Random Route Discovery Packet Drop Using Load Factor of MAC Layer in IEEE 802.11 Ad Hoc Networks *

PENGYong, XIE Dongliang, SUN Liang and CHENG Shiduan

National Laboratory of Switching Technology & Telecommunication Networks
BeiJing HaiDian District Xi Tu Cheng Road 10#, Mailbox:187, Beijing 100876, CHINA
E-mail: pengyong@bupt.edu.cn

Abstract: Recent research has shown that the original DCF access mode of IEEE 802.11 has low performance in heavy traffic and multi-hop environment. In this paper, the research focus is placed on the improvement of IEEE 802.11 networks in asymmetric contention condition multi-hop environment. Performance enhancement is implemented by the interaction of MAC layer and ad hoc routing mechanism. Different from previous mechanisms that are realized by routing layer metric measurement (e.g. delay, link state), the proposed Random Route Discovery Packet Drop (R^2DPD) mechanism introduces a MAC parameter to evaluate current wireless media contention state (i.e. load factor) to be adaptive to the MAC layer contention. This new mechanism drops ad hoc routing packets in the MAC layer according to the on-line measuring of local contention state (i.e., load factor). Stations in heavy contention environment drop route discovery packets with high probability so that the congestion is relieved. Simulation results have shown that the new MAC algorithm can achieve better performance than the original DCF algorithm.

Keywords: IEEE 802.11 Wireless LANs, Ad hoc Networks, Dynamic Source Routing.

1. INTRODUCTION

Wireless communication provides users with a lot of convenience and it is natural that the relative study on wireless technology has gradually become a research focus. As a most frequently used WLAN protocol at present, IEEE 802.11[1] protocol is an important local access method of the wireless communication. IEEE 802.11 protocol provides users with two different kinds of access methods. These two methods are DCF (Distributed Coordination Function) and PCF (Point Coordination Function). DCF is the basis access method of IEEE
802.11. People can use the IEEE 802.11 DCF to construct the infrastructure networks and ad hoc type network as well. The DCF access mode has managed to realize the asynchronous multi-access by using the contention window exponential backoff algorithm. PCF is based on DCF\[1\], it can provide user with synchronous service. DCF access method can easily be implemented and most present manufacturers use this mode to design their WLAN interface card. Recent research has shown that the traditional DCF access mode will deteriorate throughput and fairness, especially when the number of active stations is large\[4\]. Also the IEEE 802.11 protocol is found inefficient in case of multi-hop ad hoc communication topology, especially when upper transport layer is TCP\[5\].

The purpose of this paper is to improve the interaction of original DCF and routing mechanism in case of multi-hop environment with hotspot contention area. The main idea is to use the MAC layer load factor information to improve the ad hoc routing protocol. Different from those intelligent routing protocols that are realized on the routing layer, the proposed mechanism use the MAC contention measuring results because the MAC layer can sense the contention condition more accurately than the routing layer. We have first introduced a load factor parameter $l$, which is closely relative to the contention condition of current wireless channel. The value of parameter $l$ is easy to obtain. The optimum $l$ value (i.e. $l_{opt}$) corresponding to the maximum throughput is a value that can be counted easily according to the mean collision length \[2\]. We implement the net state measurement by the on-line measuring of load factor $l$. Then we use the measured MAC contention information to design the randomly route discovery packet drop (R2DPD) algorithm. Using R\(^2\)DPD, stations in heavy contention condition will have less probability in forwarding packets in order to relieve the congestion.

This paper is organized as follows. Section 2 describes some background knowledge, which will be the basis of later discussion. Section 3 describes the principle and implementation of the new algorithm R\(^2\)DPD. Section 4 is the simulation description and result analysis. We end the paper with concluding remarks in section 5.

2. BACKGROUND

2.1 IEEE 802.11 Protocol

The IEEE 802.11 MAC protocol offers two types of service to its users: asynchronous and synchronous (or, rather, contention free). These types of services can be provided on top of a variety of physical layers and for different data rates. The asynchronous type of service is always available whereas the contention free is optional. The asynchronous type of service provided by the Distributed Coordination Function (DCF) implements the basis access method of the IEEE 802.11 MAC protocol. We also call DCF a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol.

In order to improve the performance of IEEE 802.11 DCF protocol, previous researchers have done much research on the IEEE802.11 protocol. The authors of \[5\][11][12] model
IEEE 802.11 DCF using the Markov chain. [12] has also derived the maximum saturation throughput. [11] has done some modifications on the [12] modeling and proposed a novel handshake mechanism. Some other people make efforts to achieve throughput improvement by changing the packet flow process or some parameters such as Inter-Frame space and window size according to different upper layer service type [9-10]. The author of [9] has considered applying different parameters to the DCF access mode according to different applications in order to improve the efficiency of the protocol. These algorithms focus on the QoS provision in ordinary contention environment and don’t consider the heavy contention environment. In [11], the original DCF access mode is modified to fit for bi-directional data transfer, which can improve upper layer TCP goodput. But unfortunately, they have neither presented an effective way to improve DCF algorithm in heavy contention condition nor considered performance enhancement in multi-hop environment. In [2], we have analyzed the window backoff mechanism and proposed to tune the initial contention window size to a proper value in order to keep the ratio between the mean length of collisions and mean idle length near the optimal value.

In this paper, we borrow the previously proposed $l$ parameter to evaluate the contention condition of the wireless network. The routing discovery packet drop function described later is a nonlinear function of the $l$ parameter. The MAC contention aware routing will lead to performance improvement. The new mechanism is proved effective in later simulation results.

2.2 DSR Routing Mechanism Introduction

DSR [3] is a most frequently used routing protocol of ad hoc networks. It uses source routing rather than hop-by-hop routing. Each packet to be routed carries in its header the complete, ordered list of nodes through which the packet must pass. The key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward, since the packets themselves already contain all the routing decisions. This fact, coupled with the on-demand nature of the protocol, eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols.

The DSR protocol consists of 2 mechanisms: Route Discovery and Route Maintenance. Route Discovery is the mechanism by which a source wishing to send a packet to a destination obtains a source route to the destination. The source node first broadcasts a ROUTE REQUEST. This request is broadcasted through the network and is answered by a ROUTE REPLY packet from either the destination node or another node that knows a route to the destination. To reduce the cost of Route Discover, each node maintains a cache of source routes it has learned or overheard, which it aggressively uses to limit the frequency and propagation of ROUTE REQUESTs.

Route maintenance is the mechanism by which the source detects if the network topology has changed such that it can no longer use its route to the destination because two nodes listed in the route have moved out of range of each other. When Route Maintainer indicates a
source route is broken, the source is notified with a ROUTE ERROR packet. The source can then attempt to use any other route to the destination already in its cache or can invoke Route Discovery again to find a new route.

Obviously, the prior DSR routing mechanism doesn’t consider the congest condition of the wireless shared media. Even if the source node has found a route to the destination through the hotspot congest area, the packet that is sent through the discovered route will undoubtedly experience many times of collisions in the congested area. So the route will be a low quality route. At the same time, when a route becomes unavailable, upper layer routing protocol can’t distinguish whether the link down (packet retransmission exceed limit) is caused by congestion or by wireless high error rate and will invoke Route Discovery again. Obviously, too frequent rebroadcast of ROUTE REQUEST to the congested areas will undoubtedly lead to overall system performance degradation. So we consider the local contention state of 802.11 MAC to be very useful to the upper layer routing mechanism (e.g. in the course of route discovery of DSR). The above consideration has lead to the later described R²DPD protocol.

3. NET STATE DETECTION AND PROPOSED MECHANISM OVERVIEW

Suppose every station gain access to the wireless media using the DCF access mode and form an ad hoc network. A successful data packet transmission will experience the process illustrated in figure 1, where $t_v$ is defined as virtual transmission time of a station (different from [6] definition), $t_{coll}$, $t_{free}$, $t_{succ}$ and $t_{pack}$ are collision time slots, free time slots, successful transmission of other active mobile stations and data packet transmission time slots of this station, respectively. At any time, the network must be in one of the below 3 states:

State 1 (Under load state): The ratio between $t_{coll}$ and $t_{free}$ is smaller than optimal. The reason is that current contention window is too big when compared to current wireless media load and have caused unnecessary backoff delay. For the parameter set of traditional DCF, this case is unusual.

State 2 (Over load state): In heavy contention scenario, the ratio between $t_{coll}$ and $t_{free}$ is often bigger than optimal. The reason is that current contention window is too small when compared to current media load and one success transmission of data frame will often experience many times of collision. The limited self-adapt capability of the DCF exponential window backoff algorithm cannot tune the access probability of heavy contention condition stations to achieve optimal. At the same time, the contention window of the stations that encountered many times of collisions will remain a high level in a fairly long period and those stations will have less possibility of obtaining access to the wireless media compared to those stations with fewer collisions. As a result, the throughput and fairness of wireless access is weakened.

State 3 (Normal state): In light or moderate contention scenario, the ratio between $t_{coll}$ and $t_{free}$ is around the optimal value, the access probability of the stations is around the optimal and the maximum throughput is achieved at this time.
The state 2 is very usual when number of node is large. The network state transition diagram in original DCF algorithm is depicted in the figure below.

Obviously, we should carefully design the protocol in order to keep the network in the normal state. In a distributed wireless ad hoc system, stations will be always in different contention condition and will sense different wireless carrier contention property. Those stations in heavy contention hotspot area will experience much more collisions than those in light contention area. In prior routing protocol, if packet retransmission times of a congested station exceed the protocol upper limit (e.g. 7 times for packet ShortRetryLimit in IEEE 802.11 specification), the packet is discarded and the upper layer routing protocol will consider the link is down and will rebroadcast routing packet. The broadcasted routing packets will undoubtedly aggravate the net congestion and will deteriorate the overall system performance.

In order to solve the above-mentioned problem, we design a new mechanism with the name of $\text{R}^2\text{DPD}$ (Random Route Discovery Packet Drop) in the 802.11 MAC layer. The design philosophy is that routing packets should try to avoid passing those congested areas in the course of routing. The overall protocol architecture is depicted in the figure 4.

As is shown in the figure, $\text{R}^2\text{DPD}$ is based on net state detection techniques. Some previous work has been done in the field of network state detection. In [6], the authors’ algorithm is based on some statistics obtained by observing the wireless medium. The successful packet transmission of a data packet is confirmed by the successful observation of an ACK packet. The authors of [7] provide each station with a collision counter that determines how many collisions in average a packet experiences before it is successfully transmitted. Since in the wireless media, we can’t definitely know whether the missing of ACK is caused by wireless media carrier sense fault or collision of data packets, the above 2 mechanisms will both affected by wireless media error or hidden terminal phenomenon. However, from the
simulation results of [6] and [7], we can see that the two mechanisms are applicable in case of low carrier sense fault condition. In this paper, we also use the similar net state detection mechanism as that of [6]. We take the conservative assumption that any packet without an ACK is a collided packet. So, we detect the collision length by measuring the length of transmissions without ACK.

![Figure 4: R²DPD Protocol Detailed Implementation](image)

The implementation of the R²DPD algorithm is very simple. We need only do some modification to the processing when the broadcasted routing packets received by the IEEE 802.11 MAC entities. When we received a routing packet, we should first judge whether the net is in the congest state. If the network isn’t in the congest state, we simply send the packet to upper layer routing protocol as before; else, we calculate a packet drop probability $p$ and discard the routing packet silently in the MAC layer with this probability.

Before the calculation of $p$, we first introduce the load factor $l$ that is used to detect the wireless contention state. $l$ is defined as follows:

**Definition**: The wireless link load factor $l$ is defined as the ratio between the mean collision time length and the mean idle time length. $l_{opt}$ is the optimal $l$ value when the network achieve maximum throughput.

We have described the detailed implementation of measuring $l$ parameter in [2]. Also we derived the optimal $l$ value $l_{opt}$. We have arrived at the conclusion that $l_{opt}$ is independent of current active contending stations of the network. For more details about load factor please refer to [2].

The $p$ calculation mechanism is described below:

**Case 1**: When the measured load $l$ is in the $l_{opt}$’s neighbor domain $[0, l_{opt} + \sigma]$ ($\sigma$ is a constant, set as 0.3 in the simulation), we consider the station to be in the permitted contention state. In this case, R²DPD don’t drop routing packet (i.e., $p=0$).

**Case 2**: When the measured load $l$ is in the range of $(l_{opt} + \sigma, l_{thresh}]$ ($l_{thresh}$ is a preset threshold constant), we consider the station to be in the medium contention state. In this case,
we use the below nonlinear equation in $p$ calculation ($p_{max}$ is maximum of routing packet drop probability):

$$ p = p_{max} \cdot \text{Sat}_K \{ \frac{(l - l_{opt})}{(l_{thresh} - l_{opt})} \} $$  

(1)

Where:

$$ \text{Sat}_K \{x\} = \begin{cases} 
K, & x > K \\
0, & 0 \leq x \leq K \\
0, & x < 0 
\end{cases} $$  

(2)

Case 3: When the measured load $l$ is larger than $l_{thresh}$, we consider the station to be in the heavy contention state. In this case, we drop all later received routing discovery packets with maximum probability (i.e., $p = p_{max}$).

Using above described routing discovery packet drop mechanism, we implement intelligent routing, avert those areas with heavy contention in the course of routing or rerouting. At the same time, routing overhead is decreased for the controlled broadcast and rebroadcast of routing packets.

4. SIMULATION ANALYSIS

In order to see the performance $R^2$DPD in multi-hop topology, we adopted the topology in the following figure. The topology is 320*640 with a hot spot congest area at (120, 360) in the center. We set 100 active nodes in the congest area, i.e., 50 pairs of TCP connections. Besides, we set cross traffic that will affect the performance of the hot-spot area. The aim is to see how the random routing packet drop mechanism will improve the system performance (including hotspot area traffic, cross traffic capacity and total goodput, i.e., application layer achieved throughput) in multi-hop topology with hotspot area in the center. Important parameters are listed in the table1.
Table 1
Important Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel bit rate</td>
<td>2Mbps</td>
</tr>
<tr>
<td>Propagation model</td>
<td>FreeSpace</td>
</tr>
<tr>
<td>Communication range</td>
<td>100 meters</td>
</tr>
<tr>
<td>Estimated $l_{opt}$</td>
<td>0.85 (RTS/CTS)</td>
</tr>
<tr>
<td>$l_{thresh}$</td>
<td>3.0 – 15.0</td>
</tr>
<tr>
<td>$p_{max}$</td>
<td>0.5-1.0</td>
</tr>
<tr>
<td>$K$</td>
<td>1.0</td>
</tr>
<tr>
<td>Simulation start and stop time</td>
<td>0.0s, 60.0s</td>
</tr>
<tr>
<td>Hot Spot Traffic start and stop time</td>
<td>10.0s, 60.0s</td>
</tr>
<tr>
<td>Cross Traffic start and stop time</td>
<td>20.0s, 60.0s</td>
</tr>
<tr>
<td>Data packet length</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>DSR</td>
</tr>
</tbody>
</table>

The $l_{thresh}$ value is an important parameter that will affect the performance of the $R^2$DPD mechanism. In the simulation, we’ll have a look at how $l_{thresh}$ affects system performance. We set 40 communication pairs in the hotspot area. The goodput of the hotspot area is called hotspot traffic in this paper. Then we set 10 cross traffic, i.e., 10 FTP traffic on 10 TCP link. We first set 10 nodes at (160, 160) and another 10 nodes at (240, 480). The 10 TCP cross connections are set between the 20 nodes. In the below simulation results, we can see the performance comparison of the proposed algorithm and prior DCF algorithm. We can also observe how the $l_{thresh}$ parameter affects the system hotspot traffic, cross traffic and total system goodput.

![Figure 6: Goodput of 40 Comm. Pairs of Hotspot Area](image)

Our goodput calculation results are achieved from application layer bandwidth counter agent. The above figure 6 has shown the goodput of the 40 TCP connections in hotspot area.
It is obvious that the proposed algorithm outperforms the DCF algorithm. When $l_{thresh}=8.0$, we have the maximum average gain of about 7%.

![Figure 7: Goodput of 10 Comm. Pairs of Cross Traffic](image)

In above figure 7, we can see that $R^2$DPD can outperform DCF with proper tuned $l_{thresh}$ value. When $l_{thresh}=8.0$, 9.0, 10.0, 11.0, 12.0, $R^2$DPD achieves larger bandwidth and at the same time more stable than DCF (less standard derivation). However, if the $l_{thresh}$ value is not set to a proper value and is too small, the nodes will be too aggressive to drop routing packets and will lead to performance degradation of cross traffic.

![Figure 8: System Total Goodput Comparison](image)

Finally, figure 8 has illustrated the system overall goodput comparison of the two mechanisms. We can conclude from the figure that the new algorithm achieves best
performance when $l_{\text{thresh}}$ value ranges from 8.0 to 12.0.

5. CONCLUSION

In this paper, we have first analyzed the drawback of the interaction between existing ad hoc routing and MAC protocol. We consider that it is an important method to exploit the MAC layer contention information during the course of routing discovery in 802.11 wireless ad hoc networks. We have developed the above idea to the newly proposed R\textsuperscript{2}DPD mechanism. This new MAC mechanism filters routing discovery packets adaptively according to the current measured contention condition. We have done numerous simulation experiments in order to compare the new mechanism R\textsuperscript{2}DPD to DCF. The simulation results have shown that properly tuned R\textsuperscript{2}DPD mechanism can outperform prior DCF algorithm in all the characters investigated.

REFERENCES