

# 水下声音网络中基于簇的按需时间共享 MAC 协议

刘康明, 魏 锐

(黄淮学院 信息工程学院, 河南 驻马店 463000)

**摘要:** 水下声音网络中基于调制解调器的商用系统具有传输率低、前同步码长两大特征, 严重降低了媒介访问控制协议 (MAC) 的性能; 针对上述问题, 分析了调制解调器特征对当前水下声音网络两种主流 MAC 协议 (随机访问 MAC, 握手 MAC) 的影响, 提出了一种基于簇的按需时间共享 MAC 协议 (COD-TS); 该协议采用簇技术实现了局部调度和按需响应特性; 文章首先给出单跳网络的 COD-TS 协议的含义, 提出相应的吞吐量模型; 然后, 实现方案多跳场景的扩展; 仿真结果表明, 相比于已有的 Slotted ALOHA 和 RTS/CTS 等协议而言, 本文方案的吞吐量更优。

**关键词:** 水下声音网络; MAC 协议; 调整解调器; 簇; 按需响应; 吞吐量

## On Demand Time Sharing MAC Protocol Based on Cluster in Underwater Acoustic Networks

Liu Kangming, Wei Rui

(College of Information Engineering, Huang Huai University, Zhumadian, 463000, China)

**Abstract:** Commercial systems based on modem in underwater acoustic networks have two critical characteristics, low transmission rates and long preambles. It drastically degrades the performance of existing MAC protocols in the real world. In order to resolve this problem, this paper analyzed the impact of the modem characteristics on the random access-based MAC and handshake-based MAC, which are two major types of MAC protocols for underwater acoustic networks. An On-Demand Time Sharing MAC (COD-TS) based on cluster was introduced and analyzed, which adopted cluster technique to enable local schedule and on-demand response feature. COD-TS protocol for one hop network was introduced at first, the corresponding throughput model was proposed. After that, the scheme was extended to multi-hop scenarios. Simulation results show that the throughput of the proposed scheme is better compared with the existing Slotted ALOHA protocol and the RTS/CTS protocol.

**Keywords:** medium access control protocol; modem; cluster; on demand response; throughput

## 0 引言

水下声音网络 (UAN) 是部署于水下环境的一组 Ad Hoc 网络, 节点通过声波进行通信。与使用 RF 信号的陆地无线网络相比, UAN 的可用带宽较低, 传播延时较大, 动态信道较高。这些特征给网络协议尤其是水下声音网络的 MAC 协议设计带来很大困难<sup>[1-3]</sup>。

为此, 文献 [4] 提出了一种基于运动预测的水下传感器网络 MAC 协议。该文构建了水下节点的运动模型, 基于 AR 运动预测模型有效的减小了水下节点的时空不确定性对于 MAC 协议的影响, 提高了发送信息在预约时隙到达的概率。文献 [5] 提出了一个利用改进时分多址技术的媒体访问控制协议 (W-MAC)。W-MAC 利用“延缓时间”作为节点的实际数据传送时间, 采用具有睡眠策略的监测时间来避免数据碰撞, 使 W-MAC 协议在适应复杂的水下信号传送环境的同时, 把能量开销限制在一个合理的水平。

Vieira 等人对水下网络环境的长延时特征进行了建模, 提出了 Slotted-ALOHA 协议<sup>[6]</sup>, 通过引入保护时间, 提升了 Slotted-ALOHA 的吞吐量。此外, 还设计了两种 ALO-

HA 变种版本, 性能得到进一步提升。因为基于随机访问的方法无法有效避免冲突, 所以部分文献研究了竞争条件下基于握手的 MAC 协议<sup>[7-8]</sup>。虽然这些协议的理论性能较好, 但是离实际应用还有很大差距。通过一系列实地测试, 发现声音调制解调器的现实特征 (可用传输率较低, 前同步码较长) 严重降低了现有 MAC 协议的性能。虽然文献 [9] 给出了兼顾这两个因素的仿真结果, 但是既没有理论分析这些因素的影响, 也没有提供专门的解决方案。这两个特征带来的困难要求我们设计新的 MAC 协议, 以提高协议的实际性能。鉴于此, 本文提出一种基于簇的按需时间共享 MAC 协议 (COD-TS) 以及相应的吞吐量模型。最后通过仿真实验验证了本文协议的有效性。

## 1 问题分析与建模

人们已经利用近期开发的水下声音网络测试床 Aqua-TUNE 全面实验评估了各种 MAC 协议的性能<sup>[9-10]</sup>。从这些测试中可以发现, 调制解调器的部分特征严重影响了当前 MAC 协议的应用。例如, 最大传输范围无法提前确定, 且主要取决于部署环境。然而, 许多当前协议需要最大传输范围。另外, 声音传输率较低, 因此均存在前同步码较长的问题。

许多当前研究认为调制解调器的传输率约为 10 kbps 甚至更高<sup>[7-8]</sup>。然而, 可用的传输率往往远低于理论值, 对浅水环境 (即水平信道) 下更是如此。例如, 虽然 Benthos 调制解调器的传输率声称达到 15 kbps, 但是为了保证水平信道的报文

收稿日期: 2014-07-05; 修回日期: 2014-08-18。

基金项目: 国家自然科学基金 (61300218/F020806)。

作者简介: 刘康明 (1979-), 男, 河南驻马店人, 硕士, 讲师, 主要从事水下声音网络、物联网方向的研究。

差错率较低, 最大可用传输率远低于该值, 如图 1 所示。鉴于篇幅有限, 本文重点讨论给水下声音网络的 MAC 设计带来严重困难的两大特征: 前同步码较长, 传输率较低。

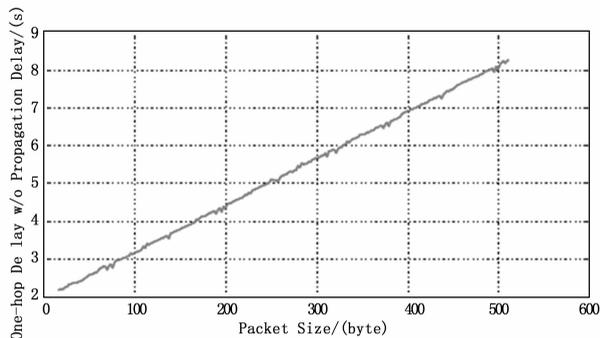


图 1 Benthos 调制解调器测得的单跳延时

一般而言, 在无线网络中往往使用小尺寸控制报文来为大尺寸数据报文预留信道。这一机制有助于缓解冲突, 因为小尺寸控制报文的冲突概率很低, 而且这也是基于握手的 MAC 协议的原则和假设。然而该假设对实践中的水下声音系统不再成立, 如图 2 所示。当前同步码较长且传输率较低时, 冲突概率迅速上升, 说明基于随机访问的 MAC 协议与实际系统存在严重冲突 (其中,  $T_{pre}$  是前同步码导致的延时,  $B$  是调制解调器有效传输率)。尤其地, 从图 2 可以看出, 即使是 20 bytes 的报文发生冲突的概率也达 0.68, 远大于传统配置时的概率。因此, 低传输率和长前同步码问题成为影响基于随机访问协议和基于握手 MAC 协议性能的关键因素。为了提高系统性能, 本文将针对水下声音系统的 MAC 协议进行研究。为了便于阐述, 首先介绍本文采用的网络模型:

- 1) 每个发送方生成的流量速率服从参数  $\lambda$  相同的泊松分布, 且所有数据报文的尺寸为  $L_D$ 。
- 2) 所有模型均在相同的网络场景下建立: 完全连接的单跳网络, 带有  $N$  个节点, 包括  $n (n \leq N)$  个发送节点。
- 3) 与文献 [8] 相同, 所有节点有相同的传输率  $B$  和最大传输范围  $R$ 。于是, 最大传输延时  $T_{mp}$  为  $R/v_p$ , 其中  $v_p$  表示传输速度。

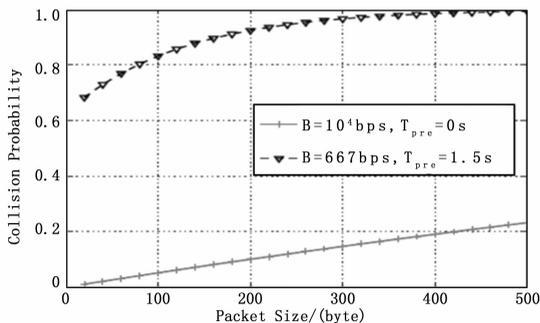


图 2 水下声音系统中报文尺寸变化时的冲突概率

## 2 基于时间共享的 MAC 协议: COD-TS

根据文献 [11] 可知, 基于时间共享的 MAC 协议不管

报文传输时间如何均可以避免报文冲突, 因此可以解决前同步码较长和传输率较低带来的挑战。为了研究基于时间共享 MAC 的性能, 文中提出了一种基于簇立即响应式时间共享 MAC 协议 (COD-TS)。该协议采用簇技术实现了局部调度和立即响应特性。文中首先引入单跳网络的 COD-TS 协议, 并以此为基础展开相关分析。然后, 将该方案拓展至多跳场景。

### 2.1 单跳网络 COD-TS 协议

在 COD-TS 中, 首先通过运行文献 [12] 中的分布式簇算法将节点分为多个簇。如图 3 所示, 簇成员  $M_1$  和  $M_2$  与簇头  $H_A$  单跳相邻, 簇头  $H_A$  负责为所有簇成员调度数据发送时间。簇的一轮通信定义为簇头两个连续调度规则更新期间交换数据报文或控制报文序列。每个簇重复每轮通信的操作。在图 3 中, 时间  $t_1$  和  $t_2$  间的时间就表示  $H_A$  簇的一轮通信。在每轮通信中, 簇头根据成员请求对成员的时隙发送进行调度。也就是说, 保证簇间的公平性, 除非调度后的总体报文传输时间超过预先定义的最大回合时间  $D_{max}$ , 否则通信回合的时间不固定。

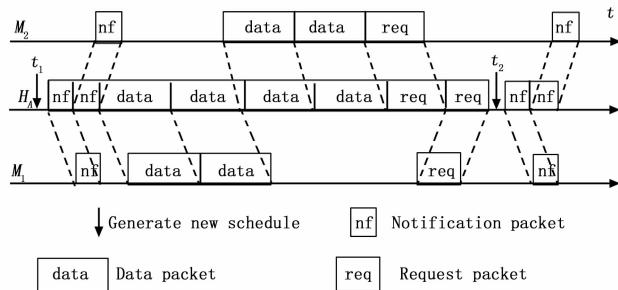


图 3 单跳网络的通信回合

每轮通信开始时生成新的调度规则, 然后向簇成员发送通知报文。这两种事件均由先前的调度更新进行安排。同时, 当前的调度更新内容还要安排本轮内数据报文和请求报文的时隙, 下一个调度规则的生成以及在下一轮如何向成员发送通知报文。例如, 在图 3 中, 当  $H_A$  在时间  $t_1$  更新调度规则时, 使用算法 1 为所有报文分配时隙, 然后到达下一调度更新时间  $t_2$ 。尤其地, 原先调度规则也在本轮安排向成员发送通知报文。收到通知报文后, 簇成员便知道何时发送请求报文和数据报文。

当簇成员  $i$  发送请求报文的调度时间到达时, 向簇头发送一个请求报文  $P_{req} \{R_i, P_i\}$ , 其中  $R_i$  表示请求报文将被发送的请求时间集合,  $P_i$  表示节点  $i$  及其相邻节点间的传播延时。可以根据相邻节点窃听报文的时间戳来评估传播时延。尤其地, 如果簇成员没有任何数据报文需要发送, 则直接跳过该请求时隙。从成员收集到请求后, 簇头再安排好调度更新时间, 利用算法 1 生成新的调度规则。

算法 1: 生成新的调度

输入:  $n, D_{max}, T_{mp}, T_{tx,D}, R_1, R_2, \dots, R_n$

输出:  $T_R, T_N, T_u, S_1, S_2, \dots, S_n$

- 1)  $T_{req\_total} \leftarrow 0, D_{mdata} \leftarrow getDSDur \{ \}$
- 2) 初始化可分配请求集合  $A_1, A_2, \dots, A_n$
- 3) while  $R_1, R_2, \dots, R_n$  不全为空 do

- 4) for  $i = 1 \rightarrow n$  do
- 5)  $T_r \leftarrow R_i$  中首个请求时隙
- 6) if  $T_r + T_{req, total} < D_{mdata}$  then
- 7) 把  $T_r$  添加到  $T_{req, total}$ , 将  $T_r$  插入  $A_i$
- 8) else
- 9) 退出 while 循环
- 10) end if
- 11) end for
- 12) end while
- 13) for  $i = 1 \rightarrow n$  do
- 14) 为  $A_i$  中的时隙寻找首批可用时隙, 将结果存储在  $S_i$  中
- 15) end for
- 16) 为来自成员的请求寻找首批可用时隙, 下一调度更新时间, 及成员通知, 并分别存储于  $T_R, T_N, T_u$  中

在算法 1 中,  $T_{ix, D}$  是尺寸为  $L_D$  的数据报文的传输时间。在第 1 行,  $getDSDur \downarrow \downarrow$  返回数据发送的最大时间。输出的  $T_R, T_N$  分别表示用于簇成员发送请求和接收通知的时隙集合。 $T_u$  表示簇头将要生成下一个新的调度规则的时间。 $S_1, S_2, \dots, S_n$  表示相应的簇成员应该发送数据报文的时隙集合。尤其地, 如果发现一个时隙可用于将报文从节点  $i$  发送到节点  $j$ , 则节点  $i$  的簇头将从节点  $i$  的角度, 使用文献 [13] 中的算法, 在时间线上映射并标记节点  $i$  的发送/接收时隙, 节点  $i$  相邻节点的接收时隙, 以及节点  $j$  的发送/接收/干扰时隙。然后, 通过定位没有与任何被标记时隙相重叠的首个时隙, 就可以找到可用时隙。

获得了新的调度规则后, 簇头可以为每个簇成员生成一个通知报文  $P_{nf} \{S_i, T_{R,i}\} (1 \leq i \leq N)$ , 并在时间  $T_{N,i}$  将其发出, 其中  $T_{R,i} \in T_R$  和  $T_{N,i} \in T_N$  分别表示成员  $i$  把请求报文发送给簇头的时间, 以及簇头将通知报文发送给成员  $i$  的时间。然后, 成员  $i$  可以在安排好的时隙内发送数据报文和请求报文。为了提升信道利用率, 每个簇成员可以在请求报文中捎带确认 ACK。然后, 簇头在通知报文中会包含相应发送方的 ACK。

### 2.2 节点吞吐量模型

因为存在时空差异, 所以要为 COD-TS 建立一种准确的节点吞吐量模型, 难度很大。出于简便考虑, 假设直到当前发送方完成发送任务  $T_{mp}$  时间后节点才可以发送数据。这一假设可以确定节点吞吐量的下界。

设  $D_{p, req}, D_{p, nf}, D_{p, ds}$  分别表示请求时隙、通知时隙和数据传输时隙的总时间。根据第 3.1 节内容,  $D_{p, req}$  和  $D_{p, nf}$  为:

$$\begin{aligned} D_{p, req} &= (N-1)(T_{ix, req} + T_{mp} + T_{guard}) \\ D_{p, nf} &= (N-1)(T_{ix, nf} + T_{mp} + T_{guard}) \end{aligned} \quad (1)$$

其中:  $T_{ix, req}$  和  $T_{ix, nf}$  分别表示请求报文和通知报文的传输时间。出于简便考虑, 将其称为总控制报文, 总控制报文的相应传输时间为:

$$D_{p, ctrl} = D_{p, req} + D_{p, nf} \quad (2)$$

如果  $D_{p, ds}$  也已知, 则一轮时间  $D_R$  为:

$$D_R = D_{p, ctrl} + D_{p, ds} \quad (3)$$

因为簇头只把数据发送时隙分配给负责发送数据的簇成员, 所以将会有  $n$  个节点参与数据发送阶段。请注意, 每个节点努力地为所有正在排队的数据报文申请发送时间, 以便它们能够在该轮内发送出所有报文, 因此只要生成的报文没有超出

信道容量,  $D_R$  必然存在稳态。在稳态下, 每轮中的节点可以发送上一轮生成的所有报文, 即对  $D_R$  应该有:

$$D_{p, ds} = n\lambda D_R T_{ix, D} + nT_{mp}, \text{ 如果 } n\lambda T_{ix, D} < 1 \quad (4)$$

然后, 合并式 (3) 和式 (4) 可以给出  $D_R$  的闭合形式:

$$D_R = \frac{D_{p, ctrl} + nT_{mp}}{1 - n\lambda T_{ix, D}} \quad (5)$$

然而, 当  $\lambda$  较高时, 信道可能饱和。此时, 簇每轮时间将受到上界  $D_{max}$  的约束。如果考虑这一约束, 则有:

$$D_R = \min \left\{ \frac{D_{p, ctrl} + nT_{mp}}{1 - n\lambda T_{ix, D}}, D_{max} \right\} \quad (6)$$

当  $n\lambda T_{ix, D} \rightarrow 1$  时  $D_{max}$  肯定变小, 因为式 (5) 的结果此时趋近于无限大。于是, 结合式 (3),  $D_{p, ds}$  为:

$$D_{p, ds} = \min \left\{ \frac{D_{p, ctrl} + nT_{mp}}{1 - n\lambda T_{ix, D}}, D_{max} \right\} - D_{p, ctrl} \quad (7)$$

考虑到本节开始时所做的假设, 为每个发送方分配的发送时隙包括  $T_{mp}$ 。除去这部分外, 剩余时间为  $D_{mdata}$ 。因此,  $D_{p, ds}$  期间发送的数据报文总数  $N_D$  为:

$$N_D = \frac{D_{p, ds} - nT_{mp}}{T_{ix, D}} \quad (8)$$

相应地,  $D_{p, ds}$  期间接收到的数据量为  $N_D L_D$ , 这也是  $D_R$  期间传递的数据量。因此, COD-TS 的节点吞吐量为:

$$\Lambda = \frac{N_D L_D}{N D_R} = \left( 1 - \frac{nT_{mp} + D_{p, ctrl}}{D_R} \right) \frac{L_D}{N(L_D/B + T_{pre})} \quad (9)$$

根据式 (9), 当  $n, N, L_D$  固定时,  $\Lambda$  由  $D_R$  确定。 $D_R$  越大,  $\Lambda$  越接近于它的上界  $\frac{L_D}{N(L_D/B + T_{pre})}$ , 而该上界也接近于最大节点吞吐量  $B$ 。因此, COD-TS 的吞吐量较高。

### 2.3 将 COD-TS 拓展至多跳网络

为了分析时间共享 MAC 的性能, 已经给出了 COD-TS 单个簇的工作原理。在本小节, 将 COD-TS 拓展至多跳场景。实际上, 如果调度规则可以避免文献 [13] 描述的 4 种类型的冲突, 便可以保证簇成员的报文可以成功到达预定接收方。请注意, ad hoc 网络的所有冲突要么由单跳相邻节点造成, 要么由隐藏的终端节点造成。因此, 为了消除冲突现象, 簇头在生成调度规则时需要知道双跳范围内成员相邻节点的最新调度规则。簇头可以利用簇结构, 并与相邻簇头 (NCH) 交换最新调度规则来获得这一信息。具体来说, 如果节点与另一节点的距离在双跳范围内, 则它们的簇头互为 NCH。例如, 在图 4 (a) 中,  $H_B$  是  $H_A$  的 NCH, 但  $H_C$  不是。为了支持 NCH 交换它们的调度规则, 簇头可以增加它们的传输功率。这种做法是可行的, 因为当前大多数调制解调器支持传输功率调节。

本文利用 CDMA 方法来发送调度交换报文。图 4 (b) 给出了 NCH 间的调度交换情况。出于简便考虑, 将用于交换 NCH 间调度规则的报文称为调度交换报文。为了避免调度更新冲突, 每个簇头在调度更新冲突期间 (SCUI) 不得生成或通知其新的调度规则。具体来说, 簇头的 SCUI 间隔表示其 NCH 生成新的调度的时间点, 与相应的调度交换报文到达该节点的时间点之间的间隔。这一规则可以保证每个簇的通信回合与其他无关时不会发生调度更新冲突。获得了一个调度更新报文后, 簇头将在自己的时间线上映射并标记所有的时隙, 据第 3.1 节描述, 该时间线还可用于寻找新的发送时隙。当生成新的调度时, 簇头便获取了所有成员双跳相邻节

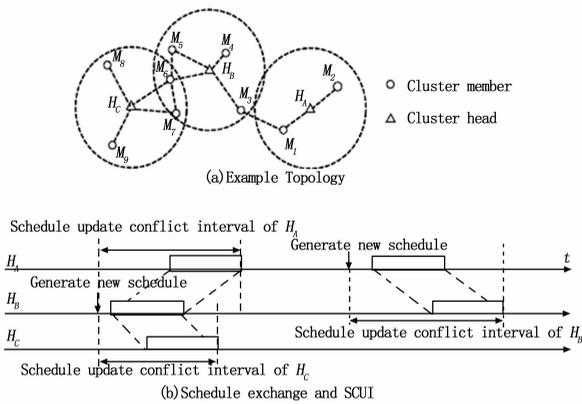


图 4 相邻簇头间的调度交换

点的最新调度信息, 因此它可以保证新的调度不会发生冲突。调度算法与算法 1 类似, 唯一区别就是它可以避免与 SCUI 重叠, 以便为  $T_u$  寻找合适的时隙和调度交换报文。然后, 簇成员遵守调度规则向同一簇内或簇外的相邻节点发送报文。

本文还引入基于 ID 的调度初始化策略, 以避免初始调度交换报文内发生报文冲突和调度冲突。具体来说, 簇头从所有的 NCH 接收到 ID 较小的调度交换报文前, 既不得生成初始调度规则也不得发送首个交换调度报文。在多跳情况下, 簇头可能接收到相邻簇内的节点 ACK。此时, 它只需要在调度交换报文内捎带确认 ACK 即可, 相应的 NCH 将会帮助传输。

### 2.4 报文差错对 COD-TS 的影响

从以上几节可以看出, 如果调度交换报文丢失, 则 COD-TS 性能将下降。虽然 COD-TS 可以避免报文冲突, 但是仍然可能因为信道状态不佳导致报文丢失。当出现这一问题时, 调度规则将会混乱, 且下一调度交换报文无法准确投递。这将严重影响 COD-TS 的性能。为此, 除了正常的调度交换外, 每个簇头还必须向其 NCH 定期发送最新调度规则。发送周期事先确定, 且要远大于通信回合, 以便避免开销过大, 同时每个簇头可以避免与这些报文的预留时隙产生调度冲突。有了这一机制后, 即使簇头因没有解码调度交换报文而发生了冲突, 在从同一 NCH 接收到周期性调度交换报文后, 也可以保证无冲突调度策略的有效性。

## 3 性能评估

本节通过仿真实验来验证节点吞吐量模型的合理性。同时比较本文的 MAC 协议与 Slotted ALOHA<sup>[6]</sup>、RTS/CTS<sup>[7]</sup> 和 Slotted-FAMA<sup>[8]</sup> 协议在多跳网络条件下的性能, 以证明时间共享 MAC 协议的优点。仿真平台为文献 [15] 中基于 NS-2 的水下声音网络模拟器 Aqua-Sim。用于验证模型的网络场景为单跳网络, 包括 4 对发送方和接收方, 共 8 个节点。在多跳网络场景下, 80 个节点均匀分布于大小为  $6000m \times 6000m \times 20m$  的网络中, 每个节点随机选择一个相邻节点作为接收方。两种场景下, 所有节点的传输范围相同, 均为 1100 米。每个发送方根据带有相同  $\lambda$  参数的泊松分布来生成大小为 400B 的数据报文。此外, 各种被评估方案中的节点维护一个永远不会溢出的长队列, 以便满足模型所做出的队列无限大的假设。鉴

于篇幅原因, 此处只给出低带宽网络节点吞吐量这一最重要指标来定量展现协议性能。

### 3.1 模型验证

从图 5 中可以看出, 3 种协议的节点吞吐量模型与相应的仿真结果均非常吻合, 因此证明了本文模型的有效性。因为冲突概率直接决定了 Slotted ALOHA 和基于 RTS/CTS 的 MAC 协议的节点吞吐量, 所以节点吞吐量模型验证后也证明了冲突概率模型的合理性。

对于 COD-TS, 当  $\lambda$  较小时, 节点吞吐量小于本文提出的下界模型。这一现象是合理的, 因为  $\lambda$  较小时需要较长时间才能到达通信回合稳态。在到达稳态前, 通信回合相对较短, 因此控制报文存在严重的开销, 降低了平均吞吐量。当  $\lambda$  较大时, 可在较短时间内在 MAC 层达到稳态的报文数量上升, 因此较短通信回合获得的较低吞吐量对平均数值不会产生太大影响。因此, 可以看出, 当  $\lambda > 0.05$  时, 实际测得的吞吐量实际上要大于等于本文模型。

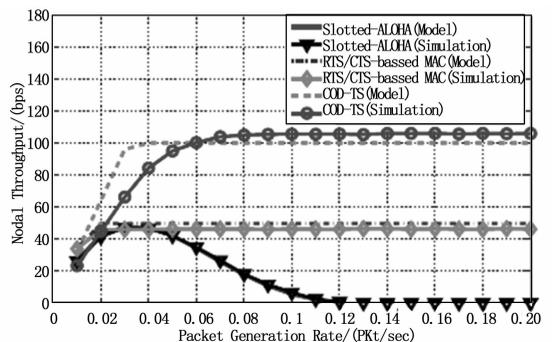


图 5 节点吞吐量模型和仿真结果比较

### 3.2 多跳网络节点吞吐量评估

(1)  $\lambda$  变化时的节点吞吐量:  $\lambda$  的变化范围为  $0.01 \text{ pkt/s} - 0.2 \text{ pkt/s}$ , 步进量为  $0.01 \text{ pkt/s}$ , 以比较 Slotted-ALOHA、Slotted-FAMA 和 COD-TS 的性能。比较结果见图 6。请注意, 多跳网络场景包括隐藏终端问题, 因此 3 种方案的节点吞吐量要低于单跳网络的测量值。

如图 6 所示, 当  $\lambda$  较小时, Slotted-ALOHA 的性能最优。这是因为 Slotted-ALOHA 的开销最小。当  $\lambda$  较小时, 冲突现象较少, 因此可以传输大多数数据报文。然而, Slotted-ALOHA 无法有效避免冲突, 因此当  $\lambda$  上升时, 冲突加剧, 吞吐量也相应趋近于 0。对于 Slotted-FAMA, 其吞吐量基本保持不变。这一结果由握手协议所决定。因为传输率较低, 前同步码较长, 所以即使  $\lambda$  较低, 到达的数据报文也超过了 Slotted-FAMA 的处理能力。此时, 报文的报文队列变满, 并以固定速率将数据报文发出。

从图 6 可以看出, 当  $\lambda$  较小时, COD-TS 的吞吐量较低。这是因为 COD-TS 使用一系列控制报文来实现其调度策略。当  $\lambda$  较小时, 通信回合较短, 因此这些控制报文, 尤其是调度交换报文, 导致严重开销, 严重影响了数据报文的可用时间。当  $\lambda$  变大时, 每回合期间可以发送更多数据报文, 因此开销的影响非常轻微。于是, 当  $\lambda$  大于  $0.05 \text{ pkt/s}$  时, COD-TS 的吞吐量上升, 并且优于其他两种方案。然而, 为了保证簇的公平性, 引入了  $D_{\max}$ , 所以受到  $D_{\max}$  的约束, COD-TS 的吞吐

量最后趋于常数。

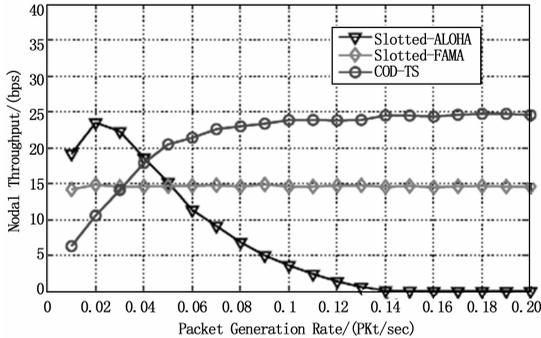


图 6  $\lambda$  变化时的节点吞吐量

(2) 报文尺寸  $L_D$  变化时的节点吞吐量: 在这组仿真中,  $\lambda = 0.08$  pkt/s,  $L_D$  的范围为 300~800 B, 步进量 100 B。图 7 仿真结果表明, COD-TS 优于其他两种方案。从图 7 可以看出, 当  $L_D$  上升时, COD-TS 和 Slotted-FAMA 的节点吞吐量上升。这是因为  $T_{pre}$  固定。具体来说,  $L_D$  越大, 报文传输的数据量越大。于是, 单位时间内传输的数据字节越多, COD-TS 和 Slotted-FAMA 的节点吞吐量上升。

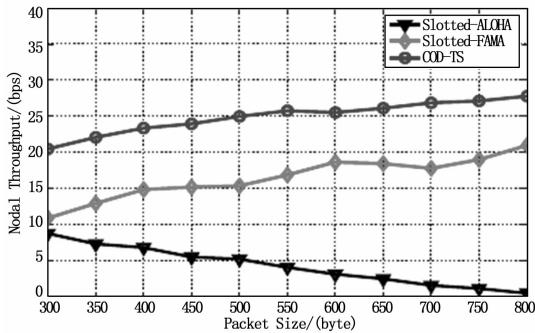


图 7 报文尺寸变化时的节点吞吐量

可以看出, Slotted-FAMA 的曲线并不平滑。这是由其时隙机制所导致。因为数据报文传输时间转换为时隙数量, 所以即使  $L_D$  只增加了一个字节, 数据报文也可能需要额外一个时隙进行传输。考虑到多增加了一个字节及一个额外时隙开销, 吞吐量肯定下降。于是, 在获得另一时隙前, 当  $L_D$  变大时, 节点吞吐量再次上升。因此, Slotted-FAMA 的曲线形状为梯形。对于 Slotted-ALOHA, 当  $L_D$  变大时, 时隙长度 (也就是一个数据报文的传输时间) 相应增加。于是, 更多报文开始排队, 多个节点在同一时隙发送报文的概率增加。因此, 当  $L_D$  增加时, 节点吞吐量下降。

#### 4 结论

为了研究实用性 MAC 协议, 文中分析了上述调制解调器特征对基于随机访问 MAC 和基于握手 MAC 的影响。通过分析, 认为基于时间共享的 MAC 协议是解决上述问题的潜

在方案。然而, 已有的协议在拓扑结构已知时提前调度时隙, 无法立即响应。为此, 本文提出了基于聚类的按需时间共享 MAC 协议方案 (COD-TS), 并提出相应的吞吐量模型。文中还进一步比较了 Slotted ALOHA、Slotted FAMA 和 COD-TS 的性能。结果与分析一致: COD-TS 优于其他两种 MAC 协议。下一步工作的重点是在实际调制解调器上部署本文的 COD-TS 协议。通过进行一系列现场测试 (湖泊和海洋) 来评估和调试本文设计, 以便增强本文协议在实践中的性能。

#### 参考文献:

- [1] Liu L, Zhou S, Cui J H. Prospects and Problems of Wireless Communication for Underwater Sensor Networks [J]. Wiley Wireless Communications and Mobile Computing, 2008, 8 (8): 977-994
- [2] 洪 璐, 洪 锋, 李正宝, 等. CT-TDMA: 水下传感器网络高效 TDMA 协议 [J]. 通信学报, 2012, 33 (2): 164-174
- [3] 郭忠文, 罗汉江, 洪 锋. 水下无线传感器网络的研究进展 [J]. 计算机研究与发展, 2010, 47 (3): 377-389
- [4] 金志刚, 苏毅珊, 刘自鑫. 基于运动预测的水下传感器网络 MAC 协议 [J]. 电子与信息学报, 2013, 35 (3): 728-734
- [5] 熊 鹏. 水声传感器网络中基于改进时分多址技术的 MAC 协议 [J]. 计算机应用, 2011, 31 (11): 2902-2908
- [6] Vieira L F M, Kong J, Lee U, et al. Gerla. Analysis of Aloha Protocols for Underwater Acoustic Sensor Networks [A]. Poster abstract in ACM WUWNet [C]. 2006: 112-118
- [7] Molins M, Stojanovic M. Slotted FAMA: a MAC Protocol for Underwater Acoustic Networks [A]. Proceedings of IEEE/MTS OCEANS [C]. 200: 1-7
- [8] Yackoski J, Shen C C. UW-FLASHR: Achieving High Channel Utilization in a Time-based Acoustic MAC Protocol [A]. Proceedings of ACM WUWNet [C]. 2008: 59-66
- [9] Petrioli C, Petrocchia R, Potter J. Performance evaluation of underwater Mac protocols: From simulation to at-sea testing [A]. IEEE OCEANS [C]. 2011: 1-10
- [10] Peng Z, Le S, Zuba M, et al, Y. Zhu, L. Pu, J. Liu, and J. H. Cui. Aqua-TUNE: A Field Testbed for Underwater Networks [A]. Proceedings of IEEE OCEANS [C]. 2011: 1-9
- [11] Ma J, Lou W. Interference-aware Spatio-Temporal Link Scheduling for Long Delay Underwater Sensor Networks [A]. Proceedings of IEEE SECON [C]. 2011: 1-9
- [12] Bandyopadhyay S, Coyle E J. Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks [A]. Proceedings of IEEE INFOCOM [C]. 2003: 1713-1723
- [13] Zhu Y, Zhou Z, Peng Z, et al. An efficient georouting aware Mac protocol for underwater acoustic networks [J]. ICST Transactions on Mobile Communications and Applications, 2011, 11 (9): 7-19
- [14] Guan Y, Sheng C C, yackoski J. MAC Scheduling for High Throughput Underwater Acoustic Networks [A]. Proceedings of IEEE WCNC [C]. 2011: 1-6
- [15] Xie P, Zhou Z, Peng Z, et al. Aqua-Sim: An NS-2 Based Simulator for Underwater Sensor Networks [A]. IEEE/MTS OCEANS [C]. 2009: 1-7