

# 蓝牙自组个人网监听模式下分散网调度算法\*

林 鸿<sup>1+</sup>, 丁文芳<sup>2</sup>, 高 强<sup>1</sup>

<sup>1</sup>(中国科学院 软件研究所,北京 100080)

<sup>2</sup>(北京科技大学 计算机科学与技术系,北京 100083)

## A Scatternet Scheduling Algorithm in Sniff Mode for Bluetooth Ad Hoc Personal Area Networks

LIN Hong<sup>1+</sup>, DING Wen-Fang<sup>2</sup>, GAO Qiang<sup>1</sup>

<sup>1</sup>(Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

<sup>2</sup>(Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing 100083, China)

+ Corresponding author: Fax: 86-10-62963059, E-mail: honglin@263.net

<http://www.iscas.ac.cn>

Received 2002-05-31; Accepted 2002-08-22

**Lin H, Ding WF, Gao Q. A scatternet scheduling algorithm in sniff mode for Bluetooth Ad hoc personal area networks. *Journal of Software*, 2003,14(8):1495~1502.**

<http://www.jos.org.cn/1000-9825/14/1495.htm>

**Abstract:** Bluetooth ad hoc personal area network is one independent wireless network composed of Bluetooth enabled mobile devices, which can freely communicate and collaborate with each other in some area without extra infrastructure. Some nodes in Bluetooth ad hoc personal area networks participate in multiple piconets in time division duplex and forward network traffic. It needs one inter-piconet or scatternet scheduling algorithm since at a time the node presents in only one of the piconets it participates. Sniff mode provides a flexible method to support inter-piconet scheduling scheme without protocol modification. A self-adaptive scatternet scheduling algorithm is presented based on sniff mode to efficiently deal with the challenging issue in this paper. According to the supervised dynamic traffic of the physical links, the algorithm adjusts in real time the allocation ratio of the node's time slots to its connected links. With the comparison and analysis of the results of the simulation in various flow models, it validates that the algorithm can enhance the throughput of the system, lower the delay of the system and then improve the performance of the network.

**Key words:** Bluetooth; Ad hoc network; scatternet; scheduling

**摘 要:** 蓝牙自组个人区域网络是由具有蓝牙功能的设备组成一个不需额外网络设备,在某个区域内一起自主协调工作,相互间自由通信的独立无线移动网络.网络中的某些节点采用时分复用方式参与多个微微网中并转发网络数据.由于节点在某时刻只能加入其中一个微微网中,这就需要微微网间或者分散网调度算法.监听模式为分散网调度算法提供了一个比较灵活的方式并且无须更改协议.提出了一种基于监听模式的自适应分散

\* Supported by the National Grand Natural Science Foundation of China under Grant No.69833030 (国家自然科学基金重点基金)

第一作者简介: 林鸿(1973—),男,四川自贡人,博士生,主要研究领域为无线通信协议和应用,无线路由协议.

网调度算法,有效解决了这个具有挑战性的课题.算法监控各个链路上的流量,并获得流量变化数据,根据这些数据实时改变节点在各个链路上的监听时隙分配比例.通过对算法仿真结果的对比和分析,证明算法能够提高吞吐量,减少传输延迟,改善网络性能.

关键词: 蓝牙;自组网络;分散网;调度

中图法分类号: TP393 文献标识码: A

蓝牙技术<sup>[1-3]</sup>是一种短距离无线数据与语音通信的开放性全球规范.蓝牙具有成本低、功耗低、尺寸小、占用免费频段、通信距离短等特点.蓝牙自组个人区域网络<sup>[4,5]</sup>是由具有蓝牙功能的设备组成一个不需额外网络设备,在某个区域内一起自主协调工作,相互间自由通信的独立无线移动网络.这种网络可以由多个共用相同信道的蓝牙节点形成的微微网,也可以是由多个微微网在相互重叠的区域形成的分散网.在分散网中,有些节点是多个微微网的成员,是不同微微网中的从节点或者主节点,这些节点称为 PMP(participant in multiple piconets)节点.它们采用时分复用方式分别参与到每个所连接的微微网中,数据才能在微微网间传递.这意味着 PMP 节点在某时刻只能参与到其中连接的一个微微网中,需要一种微微网间调度 IPS(inter-piconet scheduling)算法<sup>[6-10]</sup>,又称为分散网调度算法.

蓝牙协议中的监听模式为微微网间调度算法提供了一个比较灵活的方式并且无须更改协议.从节点在监听模式时不用监听每个数据传输时隙,因此主节点只能在特定的时隙里传输数据.如图 1 所示,从节点每间隔  $T_{sniff}$  处于监听状态,这时主从节点间可以传输数据,无论从节点收到属于自己的数据包与否,监听状态都维持  $T_{sniffattemp}$  个时隙.如果在监听状态中收到传给自己的数据包时,从节点还要继续在下面的  $T_{snifftimeout}$  个时隙和剩余的  $T_{sniffattemp}$  个时隙中较大的时隙里处于监听状态,其他时间从节点可以处于休眠状态,这时设备处于低能耗模式,设备间不能传输数据,但从节点可以参与到其他的微微网中. $T_{sniff}$ ,  $T_{sniffattemp}$  和  $T_{snifftimeout}$  的值可以根据需要灵活调整, $T_{snifftimeout}$  一般设为 0,  $T_{sniff}$  为监听周期,  $T_{sniffattemp}$  为监听时隙.

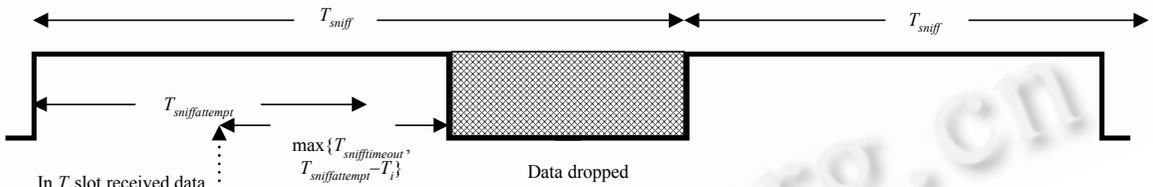


Fig.1 Relationship between sniff period and sleep period in sniff mode

图1 监听模式中监听周期和休眠周期的关系

分散网可以组成各种不同的拓扑结构,但典型的有两种:一种是参与到各微微网的 PMP 节点只作从节点,另一种是在一个微微网中作主节点,其他微微网中作从节点.我们以这两种结构讨论节点处于监听模式时的参数取值情况,这里不讨论两个 PMP 节点直接连接的情况.如图 2(a)所示,PMP 从节点  $S_B$  在多个微微网中只作为从节点,这称为单从结构 ST-SO.图 2(b)是 ST-SO 拓扑结构的一个典型应用场景.如图 3(a)所示,PMP 主节点是一个微微网的主节点,在其他微微网中为从节点,这称为主从结构 ST-MS.图 3(b)是 ST-MS 拓扑结构的一个典型应用场景.

Baatz 等人<sup>[6]</sup>的算法没有在更复杂拓扑结构中考察算法性能,不能根据流量变化改变分配点数的密度.Johansson 等人<sup>[7]</sup>对一组基于集中点 IPS 算法做了总结.Racz 等人<sup>[8]</sup>提出了一种集中点窗口的自适应流量动态变化调度算法,但缺少不同链路上集中点协调机制.Zhang 等人<sup>[9]</sup>将微微网内和网间调度结合起来,但切换表变化和流量传递而产生的额外负荷使系统负载加重,可靠性降低.Johansson 等人<sup>[10]</sup>提供一种基于跳模式轻负载调度框架,但是需要运行具体调度算法才能提高系统性能.

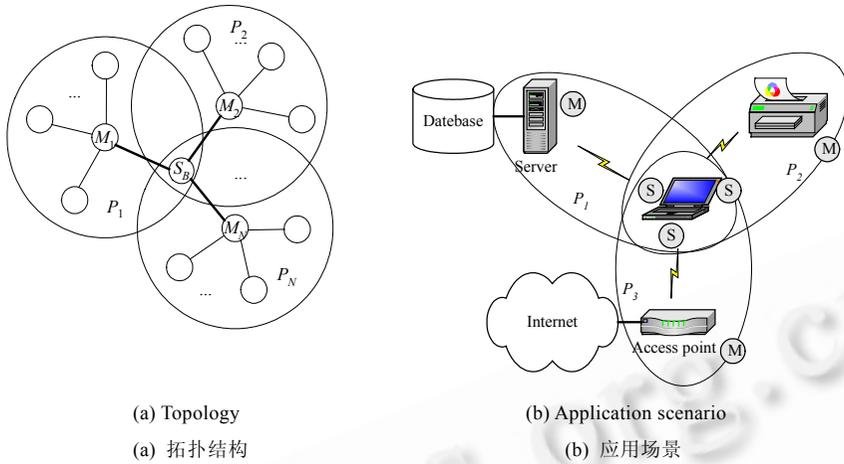


Fig.2 Topology and scenario of ST-SO

图2 ST-SO 拓扑结构和应用场景

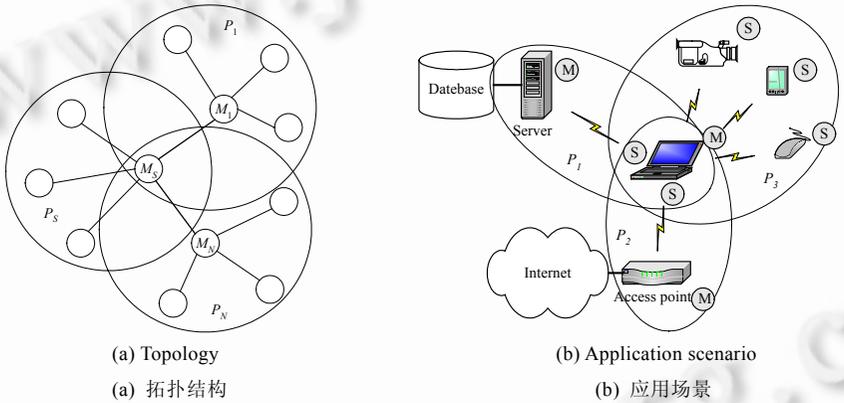


Fig.3 Topology and scenario of ST-MS

图3 ST-MS 拓扑结构和应用场景

基于集中点的IPS算法虽然能够更好地改善系统性能,但是它需要对蓝牙协议作较大的改进和扩充后才能实现.在不更改现有蓝牙协议的基础上,本文提出了一种基于监听模式的自适应分散网调度算法 SSSM-SA.算法监控各个链路上的流量,并获得流量变化数据,根据这些数据实时改变节点在各个链路上的监听时隙分配比例,从而达到提高吞吐量、减少时间延迟、改善网络性能的目的.为了衡量算法性能,本文定义了参照算法,对算法和参照算法在吞吐量、时延和能耗等性能指标上进行对比.

### 1 基于监听模式的分散网调度算法 SSSM-SA

自适应分散网调度算法是监控各个链路上的流量并获得流量变化数据.根据这些数据调整监听模式的关键参数,实时改变节点在各个链路上的监听时隙分配比例,反映链路上的流量变化,从而有效提高吞吐量,降低时间延迟,改善网络性能.目前,蓝牙协议中监听模式对微微网间调度的支持有些不方便和不明确的地方.处于监听模式后不能改变监听模式的参数.只有在退出监听模式后再进入监听模式才可能使新的参数生效.监听初始时隙由主节点决定,这使连接多个微微网的从节点上的多个链路的监听时隙可能严重相互重叠,使从节点不能很好地参与到其他的微微网中.因此对监听模式的操作方式需要作一些改进:进入监听模式以后,主节点或者从节点可以改变监听参数,不需要退出监听模式.主、从节点之间协商监听参数并在下一个时隙后开始使用新

的监听参数.监听初始时隙由从节点决定,从节点根据所连接的微微网个数和监听参数使分配在各个链路上的监听时间最大限度地不发生重叠.

### 1.1 数据流量变化的监控

PMP 节点实时监控与其他节点建立的物理链路间的数据传输,获得各个物理链路上的数据流量.PMP 节点通过计算与某节点之间在某段时间通信的数据量,可以获得某时间段的数据流量.为了更精确地计算出物理链路的流量,减少因突发数据造成数据流量的突然变化,需要对当前所获得的数据流量与历史的数据流量作加权平均.根据物理链路的数据流量,定义物理链路的繁忙程度  $B$ ,简称为忙因子,取值范围从 0.0~1.0,值越大越繁忙.当 PMP 节点处于监听模式时,监听时隙和监听周期的比值应该接近于  $B$  值,因为能传输数据的监听时隙所占的比例应该与  $B$  成正比, $B$  值越大,数据流量就越大,所需的监听时隙就越长,反之亦然.假定监听模式的参数  $T_{snifftimeout}$  为 0,则  $B$  与  $T_{sniff}$  和  $T_{sniffatempt}$  应该有关系: $B \propto T_{sniffatempt}/T_{sniff}$ .已知  $B$  和  $T_{sniff}$  值,就可以计算出  $T_{sniffatempt}$  值.为了避免因为  $B$  值的变化而产生相应监听模式参数的不断变化,只有当  $B$  值发生较大变化时,监听时隙和监听周期的值才作适当的放大或者缩小.

### 1.2 最佳监听时隙分配计算

为实现分配在各微微网上的监听时间最大限度地不重叠,需要进行一些额外的简单计算.假设某个节点连接  $N$  个微微网,它在每个微微网中都处于监听模式,监听模式的参数分别是  $T_{sniffatempt}^i$  和  $T_{sniff}^i$ ,其中  $1 \leq i \leq N$ .  $T_{sniffatempt}^i/T_{sniff}^i$  为节点在第  $i$  个微微网中的监听时间比例,因此  $T_{sniffatempt}^i$  和  $T_{sniff}^i$  参数必须满足  $\sum T_{sniffatempt}^i/T_{sniff}^i \leq 1 (1 \leq i \leq N)$ .这时,如果节点取一个统一的监听间隔  $T_{sniff}$ ,再将各个微微网的监听时隙分配到这个监听间隔中,使它们不重叠,实现监听时隙最大限度地不重叠. $T_{sniff}$  为各个  $T_{sniff}^i$  的最小公倍数,即

$$T_{sniff} = LCM(T_{sniff}^1, T_{sniff}^2, \dots, T_{sniff}^N).$$

而各个  $T_{sniffatempt}^i$  更改为  $T_{sniffatempt}^i = T_{sniffatempt}^i \cdot T_{sniff} / T_{sniff}^i$ .但是,当某个微微网中监听模式参数发生变化时,节点要重新计算  $T_{sniff}$ ,这增大了节点计算量.因此在实际应用中,假定  $T_{sniff}^i$  不是连续取值,而是 2 的指数关系,即  $T_{sniff}^i = 2^j (j > 1)$ .这时  $T_{sniff}$  的取值为  $\{T_{sniff}^i\}$  中的最大值.而  $T_{sniffatempt}^i$  只需乘以某个 2 的倍数即可,使计算量大大降低了.

### 1.3 监听参数计算

图 2 中 PMP 从节点连接了  $N$  个微微网,并是微微网  $P_1$  到  $P_N$  的从节点.作为连接这些微微网的网桥,这时应在  $S_B$  连接的微微网中都置于监听模式状态.如果  $S_B$  在微微网  $P_1$  中,当主节点  $M_1$  和  $S_B$  之间的数据流量较小时, $S_B$  会被置于监听时隙较短的监听模式,若按照微微网中的主节点决定监听模式的参数, $S_B$  有充足的时间参与到其他的微微网中.但是当主节点与从节点  $S_B$  之间的数据流量都较大时, $S_B$  会被主节点都置于某种较忙的模式,这时监听模式参数之间就发生了冲突.因此需要  $S_B$  决定和协调各个监听模式的参数.

假设  $S_B$  计算出各个链路的忙因子  $B^i, 1 \leq i \leq N$ .  $B^i$  是各个链路的绝对值,若  $\sum B^i > 1$ ,则首先需要对  $B^i$  作修整,得到相对值  $B_{new}^i, B_{new}^i = B^i / \sum B^j$ .若  $\sum B^i \leq 1$ ,则  $B_{new}^i = B^i$ .根据  $B_{new}^i$  计算从节点监听模式参数  $T_{sniffatempt}^i$  和  $T_{sniff}^i$ ,使它在不同微微网中的参与度与其忙因子成正比,并且参数满足  $\sum T_{sniffatempt}^i/T_{sniff}^i \leq 1$ .若  $T_{sniff}^i$  确定并且等于  $T_{sniff}$ ,则  $T_{sniffatempt}^i$  为  $T_{sniffatempt}^i = B_{new}^i T_{sniff}$ .

图 3(a)中 PMP 主节点  $M_S$  是微微网  $P_S$  的主节点和微微网  $P_1$  到  $P_N$  的从节点.这时  $M_S$  就不能将所有时间花在微微网  $P_S$  中,它必须在某些时间切换到其他的微微网中.但是, $M_S$  是  $P_S$  的主节点,它需要更多的时间处理  $P_S$  中的事务,这将减少  $M_S$  参与到  $P_1$  到  $P_N$  中的时间.因此必须根据微微网内和连接其他微微网的各个链路上的忙因子统一协调监听时间.假设  $M_S$  连接其他  $N$  个微微网,它们的忙因子是  $B^1$  到  $B^N$ .微微网  $P_S$  内有  $N_S$  个从节点,

它们的忙因子是  $B^{N+1}$  到  $B^{N+N_S}$ .  $B^i$  是各个链路上的绝对值,其中  $1 \leq i \leq N + N_S$ . 若  $\sum_{i=1}^{N+N_S} B^i > 1$ , 需对  $B^i$  作修整,得到相对值  $B_{new}^i$ , 其中  $1 \leq i \leq N + 1$ .  $B_{new}^1$  到  $B_{new}^N$  是  $M_S$  连接其他微微网忙因子的修正值,而  $B_{new}^{N+1}$  是微微网  $P_S$  内所有从节点忙因子的总和,它决定了  $M_S$  花在微微网  $P_S$  内的时间比例.  $B_{new}^i$  取值为

$$B_{new}^i = B^i / \sum_{j=1}^{N+N_S} B^j (1 \leq i \leq N), B_{new}^{N+1} = \sum_{j=N+1}^{N+N_S} B^j / \sum_{j=1}^{N+N_S} B^j.$$

若  $\sum_{i=1}^{N+N_S} B^i \leq 1$ , 则  $B_{new}^i = B^i$ ,  $B_{new}^{N+1} = \sum_{j=N+1}^{N+N_S} B^j$ . 根据  $B_{new}^i$  再计算从节点监听模式参数  $T_{sniffatempt}^i$  和  $T_{sniff}^i$ . 若  $T_{sniff}^i$  确定并且都等于  $T_{sniff}$ , 则  $T_{sniffatempt}^i$  有相同的计算方法为  $T_{sniffatempt}^i = B_{new}^i T_{sniff} \cdot T_{sniffatempt}^{N+1}$  为  $M_S$  留在微微网  $P_S$  服务网内从节点的时间比例.

#### 1.4 监听参数计算

只有当  $B^i$  发生较大变化时,对应的监听参数才会改变,以反映目前链路上的流量变化.当某个链路上有突发数据或者连续数据时,节点能有更多的时间比例参与到这个链路上.或者当某个链路开始处于比较空闲的时候,节点能从这个链路上释放出来参与到其他的链路上,使系统整体吞吐量较大,平均延迟较小.

假设  $\delta$  为监听模式参数改变所需的阈值,  $B_{history}^i$  为  $B^i$  的历史平均值.如果  $|(B_{history}^i - B^i) / \max(B^i, B_{history}^i)| > \delta$ , 说明  $B^i$  开始有较大的变化,已经达到了参数可改变的界限.这时,如果  $B^i > B_{history}^i$ , 说明链路上的流量有较大增长,这时  $T_{sniffatempt}^i$  应增大,节点需要更多的监听时隙,允许主、从节点之间传送更多的数据.相反,如果  $B^i < B_{history}^i$ , 说明链路上的流量有较大减少,这时  $T_{sniffatempt}^i$  应减小,节点释放出监听时隙,允许其他链路上有更多数据传送.假设  $A_{sniff}^i = |(B_{history}^i - B^i) / \max(B^i, B_{history}^i)|$ ,  $T_{sniffatempt}^i$  增大和减少的幅度与  $A_{sniff}^i$  有关,  $A_{sniff}^i$  越大,  $T_{sniffatempt}^i$  的增减幅度就越大.因此当  $B^i > B_{history}^i$  时,将  $T_{sniffatempt}^i$  更改为  $T_{sniffatempt}^i = (1 + A_{sniff}^i / 2) T_{sniffatempt}^i$ . 当  $B^i < B_{history}^i$  时,将  $T_{sniffatempt}^i$  更改为  $T_{sniffatempt}^i = (1 - A_{sniff}^i / 2) T_{sniffatempt}^i$ .

假设  $T_{sniffatempt}^i$  和  $T_{sniff}^i$  都是有界定值的,即  $T_{sniffatempt}^i \in [T_{sniffatempt}^{i, \min}, T_{sniffatempt}^{i, \max}]$ ,  $T_{sniff}^i \in [T_{sniff}^{i, \min}, T_{sniff}^{i, \max}]$ . 因此,如果更改后的  $T_{sniffatempt}^i < T_{sniffatempt}^{i, \min}$  或  $T_{sniff}^i < T_{sniff}^{i, \min}$ , 则  $T_{sniffatempt}^i$  等于  $T_{sniffatempt}^{i, \min}$  或  $T_{sniff}^i$  等于  $T_{sniff}^{i, \min}$ , 但  $T_{sniff}^i$  应增大,即  $T_{sniff}^i = 2T_{sniff}^i$  ( $2T_{sniff}^i < T_{sniff}^{i, \max}$ ) 或者  $T_{sniff}^i = T_{sniff}^{i, \max}$  ( $2T_{sniff}^i \geq T_{sniff}^{i, \max}$ ). 如果更改后的  $T_{sniffatempt}^i > T_{sniffatempt}^{i, \max}$  或  $T_{sniff}^i > T_{sniff}^{i, \max}$ , 则  $T_{sniffatempt}^i$  应减小,即  $T_{sniffatempt}^i = T_{sniffatempt}^i / 2$  ( $T_{sniffatempt}^i / 2 > T_{sniffatempt}^{i, \min}$ ) 或者  $T_{sniffatempt}^i = T_{sniffatempt}^{i, \min}$  ( $T_{sniffatempt}^i / 2 \leq T_{sniffatempt}^{i, \min}$ ). 而  $T_{sniff}^i = (T_{sniff}^{i, \max} + T_{sniffatempt}^i) T_{sniff}^i / (2T_{sniff}^{i, \max})$ .  $T_{sniffatempt}^i$  和  $T_{sniff}^i$  与节点连接的微微网个数  $N$  有关.如果是图 3(a)中的情况还加上  $N_S$ .  $T_{sniffatempt}^{i, \min}$  和  $T_{sniffatempt}^{i, \max}$  的取值可以是  $T_{sniffatempt}^{i, \min} = T_{sniff}^{i, \max} / (4N)$ ,  $T_{sniffatempt}^{i, \max} = 3T_{sniff}^{i, \max} / 4$ .  $T_{sniffatempt}^{i, \min}$  和  $T_{sniffatempt}^{i, \max}$  也可以取其他值,它们可以控制监听时隙在监听周期里的取值范围.如果改变后  $T_{sniffatempt}^i / T_{sniff}^i$  增大且造成  $\sum_{i=1}^N T_{sniffatempt}^i / T_{sniff}^i > 1$ , 则  $T_{sniffatempt}^i$  和  $T_{sniff}^i$  恢复原来的值.

#### 1.5 算法描述

假设节点连接的微微网个数为  $N$ , 忙因子为  $B^i$ , 历史忙因子为  $B_{history}^i$ , 忙因子变化率为  $A_{sniff}^i$ , 忙因子历史值的权值为  $\alpha$ ,  $L_{rx}^i$  为在上个监听时隙里接受的数据个数,  $L_{tx}^i$  为在上个监听时隙里发送的数据个数, 流量为  $F^i$ , 监听间隔为  $T_{sniff}^i$ , 监听时隙为  $T_{sniffatempt}^i$ , 可取的监听周期的最大值为  $T_{sniff}^{i, \max}$ , 最小值为  $T_{sniff}^{i, \min}$ , 可取的监听时隙的最大值为  $T_{sniffatempt}^{i, \max}$ , 最小值为  $T_{sniffatempt}^{i, \min}$ , 判断改变监听参数的阈值为  $\delta$ .

综合上述讨论,下面给出如图 2(a)中的具体基于监听模式的分散网调度算法.

(1) 初始时:  $B^i = 0, B_{history}^i = 0, T_{sniff}^i = T_{sniff}^{max} / 2, T_{sniffatempt}^i = T_{sniff}^i / N$ .

(2) 计算  $B^i: F^i = (L_{rx}^i + L_{tx}^i) T_{sniff}^i / (T_{sniff}^{max} T_{sniffatempt}^i)$ , 则  $B^i = F^i / F^{max}$ .

(3) 修正  $B^i$ : 如果  $\sum B^i > 1, B^{sum} = \sum_{i=1}^N B^i, B^i = B^i / B^{sum}$ .

(4) 计算  $T_{sniffatempt}^i$ :

$$A_{sniff}^i = |(B_{history}^i - B^i) / \max(B^i, B_{history}^i)|,$$

如果  $A_{sniff}^i > \delta$ , 则当  $B^i > B_{history}^i, T_{sniffatempt}^i = (1 + A_{sniff}^i / 2) T_{sniffatempt}^i$ ; 则当  $B^i < B_{history}^i, T_{sniffatempt}^i = (1 - A_{sniff}^i / 2) T_{sniffatempt}^i$ .

(5) 修正更改过的  $T_{sniffatempt}^i$ :

如果  $T_{sniffatempt}^i < T_{sniffatempt}^{min} T_{sniff}^i / T_{sniff}^{max}$ , 则  $T_{sniffatempt}^i = T_{sniffatempt}^{min} T_{sniff}^i / T_{sniff}^{max}$ ,  $T_{sniff}^i = 2T_{sniff}^i$ , 如果  $T_{sniff}^i > T_{sniff}^{max}$ , 则  $T_{sniff}^i = T_{sniff}^{max}$ ; 如果  $T_{sniffatempt}^i > T_{sniffatempt}^{max} T_{sniff}^i / T_{sniff}^{max}$ , 则  $T_{sniff}^i = T_{sniff}^i / 2, T_{sniffatempt}^i = (T_{sniff}^{max} + T_{sniffatempt}^i) T_{sniff}^i / (2T_{sniff}^{max})$ , 如果  $T_{sniff}^i < T_{sniff}^{min}$ , 则  $T_{sniff}^i = T_{sniff}^{min}$ .

(6) 检查监听比例: 如果  $T_{sniffatempt}^i / T_{sniff}^i$  增大且  $\sum T_{sniffatempt}^i / T_{sniff}^i > 1, T_{sniffatempt}^i$  和  $T_{sniff}^i$  不变.

(7) 调整  $B_{history}^i: B_{history}^i = \alpha B_{history}^i + (1 - \alpha) B^i$ .

## 2 仿真结果和分析

### 2.1 仿真环境

在 ns-2 网络仿真器<sup>[11]</sup>和 IBM 为 ns 开发的 BlueHoc<sup>[12]</sup>蓝牙扩展模块基础上实现本文提出的算法,并参照算法进行仿真实验,分别在变化位速率 VBR 和恒定比特率 CBR 流量模型下对算法作性能对比.其中 CBR 的总速度是 420kb/s,再平均分配到各个链路上.VBR 速度在 420kb/s 和 42kb/s 之间作随机变化.SSSM-SA 和参照算法在性能上进行比较.这种参照算法不管具体流量如何变化,在每个链路上的监听时间是平均分配的,这种调度算法称为均匀静态调度算法 SSSM-ES.SSSM-SA 和 SSM-ES 在吞吐量、延迟和能耗效率等性能指标上进行对比.吞吐量是 PMP 节点的链路上获得的平均吞吐量,单位是 kb/s.延迟是 PMP 因调度而引起的链路数据传送的平均传输延迟,单位是 ms.能耗效率是 PMP 节点与主节点通信过程中成功接受和发送的数据量,它可以表明节点在某一个链路上通信 1 秒所传送的数据字节数.单位为 kb/eu(单位能量).

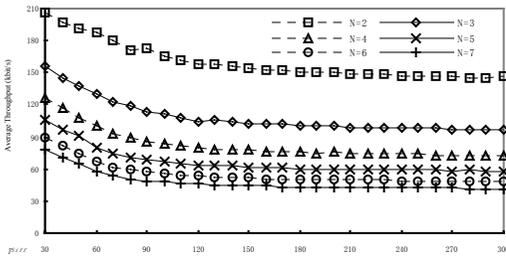
### 2.2 结果分析

图 4 是  $T_{sniff}$  不同取值时 SSSM-ES 算法对吞吐量和延迟性能的影响. $N$  从 2 到 7,  $T_{sniff}$  从 30 到 300.如图可以看出,当  $N$  一定时,随着  $T_{sniff}$  增加,节点上的平均吞吐量减小,平均延迟增大.这是因为随着  $T_{sniff}$  增加,节点需要更长的时间才能参与到下一个微微网中,这样平均延迟也就随之增大.但每次传送的数据量也增大,因此平均吞吐量虽然在较小的  $T_{sniff}$  时会减小,但是较大的  $T_{sniff}$  取值使传送数据量和延迟都增大,因此平均吞吐量随  $T_{sniff}$  增大而呈较小幅度的减小.当  $T_{sniff}$  一定时,随着  $N$  的增大,节点上的平均吞吐量也减小,平均延迟也增大.这是因为用于传送数据的时隙被分割成更小的时隙,节点需要更长的时间才能参与到下一个微微网中,这样平均延迟也就随之增大,同时,在这个更小的时隙能传送的数据更少,于是平均吞吐量也就减小了.

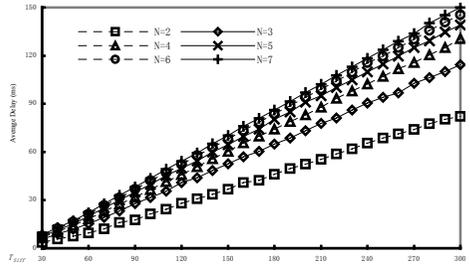
图 5 是算法 SSSM-ES 和 SSSM-SA 在 VBR 流量模型下的性能比较,图 6 是在 CBR 下的比较.参数分别是  $N \in [2, 7], \alpha = 0.5, \delta = 0.25, T_{sniff}^{max} = 60, T_{sniff}^{min} = 30, T_{sniffatempt}^{max} = 45, T_{sniffatempt}^{min} = \lceil 15 / N \rceil$ .

与前面讨论的一样,SSSM-ES 和 SSSM-SA 算法也都随  $N$  的增大,吞吐量在减小,延迟在增大.在 SSSM-SA 算法中,节点能随链路上流量的动态变化而实时改变节点在链路上的监听时隙长度,使链路上的数据能在更短的时间内更多地传送,从而最大限度地增加吞吐率,减少时间延迟,提高网络性能.因此,无论在 VBR 或者 CBR 流量模型下,SSSM-SA 算法在吞吐率和延迟上都比 SSSM-ES 算法更优.同时,SSSM-SA 算法也能提高能耗效率,这是因为算法在消耗相同的能量情况下能够尽可能地传送更多的数据,提高了链路的利用率,减少了链路的空

闲时间.



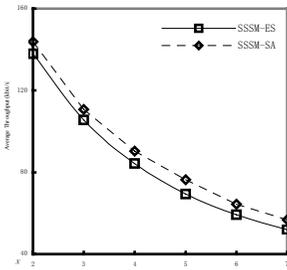
(a) Average throughput  
(a) 平均吞吐量



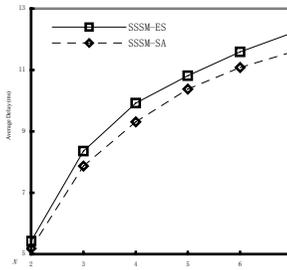
(b) Average delay  
(b) 平均时延

Fig.4 Effect of variable  $T_{sniff}$  on Performance

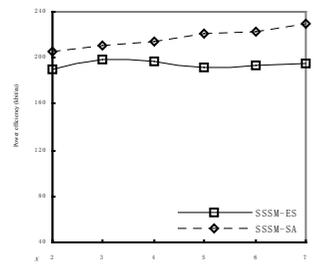
图4  $T_{sniff}$ 取值对性能的影响



(a) Average throughput  
(a) 平均吞吐量



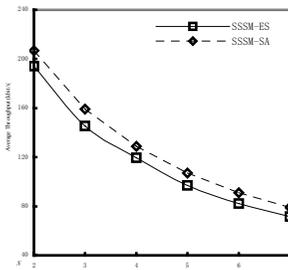
(b) Average delay  
(b) 平均时延



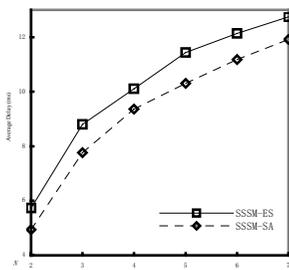
(c) Power efficiency  
(c) 能耗效率

Fig.5 Comparison between algorithm SSSM-ES and SSSM-SA in VBR flow model

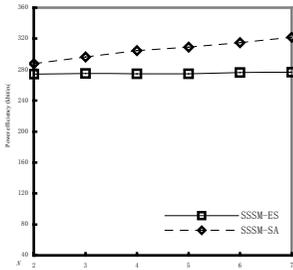
图5 VBR 流量模型下 SSSM-ES 和 SSSM-SA 算法性能比较



(a) Average throughput  
(a) 平均吞吐量



(b) Average delay  
(b) 平均时延



(c) Power efficiency  
(c) 能耗效率

Fig.6 Comparison between algorithm SSSM-ES and SSSM-SA in CBR flow model

图6 CBR 流量模型下 SSSM-ES 和 SSSM-SA 算法性能比较

### 3 结论

本文提出了基于监听模式的自适应分散网调度算法 SSSM-SA.算法监控各个链路上的流量,并获得流量变化数据,根据这些数据调整监听模式的关键参数,实时改变节点在各个链路上的监听时隙分配比例.给出了在监听模式中节点实现最佳监听时隙分配的方案,以及在不同分散网络拓扑结构中根据流量变化改变监听参数的计算方法和监听时隙分配比例控制机制.仿真实验结果证明,算法能有效地提高吞吐量,减少时间延迟,改善网络性能.自适应算法需要考虑更复杂情况下的调度机制,特别是当两个 PMP 节点直接连接时的情况,这时需要

两个节点都有较好的协调机制,以使两个节点在某段时间内不在相同微微网内的时间最少.

#### References:

- [1] The Bluetooth Special Interest Group (SIG). 2001. <http://www.bluetooth.org/>.
- [2] Haartsen J, Haghshineh M, Inouye J, Joeressen OJ, Allen W. Bluetooth: Visions, goals, and architecture. *ACM Mobile Computing and Communications Review*, 1998,2(4):38~45.
- [3] Haartsen JC. The Bluetooth radio system. *IEEE Personal Communications*, 2000. 28~36.
- [4] Siep TM, Gifford IC, Braley RC, Heile RF. Paving the way for personal area network standards: An overview of the IEEE P802.15 working group for wireless personal area networks. In: *IEEE Personal Communications*, 2000,7:37~43.
- [5] IETF MANET Working Group.2001. <http://www.ietf.org/html.charters/manet-charter.html>.
- [6] Baatz S, Frank M, Kehl C, Martini P, Scholz C. Adaptive scatternet support for Bluetooth using sniff mode. In: *Proceedings of the 26th Annual Conference on Local Computer Networks*. 2001.
- [7] Johansson P, Kazantzidis M, Kapoor R, Gerla M. Bluetooth: An enabler for personal area networking. *IEEE Network*, 2001,15(5):28~37.
- [8] Racz A, Miklos G, Kubinszky F, Valko A. A pseudo random coordinated scheduling algorithm for Bluetooth scatternets. In: *Proceedings of the 2001 ACM International Symposium on Mobile Ad hoc networking and computing*. 2001. 193~203.
- [9] Zhang W, Cao G. A flexible scatternet-wide scheduling algorithm for Bluetooth networks. In: *Proceedings of the IEEE International Performance, Computing, and Communications Conference (IPCCC)*. 2002.
- [10] Johansson N, Alriksson F, Jonsson U. JUMP mode—a dynamic window-based scheduling framework for Bluetooth scatternets. In: *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. 2001. 204~211.
- [11] ns-2: network simulator. 2000. <http://www.isi.edu/vint/nsnam/>.
- [12] BlueHoc: Bluetooth performance evaluation tool. 2001. <http://oss.software.ibm.com/developerworks/opensource/bluehoc/>.