软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(8):1729−1742 [doi: 10.13328/j.cnki.jos.004667]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

面向 IPv6 物联子网的轻量级树型转发模型*

肖 融[1,2], 陈文龙[3], 孙 波[1]

[1](北京师范大学 信息科学与技术学院,北京 100875)
[2](北京师范大学 教育学部,北京 100875)
[3](首都师范大学 信息工程学院,北京 100048)
通讯作者: 陈文龙, E-mail: wenlongchen@sina.com

摘 要: 在 IPv6 物联网中,RPL 路由模型已得到广泛的认可.然而对于规模较大的多跳网络结构,RPL 面临着部分转发节点路由容量较大的问题.而且物联子网中扁平化的地址结构使得这一问题更为突出.设计了支持 IPv6 地址自动分配的轻量级树型转发模型 TFAD(tree forwarding model with address automatically distributed),将物联子网中的节点构造成一棵层次转发树,树节点的 IPv6 地址在子树范围内高度聚合.各节点只需存储与其子节点数相当的转发项,即可完成 TFAD 模型的数据转发.此外,设计了 TFAD 模型的备份父节点机制,当网络出现故障时能够以子树为单位进行网络拓扑重构,实现物联子网的快速路由恢复.实验验证了 TFAD 模型的高效路由存储性能以及快速的路由学习能力和故障后路由恢复能力.

关键词: 物联网;转发树;子树地址范围;层次比特位
中图法分类号: TP393

中文引用格式: 肖融,陈文龙,孙波.面向 IPv6 物联子网的轻量级树型转发模型.软件学报,2014,25(8):1729−1742. http://www.jos.org.cn/1000-9825/4667.htm
英文引用格式: Xiao R, Chen WL, Sun B. Light weight and tree-based forwarding model in IPv6 IoT subnet. Ruan Jian Xue Bao/Journal of Software, 2014,25(8):1729−1742 (in Chinese). http://www.jos.org.cn/1000-9825/4667.htm

Light Weight and Tree-Based Forwarding Model in IPv6 IoT Subnet

XIAO Rong[1,2], CHEN Wen-Long[3], SUN Bo[1]

[1](College of Information Science and Technology, Beijing Normal University, Beijing 100875, China)
[2](Faculty of Education, Beijing Normal University, Beijing 100875, China)
[3](College of Information Engineering, Capital Normal University, Beijing 100048, China)
Corresponding author: CHEN Wen-Long, E-mail: wenlongchen@sina.com

Abstract: RPL has received universal acceptance in IPv6 routing of the Internet of things (IoT). However, for large-scale multi-hops networks, the RPL routing model is faced with the problem that some IoT nodes heavily consume routing table storage. Besides, the flattened address architecture in IoT subnet makes this problem more prominent. In this paper, TFAD (tree forwarding model with address automatically distributed), a light-weight and tree-based forwarding model, is proposed to support automatic IPv6 address assignment. TFAD constructs a forwarding-level-tree for all the IoT nodes that make the IPv6 addresses of nodes aggregate highly in each sub-tree. In TFAD, each node only needs to maintain a few forwarding entries, the number of which is equivalent to the number of its direct son-nodes. Moreover, the backup mechanism of parent node in TFAD is designed. This mechanism supports the network topology reconstitution based on the whole sub-tree, achieving fast route-recovery from network failure. The experiments based on real sensor nodes prove that TFAD model possesses not only high performance on routing table storage but also rapidity on routing table learning and routing recovery from failure.

Key words: Internet of things; forwarding tree; address range of sub-tree; lay bit

* 基金项目: 国家自然科学基金(61272446, 61373161); 国家科技重大专项(2012ZX03005001-001)
收稿时间: 2014-01-06; 修改时间: 2014-04-29; 定稿时间: 2014-05-29

© 中国科学院软件研究所 http://www.jos.org.cn

物联网被视为现有互联网的延伸和扩展[1],它将在环境监测、健康医疗、智能家居等各个领域被广泛实施.物联网节点的海量部署及其与传统互联网系统直连直通的诉求,都希望物联网能有一种普适的、开放的节点标识机制.显然,IP 技术是最佳选择.由于互联网 IPv4 地址即将耗尽[2],学术界普遍认为:物联网必定与 IPv6 技术相结合,通过 IPv6 地址来标识各种传感器节点[3,4].IETF 已成立多个工作组研究 IPv6 物联网技术,其中,ROLL 工作组专门针对物联网路由问题设计了 RPL 协议[5].然而在规模较大的多跳网络结构中,RPL 协议会给部分路由节点带来较大的转发项存储及处理开销.另外,物联网节点操作系统普遍使用 SLAAC(stateless address auto configuration)地址分配机制[6],易形成扁平化、难聚合的子网地址结构,对网络路由性能的提升造成障碍.

针对 IPv6 物联网的路由转发问题,本文提出了一种支持地址自动分配的轻量级树型转发模型(tree forwarding model with address automatically distributed,简称 TFAD).TFAD 模型主要适用于网络拓扑相对稳定的物联网环境,它以物联子网网关节点为树根,将所有节点构建成一棵层次转发树.在转发树中,每个节点都按照指定规则从其父节点继承一块 IPv6 地址块,并将地址分配给它的子节点.这样,转发树中任何一棵子树都拥有一个高度聚合的子树地址范围.当然,如果某个节点获得了一个子树地址范围却又没下连足够多的子孙节点,就会出现地址空间的浪费.尽管 TFAD 模型会消耗略多的地址空间,但这种 IPv6 层次地址结构非常规整,转发树中的节点只需存储极少的转发条目,即可实现物联子网节点的双向转发通信.而且,节点的转发项存储容量只与其直连子节点数相关.

TFAD 模型具有如下特点:1) 路由协议交互简单,节点间父子关系构建后,根据父节点的所辖前缀完成子节点的 IP 配置,与上下游其他节点无关,能够确保较快的路由收敛能力;2) 节点转发项数量较少,每个节点的转发项数与直连子节点数相当,极大地减少了树节点的内存消耗,具有很好的路由可扩展性;3) 快速路由恢复,以子树为单位进行拓扑重构以及事先选择备份父节点,可实现链路故障后物联子网路由体系的快速稳定.

本文第 1 节介绍相关研究工作.第 2 节介绍 TFAD 模型的核心思想及相关定义.第 3 节介绍 TFAD 转发树的构建方法.第 4 节介绍节点的转发信息存储方法及报文转发处理流程.第 5 节对 TFAD 模型进行组网实验及性能分析.第 6 节总结全文.

1 相关研究

IPv6 物联网结构有如下主要特点:1) 资源受限.不仅是计算机系统的 3 大传统资源(处理器、存储、带宽)在物联网环境中大幅受限,无源供电环境导致节点的供电能力也极为有限[7].因此,轻量级协议机制成为物联网体系的研究重点.2) 节点的多重角色.不同于传统互联网,物联网的节点还需具有自路由能力[8],为其他节点提供数据转发等功能.3) 新型链路层协议.IPv6 物联网的链路层主要基于 802.15.4 低功耗协议[9]实现,它使用 8 个字节的链路层 MAC 地址.802.15.4 协议具有复杂度低、成本小、功耗小等特点.此外,IETF 专门成立了 6LoWPAN 工作组[10],为 IEEE 802.15.4 设计适配层使其支持 IPv6.所以,IPv6 物联网转发模型面临着新的挑战,本文的轻量级转发模型也是基于上述特点展开研究.

互联网中任何一套路由转发体系都与其网络节点标识机制密不可分,节点标识是网络路由转发的前提和基础.传统传感器网络部署时,常在所辖范围内自主设置节点标识,文献[11,12]就研究了树型 ID 标识分配及标识压缩方法.自主标识机制简单易实施,但却无法支持传感器网络节点与互联网系统的端到端通信.文献[3]从多个角度阐述了物联网节点使用 IP 地址作为节点标识的优势,包括良好的开放性和可扩展性、支持轻量级协议栈实现、支持多样的上层应用、已得到广泛实施、支持端到端通信等等.文献[13]也强调物联网设计与实现的统一性,IP 技术则可为该目标提供统一的标识支撑.可见,物联网必将与 IP 技术相结合.同时,由于互联网的大规模发展导致 IPv4 地址即将耗尽,互联网本身也正从 IPv4 向 IPv6 过渡.所以,物联网环境使用 IPv6 地址作为节点标识是必然趋势.

现有 IPv6 网络主要通过 ND 协议或 DHCPv6 协议实现端系统的地址自动配置.IPv6 设计了邻居发现协议(ND)进行全局地址的无状态自动配置,由节点的第 1 跳路由器通过路由器公告(RA)消息向网络节点发布 64 位前缀,各节点将其与 EUI-64 链路层地址结合生成全局 IPv6 地址[6].另外,IPv6 网络还可利用 DHCPv6 机制实施

© 中国科学院软件研究所 http://www.jos.org.cn

节点的有状态全局地址分配[14].通过 DHCP 和 ND 机制进行地址分配,逻辑上都要求端系统与地址分配设备(网关路由器或 DHCP 服务器)直连通信.并且,子网中节点的地址是扁平结构,没有任何层次性,非常不利于物联子网中多跳寻径.可见,DHCP 和 ND 协议都无法直接施用于现有物联网典型树型结构.针对上述问题,本文面向物联网结构设计了特定的地址分配机制,为轻量级转发模型提供支撑.

作为无线传感器网络的最核心技术,路由及转发模型一直是学术研究的重要关注点.文献[15,16]对无线传感器网络路由协议进行了归纳和总结,包括层次结构和数据中心等多种路由转发模型.

层次结构路由是无线传感器网络中广泛研究的路由技术[17].它将节点分为不同层次的若干组,每组只有组头节点代表一个组与外部进行路由交互,并严格遵循层次隶属关系[18].层次路由的典型代表是分簇路由[19].它根据不同方法将传感器节点分成相应的簇,并在各节点簇中选举一个簇头节点,通过节点的多跳通信和分层路由来节约能耗,从而实现较大规模无线传感器网络节点的分级通信.TEEN(threshold sensitive energy efficient sensor network protocol)[20]还通过簇头在簇内设置阈值,减少传感信息的发送次数.然而,分簇路由需要节点支持专门的簇头选举协议.而且,如果无法解决节点标识的聚合问题,簇间路由协议交互及节点存储仍较为复杂.

数据中心路由协议围绕着传感信息到 sink 节点的传递方式进行描述.SPIN(sensor protocols for information via negotiation)[21]定义了元数据,对传感信息进行描述,并在节点间协商元数据,减少冗余信息的传递.LEACH (low-energy adaptive clustering hierarchy)[22]很好地与分簇路由思想结合在一起,传感节点向着 sink 节点发送信息,簇头节点会依据特定规则进行数据的聚集,从而减少数据传递并降低网络能耗.在近期研究中,文献[23]还介绍了一种新型的基于移动人员的物联网数据采集方法.它依据移动人员所在位置,不断构建或更新数据转发树实施传感信息采集.当然,它只适用一些特定的物联网场景.汇聚树协议(collection tree protocol,简称 CTP)[24,25]面向数据汇报场景提供了一种传感节点向根节点的数据通信方法,它与节点地址无关,每个节点根据路径链路质量选择最优父节点.节点发现父节点失效后,会按照同样的方法重新选择新的父节点.而且,CTP 只关注上行消息,本文 TFAD 模型可关注双向通信,并由于节点地址在子树范围高度聚合,TFAD 还可支持链路故障时以子树为单位切换父节点.AODV(ad hoc on-demand distance vector routing)是无线网络领域较为经典的一种路由协议[26],它专为无线自组织网络设计,也可归为数据中心路由协议.AODV 协议由源节点以组播形式发送路由请求,接收到请求的节点继续转发该组播路由请求,一直洪泛到目标节点或是某个已知去往目标节点路由的节点.AODV 协议还通过定期广播 Hello 报文来维护路由.AODV 协议的交互开销很大,并不适合大多数物联网场景.

目前,针对 IPv6 无线传感器网络,最为业界认可的路由协议当属 IETF 的 ROLL 工作组为低功耗有损网络设计的 RPL 路由协议[5].现有典型物联网操作系统 Contiki[27]和 TinyOS[28]都能很好地支持 RPL 协议.该协议将物联网节点构建成树状有向无环结构,并简单地将数据传递分为上行和下行两个方向.其中,下行数据传递支持存储模式和非存储模式.在存储模式中,每个节点会存储以其为根的子树所辖节点的路由,并以此完成路由转发.在非存储模式中,sink 节点之外的其他节点不存储路由信息,下行数据通过源路由 IP 选项头指定转发路径.由于 IP 源路由并不受业界广为接纳,现有实现主要通过存储模式进行下行数据转发.此外,该路由协议基于邻居发现机制(ND)实施网络节点的 IPv6 地址配置,网络地址仍然是扁平状结构,这使得转发树中 sink 节点或靠近树根的节点要存储大量的路由信息,导致了较低的存储性能.本文的 TFAD 模型在这方面有了明显的改善.

上述研究成果为本文工作提供了重要参考.然而,现有路由转发模型都没有很好地与节点标识研究融合,也没有解决节点标识的聚合问题.在 IPv6 物联网中,不可聚合或没有层次关系的地址结构将导致网络需要交互并存储大量的路由信息.本文设计了基于子树可聚合的地址结构,进而构建轻量级树状路由转发模型.

2 支持地址自动分配的树型转发模型

网络节点的标识体系总是与整个网络的路由转发体系密不可分,一个合理、高效的地址分配体系有利于高效转发体系的构建.我们认为,IPv6 物联子网的地址管理及路由转发应满足如下基本目标:低负载的路由交互、轻量级转发引擎、支持网络节点 IPv6 地址的简单配置管理.

© 中国科学院软件研究所 http://www.jos.org.cn

面向 IPv6 物联子网的特定网络环境,并针对上述地址管理和转发机制的基本目标,本文设计了支持地址自动分配的树型转发模型 TFAD.该模型中,所有物联网节点以网关节点为根,构造成一棵层次转发树.树型转发模型中的节点都具有明确的层数属性,用以描述该节点到根节点的跳数距离.例如,根节点的层数为 0,根节点的直连子节点的层数为 1,其他节点的层数是其父节点的层数加 1.每一节点都按照指定规则从其父节点继承一块 IPv6 地址块,并根据同样规则分配给其所辖子节点,节点地址都可在节点所属子树范围内高度聚合.由于节点地址在层次转发树中高度聚合,父、子节点间进行简单的协议交互即可完成转发信息的学习,每个节点只需存储少量的转发条目即可完成数据转发.

定义 1(层次比特位(lay bit,简称 LB)). 层次比特位是 128 位 IPv6 地址中除去网络位之外的若干连续 bit 位,TFAD 模型中,利用层次比特位标识树型结构中同一层的不同兄弟节点及其所辖子树.

为不失一般性,本文以特定 64 位子网地址范围进行分析,即假设整个物联子网分得一个“2500::/64”IPv6 网络地址段,每层的层数比特位都为 16 位,共有 5 层.此时,层次比特位是 IPv6 地址中第 65 位~第 128 位间的某 16 个连续 bit 位.图 1 就是 TFAD 模型在该物联子网构建的树型网络拓扑,节点地址中,带下划线部分就是该节点的层次比特位取值.例如,节点 2~节点 4 是同层兄弟节点,拥有相同的 16 位层次比特位(第 65 位~第 80 位),它们通过层次比特位的不同取值来标识各自的节点子树.同理,节点 5、节点 6 也拥有相同位置的 16 位层次比特位(第 81 位~第 96 位),并且取值各不相同.此外,图 1 中各节点标有对应的 IP 前缀,表示该节点所辖子树的地址范围.例如,节点 3 所辖子树的地址范围是 2500::0002:0:0:0/80,节点 5~节点 10 的地址都属于该范围.图 1 中,不同层节点管辖不同大小的地址范围,对应地址范围的掩码长度分别为 64,80,96,112,128,其中,最底层叶子节点管辖地址范围的掩码长度为 128,即只管辖自身 1 个节点.需要说明的是,TFAD 模型实际部署时,可根据具体情况设置不同的物联子网地址范围、层数和各层的层次比特位数.为便于描述,本文有如表 1 所示的符号定义.

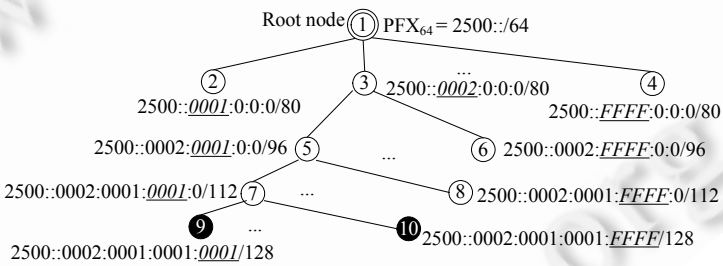


Fig.1 An example of network topology for TFAD

图 1 TFAD 网络拓扑示例

Table 1 Symbols definition of TFAD model

表 1 TFAD 模型符号定义

符号	描述
PFX_{64}	物联子网网络前缀,IPv6 子网通常都以 64 位子网掩码分配地址,本文针对该主流环境进行分析.当然,对于任意掩码位数的 IPv6 子网,TFAD 模型也可适用.
NUM_{son}	单个节点可接入的最大子节点数,可根据实施环境灵活设定
$LB[j].n$	第 j 层节点的层次比特位位数.通常,转发树中各层“层次比特位位数”取值范围不超过 16.
$LB[j].b$	第 j 层节点的层次比特位在 128 位 IPv6 地址中的起始位置
$N_i.lay$	节点 N_i 的层数
$N_i.rem$	节点 N_i 的剩余可连接数,表示节点 N_i 还可接入的子节点数,初始值为 NUM_{son}
$N_i.val$	节点 N_i 的层次比特位的取值,由其父节点分配,取值范围为 $1\sim 2^{LB[N_i.lay].n}-1$. 图 1 中第 1 层的层次比特位取值范围就是 $1\sim 2^{16}-1$.
M_i	节点 N_i 的链路层 MAC 地址
$L_{\max}$	转发树所支持的最大层次数.当某节点的层次数为 $L_{\max}$ 时,就不能再连接子节点.

定义 2(底层节点). 在层次转发树中,层数为 $L_{\max}$ 的节点称为底层节点,否则称为非底层节点.

© 中国科学院软件研究所 http://www.jos.org.cn

一个节点只要它的层数小于 $L_{\max}$,即使没有子节点,仍被称为非底层节点.显然,底层节点不能再连接子节点,而非底层节点可以下连子节点.图 1 中,实心节点为底层节点,空心节点为非底层节点.例如,节点 2 虽然并未下连子节点,但其层数小于 $L_{\max}$,属于非底层节点,它可以下连子节点.

定义 3(叶子节点). 在层次转发树中,没有下连子节点的节点为叶子节点.

叶子节点与底层节点不是同一概念.图 1 中,节点 4 和节点 6 都是叶子节点,但却不是底层节点.

定义 4(子树地址范围(address range of sub-tree,简称 ARST)). 在 TFAD 模型中,任一树节点 N_i 都拥有一个特定的子树地址范围 $ARST[i]$,以 N_i 为根的子树中所有节点的 IP 地址都属于 $ARST[i]$.子树地址范围是一个 IPv6 网段,具体描述为 $ARST[i].IP/ARST[i].len$.例如,图 1 中节点 3 的子树地址范围是 2500::0002:0:0:0/80.另外,底层节点的子树地址范围只辖一个 IPv6 地址,即节点自身的地址,此时 $ARST[i].len$=128.

定义 5(子树地址范围派生 Inherit). 父节点根据自身地址范围 $ARST[f]$及子节点层次比特位取值 $N_s.val$,派生出子节点的子树地址范围 $ARST[s]$,记作 $ARST[s]=Inherit(ARST[f])$.有:$ARST[s].len=ARST[f].len+LB[N_s.lay].n$,并且 $ARST[s].IP$ 由 $ARST[f].IP$ 在对应层次比特位上补上子节点 N_s 的层次比特位取值 $N_s.val$ 而得.

TFAD 模型中的节点都有对应的子树地址范围,是其父节点的地址范围与自身的层次比特位取值结合而得.以图 1 为例,节点 N_5 的子树地址范围为“2500::0002:0001:0:0/96”,是由父节点 N_3 的子树地址范围“2500::0002:0:0:0/80”与 N_5 自己的层次比特位“::<u>0001</u>:0:0”(下划线所指即为层次比特位)结合而得.

TFAD 模型以物联子网网关为根节点,构造一棵层次转发树.树中节点根据其所在的层次位置,拥有特定的 IPv6 子树地址范围.由于树节点 IPv6 地址具有严格的层次隶属关系,TFAD 模型各节点能够以极少的转发信息完成物联子网的数据转发过程.基本思想描述如下:

1) 节点 IPv6 地址自动配置.

物联子网中所有节点的子网掩码都为 64 位,节点 IPv6 地址根据其子树地址范围生成.如图 1 中,节点 3 的子树地址范围为“2500::0002:0:0:0/80”,则节点 3 的地址为“2500::0002:0:0:0/64”.由于 IP 地址在主机部分不能全为 0 和全为 1,所以对转发树部分节点的地址配置需要有特殊考虑.树根节点地址设置为子树地址范围加 1,图 1 中节点 1 的 IPv6 地址为 2500::1/64.另外,IP 地址主机位全为 1 的节点不允许接入层次转发树,如图 1 网络拓扑中,不能接入 1 个 IP 地址为 2500:FFFF:FFFF:FFFF:FFFF/64 的底层节点.

2) 数据转发.

物联子网节点的数据处理可分为上行转发过程(向着树根节点方向)和下行转发过程(向着树叶节点方向),这两个转发过程可覆盖物联子网内部节点间通信、物联子网节点与互联网节点间的通信.节点 N_i 收到一个待转发的数据报,若报文目的 IPv6 地址属于节点 N_i 所辖的子树地址范围 $ARST[i]$,则将向其子树中某个直连子节点转发,即为下行转发;否则,向其父节点转发,则为上行转发.

3) 路由学习与转发信息存储.

转发树中,任意父子节点间都有明确的子树地址范围隶属关系,构成天然的寻径体系.所以,TFAD 模型路由协议交互简单,建立好单跳父子节点关系并分配好 IP 地址,转发路径也就随之建立.网络节点的转发信息包括各直连子节点的层次比特位和链路层 MAC 地址的对应项,还包括父节点的链路层 MAC 地址.图 1 中,节点 3 的转发信息包括层次比特位“0001”对应链路地址为 M_5、层次比特位“FFFF”对应链路地址为 M_6 等等.此外,节点 3 还需存储父节点的链路地址 M_1.

定义 6(路由恢复时间). 在 TFAD 模型中,无线网络拓扑变化会导致若干节点到根节点的短暂数据传输中断,从传输中断到传输恢复正常的时间段称为路由恢复时间.路由恢复时间与特定网络结构、特定网络状态相关.需要强调,节点间的正常传输包括双向可达.

TFAD 模型支持以子树为单位进行拓扑重构,从而实现网络故障后的快速路由恢复机制.在层次转发树中,每一节点都会预先设置备份父节点,备份父节点的层数必须不大于父节点的层数,并且尽可能地接近父节点的层数.这样,节点或链路故障时即可实施子树整体迁移,实现节点路由的快速恢复.快速恢复机制详见第 3 节.

© 中国科学院软件研究所 http://www.jos.org.cn

3 转发树构建方法

首先,需要对物联子网根节点进行初始化配置.根节点实际还充当物联网节点与传统互联网通信的网关.根节点与其子节点的通信接口称为下连接口,与互联网相连的接口为上连接口.需要给根节点指定该物联子网的64 位 IPv6 地址前缀 PFX_{64},根节点下连接口的 IPv6 地址为 $PFX_{64}+1$,掩码长度为 64.显然,根节点的子树地址范围就是 PFX_{64}.另外,根节点上连接口以传统方式接入 IPv6 互联网,并向外发布 PFX_{64} 对应的物联子网路由.

接着,物联子网各节点按照指定规则依次加入层次转发树.在 TFAD 模型中,物联子网中新增节点会通过广播"Hello 请求报文"探测直连邻居节点,已加入转发树的节点收到"Hello 请求报文"后会发送"Hello 应答报文".两种 Hello 报文主要字段包括:"层数"描述报文发送节点 N_i 的层数,即 $N_i.lay$;"剩余连接数"描述报文发送节点 N_i 还可接收的子节点数,即 $N_i.rem$ 字段.对于尚未加入转发树的节点,其 Hello 报文中节点层数设置为–1.当一个新节点 N_n 准备加入层次转发树时,它会通过 Hello 报文搜寻其一跳可达的邻居节点,确定一个尽量靠近转发树根(即层数较小)的邻居节点作为父节点 N_f.新节点 N_n 通过节点 N_f 加入物联子网的层次转发树中.

TFAD 模型的核心思想是:将物联子网中的节点构造成一棵层次转发树,并且树节点的 IPv6 地址在子树范围内高度聚合,使得节点根据子节点地址信息就可完成数据转发.层次转发树的构建过程主要考虑两个较为典型的度量参数:子节点数和到网关的转发跳数.下连子节点数的限制是为了各节点的转发负载均衡,转发跳数最少是为了数据快速到达网关.实际部署中,可根据特定环境及需求选择其他度量参数来决定层次转发树的构造方法.

节点 N_n 加入物联子网的过程,即 N_n 发现父节点的过程,描述为 PROC_1,具体步骤如下:

1) N_n 启动邻居搜寻超时定时器 TM_1,超时时间为 t_1.
2) 在 t_1 时间段内,新节点 N_n 通过 Hello 报文搜寻其能够一跳连接的所有邻居节点集合 S_1.
3) TM_1 超时后,如果 S_1 为空,则重新跳转到步骤 1;否则,在 S_1 中挑选已连入层次转发树(即该节点层数不为负数)且子节点数尚未达到最大值(即还可接收子节点数大于 0)的节点,令此集合为 S_2,满足:$S_2=\{N_i|N_i.lay\geqslant 0\wedge N_i.rem>0\}$;如果集合 S_2 为空,则重新跳转到步骤 1).
4) 如果节点 N_n 没有父节点,则执行步骤 5);如果节点 N_n 已有父节点 N_f 但没有备份父节点,则跳转到步骤 7).
5) 在 S_2 中挑选层数最小的节点集合(可能有多个节点同时为层数最小),令此集合为 S_3,满足:$S_3=\{N_i|N_i.lay\leqslant N_j.lay,\forall N_j\in S_2\}$;显然,由于 S_2 不为空,则 S_3 必定不为空.
6) 如在 S_3 中只有 1 个节点,则它成为新节点 N_n 的接入父节点 N_f,重新跳转到步骤 1)(寻找备份父节点);否则,表示有多个节点共同满足层数最小,则选择子节点数最少的两个节点,这两个节点的子节点数可能相同,也可能不同.在这两个节点中,选择子节点数最小的节点(如果两个节点的子节点数相同,则任选一个)为父节点 N_f,选择另一个节点作为备份父节点 N_b,备份父节点 $N_b.rem$ 减 1.然后删除定时器 TM_1,算法结束.
7) 在 S_2 中挑选出层数小于或等于父节点层数,且与父节点层数尽量接近的节点组成集合 S_4.满足:$S_4=\{N_i|(N_i.lay\leqslant N_f.lay)\wedge(N_f.lay-N_i.lay)\leqslant(N_f.lay-N_j.lay),\forall N_j\in S_2\}$.
8) 如果 S_4 为空,则重新跳转到步骤 1);否则,转到步骤 9).
9) 如在 S_4 中只有 1 个节点,则它成为节点 N_n 的备份父节点 N_b;否则,选择子节点数最小的节点(如果有多个子节点数最小的节点,则任选一个)为备份父节点 N_b,备份父节点 $N_b.rem$ 减 1.然后删除定时器 TM_1,算法结束.

父节点发现过程利用邻居搜寻定时器 TM_1,基本能够确保节点 N_n 通过最优的父节点接入层次转发树.此后,考虑拓扑的稳定性以及父节点切换代价,节点在能正常工作的情况下不会再寻找更优父节点.当然,如果父节点及备份父节点全部失效,节点 N_n 将重新启动 PROC_1 过程,寻找新的最优父节点并加入转发树.

新增节点 N_n 确定父节点后,将向父节点发送"节点加入请求报文",父节点回送"节点加入应答报文",相互获得路由转发所需信息.具体步骤描述为 PROC_2.

© 中国科学院软件研究所 http://www.jos.org.cn

1) 新增节点 N_n 向父节点发送“节点加入请求报文”,通告自身的 MAC 地址.
2) 父节点 N_f 接收到 N_n 的“节点加入请求报文”后,将可接入子节点数 $N_f.rem$ 减 1.
3) N_f 为新增子节点 N_n 分配层次比特位 $N_n.val$;若 N_n 是 N_f 的第 k 个子节点,则新增节点的层次比特位取值:$N_n.val=k$.
4) 父节点增加一条转发项,即子节点 N_n 的层次比特位以及对应子节点的 MAC 地址所组成的转发信息项:$(N_n.val,M_n)$.
5) 父节点计算新增子节点 N_n 的子树地址范围 $ARST[n]$,并向 N_n 发送“节点加入应答报文”通告 N_n 对应的子树地址范围,有:$ARST[n]=Inherit(ARST[f])$.
6) 新增节点 N_n 获取自己处于转发树的层数:$N_n.lay=N_f.lay+1$,并根据协议报文存储父节点的链路层地址为 M_f.
7) N_n 设置自身 IPv6 地址为 $ARST[n].IP/64$.

在 TFAD 模型中,节点加入到层次转发树的过程涵盖了节点的地址分配及转发信息生成,整个过程涉及 4 种协议报文:Hello 请求、Hello 应答、节点加入请求、节点加入应答.TFAD 转发树构造过程包含了节点地址分配机制,能确保节点 IP 地址总能在子树范围聚合成一个网络前缀,这使得节点的转发项数很少,是直连子节点数加 1.而且,TFAD 的路由收敛时间相对较短,新节点的加入只对其直连父节点产生影响,无需向根节点层层通告.反之,在 RPL 模型中,新节点加入到树型拓扑时必须由其父节点向着根节点的方向层层通告,构建新增节点的下行路由通道.

TFAD 模型备份父节点机制可有效实现网络故障时的快速路由恢复,每个节点都会尽力寻找备份父节点.当节点 N_i 发现与父节点的无线链路失效时,若有备份父节点 N_b,则会将 N_i 为根的转发子树整体移动到备份父节点的子树中.当然,若 N_i 无下连子节点,则会以单个节点实施父节点切换.对于 TFAD 模型中的任一节点 N_i,其备份父节点的层数总是不大于父节点的层数,该特性确保了链路失效时以 N_i 为根的转发子树整体移动的可行性.需要说明:拥有同一父节点的兄弟节点可能拥有不同的备份父节点,当父节点失效时,它们会选择各自的备份父节点作为自己新的父节点.

节点与备份父节点确定父子关系后,参照 PROC_2 完成信息交互,不同的是:$N_b.rem$ 不需要减 1,因为在 PROC_1 中 $N_b.rem$ 已经减 1.这样可以保证当备份父节点 N_b 切换为 N_i 的父节点时,以 N_i 为根的转发子树一定能连接到 N_b 节点.另外,由于此时 N_i 是带着一棵子树加入到 N_b 子树,还需从 N_i 开始,朝着树叶方向依次向各自的子节点通告新的子树地址范围和各节点所在层数.最重要的是:受影响子树中各节点的转发表项,在子树迁移过程中无需修改.所以,以子树为单位的拓扑重构过程极为简单,能够实现故障时的快速路由恢复.

假设图 2(a)为前期稳定状态时的某个转发树拓扑.当节点 N_5 发现与节点 N_3 的无线链路发生故障,则会将以 N_5 为根的子树整体切换到备份父节点的子树中.有两种可能的切换结果:第 1 种情况是备份节点与父节点在同一层,第 2 种情况是备份节点比父节点的层数要小.情况 1 如图 2(b)所示,若 N_5 的备份父节点为 N_2,则 N_2 与 N_5 参照 PROC_2 交互信息,构建新的转发项.此外,N_2 还需向 N_5 子树层层发布新的子树地址范围.不过,由于 N_5 子树的层次比特位分配没有变化,所以整个 N_5 子树节点的转发表无需改变.情况 2 如图 2(c)所示,若 N_5 的备份父节点为 N_1,除了完成情况 1 所述工作,还需将 N_5 子树中各节点的层数减 1.

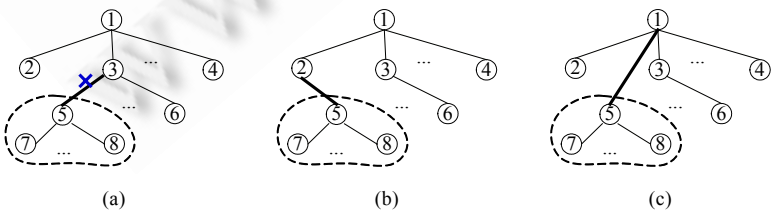


Fig.2 Rapid routing recovery after link failure
图 2 链路故障后的路由快速恢复

© 中国科学院软件研究所 http://www.jos.org.cn

若 N_i 没有备份父节点,则需通知其子树中所有节点解散,让它们各自重新寻找新的父节点.当然,在这种情况下,也可以设计其他优化算法来实现该子树某一部分整体移动.另外,还可以设计预留资源方法确保转发树中各节点都能拥有一个备份父节点.限于篇幅,本文不对上述机制详细描述.

对于子节点 N_i 的离开,其原父节点 N_f 需要完成资源回收及信息更新.主要行为描述为 PROC_3.需要说明:若离开的节点 N_i 下连一个子树,父节点的行为也是如此.

1) 父节点 N_f 的可接入子节点数 $N_f.rem$ 加 1;
2) N_f 回收子节点 N_i 对应的层次比特位,可供以后的新增子节点使用;
3) 父节点删除 N_i 对应的转发项:$(N_i.val,M_i)$.

此外,TFAD 模型能够方便地实施规模扩展.如图 3 所示,对于规模更大的物联子网环境,可构造基于 TFAD 的层次路由体系.整个物联子网包括多个网关,每个网关都是一个 TFAD 层次结构的根节点,各网关可按照传统域内路由协议互连,并与互联网连接.当然,层次路由体系中每个根节点需要发布其转发树标识以相互区分,协议交互更为复杂,本文专注于单棵转发树的路由转发分析,对多层路由体系不做细述.

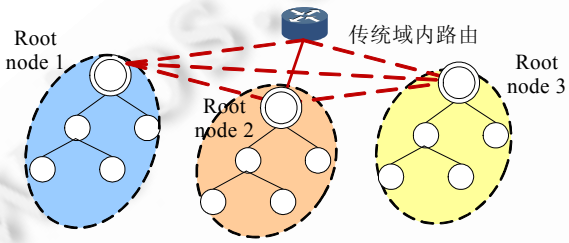


Fig.3 Hierarchical routing architecture with multi-gateway based on TFAD

图 3 基于 TFAD 的多网关层次路由体系

4 表项存储及转发处理

对于 TFAD 模型转发树中的任一节点 N_i,其转发项结构为

$$(LB[j],MAC_j),0\leqslant j\leqslant NUM_{son}.$$

该转发项用于存储 N_i 的第 j 个子节点的层次比特位及其 MAC 地址.当 j 为 0 时,该转发项对应 $LB[j]$的值为 0,MAC_j 存储了 N_i 的父节点 MAC 地址.根据前文所述,层次比特位位数最多为 16,占 2Bytes,加上 8Bytes 的 MAC 地址,一条转发项只需占用 10Bytes.典型 IPv6 路由系统中,一条 IPv6 转发项的长度至少为 25Bytes(16 字节的目的网段、1 字节的掩码长度、8 字节的下一跳链路地址).本文设计的 TFAD 转发项的长度为 10 个字节,只是原来的 40%,极大地节省了节点存储资源.除主体转发项外,节点 N_i 还要存储其子树地址范围 $ARST[i]$,可据此判断一个报文的目的端是否属于 N_i 子树.

图 4 是层次转发树中,某节点 N_i 的报文转发处理过程.由于 TFAD 模型中各节点 IP 地址是严格按子树聚合,所以可根据目的地址是否属于 N_i 的子树地址范围来判断上行流量和下行流量.上行流量转发给 N_i 的父节点,下行流量通过转发表查询发送给 N_i 的某个子节点.具体步骤如下:

1) 通过无线接口收到一个 IPv6 报文,取其目的 IPv6 地址 IP_{dest}.
2) 若 IP_{dest} 属于节点 N_i 的子树地址范围 $ARST[i]$,则说明目标节点处在以 N_i 为根的子树中,跳转至步骤 4);若 IP_{dest} 不属于 $ARST[i]$,则执行步骤 3).
3) 若此报文源 MAC 地址为 N_i 父节点的链路层地址,则说明发生路由环,丢弃报文,转发结束;否则,转发报文给 N_i 的父节点,父节点链路层地址为转发表中的$(0,MAC_0)$转发项的 MAC_0,转发结束.
4) 若 IP_{dest} 为本节点地址,则提交上层协议处理,转发结束;否则,执行步骤 5).
5) 取 IP_{dest} 中对应 N_i 节点层次比特位的值,即 IP_{dest} 中从第 $LB[N_i.lay].b$ 位开始,到第$(LB[N_i.lay].b+LB[N_i.lay].n-1)$位的取值,并在转发表中查找匹配项;若查找失败,则丢弃报文,结束;若匹配成功,则将

© 中国科学院软件研究所 http://www.jos.org.cn

报文转发到匹配项对应的链路层地址.

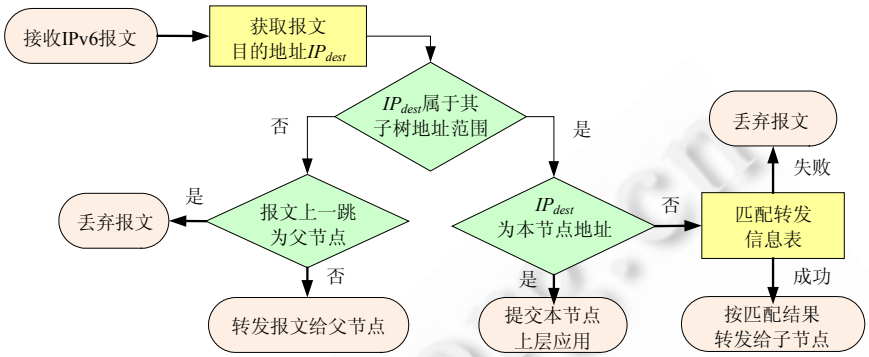


Fig.4 Packets forwarding procedure

图 4 报文转发过程

TFAD 转发模型有如下特点:1) 3 层路由转发与 2 层链路地址查找集成在一起,可快速完成整个转发处理过程;2) 转发表项简单、实用,只存储“层次比特位”与对应的 MAC 地址;3) 可支持链路故障时转发系统的快速恢复,子树整体切换时无需修改大多数节点的主体转发表项.

5 实验及性能分析

在 TFAD 模型中,由于节点只存储各子节点层次比特位和对应的链路层地址,存储性能有较大的优化.为了方便分析,令构建的转发拓扑为满 m 叉树,即,除了底层节点以外,每个节点都有 m 个子节点,共有 n 层.其中,第 0 层有 1 个根节点,第 1 层有 m 个节点,第 i 层有 m^i 个节点.为了便于分析,我们不考虑第 2 节所述的 IP 地址主机位全为 1 的节点.令转发树中节点总数为 Num_{nodes},则有:

$$Num_{nodes}=m^0+m^1+\ldots+m^{n-1}=\left(\sum_{i=0}^{n-1}m^i\right).$$

TFAD 模型中,每个非底层节点存储的转发项数为$(m+1)$,即,m 项子节点和 1 项父节点的转发信息;另外,底层节点只需存储 1 项父节点的转发信息.因此,可计算 TFAD 模型所有非底层节点转发项数共为 $(m+1)\times\sum_{i=0}^{n-2}m^i$.所有底层节点转发项数共为 m^{n-1}.所以,转发树所有节点转发项数为

$$(m+1)\times\sum_{i=0}^{n-2}m^i+m^{n-1}.$$

本文以 RPL 转发模型为比较对象,对 TFAD 转发表存储性能进行分析.RPL 模型中,每个节点也需存储 1 项父节点的转发信息;另外,还需为其所有的子孙节点存储转发信息.第 0 层根节点转发项数为 $\sum_{i=1}^{n-1}m^i$, 第 1 层共有 m^1 个节点,每个节点转发项数为 $\sum_{i=2}^{n-1}m^i$; 以此类推,第 j 层共有 m^j 个节点,每个节点转发项数为 $\sum_{i=j+1}^{n-1}m^i$,满足 $j<n-1$.另外,共有 m^{n-1} 个底层节点,它们只需存储 1 项父节点转发信息.所以,RPL 模型中所有节点转发项数为

$$\sum_{j=0}^{n-2}\left(m^j\times\sum_{i=j+1}^{n-1}m^i\right)+m^{n-1}.$$

显然,TFAD 模型中整个转发树的转发项数量相对于 RPL 模型有大幅减少.继续分析单节点转发项数量,令转发树中各节点的子节点数 m 分别为 1,2,3,针对 5 层树结构比较 RPL 模型与 TFAD 模型的单节点转发表容量,如图 5 所示.两种模型的底层节点,都只需存储去往父节点的 1 条转发项.当 m 等于 1 时,即物联子网为线型连接

© 中国科学院软件研究所 http://www.jos.org.cn

拓扑.对于靠近树根的节点,RPL 模型转发项数比 TFAD 模型略多,并且越往叶子节点靠近,转发项数差异越小.然而随着子节点数 m 的增长,RPL 模型中靠近树根的节点转发项数却急剧增长.但 TFAD 模型中各节点存储的转发项数只与其直连子节点数相当,不会随着层数减小而增长.此外,TFAD 模型节点的小容量转发表,还可提高报文转发过程中表项的匹配速度.所以,与 RPL 模型相比,TFAD 模型对物联网节点的转发性能也有较大提高.

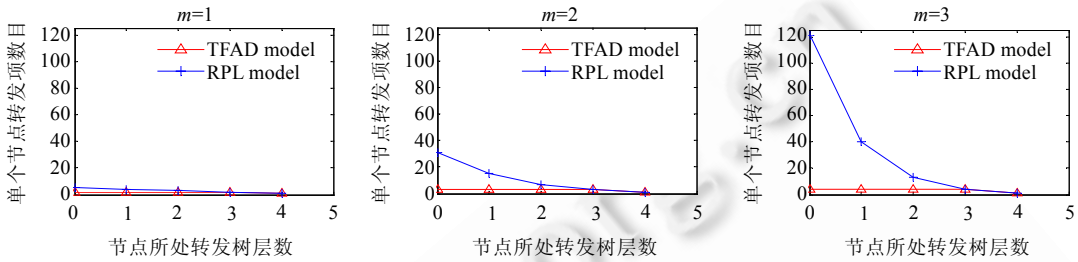


Fig.5 Analysis of forwarding items number of one node in TFAD model

图 5 TFAD 模型单节点转发项数分析

本文通过真实物联网节点组网,测试了 TFAD 模型的路由性能,节点参数描述见表 2.

Table 2 Parameters of IOT nodes

表 2 物联网节点参数

参数	具体描述
处理器	型号 MCU8051,主频 32MHz
链路层协议	IEEE 802.15.4
传输速率	250Kbps
通信范围	无障碍直线范围 100m
存储	SRAM: 8KB, FLASH: 128KB
操作系统	Contiki 2.6

实验在各个节点的链路层实施 MAC 地址过滤,构建图 6(a)所示的典型网络拓扑结构.通过同类型无线节点集成串口,与普通主机连接,形成报文监控节点.监控节点可捕获 802.15.4 协议报文,并在 Wireshark 工具中显示抓包信息.实验中,设置各节点加入路由树后会定时向节点 1 发送类似 PING 机制的 echo request 报文,节点 1 收到后会回送 echo reply 报文.那么,各节点路由学习成功后会在图 6 椭圆范围出现相应的双向通信报文.实验通过各节点首次与节点 1 正常双向通信来判定其路由学习完成时间.所以,实验中只需部署一个监控节点,对图 6 中的椭圆范围进行报文捕获,即可获得各节点路由学习时间.

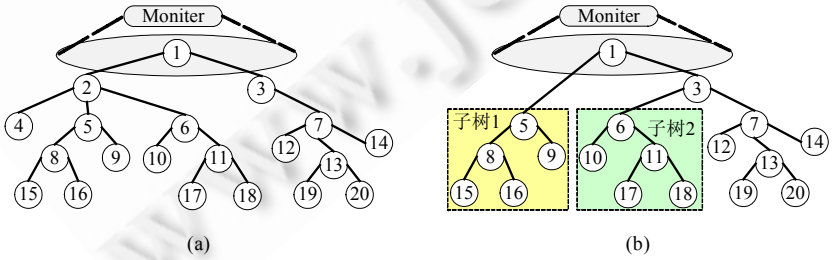


Fig.6 Topology of network with real nodes

图 6 真实节点网络拓扑

分别在各节点中实施 RPL 模型和 TFAD 模型,RPL 协议是 Contiki 操作系统自带功能,并且节点加入方式为主动请求方式.图 7 介绍了两种模型在 Contiki 中实现的功能结构图,给出了与本文相关的主要功能函数.图 7(a)

© 中国科学院软件研究所 http://www.jos.org.cn

描述了 RPL 的主要功能模块.控制层面:RPL 协议收到 DAO 报文[5]后,生成路由项并添加到全局路由表中.数据层面:底层接收报文后进行分析,将目的 IP 地址为本地的报文交由应用层处理,也包括路由协议报文.对于其他报文实施转发处理,查找路由获得下一跳地址,继续查找邻居 MAC 地址并通过 6LoWPAN 封装发送.图 7(b)是本文设计的 TFAD 模型基本实现,浅灰色的模块与 RPL 不同,其他模块的实现与 RPL 相同.控制层面:收到新的子节点加入请求则增加转发项,转发项包含子节点的层次比特位及该节点的 MAC 地址.数据层面:转发处理与 RPL 有明显差异,由于转发项包含 MAC 地址,则无需再查找邻居 MAC 地址.

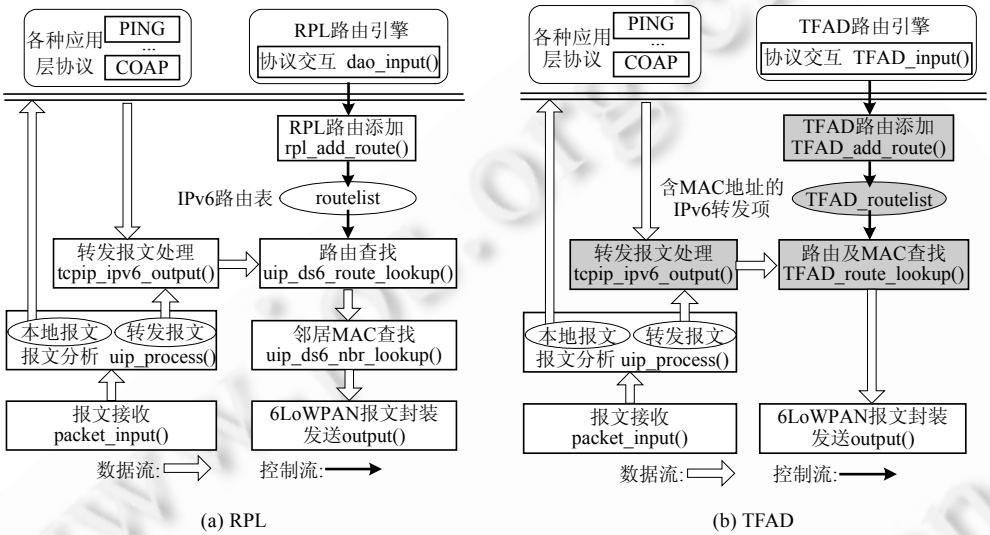


Fig.7 Contiki-Based implementation of two forwarding models

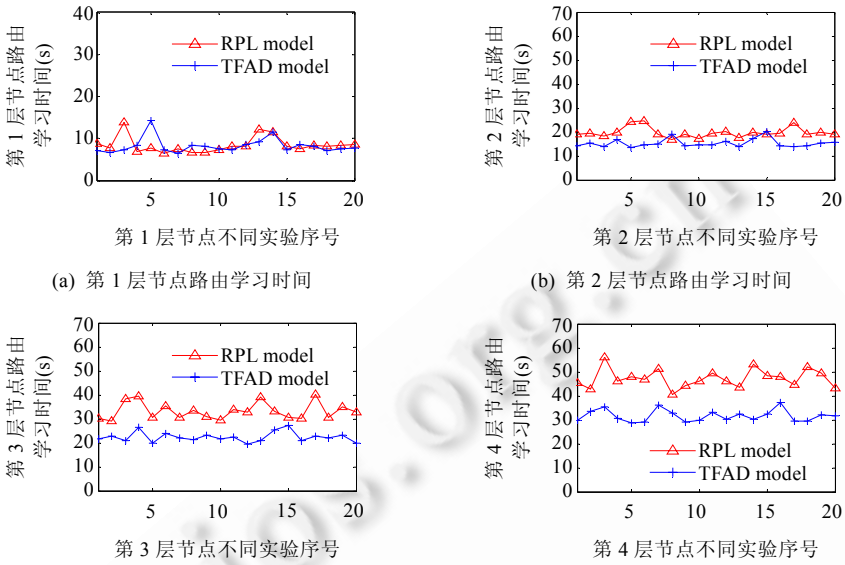
图 7 两种转发模型基于 Contiki 的实现

图 8 是两种模型下不同层数节点多次实验的路由学习时间分析,每层节点取 20 次实验数据.对于第 1 层节点,两种模型的路由学习时间相差无几;但对于 2 层及 2 层以上节点,TFAD 模型中节点路由学习速度明显优于 RPL 模型.而且,随着节点所处层数的增加,两种模型中节点的路由学习时间差更为明显.所以,对于无线节点构成的层次拓扑树,其层数越大,则 TFAD 模型的快速路由学习优势就更为明显.实验中发生的无线传输偶尔丢包,会导致路由学习时间的延长,表现为图 8 中折线的突然升高.

本文还基于图 6(a)拓扑进行了两种模型网络状态变化后的路由恢复实验.基于图 6(a)拓扑的网络稳定后,将节点 2 和节点 4 关闭,网络拓扑变化为图 6(b).其中,节点 5 为根的子树(子树 1)将以节点 1 为自己新的父节点,该子树所有节点的层数都发生了变化;节点 6 为根的子树(子树 2)将以节点 3 为自己新的父节点,该子树所有节点的层数都未发生变化.在 TFAD 实验中,图 6(a)拓扑稳定时节点 5 的备份父节点为节点 1,节点 6 的备份父节点为节点 3.整个实验仍通过监控节点捕获报文,当节点 N_i 能与节点 1 正常双向通信,才认为节点 N_i 的路由恢复正常.

在上述网络拓扑变化情况下,TFAD 模型和 RPL 模型中不同层数节点的路由恢复时间如图 9 所示.受影响子树的第 0 层节点在 TFAD 模型与 RPL 模型中的路由恢复时间相差不大,如图 9(a)所示,这是因为切换过程涉及的协议控制报文数量相当.然而对于受影响子树的其他节点,TFAD 模型的路由恢复时间却有较为明显的优势,如图 9(b)和图 9(c)所示.而且,无论备份父节点与父节点层数是否相同,路由恢复实验中所有节点在 TFAD 模型中的路由恢复时间都较短并且变化不大.此外,针对受影响子树部署监控节点,捕获协议报文进行分析,可以发现:TFAD 模型在子树整体迁移过程中,只需要极少的控制消息,主要用于通告子树地址范围和各节点所在层数的变化;而且,TFAD 模型的网络拓扑变化不用像 RPL 模型那样层层向上通告,而只需向新的父节点通告即可.这也是 TFAD 模型快速路由恢复的一个关键因素.

© 中国科学院软件研究所 http://www.jos.org.cn

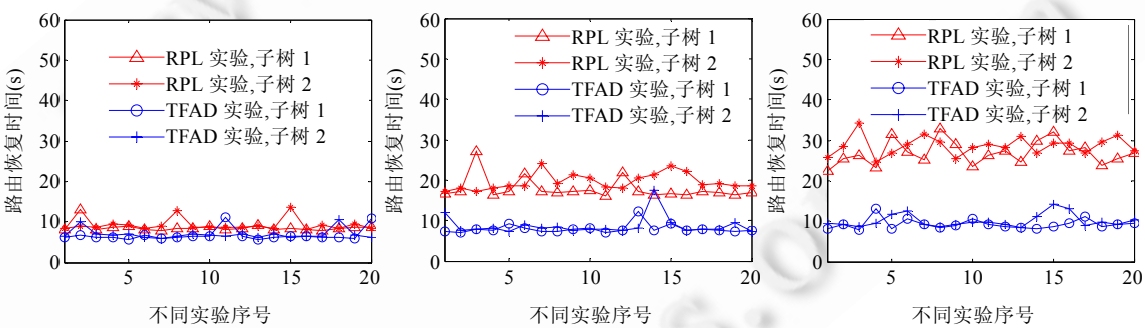


(a) 第 1 层节点路由学习时间 (b) 第 2 层节点路由学习时间

(c) 第 3 层节点路由学习时间 (d) 第 4 层节点路由学习时间

Fig.8 Time for achieving routes of nodes with different layers

图 8 不同层数节点获取路由时间



(a) 受影响子树第 0 层节点 (b) 受影响子树第 1 层节点 (c) 受影响子树第 2 层节点

Fig.9 Routing recovery time of nodes with different layers in the influenced sub-tree

图 9 受影响子树中不同层数节点路由恢复时间

综上可知:在多跳(2 跳以上)路由结构中,TFAD 模型的路由学习速度和网络状态变化后的路由恢复速度都比 RPL 模型有较大的优势;而且随着网络路由跳数增大及各节点子节点数增多,TFAD 比 RPL 具有更好的路由存储性能.TFAD 模型付出的代价是 IPv6 地址空间的低利用率,具体的利用率与实际部署情况相关.TFAD 和 RPL 路由模型的各项性能比较见表 3.

Table 3 Performance comparison for two models

表 3 两种模型的性能比较

	RPL	TFAD
路由存储性能	一般,需存储子树中所有节点地址对应路由项	很好,路由容量与直连子节点数相当
收敛速度	一般	较快
故障恢复	一般	较快
协议复杂性	简单	简单
地址空间	利用率高,无地址空间浪费	利用率略低,存在地址空间浪费

© 中国科学院软件研究所 http://www.jos.org.cn

6 总 结

RPL 协议是 IPv6 物联网中最广为认可的路由机制,但在规模略大的多跳网络结构中,面临着转发节点存储及处理负载过重的问题.而且,物联网中扁平化 IPv6 地址体系使得这一问题更加突出.本文设计了节点地址可基于子树聚合的轻量级转发模型 TFAD,将物联网节点构造成一棵层次转发树,每一子树中的节点地址总能汇聚为一个 IPv6 网段.各节点只需存储与其子节点数相当的转发项,即可支持 TFAD 模型的数据转发,极大地减少了树节点的存储消耗,具有很好的路由可扩展性.性能分析表明:随着网络规模的逐渐增大,TFAD 模型相对于 RPL 模型具有更好的存储性能和路由收敛速度.TFAD 模型的协议交互简单,只需极少的父子节点信息交互即可实现子网内节点的路由信息生成.每一子树都有对应的 IP 网段,子树内部的拓扑结构信息对上层节点透明.另外,还设计了 TFAD 模型的备份父节点机制,支持节点或链路故障后以子树为单位进行网络拓扑重构,从而实现物联子网的快速路由恢复.基于真实物联网节点的组网实验,验证了 TFAD 模型节点的路由学习速度较 RPL 模型占优;而对于网络故障后的路由恢复,TFAD 模型也更为快速.

当然,TFAD 模型层次地址结构可能会导致物联子网内 IPv6 地址的少许浪费,但可以优化 TFAD 地址分配机制以节约地址.另外,虽然 TFAD 模型可根据部署情况设置不同的层数和层次比特位数,但本文方案仍有“同层节点总是具有相同的层次比特位”的约束.可对 TFAD 模型优化,使得各子树能够根据实时连接状态,调整单个子树范围内的层数及层次比特位,从而有效提升 TFAD 模型的灵活性.而且,IPv6 较大的子网地址范围能够为该思想提供有力的支撑.本文的后续工作将对上述问题展开研究.

References:

[1] Commission of the European Communities. Internet of things—An action plan for Europe. 2009.

[2] Meyer D, Zhang L, Fall K. Report from the IAB workshop on routing and addressing. RFC 4984, 2007.

[3] Dunkels A, Vasseur JP. IP for Smart Objects. ISPO Alliance White Paper, 2008.

[4] Ma YW, Lai CF, Huang YM, Chen JL. Mobile RFID with IPv6 for phone services. In: Proc. of the 13th IEEE Int’l Symp. on Consumer Electronics. Kyoto: IEEE Press, 2009. 169−170. [doi: 10.1109/ISCE.2009.5156859]

[5] Winter T, Thubert P, Hui J, Kelsey R, Levis P, Pister K, Struik R, Vasseur JP, Alexander R. RPL: IPv6 routing protocol for low-power and lossy networks. RFC 6550, 2012.

[6] Narten T, Nordmark E, Simpson W, Soliman H. Neighbor discovery for IP version 6 (IPv6). RFC 4861, 2007.

[7] Akkaya K, Younis M. A survey on routing protocols for wireless sensor networks. Ad Hoc Networks, 2005,3(3):325−349. [doi: 10.1016/j.adhoc.2003.09.010]

[8] Atzori L, Iera A, Morabito G. The Internet of things: A survey. Computer Networks, 2010,54(15):2787−2805. [doi: 10.1016/j.comnet.2010.05.010]

[9] IEEE.802.15.4. http://www.ieee802.org/15/pub/TG4.html

[10] Kim E, Kaspar D, Gomez C, Bormann C. Problem statement and requirements for IPv6 over low-power wireless personal area network (6LoWPAN) routing. RFC 6606, 2012.

[11] Jobin J, Krishnamurthy SV, Tripathi SK. A scheme for the assignment of unique addresses to support self-organization in wireless sensor networks. In: Proc. of the IEEE 60th Vehicular Technology Conf. Los Angeles: IEEE Press, 2004. 4578−4582. [doi: 10.1109/VETECF.2004.1404946]

[12] Lin JL, Liu YH, Ni LM. SIDA: Self-Organized ID assignment in wireless sensor networks. In: Proc. of the IEEE Int’l Conf. on Mobile Adhoc and Sensor Systems. Pisa: IEEE Press, 2007. 1−8. [doi: 10.1109/MOBHOC.2007.4428604]

[13] Chen HM, Cui L, XIE KB. A comparative study on architecture and implementation methodologies of Internet of things. Chinese Journal of Computers, 2013,36(1):168−188 (in Chinese with English abstract).

[14] Droms R, Bound J, Volz B, Lemon T, Perkins C, Carney M. Dynamic host configuration protocol for IPv6 (DHCPv6). RFC 3315, 2003.

[15] Villalba LJG, Orozco ALS, Cabrera AT, Abbas CJB. Routing protocols in wireless sensor networks. Sensors, 2009,9(11):8399−8421.

© 中国科学院软件研究所 http://www.jos.org.cn

[16] Singh SK, Singh MP, Singh DK. Routing protocols in wireless sensor networks—A survey. Int'l Journal of Computer Science & Engineering Survey (IJCSES), 2010,1(2):63−83. [doi: 10.5121/ijcses.2010.1206]

[17] Estrin D, Govindan R, Heidemann J, Kumar S. Next century challenges: Scalable coordinate in sensor network. In: Proc. of the 5th ACM/IEEE Int'l Conf. on Mobile Computing and Networking. New York: ACM Press, 1999. 263−270.

[18] Singh SK, Singh MP, Singh DK. A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks. Int'l Journal of Advanced Networking and Application, 2010,2(2):570−580.

[19] Shen B, Zhang SY, Zhong YP. Cluster-Based routing protocols for wireless sensor networks. Ruan Jian Xue Bao/Journal of Software, 2006,17(7):1588−1600 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1588.htm [doi: 10.1360/jos171588]

[20] Manjeshwar A, Agrawal DP. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In: Proc. of the 15th Int'l Parallel and Distributed Processing Symp. (IPDPS). San Francisco: IEEE Press, 2001. 2009−2015. [doi: 10.1109/IPDPS.2001.925197]

[21] Kulik J, Heinzelman W, Balakrishnan H. Negotiation-Based protocols for disseminating information in wireless sensor networks. Wireless Networks, 2002,8(2/3):169−185. [doi: 10.1023/A:1013715909417]

[22] Heinzelman WB, Chandrakasan AP, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. on Wireless Communications, 2002,1(4):660−670. [doi: 10.1109/TWC.2002.804190]

[23] Li ZJ, Liu YH, Li M, Wang JL, Cao ZC. Exploiting ubiquitous data collection for mobile users in wireless sensor networks. IEEE Trans. on Parallel and Distributed Systems, 2013,24(2):312−326. [doi: 10.1109/TPDS.2012.92]

[24] Fonseca R, Gnawali O, Jamieson K, Kim S, Levis P, Woo A. TEP 123: The collection tree protocol. 2006.

[25] Gnawali O, Fonseca R, Jamieson K, Moss D, Levis P. Collection tree protocol. In: Proc. of the 7th ACM Conf. on Embedded Networked Sensor Systems. New York: ACM Press, 2009. 1−14. [doi: 10.1145/1644038.1644040]

[26] Perkins C, Belding-Royer E, Das S. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, 2003.

[27] Contike. http://www.contiki-os.org/

[28] TinyOS. http://www.tinyos.net/

附中文参考文献:

[13] 陈海明,崔莉,谢开斌.物联网体系结构与实现方法的比较研究.计算机学报,2012,36(1):168−188.

[19] 沈波,张世永,钟亦平.无线传感器网络分簇路由协议.软件学报,2006,17(7):1588−1600. http://www.jos.org.cn/1000-9825/17/1588.htm [doi: 10.1360/jos171588]

肖融(1974－),女,贵州沿河人,博士生,讲师,CCF 会员,主要研究领域为物联网,新型互联网体系结构.
E-mail: xiaorong@bnu.edu.cn

孙波(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机应用技术.
E-mail: tosunbo@bnu.edu.cn

陈文龙(1976－),男,博士,讲师,CCF 会员,主要研究领域为物联网,高性能路由系统.
E-mail: wenlongchen@sina.com

© 中国科学院软件研究所 http://www.jos.org.cn