

PaperPass检测报告简明打印版

比对结果（相似度）：

总体：8 %（总体相似度是指本地库、互联网的综合比对结果）

本地库：8 %（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）

互联网：0 %（互联网相似度是指论文与互联网资源的比对结果）

编号：5730857B95A4FH4DE

标题：无线Mesh视频传输网络QoS优化技术研究

作者：王锋

长度：37695 字符(不计空格)

句子数：1047句

时间：2016-5-9 20:41:31

比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源

查真伪：<http://www.paperpass.com/check>

句子相似度分布图：



本地库相似资源列表（学术期刊、学位论文、会议论文）：

没有找到与本地库相似度高的资源！

互联网相似资源列表：

没有找到与互联网相似度高的资源！

全文简明报告：

无线Mesh视频传输网络QoS优化技术研究

{ 43 %：无线 Mesh 网络作为一种新兴的无线网络技术，以其特有的优势正在对人们的生产生活产生日益显著的影响。} 传统的 AP 网络设备，在为移动设备提供无线接口的同时，需要以有线形式接入外部网络，极大的限制了 AP 网络的灵活性和覆盖范围。另一种更加泛在的无线网络-3G/4G 网络，虽然能够提供足够的网络覆盖范围，但网络基础设施部署和维护成本极高。{ 42 %：在这样的背景下，无线 Mesh 网络以其优秀的灵活性、易部署等特性得到了广泛的尝试。} { 40 %：目前，无线 Mesh 网络在校园、城市、野外、救灾、油田等很多场景下得到积极的推广和应用。} 但在提供便捷、灵活的网络接入方式的同时，也因其高度动态的路由机制带来了很多网

络应用上的不足，比如无线链路的不稳定性，严重的信号间干扰等，网络拓扑不稳定等。无线 Mesh 网络在整体协议栈上缺乏 QoS 保障机制，MAC 技术依赖 802.11 系列协议设定，但又无法直接引入 802.11e 的 QoS 保障技术。

{ 41 % : 本文的主要工作集中在解决无线 Mesh 网络在视频监控场景下的 QoS 保障所面临的挑战，所提出的技术方案在无线 Mesh 网络实际部署中亦具有指导意义。 }

本文的创新点主要有:

大规模无线 Mesh 网络实践; 跨层的 QoS 保障技术。

关键词: Mesh 网络; 视频监控; QoS; 正交信道

Abstract

Abstract

An abstract of a dissertation is a summary and extraction of research work and contributions. Included in an abstract should be description of research topic and research objective , brief introduction to methodology and research process , and summarization of conclusion and contributions of the research. An abstract should be characterized by independence and clarity and carry identical information with the dissertation. It should be such that the general idea and major contributions of the dissertation are conveyed without reading the dissertation.

An abstract should be concise and to the point. It is a misunderstanding to make an abstract an outline of the dissertation and words “ the first chapter ” , “ the second chapter ” and the like should be avoided in the abstract.

Key words are terms used in a dissertation for indexing , reflecting core information of the dissertation. An abstract may contain a maximum of 5 key words , with semi-colons used in between to separate one another.

Key words: Mesh Network; video surveillance; QoS; orthogonal channel

1.1 背景及意义

第1章 引言

第1章 引言

{ 41 % : 自1970年第一次无线数据通信演示始[5] , 无线设备迅速渗透到人们的生产生活的方方面面 , } { 41 % : 小到嵌入式微处理器、移动设备 , 大到交通工具、航空航天设备都囊括在内。 } 在人们的日常生活中 , 最常见的无线网络就是 AP 网络和 3 G/4 G 以及正在兴起的 5 G 网络 , 前者提供有限覆盖范围和低灵活度的网络接入 , 但优点是部署成本低廉 , 且网络性能稳定; 后者极大的扩展了无线网络的覆盖范围 , 方便用户使用 , 但也存在部署和维护成本高的缺点。如今伴随着工业 4.0 的兴起 , 越来越多的传统设备智能化 , 另一方面智能终端越来越普及 , 人均入网设备快速增长。无线网络的接入需求正在以惊人的速度增长。这就带来迅速增长的无线接入需求和不完善的无线网络支撑体系之间的矛盾。 { 46 % : 面对这一困境 , 无线 Mesh 网络作为一种可能的无线网络解决

方案受到工业界的重视。} {45%：相对于传统的无线接入方式，无线 Mesh 网络具有自组织、自治愈、网络拓扑灵活、部署成本低廉等特性。} 这些特性使得无线 Mesh 网络在灾后恢复、缺少有线基础设施的大型建筑物或者区域、蜂窝网络无法良好覆盖的区域等场景下具有无可比拟的优势。

基于上述背景，本文工作的主要内容为基于无线 Mesh 网络设计实现部署灵活、低成本、高性能的大型工业视频监控系统，该系统将在缺乏网络基础设施的野外油田等项目中部署并长期运行。本文工作对无线 mesh 网络系统的大规模部署应用提供实际的借鉴意义，探索了其中 QoS 保障、性能优化、网络整体规划等核心问题，并提出有效的优化方案。 {45%：同时对于工业视频监控网络提供一种新的高效的设计思路。}

1.2 主要工作

本文基于室内大型实验床和室外实际运行的无线 Mesh 网络系统进行了大量的实验，在实验过程中发现已有系统在 QoS 保障方面的不足，并针对这些问题设计实现优化方案。同时开发了一系列网络管理配置相关的工具，并深入 Mesh 网络各层进行以提升系统整体 QoS 为目标的优化。本文主要工作现罗列如下：

1. 采用业界较为广泛使用的一种无线 Mesh 网络协议 – batman-adv，基于 linux 系统和 Mikrotik 硬件平台配置实际运行的 Mesh 节点设备，在此基础上搭建规模化的无线 Mesh 网络系统。

第1章 引言

2. 以 1 为基础，进行大量实验，从实验中发现已有 Mesh 网络在 QoS 保障方面的不足。归纳为三点：无线信道的开发性导致的无法规模化部署、视频监控场景下的视频大数据导致的网络带宽紧张、设备移动时路径切换时延过长。

3. 针对 2 中的问题设计实现子网信道隔离的部署方案、跨层视频帧权重差分技术、路径质量敏感的动态切换阈值算法。

4. 为了提供对于大规模 Mesh 网络的管理、诊断、可视化等功能，开发了一系列工具： {47%：包括静态路由设定、外接显示模块、基于 web 端的可视化和诊断系统等。}

5. 在伊拉克哈法亚油田等实际项目中部署项目系统，长期运行并监控维护。

1.3 文章组织结构

{45%：本文第一章引言部分介绍文章的背景和意义，并介绍文章的主要工作和贡献。} 第二章介绍该领域的相关工作，面临的挑战。 {42%：第三章介绍项目软硬件平台、项目设计和相关的知识。} 第四章详细叙述项目中的三个核心创新点的设计实现及带来的性能提升。第五章总结全文并展望后续工作。

本章首先介绍 Mesh 网络架构，对现有的工业界和学术界的 Mesh 网络实验平台和项目做综述，然后总结它们的贡献、优势和缺陷。然后介绍 QoS 在 802.11 协议族中的基本实现机制，并讨论在 Mesh 网络中应用 QoS 保障面临的困难。 {43%：最后聚焦在 Mesh 网络基础上的视频传输 QoS 保障技术的研究应用。}

2.1 无线 Mesh 网络

在存在基础管理结构的无线网络中，路由信息通常由中央控制的角色提供，如 AP 网络中的 AC。这样可以实现网络资源的全局调度和管控。相反，在 ad-hoc 模式下，不存在中央控制的角色，网络中的路由构建由参与网络的节点设备自发交互完成。 { 41 % : 作为典型的 ad-hoc 模式的无线网络，无线 Mesh 网络中的节点之间通过交互链路状态和局部路由信息从而构建起全局的网状网络架构。 }

无线 Mesh 网络由 Mesh 网关，Mesh 路由器，Mesh 客户端组成 citemeshsurvey。 { 41 % : 其中，Mesh 路由器提供路由构建功能，Mesh 网管提供异构网络接入的功能，通常 Mesh 网关同时也是 Mesh 路由器。 } Mesh 客户端为 Mesh 网络的用户。图 2.1 给出了 Mesh 网络的网络架构示例。

图 2.1 无线 Mesh 网络架构图

{ 61 % : 无线 Mesh 网络是具有广阔应用前景的下一代无线网络技术。 } 其自组织、移动 3

{ 41 % : 和灵活的网络构建使其能够适用于很多传统网络无法覆盖的场景，比如: } 救灾、野外、战场等临时性应用以及其他缺乏有线网络基础设施，并且可以作为 3G/4G 网络的一个很好的延伸。 { 47 % : 无线 Mesh 网络的主要优势体现在快速、低成本的部署和大范围的网络覆盖。 }

{ 49 % : 在过去的十多年里，伴随着无线 Mesh 网络的研究热潮，很多的项目开发了不同的 Mesh 网络路由协议， } 802.11 协议簇也将 Mesh 网络的标准制定纳入讨论并以提出完善的修正案 802.11 s [6]。该修正案中除了提出一种推荐的 Mesh 组网协议，同时保留组网协议的灵活性，规定可以替换其他 Mesh 组网协议。各种不同的路由协议仍然以各自的社区为依托不断发展完善。

2.2 无线 Mesh 网络中的路由技术

{ 40 % : 在无线 mesh 网络中，当节点互相不在信号覆盖范围内时，就需要中继节点代为转发。 } 很多的 Mesh 网络协议可以完成这一功能，本节就描述其中一些典型的协议以及它们之间的差异。

{ 43 % : 无线 Mesh 网络通常采用分布式路由，即网络中的节点分发并采集其他节点的局部路由信息。 } { 48 % : 然后，每个节点根据收集到的网络路由信息，决定到达目的节点的最佳路径。 } { 53 % : 分布式路由协议又可以分为主动式和反应式两种不同的类别。 } { 48 % : 主动式路由协议中，节点周期性的广播自己的存在，并携带自己所知道的局部路由信息; } { 64 % : 反应式路由协议则在需要发送数据的时候，即时获取路由信息。 } { 46 % : Mesh 网络的架构和无线介质的特殊性给路由构建带来了一些特殊的挑战。 }

- 竞争使用共享的无线信道会限制网络的性能。

- 用于网络构建的数据包造成额外的开销。

{ 48 % : - 需要引入漫游机制解决移动节点的接入问题。 }

{ 41 % : - 当网络中一个节点失效，可能导致多条路由随之失效。 }

各种不同的协议采用不同的方式解决这些问题，但殊途同归，最终的目标都是最小化网络构建的额外开销，同时保证最大化网络吞吐量、网络的性能，保持连接的有效性。

2.2.1 主要路由技术

{ 54 % : 如前所述, Mesh 网络中的路由技术可以分为主动式路由和反应式路由。 } { 45 % : 如果路由信息的收集和路由的计算在节点需要发送数据时才进行, 则该路由方式成为反应式路由。 } 反之, 如果网络的信息分布式存储在网络中的节点中, 且每当网络中的状态发生变化, 该变化会即时广播全网络, 相关的节点即时更新自己的路由

第2章 相关工作

{ 51 % : 信息, 这种路由方式称为主动式路由。 }

{ 41 % : 大多数 Mesh 路由协议收集信息和判定路由时以如下两种方式为主要判定 }

依据: 链路状态和距离向量。 { 47 % : 基于链路状态的路由, 节点将自己相邻链路信息组织成有向图的形式, 洪泛 }

全网。 每个节点可以收集到其他节点的局部链路拓扑和质量信息, 并基于此构建整个网络的拓扑信息, 并基于不同拓扑路由的权重计算出最短路径, 即发送数据时的路由。

{ 45 % : 基于距离向量的路由, 节点仅知道目标节点数据需要送达的下一跳节点, 即数据发送的方向。 } { 41 % : 而最佳的下一跳节点的选择则基于到达目标节点的总跳数和每一跳的链路质量。 } { 42 % : 距离向量路由方式不需要计算完整的网络拓扑, 因此消耗的额外开销更少。 } 缺点是, 得到的路由通常不是全局最优的。

{ 43 % : 分布式路由需要处理网络中的路由环路现象。 } 路由环路通常出现在两个或者更多的节点之间, 路由环路会导致数据陷入环路, 造成额外的网络开销, 且数据始终无法送达最终的目的节点。

2.2.2 优化的链路状态路由 (OLSR)[1]

{ 45 % : 优化的链路状态路由 (OLSR) 广泛的应用于多种基于嵌入式 Linux 平台的 Mesh 网络中。 } 通常作为守护进程运行在网络层, 属于主动式路由, 且基于链路状态做路由决策。

在最初的 OLSR 实现中, 网络拓扑图的计算很慢且经常在网络状态发生更新时失效。 路由变化频繁, 易形成路由环路, 造成网络的低可用性。 为了解决这些问题, OLSR 引入了信的路由决策参数--传输次数期望 (ETX)。 新的参数带来了性能的显著提升, 但仍然没有解决路由环路的问题。 通过增加拓扑控制数据包的发送频率可以一定程度上降低路由环路的形成次数。 由此造成的额外的路由开销可以通过限制洪泛距离予以限制--仅向三跳之内的邻居节点洪泛路由变化信息, 这种方法在实践中被证明是有效的。 这种技术称为鱼眼机制 (Fish Eye), 该技术保证了 OLSR 的可用性, 但目前还不能完全消除路由环路现象。

2.2.3 Ad-hoc 网络需求驱动的距离向量路由协议 (AODV)[2]

{ 58 % : Ad-hoc 网络需求驱动的距离向量路由协议 (AODV) 是一种基于距离向量的反应式路由协议。 } { 43 % : 在该路由协议中, 当有节点需要发送数据到目的节点, 就先向网络广播路由请求数据包, 通过网络中其他节点包括目的节点的相应, 构建瞬时的有效路由。 } { 42 % : 当网络处于相对空闲状态时, 该协议最大程度上减少了额外的路由开销。 }

{ 43 % : 第2章 相关工作 该协议在对能耗要求极高的传感器网络中具有很好的应用价值。 }

2.2.4 针对移动 Ad-hoc 网络的路由协议 (BATMAN)[3]

针对移动 Ad-hoc 网络的路由协议 2006 年诞生, 最初作为 OLSR 的替代者, 属于基于距离向量的主动路由协议。 在最初的三个版本中, BATMAN 引入了许多特性, 包括: 不对称链路, 多接口支持等。 { 42 % : 类似于 OLSR, 该协议同样作为用户层的守护进程运行网络层。 }

2.2.5 BATMAN 升级版 (BATMAN-adv)[4]

因为 BATMAN 运行再用户层所带来的性能问题, BATMAN-adv 在 2007 年开始投入研发。 { 52 % : 因为运行在内核态, 新的协议节省了大量的内核态和用户态之间拷贝数据包的开销。 } { 43 % : 另一方面, 新的协议运行在链路层, 用 mac 地址做路由。 }

BATMAN-adv 在链路层对数据做封装, 对网络层透明, 因此可以兼容其他的网络层协议, 也使得其他特性的加入更加容易。 现在的 BATMAN-adv 支持非 Mesh 设备的桥接和漫游。

2011 年, BATMAN-adv 已经添加进 Linux 内核开发主线树。 2.3 802.11 协议簇 QoS 支持

{ 50 % : 802.11 标准族制定对无线局域网的 QoS 保障的标准 802.11e [7]。 } { 43 % : 802.11 e 将 mac 层数据按照数据的业务类型划分为四个不同的优先级队列, 每一个队列在竞争使用无线信道是具有不同的优先级, } { 42 % : 优先级通过竞争窗口大小的设置从而控制竞争成功的概率实现。 } { 48 % : 在第四章中将对该优先级队列的机制详细介绍。 }

2.4 无线 Mesh 网络信道分配相关研究

{ 44 % : Ashish Raniwala 等人在 2004 年就进行了无线 Mesh 网络信道分配方面的研究工作 [8]。 } 作者设计了多网卡接口 Mesh 路由设备, 并基于该设备进行组网建模。 另外提出了一种对干扰的评估方案。 最终通过对信道干扰进行建模, 然后以最小化 Mesh 路由节点之间的干扰为优化目标, 最终导出启发式的信道分配最优化算法。 实验结果显示所提信道分配方案相对于传统分配方法取得了显著的性能提升。

文章 [9] 中, Hisham 针对基于认知无线模块的无线 Mesh 网络的存到分配问题, 尝试寻找一种能够提供链路级别 QoS 保障, { 40 % : 最大化网络覆盖, 同时减少控制信道需求的信道分配方案。 } { 47 % : 并提出了一种基于接收者的信道分配方案同时满足前述目标。 }

上述工作都是基于特定的硬件平台, 或者是支持多网卡接口或者是基于认知无线电模块技术, 但目前实际商用的无线设备并不具有这样的特性。 相关的工作还有 [10] [11] 提出了类似 [8] 但从最小化干扰的角度触发的思路。 文章 [12] 从对干扰建模到路由 QoS 技术提出了一种综合的跨层方案。

2.5 无线 Mesh 网络视频传输相关研究

基于无线 Mesh 网络搭建高性能实时视频流传输系统方面已经有过诸多的探索性工作。

A. {41%: Rowe等人的文章提出了一种无线分布式实时监控系统 OmniEye[13], 提出在视频流应用的场景中, } 为了在摄像头数量增多时最大限度的保证图像质量, 需要提升网络的高可用性。实验发现, 使用标准 802.11 标准的 MAC 协议 DCF 信道接入机制, 当每个摄像头的视频流占用带宽在 1 Mbps 时, 系统最大可负载的摄像头数量大约为 5-6 个。当摄像头超过这个数量时, 就会出现严重的抖动现象影响视频质量。文章认为性能的显著下降来自于多跳场景下隐终端的影响。针对这一问题, 文章提出了一种时间同步的应用层 MAC 协议- TSAM, 该协议运行在 802.11 之上。TSAM 禁用了 802.11 的冲突退避机制, 并基于时分复用技术给每一个节点分配时间窗口, 从而消除了竞争, 保证无线信道的最大可用性。实验显示, TSAM 在控制端到端延时的同时通过顺序时间窗口保证了节点之间的同步。该工作对节点设备的时钟同步要求较高, 且需要修改现有 802.11 协议的冲突退避机制, 不能够很好的兼容现有协议。

Choumas 等人提出并特征化了一种针对视频流多播的机会路由算法[14], 该算法特别适用于两跳之内的无线 Mesh 网络。{49%: 文章通过分类不同的视频数据并赋予不同的权重, 从而提升 QoS。} 对于视频流数据而言, 延时是一个重要的考量的指标, 相比之下, 即使因此发生一定的丢包都可以接受。基于此, 文章提出 ViMOR, 一种视频流多播机会路由协议, 且聚焦于跳数小于等于两跳的拓扑结构。相比于 MORE[15], ViMOR 能够提升网络的吞吐量同时提升视频接受的质量。

文章[16]中, Le Dang 等人提出了一种挑选对等节点的参数-MSM(Multiplication Selector Metric)。该参数可以解决传统的基于假发的参数面临的两个局限: {61%: 瓶颈链路的识别和跳数计数。} 且可以和任意的链路质量感知参数协同而不需要额外的网络开销。在此基础上, 文章提出了一种跨层的基于无线链路质量感知的对等节点选择机制-WLO(Wireless Link quality-aware Overlay)。WLO 根据 MSM 参数的值在存有目标内容的对等网络节点中选择最优的节点。仿真也取得了不错的性能提升。

第2章 相关工作

Video Multicast over Wireless Mesh Networks with Scalable Video Coding(SVC)文章[17]中, {43%: 作者 Xiaqing Zhu等人致力于优化无线 Mesh网络中视频多播场景下的速率分配问题。} 构建了一个综合无线链路容量各向异性、相邻链路竞争、不同视频的误码率特征等的优化框架, 并提出了一种分布式的速率分配方案来最小化整体网络中传输视频的误码率, 而不给网络带来额外的网络开销。{50%: 仿真显示该方案要显著优于基于 TCP 的启发式算法。}

另外, 随着今年多无线模块的无线产品越来越普及, 围绕多无线模块的性能优化也吸引了很多的研究工作 [18][19]。文章[20][21][22]等通过不同的角度设计了不同的跨层方案试图整体上提升 Mesh 视频传输网络的性能。

2.6 移动场景下的无线 Mesh 网络 QoS 保障相关研究

E. M. Royer和 Chai- Keong Toh在1999年发表的文章[23]当时主流的几种移动 adhoc网络协议做了对比研究, 并提出了移动 Mesh网络的几点主要的衡量指标, 包括 QoS、性能、延时等。之后关于移动 adhoc 网络的研究蔚然成风, 吸引了很多研究者投入进来。

{52%: 文章[24]提出了广播风暴在移动 adhoc 网络中的危害。} 广播技术在网络中是一种常见的可以协助解决很多问题的技术。但是在移动 adhoc 网络中, 因为网络设备的移动性, 广播的频率会成倍放大以适应网络的动态变化。{47%: 而因为无线信道的开放性, 高频率的广播会占用额外的带宽资源。} 文章中, 作者提出了一些方法来减少冗余的广播数据, 并且差异化重广播的时间点, 从而减轻了广播风暴的影响。

Lee 等人在文章[25]中提出了一种基于 IP 的 QoS 框架 INSIGNIA，以支持移动 adhoc 网络中的适应性服务。该框架基于 IP 网络的灵活性、鲁棒性、可扩展性设计 INSIGNIA，使之能够支持快速回复、端到端适应等特性。

文章[26]聚焦于延时限制的拓扑通知问题，将延时和干扰相结合。 { 62 % : 并基于此提出了一种分布式跨层算法。 } 进一步的，作者还将节点移动性对控制算法的影响纳入了考虑，删除了一些不稳定的拓扑链接。仿真结果显示，该算法相对于 AODV [2] 可以取得 50% 以上的性能提升。

在移动链路稳定性、无缝漫游、链路容量等方面也积累了很多的研究工作 [27] [28] [29] [30] [31]。

3.1 项目描述

{ 43 % : 本项目旨在构建一套稳定、健壮、大规模的无线 Mesh 网络基础设施，以支持大量的实时监控视频数据传输。 }

项目基于的硬件平台为 Mikrotik RB411 U 电路板，该电路板搭载 Athros AR7130 处理器，工作频率 300 MHz，内存 32 M，天线为 Broadcom，工作在 5 GHz 频段，增益 12 dBi。该硬件平台正常通信范围为 3.5 km。另外，配置 Atheros AR9220 无线模块，该模块对 EDCA 的队列机制提供硬件级支持。节点另外设计有封装外壳，防水防爆，以适应不同的工业应用场景。

节点运行的操作系统为 OpenWRT 系统。 { 46 % : 基于 Linux 2.4.30 内核，专业设计用于嵌入式无线网络设备。 } 开发系统为 Ubuntu 12.04，编译工具基于 OpenWRT 提供的相应硬件平台的编译环境配置，引入 BATMAN-adv 路由协议的包。所有使用的软件系统均来自开源项目。

实验章节中使用公开的视频序列[32]和普通的 D-link 摄像头获取视频图像。 3.2 BATMAN-adv 路由协议

为了深入研究项目的 Mesh 网络特点，并进行深度开发，有必要首先理解 Mesh 网络核心协议 BATMAN-adv 的技术原理。BATMAN-adv 协议支持 Mesh 网络的自组网和数据路由两大核心功能。

3.2.1 数据包类型

BATMAN-adv 协议定义了六种不同的数据包类型，用于网络的路由构建、数据包传输以及网络拓扑可视化。网络中的节点称为 Originator。这里首先对每一种数据包类型做描述。

Originator Messages 通常简称为 OGMs，为 BATMAN-adv 协议的核心数据包。 { 42 % : 它用于发现网络中的节点，每个节点通过周期性的广播 OGMs 申明自己的存在，以加入网络。 } 3.1 节点广播 OGMs，邻居节点接收到后更新自身的路由表并重新广播出去，这样两个相互无法直接沟通的节点就能够知道彼此的存在，从而构建有效的多跳通信路由。

{ 41 % : 图 3.1 OGMs 传播示意图，A 点广播 OGM，接受节点更新路由表并转发该 OGM }

OGMs 的存在主要服务于两个目的： { 52 % : 1) 申明节点的存在，以及到达申明节点的可能的下一跳节点。 } { 41 % : 2) 测量链路的质量，通过多跳重广播积累路径的整体质量。 } 后续会详细描述。 OGM 包中包含如下信息：

Originator地址-用于鉴别生成该OGM的原始节点。

序列号-用于链路质量度量和重复包检测。

传输质量 (TQ) -描述到达 Originator 的整条路径的链路质量。 前驱节点地址-用于检测并丢弃以广播出去的 OGM。

TTL-用于限制OGM的最大传输跳数。

网关标识-用于标识接入外网的节点。

因特网控制报文 (Internet Control Message Packet) 简称 ICMP，用以支持一部分由 IP 版本的 ICMP 提供的特征。因为 BATMAN-adv 运行在 mac 层，因此网络中的节点不能够通过 IP 地址到达，因此协议提供了 mac 地址到 ip 地址的映射机制，并基于此设计了 mac 地址版本的 ping 工具。

单播包 (Unicast Packet) 单播数据包封装来自上层的单播数据。除了上层数据本身，单播包还在包头中添加目的地址和 TTL 字段。

分拆单播包 (Fragmented Unicast Packet) 碎片化单播数据包。BATMAN-adv 封装来自上层的数据，可能超出 mac 层 MTU 长度的限制，这就导致数据包需要被碎片化并在终点重新聚合。碎片化单播数据包除了分拆的数据外，还需要在包头添加序号用以聚合，并添加 originator 的标识和碎片末尾标识。

{ 42 % : 广播数据包 (Broadcast packet) 广播包用于向全网所有节点广播数据。 } { 46 % : 除了数据部分，包头还包含序列号，源节点地址和 TTL。 }

{ 41 % : 可视化数据包 (Visualization packets) 可视化数据包用于支持动态的图形化网络拓扑。 } 用户需要先设定其中一个节点为服务器，服务器节点会向网络中其他节点发送可视化数据包，搜集其他节点的信息，从而构建出整体网络的拓扑图。

3.2.2 节点发现

如前所述，BATMAN-adv 协议中所有节点会周期性的向其他节点广播 OGMs，当一个节点接收到 OGM 后，会做如下处理：

检查该 OGM 包中的 originator 是否为自身，如果是，则发送者为直接邻居，路由表做相应更新。

{ 40 % : 检查该 OGM 包的前一跳发送节点是否为自身，如果是，表示该 OGM 已经处理过，直接丢弃。 }

{ 50 % : 检查 originator 是否已经在路由表中存在，如果否，创建该路由表项。 }

更新到达 originator 的路由表项。

更新 TQ 和 TTL 值，重广播该 OGM。

在此基础上，还会对路由循环和重复 OGM 做检查。 3.2.3 链路质量估计

TQ 值是用以估算链路质量的核心度量值，在 OGM 的传输过程中，TQ 值逐跳计算并累计。TQ 值实际上描述了数据包在该链路上按预期到达的概率。该值存储为一个 8 位的值，大小位 0-255。

图 3.2 计算 TQ

接收质量定义为 RQ。 { 42 % : 如图 3.2，节点 B 发送 OGM 并被节点 A 接收到，A 点根据来自 B 的 OGM 序列号计算从 B 到达 A 的单向链路 RQ。 } 该过程通过一个大小为 N 的滑动窗口(默认大小 128)实现，在该滑动窗口中将纪录从最后接收到的 OGM 的序列号向前推 N 个对应的 OGM。 { 41 % : RQ 值即为滑动窗口内序列号对应的 OGM 中实际接收到的比例。 }

回程质量定义为 EQ。 { 40 % : 节点 A 发出的 OGM 到达 B 后，B 会重广播该 OGM，A 在此收到该 OGM 即可计算链路的回程质量。 } 计算类同 RQ，同样通过一个纪录序列号的滑动窗口实现。

传输质量定义位 TQ。 { 49 % : 传输质量指代数据包从节点 A 发出，被节点 B 正确接收的概率。 } { 43 % : 因为 EQ 定义为数据包双向回程正确接收的概率，RQ 定义为链路反向 }

第3章 项目设计的正确接收概率，由此即可计算 A 点到 B 点的链路传输质量 TQ:

{ 46 % : 网络中的节点从 OGM 中可以获得前一跳节点到达目的节点的 TQ 值。 } { 41 % : 如图3.3所示，A 点发出 OGM，该 OGM 传播途径的每一个节点根据 OGM 中纪录的全局 TQ 值， } 结合自身邻居表存储的本地 TQ 值，更新自身到达 A 点的 TQ 值，并将 OGM 中的 TQ 值同步更新，之后重广播该 OGM。

图 3.3 TQ 传播

如果节点 A 发出的 OGM 通过两条不同的路径到达某一节点，接收节点分别计算两条路径的全局 TQ 值，并选择链路质量更好的那一条作为最优路由路径。没经过一次节点重广播，OGM 中的全局 TQ 值会加上一个惩罚值，这一实现使协议在选择时更佳倾向于选择跳数更少的路由，全局来讲，这有利于优化信道资源的竞争。

3.2.5 路由选择

{ 41 % : 不同于很多其他协议计算整个网络的拓扑，存储到达目的节点的整条路由，BATMAN-adv 仅存储到达目的节点的最优下一跳节点。 } { 56 % : 下一跳节点收到该数据包后，会继续寻找最优下一跳发送。 }

图 3.4 给出了一个路由决策的场景示例。节点 F 发送一个数据包到节点 A。节点 F 所知道的信息仅为到达节点 A 有两条有效路径，两条路径分别对应两个不同的 TQ 值。此时，如果途经 D 的路径 TQ 值更高，则 F 选择将数据包交给 D。D 点经过同样的决策过程，将数据包交给 C，最终由 C 完成最后一跳传输。

3.3 全局 QoS 保障技术

本项目的主要目标是在视频传输 Mesh 网络中提供 QoS 保障，保证在不同场景下，用户能够接收到流畅的视频流数据。在第二章相关工作中已经介绍了目前该领域的相关工作的研究现状，这些工作或多或少存在大规模应用

的弊端。因此本文基于全面详尽的实验，提出了一套优化整体视频传输 Mesh 网络 QoS 的技术方案。整个方案分为如下三项核心技术：

3.3.1 子网信道隔离

无线网络中核心的技术挑战在于无线信道的开放性，这就意味着当多个设备需要发送数据时，同一时刻只能有一个设备有效传输。虽然现在有诸如 OFDM[参考]、MIMO[参考]等技术来应对这一问题，但目前市面上的硬件产品并不都能够很好的支持这些技术，因此不具有普适性。子网划分即将整个大的 Mesh 网络划分为多个子网，相邻子网间采用相互正交的信道，从而降低节点间相互信道竞争造成的干扰。整体上提升网络的吞吐量。

3.3.2 跨层视频帧权重差分

该方法引入 802.11 协议簇的 QoS 保障技术，该技术在系列标准的 802.11e[参考]子标准中规定。其核心在于将应用层业务数据映射到不同的 mac 层优先队列，不同的优先队列有不同的信道接入优先级，从而保障重要的对时延敏感的数据尽快获取信道传输。

3.3.3 路径质量敏感的动态切换阈值

在移动场景下，无线 Mesh 网络的组网协议会面临更加严峻的挑战，尤其是高速移动的场景下，batman-adv 协议的路由切换机制存在缺陷， { 41 % : 后面的章节将通过深入细致的实验验证，并提出有效的解决方案。 }

上述协议可以在大多数场景中提供稳定可靠的路由功能，但是针对不同的需求仍然存在很多优化的空间。为了深入的探索协议的运行机制、不同参数的影响，需要深入协议源码进行实验探究。另外，本项目旨在搭建实际运行的系统，因此所有实验均在实际的设备上运行测量。为此，开发了手动设定固定路由工具、外接显示模块装置等功能模块。

静态路由设定模块提供给 Mesh 网络管理员手动设定固定路由的工具，包括命令行即时交互接口。这个工具可以很大程度上提升实验和网络设定的自由度。在实验中，Mesh 网络可能根据协议形成固定的拓扑结构，一旦产生外界的数据压力，拓扑结构就会变化，我们很难进行一些压力测试，比如切换条件等一些临界状态的测试。另外在实际部署中，手动设定部分稳定链路的路由可以保障整体网络拓扑的稳定，避免因为一些链路的抖动，造成整个网络拓扑变化频繁，影响性能。

外接显示模块提供 rssi 轮询显示和带宽测试两项功能。正常模式下，轮询的显示邻居节点的 rssi 值，当用户需要测量到达子网簇首的带宽时进行模式切换即可实时测量。

{ 50 % : 3.5该模块通过硬件和软件两部分配合实现。 } 硬件部分采用单片机加数码管，并集成 USB 接口，通过 USB 接口直接连接到 Mesh 主板。软件部分在 Mesh 主板运行一个后台 shell 进程检测 USB 口输入信号。正常模式下，后台进程周期性的通过 iwlist 命令扫描周围邻居节点的 rssi 值。当用户需要测试部署位置到达子网簇首的有效带宽时，通过按测试按键，硬件模块会通过 USB 接口通知后台进程进行带宽测试，然后返回测量数值显示在数码管上。

网络可视化与诊断系统当网络规模达到一定规模，就很难通过终端监控网络拓扑、网络性能和网络中节点健康状况。为了更好的提供对于大规模网络的监控和维护等功能，我们开发了基于 web 端的可视化及诊断系统。我们在一台服务器上搭建 apache 服务端，提供 web 服务，同时该服务器连接 Mesh 网络中的某一个开通了可视化管理功

能的节点，开通了可视化管理功能的节点使用可视化数据包获取网络拓扑信息，作为可视化服务器的角色存在与 Mesh 网络中。客户端通过 apache 提供的端口登录 web 服务，web 页面实时显示网络的拓扑形态，同时显示每条链路的 TQ 值指代其链路质量。同时 web 页面封装了一个简单的 shell 控制台，可以直接和节点交互。

本章详细介绍项目系统的实现，集中于介绍项目的三个创新点：子网信道隔离的部署方案、跨层视频帧权重差分技术、路径质量敏感的动态切换阈值算法。其中，子网信道隔离从整体网络规划上知道大型 Mesh 网络的架构；跨层优先级队列映射在将 802.11 系列标准的 QoS 技术引入，针对视频传输提供了强大的 QoS 保障；移动场景下的 QoS 保障专注于优化 Mesh 节点移动带来的路径切换时延，并有助于优化固定 Mesh 网络中的路由震荡现象。

本章工作基于细致全面的实验和对 BATMAN-adv 代码的在实际系统的调试和理解，带来的网络性能上的提升巨大。在系统实践中也积累了大量的无线网络规划经验和 linux 内核模块开发调试的技术，为我们的多个对外项目的实施提供了指导。

4.1 子网信道隔离

{ 44 % : 出于成本考虑，项目所选择的硬件平台仅支持单一无线接口，这就意味着每一个节点设备同一时间只能工作在一个无线信道。} 而无线信道的开放性所带来的信道竞争排斥多路并发传输。{ 42 % : 不同的无线终端在进行数据传输之前需要先竞争使用信道。} 这个阶段称为信道接入。

在 802.11 定义了基本的信道接入方式：{ 47 % : 分布式协调功能，该方式本质上就是一种 CSMA/CA 多路访问载波侦听冲突避免机制。} { 44 % : CSMA 机制在工业界广泛使用，比如以太网中使用的 CSMA/CD，其中 CD 表示冲突检测。} { 42 % : 在无线网络中，因为隐终端的存在，CD 的方式不可靠，因此多使用 CA 的方式。} CSMA/CA 工作流程如下：当一个终端需要占用信道发送数据时，如果信道繁忙(其他终端正在使用信道)，{ 43 % : 那么就会推迟一段时间进行下一次尝试，如果信道检测为空闲就可以立即发送数据。}

4.1.1 相邻链路干扰 因为信道竞争和隐终端的影响，如果所有节点选择同一信道则会造成严重的

{ 62 % : 信道干扰，导致网络总体吞吐量的急剧下降。}

首先是相邻链路的信道竞争。图 4.1 给出了实验图示，节点 1 和节点 4 之间通信的同时节点 2 和节点 3 通信，产生信道竞争，导致每条链路的实际有效带宽下降。实验结果如下表 4.1 所示。

可以明显看出当两条相邻链路同时发送数据时，各自的吞吐量降为原先的一半左右，当相邻的并发链路数量过多时即可能导致每条链路的实际吞吐不足以支持视频流传输。

4.1.2 隐终端

隐终端指当网络中存在多个终端时，某终端只能在信道竞争时获知相邻终端的存在，而无法感知其他更远处终端的存在，处于远处的未被感知的终端即称为隐终端。当某终端侦听信道判断当前信道空闲时，就会发送数据，但可能与此同时远处的隐终端也认为信道空闲并发送数据，假设此时两份数据的接收者处于两者的物理位置的中间，则两份数据同时到达将造成混乱，无法分辨，从而无法应答。{ 42 % : 导致两边的终端不得不反复的重传，甚至发生数据包丢失。} 以下实验就是探究隐终端在 Mesh 网络中的影响。

首先进行单跳实验，网络拓扑如图 4.2。 { 41 % : 客户端连接 1 节点，服务端连接 5 节点，图中共计 5 个节点组成一个稳定的无线 Mesh 网络。 }

{ 46 % : 在客户端和服务端之间，通过 Mesh 网络进行 100 次测试。 } { 40 % : 每次测试运行 20 秒，客户端上的发送进程分别以 1 至 100 间隔 1Mbps 的发送带宽向服务端发送 UDP 数据包。 } 每次测量结束，服务端上的服务进程会统计客户端此次通信数据的有效带宽、时延抖动等数据并告知客户端进程。所有测量结果在客户端汇总整理。

然后进行隐终端存在的实验，网络拓扑如图 4.3。客户端机器通过交换机与 Mesh 网络中的 1 节点和 5 节点连接，服务端机器与 Mesh 网络中的 3 节点相连， { 43 % : 在该网络中，1 节点和 5 节点互相不在覆盖范围内，不知道对方的存在。 } 在客户端和服务端之间，客户端虚拟两个进程，利用 Iperf 工具并控制 1 节点和 5 节点并发向 3 节点发送数据，调节发射功率从 500 次依次增大到 1700，每次通过 Mesh 网络进行 100 次测试。 { 40 % : 每次测试进行 20 秒，客户端上的两个进程分别以 1 到 100 间隔 1Mbps 的发送带宽向服务端发送 UDP 数据包。 } 每次测量结束，服务端的进程统计客户端进程此次通信数据的有效带宽等数据并告知客户端进程。所有测量结果在客户端汇总整理。以上实验过程在 1 节点和 5 节点 RTS/CTS 打开与关闭的情况下分别进行依次。

单跳带宽测试实验结果如图 4.4(a) 所示，可以看到在无干扰情况下，硬件平台支持的最大带宽接近 100Mbps。

相较之下，隐终端存在的带宽测试实验结果如图 4.4(b) 所示，该图示中同时呈现了 RTS/CTS 打开和关闭情况下实际有效带宽。可以明显看出 RTS/CTS 打开情况，实际有效带宽提升在 10Mbps 至 20Mbps。 { 47 % : 但考虑到 RTS/CTS 打开，因为需要发送 RTS/CTS 控制帧，所以会造成一定程度上的额外资源开销。 }

进一步引入发送功率作为因变量。之所以考虑发送功率，是因为发送功率和干扰范围密切相关，当发送功率升高时，其干扰范围就会扩大，从而导致网络整体吞吐量下降。反之，发送功率过小，则会导致网络形成更多的多跳路由，而每多一跳同样会带来额外的干扰。依次我们对发送功率同样进行了细致的实验辅助建模，探索其和有效带宽的关系。

实验结果如图 4.5(a) 所示，可以发现当发送带宽较小，即使较小的发送功率也可以满足要求。但当发送带宽超过 50Mbps 后，发送功率就会成为限制有效带宽的一个重要因素，随着发送功率进一步增大，有效带宽逐渐达到饱和。

综合发送带宽和发送功率绘制图 4.5(b)。可以看到 RTS/CTS 打开的情况下，有效带宽呈现出更好的状态。因此在实际项目系统的测试和部署中，如不特殊说明，将选择打开 RTS/CTS 功能。

4.1.3 子网划分与信道分配

由上述实验可以看出相邻链路干扰和隐终端的存在会给 Mesh 网络的实际有效带宽带来很大的干扰。因此在进行总体网络规划的时候，选择进行子网划分，子网内部工作在同一信道，子网之间采用正交信道，避免相互干扰。

5GHz 在高频段有 10 个可用信道(根据所在国家和地区有差异)。设置 20Mbps 的信道宽度，则可以划分为 5 个互相正交的独立信道。根据四色定理，平面空间中的区块可以用四种颜色着色，相邻区块不会出现重色。 { 62 % : 据此，我们将整个网络划分为多个相邻的子网。 } 子网内部独立自组织形成 Mesh 网络，每个子网选择一个

节点作为簇首节点，簇首节点作为该子网联通外部的网络的网关。 { 40 % : 上层簇首节点因为数量可控，我们通过桥接形式接入远距离定向无线传输设备。 } 这样最终的网 络架构将是一种两层结构。 该结构既保留了 Mesh 网络的独特优势，又进一步通过 分层分信道划分子网，到达隔离干扰域，优化整体网络吞吐量的目的。

划分子网的另一个好处是，在我们的系统目标场景中，油田是一个重要的应用场景，而在野外的油田分布十分分散， 一套 Mesh网络形成大范围覆盖效果并不理想，并且会需要大量的中继节点，造成严重的资源浪费。 为此我们可以在每一个 油田或相邻的几个油田部署独立的一套 Mesh 子网，完成对该油田区域的覆盖，可以 提供充足的带宽用于监控、生产数据的传输。 最终通过簇首桥接远距离定向无 线设备完成数据汇总至油田的控制中心。 如 [图] 所示。

基于以上实验论证，我们搭建了较大型的室内实验平台，使用 3.1 节中介绍的 硬件平台配置了 101 个 Mesh 节点。 { 41 % : 将这 101 个节点划分为 10 个 Mesh 子网，每个 Mesh 子网的簇首桥接一个远距离无线设备，无线设备与 sink 直接通信。 } 实验平台 如图 4.6所示。 因为原硬件平台天线信号覆盖范围最大可达3~5 km，显然会造成整个实验床的全覆盖， 形成严重的节点之间的干扰，实验验证在这种情况下视频完全无法传输。 于是我们给天线加装衰减器，通过调节衰减器的衰减性能将天线 信号覆盖范围控制在三米左右。 这样可以基本保证划分的子网做到一定的区分。

实验床的节点物理位置分布成如图 4.7所示，图中每种颜色代表一个独立的信 道，相互正交，相邻信道间隔知道为两个独立信道，因此不存在干扰。 { 41 % : 图中蓝、黄、红、紫分别代表5 GHz频段的142、149、153、157号信道， } 这四个信道用于分配给子网，根据四色定理，存在这样的方案使得任意相邻信道之间不存在干扰。 因为不同子网间节点在不信道，互相之间不可见，因此组网协议在不同子网间也是 不可组网的，这样就实现了子网的隔离，而子网内部可以自由组网。

这样的子网是相互独立的，而我们最终需要的是一个整体的网路，因此我们使用两层架构， 在子网之上，每个子网的簇首节点桥接一个上层节点，该节点采用定向发送方式，因此虽然需要更高的传输功率但干扰范围可控，且独立占据一个与所有子网都不相互干扰的信道。

完成配置部署后我们进行了实验验证，该方案带来的性能提升十分巨大，如 表 4.2所示。 实验比较了不同的信道划分方案，主要分两类，一种是所有节点均采

用同一信道，全干扰状态下进行视频传输实验，另一种是上述子网划分方案。 在 第一种单一信道实验中，用三种不同且相互正交的信道分别进行，以排除环境种 可能的信号干扰。 在每个子网中接入 3 路视频，一共 30 路视频，每一路视频占用 传输带宽设置为 2Mbps。 实验结果以实际视频的传输质量为参照。

从表 4.2中显而易见，子网划分方案带来的性能提升是显著的，单一信道在这 样的全干扰状态下几乎无法正常工作。

4.2 跨层视频帧权重差分

在过去的二十多年中，无线局域网中的视频流传输研究吸引了大量的研究人 员的努力，其中很多的工作都聚焦在 QoS 管理方面。 比如，802.11e 定义的 EDCA，在 mac 层实现了不同的权重队列，每个队列对应不同的发送优先级，上层过来的业 务数据根据类型被映射到不同的权重队列中。 { 49 % : 视频和音频数据通常具有更高的机会进入高优先级的发送队列。 } 依次，相对于传统的 802.11 的信道竞争机制，EDCA 的支持下，视频音频数据将获得刚好的传输带宽，从而带来更好的用户体验。

另一方面,视频帧编码技术将视频编码为不同权重的帧,权重更高的帧对于视频的解析更重要,甚至仅有高权重帧的情况下,也可以解码出流畅但不甚清晰的视频。由此,将视频帧解码为不同权重的帧映射到不同的EDCA权重队列中,就可以实现在带宽资源过度匮乏的情况下,优先传输高权重视频帧的目的。

目前,一些研究工作已经将这一想法引入了无线 Mesh 网络,但是并没有取得显著的效果。特别是在网络规模扩大或者视频数据量增大的情况下,带宽很容易耗尽[33]。其他研究工作提出网络中的在线视频数据削减[34]。但是,计算机视觉技术会需要很大的资源开销,这对计算资源受限的 mesh 节点是一个极大的挑战。另一方面,相关工作大多数都是通过仿真软件来完成,缺少实际系统的验证。

通过细致的实验分析,我们发现现阶段很多工作提出的映射机制和入队算法越来越复杂化,能够优化的空间越发有限。一个可能的性能突破的方向是对数据进行更加细粒度的分析。例如对于视频数据帧,不仅仅停留在编码帧的层面,而是

深入观察相同的编码帧之间的差异,达到更好的映射效果。该想法基于我们实验中的三点发现: { 48% : 属于同一类型的编码帧在解码过程中的权重不同。 } { 42% : 典型的视频帧编码标准比如 H.264 和 MPEG-4 将视频帧编码为三种不同类型的帧,分别是 I 帧、P 帧和 B 帧。 } { 40% : 实验发现,属于同一个组的 P 帧权重依次递减,B 帧也呈现同样的规律。 }

因为同一个编码帧可能在网络层或者更低层被分割为不同的数据包,这些数据包同样会呈现不同的重要性。因此,在设计映射机制的时候,应该深入不同该层次的数据包,而不是停留在编码帧的层面。

映射机制应该具有足够的灵活性,使不同类型的编码帧都有机会使用高权重的发送队列,最大限度的利用有限的带宽。

基于以上,本项目提出一种无线 Mesh 网络视频帧的 mac 层权重队列映射机

制,全面的考虑了编码帧类型、帧出现的位置、更细粒度的数据包形成最终的映射方案。该方案在高效的网络传输和有限的节点计算资源之间做到很好的平衡。

4.2.1 视频帧分层编码

视频数据一直是数据量最大的业务数据类型,再如今大数据火热的阶段,尤其被关注,在无线网络中视频数据同样也给网络传输带来了极大的挑战。 { 48% : 不同的视频编码技术应运而生,在这之中,H.264 和 MPEG-4 被业界广泛接收和使用。 } { 40% : 这种类型的编码方式在性能和编码复杂度上都能够很好的满足大多数的应用场景。 }

{ 48% : 分成编码技术将视频编码为三种不同类型的帧: } I 帧、P 帧、B 帧。每一种帧在视频解码时都扮演不同的角色。

I 帧(帧间编码): { 43% : 该帧数量最少,包含全部用来解码其自身的信息,这就以为这 I 帧的解码不需要依赖其他 I 帧或者 P 帧、B 帧。 } { 51% : I 帧通常压缩为一个静态图像,体态也较 P 帧、B 帧更大。 }

P 帧(前向预测帧): { 52% : 该帧包含它之前最近的 I 帧和 P 帧之间的差异,因此需要依赖之前的 I 帧和 P 帧来解

码。}

B帧(双向预测帧): { 47 % : 该帧压缩尺度最大, 需要依赖其之前和之后的I帧和P帧 来解码。 }

根据 H.264 和 MPEG-4 编码, 一小段视频会被解析编码为一组重新组织过的 帧, 这一组帧称为 GOP。 { 50 % : 每个 GOP 总是由一个 I 帧开始, 之后跟随一段连续的 P 帧和 B 帧。 } GOP 的结构可以表示为两个参数 $G(N, M)$, 其中 N 表示 GOP 中包含的所有帧的数量(两个连续 I 帧之间的距离), { 46 % : M 表示 I 帧和其后第一个 P 帧之间(或者两个连续 P 帧之间)的距离。 } 图 4.8 为 $G(12, 3)$ 的结构示例, 分解后即为:

' IBBPBBPBBPBB '。

如图 4.8 所示, GOP 内部的不同帧之间紧密相连, 不同的 GOP 之间相互独立。

{ 57 % : 实质上, I 帧不依赖于任何其他帧, 可以单独解码, P 帧依赖于其之前的 I 帧或者 P 帧, B 帧依赖于其前后的 I 帧或者 P 帧。 } { 44 % : 也就是说, 如果在传输过程中, 丢失了 I 帧则整个 GOP 都无法解析。 } { 51 % : 类似的如果一个 P 帧丢失, 则其后的 P 帧和 B 帧均 无法解码。 } 而 B 帧丢失仅影响其自身, 其他数据帧不会收到任何影响。 { 51 % : 由此我们 可以总结三种类型帧的重要性, 显然 I 帧 > P 帧 > B 帧。 }

综上, 处于 QoS 保障的考虑, 当带宽紧缺时, 需要保证重要性更高的帧优先 传输。

4.2.2 802.11e 增强分布式协调访问 (EDCA)

{ 41 % : 传统的 IEEE802.11 标准分布式信道接入协调功能 (DCA) 做为基础的信道接入 方式, 该方式基于 CSMA/CA, 不提供 QoS 保障。 } 为了支持区分的服务, 802.11e 引入了一种混合协调功能 (HCF), 其中包含两种并行机制: { 61 % : 混合控制信道访问 (HCCA) 和增强分布式协调访问 (EDCA)。 } 本项目中基于 EDCA 功能展开。

EDCA 通过引入四个不同的接入类型 (AC) 来提供 QoS 保障。 传统的 802.11 信道接入方式维护一个发送队列, 所有业务数据平等的进行信道竞争, 不同于此, 每个 AC 的数据维护各自的发送队列, 通过竞争窗口控制每个队列竞争获得信道的概率不同。 { 46 % : 如图 4.9 所示, 四个队列按权重从低到高依次是: } AC_BK(背景数据), AC_BE(尽 最大努力传送), AC_VI(视频), AC_VO(声音), 依次编号为 AC(0), AC(1), AC(2), AC(3)。 不同的权重通过设置不同的信道竞争参数实现, 包括竞争窗口界限、仲裁帧间隔、发送机会限制。

默认情况下, QoS 支持在 EDCA 中的实现是通过将实时数据包括音频视频映

射到 AC(2) 和 AC(3) 中, 其他的数据则映射到 AC(0) 和 AC(1)。 { 42 % : EDCA 机制不考虑 视频帧的不同类型之间重要性的区别, 将所有的编码都映射到 AC(2) 中。 } 然而, 在视频监控的场景中, 除了少量的控制数据, 几乎所有的数据都是视频数据, 这些数据都被映射到同一队列, 很容易造成该队列的饱和, 进一步造成丢包, 影响传输质量。 因此, 在无线 Mesh 网络中, 尤其在视频流作为主要传输数据的场景中, 应该充分挖掘视频帧编码之间的差异, 利用 EDCA 四个队列的功能来实现更好的用户 观看体验。 之前已经有一些工作做过这方面的尝试, 图 4.10 给出了静态映射和动态映射的对比。 静态映射, 比如[35], 仅仅将不同重要性的帧固定的放入固定的队列, 这样会造成优先队列资源的浪费。 动态映射, 比如 [36], 进一步考虑网络地动态状况和队列地占用情况, 动态地进行队列映射, 提高了网络地利用度。

4.2.3 视频帧映射机制

基于 GOP 编码结构和 EDCA 机制，结合细粒度地分析视频数据包，我们提出了一种跨层映射方案，实验显示该所提方案相对于朴素地基于 EDCA 传输方案，优化效果达 50%。

{ 59 % : 在视频监控系统中，视频数据占所传输带宽的绝大部分。 } 在衡量视频传输质量的时候，除了传统网络度量参数如丢包率等，更重要的是视频观看体验， { 71 % : 目前普遍使用的衡量指标是峰值信噪比 (PSNR) 和结构相似性指标 (SSIM)。 } { 48 % : 视频传输的流畅、实时、清晰是视频监控系统的首要目标。 }

受通用视频编码技术中的分层编码的影响，可以认为不同类型的帧对视频的传输质量贡献不同。 { 43 % : 这是由分层编码中 GOP 内部不同帧之间的依赖关系决定的。 } 如前所述，I 帧独立于其他帧，可独立解码其压缩率也最低； { 53 % : P 帧依赖于其之前的 I 帧或者 P 帧，压缩率次之； } { 58 % : B 帧压缩率最高，但需要依赖其前后的 I 帧或者 P 帧来解码。 } 不难发现，I 帧丢失则该 GOP 内全部的帧都无法解码； { 43 % : P 帧丢失则其后的所有帧无法解码，B 帧丢失则仅其自身无法解码，不影响其他帧。 }

{ 44 % : 图 4.11 和图 4.12 给出了 I 帧、P 帧、B 帧以及同一个 GOP 中不同位置的 P 帧丢失对视频质量的影响。 } 可以看出，和我们预测的相同，在相同的数量上， { 57 % : I 帧对视频质量的贡献明显大于 P 帧， } { 41 % : P 帧中出现在靠前位置的 P 帧影响力又要大于之后的 P 帧， } 而 B 帧对视频质量的影响最小。

之前的关于映射机制的工作，仅仅按照编码帧的类型进行映射，而没有考虑不同位置帧的重要性不同。我们将这一点引入映射机制，综合考虑帧的类型和位置。下面详细介绍该映射机制的细节原理和实现。其中核心分为两部分，帧权重计算和帧映射。

信道 (L1-L2)

GOP 参数

数据包权重值 受当前帧影响的帧数量 影响当前帧的帧数量 权重值计算公式参数

{ 42 % : 权重值基线 帧分解的第一个数据包的额外权重 P 帧在 GOP 中的位置索引 }

AC((i)) 的最大队列缓存 AC((i)) 当前使用长度

{ 43 % : 对于 I 帧，因为其相对于 P 帧和 B 帧数量极少，且对于解码 GOP 起到至关重要的作用，所以在权重上赋最大值 1，即为 1； } 对于 P 帧和 B 帧，它们的重要性和它们之前的需要依赖的帧以及它们后续的依赖于它们的帧有关。这里我们定义受当前帧影响的帧数量为 f_0 ，对当前帧有影响的帧数量为 f_1 。进一步的定义当前帧的权重计算公式为：

$$= g(f_0 - f_1) \quad (4-1)$$

在这里， $(0, 1)$ 是一个权重常量，用来平衡 f_0 和 f_1 ， $f_0 - 1$ (所有的 P 帧和 B 帧至少被 I 帧影响)， $f_1 - 1$ (每一个帧至少影响它本身)， $g(x)$ 是一个单调递增函数。这就意味着，越多的帧依赖于当前帧，或依赖于当前帧的帧越少，则当前帧的权重就

越高。 { 43 % : 更直观的讲, 就是在一个 GOP 中出现越早的帧相对于晚出现的帧权重更高。 } 27

第4章 项目实施 为了降低 Mesh 节点的计算开销, $g(x)$ 定义为:

$g(x) = a(\log(x) + b) + b_0$ (4-2) 带入 表达式则有 P 帧和 B 帧数据包的权重值计算公式:

$= a(\log f_0 + f_1 \log + b) + b_0$ (4-3) 上式中, b_0 是作为权重值的基线, 确保 B 帧能够有机会进入优先级较高的队

列。 a 、 b 和 b_0 用来限制 在 $[b_0, 1]$ 的范围内。 可推出 a 、 b 值如下:

$$a = 1 - b_0, b = \log(N/M) \quad (4-4)$$

$$\log(N) - \log(N/M)$$

上式中, $\log(N)$ 和 $\log(N/M)$ 分别为 $\log(f_0 - f_1)$ 的最大值和最小值。 对于一个 GOP 中的第 p 个 P 帧, $f_0 = N - 1 - p$, $f_1 = p$ 。 对于任意子第 p 个 P 帧 后的 B 帧, 有 $f_0 = 1$, $f_1 = 2$ 。

目前为止, 我们定义了普通的视频帧数据包的权重计算方法, 该方法充分利用的 GOP 结构中不同类型帧不同的重要性。 下面我么考虑利用同类型不同位置帧 重要性不同的性质, 其中关键的是一个视频帧被底层协议拆分为一个个的数据包时, 最重要的是第一个数据包, 其中包含了编码重要的信息。 我们选择添加一个额外的值 h 来提升第一个数据包的权重。 如果出现 $h \geq 1$, 就设置 $h = 0.99$ 。 之所以这里不设为 1 是考虑到防止 P 帧或者 B 帧抢占 I 帧的优先权, 在保证 I 帧绝对优先权的基础上最大限度提升 P 帧或者 B 帧较重要的数据包的传输几率。

B. 帧映射

{ 40 % : 所谓帧映射就是将属于不同类型帧的数据包映射到不同的 EDCA 的 AC 中, 区分他们在发送时的优先权。 } 根据上一节涉及的权重计算方法, 可以计算出每一个 视频帧对应的权重值, 帧映射即可以根据该权重值完成。 映射的直接原则是充分 使用高权重队列以降低传输延迟和丢包率。 当网络质量良好, 传输数据量较小时, 甚至可以将 B 帧映射到 AC(0) 中, 所有帧都可以以最低延迟, 最高的优先权完成 传输。 相反, 当网络质量较差, 网络传输数据量过大, 造撑拥堵时, 映射机制优先 将 I 帧映射到最高优先权队列, 保证 I 帧的传输质量。

{ 47 % : 当一个数据包到达时, 映射机制将顺序从最高优先权队列 AC(3) 开始依次检 }

测。 如果发现当前 AC 有足够的缓冲区余量, 且满足映射标准, 即将数据包插入该 AC 中。 判定的标准为, 给定一个数据包的权重假设为 w , 如果 $w \geq \text{threshold}(i)$, 就将数据包插入队列 AC(i); 否则, 继续检查优先权次之的队列。 其中, $\text{threshold}(i)$ 和 $\text{qlen}(i)$ 是 AC(i) 可以通过底层 API 读取的最大队列长度和当前队列占用。 如果一个数据包不能够插入 AC(3), AC(2) 和 AC(1), 那么就直接插入 AC(0) 而不需要进行进一步的检查。 映射算法尽可能设计的简单以减少 Mesh 节点的计算开销, 因为在工业应用中, Mesh 节点能源和计算资源都比较稀缺。

{ 43 % : 上面的队列映射策略给予了更高权重的数据包更高的概率插入高优先级队列, 同时阻止低权重的数据包阻塞队列。 }

C. 映射系统实现

队列映射如图 4.13 所示，跨层架构设计也在图 4.14 中展示。在结构设计上，因为 BATMAN-adv 协议主要工作在 mac 层，且 EDCA 的四个 AC 也在 mac 层实现，所以我们的权重计算和映射模块均在 mac 层实现。尽管如此，我们需要在考虑来自引用层的数据，网络层的 ToS 字段，以及物理层的队列大小及占用情况。实际系统中，每个视频帧的数据包，其权重均在产生该数据包的第一个 Mesh 节点计算。当一个 Mesh 节点接到请求需要发送视频数据时，首先解构 GOP 为 I 帧、P 帧和 B 帧，然后将各个帧在网络层封装为较小的数据包，之后就根据每个数据包所属帧的类型、位置和数据包是否为帧的头数据包等计算数据包的权重。之后数据包的权重值一直维护在数据包中，在网络传输的过程中不会改变该权重值。在最近的实现中，我们将权重值存储在网络层包头的 ToS 字段中。

数据包传输到网络中间任一节点时，该节点队列映射也按如图 4.13 所示进行，但不需要重新计算权重值，只需要从 ToS 字段中取的该权重值即可， { 45 % : 之后就根据该权重值，将数据包插入相应的 AC 中。 }

第4章 项目实现

4.2.4 实验验证 A. 实验方法

在 3.1 节已经介绍了项目所采用的软件系统和硬件平台。更细致地，OpenWRT 内置的 mac80211 模块控制数据包的收发。队列选择是数据包发送过程中的一个子环节，核心工作就是修改队列选择算法。每当接收到一个数据包，首先检查改数据包包头，如果是一个符合 H.264 编码标准的数据包就抓取其中的帧类型、GOP 信息和帧编码等。通过公式 4-3 计算出数据包的权重。之后，根据计算得到的权重

{ 43 % : 值和队列的负荷情况将数据包插入合适的队列中。 } { 41 % : 转发该数据包时，将权重值记录在 IP 数据包包头的 ToS 字段中，之后的转发节点不需要重复计算。 }

实验中使用公开的视频序列[32]中的“foreman”作为基准进行比对。将视频转换为 25 帧每秒的 GOP(12, 3) 格式。考虑 Mesh 延时，将延时超过 1 秒的视频帧丢弃。

实验中采用的测量参数为 PSNR、SSIM 和 DFR，均为视频图像实验分析常用的标准参数。其中 PSNR 描述图像像素级的差异，对于视频则秒速视频帧图像差异的平均值；SSIM 描述解构差异，更能够体现人的观看体验。DFR 描述解码成功的帧占总帧数的比例，属于帧级别的参数。

我们选取了之前的类似工作做对比。其中 SMM[37] 基于帧类型；DMM[38] 基于帧类型和流量负载；AMM[39] 考虑了帧类型、流量负载和 GOP 结构；EDCA 级 802.11 默认的映射方法，也即所有视频映射到优先级次优的队列。

B. 实验结果及分析

总体性能上，我们选取参数 $\alpha = 0.6$ ， $b_0 = 0.2$ ， $h = 0.6$ ，AC(0) 到 AC(3) 的 threshold 分别设为 [, 80, 50, 50]。图 4.16 给出了不同映射算法在视频图像上的直观体现。图 4.17 给出了 PSNR 和 SSIM 两个参数的比。

其中 MeshMapping 即本文所提出的映射算法。可以看到 MeshMapping 在 PSNR 参数上要明显优于其他算法，在 SSIM 也超过其他算法。相对于 EDCA 更是能够在 PSNR 和 SSIM 两个参数上分别优化 74% 和 54%。

进一步地，我们研究公式 4-3 中参数对性能的影响，包括 α ， b_0 ， h 。选用视频资源中的“coastguard”，“container”和“foreman”三种视频来做实验验证。

参数影响之 α 。 结果如图 4.18 所示。原则上， α 用来平衡 f_0 和 f_1 ，这两个参数分别用来表示受该帧影响的帧数量和影响该帧的帧数量。{51%：越小则 f_1 对权重影响越大，反之 f_0 影响越大。} 图 4.18 给出了当 α 从 0.1 增长到 1.0 的过程中，各参数值的变化。从 (a) 可以看出，随着 α 的增长，P 帧的权重值逐渐增长，B 帧的权重逐渐降低。另外，如果 α 过小，序列靠前的 B 帧甚至会比靠后的 P 帧权重更高。反之，如果 α 过大，所有 B 帧的权重会非常低。从 (c) 可以更清楚的看出这一点，当 α 大于 0.7 后，DFR 值会急剧下降，这样会导致 B 帧数据包丢失的概率提高。结合 (b) 可以看出，当 α 处于一个合理的范围，[0.4, 0.6] 区间内，可以获得最好的解码质量。

参数影响之 b_0 。 结果如图 4.19 所示， b_0 用来确保序列靠前的 B 帧能够有机会进入高优先级的队列。(a) 显示随着 b_0 的增长，P 帧和 B 帧的权重值都会相应的提升，意味着更多的 B 帧能够有机会进入高优先级的发送队列。{45%：在 (c) 中，随着 b_0 更多的 B 帧进入高优先级队列，导致 DFR 值增长。} 但是，从 (b) 中可以看出，这样并不意味着视频质量的增长，更多的 B 帧进入高优先级队列，就可能导致 P 帧被分配到低优先级的队列，从而导致 SSIM 值反而降低。这一现象在视频图像较大时尤为明显。综合来看，我们建议合适的 b_0 取值范围应在 [0.2, 0.4]，在这个范围内 SSIM 值和 DFR 都相对可观。

参数影响之 h 。 {41%：结果如图 4.20 所示， h 用来提升帧的第一各数据包的权重值。} 当 h 提升时，DFR 值显著提升，因为更多的头数据包被接受到。但是，尽管更多的头数据包被成功接收并解码，SSIM 值却没有相应的提升。原因在于，头数据包仅仅保证一个帧可以解码，但不保证其他帧的接收，而 SSIM，以及 PSNR 都更加依赖于整个帧的解析情况。总体来说，更多能够解码的帧带来更好的流畅体验。在实践中，我们通常将 h 的值设定在 [0.6, 0.8] 区间内，保证一个相对较好的头数据包的发送成功率。

视频内容的影响。 实验结果在视频大小，时长，内容丰富性上有较大差异。比如，“container”是一段监控视频，记录缓慢的桥梁的画面；相比之下，“coastguard”记录了一段摩托艇在海上冲浪的视频，画面变化非常快，因此，在实时性上，“coast-guard”的要求比“container”要高很多。从图 4.18，4.19 和 4.20 可见，在任何情况下，无论上述的参数如何设定，“container”都可以以很高的质量呈现。所以，本文所设计的映射算法在数据量大，延时要求高，丢包严重的场合更加能够体现其优势，也可以预期其在复杂的工业监控场景下的良好的性能。

4.3 路径质量敏感的动态切换阈值

无线 Mesh 网络目前还处在一个高速发展的阶段，不管是学术界还是工业界目前都还在尝试不同的技术。物理层上，MIMO、多无线模块、软件无线电等技术都被尝试应用到 Mesh 网络中以获得更好的网络性能。链路层上，QoS 技术、时分复用技术等也被广泛应用。网络层上更是围绕组网协议这一核心模块集中了大量的研究工作，在 2.2 节中已经列举了诸多目前正在使用的 Mesh 组网协议，它们在链路选择、组网方式上或多或少都存在差异，而这些协议目前共同面对的一项挑战就是不能很好的应对移动场景下漫游的切换时延问题。本节的工作集中在移动场景下如何保证客户端接入在不同的 Mesh 主干网络节点之间切换时做到低时延。

在 3.2 节中详细介绍了 BATMAN-adv 协议的数据包类型、节点发现过程、链路质量估算和路由选择过程。其中链路质量的估算直接影响到链路的选择，决定了能否及时的感知链路状况变化，客户端发生漫游，并且时换链路的直接决定条件。但另一方面，如果对链路状态过于敏感，则可能导致路由震荡，甚至路由失效。这些在

后面的实验验证部分都会通过实际场景论述。 因此我们需要在路由对链路变化感知敏感度和路由切换频率之间进行权衡。

如 3.2.3 节所介绍，在 BATMAN-adv 协议中，TQ 值作为链路估算的标准值，描述在链路上传输成功的概率。TQ 值记录在 OGM 包中，随着 OGM 传播至网络中其他节点，在传播过程中逐跳累乘当前链路 TQ 值，并在每多一跳时减少一个惩罚

值，该惩罚值默认为 30，作为对多跳路径的惩罚，这意味着在网络构建中偏向于选择尽量短的路径。TQ 值存储为一个 8 位的数值，大小为 0-255，数值越大表示链路的质量越好，255 即表示该路径上的数据可以保证百分之百的传输成功率。另外，TQ 值在最终被使用时还会经过一个滑动窗口做滑动平均，窗口默认大小为 5，即对最近的 5 次计算得到的 TQ 值做滑动平均，以此防止网络的抖动对路由选择造成影响，而链路的抖动在无线网络中十分常见。

因为节点无法直接计算发送的成功概率，所以在实现中通过接收质量 RQ(从

该邻居点成功接收的概率)和回程质量 EQ(成功发送至该邻居节点并成功收到应答

的概率)得到。 计算过程再 3.2.3 章节中详细介绍。 最终的单跳链路的 TQ 计算公

式为 $TQ = EQ$ 。 整条多跳路径的 TQ 值为该多跳路径上每一条单跳链路 TQ 值的 RQ

乘积。

当链路质量能够支持所有的发送数据包都能够被接收方成功接收，则 TQ 值

为最大的 255。 相反，当网络中的链路质量出现波动，比如一条链路上出现遮挡或者节点移动，则此时链路的 TQ 值下降， { 42 % : 下降后的 TQ 值随着 OGM 广播至全网其他节点，接收到的节点做相应的路由表更新。 } 而如果在 OGM 广播到相应节点更新的这段时间内，节点正在进行通信就会出现通信中断，中断时长视 OGM 的传播速度和相应节点路由表更新速度而定。 这就会引入一个时间差。 如图 4.21，一个客户端从之前接入的节点移动至另一位置，通过另一个 Mesh 节点接入 Mesh 网络，客户端仍然能够和网络中的节点正常通信，因为网络中的每一个节点都会通过接入点的 OGM 知道客户端的存在。 然而，OGM 的产生和传输需要一段时间，这段时差内，网络中的其他节点仍然将传送至客户端的数据包交给之前的接入点，而之前的接入点在这段差内也同样不知道客户端的状态，所以实际上客户端是无法和网络中其他节点交互的。 这段时差随网络规模而变，在实验网络中可长达数秒，如图 4.22 这样的多跳场景，1 号节点需要 6 个 OGM 的转发周期才能获知客户端当前的接入状态，我们称之为多跳灾难，如果在传输过程中任意一跳丢失会导致传输时间间隔成倍增长。 这对于对时延敏感的数据，尤其是 VoIP 或者视频监控数据是无法忍受的。

4.3.1 BATMAN-adv 的漫游支持

所谓漫游，指当客户端在网络中移动，从一个接入点移动到另一个接入点的过程中不会失去连通性。 { 43 % : 传统的没有漫游技术支持的网络，如上节所述，当客户端从一个接入点移动到另一个接入点， } 需要新的接入点通过 OGM 以广播的形式通知全网中的其他节点客户端的接入状态，直到所有节点都获知客户端的接入状态变化才能够确保客户端能够与网络中其他节点通信。 { 41 % : 而这个过程随网络规模的扩大而扩大，对于时延敏感的应用场景无法忍受。 }

因此，BATMAN-adv 在 2011 年引入了对漫游的支持。该机制的核心在于添加一种称为漫游数据包的包类型，当客户端移动到新的接入点，仍然能够充分利用原先已经建立的连接，即使此时原先的连接已经不是最优的方案。

当一个节点检测到漫游事件时，就会发送一个漫游数据包，以快速恢复原先的连接。那么如何检测到漫游事件呢？当节点检测到客户端接入时会首先检查当前路由表中是否有到达该客户端的路由，如果有，则说明该客户端是从之前的接入点漫游过来的。此时如果要通过 OGM 包告知网络中之前正在与该客户端通信的节点显然很困难，因为新的接入节点并不知道客户端之前的通信节点有哪些。

时也无法保证有没有新的节点在寻求与客户端通信的有效路由。此时，最有效的

{ 44 % : 方法是利用客户端之间的接入点已建立的有效连接。 } { 53 % : 于是，新的接入点通过单播包的形式通知旧的接入点， } “客户端已经通过我接入网络”，该数据包即漫游数据包。旧的接入节点就可以快速的建立到达客户端的临时路由，网络中其他节点则可以暂时通过旧接入点和临时路由完成与客户端的通信。这个阶段网络中的其他节点是不知道客户端发生了漫游的，但也不应该一直使用这条临时路由通信，因为这条路由并不是协议选择的最佳路由。所以，与此同时，新接入点的 OGM 包会广播通知全网节点客户端新的接入状态，于是在 OGM 广播全网完成后，{ 45 % : 所有节点通过信的接入点重新建立起到达客户端的有效路由。 }

的数据。

(b) 客户端 cli 从接入点 B 移动到接入点 A，这个时间段内 ext 发送给 cli 的数据

再 A 点丢失。

(c) 一旦 cli 与 B 点连接成功，B 点就会发送一个漫游数据包，以单播包的形式发送给 A 点。

(d) A 点就开始将发送给 cli 的数据转发给 B 点。

{ 42 % : (e) B 点继续广播 OGM 至网络中的其他节点，其他节点更新 cli 的新的接入状 }

态。

(f) 所有节点收到 B 点的 OGM 后就会构建新的有效路由，直接将发送给 cli 的

数据交给 B 点而非之前的 A 点。

4.3.2 TQ 值差异导致的切换延时

BATMAN-adv 引入的漫游机制很好的优化了客户端漫游的切换时延问题。{ 41 % : 但是，在实验中我们发现，客户端在漫游的过程中并不是突然从一个接入点移动到另一个接入点， } 很多情况下是在两个接入点之间逐渐过度，移动的过程中同时与 A 点和 B 点建立有效连接，在选择路由时根据两条路径的 TQ 值大小决定。在这样的场景下，如果两条路径的 TQ 值差异较大，客户端从 TQ 值较大的接入点向 TQ 值较小的接入点移动就会出现长时间的连接中断或者通信质量差。下面的实验将进一步说明这一现象。

A. 实验方法

在固定位置放置三套 Mesh 节点，位置成一条直线，相邻节点相距 10 米。通过加装衰减器，调整功率使得仅相邻节点可互相感知，1 号节点和 3 号节点不能够直接通信，每个节点的有效信号覆盖范围如图所示。此时在一辆车上放置一个节点，节点同样控制在和固定节点相同的功率。{ 40 % : 3 号节点连接一个摄像头，小车节点连接笔记本电脑实时现实摄像头图像。 } 实验过程如下：

{ 43 % : 1.车辆按图示方向从 1 号点出发到达 3 号点，再返回 1 号点。 } 在这个过程中记录 TQ 值的变化，当前路径条数以及视频质量变化情况。

2.将 TQ 值计算的滑动窗口长度从 5 缩减到 1，重复 1 的过程。

3.将包 OGM 包发包频率从默认的 1000ms 调整为 200ms，重复上述实验。

38

B. 实现过程及现象

(1) 试验车从 A 点开始行驶，此时 4 号节点和 3 号节点之间通过 3 跳路径连接，4-]1-]2-]3。整条路径的 TQ 值为 190，视频传输正常；小车前进至 B 处的过程中，3 跳路径的 TQ 值逐渐减小。而此时另一条可用的 2 跳路径，4-]2-]3 的 TQ 值逐渐增大，当两跳路径的 TQ 值增大到超过 3 跳路径一定值时，实际传输路径理解切换为 2 跳路径。

(2) 试验车继续驶离 B 点，驶向 C 点，2 跳路径的 TQ 值继续增大，当试验车驶过 2 号节点后，2 跳路径的 TQ 值开始减小，于此同时建立其另一条有效单跳路径，4-]3，且单跳路径的 TQ 值开始逐渐增大，最终实际路由路径平滑切换到单跳路径。

(3) 之后试验车开始向反方向行驶，单跳 TQ 值逐渐减小，两跳路径(4-]2-]3) TQ 值逐渐增大，{ 46 % : 当试验车靠近 2 号节点时，视频断开连接。 } 经过 30 秒左右视频连接恢复。

(4) 试验车继续向 A 点行驶，2 跳路径 TQ 值逐渐减小，3 跳路径(4-]1-]2-]3) TQ 值逐渐增大，当试验车靠近 A 点时，视频断开连接。经过 30 秒左右视频连接恢复。

(5) 改变 TQ 值计算的滑动窗口大小会对实验效果产生影响，一定程度上会加快 TQ 值的更新，但窗口过小，比如设为 1 的情况下会造成负面影响，在网络不稳定的情况下导致 TQ 值变化剧烈。

(6) 改变 OGM 发包频率效果显著，TQ 切换速度显著加快，实验中讲 OGM 发包频率从 1000 ms 改为 200 ms，相应的试验车反向行驶到 2 号节点时，视频断开的的时间间隔由 30 秒缩短为 5 秒。同样的，当试验车反向行驶，靠近 A 点时，视频断开的的时间间隔也缩短为 4 秒左右。

由以上实验我们发现当路径的切换是从长路径(跳数更多的路径)向短路径(跳数更少的路径)切换时，可以达到完全的无缝切换；但是，当切换是从短路径向长路 39

第4章 项目实施

{ 60 % : 径切换时, 通信会有一段时间的断开。} 另外, 缩小 TQ 值计算的滑动平均窗口或增加 OGM 发包的频率均可以增强链路对 TQ 值变化的敏感性, 很大程度上增加切换的速度。

C. 实验结果及分析

实验暴露的问题是, 为什么从短路径切换到长路径会出现一段时间的通信中断。这个问题可以从 BATMAN-adv 的路由选择算法得到答案。参考实验示意图 4.24, 试验车回程过程中, 初始时, 试验车与 3 号节点之间显然通过单跳路径 (4-J3) 通信; 当试验车行驶到 C 点时, 两跳路径 (4-J2-J3) 就成为一条备选路径, 此时 4 号节点有两种路径选择, 但此时会继续选择单跳路径 (因为两跳路径的 TQ 惩罚, 此时单跳路径的 TQ 值会明显大于两跳)。此时单跳路径的 TQ 值显示为 230, 然而由于 TQ 值更新的滞后性 (为了减小网络抖动带来的影响), 实际 TQ 值是小于 230 的; 而两跳路径的 TQ 值显示为 225。

当试验车继续行驶, 会出现这样一段时间: 单跳路径的 TQ 值显示为 225 左右, 然而因为 TQ 值更新滞后, 实际要远小于 225 (差值与试验车行驶速度有关), 导致单条路径无法正常通信。而两跳路径 TQ 值也为 225 (因为需要减去 30 的多跳惩罚值, 两跳路径的 TQ 值最大为 225)。于是, 即使两跳路径的通信质量明显好于单跳路径, 但由于 TQ 值无法及时真实地反应这个差异, 也就存在这样一个时间窗口无法正常通信。相反的, 如果从长路径切换到短路径, 就不存在这样一个时间窗口。因为如果短路径能够支持正常通信, 那么它的 TQ 值不会明显小于长路径, 当长路径的 TQ 值衰退到达一定程度就会触发路径切换。

至于缩短 TQ 值滑动平均窗口和增加 OGM 发包频率, 虽然增加了链路切换的敏感性, 但可能造成路径切换频繁, 在一些场景下可能链路质量波动频繁, 导致链路切换频繁, 进而导致网络无法提供正常的数据传输功能。这也会成为网络遭受 DOS 攻击的一个隐患。

4.3.3 路由切换优化

在实验阶段已经尝试了两种解决方案, 并做了分析, 一种是缩减 TQ 值计算的滑动窗口, 另一种是增大 OGM 的发包频率, 两者同样都会造成链路的抖动, 导致无法提供正常的数据传输服务。但是两种方法确实可以很好的提升路由切换的速率。于是问题转化为如何在较小的滑动窗口和较高的 OGM 发包频率下减少路由抖动的产生?

BATMAN-adv 在路由选择算法上考虑了路由抖动的问题。 它的实现思路很简 40

第4章 项目实施

单, 只有在发现一条路径的整体 QT 值明显优于当前使用的链路时才会切换到新的路径上。所谓明显即存在一个固定的切换阈值, 只有当存在一条链路的 TQ 值高于当前链路 TQ 值至少切换阈值时才会进行链路切换。固定阈值的方法在一些场景下适用, 但是也存在显著的缺陷, 如果阈值设得过低, 则不能防止路由抖动; 如果阈值设得过高, 那么当当前路径 TQ 值较低, 以致不能提供良好的传输带宽时, 不能够及时切换到 TQ 值较大的链路上。那么能否找到一个合理的阈值, 在两者之间取得平衡呢? 很遗憾, 不存在这样的万能阈值, 因为无线链路本身受环境影响非常大, 在不同的环境中, 这个阈值的取值会千差万别。那么, 如何设定一个合理的路径切换机制呢?

面对上述问题，我们引入了新的动态路由切换指标。新的指标可以直接和 TQ 值衔接，不需要改变现有 TQ 值的计算方法。{43%：目标是尽量选择质量较好的链路，同时尽量减少路由波动，维持网络拓扑的稳定。} 设新的动态路由切换指标为 h 。 h 和 BATMAN-adv 中原先的切换阈值功能类似，但会随着 TQ 值和切换频度的变化而变化。

BATMAN-adv 在进行路由决策时，如果是第一次建立此条路由，则在路由表中选择 TQ 值最大的路径；如果已经存在一条正在使用的路径，则在路由表中查看，是否存在 TQ 值显著大于当前路径 TQ 值的路径，如果存在则替换当前路径，否则仍然使用当前路径。引入 h 后，路由决策算法为：{46%：如果第一次建立某条路由时，在路由表中选择 TQ 最大的路径；} 如果已经存在一条正在使用的路径，则在路由表中查看，是否存在 TQ 值超出当前路径 TQ 值至少 h 的路径，若存在则以之替换当前路径，否则仍使用当前路径。

具体的， h 定义如下：

$$h = \max\{a \log TQ + b \log c + h_0, h_0\} \quad (4-5)$$

其中， c 为选择当前路径的频次，利用路由表项的留用字段存储，每次选择该路径时加 1。 h_0 表示路径切换的最低阈值，可以理解为 BATMAN-adv 原来的切换阈值，但取值较低，作用仅在于过滤因链路轻微的暂态波动干扰。在该式中， $\log TQ$ 的取值范围为 $[a \log TQ - 8a, 0]$ ， $b \log c$ 的取值范围为 $[0, b \log c - 5b]$ 。而实际上，同一条路径切换频次在我们的统计数据中，极少达到千级，大多数在几十次， $b \log c$ 的取值范围在 0 到 $b \log c - 5b$ 的范围内。实验中，我们取 h_0 为 5，这样得到 h 的取值范围为 $[5, 8a + 5b + 5]$ 。公式中，两个取整项是为了和 BATMAN-adv 的 TQ 值数据类型保持一致。

上述公式的实践解释如下：当当前路径 TQ 值较低时，意味着路径质量差，此

第4章 项目实施

时会更加偏向于能够尽快的切换到通信质量较好的链路，所以当 TQ 值低的时候，切换阈值 h 较低，当 TQ 增大时，切换阈值逐渐增大，显然当 TQ 值较大时，也就是说链路质量足够好，此时即使另一条链路质量有一个明显的优势，当前链路也足够承载数据量，出于维持网络拓扑稳定的目的，尽量少的切换，此时切换阈值 h 相应的较大。另一方面，从切换频次的角度考虑，如果很少切换到当前链路，也就是当前链路的 c 较低，{41%：有理由相信可能是链路质量波动导致暂时的链路质量优势，倾向于选择其他更稳定的链路，} 于是切换阈值 h 自适应的调整为较低值；如果当前链路切换频次较高，也就有理由相信当前链路相对较稳定的维持在高质量水平，倾向于不进行链路切换，于是切换阈值 h 自适应的调整为较大。

4.3.4 动态阈值实现 A. BATMAN-adv 结构

BATMAN-adv 是二层协议，作为一个 Linux 内核模块加在到系统内核。它作为网络栈中单独一层向上层的网络层提供虚拟的 Mesh 接口，并向下层提供网络层接口。如图 4.25 所示。上层网络层只需要使用 BATMAN-adv 提供的接口，Mesh 网络本身对网络层透明。同时下层使用虚拟的网络层接口，同样不需要知道 Mesh 网络本身的存在。这就使得 Mesh 网络如同一个虚拟的交换机，原来网络栈中的一切角色都只需要向其提供的接口发送数据，而不需要任何改变。

B. BATMAN-adv 功能模块及模块间交互

BATMAN-adv 模块载入系统内核并通过配置添加一个系统网络接口后，模块即完成注册，并开始接收 BATMAN-adv 类型的数据包。相应的以太网类型和虚拟接口都完成创建。BATMAN-adv 可以拆借为很多个相互协作的功能模块，在图 4.26 中描述了它们之间的相互关系和数据流向。

图 4.26 BATMAN-adv 功能模块

soft-interface.c 模块中的函数使用虚拟网络接口。发送数据包时，数据包由 soft-interface.c 模块进入 unicast.c 模块，unicast.c 模块将数据包进行 batman 包头封装，然后转交给 send.c 模块发送。

{ 57 % : 在 hard-interface.c 模块中设置网络设备的回调函数。 } 当网络设备的接收缓存中有数据时就会触发 hard-interface.c 中的函数进行处理。 { 43 % : 模块接收到数据包后，转交给 routing.c 模块做进一步处理。 } { 49 % : 如果这个数据包是需要转发的，就会进一步转交给 send.c 模块。 } 如果是目的地址是自己，就会转交给 soft-interface.c 模块。

C. BATMAN-adv 数据发送与转发

每个 BATMAN-adv 节点需要维护 Originators 表用以记录网络中所有 BATMAN-adv 节点。Originators 表存储节点的信息，并在转发和发送本地数据包时用来做辅助决策。 { 44 % : 同时每个节点还维护邻居表用以记录所有的邻居信息，并在进行路由选择时选择最佳的邻居节点作为下一跳。 } 之后，还需要维护一个转换表，用来记录非 BATMAN-adv 网络节点的设备的接入情况。

当节点需要发送数据时，首先查看目的地址是 BATMAN-adv 设备，如果不是就查找转换表，找到该目的设备连接在哪个 Mesh 节点下。然后以该 Mesh 节点作为目的节点进行路由。路由过程如下： { 41 % : 根据发送的目的节点地址搜寻 Originator 表，找到之后在确定到达目的节点的最佳下一跳节点，然后在 Neighbors 表中找到该邻居节点，并将数据包发送给该邻居节点。 } 图 4.27 给出了发送过程细致的函数处理过程。

协议栈中上层通过注册的回调函数 interface_tx() 将数据交给 BATMAN-adv。该函数会检查源地址，如果该数据包来自一个新连接到节点的设备就需要在转换表中注册该设备的地址。目的地址需要检查是否为单播包或者广播包，如果是单播包就交给 unicast_send_skb()，在该函数中搜索转换表查找目的节点。 { 43 % : 然后，在 routing.c 中查找到达目的节点的最佳下一跳邻居，并完成 batman 包头的封装。 } 之后交由 send_skb_packet() 进行发送。

转发和发送本地数据包，其流程十分相似，唯一的区别在于来源不同，导致在预处理上有些许差异：转发的数据包需要先判断数据包的类型，以及做一些路由环路侦测等处理；而本地数据包则直接从上层虚拟 Mesh 接口获得，完成 BATMAN-adv 的封装。之后两种来源的数据包的路由和发送机制完全相同。

转发数据包从 hard-interface.c 模块进入，首先由 recv_unicast_packet() 函数处理， { 41 % : 检查源地址，如果本节点不是目的节点就将数据包交给 routing.c 模块由 route_unicast_packet() 函数完成路由和包头数据的更替， } 之后交给 send_skb_packet() 完成发送。如果本节点是目的节点，数据包就直接交给 soft-interface.c 模块的 interface_rx() 函数，进行 batman 包头剥离，并传递给上层。

D. BATMAN-adv 中实现动态阈值切换

考虑如何在 BATMAN-adv 中实现动态阈值切换，我们发现动态阈值切换主要的作用范围在于单播包做路由决

策时。 { 42 % : 当节点发送本地数据包或者转发数据包时 , 都会在发送之前选择最佳的下一跳节点 , 也就是进行路由决策。 } 如果之前没有在干路径上传输过数据 , 则需要从路由表中选择最佳下一条 , 即 TQ 值最大的路径中的下一跳节点; 如果之前传输过 , 此时针对该路径已经存储了一个默认路由节点 , 这时进行路由决策就需要查看路由表项 , 是否存在某条路径的 TQ 值显著优于默认路径 , 所谓显著就是说高出当前路径 TQ 值一定的阈值。

44

第4章 项目实施

基于前面对 BATMAN-adv 功能模块的介绍 , 显然进行路由决策的模块为 routing.c。 于是我们深入 routing.c 找到 bat_iv_find_router() 函数 , 该函数中进行阈值 比对。 此处 , 我们将原先的静态阈值替换为动态阈值 , 动态阈值通过独立模块计算 返回。

综上 , 动态阈值切换的实现基于原 BATMAN-adv 的功能模块交互方式及数据流 , 增加了动态阈值计算模块 dynamic_threshold.c , 该模块和 routing.c 模块交互 , 当 routing.c 进行路由决策时 , 就会向动态阈值计算时就会调用 dynamic_threshold.c 相关函数进行计算。 另外 , 需要在路由表项的数据结构中添加路径切换计数字段 c , 用以记录当前路径切换次数。 添加动态阈值计算模块后的功能模块结构如下 图 4.28

图 4.28 动态阈值计算模块与 BATMAN-adv 交互图示 至此就完成了动态阈值标量 h 的设计与实现 , 那么它的实际效果如何我们通

过如下实验加以验证。

4.3.5 实验验证 A. 整体性能

实验的目的是验证该算法在路径 TQ 值相近且频繁波动的情况下能否维持相 对稳定有效的路由 , 以及公式 4-5 中个参数的影响。 该实验同样基于图 4.15 示实际 监控系统进行。

实验中将两个重要的影响参数进行设定 , 将 TQ 值的计算窗口长度从默认的 5 缩减到 3 , 将 OGM 的发包间隔从默认的 1000ms 缩减到 200ms。 首先设定实验网 络拓扑 , 如图 4.29 所示 , 1 号点有两条有效路由可以和 4 号点通信 , 通过 2 号点 中继 , 或者通过 3 号点中继。 在 1 号点上连接摄像头 , 那么 4 号点会通过 1-|2-|4 或者 1-|3-|4 两跳路径收到摄像头的视频图像。 此时如果在两条路径的第一跳上

存在一个干扰物来回移动 , 因为 5 GHz 频段信号绕障碍的能力较差 , 当有障碍物移动到 1 号点和 2 号点之间时会影响 1-|2-|4 链路的通信质量 , 降低 1-|2-|4 路径的 TQ 值 , 使得通过该条路径传输的视频图像卡顿; 同样地 , 当障碍物移动到 1 号点 和 3 号点之间时会直接降低 1-|3-|4 路径的 TQ 值 , 使得通过该条路径传输的视频 图像卡顿。 适当调节衰减器和发送功率 , 因为在实验设定中 , TQ 值的计算窗口和 OGM 的发包间隔都缩短了 , 所以很容易达到如下状态: 当障碍物移动到 1-|2 链路上时 , BATMAN-adv 会自动切换传输路径至 1-|3-|4; 反之 , 当障碍物移动到 1-|3 链路上时 , BATMAN-adv 会自动切换到路径 1-|2-|4 上。 这样以一定的频率来回 移动障碍物就会造成路由震荡 , 无法正常传输数据 , 视频无法观看。 { 41 % : 而实际上 , 此时总是至少存在一条有效路径可以进行数据传递的。 }

分别对 BATMAN-adv 所使用的固定阈值和本章所提出的动态阈值算法做一组 实验 , 每组实验以 10 秒的周期来回移动障碍物 , 持续 10 分钟。 实验中设定 h0 为 5。

可以发现,引入动态阈值标量 h 后,前期 (0.2 分钟) 路

由抖动仍然存在,之后切换会越来越缓慢,且越来越少,最后一直维持在单一路由

上。结合公式 4-5 分析如下: 在整个实验过程中,两跳链路的 TQ 值都随着障碍物的移动呈现周期性的波动状态,如图 4.30(a)。实验刚开始时,数据传输在 1-]2-]4 上进行,此时链路 1-]2-]4 的 c 值为 1,链路 1-]3-]4 的 c 值为 0。当障碍物移动到 1-]2-]4 路径上时,该路径的 TQ 值下降至低于路径 1-]3-]4,于是传输路径切换至 1-]3-]4。之后当障碍物移动到另一条链路后,传输路径再次切换。值得注意的是,在整个实验过程中,TQ 值最差也不低于 160,这个 TQ 值是可以保障我们实验中的视频稳定传输的,然而因为路由频繁切换导致视频无法传输。在前两分钟内这个切换的过程对于固定阈值和动态阈值的没有差异,都会导致路径近似于周期性的切换。但在这个过程中动态阈值计算公式中的 c 值在逐渐变大,当到达两分钟左右的时候, c 值增大到 3,动态阈值此时的大小要比初始时高出 b^3 ,这就保障了路径切换的标准增大,在实验中表现为路径维持在路径 1-]3-]4 上。如图 4.30(b) 所示,途中路径编号 1 为 1-]2-]4,2 为 1-]3-]4。

另一个可能的疑虑是当 TQ 值不高,但因为频繁切换后 c 值升高,导致动态阈值整体较大,会不会导致此时即使出现一条 TQ 值很高的路径,仍然会受制于动态阈值而无法切换呢? 实际上,在实际实验中如果 TQ 值较低,那么 c 值很难达到一个很大的值。假设此时两跳路径 TQ 值较低,均在 150 以下,向同一条路径切换两次后该路径 c 值为 2。此时动态阈值就会显著提高 b^2 ,如果再次切换后 TQ 值必然已经达到一个较高的水平。

B. 各项参数对动态阈值的影响

参数影响之 a 。 a 的作用是根据 TQ 值的不同,调整动态阈值的大小。当 TQ 值较小时,倾向于更快的切换到较好的链路上,因为此时链路无法很好的支持数据传输。动态阈值变小会导致一定时间的路由波动的可能性,关于波动的抑制可以通过 c 值来调节。另一方面,当 TQ 值较大,此时显然链路足以支撑当前数据的正常传输,因此倾向于保持在该条链路上,即使此时存在一个更好的链路也可以

暂不考虑。{41%: a 取值偏小时会导致 h 收敛速度慢,偏大则可能导致无法收敛到合理的值。} 实践中一般建议取值范围为 $[2, 3]$ 。

参数影响之 b 。 b 的作用是辅助 c 值达到合理的收敛速度,和 a 类似,如果取值偏小会导致 h 收敛速度慢, {41%: 取值偏大则会导致 h 无法收敛到合理值。} 实践中一般建议取值范围为 $[4, 6]$ 。

参数影响之 c 。 c 的作用是当链路出现波动的时候起抑制作用,比如当 TQ 值较小导致 a 值较小进而导致链路波动时,在链路波动的过程中,每一次切换相应链路上的 c 值都会增加,而 c 值的增加会带动动态阈值 h 的增加,从而抑制之后的链路切换。

值得注意的是,我们在公式中没有限定 c 值的范围,在上一节也涉及过是否会出现 c 值较大导致链路停滞在较差水平而不进行切换。现做如下讨论: 因为 TQ 值最大为 255,现假设 TQ 值从 0 开始向较高 TQ 值链路切换。初始阶段因为 TQ 值和 c 值均为 0,所以只有 h_0 一项,假设 h_0 为 5,也就是说此时的动态阈值 h 的值为 5。那么当另一条链路 TQ 值大于 5 就会完成第一次切换,新的链路动态阈值为大于等于 $a \log 5 + 5$ 。完成第二次切换后,新的链路动态阈值变为大于等于 $a \log(a \log 5 + 5) + 5$ 。当切换至之前的路径时, c 值大于等于 2, $\log c$ 开始发挥作用。新的链路动态阈值也就增加了 $b \log c$ 项。假设 a 取值为 2, b 取值为 3,那么在 8 次切换后,动态阈值 h 的值将达到 25,而此时选择的链路质量也会显著好于之前的链路。整个过程的各参数值变化

参见表 4.5，其中最后一行的 258 加号，为假设推导项。因为 TQ 值最大为 255，达到 255 后不再增加，所以也就不会再发生路径切换。从表中可以看出动态阈值 h 收敛于 25，表中所推导的数据是在 TQ 值理想最低情况下，可以判定动态阈值 h 最高收敛值为 25，收敛条件是 TQ 值接近 255，亦可判定 c 值不会超过 6。进一步地，当节点掉电或者其它原因导致某条链路从路由表中移除，重新构建该链路后所有参数亦重新初始化。

参数影响之 h_0 。 h_0 的作用和固定阈值相近，但它的主要目的是为了在 TQ 值和 c 都相对较小时，提供一个阈值的下限，加快收敛速度。所以它的取值相对于固定阈值的设定也会相对较小。实践中一般建议取值范围为 $[5, 10]$ 。

4.4 本章小结

本章详细介绍了系统的实际实现，尤其是本项目的三个核心创新点。4.1 节针对无线信道干扰的问题设计了子网信道隔离的部署方案，将上百个节点组成的全干扰网络划分为 10 个互相独立的子网，每个子网内自组织形成 Mesh 网络，上层通过远距离无线传输设备将所有子网的数据通过簇首节点传输至汇聚

节点。这样的网络规划可以最大限度地发挥当前 Mesh 路由协议的承载能力，同时充分利用 5GHz 频段的频谱资源。实现网络性能的巨大提升。实验显示，划分子网进行信道隔离后，原先近乎瘫痪的全干扰网络可以流畅的传输所有视频数据。

4.2 主要工作是设计了一种包级别的视频帧映射算法，用以提升在工业视频监控场景下的 Mesh 网络视频传输性能。该映射算法综合考虑了多种对视频解码有影响的参数，如视频帧格式、视频帧编码结构、帧序列以及头数据包的特殊性等，设计公式 4-3 综合以上参数，生成每个数据包对应的权重。得到数据包权重后，结合队列的占用情况，将不同的数据包插入合适的发送队列。最后，设计并实现了上述算法，在真实的系统中长期运行。相对于传统 EDCA、静态映射算法和动态映射算法，该映射算法在 PSNR 值上分别能够提升 50%，16%，5%。

4.3 首先从 BATMAN-adv 的路由算法及链路质量判定方法引出该路由机制在漫游场景下对低延迟要求的服务无力支持的窘境。随后介绍了一种优化方案，该方案中通过短暂的复用之前建立的有效路由，完成路由重建的时间间隙中的通信需求，在这个时间间隔中，所有到达漫游节点的数据由之前的接入点引导到当前接入点，从而完成通信。该方案在瞬时漫游场景下表现良好，但是如果漫游节点在两跳链路之间缓慢移动，那么漫游过程中的路径切换时延将无法避免。一个直观而有效的方法是缩减 TQ 值计算窗口，同时减小 OGM 的发包频率。这两种方法都能够增加路由计算对链路质量变化的敏感度，从而更快的在上述场景下完成路由切换。但是另一方面，增加链路敏感度可能导致路由震荡，该现象往往出现在质

量相近且波动频繁的链路之间。对于视频传输，路由震荡可能导致视频卡顿，甚至完全无法观看。同时存在遭受 DOS 攻击的风险。

针对这一问题，本章提出了一种新的动态阈值标量，定义为 h ，并在公式 4-5 中阐明其计算方法和意义。虽有通过实验对比动态阈值标量和静态阈值之间的性能差异。实验显示，动态阈值的性能提升是巨大的。

之后，讨论了公式 4-5 中的参变量 a 、 b 、 c 、 h_0 取值不同对动态阈值 h 的影响，影响主要是从收敛速度，确保收敛至合理值等方面讨论，最后给出了不同参变量的合适的取值范围。

第 5 章 总结与展望

回顾全文，首先在第一章介绍了项目的背景和意义，无线 Mesh 网络的快速发展对人们的生产生活所产生的影响越来越大，尤其在一切基础网络设施缺乏的特殊场合，比如灾后重建、野外大规模作业等，Mesh 网络能够充分发挥其易部署、低成本、高可靠性的特点，为这类工作提供很好的支持。然而 Mesh 网络因为其特殊的网络构建方式和无线信道的开发性，导致难以提供 QoS 保障。 { 42 % : 进而第二章介绍了相关领域的研究现状，以及目前 Mesh 网络的主流协议。 }

{ 45 % : 在此基础上，第三章提出了文章三个核心的创新点: } 子网信道隔离的部署方案、跨层视频帧权重差分技术、路径质量敏感的动态切换阈值算法。三个创新点围绕提升 Mesh 网络整体 QoS 展开。子网信道隔离的目的是在网络规划层面优化整体网络性能；视频帧优先级队列映射的作用在于对视频帧进行不同的权重分解，以保证高权重的数据包能够优先传送；移动场景下的 QoS 保障解决的问题是增强协议对链路质量变化的敏感度，从而缩减漫游中的路径切换时延，同时优化路由震荡的问题。

{ 41 % : 第四章分三部分详细介绍了三个核心创新点在实际无线 Mesh 网络系统中如何设计实现。 } 自网信道隔离通过在 5GHz 频段选取五个相互正交的信道，使得相邻子网之间相互隔离，并在 Mesh 子网上层架设一层无线桥接网络。这样的实现不仅充分利用了 5GHz 频段的频谱资源，还将大的网络切割为较小的 Mesh 子网，这种模块化的思想为后期的网络管理与维护及网络规模的扩建提供了支撑。跨层视频帧权重差分技术设计了一个跨层的优先级队列映射机制。视频源采用 GOP 分层编码，第一跳节点计算每一个视频帧数据包的权重，计算过程结合帧的类型、帧的位置以及是否为帧的头数据包。计算完成后将权重值存储在网络层包头的 ToS 字段，后继节点不需要计算转发数据包的权重值。之后在发送阶段映射模块会根据权重值将数据包插入合适的队列。路径质量敏感的动态切换阈值算法针对节点漫游时可能产生的路径切换时延，重新调整了原 BATMAN-adv 协议的 TQ 值滑动窗口和 OGM 发包频率。并设计实现了动态阈值标量的计算模块，从而避免了因为提升链路质量敏感度导致的路由震荡。在介绍三个创新点的设计实现中贯穿了大量时延以验证不同的功能带来的性能上的提升。

最终的实现的系统相对于之前的原始系统在整体系统性能上达到极大的提升。子网信道隔离使得实验 Mesh 网络的性能提升达到 10 倍以上。视频真优先级队列

映射针对视频质量度量指标 PSNR 的相对于传统 EDCA 方式提升分别在 50% 以上。移动场景下的 QoS 保障使得短路径到长路径的切换时延从 30 秒缩减到 4 秒左右，并有效抑制了路由震荡。

5.2 工作展望

{ 40 % : 本文的工作部分解决了目前工业界大规模无线 Mesh 网络应用中的 QoS 保障的问题，单仍然存在很多可以优化和改善的地方。 } 比如子网信道隔离，虽然充分保障了子网之间的相互零干扰，但是占用了过多的信道资源，可能会干扰其他非 Mesh 网络设备的正常工作，这一点我们在伊拉克实际部署时就已经遭遇过。如何在占用更少信道资源的情况下保障干扰最小化是一个值得研究的课题，它的核心其实就是信道分配算法，学术界对这个问题也有过很多的研究工作。再如跨层视频帧权重差分技术，引入了 802.11 e 的 EDCA 机制，尝试了新的权重计算方法和映射机制，但是也还存在一定的局限性，比如现阶段只适用于视频监控等少数场景，另外多跳路径上不同节点汇入的数据如何进行负载均衡也是一个值得研究的课题。最后路径质量敏感的动态切换阈值算法，做到了在路径切换延时的大幅优化和路由震荡抑制，但是显然 4 秒的延迟对于时延敏感的数据服务仍然存在巨大的提升空间，同时路由震荡本文的方法抑制效果在前期并不理想，是否存在类似毒性逆转等传统网络中使用的方法可以更好的解决此问题也是值得进一步研究的。 { 46 % : 总之，本文的工作取得了显著的成果，但还存在大量的问题值得研究。 }

我们也看到，现在无线 Mesh 网络技术发展迅猛，国内外很多公司都加入 Mesh 网络研发的队伍，比如华为、微

软、思科、Aruba、Strix 等。 学术界也一直在研究 其中的路由、安全、QoS 保障等方面的问题，典型的项目有 MIT 的 RoofNet 项目、约翰霍普金斯大学的 SMesh 项目等。 IEEE802.11 几年前已经成立了 Mesh 网络工作组 802.11s，该标准目前正在稳步推进。 { 52 % : 随着相关的研发工作的投入，有理由相信无线 Mesh 网络将在人们的日常生产生活中发挥越来越重要的作用。 }

检测报告由PaperPass文献相似度检测系统生成

Copyright 2007-2016 PaperPass