

一种基于双令牌的蓝牙调度算法^{*}

任秀丽^{1,2} 全成斌² 范力军² 栾贵兴³

(吉林师范大学计算机学院 四平 136000)¹ (东北大学信息科学与工程学院 沈阳 110004)²

(中国科学院沈阳计算技术研究所 沈阳 110004)³

摘 要 调度算法是短距离无线通信技术的主要问题。蓝牙网络中使用的 Round Robin (RR) 轮询调度算法严重影响了系统的性能, 本文针对这个问题提出了一种改进的蓝牙调度算法——双令牌随机(DTPR)轮询算法, 此算法有两个重要特点: 一是访问采用随机次序解决从设备之间的不公平性; 二是采用优先级调度策略, 对拥有令牌的主从设备进行调度, 缩短了轮询周期。并对本文提出的算法与现已提出的算法进行了仿真实验, 结果表明, 本文提出的算法在系统的吞吐量、传输时延和公平性方面都优于现有的其它算法。

关键词 蓝牙技术, 调度算法, 令牌, 吞吐量, 时延, 公平性

A Bluetooth Scheduling Algorithm Based on Both Tokens

REN Xiu-Li^{1,2} QUAN Cheng-Bin² FAN Li-Jun² LUAN Gui-Xing³

(School of Computer, Jilin Normal University, Siping 136000)¹

(School of Information Science & Engineering, Northeastern University, Shenyang 110004)²

(Shenyang Institute of Computing Technology, The Chinese Academy of Sciences, Shenyang 110004)³

Abstract Scheduling algorithm is a primary question in short range wireless communication. The Round Robin polling algorithm of Bluetooth significantly impacts on the system performance. In this paper Double Token Pseudo Random polling algorithm is proposed. The algorithm has two important characteristics. The first one is that all slaves are visited by the Pseudo-Random order to solve the unfairness for all slaves. The second one is that a pair of the master-slave devices holding two Tokens is firstly scheduled. The length of the polling cycles is reduced. The simulation results show that the proposed algorithm is outperformed in throughput and delay behavior and has a fairness among all slaves comparing with these practical algorithm presented in this paper.

Keywords Bluetooth technology, Scheduling algorithm, Token, Throughput, Delay, Fairness

1 引言

蓝牙技术是一种新型的无线通信技术, 工作在全球通用的 2.4GHz ISM 频段上。它使笔记本电脑、PDA、蜂窝电话和其它便携设备方便快捷地建立无线连接, 并能通过多种方式形成短距离的自组网(ad hoc 网)^[1]。

任意两个或多个激活的设备在一定的范围内建立一种特殊的连接, 形成一个蓝牙网, 即微微网(piconet)。每个微微网有 1 个主设备和最多 7 个激活的从设备。蓝牙系统采用全双工分时(TDD)传输方案实现双工传输。信道被分成长度为 625μs 的时隙, 主设备和从设备交替地进行传输。主设备在偶数时隙上进行信息传输, 称为下行链路传输; 而从设备在奇数时隙上进行信息传输, 称为上行链路传输。每个微微网对应一个特定的跳频序列, 由蓝牙主设备地址决定, 所有参与通信的设备均采用同一跳频序列。

主设备和从设备之间能够建立两种类型的连接: 同步面向连接和异步无连接。在规定间隔上, 同步面向连接使用保留时隙提供电路交换服务。异步无连接提供分组交换服务, 分组可以占用 1、3 和 5 个时隙。蓝牙使用一种快速确认和重传方案确保可信的数据传输。主设备轮询每个从设备, 在微微网内

控制业务量。一个从设备只有在主设备轮询之后才允许进行传输。

主设备管理和分配时隙。在下行链路传输的时隙, 主设备决定发送数据包还是信令包给从设备, 从设备接收到主设备发送的包之后, 才允许给主设备发送响应包, 所以从设备的传输完全受主设备传输的控制。主设备轮询哪一个从设备决定着系统的性能。因此, 在蓝牙网络中, 调度算法是一个主要问题。

文[2,3]表明, 蓝牙标准使用 Round Robin (RR) 轮询调度算法造成带宽的浪费, 所以在 MAC 层上必须设计一个有效的调度算法。针对这一问题, 本文提出了双令牌随机(DTPR)轮询算法, 此算法避免了发送 POLL 包或 NULL 包造成时隙的浪费。对已提出的算法与本文提出的算法进行了仿真, 实验结果表明, 本文提出的算法在提高系统的吞吐量、降低时延和改善公平性方面都优于其它算法。

2 相关的调度算法

在蓝牙网络中, RR 调度算法的实现就是对所有的从设备平均分配轮询次数, 对每个主从对只限传输一次, 它并没有考虑一个从设备所需要的轮询次数, 结果, 不需轮询的从设备

^{*} 本文受国家科技部科技型中小企业技术创新基金项目(01C26212110295)资助。任秀丽 博士研究生, 主要研究方向为无线网络与通信。全成斌 博士研究生, 主要研究方向为网络与通信。范力军 主要研究方向为嵌入式系统。栾贵兴 博士生导师, 主要研究方向为计算机体系结构、计算机网络与通信。

代替了需要轮询的从设备,因此,传输时隙被浪费,大大降低了系统的性能。而且它采用一种固定的轮询方式,即对从设备的轮询次序始终保持不变,这样在对所有从设备进行调度时产生了不公平性。

基于存在的这些问题,很多文献里提出了对蓝牙网络中调度算法的改进,根据这些算法的特点,可分成以下两大类。

2.1 理想化的调度算法

理想化的调度算法就是假设主设备掌握微微网的全部信息,它能有效地提高系统的性能。文[4,5]中提出 HOL 优先调度算法和 HOL K-公平调度算法,主设备根据主设备和从设备队列里的数据包的状态进行调度。如果主设备和从设备队列里都没有数据包,则这个从设备就不能被调度。

文[6]中提出了 B1 和 B2 的调度算法也是基于知道主设备和从设备的队列状态,B1 算法采用完全发送机制。当主设备与某一个从设备进行通信,只有在主设备和从设备队列里的数据包全部发送之后,才去轮询下一个从设备,但对这个从设备有一个要求就是它的主设备队列和从设备队列的长度之和是在整个微微网内的所有主从设备队列之和中取最大值。在高负载的情况下,当一个队列空,另一个队列不空时,B1 算法浪费了一些时隙。B2 算法是在 B1 算法的基础上提出的,在某种程度上,它采用的是部分完全发送机制,也就是说,当一个队列空时转向下一个主从队列,主从队列都不空时首先被访问。

在理想化的条件下,上述提出的改进算法基本上满足蓝牙网络的 QoS 要求,鉴于这些算法,提出了几种实际的调度算法。

2.2 实际的调度算法

实际的调度算法没有像上面提到的假设,它在实际工作中能够实现。文[7]中提到了 Exhaustive Round Robin(ERR)轮询算法,它是基于 RR 轮询算法提出的。这个算法的特点是当一对主从设备获得发送权时,它们就相互发送数据包,直到两个设备队列变空时,发送权才转交给下一主从对。但是,使用 ERR 轮询算法的主要缺点是信道可能被业务量大的设备占用,系统出现了不公平性。根据存在的缺点,文[6]中又提出了 Limited Round Robin(LRR)轮询算法,它是以固定的轮询次序对从设备进行调度,在每次循环内,主从队列对的传输次数都给予限制。虽然在公平性方面有所提高,但它还有不足之处,就是对业务量少的主从队将会浪费很多时隙,因为此时有可能主从队列处于空闲状态,导致双方发送大量的 POLL 包或 NULL 包,造成系统性能下降。本文基于 LRR 算法,提出了一种改进的蓝牙调度算法——双令牌随机(DTPR)轮询算法。

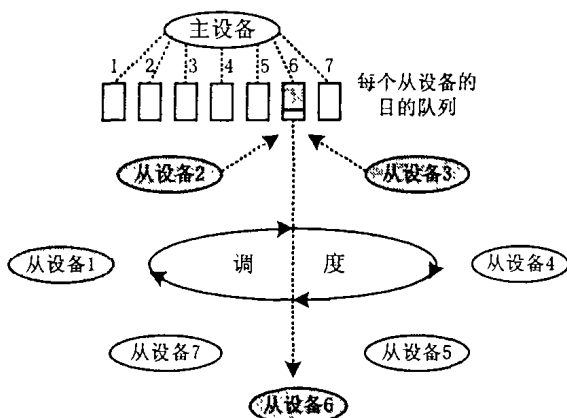


图1 蓝牙网络的拓扑结构

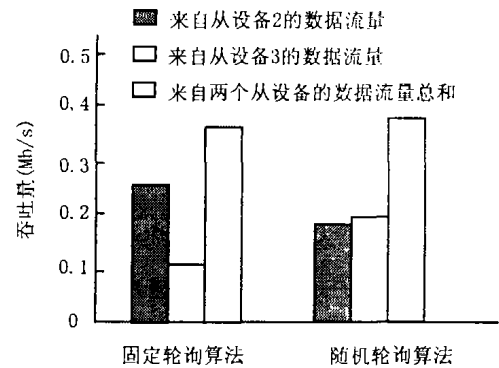


图2 不同调度算法两从设备的吞吐量

3 双令牌随机轮询算法

本文提出的调度算法有两个重要的特点,一个是通过随机次序的方式访问所有的从设备;另一个是减少对没有数据发送的从设备的访问次数。

3.1 随机轮询算法

随机轮询算法就是随机地访问所有的从设备,对系统的性能有重要的影响。假设一个蓝牙网络的拓扑结构如图1所示^[8],它由一个主设备和七个从设备组成。蓝牙网络的特点决定了两个从设备之间的通信必须经过主设备转发,也就是说,当图1中的从设备2与从设备6进行通信时,从设备2将数据发送给主设备并存储在目的队列中,然后,再把数据发送到目的从设备6。现在,若有从设备3也与从设备6进行通信,由于从设备2和从设备3共享主设备的队列,如果采用蓝牙的固定(RR)轮询算法,即从1到7这种方式,在主设备队列里,来自从设备3的数据包不仅依赖于从设备3与主设备的连接,而且还依赖于从设备2与主设备的连接。如果主设备先访问从设备2,那么主设备的队列有可能由从设备2发送的数据包充满,所以采用固定轮询算法不可能对所有的从设备给予公平性。

解决这种不公平性的方法是采用随机轮询算法(Pseudo-random RR)。此算法是使主设备采用随机方式轮询所有的从设备,即在每次轮询开始,主设备把编号1到7随机地分配给每个从设备,然后,按编号递增的顺序访问从设备。

为验证使用此方法提高了公平性,做了两个仿真实验,仿真器是基于 ns-2^[9]扩展的 BlueHoc^[10],由 IBM 研究开发的,在基带层和 L2CAP 协议层进行了仿真。为了比较固定轮询算法和随机轮询算法的性能,在第一个实验中,从设备2和从设备3与目的从设备6分别建立一个 TCP 连接,如图1所示。实验结果如图2所示,从图上可以看到,固定轮询算法对两个数据流给出不平等的吞吐量,而随机轮询算法在业务流之间几乎平分带宽。在第二个实验里,保持连接类型为 TCP,增加了一些新的连接,也就是说,由1到6的从设备向从设备7发送数据,测得的实验数据见表1。从表上的数据看,随机轮询算法比固定轮询算法具有公平性。

3.2 双令牌随机轮询算法

双令牌随机(DTPR)轮询算法是根据 LRR 算法改进的,根据发送和接收的数据包来判断下一周期是否访问该设备,减少对没有数据发送的从设备的访问次数。本算法采用随机分配令牌的方法对持有双令牌的设备优先进行访问,避免发送 POLL 包和 NULL 包,造成时隙的浪费。首先介绍本算法中涉及到的一些变量的含义:

表1 固定轮询算法和随机轮询算法的吞吐量(Mb/s)

固定轮询算法							
从设备 吞吐量 连接数	1	2	3	4	5	6	总吞 吐量
1	0.38						0.38
2	0.24	0.15					0.39
3	0.05	0.24	0.09				0.38
4	0.02	0.04	0.21	0.12			0.39
5	0.03	0.04	0.06	0.22	0.05		0.40
6	0.07	0.15	0.02	0.04	0.03	0.08	0.39

随机轮询算法								
连接数	从设备 吞吐量	1	2	3	4	5	6	总吞 吐量
	1	0.38						0.38
	2	0.19	0.20					0.39
	3	0.11	0.12	0.15				0.38
	4	0.10	0.09	0.09	0.11			0.39
	5	0.07	0.09	0.08	0.08	0.07		0.39
	6	0.07	0.06	0.06	0.08	0.06	0.07	0.40

TK:拥有双令牌的主从设备对的数值;TOK₀和 TOK₁:分别是主从队列的令牌和从主队列的令牌;i:取值为1到7(从设备的编号);0:主设备的编号;T_s:系统设置定时器变量;T_L:规定的一个时限值;T,N:分别存放每个从设备时间标志变量和计数器变量的集合;L:数据包的传输次数;Q₀和 Q₁:分别是主从队列和从主队列。算法具体描述如下:

- (1) 初始化,分配给每一对主从设备两块令牌,主从队(TOK₀)和从主队列(TOK₁)各一块。对 T,N,TK,T_s和 T_L 赋初值,即 T=N={},TK=7,L=3,T_s=0,T_L=50ms。
- (2) 每次轮询开始,伪随机数发生器产生7个伪随机数,分配给激活的从设备,主设备按编号递增的顺序对从设备进行访问,对i赋值为1。
- (3) if T_s>T_L then go to (4)
 - else { /* 访问有两块令牌的主从设备对 */
 - if TK=0 then go to (5)
 - else { if i>7 then go to (2)
 - else { /* 有三种情况 */
 - if TOK₀ and TOK₁ then {Polling (Slave_i), 传输L次, i=i+1, go to (3)}
 - if TOK₀ or TOK₁ then { /* 证明有一方队列处于空闲状态,跳过访问,检查时间标志变量 */
 - if 此设备的时间标志变量∈T then { i=i+1, go to (3)}
 - else {系统时间赋给此设备的时间标记变量,存入集合T中, i=i+1, TK=TK-1, go to (3)}
 - if (not TOK₀) and (not TOK₁) then { /* 双方队列都处于空闲状态,跳过访问,检查计数器

变量 */

if 此设备的计数器变量∈N then {此设备的计数器值加1, i=i+1, go to (3)}

else {此设备的计数器值加1, 存入集合N中, i=i+1, TK=TK-1, go to (3)} }

(4) T_s=0, 检查集合N, if N={} then go to (3)

else { /* 判断集合中各个计数器所对应主从设备对的队列情况 */

if Q₀≠NULL and Q₁≠NULL then {分配两块令牌, 并且给计数器值大的主从设备对分配优先权, TK=TK+1, go to (2)}

if Q₀≠NULL or Q₁≠NULL then {分配一块令牌, 系统时间赋给此设备的时间标志变量, 存入集合T中, go to (3)}

(5) 判断集合T, if T={} then go to (6)

else { /* 访问有一块令牌的主从设备对 */

if T_s≤T_L then {访问时间标志最小的从设备 Polling (Slave_i), 传输L次, go to (3)} else go to (4)}

(6) 结束访问所有的激活从设备。

4 性能的评价

为了评估本文提出的双令牌随机轮询算法的性能,进行了仿真实验,将此算法与上文提到的实际算法进行了仿真比较,即与RR轮询算法、ERR轮询算法和LRR轮询算法在系统的时延和吞吐量方面作了一下对比。实验过程中,采用的仿真器是前文提到的仿真器。仿真模型如图1。在仿真过程中,对算法LRR和本文提出的算法中涉及到的参数L都赋值为3,仿真时间为120s。基带层的数据包类型为DH1、DH3和DH5,它们相应的负载长度为27字节、183字节和339字节。

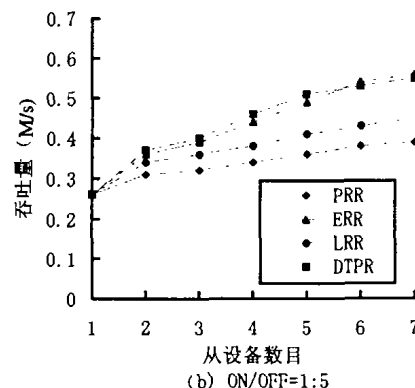
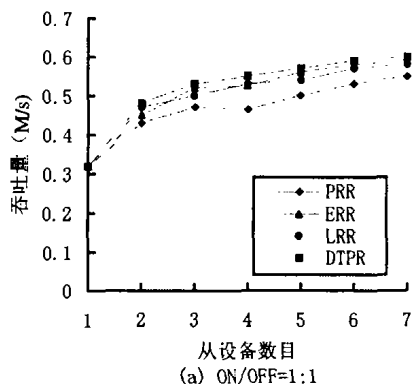


图3 不同ON/OFF比率下DTPR算法与PRR、ERR和LRR算法系统吞吐量的比较

(下转第88页)

- 9 Tian F, DeWitt D, Chen J, Zhang C. The Design and Performance Evaluation of Alternative XML Storage Strategies: [Technical Report]. CS Dept., University of Wisconsin, 2000
- 10 Schoning H. Tamino—A DBMS designed for XML. In: Proc. ICDE Conf., 2001
- 11 Jagadish H V, et al. Timber: A native XML database: [Technical report]. University of Michigan, April 2002
- 12 <http://exist.sourceforge.net/>
- 13 McHugh J, et al. Lore: A Database Management System for Semistructured Data. SIGMOD Record, 1997, 26(3)
- 14 Salminen A, Tompa F W. System Desiderata for XML Databases. In: Proc. of the 27th VLDB Conf. Roma, Italy, 2001
- 15 Abiteboul S. Querying semi-structured data. In: Proc. of the Int. Conf. on Database Theory, Greece, 1997
- 16 Calvanese D, De Giacomo G, Lenzerini M. Representing and Reasoning on XML Documents: A Description Logic Approach. J. Logic and Computation, 1999, 9(3)
- 17 Nestorov S, Abiteboul S, Motwani R. Extracting schema from semistructured data. In: Proc. of ACM SIGMOD Conf. on Management of Data. Seattle, WA, 1998
- 18 Papakonstantinou Y, Garcia-Molina H, Widom J. Object exchange across heterogeneous information sources. In: Proc. of the Eleventh Intl. Conf. on Data Engineering, Taipei, Taiwan, March 1995
- 19 Fernandez M, et al. An algebra for XML query. In Foundations of Software Technology and Theoretical Computer Science, (FSTTCS), Delhi, India, 2000
- 20 Murata M. DTD Transformation by Patterns and Contextual Conditions. In: Proc. SGML/XML'97 Conf. 1997
- 21 Buneman P, et al. Beyond XML query languages. In: Proc. of the Query Language Workshop (QL'98), 1998
- 22 Wallace M, Ranciman C. Haskell and XML: Generic Combinators or Type-Based Translation. In: Proc. of the 4th ACM SIGPLAN Conf. on Functional Programming (ICFP. 99)
- 23 David Maier Department of Computer Science and Engineering Oregon Graduate Institute. Database Desiderata for an XML Query Language. <http://www.w3.org/TandS/QL/QL98/pp/maier.html> May 2003
- 24 Suciu D. On Database Theory and XML. SIGMOD Record, 2001, 30(3)
- 25 <http://ozone-db.org/ozone-main.html>

(上接第49页)

在第一个实验里,比较这些算法的时延特性,仿真蓝牙网络中,主设备与每个从设备建立一个 ON:OFF 的 TCP 连接,每个设备都有业务源发生器,它们产生信息文件长度为10个 TCP 数据包, TCP 数据包的大小为512字节。图3给出了在不同 ON/OFF 的比率的情况下,各种轮询算法建立 TCP 连接的系统吞吐量,从图上可以看出,ERR 和 DTPR 两种算法比其它算法给出了很高的吞吐量,在 ON/OFF 低比率情况(1:1),DTPR 算法比 ERR 算法更好,而使用 ERR 算法在从设备之间引起了不公平性。PRR 算法表现出最低的性能。

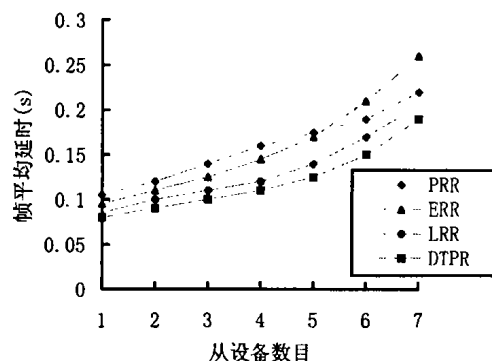


图4 在 ON/OFF 比率为1:1情况下,DTPR 算法与 PRR,ERR,LRR 算法帧平均时延的比较

图4给出了端到端的传输时延,结果表明,DTPR 算法比其它算法有更低的时延。并对本文提出的算法在公平性方面也做了进一步的证实,从表2可知,所有建立连接上的数据流平分带宽,也就是说,DTPR 算法对所有的数据流是公平的。

表2 双令牌随机轮询算法系统的吞吐量(Mb/s)

从设备 吞吐量 连接数	1	2	3	4	5	6	总吞 吐量
1	0.60						0.60
2	0.30	0.31					0.61
3	0.20	0.19	0.21				0.60
4	0.16	0.14	0.15	0.16			0.61
5	0.12	0.14	0.12	0.13	0.11		0.62
6	0.11	0.10	0.09	0.10	0.12	0.11	0.63

结束语 本文针对蓝牙网络中调度算法存在的问题,提出了一种改进算法——双令牌随机轮询算法,此算法解决了蓝牙网络中从设备之间的不公平性,避免了空轮询造成时隙的浪费,增加了对业务量多的从设备的调度。对现已提出的实际轮询算法与本文提出的算法进行了仿真,结果表明,该算法提高了系统的性能,可支持更高性能蓝牙产品的开发。

参考文献

- 1 Haartsen J. The Bluetooth radio system [J]. IEEE Personal Communications, 2000, 7(1): 28~36
- 2 Johansson N, Korner U, Johansson P. Performance evaluation of scheduling algorithms for Bluetooth [A]. In: Proc. of BC'99 IFIP TC 6 Fifth Intl. Conf. on Broadband Communications [C], Hong Kong: Kluwer Academic Press, 1999. 139~150
- 3 Zillbess S, Stahl W, Matheus K, Haartsen J. Radio network performance of Bluetooth [A]. IEEE ICC 2000 [C], New Orleans: IEEE Computer Society Press, 2000. 1563~1567
- 4 Garg S, Kalia M, Shorey R. MAC scheduling policies and SAR policies for Bluetooth: a master driven TDD pico-cellular wireless system [A]. MoMuc 99[C], San Diego: Academic Press, 384~386
- 5 Kalia M, Bansal D, Shorey R. Data scheduling and SAR for Bluetooth MAC [A]. IEEE VTC 2000 [C], Tokyo: Princeton University Press, 716~720
- 6 Capone A, Kapoor R, Gerla M. Efficient Polling Schemes for Bluetooth Picocells [A]. IEEE ICC 01 [C], MIT Press, 2001. 1990~1994
- 7 Liron Har-Shai, Ronen Kofman, Gil Zussman, Adrian Segall. Inter-piconet scheduling in Bluetooth scatternet [A]. In: Proc. of the OPNETWORK 2002 Conference [C], Washington: ICC Press, 2002. 1~11
- 8 Lee Y Z, Kapoor R, Gerla M. An Efficient and Fair Polling Scheme for Bluetooth. MILCOM 2002, Berlin: Springer, 2002. 1~7
- 9 The Network Simulator ns-2 [EB/OL]. <http://www.isi.edu/nsnam/ns/>. 2002-12-25
- 10 BlueHoc, IBM Bluetooth Simulator [EB/OL]. <http://oss.software.ibm.com/developerworks/opensource/bluehoc/>. 2001-02-25