國立交通大學

資訊科學系

碩士論文

行動隨意網路拓樸資訊收集之分析

An Analysis of Topology Information Gathering in Mobile Ad Hoc Networks

研究生：王獻綱

指導教授：陳　健　教授

中華民國九十四年六月

行動隨意網路拓樸資訊收集之分析

An Analysis of Topology Information Gathering in Mobile Ad Hoc Networks

研究生：王獻綱　　Student：Hsien-Kang Wang

指導教授：陳健　　Advisor：Chien Chen

國立交通大學

資訊科學系

碩士論文

A Thesis
Submitted to Department of Computer and Information Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
In partial Fulfillment of the Requirements
For the Degree of
Master
In

Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

行動隨意網路拓樸資訊收集之分析

研究生：王獻綱　　　　指導教授：陳　健

國立交通大學資訊科學系

中文摘要

拓樸資訊經常被使用在網路常見的功能如廣播、繞徑等。單一跳躍鄰居列表資訊是拓樸資訊的基本單元，用以得知更多的拓樸資訊。此種資訊可由在整個網路上的所有節點交換一次招呼封包取得。然而，行動隨意網路可移動的特性使得單一跳躍鄰居列表隨著時間不同而頻繁地改變，並且使得鄰居列表資訊產生錯誤。在本論文中，我們將推導節點在一段時間內，離開原來的傳輸範圍的機率，以分析移動的程度將對此資訊錯誤率的關係，並驗證此錯誤率對網路效能產生的影響以決定招呼封包的傳送週期，藉由此機率模型，推導鏈結變化率，以得知適當的傳送週期。為了取得正確的單一跳躍鄰居列表，招呼封包必須週期性的發送。然而，資訊的精確度與傳送週期的關係是密不可分的，這意味著高精確度往往會伴隨著大量的招呼封包產生，其結果將導致可觀的成本。許多的研究假設單一跳躍鄰居列表是已知的，因此這項成本就可以被忽略。當實作在真實世界時，這是不合理的，我們將透過模擬來說明此問題。最後利用 ns2 來比較分析結果與模擬測試是否符合。

關鍵字：拓樸、機率分析、單一跳躍鄰居列表、廣播、行動隨意網路。

An Analysis of Topology Information Gathering in Mobile Ad Hoc Networks

Student: Hsien-Kang Wang **Advisor: Dr. Chien Chen**

Institute of Computer and Information Science
National Chiao Tung University

Abstract

Topology information is frequently used in many network functions such as broadcast, route discovery, etc. One-hop neighbor lists are the basic of topology information and used for complicated topology information. The lists can be obtained by all nodes of the network exchanging hello packets exactly once. However, the mobility feature of the wireless Mobile Ad Hoc Networks (MANETs) causes the list different frequently. It also causes the error of the 1-hop neighbor lists. In this paper, we analyze the relationship between the degree of mobility and neighbor information error rate by deriving the probability that a node leaves its original transmission range. We also verify the influence of neighbor information error rate on network performance so as to determine the transmission period of hello packets. According to the probability model, we can deduce the link change rate. The proper hello period can also be obtained by the value of link change rate. In order to obtain accurate 1-hop neighbor information, the hello packets must be sent periodically. However, the accuracy is highly related with the hello transmission period. It means that a large number of hello packets are needed to maintain high accuracy, and consequently a considerable overhead follows. Many researches make an assumption that 1-hop neighbor lists are already known, thereby neglecting the overheads stem from hello messages. It is not reasonable when it comes to real world implementation. We demonstrate the problem through a series of simulations. Hello packet period is also in connection with mobility model. The mobility model dominates the variation speed of topology information. Therefore, the accuracy of topology information is also affected. In this paper, we analyze the relationship between hello packet period, mobility model and the accuracy of topology information. We also derive the probability that a node leaves its original transmission in certain time. The probability can be used to determine proper hello packet period. Last, the ns2 simulator was adopted to compare the results of the analysis and simulation.

Keywords: topology; probability analysis; 1-hop neighbor list; broadcast; mobile ad hoc networks (MANETs).

誌謝

本篇論文的完成，要歸功於我的指導教授陳健博士，老師不厭其煩的容忍我在做研究的路程中所犯的錯誤，當我遇到瓶頸時幫助我尋找突破的方向，讓我受益良多，也因此學得如何從頭到尾以科學的精神面對一個難題。讓我得以順利完成本篇論文，在此表達最誠摯的感謝。同時也感謝我的論文口試委員，交大的曾煜棋教授、簡榮宏教授及清大的陳志成教授，他們從客觀的角度提出了許多的寶貴意見，讓我的論文更加完整。

感謝我的夥伴徐勤凱，由於我們的互相討論及在論文上的協助，使我的研究更為完整，在艱苦的碩士生涯給我相當大的幫助及啟發。另外我也要感謝實驗室的學長、同學、學弟們，吳奕緯、陳盈羽、林俊源、徐勤凱、陳咨翰以及劉上群等人，他們在都在我最失意，最消沈時給我友情的溫暖，讓我能重拾信心，繼續走完人生最後的學生時期。

感謝我的朋友，丁瑛怡、林世平、黃智強、張榮仁及王展宇，他們以過來人的經驗，指導了我做出了許多正確的抉擇，而不致迷失了自我。，這些好朋友們就像是明燈般照亮了昏暗的旅程。

最後，我要感謝我的家人對我的關懷及支持，親情是無法取代的，我由衷感謝他們支持我在退伍後續繼求學，我要向他們致上最高的感謝。

Table of Content

Chapter1: Introduction

Topology control in Mobile Ad Hoc network has been widely discussed. In order to have accurate topology information, sending hello packets is considered an effective way to achieve the goal. However, the accuracy of topology information relies on the delay between the error happening and update process. It means that a large number of hello packets are expected to maintain high accuracy. Excessive control packets will consume precious resources in wireless network such as energy and channel, and thus deteriorate the network performance.

In this paper, we study the link change rate in order to find the proper value of hello period in the first part. This rate is related to the node speed. In the second part, we focus on the reduction of hello packets without sacrificing the network performance.

The remainder of the thesis is organized as follows. Chapter 2 presents the theoretical period of hello packets. The reduction of hello packets is proposed in Chapter 3. Finally, conclusions are summarized in Chapter 4.

Chapter 2: The period of hello packets

2.1 Introduction

The influence of mobility in the network120

is that the current 1-hop neighbor lists may be incorrect. When the moving speed is high, the neighbor error rate would also be serious. One way to settle the problem is all nodes in the network sending hello packets to correct all erroneous information. Thus, there is a relation between the error rate and the hello period.

We probe into the mobility feature by analyzing the probability that a node leave its original transmission range in terms of time axis. The probability can indirectly indicate the accuracy of the topology information. We believe that the accuracy information can provide the network administrator to determine the hello period. We adopt the Random waypoint model that is random and uniform distributed on node speed, moving direction and destination. The pause time is set to zero so that all nodes will keep moving.

2.2 The analysis model

First, we will derive the relative velocity of any two moving vectors under Random Waypoint model. As shown in Figure 1, if we take node A as the reference point, then from the viewpoint of node A, node B moves at a relative velocity v_r, and node A remains static. The relative velocity v_r is the average of all the possible combination of node A and node B that forms a triangle with v_r. Thus

$$v_{r=} \frac{\int_0^{2\pi} \sqrt{a^2+b^2-2ab\cos\theta}\, d\theta}{2\pi} \tag{1}$$

Since the speed is uniform distributed between 0 and v_{max} , the average speed of a node is $v_{max}/2$.

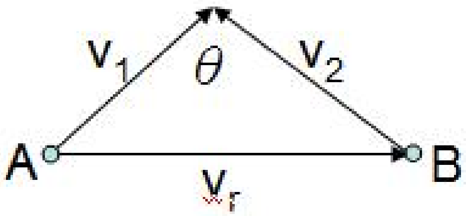

Figure 1. The relative speed

Have the relative speed in mind, we can start the probability that a node leave its original transmission range after a certain time. As shown in Figure 2, the probability of a node locates at node A is $\frac{2\pi(R-x)}{\pi R^2}$. Consider node A moves t second at the speed v_r , the smallest probability of node A leave the range is the situation when x equals v_r *t, which exists only one direction toward the circumference. However, under the condition node A moves the distance v_r*t, it still has the chance out of the range if it locates between node A and node C. That is, x is smaller than v_r*t. Thus, the probability that node A leaves the range is $\frac{2\alpha}{360}$ since the direction is uniform distributed. The value of α is the largest angle that node A reaches the circumference with the distance v_r*t. It depends on the length of x and the angle θ. Thus, we should derive θ first. From the cosine formula, we have

$$\cos\theta = \frac{(R-x)^2 + (vt)^2 - R^2}{2(R-x)(vt)} \tag{2}$$

Therefore, the probability a node leave the range after t seconds can be calculated as follows:

$$\int_0^{vt} \frac{2\pi(R-x)}{\pi R^2} * \frac{2(\frac{\pi}{2}-\theta)}{2\pi} dx \tag{3}$$

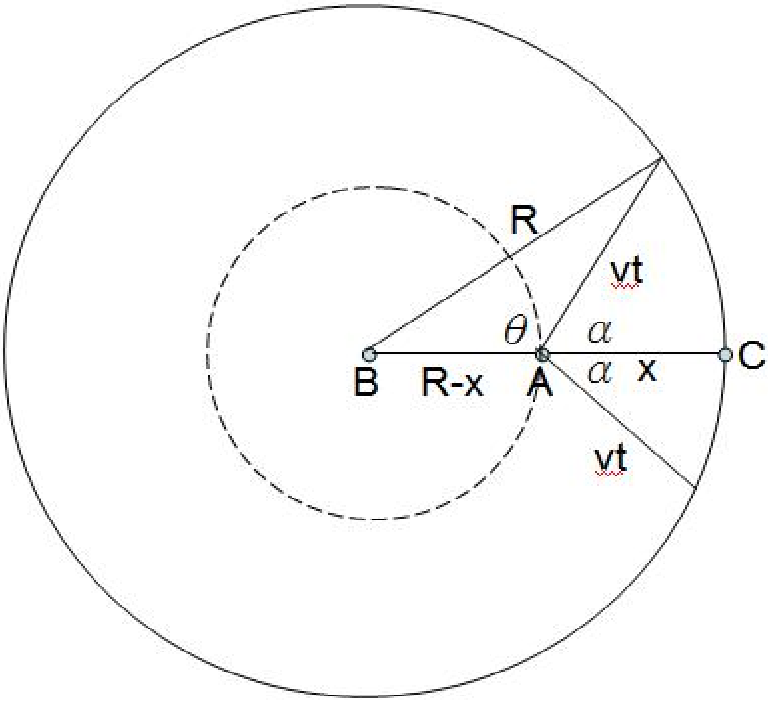

Figure 2. The probability model of a node leaves the range.

2.3 Simulation Environment setup

We use event-driven simulator ns2 [9] to simulate the performance. The simulated network contains 100 nodes. These nodes are placed randomly using a “setdest” program supported by the ns2. The map size is 5*5 and 7*7, where a unit is of the length of communication radius. The communication radius of the node is set to 250 meters. The mobility pattern is Random Waypoint model. Hello packets are sent periodically according the predetermined interval during the entire simulation time.

2.4 The comparison of analysis and simulation results.

To simulate different degree of mobility, we choose two speed intervals, which are ranging from 0 m/s to 10 m/s and 0 m/s to 20 m/s. Since each node have a probability leaving the range after a certain time, we measure the remaining number of numbers with the time goes by.

From equation (3), we have the probability in terms of different time, which is shown in blue line in Figure 3. The purple line is the simulation results. We can observe that the two lines are very close. The same situations are also shown in Figure 4 with the other speed interval. We can observe that the analysis and simulation results are near consistent. Note that the slope in Figure 4 is steeper than in Figure 3. This is because the different degree of mobility would have different rate of departure. It can also indicate that when moving speed is faster, the information update should be more frequent.

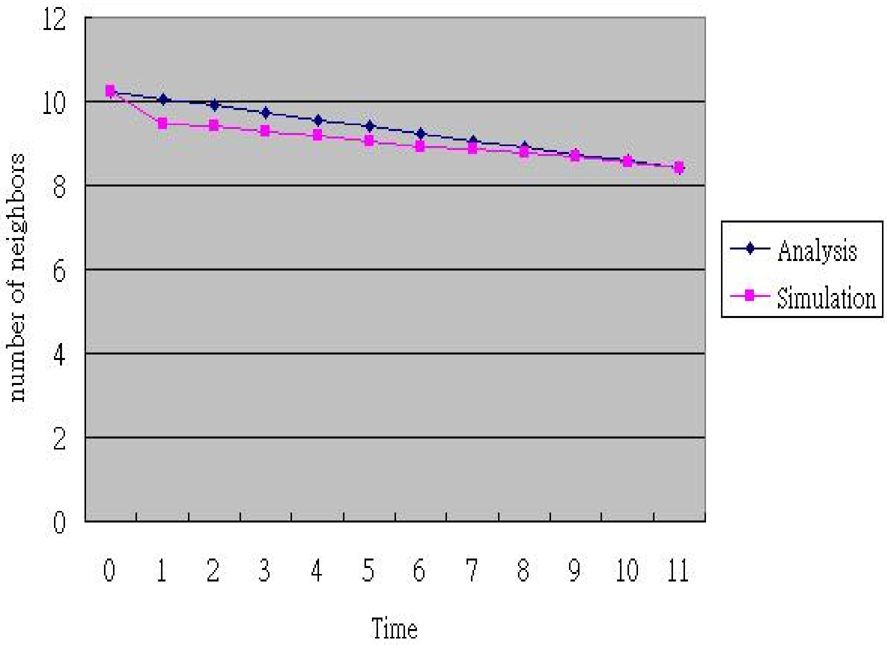

Figure 3. The comparison under the speed interval between 0 to 10 m/s

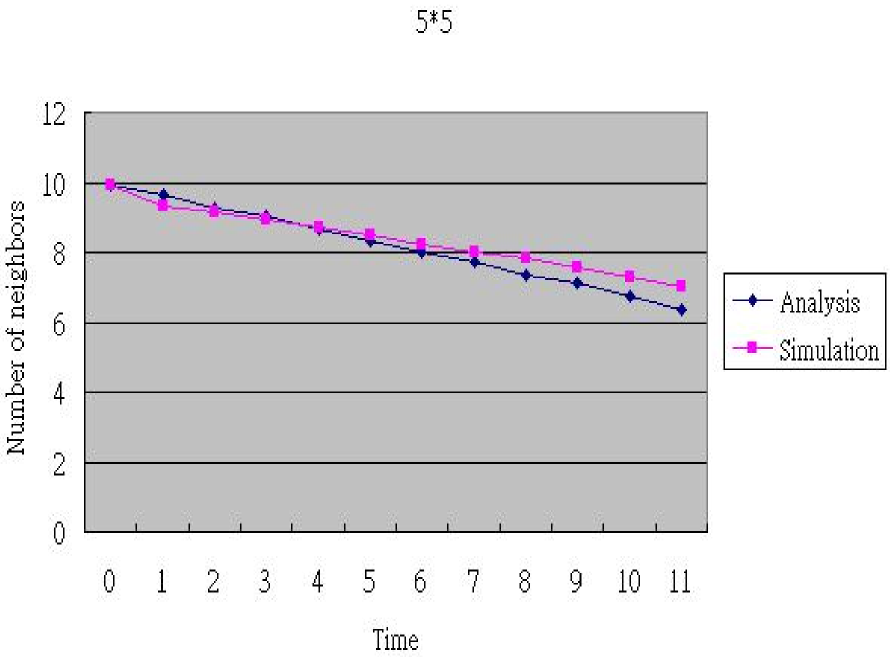

Figure 4. The comparison under the speed interval between 0 to 20 m/s

2.5 The influence of error rate on performance

2.5.1 Introduction

In the previous section, we derive the error rate in the view of time. The next we need to investigate is how the error rate would affect the performance. Broadcast in MANETs is one of the applications that needs precise rebroadcast choice; otherwise, the broadcast storm problem [1], which leads to excess rebroadcasts, may occur. Many schemes have been proposed to settle this problem. These schemes can be classified into neighbor knowledge schemes [2] [7] [8] and non-neighbor knowledge schemes [1] [3]. In general, neighbor knowledge schemes perform better than non-neighbor knowledge schemes in terms of rebroadcast numbers, for the nodes are aware of their neighborhood.

2.5.2 The selected broadcast scheme

In this section, we choose a proposed neighbor-knowledge scheme to test the influence of neighbor error rate on broadcast performance. The simulated broadcast scheme is self-pruning, described as follows. When receiving a broadcast request, the node compares its neighbor list to the sender’s neighbor list. If the additional nodes it can cover are more than one, it rebroadcast; otherwise, the request will be drop. The disadvantage of original self-pruning is that it has similar behavior to flooding. The reason is that each node rebroadcasts if it can cover only one additional node. Thus we define a rebroadcast threshold which counts the number of additional nodes. If the additional nodes it can cover exceed rebroadcast threshold, it rebroadcasts the packet; otherwise, the packet refrains from rebroadcasting.

2.5.3 Simulation results

In order to test the impact of neighbor error rate on the broadcast performance, we use static network topology. The errors come from two situations. First, node A locates in node B's transmission, but node A does not aware of it. Second, node A does not locate node B's transmission range, but node B think it is. Different degree of error rate was simulated to observe the impact on performance. The metric used is coverage, which mean the ratio that the nodes in the network receive the same broadcast packet.

In Figure 5, we simulate two kinds of topology, 5*5 and 7*7. In 5*5, we can observe that the impacts of error rate are slighter, because the network topology is denser. Wrong decisions do not make a great influence on coverage. While in 7*7, the network is sparser. Wrong decisions are more likely to disconnect the network, as we can observe the steep slope in Figure 5.

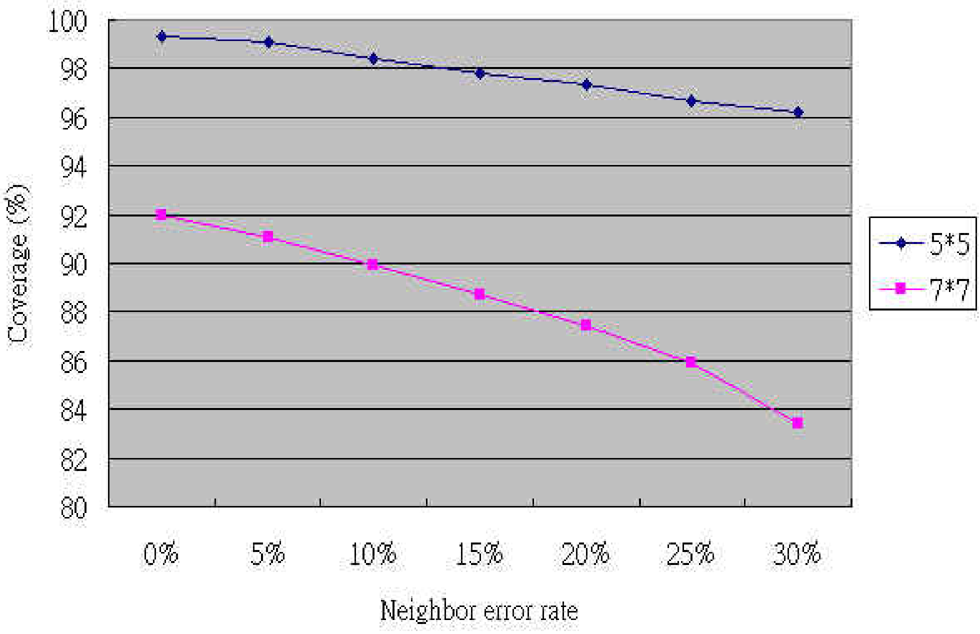

Figure 5. The impact of neighbor error rate on coverage.

Pervious discussions demonstrate the influence of neighbor error rate on the coverage performance. We can now come up with the main target – the relationship of the coverage and the time period. As we can observe in Figure 6, there is a mapping between the error rate and time period. Compared with Figure 5, we can obtain an hello period if the coverage performance is required. For example, if we want coverage performance at least 90% under 7*7 map, the neighbor error rate should be about 10%. If the speed interval is 0 to 15 m/s, we can infer that the time period 5 may meet the requirement.

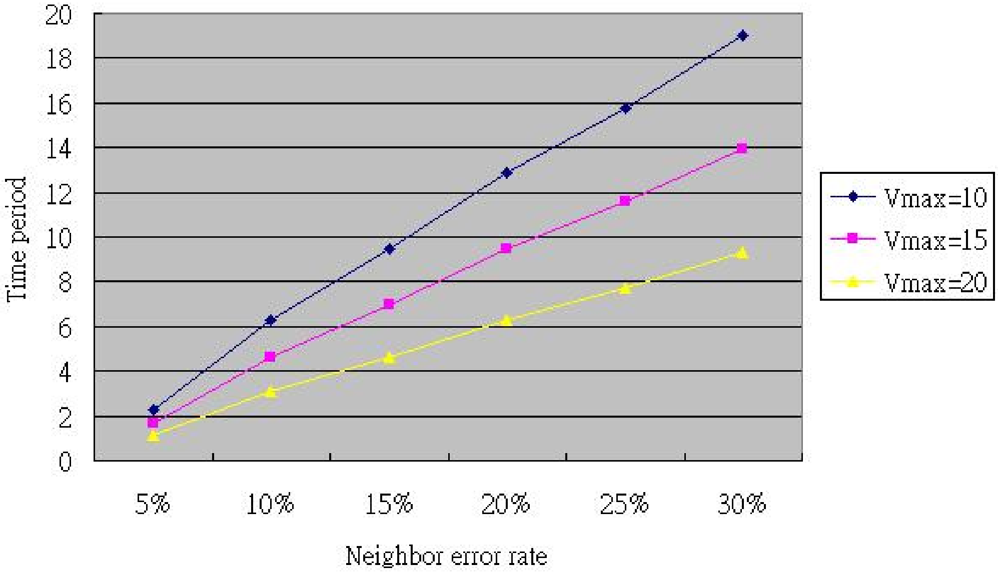

Figure 6. The error rate and the time period

2.6 The theoretical value of hello packets

In this section, we study the theoretical period of hello packets. In [22], P. Samar et. al proposed the largest update period such that the expected delay between the detection of a link change and the next update is small enough. The expected value of the delay is calculated as follows

$$E(t)=\frac{T}{1-e^{-\lambda T}}-\frac{1}{\lambda}\leq\alpha \tag{4}$$

T denotes the update period (hello period), and λ denotes the link change rate which is the average link breakage and arrival per second. α denotes the QoS parameter which can be defined by network administrator. In our model mentioned before (equation (3)), the probability of a link break (P_b)can be calculated (0.032 in our case). Since the expected link arrival rate and link breakage are the same [22], the link change rate can be calculated as follows:

$$\lambda=P_b\cdot\rho\cdot 2 \tag{5}$$

For example, in 5x5 and 7x7 topology (ρ equal 6.4 and 12.8 respectively), the link change rates are 0.4/s and 0.8/s respectively. According to (4) and (5), we can have the relation between delay (α) and link change rate (λ) and hello period (T). We show the relation in Figure 7.

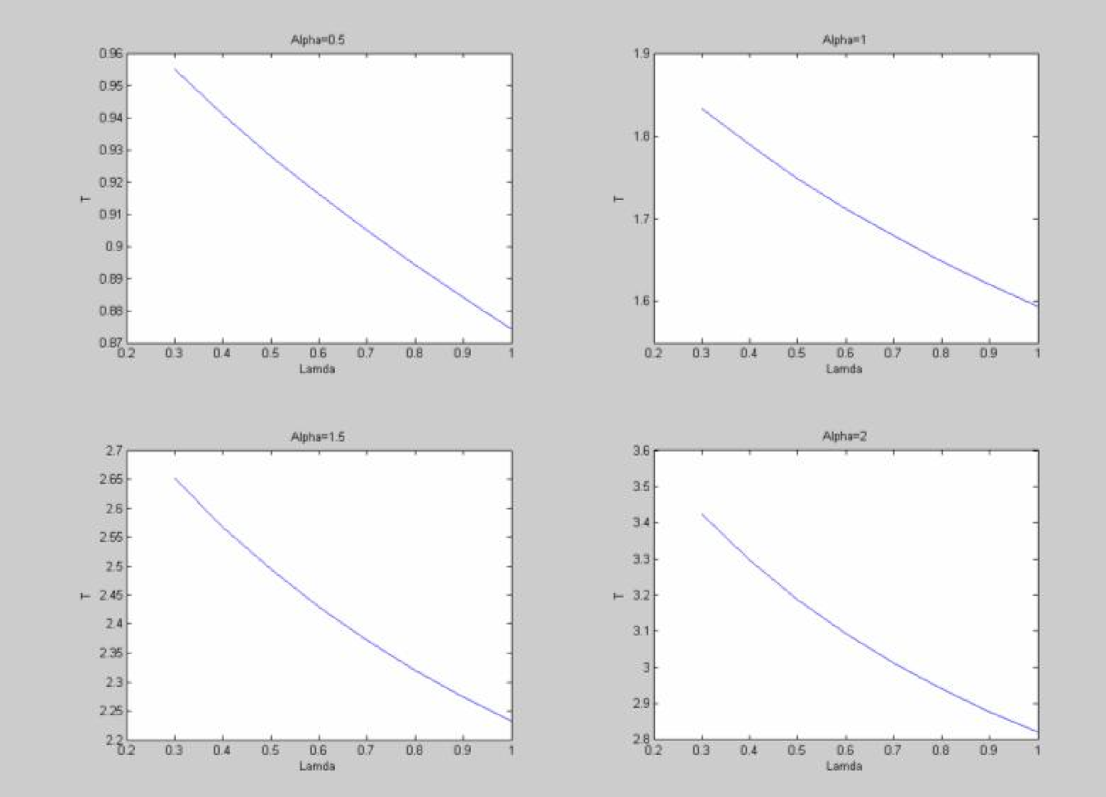

Figure 7. The theoretical hello period under different α and λ.

Chapter 3: The reduction of hello packets

3.1 Introduction

Network topology is the basis of many applications. Well-known applications such as broadcast and route discovery all depend on the accurate topology information. Topology information discusses the connection relationship between nodes. This relationship determines the shape of the network topology. In the static network, the topology information can be gathered by exchanging neighbor lists, and the correctness of topology information can last until network terminated. In Mobile Ad Hoc Networks (MANETs), nodes are mobile so that the topology information varies from time to time. In MANETs, if two nodes can’t communicate directly, the intermediate nodes are needed to relay packets. Since the resources such as energy and channel in wireless environment are very limited, it is very important to make precise relay choices. Otherwise, unnecessary relays will cause additional cost and shorten the network lifetime. Good relay choices rely on precise neighbor information. Broadcast in MANETs is one of the applications that needs precise relay choice; otherwise, the broadcast storm problem [1], which leads to excess rebroadcasts, may occur. Many schemes have been proposed to settle broadcast storm problem. These schemes can be classified into neighbor knowledge schemes [2] [7] [8] and non-neighbor knowledge schemes [1] [3]. In general, neighbor knowledge schemes perform better than non-neighbor knowledge schemes in terms of rebroadcast numbers.

Obviously, better relay choices can be made if nodes are aware of network topology. Among all neighbor knowledge schemes, 1-hop neighbor list is the most fundamental and effective [2]. Traditionally, 1-hop neighbor lists can be obtained by sending hello packets. In order to adapt to dynamic environment of MANETs, hello packets should be sent periodically. If the speeds of mobile nodes are high, the period should be shortened in order to maintain high accuracy of topology information. Thus, the hello packets grow significantly with a shorter hello period. It has been assumed that the overheads derived from hello packets can be ignored because their packet size is small compared with those of broadcast data packets. In this paper, we first demonstrate through simulation that the overheads caused by hello packets actually can not be ignored while broadcast in MANET. We measure the overhead by counting the energy consumed by all the packets transmitted and received in self-pruning broadcast scheme. The simulation results indicate that the energy consumption caused by hello packets accounts for 64% of the total energy consumption when the number of broadcast requests is 50, and when the number of broadcast requests is 400, the hello packets still consumes 17% of the total energy. We may reasonably conclude that the overheads derived from hello packets are indispensable.

Then we propose a method to reduce the overheads greatly. Intuitively, the most effective way is to reduce the number of hello packets directly, thus saves energy and reduces channel contenders. Authors in [2] mentioned that the information of a hello packet can piggyback in a data packet. As a result, no matter what kind of packet a node sends, the receiver could have the same knowledge of information as delivered by hello packets, such as source node, forwarder node, signal strength, etc. More specifically, data packets can lengthen the period of hello packets, because the amount of the needed hello packets is fewer to maintain accuracy. In this paper we derive a practical way to calculate a new effective hello period from the number of broadcast requests. We will further quantify this effect by simulation. The proposed method can reduce the number of hello packets while keeping high accuracy of neighbor information.

The rest of the paper is organized as follows. Section Ⅱ describes the overhead of hello packets. Section Ⅲ shows how to reduce hello packets. Section Ⅳ shows the simulation environment and results. A few concluding remarks are given in section Ⅴ.

3.2 Overheads of hello packets

In this section, our simulation results show that the overheads of hello packets are indispensable from the view of energy consumption and collision incurred by hello packets alone. In our simulation, we assume that the node number is 100, and simulation time is 200 seconds. We use the default energy model in ns2 simulator. The details of the simulation environment are listed in Table 1. In Figure 1, energy consumptions are shown by a bar chart, and collision numbers are shown by a line chart. The X-axis represents the size of hello packet. The energy consumptions are proportional to the size of hello packets. We compare three different hello periods. We can observe that the energy consumption increases as the length of hello period decreases. The collision metric is the average number of collisions per transmitted packets. We can observe that when the length of hello period is shorter, the number of collisions is lower. This is because the total number of transmitted packets is much more in short hello period case. However, short hello period still causes the largest number of actual collisions. There are two issues which should be noticed. First, the figure only shows the influences of hello packets, when data packets are also considered the energy consumptions and collision numbers will further increase. Second, the figure shows the absolute value of energy consumption and collision number. We will show the relative results in Figure 2.

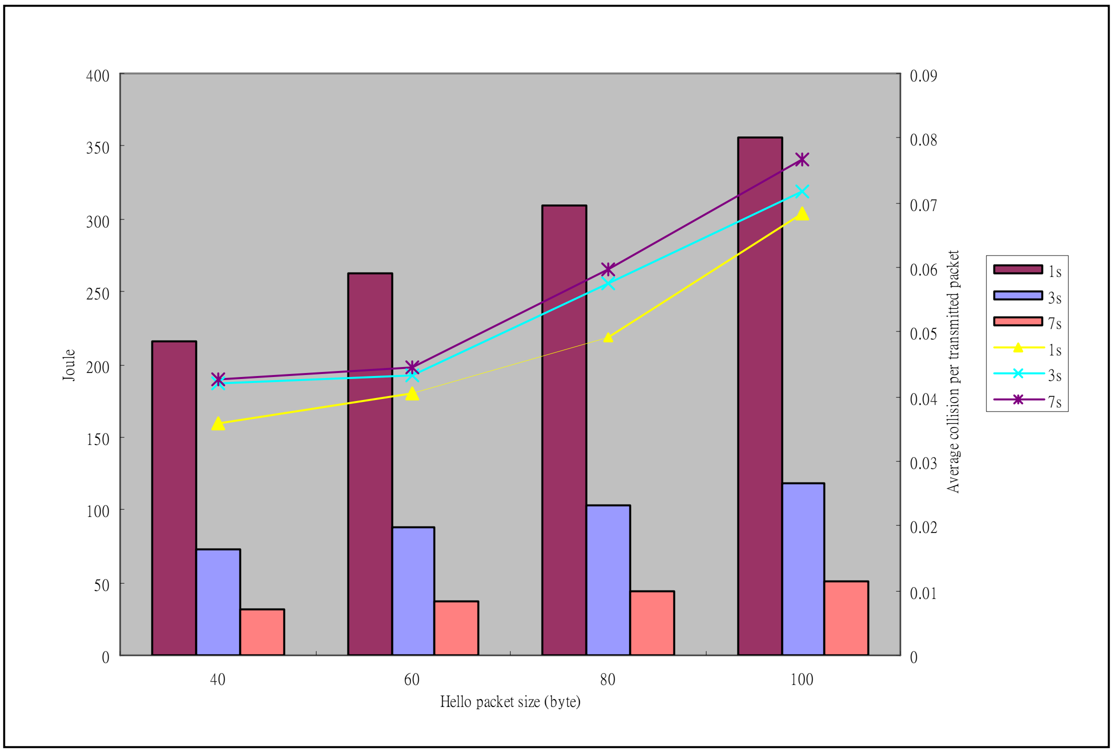

Figure 1. Energy consumption and collision of hello packets

In Figure 2, we want to show that ignoring hello packet overheads are not realistic in the real world. The original objective to address the broadcast storm problem is to reduce the number of rebroadcasts. Thus, it is important to know clearly whether the overheads of neighbor knowledge broadcast schemes outperform non-neighbor knowledge broadcast schemes. However, the total costs of a neighbor knowledge broadcast scheme should include the overheads stem from both hello packets. In Figure 2, we can observe the relationship between energy consumptions, number of collisions, and number of broadcast requests. We fix the size of the hello packet to 40 bytes and the hello and data packet size ratio to 1/8. We can observe that when request numbers are 50 and 100, the total energy consumption including hello packets for a neighbor knowledge broadcast scheme (self-pruning) are even higher than flooding which is the worst broadcast scheme in term of energy consumption. Even if when the number of broadcast requests is large, the saving is still not commendable, as we can observe that there is just about 11% energy saving compared with flooding when number of broadcast requests is 400. Compared with other effective non-neighbor knowledge scheme, the saving can be quite slight or even none. Take 40 byte hello packets and 50 broadcast requests as an example; hello packets consumed 64% of total energy. Even when number of broadcast requests is 400, it still consumed 17% of the total energy. From the simulation results, we can conclude that the overheads of hello packets should not be overlooked. However, if we can reduce the number of hello packets while keeping high accurate neighbor information, we can reduce total energy consumptions and collisions. We achieve this objective in the next section.

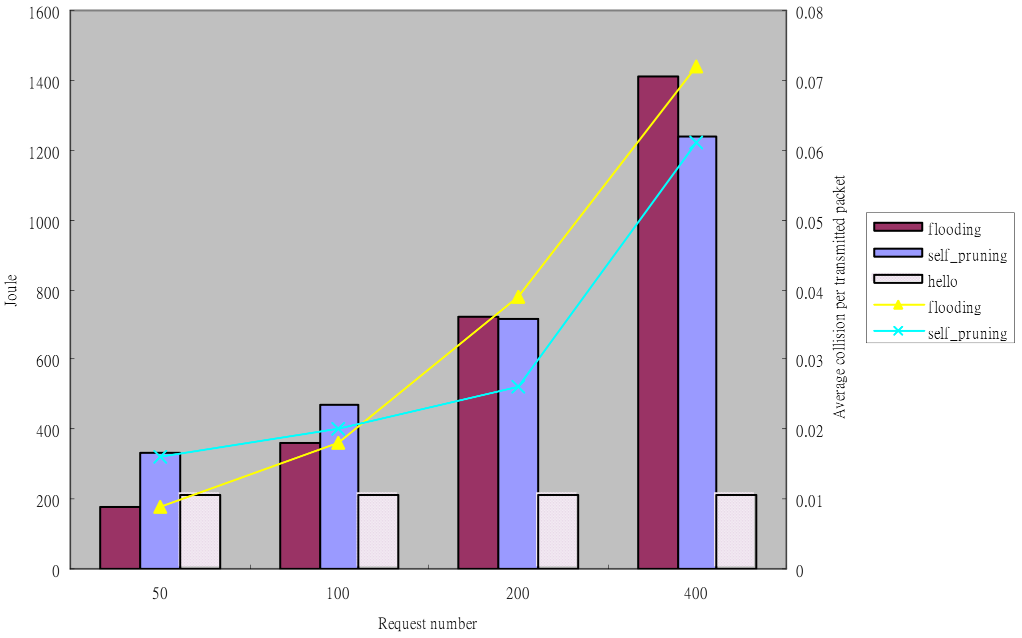

Figure 2. Energy consumption and collision for flooding and self pruning

3.3 Reduction of hello packets

The objective of this research is to reduce the overheads caused by hello packets without sacrificing the performance. Authors in [2] mentioned the hello packets can be piggybacked in data packets; however, the actual effect of the piggybacked hello packets is not quantified. In this paper, we analyze the saving of hello packets by using piggybacked hello packets which are defined as the broadcast request packets carries the neighbor information for the hello packets. We define an effective hello period that takes the piggybacked hello packets into account. We quantified the effect of piggybacked hello packet, and then a new effective hello period (longer than original hello period) can be derived.

Let us start with the replacement of hello packets under flooding case. The number of broadcast request packets in flooding can be calculated since each node in the network relays a new received broadcast request exactly once. Therefore, the maximum number of piggyback hello packets can be calculated as total number of broadcast requests multiply number of nodes as in Equation (1)

Number of packets can be used to piggyback hello packets =

$$\text{average number of requests} * \text{time period} * \text{node number} \tag{1}$$

Given a hello period and number of node, the total number of hello packet within a time period can be calculated.

$$\text{Original hello packets} = \frac{\text{time period}}{\text{hello period}} * \text{node number} \tag{2}$$

Therefore, in order to maintain same number of hello packets, we are still lacking

$$\text{Insufficient hello packets} = \text{original hello packets} - \text{piggybacked hello packets} \qquad (3)$$

Finally, we can derive a new effective hello period by evenly distribute the responsibility to deliver insufficient hello packets to all nodes within a time period.

$$\text{Effective hello period} = \frac{\text{time period} * \text{node number}}{\text{Insufficient hello packets}} \qquad (4)$$

Instead of original hello period, we now send the hello packets periodically according to the effective hello period. We use an example to describe the above formulas. Consider that there are 100 nodes with 1 second hello period and 100 broadcast requests in 200 seconds time period. The insufficient hello packets drop from 20,000 to 10,000. That is, the hello period is doubled. According to the analysis, we can have a snapshot of how many hello packets can be saved, as shown in Figure 3. In Figure 3, X-axis is the number of broadcast requests. Given 100 nodes, the number of hello packets for three different hello periods are compared in 200 second time period by using effective hello period. When the total number of broadcast requests is 100 and the length of hello period is 1 second, half of the hello packets can be saved by effective hello period. Notice that in this example when the hello period equals to 7 and number of request is greater than 25, there are some cases that the number of hello packets transmitted per second is equal to 0 because all the hello packets can be piggybacked by the broadcast packets according to our effective hello period calculation. However, in the real networks, if there exist some nodes do not initiate any broadcast requests or do not be asked to rebroadcast for other nodes; these nodes may be invisible to the whole network, therefore the coverage performance could be degraded. Thus, even the number of rebroadcast data packet exceeds the expected hello packet number or the effective hello packet period is too long, network nodes should still have a chance to inform the network of their presence in time. For that reason, the effective hello period should have a bound.

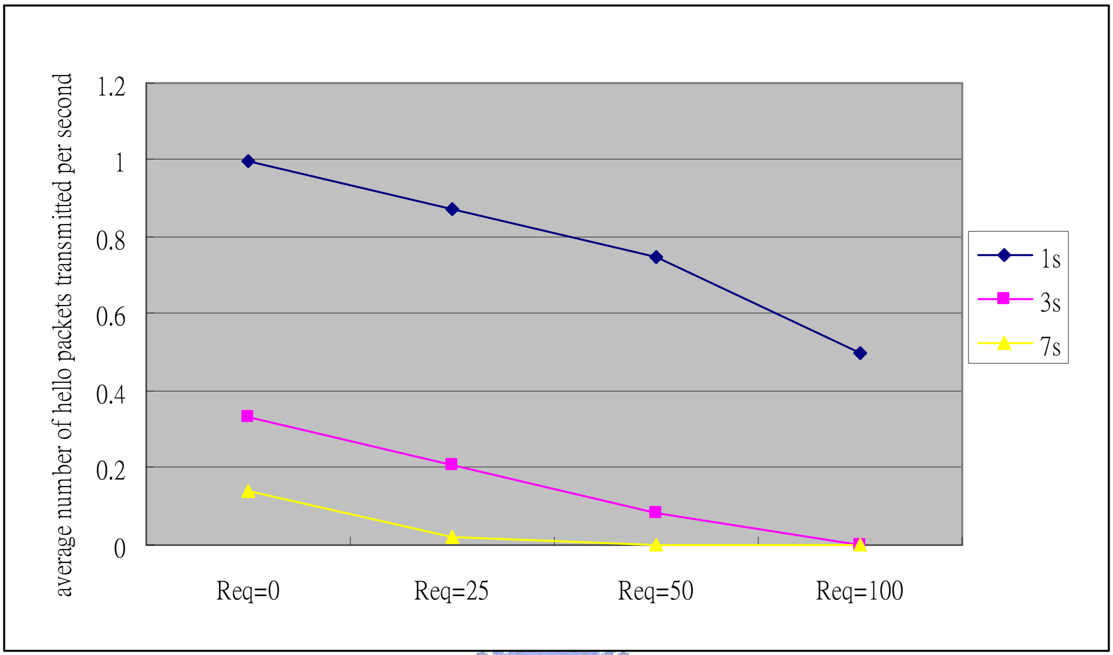

Figure 3. Transmitted hello packets vs. different broadcast requests under flooding

Since 1-hop neighbor list is our main concern, a node should issue a notification after each transition (a transition means a node moves from one location to another location under random waypoint mobility model). Thus, we define the average of a transition time as an upper bound for a node should at least issue a hello packet. From equation (1), we can obtain the relative speed V_r. Authors in [10] analyze the stochastic properties of random waypoint model. The expected length of one transition on a circular area of radius R is 0.9054R. As a result, we can have a bound of effective hello period, which equals 0.9054R / Vr under random waypoint mobile model.

Since flooding does not need neighbor information, we should further invest a broadcast scheme which relies on hello packets to obtain accurate neighbor information. We use a neighbor knowledge broadcast scheme called self-pruning [2]. In self-pruning, each network node adds its 1-hop neighbor list to the headers of its broadcast packets. When a node receives a broadcast request packet, it compares its own neighbor list to the neighbor list contained in the packet header. If the number of different neighbor nodes exceeds a predefined rebroadcast threshold, the node will rebroadcast; otherwise, it will drop the request. The value of threshold determines the additional cover of a node. As a result, accurate 1-hop neighbor information is needed for the function of self-pruning broadcast scheme.

Since the objective of an effective broadcast scheme should reach the best coverage using smallest number of rebroadcasts. Accordingly, the value of the threshold should also conform to this goal. Therefore, we define an Effective Coverage and Rebroadcast Ratio (ECRR), as a ratio of coverage performance over number of rebroadcasts under a reasonable coverage performance (e.g. > 95% in 5x5 maps). Figure 5 and 6 show the coverage performance and number of rebroadcast of self-pruning as compared with blind flooding under different rebroadcast thresholds in different size of networks. From these figures, self-pruning gives best ECRR when the rebroadcast threshold is 7 and 4 in 5x5 and 7x7 maps respectively. Thus, the number of rebroadcast packets that can be used to piggyback hello packets in self-pruning is only 56% of pure flooding in 5x5 and 63% in 7x7 respectively. As a result, we can recalculate the number of broadcast data packets that can be used to piggyback hello packet under self-pruning are 53% and 63% of broadcast data packets in flooding when network size is 5x5 and 7x7 respectively.

We also use an example to describe this variation. Consider 100 nodes with 3 seconds hello period and 50 broadcast requests in 200 seconds simulation time in 7x7 map. The original number of hello packets is 6700. The number of data packets that can be used to piggyback hello packets is 50*100*63% which equals to 3150. Thus, the effective hello period can be calculated as 200*100 / (6700-3150), which equals 5.6 seconds.

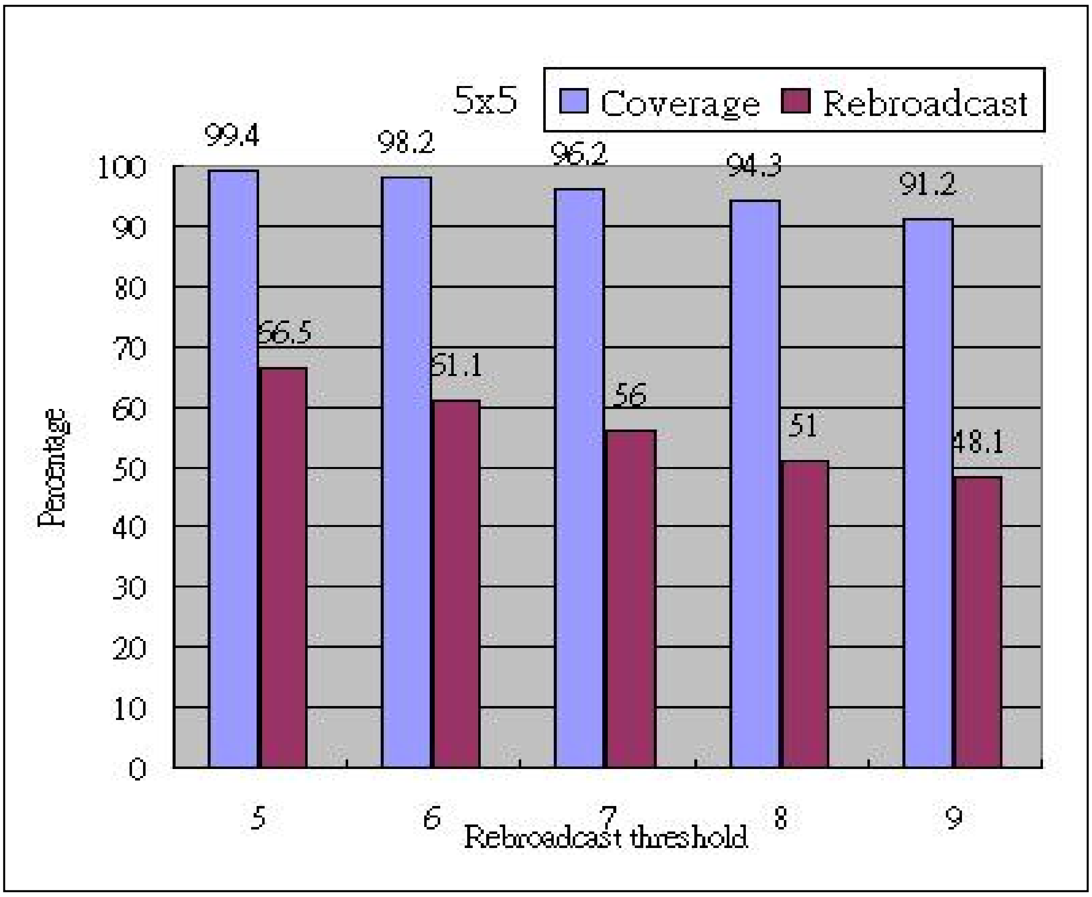

Figure 5. Performance of self-pruning under different rebroadcast threshold in 5x5 map

According to this adjustment, we can also have a snapshot of how many hello packets can be saved in self pruning. As shown in Figure 7, the saving of hello packet in self-pruning is less than the saving in flooding using effective hello period, since number of packets can be used to piggyback hello packets in self-pruning is less Notice that the average number of hello packets per second are always have values greater than 0, since we adopt a bound for effective hello period as 18 seconds in this case (mobility speed is between 0 and 20 m/s).

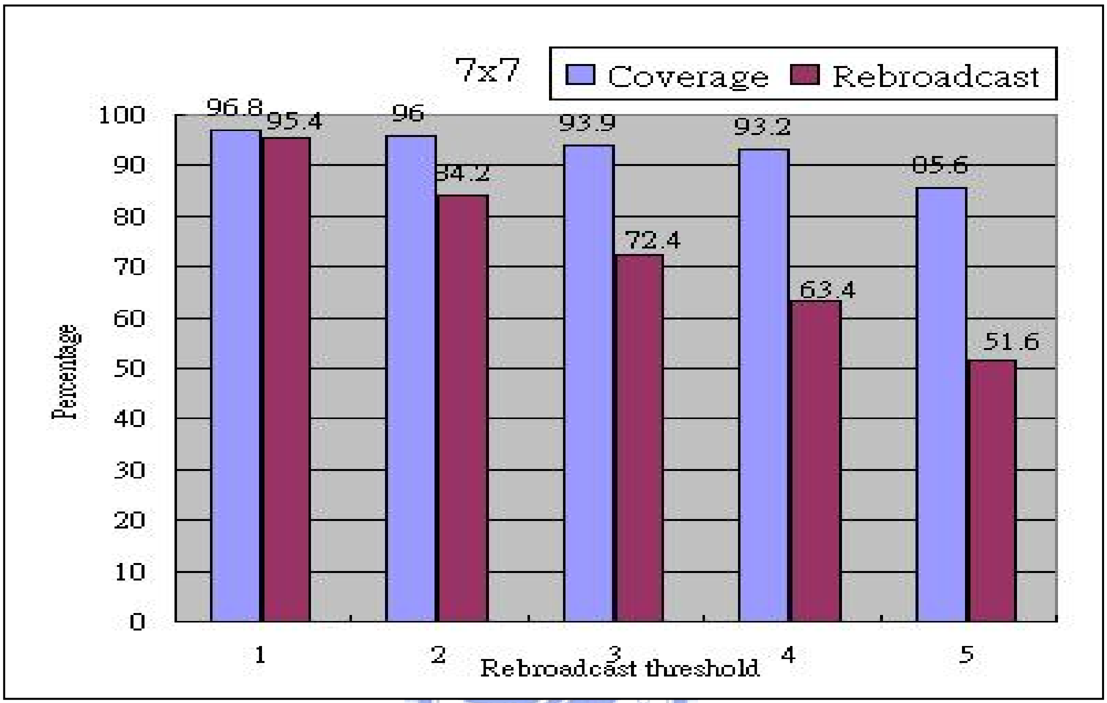

Figure 6. Performance of self-pruning under different rebroadcast threshold in 7x7 map

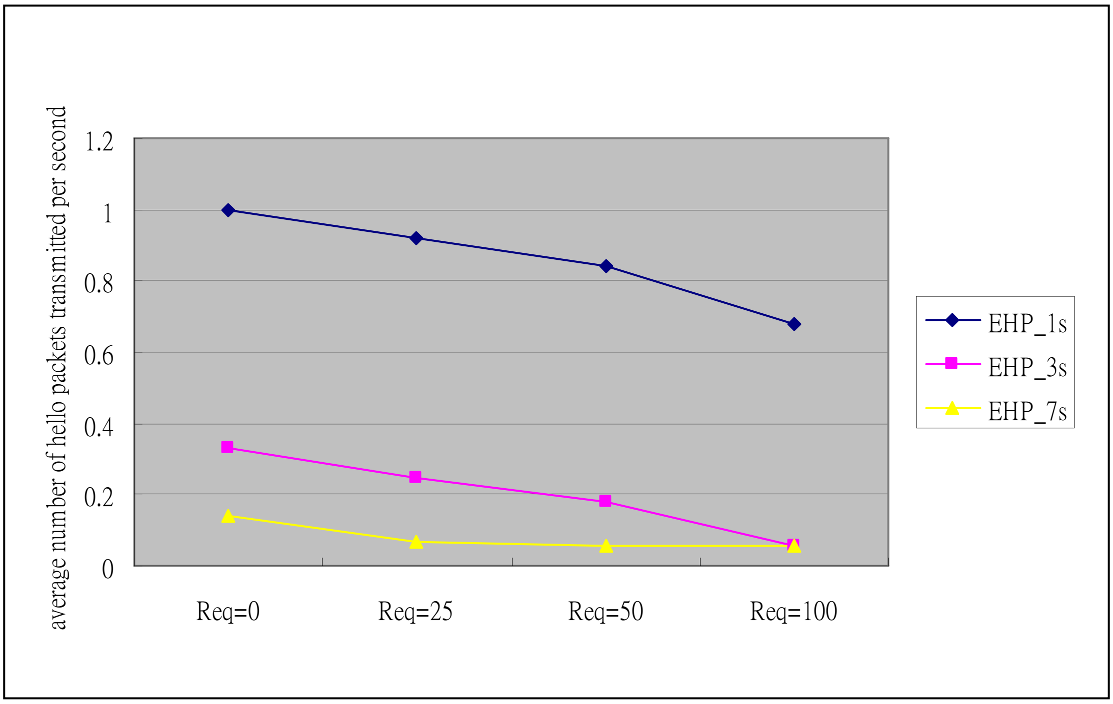

Figure 7. Transmitted hello packets vs. different broadcast requests under self-pruning

3.4 Simulation Environment and results

We use NS-2 [9], an event-driven simulator, as our simulation tool. The simulation network contained 100 nodes placed randomly in a map of L×L units, where a unit is the length of communication radius set to 250 meters. The random waypoint model for mobility patterns was adopted. The topologies were generated randomly by the “setdest” program supported by NS-2. The moving speed was randomly distributed from 0 to 20 (m/s), and the pause time was set to 0. The MAC layer was constructed using the IEEE 802.11 standard, which is implemented in NS-2. Broadcast data packets are gathered from the broadcast requests in 200 second simulation time. Hello packets are sent periodically during the entire simulation period. The simulation results were averaged by the results of 15 simulation runs. The detailed parameters are summarized in Table I.

Table1. Simulation parameters

Simulation parameter	**Value**
Simulator	Ns2(2.27)
Node number	100
Network range	5*5, 7*7
Simulation time	200s
Mobility Speed	0-20m/s
Transmit power(W)	0.665
Receive power(W)	0.395
Idle power(W)	0

We consider the following performance metrics：

· Coverage – the percentage of nodes which can be covered as compared with blind flooding

- Energy consumption – the total energy consumed by hello and broadcast packets in Joule.
- Collision – the average number of collisions per transmitted packet
- Hello packet number – total number of hello packets generated during entire simulation period

To achieve more saving in the hello packets than effective hello period (EHP), we further enhance EHP by sending hello packets only if no broadcast data packets transmitted during the effective hello period instead of sending hello packet at end of each effective hello period, We refer to it as enhanced effective hello period (E_EHP) in the following simulations. We simulate 1 second, 3 second, and 7 second of hello periods with 25, 50, and 100 broadcast requests. The effective hello periods are summarized in Table II.

Table 2. The effective period used in simulation

	effective_period
1s req=25	1.09
req=50	1.19
req=100	1.46
3s req=25	3.9
req=50	5.6
req=100	18
7s req=25	15
req=50	18
req=100	18

Figures 8, 9, and 10 illustrate the coverage and total number of hello packets under different broadcast requests with hello periods equal 1, 3 and 7 seconds. The bar charts show the coverage performance, while the line charts indicate the percentage of hello packet number used by EHP and E_EHP compared with original hello period. Periodical hello sent fixed number of hello packets, the actual numbers are 20000 for 1 second, 6700 for 3 second and 2900 for 7 second during 200 second simulation time. Compared with EHP and E_EHP, it has the most accurate 1-hop neighbor list, so the coverage performance is the best, as we can observe in the first bar in these figures. The EHP we proposed performs compatible with periodical hello. There is only about 2% degradation in coverage performance. However, compared with total savings in hello packets, in Figure 8, when the number of requests number is 25, the EHP can only save about 8% of hello packets, and this is because the piggybacked hello packets are fewer. However, when number of requests is 100, the saving can reach 32%. In Figure 10, the E_EHP can save even more hello packets as 88%, 93% and 96% with only about 5% degradation of coverage performance.

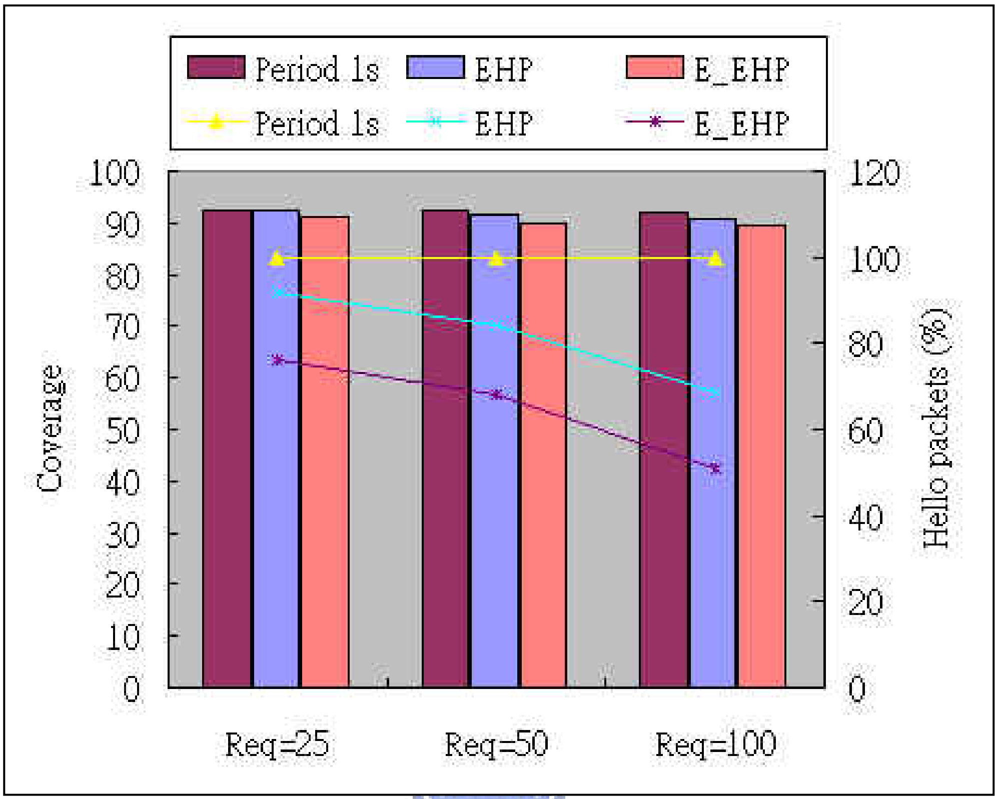

Figure 8. Coverage and hello number on 1s period

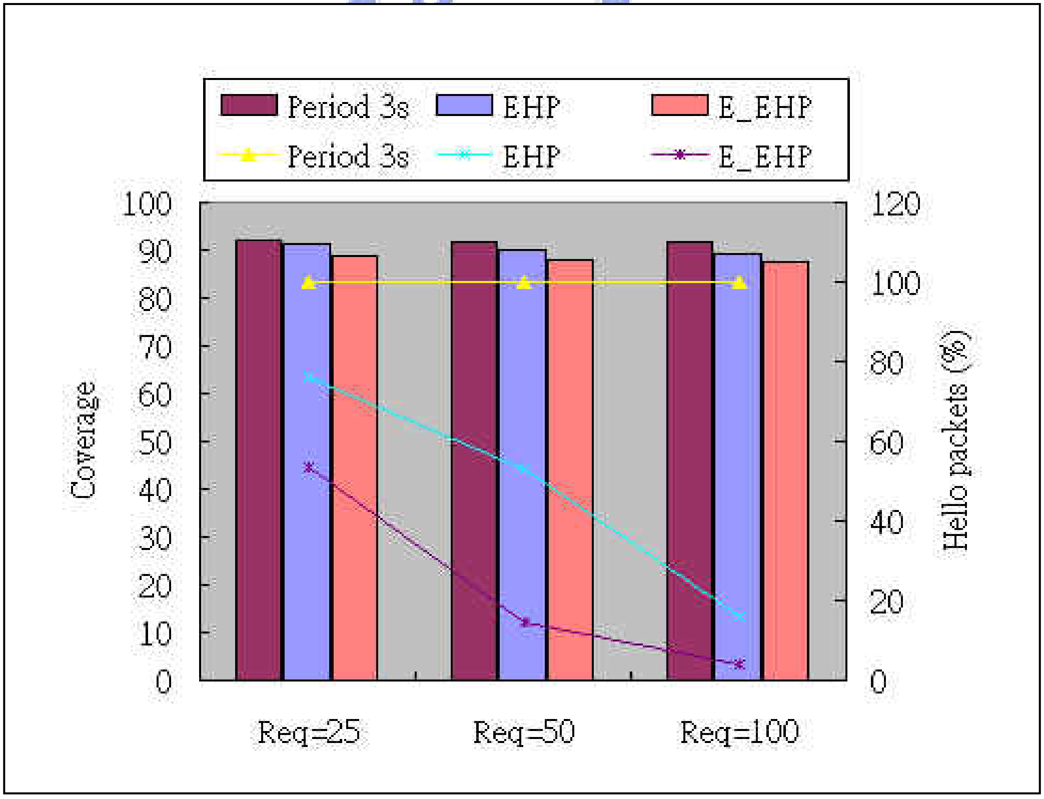

Figure 9. Coverage and hello number on 3s period

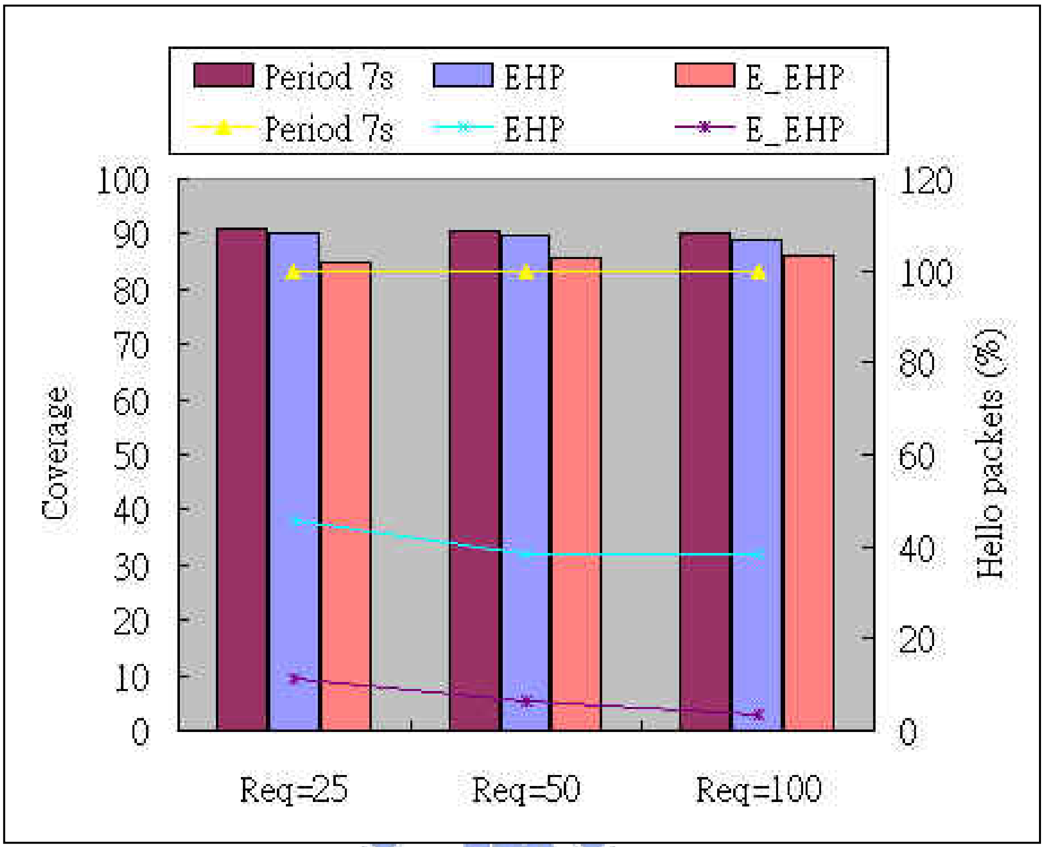

Figure 10. Coverage and hello number on 7s period

Figures 11, 12, and 13 illustrate energy consumption and number of collision under different hello/data packet size ratios. The bar charts show the energy consumption, while the line charts indicate the collision number. The number of broadcast requests is fixed to 50. We change the hello and data packet size ratio by fixing the size of hello packet and adjusting size of broadcast data packet. The size of hello packet is 40 bytes and the ratios of hello and data packet size are 1/2, 1/4, and 1/8 respectively. When the ratio decreases, (the data packet size increases) the energy consumption increases. Since the objective of our effective hello period is to reduce the number of hello packets, the total amount of energy consumption will decrease as well. In Figure 11, when the broadcast requests is 25, since the reduction of hello packets is fewer, the saving of energy is lower. With the growing of the number of broadcast requests, the savings incline. Figures 12 and 13 indicate similar saving trend.

The reduction of hello packet also influences the number of collisions directly, since the contenders for the communication channel are less. When the ratios of hello and data packet size become smaller (the size of data packet become larger), the transmission time of data packets becomes longer. Therefore, the collision probability rises. Hence, as shown in Figure 11, 12 and 13, the number of collision tends to increase with the growth of data packet size. However, the average collision per transmitted packet is lower when the period is shorter. This is because of total number of transmitted packets are much more in short hello period case. However, short hello period still causes the largest number of actual collisions.

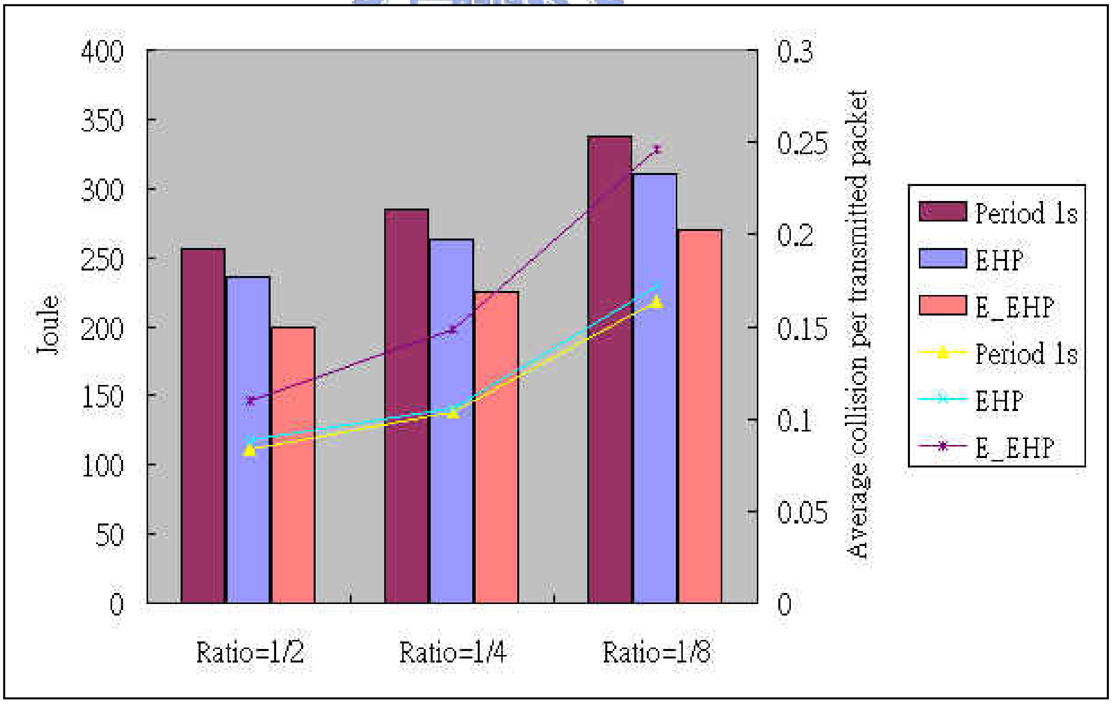

Figure 11. Energy and collision performance on 1s period

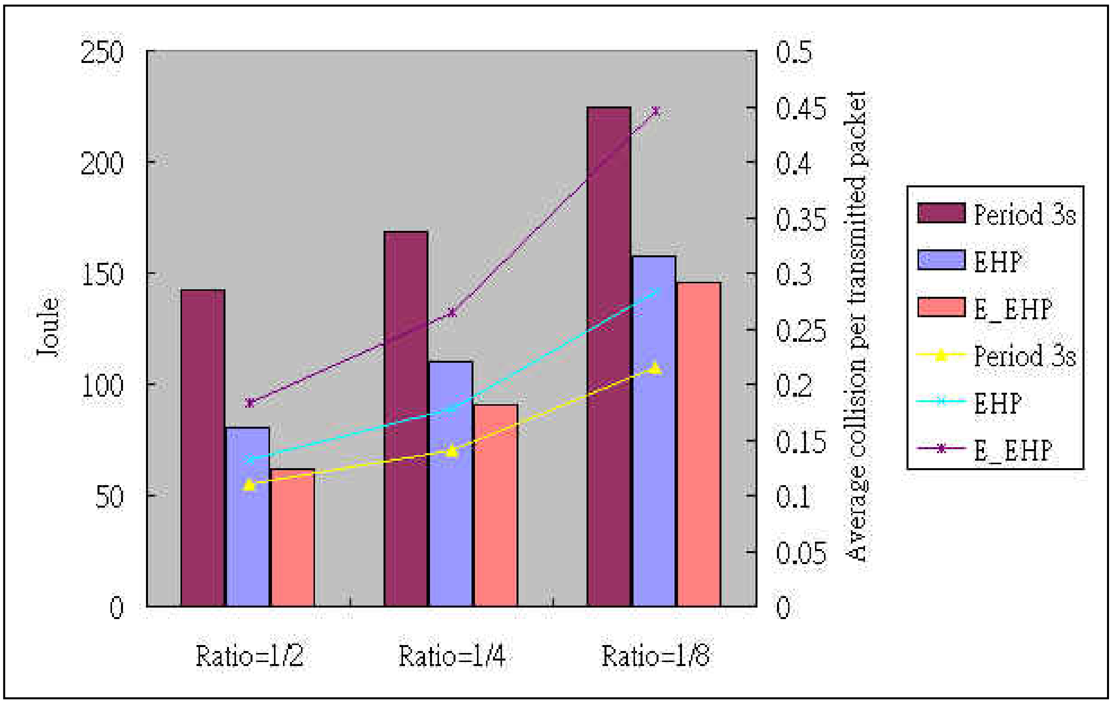

Figure 12. Energy and collision performance on 3s period

Figure 14 and 15 shows the energy consumption of effective hello period under different broadcast requests in 5x5 and 7x7 maps. We can observe that the energy consumptions are less in EFP and E_EFP compared with flooding in most cases. Only under such a circumstance the use of neighbor knowledge broadcast schemes do make sense. Notice that the energy consumption in 5x5 is larger than in 7x7. This is because the average number of neighbors in 5x5 map are more than in 7x7, and thus 5x5 map consumes more receiving power.

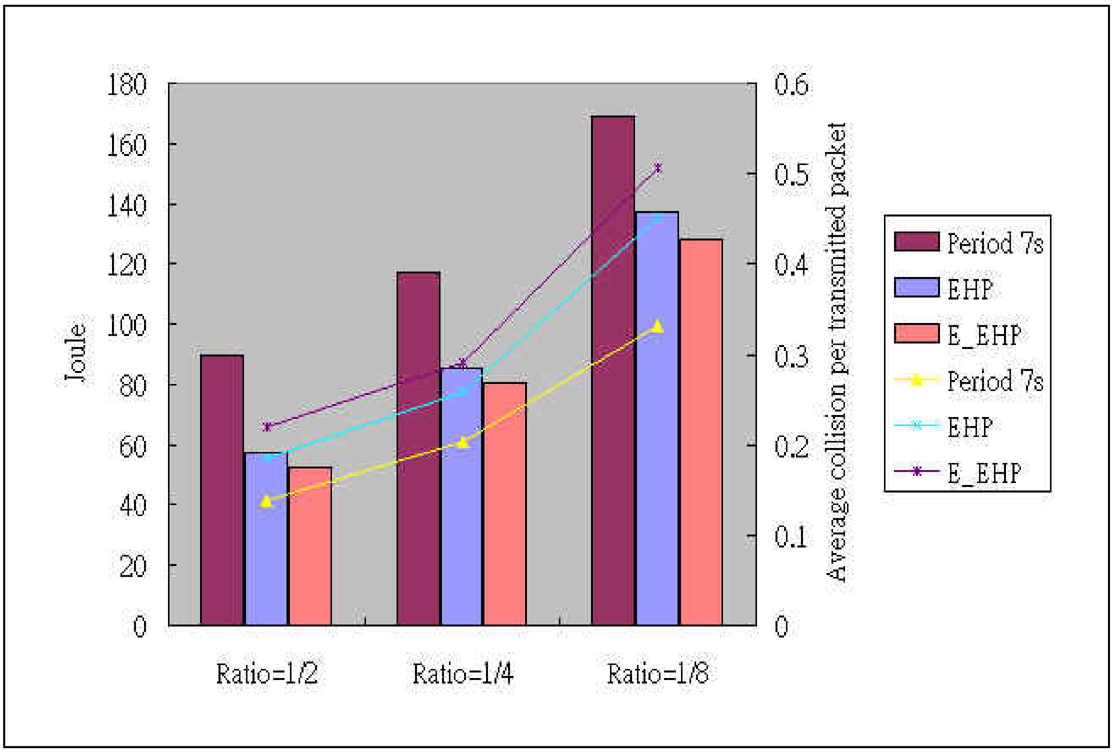

Figure 13. Energy and collision performance on 7s period

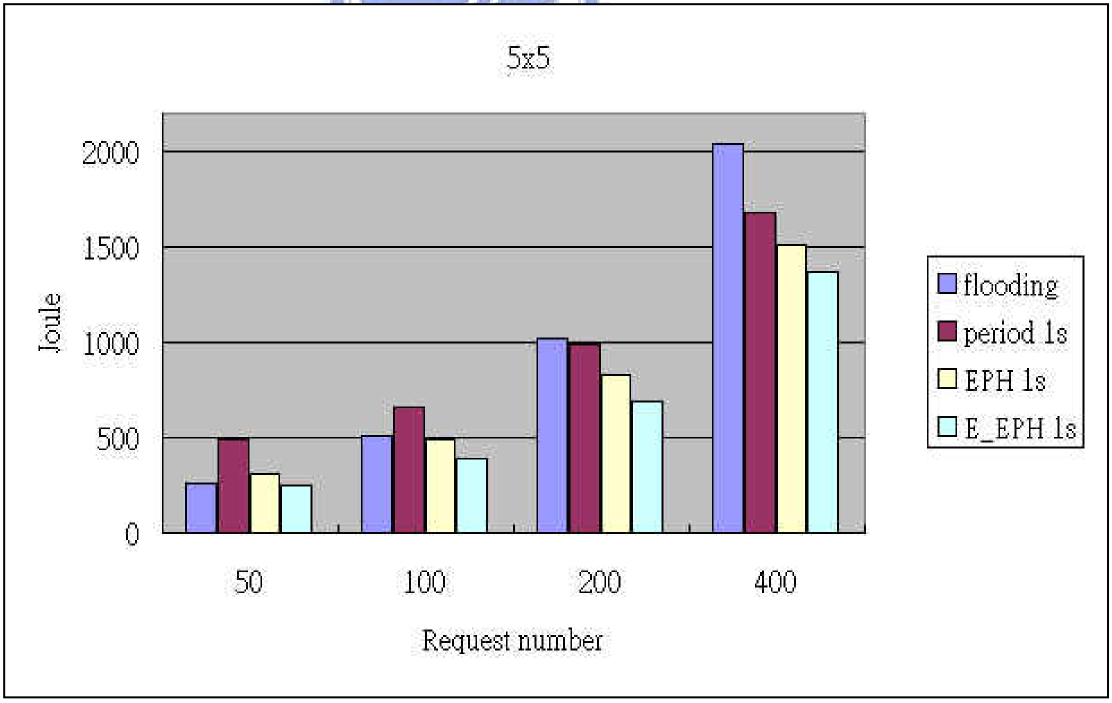

Figure 14. Comparison of energy consumption on different broadcast requests under 5x5 map

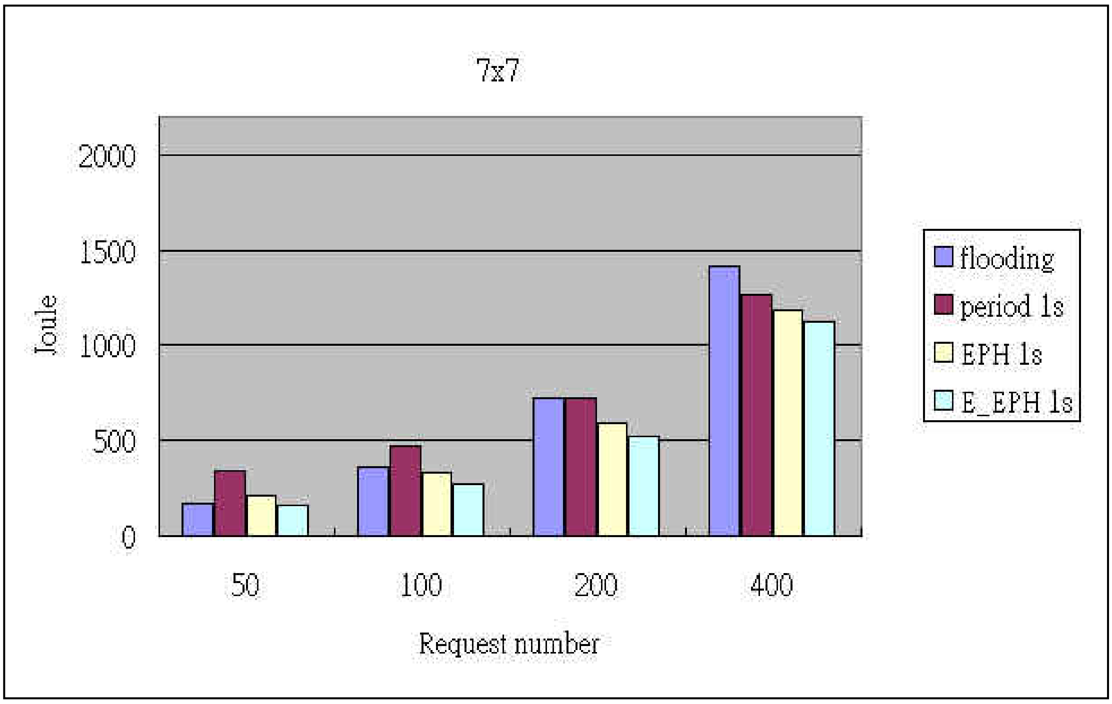

Figure 15. Comparison of energy consumption on different broadcast requests under 7x7 map

Chapter4: Conclusion

In the first part, we analyze the probability that a node leaves its original transmission range. According to this model, we can deduce the link change rate that the variation of the neighbor of a node. Lastly, we discuss the proper transmission period of hello packets under certain QoS constraint.

Second, we analyzed the most frequently used topology information—1-hop neighbor list in wireless MANETs broadcast schemes. This topology information can be obtained by periodical hello packets. Many researchers consider the overhead derived from hello packets is negligible, since the size of a hello packet is small. We believe that it is certainly the most popular delusion about hello packets. It results in abuse usage of hello packets. This situation leads to extra energy consumption and collisions. We showed that the overhead can be significant through our quantification process. We further proposed the effective hello period to reduce the hello packet overheads. From the simulation results, we can show that our proposed method can reduce the overhead notably while still maintaining high coverage performance. We believe this analysis is likely to be essential for a real world MANET implementation.

Reference:

[1] Y-C Tseng, S-Y Ni, Y-S Chen, J-P Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network", *Wireless Networks*, Volume 8 Issue 2/3, March 2002

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

[3] C. Zhu, M. J. Lee and T. Saadawi, "A border-aware Broadcast Scheme for Wireless Ad Hoc Network", *IEEE Consumer Communications & Networking 2004*, Las Vegas, January 2004

[4] Y-C Tseng, S-Y Ni and E-Y Shin,"Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network", *IEEE Transaction on Computers*, Vol.52, NO.5, May 2003

[5] B. Williams and T. Camp,"Comparison of broadcasting techniques for mobile ad hoc networks", *in Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp.194-205, 2002, Available: citeseer.nj.nec.com/williams02comparison.html

[6] J. Broch, D. B. Johnson and D. A. Maltz, "The dynamic source routing protocol for mobile ad hoc networks", *IETF Internet draft*, draft-ietf-manet-dsr-03.txt, 22 October 1999.

[7] W. Peng and X. Lu." On the reduction of broadcast redundancy in mobile ad hoc networks", In Proc. *First annual workshop on Mobile and Ad Hoc Networking and Computing*, pp.129-130, August 2000.

[8] W. Peng and X. Lu. "AHBP: An efficient broadcast protocol for mobile ad hoc networks", *Journal of Science and Technology*", Beijing, China 2002.

[9] K. Fall and K. Varadhan, ”ns Notes and Documents,” The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2000, Available at http://www.isi.edu/nsnam/ns/ns-documentation.html.

[10] C. Bettstetter, H. Hartenstein, X. Pérez-Costa, “Stochastic Properties of the Random Waypoint Mobility Model”, *Wireless Networks***,** Volume 10 Issue 5, September 2004

[11] D. Yu, H. Li, and G. Ingo, ”Path availability in ad hoc network**”,** *Telecommunications*, 2003. *ICT* 2003. 10th International Conference on Volume 1, 23 Feb.-1 March 2003 pp.383-387 vol.1

[12] D. Yu and H. Li, ” Lower bound on path availability in ad hoc network”, *VTC* 2003-Spring. The 57th IEEE Semiannual Volume 3, 22-25 April 2003 pp.1758-1762 vol.3

[13] S. Jiang, D. He, and J. Rao, “A prediction-based link availability estimation for mobile ad hoc networks”, *INFOCOM* 2001. *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*. Proceedings. IEEE Volume 3, 22-26 April 2001 pp.1745-1752 vol.3

[14] A.B. McDonald and T. Znati, ”A path availability model for wireless ad-hoc etworks”, *WCNC* 1999 IEEE 21-24 Sept. 1999 pp.35-40 vol.1

[15] G. Ingo and H. Li, ” Link expiration times in mobile ad hoc networks”, *Local Computer Networks*, 2002. Proceedings. *LCN* 2002. 27th Annual IEEE Conference on 6-8 Nov. 2002 pp.743-750

[16] P. Samar and, S.B. Wicker, ” Characterizing the communication links of a node in a mobile ad hoc network”, *Communications*, *IEEE International Conference* on Volume 7, 20-24 June 2004 pp.4092–4096 Vol.7, 2004

[17] D. Kim, C-K Toh, C., J.-C. and Manzoni, P., ” A bounding algorithm for the broadcast storm problem in mobile ad hoc networks”, *Wireless Communications*

and Networking, 2003. *WCNC* 2003. 2003 IEEE Volume 2, 16-20 March 2003 pp.1131-1136 vol.2

[18] C. Bettstetter, H. Hartenstein and X. Perez-Costa, ” Mobility, Modeling, and Management: Stochastic properties of the random waypoint mobility model: epoch length, direction distribution, and cell change rate”, *Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile system*, 2002

[19] D. Trajanov, S. Filiposka, M. Efnuseva and A. Grnarov, “Routing & performance modelling: Ad hoc networks connection availability modeling”, *Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous network*, 2004

[20] J. Wu and F. Dai, ” Efficient Broadcasting with Guaranteed Coverage in Mobile Ad Hoc Networks”, *IEEE Transactions on Mobile Computing*, Volume 4, Issue 3, May-June 2005 pp.259-270

[21] F. Dai and J. Wu, “Performance comparison of Broadcast Protocols for Ad Hoc Networks Based on Self-Pruning”, *IEEE Trans. Parallel and Distributed Systems*, vol. 15, no. 11, pp.1027-1040, Nov, 2004

[22] P. Samar and S. Wicker, “On the behavior of communication links of a node in a multi-hop mobile environment”, *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp.145-156, 2004