

支持 QoS 保障的 Ad hoc 网络多信道 MAC 协议

张香芝, 沈明玉

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘要: 提出了一种提供 QoS 保障的多信道 MAC 协议, 该协议把不同种类业务划分优先级和帧间隔时间, 以保障优先级高的数据较早接入信道, 同时把高层数据按目的地址的不同建立发送链表, 采用自动重传技术, 实现数据信道上连续的帧交换过程。仿真结果表明, 该方法大大提高了数据信道利用率, 改善了系统性能。

关键词: 无线自组织网络; 服务质量; 介质接入控制协议; 多信道

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2010)15-0043-04

Multi-channel MAC protocol with QoS guarantee for Ad hoc networks

ZHANG Xiang-Zhi, SHEN Ming-Yu

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: A multi-channel protocol which provided QoS guarantee is presented. This protocol designate data from priority and frame interval, to make sure data of high pritrity can access channel earlier. Lnk tables of the high-level data for sending are established by destination address. Automatic re-transmission technology is used to realize the continual frame exchange process on the data channel. Simulation results show that the presented method can greatly improve the data channel utilization and performance of the system.

Key words: Ad hoc; QoS; medium access control protocol; multi-channel

无线自组织网络(Ad hoc)是一种具有无中心、自组织、快速展开和移动等特点的对等式网络, 又被称作多跳网络(Multi-hop Network)或自组织网络(Self-organized Network)^[1]。随着自组网业务的多样化以及逐渐向公用网络领域渗透, 服务质量 QoS(Quality of Service)保障问题越来越重要, 介质接入控制(MAC)协议是自组网 QoS 体系中最基本的部分, 主要用来管理和协调多个用户共享可用的频谱资源, QoS 路由和信令都要依赖它。MAC 需要解决两个问题^[2]: 如何将频谱划分为不同的信道, 如何将信道资源分配给不同的用户。在 Ad hoc 网络 MAC 协议领域, 多信道技术成为研究热点, 很多的多信道 MAC 协议能较好地解决信道争用、隐藏终端、暴露终端^[3-4], 与单信道技术相比, 其具有提高系统吞吐量、降低时延等优点^[3-5], 但是大部分不能为实时业务提供 QoS 保障。本文结合多信道 DCA^[5]协议信道预约的思想, 给不同种类的业务划分优先级和帧间隔, 保证优先级高的业务较

早地预约数据信道, 同时把高层数据按目的地址的不同建立发送链表, 实现数据信道上连续的帧交换, 改善了系统吞吐量和时延特性。

1 协议基本思想

把整个信道分为 1 个控制信道和 n 个数据信道, 这些子信道互不重叠且带宽相等, 每个节点配置两部半双工收发机, 1 个用于控制信道, 1 个可在 n 个数据信道间切换使用。通信双方通过在控制信道上交换 RTS/CTS/RES 帧来预约数据信道, 然后在数据信道上交换 DATA/ACK 帧进行通信。每个节点(如 A)保存信道使用列表 CUL[] 和空闲信道列表 FCL。CUL[] 表项有 3 个元素: CUL[i].host 是 A 的一个邻居地址, CUL[i].ch 是被邻居 CUL[i].host 占用的数据信道, CUL[i].rel_time 指 CUL[i].ch 信道的释放时间。FCL 表可由 CUL[] 计算得出。

建立发送链表, 采用自动重传技术, 双方可实现数据信道上连续的 DATA/ACK 帧交换过程, 直到通信完成

《微型机与应用》2010 年 第 29 卷 第 15 期

网络与通信 Network and Communication

或者达到预定通信时间。下面说明协议中的几个重要规则。协议中用到的标识符含义如表 1 所示。

表 1 协议中用到的标识符含义

标识符	含义
MAX T	数据信道的一次最大通信时间
T ifs	帧间隔
T rts	RTS 帧的传输时延
T cts	CTS 帧的传输时延
T res	RES 帧的传输时延
T_ack	ACK 帧的传输时延
T pro	最大传播时延
T curr	当前时间
NAVcts	CTS 帧中的网络分配向量
NAVres	RES 帧中的网络分配向量
NAVrts	RTS 帧中的网络分配向量
CUL[]	信道使用列表
FCL[]	空闲信道列表

1.1 发送链表

按目的地址的不同把高层数据加入到相应链表中，发送数据链表的结构形式如图 1 所示。

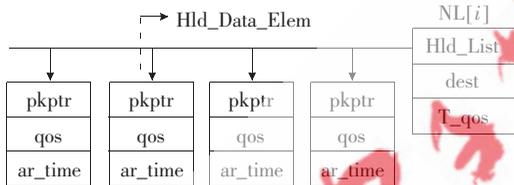


图 1 发送链表结构图

Hld_Data_Elem 是高层数据的表示结构，pkptr 指向高层数据，qos 是描述分组优先级的整数，实时(语音、视频等)分组比数据分组优先级高，ar_time 是高层数据的到达时间。NL[] 是链表头结构，具有同一目的地址的分组均放在此链表中，Hld_List 是链表头指针，dest 是目的地址，T_qos 是表示此链表中所有分组的平均优先级，由链表中所有分组的 qos 和 ar_time 值决定：

$$T_{qos} = \frac{\sum(Q \times qos + T \times (cur_time - ar_time))}{data_num}$$

Q、T 分别为 qos 和 ar_time 的比例度量值，data_num 是当前链表中分组个数，T_qos 值由链表中所有分组 qos 和 ar_time 值确定且实时更新。

1.2 控制信道规程

在控制信道上交换 RTS/CTS/RES 帧预约数据信道，RTS/CTS/RES 帧格式如图 2 所示。

RTS 帧格式	Dest	NAVrts	Fcl	N	Tdl	Ndl
CTS 帧格式	Dest	NAVcts(Test)	Di			
RES 帧格式	Dest	NAVres	Di			

图 2 RTS/CTS/RES 帧格式

RTS 帧中的 N 域表示发送链表中的高层数据个数；Tdl(Total data length)域是 N 个高层数据的总长度，目的节点可以结合 N、Tdl 域计算数据信道上的总通信时间。Ndl(next data length)域是节点 A 当前发送链表中下一个要发送的数据长度，用于节点 B 设置超时定时器。

要在数据信道上实现连续的帧交换功能，则在控制帧交换过程中，网络分配向量 NAVcts 必须有效地预测数据信道 Dj 上的通信时间，NAVcts 值被设为发送链表中的数据正常交换完成所需要的时间，NAVcts 的计算过程如下：

源节点的当前发送链表中有 4 条数据，在数据信道上正常交换时的传输序列如图 3 所示，由此可以推出：

$$NAV_{cts} = T_{dl_1} + N_A \times T_{ack} + 2N_A \times T_{prop}$$

其中，若帧交换过程中出现异常需要重传，则数据信道上的通信时间将会大于 NAVcts 值，这种异常情况的处理见下文。

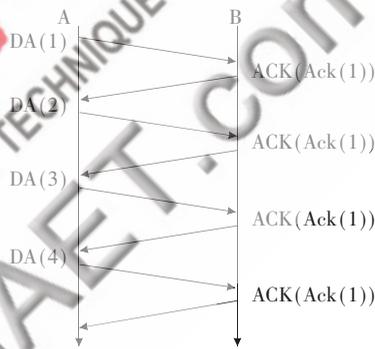


图 3 正常的数据信道帧交换图

1.3 数据信道规程

数据信道上交换 DATA/ACK 帧，DATA/ACK 帧格式如图 4 所示。

DATA 帧格式	Dest	Data	Ndl	Seq	Itrp
ACK 帧格式	Dest	Ack			

图 4 DATA/ACK 帧格式

DATA 帧中的 Seq 域用于对方节点辨别到来是否是重复帧，正常情况下此位按 0、1 交替变化，在重传时不发生变化，此位是必要的，因为节点超时定时器发生时，可能是正确的应答帧在链路上丢失造成的，这样重传时设置 Seq 位不变，对方节点便知道是重复帧而丢弃它；Itrp 域是强制中断位，正常情况下此位为 1，当为 0 时，表示要求立即停止通信，之后双方交换 ACK 帧结束本次通信。

系统记录数据信道上的通信时间，当发生重传时，源节点在发送本条数据时，计算发送链表中下一条数据正常传输结束的时刻会不会超过 NAVcts 值表示的时刻，如果超过，则把 Itrp 位置 0，强制结束通信，如图 5 中所示，目的节点收到 DATA(Itrp(0)) 时，返回 ACK 帧，之后结束通信。A 中未传完的数据依然留在发送链表

网络与通信 Network and Communication

中。因此, Itrp 位保证了发生错误重传时, 数据信道上总的通信时间不大于 NAV_{cts} 表示的时间, 既充分利用了信道资源, 又及时释放信道不至于造成冲突。

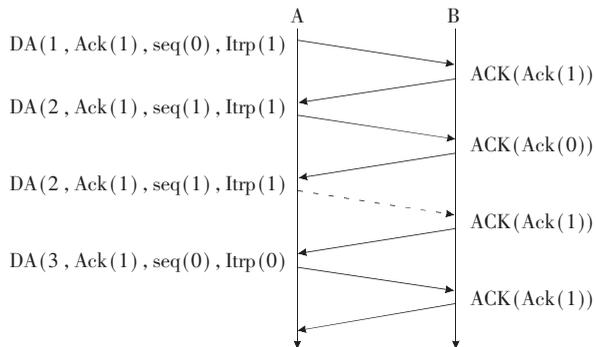


图5 有重传的数据信道帧交换图

2 本文协议的描述过程

本文协议的描述过程如下, 其分组交换时序图如图6所示。

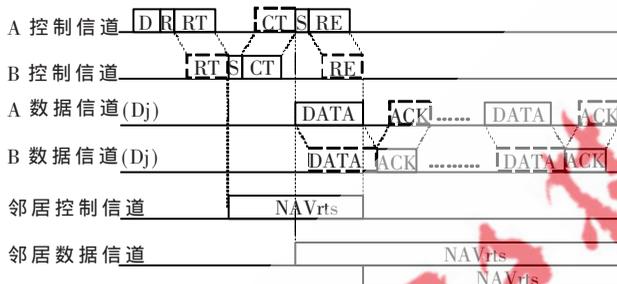


图6 分组交换时序图

(1) 节点(如 A)发送链表不全为空时, 选择 T_{qos} 权值最大的链表作为当前发送链表, 其目的地址为节点 B, 在 RTS 帧前, 节点 A 做三项检查:

(a) 保证目的节点 B 的数据信道收发机空闲。在 CUL 表中不能有:

$$CUL[i].host=B \text{ 且 } CUL[i].rel_time > T_{curr} + (T_{rts} + T_{sifs} + T_{cts})$$

(b) 保证本节点的数据信道收发机空闲。在 CUL 表中不能有:

$$CUL[i].host=A \text{ 且 } CUL[i].rel_time > T_{curr} + (T_{rts} + T_{sifs} + T_{cts})$$

(c) 保证本节点 A 有空闲的数据信道。至少一数据信道 D_j 满足:

$$CUL[i].ch=D_j \text{ 且 } CUL[i].rel_time \leq T_{curr} + (T_{rts} + T_{sifs} + T_{cts})$$

再把满足条件的信道记入 FCL 表, 设置好 NAV_{rts}、Fcl、N、Tdl、Ndl 等域, 向 B 发送 RTS 帧。

(2) 收到 RTS 帧后, B 检查是否有匹配的空闲数据信道, 能否满足 FCL_A 与 FCL_B 有匹配项或者对 $D_j \in FCL_A$ 有: $CUL[i].ch=D_j$ 且 $CUL[i].rel_time \leq T_{curr} + T_{cts}$ 存在。

(a) 若满足, 选择一个空闲数据信道(如 D_j), 设置

NAV_{cts} 等域, 向 A 返回 NAV_{cts}(D_j , NAV_{cts}) 帧。然后把数据信道收发机切换到 D_j , 准备接收 DATA 帧。

(b) 若不满足, 返回 CTS(0, T_{est}) 帧, 其中 T_{est} 是 B 节点有空闲信道的最小估计时间。

(3) 当非目的节点收到 RTS 帧时, 控制信道上执行退避, 以避免控制信道上发生冲突, 退避时间为:

$$NAV_{rts} = T_{cts} + T_{res} + 2 \times T_{sifs} + 2 \times T_{prop}$$

(4) 收到 B 的 CTS(D_j , NAV_{cts}) 后, 节点 A 执行:

(a) 向 CUL 表中增加一表项:

$$\begin{aligned} CUL[i].host &= B; \\ CUL[i].ch &= D_j; \\ CUL[i].rel_time &= T_{curr} + NAV_{cts}; \end{aligned}$$

(b) 把数据信道收发机切换到 D_j 上发送 DATA 帧, 发送完设置超时定时器。

(c) 发送广播帧 RES(D_j , NAV_{res}), 其中:

$$NAV_{res} = NAV_{cts} - T_{sifs} - T_{res} - T_{prop}$$

若收到的是 CTS(T_{est}) 帧, 则 A 退避 T_{est} 时间重新发送 RTS 帧。

(5) 非源节点 C 收到 B 返回的 CTS(D_j , NAV_{cts}) 帧时:

(a) 向 CUL 表中增加一表项:

$$\begin{aligned} CUL[i].host &= B; \\ CUL[i].ch &= D_j; \\ CUL[i].rel_time &= T_{curr} + NAV_{cts} \end{aligned}$$

若收到的是 CTS(T_{est}), 不做任何处理。

(6) 非目的节点收到来自 A 的广播帧 RES(D_j , NAV_{res}) 时, 向 CUL 表中增加一项:

$$\begin{aligned} CUL[i].host &= B; \\ CUL[i].ch &= D_j; \\ CUL[i].rel_time &= T_{curr} + NAV_{res} \end{aligned}$$

(7) 当收到 A 的 DATA 帧时, B 返回 ACK 帧:

(a) 检查 DATA 帧的 Itrp 位是否为 0, 若是, 则回复 ACK 帧后终止通信。

(b) 检查 DATA 帧的 Seq 位, 确认是否为重发帧, 再根据情况决定是否丢弃 DATA 帧。

(c) 根据 DATA 帧的 Ndl 域值, 在回复 ACK 帧后设置超时定时器, 若 Ndl 值为 0, 表示源节点数据发送完毕, 回复 ACK 后结束通信。若 DATA 帧错误或定时器超时回复 ACK(Ack(0)) 以示重传。

(8) 当收到节点 B 的 ACK 帧时, A 回 DATA 帧:

(a) 若 ACK 帧中 Ack 位为 0, 则重传上一条 DATA 帧, 其中 Seq 位保持不变。

(b) 计算下一条数据传输完成后立即结束本次通信时刻会不会超过 NAV_{cts} 值表示的时刻, 若超过, 回复 DATA 帧中 Itrp 位置 1, 就此终止通信。

若 ACK 帧错误或者定时器超时, 则重传上一条 DATA 帧。

本协议最大的优点在于建立发送链表, 结合自动重传请求技术, 实现连续的帧交换过程, 减少了控制帧

《微型机与应用》2010 年 第 29 卷 第 15 期

交换次数,既增加数据信道上的通信时间,又减少了控制信道上的冲突。

4 仿真结果分析

在相同的场景中,从吞吐量、分组平均时延两方面比较本文协议和 DCA 协议的性能。仿真条件:在 $3\text{ km} \times 3\text{ km}$ 的范围内放置 50 个节点,最大通信距离为 300 m,仿真时间为 400 s,分组长度 1 024 B,发包率服从 Poisson 分布,其中实时业务分组随机产生,数据分组帧间隔 $50\ \mu\text{s}$,实时分组帧间隔 $20\ \mu\text{s}$ 。仿真结果对比如图 7、图 8 所示。

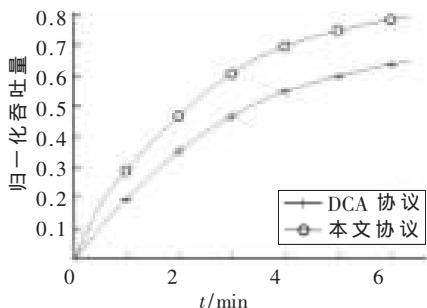


图 7 本文协议与 DCA 协议的吞吐量比较

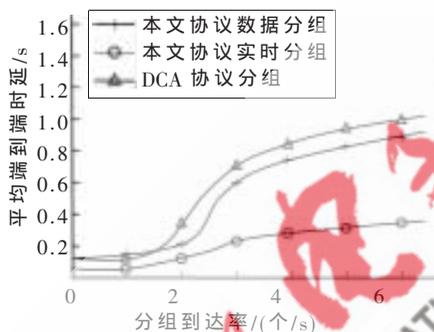


图 8 本文协议与 DCA 协议的分组时延比较

DCA 协议中,每条数据发送前需要进行一次信道预约,而本文协议实现了数据信道上的连续帧交换过程,一次信道预约可以完成多个分组交换,既大大避免了控制信道上的冲突,又提高了数据信道上的平均通信时

间,系统的吞吐量和平均时延特性得到明显改善。同时,本文协议给实时业务规定较高的优先级和较小的帧间隔时间,可以保证先于普通数据分组接入信道,平均时延显著低于数据分组,因此可以从一定程度上为实时业务提供 QoS 保障。

本文提供了一种支持 QoS 保障的 Ad hoc 网络多信道 MAC 协议,给不同类型的业务划分不同的优先级和帧间隔,以保证高优先级业务较早地接入信道。同时把高层数据按其目的地址的不同建立发送链表。采用自动重传技术,实现了数据信道上连续的帧交换过程,大大减少了控制信道上的冲突,提高了数据信道上的平均通信时间,系统吞吐量和时延特性得到明显改善。

参考文献

- [1] 郑少仁,王海涛,赵志峰,等. Ad hoc 网络技术[M]. 北京:人民邮电出版社,2005.
- [2] RAMANATHAN R, REDI J. A brief overview of Ad hoc network: challenges and directions [J]. IEEE Communication Magazine, 2002,40(5):20-22.
- [3] SO J, VAIDYA N. A multi-channel MAC for Ad hoc wireless networks [J]. Dept. of Electrical and Computer Engineering, 2003(1):254-257.
- [4] JIN K T, CHO D H. Multi-code MAC for multi-hop wireless ad hoc network [C]. Vehicular Technology Conference 2002 Proceedings, USA. 2002,2:1100-1104.
- [5] WU Shih-Lin, TSENG Yu-Chee, LIN Chih-Yu, et al. A new multi-channel MAC protocol with on-demand channel assignment for mobile Ad hoc networks [C]. Dallas: in Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks(I-SPAN), 2000: 232-237.

(收稿日期:2010-03-03)

作者简介:

张香芝,女,1983年生,硕士研究生,主要研究方向:Ad hoc 网络中信道接入协议的研究。

沈明玉,男,1962年生,副教授,硕士生导师,主要研究方向:Ad hoc 网络和无线传感器网络的研究。