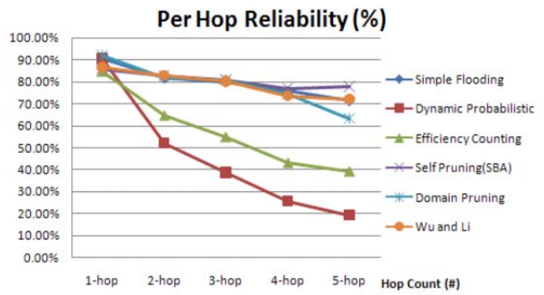
4.3 Experimental Results

4.3.1 Longest Path Reliability

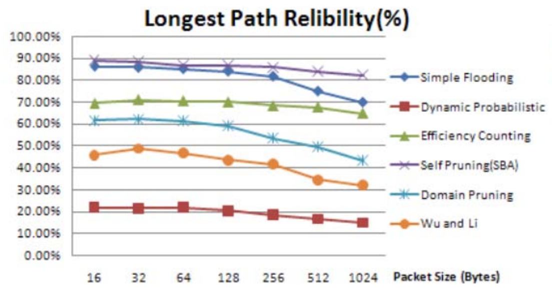
Figure 6(a) shows the longest path reliability with different broadcast data size on the triangle mesh. The reliability of the simple flooding, domain pruning, and Wu and Li algorithms are degraded about 18~20% as the packet size increasing. This is reasonable because a lengthy packet has higher probability to collision with others when transmitting on the air. On the other hand, the degradation of reliabitliy is relative small in ECB and SBA algorithms because of the delay mechanism which result in a staggered transmission and ligher collision probability. Besides, the dynamic probabilisitc algorithm also suffers from the enlargment of packet size. It is also reasonsable because the reliability of this algorithm is much lower than the one of others. Therefore, the collision probability of the dynamic probabilisitc algorithm is also lower than the probability of other algorithms. The reason of the lower reliability coming comes from the aggregation of the increased hop count. Apprarently, the retransmission probabilities for both of the dynamic probabilistic and ECB algorithms is exponentally decreasing as the hop counts. Figure 6(b) verifies our explanation by showing per-hop reliability. The per-hop reliability decreases dramatically for the dynamic probabilistic and ECB algorithms, but it is more steady for other algorithms. Figure 6(c) and 6(d) show the experimental results of longest path reliability on the 2D mesh and 3D mesh respectively. The results are similar to the case of the triangle mesh. For these reasons, it is recommended to adopt the delay-based algorithm for transmitting a service-discovering broadcast message. It is also recommended to avoid using the probabilistic-based algorithm which leads to the degradation of per-hop reliability.



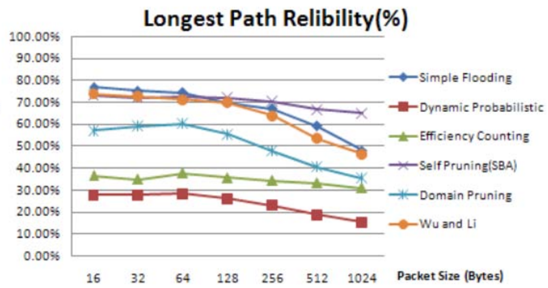
(a) Longest path reliability on the triangle mesh



(b) Per-hop reliability on the triangle mesh



(c) Longest path reliability on the 2D mesh

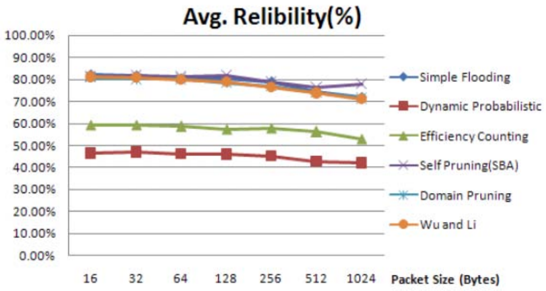


(d) Longest path reliability on the 3D mesh

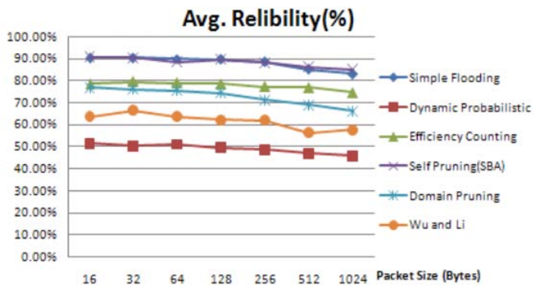
Figure 6 Details of Reliability

4.3.2 Average Reliability

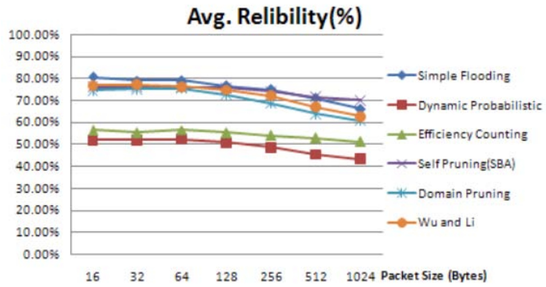
Figure 7 shows the average reliability, which is the mean reliability of all mesh nodes, for different topologies. According to the results, simple flooding, SBA, and the dynamic probabilistic algorithms are more independent on the topology change. Thus, the simple flooding and SBA outperform than other algorithms, and the dynamic probabilistic algorithms are worse than others. Due to lighter collision mentioned in the previous section, the average reliability of SBA is slightly better than simple flooding especially for larger size of data. Overall, the SBA shows the best reliability among all algorithms, which is much different from the previous simulations in [21-22]. The reasons are that the delay-based mechanism allieviates the collection problem, and the remaining list after the observation phase enhances reliability.



(a) Average reliability on the triangle mesh



(b) Average reliability on the 2D mesh

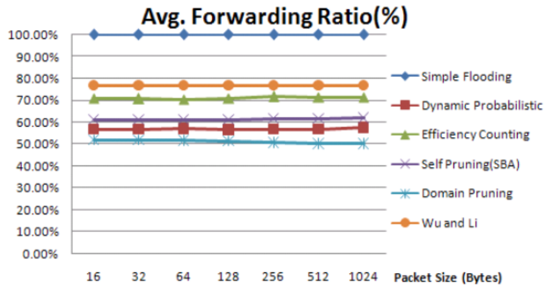


(c) Average reliability on the 3D mesh

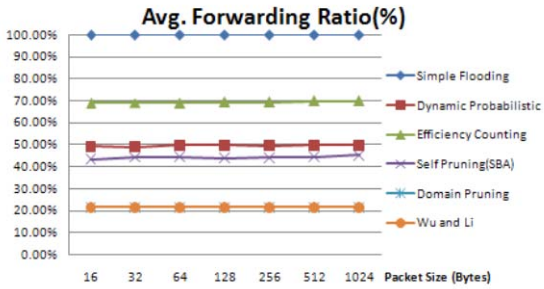
Figure 7 Average Reliability

4.3.3 Forwarding Ratio

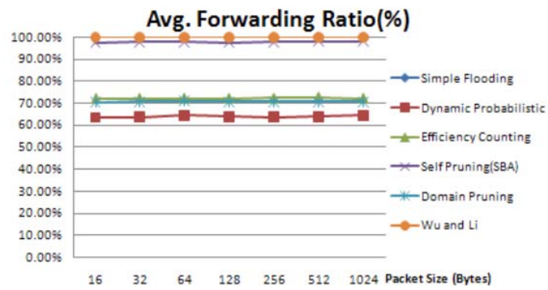
The forwarding ratio is the re-transmission ratio of a received broadcast frame. Figure 8(a) shows the average forwarding ratio for different size of broadcast on a triangle mesh. Apparently, the packet size is not a factor of the forwarding ratio. The results on a 2D and 3D mesh, shown in Figure 8(b) and Figure 8(c) respectively, are also compliant with the same observation. Through different topologies, we find that the average forwarding ratio of simple flooding (100%), dynamic probabilistic algorithm (50%), and ECB (70%) are not influenced by the topology. However, the topologies is the major factor that influences the forwarding ratio of the algorithms using neighbor information. In particular, the Wu and Li algorithm varies most (20% to 100%) among all algorithms, because a mesh node using Wu and Li is easy to become a gatway node by as judging the disconnectivity of its 1-hop neighbors. Thus, the Wu and Li algorithm would act as flooding in a non fully connected topology.



(a) Avg. Forwarding Ratio on the triangle Mesh



(b) Avg. Forwarding Ratio on the 2D Mesh



(c) Avg. Forwarding Ratio on the 3D Mesh

Figure 8 Average Forwarding Ratios

4.3.4 Broadcasting Efficiency

In order to evaluate the efficiency of a broadcast algorithm, we define an index, broadcasting efficiency, which considers both the forwarding efficiency and the reliability factors to express the contribution of each forwarding frame to the reliability. To explain our definition, some variables are defined here:

N(p): number of neighbors for node ***p***;

R(p): observed number of effective broadcasts received by node ***p***;

T(p): expected number of forwarding times issued by node ***p***;

PER: the packet error rate.

The main idea of the index is the result of multiplying the average reliability by a factor which represents the forwarding efficiency. To evaluate the forwarding efficiency, we define the 'merit' of a forwarding, i.e., ***R(p)***, which counts the number of effective (correctly and non-duplicated) broadcasts. Therefore, of the higher merit an algorithm is, the more efficient it is. By taking the PER into consider, finally, the forwarding efficiency is the number of observed effective broadcasts divided by the number of expected received broadcasts for each algorithm, i.e., the number of expected broadcasts is contributed from $T(sender) \times N(sender) \times PER$ for the broadcast sender, and $T(p) \times (N(p) - 1) \times PER$ for other nodes. Thus, the definitions of forwarding efficiency and broadcasting efficiency are listed as follows:

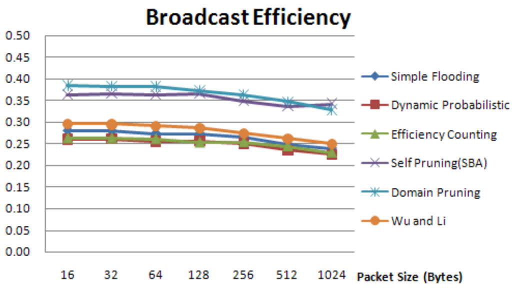
$$Forwarding\ Efficiency = \frac{\sum_{nodes} R(p)}{T(sender) \times PER + \sum_{nodes} T(p) \times (N(p) - 1) \times PER}$$

$$Broadcast\ Efficiency = Forwarding\ Efficiency\ \times Avg.Reliability$$

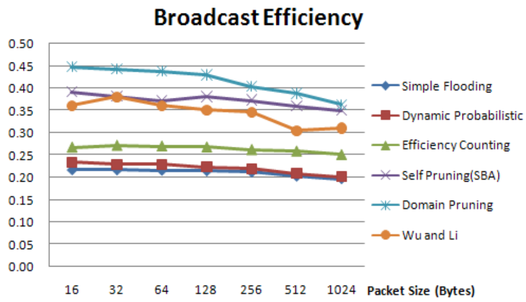
Based on the index, a broadcast algorithm is more efficient if both the average reliability and forwarding efficiency are higher in the meantime. Thus, the higher index value a broadcast algorithm shows, the more efficient the broadcast algorithm is.

The broadcasting efficiency results of each algorithm are drawn in the Figure 9. We can find that the DP algorithm outperforms than all other algorithms not only for different size of broadcast data but also for different topologies. By checking the Figure 7 and Figure 8, we can know that its advantage comes from the low forwarding ratio which is the result of gateway selection, which effectively reduces the number of forwarders. On the other hand, the efficiency of another gateway selection algorithm, the Wu-and-Li algorithm, degrades to the worst algorithm, the simple flooding, in the 3D mesh topology. The reason is that the Wu-and-Li algorithm statically determines whether it should be a gateway node by checking the local-CDS, but the gateway node is run-time decided by the sender in the DP algorithm. The local-CDS in the 3D mesh topology contains no fully connected local topology, so all mesh nodes become forwarders, which leads to the Wu-and-Li algorithm degrades to the simple flooding algorithm. Figure 8(c), where the forwarding ratio of the Wu-and-Li algorithm is 100%, also verifies the conclusion.

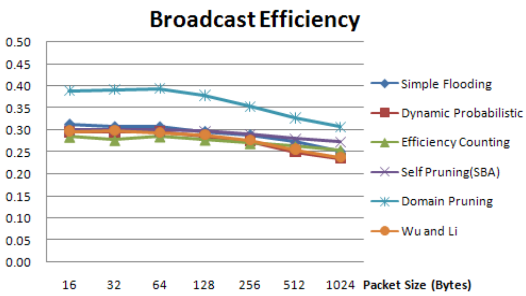
In addition, although the SBA is best at both of the longest path reliability and average reliability as shown in previous section, its broadcasting efficiency is not the best one among all algorithms. Conceivably, the forwarding method is a little more inefficient in SBA, which results in worse broadcasting efficiency than the index of DP. Besides, the results of the simple flooding, dynamic probabilistic algorithm and ECB are all worst and similar when using them under the same topology.



(a) Broadcast efficiency on the triangle mesh



(b) Broadcast efficiency on the 2D mesh



(c) Broadcast efficiency on the 3D mesh

Figure 9 Broadcast Efficiency

4.4 Summarization

The lessons learned from the evaluations under the real-world platform are summarized as follows,

1. The reliability of simple flooding is not as good as the results in simulation.

In the simulation results presented by [21-22], the reliablity of simple flooding is approximate to 100%. However, in our real-world experiments, the reliabilities of simple flooding, as shown in Figure 7 and Figure 8, vary from 50% to 90% and significantly differ from simulation. We deem that it is because the interference and collision are underestimated in simulations. Therefore, our work evaluating the broadcast algorithms in the real-world testbed could reveal the properties of those algorithms more correctly.

2. Probability-based algorithms are not suggested.

The probability-based algorithms show terrible reliability and efficiency in our evaluations, because its re-transmission probability is accumulatively decreasing with the increasing hop counts. Though the probability-based algorithms are the variant of simple flooding, according to our experiments, their efficiency is equal or less than the simple flooding in the real world evaluating environment. Therefore, it is not suggested to use the probability-based algorithms in wireless mesh networks.

3. It is recommended to use the self-pruning algorithm in small-scaled mesh networks, and use the domain-pruning algorithm in large-scaled mesh networks.

The reason for the first one is that the self-pruning algorithm provides the best reliability with the side-effect of queuing latency. Therefore, it would be acceptable to use the self-pruning algorithm for a small-scaled mesh network. For a large-scaled mesh network, the domain-pruning is recommended due to its good efficiency, which presents lower forwarding ratio, slighter collision probability, and acceptable reliability.

Chapter 5 Conclusions

In this work, we investigate the real design issues for the implementation of broadcast algorithms in the wireless mesh network. A control message is introduced to collect 2-hop neighbor information, select gateway nodes, and provide the compatibility with the wireless standard at one time. We design a flexible architecture above which it is easy to change the broadcast algorithm. For each broadcast algorithm, a three-phase execution concept is abstracted and devised, which simplifies the implementation. We evaluate and discuss five representative broadcast algorithms on the real-world testbed. Through comparing the reliability for the simple flooding algorithm in simulation with the one in the real-world testbed, we show that the simulation results significantly differ from testbed. We assume the main reason of the difference is that the simulators are unable to precisely reflect the packet error rates on overlapped collision domains. Thus, the evaluation of broadcast algorithms with the real-world testbed could reveal the properties of those algorithms more correctly. The lessons learned are the reliability credited to fewer collisions for the delay-based algorithms, the untrustworthy of probabilistic-based algorithms due to the decreasing re-transmission probability by hop count, the efficiency of domain-pruning algorithm because of the run-time gateway selection, and the different topology density and connectivity favoring different algorithms.

Through observing the experimental results, we show that every algorithm has its advantages and disadvantages. Therefore, in the future work, one improvement method could come from mixing the features of several broadcast algorithms together. For example, in the domain-pruning algorithm, if a sender brings in the listening-during-delay concept used in SBA to actively listen whether the retransmissions are performed by its forwarders, the transmission failure would be reduced and the overall reliability could be raised. Another improvement idea resulting from the observation mentioned in chapter 4 is that different algorithms favor different scaled of mesh networks. Hence, in the future work, we will devise an adaptive broadcast algorithm that selects a most suitable algorithm from several implemented algorithms according to the observed topology in the runtime.

References

[1] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "*Flooding for reliable multicast in multi-hop ad hoc networks.*" International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM), pp. 64–71, 1999.

[2] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "*The broadcast storm problem in a mobile ad hoc network.*" ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp. 151–162, 1999.

[3] Lou, W., Wu, J, "*Toward broadcast reliability in mobile ad hoc networks with double coverage.*" IEEE Trans. on Mobile Computing 6(2), pp. 148-163, 2007

[4] Plummer, D., "*An Ethernet Address Resolution Protocol*". RFC 826, MIT-LCS, November 1982.

[5] IEEE 802.1D, "IEEE standard for local and metropolitan area networks--Common specifications--Medium access control (MAC) Bridges", ANSI/IEEE Standard 802.1D, 1998.

[6] IEEE P802.11s/D1.06, "*Draft amendment to standard IEEE 802.11: ESS Mesh Networking*." IEEE, work in progress, 2007.

[7] Zhang, Q., Agrawal, D.P., "*Dynamic probabilistic broadcasting in manets.*" Parallel and Distributed Computing 65(2), pp. 220-233, 2005.

[8] A Mohammed, M Ould-Khaoua, L Mackenzie: "*An Efficient Counter-Based Broadcast Scheme for Mobile Ad Hoc Networks.*" Lecture Notes in Computer Science – Springer, 2007.

[9] Sun Qiang, Li Layuan, Chen Niansheng, Aziz S, "*A Cross-Layer Design for Broadcast Algorithm in MANETs.*" IEEE Wireless Communications, Networking and Mobile Computing, 2006, WiCOM 2006, 2006.

[10] Li Layuan, Zheng Feng, Chunlin Li and Qiang Sun, "*A Distributed Broadcast Algorithm for Wireless Mobile Ad Hoc Networks.*" Advances in Multimedia Modeling – Springer, 2007.

[11] Peng, W., Lu, X.-C, "*On the reduction of broadcast redundancy in mobile ad hoc networks.*" 1st ACM international symp. On Mobile ad hoc networking & computing (MOBIHOC), pp. 129–130, IEEE Press, Piscataway, NJ, USA, 2000.

[12] H. Lim and C. Kim, "*Multicast tree construction and flooding in wireless ad hoc networks.*" ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), 2000.

[13] Lim, H., Kim, C., "*Flooding in wireless ad hoc networks.*" Computer Comm. 24(3–4),

pp. 353-363, 2001.

[14] Lou, W., Wu, J., “*On reducing broadcast redundancy in ad hoc wireless networks.*” IEEE Trans. on Mobile Computing 1(2), pp. 111-123, 2002.

[15] Bin Shen, Bo Li, Bing-Xin Shi, “*Improved Approach to Enhanced Internet Connectivity for Mobile Ad Hoc Networks.*” Journal of Southeast University (English Edition), March 2007.

[16] Wu, J., Li, H., “*On calculating connected dominating set for efficient routing in ad hoc wireless networks.*” DIALM ’99: 3rd Int. Workshop on Discrete algorithms and Methods for Mobile Computing and Comm., pp. 7-14. ACM Press, New York, NY, USA, 1999.

[17] A. Qayyum, L. Viennot, and A. Laouiti, “*Multipoint relaying: An efficient technique for flooding in mobile wireless networks.*” Technical Report 3898, INRIA - Rapport de recherche, 2000.

[18] François Ingelrest, David Simplot-Ryl, “*Maximizing the Probability of Delivery of Multipoint Relay Broadcast Protocol in Wireless Ad Hoc Networks with a Realistic Physical Layer.*” 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2006), 2006.

[19] Keshavarz-Haddad, Alireza, Ribeiro, Vinay, Riedi, Rudolf, “*DRB and DCCB: Efficient and Robust Dynamic Broadcast for Ad Hoc and Sensor Networks.*” IEEE Communications Society on Sensor, Mesh and Ad Hoc Communications and Networks(SECON) , 2007.

[20] Hasegawa, Keigo, Fuji, Takeo, Umebayash, Kenta, Kamiya, Yukihiro, Suzuki, Yasuo, “*A Dialogue-based MAC Protocol for Efficient Broadcasting in Ad Hoc Networks.*” Mobile and Wireless Communications Summit, 16th IST, 2007.

[21] Williams, B., Camp, T., “*Comparison of broadcasting techniques for mobile ad hoc networks.*” 3rd ACM Int. Symp. on Mob. Ad Hoc Networking & Computing, pp. 194-205. ACM Press, New York, 2002.

[22] Talmai Brand˜ao de Oliveira, Victor Franco Costa, and Fab’ıola Greve, “*On the Behavior of Broadcasting Protocols for MANETs Under Omission Faults Scenarios.*” Dependable Computing– Springer, 2007.

[23] I. F. Akyildiz, et al., “*Wireless mesh networks: a survey.*” Computer Networks, vol. 47, issue 4, Pages 445-487, 2005.

[24] J. Bicket, et al., "*Architecture and evaluation of an unplanned 802.11b mesh network.*" Proc. 11th Ann. Int’l Conf. Mobile Computing and Networking, pp. 31-42, 2005.

[25] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K.-C. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan., "*Effects of Detail in Wireless Network Simulation.*" SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, 2001.

[26] Kwan-Wu Chin, John Judge, Aidan Williams, Roger Kermode, "*Implementation experience with MANET routing protocols.*" ACM SIGCOMM Computer Communication Review, v.32 n.5, November 2002.

[27] F. Haq and T. Kunz, "*Simulation vs. emulation: evaluating mobile ad hoc network routing protocols.*" International Workshop on Wireless Ad-hoc Networks (IWWAN'05), May 2005.

[28] Wilson T.H. Woon, Tat-Chee Wan, "*Performance evaluation of IEEE 802.15.4 wireless multi-hop networks: simulation and testbed approach*." InderScience International Journal of Ad Hoc and Ubiquitous Computing, pp. 57-66, 2008