

按需轮循的蓝牙微微网调度算法与性能评估

杨帆, 王珂, 钱志鸿

(吉林大学通信工程学院, 吉林长春 130022)

摘要: 提出了一种低时延、高信道利用率与公平性好的按需轮循的蓝牙微微网调度算法. 首先通过速率估计过程计算出上行链路分组到达率的估计值, 然后在主调度过程中分别计算各链路对自上次轮循时刻到当前时刻的分组到达数量的估计值与等待时间, 根据提出的动态轮循选择规则选出优先度最高的链路对进行轮循, 再根据自适应轮循间隙规则计算出下次轮循应该进行的时间. 分别在混合、下行与上行三种传输模型的仿真中进行了与其他经典算法的性能比较. 结果表明: 算法在前两种传输模型中具有最佳的整体性能, 尤其是时延分别降低了 24.1% 与 41.4% 以上, 在第三种传输模型中也具有较佳的整体性能. 该算法效率高、实现复杂度低, 可应用于蓝牙与其他无线网络.

关键词: 蓝牙微微网; 调度算法; 按需轮循; 时延; 信道利用率; 公平

中图分类号: TN915 **文献标识码:** A **文章编号:** 0372-2112 (2007) 04-0647-06

Polling on Demand Scheduling Algorithm for Bluetooth Piconet and Performance Evaluation

YANG Fan, WANG Ke, QIAN Zhi hong

(College of Communication Engineering, Jilin University, Changchun, Jilin 130022, China)

Abstract: A polling on demand scheduling algorithm for bluetooth piconet is presented, which has good characters of low delay, high channel utilization and desired fairness. The algorithm computes out estimate of packet arrival rates for upstream links by the rate estimating process, and then it works out the estimate of packet arrival number and latency for every link pair from its last poll time through present moment. According to the dynamic polling selection regulation presented in the paper, the scheduler selects a link pair that has the highest priority to poll, and computes the next polling time according to the adaptive polling interval regulation. The three traffic modes, mixed, downstream and upstream are used to compare the performance between this algorithm and other classical algorithms by simulations. The results indicate that this algorithm has the best integral performance in the first two traffic modes, and particularly, delays decrease by at least 24.1% and 41.4% respectively. The algorithm also has relatively good behaviors in the third traffic mode. This algorithm is efficient and has low complexity, and it can be used for bluetooth and other wireless networks.

Key words: bluetooth piconet; scheduling algorithm; polling on demand; delay; channel utilization; fairness

1 引言

随着蓝牙 2.0 新标准^[1]的颁布, 更快的传输速率与更方便的组网模式使蓝牙技术在上世界上得到了更广泛的应用. 典型的蓝牙微微网由 1 个主节点与最多 7 个活动从节点构成. 主节点负责对整个微微网进行控制, 只有主从节点间可以直接交换数据. 主节点对选定的从节点进行轮循时, 发送数据分组或单时隙的 POLL 分组, 被轮循的从节点在随后的时隙中返回数据分组或单时隙的 NULL 分组. 主节点按照预定的调度算法来控制整个微微网的通信, 对每个从节点进行轮循. 轮循规则首

先要保证每个有数据需要发送的从节点都有机会得到轮循, 其次要保证一定的公平性. 当前蓝牙标准一直使用的 Round Robin^[2]调度算法存在很多不足, 对于蓝牙这样具有双向链路的无线网络效率低下. 近几年来, 大量的文献提出了蓝牙微微网的轮循调度算法^[2, 5~16]. 但目前并没有一种算法能够同时具有较低的时延、较高的信道利用率以及较佳的公平性与较低的算法复杂度, 也没有一种算法能够同时适应各种传输模式以及各种流量条件.

本文根据以往调度算法的不足, 提出了一种按需轮循的蓝牙微微网调度算法. 算法在大部分流量条件与

传输模型中同时兼有低时延、高信道利用率与高公平性的优秀性能,能够自适应地按照各链路的流量情况动态调整各链路对轮循的优先度以及链路对间的动态轮循间隙,即实现按需轮循的目的.算法实现简单,没有修改任何蓝牙现有协议规范.此外,本文提出了一组评估蓝牙微微网调度算法的性能指标,并通过全面的仿真验证了算法的有效性.

2 按需轮循的蓝牙微微网调度算法

假设微微网 P 由 1 个主节点 m 与 N ($N \leq 7$) 个活动从节点 s_i ($i \leq N$) 构成.主节点的调度器维护 $2N$ 个链路,其中主到从的 N 个下行链路为: $(Lm_1, Lm_2, \dots, Lm_N)$, Lm_i 表示其中某个下行链路;从到主的 N 个上行链路为: $(Ls_1, Ls_2, \dots, Ls_N)$, Ls_i 表示其中某个上行链路.令 L_i 为第 i 个链路对,即 $L_i = \{Lm_i, Ls_i\}$.设上行链路 Ls_i 中从节点 s_i 的平均分组到达率为 λ_{s_i} ,下行链路 Lm_i 中主节点 m 的平均分组到达率为 λ_{m_i} .则链路对 L_i 的平均分组到达率 $\lambda(L_i)$ 为 $\lambda_{m_i} + \lambda_{s_i}$.整个微微网的平均分组到达率为: $\lambda(P) = \sum_{i=1}^N [\lambda(m_i) + \lambda(s_i)]$. N 个链路对的分组到达率的协方差定义为^[16]:

$$COV = \frac{N \left[\sum_{i=1}^N \lambda^2(L_i) \right] - \frac{1}{N} \left[\sum_{i=1}^N \lambda(L_i) \right]^2}{(N-1)} \quad (1)$$

假设到达的基带数据分组类型仅为 DH1、DH3 与 DH5,分别占用 1、3 与 5 个时隙,其有效载荷分别为: 27、183 与 339 字节^[3,4].假设 DH1、DH3 与 DH5 分组产生的概率分别为 p_{DH1} 、 p_{DH3} 与 p_{DH5} ,则到达分组的平均字节长度与平均占用时隙数分别表示为式(2)与式(3):

$$\bar{L} = 27p_{DH1} + 183p_{DH3} + 339p_{DH5} \quad (2)$$

$$\bar{d} = p_{DH1} + 3p_{DH3} + 5p_{DH5} \quad (3)$$

整个蓝牙微微网的网络负载可定义为:

$$\rho = \frac{1}{1600} \sum_{i=1}^N \lambda(L_i) \bar{d} \quad (4)$$

根据蓝牙网络中分组到达的一般规律并简化分析,假设上下行链路的分组到达分别服从强度为 λ_{s_i} 与 λ_{m_i} 的 Poisson 过程.用 $M(t)$ ($t \geq 0$) 表示 $[0, t]$ 时段内分组到达的个数.由 Poisson 过程的定义可知在时间间隔 $[t, t + \Delta t]$ 内出现 k 个分组的概率分布为:

$$P(M(t, t + \Delta t) = k) = \frac{(\lambda \Delta t)^k}{k!} e^{-\lambda \Delta t}, k = 0, 1, 2, \dots \quad (5)$$

在时间间隔 $[t, t + \Delta t]$ 内到达分组的数学期望为:

$$Q = \sum_{k=0}^{\infty} k \cdot P(M(t, t + \Delta t) = k) = \lambda \Delta t \quad (6)$$

其中 λ 为分组平均到达率.

2.1 分组到达率估计过程

本过程通过类似 Round Robin 的方法在短时间内估计各上行链路队列中的分组到达率,并将继续在随后主调度过程中的每次轮循时继续更新.假设 $t = 0$ 时整个调度过程开始,主节点 m 按照固定的次序 ($s_1, s_2, \dots, s_N$) 周期性地轮循各从节点.当某从节点 s_i 得到轮循机会时,主节点 m 向 s_i 发送一个数据分组或 POLL 分组, s_i 则返回数据分组或 NULL 分组,然后主节点 m 继续轮循下一个从节点 s_{i+1} ,如此反复.令 $UL_{s_i}(T)$ 为在 $[0, T]$ 秒的时间内上行链路 Ls_i 所发出的所有分组总数,则 T 秒内 Ls_i 的平均分组发送率为: $UL_{s_i}(T)/T$.假设系统调度能达到稳定,则当 $T \rightarrow \infty$ 时, $UL_{s_i}(T)/T \approx \lambda_{s_i}$.仿真中得知 $T = 0.5$ 秒时即可达到理想的结果.令 $\tilde{\lambda}_{s_i}(Ts_i)$ 为时间 Ts_i 时各上行链路中分组到达率 λ_{s_i} 的估计值, Ts_i 与 Tm_i 为各 Ls_i 与 Lm_i 在 T 秒前的最后一次轮循的时间.定义上次轮循时刻 $T_{\text{last_poll}}(Ls_i)$ 与 $T_{\text{last_poll}}(Lm_i)$,分别为上下行链路上次被调度器轮循的时刻,调度器将维护 $2N$ 个上次轮循时刻值.按照这样的过程, T 秒时主节点中的调度器可以获得以下各组数值:

(1) 各上行链路分组到达率的估计值: $[\tilde{\lambda}_{s_1}(Ts_1), \tilde{\lambda}_{s_2}(Ts_2), \dots, \tilde{\lambda}_{s_N}(Ts_N)]$;

(2) 各上行链路发出的分组总数: $[UL_{s_1}(T), UL_{s_2}(T), \dots, UL_{s_N}(T)]$;

(3) 各下行链路发出的分组总数: $[UL_{m_1}(T), UL_{m_2}(T), \dots, UL_{m_N}(T)]$;

(4) 各链路的上次轮循时刻: $[T_{\text{last_poll}}(Ls_1), T_{\text{last_poll}}(Ls_2), \dots, T_{\text{last_poll}}(Ls_N)]$ 与 $[T_{\text{last_poll}}(Lm_1), T_{\text{last_poll}}(Lm_2), \dots, T_{\text{last_poll}}(Lm_N)]$, 即 $[Ts_1, Ts_2, \dots, Ts_N]$ 与 $[Tm_1, Tm_2, \dots, Tm_N]$.

2.2 主调度过程

结束分组到达率估计过程后即开始主调度过程,开始时刻为 T 开始的第一个主节点发送时隙.主节点的调度器将按照下述的 4 个子过程与规则选择出最需轮循的从节点,完成数据传输与更新并计算下一次主调度过程的最佳进行时间.

2.2.1 动态轮循选择规则

调度器首先计算 N 个上行链路分别在其上次轮循时刻 $T_{\text{last_poll}}(Ls_i)$ 到当前时刻 (t 时) 所经过的时间 $[t - T_{\text{last_poll}}(Ls_i)]$ 内到达的分组总数的估计值:

$$\tilde{Q}(Ls_i, t) = \tilde{\lambda}_{s_i}(T_{\text{last_poll}}(Ls_i)) \cdot [t - T_{\text{last_poll}}(Ls_i)] \quad (7)$$

根据 2.1 节所述,主调度过程首次工作时, $T_{\text{last_poll}}(Ls_i)$ 的初值为 Ts_i .调度器再分别统计当前时刻 N 个下行链路队列中等待发送的分组数: $Q(Lm_i, t)$.则 t 时,链路对 L_i 中分组总数的估计值为:

$$\tilde{Q}(L_i, t) = \tilde{Q}(L_{S_i}, t) + Q(L_{m_i}, t) \quad (8)$$

令 $F_i(t) = \tilde{Q}(L_i, t) \setminus \sum_{i=1}^N \tilde{Q}(L_i, t)$, 则 $F_i(t)$ 表示 t 时各链路对的上下行链路中分组总数的估计值占所有链路对的队列中分组总数估计值之和的比例。这样 $F_i(t)$ 体现了各链路对在当前时刻拥有数据分组数量的比例, 为满足调度的效率因素以及降低单一链路对的时延, 应该选择具有最大 $F_i(t)$ 值的链路对进行轮循。

此外, 为兼顾调度的公平性并减小整体时延, 还需要计算各链路对中的上下行链路自最后一次轮循到现在所经历的时间间隔之和, 即:

$$\Delta t(L_i) = t - T_{\text{last_poll}}(L_{S_i}) + t - T_{\text{last_poll}}(L_{m_i}) \quad (9)$$

令 $D_i(t) = \Delta t(L_i) \setminus \sum_{i=1}^N \Delta t(L_i)$, 则 $D_i(t)$ 表示各链路对在当前已等待轮循的时间的比例。为满足调度的公平性, 减小整体时延, 避免某些链路对产生饥饿, 拥有最长 $\Delta t(L_i)$ 的链路对应该得到本次轮循机会。

综合考虑调度的效率与公平因素, 定义调度的整体优先度如下:

$$P_i = \alpha F_i(t) + (1 - \alpha) D_i(t), \quad 0 \leq \alpha \leq 1 \quad (10)$$

具有最大 P_i 值的链路对为本次应该得到轮循机会的链路对。其中的 α 用来控制 $F_i(t)$ 与 $D_i(t)$ 的权重, 当 α 接近 1 时调度系统倾向于效率优先, 当 α 接近 0 时调度系统倾向于公平性优先。 α 的取值将在 2.2.4 节中描述。若各链路对的 P_i 值相同, 则随机选择某链路对进行轮循。

2.2.2 数据交换过程

假设按照上述的方法选择出本次应该轮循的链路对为 L_j , 则 t 时主节点向从节点 s_j 发送队列 L_{m_j} 中首个数据分组, 若 L_{m_j} 为空, 则仅发送单时隙的 POLL 分组。假设 m 向 s_j 发送的分组所占时隙为 $slot_m$, 则当 $\omega = t + 0.000625 \cdot slot_m$ 时, 从节点 s_j 向主节点发送队列 L_{S_j} 中首个数据分组。若 L_{S_j} 为空, 则返回单时隙的 NULL 分组。假设 s_j 向 m 发送的分组所占时隙为 $slot_s$, 当 $\theta = t + 0.000625 \cdot (slot_m + slot_s)$ 时, L_j 的调度结束。该过程同样类似 Round Robin 机制, 即上下行链路最多只发送一个分组。这样可以避免由于持续为某一个链路对进行服务而影响整体时延与公平性, 仿真结果证明了采用该机制的有效性。

2.2.3 更新数据

时间 $\theta = t + 0.000625 \cdot (slot_m + slot_s)$ 时, 调度器更新本次调度引起的初值变化:

(1) 更新上行链路 L_{S_j} 在时间 $\omega = t + 0.000625 \cdot slot_m$ 秒内发送的所有分组总数: $UL_{S_j}(\omega) = p_s + UL_{S_j}(T_{\text{last_poll}}(L_{S_j}))$, 其中 p_s 为本次发送的分组数, $p_s = 1$ 或 p_s

$= 0$ 。主调度过程首次工作时, $UL_{S_j}(T_{\text{last_poll}}(L_{S_j}))$ 的初值为 $UL_{S_j}(T_{S_i})$ 。

(2) 更新下行链路 L_{m_j} 在时间 t 秒内发送的所有分组总数: $UL_{m_j}(t) = p_m + UL_{S_j}(T_{\text{last_poll}}(L_{m_j}))$, 其中 p_m 为本次发送的分组数, $p_m = 1$ 或 $p_m = 0$ 。主调度过程首次工作时, $UL_{m_j}(T_{\text{last_poll}}(L_{m_j}))$ 的初值为 $UL_{m_j}(T_{m_i})$ 。

(3) 更新上行链路 L_{S_j} 的上次轮循时刻:

$$T_{\text{last_poll}}(L_{S_j}) = t + 0.000625 \cdot slot_m;$$

(4) 更新下行链路 L_{m_j} 的上次轮循时刻:

$$T_{\text{last_poll}}(L_{m_j}) = t;$$

(5) 由于调度器无法知道上行链路的队列中分组到达的情况, 所以每次轮循后需要根据所获得的最新信息重新计算一次上行链路 L_{S_j} 中分组到达率的估计值:

$$\mathbb{X}_j(T_{\text{last_poll}}(L_{S_j})) = UL_{S_j}(T_{\text{last_poll}}(L_{S_j})) / T_{\text{last_poll}}(L_{S_j}) \quad (11)$$

2.2.4 计算下次轮循的时间

链路对 L_j 轮循结束后, 调度器应该根据当前情况分析是否立即根据动态轮循选择规则进行下次轮循还是经过一定时隙等待后再进行下次轮循。在微微网负载较轻时, 如果相邻两次轮循相继发生将可能浪费时隙, 因为此时由于网络负载较轻, 各链路对的队列中可能没有任何数据分组等待发送, 这样将增加 POLL-NULL 分组发生的次数, 影响信道利用率并增加蓝牙的功耗。通过分析仿真研究, 提出的解决策略为: 如果本次调度出现了 POLL-NULL 分组, 则按照如下的自适应轮循间隙规则进行处理:

自适应轮循间隙规则: 令 Ψ_t 为微微网在时间 θ 时所有链路分组到达率之和乘 γ :

$$\Psi_0 = \gamma \cdot \left[\sum_{i=1}^N (UL_{S_i}(T_{\text{last_poll}}(L_{S_j})) + UL_{m_i}(\theta) + Q(L_{m_i}, \theta)) \right] / \theta \quad (12)$$

式中 γ 为调整系数, 通过仿真确定 $\gamma = 0.8$ 为理想值。

θ 时各链路对中上下行链路分别在其上次轮循时刻到现在所经过的时间内到达的分组总数的估计值为

$P = \sum_{i=1}^N \tilde{Q}(L_i, \theta)$ 。若 $P < \Psi_0$, 则 θ 时不进行任何调度, 直到 2 个时隙后主节点时隙到来时再重新计算 P 值, 若依然满足 $P < \Psi_0$, 则继续重复上述规则; 若 $P \geq \Psi_0$, 则立即开始本次调度。

此外, 定义 $\alpha = 1 - \Psi_0$, 这样式 (10) 能够成功实现本调度算法在各传输模型的大部分流量情况下时延、信道利用率与公平性的最佳统一。 α 成为一个变量, 能够随时根据网络特性得到自适应的调节。从仿真研究得知 $\gamma = 0.8$ 能够得到在混合传输与下行传输模型下最低的时延, 在上行传输模型中也能得到较低的时延。

同时, 在各种传输模型中, $\gamma = 0.8$ 也能得到较高的信道利用率与公平性. 如果需要满足特定传输模型的需要, 可以灵活的调整 γ 的值.

3 仿真与性能评估

3.1 性能指标定义

目前的蓝牙规范并没有统一的标准来评估蓝牙微网调度算法的性能. 本文对一些传统网络调度算法的评估指标经过调整与重新定义, 提出了三种主要的评估蓝牙微微网调度算法的性能指标.

平均分组时延: 定义为数据分组自最后一个位进入队列后, 到最后一个位被发送后在队列中等待时间的平均值.

信道利用率- η : 部分体现调度算法的效率, 定义为调度过程中用于传输数据分组的时隙总数与所有时隙总数的比例. 本文采用无效时隙占总时隙的比例(记为 W)来表示 η , 即 $W = 1 - \eta$. 定义为调度过程中出现的 POLL 或 NULL 分组占用的时隙总数与所有时隙总数的比例. 该值越低则说明调度算法越有效.

公平性: 本文采用文献[12]提出的其中一个方法来定义蓝牙微微网调度算法的公平性:

$$\text{Fairness index} = \left[\sum_{i=1}^N f(L_i) \right]^2 / \left[\sum_{i=1}^N f^2(L_i) \right] \quad (13)$$

式中: $f_i = \begin{cases} bw_i / BW_i & (bw_i \leq BW_i) \\ 1 & (bw_i > BW_i) \end{cases}$, bw_i 表示链路对 L_i

实际得到的带宽与各链路对实际得到的带宽之和的比例, BW_i 表示链路对 L_i 应该得到的带宽与各链路对应应该得到的带宽之和的比例. Fairness index 的范围为 $[0, 1]$, Fairness index 越接近于 1, 则表明算法公平性越高.

3.2 仿真结果与分析

本算法的仿真是与其他算法在完全一样的流量环境中实现的. 仿真模型由 1 个主节点与 7 个从节点构成, 各链路采用如下的泊松分组到达率:

上行链路: $\lambda_{1-3} = v_1$, $\lambda_{4-7} = v_2$, 下行链路: $\lambda_{1-3} = v_3$, $\lambda_{4-7} = v_4$. 通过 v_1, v_2, v_3 与 v_4 的不同值可以获得式(1)定义的各链路对协方差的不同值.

此外, 仿真中设定 DH1、DH3、DH5 分组出现的概率相等. 假设调度过程中每个数据分组均能正确地传送.

3.2.1 混合传输模型

混合传输模型即同时存在上下行传输, 这是比较常见的蓝牙微微网应用场景. 仿真中 $v_1 = v_3, v_2 = v_4$, 表示上/下行链路的流量相等.

图 1 说明了 $\rho = 0.4$ 时, 平均分组时延随各链路对分组到达率协方差(COV)的变化关系. 由图可知本算法得到了最低时延, 而且明显优于其他 3 种算法. 时延比 FEP^[11] 算法降低了 24.1%, 此外, 在 $\rho = 0.8$ 时, 比

ERR^[5] 降低了 44.6%. 图 2 表示在 $\text{COV} = 0.6$ 时, 时延随负载的变化关系, 本算法依然得到最低的时延值. 图 3 说明了 $\rho = 0.8$ 时无效时隙比例(W)随 COV 的变化关系. 结合其他仿真结果可知: 本算法随 COV 与 ρ 的增加, 信道利用率明显提高. 在保持较低分组时延的情况下仍然得到较高的信道利用率. 图 4 对比了不同 ρ 时公平指数(Fairness index)随 COV 的变化关系. 仿真结果表明本算法得到了最高的 Fairness index.



图 1 $\rho=0.4$ 时, 混合传输模型中时延随 COV 的变化

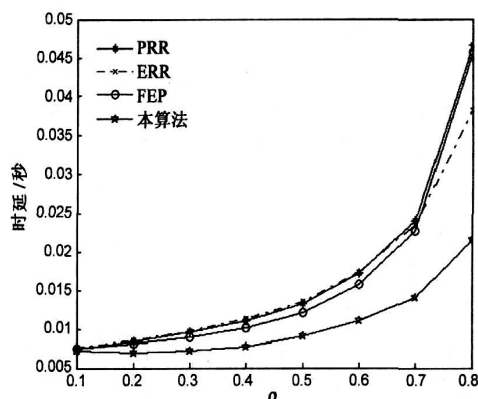


图 2 $\text{COV}=0.6$ 时, 混合传输模型中时延随 ρ 的变化

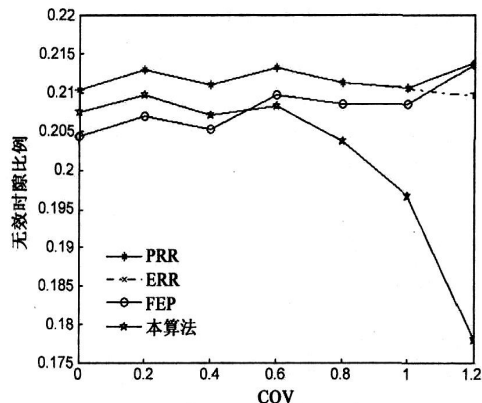


图 3 $\rho=0.8$ 时, 混合传输模型中 W 随 COV 的变化

3.2.2 下行传输模型

下行传输模型即只存在主节点到从节点的下行链路传输, 典型的应用情形有 FTP 等. 由图 5 与图 6 可知,

于其他 3 种算法. $\rho=0.4$ 时, 本算法的时延比 FEP 算法降低了 41.4%. 此外, 在 $\rho=0.8$ 时, 比 ERR 降低了 60.8%. 图 7 与图 8 的信道利用率对比中, 本算法也得到了最佳值, 即拥有最低的无效时隙比例. $\rho=0.4$ 时, 本算法的无效时隙比例比 FEP 算法降低了 71.5%, 在 $\rho=0.8$ 时, 比 ERR 降低了 19.6%. 图 9 与图 10 对比了本算法与其他算法的 Fairness index, 本算法均得到了最高的 Fairness index, 说明算法具有最高的公平性.

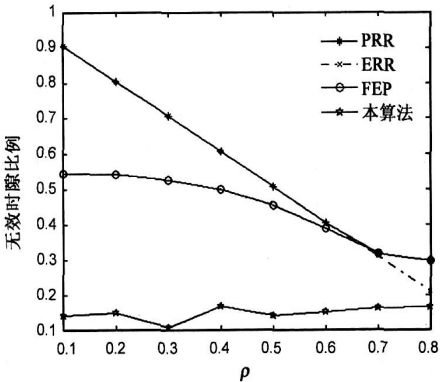


图 8 COV=0.6 时, 下行传输模型中 W 随 ρ 的变化

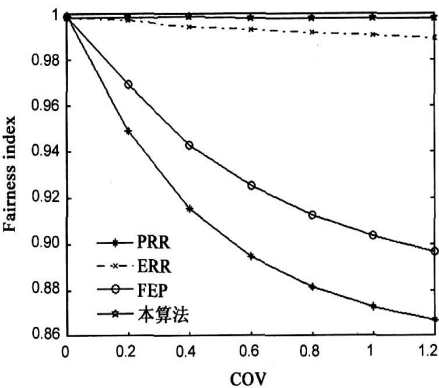


图 9 $\rho=0.8$ 时, 下行传输模型中公平指数随 COV 的变化

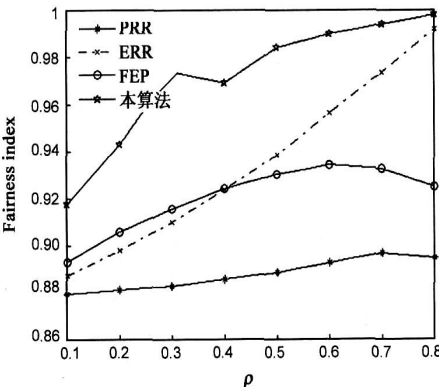


图 10 COV=0.6 时, 下行传输模型中公平指数随 ρ 的变化

3.2.3 上行传输模型

上行传输模型即只存在各从节点到主节点的上行链路传输. 由仿真结果可知, 本算法在 ρ 为 0.4 时得到了最低的平均分组时延, 在 ρ 为 0.8 时, 其时延性能略

差于 ERR 算法. 在信道利用率对比中, 当 ρ 为 0.4 时, 本算法与 PRR、ERR 性能相近, 略差于 PER 算法. 当 ρ 为 0.8 时, 本算法与 ERR 算法的性能相近, 优于其他两种算法. 在仿真中发现: 若略微调节式(12)中的 γ 值即可显著提升本算法的信道利用率. 此外, 本算法均得到了较高的 Fairness index, 与 ERR 算法差异并不显著.

4 总结

蓝牙微微网因其自身的特殊性质使其调度算法在很大程度上不同于其他常规的无线网络调度算法. 与其他算法相比, 本算法成功地在下行传输与混合传输模型中获得了时延、信道利用率与公平的明显最佳值, 在上行传输模型中也获得了较好的性能. 值得注意的是: 本算法在大部分情况下同时提高了效率与公平性. 算法提出的两个执行过程有效地使整体性能得到保证, 在主调度过程中提出的自适应轮循环间隙规则在保持一定的时延指标的要求下有效地降低了系统的无效数据时隙与 POLL-NULL 分组总数, 提出的动态轮循选择规则能够按照优先度的大小去轮循链路对. 本算法非常适合蓝牙混合传输与下行传输的应用.

参考文献:

- [1] Bluetooth Special Interest Group. Core specification of the Bluetooth system, version 2.0+EDR [OL]. <http://www.bluetooth.org>. 2004. 11.
- [2] H S Liron, R Kofman, G Zussman et al. Load adaptive inter piconet scheduling in small scale bluetooth scatternets [J]. IEEE Communications Magazine. 2004, 42(7): 136–142.
- [3] 杨帆, 王珂, 钱志鸿. 蓝牙分组传输性能分析与自适应分组选择策略 [J]. 通信学报, 2005, 26(9): 97–102.
- [4] 钱志鸿, 杨帆, 周求湛. 蓝牙技术原理开发与应用. 北京: 北京航空航天大学出版社. 2006, 3(1): 2–15.
- [5] Ka Lok Chan, Vojislav B. Misić, Jelena Misić. Efficient polling schemes for bluetooth picocells revisited [A]. Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) [C]. Big Island, Hawaii: IEEE, Vol. 9, 2004: 90307. 2.
- [6] Y-Z Lee, R Kapoor, M Gerla. An efficient and fair polling scheme for Bluetooth [A]. In Proceedings MILCOM 2002 [C]. Anaheim: IEEE, 2002, 2: 1062–1068.
- [7] R Bruno, M Conti, E Gregori. Bluetooth: architecture, protocols and scheduling algorithms [J]. ACM Cluster Computing 5, April, 2002: 117–131.
- [8] G Zussman, U Yechiali, A Segall. Exact probabilistic analysis of the limited scheduling algorithm for symmetrical bluetooth piconets [A]. Proc. IFIP TC6 PWC'03 [C]. Venice, Italy. LNCS Vol. 2775. Springer. Sept. 2003.
- [9] M Kalia, D Bansal, R Shorey. Data scheduling and SAR for Bluetooth MAC [A]. Proc. of IEEE 51st Vehicular Technology Conference (VTC 2000: Spring) [C]. Tokyo, Japan: IEEE. Vol. 2: 716–720.
- [10] M Kalia, D Bansal, R Shorey. MAC scheduling and SAR policies for bluetooth: a master driven TDD Picocellular wireless system [A]. In MoMUC [C]. May 1999. 384–388.
- [11] Johansson N, Korner U, Johansson P. Performance evaluation of scheduling algorithms for Bluetooth [A]. Proceedings of BC'99 IFIP TC 6 Fifth International Conference on Broadband Communications [C]. Hong Kong: Kluwer Academic Press, 1999: 139–150.
- [12] R Jain, D Chiu, W Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems [R]. DEC Research Report TR 301, September 1984.
- [13] Rohit Kapoor, Andrea Zanella, Mario Gerla. A fair and traffic dependent scheduling algorithm for bluetooth scatternets [J]. Mobile Networks and Applications, 2004, 9(1): 9–20.
- [14] Jelena Misić, K L Chan, V B Misić. Performance of TCP traffic in Bluetooth piconets: a queueing theoretic analysis [R]. University of Manitoba, Winnipeg, Manitoba, Canada, July 2004.
- [15] J Misić, K L Chan, V. B. Misić. Performance of Bluetooth piconets under E-rtimed scheduling [R]. University of Manitoba, Winnipeg, Manitoba, Canada, May 2003.
- [16] J B Lapeyrie, T Turetli. FPQ: a fair and efficient polling algorithm with QoS support for Bluetooth piconet [A]. In Proceedings 22nd Annual Joint Conference of the IEEE Computer and Communications Societies IEEE INFOCOM 2003 [C]. vol. 2, New York, NY, Apr. 2003: 1322–1332.

作者简介:



杨帆 男, 1977 年出生于上海, 吉林大学通信工程学院博士研究生. 主要研究方向为 Ad Hoc 网络、无线传感器网络、蓝牙技术理论研究及应用等.

E-mail: yangfan@email.jlu.edu.cn



王珂 男, 1955 年出生于河北秦皇岛, 博士, 吉林大学通信工程学院教授, 博士生导师. 主要研究方向为无线通信理论及应用、无线导航及数据处理等.