

PRMA 中的允许概率动态决定算法

An Algorithm for Dynamically Determining the Access Probability

陈 晖 徐福培

(南京大学计算机科学与技术系 南京210093)

Abstract Packet Reservation technology is frequently used in Media Access Control protocols. To increase the utilization of bandwidth, D. J. Goodman et al. presented the p-persistent algorithm, in which an access probability p is used to reduce collision. This idea is applied in many protocols. In this paper, the distribution of the number of accessing terminals is forecasted to determine the access probability dynamically. This method helps the protocol adaptive to various system environments automatically.

Keywords MAC, ALOHA, Packet reservation

1. 背景

预约(Reservation)技术是在1973年由 L. Roberts 首次提出,并被美国国防先进项目研究署(DARPA)首先用于 SATNET 中。1989年, D. J. Goodman 和 R. A. Valenzuela^[1]等人提出了基于分组预约(Packet reservation)的 MAC 接入协议 PRMA-分组预约多址接入。此后从20世纪90年代中期开始到现在,出现了很多基于分组预约的 MAC 层接入协议,如 integrated PRMA^[2]、centralized PRMA^[3]、DRMA^[4]、APRMA^[5]、PRMA/DA^[6]、DRISMA^[7]等,应用领域覆盖了卫星通信系统、无线通信系统以及 HFC 网络等各方面。PRMA 工作于星型网络下的时分复用(TDMA)信道,同时支持交谈终端的语音数据传输和一般信息终端的数据包传输。信道被同步为时隙(slot),并在终端组织为帧(frame),终端以 Slotted-ALOHA 的方式竞争时隙,不同的是当竞争失败需要再次竞争时,终端改以允许概率 p 发送请求。竞争成功后每帧中的此时隙被该终端预约,独占使用。L. Roberts 最初提出的预约方案被称为 Explicit Reservation(Reservation ALOHA),是将 Slotted-ALOHA 中每一帧最后的一个时隙分割为更小的微时隙(mini-slot)用于竞争,终端采用和 Slotted-ALOHA 中一样的方法竞争微时隙传送自己的请求(request)给基站。基站在接到请求之后在其他的时隙中分配尚未被使用的给终端使用。X. Qiu 和 V. O. K. Li^[4]综合了这两种协议的优点,提出了将未被预约的时隙全部分割为微时隙供申请使用等。1998年 H. C. B. Chan^[7]等又提出了动态允许概率。本文介绍的算法就是在这些协议的基础上,更好地动态决定允许概率。

2. 模型和协议

本文通过建立基站(Center Base)与语音终端(Voice Terminal)之间的通信模型来说明算法。为了简化起见,假定所有的 N 个终端完全相同,每个终端都只有两种状态:空闲状态(Idle)以及活动状态(Active),两种状态之间的转变概率满足指数分布。基站和终端之间是以星形拓扑网络相连,所有的通信都发生在基站和终端之间。从基站到终端的通信(下传)信道是高频的,由基站自身根据需要分配。终端到基站的通信(上传)信道以类似 Slotted-ALOHA 的方式组织。每 F 个时隙组织为一帧并由基站同步。每个时隙又可以根据需要被分为 k 个较小的时隙。终端以基站给定的某一允许概率 p , 占用微时隙发送自己的请求包。此时的终端标记为竞争状态(contending)。基站接收到请求包后,根据情况分配一个空闲

(free)时隙给这个终端,同时标记该 slot 为已预约(reserved),在后续帧中都会保留给该终端使用,直到终端传完所有的数据后通知基站,这时的终端就标记为预约状态(reservation)。在发送请求包期间若两个终端在同一微时隙中发出其请求包,则会发生冲突(collision),基站将无法收到任何一个请求。发生冲突的终端在接受到基站传来的竞争结果为冲突时,在下一微时隙继续以概率 p 发送请求包。

系统参数除了终端数 N 、每帧时隙数 F 、每时隙所分的微时隙数 k ,还有终端的平均空闲时间 $1/\alpha$,平均活动时间 $1/\beta$ 。

作为一个理想模型,假设通信没有任何传输延迟,基站可以及时地返回控制信息(允许概率 p 、竞争结果以及预约情况等)给终端。每一帧开头,基站还会将所有时隙的预约状态下传给每一终端,没有预约的时隙将全部被分割为微时隙用于终端竞争发送请求包。所有终端只在每帧的开始时刻发生状态变化(由空闲变为活动或由活动变为空闲)。模型中使用的变量和符号定义如下:

·在第 n 帧开始时(终端的状态已经发生过变化),处于已预约状态的终端数目为 $r_{n,0}$,处于竞争状态的终端数目为 $c_{n,0}$;在第 n 帧的第 $m+1$ 个微时隙开始时,处于 reserved 状态的终端数为 $r_{n,m}$,处于竞争状态的终端数为 $c_{n,m}$ 。从前面可以看出,第 n 个帧的微时隙的个数 $m_n = \begin{cases} (F-r_{n,0}) \times k & F-r_{n,0} > 0 \\ 0 & F-r_{n,0} \leq 0 \end{cases}$

·在第 n 帧的第 $m+1$ 个微时隙开始时,有 x 终端参加竞争的概率表示为 $C_{n,m}(x)$;在第 n 帧的开始时,有 y 终端参加竞争的概率表示为 $C_{n,0}(y)$ 。

· $p_{n,0}$ 表示在第 n 帧开头的允许概率; $p_{n,m}$ 表示在第 n 帧的第 $m+1$ 个微时隙开头的允许概率。

·终端从空闲态到活动态的概率为 $P_{ai} = 1 - e^{-\beta f}$,终端从活动态到空闲态的概率为 $P_{ia} = 1 - e^{-\alpha f}$ 。这是由系统参数 α 和 β 决定的。

·在第 n 帧结束的时候,还剩下 u_n 个处于竞争状态的终端。

显然在这样的模型假设下,基站的任务就是要寻找允许概率 p ,使得在每一微时隙构造过程中,以 p 为允许概率接入,成功的概率最大。即寻找使函数

$$S_{n,m} = \sum_{c_{n,m}=1}^{N-r_{n,m}} C_{n,m}(c_{n,m}) \times c_{n,m} \times p \times (1-p)^{c_{n,m}-1} \quad (1)$$

取最大值的 p 。整理得所求的 p 是方程

$$\sum_{c_{n,m}=1}^{N-r_{n,m}} C_{n,m}(c_{n,m}) \times c_{n,m} \times (1-p)^{c_{n,m}} \times (1-c_{n,m} \times p) = 0 \quad (2)$$

在 $(0, 1/(N-r_{n,m}))$ 上的解。

如果在第 n 个帧的第 m 个微时隙构造前, 基站知道此时参加竞争的终端的数目 $N_{c_{n,m}}$, 可得 $p = 1/N_{c_{n,m}}$, 即最佳的允许概率应该等于实际参与竞争的终端数目的倒数。

实际的情况中, 基站往往不可能知道某个时候参加竞争的终端的实际数目。H. Chan 和 V. Leung^[7] 提出将允许概率设为:

$$p = \sum_{i=0}^{N-N_r-1} \frac{1}{1+i} \binom{N-N_r-1}{i} P_{ia}' (1-P_{ia})^{N-N_r-1-i} \quad (3)$$

其中 N_r 是已经申请到 slot 处于预约状态的终端数目。对于一个处于竞争状态的终端来说, 如果另外还有 i 个终端同时处于竞争状态, 它应该取 $1/(i+1)$ 为允许概率。虽然基站不知道 i 的值, 但是可以假设在前一帧中, 所有参加竞争的终端全部都成功地发送了请求而成为预约状态, 即 $u_n = 0 \forall n$ 。这样 i 应该服从分布:

$$P(i) = \binom{N_r-N_r-1}{i} r^i (1-r)^{N_r-N_r-1-i} \quad (4)$$

可以看出式(3)所决定的 p 是所有可能的允许概率 p 的期望。

此允许概率和最优允许概率依然有较大的距离。因为 i 的分布不是真实值, 只是一个理想情况下的近似; 同时所有可能值的期望也远不等于方程(2)的解。

在知道分布的情况下, 可以用简单数值的方法去求近似解。实际上, 可以让 p 以一定的步长从 0 增加到 $1/N_r$, 同时计算公式(1)中相应的 $S_{n,m}$, 找出 $S_{n,m}$ 的最大值及其相应的 p 的值。解的质量和所取的步长有关。步长越小, 解就越精确。所以, 只要给定一个分布 $C_{n,m}(c_{n,m})$, 便可以得到 p 的一个近似解。但最主要的问题就是如何得到分布 $C_{n,m}(c_{n,m})$ 。

3. 分布更新和允许概率动态决定算法

我们知道, 基站能够知道每个微时隙竞争的结果。可以利用这些结果求出在每一个微时隙开始时, 处于竞争态的终端的概率分布, 从而更加精确地计算出允许概率 p 的最优值。

在第 n 帧的第 m 个微时隙开始前, 假设竞争态终端的数量满足分布 $C_{n,m-1}(c_{n,m-1})$, 而该微时隙申请成功的概率为 $S_{n,m}$, 结果为空的概率是 $I_{n,m}$, 冲突的概率为 $Co_{n,m}$ 。可以得到以下公式:

$$S_{n,m} = \sum_{c_{n,m-1}=1}^{N-r_{n,m-1}} C_{n,m-1}(c_{n,m-1}) \times c_{n,m-1} \times p_{n,m-1} \times (1-p_{n,m-1})^{c_{n,m-1}-1} \quad (5)$$

$$I_{n,m} = \sum_{c_{n,m-1}=1}^{N-r_{n,m-1}} C_{n,m-1}(c_{n,m-1}) \times (1-p_{n,m-1})^{c_{n,m-1}} \quad (6)$$

$$C_{n,m}(c_{n,m}) = \frac{C_{n,m-1}(c_{n,m-1}) \times (1 - (1-p_{n,m-1})^{c_{n,m-1}-c_{n,m-1}} \times p_{n,m-1} \times (1-p_{n,m-1})^{c_{n,m-1}-1})}{\sum_{c'_{n,m-1}=1}^{N-r_{n,m-1}} C_{n,m-1}(c'_{n,m-1}) \times (1 - (1-p_{n,m-1})^{c'_{n,m-1}-c'_{n,m-1}} \times p_{n,m-1} \times (1-p_{n,m-1})^{c'_{n,m-1}-1})} \quad (12)$$

通过上述公式, 我们可以随时改变竞争终端数目的概率分布。但是在每两个帧之间, 终端的状态会发生一些变化。假定 $c'_{n,m}$ 表示在第 n 个帧结束之时竞争态终端的数目, 其中有 i 个终端在 n 帧到 $n+1$ 帧之间变为空闲态。而 $c_{n+1,0}$ 表示在第 $n+1$ 帧开始时, 系统所含的竞争态终端的数量。易见, 在 n 帧到 $n+1$ 帧之间, 有 $c_{n+1,0} - c'_{n,m} + i$ 个终端从空闲态变换到竞争态。于是, 可以得到:

$$C_{n+1,0}(c_{n+1,0}) = \sum_{c'_{n,m}} \sum_i (C_{n,m}(c'_{n,m}) \times \binom{c'_{n,m}}{i} \times P_{ai}' \times (1-P_{ai})^{c'_{n,m}-i})$$

$$Co_{n,m} = \sum_{c_{n,m-1}=1}^{N-r_{n,m-1}} C_{n,m-1}(c_{n,m-1}) \times (1 - (1-p_{n,m-1})^{c_{n,m-1}-c_{n,m-1}} \times p_{n,m-1} \times (1-p_{n,m-1})^{c_{n,m-1}-1}) \quad (7)$$

如果第 n 帧的第 m 个微时隙开始时处于竞争状态的终端数量为 $c_{n,m-1}$, 每一个终端以 $p_{n,m-1}$ 的概率竞争微时隙, 可能结果有三种: 成功、空闲和冲突。如果结果为空闲, 利用 Bayesian 定理, 可以知道:

$$P(c_{n,m-1} | idle) \times P(idle) = P(idle | c_{n,m-1}) \times P(c_{n,m-1}) \quad (8)$$

其中: $P(c_{n,m-1} | idle)$ 表示在已知该微时隙的竞争结果为空闲的情况下, 起初有 $c_{n,m-1}$ 个竞争态的终端的概率; $P(idle)$ 指竞争结果为空闲的概率; $P(idle | c_{n,m-1})$ 表示起初有 $c_{n,m-1}$ 个竞争态终端的情况下, 最后结果为空闲的概率; $P(c_{n,m-1})$ 表示在第 n 帧第 m 个微时隙开始时, 有 $c_{n,m-1}$ 个终端处于竞争状态的概率。

$P(idle)$ 的值可以从式(6)得出, $P(c_{n,m-1})$ 的概率也可以从分布 $C_{n,m-1}(c_{n,m-1})$ 中得出, $P(idle | c_{n,m-1})$ 的概率可以求得为 $(1-p_{n,m-1})^{c_{n,m-1}}$ 。因此, 可以得出 $P(c_{n,m-1} | idle)$ 的值。实际上, 概率 $P(c_{n,m-1} | idle)$ 表示在已知竞争结果为空闲的情况下, 第 n 帧第 m 个微时隙开始时, 有 $c_{n,m-1}$ 个终端处于竞争状态的概率。由此可得:

$$\begin{aligned} C_{n,m}(c_{n,m-1}) &= P(c_{n,m-1} | idle) \\ &= \frac{P(idle | c_{n,m-1}) \times P(c_{n,m-1})}{P(idle)} \\ &= \frac{(1-p_{n,m-1})^{c_{n,m-1}} \times C_{n,m-1}(c_{n,m-1})}{\sum_{c'_{n,m-1}=1}^{N-r_{n,m-1}} C_{n,m-1}(c'_{n,m-1}) \times (1-p_{n,m-1})^{c'_{n,m-1}}} \quad (9) \end{aligned}$$

当结果为冲突或成功时类似。所以我们有:

1. 当竞争结果为成功时:

$$\begin{aligned} C_{n,m}(c_{n,m-1}-1) &= \frac{C_{n,m-1}(c_{n,m-1}) \times c_{n,m-1} \times p_{n,m-1} \times (1-p_{n,m-1})^{c_{n,m-1}-1}}{\sum_{c'_{n,m-1}=1}^{N-r_{n,m-1}} C_{n,m-1}(c'_{n,m-1}) \times c'_{n,m-1} \times p_{n,m-1} \times (1-p_{n,m-1})^{c'_{n,m-1}-1}} \quad (10) \end{aligned}$$

2. 当结果是空闲时:

$$C_{n,m}(c_{n,m-1}) = \frac{C_{n,m-1}(c_{n,m-1}) \times (1-p_{n,m-1})^{c_{n,m-1}}}{\sum_{c'_{n,m-1}=1}^{N-r_{n,m-1}} C_{n,m-1}(c'_{n,m-1}) \times (1-p_{n,m-1})^{c'_{n,m-1}}} \quad (11)$$

3. 当竞争结果为冲突时:

$$\begin{aligned} &\times \left(\binom{N-r_{n,m}-c'_{n,m}}{c_{n+1,0}-c'_{n,m}+i} \times P_{ia}'^{c_{n+1,0}-c'_{n,m}+i} \times (1-P_{ia})^{N-r_{n,m}-c'_{n,m}-c_{n+1,0}+i} \right) \\ &P_{ia}'^{N-r_{n,m}-c'_{n,m}-c_{n+1,0}-i} \quad (13) \end{aligned}$$

其中 $c'_{n,m}$ 和 i 的变化范围是: $0 \leq c'_{n,m} \leq N-r_{n,m}$, $0 \leq i \leq c'_{n,m}$, $0 \leq c_{n+1,0} - c'_{n,m} + i \leq N-r_{n,m} - c'_{n,m}$ 。

我们还知道, 第一帧开始前处于竞争状态的终端数量的概率分布如下:

$$C_{1,0}(c_{1,0}) = \binom{N}{c_{1,0}} \times P_{ia}'^{c_{1,0}} \times (1-P_{ia})^{N-c_{1,0}} \quad (14)$$

这样, 就可以利用以上结论动态更新竞争态终端数的概率分布, 从而动态决定允许概率 p 。

4. 实验和结果

使用动态允许概率决定算法,每个微时隙都可以更新允许概率 p 的大小。但是该模型的模拟实验耗费的时间很大,不容易得到包丢失率、带宽利用率等结果。我们采用另一种方法。在第2节中得到结论,最优的允许概率应该等于实际竞争状态的终端数目的倒数。所以将算法给出的允许概率和此最优允许概率相比较。设系统中 $F=120$, $N=150$, $k=6$, $1/\alpha=0.015$, $1/\beta=0.01$ 。此系统模拟了8M的信道。终端状态的变化频率是正常的150倍(这是为了模拟大量终端的效果)。结果可见图1。X轴坐标表示微时隙序列,Y轴表示允许概率。图中的实线表示动态决定算法计算的允许概率,虚线表示最佳允许概率。可以看出,利用此模型得到的允许概率和最佳允许概率极为接近。并且在一帧中,这种差距会越来越小。

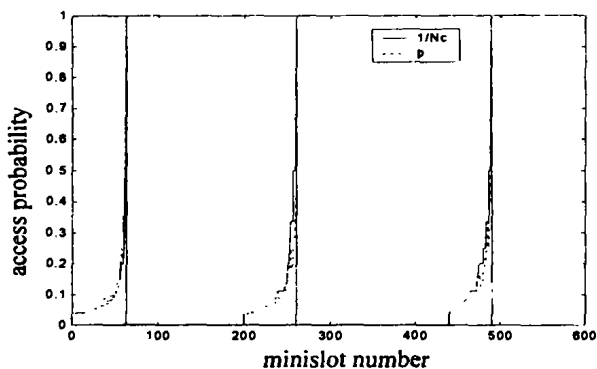


图1 正常情况下的动态允许概率和最优允许概率

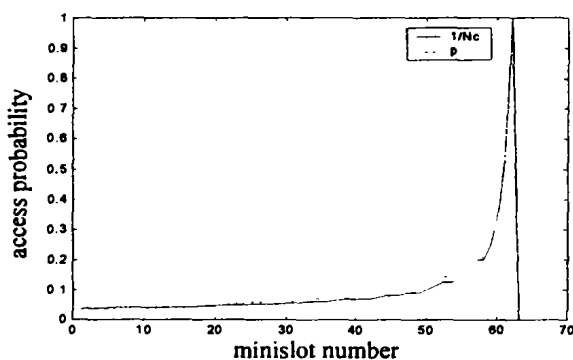


图2 错误分布下的动态允许概率和最优允许概率

同时我们发现,即使某帧开始时刻的竞争态终端数目的分布是错误的,在经过部分微时隙后,计算的允许概率也会同样接近最优允许概率。若在帧开始时,终端数目的分布为平均分布,则结果可见图2。

结论 本文提出的允许概率动态决定算法,进一步优化了 p -persistent 算法。为了实现该算法,我们还建立了一个简单模型,并给出了该模型下的协议。该算法有着广泛的应用,很多使用了分组预约技术和 p -persistent 算法的协议中都可以利用这种算法。虽然在我们的模型中只有语音传输,但在实际应用中往往采用一些其他技术集成数据或其他多媒体数据的传输。通过实验得出,此算法能使得系统允许概率接近最优允许概率。因为在最优允许概率下,每个微时隙竞争成功的极限值为 $1/e$,所以算法能够很好地适应各种各样的系统参数。

参考文献

- Goodman D J, et al. Packet reservation multiple access for local wireless communications. *IEEE Trans. on Commun.*, 1989, 37: 885~890
- Wong W, Goodman D J. Integrated data and speech transmission using packet reservation multiple access. In: *Proc. ICC'93, Geneva, Switzerland*, 1993. 172~176
- Bianchi G, et al. C-PRMA: the centralized packet reservation multiple access for local wireless communications. In: *Proc. Globecom'94, San Francisco, USA, Nov. 1994*. 1340~1346
- Qiu X, Li V O K. Dynamic reservation multiple access (DRMA): A new multiple access scheme for personal communication system. *ACM/Baltzer Wireless Networks*, 1996, 2: 117~128
- Khan F, Zeghlache D. Analysis of aggressive reservation multiple access scheme for wireless PCS. In: *Proc. ICC'96, Dallas, TX, June 1996*. 1750~1755
- Kim J G, Widjaja I. PRMA/DA: A new media access control protocol for wireless ATM. In: *Proc. ICC'96, Dallas, TX, June 1996*. 240~244
- Chan H C B, Leung V C M. Dynamic reservation integrated services multiple access for wireless ATM. In: *Proc. IEEE ICUPC'98, Florence, Italy, Oct. 1998*. 841~846

(上接第134页)

- Maffei S. A Fault-Tolerant CORBA Name Server. In: *Proc. of the 15th IEEE Symposium on Reliable Distributed Systems*. IEEE Press, Niagara-on-the-lake, Canada, Oct. 1996
- Birman K P, Joseph T A. Exploiting virtual synchrony in distributed systems. In: *Proc. of the ACM Symposium on Operating System Principles*, 1987. 123~138
- van steen M, Hauck F J, Homburg P, Tanenbaum A S. Locating Objects in Wide-Area Systems. *IEEE Communications Magazine*, Jan. 1998. 104~109
- Birman K, Joseph T. Reliable Communication in the presence of failures. *ACM Trans on Computer Systems*, 1987. 47~76

- 史殿习, 吴泉源, 王怀明, 邹鹏. 组通信中虚拟同步协议的研究与设计. *计算机研究与发展*, 2000, 37(10): 1192~1196
- Gamma E, Hekm R, Johnson R, Vlissides J. *Design patterns Elements of Resuable Object-Oriented Software*. Addison Welse, 1995
- Programming with Webogic JNDI. <http://edocs.bea.com/wls/docs60/jndi/jndi.html>
- OrbixNames Programmer's and Administrator's Guide. <http://www.iona.com>
- Baratloo A, et al. Filterfresh: Hot Replication of Java RMI Server Objects. In: *4th USENIX Conf. on Object-Oriented Technologies and Systems (COOTS'98)*