

# 基于 DSDV 的无线 Mesh 网络跨层路由设计

张 博<sup>1</sup>, 罗卫兵<sup>2</sup>, 贾 方<sup>1</sup>

(1. 武警工程学院研究生管理大队, 西安 710086; 2. 武警工程学院通信工程系, 西安 710086)

**摘 要:** 在 DSDV 协议的基础上, 提出一种跨层路由设计的方法, 选择一条具有合理长度和最小重传概率的路由进行测试。该链路质量机制可应用到基于 IEEE802.11x 的无线 Mesh 网络中, 无须对现有的 MAC 层进行修改, 即可作为一个单独模块加入到现有协议。在多跳网络中使用该跨层方案进行路由选择, 能节省系统开销, 优化系统的整体性能。

**关键词:** 无线 Mesh 网络; 跨层设计; 路由协议

## Wireless Mesh Networks Cross-Layer Routing Design Based on DSDV

ZHANG Bo<sup>1</sup>, LUO Wei-bing<sup>2</sup>, JIA Fang<sup>1</sup>

(1. Post-graduate Management Team, Engineering College of APF, Xi'an 710086;

2. Department of Communication Engineering, Engineering College of APF, Xi'an 710086)

**【Abstract】** Based on DSDV protocol, an approach of cross-layer routing design is proposed. A reasonable length and the smallest probability of re-routing has been chosen and tested. This is a quality link mechanism which can be applied to Wireless Mesh Networks(WMN) based on IEEE802.11x. It does not require amendments of the existing MAC layer and can be added to the existing protocol as a separate module. This paper uses the cross-layer routing design to choose the routing in multi-hop networks, which can save the system expenditure and optimize system's integrated performance.

**【Key words】** Wireless Mesh Networks(WMN); cross-layer design; routing protocol

传统网络协议层需要彼此透明以保持各层协议设计的独立性, 但在无线 Mesh 网络(WMN)中, 无线环境的不确定因素, 导致无线信道容量和误码率等时变特性, 传统分层设计方法无法保证网络的最佳资源利用率和用户业务的 QoS 需求。为改善无线网络的性能, MAC、路由协议和传输协议, 甚至应用层必须与其下各层进行有效的信息交互, 以适应其各层特性的变化。

### 1 路由协议跨层机制的基本要求

#### 1.1 跨层设计原理

跨层设计是一种综合考虑协议栈各层次设计与优化, 并允许任意层次和功能模块之间自由交互信息的方法。在原有的分层协议栈基础上, 集成跨层设计与优化方法, 可得到一种跨层协议栈<sup>[1]</sup>。在分层设计方式中, 多个层须进行重复计算和交互来得到一些其他层次很容易得到的信息。跨层设计与优化的优势在于通过使用层间交互, 不同的层次可充分利用网络资源, 达到系统总吞吐量最大化、总传输功率最小化、QoS 最优化等目的。跨层设计的理论模型如图 1 所示<sup>[2]</sup>。

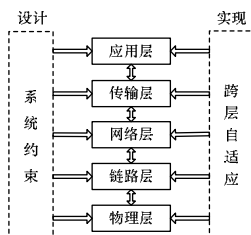


图 1 跨层设计理论模型

从跨层设计的角度看, 有必要考虑层与层之间的依赖关系, 加强层与层之间的信息交互和共享, 从而使网络得到整体优化。图 2 为各层间跨层信息交互过程<sup>[2]</sup>。

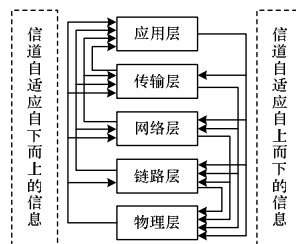


图 2 各层间跨层信息交互过程

#### 1.2 路由协议设计对跨层设计的要求

采用传统的最短路径准则设计无线 Mesh 网络的路由协议, 不足以构造一个良好的路由, 即传输时延、吞吐量和可靠性等性能无法达到理想的指标。原因在于最短路径准则没有考虑到其下物理信道特性的变化对 MAC 层接入性能的影响等因素, 以及上层对 QoS 的要求, 造成所选路径无法适应底层性能的变化, 也可能造成传输层性能的较大变动。此外, 就无线信道的特点而言, 即使信道环境在通信期间没有产生变化, 最短路径也未必意味着就是最优路径。在同样的 BER 条件下, 传输距离越长, 所支持的数据传输速率就越低, 即

**作者简介:** 张 博(1980 - ), 男, 硕士研究生, 主研方向: 无线数据通信; 罗卫兵, 教授、博士; 贾 方, 硕士研究生

**收稿日期:** 2008-06-10 **E-mail:** 49329986@qq.com

长距离的最短路径比短距离的非最短路径的传输速率或吞吐量低<sup>[2]</sup>。因此,须为 WMN 设计一种新的能满足网络业务 QoS 需求的路由准则。

## 2 基于 DSDV 的 WMN 跨层路由设计

无线 Mesh 网络路由协议的跨层设计有时也被称为路由协议的自适应设计。在无线网络中,传输媒介的变化以及干扰的不可预测性,导致了网络性能在时间和空间上的剧烈波动。路由协议的跨层设计打破现有的网络分层结构设计理念,将分散在各层的网络信息通过跨层管理模块传递到 WMR(Wireless Mesh Routers)层,可最大程度地挖掘无线网络的潜在优势。路由协议的跨层设计主要是以全局的观点看待 MAC 物理层的一些参数和信息,甚至允许路由协议去控制其他一些协议层的参数设置,从而实现更加可靠、更加高效的路由路径。

### 2.1 DSDV 路由协议机理

DSDV,即目的节点序列距离矢量路由协议,是一种典型的先应式路由协议。协议采用最短路径法则,每个节点维护一张包含到达节点的路由信息的路由表,并根据每个节点周期性的广播消息来更新路由表,以适应网络的拓扑变化<sup>[3]</sup>。协议中使用目的端顺序号,避免因使用过时的路由信息而产生无效的路径(包括路由环路和中断的路径)。每一个目标节点的路径记录对应一个目的地序列号,这个序列号由目标节点产生。当节点从邻居节点的距离/向量报文中得到某一目标节点的另一个路由记录时,若新记录的顺序号比已有的记录新,则节点使用新路由记录;若 2 个路由记录的顺序号相同,则节点使用路径长度更短的记录;否则,目标节点的记录不变。其特点如下<sup>[4]</sup>:

(1)获取路由的延时小,较适合具有实时要求的应用。

(2)引入目的节点序列号,既能区别路由的新旧,又能有效避免路由环路的产生。当节点收到多个不同的矢量表数据包时,采用序列号较大,即较新的路由来计算,如果序列号相同则看谁的路径更短。

(3)每个节点都维护一个路由表,路由表的每条路由都有一个目的节点地址、到达目的节点所需的跳数以及从该目的节点收到的序列号。

(4)路由表有 2 种更新方式:1)时间驱动的更新(全部更新),是周期性的,拓扑更新信息包括整个路由表。这样的更新方式可以让新加入的节点及时了解网络拓扑,适用于拓扑变化快的情况。2)事件驱动的更新(逐步更新),只有当路由发生改变时才触发更新,仅包含上次对整个路由表进行广播后又更新的内容。这种方式的更新能及时反映拓扑的变化,适用于网络变化缓慢的情况。

DSDV 可有效减少端到端的时延,从一定程度上满足各种应用对 QoS 的要求。而且 DSDV 触发路由更新的方式既有时间驱动,又有事件驱动,各节点的路由表能在较短时间内达到较为稳定的状态,因此,考虑将 DSDV 作为 WMN 中无线路由器与其覆盖区域内用户节点间的路由协议。同时,WMN 网络中的节点也不会像 MANET 中的节点那样具有较高的移动速率,所以定期更新的时间间隔可较长。如果能在 DSDV 的路由表存储到达同一目的节点的多条路由,能在一定程度上提高数据分组的发送成功率。

### 2.2 跨层路由设计方案

本文在文献[5]的基础上加以调整。在信息传递的过程中,由于会发生帧丢失或者网络拥塞的情况,节点在发送

DATA 帧后,如未收到回应的 ACK 或在发送 RTS 后没有收到返回的 CTS,就会发生重传。因此,对应 MAC 层每次成功传输的 DATA 和 RTS 的重传概率代表链路的质量和拥塞情况。如果移动节点在媒介接入时使用了 DCF 机制,在发送 DATA 之前会有 RTS/CTS 交互阶段来“预订”信道,避免出现隐藏/暴露终端的问题。在“预订”信道后,如节点发送 DATA 帧后没有收到对应的 ACK 帧,就判定帧丢失是由信道状况较差造成的。如不使用 RTS/CTS 交互,DATA 帧的丢失可认为是因为差的信道状况或发生了冲突。

本文使用 IEEE802.11 管理信息库(MIB)中的 ACKFailureCount 和 RTSFailureCount,它们分别代表 DATA 帧和 RTS 帧的重传次数,以避免对 MAC 进行修改。由于 IEEE802.11 中广播的分组无须进行回复,它们不能用来衡量帧重传概率,因此只能利用单播分组的传输才能进行这种测量。结合类似指数加权移动平均数(EWMA)的方法,在周期  $T$ (在 DSDV 中路由表的更新周期)内使用帧传输成功率(FTE)移动平均数,并根据网络的对最新的 FTE 和此前的平均 FTE 指定不同的权重,这样还能避免由于帧传输成功率在无线链路中的起伏较大而造成所选路径变化过快的情况。图 3 为使用 RTS/CTS 帧交互“预订”信道的 MAC 层帧传输过程<sup>[5]</sup>。

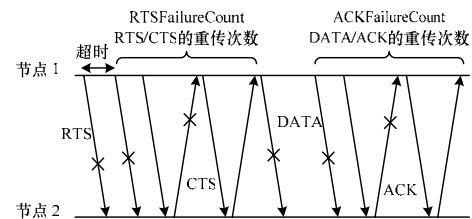


图 3 MAC 层帧传输过程

以 RTS 为例,DATA 帧的重传过程与 RTS 帧类似。RTS 的重传次数代表节点 1 和节点 2 之间信道竞争的程度,也代表对它们之间链路质量的估计。当节点 1 发送 RTS 帧要求“预订”从节点 1 到节点 2 的信道后,它会等待一个特定的时间段。若在此时间段内没有收到对应的 CTS,节点 1 就会重发 RTS,直到收到来自节点 2 的 CTS,此时它就获得了一个无冲突的信道。各节点都使用 MIB 变量来得到某条链路上对应成功传输一个 DATA 帧的重传次数,以计算出该条链路的 FTE。假设对应从节点 1 到节点 2 成功传输第  $i$  个分组的重传次数为  $Failure_{12}(i)$ ,则  $Failure_{12}(i)$  实际上就是 DATA 和 RTS 的重传次数的和,其值为

$$Failure_{12}(i) = ACKFailureCount_{12}(i) + RTSFailureCount_{12}(i) \quad (1)$$

在理想情况下,对 DATA 和 RTS 仅有一次传输。因此,从节点 1 到节点 2 的 DATA 帧的传输成功率  $FTE_{12}(i)$  的计算公式为

$$FTE_{12}(i) = 1 - \frac{Failure_{12}(i)}{Failure_{12}(i) + 2} \quad (2)$$

该参数是成功发送 DATA 和 RTS 帧的概率。重传次数越大, FTE 的值越小;当没有发生帧丢失时,重传次数为 0,  $Failure_{12}(i)$  为 1,即从节点 1 到节点 2 的分组 100% 被发送成功。当重传次数超过一个预设的阈值时,发送者将停止发送分组,此时  $Failure_{12}(i)$  的值为 0,这将在相当程度上减小了传输成功率的移动平均值<sup>[5]</sup>。

在广播路由表的周期  $T$  内,节点 1 对计算出来的  $FTE_{12}(i)$

取  $EWMA$  值,代表节点 1 和节点 2 之间的链路质量和拥塞情况。该平均值为

$$\overline{FTE}_{12}(i) = (1 - \alpha_i) \times FTE_{12}(i-1) + \alpha_i \times FTE_{12}(i) \quad (3)$$

其中,  $\overline{FTE}_{12}(i)$  代表在传输第  $i$  个分组时  $FTE$  的移动平均值,  $\alpha_i$  的取值取决与信道变化的速度。计算出的  $FTE$  被设定了一个较小的权值 5%, 而之前平均  $FTE$  的权值为 95%, 即  $\alpha_i$  取值 5%。如果信道状态不稳定, 可加大权值到 20%, 进而减小由于无线链路的瞬态特性给网络带来的波动。

计算时还应考虑到跳数的影响, 以避免选择一条跳数太大的链路, 将该参数记为  $HopMetric$ 。选取一个从节点 1 到目的节点 6 的多跳路径, 计算方法如下:

$$HopMetric_{16} = 1 - \frac{HopCount_{16}}{NodeNum} \quad (4)$$

其中,  $HopCount_{16}$  是当前从源节点 1 到目的节点 6 的最小跳数, 这一参数在标准 DSDV 中已经存在, 无需额外计算;  $NodeNum$  是网络中最大的节点个数<sup>[5]</sup>。

最后将  $\overline{FTE}_{16}(i)$  和  $HopMetric_{16}$  结合为一个参数, 以选择出到达目的节点的最佳路径, 即

$$RouteMetric_{16} = \overline{FTE}_{16}(i) \times HopMetric_{16} \quad (5)$$

### 2.3 路由测试

测试地点为西安市西郊某地, 设置 8 个 Mesh 点进行测试, 包括视通 (LOS) 和非视通 (NLOS) 环境。在 0.5 km ~ 1.5 km 范围内建立测试点。

测试设备: 8 个 802.11g 功能的 Mesh 基站。

测试拓扑图如图 4 所示。

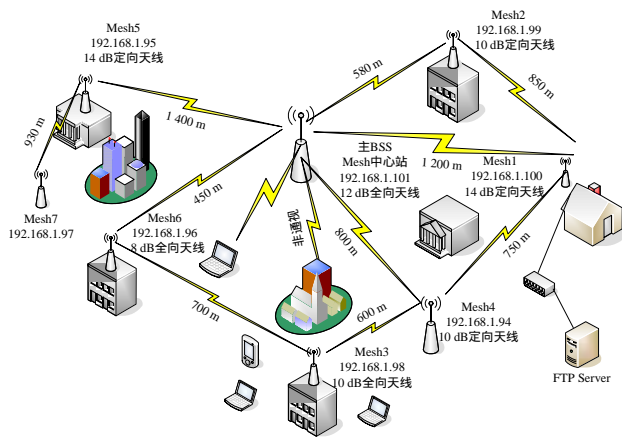


图 4 拉距测试拓扑

测试方法: 在 Mesh1 节点接入 1 台 FTP 服务器提供 24 MB 带宽下载测试。各个节点间的连通性采用 Ping 进行测试, 观察结果; 下载数百兆文件进行加压测试, 观察连接情况; 多节点同时下载, 观察连接情况。使用 Chariot 5.0 测试节点的实际带宽, NetStumbler 0.4.0 测试无线信号的质量。测试结

果见表 1。

表 1 路由测试结果

节点编号	距离/km	跳数	平均下载速率 ( $MB \cdot s^{-1}$ )	压力测试	信号质量	连通状态
主BSS	1.20	1	22.3	通过	好	连通
Mesh1	0.00	0	23.5	通过	好	连通
Mesh2	0.85	1	22.7	通过	好	连通
Mesh3	1.35	2	20.3	通过	好	连通
Mesh4	0.75	1	22.9	通过	好	连通
Mesh5	2.60	2	18.8	通过	良	连通
Mesh6	2.10	3	15.6	通过	良	连通
Mesh7	3.50	3	9.7	通过	可用	连通

### 2.4 桥接测试

测试 2 个 Mesh 站点远距离桥接的连通性。在 2 个制高点相距 13.70 km 测试(GPS 距离), 使用 24 dB 零风阻栅格抛物面天线。

测试设备: 2 个 802.11g 功能的 Mesh 基站; 2 副 24 dB 栅格抛物面天线。

测试方法: 2 个节点间的连通性采用 Ping 进行测试, 观察结果。使用 Chariot 5.0 测试连接带宽, 取 5 次测量值平均。

测试结果: 连通性测试, 通过; 平均带宽为 33.6 MB。

### 3 结束语

跨层机制在进行路由表更新时, 采用了一种链路质量机制, 节点会综合考虑跳数和链路质量, 选择更好的路由来更新到达某一目的节点路由的下一跳。测试结果表明, 在无线 Mesh 网络中使用上述跨层路由算法, 考虑到重传次数和帧成功传速率, 改进了传统的 DSDV 协议, 虽然增大了路由算法的复杂程度, 却使整个网络的吞吐量有了很大的提升, 减少了处理和通信开销, 缩短了端到端的时延, 优化了系统的整体性能。

### 参考文献

- [1] Iannone L, Khalili R, Salamatian K, et al. Cross-layer Routing in Wireless Mesh Networks[C]//Proc. of the 1st International Symposium on Wireless Communication System. [S. l.]: IEEE Computer Society Press, 2004.
- [2] 方旭明. 下一代无线因特网技术: 无线 Mesh 网络[M]. 北京: 人民邮电出版社, 2006.
- [3] Pham P P, Perreau S, Jayasuriya A. New Cross-layer Design Approach to Ad Hoc Networks Under Rayleigh Fading[J]. IEEE Journal on Selected Areas in Communications, 2005, 23(1): 28-39.
- [4] 张鑫, 张顺颐, 孙雁飞. 无线 Mesh 网络中路由算法的研究[J]. 江苏通信技术, 2007, 23(4): 6-9.
- [5] 张勇, 郭达. 无线网状网原理与技术[M]. 北京: 电子工业出版社, 2007.

编辑 金胡考

(上接第 63 页)

- [14] Kargupta H, Huang Weiyun, Sivakumar K, et al. Collective Principal Component Analysis From Distributed, Heterogeneous Data[C]//Proc. of PKDD'00. Freiburg, Germany: [s. n.], 2000: 452-457.
- [15] 唐懿芳, 牛力, 张师超. 多数据源关联规则挖掘算法研究[J].

广西师范大学学报: 自然科学版, 2002, 20(4): 27-31.

- [16] Zhang Shichao, Zhang Chengqi, Wu Xindong. Knowledge Discovery in Multiple Databases[M]. London, UK: Springer-Verlag, 2004.

编辑 金胡考

