

分布式排队中退避树的深度优先遍历算法

王文霖, 张延贺, 吴炜, 柏琛, 王斌

(南京邮电大学通信与信息工程学院, 江苏 南京 210003)

摘要: 分析传统分布式排队 (DQ) 的调度过程及退避树操作规则, 设计了一种深度优先遍历的改进算法。结合完全二叉树特例分析和随机重构的一般性推算, 对改进算法的系统吞吐性能进行了理论分析和仿真评估, 给出了 DQ 帧争用时隙的最优配置条件和基于开源软件 NS-3 的扩展仿真。仿真结果表明, 所提算法的最大吞吐量可稳定达到信道物理容量的 70%。

关键词: 随机多址接入; 分布式排队; 指数退避树; 深度优先搜索; 性能分析

中图分类号: TN915

文献标识码: A

DOI: 10.11959/j.issn.1000-436x.2021044

Depth first traversal algorithm for the back-off tree of distributed queuing

WANG Wennai, ZHANG Yanhe, WU Wei, BAI Chen, WANG Bin

School of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: An analytic model was provided for the conventional distributed queuing (DQ) and its back-off tree operations, followed by a design of improving algorithm based on depth first traversal. Combing the specific analysis of complete binary tree with generalized extension by random tree reconstruction, the performance of proposed algorithm was evaluated on the throughput in both theory and simulation experiment. A theoretic optimal solution of contention slots of DQ frame and a brief description of simulation extension based on the open source NS-3 were presented. The simulation results show that the maximum stationary throughput by the proposed algorithm reaches 70% of the physical capacity of channel.

Keywords: random multiple access, distributed queuing, exponential back-off tree, depth first searching, performance evaluation

1 引言

分布式排队 (DQ, distributed queuing) 是综合了时分复用 (TDM, time division multiplex) 和随机多址接入 (RMA, random multiple access) 2 种手段的媒质访问控制方法, 其最初目标是高效解决有线电视电缆的多站接入冲突^[1]。由于 DQ 的吞吐性能和重载稳定性接近理想的无冲突 M/D/1 排队系统^[2], 其近年来被扩展到各类无线接入网, 特别是终端密度高、业务突发性强、功能复杂度低的物联网 (IoT, Internet of things)^[3-8], 甚至被视为 ALOHA 技术的

终结者^[9]。

DQ 系统包含一个集中式协调者和 2 个分布式虚拟队列, 分别为争用请求队列 (CRQ, contention request queue) 和数据发送队列 (DTQ, data-transmit queue)^[1]。CRQ 的离队站点随机争用共享信道, 争用成功时进入 DTQ, 否则返回 CRQ 退避。DTQ 的离队站点以 TDM 方式无冲突地占用信道。共享信道的争用状态由协调者监测和反馈, 其功能可以由小区制基站或接入点 (AP, access point) 实现^[3-4], 也可以由自组织网络的簇头临时实现^[5]。DQ 时隙结构和争用信号方式有很大的灵活性, 能适应不同的应

收稿日期: 2020-08-11; 修回日期: 2020-12-16

基金项目: 国家自然科学基金资助项目 (No.61871234, No.61475075)

Foundation Item: The National Natural Science Foundation of China (No.61871234, No.61475075)

用环境。

文献[3]针对长期演进技术 (LTE, long term evolution) 物理随机接入信道 (PRACH, physical random access channel) 提出了一种基于前导信号的 DQ 应用方案, 发现用户设备的接入阻塞率、吞吐量和能耗都有显著优势, 尤其是在重载时阻塞率保持为零。文献[4]发现 DQ 用于窄带物联网 (NB-IoT, narrow band Internet of things) 时, 在保持 100% 接入成功率的同时, 通过资源分组能进一步减少接入时延。文献[5]针对 IEEE 802.11 WLAN 提出了一种基于请求发送 (RTS, request to send) 控制帧的 DQ 争用时隙方案, 发现吞吐量至少能提升 25%。文献[6]将 DQ 用于 Ad-Hoc 网络, 发现重载时 DQ 吞吐量可比传统的 IEEE 802.11 技术提高 85%。文献[7]针对 LoRaWAN 设计了一种 DQ 验证原型, 估算发现, 在增大 TDM 时隙的占比后, 相同性能条件下能将接入容量从 1 500 个终端提高到 5 712 个终端。本文的先期工作提出另一种 DQ 结合 LoRaWAN 的方案, 通过数值仿真发现, 与现有技术标准的接入方案相比, 吞吐量可增大 2.6 倍, 500 个终端并发时的平均时延可降低 54%^[8]。

相比于 ALOHA, DQ 表现出全方位的性能优势, 原因有以下 2 个方面: 1) DQ 将争用冲突压缩在相对短时的争用时隙; 2) DQ 采用树型退避算法以指数形式递减并发站点数。相比于早期的树型随机多址^[10], DQ 分配了多个争用时隙。争用时隙数目 m 越大, 冲突解决时效性越好, 但相应的时间开销也越大, 反而会降低有效吞吐量。文献[2]的理论分析发现, $m=3$ 是较好的折中条件。文献[8]的数值仿真发现, $m=3$ 这一条件仅适用于中等规模的多站并发。在不增大 m 的前提下提高冲突解决时效, 目前尚未见公开报道。

DQ 调度的重载性能稳定, 功能相对简单, 对以 ALOHA 技术为基础的物联网接入等新型业务有很大的应用潜力^[9], 因此受到研究者的关注。本文分析 DQ 退避树遍历的计算优化, 以进一步提高冲突解决时效性, 研究用深度优先方法代替传统的广度优先方法。

2 DQ 工作机制

2.1 系统组成与时序

ALOHA 完全由数据站组成^[10], 而 DQ 系统则引入一个集中式协调者站点。为叙述方便, 以下假

设协调者功能部署在小区中心基站或 AP, 数据站为小区终端, DQ 的调度对象是并发终端至基站上行的数据传输。

DQ 以时分复用方式共享信道, 重复的 DQ 周期包含上行争用时隙 (CS, contention slot) 和数据时隙 (DS, data slot), 以及协调者至所有终端的下行反馈时隙 (FS, feed-back slot)^[1]。CS 嵌套 m 个小时隙/子时隙 (MS, mini-slot)。DQ 时域共享信道的时隙划分结构如图 1 所示。

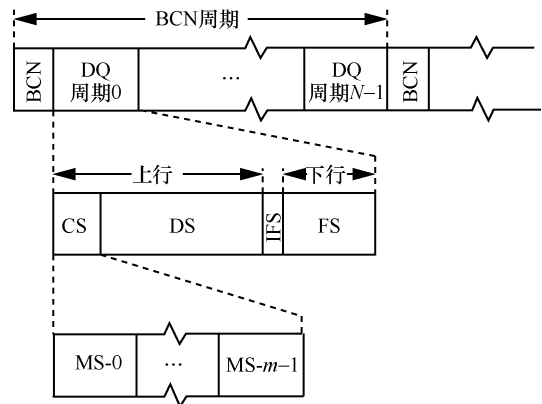


图 1 DQ 时域共享信道的时隙划分结构

图 1 中, 信标 (BCN, beacon) 广播起到定时同步的作用, 帧间空隙 (IFS, inter-frame space) 用于容纳上下行传播时延。N 个 DQ 调度周期构成一个 BCN 周期, N 值由 BCN 广播通告或者动态配置。

在一个 BCN 周期内新到的终端发送请求, 并推至下一个 BCN 周期。争用时隙包含 m 个子时隙供终端随机争用, 其中 $m \geq 2$ 。一个 DQ 调度周期只包含一个数据时隙, 多个争用成功的终端顺序排队占用多个 DQ 周期内的 DS。

静态系统配置固定的子时隙数 m , 以及固定时长的 MS、CS、DS 和 FS。

2.2 分布式排队过程

为方便说明, DQ 终端标识记为 $k, k \in [0, K_n - 1]$, 其中, K_n 为第 n 个 DQ 周期内并发争用 DS 的终端总数, $n \in [0, N - 1]$ 。争用终端 k 在 m 个 MS 中以等概率随机选择一个发送请求帧或请求信号, 选中的 MS 索引记为 $r_k, r_k \in [0, m - 1]$ 。

协调者监听 CS 内所有 MS。当仅有一个终端占用 MS 时, 由于可以正确恢复出请求信息, 因此易被判定为争用成功; 当 2 个及 2 个以上终端占用 MS 时, 由于存在互干扰, 因此易被判定为争用失败; 当没有终端占用 MS 时, 因无载波信号而易被

判定为空闲。

协调者在 FS 以广播方式反馈 MS 的监听结果或 MS 状态, 记为 a_i , $i \in [0, m-1]$ 。不失一般性地, 设 $a_i=0$ 表示空闲, $a_i=1$ 表示争用成功, $a_i=2$ 表示冲突或争用失败。

所有争用终端在每个 DQ 周期 FS 接收来自协调者的 a_i 反馈结果, 独立计算式(1)。

$$\begin{aligned}
 G &= \sum_{i=0}^{m-1} \delta(a_i, 1) \\
 C &= \sum_{i=0}^{m-1} \delta(a_i, 2) \\
 g_k &= \sum_{i=0}^{r_k} \delta(a_i, 1), a_{r_k} = 1 \\
 c_k &= \sum_{i=0}^{r_k} \delta(a_i, 2), a_{r_k} = 2
 \end{aligned} \tag{1}$$

其中, G 为争用成功的终端个数, C 为争用失败的终端组的个数, g_k 为终端 k 争用成功时的次序号, c_k 为终端 k 争用失败时的次序号。当 $a_i=n$ 时 $\delta(a_i, n)=1$, 否则 $\delta(a_i, n)=0$ 。

协调者在监听 MS 状态的同时, 同理计算 G 和 C , 并分别累加到由其集中维护的 DTQ 长度 L_T 和 CRQ 长度 L_C 。

如果 $a_i=1$, 且 $r_k=i$ 在 G 个成功者中的次序为 g_k , $g_k \in [0, G-1]$, 则终端 k 进入 DTQ 队尾, 并由该终端自身记录其在 DTQ 中的位置为

$$p_k = L_T + g_k \tag{2}$$

其中, L_T 由协调者统计和广播通告, 其随 G 递增、随 DQ 周期递减; g_k 是对 a_i 的统计结果; p_k 随 DQ 周期递减。

如果 $a_j=2$, 且 $r_k=j$ 在 C 个失败组中的次序为 c_k , $c_k \in [0, C-1]$, 则所有选中 j 的终端返回 CRQ 队尾, 并由这些终端自身记录其在 CRQ 中的位置为

$$q_k = L_C + c_k \tag{3}$$

其中, L_C 由协调者按 DTQ 相同的方法计算和通告, c_k 是对 a_i 的统计结果, q_k 随 DQ 周期递减。

式(1)~式(3)的计算依赖于协调者在 FS 的广播反馈, FS 和 m 个 MS 的总时长是分布式排队的系统开销。

2.3 冲突解决过程

队列 CRQ 和 DTQ 在 DQ 中并无物理实体, 排队功能分布在各个终端, 所以是虚拟的。 L_T 和 L_C 由协调者在 FS 中随 a_i 一起广播给所有终端。显然, 每个

DTQ 位置上最多只能有一个终端, 而每个 CRQ 位置上的一组终端至少有 2 个成员, 它们是后继 DQ 周期参与争占的终端数, 即 K_n 。

以 $m=2$ 为例, 设初始时 $K_0=18$, 有 $L_C=0, L_T=0$ 。在初始 DQ 周期 T_0 , 假设 2 个 MS 各有 9 个终端选中, 则 $G=0, C=2, L_C=2, L_T=0$ 。由式(1)~式(3)可知, 协调者将队列长度 L_C 和 L_T 以及 MS 状态以广播方式反馈给所有终端, 冲突终端依此分为两组退避重入 CRQ。结果是, CRQ 队首 2 个位置各有一组终端, 每组各有 9 个终端。因此, 在 DQ 周期 T_1 和 T_2 , 有 $K_2=K_3=9$ 。

在 DQ 周期 T_1 , 设 2 个 MS 各有 5 个和 4 个站点, 可得 $G=0, C=2, L_C=3$, 依次类推。变量 K_n 、 G 、 C 、 L_T 、 L_C , 以及终端 k 在队列中的位置 p_k 和 q_k , 取决于该终端选取 MS 的具体情况。图 2 描述了其中一种可能的过程。其中, 矩形框表示小时隙, 其中数字表示争用终端的数量, 粗边框表示冲突, 细边框表示成功; 细箭头线表示终端组的树型分割, 粗箭头线表示树的遍历次序。

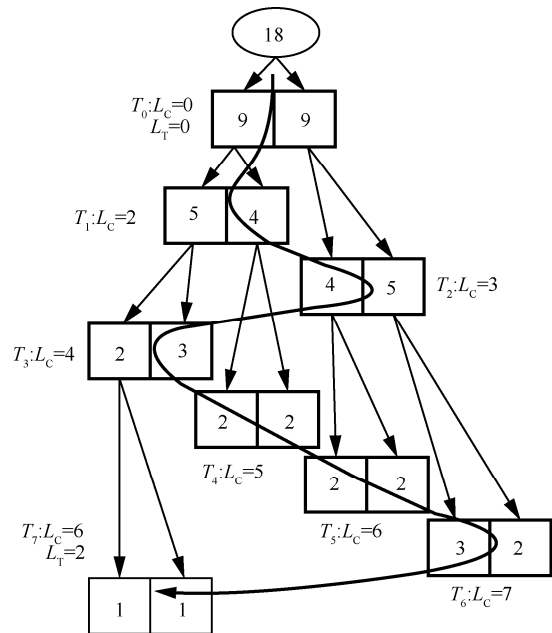


图 2 DQ 冲突退避与解决的过程示例

图 2 中, 2 个一组的矩形框表示 2 个 MS, 中间数值表示并发争用的终端数。争用失败的终端按所选 MS 的序号, 进入 CRQ 队尾退避等待和重新争用。从图 2 给出的特定时序可以看出, 在第 8 个 DQ 周期 T_7 , 冲突得到部分解决, L_C 开始减小, DTQ 有成功者入队。

需要说明的是, 图 2 对应于终端以大概率平均

选择 MS 的情况，如果终端选择出现随机偏离，将增加退避树的深度及调度的遍历时长。

3 改进算法

3.1 树遍历的优化机会

DQ 冲突退避采用了树型分割方法对一组终端迭代细分，对应的遍历次序具有宽度优先搜索 (BFS, breadth first searching) 特点。当初始终端数远大于 MS 配置数，即 $K_0 \gg m$ 时，到达冲突解决的叶节点需要遍历大量存在冲突的树的中间节点。

从图 2 可以直观地看出，在相同 MS 配置数的情况下，换用深度优先搜索 (DFS, depth first searching) 可以更快地到达冲突解决的叶子节点，竞选出 DTQ 的入队终端。图 3 描述了一个可能的遍历过程。

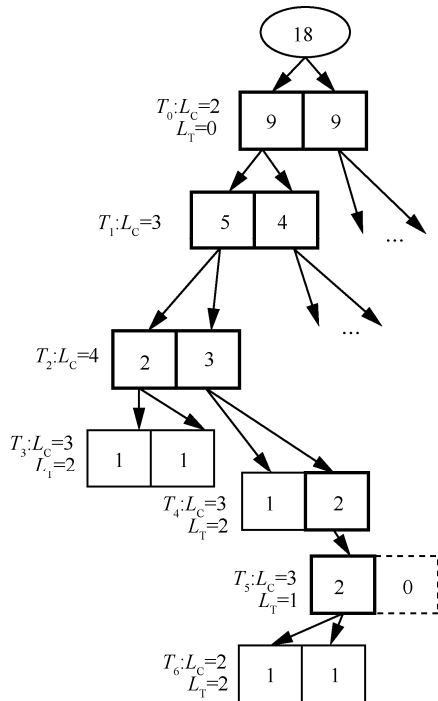


图 3 DFS 遍历冲突解决叶子节点示例

图 3 中，虚边框表示 MS 未被任何终端选中，即空闲状态。为简化绘制，图 3 主要给出 DFS 搜索至前 2 个叶子节点的过程，省略了后继的遍历细节。

从图 3 可以直观地看出，在第 4 个 DQ 周期 (T_3) 得到 2 个 DTQ 入队终端，这比 BFS 提前了 4 个 DQ 周期。因此，基于 DFS 的 CRQ 退避规则能更快地解决争用冲突。

3.2 算法设计

改进算法沿用与 BFS 相同的 MS 监听和 DTQ 调度规则。协调者在得到 MS 状态后，同样将 L_T 、

L_C 和 MS 的状态反馈给所有终端，同时进行队列长度更新。

$$L_T = L_T + G - 1 \quad (4)$$

$$L_C = L_C + C - 1 \quad (5)$$

其中， G 是争用成功的 MS 个数， C 是冲突失败的 MS 个数，减 1 对应于队首离队情况。另外，队列长度减至 0 时不再减小。

DFS 的作用主要体现在冲突时终端的 CRQ 位置更新和计算。BFS 中，终端返回 CRQ 队尾，而在 DFS 中则进入 CRQ 队首以获得优先重试机会。这就要求在 CRQ 中排队的终端后移位置，具体计算式为

$$q_k = q_k + C - 1 \quad (6)$$

其中， C 由各终端独立地从 FS 反馈信息计算得出，减 1 计算对应于排队前移。

对于从 CRQ 离队但争用失败的终端 k ，返回 CRQ 的位置就是计算前述的变量 c_k ，即终端所选 r_k 在 C 个冲突 MS 中的次序。算法 1 给出了分布的终端独立执行 DFS 退避的伪代码，其中， $a[i]=0、1、2$ 分别表示 MS 处于空闲、成功、冲突状态， $r=0,1,\dots,m-1$ 表示终端所选 MS 的索引号， q 表示终端在 CRQ 中的位置。

算法 1 CRQ 位置计算与更新的改进算法

输入 $L_T, L_C, a[i], r \quad //i=[0, m-1]$

输出 q

- 1) CRQ-POS (L_T, L_C, a, r, q) {
- 2) int $i, c = 0;$
- 3) if ($q > 0$) //已在 CRQ 排队
- 4) for ($i=0; i < m; i++$)
- 5) if ($a[i] == 2$)
- 6) $c++;$ //冲突计数
- 7) end if
- 8) end for
- 9) $q = q + c - 1;$ //后移
- 10) else // $q \leq 0$
- 11) switch ($a[r]$)
- 12) case 1: //争用成功
- 13) DTQ-POS (L_T, a, r); //出队进 DTQ
- 14) break;
- 15) case 2: //争用冲突
- 16) for ($i=0; i < r; i++$) //从 CRQ 队头
- 17) if ($a[i] == 2$)
- 18) $c++;$ //计算序号

```

19)   end if
20)   end for
21)   q = c;
22)   break;
23)   end switch case
24)   end if
25)}

```

算法 1 中 CRQ-POS 针对退避等待和争用状态分别处理。第 2)~9)行是已处于退避等待终端的位置后移,第 12)~14)行及函数 DTQ-POS 是对争用成功计算 DTQ 位置,第 15)~22)行针对争用失败按序进入 CRQ 队头。需要注意的是,以上伪代码省略了 MS 为空闭和通信出错的处理。

CRQ-POS 的唯一预设条件是在调用 CRQ-POS 之前,当终端 CRQ 位置 $q=0$ 时,该终端独立地以一致概率在区间 $[0, m-1]$ 生成随机整数,并记录在参量 r 中。

算法 1 中第 13)行调用的函数 DTQ-POS 的主要功能是确定争用成功的终端在 DTQ 中的位置,按式(2)计算。

相比于 BFS,以上基于 DFS 的改进算法的复杂度仅增加了终端在 CRQ 队内排队等待的后移计算,即算法 CRQ-POS 的第 2)~9)行。如图 3 所示,这种极小量的计算开销带来了改善系统吞吐性能的预期。当然,这里的后移计算,其前提是 CRQ 队内的退避终端要持续不断地监听和处理来自协调者的反馈信息。

4 吞吐性能对比分析

4.1 吞吐量最大条件

图 1 表示的 DQ 周期中,将 MS 的时长记为 T_M , DS、FS 和 IFS 的时长分别记为 T_{DS} 、 T_{FS} 和 T_{IFS} ,则共享信道的吞吐量为

$$S = \frac{uT_{DS}}{(u+v)(mT_M + T_{IFS} + T_{DS} + T_{FS})} \quad (7)$$

其中, u 表示有数据传输的 DS 总数, v 表示无数据传输的 DS 总数。

设一个 BCN 周期正好容纳所有终端的数据发送,则 $u=K_0$, $N=u+v$ 。

从式(7)可知, v 和 m 越小, S 越大。但 m 越小,对应于 CS 内嵌的 MS 越少,终端争用的冲突概率越大,导致 v 越大。所以,以吞吐量最大为目标,最佳 m 需满足

$$(mT_M + T_F) \left(\frac{dv}{dm} \right) + (K_0 + v)T_M = 0 \quad (8)$$

整理得到

$$\frac{dv}{dm} = -\frac{(K_0 + v)T_M}{mT_M + T_F} = -\frac{(K_0 + v)}{m + \alpha} \quad (9)$$

其中, $T_F=T_{IFS}+T_{DS}+T_{FS}$ 表示 DQ 周期扣除 CS 的时长, $\alpha=\frac{T_F}{T_M}$ 。

4.2 完全树的对比分析

图 4 给出了 $K_0=32$ 时冲突退避构成完全二叉树的特例。图 4(a)对应 BFS,图 4(b)对应 DFS。

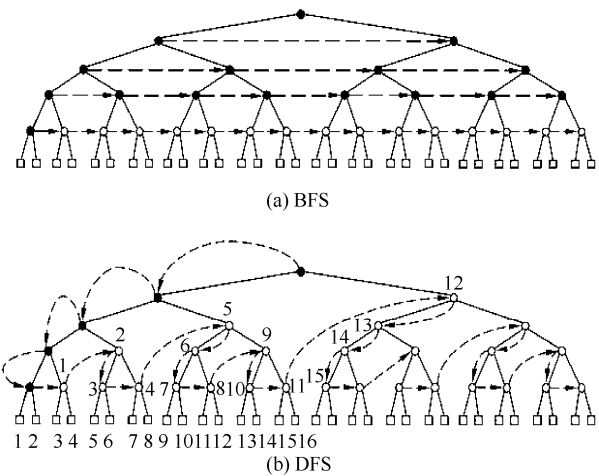


图 4 $K_0=32$ 时冲突退避构成完全二叉树的特例

图 4 中,树中圆节点表示 MS 冲突的情况,方形叶子节点表示冲突解决。树遍历过程只涉及圆节点,实心圆表示 v 的贡献项,空心圆表示因 DS 内有数据传输而对 v 无贡献的情况。

图 4(a)树根对应树的第 0 层,初始时 DS 无数据发送, $v=1$, 2 个 MS 各分 16 个终端;第 1 层 2 个中间节点,对应 DS 同样无数据发送,所以 $v=1+2=3$;第 3 层, $v=15$;第 4 层左侧第 1 个节点仍有冲突, $v=16$;第 5 层为 2 个叶子节点,所以第 4 层后继节点遍历时代 DS 有数据发送,它们对 v 无贡献。因此可得 $v=16$ 。

图 4(b)树根至最左侧叶子节点,深度为 4,其中 DS 无数据发送,所以 $v=4$ 。该树的后继遍历中,所至中间节点均有对应的叶子节点,如图 4(b)中序号为 1~16 的节点所示,它们对 v 无贡献。因此可得 $v=4$ 。

以上针对 $K_0=2^L$ 的完全二叉树,从图 4(a)表示的 BFS 遍历可得, $v=2^{L-1}=16$,从图 4(b)表示的 DFS 可得, $v=L-1=4$,其中, $L=\text{lb}K_0$ 为树的深度。对于 $m \geq 2$ 的一般情况,可得 $L=\text{log}_m(K_0)$,且

$$v_{\text{BFS}} = m^{L-1} = \frac{K_0}{m} \quad (10)$$

$$v_{\text{DFS}} = L - 1 = \log_m(K_0) - 1 \quad (11)$$

所以，有

$$\frac{dv_{\text{BFS}}}{dm} = \frac{-K_0}{m^2} \quad (12)$$

$$\frac{dv_{\text{DFS}}}{dm} = \frac{1}{K_0 \log(m)} \quad (13)$$

将式(12)代入式(9)，得到 BFS 的最优条件为

$$\frac{K_0}{m^2} = \frac{\left[K_0 + \frac{K_0}{m} \right]}{m + \alpha} \quad (14)$$

求解可得

$$m_{\text{BFS_MAX}} = \sqrt{\alpha} \quad (15)$$

代入式(7)可得

$$S_{\text{BFS_MAX}} = \frac{\frac{T_{\text{DS}}}{T_{\text{M}}}}{(1 + \sqrt{\alpha})^2} \quad (16)$$

对于 DFS，从式(13)可知，因为 $m > 1$ ，所以 $dv_{\text{DFS}}/dm > 0$ 。代入式(8)，左侧总是大于 0，所以吞吐量最大条件是 $m_{\text{DFS_MAX}} = 2$ ，有

$$S_{\text{DFS_MAX}} = \frac{K_0 T_{\text{DS}}}{(K_0 + \text{lb}K_0 - 1)T_{\text{M}}(2 + \alpha)} \approx \frac{\frac{T_{\text{DS}}}{T_{\text{M}}}}{2 + \alpha} \quad (17)$$

因为 $K_0 \gg 2$ ，所以 $K_0 + \text{lb}K_0 - 1 \approx K_0$ 。

定义加率比 $f = S_{\text{DFS_MAX}}/S_{\text{BFS_MAX}}$ ，则有

$$f = \frac{(1 + \sqrt{\alpha})^2}{2 + \alpha} \quad (18)$$

易得，当 $\alpha = 4$ 时，有最大加速比 $f = 1.5$ 。

4.3 一般随机树的对比分析

以上推算虽然只适用于完全树的特例，但 DFS 性能优于 BFS 的结论适用于一般性情况。这是因为，当 $K_0 \gg m$ 时，终端均匀争用各小时隙的概率最大，完全树分布对性能贡献权重也最大。此外，一般随机树的情况可等效于完全树的随机重构。

从 $K_0 = m^L$ 完全树出发，随机选取一个叶子节点，将其修减，得到 $K_0 - 1$ 的随机树；将其移接到其他随机选取的叶子节点，则得到一个非平衡的随机树。修减一个叶子节点，对变量 v 至多贡献一次正

计数。移接一个叶子节点，将对 BFS 的 v 增加一个计数，而对 DFS 的 v ，只在前遍历 $L_1 = 0$ 时才有一次加 1 的贡献。总体上，完全树随机重构后，基于 DFS 的冲突解决仍然快于 BFS。

当然，完全树随机重构后，式(16)和式(17)最大吞吐量条件会存在偏离。相应地，式(18)给出的加速比只可视为一个近似估计。下面，通过随机化网络仿真说明更一般的情况。

5 数值仿真验证

5.1 仿真软件设计

如前文所述，传统基于 BFS 的 DQ 调度和本文设计的基于 DFS 的改进算法采用相同的时域信道划分结构和相同的 DTQ 排队规则，而 CRQ 排队规则有所差别。因此，本文基于开源 NS-3 离散事件仿真软件，设计了的状态机和事件类型，DQ 仿真的状态迁移与事件如图 5 所示。

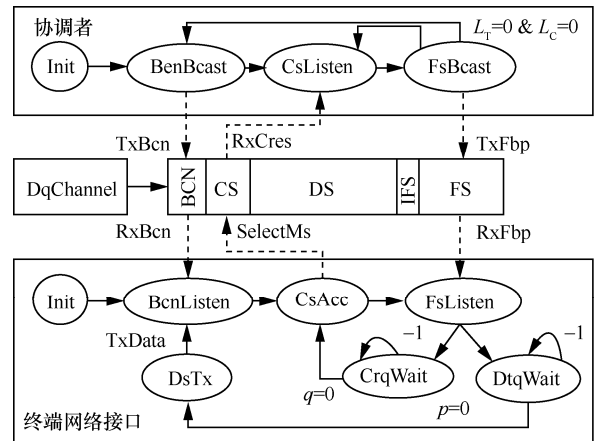


图 5 DQ 仿真的状态迁移与事件

图 5 中的仿真对象类 DqChannel 按固定的定时事件分别调用协调者和终端网络接口的接口函数，完成信标帧的收发、CS 争用与监听和反馈帧 Fbp 的收发。

协调者和终端网络接口派生于对象类 ns3::NetDevice，DqChannel 派生于 ns3::Channel。终端在争用接入 (CsAcc) 状态随机选占一个小时隙 (SelectMs)，协调者统计争用请求 (Rx Cres) 并在状态 FsBcast 发送反馈帧 (Tx Fbp)。所有终端数据发送完成后，协调者再次发送信标，以支持连续多次仿真。

终端网络接口在每个 BCN 周期内发出一数据帧，在状态 FsListen 收到反馈帧后，依据 DQ 规

则进行 CRQ 排队 (CrqWait) 或 DTQ 排队 (DtqWait)。终端属性 p 和 q 分别记录终端在 DTQ 和 CRQ 中的位置, 并利用 NS3 的变量跟踪机制提供监测和统计。当 $p=0$ 时, 终端迁移至数据发送状态 DsTx, 在紧随其后的 DQ 周期的 DS 发送数据 TxData, 然后进入状态 BcnListen 等待下一轮仿真。

针对 DQ 调度性能的仿真, 图 5 设计的简化模型假设所有数据长度固定不变, 时长均为 T_{DS} 。而共享时域信道的划分由 DqChannel 相应属性提供配置接口。本文仿真实验设置 $T_{DS}=0.3\text{ s}$, $T_{BCN}=T_{FS}=0.1\text{ s}$, $T_{IFS}=2\text{ ms}$, $T_M=0.01\text{ s}$, 相应地, $T_F=0.4\text{ s}$, $\alpha=40$ 。

5.2 争用时隙优化条件

图 6 和图 7 分别给出了 BFS 和 DFS 中 CRQ 和 DTQ 排队长度随时间变化趋势。

图 6 描述了 BFS 解决信道争用的排队长度变化。初始时刻 ($T=0$), 并发终端数 $K_0=1\ 000$, DTQ 长度 $L_T=0$, CRQ 长度 $L_C=1$ 。

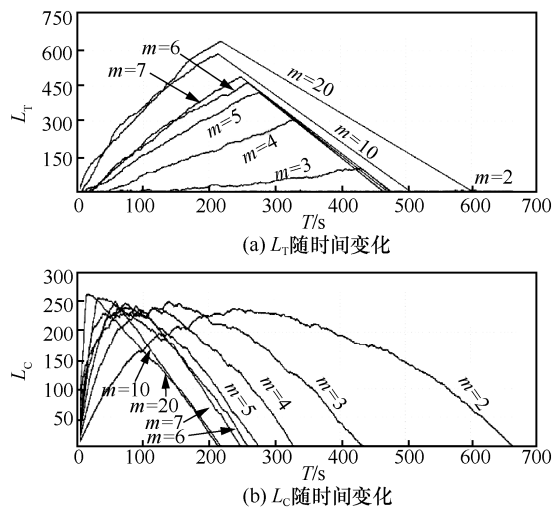


图 6 BFS 中 DTQ 和 CRQ 排队长度随时间变化趋势

从图 6(a)的 L_T 曲线可以看出, L_T 随时间增加先增大, 达到最大值后线性递减至 0。 $m=2$ 时, 210 s 左右 L_T 达到最大值, 650 s 左右 L_T 减小至 0。 $m=20$ 时, L_T 达到最大值和 0 值的时间分别约为 210 s 和 600 s, L_T 达到最大值时间对应于 $L_C=0$ 。这是因为, CRQ 内退避终端清零时, 后继没有入队而只有出队的 DTQ 只会随时间递减。

从图 6(b)的 L_C 曲线可以看出, 随着时间增加, L_C 先增大, 达到最大值后再逐渐减小到 0。 m 值越大, 最大值和 0 值出现时间越小。 $m=2$ 时, 250 s 左右 L_C 达到最大值, 650 s 左右 L_C 减小至 0。 $m=20$ 时,

L_C 达到最大值和 0 的时间分别约为 30 s 和 210 s。

从图 6 可见, 当 $m=4, 5, 6$ 时, $L_T=0$ 的时间最小, 即 K_0 个终端全部完成数据发送的时间最少。根据式(15), 吞吐量最大的理论条件是 $m_{BFS_MAX} \approx 6$, 这与仿真结果是吻合的。

图 7 描述了相同初始条件下 DFS 解决信道争用的排队长度变化, 其中 L_T 的变化趋势与图 6 相似, 但 L_C 的变化幅度比图 6 小一个数量级。因曲线重叠, 图 7(b)分别给出了 $m=2, 3, 20$ 的 L_C 时变曲线。同样的规律是, 当 $L_C=0$ 时, L_T 达到最大; 当 $L_T=0$ 时, K_0 个终端全部完成数据发送。

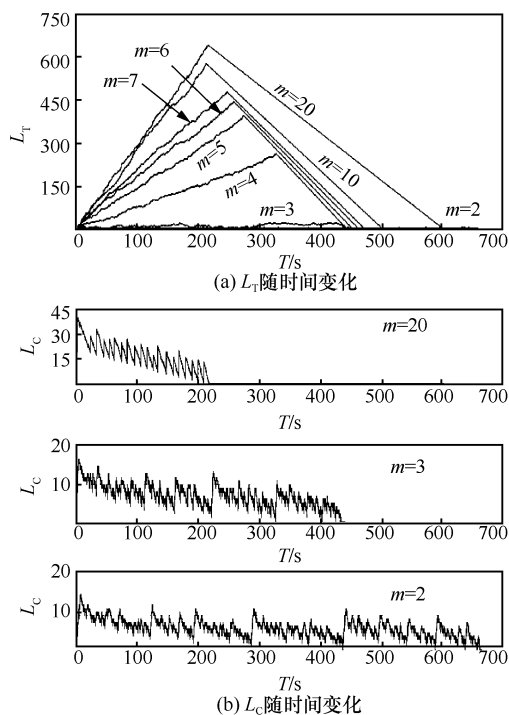


图 7 DFS 中 DTQ 和 CRQ 排队长度随时间变化趋势

从图 7 可以看出, $m=3$ 的 DTQ 排队长度 L_T 最先减少至 0, 这与式(17)的估算结果吻合。

文献[2]在忽略 CRQ 与 DTQ 的相互耦合的情况下, 为传统 DQ 给出的乐观估计是, 吞吐量最大条件是小时隙配置 $m=3$ 。依据仿真实验, 传统 DQ 最佳条件应为 $m=4$, 基于 DFS 的改进算法的最佳条件为 $m=3$ 。这是因为, DFS 能更快地搜索到退避树叶节点, 为冲突解决提供同时进行数据传输的机会, 减少 DS 空转概率。

5.3 吞吐性能对比

本节以小时隙最佳配置为条件, 对 DQ 吞吐性能随并发终端数的情况进行随机实验, 统计结果如图 8 所示。

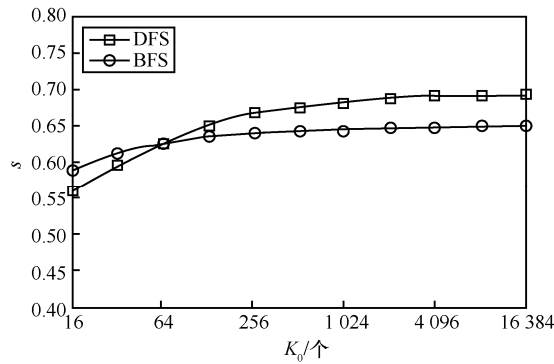


图 8 归一化吞吐量随并发终端数的变化

图 8 所示的仿真中，DFS 中设置 $m=3$ ，BFS 中设置 $m=4$ 。仿真总时长设为 8×10^5 s，针对 $K_0=16384$ ，得到 10 组 BCN 周期；针对 $K_0=16$ ，得到 9 000 组以上 BCN 周期。归一化吞吐量为 $s = \frac{K_0 T_{DS}}{T_{tot}}$ ，其中 T_{tot} 为所有 BCN 周期完成的仿真时间。

当 $K_0=16384$ 时，DFS 和 BFS 仿真中的 T_{tot} 平均值分别为 7 085.291 s 和 7 537 s。对应 ALOHA，按 $G = K_0(T_{tot}/T_{DS})^{-1}$ 计算归一化负载分别为 0.96 和 0.69。依文献[10]，可得归一化吞吐量分别为 0.141 和 0.173，均小于理论上的最大值 0.184。

从图 8 可见，DQ 的归一化吞吐量大于 0.55，重负载时超过 0.65，反映出远好于 ALOHA 的稳定性。当并发终端数 $K_0 > 64$ 时，相比于 DQ 原有算法，改进算法的吞吐量有最大 6% 的提升，接近信道物理容量的 70%。表 1 给出了 BFS 和 DFS 算法下所有终端完成数据发送的总时长及加速因子计算结果。

表 1 BFS 和 DFS 算法下所有终端完成数据发送的总时长及加速因子计算结果

$K_0/\text{个}$	T_{BFS}/s	T_{DFS}/s	$f = \frac{T_{BFS}}{T_{DFS}}$
16	8.15	8.59	0.95
32	15.60	16.10	0.97
64	30.61	30.57	1.00
128	60.30	58.84	1.03
256	119.85	114.88	1.04
512	238.37	226.04	1.06
1 024	475.13	448.03	1.06
2 048	946.53	890.68	1.06
4 096	1 891.09	1 775.33	1.07
8 192	3 772.36	3 545.94	1.06
16 384	7 537.00	7 085.29	1.06

6 结束语

分布式排队综合了 ALOHA 随机多址和时分复用的技术手段，通过树型退避方法压缩多终端接入的连续冲突概率，使共享信道的有效利用率得到大幅提高。共享信道的争用时隙配置与优化是 DQ 技术的应用关键。本文分析了 DQ 工作机制，针对争用请求队列采用的宽度优先遍历树，设计了基于深度优先的改进算法，并给出了理论和仿真的分析结果，证明了改进算法可以进一步提高吞吐性能。此外，本文采用了随机树分析法估算出 DQ 争用时隙数的最佳配置条件，并通过仿真实验进行了验证。本文预设 DQ 系统具有理想的定时同步性能，这在实际应用中，特别是应用于因休眠节能而不能持久同步的物联网时，是值得进一步深入探索的问题。

参考文献:

- [1] XU W. Distributed queueing random access protocols for a broadcast channel[D]. Chicago: Illinois Institute of Technology, 1990.
- [2] XU W, CAMPBELL G. A distributed queueing random access protocol for a broadcast channel[J]. SIGCOMM Communications, 1993, 23(4): 270-278.
- [3] LAYA A, ALONSO L, ALONSO-ZARATE J. Contention resolution queues for massive machine type communications in LTE[C]//2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications. Piscataway: IEEE Press, 2015: 2314-2318.
- [4] XING S, WEN X, LU Z, et al. A novel distributed queueing-based random access protocol for narrowband-IoT[C]//2019 IEEE International Conference on Communications. Piscataway: IEEE Press, 2019:1-7.
- [5] ALONSO L, FERRUS R, AGUSTI R. WLAN throughput improvement via distributed queueing MAC[J]. IEEE Communications Letters, 2005, 9(4): 310-312.
- [6] ALONSO-ZARATE J, KARTSAKLI E, ALONSO L, et al. Performance analysis of a cluster-based MAC protocol for wireless ad hoc networks[J]. EURASIP Journal on Wireless Communications and Networking, 2010: doi.org/10.1155/2010/25619.
- [7] ZHANG K, MARCHIORI A. Demo abstract: PlanIt and DQ-N for low-power wide-area networks[C]//2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation. Piscataway: IEEE Press, 2017: 291-292.
- [8] WU W, LI Y, ZHANG Y, et al. Distributed queueing-based random access protocol for LoRa networks[J]. IEEE Internet of Things Journal,

2020, 7(1): 763-772.

[9] LAYA A, KALALAS C, VAZQUEZ-GALLEGO F, et al. Goodbye, ALOHA![J]. IEEE Access, 2016, 4: 2029-2044.

[10] 徐秉铮. 随机多址通信技术[J]. 通信学报, 1982, 3(3): 78-84.

XU B Z. Random access techniques[J]. Journal on Communications, 1982, 3(3): 78-84.



吴炜（1993- ），女，江苏常州人，南京邮电大学博士生，主要研究方向为宽带无线通信技术。

[作者简介]



王文骥（1966- ），男，江苏南京人，博士，南京邮电大学教授、博士生导师，主要研究方向为通信网协议与无线通信技术。



柏琛（1995- ），男，江苏盐城人，南京邮电大学硕士生，主要研究方向为宽带无线通信技术。



张延贺（1994- ），男，山东枣庄人，南京邮电大学硕士生，主要研究方向为宽带无线通信技术。



王斌（1970- ），男，江西九江人，博士，南京邮电大学副教授、硕士生导师，主要研究方向为移动通信和下一代网络。