

## MANET 中有利于信道重用的按需信道分配和路由协议

彭立阳,周继鹏

(暨南大学 信息科学技术学院, 广州 510632)

(pengliyang@163.com)

**摘要:**从信道重用的角度出发,设计出一种简单而有效的按需固定信道分配机制和路由的协议(CA-AODV-R)。该协议将信道分配放到路由层进行,通过在路由发现时的 RREQ 和 RREP 中携带信道信息来分配固定信道,避免了 MAC 层动态信道分配协议(如 DCA 等)需要频繁地调用信道分配算法的问题。CA-AODV-R 使用的固定信道分配算法为把数据信道按编号从小到大排列后按每 3 个划分为一个小组,同一条路由发现路径上的后继节点分配固定信道时优先在其前驱节点的固定信道所在小组内选择空闲信道。仿真结果表明,CA-AODV-R 协议相对于单信道 AODV 能够大幅度提高网络吞吐量和分组投递率并降低网络的端到端时延。

**关键词:**自组网络;多信道;信道分配;信道重用;路由协议

**中图分类号:** TP393.04; TN915.04 **文献标志码:** A

## Channel assignment scheme and routing protocol with channel reuse in multi-channel mobile Ad Hoc network

PENG Li-yang, ZHOU Ji-peng

(College of Information Science and Technology, Jinan University, Guangzhou Guangdong 510632, China)

**Abstract:** From the angle of channel reuse, a simple and effective on-demand fixed channel assignment scheme and routing protocol (CA-AODV-R) was proposed, which combined channel assignment with the classical AODV routing. By carrying channel information in RREQ and RREP/HELLO, this protocol could avoid frequently calling channel allocation algorithm and did not require modifications to IEEE 802.11. A new fixed channel assignment algorithm that divided every three data channels into a group was also proposed. The simulation results show that compared to the single-channel AODV protocol, CA-AODV-R protocol can enhance network throughput and packet delivery rate, and reduce the end-to-end delay of the network.

**Key words:** Ad Hoc network; multi-channel; channel assignment; channel reuse; routing protocol

### 0 引言

IEEE 802.11 物理层规范提供了多重无交叠信道,例如 IEEE 802.11b<sup>[1]</sup>有 3 个未交叠信道,可以同时使用信道 1、6、11 进行工作,而 IEEE 802.11a<sup>[2]</sup>有 12 个未交叠信道,但是 IEEE 802.11 介质访问控制协议并不是为操作多个信道而设计的。文献[3-4]分别提出了基于介质访问控制(Media Access Control, MAC)层的动态信道分配协议(Dynamic Channel Allocation, DCA)及 DCA-R,其基本思想是通过在两个节点之间的 RTS→CTS→RES 的控制包交换过程中携带信道信息来预约信道。文献[5-6]给出了一种将路由和信道分配结合的多信道分配机制 CA-AODV 及其扩展 E2-CA-AODV,主要思想为在路由的过程中分配信道,同时将信道分为一个控制信道和多个数据信道,控制信道用于请求发送/允许发送协议(Request To Send/Clear To Send, RTS/CTS)的交换,而数据信道用于确认字符(Acknowledge Character, ACK)和数据包的传输。

固定接口的信道最优分配问题是一个 NP 难问题<sup>[7]</sup>,因此是非常复杂的。每一种信道分配和路由协议都有其优点和自身的局限性,只有在其特定的环境和条件下才具有最优性能。如 DCA-R 要求知道两跳范围之外邻居的信道使用情况

来提高信道重用率;CA-AODV 则在信道分配时随机选择一个空闲信道,这样网络容易出现因信道用完而无空闲信道可分配的情况。

信道重用<sup>[4]</sup>,即如果同一时刻在网络中能够使用更少的信道使得更多的通信对进行无冲突通信,那么该网络信道重用率就越高,提高信道重用率可以有效减少网络中因信道被用完而无空闲信道可使用的情况的发生。基于此,本文以经典 AODV<sup>[8]</sup>路由协议为基础,提出了一种有利于提高信道重用率的信道分配算法和路由协议。该协议通过将信道分配与路由发现同步进行,与在 MAC 层进行信道分配的机制相比,能够大大减少信道分配算法调用的频率。

### 1 设计思想

将所有信道划分为一个公共控制信道(Control Channel, CC)和多个数据信道,网络中的每个节点配备两个接口、一个控制接口和一个数据接口。网络中的所有节点的控制接口都在公共控制信道 CC 上接收和发送 RREQ、RREP 及 HELLO 等路由控制消息,当路由建立好后,节点即可将数据接口切换到不同的数据信道发送数据包。

本文提出的协议在 AODV 路由发现的过程中对路由路径上的节点分配信道。该协议将所有数据信道根据编号从小

收稿日期:2010-07-26;修回日期:2010-08-30。

基金项目:广东省教育部产学研结合基金资助项目(2010B090400164);广东省自然科学基金资助项目(7005946)。

作者简介:彭立阳(1982-),男,湖南隆回人,硕士研究生,主要研究方向:无线自组网络;周继鹏(1962-),男,陕西周至人,教授,博士,主要研究方向:计算机网络、分布式系统、移动计算。

到大按每3个划分为一个小组,同一条路由发现路径上后面的节点信道分配优先在与其前驱节点分配的固定信道所在的同一个小组内进行。

该协议的数据包传输是基于接收节点的固定信道而进行的,即路由路径上的每个节点都会为自己分配一个固定信道用于接收数据,所有要给该节点发送数据包的节点都需要先将自己的数据接口切换到该固定信道才可进行发送。

同一条路由路径上相邻两节点的固定信道信息和邻居表携带在路由请求(RREQ)包和路由应答(RREP)包中进行交换,通过邻居节点间的HELLO消息进行更新,不同路由路径上的相邻节点则通过周期性地广播HELLO消息包交换固定信道信息和邻居表。

## 2 信道分配

### 2.1 信道分配策略

本文中使用的相关符号及其意义如表1所示。网络中每个节点维护一个邻居表(Neighbour Table, NT),表中记录了节点的一跳范围邻居和使用的固定信道。节点在发送RREQ、RREP或周期性的HELLO消息时会将自己的邻居表和自己的固定信道携带在这些包中。当一个节点A收到另一个邻居节点B的RREQ或HELLO消息包时,就会检查该邻居节点B是否已经存在于邻居表 $NT_A$ 中,如果不存在就加入该邻居;否则就利用包中的信道信息更新 $NT_A$ 中节点B的固定信道。

表1 符号及其描述

符号	描述
CC	控制信道
DC	数据信道集合
GDC	将数据信道集合DC划分为小组后的信道集合
NDC	数据信道集合中的信道总数目
$RT_A$	网络中某个节点A维护的路由表
$NT_A$	网络中某个节点A维护的一跳范围邻居表
$NT_r$	节点收到的携带在RREQ/RREP/HELLO消息包中的邻居表
$CUT_A$	网络中某个节点A维护的信道使用表
$FC_A$	网络中某个节点A所分配的固定信道,值为-1表示未分配

每个节点同时还维护一个信道使用表(Channel Using Table, CUT),该表记录了每个信道是否已经被其两跳范围内某个邻居选择为固定信道。CUT表的构造和更新过程如下:当一个节点A收到某个邻居节点B的RREQ、RREP或HELLO消息包时,就会利用携带在这些包中的邻居表 $NT_B$ 里面的信道信息更新 $CUT_A$ 表,即如果信道 $i$ ( $i$ 为信道集中的某个信道)存在于 $NT_B$ 中,则认为信道 $i$ 已被节点A的某个邻居选择为固定信道,于是将 $CUT_A$ 中该信道的生命期加上一个时间 $T$ ;如果 $CUT_A$ 中的某个信道的生命期长时间没有被更新而导致过期,就认为该信道已经没有被节点A的所有一跳和两跳范围内邻居选择为固定信道,因此应该将其重新复位为空闲状态,以便该节点需要选择固定信道时备用。

信道分配的过程是伴随在路由发现的过程中完成的。路由请求发起节点(假设为S)在发送RREQ请求前首先检查自己是否已经拥有了一个固定信道,如果没有则从 $CUT_s$ 表中选择一个,然后广播RREQ给邻居。收到RREQ的邻居节点(假设为A)如果还没有自己的固定信道,则从 $CUT_A$ 中选择一个空闲信道作为自己的固定信道,然后再确定是回应还是转发该RREQ。如果在分配信道的时候出现了信道冲突,则将冲突记录在建立的逆向路由中,待路由回应时将该冲突信

息携带在RREP中反馈给冲突节点以便重新选择固定信道。以上信道分配过程用流程图表示如图1所示。

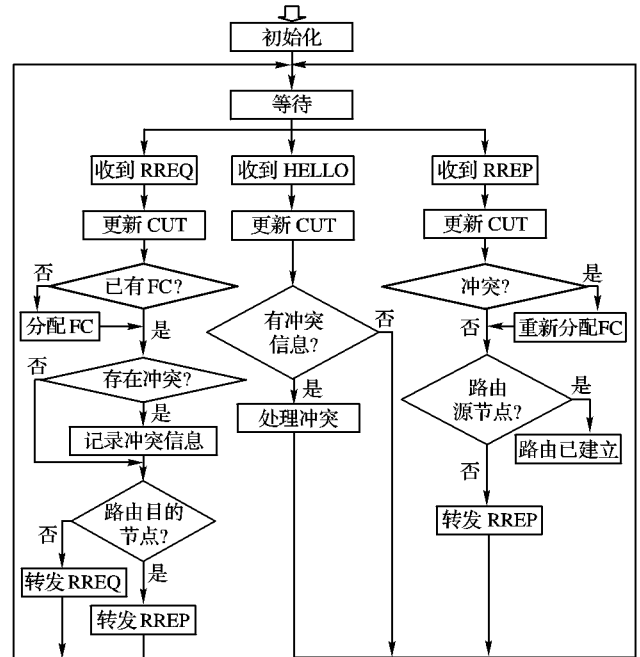


图1 信道分配流程

### 2.2 信道重用模式

对于网络中单独的一条路由路径,至少需要3个数据信道就可以达到信道重用效果。将所有数据信道按编号从小到大排列,然后按每3个信道为一小组划分为第0组,第1组, ..., 第 $n$ 组。路由发起的源节点从靠前的信道小组中选择一个空闲信道为自己的固定信道。同一条路由发现路径上后面的节点信道分配优先在与其前驱节点分配的固定信道所在的同一个小组内进行,当在同一个小组内找不到空闲信道时就返回到前面的信道小组随机选择空闲信道,如果仍然找不到空闲信道才去后面的信道小组选择一个空闲信道。这样就能让更多后面的信道空闲出来备用,从而有利于信道重用率的提高。若所有数据信道集合为 $DC = \{1, 2, 3, \dots\}$ ,则按照上述方法划分后的数据信道集为:

$$GDC = \{\{0, 1, 2\}, \{3, 4, 5\}, \{6, 7, 8\}, \dots\}$$

假设一条路由发现路径上的第 $i$ 个节点分配的固定信道为 $channel1$ 时,则第 $i+1$ 个节点优先在信道 $channel1$ 所在的信道小组 $group1$ 中选择一个空闲信道。如果该小组内无空闲信道,则在第1组至第 $group1-1$ 组内随机选择空闲信道 $channel2$ 。如果仍找不到就去第 $group1+1$ 组选择。第 $i+2$ 个节点则应该优先去信道 $channel2$ 所在小组选择空闲信道而不是在信道 $channel1$ 所在小组选择,后面的节点依次类推。

### 2.3 信道选择算法

路由发现路径上的节点在已知其前驱节点的固定信道的情况下给自己分配固定信道的算法如下:

```

/* 寻找空闲信道, preNodeChannel 为前驱节点所用信道 */
get_Free_Fixed_Channel(preNodeChannel) {
    // 计算前驱节点信道所在组号 groupNo
    groupNo = getGroupNo(preNodeChannel);
    // 尝试在第 groupNo 组寻找空闲信道
    for (int i = 1; i <= 3; i++) {
        nextChannel = getNextChannelInGroup();
        if (-1 == channel_usage_list[nextChannel]) {
            findChannel = nextChannel;
        }
    }
    // END FOR
}
  
```

```

// 没有找到,去前面的信道随机选择空闲信道
if ( -1 == findChannel ) {
    findChannel = RandomFreeChannel(1, groupNo * 3);
}

// 仍未找到空闲信道,继续去后面的小组中寻找
if ( -1 == findChannel ) {
    for (int i = (groupNo + 1) * 3; i <= NDC; i++) {
        if ( -1 == channel_usage_list[i] )
            { findChannel = i; }
    }
} // END FOR
} // END IF
return findChannel;
}

```

### 3 协议描述

#### 3.1 路由表(RT)结构

在标准 AODV 路由表的基础上再添加一个下一跳节点分配的固定信道字段和信道冲突标记字段,这样修改后的路由表包含的主要信息为:目的节点(Dest)、序列号(Seq)、下一跳(NextHop)、跳数(Hop)、路由生存时间(LifeTime)、下一跳使用的固定信道(NextHopChannel)、信道分配冲突标记(ConflictFlag)。

#### 3.2 邻居表(NT)结构

每个节点除了维护一个路由表(RT)外,还维护一个邻居表 NT。在节点的邻居表中记录了该节点所有的一跳邻居和使用的固定信道,节点在收到邻居的 Hello 消息后更新自己的邻居表。

#### 3.3 信道使用表(CUT)结构

每个节点维护一个 CUT,在该表中记录了每个数据信道的生存时间 ChannleExpireTime,当节点收到邻居的 HELLO 消息时就会根据收到的固定信道和邻居表 NT<sub>i</sub> 更新自己的信道使用表中相应信道的生存期,若生存时间为 -1,则表示该信道空闲可分配给该节点。

Dest	Seq	Hop	LTime	NHOp	NChannel	Conflict
------	-----	-----	-------	------	----------	----------

(a) 路由表结构

NeighborIP	FixedChannel	ExpireTime
------------	--------------	------------

(b) 邻居表结构

ChannelIndex	ChannleExpireTime
--------------	-------------------

(c) 信道使用表结构

图2 几个表的结构

#### 3.4 路由算法

图3~4分别描述了在具有5个节点、6个数据信道的网络中,源节点S发起路由请求和收到路由回应的过程。当某个节点维护的 CUT 中的某个信道已经被该节点的一跳或两跳范围邻居节点分配为固定信道时,则该节点在分配或重新分配固定信道时不能选择这个信道作为自己的固定信道,否则可能会引起冲突。在图3~4的 CUT 中以下划线标记这些信道。

##### 3.4.1 路由请求过程

1) 路由请求发起节点(如图3所示节点S)首先检查自己是否已经分配了固定信道,如果还没有分配固定信道则先在自己的信道使用表(CUT<sub>s</sub>)中按照信道分配算法选择一个空闲信道作为自己的固定信道同时更新该信道的 ChannleExpireTime;然后创建一个携带自己固定信道和邻居表的路由请求包(RREQ),广播给其邻居。

2) 收到 RREQ 的节点(如A)首先根据收到的 RREQ 中

的固定信道和邻居表(NT<sub>i</sub>)更新自己信道使用表(CUT<sub>A</sub>)中的相应信道的 ChannleExpireTime;然后,如果该节点还没有自己的固定信道,则先从自己的信道使用表(CUT<sub>A</sub>)中分配一个固定信道;接着,该节点建立到路由源节点(图中为S)的逆向路由,其中包含了逆向路由的下一跳的固定信道和冲突标记。

3) 如果收到 RREQ 的节点(如A)没有到目的节点(如D)的路由,则将自己的固定信道和邻居表携带在该 RREQ 中重新广播给自己的邻居;否则单播 RREP 至路由源节点。

4) 所有收到 RREQ 的节点按照2)和3)转发 RREQ 直至节点有到达目的节点的有效路由或目的节点收到 RREQ。

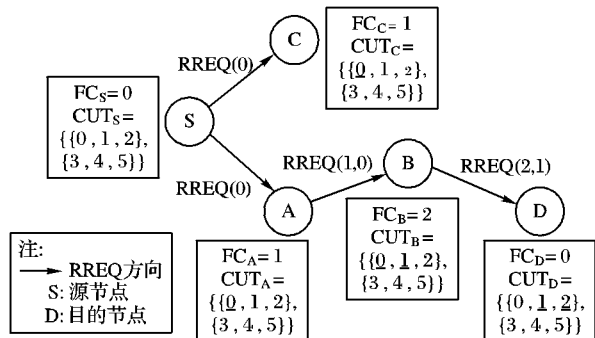


图3 路由请求

##### 3.4.2 路由回应过程

1) 目的节点(如图4所示节点D)在收到 RREQ 包后会向路由请求源节点单播一个 RREP 包,该 RREP 包中携带有自己的固定信道和邻居表以及冲突标记。

2) 收到 RREP 的中间节点(如B)首先根据收到的 RREP 中的固定信道和邻居表(NT<sub>i</sub>)更新自己信道使用表(如CUT<sub>B</sub>)中的相应信道的 ChannleExpireTime。如果 RREP 中信道冲突标记为 True,则重新分配一个可用信道作为自己的固定信道;然后,该节点建立正向路由,并将自己的固定信道和邻居表携带在该 RREP 包中继续转发。

3) 重复2)直至路由请求源节点(如S)收到该 RREP 包,并创建到目的节点(如D)的正向路由。

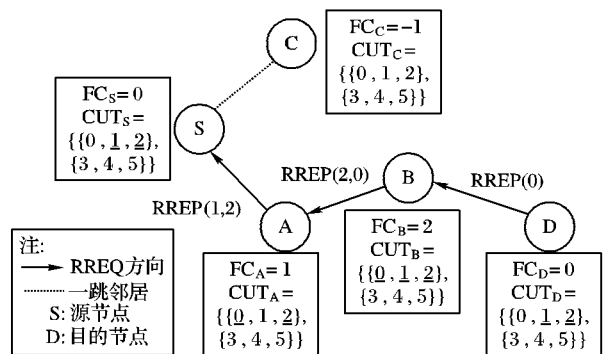


图4 路由应答

至此,整个路由发现过程已经完成。路由请求路径上的某个节点如果在一段时间内没有收到针对该 RREQ 包的 RREP 包,对应的路由表项将被删除,该节点分配的固定信道将会因为生存时间过期而复位成 -1。

### 4 冲突处理机制

冲突主要存在于以下两种情况:

1) 当一条待建路由上的节点A向一条已经建立的活跃路由上的节点B发送 RREQ 请求,则在这两条路由的交叉点发生冲突。

2) 分别位于两条活跃路由上的两个相邻节点通过 Hello

消息获知信道冲突。

针对第1种情况,活跃路由上的节点B将会在建立逆向路由的时候将路由表项中的冲突标记 ConflictFlag 置为1。当节点B发送路由回应 RREP 时会将该信息携带在 RREP 中反馈给节点A以通知其重新分配信道。

针对第2种情况,节点在每次分配信道时即生成一个在区间  $[1, M]$  的随机数 NodeNumber ( $M \gg 1$ )<sup>[5]</sup>。节点将自己分配到的固定信道和邻居表以及该随机数携带在 Hello 消息中周期性地发送给自己的所有邻居节点。收到 Hello 消息包的节点如果是非活跃路由路径上的节点则只需更新自己的 NT 和 CUT; 否则,还要将自己的 NodeNumber 与收到的 NodeNumber 比较大小,如果自己的 NodeNumber 较大则需要重新分配一个固定信道给自己。

## 5 仿真实验

NS2<sup>[9]</sup>是一种面向对象的、离散的、事件驱动的开源网络仿真工具,其组件的主要功能使用C++编写,运算速度快,而网络参数的配置则用 OTcl 编写,修改极其方便。本章通过使用 NS2 网络仿真平台并采用 Ramon 教授的 NS2 多信道扩展方案<sup>[10]</sup>对 CA-AODV-R 算法性能进行模拟实验,并与基于 802.11 MAC 的单信道 AODV 进行了比较。实验主要测试了网络中的每条 CBR 流以不同的速率发送数据包时总的吞吐量、平均端到端时延以及分组投递率3个性能指标。

### 5.1 模拟环境设置

假设在  $1200\text{ m} \times 1200\text{ m}$  的平面区域内随机分布了50个节点,节点使用 IEEE 802.11 DCF 介质访问控制协议,单个节点的传输半径为 250 m,每个包的大小为 512 B。网络随机生成4条 CBR 流,将每条流的负载分别从 40 Kbps 到 1000 Kbps 进行模拟,每次模拟时间为 300 s。

### 5.2 仿真结果与分析

从图5可以看出,使用 AODV 的网络总的吞吐量在 0.8 Mbps 以下,而采用本文提出的 CA-AODV-R 协议当信道数为12个时,网络总的吞吐量接近于单信道 AODV 的吞吐量的2~3倍,在每条 CBR 流的发送速率为 800 Kbps 时达到最大值约为 1.85 Mbps,随后随着发送速率的继续提高吞吐量开始下降。

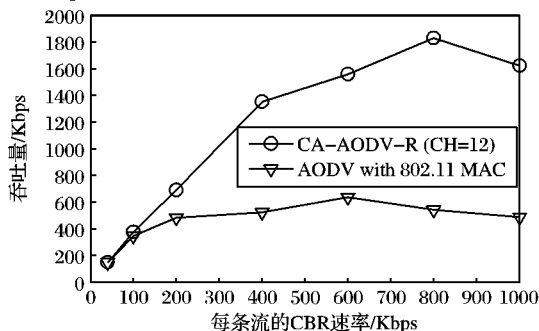


图5 网络总的吞吐量

图6~7分别给出了包的平均端到端时延和分组投递率的模拟结果,由图可见,CA-AODV-R 的端到端时延要比 AODV 低得多,且随着 CBR 流的速率的提高,延时上升的速度比相应的 AODV 的上升速度要慢,而使用 CA-AODV-R 协议的分组投递率也比使用 AODV 有明显地提高。

## 6 结语

本文提出了一种有效地适用于 Ad Hoc 网络的按需多信道路由协议 CA-AODV-R,从信道重用的角度在路由发现时分配信道,通过将数据信道按每3个划分为一个小组,其信道分

配算法简单而高效,且不需要改动 IEEE802.11 信道接入子层 (MAC) 协议。仿真结果表明 CA-AODV-R 协议与标准 AODV 比较,明显提高了网络总的吞吐量和分组的投递率,降低了网络的平均端到端时延。

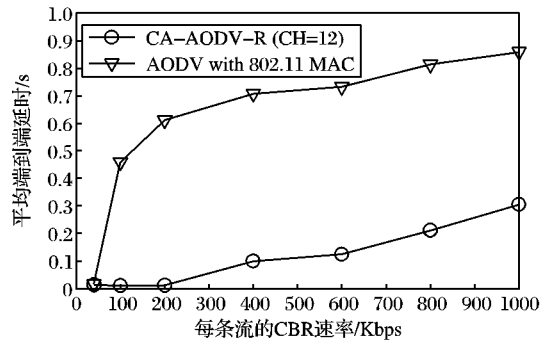


图6 平均端到端时延

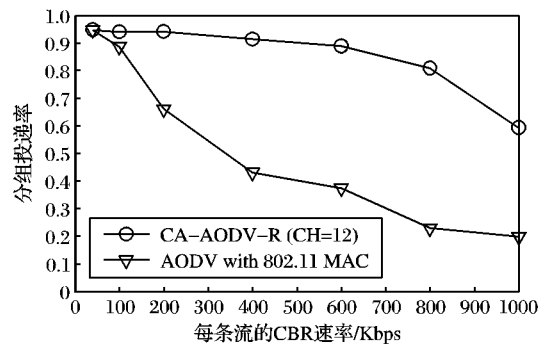


图7 分组投递率

### 参考文献:

- [1] IEEE Standard 802.11. Part 11: IEEE 802.11b wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed physical layer extension in the 2.4GHz band [S]. 2007.
- [2] IEEE 802.11a Working Group. Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications amendment 1: High-speed Physical Layer in the 5GHz band [S]. 1999.
- [3] WU S-L, LIN C-Y, TSENG Y-C, et al. A novel MAC protocol with on-demand channel assignment for multi-hop mobile Ad Hoc networks [J]. Electrical Engineering, 2004, 11(4): 361-373.
- [4] WU S-L, YANG J-Y. A novel channel assignment scheme for improving channel reuse efficiency in multi-channel Ad Hoc wireless networks [J]. Computer Communications, 2007, 30(17): 3416-3424.
- [5] GONG M X, MIDKIFF S F, MAO S. On-demand routing and channel assignment in multi-channel mobile Ad Hoc networks [J]. Ad Hoc Network, 2009, 7(1): 63-78.
- [6] GONG M X. Improving the capacity in wireless Ad Hoc networks through multiple channel operation: design principles and protocols [EB/OL]. [2010-05-10]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.99.2019&rep=rep1&type=pdf>.
- [7] 毕坤, 顾乃杰, 任开新, 等. 混合式无线 mesh 网络中信道分配算法研究 [J]. 小型微型计算机系统, 2009, 30(5): 812-816.
- [8] RFC 3561, Ad Hoc on-demand distance vector (AODV) routing [S]. 2008.
- [9] NS-2 [EB/OL]. [2010-05-10]. <http://nsnam.isi.edu/nsnam/index.php>.
- [10] Adding multiple interface support in NS-2 [EB/OL]. [2010-05-10]. <http://personales.unican.es/aguerocr/files/ucMultifaces-Support.pdf>.