

一种可扩展的无线 Mesh 适配树路由协议

王江涛, 葛 强, 钱 炜, 蔡得菊

(中国十七冶集团有限公司 信息化部, 安徽 马鞍山 243061)

摘 要: 提出一种可扩展的无线 Mesh 适配树路由协议(SMTRP), 解决无线 Mesh 随着网络规模扩大引起路由延迟增大, 网络性能严重下降的问题. 该协议引入 Mesh 适配树和自适应地址块机制, 通过将网络拓扑反映在逻辑地址上, 每个节点仅需维护自己的 N 跳邻居节点的信息, 判断出数据包该往哪个方向转发, 无需维护大量的路由表, 节省了存储空间, 减少了通信开销以及能量消耗. 理论证明了 SMTRP 算法的可扩展性, 并在 OPNET 仿真平台上验证了 SMTRP 协议的有效性.

关键词: 无线 Mesh; 路由协议; 分布式链路状态; 自适应地址块

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2012)05-0086-04

A Scalable Wireless Mesh Tree Routing Protocol

WANG Jiang-tao, GE Qiang, QIAN Wei, CAI De-ju

(Department of Information Technology, China MCC17 Group Company Limited, Maanshan 243061, China)

Abstract: This paper puts forward the scalable Mesh Tree Routing Protocol (SMTRP) to solve the serious decrease of network performance resulted from the increased routing latency in the large-scale network. By reflecting the network topology to logic addresses, introducing Mesh Adaptive Tree (MAT) and Adaptive Block Addressing (ABA) mechanism, each node is only required to maintain its own neighbor node's information within N hops. As a result, this protocol can verdict the direction of delivering the data packets, and dispense with maintenance of abundant routing tables, thus save a lot of storage space, reduces the communication overhead and energy consumption. The scalability of SMTRP Protocol has been proved theoretically and the validity of it has been validated by OMNET simulate platform.

Key words: wireless mesh; routing protocol; distributed link state; adaptive block addressing

1 引言

无线 Mesh 现有路由协议都存在一些实现的难题和不足, 典型的如网关协议随着规模扩大, 端到端可靠性降低, 从而导致 QoS 明显下降^[1-3]. 此外, 在一个很大的无线网络中即使路由已经建好, 路径上节点状态也可能会变化, 路由协议将无法找到一条可靠的路由路径^[4]. 无线 Mesh 开放式的网络架构可以获得较好的可扩展性, 然而缺点是通信协议随着网络规模的扩大, 路由协议的开销急剧上升, 算法收敛困难, 路由延迟增大, 路由修复困难, 网络性能严重下降.

本文提出的无线 Mesh 适配树路由协议 (Scalable Mesh Tree Routing Protocol, SMTRP) 省去了数据包转发之前路由路径的发现时间, 同时每个节点仅仅维护自己的 N 跳的邻居节点的信息, 减少存储空间和通信开销.

2 SMTRP 路由协议的实现

SMTRP 路由协议主要包括两个部分: 一是 Mesh 树的初始化, 包括节点的关联、地址块的分配以及邻居关系的建立; 二是路由阶段.

2.1 Mesh 树的建立

根据无线 Mesh 网络中节点加入的先后顺序构

建 Mesh 树,树型网络结构形成之后,根据每个路由器设定的允许其他路由器和客户端接入数量,自根路由器节点而下,分配临时网络地址。

(1) 节点关联

最大程度地设置最佳根节点,可以减少网络拓扑结构的大幅度变化带来的开销过大等问题^[5]。在关联阶段,手工方式将最早加入网络的功能较强、移动性较小、位于区域中心的路由器设置为根节点,产生一个仅有一个节点的一级核心树。然后当有新的节点加入网络后,不断在无线公共信道上广播发送 RREQ 报文,与网络中已经存在的节点进行关联。如果某个 Mesh 路由器节点要接入的路由节点的数目不确定,它可以用一个接受度^[6](Acceptance Degree, AD),表明它接受其他 Mesh 路由节点“愿望”的大小。这样便于控制 Mesh 路由器的转发能力,控制路由器的用户接入,有利于协调全网内的资源分配和每个路由器负载的均衡,避免了树形网络中某一个节点负担太大,以最大化网络容量。如,一个 4 级 AD,可以用“3”表示无条件接受,用“2”表示有条件接受,用“1”表示勉强接受,用“0”表示拒绝。

(2) 地址块分配

树形网络建立之后,路由器根节点广播一个后继节点统计信息,网络中每个路由器负责统计以它为根的所有孩子节点需要提供的本地接入地址数目。从树形最低端的节点开始向它的父节点发送一个地址块请求帧 AREQ,其中包含一个子节点计数器,其值设为 1,表示初始只有自己一个节点。网络中的每个路由器节点都从自己的子节点处收集这些 AREQ,计算出其中的计数值总和,然后加 1。接着,此节点路由器生成一个地址请求帧 AREQ,这个新的 AREQ 的内容包括上一步算出的数值 X,然后逐步向上层路由器汇报,直到根节点。等待所有的节点信息都返回到根节点后,计算出其中的计数值总和,由根节点负责统一分配地址块。为了路由表的简洁,根节点向下分配的地址不再是 IP 地址,而是编号连续的临时地址 ID,这些连续的临时地址 ID 分配给路由器,供路由器协调分配给以后接入路由器和本地用户。临时分配的地址 ID 和移动用户节点固有的 IP 地址形成固定的映射。

(3) 邻居关系建立

每个节点都维护自己的一张 N 跳邻居节点列表,一般将 N 设为 2 或者 3,此处将 N 设置为 2。邻居关系的建立是在 Mesh 适配树形成的过程中就形成的,当一个新节点刚刚加入网络时,将向周围广播

发送 RREQ 报文,周围节点中已经加入树的,回复 RREP 报文,同时,接收到 RREQ 报文的节点将此新节点加入到其邻居节点列表里。此节点到邻居节点的跳数用 TTL 的初始值减去 TTL 的剩余值。核心树形成之后,路由器向邻近路由节点发送邻居加入请求报文,建立路由器之间的横向连接,形成邻居节点,邻居 WR 的级别之差的绝对值小于等于“1”。

2.2 路由阶段

客户端节点不具有转发功能,当某一客户端节点向其他节点发送消息时,需要通过本地接入节点将其把数据包转发出去,这也就可以将 Mesh 网络中的所有路由问题归结为一维平面上的数据包转发过程。当一个节点 S 需要向节点 D 发送数据包的时候,它总要通过自己的邻居节点将数据包转发出去,因此找到要通过自己的哪一个邻居节点将数据包转发出去是本路由算法的关键。路由发现的伪代码如下:

```
func_nextHop(dst)
```

```
{
    if(dst 在本节点的 neighbor list 列表中)
    {
        directionNb=dst;
        next_hop=getOneHopNeighbor(directionNb);
        return next_hop;
    }
    else if(dst 是本节点 neighbor list 中节点的后代节点)
    {
        directionNb=找到一个 neighbor 使得这个节点是 level 最大的节点;
        next_hop=getOneHopNeighbor(directionNb);
        return next_hop;
    }
    else
    {
        hops2root=本节点的 neighbor list 中,level 比本节点小,同时节点的(hops+tree_level)最小的值;
        minHops=本节点的 neighbor list 中,具有 hops2root 值,同时具有最大 level 的节点的 hops 值;
        directionNb=本节点的 neighbor list 中,选取具有 hops2root 值同时 tops 属性为 minHops 的节点;
        next_hop=getOneHopNeighbor(directionNb);
        return next_hop;
    }
}

func_getOneHopNeighbor(directionNb)
{
    current_hops=directionNb 的 hops 值(即本节点到 directionNb 节点的跳数);
```

```

if (current_hops==1)
    return directionNb;
else
{
    while current_hops>1
    {
        for (本节点的 neighbor list 中, 直接与 directionNb 相
        连的节点 Nbi);
        if (Nbi 的 hop_number==current_hops-1)
        {
            func_getOneHopNeighbor(Nbi);
            current_hops=current_hops-1;
        }
    }
    end while
}
}

```

3 SMTRP 协议可扩展性分析

3.1 性能参数指标

三元组决定了网络的可扩展性, 根据文献[3]定义的不同性能指标的可扩展性因子分析如下:

(1) 设 $T_r(\lambda_1, \lambda_2, \dots)$ 是最小传输负荷, $\lambda_1, \lambda_2, \dots$ 表示网络的参数(如网络大小、移动率、数据生成率等), 一个网络扩展性的参数 $\lambda_i(\Psi_{\lambda_i})$ 定义为:

$$\Psi_{\lambda_i} \triangleq \lim_{\lambda_i \rightarrow \infty} \frac{\log T_r(\lambda_1, \lambda_2, \dots)}{\log \lambda_i}$$

(2) 设路由协议 X 引起的总开销为 $X_{OV}(\lambda_1, \lambda_2, \dots)$, X 取决于参数 $\lambda_1, \lambda_2, \dots$, 则相对于参数 $\lambda_i(\rho_{\lambda_i}^X)$, 协议 X 的规模因子定义为:

$$\rho_{\lambda_i}^X \triangleq \lim_{\lambda_i \rightarrow \infty} \frac{\log X_{OV}(\lambda_1, \lambda_2, \dots)}{\log \lambda_i}$$

考虑到网络的大小, 为使网络具有可扩展性, 要求 $\Psi_N \leq 1$. 对于参数 λ_i , 当且仅当参数 λ_i 增加时, 由协议引起的总开销 X_{OV} 的增加不大于网络的最小业务负荷, 即当且仅当 $\rho_{\lambda_i}^X \leq \Psi_{\lambda_i}$, 称路由协议 X 是可扩展的; 对于网络大小 N , 当且仅当 $\rho_N^X \leq 2$, 路由协议 X 是可扩展的; 对于节点的移动速率 λ_k , 当且仅当 $\rho_{\lambda_k}^X \leq 0$, 路由协议 X 是可扩展的; 对于网络的业务负荷 λ_t , 当且仅当 $\rho_{\lambda_t}^X \leq 1$, 路由协议是可扩展的.

3.2 SMTRP 的可扩展性

设在 n 个网络范围内分布有 N 个节点, 当平均最优分组长度为 L 时, 各分组需要传输 $(n-1)$ 次, 由于每秒产生的数据分组为 $\lambda_t N$ (λ_t 为每秒产生的业务量), 所以传输所有这些分组的附加带宽为 $(n-1)(\lambda_t N - L)$.

最先收到 RREQ 报文的节点发送应答信息 RREP 来建立父子关系, 当节点收到 RREP 时, 发送 RACK 报文予以确认. 设节点加入 RREP 的平均速率为每秒 λ_s ($\lambda_s \leq \lambda_t$), 则 Mesh 树建立的网络负荷为 $\frac{3}{2} \lambda_s (N-1)(\max Hops + 1)$. 而数据包的发送由于不采用事先发现路由后再传输数据包的机制, 而是直接根据每个节点所谓好的邻居状态直接转发数据包, 所以传统意义上路由发现所带来的网络开销可以忽略.

在 SMTRP 中节点位置更新消息发送速率为每秒 λ_k , 位置更新负荷为 $\lambda_k N$, 无线 Mesh 网络中 SMTRP 路由协议的总的开销为: $(n-1)(\lambda_t N - L) + \frac{3}{2} \lambda_s (N-1)(\max Hops + 1) + \lambda_k N < \lambda_t N (n-1) + \frac{3}{2} \lambda_s (N-1)(\max Hops + 1) + \lambda_k N = \Theta[\lambda_t N (n-1)]$. 结果, $\rho_{\lambda_t}^{MTDLS} = 1, \rho_N^{MTDLS} = 1, \rho_{\lambda_k}^{MTDLS} = 0$.

从上面的理论分析可知, SMTRP 协议对节点的移动速率、网络的业务负荷、节点数目都具有可扩展性.

4 仿真比较

通过网络仿真软件 OPNET modeler 10.5, 利用其已有的网络模块, 建立无线 Mesh 路由仿真平台, 并在此平台上通过对包传递比率、端到端平均跳数、端到端平均时延、每跳平均延迟和端到端通信效率(吞吐量)等参数比较传统的 AODV 路由协议和 SMTRP 路由协议的网络性能. 选择 AODV 作为对比, 是因为它已被 IEEE802.11s 提出作为解决 Mesh 接入点路由流量的协议. 比较结果, SMTRP 路由协议具有性能优势.

包传递比率的仿真图如图 1 所示. 当网络规模控制在 100 个节点以内时, 两种路由协议都可以达到 90% 以上的包传递比率. 随着网络范围的变大, 这两种包传递比率急剧下降, 而且 AODV 下降的比 SMTRP 迅速. 这是因为, 随着网络规模的扩大, 以及网络中越来越多的出现动态节点, AODV 路由发现过程中建立起来的路由路径很可能不再存在, 这样就需要重新进行路由发现过程, 大大降低了包传递比率.

图 2、图 3 显示网络端到端的平均跳数和端到端的平均时延, 这两个评估参数从两个角度对比了两种路由协议对最优路径的查找能力. 可以看出 AODV 在网络规模较小时, 比 SMTRP 更快找到距

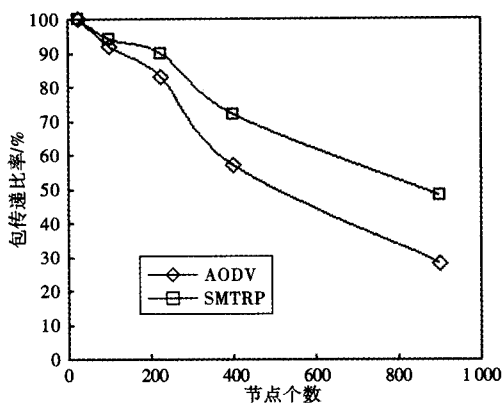


图1 包传递比率

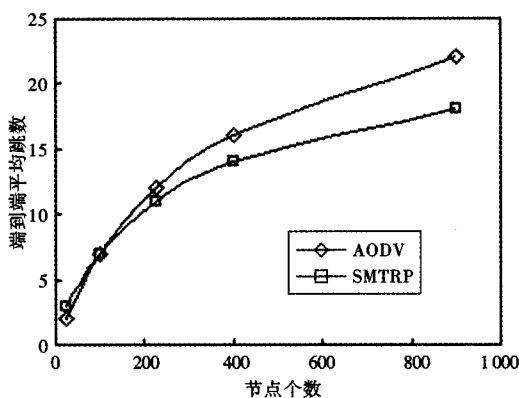


图2 端到端平均跳数

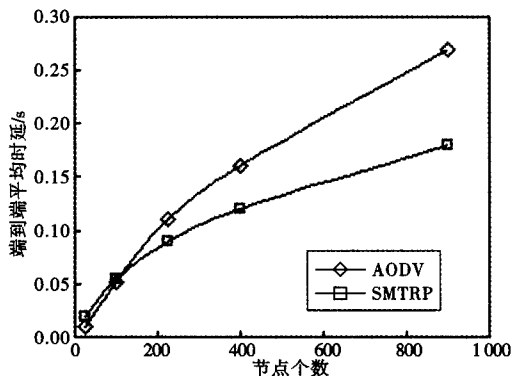


图3 端到端平均时延

离更短的路由路径。这是因为 AODV 是一种按需路由协议,对每一个要发送的数据包都进行路由查找过程,这样在规模不大时,总是能尽快的找到距离最短的路由路径。而 SMTRP 不管网络规模的大小,都要先进行建树和地址分配工作,采取向逻辑树上层发送的机制,这样就容易有路由绕行,而且对于小规模网络来说,也浪费了时间。随着网络规模的扩大,路由路径的变长,SMTRP 的这种弊端所占的比例越来越小,而 AODV 由于路由路径变长后的不稳定性,会严重影响到端到端的平均跳数,因此 SMTRP

在较大规模网络下比 AODV 更能显示其优越性。

随着网络的增大以及网络中不稳定节点的增多,每跳平均延迟必然延长,如图4所示。除了在25个节点规模的网络中 SMTRP 的每跳延迟大于 AODV 的网络,其他四个场景规模的网络中,SMTRP 的性能都明显好于 AODV。这是 SMTRP 省去路由发现以及路由修复过程的带来的明显好处。

图5中描述了两种路由协议在五种网络场景下的端到端通信比特率。同样在网络超过200个节点时的较大规模时,SMTRP 比 AODV 有更高的端到端通信比特率,但差别不大。

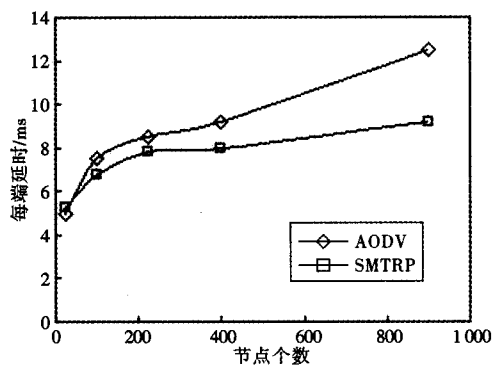


图4 每跳平均延迟

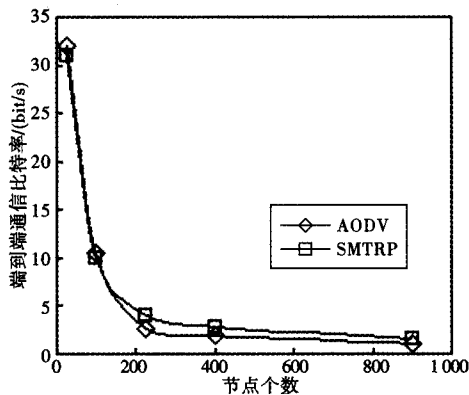


图5 端到端通信效率

5 结束语

本文借鉴核心树的逻辑拓扑结构和自适应地址块分配机制,建立了 Mesh 适配树的逻辑拓扑结构,提出并实现了可扩展的无线 Mesh 适配树路由协议 SMTRP,该协议通过将网络拓扑反映在逻辑地址上,然后根据每个节点所维护的邻居节点状态列表,可以判断出一个数据包该往哪个方向转发。在理论

(下转第94页)

表 1 系统重构时间对比

	Static	RM1	RM2
比特流大小/kB	1 532	143	246
重构时间/ms	4 398	339	604
重构时间比	1	0.077	0.137

7 结束语

本文完成了基于可重构平台的外部总线设计与实现,并给出了一个动态可重构的实例.该总线结构简单,易于实现,可配置,并可以支持多个从设备,应用广泛等特点.可重构计算是未来多核处理器的一个发展趋势,在系统中可预先包含多个不同功能的配置文件,根据现场的需要进行相应的配置,实现部分动态可重构的功能,既节省了开发成本,又满足了一些特殊系统的设计要求,也体现出可重构 SoC 现场升级,灵活运用的设计理念^[6].该方案的提出,使系统的开发过程大为简化,用户只需要简单的步骤就可以完成新的系统或者对原来的系统进行升级,此开发过程可以满足军用,宇航领域对系统高密度、小型化、智能化需求,也适合航天、军用方面多品种少批量产品开发,有着广泛的应用前景.

参考文献:

- [1] Daniel Kriesten, Volker Pankalla, Ulrich Heinkel. An application example of a run-time reconfigurable embedded system[C]//2010 International Conference on Reconfigurable Computing, 2010:97-112.
- [2] 王刚,张激.基于 FPGA 的 SOPC 嵌入式系统设计与典型实例[M].北京:电子工业出版社,2009.
- [3] 李炳新,汪波,张辉.用于可重构计算的 FPGA 开发平台的研究[J].微电子学与计算机,2010,27(11):139-142.
- [4] Fabrizio Ferrandi, Marco D. Santambrogio, Donatella Sciuto. A design methodology for dynamic reconfiguration: the caronte architecture[C]//Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05). Italy: IEEE, 2005.
- [5] 许骏,晏渭川,彭澄廉.基于模块的动态可重构系统设计[J].计算机工程与设计,2008,29(8):1367-1369.
- [6] 漆锋滨,王珊珊,姜小成,等.可重构计算及可重构编译技术研究[J].微电子学与计算机,2007,24(10):36-38.

作者简介:

蔡一茂 男,(1986—),硕士研究生.研究方向为可重构计算.

(上接第 89 页)

上证明 SMTRP 的可扩展性之后,仿真实验表明,新路由由协议相比传统的协议节省了大量存储空间,并且减少了通信开销和能量消耗.

参考文献:

- [1] Akyildiz I F, Wang X. survey on wireless mesh networks [J]. IEEE Communications Magazine, 2005 (43): 23-30.
- [2] 丘小兰,李陶深,葛志辉.无线 Mesh 网络多网关选播 QoS 路由模型研究[J].微电子学与计算机,2010,27(8):116-118.
- [3] Jain K, Padhve J, Padmanabhan V, et al. Impact of interference on multi-hop wireless network performance [C]//Proceedings of the 9th Annual International Conference on Mobile Computing and Networking.

New York: ACM, 2003:66-80.

- [4] 杨孟珂,杨亚涛,白中英.基于 HWMP 协议的无线 Mesh 网络多网关路由协议研究[J].微电子学与计算机,2009,27(12):4-7.
- [5] Hester L, Huang Y, Allen A, et al. NeuRFon-net-form: a self-organizing wireless sensor network [C]//Proceedings of the 11th IEEE ICCCN Conference. Miami, Florida:IEEE, 2002:364-369.
- [6] Zheng J, Lee M J. A resource-efficient and scalable wireless mesh routing protocol [J]. Ad Hoc Networks, 2007: 704-718.

作者简介:

王江涛 男,(1978—),博士,高级工程师.研究方向为计算机网络.