

无线传感网多信道的 MAC 层协议

司宏林^{1,2}, 王晓蔚¹

(1. 东南大学 计算机科学与工程系, 江苏 南京 210096;

2. 安徽电气工程学校, 安徽 合肥 230031)

摘要:无线传感网(Wireless Sensor Networks, WSNs)作为计算机、通信和传感器三项技术相结合的产物,已成为计算机与通信领域一个活跃的研究分支。信道接入协议是无线传感网协议的重要组成部分,基于多信道的信道接入协议具有一些特殊的优点。分析了无线传感网多信道的 MAC(Media Access Control)协议:DCA-PC(Dynamic Channel Assignment with Power Control),并与其他的多信道协议作了比较。

关键词:无线传感网;MAC层协议;多信道;功率控制

中图分类号: TN915.04

文献标识码: A

文章编号: 1005-3751(2006)03-0232-03

MAC Protocols of Multi-Channel Based on Wireless Sensor Networks

SI Hong-lin^{1,2}, WANG Xiao-wei¹

(1. Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China;

2. Anhui Electrical Engineering School, Hefei 230031, China)

Abstract: As a result of combination of computing, communication and sensor technology, WSNs have become an active branch of computer science and communication. Channel access protocol is an important part of WSNs protocol, and multi-channel based on channel access protocol has its special advantages. DCA-PC, one of such protocols, is emphatically analyzed and compared with other typical protocols.

Key words: wireless sensor networks; MAC protocol; multi-channel; power control

0 引言

信道接入协议对网络的性能起着决定性的作用,是 WSNs 技术的研究重点之一。单信道的信道接入协议在节点规模增大时,网络性能明显变差。采用多信道机制,可以提高网络吞吐量,减小传播延时,降低冲突的概率,更易于支持网络的 QoS(Quality of Service)保障。多信道协议主要应解决两个问题^[1]:信道分配和接入控制。信道分配是要为不同的通信节点分配相应的信道;接入控制是确定节点接入信道的时机,解决好竞争和冲突的避免问题。文中介绍 3 种多信道的信道接入协议:多信道的 CSMA(Carrier Sense Multiple Access),FAMAC(Frequency Assignment based multi-channel Multiple Access Control),DCA-PC,其中 DCA-PC 是文中的重点。

1 基于 CSMA 和 FAMAC 的多信道 MAC 层协议

1.1 多信道 CSMA

多信道 CSMA 是一种基于载波监听的信道接入协

议^[2]。其设计目标是通过使用多信道来减少隐终端问题的影响,减少数据报文的冲突。它采用准信道预留技术,通过分布式的载波监听来对多个信道进行分配。

准信道预留技术的原理^[2]是:当节点发送报文时,它优先选择上次使用过的信道;如果该信道忙,就通过载波监听随机选择一个空闲信道发送数据。

多信道 CSMA 将可用信道分为 N 个不重叠的信道,一般而言 N 要小于无线传感网网络中节点的数目。每个子信道的带宽是整个信道带宽的 $1/N$ 。

1.2 FAMAC 协议

FAMAC^[3]是一种基于频率分配的多信道接入协议,设置一个控制信道和多个数据信道。实现时,所有节点使用一个固定的频率作为控制信道,完成 RTS(Request To Send)和 CTS(Clear To Send)的交互。处于空闲状态的节点驻留在控制信道上。另外,为每个节点分配一个不同的频率,作为该节点的数据信道值。信道选择时,发送者向接受者发送的 RTS 中,携带自己的频率信息。接收者收到 RTS 后,记录发送者的频率,回送 CTS,将电台切换到发送者的频率上,接收数据。

以上两种协议都是针对多信道设计的,较好地解决了接入控制和信道选择问题,隐终端和暴露终端的影响也得到了很好的控制。由于 WSNs 的节点通常运行在人无法

收稿日期:2005-07-14

基金项目:江苏省自然科学基金资助项目(BK2004067)

作者简介:司宏林(1969—),女,安徽合肥人,硕士研究生,研究方向为无线传感网络;王晓蔚,副教授,硕士生导师,研究方向为计算机系统结构。

接近的恶劣甚至危险的远程环境中,能源无法代替,难以补充,所以能量受限的约束条件要求其 MAC 协议首先要关注的就是能量效率,要尽可能地节约能源,而上述两种协议对此未作足够的考虑。DCA-PC 协议从节约能耗的观点出发,在 MAC 协议设计中探索了集成功率控制和多信道接入两种机制的可能性。

2 采用功率控制的多信道协议 DCA-PC

DCA-PC 协议首先在 WSNs 的 MAC 层设计中将功率控制的概念与多信道接入结合起来,节约能耗,减少邻居节点共用信道时的相互干扰,提高信道复用度。

2.1 信道模型

带宽被划分为一个控制信道和 n 个数据信道 D_1, D_2, \dots, D_n 。控制信道用于控制报文的传送,目标是为节点合理分配数据信道,避免冲突;数据信道用于数据报文和 ACK(Acknowledgement)报文的传送。

从避免冲突,提高信道预约成功率的角度出发,控制报文发送采用最大功率;为了降低能耗,提高信道复用度,将数据报文的发送功率划分为大小不同的若干等级,通过 RTS-CTS 握手,可计算出双方通信必需的最小功率,数据报文发送采用最接近最小发送功率的那个功率等级。

为实现动态信道分配和功率控制,每个节点保存 3 个数组。以节点 A 为例来说明:

* Power[i] 功率列表:节点 A 向节点 i 发送数据报文时应采用的功率等级,可利用功率控制的原理^[4]计算得出。

* CUL(Channel Usage List)[i] 信道使用状况列表: A 节点获悉的已用信道列表。CUL[i] 有 4 个域: CUL[i]. host 记录节点 A 的邻居节点的主机; CUL[i]. ch 记录 CUL[i]. host 占用的数据信道; CUL[i]. rel-time 表示释放 CUL[i]. ch 数据信道的时间; CUL[i]. int 记录 CUL[i]. host 发送的信号是否会被节点 A 听到(CUL[i]. int 值为 1 或 0)。

* FCL 空闲信道列表:节点发送数据时的可用信道列表,可根据 CUL 计算得出。

2.2 完整的协议过程

下文阐述了多信道完整的协议过程,其中表 1^[4]列出了协议表示过程中用到的常量和变量;图 1^[4]为报文交换时序图。

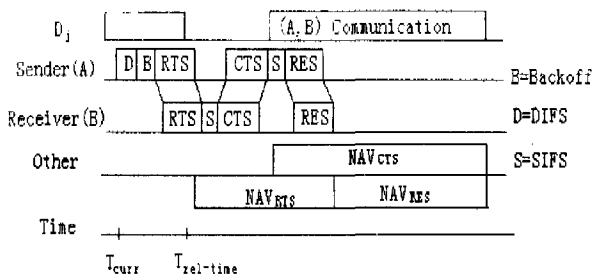


图 1 报文交换时序图

表 1 协议表示过程中用到的常量和变量

T_{SIFS}	Length of short inter-frame spacing
T_{DIFS}	Length of distributed inter-frame spacing
T_{RTS}	Time to transmit a RTS
T_{CTS}	Time to transmit a CTS
T_{RES}	Time to transmit a RES
T_{curr}	The current clock of a host
NAV_{RTS}	Network allocation vector on receiving a RTS
NAV_{CTS}	Network allocation vector on receiving a CTS
NAV_{RES}	Network allocation vector on receiving a RES
L_d	Length of a data packet
B_d	Bandwidth of a data packet
τ	Maximal propagation delay

(1) 主机 A 要向 B 传送数据报文时,首先检查下列两个条件^[4]。

a) B 应为满足下列条件的主机:

$$CUL[i].rel-time \leq T_{curr} + (T_{DIFS} + T_{RTS} + T_{SIFS} + T_{CTS})$$

否则在 RTS, CTS 报文交互之后,对应的数据信道 CUL[i].ch 仍处于忙态。

b) 对所有数据信道 CUL[i].ch,至少有一条 D_j 满足:

$$(CUL[i].ch = D_j) \rightarrow \{ CUL[i].rel-time \leq T_{curr} + (T_{DIFS} + T_{RTS} + T_{SIFS} + T_{CTS}) \} \vee \{ (CUL[i].int = 0) \wedge (Power[CUL[i].host] > Power[B]) \}$$

两条条件满足时, A 将所有满足条件 b) 的 D_j 加入到自己的 FCL 中;否则 A 将退避等待。

(2) A 向 B 发送 RTS(含有 FCL, L_d 的信息)。

(3) B 收到 RTS(FCL, L_d) 后,检查 FCL 中是否有这样的数据信道 D_j , 对所有的 i, 均满足

$$(CUL[i].ch = D_j) \rightarrow \{ CUL[i].rel-time \leq T_{curr} + (T_{SIFS} + T_{CTS}) \} \vee \{ (CUL[i].int = 0) \wedge (Power[CUL[i].host] > Power[A]) \}$$

若有,从所有满足条件的信道中选择出第一条 D_j , 向 A 回复 CTS(含有 D_j , NAV_{CTS} , P_{CTS})。这里,

$$NAV_{CTS} = L_d/B_d + T_{ACK} + 2\tau$$

$$P_{CTS} = Power[A]$$

同时 B 将数据收发器调整到 D_j , 等待接收来自 A 的数据报文。

若没有找到满足条件的 D_j , B 将向 A 回复 CTS (Test) 信号,

$$Test = \min \{ \forall i, CUL[i].rel-time \} - T_{curr} - T_{SIFS} - T_{CTS} \quad (Test \text{ 是正在使用的数据信道的最早释放时间减去 CTS 报文交换时间})$$

(4) 除 B 以外的其他主机收到 A 的 RTS(FCL, L_d) 时,要计算一个退避时间,此时间内不使用控制信道:

$$NAV_{CTS} = 2T_{SIFS} + T_{CTS} + T_{RES} + 2\tau$$

以避免冲突。

(5) A 发送 RTS 后,要经过 $T_{SIFS} + T_{CTS} + T_{RES} + 2\tau$ 时期,等待 B 的 CTS 的到来。若未收到 CTS,则退避重发。

(6) A 收到 B 的 CTS (D_j, NAV_{CTS}, P_{CTS}) 后,执行以下步骤:

a) 添加 CUL[k] 到 CUL 信道使用列表中:

$$CUL[k].host = B$$

$$CUL[k].ch = D_j$$

$$CUL[k].rel-time = T_{curr} + NAV_{CTS}$$

$$CUL[k].int = 1$$

b) 用最大功率在控制信道上广播 RES (D_j, NAV_{RES}, P_{RES}) 报文,

$$NAV_{RES} = NAV_{CTS} - T_{SIFS} - T_{RES}$$

$$P_{RES} = Power[B]$$

c) 用 Power[B] 在数据信道 D_j 上传送数据报文(步骤 b)与 c) 同步进行)。

若 A 收到的是 B 的 CTS(Test) 信号,则 A 回到步骤 1 退避等待,或者重新选择新释放的数据信道。

(7) 除 A 以外的其他主机收到 B 的 CTS (D_j, NAV_{CTS}, P_{CTS}) 时,将更新自己的 CUL,与步骤 6a) 类似。但应注意:

$$CUL[k].rel-time = T_{curr} + NAV_{CTS} + \tau$$

$$CUL[k].int = \begin{cases} 0, & \text{如果 } Power[B] > P_{CTS} \\ 1, & \text{如果 } Power[B] \leq P_{CTS} \end{cases}$$

若收到的是 B 的 CTS(Test) 信号,则忽略该报文。

(8) 若除 A 以外的其他主机收到 RES (D_j, NAV_{RES}, P_{RES}) 报文,则添加 CUL[k] 到 CUL 信道使用列表中:

$$CUL[k].host = A$$

$$CUL[k].ch = D_j$$

$$CUL[k].rel-time = T_{curr} + NAV_{RES}$$

$$CUL[k].int = \begin{cases} 0, & \text{如果 } Power[A] > P_{RES} \\ 1, & \text{如果 } Power[A] \leq P_{RES} \end{cases}$$

(上接第 231 页)

```
<forward name="failed" path="/error.jsp"/>
</action>
</action-mappings>
</struts-config>
```

4 结束语

Struts 是目前非常流行的基于 MVC 的 Java Web 框架。这种框架使系统开发过程各个模块更加细化。用户界面、业务过程、业务逻辑和业务控制的分离,使得程序开发者能很好地协调,开发过程也更简洁,同时也提高了应用软件的柔韧性。由于篇幅有限,文中实例只是简要地讲述了 Struts 的用法,要想真正掌握其开发技巧,还须进一

(9) B 收到 A 的完整数据报文后,用 Power[A] 在数据信道 D_j 上回送 ACK 报文。

3 总结

通过以上分析,能看出 DCA-PC 协议具有以下特点:首先,能按需为节点分配信道。当节点有信号需要传送时,协议通过 RTS/CTS/RES 握手控制报文取得一个信道,传送任务完成后释放该信道。其次,在这种按需分配传送的机制下,网络信道数目的分配与网络的规模,拓扑和复杂度无关。另外,不需要在全网范围内提供时钟同步。具有基于 CSMA 和 FAMAC 的多信道协议不具备的优势。现有的其他多信道协议往往只是单独考虑信道接入和功率控制的问题,而 DCA-PC 独创性地将二者结合起来。研究结果表明,使用多信道和动态信道分配技术,可以显著提高网络的吞吐量。通过在数据信道上使用功率控制技术,增加了频率的空间复用度,在提高网络吞吐量的同时,也降低了网络的能耗^[5],对改善 WSNs 的性能具有重要意义。

参考文献:

- [1] Zheng Tao, Radhakrishnan S, Sarangan V. PMAC: An adaptive energy-efficient MAC protocol for Wireless Sensor Networks [J/OL]. <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>, 2005-04.
- [2] 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.
- [3] 赵志峰, 郑少仁. 基于多信道的 Ad Hoc 网络信道接入协议性能分析[J]. 军事通信技术, 2002, 23(4): 1-5.
- [4] Tseng Yu-Chue, Wu Shin-Lin, Lin Chih-Yu, et al. A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks[J]. The Computer Journal, 2002, 45(1): 101-110.
- [5] 任丰原, 黄海宁, 林 闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282-1290.

步掌握相关知识。

参考文献:

- [1] 孙卫琴. 精通 Struts: 基于 MVC 的 Java Web 设计与开发[M]. 北京: 电子工业出版社, 2004. 9-16.
- [2] 黄若波, 程 峰, 程繁科. 实战 STRUTS[M]. 北京: 机械工业出版社, 2005. 10-12.
- [3] 丰 伟. Struts 教程[EB/OL]. <http://www.xpbook.com/soft/2779.htm>, 2004-12-16.
- [4] 孙卫琴. 创建 Web 应用和 Struts 框架的配置文件[EB/OL]. <http://linux.ccidnet.com/pub/article/c322-a181127-p1.html>, 2004-11-24.
- [5] 丁 鹏, 刘 方, 邵志峰, 等. STRUTS 技术揭秘及 WEB 开发实例[M]. 北京: 清华大学出版社, 2004. 20-28.