Performance analysis of broadcast in mobile ad hoc networks with synchronized and non-synchronized reception

Kuang-Hung Pan^{a,*}, Hsiao-Kuang Wu^b, Rung-Ji Shang^c, Feipei Lai^{a,c}

^aDepartment of Electrical Engineering, National Taiwan University, Taipei, Taiwan

^bDepartment of Computer Science and Information Engineering, National Central University, ChungLi, Taiwan

^cDepartment of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

Abstract

Analyzing the performance of broadcast in mobile ad hoc networks is necessary because of the importance of broadcast in multiuser communications and the characteristic difference between wireless communications and wired communications. If the time lag between collided packets is small on the order of a symbol, the reception is *synchronized*; otherwise, the reception is *non-synchronized*. We find that there is a time complexity gap exponential with the degree of the network between the performance of synchronized and non-synchronized reception. Besides, we also take into account the possibility that a processor is busy with other tasks, and we find that allowing the processors to be temporarily busy with other tasks will not degrade the performance significantly. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Network protocols; Performance analysis; Mobile ad hoc networks; Distributed broadcast algorithms

1. Introduction

Broadcast, i.e. sending a message initiated from a user to other users in the network, is an important task in distributed systems and mobile networks. It can be used for disseminating information among a set of receivers [1] and exchanging messages in distributed computing [2]. Applications of broadcast include sending control messages to all users and video conferencing [1].

In single-hop wireless networks with a base station, such as cellular networks, broadcast can be done by sending the message via the base station to every mobile user in the downlink channel [3]. However, in a mobile ad hoc network [4], there is not a base station that can communicate with all users directly, and not all users can receive the messages from all other users directly [2]. Therefore, ad hoc mobile networks belong to multihop networks, in which it is not so simple to solve the broadcast problem. Modern applications of multihop mobile networks are given in Refs. [5–8], such as disaster recovery, e.g. search and rescue in fire or earthquake. Another example [5–8] is ad hoc personal communications networks, which could be rapidly implemented, for example, on a campus, to support collaborative computing and provide access to the Internet during special events like concerts and festivals.

Numerous researches have been conducted to investigate the broadcast problem in multihop wireless network. Chlamtac and Weistein proposed a centralized broadcast algorithm with time complexity $O(R \log^2 V)$, where R is the radius of the network, i.e. the maximum number of edges in the shortest path between two nodes, and V is the number of mobile users in the network [9]. Broadcast protocol based on multi-cluster architecture [10] was proposed in Ref. [11]. In Ref. [12], it was proved that any broadcast algorithm requires $\Omega(\log^2 V)$ time slots to finish for a network with radius equal to two. In Ref. [13], it was proved that any distributed broadcast algorithm requires $\Omega(R \log V)$ time slots to finish. Bar-Yehuda et al. in Ref. [14] proposed a distributed randomized algorithm, which needs only $O((R + \log V/\epsilon) \log V)$ time slots to finish with probability $1 - \epsilon$. This algorithm is exponentially superior to any distributed deterministic algorithm, which takes $\Theta(V)$ time slots even for a radius-3 network. Based on this randomized algorithm, a routing and multiple broadcast algorithm was proposed in Ref. [15]. In Ref. [16], a $\Omega(D \log(V/R))$ lower bound for randomized broadcast algorithm was provided by Kushilevitz and Mansour.

In previous researches such as Refs. [9–16], it was often assumed that the reception of a packet is always successful when there is only one packet received, and it always fails when more than one packet are received at the same time. In a real wireless channel, even if there is only one packet

* Corresponding author.

E-mail address: kpan@orchid.ee.ntu.edu.tw (K.-H. Pan).

Fig. 1. Illustration of the physical layer multipath and propagation delay.

transmitted, it is possible that this packet is not acquired successfully. When there are more than one packet received, it is possible that a packet is acquired successfully [15]. Therefore, in the previous work [17], we analyzed the performance of a distributed randomized broadcast algorithm with consideration of channel reliability. The analysis shows that the performance of randomized broadcast algorithm is satisfactory with channel reliability. Furthermore, adjusting the retransmission probability is a good way to obtain better performance [17].

However, power level is only one of the several important issues in wireless networks. For example, another critical issue is the timing of the signals. Therefore, in this paper, we analyze the performance of broadcast with more considerations than those in Ref. [17]. The impact of the time lag at *symbol (bit) level between collided packets* is analyzed. If the time lag between packets is small, we say the reception is *synchronized*. If the time lag is large, we say the reception is *non-synchronized*. To correctly obtain the information from the signal, the receiver has to know when the signal starts and ends. Therefore, investigating these timing issues will be very important to the performance of wireless networks.

Related research topics appear in the analysis of single-hop networks. The phenomenon that a collision does not necessarily destroy packets is called capture [2,18,19]. Capture could enhance the performance of random access protocol radio such as ALOHA, and capture influences the performance of CDMA (code division multiple access) [19]. The goal of the MAC (medium access control) in those topics is to achieve multiple access. Therefore, the packets involved in a collision have different contents. The broadcast problem here contains only single message; therefore, it is successful as long as at least one packet is received successfully. Thus, the MAC is utilized to carry this packet to all users. Besides, there can be more than one hop in the network, so the packet has to be transmitted hop by hop to reach every user. Therefore, the packets involved in a collision have the same contents, if the packets are broadcast in ordered sequence.

Furthermore, the probability that a processor is so busy with other tasks that temporarily does not execute

the broadcast algorithm is included. This may result from some emergency event that a processor has to deal with, or power-saving considerations, etc. In another point of view, allowing a processor to be busy with other things also gives the processor more freedom since it can deal with other important jobs. Thus, it would be desired to analyze the performance of allowing a processor to be busy with other things and temporarily not carrying out the algorithm.

In this paper we analyze the performance of broadcast with synchronized and non-synchronized reception in mobile ad hoc networks. We examine the performance gap between them. How to adjust the parameters in the broadcast algorithm to achieve better performance is also investigated. The rest of the paper is organized as follows. Section 2 analyzes the probability of successful reception of a packet. Performance of broadcast algorithms is analyzed in Section 3. Section 4 gives numerical examples and discussions. Conclusions are provided in Section 5.

2. Probabilities of successful reception

This paper takes into account the influence of multipath fading [19]. Fading is used to describe the fluctuation of a signal over a period [19]. The cause of physical-layer multipath is illustrated in Fig. 1, where there are many possible paths for the signal to propagate from the transmitter to receiver, and the propagation length of each path is quite likely to be different. In this paper, it is assumed that the time lag between components from the paths is within a symbol interval, and the sum of their amplitude is Rayleigh distributed [19]. Rayleigh distribution is often used to model the statistical characteristics of the received envelope of a signal transmitted through many paths. The pdf (probability density function) of the signal voltage r with Rayleigh fading is

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & \text{for } 0 \leq r \leq \infty, \\ 0 & \text{for } r < 0, \end{cases}$$

where σ is the rms (root-mean square) value of the received signal voltage, σ^2 is the time-average power of the received signal.

Equivalently, the power level $S = r^2$ is exponentially distributed [19–21]

$$p(S) = \frac{1}{2\sigma^2} \exp\left(-\frac{S}{2\sigma^2}\right).$$

A digital modulator transforms discrete inputs into the continuous waveform at a suitable frequency, and the corresponding demodulator transforms the continuous waveform back to discrete outputs. In this paper, incoherent binary FSK is used for modulation and demodulation. BFSK

Fig. 2. The demodulator for incoherent binary FSK.

(binary frequency shift keying) converts “0” into sinusoidal wave with frequency ω_c and “1” into sinusoidal wave with frequency $\omega_c + \Delta\omega$. An implementation of the demodulator is illustrated in Fig. 2. The demodulator compares the outputs at the two frequencies and decides the output to be “0” or “1” [20,21].

Synchronization issues can be investigated at different levels, such as carrier phase, bit, word [20,21], or algorithm [22]. The definitions of synchronized and non-synchronized

reception in this paper are as follows. *If the time lag between simultaneously received packets is small enough in comparison with the duration of a symbol (bit), the reception is called synchronized.* The relation between time lag and synchronized reception is illustrated in Fig. 3. For example, in Fig. 3(a), the time lag between the first and the second packets, T_l , is quite small compared with the symbol duration. Otherwise, the reception is called *non-synchronized*. For example, in Fig. 3(b), the time lag is larger

(a) synchronized reception

(b) non-synchronized reception

Fig. 3. Illustration of propagation delay and synchronization.

than the symbol duration. Such non-synchronized reception may occur because of imperfectly synchronized clocks at the processors and propagation delays. In the following, it is assumed that every slot is designed to contain a guard period [19] to allow all the paths to arrive in a slot interval. Thus, the packets in previous slots do not interfere with the packets in the current slot, as in Fig. 3.

One cause of non-synchronization is imperfectly synchronized clocks at the processors. That is, it is difficult for the processors to synchronize all the processors perfectly at the setting up of the network. If the time lag is not too large, we can add some guard period, as in Fig. 3(b), to avoid the packet in the previous slot collide with packets in the next slot. Therefore, the time lag viewed at packet level does not influence the performance too much. What needs investigation is the influence of time lag at bit level. In mobile ad hoc networks, the timing of the clocks can be synchronized with the advent GPS (global positioning system) [23] technology. Also, in wireless local area networks, the synchronized reception is likely to occur since the distances between nodes are small. For example, the time lag due to a distance of 15 m will be $15/(3 \times 10^8) = 5 \times 10^{-8} \text{ s} = 0.05 \text{ } \mu\text{s}$ (microsecond). Since the speed of electromagnetic wave is $3 \times 10^8 \text{ m/s}$. Even with a data rate as high as 2 Mbs (10^8 bits per second), the bit interval will be $(1/2 \times 10^6) = 0.5 \text{ } \mu\text{s}$, which is much larger than the time lag due to propagation delay. Therefore, it is of interest to examine the differences in the performance with synchronized and non-synchronized reception.

Here we are going to analyze the probability of successful reception of a packet. Assume the average energy per bit [20] for each received packet is E_b . The power level of the Gaussian noise at the receiver is N_0 . We assume the fading is slow [19]. Thus, if one bit is not received successfully, it probably means that fading is quite severe for the whole packet containing the bit, and the packet is unlikely to be received successfully. Therefore, analyzing bit error probability is an approximate of the probability that a packet can be received successfully.

The probability of bit error for incoherent binary FSK in a Rayleigh fading channel is

$$P_b = 1/(2 + \bar{z}), \quad (1)$$

where $\bar{z}$ is the average signal-to-noise ratio [20,21].

To analyze the relation between time lag and the probability of successfully reception, we can use the formula of probability of error for incoherent binary FSK [20]:

$$\int_0^\infty f_{R_1}(r_1) \left[\int_0^\infty f_{R_2}(r_2) dr_2 \right] dr_1,$$

where r_1 and r_2 are the magnitude of the signal $r_1(t)$ and $r_2(t)$ respectively, $f_{R_1}(r_1)$ and $f_{R_2}(r_2)$ are their probability density function.

Given T_1 , the time lag, T_b , the bit duration, W_1 , average power level of packet 1, W_2 , average power level of packet

2, θ , the phase difference between packets 1 and 2, and N_0 , the average noise power, we can calculate the probability of error in a manner similar to the approach of calculating the probability of error for intersymbol interference with multipath transmission in Ref. [20]. That is, we can treat the interference from other users as a multipath component since we will be considering a broadcast algorithm where the packets from different users involved in a collision has the same content.

In the following, we are going to analyze the relation between the number of packets received and the probability of successful reception with synchronized and non-synchronized reception. In the case of *synchronized reception*, i.e. the time lag between the packets is small enough as in Fig. 3(a), the output is the sum of the individual components transmitted from each processor. Since each packet contains the same symbols (bits), the sum of the simultaneously arriving packets has the same symbol (bit) sequence as each packet. For most of the time, the signal simultaneously received can be viewed as constructive interference.

The number of packets arriving at the same time is denoted as n . To truly achieve constructive interference, not only must the symbols overlap but the phase of the carriers must also be synchronized enough at the receiving antenna so that the summation of the two signals will result in larger power level helpful to the demodulation of incoherent FSK. We find that constructive interference is possible because with Rayleigh fading, the in-phase and quadri-phase components of the interfering signal are Gaussian random variable [19–21]. Since the sum of Gaussian random variables is a Gaussian random variable with variance equal to the sum of the variance of all components of it [19–21,24]. Therefore, the average energy per bit of the sum of all received packets is $n \times E_b$ because each packet has the same symbol sequence and the reception is synchronized. Thus, the probability of bit error has the same form as Eq. (2), and $\bar{z} = n \times E_b/N_0$.

In the case of *non-synchronized reception* as in Fig. 3(b), the signals simultaneously received can be viewed as destructive interference for most of the time. That is, the signals from other packets are only interfering as noise. With Rayleigh fading, the power level of the sum of the interfering signals can be calculated as $(n - 1) \times E_b$, and $\bar{z} = E_b/(N_0 + (n - 1)E_b)$.

3. Performance of broadcast algorithms

3.1. System model

The network is represented by an undirected graph G , in which a node represents a mobile user. The existence of an edge between two nodes means that the two nodes have a chance to be connected. There is at most one edge between two nodes. The two nodes with an edge between them are called neighbors to each other. The degree of a node is the

Fig. 4. Examples of mobile network graphs.

number of neighbors this node has. The degree (D) of a network is the maximum degree. The radius (R) of the network is the maximum number of edges in the shortest path between two nodes. Examples of this graph representation are given in Fig. 4.

All processors are assumed to have the same characteristics, and are dedicated to executing the algorithm in a time slot with the same probability p_r .

3.2. Comparison of algorithms

The goal of a broadcast algorithm is to let as many users in the network as possible to receive a message successfully. Each user in the network executes the broadcast algorithm.

Since we are considering a mobile ad hoc network without a base station, centralized algorithm is not easy to carry out because the number of centralized control channels scales with the number of users of the network [14]. Besides, the use of central control is inefficient because the control messages may not be successfully received due to either insufficient power level or collision.

Next, consider deterministic algorithms. Since there is uncertainty in channel condition, i.e. whether two nodes are connected or not is time varying and random, it is difficult to make a deterministic decision based on knowing the connectivity in advance. Because of the randomness in channels, it is also hard to make a scheduling of transmissions in advance and let the broadcast packet carry this scheduling message. Thus, a deterministic algorithm is also difficult to carry out.

Trying to exploit feedback or acknowledgement mechanism is not efficient either. The feedback message may be lost due to randomness in channel, or be garbled due to collision of feedback message. The feedback information may also be useless because it tells only the past condition of the channel, and may not be very helpful in determining the future condition of the channel.

3.3. Distributed and randomized algorithm

Based on the above reasoning, a successful broadcast algorithm should make a decision in a distributed manner. Each processor makes a decision individually, and it uses a non-deterministic rule to decide when to transmit. It also does not need feedback message. For example, after

receiving a packet, a processor can decide whether to transmit this packet or not randomly.

Therefore, a more practical approach to fulfilling the task of broadcast is a *distributed and randomized* algorithm [22], which satisfies the above requirements. In this paper, we choose the randomized distributed algorithm proposed by Bar-Yehuda et al. [14] as the example for the performance analysis of distributed randomized algorithm with consideration of channel reliability. In this algorithm, any mobile user receiving the message to be broadcast in this time slot will broadcast the message at next time slot. To avoid the situation that too many nodes are transmitting and collisions of packets are too frequent, the algorithm introduces a procedure called “Decay” [14]. In the “Decay” procedure, a processor that is transmitting at this time-slot may stop or continue transmitting the same packet at the next time slot with a specified probability (this probability will be abbreviated as q). This process repeats for $\lceil \tau \rceil$ time slots.

Procedure Decay($\lceil \tau \rceil, m$); // m : the message to be broadcast.

repeat at most $\lceil \tau \rceil$ times (at least once)
transmit m to all neighbors;
set coin = 1 with probability q and 0 with probability $1 - q$;
until coin = 0;

The whole randomized algorithm uses $\lceil \tau \rceil$ as the number of time slots in a stage, i.e. the transmitting processors restart executing “Decay” every $\lceil \tau \rceil$ time slot [14]. In time slot 0, a processor, called the source, generates and broadcasts the message m to other users. Then every processor executes the following broadcast procedure.

Procedure Broadcast;
Wait until receiving the message m ;
do T times
Wait until (Time mod $\lceil \tau \rceil$) = 0;
Decay($\lceil \tau \rceil, m$);
od.

For the situation that a sequence of packets have to be broadcast, the above algorithm can serve as one step in a more complete algorithm. The packets are broadcast in sequel, so the collisions of packets involve only packets of the same content.

How to adjust this algorithm to achieve better performance for synchronized and non-synchronized reception will be investigated in the next section.

3.4. Performance of broadcasting

The algorithm used in both the cases of synchronized and non-synchronized reception is the same. The difference relies on the system parameters $\lceil \tau \rceil$. Consider the situation in which there are n neighbors trying to broadcast the message m to user i at time slot 0. Let $f(n, t)$ denote the

probability that the probability that i receives m in t or fewer slots i from the n transmitting neighbors. As in Ref. [14], **Procedure Broadcast** uses $\lceil \tau \rceil$ as the number of time slots in a stage, i.e. the processors restart executing “Decay” every $\lceil \tau \rceil$ time slot. Since the maximum value of degrees is D , i.e. the upper bound of n is D , we choose $\lceil \tau \rceil$ a minimum value such that $f(D, \lceil \tau \rceil) \geq P_s$. Therefore, the difference in performance of synchronized and non-synchronized reception can be known by the analysis of $f(n, t)$ in each case. Thus, in the following paragraphs, $f(n, t)$ will be analyzed.

The recurrence relation [25,26] of the probability that i receives m before time t from one of the n transmitting neighbors (denoted as $f(n, t)$) is

$$f(n, t) = P_{\text{success}}[n] + 1 - P_{\text{success}}[n] \sum_{j=0}^n \binom{n}{j} q^j (1 - q)^{n-j} f(j, t-1), \quad (2)$$

where $P_{\text{success}}[n]$ represents the probability that i receives m successfully in the 0th time slot, i.e. the probability that the packet is successfully received under the condition that n packets are received at the same time, and in the term

$$(1 - P_{\text{success}}[n]) \sum_{j=0}^n \binom{n}{j} q^j (1 - q)^{n-j} f(j, t-1), (1 - P_{\text{success}}[n])$$

is the probability that i does not receive m successfully in the 0th time slot, and

$$\binom{n}{j} q^j (1 - q)^{n-j}$$

represents the probability of occurrence of j neighbors keeping transmitting in the next time slot, which is multiplied by $f(j, t-1)$, the probability of successful reception for each case. The probability $P_{\text{success}}[n]$ for synchronized reception and non-synchronized reception can be calculated from the analysis in the previous section as follows. Considering the probability that a processor is taking a rest, the probability that a user receives a packet successfully is

$$P_{\text{success}}[n] = (1 - p_r) \sum_{j=0}^n \binom{n}{j} (1 - p_r)^j p_r^{(n-j)} P_{\text{received}}[j], \quad (3)$$

where $(1 - p_r)$ is the probability that the receiving node is not taking a rest,

$$\binom{n}{j} (1 - p_r)^j p_r^{(n-j)}$$

represents the probability that j neighbors are not taking a rest and are transmitting the packet in this time slot, and $P_{\text{received}}[j]$ is the probability that a packet can be received correctly in a collision of j packets, which is obtained from the analysis of the successful reception. As we will see later,

the non-synchronized interference will likely destroy the desired packet, and the synchronized interference can be approximated as an enhancement of the desired signal. Therefore, $P_{\text{received}}[j] \approx j(1 - p_{\text{ns}})^{j-1}$, where p_{ns} is the probability that a interfering packet becomes non-synchronized, $(1 - p_{\text{ns}})^{j-1}$ is the probability that one packet can be successfully received because the other $j-1$ packets are synchronized, and this term is multiplied by j since there are j possible selections of the desired packet with j simultaneously received packets.

4. Numerical examples and discussions

As indicated in Sections 1 and 2, in a network G , the physical environment and the design of working power determine the network topology and the parameters R and D . There is a tradeoff between D and R . If the transmitters increase their power levels, R could decrease and D could increase. Therefore, evaluating the performance of the broadcast algorithm is very helpful not only in saving time but also in increasing power efficiency. From the result of Ref. [14], it seems that minimizing the time complexity requires minimizing R and D . Besides, from the result in Ref. [17], different retransmission probability (q) also influences the performance a lot. How to decide an optimum value depends on the values of $f(n, t)$ and $\lceil \tau \rceil$. Based on their importance, we are going to give numerical examples of $f(n, t)$ and $\lceil \tau \rceil$ in this section. And this optimization problem will be examined later in this section. Before that, we evaluate Eq. (2) using recursive method [25,26]. First of all, we have to evaluate $P_{\text{received}}[n]$, the probability that the packet is received correctly; then can we analyze $f(n, t)$ by Eq. (2); then $\lceil \tau \rceil$ is obtained as a minimum value such that $f(D, \lceil \tau \rceil) \geq P_s$.

4.1. Relation between time lag and probability of successful reception

$P_{\text{received}}[j]$ can be obtained from the analysis in the previous section. To investigate the impacts of the time lag, in the following we examine the case with two simultaneously received packets, i.e. $j = 2$. In practical systems, to improve the immunity to reception errors due to noise, error control coding is often exploited. Here we adopt the Hamming coding, which is a common block error control code [20]. The encoding of the Hamming code is to multiply four bits of messages by the generator matrix $\mathbf{G}$ to obtain a block of seven encoded bits, where

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and the decoding is to multiply the block of seven encoded

Fig. 5. The probability of successfully reception with different time lag when $n = 2$.

bits by the parity-check matrix $\mathbf{H}$,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix},$$

to know if there are any mistakes in the first bits of the received block [21]. Hamming code can be used to detect and correct any single error in the received block [20]. Therefore, the probability of successful reception is calculated as

$$P_{\text{received}}[j] \Big|_{j=2} = ((1 - P_b)^7 + 7P_b(1 - P_b)^6)^{L/4},$$

where L is the length of the original message without error correction coding, $L/4$ is the number of blocks each with seven bits, and P_b is the probability of bit error. For incoherent binary FSK in a collision of two packets with Rayleigh fading, $P_b = 1/(2 + \bar{z})$, where $\bar{z}$ is average signal-to-noise ratio $= E_b(1 + (T_b - T_1)/T_b)/(N_0 + T_1/T_b)$, E_b is energy per bit, N_0 is noise level, T_b is bit duration, and T_1 is time lag between the two packets.

In Fig. 5, the relation between the probability of successful reception $P_{\text{received}}[2]$ and the time lag is plotted. The contents of the two simultaneous received packets are the

same, since it is assumed that the users broadcast a packet at a time. We denote the power level of the two packets by W_1 and W_2 , respectively. For the case of the same packet power ($W_1 = W_2 = 20$ dB), the probability of successful reception decreases with the increasing time lag. For time lag equal to one, i.e. time lag equal to symbol duration, the probability of successful reception is almost zero. The situation for time lag greater than one is not plotted because the probability of successful reception does not vary too much from time lag equal to one. Therefore, we can say that the reception begins to be non-synchronized after about time lag equal to 0.2 for the case of $W_1 = W_2 = 20$ dB. With the time lag increasing, the probability of successful reception can be raised only by raising the signal to interference ratio, as the $W_1 = 20$ dB, $W_2 = 10$ dB curve in Fig. 5, since the interference is destructive. For synchronized reception, the probability of successful reception for $W_1 = W_2 = 20$ dB is larger than that for $W_1 = 20$ dB, $W_2 = 10$ dB because the interference is constructive.

4.2. Probability of successful reception with n packets simultaneously received $P_{\text{received}}[n]$ for different cases

The influences of synchronized and non-synchronized reception on $P_{\text{received}}[n]$ were analyzed, and we obtain the numerical examples. We have the following observations from the numerical results in Fig. 6, where p_{ns} is the probability of a packet being non-synchronized received.

Then, it can be seen that $P_{\text{received}}[n]$ decreases more quickly with p_{ns} (the probability of being non-synchronized) increasing. That means, with the probability of being non-synchronized reception increasing, the probability of successful reception decreases. This stems from the phenomenon that for synchronized reception, simultaneous reception of more than one duplicate of the same packet

(a) relation between $P_{\text{received}}[n]$ and n

(b) relation between $(P_{\text{received}}[n])$ and p_{ns} . n = the number of packets received,

Fig. 6. The probability of successfully reception $P_{\text{received}}[n]$ in a collision of n packets.

(a.1) synchronized reception: relation between P_{success} and p_r . (b.1) non-synchronized reception ($p_{\text{ns}} = 0.1$): relation between P_{success} and p_r . (a.2) synchronized reception: relation between P_{success} and n . (b.2) non-synchronized reception ($p_{\text{ns}} = 0.2$): relation between P_{success} and n .

Fig. 7. The probability that the packet is received correctly from n neighbors $P_{\text{success}}[n]$, n is the number of packets received, p_r the probability that a processor is not transmitting or receiving, P_{success} the probability of successful receiving the packet from n neighbors with the packet, p_r : the probability that a processor is busy with other tasks.

Fig. 7. (continued)

results in better performance. For non-synchronized reception, the interfering packets cause adverse effects.

4.3. Probability of successful reception from n neighbors $P_{\text{success}}[n]$ for different cases

The influences of synchronized and non-synchronized reception on $P_{\text{success}}[n]$ were analyzed, and we obtain the numerical examples. For synchronized reception, the probability of being non-synchronized (p_{ns}) is 0, and for non-synchronized reception, p_{ns} is assumed to be 0.2. The probability that a processor is not carrying out broadcast (p_r) is chosen from 0.03 to 0.27. We have the following observations from the numerical results in Fig. 7.

First of all, it can be seen from Fig. 7(a.1) that $P_{\text{success}}[n]$ decreases with p_r increasing for synchronized reception. However, in Fig. 7(b.1), $P_{\text{success}}[n]$ not always decreases with p_r with n for non-synchronized reception. In fact, when $n = 3$, $P_{\text{success}}[n]$ indeed decreases with p_r . For n larger than three, $P_{\text{success}}[n]$ might increase with p_r , and this phenomenon is especially true for larger n . The reason for this phenomenon is that in non-synchronized reception, the probability that a user receives the packet correctly decreases with the number of collided packets. Therefore, if there are some processors not transmitting a packet, the

number of collided packets could decrease, so the probability of successful reception becomes higher.

4.4. $f(n, t)$ for different parameters

The influences of different parameters on $f(n, t)$ were analyzed, and we obtain the numerical results in Fig. 7 where there are curves of $f(n, t)$ versus t . To examine the effects of different p_{ns} and p_r , we analyzed the cases with $p_{\text{ns}} = 0.25, 0.5, 0.75$ and $p_r = 0.09, 0.18, 0.27$. For synchronized reception, since the larger the n , the better the performance, the retransmission probability q is chosen as one. The result of synchronized reception is shown in Fig. 8(a). For non-synchronized reception, retransmission probability q is chosen as a value smaller than one to solve collisions. Thus, we obtain the results of the cases $q = 0.25, 0.5$ and 0.75 as in Fig. 8(b). We have the following findings from Fig. 8.

In Fig. 8(a), $f(n, t)$ quickly approaches one as t increases. Basically, the larger the signal-to-noise ratio and the lower the probability that a processor is not carrying out broadcast, the better the performance for $t = 1, 2$. However, there is almost no difference for $t \geq 2$.

In Fig. 8(b), the curves increase with t at different rates. Fig. 8(b.1)–(b.3) shows the cases with $q = 0.25, 0.5$ and 0.75 , respectively. The larger the q , the more slowly $f(n, t)$ increases with t . Meanwhile, the asymptotic value of $f(n, t)$ with respect to t increases with increasing q . This is because

(a.1) synchronized reception: relation between $f(n, t)$ and t . (a.2) synchronized reception: relation between $f(n, t)$ and n . (a.3) synchronized reception: relation between $f(n, t)$ and p_r . (b.1.1) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.25$, $p_{ns} = 0.25$

Fig. 8. The probability that the packet is received correctly from n neighbors in time t ($f(n, t)$), n is the number of packets received, p_r the probability that a processor is not transmitting or receiving.

(b.1.2) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.25$, $p_{ns} = 0.5$ (b.1.3) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.25$, $p_{ns} = 0.75$ (b.2.1) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.5$, $p_{ns} = 0.25$ (b.2.2) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.5$, $p_{ns} = 0.5$

Fig. 8. (continued)

(b.2.3) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.5$, $p_{ns} = 0.75$ (b.3.1) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.75$, $p_{ns} = 0.25$ (b.3.2) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.75$, $p_{ns} = 0.5$ (b.3.3) non-synchronized reception: relation between $f(n, t)$ and t , $q = 0.75$, $p_{ns} = 0.75$

Fig. 8. (continued)

(a) non-synchronized reception: relation between $\lceil \tau \rceil$ and D , $q = 0.5$ (b) non-synchronized reception: relation between $\lceil \tau \rceil$ and D , $q = 0.75$

Fig. 9. The time slots needed before the probability of successful reception from n neighbors is larger than 0.5 ($\lceil \tau \rceil$), n is the number of packets received, p_{ns} the probability of being non-synchronized, p_r the probability that a processor is not transmitting or receiving.

for smaller q , the collisions of packets can be solved faster with fewer neighbors transmitting. However, in the long run, very few neighbors keep transmitting, so $f(n, t)$ in the long run is smaller. For larger q , the situation is just opposite. Collisions of packets cannot be solved quickly because there are still many neighbors transmitting. However, in the long run, there are more neighbors transmitting, so $f(n, t)$ in the long run is larger.

The cases with $p_{ns} = 0.25, 0.5, 0.75$ are plotted in Fig. 8. It can be seen that the larger the probability of being non-synchronized, the lower $f(n, t)$ is. Adjusting the probability of transmitting (q) provides a method for improving the performance. For example, as in Fig. 8(b.3), with $q = 0.75$, all the curves have better performance than that in Fig. 8(b.1.2) with $q = 0.25$ and 0.5 . Similarly, the same phenomenon occurs between the probability that a processor is not carrying out broadcast of (p_r) and q .

4.5. $\lceil \tau \rceil$ for different parameters

In Fig. 9(a) and (b), we evaluate another important parameter $\lceil \tau \rceil$ —number of time slots required in a stage of the broadcast algorithm. Following the convention in Ref. [14], we chose the same value $P_s = 1/2$ as the required value of $f(D, t)$ at the end of a stage. What differs from Ref. [14] is the number of required time slots in each

stage, i.e. $\lceil \tau \rceil$. It can be proved that [1,14] with the probability of successful reception in a stage ($P_s = 1/2$), the number of time slots needed is within $\lceil \tau \rceil \times T(\epsilon)$ with probability larger than $1 - \epsilon$, where

$$T(\epsilon) = 2R + 5M(\epsilon)\text{Max}(\sqrt{R}, M(\epsilon))$$

and

$$M(\epsilon) = \sqrt{\log \frac{N}{\epsilon}},$$

where R is the radius of the network.

For synchronized reception, from the results of $f(n, t)$ such as that in Fig. 8(a), we can see that one time slot is enough for $f(n, t)$ to be larger than $P_s = 0.5$. Therefore, for synchronized reception, $\lceil \tau \rceil = 1$.

For non-synchronized reception, we investigate the influences of different p_r and q . In Fig. 9(a) and (b), the curves $\lceil \tau \rceil$ versus D for $p_{ns} = 0.25, 0.5, 0.75$, $p_r = 0.09, 0.18, 0.27$, and $q = 0.5, 0.75$ are demonstrated. The case $q = 0.25$ was not considered because we found that $P_s = 0.5$ cannot be reached for larger p_{ns} and p_r with $q = 0.25$. We have the following findings from these numerical results in Fig. 9(a) and (b).

Generally, the number of time slots needed ($\lceil \tau \rceil$) such that $f(D, \lceil \tau \rceil) \geq 0.5$, increases with D increasing. And the relation is almost logarithmic asymptotically, i.e. $\lceil \tau \rceil$ grows with rate $\log(D)$, for non-synchronized reception.

From Fig. 9(a) and (b) and the fact that $P_s = 0.5$ cannot be reached for larger p_{ns} and p_r with $q = 0.25$, the best value of q is around 0.5. If q is too small, it is possible that all processors stop transmitting too soon. If q is too large, the detrimental collisions will continue to occur for a longer period, and $f(n, t)$ will be increasing with t at a lower rate, as Fig. 8(b.3.1)–(b.3.3).

The impact of p_r is not serious as long as p_r is within a reasonable range. Therefore, allowing the processors not to carry out broadcast temporarily does not damage the performance a lot. The processors can thus sometimes save power or deal with other urgent task. In fact, when the number of simultaneous transmitting neighbors is large, allowing the processors not to carry out broadcast might also be a method to solve collision of packets. For example, as in Fig. 9(a) and (b), when $p_{ns} = 0.75$, $\lceil \tau \rceil$ is smaller with larger p_r (0.18, 0.27) than with smaller p_r (0.09).

4.6. Optimization of the network topology

The network topology can be decided by the transmission range [2] and the antenna direction (pattern) [17]. From the above analysis, we can investigate the optimization problem mentioned in the beginning of this section.

From the result in Ref. [14] and Theorems 3 and 4 in Ref. [1], the time needed to carry out broadcast for non-synchronized reception is proportional to $\lceil \tau \rceil \times T(\epsilon)$, where $\lceil \tau \rceil$ grows

Fig. 10. Logical graph representations of a network with nine users.

almost logarithmically with D asymptotically,

$$T(\epsilon) = 2R + 5M(\epsilon)\text{Max}(\sqrt{R}, M(\epsilon)).$$

Let $N = 9$ and $\epsilon = 0.01$, then $\log_2(N/\epsilon) = 9.8$, $M(\epsilon) = 3.13$.

Consider the three examples in Fig. 10: (a) $D = 2$, $R = 4$; (b) $D = 3$, $R = 3$; (c) $D = 8$, $R = 2$. We have $\text{Max}(\sqrt{R}, M(\epsilon)) = 3.13$, and $T(\epsilon) = 2R + 5 \times 9.8 = 2R + 49$.

In the following, the value of $\lceil \tau \rceil$ is from Fig. 9(a). Consider the case $p_{ns} = 0.5$, $p_r = 0.18$.

- (a) $D = 2$, $\lceil \tau \rceil = 1$; $R = 8$, $T(\epsilon) = 2 \times 8 + 49 = 65$; needs $1 \times 65 = 65$ time slots.
- (b) $D = 3$, $\lceil \tau \rceil = 2$; $R = 3$, $T(\epsilon) = 2 \times 3 + 49 = 55$; needs $2 \times 55 = 110$ time slots.
- (c) $D = 8$, $\lceil \tau \rceil = 3$; $R = 2$, $T(\epsilon) = 2 \times 2 + 49 = 53$; needs $3 \times 53 = 159$ time slots.

Therefore, for this case, the smaller the D , the lesser the time needed. R may be allowed to be larger. This is because the occurrence of collisions is the main obstacle of broadcast for non-synchronized reception.

For *synchronized reception*, the situation is in contrast to the situation for non-synchronized reception. Therefore, the smaller the R , the less the time needed. This is because the successful reception probability is better for larger D , and the time needed by broadcast algorithm is in proportion to the radius (R).

5. Conclusions

In this paper, we analyze the probability of successful reception of the packet when more than one packet is received at the same time. The probability of successful reception is not only dependent on the reception power of packets [17], but also depends on the time lag and the contents of the collided packets. There is a great gap between the performance of synchronized and non-synchronized reception. In broadcast, the performance of synchronized reception of several packets is in fact better than that of reception of only one packet. This is because, the multiple packets received contain the same content, the symbols are synchronized, and the envelope magnitude of the sum of the signals is larger than that of only one signal. Besides, incoherent demodulation is used, so the envelope magnitude decides whether a packet can be received correctly. On the contrary, non-synchronized reception of the packets caused adverse effects. Since the symbols are not synchronized, the content of one packet acts as noise

to other packets. Therefore, the receiving processor has a smaller chance to receive any one of the packets successfully.

The substantial difference in performance of synchronized and non-synchronized reception also influences how the broadcast algorithm is used. For synchronized case, the processors in fact do not need to use “decay” to solve collision of packets, since collision of packets in fact enhances the probability of successful reception. For non-synchronized case, the processors need to use “decay” to solve collision of packets, since non-synchronized reception of packets might destroy the packet. Therefore, the number of time slots needed in a stage is practically one for synchronized reception, and grows logarithmically with the maximum number of neighbors (i.e. $\log D$) for non-synchronized reception.

We also investigate how to improve the performance. When collisions are not frequent and there are only few neighbors, raising q can keep the neighbors transmitting for longer time. When collisions are frequent, the time and power needed can be reduced by using smaller q to reduce the number of collisions. Raising the transmitter power and thus signal-to-noise ratio can only improve the performance within a limit, after which increasing power level does not result in great improvement. Rather, choosing a proper value of retransmission probability q is a more effective way to improve the performance.

How to optimize the network topology to reduce the time needed for broadcast also differs in the cases of synchronized and non-synchronized reception. Minimizing the time complexity requires minimizing both the radius of the network R and the degree of the network D . However there is a tradeoff between D and R . For example, if the transmitters increase their power levels, R could decrease and D could increase. To reduce the time needed for broadcast, trying to have *smaller* R is a better method for *synchronized reception*, but trying to have *smaller* D is a better method for *non-synchronized reception*.

Practical processors might not carry out broadcast temporarily. In fact, for the case that collision is detrimental and collisions are frequent, allowing the processors not to carry out broadcast temporarily might also be a method to solve collision of packets. We found that the performance degrades with p_r (probability that a processor does not carry out broadcast) for synchronized reception, and the performance sometimes improves with p_r for non-synchronized reception.

In mobile ad hoc networks, broadcast provides an effective way to implement communications among users under the condition that there is no base station. This is especially true for wireless local area networks, where synchronized reception can be more easily achieved. Furthermore, the mobility of users increases the cost of maintaining a routing table needed by deterministic algorithms. Therefore, randomized algorithms are more appealing. There are several directions for future work. First, the performance of multiple messages broadcast,

routing, and multicast can be investigated [15,27]. Second, other possible methods of broadcast in mobile ad hoc networks can also be studied. For example, how to carry out broadcast in a completely asynchronous (i.e. unslotted) manner can be studied. Third, we can carry out more accurate analysis of the probability of successful reception with practical considerations. One consideration is the correlation between consecutive events, such as severe fading, non-synchronized reception, and a processor's pause in carrying out broadcast algorithm. Another consideration is the relation between successful reception at packet level and signaling level. The accuracy of the synchronization of the clocks can also be considered. Fourth, we will investigate possible performance improvements by the use of receiver diversity and RAKE receiver to distinguish and combine packets from different users [28].

Acknowledgements

The authors would like to thank the reviewers for their constructive comments.

This work was supported by Taiwan Ministry of Education under the project 89-H-FA07-1-4 Learning Technology.

References

- [1] D.P. Bertsekas, R.G. Gallager, Data Networks, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1992.
- [2] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [3] U. Black, Mobile and Wireless Networks, Prentice-Hall, Englewood Cliffs, NJ, 1996.
- [4] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Standards Board, June 1997.
- [5] T.-W. Chen, J.T. Tsai, M. Gerla, QoS routing performance in multihop, multimedia, wireless networks, in: Proceedings of ICUPC'97 (Sixth IEEE International Conference on Universal Personal Communications), vol. 2, pp. 557–561.
- [6] C.-C. Chiang, M. Gerla, Routing and multicast in multihop, mobile wireless networks, in: Proceedings of ICUPC'97 (Sixth IEEE International Conference on Universal Personal Communications), vol. 2, pp. 546–551.
- [7] C.-C. Chiang, H.-K. Wu, W. Liu, M. Gerla, Routing in clustered multihop mobile wireless networks with fading channel, in: Proceedings of IEEE Singapore International Conference on Networks (SICON'97).
- [8] C.R. Lin, M. Gerla, MACA/PR: an asynchronous multimedia multihop wireless network, in: Proceedings of INFOCOM'97, vol. 1, pp. 118–125.
- [9] I. Chlamatic, O. Weinstein, The wave expansion approach to broadcasting in multihop radio networks, IEEE Transactions on Communications COM39 (3) (1991) 426–433.
- [10] M. Gerla, J.T. Tsai, Multiculster, mobile, multimedia radio networks, Wireless Networks 1 (1995) 255–265.
- [11] E. Pagani, G.P. Rossi, Reliable broadcast in mobile multihop packet networks, in: Proceedings of MOBICom'97, pp. 34–42.
- [12] N. Alon, A. Bar-Noy, N. Linial, D. Peleg, A lower bound for radio broadcast, Journal of Computer and System Sciences 43 (1991) 290–298.
- [13] D. Bruschi, M. Del Pinto, Lower bounds for the broadcast problem in mobile radio networks, Distributed Computing 10 (1997) 129–135.
- [14] R. Bar-Yehuda, O. Goldreich, A. Itai, On the time-complexity of broadcast in multi-hop radio networks: an exponential gap between determinism and randomization, Journal of Computer and System Sciences 45 (1992) 104–126.
- [15] R. Bar-Yehuda, A. Israeli, A. Itai, Multiple communication in multihop radio networks, SIAM Journal of Computation 22 (1993) 875–887.
- [16] E. Kushilevitz, Y. Mansour, An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks, SIAM Journal of Computation 27 (1998) 702–712.
- [17] K.-H. Pan, H.-K. Wu, R.-J. Shang, F. Lai, Performance analysis of distributed broadcast algorithms in wireless networks with consideration of channel reliability, in: Proceedings of PDCN'98 (Second IASTED International Conference on Parallel and Distributed Computing and Networks), pp. 549–554.
- [18] T. Tsiligrirides, Queueing analysis of a hybrid CSMA/CD and BTMA protocol with capture, Computer Communications 19 (1996) 763–787.
- [19] T.S. Rappaport, Wireless Communications: Principles and Practice, Prentice-Hall, Englewood Cliffs, NJ, 1996.
- [20] R.E. Ziemer, W.H. Tranter, Principles of Communications, 4, Houghton Mifflin, Boston, MA, 1995.
- [21] S. Haykin, Communication System, 3rd ed., Wiley, New York, 1994.
- [22] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge, New York, 1995.
- [23] E.D. Kaplan (Ed.), Understanding GPS: Principles and Applications Artech House, Norwood, MA, 1996.
- [24] S. Ross, A First Course in Probability, 5th ed., Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [25] J.W. Grossman, Discrete Mathematics: an Introduction to Concepts, Methods, and Applications, Macmillan, New York, 1990.
- [26] R.L. Graham, D.E. Knuth, O. Patashnik, Concrete Mathematics, Addison-Wesley, Reading, MA, 1991.
- [27] G.D. Stamoulis, J.N. Tsitsiklis, Efficient routing schemes for multiple broadcasts in hypercubes, IEEE Transactions on Parallel and Distributed Systems 4 (7) (1993) 725–739.
- [28] H.V. Poor, G.W. Wornell (Eds.), Wireless Communications: Signal Processing Perspectives Prentice-Hall, Englewood Cliffs, NJ, 1998.