

定向自组织网络中一种基于双令牌的 非同步邻居发现协议

张航^{1,2}, 李波¹, 闫中江¹, 李心茹¹, 杨懋¹, 秦建存³

(1.西北工业大学 电子信息学院, 陕西 西安 710129;
2.中国电子科技集团公司第五十四研究所, 河北 石家庄 050081;
3.中电网络通信集团有限公司, 河北 石家庄 050081)

摘要:针对基于单令牌(Token)的扇区天线邻居发现协议(SAND)发现速度慢的问题,提出了一种非同步下基于双Token的扇区天线邻居发现协议(D-SAND)。双Token分为主Token和从Token,主Token持有节点(MTH)探测邻居节点,通过波束扫描的形式发送Hone-In信息,建立基本连接后与各个邻居节点完成邻居发现过程,然后将该Token传递给下一跳主Token持有节点(nMTH),同时MTH将nMTH的位置信息发送给父Token持有节点(MTP),接着MTP节点进行从Token生成决策,查看是否存在与nMTH距离大于 $2R$ 的邻节点(其中 R 表示单跳节点之间的最大通信距离),如果存在则由父Token持有节点产生临时的从Token传递给该节点进行邻居发现。之后对双Token并行工作互不干扰的时间条件进行了理论分析,通过仿真表明,相比快速扇区天线邻居发现协议(Q-SAND),提出的D-SAND协议全网邻居发现时间缩短约10%。

关键词:定向自组织网络;非同步;邻居发现;扇区天线

中图分类号: TG925

文献标志码: A

文章编号: 1000-2758(2021)01-0062-09

定向自组织网络是采用定向波束进行通信的自组织网络。因为定向波束带来的窄波束、高增益特性,使定向自组织网络具备了单跳传输距离远、传输速率高、抗干扰能力强的优势。

定向自组织网络的挑战在于其采用了定向窄波束进行通信,各通信节点在进行有效通信之前,需要先进行基于窄波束的邻居发现,确定周围具体有几个邻居节点存在,并识别所在波束。如果各邻居节点在非同步条件下完成此项工作,则尤为困难。

现有研究按照是否存在统一的同步信息,可分为基于全网同步的发现算法和全网非同步邻居发现算法。非同步邻居发现算法不依赖统一的外同步源授时,具有重要的研究价值。

SAND协议是一种基于全网非同步下的邻居发

现算法,采用令牌(Token)传递的形式依次进行邻居发现,在之后Q-SAND等协议在此基础上进行了改进,其基础同样是基于Token传递完成的。该类型协议的问题是邻居发现时间长,Token需要被所有节点持有,并顺序进行邻居发现,邻居发现时间和全网节点数是呈线性正比关系。

本文提出了一种基于双Token的定向自组织非同步邻居发现协议,在发现过程中,以不互相干扰为原则,产生双Token,由双Token持有节点并行完成邻节点发现,从而缩短全网完成邻居发现的时间。

主要贡献点:

1) 针对非同步下的定向波束邻居发现问题,提出了一种双Token协议,相比单Token协议,全网邻节点发现效率得到提升。

收稿日期: 2020-06-02

基金项目: 国家自然科学基金(61771392, 61771390, 61871322 和 61501373)和航空科学基金(201955053002, 20185553035)资助
作者简介: 张航(1983—),西北工业大学博士研究生,主要从事无线自组织网络协议研究。

通信作者: 闫中江(1983—),西北工业大学副教授,主要从事无线网络组网协议设计研究。e-mail: zhjyan@nwpu.edu.cn

2) 进行了理论分析,分析了双 Token 干扰避免的条件,验证了双 Token 传递的可行性和有效性。

3) 完成了仿真,对双 Token 协议与单 Token 协议进行了对比分析,可以看到双 Token 较单 Token 协议节省了全网邻节点发现时间。

1 研究现状

按照节点的时钟状态,可以将邻居节点发现协议分为同步邻居节点发现和异步邻居节点发现。

文献 [1] 提出的 SAND (sectored antennas neighbor discovery) 协议是一种基于扇区天线的串行化邻居发现协议,不同节点在异步条件下完成邻居发现,为了避免节点之间的冲突干扰,采用 Token 传递机制协调不同节点的工作过程,只由当前的 Token 持有节点执行邻居节点发现。该协议中的节点均采用扇区可切换的定向天线,节点之间不需要进行全局时间同步,即可以完成全定向扇区天线的节点间邻居发现。是当前用于全定向波束邻居发现的一种先进有效的协议机制。

基于 SAND 协议,文献 [2] 中提出了一种 Q-SAND (quick SAND) 协议,对 SAND 协议进行了改进,以减少邻居发现时间。文献 [6] 提出了 DANDi 协议,在 Hello-Reply 阶段提出了一种时隙分配的优化方法。

以下重点对 SAND 协议和 Q-SAND 协议进行介绍。SAND 协议的协议操作主要分为 3 个阶段,包括:①吸引邻居的注意;②Hello-Reply 过程;③令牌传递或释放。具体的操作过程如下:

1) 吸引邻居的注意

该阶段又称为 Hone-In 阶段。Token 持有节点在该阶段依次向自己的每个扇区方向发送一系列的 Hone-In 消息,提醒收到该消息的节点即将进行邻居节点发现。Token 持有节点的邻居节点处在快速扫描模式,周期性遍历自己的每个扇区,在收到该 Hone-In 消息后,邻居节点将停止扫描,保持扇区指向 Token 持有节点,并启动计时器,在一定时间与 Token 持有节点同时进入邻居节点发现阶段。

2) Hello-Reply 过程

在 Hone-In 阶段结束后,所有收到 Hone-In 消息的潜在邻居节点切换到第一个扇区,Token 持有节点开始通过轮询邻居并在随后的若干时隙等待它们的响应来发现它的邻居。这种双向握手方式可以确

保发现节点之间的双向链接。在 Hello-Reply 过程中,Token 持有节点和它的潜在邻居节点都按照预先确定的顺序(顺时针或逆时针)依次遍历其 K 个扇区,确保对所有的扇区组合方式进行测试,以便于选出最好的通信链路。SAND 协议中一个 Token 持有节点的 Hello-Reply 过程需要尝试 K^2 种扇区组合方式。

3) 令牌传递或释放

令牌传递或释放又称为 Token Passing 或 Token Releasing。Token 持有节点完成前 2 个阶段后,会根据自身和周围邻居节点状态选择进行令牌传递。令牌传递的目的节点是下一跳未持有过 Token 的节点或者是自己的父节点。令牌释放则是针对当前拿到 Token 的节点已持有过 Token 即已经完成邻居发现的状态所执行的一种释放机制,以将 Token 交回初始节点。

当 Token 回到第一个 Token 持有节点,且网络中所有的节点都已持有过 Token 时,全网的邻居节点发现过程结束。

Q-SAND 协议引入了扇区方位,依靠方位信息,对于扇区个数 K 为偶数的节点,只需要尝试 K 种扇区组合;对于扇区个数 K 为奇数的节点,需要尝试 $2K$ 种扇区组合,即可以找到与邻居节点的最佳匹配扇区,与 SAND 协议中尝试 K^2 种扇区组合相比,Q-SAND 协议可以减少 Hello-Reply 阶段的通信开销提高节点发现效率。在其他阶段与 SAND 协议保持一致。因此 Q-SAND 协议的邻居发现总时间相比 SAND 协议得到提升。

2 系统模型

定向自组织网络的节点相互之间对等,在本协议中根据邻居发现时的角色定义了若干节点类型,分别是:

主 Token 持有节点(master token holder, MTH): 该节点当前持有主 Token,主动进行邻居节点发现,完成相应流程后,将主 Token 传递给下一个目标节点。下一跳的 MTH 称为 nMTH。

从 Token 持有节点(slave token holder, STH): 该节点当前持有从 Token,进行邻居节点发现,完成相应流程后,将从 Token 销毁。

主 Token 持有节点的父节点(master token parent, MTP): 当前主 Token 持有节点的上一级

Token 来源节点。该节点传递主 Token 给下一跳节点之后转换为 NN 节点,在接收到启动信息后才按照该角色进行工作。

邻居节点 (neighbor node, NN): 除 MTH、STH、MTP 之外的节点统称 NN 节点,如上所述 MTP 启动之前也为 NN 节点。

每个通信节点均由 K 个定向扇区天线组成,但需要注意实际的定向波束方向图不是理想的,旁瓣仍然有增益,但一般会显著小于主瓣增益。定义 R 为单跳最大通信距离。

通信节点依靠邻居发现可定位到周围的邻居节点及其所在的波束,在通信时,波束可以快速调整到需要通信的节点方向完成通信。

所有节点在存在方位信息的情况下,通信节点将指向正北方向的扇区定义为扇区 0。节点之间的扇区连线存在如下关系:

定义 K_a 为图 1 左扇区图的某一个扇区, K_b 为右扇区图与左扇区图连线上的扇区。

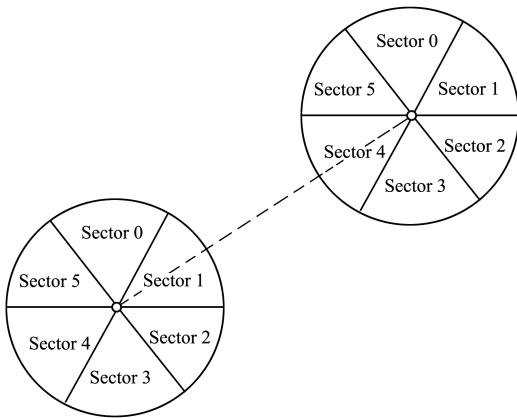


图 1 在 K 为偶数时,扇区之间的连线关系

K 为偶数时, $K_b = \text{mod}(K_a + K/2)$ 。如图 1 所示, K 为 6, 扇区 K_a 为 1, 所对应的 K_b 为 4。该连线关系反映了节点通信时,扇区波束的相互匹配关系。

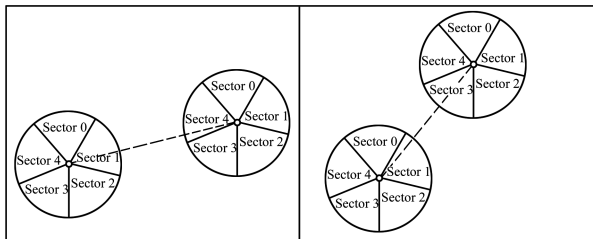


图 2 在 K 为奇数时,扇区之间的连线关系

如图 2 所示, K 为奇数时, $K_b = \text{mod}(K_a + (K + 1)/2)$ 或者 $K_b = \text{mod}(K_a + (K - 1)/2)$ 。出现 2 种对应关系主要是由节点对之间相对位置不同造成的。

在后续进行计算和仿真时,为简单起见假定扇区 K 为偶数。当 K 为奇数时,需要考虑图 2 中的 2 种情况。

3 协议描述

3.1 协议机制

D-SAND 协议的核心思想是提出了一个基于双 Token 的邻居发现协议,持有双 Token 的节点可并行工作,解决由于单 Token 依次传递带来的发现慢的问题。主 Token 持有节点 (MTH) 完成发现过程后,把主 Token 传递给 nMTH,同时将 nMTH 的位置信息告知 MTP,接着由 MTP 查看是否存在与 nMTH 距离大于 $2R$ 的邻节点 (R 为单跳节点最大通信距离),如果存在则由 MTP 产生临时的并发 Token,并传递给该节点进行邻居发现。

由于从 Token 是在主 Token 传递过程中临时产生的,接下来以 MTH 为主线介绍 D-SAND 协议整体工作流程。图 3 给出了 MTH 进行邻居发现时的基本帧结构,其中包括 Hone-In 阶段、Hello-Reply 阶段和主 Token 传递阶段。



图 3 MTH 的基本帧结构示意图

此外,与 MTH 类似,STH 也有 Hone-In 和 Hello-Reply 阶段,其工作过程是相同的,而 STH 完成 Hello-Reply 后进行从 Token 销毁。对于已经完成了邻居发现的节点,再次收到主 Token 时执行 Token 释放机制。

1) Hone-In 阶段

MTH 在每个扇区周期性的广播 h 个 Hone-In 消息,其周期为 $t_{\text{Hone-In}}$,以此来吸引 NN 节点的注意,并使邻节点与 MTH 保持一个初步的同步。为了保证在该扇区下的 NN 节点都能够收到消息,NN 节点扇区切换速度比 MTH 扇区切换快,即 NN 节点扇区切换时间 t_{switch} 应大于 $t_{\text{Hone-In}}$ 。 h 应满足

$$h \geq K t_{\text{switch}} / t_{\text{Hone-In}} \quad (1)$$

为了保证 NN 节点能够完整接收 Hone-In 消息,取 $t_{\text{switch}} = 2t_{\text{Hone-In}}$ 有

$$h \geq 2K \quad (2)$$

MTH 在 Hone-In 阶段总计发送 hK 个 Hone-In 消息 (K 为扇区数量) 即进入 Hello-Reply 阶段。Hone-In 消息中包含了启动下一阶段剩余的消息分组个数, NN 节点在收到 Hone-In 消息后将自己的工作扇区指向 MTH, 并依据剩余消息分组个数启动计数器倒计时, 倒计时计数到零后, 所有的 NN 节点进入 Hello-Reply 状态。

2) Hello-Reply 阶段

该阶段的主要目的是发现周围邻居节点与 MTH 所在扇区, 并形成可通信的邻居表。

MTH 在每个扇区周期性的广播 Hello 包, NN 节点基于与 MTH 的扇区对应关系进行接收, 并在收到 Hello 包后进行回复。回复过程 NN 节点分时隙进行, 时隙数量为 N_{slots} , 并随机选择一个时隙进行回复, 以减小多个节点同时回复带来的碰撞概率。完成一轮 Hello-Reply 之后, 启动下一轮, 轮数为 N_{rounds} , 以让所有的 NN 节点能够被 MTH 发现。

在该阶段结束时, NN 节点的扇区工作波束指向 MTH。

3) Token 传递阶段

Token 传递阶段 (Token Passing, TP) 主要包括: 主 Token 传递和从 Token 传递的过程。图 4 给出了 MTH、MTP、STH 和 NN 4 种节点类型的转换关系。

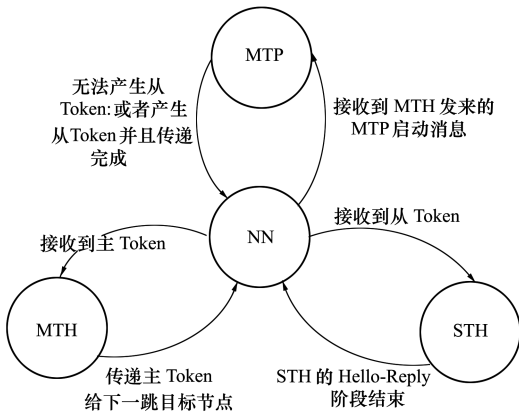


图 4 MTH、MTP、STH 和 NN 4 种节点类型的转换关系

(1) 主 Token 传递

MTH 在完成 Hello-Reply 阶段后形成邻居表, 该邻居表用 N_m 表示。该表记录的信息有: 邻居节点

号、邻节点所在扇区及其位置信息, 并标记出不同的邻居节点类型: 第一类为未完成邻居发现的节点, 为可能的 nMTH 集合; 第二类为已经完成邻居发现, 但未持有过主 Token 的节点 (成为过 STH, 并销毁了从 Token); 第三类为已完成邻居发现且持有过主 Token 的节点。

MTH 的主 Token 传递对象首先从邻居表中的第一类节点集合中选择, 作为 nMTH; 否则从第二类节点集合中选择; 如果前 2 类节点均不存在, 则将主 Token 传递给 MTH 的父节点。

MTH 在主 Token 传递阶段向所有扇区依次广播 GoToFastScan 消息。该消息的主要作用是使得 MTP 和下一跳 Token 持有节点之外的邻居节点回到快速搜索状态, 如果存在 nMTH 的情况下启动 MTP, 并将主 Token 传递给 nMTH, 如果不存在 nMTH 则不启动 MTP, 意味着不具备从 Token 产生的条件。

GoToFastScan 消息中包含: MTH 的邻居表、NN 节点快速扫描指示、MTP 启动指示、nMTH 的 ID 号和位置信息。如果需要启动 MTP, 则 GoToFastScan 消息先向 MTP 所在扇区发送, 收到该消息后, MTP 启动从 Token 产生决策机制, 同扇区的其他 NN 节点则进入快速扫描状态。之后 MTH 依次指向其他扇区继续发送消息。主 Token 下一跳节点也会在所在扇区收到 GoToFastScan 消息, MTH 在该扇区发送完 GoToFastScan 消息后, 接收返回的 ACK, 确保主 Token 正确传递了出去。每一条 GoToFastScan 消息中均含有一个倒计时器, 用于下一跳主 Token 持有节点在收到消息后, 计算还有多长时间启动工作。

(2) 从 Token 传递

MTP 本身维护了一张自有邻居表 (N_p), MTP 接收到 GoToFastScan 消息之后, 提取 MTH 的邻居表 (N_m)。

MTP 决策是否可以产生从 Token 过程如下:

将 N_m 和 N_p 进行对比, 检查在 N_p 集合的节点中是否存在一个非共有邻居集合, 即可能的 STH 的集合 N_s 。

$$N_s = \{n_i \mid n_i \in N_p \text{ 且 } n_i \notin N_m\} \quad (3)$$

定义 $D(N_s, \text{nMTH})$ 表示 N_s 中与 nMTH 的距离集合

$$d_{\text{max}} = \max(D(N_s, \text{nMTH})) \quad (4)$$

要求 $d_{\text{max}} > 2R$, 其中 R 为单跳最大通信距离。

如果存在 $d_{\text{max}} > 2R$ 的节点, 则将该距离对应的

N_s 中的节点作为 STH, 从 Token 由 MTP 生成传递给该从 Token 持有节点。

这样 STH 和 nMTH 一定是相距两跳之外的, 可以保证 STH 和 nMTH 之间不存在一个节点既是 STH 的邻居, 又是 nMTH 的邻居。

MTP 决策从 Token 可以产生之后, MTP 将扇区波束对准目标 STH, 并持续发送 mini-Hone-In 消息, mini-Hone-In 消息中包含 Token 节点信息, 只在一个扇区进行发送。目标 STH 收到消息之后, 扇区波束停止切换, 接收到从 Token 之后, 回复 ACK, 同时 MTP 收到 ACK 之后, 转换到快速扫描状态。

目标 STH 收到从 Token 之后, 开始完成 Hone-In 和 Hello-Reply 过程, 完成 Hello-Reply 之后, 则将从 Token 进行销毁, 不再进行传递。其邻节点自动进入到快速扫描状态。

4) Token 释放机制

Token 释放机制 (token releasing, TR) 是针对主 Token 在不同节点之间传递时, 经过的节点如果发现自身已经完成过邻居发现, 则将主 Token 释放给未持有过主 Token 的邻居节点, 如果不存在此节点, 则将此节点交回父节点。Token 释放采用 mini-Hone-In 机制在目标节点所在扇区持续发送消息。

依照上述传递和释放规则, 主 Token 可以在网络内所有节点遍历一次, 当主 Token 再次回到起始源节点, 并且源节点不存在未发现邻居的邻节点时, 则全网的邻居发现结束。

由上一小节描述可知, 从 Token 仅是临时产生的, 持有过从 Token 的节点会标识自己已经完成了邻居发现过程, 当主 Token 再次到达该节点, 不再进行邻居发现, 直接执行 Token 释放的过程。

3.2 协议复杂度分析

D-SAND 协议与 Q-SAND 协议在 Hone-In、Hello-Reply 阶段传递信息相同, 处理方式相同, 所以在这两个阶段协议复杂度相当; 在 Token 传递阶段, D-SAND 协议相比 Q-SAND 协议在计算复杂度上主要增加了 STH 的决策算法。该决策过程主要是计算 MTP 邻居节点与 nMTH 节点的距离, 并选出与 nMTH 距离最大的节点。该计算过程通过遍历算法即可以进行运算完成。设邻居表中最大邻居数为 n , 计算时间复杂度可以表示为 $O(n)$ 。存储资源与 n 也呈线性关系, 存储空间复杂度也为 $O(n)$ 。

决策算法在运算时, 计算负载实现了节点间的分担, 只增加了 MTP 节点一定的运算量, 不会带来

MTH 节点运算负荷的加重。

从全网发包数量上分析, 可以反映出协议在网络上运行的复杂度。如果不能产生从 Token 时, 发包数量和 Q-SAND 协议相同。在可以产生从 Token 时, MTP 将发送 mini-Hone-In 消息给 STH 节点, 该发包过程是 D-SAND 协议特有的, 但 STH 工作过程相比 MTH 又减少了 Token 传递阶段, 总体来说, 从 Token 发现过程相比主 Token 发现过程会带来发包略有增加, 而且仅在双 Token 并行工作时, 发包数量才会发生增加。

3.3 协议运行示例图

图 5 说明了双 Token 的形成和传递过程。具体说明见图 5 中注释。

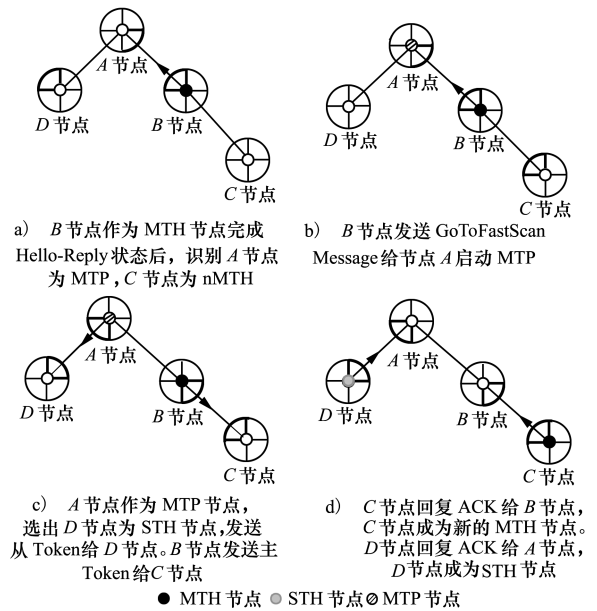


图 5 并发 Token 的形成和传递示例图

3.4 理论分析

1) 双 Token 并行工作时间分析

如图 6 所示, 从 Token 持有节点 STH 是与主 Token 持有节点 nMTH 并行工作的, 如果主 Token 在从 Token 工作期间已经被传递, 则主 Token 节点可能与 STH 存在共有邻居。因此, STH 邻居发现完成时间相比 nMTH 的发现完成时间应提前或同时结束。

令 $t_0, t_1, t_2, t_3, t_4, t_5$ 。 t_0 表示 MTH 完成 Hello-Reply 的时刻, t_1 表示完成 MTH 向 MTP 发送 GoToFastScan 的时刻, t_2 表示完成 MTH 向 nMTH 发送 GoToFastScan 的时刻, 即 nMTH 开始 Hone-In 的

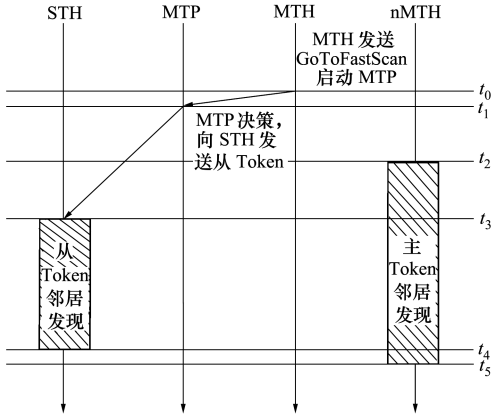


图 6 双 Token 并行工作时间

时刻, t_3 表示 STH 开始 Hone-In 的时刻, t_4 表示 STH 完成邻居发现的时刻, t_5 表示 nMTH 完成邻居发现的时刻。由上述分析可知, $t_5 \geq t_4$ 。不失一般性, 将 t_0 设为 0。

$$t_1 = t_{\text{GoToFastScan}} \quad (5)$$

$$t_2 = T_{TP} = (K - 1)t_{\text{GoToFastScan}} + t_{\text{TokenAck}} \quad (6)$$

$$t_3 = t_1 + T_{\text{mini-Hone-In}} = t_{\text{GoToFastScan}} + (h - 1)t_{\text{Hone-In}} + t_{\text{TokenAck}} \quad (7)$$

$$t_4 = t_3 + T_{HI} + T_{HR} = t_{\text{GoToFastScan}} + (h - 1)t_{\text{Hone-In}} + t_{\text{TokenAck}} + T_{HI} + T_{HR} \quad (8)$$

$$t_5 = t_2 + T_{HI} + T_{HR} + T_{TP} = T_{HI} + T_{HR} + 2T_{TP} \quad (9)$$

$$t_5 - t_4 = 2T_{TP} - (t_{\text{GoToFastScan}} + (h - 1)t_{\text{Hone-In}} + t_{\text{TokenAck}}) = 2(K - 1)t_{\text{GoToFastScan}} + 2t_{\text{TokenAck}} - (t_{\text{GoToFastScan}} + (h - 1)t_{\text{Hone-In}} + t_{\text{TokenAck}}) = (2K - 3)t_{\text{GoToFastScan}} + t_{\text{TokenAck}} - (h - 1)t_{\text{Hone-In}} \quad (10)$$

不失一般性, 设定

$$t_{\text{GoToFastScan}} = t_{\text{Hone-In}} = 1/2t_{\text{TokenAck}} = \tau \quad (11)$$

为了避免产生额外的干扰, 需要有

$$t_5 - t_4 \geq 0 \quad (12)$$

将(10)至(11)式代入(12)式可得

$$h \leq 2K \quad (13)$$

再根据(2)式, $h = 2K$ 时, 双 Token 可实现并行工作不产生额外的干扰。

2) 邻居发现总时间计算

Q-SAND 协议的时间为

$$T_{\text{QSAND}} = n(T_{HI} + T_{HR} + T_{TP}) + (n - 2)T_{TR} \quad (14)$$

Q-SAND 协议的总传递次数为 $2(n - 1)$, 其中 n

代表全网 n 个节点。其中每一个节点均要完成一次发现的全过程, 即进行 n 次的 Token 传递, 总传递次数为 Token 传递和 Token 释放次数的总和, 那么 Token 释放的次数为 $2(n - 1) - n = n - 2$ 。

其中 Hone-In 阶段的时长为

$$T_{HI} = hKt_{\text{Hone-In}} \quad (15)$$

Hello-Reply 阶段的时长为

$$T_{HR} = KN_{\text{rounds}}N_{\text{slots}}t_{\text{slots}} \quad (16)$$

Token 传递阶段的时长为

$$T_{TP} = (K - 1)t_{\text{GoToFastScan}} + t_{\text{TokenAck}} \quad (17)$$

Token 释放机制的时长为

$$T_{TR} = (h - 1)t_{\text{Hone-In}} + t_{\text{TokenAck}} \quad (18)$$

D-SAND 协议的时间为

$$T_{\text{DSAND}} = (n - n')(T_{HI} + T_{HR} + T_{TP}) + (n + n' - 2)T_{TR} \quad (19)$$

式中, n' 为并发 Token 的次数, 即有 n' 个节点的邻居发现是并行进行的, 在统计节点进行邻居发现的时间时不需要计算在总时间内。

相比 Q-SAND(14) 式, D-SAND 协议的主 Token 总传递次数仍为 $2(n - 1)$, 总传递次数为主 Token 传递和主 Token 释放次数的总和。 $n - n'$ 个节点要进行邻居发现全过程, 那么主 Token 释放的次数为 $2(n - 1) - (n - n') = n + n' - 2$, 相比 Q-SAND 的 Token 释放次数增加了 n' 次。

由于 $T_{HI} + T_{HR} + T_{TP}$ 远大于 T_{TR} , 可知 n' 越大, D-SAND 的总时间将越小。

4 仿 真

搭建 MATLAB 仿真平台, 对提出的双 Token 的 D-SAND 协议和单 Token 的 Q-SAND 协议进行了仿真比较。协议的仿真参数见表 1。

表 1 D-SAND 和 Q-SAND 协议在仿真中的参数

协议机制	参数	数值/ms
快速扫描模式	t_{switch}	3
Hone-In 阶段	$t_{\text{Hone-In}}$	1.5
	h	12
Hello-Reply 阶段	N_{slots}	5
	t_{slots}	1.5
Token 传递阶段	N_{rounds}	5
	$t_{\text{GoToFastScan}}$	1.5

仿真场景参数设置见表 2,节点在一定区域内随机分布,在仿真时要求随机分布的各个节点间需要形成可连通的网络拓扑,如果出现随机的孤立节点,则此次形成的网络拓扑不计入统计结果。每个节点都有一个独立的 ID,以 ID 为 1 的节点作为起始的 MTH。

表 2 网络场景仿真参数

网络场景设置	数值
节点分布区域/km	600×600
单跳最大通信距离/km	100
单节点波束个数	6
固定节点数下网络仿真次数	20

首先仿真了不同节点数下的从 Token 的并发次数和并发概率,如图 7 所示。节点个数一定的情况下,并发次数和网络拓扑是有关联的,为了说明趋势,按照表 2 的参数取 20 次平均。如图所示,网络中节点数增加后,并发概率在随之增大。

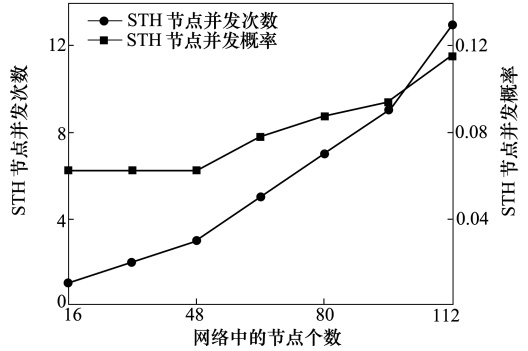


图 7 不同节点个数下的 STH 并发次数及并发概率

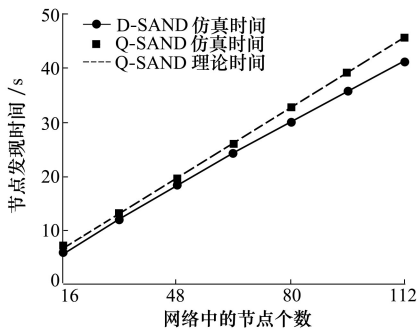


图 8 不同节点数量下的 D-SAND 和 Q-SAND 发现时间

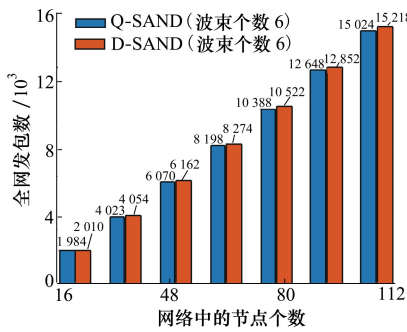


图 9 不同节点数量下的 D-SAND 和 Q-SAND 全网发包数对比

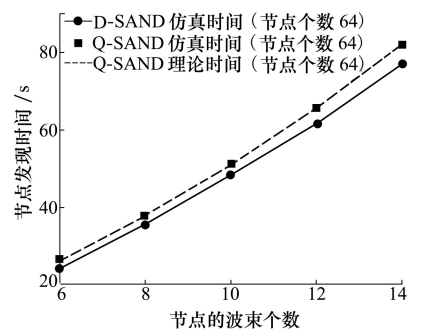


图 10 波束个数对 D-SAND 和 Q-SAND 发现时间的影响

图 8 仿真了全网的邻居发现时间,对 D-SAND 协议和 Q-SAND 协议进行了对比,Q-SAND 理论时间和仿真时间是保持一致的,验证了基础仿真的正确性。

Q-SAND 协议相比经典的 SAND 协议,改进目标是缩短全网邻居发现的时间。由图 8 可以看到本文提出的 D-SAND 协议相比 Q-SAND 协议更优,且随着节点数量的上升,D-SAND 相比 Q-SAND 优势越发明显,原因是节点数量增长,双 Token 并发的概率也在随之增加。从整体上来看,D-SAND 发现时间相比 Q-SAND 提升约 10%。

图 9 对 D-SAND 和 Q-SAND 2 种协议在不同网络节点数下的发包数量进行了仿真分析,由仿真可见,D-SAND 协议增加发包数量有限,以仿真中发包数量最多的 112 节点为例,D-SAND 协议增加的发

包数量比例为 1.2%。

最后对波束个数变化和节点发现时间的关系进行了仿真,如图 10 所示,固定节点个数为 64,D-SAND 和 Q-SAND 的发现时间随波束数量增加线性增长。

5 结 论

本文提出了一种异步条件下全定向扇区自组织网络的邻居发现 D-SAND 协议,提出并采用双 Token 工作机制,实现了不同通信节点的并行邻居发现。相比基于单 Token 的快速邻居发现协议 Q-SAND 协议进一步缩短了邻居发现时间,更好地适用于全网无统一时钟源条件下的定向波束自组织网络。

参考文献:

- [1] FELEMBAN E, MURAWSKI R, EKICI E, et al. SAND: sectorized-antenna neighbor discovery protocol for wireless networks[C] //2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Boston, MA, 2010: 1-9
- [2] GAMMARANO N, SCHANDY J, STEINFELD L. Q-SAND: a quick neighbor discovery protocol for wireless networks with sectorized antennas[C] //2018 9th Argentine Symposium and Conference on Embedded Systems, Cordoba, 2018: 19-24
- [3] MURAWSKI R, FELEMBAN E, EKICI E, et al. Neighbor discovery in wireless networks with sectorized antennas[J]. Ad Hoc Networks, 2012, 10(1): 1-18
- [4] YANG Annan, LI Bo, YAN Zhongjiang, et al. A bi-directional carrier sense collision avoidance neighbor discovery algorithm in directional wireless ad hoc sensor networks[J]. Sensors, 2019, 19(9): 2120-2138
- [5] YAN Zhongjiang, LI Bo, ZUO Xiaoya, et al. A heuristic clique based stdma scheduling algorithm for spatial concurrent transmission in mmwave networks[C] //2015 IEEE Wireless Communications and Networking Conference, New Orleans, LA, 2015: 1036-1041
- [6] GAMMARANO N, SCHANDY J, STEINFELD L. DANDi: dynamic asynchronous neighbor discovery protocol for directional antennas[C] //2018 VIII Brazilian Symposium on Computing Systems Engineering, Salvador, Brazil, 2018: 16-23
- [7] NUR F N, SHARMIN S, HABIB M A, et al. Collaborative neighbor discovery in directional wireless sensor networks: algorithm and analysis[J]. Eurasip Journal on Wireless Communications & Networking, 2017, 2017(1): 1-15
- [8] ZHANG Zhensheng. DTRA: directional transmission and reception algorithms in w lans with directional antennas for QOS support [J]. IEEE Network, 2005, 19(3): 27-32
- [9] LI Qianqian, LI Bo, YAN Zhongjiang, et al. Multi-channel multiple access protocol based on classified time slots for directional ad hoc networks[C] //2018 IEEE International Conference on Signal Processing, Communications and Computing, Qingdao, 2018: 1-6
- [10] ZHANG Hang, LI Bo, YAN Zhongjiang, et al. A pseudo random sequence based multichannel MAC protocol for directional ad hoc networks[C] //13th International Conference, Quality, Reliability, Security and Robustness in Heterogeneous Systems, Dalian, 2017: 172-182
- [11] SANTOSA R A, LEE B, YEO C K, et al, Distributed neighbor discovery in ad hoc networks using directional antennas[C] // The 6th IEEE International Conference on Computer and Information Technology, Seoul, 2006: 97-97
- [12] CAI Hao, WOLF T. On 2-way neighbor discovery in wireless networks with directional antennas[C] //2015 IEEE Conference on Computer Communications, Kowloon, 2015: 702-710
- [13] PEI Guangyu, ALBUQUERQUE M A, KIM J H, et al. A neighbor discovery protocol for directional antenna networks[C] //2005 IEEE Military Communications Conference on MILCOM, Atlantic City, NJ, 2005: 487-492
- [14] MIR Z H, JUNG W S, KO Y B. Continuous neighbor discovery protocol in wireless ad hoc networks with sectorized-antennas[C] //2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangju, 2015: 54-61
- [15] GAO Mingjin, SHEN Rujing, MU Linhong, et al. An anti-collision neighbor discovery protocol for multi-node discovery[C] //2019 11th International Conference on Wireless Communications and Signal Processing, Xi'an, China, 2019: 1-5

An asynchronous neighbor discovery protocol based on double tokens in directional ad hoc networks

ZHANG Hang^{1,2}, LI Bo¹, YAN Zhongjiang¹, LI Xinru¹, YANG Mao¹, QIN Jiancun³

(1.School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China;
2.The 54th Research Institute of CETC, Shijiazhuang 050081, China;
3.CETC Network & Communications Co. Ltd, Shijiazhuang 050081, China)

Abstract: Aiming at the problem of slow neighbor discovery speed of sector antenna neighbor discovery (SAND) protocol based single token, an asynchronous sector antenna neighbor discovery protocol based double tokens (D-SAND) is proposed. The double tokens include one master token and one slave token. The master token holder (MTH) starts its neighbor nodes discovery process by sending hone-in message through beam scanning, and ends this process when each neighbor node is discovered. Then, MTH passes the master token to the next hop master token holder (nMTH). At the same time, MTH sends the location information of the nMTH node to its parent master token holder (MTP) from which it obtains its token. MTP determines whether to generate a slave token according to that if there exists a neighbor node with a distance greater than $2R$ from nMTH, where R represents the communication distance. If there exists one such neighbor, MTP generates a slave token and passes it to that neighbor node, who can begin the neighbor discovering process. The time conditions are theoretically derived when these two tokens begin their neighbor discovering process respectively and without interference to each other. Simulation results show that compared to Q-SAND(quick-SAND), the proposed D-SAND protocol reduces the network-wide neighbor discovery time about by 10%.

Keywords: directional ad hoc network; asynchronization; neighbor discovery; sector antenna

引用格式:张航, 李波, 闫中江, 等. 定向自组织网络中一种基于双令牌的非同步邻居发现协议[J]. 西北工业大学学报, 2021, 39(1): 62-70

ZHANG Hang, LI Bo, YAN Zhongjiang, et al. An asynchronous neighbor discovery protocol based on double tokens in directional ad hoc networks[J]. *Journal of Northwestern Polytechnical University*, 2021, 39(1): 62-70 (in Chinese)