

CT-TDMA: 水下传感器网络高效 TDMA 协议

洪璐, 洪锋, 李正宝, 郭忠文

(中国海洋大学 信息科学与工程学院 计算机科学与技术系, 山东 青岛 266100)

摘要: 针对水下无线传感器网络 (UWSN, underwater sensor networks) 提出以发送端为中心以连续时间为计量单位的冲突状态模型——局部冲突状态图及其分布式构建算法, 并在此基础上设计了基于启发式规则的水下传感器网络 TDMA 协议 (CT-TDMA, continuous time based TDMA)。CT-TDMA 利用 UWSN 中同一接收节点与不同发送节点之间链路时延的差异性, 减少在目的端的接收帧之间的空闲时间, 从而提高网络流量; 基于启发式规则的分配算法, 能有效缩短连续时间轴上的时刻分配所花费的时间。模拟实验证明: CT-TDMA 与以 ST-MAC 为代表的按时隙分配的 TDMA 方案相比, 网络流量提高了 20%, 数据分组的端到端时延降低了 18%; 与由全局知识所计算出的最优分配策略相比, 网络流量达到了 80%, 端到端时延仅延长了 12%。

关键词: 水下传感器网络; MAC 协议; TDMA; ST-MAC

中图分类号: TP212.9

文献标识码: A

文章编号: 1000-436X(2012)02-0164-11

CT-TDMA: efficient TDMA protocol for underwater sensor networks

HONG Lu, HONG Feng, LI Zheng-bao, GUO Zhong-wen

(Department of Computer Science, College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China)

Abstract: Aimed at underwater acoustic sensor networks (UWSN), a novel sender based conflict model with the schemes of allocating continuous time was presented, including local conflict graph (LCG) and a distribute algorithm to generate LCG. Moreover, CT-TDMA, an efficient TDMA protocol based on the conflict model was also proposed, which used heuristic priority rules to allocate transmitting moments for all nodes. CT-TDMA exploits the diversity of propagation delay of different links in UWSN to decrease the idle time between packets at the same receiving node, which helps in improving the throughput. And a heuristic schedule algorithm is applied to shorten the process of allocating continuous time for each node. Simulation results show that, compared with traditional TDMA protocols such as ST-MAC, network throughput of CT-TDMA has increased 20% and end to end delay has decreased 18%; compared to the theoretically optimal scheme with global knowledge, CT-TDMA has achieved 80% network throughput and the end to end delay is only 12% longer.

Key words: underwater sensor networks; MAC protocols; TDMA, ST-MAC

1 引言

水下无线传感器网络 (UWSN, underwater sensor networks) 将采集到的海洋环境数据发送给用户来辅

助决策, 在环境监测、资源勘探、灾害预警和潜艇探测等民用和军用领域均具有广阔的应用前景^[1~3]。

UWSN 的重要特点是, 主要通信方式只能采用水声通信。这是因为高频无线信号会被海水吸收,

收稿日期: 2010-06-17; 修回日期: 2010-11-18

基金项目: 国家自然科学基金资助项目(60703082, 60873248, 60933011, 60970129); 青岛市科技发展计划基金资助项目(10-3-4-1-6-jch)

Foundation Items: The National Natural Science Foundation of China (60703082, 60873248, 60933011, 60970129); Science and Technology Development Program of Qingdao (10-3-4-1-6-jch)

光信号也会因海水散射和反射而大量损耗,两者均不能满足水下长距离通信的要求。与无线电信号相比,水声通信的主要特点是:传播时延大,水声信号的传播速率仅为 1 500m/s;通信距离长且带宽低,通信距离一般在 1~10km 级别,通信带宽仅在 10kHz 级别;误码率高、多途现象和多普勒频移现象均会导致传输错误^[4,5]。

由于水声通信的传播时延高和通信距离长,发送端无法判断信道是否冲突;同时,由于低带宽和误码率高的特性,使得 UWSN 中每次通信的分组长度必须受到限制。因此,解决 UWSN 的 MAC 问题的关键在于接收端判断冲突是否发生。

目前,国内外的研究者在 UWSN 的 MAC 问题上展开了深入的研究,主要包括接入竞争控制协议和无冲突接入协议 2 种。接入竞争控制协议,通过以接收端为中心的虚拟载波监听协议来解决 MAC 问题^[6~10],但由于水声通信的时延长,虚拟载波监听的握手机制降低了数据通信时间的比例,导致网络流量较低。

无冲突接入协议主要采用时分复用接入(TDMA),即采用集中式的方式来向 UWSN 内各个节点分配通信时间片^[11~14]。此类协议可根据数据传输完成需要的时隙数分为 2 种。第一种 TDMA 算法要求,一跳范围内的数据传输必须在一个时隙完成,称之为单时隙协议^[11,12]。单时隙协议的主要问题在于,为了保证不同距离的节点对在一个时隙内均能完成通信,时隙必须由最大的链路时延决定。时隙的长度是发送帧时和网络中最大链路时延之和,该问题同样影响了网络流量的提高。

为了提高网络流量,研究者提出了允许跨时隙完成数据传输的 ST-MAC 协议^[13]。该协议假设所有的链路时延都是帧时的整数倍,利用启发式规则完成多跳网络的时隙分配,实现了较高的网络流量。

基于时隙的分配本质上是分配各个节点的发送时刻。如果能不采用按时隙分配,而直接在连续时间轴上对每个节点分配发送时刻,将更好地利用 UWSN 网络中同一接收节点与不同发送节点之间链路时延的差异性,降低接收帧之间的空闲时间,提高网络流量。因此,本文提出了以连续时间分配为出发点的水下传感器网络的 TDMA 协议(CT-TDMA, continuous time based TDMA)。

CT-TDMA 的核心思想是,每个节点利用收集到的局部网络拓扑信息,以自身的发送行为可能引

起的冲突为依据,分布式地计算局部冲突状态图(LCG, local conflict graph)。所有节点按照启发式规则计算自己的优先级,然后根据 LCG 图中的所有邻居的优先级顺序完成发送时刻的标定。优先级计算的启发式规则综合了节点的度、负载和链路时延等重要因素,在有效缩短连续时间轴上时刻分配问题的计算时间的同时,保证了 TDMA 协议的流量和端到端时延等性能指标。

本文的主要贡献如下。

1) 针对水下传感器网络在 MAC 层设计上的独特问题——同一接收节点与不同发送节点之间链路时延的差异性不可忽略,设计了以发送端为中心以连续时间为计量单位的冲突状态模型——局部冲突状态图。在此基础上,本文提出了分布式的局部冲突状态图构建算法:每个节点利用以通信半径和干扰半径之和为半径的覆盖圆内的网络拓扑信息,分布式计算局部冲突状态图。

2) 提出了基于启发式规则的水下传感器网络 TDMA 协议,即节点以其局部冲突状态图为分配依据,以节点的度、负载和链路时延计算优先级,按照优先级顺序进行发送时刻标定。该启发式算法有效缩短了连续时间轴上时刻分配问题的运行时间。

3) CT-TDMA 协议采用基于连续时间的接入分配算法,与基于时隙的 TDMA 协议相比,缩短了目的端的接收帧之间的空闲时间,提高了接收端的信道利用率及整个网络的流量。模拟实验证明:在实验时间段内,CT-TDMA 方案与以 ST-MAC 为代表的 TDMA 方案相比,网络流量提高了 20%,数据分组的端到端时延降低了 18%;与由全局知识所计算出的最优分配策略相比,网络流量达到了 80%,数据分组的端到端时延仅延长了 12%。

本文其余部分组织如下:第 2 节对 CT-TDMA 设计方案进行详细介绍;第 3 节分析 CT-TDMA 的仿真实验结果,第 4 节概述近几年 UWSN 的 MAC 问题的相关研究;第 5 节是结束语。

2 CT-TDMA 设计

本节首先通过示例说明 CT-TDMA 的设计出发点,进而对基于连续时间的冲突状态模型——局部冲突状态图进行了详细说明。其次详细介绍 CT-TDMA 的执行过程,并给出关键函数和过程的伪代码描述。然后讨论了 CT-TDMA 中用于判断节点接入优先级的启发式规则。最后,结合实例来展

示 CT-TDMA 的运行过程和运行结果。

2.1 设计出发点

解决 UWSN 的 MAC 问题的关键在于,唯有接收端才能判断数据分组冲突是否发生。如图 1(a)所示,节点 A 和 B 向节点 C 发送数据。设 D_{AC} 和 D_{BC} 为链路 AC 和 BC 的传播时延且 $D_{AC} < D_{BC}$, T 为数据分组发送时间即帧时, t_a 和 t_b 为节点 A 和 B 的传输时刻。

在陆地传感器网络中,由于无线电信号的传播时延可以忽略,因此 TDMA 方案的时隙长度为 T 。只要由发送端 A 和 B 选择不同的时隙发送即可避免冲突。然而, UWSN 的链路的传播时延不可忽略,为了保证相邻时隙发送的帧无冲突,在本例中时隙长度至少为 $D_{BC} + T$ 。因此,接收端 C 接收到的帧序列在时间轴将变得稀疏,如图 1(b)所示。

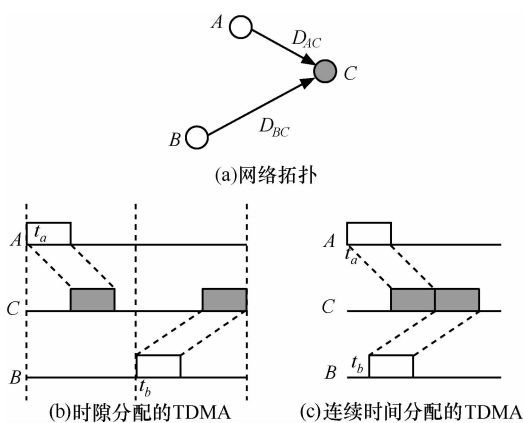


图 1 TDMA 不同策略下的发送时间的分配

而理想的 TDMA 效果应如图 1(c)所示, A 选择 t_a 时刻发送数据帧,则该帧到达 C 的时刻为 $t_a + D_{AC}$ 。为使得 C 接收到的帧序列间的空闲尽可能小,那么 B 发送的帧到达 C 的时刻应为 $t_a + D_{AC} + T$ (即 A 发送帧的帧尾和 B 发送帧的帧头相接),所以 B 节点的发送时刻应为 $t_b = t_a + D_{AC} + T - D_{BC}$ 。

图 1 给出了 UWSN 网络的普通拓扑情况。该例说明基于连续时间的接入分配相比于基于时隙的 TDMA 方案,能够提高接收端的流量。基于连续时间的分配充分利用了 UWSN 的链路时延长和数据分组短的特性,这正是 CT-TDMA 设计的出发点。

2.2 连续冲突状态模型

为了清晰地描述 CT-TDMA 的连续冲突状态模型,本节首先对运行环境做一些基本设定。

1) 使用圆型通信和干扰模型,即节点的通信范围和干扰范围都是一个以自己为圆心的圆。网络中

节点都是同构的,具有相同的通信半径 R_T (transmission radius)和干扰半径 R_I (interference radius)。传感器网络中的干扰半径 R_I 是对节点信号的干扰覆盖范围的一种描述:当源节点进行发送时,以源节点为圆心、 R_I 为半径的覆盖圆内的所有节点都由于源节点发送数据所带来的干扰,而不能正常接收来自其他节点的数据分组^[15, 16]。 R_I 在网络部署前进行测定,并与通信半径一样作为协议的初始化参数进行设定。

2) 节点通过时钟同步算法,保证足够的同步精确度。

CT-TDMA 的连续冲突状态模型包括局部冲突状态图和禁止时间计算规则。其关键术语的定义如下。

定义 1 覆盖范围函数 $Cov(N, R)$: 以节点 N 为圆心,以 R 为半径的区域。如节点 $N(x_n, y_n)$ 通信范围 $Cov(N, R_T) = \{(x, y) | (x - x_n)^2 + (y - y_n)^2 \leq R_T^2\}$; 干扰范围 $Cov(N, R_I) = \{(x, y) | (x - x_n)^2 + (y - y_n)^2 \leq R_I^2\}$

定义 2 求取某区域内的节点集的操作 $Nodeset(C)$: C 为水下载感器网络的某个部署区域,操作返回属于该区域的节点集,即 $Nodeset(C) = \{N | (x_n, y_n) \in C\}$ 。

定义 3 冲突: 节点不能够同时发送和接收,不能够同时接收 2 个以上的数据分组,所有违反这 2 条原则的发送和接收均称为冲突。以图 2 为例,如果 i 和 j 的信号同时到达 k , k 将无法接收 i 的数据分组;如果 i 的信号到达 k 时 k 正在发送, k 也无法接收 i 的数据分组,此 2 种情形均称为冲突。

定义 4 冲突邻居、冲突目标节点: 节点之间的冲突关系是网络物理拓扑和逻辑拓扑的直接反映。设 N 为网络中的任一发送节点,节点 N 的发送操作对周围节点可能造成的冲突主要表现在:对于不同于 N 的节点 M 而言,如果存在节点 P ,满足 P 的位置在物理拓扑中位于 M 和 N 其中一个节点的通信范围内,同时位于另一节点的干扰范围内,并且,在逻辑拓扑中存在链路 $N \rightarrow P$ 或者 $M \rightarrow P$,那么 P 节点在接收 N 或者 M 其中一个节点的数据分组时,可能会受到来自另一节点的发送信号的干扰而无法接收,即在 P 点形成冲突。

对于节点 N 来说,其干扰范围内的所有邻居都有可能由于节点 N 的发送而成为冲突发生点。因此,节点 N 的冲突目标节点和冲突邻居定义如下:

当节点 P 、 M 同时满足下列拓扑条件时,节点

P 是节点 N 的冲突目标节点, 节点 M 是节点 N 的冲突邻居:

① 物理拓扑条件:

$$P \in \text{Nodeset}(\text{Cov}(N, R_T) \cap \text{Cov}(M, R_T)) \parallel$$

$$P \in \text{Nodeset}(\text{Cov}(N, R_I) \cap \text{Cov}(M, R_T))$$

② 逻辑拓扑条件: $(N \rightarrow P) \parallel (M \rightarrow P)$

该定义说明, 节点的冲突邻居必位于它的 $(R_I + R_T)$ 的覆盖范围内。如图 2 中, 节点 k 处于节点 i 的通信范围和节点 j 的干扰范围内, 同时链路 $i \rightarrow k$ 存在, 因此节点 j 是节点 i 的冲突邻居, 节点 k 为相应的冲突目标节点。对于节点 n 来说, 节点 n 和节点 i 相距较远, 且两者的通信范围和干扰范围的交集范围内不存在节点, 因此不满足上述定义中的物理拓扑条件, 所以节点 n 不是节点 i 的冲突邻居。

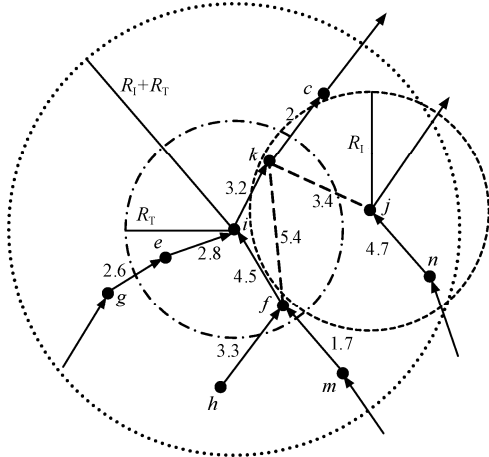


图 2 节点 i 的局部网络拓扑结构

定义 5 局部冲突状态图 (LCG, local conflict graph): 节点 i 的冲突状态图是一个以节点 i 为中心的带权多重图, 描述所有和 i 的发送行为有关的冲突。LCG 中的顶点为 i 及其所有的冲突邻居。LCG 图中边的权值定义为: 如果节点 j 为节点 i 的冲突邻居, 其冲突目标节点为 k , 且链路 ik 和 jk 的时延分别为 D_{ik} 和 D_{jk} , 则边 $i \rightarrow j$ 的权值为

$$W_{ij}(k) = D_{ik} - D_{jk} \quad (1)$$

为了表达式的一般性, 定义 $D_{ii}=0$ 。

$W_{ij}(k)$ 用来描述节点 i 和节点 j 发送的数据在接收点 k 产生冲突的状态。 $W_{ij}(k)$ 取正值表示, 如果节点 i 选择比节点 j 早 $W_{ij}(k)$ 的时间进行发送, 则会在 k 产生冲突。相应地, 如果 $W_{ij}(k)$ 为负值, 表示节点 i 选择比节点 j 晚 $|W_{ij}(k)|$ 的时间进行发送时会产生冲突。由于节点的所有冲突邻居均位于该节点的 LCG

中, 因此冲突邻居也就是节点的 LCG 邻居。

以图 2 为例, 节点 k 是节点 i 和节点 j 的冲突目标节点, 那么节点 i 的 LCG 图中存在边 $i \rightarrow j$, 权值为 $W_{ij}(k) = D_{ik} - D_{jk} = -0.2$ 。其含义为: 如果节点 i 选择比节点 j 晚 0.2s 进行发送, 则在节点 k 将同时接收到节点 i 的发送信号和节点 j 发向其他节点的数据信号的干扰能量, 从而发生冲突。

对于节点 i 和节点 k 来说, $\text{Nodeset}(\text{Cov}(i, R_T) \cap \text{Cov}(k, R_I)) = \{i, k\}$, 并且仅链路 $i \rightarrow k$ 存在, 由定义 4 可得, 节点 k 既是节点 i 的冲突邻居, 也是节点 i 和节点 k 冲突的目标节点, 且 $W_{ik}(k) = W_{ik} - W_{kk} = 3.2$ 。这说明如果节点 i 选择比节点 k 早 3.2s 进行发送, 那么当节点 i 的数据信号到达节点 k 时, 节点 k 正在向其他节点发送信号, 无法接收节点 i 的数据, 产生冲突。

再分析节点 i 和节点 f 的冲突情况, 由定义 4 的物理条件得到, $\text{Nodeset}(\text{Cov}(i, R_T) \cap \text{Cov}(f, R_I)) = \{i, k, f\}$ 。由于链路 $i \rightarrow f$ 不存在, 根据定义 4 的逻辑拓扑条件, f 不是冲突目标节点。该论断的含义是, f 不接收 i 的数据分组, 因此 i 发送的信号到达 f 时, f 可以进行发送, 即在 f 节点处不会产生 i 和 f 的冲突。但是链路 $i \rightarrow k$ 存在, 说明 k 是 i 和 f 的冲突目标节点, 其含义是 i 和 f 的信号可能同时到达节点 k 。同时, 链路 $f \rightarrow i$ 存在, 说明 i 同样是 i 和 f 的冲突目标节点, 其含义是 i 在接收 f 的信号时, 不能向其他节点发送数据。因此节点 i 的 LCG 图中, i 和 f 之间存在 2 条边, 其权值分别为 $W_{if}(k) = D_{ik} - D_{fk} = -2.2$ 和 $W_{if}(i) = D_{ii} - D_{if} = -4.5$ 。

定义 6 禁止时间: 节点的禁止时间是指在时间轴上的一个时间片段内, 节点不能够选择发送时刻, 否则会引起冲突。

为保证 2 个节点的帧到达目标时不冲突 (即在接收端的时间轴上不重叠), 对每个发送节点来说均有一段不能发送, 称为禁止时间。以图 3 为例, 在节点 i 的 LCG 中, 如果其邻居 j 选择发送时刻 t_j , 则由节点 j 决定的节点 i 的禁止时间定义为 $(F_l(i, j, k), F_r(i, j, k))$ 。其中

$$F_l(i, j, k) = t_j - W_{ij}(k) - T \quad (2)$$

$$F_r(i, j, k) = t_j - W_{ij}(k) + T \quad (3)$$

禁止时间的物理含义为, 由于节点 j 的存在和其发送时刻 t_j 的确定, 节点 i 在 $(F_l(i, j, k), F_r(i, j, k))$ 时间段内不能发送数据帧。如图 3 所示, 如果节点

i 选取的发送时刻大于 $F_l(i, j, k)$, i 发送的帧尾将与 j 发送的帧头在相关接收点 k 产生冲突; 当 i 选取的发送时刻稍小于 $F_r(i, j, k)$, i 发送的帧头将与 j 发送的帧尾在相关接收点 k 产生冲突。

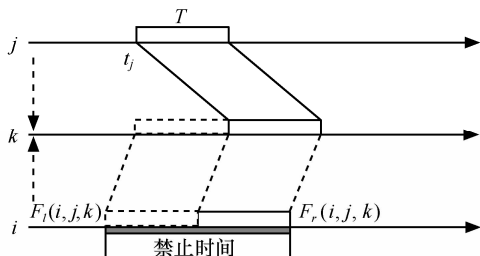


图 3 禁止时间的计算

以图 2 中的节点 i 和 j 的冲突为例, $W_{ij}(k) = -0.2$, 假设节点 j 选择第 2s 发送($t_j=2$), 帧时 $T=1s$, 则根据式(2)和式(3)可得节点 j 给节点 i 带来的禁止时间为(1.2, 3.2)。以节点 i 和 f 为例, 根据前面的分析, 冲突状态图中边 $i-f$ 的权值有 2 个, 因此假设 f 选择 0s 开始发送, 根据公式计算出的 f 给 i 带来的禁止时间也是 2 段, 为 $(3.5, 5.5) \cup (1.2, 3.2)$ 。

2.3 CT-TDMA 的主要算法

本节首先介绍局部冲突状态图的生成过程, 然后介绍 CT-TDMA 的算法流程。CT-TDMA 可以用于任何网络拓扑结构, 但为了简化说明, 本节针对最广泛使用的树形拓扑结构进行介绍。

2.3.1 LCG 生成算法

在初始化阶段, 每个节点 i 收集 $Cov(i, R_T+R_I)$ 范围内的邻居节点的地理位置信息和逻辑链路信息, 并计算其冲突区域内所有节点对的信号传播时延, 得到以该节点为中心的局部网络拓扑结构。对于位于节点通信半径之外的邻居, 节点通过多跳传输的方式获得其对应邻居的相关信息。在计算得到 $Cov(i, R_T+R_I)$ 内的局部网络拓扑结构后, 节点运行 createLCG() 函数, 按照定义 5 中给出的规则生成自己的 LCG。

利用定义 4 验证 j 是否是 i 的冲突邻居。PTC() 和 LTC() 函数用来对节点是否符合冲突的物理拓扑条件(physical topology condition)和逻辑拓扑条件(logical topology condition)进行判定。针对每一个邻居 j 进行分析: 如果存在某个节点 k , 使得 k, j 符合冲突的 2 个条件, 满足节点 j 是节点 i 的冲突邻居, k 是 i 和 j 冲突的目标节点, 则计算 $W_{ij}(k) = D_{ik} - D_{jk}$, 并记录到 i 的 LCG 中。

算法 1 createLCG ()

```

1) Input:  $V(i)$ 
2) Output:  $S_{LCG}(i)$ 
3) PTC( $N, P, M$ ) { //物理拓扑条件判定
4) if ( $P \in Nodeset(Cov(N, R_T) \cap Cov(M, R_I)) \parallel$ 
 $P \in Nodeset(Cov(N, R_I) \cap Cov(M, R_T))$ )
5) return true
6) else
7) return false
8) }
9) LTC( $N, P, M$ ) { //逻辑拓扑条件判定
10) if ( $\exists N \rightarrow P \parallel \exists M \rightarrow P$ )
11) return true
12) else
13) return false
14) }
createLCG()
15) new  $V_{LCG} = V(i);$  //  $Cov(N, R_T+R_I)$  邻居顶点集
16) while ( $V_{LCG} \neq null$ ) {
17)  $\forall j \in V(i)$  //讨论边  $i-j$  的情况
18)  $V_{LCG} = V_{LCG} - \{j\}$ 
19) new  $V' = V(i)$  //以下逐个节点分析,  $i$  节点和  $j$  节点是否会在  $V(i)$  中的节点产生冲突
20) while ( $V' \neq null$ ) { //遍历所有可能的冲突目标节点
21)  $\forall k \in V'$ 
22)  $V' = V' - \{k\}$ 
23) If ( $PTC(i, k, j) = true \ \&\& \ LTC(i, k, j) = true$ ) { //冲突邻居和冲突目标节点判定
24)  $W_{ij}(k) = D_{ik} - D_{jk}$ 
25)  $S_{LCG}(i) = S_{LCG}(i) \cup \{W_{ij}(k)\}$  //将  $i$  节点和  $j$  节点的一条边的权值记录到  $S_{LCG}$  中
26) } //end if
27) } //end while ( $V' \neq null$ )
28) } //end while ( $V_{LCG} \neq null$ )
29) return  $S_{LCG}(i)$ 

```

令 $V(i)$ 是节点 i 的所有 $Cov(I, R_T+R_I)$ 内的邻居节点的集合, $S_{LCG}(i)$ 为 i 的冲突状态图内所有的边的权值集合, 算法 1 描述了节点局部冲突状态图的生成过程。

以图 2 的拓扑为例, 节点 i 运行 createLCG() 算法后, 生成的 LCG 如图 4 所示。根据前面的分析,

节点 f 和 i 冲突的目标节点有 2 个, 根据定义 5 计算出边 $i-f$ 有 2 个权值, 因此在局部冲突状态图中体现为具有不同权值的多重边。如果 2 个节点对会在多个节点处产生冲突, 则 LCG 图中这 2 个节点对之间就会有多重边, 且多重边的个数即为冲突目标节点的个数。因此, 由于节点冲突的多样性, LCG 为带权的多重图。

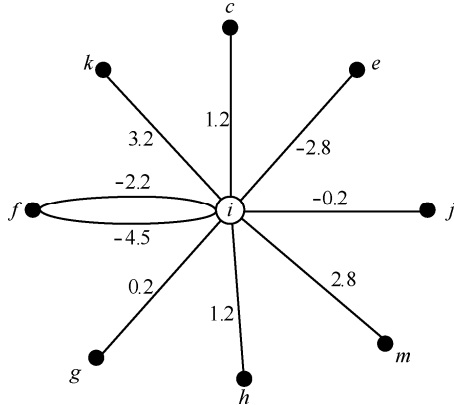


图4 节点 i 的局部冲突状态图 (LCG)

2.3.2 CT-TDMA 算法流程

每个节点生成自己的 LCG 图后, 就根据由 LCG 图每条边所对应的各个禁止时间, 来选择自己的发送时刻。即选择和所有禁止时间不冲突的最早时刻, 作为自己的发送时刻。

但是, 当前节点的发送时刻的确定, 取决于其 LCG 邻居的发送时刻。CT-TDMA 采用优先级和随机选择相结合的方式, 安排节点的发送时刻标定顺序。即优先级最高的节点最先选择发送时刻, 选择的基本策略是在所有可用的时间段内选择一个最早的时间点 (选择的过程称为标定)。如果 LCG 图中所有未标定节点的优先级相同, 各个节点随机选择发送时刻。优先级的计算方法, 将直接影响 CT-TDMA 的执行效率。本节先介绍 CT-TDMA 的算法流程, 而计算优先级的启发式规则将在 2.3.3 节中进行讨论。

CT-TDMA 的完整算法流程包括以下 7 步。

- 1) 节点生成自己的 LCG 并计算自己的标定优先级。
- 2) 每个节点向所有的 LCG 邻居广播自己的标定优先级, 并将自己的状态置为 “unlabeled”。然后, 每个节点以分布式方式运行 3)~6) 步。
- 3) 如果当前节点在其 LCG 图中的所有未标定邻居中拥有最高的优先级, 则节点根据由 LCG 得

到的所有禁止时间段, 选择最早的可发送时刻作为自己的发送时刻, 并将其广播给所有的 LCG 邻居, 并将状态置为 “labeled”。

4) 如果当前节点的所有未标定邻居中存在优先级高于自己的邻居, 则等待优先级高的节点先标定, 等待接收相关节点的发送时刻结果并更新自己的禁止时间, 返回第 3) 步。

5) 如果当前节点在其 LCG 图中与其他未标定的节点拥有相同的优先级, 则该节点在时间轴上除禁止时间之外的时刻随机选择其发送时刻, 并广播给所有同优先级的未标定状态的 LCG 邻居。

6) 随机标定的节点选定发送时刻并与所有同优先级邻居交换选定结果后, 运行 Judge() 函数判断是否会产生冲突, 若无冲突, 则置为 “labeled”, 否则置为 “unlabeled”, 并返回第 5) 步。

7) LCG 图中的所有节点状态均为 “labeled” 时, 算法结束。

算法 2 Label()

Initialization:

- 1) $S_c(i) = null; S_T(i) = unlabeled$
- 2) $S_{uc}(i) = \{j | j \text{ is } i\text{'s LCG neighbor}\}$

Label()

- 3) while ($\exists j \in S_{uc}(i), P(i) < P(j)$) { // 顺序标定, 节点优先级非最高, 等待并接收其他节点标定结果
- 4) receive node j 's selection t_j
- 5) $S_c(i) = S_c(i) \cup \{j\}$
- 6) $S_{uc}(i) = S_{uc}(i) - \{j\}$
- 7) calculate $F_l(i, j, k), F_r(i, j, k)$ // 计算禁止时间
- 8) } // end while
- 9) loop:
- 10) if ($\exists j \in S_{uc}(i), P(i) = P(j)$) { // 优先级相同则随机标定
- 11) choose t_i randomly on available time axle
- 12) send t_i and receive other nodes' transmitting moments
- 13) for each $j (j \in S_{uc}(i) \ \&\& \ P(i) = P(j))$ calculate $F_l(i, j, k)$ and $F_r(i, j, k)$
- 14) if (Judge() = false)
- 15) continue loop
- 16) else {
- 17) $S_T(i) = labeled$
- 18) return t_i

```

19) }
20) } //随机标定结束
21) else{ //节点自己的优先级最高,开始标定
22)    $t_i=0$  //从 0 开始寻找最早的可用时间
23)   for each  $j \in S_c(i)$  {
24)     if( $F_l(i, j, k) < t_i < F_r(i, j, k)$ ) //如果当前的
 $t_i$  位于禁止时间内, 则增大  $t_i$ 
25)        $t_i = Fr(i, j, k)$ 
26)   }
27) }
28)  $S_T(i) = labeled$ 
29) return  $t_i$ 

```

节点使用 Label()函数实现标定功能,即上面的 3)~6)步, Judge()函数被 Label()调用,用来判断一次随机选择时间以后是否会产生冲突。在随机标定中,节点获得邻居随机选定的时刻后仍然使用定义 6 计算禁止时间。Judge()函数判断冲突的方式是,如果节点自己随机选定的时刻处于计算出的“禁止时间”之内,则认为会产生冲突,此次随机标定即作废,节点及其邻居重新进行随机标定。

定义 $S_c(i)$ 为节点 i 的已经标定的 LCG 邻居集合, $S_{uc}(i)$ 为节点 i 的所有未标定的 LCG 邻居的集合, $S_T(i)$ 为节点 i 的标定状态(labeled 或 unlabeled), $P(i)$ 为节点 i 的标定优先级。算法 2 和算法 3 分别给出了 Label()和 Judge()函数的伪代码描述。

在 Label()函数中引入随机标定的设计,是因为在某些网络中可能存在大量节点的优先级相同。虽然可以采用节点 ID 等强制优先级方式,但类似的优先级策略对优化最终的总传输时间没有帮助。随机标定可以减少算法的执行时间。比如,在极端的网络拓扑条件下(如线形、星形),当节点的优先级均相同时,顺序标定算法的时间复杂度为 $O(n)$ 而随机标定的时间复杂度为 $O(\log n)^{[17]}$ 。其中, n 为网络中的节点个数。

算法 3 Judge()

```

1) Input:  $t_i, S_{uc}(i), V(i)$ 
2) Output: true or false
Judge()
3) Boolean  $B=true$ 
4) for each  $j \in S_{uc}(i)$  {
5)   if( $\exists k \in V(i), F_l(i, j, k) < t_i < F_r(i, j, k)$ )
6)      $B=false$ 
7) }

```

```

8) return  $B$ 

```

2.3.3 标定优先级确定

在 UWSN 环境下,对于所有节点的连续时间的接入分配,等价于一个连续轴上的分布式着色问题,因此确定最优全局分配策略是 NP-hard 问题。于是,在上一小节介绍的 CT-TDMA 算法流程中,使用了按照优先级顺序进行标定的算法。这样,优先级的确定直接影响 CT-TDMA 算法本身的执行效率。本节对优先级的计算方法进行详细说明。

CT-TDMA 的算法目的是提高网络流量,即所有节点均发送一次数据所使用的总时间越短越好。针对这个设计目标,提出重要性依次降低的 3 条启发式的优先级制定规则如下。

1) 在冲突状态图中拥有最大度的节点应该优先分配发送时刻。节点的度越大意味着它潜在的冲突节点越多。度最大的节点首先分配发送时间,可以使受其影响的大量节点的禁止时间尽可能地靠前,从而缩短最后算法选定的总时间。

2) 负载较重的节点优先发送。负载较重的节点优先发送有助于缩短每个分组端到端的平均时延,实现网络的公平性。

3) 网络中拥有较大链路时延的节点优先发送。时延较高的链路如果发送的较晚,到达目标时的时间就会更晚。而优先分配时延较高的链路,使得高时延的链路早发送,低时延的链路晚发送,有助于提高节点的传输并行度。

以上述 3 条启发式规则为依据,可以实现定量计算节点的优先级指标。 $P(i)$ 表示节点 i 的标定优先级,采用式(4)进行计算。式(4)中使用的主要变量定义为: d_i 为 i 在它自己的 LCG 图中的度, L_{max} 为节点缓冲区的大小, L_i 为节点 i 的缓冲区队列长度。定义从节点 i 到基站的传输路径上的下一跳节点 k 为节点 i 的父节点, D_{ik} 为节点 i 到其父节点 k 的链路传播时延。定义 D_{max} 是整个网络中的最大的链路传播时延,即网络中距离最长的一跳链路长度与水声信号的传播速度的比值,该值在网络部署时可以进行估计并设定在所有节点中。由于 CT-TDMA 协议运行的环境为静态网络,该值在协议运行过程中不会改变。而且,该值仅涉及到优先级的计算,对于所有节点的计算效果是等价的,因此该值的估计精度对于本文提出的 CT-TDMA 协议的运行效果没有直接影响。

$$P(i) = d_i \times L_{\max} + L_i + \frac{D_{ik}}{D_{\max}} \quad (4)$$

3 条启发式规则的重要性在式(4)中直接体现: 拥有最大度的节点拥有最高的标定优先级; 如果节点的度相同则负载较重的节点优先级高; 如果节点的负载也相同, 则拥有最长链路时延的节点优先级高。当网络的拓扑结构, 节点负载等条件不发生变化时, 网络可以根据 CT-TDMA 计算出的调度顺序持续的运行。而当网络拓扑结构或者节点负载等条件发生变化时, 即节点的标定优先级发生变化时, 协议将在整个网络重新运行, 以保持节点的发送时间分配始终为当前的最优策略。

2.4 算法实例

为了清晰地说明 CT-TDMA 的基本思想、工作流程及特点, 本节给出一个简单的网络实例进行说明, 如图 5(a)所示。链路上的标记为该条链路的时延(单位为 s), 设帧时 T 为 1s。

各节点首先运行 createLCG()函数生成自己的局部冲突状态图, 4 个节点的 LCG 如图 5(b)所示。

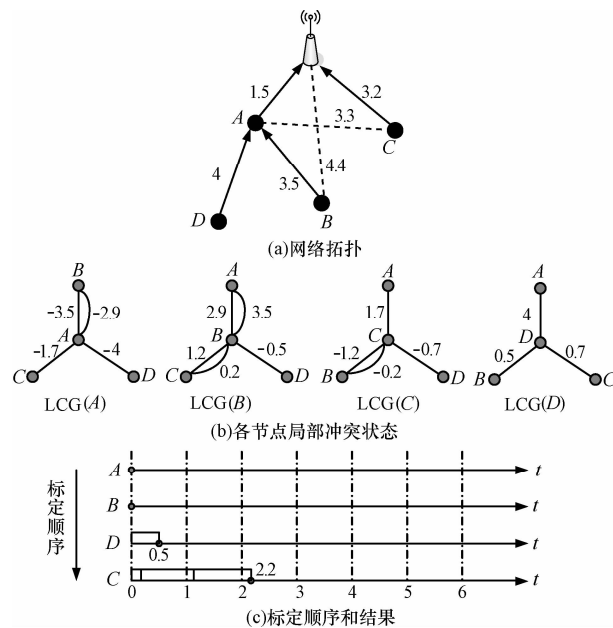


图 5 CT-TDMA 的执行过程
(实线表示逻辑链路, 虚线表示点到点之间的时延)

然后, 所有节点计算自己的优先级。在图 5(b)中 4 个节点的度相同, 在所有节点产生数据分组的频率相同的情况下, 对于 A 节点来说, 它不仅需要发送自己的数据分组, 还要转发来自 B 和 D 的分组, A 节点的缓冲区队列长度必将比 B、C、

D 节点的更长, 因此 A 节点的负载最重, 优先级最高。B、C、D 负载相同, 因此按链路的时延排序, 相关链路时延高者优先级高, 最终标定顺序为 A、B、D、C。

因此, A 首先选择 0 时刻并广播给 B、C、D。B 根据公式计算出 A 对 B 的禁止时间为 $(-3.9, -1.9) \cup (-4.5, -2.5)$, 所以 B 也可选择 0 时刻。D 根据 A 和 B 的标定结果, 计算 A 和 B 对自己的禁止时间为 $(-5, -3) \cup (-1.5, 0.5)$, 于是 D 选择自己的发送时刻为 0.5。最后, C 计算 A、B、D 对自己带来的禁止时间为 $(-2.7, -0.7) \cup (0.2, 2.2) \cup (-0.8, 1.2)$, C 选择自己的发送时刻为 2.2s。4 个节点的正数禁止时间在图 5(c)中给出。禁止时间的负数部分不影响节点发送时刻的选择, 故未给出。最后, A、B、C 和 D 4 个节点分别确定自己的发送时刻为 0s、0s、2.2s 和 0.5s。

本例说明, CT-TDMA 提高了数据发送的并行度。例如, A 和 B 节点都在 0 时刻发送而不会冲突, 这在传统的 WSN 中是无法实现的。UWSN 正是由于传播时延高、通信距离长和数据分组短, 导致了不同发送节点的数据分组到达接收端的时刻存在差异。CT-TDMA 利用 UWSN 的这种特性, 避免了冲突, 提高了网络中数据传输的并行度, 达到提高网络流量的目的。

3 仿真

本节介绍 CT-TDMA 的仿真结果, 并对算法的性能进行分析和对比。使用了 MATLAB 等软件对算法进行了仿真, 在仿真过程中, 帧时 T 设定为 1s, 节点的通信半径为 3000m, 干扰半径为 3500m, 网络包括从 5 个节点到 60 个节点的 4 种规模。根据网络规模的变化, 节点被部署在最大 $12\text{km} \times 12\text{km}$ 的范围内, 通过随机部署的方式, 利用部署密度限制节点和邻居节点之间的距离在 450m 到 3000m 之间。因为水声信号的速度为 1500m/s, 所以网络中链路的时延被控制在 0.3s~2s 之间。

由于网络拓扑结构对节点的传输并行度有重要影响, 因此对比了一般的网络拓扑结构(节点随机投放并自组织成树形拓扑结构), 以及特殊的星形、线形和网格, 共 4 种拓扑结构对算法的影响。在 4 种拓扑结构下, 均采用 4 种网络规模完成了 16 组实验。在仿真中使用平均每秒钟整个网络发送的有效信息量作为评价网络流量的指标, 即节点发送

的数据信息字节数。ST-MAC 帧长度为 45byte^[13] (40byte 有效数据), CT-TDMA 的帧长度为 105byte (100byte 有效数据)。在仿真中模拟了 4 种不同的接入分配算法。

1) **Optimal**: 实验网络部署情况下的理论上最优的分配策略, 该策略能够最大限度地利用信道和提高节点的传输并行度, 为其他算法提供理论的最佳上界。最优策略的求解, 即针对实验环境下的网络拓扑, 在所有可能的节点接入方案中, 选取全部节点的一轮接入时间最短的接入方案。

2) **CT-TDMA**: 本文提出的分配策略。

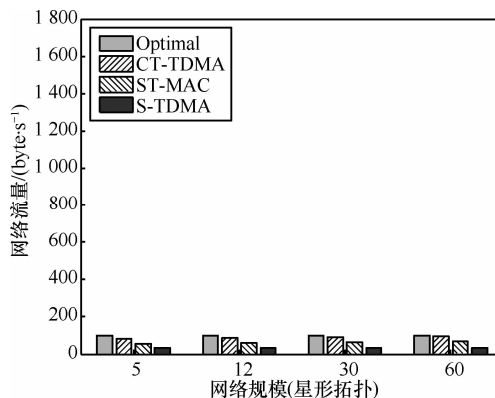
3) **ST-MAC**: 文献[13]提出的跨时隙 TDMA 分配策略。

4) **S-TDMA**: 传统的 UWSN 单时隙 TDMA。

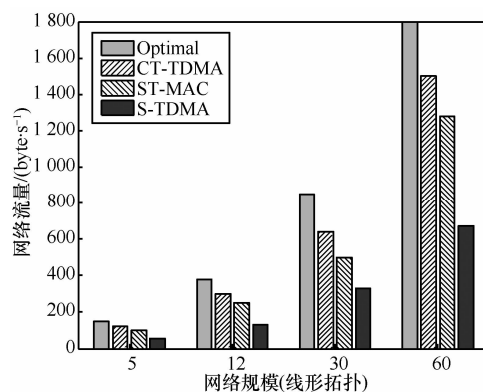
图 6 中对比了 4 种调度算法在 4 种不同网络拓扑结构下的流量情况。可以看出, 网络拓扑结构对流量有明显的影响, 线形、树形、网格到星形网络流量依次降低, 这是由于随着拓扑的变化节点的密度逐渐增加, 即共享同一个信道的节点数目越来越多, 因此能够同时并行传输的节点数目就变得越来越少。除星形拓扑之外, 在其他拓扑结构和协议下, 网络流量均随着网络规模的增加而增加。星形拓扑结构表现特殊的原因是, 在该环境中所有节点均一跳传向基站, 因此任意 2 个节点均互为冲突邻居, 所有的节点共享同一个信道, 无法提高数据传输的并行度。

图 6 显示, 当网络规模较小时, CT-TDMA 和 ST-MAC 均能够达到或接近理论最优的效果。当网络规模逐渐增大时, CT-TDMA 和 ST-MAC 的流量均不能达到理论最优值, 但 CT-TDMA 的性能始终优于 ST-MAC (平均高 20% 以上), 并达到理论最优值的 80% 左右。而 S-TDMA 的表现最差, 因为该方案较长的时隙给接收端带来大量空闲时间。

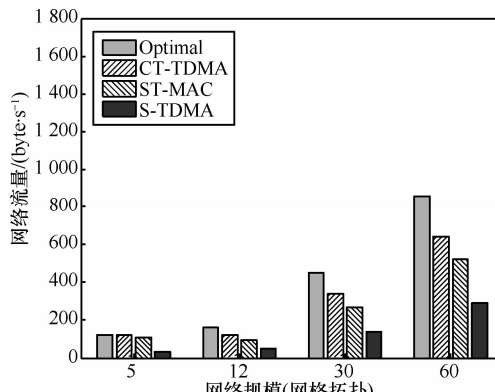
CT-TDMA 优于 ST-MAC 的原因在于, ST-MAC 采用时隙调度策略为节点分配发送时间, 这种策略规定节点只能在每个时隙的开始时刻发送帧。此外, ST-MAC 要求所有链路时延是帧时的整数倍。为了确保这一点, ST-MAC 协议中的分组长度必须尽可能小, 进一步引起帧中有效数据比例下降。其原因是, 虽然帧长度缩短, 帧中的校验和确认位并不减少。CT-TDMA 采用连续时间分配策略, 并不需要 ST-MAC 的上述设定, 因此可以应用更长的数据帧, 从而提高了网络流量。



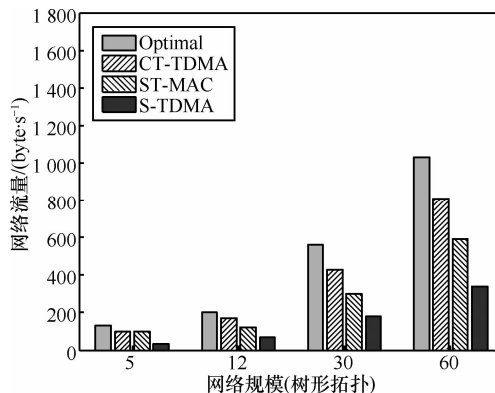
(a) 星形拓扑下的流量



(b) 线形网络拓扑下的流量



(c) 网格拓扑下的流量



(d) 树形拓扑结构下的流量

图 6 4 种协议在不同网络规模和拓扑结构下的流量对比

图 7 记录了 4 种协议在一般网络情况即树形拓扑结构下的端到端时延情况。传感器网络的数据传输模式为所有的节点向基站传输数据, 因此端到端时延即为数据分组从源节点到基站的传播时延。从图 7 可以看出, 随着网络规模的增大, 节点到基站的平均跳数增加, 端到端时延也相应增加。由于 CT-TDMA 使用了连续时间分配策略, 在接收节点处缩短了帧与帧之间的空闲时间, 提高了传输效率, 其平均端到端时延比 ST-MAC 降低了约 18%, 并且只比理论最优策略的时延高 12%。可见, CT-TDMA 不仅网络流量要高于传统协议, 平均的端到端时延也更低, 实时性更好。

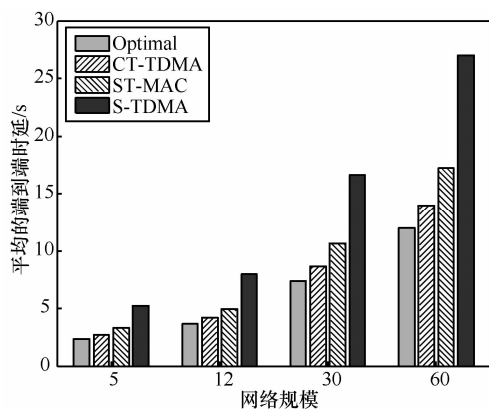


图 7 4 种协议的平均端到端时延

从计算代价的角度分析, ST-MAC 为集中式算法, 需要从所有的全局调度方案中寻找最优方案, 而 CT-TDMA 采用分布式的调度策略, 每个节点的计算代价都比 ST-MAC 小很多, 因此计算时间更短。从能量消耗的角度分析, ST-MAC 和 CT-TDMA 均属于 TDMA 协议, 数据传输阶段没有冲突, 因此能量效率都很高。额外的能量消耗仅在于协议的初始化阶段, ST-MAC 需要基站收集全网络的信息和分发调度策略, CT-TDMA 仅需要各个节点与其 $Cov(R_T+R_T)$ 范围内的邻居进行信息交换。当网络规模急剧增大时, ST-MAC 洪泛和广播的通信开销要远大于 CT-TDMA 的分布式信息交互。

4 相关工作

近年来, UWSN 的 MAC 问题由于水声通信的独特特点引起广泛关注。相关研究成果可分为基于竞争的接入控制协议和无冲突的接入协议 2 类。

接入竞争控制协议, 通过以接收端为中心的虚拟载波监听协议来解决 MAC 问题。文献[9]使用

RTS/CTS 握手机制来建立信道预约。节点通过截获邻居信息的方式计算各个邻居节点的忙时间, 然后在这段时间内停止侦听信道。文献[6]研究了水声信道特征对 Aloha 协议的影响, 进而提出水声通信的冲突状态具有时空不确定性的特点。文献[8]和文献[10]基于 Aloha 协议, 使用一个短帧来预约信道, 避免数据分组的冲突。T-Lohi 协议^[7]针对水声信道冲突的时空不确定性设计了一个特殊的退避算法, 在高负载的环境中具有较高的流量。

无冲突的接入协议基于 TDMA 和 CDMA。针对 UWSN 设计的 TDMA 协议分为单时隙和跨时隙 2 种, 单时隙协议要求一跳范围内的数据传输必须在一个时隙完成。UWAN-MAC^[11]允许节点随机地选择时隙进行传输。文献[12]提出的混合协议将 TDMA 和竞争协议相结合, 充分利用了竞争协议时延较低而 TDMA 流量较高的优点。

跨时隙的协议允许使用多个时隙完成数据分组传输。最近提出的 ST-MAC^[13]是第一个跨时隙的 TDMA 协议, 该协议以整个网络的冲突状态图为基础, 采用集中式算法计算各个节点的最优发送时隙。ST-MAC 需要收集全局信息并且假设网络中的任意链路时延都是发送帧时的整数倍, 通过分配时隙的方式, 使得不同节点的帧到达目的端的时隙不相同(即不会产生冲突)。该协议的时隙长度比传统协议要短的多, 利用链路时延的差异性减少了接收帧之间的空闲时间, 与传统的单时隙的 TDMA 相比提高了网络流量。本文提出的 CT-TDMA 和 ST-MAC 相比, 不依赖于时延与帧时关系的假设, 并且采用连续时间的分配方式代替时隙分配, 减少了信道空闲时间, 提高了网络流量。

除了 TDMA 之外, 研究者还提出了针对水下环境的改进 CDMA 协议^[18, 19], 文献[19]提出的混合协议使用对节点分簇的策略, 簇内节点通信使用 TDMA 协议, 不同的簇之间的节点通信则使用 CDMA 协议。然而目前在 UWSN 中使用 CDMA 协议还面临很多困难, 对大规模的传感器节点分配伪随机码是一项艰巨的挑战, 此外, 水声通信中使用 CDMA 的远近效应问题也没有得到很好地解决^[18]。

5 结束语

针对水下无线传感器网络时延高、带宽低和误码率高的特性, 本文提出了以发送端为中心以连续时间为计量单位的冲突状态模型——局部冲突状态

图 (LCG) 及其分布式构建算法, 并在此基础上设计了基于启发式规则的水下传感器网络 TDMA 协议。CT-TDMA 利用 UWSN 中同一接收节点与不同发送节点之间链路时延的差异性, 与现有协议相比进一步减少了节点接收帧序列的空闲时间, 提高了网络流量。并且, 基于启发式规则的分配算法, 有效缩短了节点在连续时间轴上进行时刻分配的运行时间。仿真表明, 与 ST-MAC 和 S-TDMA 等基于时隙分配的水下传感器网络 TDMA 协议相比, CT-TDMA 能够实现更高的网络流量和更低的端到端传播时延。

参考文献:

- [1] 李建中, 高宏. 无线传感网络的研究进展[J]. 计算机研究与发展, 2008,45(1): 1-15.
LI J Z, GAO H. Research progress of wireless sensor networks[J]. Journal of Computer Research and Development, 2008,45 (1): 1-15.
- [2] 李瑞芳, 李仁发, 罗娟. 无线多媒体传感器网络 MAC 协议研究综述[J]. 通信学报, 2008,29 (8): 111-123.
LI R F, LI R F, LUO J. Research review on MAC protocols of wireless multimedia sensor networks[J]. Journal on Communications, 2008, 29 (8): 111-123.
- [3] HEIDEMANN J, LI Y, SYED A. Research challenges and applications for underwater sensor networking[A]. Proc WCNC[C]. Las Vegas, NV, USA, 2006.228-235.
- [4] SYED A, HEIDEMANN J. Time synchronization for high latency acoustic networks[A]. Proc INFOCOM 2006[C]. Barcelona, Catalunya, Spain, 2006.1-12.
- [5] HARRIS A F III, STOJANOVIC M, ZORZI M. When underwater acoustic nodes should sleep with one eye open: Idle-time power management in underwater sensor networks[A]. Proc ACM WUWNET[C]. Los Angeles, CA, USA, 2006.105-108.
- [6] SYED A, YE W, HEIDEMANN J. Understanding spatial-temporal uncertainty in medium access with ALOHA protocols[A]. Proc ACM WUWNET[C]. Montr al, Qu bec, Canada, 2007.41-48.
- [7] SYED A, YE W, HEIDEMANN J. T-lohi: a new class of MAC protocols for underwater acoustic sensor networks[A]. IEEE INFOCOM[C]. Phoenix, AZ, USA, 2008.231-235.
- [8] GUO X, FRATER M, RYAN M. A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks[A]. OCEANS 2006-Asia Pacific[C]. Singapore, 2006.1-6.
- [9] MOLINS M, STOJANOVIC M. Slotted FAMA: a MAC protocol for underwater acoustic networks[A]. OCEANS 2006-Asia Pacific[C]. Singapore, 2006.1-7.
- [10] CHIRDCHOO N, SOH W, CHUA K. Aloha-based MAC protocols with collision avoidance for underwater acoustic networks[A]. Proc IEEE INFOCOM[C]. Anchorage, Alaska, USA, 2007.2271-2275.
- [11] PARK M, RODOPLU V. UWAN-MAC: an energy-efficient MAC protocol for underwater acoustic wireless sensor networks[J]. IEEE Journal of Oceanic Engineering, 2007, 32(3): 710-720.
- [12] KURTIS I, KREDO B, MOHAPATRA P. A hybrid medium access control protocol for underwater wireless networks[A]. Proc ACM WUWNET[C]. Montr al, Qu bec, Canada, 2007.33-40.
- [13] HSU C C, LAI K F, CHOU C F. ST-MAC: spatial-temporal MAC scheduling for underwater sensor networks[A]. Proc IEEE INFOCOM[C]. Rio de Janeiro, Brazil, 2009.1827-1835.
- [14] HONG L, HONG F, GUO Z W. A TDMA-based MAC protocol in underwater sensor networks[A]. The 4th International Conference on Wireless Communications, Networking, and Mobile Computing (WICOM'08)[C]. Dalian, China, 2008.1-4.
- [15] DOWNEY P. The Behavior of a Flooding Protocol in a Wireless Sensor Network[R]. Report for the Honours Programme of the School of Computer Science and Software Engineering, the University of Western Australia, 2003.
- [16] LIU YH, LIONEL M. Ni: a generalized probabilistic topology control for wireless sensor networks[A]. INFOCOM 2009[C]. Rio de Janeiro, Brazil, 2009.2776-2780.
- [17] LUBY M. A simple parallel algorithm for the maximal independent set problem[J]. SIAM Journal on Computing, 1986, 15(4):1036-1053.
- [18] TAN H X, SEAH W K G. Distributed CDMA-based MAC protocol for underwater sensor networks[A]. LCN[C]. Clontarf Castle, Dublin, Ireland, 2007.26-36.
- [19] SALVA-GARAU F, STOJANOVIC M. Multi-cluster protocol for ad hoc mobile underwater acoustic networks[A]. Proc IEEE Oceans Conf[C]. San Diego, CA, USA, 2003.91-98.

作者简介:



洪璐 (1984-), 男, 山东济宁人, 中国海洋大学博士生, 主要研究方向为传感器网络和容迟网络。

洪锋 (1977-), 男, 山东青岛人, 中国海洋大学副教授, 主要研究方向为传感器网络、对等计算和网格计算。

李正宝 (1981-), 男, 山东济宁人, 中国海洋大学博士生, 主要研究方向为传感器网络和容迟网络。

郭忠文 (1965-), 男, 山东青岛人, 中国海洋大学教授、博士生导师, 主要研究方向为传感器网络和海洋信息分布式处理技术。