文章编号：1671-0576(2024)02-0040-05

基于表驱动的路由协议设计与 FPGA 实现

范明慧， 彭 澎， 董国英， 马景馨， 汪吕喜

（上海无线电设备研究所，上海 201109）

摘 要：针对 ad hoc 网络的拓扑可变性，设计了一种基于表驱动的网络路由协议。该路由协议中的每个节点维护一个拓扑集和一张到其他节点的路由表，当网络的拓扑结构变化时，各节点实时更新网络的拓扑项和路由表。同时在 FPGA 上实现了一种基于有限状态机(FSM)的路由表管理。经验证，该路由协议可以快速响应拓扑的变化，路由时延低，硬件资源占用少且功耗小。

关键词：ad hoc 网络；表驱动路由协议；FPGA；拓扑项

中图分类号：TN957.52 **文献标志码**：A

DOI：10.3969/j.issn.1671-0576.2024.02.007

Design and FPGA Implementation of Routing Protocol Based on Table Driven

FAN Minghui， *PENG Peng*， *DONG Guoying*， *MA Jingxin*， *WANG Lyuxi*

(Shanghai Radio Equipment Research Institute, Shanghai 201109, China)

Abstract: Aiming at the topology variability of ad hoc network, a network routing protocol based on table driven was designed. Each node in the routing protocol maintained a topological set and a routing table to other nodes. When the network topology structure changed, each node updated the topology item and routing table in real time. At the same time, the routing table management based on finite state machine (FSM) was implemented on FPGA. It is proved that the routing protocol can respond to the change of topology quickly, with low routing delay, less hardware resources occupation and less power consumption.

Key words: ad hoc network; table driven routing protocol; FPGA; topology item

0 引言

ad hoc 网络[1]具有无中心控制节点、路由多跳、拓扑动态等特点，可以适用于不能预设网络设施或需要快速自动组网的场合。设计高效、动态的路由协议成为规划无线自组网的一个挑战，路由协议必须能够适应节点移动所带来的网络拓扑结构迅速变化。

根据路由建立时机与数据发送的关系，

收稿日期：2023-10-13

作者简介：范明慧(1989—)，女，硕士，工程师，主要从事数据链无线自组网技术研究。

ad hoc 网络路由协议[2]通常可分为主动路由协议和按需路由协议。主动路由协议主要有目的序列的距离矢量路由(destination sequenced distance vector routing，DSDV)[3]、优化链路状态路由(optimized link state routing，OLSR)[4]、鱼眼状态路由(fisheye state routing，FSR)[5]等；按需路由协议主要有动态源路由(dynamic source routing，DSR)[6]、按需距离矢量路由(ad hoc on-demand distance vector routing，AODV)[7]等。按需路由是一种被动路由方式，其路由表是按需建立的，无需实时维护，一旦路由过期或者链路断开，路由表就失去作用了。

表驱动路由协议(table driven routing protocol)是一种主动路由协议，在路由过程中，每个节点分别在本地维护一个完整的路由表，当有数据需要发送时，直接根据目的节点地址查表获得下一跳需要发往的节点地址，并在每个中间节点依次查找，最终使数据以最短的路径抵达目的节点。

本文提出一种基于表驱动路由协议的路由表建立方法，并在 FPGA 硬件平台上进行性能验证。

1　路由协议概述

路由是指在网络中选择路径进行数据传输的过程。路由协议就是根据网络的状态信息，寻找最短的数据传输路径的网络协议。数据传输路径的选择会对网络性能产生巨大影响，因此路由协议对于网络来说十分重要。基于表驱动路由协议的路由，各节点在网络拓扑结构发生变化时发送更新信息，收到更新信息的节点更新本节点的路由表，以便及时准确地维护路由信息。不同的表驱动路由协议的区别在于拓扑更新信息在网络中传播的方式和需要存储的表的类型不同。

本文所提的表驱动路由协议的核心在于拓扑项的传输和路由项的缓存。处于网络拓扑中的节点通过无线数据传输进行通信，便可以得到本地链路(拓扑项)，同时也可以通过无线数据传输将本地拓扑项转发至邻居节点。当网络拓扑结构发生变化时，网络中可以建链的各节点间互相发送各自拓扑项，并根据本地链路拓扑和接收到的网内拓扑在本地维护一个由拓扑项组成的拓扑集。

节点在更新拓扑集时，根据路由算法计算出所有的路由项，所有的路由项组成一个路由表。在传输数据时，若节点与目的节点不能直接建链通信，则根据目的节点地址查找路由表即可选出通信路径。各节点根据路由算法分布式计算节点到网内目的节点的路径。

2　拓扑集的建立

2.1　拓扑集初始化

节点拓扑集包含多个拓扑项，每个拓扑项代表一条目的节点和与其建链节点的通信链路。网内各节点在本地维护一个拓扑集，该拓扑集由本地拓扑项和接收到的网内拓扑项组成。拓扑项包含目的节点地址及与目的节点建链成功节点地址等数据项，拓扑项如表 1 所示。

表 1　拓扑项

数据项	含义
T_Dest_Addr	目的节点地址
T_Last_Addr	与目的节点建链成功节点地址

当节点存在一条本地链路时，根据无线通信数据，可以得到链路节点的地址信息，即可建立一个本地链路对应的拓扑项(本地拓扑项)，完成拓扑集初始化。

2.2　拓扑集维护

在 ad hoc 网络中，移动节点能以任意速度和方式在网络中移动，同时通过无线通信信道发送功率的变化、无线信道间的干扰、环境因素的影响等信息，节点间通过无线通信形成的网络拓扑结构随时会发生变化。在网络拓扑发生变化时，需要对本地拓扑集进行维护。本地拓扑集的维护主要包括：本地拓扑项的更新与发送，本地链路消息的发送与接收。

在网络运行过程中，若网络拓扑结构发生变化，节点会将根据变化产生的本地链路消息发送给所有邻居节点，其中本地链路消息包括节点地址和对应的邻居节点地址。同时节点还需要对本地拓扑项进行更新：链路状态从连接变成断开，则删除对应的拓扑项；链路状态从断开变成连接，则增加对应的拓扑项。

节点接收到本地链路消息后，解析出消息中的节点地址，将其与拓扑集中所有拓扑项的 T_Last_Addr 进行比较，若拓扑集中不存在该节点地址的拓扑项，则将该条链路消息发送给所有邻居节点，同时建立新的拓扑项放入拓扑集，拓扑项中 T_Dest_Addr 为本地链路消息中的邻居节点地址、T_Last_Addr 为本地链路消息中的节点地址；若拓扑集中存在该节点地址的拓扑项，则更新拓扑项的 T_Dest_Addr。

2.3 拓扑集的 FPGA 实现

拓扑集的 FPGA 实现主要是在 FPGA 上实现 T_Dest_Addr 和 T_Last_Addr 两个数组寄存器的缓存，其流程图如图 1 所示。网络节点的数据解析模块从接收的无线通信数据中解析出帧格式和发送节点地址。将解析的发送节点地址缓存至节点地址 T_Last_Addr 对应的 T_Dest_Addr 中；根据解析的帧格式，若该帧为路由信息帧，则将信息帧中的节点地址和邻居节点地址分别缓存至 T_Last_Addr 和 T_Dest_Addr。

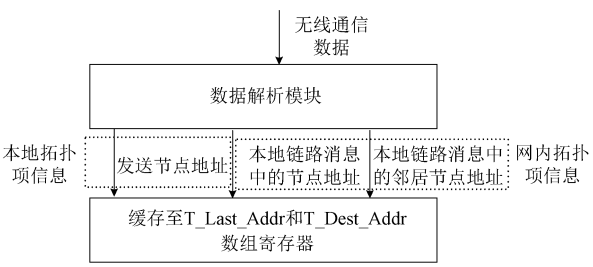

图 1 拓扑集实现的流程图

以 6 个节点的 ad hoc 网络为例描述拓扑集的建立过程，网络拓扑如图 2 所示。

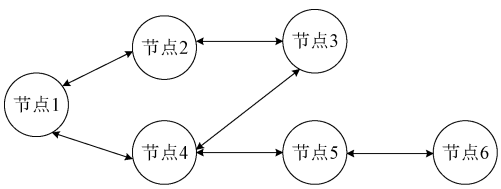

图 2 网络拓扑图

在网络运行过程中，网内各节点根据网络路由协议建链，节点 1 与其能直接建链的节点 2 和节点 4 完成无线通信。各节点对接收到的通信数据进行解析，可得到通信节点地址和对应的节点编号，完成本地拓扑项建立，并将对应数据缓存至 T_Dest_Addr 和 T_Last_Addr 两个数组寄存器。设 T_Dest_Addr[n]，T_Last_Addr[n]分别表示 T_Dest_Addr 和 T_Last_Addr 两个数组寄存器的第 n 个寄存器单元，其中 $n=0,1,\cdots,5$。具体缓存情况为：节点 1 地址缓存至 T_Last_Addr[0]；节点 2 地址、节点 4 地址缓存至 T_Dest_Addr[0]。

节点 2 将网络运行中得到的本地拓扑项和缓存的节点 3 拓扑项通过本地链路消息发送给节点 1，节点 1 将解析出的节点地址和邻居节点地址缓存至 T_Dest_Addr 和 T_Last_Addr 两个数组寄存器。具体缓存情况为：节点 2 地址缓存至 T_Last_Addr[1]；节点 1 地址、节点 3 地址缓存至 T_Dest_Addr[1]；节点 3 地址缓存至 T_Last_Addr[2]；节点 2 地址、节点 4 地址缓存至 T_Dest_Addr[2]。

同理，节点 4 将本地拓扑项和缓存的节点 5、节点 6 的拓扑项通过本地链路消息发送给节点 1，节点 1 将解析出的节点地址和邻居节点地址缓存至 T_Dest_Addr 和 T_Last_Addr 两个数组寄存器。具体缓存情况为：节点 4 地址缓存至 T_Last_Addr[3]；节点 1、节点 3、节点 5 的地址缓存至 T_Dest_Addr[3]；节点 5 地址缓存至 T_Last_Addr[4]；节点 4 地址、节点 6 地址缓存至 T_Dest_Addr[4]；节点 6 地址缓存至 T_Last_Addr[5]；节点 5 地址缓存至 T_Dest_Addr[5]。

通过上述的解析与缓存，节点 1 即可获得本地拓扑集，用于路由表的计算。

3 路由表计算

3.1 路由表算法

在 ad hoc 网络中，每个节点都可以作为路由器来维护本地路由表。节点不仅具备数据发送和接收功能，也具备路由的选择、存储和转发功能。节点路由表由路由项组成，而路由项又由多个数据项组成，路由项如表 2 所示。

表 2 路由项

数据项	含义
R_Dest_Addr	目的节点地址
R_Last_Addr	与目的节点建链的最后一跳节点地址
R_Next_Addr	与目的节点建链的下一跳节点地址
R_Dist_Num	到目的节点的跳数

路由表的建立过程是网络中每个节点路由的计算过程。通过寻找以本节点为根节点的生成树方式，计算本节点到拓扑集中所有其他节点的路径。具体算法如下。

步骤 1：为拓扑集中的每个 T_Last_Addr 等于本节点地址的拓扑项（本地链路）生成一个路由项，并将该拓扑项设为已计算。

路由项中数据项的取值方式为：路由项中的数据项 R_Last_Addr 等于拓扑项中本节点地址的 T_Last_Addr；数据项 R_Dest_Addr 和 R_Next_Addr 等于拓扑项中本节点地址对应的 T_Dest_Addr；R_Dist_Num 的初始值设为 1。

步骤 2：为新建立的路由项（原路由项）寻找 T_Last_Addr 等于该路由项 R_Dest_Addr 且 T_Dest_Addr 不等于本地链路的拓扑项（一个原路由项可对应多个拓扑项），为这些拓扑项的 T_Dest_Addr 建立新路由项，并将这些路由项设为已计算。

新路由项中数据项取值方式为：路由项的数据项 R_Dest_Addr 为拓扑项的 T_Dest_Addr；路由项的 R_Last_Addr 为拓扑项的 T_Last_Addr；路由项的 R_Next_Addr 为原路由项的 R_Next_Addr；路由项的 R_Dist_Num 为原路由项的 R_Dist_Num 加 1。

步骤 3：判断拓扑集中所有的拓扑项是否均已经计算，若是则计算结束，否则继续执行步骤 2。

3.2　路由表的 FPGA 实现

目前 ad hoc 网络路由协议的实现大多基于 NS2、GloMoSim、OPNET 等软件平台，但是在工程应用中，路由协议还需要在嵌入式硬件平台上实现。本文拟采用 FPGA 实现路由协议，以更快地建立和更新动态网络拓扑。

本文基于有限状态机（finite state machine，FSM）[8-9] 实现路由表管理，其流程图如图 3 所示。

当 ad hoc 网络的拓扑运行至图 2 所示状态时，拓扑集变化使能触发路由表生成功能。设 R_Dest_Addr[n]，R_Last_Addr[n]，R_Next_Addr[n] 分别表示 R_Dest_Addr，R_Last_Addr 和 R_Next_Addr 数组寄存器的第 n 个寄存器单元，$n=0,1,\cdots,5$。根据图 3，路由

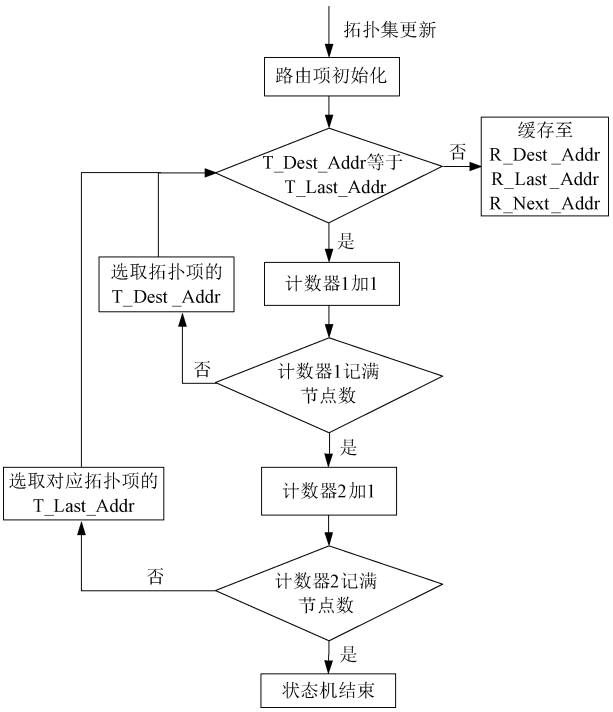

图 3　路由表管理流程图

表的实现步骤如下。

步骤 1：路由项初始化。设三个寄存器的初始值为 0；T_Last_Addr 等于本地节点地址即节点 1 地址时，根据路由算法将对应数据缓存至 R_Dest_Addr、R_Last_Addr、R_Next_Addr 数组寄存器。具体缓存情况为：

a）节点 1 地址对应的 T_Dest_Addr 为节点 2 地址和节点 4 地址，则将节点 2 地址缓存至 R_Dest_Addr[1]、节点 4 地址缓存至 R_Dest_Addr[3]；

b）节点 1 地址缓存至 R_Last_Addr[1]和 R_Last_Addr[3]；

c）节点 1 地址对应的 T_Dest_Addr 为节点 2 地址和节点 4 地址，则将节点 2 地址缓存至 R_Next_Addr[1]、节点 4 地址缓存至 R_Next_Addr[3]。

步骤 2：寻找 T_Last_Addr 等于该路由项的 R_Dest_Addr 且 T_Dest_Addr 不等于本平台地址的拓扑项，即令 T_Last_Addr 为节点 2 地址和节点 4 地址，建立新的路由项。具体缓存情况为：

a）节点 2 地址对应的 T_Dest_Addr 为节点 1 地址和节点 3 地址，由于节点 1 地址为本地节点地址，则将节点 3 地址缓存至

R_Dest_Addr[2]；

b）节点 2 地址缓存至 R_Last_Addr[2]；

c）原路由中节点 1 地址对应路由项的寄存单元 R_Next_Addr[2]缓存节点 2 地址，则将节点 2 地址缓存至 R_Next_Addr[2]；

d）节点 4 地址对应的 T_Dest_Addr 为节点 1 地址、节点 3 地址和节点 5 地址，由于节点 1 地址为本节点地址，节点 3 地址已经计算，则将节点 5 地址缓存至 R_Dest_Addr[4]；

e）节点 4 地址缓存至 R_Last_Addr[4]；

f）原路由中节点 1 地址对应路由项的寄存单元 R_Next_Addr[4]存储节点 4 地址，则将节点 4 地址缓存至 R_Next_Addr[4]。

根据图 2 可知，节点 1 到节点 3 的路径有两条，即节点 1—节点 2—节点 3、节点 1—节点 4—节点 3，由于节点 2—节点 3 路径先进行了计算，则节点 4—节点 3 的路径不再进行计算。

步骤 3：寻找 T_Last_Addr 等于该路由项的 R_Dest_Addr 且 T_Dest_Addr 不等于本平台地址的拓扑项，即令 T_Last_Addr 为节点 3 地址和节点 5 地址，建立新的路由项。具体缓存情况为：

a）节点 3 地址对应的 T_Dest_Addr 为节点 2 地址和节点 4 地址，由于节点 2 地址和节点 4 地址均已经计算，则不再进行计算；

b）节点 5 地址对应的 T_Dest_Addr 为节点 4 地址和节点 6 地址，由于节点 4 地址已经计算，则将节点 6 地址缓存至 R_Dest_Addr[5]；

c）节点 5 地址缓存至 R_Last_Addr[5]；

d）原路由中节点 5 地址对应路由项的寄存单元 R_Next_Addr[5]存储节点 4 地址，则将节点 4 地址缓存至 R_Next_Addr[5]。

根据上述步骤可得到节点 1 到其余节点的路由表，再根据目的节点地址，查询 R_Dest_Addr 数组对应的编号，通过 R_Next _Addr 数组得到下一跳的节点地址。

3.3 路由性能分析

本文提出的路由协议在 Xilinx 公司的 FPGA 芯片 XC7K410T 上实现。当节点拓扑项发生变化时，若网内节点个数为 N，则路由表生成的最长时间为 $2N^2+3$ 个时钟周期。以图 2 所示网络为例，当 N 为 6 时，在 100 MHz 时钟下，路由表生成的最长时间为 750 ns，芯片功耗为 0.006 W。芯片各部分资源使用情况如表 3 所示。

表 3 芯片资源使用情况

资源类别	数量	使用资源占比/%
触发器(Slice Flip Flops)	1 889	0.43
查找表(Slice LUT)	1 310	0.60
块存储器(Block RAM/FIFO)	0	0

可见，使用嵌入式硬件平台实现路由协议可以快速响应拓扑变化，路由时延低，硬件资源占用少且功耗小。

4 结论

本文提出了一种基于表驱动的 ad hoc 网络路由协议设计和 FPGA 硬件实现方法，该设计和实现方法的核心为拓扑项的实现和路由表有限状态机的设计与实现。经验证，使用 FPGA 平台实现路由协议，路由时延低，资源占用少且功耗小。

参考文献

[1] 王海涛. Ad Hoc 网络[J]. 电信技术，2005(8)：97-99.

[2] 曹建玲. 无线自组织网络路由协议及应用[M]. 北京：电子工业出版社，2015.

[3] MAHDIPOUR E，RAHMANI A M，AMINIAN E. Performance evaluation of destination sequenced distance vector (DSDV) routing protocol[C]// International Conference of Future Networks. Piscataway，NJ：IEEE Press，2009：186-190.

[4] DHURANDHER S K，OBAIDAT M S，GUPTA M. A reactive optimized link state routing protocol for mobile ad hoc networks [C]// IEEE International Conference on Electronics，Circuits and Systems. Piscataway，NJ：IEEE Press，2010：367-370.

[5] PEI G Y，GERLA M，CHEN T W. Fisheye state routing：a routing scheme for ad hoc wireless networks[C]// IEEE International Conference on Communications. Piscataway，NJ：IEEE Press，2000，1：70-74.

(下转第 49 页)

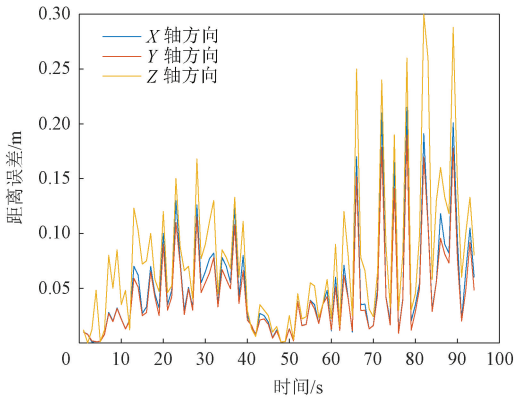

图 3　通用算法与简化算法的距离误差计算结果比较图

表 1　通用算法与简化算法计算量对比

算法	运算次数		
	乘法	除法	加法
通用算法	93	6	87
简化算法	14	6	9

综上所述，简化算法在计算误差与计算量上都能满足星上计算的需求。

3　结论

本文介绍了星上天线波束指向角的简化算法，该算法与星上天线波束指向角通用算法最大的区别在于转换坐标系的选择。通用算法采用 J2000 坐标系，该坐标系与时间密切相关，且坐标转换过程中涉及地球极移模型和章动岁差模型的选取问题，计算过程复杂；简化算法通过建立当前时刻与 CGCS2000 坐标系重合的惯性坐标系 I，避免了通用算法中的复杂模型的选取与计算消耗，实现了从 CGCS2000 坐标系到轨道坐标系 O 的快速转换。

参考文献

[1] 汪春霆，翟立君，徐晓帆. 天地一体化信息网络发展与展望[J]. 无线电通信技术，2020，36(2)：493-504.

[2] 阎鲁滨. 星载相控阵天线的技术现状及发展趋势[J]. 航天器工程，2012，21(3)：11-16.

[3] 李广宇. 天球参考系变换及其应用[M]. 北京：科学出版社，2010.

[4] 张捍卫，郑勇，马高峰. GCRS 与 ITRS 之间的坐标转换研究[J]. 大地测量与地球动力学，2011，31(1)：63-67.

[5] 高龙超. 定向通信系统天线波束指向误差分析[J]. 航空电子技术，2021，52(4)：56-61.

[6] 张毅，肖龙旭，王顺宏. 弹道导弹弹道学[M]. 长沙：国防科技大学出版社，2005：5-30.

[7] 郝辉，李雪瑞，舒健生，等. 导弹常用空间直角坐标系间转换方法[J]. 四川兵工学报，2013，34(2)：18-20.

[8] 严恭敏，戴晨杰，陈若彤. 地球自转模型误差对高精度惯导系统定位精度的影响分析[J]. 中国惯性技术学报，2022，30(2)：153-158.

(上接第 28 页)

[6] 刘荔娜. 基于 LDRA TBrun 软件集成测试的研究[J]. 电脑与电信，2020(4)：64-67.

[7] 黄晨，董燕，于倩，等. 基于目标码的测试覆盖不可达分析方法[J]. 测控技术，2017，36(1)：100-103，107.

[8] 刘高军，钱程. 基于边界值分析法的 CNONIX 标准测试用例生成方法研究[J]. 工业技术创新，2015，2(2)：228-233.

(上接第 44 页)

[6] JOHNSON D，MALTZ D. Dynamic source routing in ad hoc wireless networks[J]. Mobile Computing，1996，335(1)：153-181.

[7] PERKINS C E，BELDING-ROYER E M，DAS S R. Ad hoc on-demand distance vector (AODV) routing[Z]. Reston：The Internet Society，2003.

[8] 陈勇. 有限状态机的建模与优化设计[J]. 重庆工学院学报(自然科学版)，2007，21(5)：55-58.

[9] 李森. 面向 FPGA 的 DSR 路由表项设计与实现方法[J]. 电子技术应用，2018，44(12)：89-92.