

基于 WIFI 的公交车载数据传输调度策略

孙彬^a, 陈朋^b, 冯骏^b, 俞立^b

(浙江工业大学 a. 计算机学院; b. 信息工程学院, 杭州 310014)

摘要: 为降低车辆的整体能耗并减少数据传输时间, 提出一种基于无线宽带(WIFI)的公交车载数据传输调度策略。建立公交车载无线传输问题的数学模型, 应用遗传算法对其进行求解, 得到优化的车辆传输序列。根据该传输序列, 结合相应的流程对车辆的传输进行调度。实验结果表明, 该调度策略与以先来先服务为基本原则的传统方式相比, 能节省 16.8% 的能耗, 减少 38% 的高优先级数据传输时间。

关键词: 无线宽带; 公交车载; 调度策略; 遗传算法

Schedule Strategy for Public Transport Vehicle Data Transmission Based on WIFI

SUN Bin^a, CHEN Peng^b, FENG Jia-qin^b, YU Li^b

(a. College of Computer; b. College of Information Engineering, Zhejiang University of Technology, Hangzhou 310014, China)

【Abstract】 In order to reduce the cost of power and the time of transmission in the public transport vehicle wireless data transmission, this paper proposes a schedule strategy based on Wireless Fidelity(WIFI). The strategy models a solution by applying genetic algorithm to obtain a public transport vehicle sequence which satisfies the schedule process. Simulation results demonstrate that compared with the traditional First Come First Served(FCFS) policy, the proposed schedule strategy performs much better by saving 16.8% power cost and reducing 38% high priority data transmission time.

【Key words】 Wireless Fidelity(WIFI); public transport vehicle; schedule strategy; genetic algorithm

DOI: 10.3969/j.issn.1000-3428.2011.14.093

1 概述

基于无线宽带(Wireless Fidelity, WIFI)的公交车载无线数据传输是指公交车载嵌入式设备通过 WIFI 局域网连接服务器, 利用 FTP 协议传输公交车所需的更新数据。目前, 由于缺乏合理的调度策略, 车辆的传输通常采用先来先服务(First Come First Served, FCFS)的基本原则^[1]。但由于服务器资源有限, 当车辆较多时该方法易导致服务器负载, 影响传输效率, 增加车辆能耗, 在固定时间内无法完成高优先内容的传输。

因此, 要解决的问题可简要描述为: 在一个具有固定带宽的服务器上, 对于等待传输的车辆给出一个合理的传输序列, 在最短的时间内, 完成车辆的更新。文献[2]提出在线 FTP 问题, 可以被抽象为单机调度问题, 目标是加权总完工时间最小。文献[3]提出了多客户竞争共同资源这类机器调度问题的模型。但由于公交车载无线数据传输问题的一些特点, 如调度以总能耗最小为目标以及高优先数据的优先传输, 需要综合考虑所有车辆的等待时间和工作时间。目前的模型和方法并不能够完全符合实际的需要。为此, 本文提出一种基于 WIFI 的公交车载数据传输调度策略。

2 公交车载无线传输调度策略

2.1 基本思想

车辆在传输过程中分为下载和等待 2 个状态, 由于前者比后者更加耗能, 因此本文提出利用服务器收集车辆的传输信息, 通过检测当前服务器剩余带宽是否满足其最大传输速度的方式, 决定车辆的传输状态。该方式避免了服务器的负载, 同时减少了车辆的下载时间, 节省了能耗。但是不合理的车辆传输次序会导致服务器带宽利用率不足, 增加车辆

的等待时间, 而等待时间也是影响车辆总体能耗的重要因素。因此, 调度策略的目标是得到一组优化的车辆传输序列, 在保证车辆的最大传输速度的条件下, 减少等待时间, 满足高优先级内容优先下载的同时节省车辆在更新过程中的能耗。

2.2 调度流程

公交车载无线传输调度流程如图 1 所示。

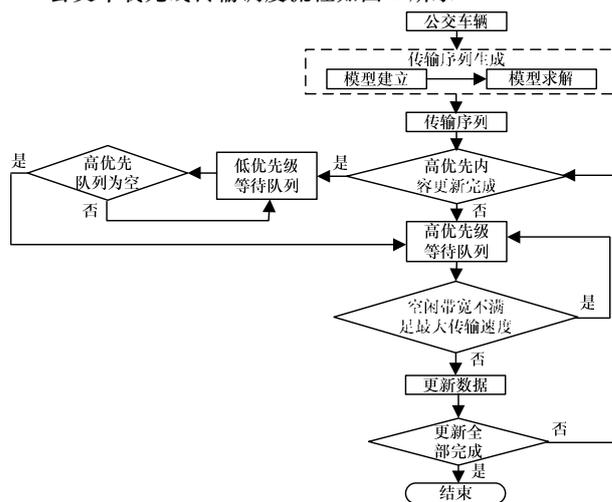


图 1 公交车载无线传输调度流程

基金项目: 浙江省自然科学基金资助项目(Y1080163); 浙江省教育厅重大科技攻关计划基金资助项目(ZD2007003)

作者简介: 孙彬(1986—), 男, 硕士研究生, 主研方向: 网络传输调度; 陈朋, 讲师、博士; 冯骏, 硕士研究生; 俞立, 教授、博士、博士生导师

收稿日期: 2011-01-06 E-mail: fanmzhai@163.com

公交车载无线传输调度策略分为2个阶段:

(1)第1个阶段: 服务器通过Socket通信收集车辆的最大传输速度, 更新数据大小等信息。针对公交车载传输调度问题的特点, 建立调度优化模型并求解, 得到优化的传输序列。

(2)第2个阶段: 车辆根据传输序列依次访问服务器下载高优先级的数据, 完成后进入等待队列, 直到场站内的所有车辆的高优先级数据更新完成, 才进行其他数据的更新, 这样可以保证在尽可能短的时间内完成高优先级数据的更新。

3 模型的建立与求解

3.1 模型的建立

调度优化的过程即是求解在满足前文约束条件的前提下使得车辆等待时间最小的传输序列的过程。离散决策变量 $x_i(x_i \in [1, n])$ 表示待更新的车辆, 与之对应的 $y_i(y_i \in [1, n])$ 表示该车在服务器上进行数据传输的次序即工序, 状态节点 (x_i, y_i) 表明任务 x_i 在工序 y_i 进行更新, 满足 $x_i \neq x_j, y_i \neq y_j, \forall i, j \in [1, n], i \neq j$ 。最小化目标函数 $f: S \rightarrow \mathbf{R}^+$, 其中 S 为解空间, $S = \{s = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}\}$, 其优化目标可描述为:

$$F = \min f(s), s \in S \quad (1)$$

公交车载无线传输问题的数学表述如下, 各变量定义如表1所示。

表1 变量定义

符号	定义
P	服务器带宽
D	当前下载速度和
N	公交车数
I	公交车集合
v_i	第 <i>i</i> 辆公交车的最大传输速度
A_i	第 <i>i</i> 辆公交车的待下载更新数据量
w_i	第 <i>i</i> 辆公交车的等待时间
C_i	第 <i>i</i> 辆公交车在传输过程中花费的总时间
R	当前正在下载的公交车集合, $R \subset I$
T	当前正在下载的公交车花费的总时间集合
U	已经完成下载的公交车集合, $U \subset R$
Z	已经完成下载的公交车花费的总时间集合

$$C_i = A_i \div v_i + w_i \quad (2)$$

为了保证车辆的最大传输速度, 当前下载速度总和不能大于服务器的总带宽, 有:

$$D = \sum_{i \in R} v_i \leq P \quad (3)$$

当 $t_{i-1} \in T$ 时表明轮到第*i*辆车准备进入下载, 它是否能进入下载取决于下载队列中最快结束下载的车次的传输速度和当前的下载速度总和。如果当前下载总流量与第*i*辆车的最大传输速度和不大总带宽即 $D + v_i \leq P$, 则第*i*辆车可以立即进入下载队列进行下载, 第*i*辆车的等待时间与它前一辆车相同, 即为 $w_i = w_{i-1}$ 。

当 $D + v_i > P$ 时, *i*车需要等待前续车辆完成下载, 释放带宽。设下载队列中最快结束下载的车次为 a , 则 $t_a = \min(T)$, $a \in R$, 当其下载结束, 服务器的空闲带宽能满足第*i*辆车的最大传输速度, 有:

$$D - v_a + v_i \leq P, D + v_i > P \quad (4)$$

则*i*车的等待时间为 $\min(T)$ 。

如果直到车次*k*下载完成时, 服务器的空闲带宽才满足*i*车的最大传输速度, 有:

$$D - \sum_{x \in U} v_x + v_i \leq P, D + v_i - v_k - \sum_{x \in U} v_x > P \quad (5)$$

则*i*车的等待时间为 $\min(T - Z)$ 。

综上所述, 第*i*辆车的等待时间为:

$$w_i = \begin{cases} w_{i-1}, D + v_i \leq P \\ \min(T - Z), P + \sum_{x \in U} v_x \leq D + v_i < P + \sum_{x \in U} v_x + v_k \end{cases} \quad (6)$$

由式(5)可得模型的目标为所有车辆等待时间之和, 目标函数定义如下:

$$f = \sum_{i \in I} w_i \quad (7)$$

3.2 模型的求解

本文采用遗传算法, 根据车辆的一组传输序列作为个体, 对模型进行求解, 得到一组目标函数值为最小等待时间的传输序列解^[4-5]。

遗传算法的主要步骤如下:

(1)首先设定群体规模为 $N_{pop-size}$, 最大进化代数数为 $M_{max-gen}$, 交叉概率为 P_c 和变异概率为 P_m , 车辆集合为 $\{1, 2, \dots, n\}$, 随机产生 $N_{pop-size}$ 个 $1, 2, \dots, n$ 的整数排列作为初始群体 $\{V_1, V_2, \dots, V_{N_{pop-size}}\}$ 。

(2)计算适应度的值。模型的目标函数式(3)作为遗传算法的适应度函数:

$$f(V) = \sum_{i=1}^n w_i \quad (8)$$

其中, $f(V)$ 是个体 V 的适应度。

(3)采用赌轮盘方式从种群中选择个体。每个个体选则概率 $P(V)$ 与其适应值呈反比:

$$p(V) = \frac{\sum \frac{1}{f(V)}}{f(V)} \quad (9)$$

(4)对种群进行交叉和变异操作。采用交换子串的交叉算子产生一个 $[0, 1]$ 区间的随机数, 如果该随机数小于交叉概率 P_c , 则首先随机选择2个个体作为父代; 然后在基因串上随机选择2点作为交叉点, 交换2个父代基因串上交叉点之间的子串, 形成2个原型子代; 比较2个原型子代在两个交叉点之间的子串, 找出对于2个原型子代来说是多余和没有的基因; 随机用等位基因将多余的基因用没有的基因替换, 形成2个子代; 最后重复该过程, 直到得到满足约束的子代为止。采用移动变异算子, 产生一个 $[0, 1]$ 区间的随机数, 当随机数小于变异概率 P_m , 则随机选择一个个体作为父代, 在基因串上随机选择2点作为变异点, 交换2点的基因值。

(5)令迭代次数加1, 继续计算适应度值, 保留历史最优解。如到达最大值代数则终止并输出最优解, 否则转步骤(2)。

主要计算流程如图2所示。

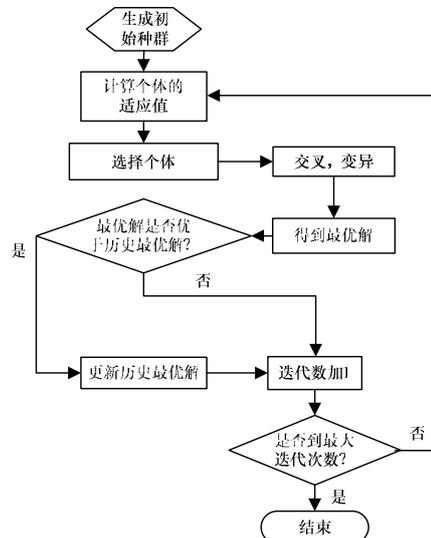


图2 遗传算法流程

4 实验结果与分析

将上述算法应用于如下典型的公交车载无线传输调度问题: 设服务器总带宽上限为 2 MB; 在实际工程中, 车载 WIFI 模块工作时的功率为 500 mW, 而等待时功率为工作时的 1/3; 车辆的传输速度主要受其与场站 AP 距离的影响, 取值为 2 MB/s~0.2 MB/s 之间。现抽取 10 辆车作为样本数据, 进行实验。车辆信息如表 2 所示。

表 2 车辆信息

车辆编号	更新数据总量/MB	高优先级数据量/MB	传输速度/(MB·s ⁻¹)
1	600	200	1.0
2	500	166	2.0
3	400	133	0.5
4	634	211	0.5
5	624	208	0.2
6	678	226	0.3
7	477	159	0.5
8	498	166	1.0
9	512	170	1.4
10	488	162	0.5

采用本文提出的传输调度优化策略, 利用遗传算法求解模型的近似最优解, 遗传算法参数设置为: 种群数目 $N_{pop-size}=50$; 交叉概率 $P_c=0.8$, 变异概率 $P_m=0.04$; 迭代次数 $M_{max-gen}=100$; 仿真在 100 次迭代后得到最优目标函数值为: 121, 所获得的最佳传输序列为: 2→8→1→3→9→4→5→6→7→10。遗传进化曲线如图 3 所示。

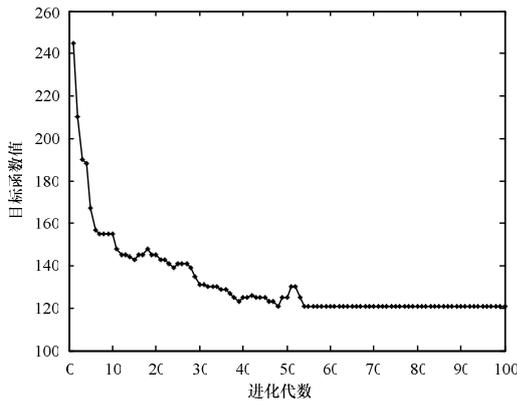


图 3 遗传进化曲线

可以看出, 曲线在进化开始时减少迅速, 而在进化末期则趋于平缓, 显示出该算法具有较高的寻优效率和较快的收敛速度, 能在短时间内求得模型的近似最优解。

下面的实验分别从所有车辆的总能耗和高优先级内容更新情况 2 个方面比较传统的 FCFS 策略和本文提出的调度优化策略的性能。2 种策略的性能比较如图 4 所示。

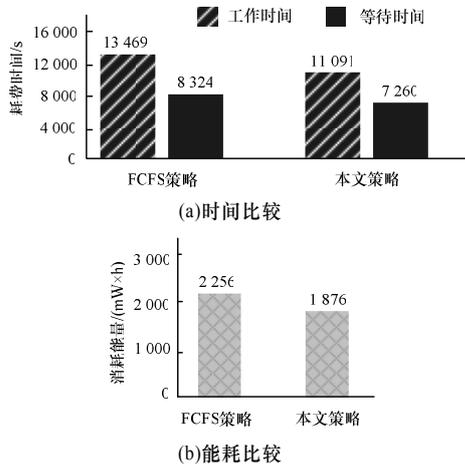


图 4 2 种策略的性能比较

可以看出, 本文提出的调度优化策略, 保证了车辆的最大传输速度。与传统的 FCFS 策略相比, 这种方式减少了车辆的工作时间和等待时间。车载设备在等待状态下更加节能, 本文提出的策略与 FCFS 策略相比节省了 16.8% 的能耗。因此, 本文提出的调度策略节约能耗方面具有更加良好的性能。

2 种策略的单位车辆所需的耗能对比如图 5 所示。

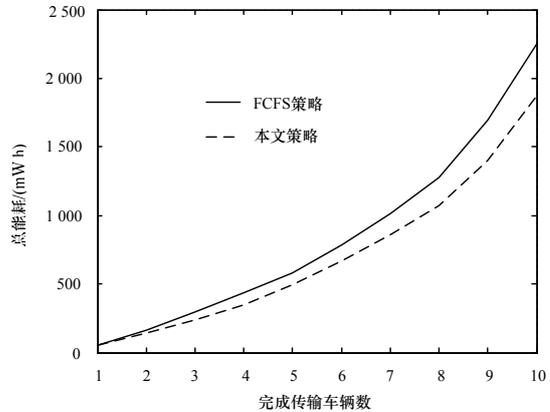


图 5 2 种策略的车辆能耗对比

可以看出, 当所需完成更新的车辆较少时, 2 种策略所消耗的能量比较接近, 节能效果并不明显; 随着车辆数目的增加, 采用本文提出的策略的能耗曲线增长幅度小于 FCFS 策略, 节能效果逐渐明显。

2 种策略的车辆完成高优先级数据时间对比如图 6 所示。

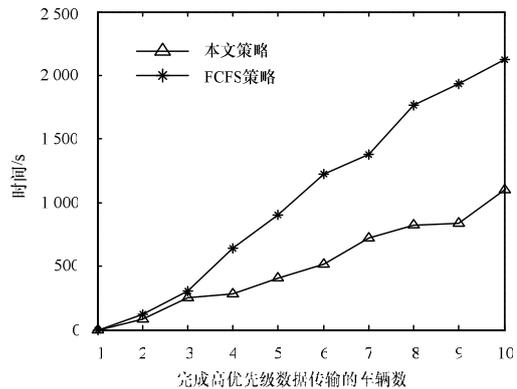


图 6 2 种策略的车辆完成高优先级数据时间对比

可以看出, 当车辆较少时, 2 种策略在完成高优先级内容的更新中所花费的时间相差不大。但是如果车数较多, 采用本文策略的完成时间曲线增长幅度小于 FCFS 策略, 所花费的时间小于 FCFS 策略。而且随着车辆数目的增加, 减少的幅度更加明显, 2 种策略完成 10 辆车的高优先级内容时间分别为 2 478 s 和 1 534 s, 本文提出的策略所需时间减少了 38%。

根据此实验样本数据, 假设有以上 10 种参数的车辆各 10 辆, 共 100 辆车进行实验。实验情况如下: 采用 FCFS 策略总共耗能约为 231.42 W·h, 完成全部高优先级内容的更新需要近 7.6 h; 而本文提出的策略总耗能为 192.07 W·h, 相比 FCFS 策略节省了约 17% 的能耗, 完成全部高优先级内容的更新约为 4.2 h。假设场站凌晨 12 点开始进行更新, 到早上 6 点更新结束开始发车, 那么采用本文提出的策略此时已经对所有车辆的高优先级内容更新成功, 保证了当日的正常播放流程。

(下转第 279 页)