论文题目：针对多媒体流的802.11可靠多播LBP协议改进研究

专　　业：计算机软件与理论

硕 士 生：李钊文

指导教师：周晓聪 副教授，尹冬生 讲师（第二导师）

摘　要：

Leader-Based Protocol（LBP）是最经典的802.11可靠多播协议之一。近年来基于LBP协议的研究和改进工作在世界各地都有进行，但针对多媒体流进行协议优化的研究工作甚少。本文针对多媒体流对低延迟、低抖动和稳定传输的需求，对LBP协议进行优化，提出Multimedia Leader-Based Protocol（MLBP）协议，增加基于概率的错误重传恢复机制，基于优先级的QoS缓冲队列调度算法，以及多媒体多播流资源预约机制等。通过ns-3网络仿真实验证明，MLBP协议更适合用于多媒体流的传输，同时也提供可靠的链路层传输。

关键词：802.11，多媒体，可靠多播，Leader-Based Protocol

Title: Improvement research of 802.11 reliable multicast protocol LBP based on multimedia stream

Major: Computer Software and Theory

Name: Zhaowen Li

Supervisor: Associated Prof. Xiaocong Zhou

Lecturer Dongsheng Yin (Second supervisor)

Abstract:

Leader-Based Protocol (LBP) is one of the most famous 802.11 reliable multicast protocols. In these years, research and improvement jobs based on it have been done all around the world. But contribution on how to optimize the protocol based on multimedia stream was rarely proposed. This article faced the need for low delay, low jitter and stable transmission of multimedia stream, optimized the LBP, and proposed a new protocol called Multimedia Leader-Based Protocol (MLBP), with additional functiones called probability-based retransmission recovery, priority-based QoS queue scheduling algorithm, and resource reservation mechanism of wireless channel. The ns-3 simulation experiments proved that MLBP is more proper for the multimedia stream, and it does provide reliable link layer transmission, too.

Key Words: 802.11, Multimedia, Reliable multicast, Leader-based protocol

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：李剑文

日　期：2010.6.6

学位论文使用授权声明

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅，有权将学位论文的内容编入有关数据库进行检索，可以采用复印、缩印或其他方法保存学位论文。

学位论文作者签名：李剑文　　导师签名：[illegible]

日期：2010年6月6日　　日期：10年6月2日

引 言

IEEE（电气与电子工程师学会）在历时五年的研究和审查之后，终于在2009年9月正式批准802.11n为高速无线局域网标准。在该标准支持下的产品理论速率为300Mbps[24]，相比原来的802.11a/b/g，更适合用于传输大流量的数据业务，尤其是多媒体甚至是高质量多媒体这一类的对带宽要求很高的业务。

从2008年底开始，中国移动作为中国最大的移动运营商，提出全面推广Wi-Fi业务，采用802.11b标准的技术，从之前的只面向商务客户，逐渐推广到普通大众用户[31]。而据中国移动的一份内部招标文件透露，2009年中国移动将新建WLAN热点10.8万个，其中胖AP约1.8万个，瘦AP约9.06万个，知情人士称，这些设备均兼容WAPI标准。同时中国移动集团将在2009年各省的建设资金中设置宽带接入预覆盖建设专项资金，以加强TD + Wi-Fi网络建设[32]。而据iSuppli公司的最新数据显示，截至2009年底，中国电信的Wi-Fi热点数量已经达到9.5万个，相比2008年底的3万多个，增长超过200%；中国移动的Wi-Fi热点数量达到9万个，相比2008年底的1万多个，增长了近800%；而中国联通的Wi-Fi热点数量增加到2.5万个[33]。从以上所有数据看来，无线Wi-Fi技术的部署和全面推广，已经是大势所趋。在不久的将来，无论在繁华的闹市中心，还是在恬静的民居小巷，也许都能被Wi-Fi的无线信号覆盖，普罗大众也能享受到方便快捷而且高质量的无线网络接入服务。

2007年1月10日，苹果在2007 Mac World大会上正式发布了其旗下的第一款手机产品iPhone[34]。2010年1月28日，苹果在加州旧金山前卫艺术中心举行发布会，正式发布了传闻已久的平板电脑，型号是iPad[35]。无论是iPhone还是iPad，还是后继的众多厂家的模仿或自主创新的，准备与苹果一争高下的电子终端产品，都离不开无线Wi-Fi接入的功能，因为这些移动终端产品的本身就是定位为便携式，不可能长期使用有线网络进行连接。而在这些手机或者平板电脑的终端上通过无线网络使用在线多媒体业务，就离不开Wi-Fi这种高带宽高速率的无线接入方式了。虽然目前有3G这种新兴的手机上网接入方式，但其高昂的费用以及有限的带宽，使其具有很大的局限性。而随着这些新产品的推广和流行，

越来越多的用户会使用这些便携式的无线终端产品进行网络访问，所使用的服务，除了传统的浏览网页和收发邮件服务之外，更多的是新兴的在线多媒体服务。根据《中国互联网络发展状况统计报告》[25]显示，截止到 2009 年底，网络视频用户规模达到 2.4 亿，较 2008 年底增长 3844 万，使用率为 62.6%，下降了 5.1 个百分点。而在手机用户上网的应用类别中，手机电视所占比率为 16.5%，而手机在线收听音乐或下载的比率更是高达 50.4%。从此可以看出，在无线终端的网络应用中，不乏大量的用户对多媒体应用需求。无线局域网作为最后一公里[26]的其中一种接入技术，由于其逐渐的被广泛使用，而在其上承载多媒体业务便成为一种重要的需求。

802.11 无线多播的需求

在当今无线网络不断普及的时代，802.11 协议作为一种无线局域网的接入层协议，担当起了网络接入层的重要角色。如何在传统 802.11 无线接入层上顺利承载多媒体业务，并提供较好的用户体验，是 802.11 协议值得进行改进的一个方面。传统 802.11 协议一般使用单播方式进行数据传输，然而在有线以太网多年的发展历史中证明，多播方式对于解决多媒体分享问题具有其独特的优势，但 802.11 协议的多播机制一向被视为不可靠的[5]，不能作为一种常规的数据传输方式，甚至在许多网络产品中都自动禁用了多播功能。

未来的网络环境将会出现大量的可移动终端设备，并且在不同的无线基站覆盖范围之间移动。在同一个基站单元中，有可能会出现多个设备或终端是对某一个来自远端的业务流是感兴趣的，并且同时准备接收该业务流的数据。为了能更好地利用无线链路的带宽资源，使用多播形式转发该业务流的数据就显得非常有优势。在采用单播传输方式的情况下，每一个需要接收该业务流数据的终端站点都必须获得一份独立的数据拷贝，即同样的数据需要进行多份拷贝和分发，以满足每一个终端站点的需要，如图 0-1 所示。这不但大量耗费服务器的处理资源，也会造成在 AP 这种网络瓶颈位置容易出现流量拥塞，最终使得大量的终端站点不能正常接入无线信道，只有少数站点能获得服务。而采用多播情况下，无线 AP 只需要转发该业务流数据的一个副本就可以了，而不需要以单播方式发送多个副本给每一个需要接收的终端，节省了无线带宽资源，如图 0-2 所示。也不需

要像广播方式那样，使得其他不需要接收该业务流数据的终端也被迫要处理这些数据帧，而往往这些终端才是占该无线局域网的多数，使得大量的终端资源被浪费。

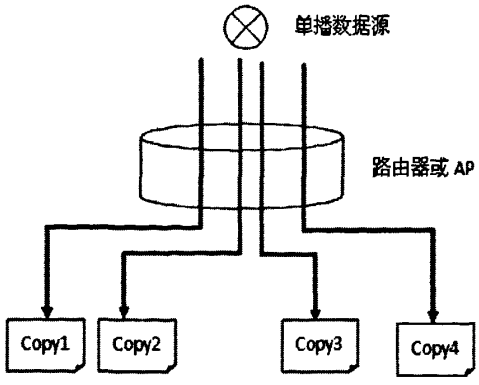


图 0-1 单播数据源流量示意图

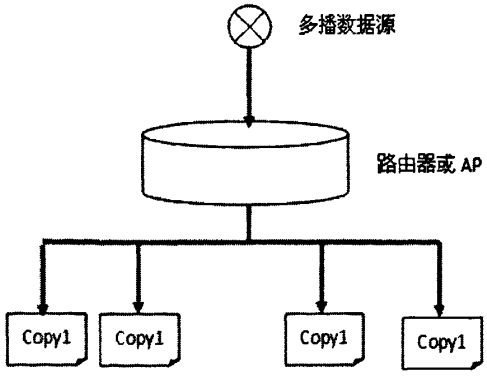


图 0-2 多播数据源流量示意图

许多的应用都适合使用多播这种方式进行数据传送，只要是单一的业务数据源同时分发数据到多个终端接收者的模式，多播方式都是非常具有优势的。例如多媒体视像会议、共享白板、远程教育、多人网络游戏、分布式计算等，而更加典型的一种应用，便是大型体育赛事或时政要闻的直播。

2010 年 11 月 12 日， 2010 年亚运会将会在广州隆重开幕。届时，40 多个亚运会竞赛项目将会在各大场馆逐一进行角逐。而在此期间，完成各项赛事的实时直播，便成为了广州亚组委需要特别关注的一项重要任务。为了让广大人民群众能更及时地收看到赛事的进展情况，除了传统的电视直播[36][37]和电台直播这

些媒体之外，还会有更多的移动的直播方式，如手机电视和无线终端接入互联网的在线视频等，所以无线网络的建设尤为重要。根据广州市越秀区和中移动广州分公司所签订的共建“信息越秀”信息化战略合作框架协议，亚运会前，广州市区将实现100%无线网络覆盖。在2010年亚运会召开之前，广州市将建成11000多个无线AP接入点、6820多个宽带移动通信网络接入点，实现无线宽带接入覆盖全城所有亚运场所及公共活动场所的目标，建成具有国际先进水平的无线城市框架[38]。

面对即将来临的如此盛大的体育赛事，无线网络承载多媒体的优势便显现出来。无线网络接入方式的随时性和随处性，使得大量的观众会使用无线终端进行赛事直播的收看或收听。而这样一种实时性的多媒体业务，自然符合单数据源分发到多个接收终端的模式，因此，如果用户手中的无线终端能支持多播模式，那么在同一个无线局域网覆盖范围内的用户，就不会因为大家同时访问同一个资源而造成网络拥塞，最终导致赛事观看的延误。多播模式将为无线网络用户提供非常好的实时多媒体服务。

针对多媒体优化

传统的端到端的传输层错误恢复过程是需要较长的延时的，这对于时延敏感的多媒体应用来说，就显得不太适合。但链路层的错误恢复则只需要非常短的时延，足够应付多媒体应用对时延的严格要求。所以研究链路层的错误恢复机制就显得非常有价值[1]。从本地和远端的两种错误恢复方式来看，本地的恢复显得更加快捷和及时，延时更短，而由远端进行恢复则显得反应太慢了。通常本地的错误恢复对于无线局域网环境来说，就是指在AP上的链路层错误恢复。

虽然可靠性的增加使得无线链路层多播不再是一块“鸡肋”，而变成了可以承载更多业务的一种数据转发方式，但当我们回想起最初发掘多播模式的动机时，更多的还是希望用其承载多媒体业务流。而对于多媒体业务流，尤其是在线多媒体这种追求实时性的服务来说，可靠性固然重要，但更重要的，是稳定性。低时延，低抖动，稳定带宽传输，这才是影响终端用户体验的最关键的因素，而这些方面的研究，也正是当前无线可靠多播协议领域所缺乏的。

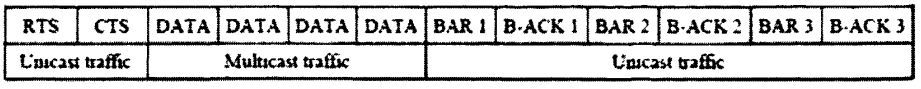
相关工作

Kuri.J 等[1]针对无线局域网链路层可靠多播机制容易产生确认帧冲突的问题，首先提出了一种新的可靠多播协议 Leader-based Protocol（LBP），利用选举 Leader 为代表节点进行确认信息的回复这种机制，解决了多播时确认帧冲突的问题，并在一定程度上保证了多播的可靠性。Kuri，J 等还对比了另外两种可靠多播协议 Delayed feedback-Based Protocol（DBP）和 Probability-Based Protocol（PBP），从实验结果上发现 LBP 在性能上更具有优势，成为了许多学者在无线局域网可靠多播的后续研究工作的基础。

Han-Chieh Chao 等[2]基于 Kuri.J 提出的无线可靠多播协议 Leader-Based Protocol[1]，结合另外一种同样是经典的无线可靠多播协议 Delayed feedback-Based Protocol 的思想，提出了 LBP 的改进协议 RLBP（Random LBP）。该协议将优先级延时的思想加入到 Leader 的选举方法上。在每一次的确认帧发送过程中，所有 Station 通过随机选取延时来决定确认帧发送的优先顺序。随机选取的延时最短的 Station 不但可以优先发送确认帧，并且同时担任新的 Leader，代表所有 Station 进行后续的帧确认操作。这使得 Leader 的产生过程随机化和即时化，不需要额外的选举过程，同时节省了 Leader 表的额外维护开销。但是 RLBP 协议容易产生一个问题，容易因网络延迟的问题出现多个 Leader 同时当选，从而产生多 Leader 冲突。为了解决这个问题，Dongkyun K 等[3]将 RLBP 继续进行改进，增加了 Leader 的互相抑制机制。在同时出现多个 Station 当选 Leader 的情况下，先当选者自动解除 Leader 角色，让位给后来者，使得最后一个认为自己当选 Leader 的 Station 可以顺利完成 Leader 的任务，而不会产生多 Leader 冲突，从而保证在同一个无线局域网内同一时间只有一个 Leader 存在。同时该文为 RLBP 增加了 cl-RTS（clear-leader RTS）帧，用于清除多 Leader 冲突的状态，发起重新选举 Leader 的过程，从而更进一步保证不会出现多 Leader 的混乱局面。

Andrey Lyakhov 等[6]提出了一种基于 802.11e 的多播协议。该协议采用的是 Leader-Base ACK 的模式。而且 Leader 不仅限于单个，而是设置多个 Leader。结合 802.11e 的 Block Data（B-DATA）和 Block ACK（B-ACK）的机制，使得多播数据帧可以多个帧同时在一次发送机会中连续突发（Packet Burst），在一定程度

上提高了吞吐率。然后 AP 对每个 Leader 进行轮询，发送 Block ACK Request（BAR)帧,请求确认对 B-DATA 的确认。每个 Leader 收到 BAR 之后发送 B-ACK 进行确认。完成对所有 Leader 的轮询后才结束该次的帧发送过程。协议的时序图如图 0-3 所示：



ELBP burst structure (B=4, J=3)

图 0-3 ELBP 帧突发序列

从图上可以看出，只有 B-DATA 帧是以多播形式发送，RTS/CTS 握手过程，以及 BAR/B-ACK 的轮询确认过程依然是采用单播形式进行。虽然 BAR/B-ACK 的轮询确认过程提高了原来单 Leader 的 LBP 协议的可靠性，但同时也消耗了更多宝贵的带宽资源。而且对于 RTS/CTS 握手过程，A. Lyakhov 等只是简单地用随机选取一个 Leader 进行单播来解决，并没有深入去解释其中会出现的问题。

Gupta 等[7]提出了一种称为 Dual Busy Tones 的变式 NAK 的方式，解决了多播方式下确认帧冲突的问题。由于采用的是 NAK 否定确认的方式，所以 ACK 和 CTS 这两种确认帧都不需要了。由于其变式 NAK 采用的是 Busy Tones 信号的方式，并不是真正的帧，就算产生冲突后信号重叠，其否定确认的功能并没有丢失，所以并没有所谓冲突造成丢帧的问题，从而在根本上解决了确认帧冲突的问题。既避免了使用确认帧，同时这种多播机制又拥有一定的可靠性，使得该协议成为了一种新型的可靠多播协议。该协议的时序如图 0-4 所示。

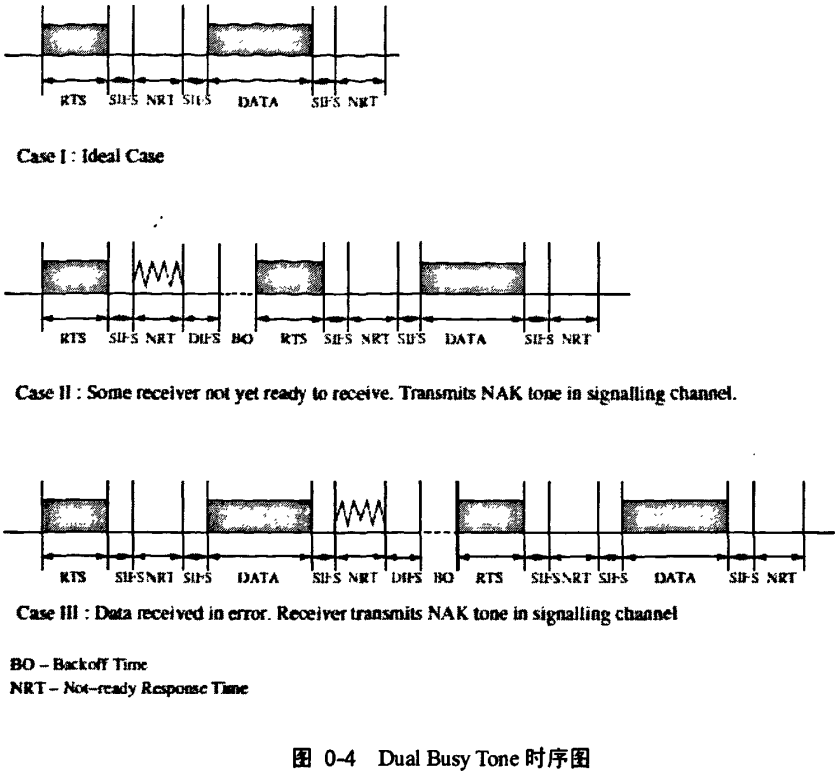


图 0-4 Dual Busy Tone 时序图

每当 RTS 帧或 DATA 帧发送之后，AP 都会等待一段时间 Not-ready Response Time（NRT），如果任何一个 Station 未准备好接收 Data 帧，或者不能正确接收到一个完整的 Data 帧时，就会发送 Busy Tone，以引起 AP 注意，启动错误恢复。

Zhao Li 等结合他们之前提出的 Beacon-driven Leader Based Protocol（BLBP）协议[8]，加上 Packet-Level 的 FEC 前向校验机制，提出了一种新的混合 LBP 协议 Hybrid Leader-Based Protocol（HLBP）[4]。在 BLBP 协议中，Zhao. Li 等为原始的 LBP 协议加入了 Beacon 帧，捎带了 Data 帧的序列信息和延时信息，使得每个 Station 都能辨别即将收到的 Data 帧是否已经接收过，即使该 Data 被损坏发生丢帧。这种情况下许多不必要的重传得以避免，减少了无线链路的带宽消耗。而在 HLBP 协议中，Zhao. Li 等更增加了数据帧级的前向校验（Packet-Level FEC），使得在较高丢帧率、信道恶化严重的时候，该协议依然能保持低开销的多播帧传送，进一步提高了链路带宽的利用率。但该协议在确实需要重传恢复时，其开销是非常大的。

Xiaoli Wang等[9]在Leader-Based Protocol的基础上，提出显式指定Leader的过程，并且不再采用单Leader的方式，而是随机指定三个Leader进行确认帧的回复，改善了LBP协议单Leader的有限可靠性。在指定Leader的过程同时也捎带了Leader回复的优先级顺序，从而解决了多Station同时回复确认容易冲突的问题。但该协议并没有采用ACK帧，而是采用CTS帧捎带上一次收到Data帧的回复ACK信息的方式，进一步减少了信道带宽开销的同时，又保证了可靠性。但该协议对CTS帧和DATA帧的帧格式修改比较大，使得实现难度和成本可能都比较高。

Ping Ding等[10]从另一个角度出发，重新提出了一种多AP间的多播协调机制，不需要增加CTS帧和ACK帧等额外的feedback机制，完全使用原始的802.11无线多播机制，增加了一个基于MCDI（Multicast Collision Detection Interval，多播冲突检测间隔）的等待时间。于是，因多AP信号覆盖范围重叠产生的多播互相影响的情况将大大减少。但该协议并没有解决单AP覆盖范围内多播的可靠性问题，以及在出现大量单播背景流时，该协议是否能在信道资源竞争中保持稳定的多播流量的问题。更长的等待时间意味着更低的信道使用优先级，更小的机会争用到信道的使用权。

S.A.Khayam等[11]针对如何利用FEC来降低丢帧率，尽可能地挽救因信道质量恶化而造成校验错误的帧，而不是简单地将其丢掉。该文提到了先前的研究都忽略了甄别错误帧的过程，到底一个错误帧是payload错误还是header错误。他们企图利用同一个会话的前后帧的Header相似性，甄别出只是header错误的帧，从而挽救更多payload完整的数据帧，而在其他协议中这些帧都是会被直接丢弃的。在该协议下，重传次数明显减少，信道利用率和吞吐率都得到了提升。

F.Fu等[12]从无线局域网的接入点入手，设计新的资源分配管理协议，使得多个Station互相竞争接入时更加公平。该文将所有Station都假想为理智而且贪婪的，在任何情况下都极想争夺最多的无线信道资源，甚至会欺骗基站来获得更多的信道使用时间。但该协议迫使所有Station都要为其已经争用到的资源付出代价。F.Fu等提出了一种新的公平性评判标准，不是根据Station用了多少资源，而是根据对其他Station的接入造成了多大的不便和损失，如延迟和丢帧，来判定在后续的信道争用过程中该Station需要牺牲多少接入机会来进行弥补，

以求获得总体上的公平。F.Fu等称之为一种willingness-to-pay的机制，每一个Station要愿意付出代价，才能获得更多的资源，所有station都必须评估自己获取资源后所产生的风险。实验证明在这种机制下传输video的质量可以得到明显的提高。

Li Y.等[13]提出了一个Content-Aware的思想，直接关注到应用层的层面。通过分析视频帧的细节，尤其是关注动作和场景间切换比较多的片段，提升其优先级，将相对变化较少甚至静止的片段降低优先级，以此作为根据来重新安排帧的封装、分段和发送等，并且做成一种自适应机制，自动动态调整帧的安排，以此来提高视频流在信道质量较差的无线局域网中传播时的质量。

Lohier, S.等[14]提出了两种区分无线链路丢帧原因的算法，称为Loss Differentiation Algorithms(LDA)。一种是基于MAC层的LDA，主要是用Signal to Noise Ratio（SNR）作为参照指标。因为距离增大或者出现障碍物而导致信号衰减，还是因为信道拥塞而导致丢帧，在同时期其他所接收到的帧的SNR值是不同的。SNR值大的时候，证明接收信号功率依然很大，不可能是因为信号衰减，此时则判定为拥塞丢帧，反之则可认为是信道质量下降导致信号衰减，判定为信号衰减丢帧。当因为信号衰减而丢帧时，则增大重传限制次数，企图在传输层发生超时之前，信道质量重新恢复稳定，从而在链路层就能实现错误恢复，不必经由传输层发起重传恢复，减少了错误恢复的时延。当因拥塞丢帧时，则直接放弃重传，避免进一步增大信道负荷，让传输层来负责拥塞控制和重传恢复的任务。另一种算法是基于传输层和MAC层的跨层实现。在传输层利用Round Trip Time（RTT）来计算，但该计算结果并不准确反应信道情况，所以要结合MAC层的重传次数来联合判定，如果RTT增大而且重传次数达到上限，则认为是因为信道质量恶化，这样就没必要在传输层缩小争用窗口进行拥塞控制和降低发送速度，而是增大重传阈值，让链路层继续尝试进行重传恢复，减少时延。而当重传还未达到上限时RTT已经增大，就判定为拥塞原因，此时就采用传输层的拥塞控制机制进行解决。

Sungjoon Choi等[15]针对无线局域网中多播发送速率低的问题，提出了一种新的自适应的速率调节机制Leader-Based Multicast with Auto Rate Fallback protocol（LM-ARF）。在无线局域网中多播默认是按照广播的方式发送的，即只

能采用基本速率（802.11a的基本速率为6Mbps），为的是能够让覆盖范围内的所有站点都能兼容该速率。S.Choi等利用前人已经提出的一种ARF机制，结合可靠多播协议Leader-Based Protocol运用到无线网可靠多播机制上，实现了新的协议LM-ARF，而且该协议可以和现有的802.11体系兼容，同时还考虑到了流量公平性和移动的问题。

本文的主要贡献以及内容安排

本文主要提出了一种新的无线可靠多播协议MLBP（Multimedia Leader-Based Protocol），该协议在原有的LBP协议基础上进行改进，增加了对多媒体业务流的支持，使其在承载多媒体业务流时能够获得更稳定的吞吐率，更小的时延和抖动。

本文将在第一章对802.11标准的体系结构进行陈述，主要针对与本文相关部分的内容如MAC层的实现等进行详细的解释。第二章将对本文所采用的实验平台ns-3网络仿真系统进行介绍，特别针对本文协议MLBP具体进行实现的模块，即无线Wi-Fi网络设备模型进行讲解。第三章将会对三种经典的无线可靠多播协议进行介绍和分析，包括Leader-based protocol，Delayed feedback-based protocol和Probabilistic feedback-based protocol等。第四章将会引入本文设计的MLBP协议，并介绍其协议实现细节，包括协议针对多媒体进行优化改进的三种新机制：基于概率的信道错误重传恢复、基于优先级的QoS缓冲队列调度算法以及多媒体多播流资源预约机制。第五章将会对MLBP协议进行系统建模，通过数学模型分析其性能并推导出分析结果，从理论上证明MLBP协议的优势。第六章将会介绍本文的仿真实验的部署和设定，还有多种实验方案的设计，包括信道质量恶化测试实验和高负载背景流拥塞测试实验两大类，以及相应的实验结果展示及分析结论。同时还会介绍一种构建基于ns-3仿真平台的可控丢帧率无线信道的方法。第七章将会对全文进行总结并提出展望。

第1章 802.11标准体系结构

主流的无线局域网产品最早来自于IEEE组织1997年6月批准的802.11标准[24]。和其他IEEE 802系列协议相同，802.11只涉及ISO/RM七层网络模型中的最低两层——物理层和数据链路层的介质访问控制子层。802.11标准的逻辑结构如图1-1所示，其中阴影部分为802.11标准涉及的部分。每个采用802.11标准的无线站点都包括一个LLC子层、MAC子层和5个PHY层(跳频扩频FHSS、直接序列扩频DSSS、红外Infrared、正交频分复用OFDM、多输入多输出MIMO)中的一个。

数据链路层	LLC子层				
	MAC子层				
物理层	跳频扩频 (FHSS)	直接序列扩频 (DSSS)	红外 (Infrared)	正交频分复用 (OFDM)	多输入多输出 (MIMO)

图 1-1 802.11标准体系结构示意图

由于本文所设计的协议及所部署的实验主要基于802.11a标准，以下只针对802.11a标准作简单介绍。

1.1 802.11a标准

IEEE 802.11a是802.11原始标准的一个修订标准，于1999年获得批准。802.11a标准采用了与原始标准相同的核心协议，工作频率为5GHz，使用52个正交频分多路复用副载波，最大原始数据传输率为54Mb/s，达到了现实网络中等吞吐量（20Mb/s）的要求。如果需要的话，数据率可降为48，36，24，18，12，9或者6Mb/s。802.11a拥有12条不相互重叠的频道，8条用于室内，4条用于点对点传输。它不能与802.11b进行互操作，除非使用了对两种标准都采用的设备。

由于2.4GHz频带已经被到处使用，采用5GHz的频带让802.11a具有更少冲突的优点。然而，高载波频率也带来了负面效果。802.11a几乎被限制在直线范围内使用，这导致必须使用更多的接入点；同样还意味着802.11a不能传播得像802.11b那么远，因为它更容易被吸收。

尽管2003年的世界无线电通信会议让802.11a在全球的应用变得更容易，不同的国家还是有不同的规定支持。美国和日本已经出现了相关规定对802.11a进行了认可，但是在其它地区，如欧盟，管理机构却考虑使用欧洲的HIPERLAN标准，而且在2002年中期禁止在欧洲使用802.11a。在美国，2003年中期联邦通信委员会的决定可能会为802.11a提供更多的频谱[24]。

1.2 802.11无线网络MAC层协议

由于无线网络环境相对有线网络环境显得更错综复杂，802.11的MAC子层协议与以太网的MAC子层协议是有很多差异的。在以太网中，每一个工作站都只需要等待线路静默下来就可以开始传送数据了。如果在一开始的64比特数据没有收到一个突发的噪声反馈信号，则整个数据帧都可被认为正确传送了的。但在无线网络中，情况并不是这样简单的。

802.11的接入采用CSMA/CA（载波侦听多路接入/冲突避免）机制，与有线以太网如802.3采用的CSMA/CD（载波侦听多路接入/冲突检测）机制不同的是，CSMA/CA更倾向于尽可能避免冲突的发生，但CSMA/CD则只考虑冲突发生时的检测机制，不会在意冲突的发生会带来其他的副作用，如发送节点的功率浪费，而这些都是无线节点需要考虑的[17][18][23]。

1.3 CSMA/CA

由于无线信道传输条件的特殊性，以及信号强度变化范围非常大，就使得发送站无法像有线局域网一样使用冲突检测的方法来确定是否发生了冲突。IEEE 802.11规定介质访问控制子层采用冲突避免（CA）协议，而不是冲突检测（CD）协议。为了尽量减少数据的传输碰撞和重试发送，防止各站点无序地争用信道，无线局域网中采用了与以太网CSMA/CD相类似的CSMA/CA（载波监听多路访问/冲突防止）协议。CSMA/CA通信方式将时间域的划分与帧格式紧密联系起来，保证某一时刻只有一个站点发送，实现了网络系统的集中控制。

因传输介质不同，CSMA/CD与CSMA/CA的检测方式也不同。CSMA/CD通过电缆中电压或相位的变化来检测，当数据发生碰撞时，电缆中的电压或相位就会随着发生变化；而CSMA/CA采用能量检测（ED）、载波检测（CS）和能量

载波混合检测3种方式检测信道是否空闲。[27]

CSMA/CA协议主要使用两种方法来避免碰撞：

- 设备欲发送帧，且侦听到信道空闲时，维持一段时间后，再等待一段随机的时间依然空闲时，才送出数据。由于各个设备的等待时间是分别随机产生的，因此很大可能有所区别，由此可以减少冲突的可能性。
- RTS-CTS 握手（handshake）：设备欲发送帧前，先发送一个很小的RTS（Request to Send）帧给目标端，等待目标端回应CTS（Clear to Send）帧后，才开始传送。此方式可以确保接下来传送数据时，不会发生冲突。同时由于RTS帧与CTS帧都很小，让传送的无效开销变小。

1.4 DCF

DCF（Distributed Coordination Function）是802.11无线标准的MAC技术的基础。DCF引入了CSMA/CA的思想，并加上了二进制指数退避算法、虚拟侦听和RTS/CTS握手机制以及主动确认机制等，建立起一种可靠的802.11无线MAC层的传输方式[28]。

1.4.1 冲突避免——帧间间隔IFS（InterFrame Spacing）

802.11定义了四种IFS，每一种IFS都有其特定的用途。在所有帧（包括数据帧和控制帧）发送之后，所有工作站都必须等待一段定义好的时间才能进行下一帧的发送。通过不同长度的IFS的定义和区分使用，使得802.11标准设计的帧都按照一种优先级顺序进行信道争用，而不会造成不同类的帧之间发生冲突[17]。以下按照四种IFS的时延长短逐一介绍：

- **SIFS**

首先最短的是SIFS。这是用来允许一些相对独立的站间对话能够不受干扰地顺利进行。这里包括接收机收到RTS之后回复CTS要等待的时间是SIFS，接收机收到数据帧之后回复ACK要等待的时间是SIFS，还有在分段并发传送时发送机收到ACK后继续发送下一数据帧的等待的时间也是SIFS。因为SIFS是最短的IFS，所以采用这种IFS进行帧交换的过程相当于是优先级最高的，最先抢占到信道的使用权的[16][17]。

- **PIFS**

PIFS 是比 SIFS 稍微长一点的第二级别的 IFS。当基站需要发送信标帧或者轮询帧时，就需要等待 PIFS，由于 PIFS 是除了 SIFS 之外最短的 IFS，比 DIFS 还要短，所以 PCF 的轮询过程就不会被其他工作站企图使用 DCF 的争用模式而打断，基站可以更快地抢占信道进行轮询过程。但由于 PCF 在目前的大多数实际产品中都越来越少使用，所以这种 IFS 逐渐被人忽略，或者被其他新协议进行利用[16][17]。

- **DIFS**

这是 DCF 模式下的每一个工作站争用信道时所使用的 IFS，每一个工作站需要发送数据帧时，都必须在上一个在信道上传输的帧结束传输后再等待 DIFS 才能尝试争用信道以便发送新的一帧。如果发生冲突，二进制指数退避机制也要加入其中[16][17]。

- **EIFS**

这是最长的一个 IFS，如果一个工作站收到了一个坏帧，就需要等待一个较长的 EIFS 时间再采取下一步的动作，因为此时可以认为信道上的干扰是非常严重的，与其继续发送帧而且极大可能会丢失的，倒不如节省功率等待一段较长的时间期望信道恢复正常再继续发送[16][17]。

1.4.2 二进制指数退避算法

虽然 DCF 定义了 DIFS，作为所有 Station 争用信道接入权时的等待时间，但仍然会很容易出现 Station 间的冲突问题。当 Station 侦听到信道繁忙时，有可能会对帧的发送过程造成时延。但 Station 也很容易会侦听到信道重新恢复空闲的时候，于是所有 Station 等待同样的 DIFS 延时后就会再次造成冲突。DCF 为此定义了一种指数退避算法[16][24]。每一个 Station 在侦听到信道空闲并等待了 DIFS 的时延之后，自行计算一个随机时延值，如

BackoffTime = Random × aSlotTime　公式 (1-1)所示：

$$\text{BackoffTime} = \text{Random} \times \text{aSlotTime} \quad \text{公式 (1-1)}$$

其中 aSlotTime 为预设值，一般由 AP 发布到局域网内所有 Station 进行统一，而 Random 函数则产生一个随机数，使得各个 Station 的退避时间都不一致，可以错

开争用接入的时间。每个 Station 若在该次退避中未能争用到信道的使用权，则记录下未倒计时完的退避时间，在下一次信道空闲时，同样先等待 DIFS 帧间隔之后，只需要继续等待完前次剩下的退避时间，如此往复，由于所剩的退避时间越来越短，其优先级也会越来越高，直到争用到信道使用权为止。以这种方式来进行 Station 的接入安排，不但可以避免冲突，还可以保证接入的公平性，不会出现 Station“饿死”的现象。DCF 的这种指数退避的时序过程如图 1-2[29]所示。

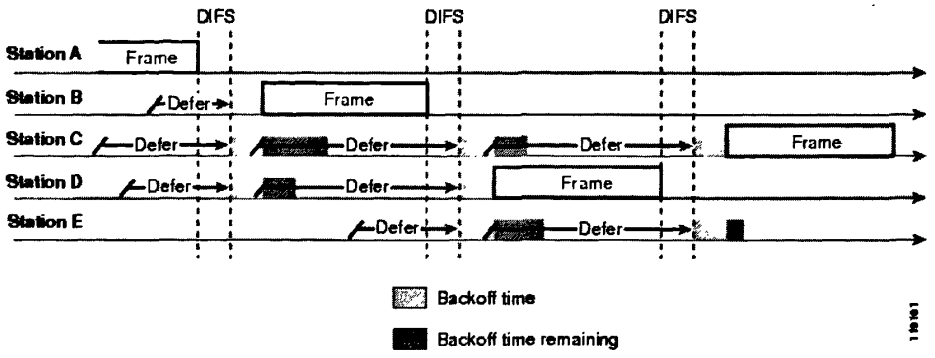


图 1-2　DCF 指数退避时序图

1.4.3 隐藏站/暴露站问题

无线局域网不得不说的一个潜在问题就是隐藏站和暴露站。因为不是所有工作站的无线电信号都可以互相覆盖，所以在一个小区里，一个工作站的传输数据的信号不一定能遍及该小区内的每一个角落。这时就会出现隐藏站和暴露站的现象[17]。如图 1-3 所示。隐藏站的问题可以举例描述如下，A，B，C 分别为三个无线工作站，各自的无线信号覆盖范围分别为 ra、rb 和 rc。当工作站 C 要传送数据给工作站 B，因为工作站 A 不在 C 的覆盖范围内，所以 A 侦听信道时并不能侦听到 C 的传输信号，A 就会断定与 B 之间的信道是空闲的，可以开始发送数据。一旦 A 也开始发送数据给 B 时，A 和 C 各自发送的数据就会在 B 处产生冲突，互相干扰，因此 B 无法接收任何一方传来的数据，导致 A 和 C 发送的数据都是无效的被丢弃的，但 A 和 C 都不知道这个情况的发生，继续发送数据，然后导致更多的数据丢失，信道质量严重下降。

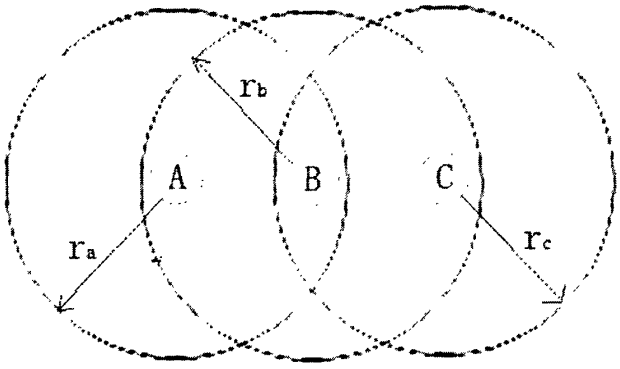


图 1-3　隐藏站/暴露站问题

除了隐藏站问题还有暴露站问题[17]。同样如图 1-3 所示，当 B 要发送数据给 C 时，B 会侦听信道，这时如果 A 也发送数据给另一个站（图上忽略），由于 A 可以覆盖到 B，所以 B 侦听到 A 的信号之后会以为信道忙，于是放弃发送数据到 C，但实际上 B 发送数据到 C 是没有问题的，因为 B 和 C 之间的信道是空闲的，不会受到 A 信号的干扰。

另一方面，大多数的无线电信道是半双工的，也就是说工作站不可能在同一个频率上发送数据的同时侦听到噪声反馈信号。所以 802.11 并没有采用以太网 MAC 层的 CSMA/CD 机制[16][17]。

为了解决上述问题，802.11 支持两种模式的操作。首先是 DCF（Distributed Coordination Function）模式，并没有使用任何中央控制机制（在某种程度上类似以太网）；另一种是 PCF（Point Coordination Function），使用一个基站来控制小区内的所有工作站的通信活动。所有 802.11 协议的实现方案都会支持 DCF，但 PCF 则是可选的。而其中 DCF 中解决该问题的方案，就是下文所提到的虚拟侦听的方法。

1.4.4 虚拟侦听和 RTS/CTS 会话

当 DCF 启用时，802.11 使用一种协议称作 CSMA/CA（载波侦听多址接入/冲突避免）。在这种协议下，物理信道侦听和虚拟信道侦听协同工作。物理侦听方法，是指当一个工作站想发送数据时，它会侦听信道，如果信道空闲，则开始发送，并且在发送过程中不会再侦听信道，直到将整个帧都发送完，当然这时该帧就很容易受干扰而在接收机处无法纠错而被丢弃。如果侦听到信道忙，发送机

会等待一段时间直到信道再次空闲，然后才开始发送。如果发生冲突，冲突的工作站会自动退避一个随机时间，使用以太网的二进制指数退避算法，退避结束后才重新开始侦听过程[16][17]。

虚拟侦听的方法如图 1-4 所示，在这个例子中，A 想发送数据给 B，A 的信号也可以覆盖到 C。D 则只是被 B 覆盖到而并没有被 A 覆盖。当 A 开始发送数据给 B 时，A 会先发送一个 RTS 帧给 B 请求允许发送一个数据帧，当 B 收到 RTS 帧之后，会作出判定是否接收数据帧，如果决定接收，则 B 会回复一个 CTS 帧给 A。接收到 CTS 帧之后 A 就开始发送数据帧，并且启动 ACK 回复计时，等待 B 的 ACK 回复。如果 B 正确收到了 A 的数据帧，就会回复 ACK 帧，结束这次帧交换。如果 A 的 ACK 计时器超时了仍然没有收到 B 的 ACK 回复，整套帧交换的协议就会重新开始。

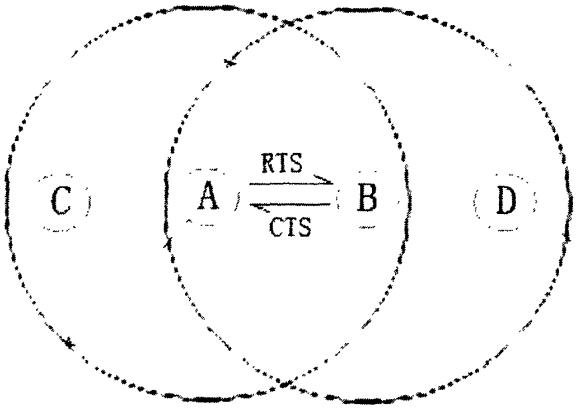


图 1-4　虚拟载波侦听的 RTS/CTS 握手过程

换一个角度从 C 和 D 来分析，由于 C 在 A 的覆盖范围内，所以 C 会收到 A 的 RTS 帧，然后 C 就知道有工作站即将要发送数据帧了，为了避免自己发送的帧遇到冲突，C 会采取干脆停止发送并且等待一段时间，确定 A 的帧交换过程结束后才开始自己的发送。而这段时间就用一个称为 NAV（Network Allocation Vector）的值来保存，因为从 RTS 帧中可以收到关于这次帧交换的一些信息，包括根据数据帧长计算出的需要占用信道的时间，C 就可以读出这个时间值并保存在 NAV 中，这个时间包括 B 回复 CTS 帧的时间，A 发送数据帧的时间，B 再回复 ACK 帧的时间，以及所有的帧间间隔时间。同样地，D 虽然不在 A 的覆盖范围内，但在 B 的覆盖范围内，可以收到 B 回复的 CTS 帧（假如 B 同意与 A 进行

帧交换)。通过 CTS 帧里面读取的数据，D 也可以计算出自己的 NAV 值，即包括 A 发送数据帧的时间，B 回复 ACK 的时间，以及所有的帧间间隔时间。对于 C 和 D 来说，都设置好了 NAV 值之后，在 NAV 所表示的这段时间之内，C 和 D 都不再试图发送任何数据，甚至不去侦听信道，直到 NAV 的计时结束后才重新开始侦听信道的过程。这个过程就称为虚拟侦听，因为 C 和 D 并没有真正去侦听物理信道，而只是通过计算出 NAV 值之后就认为信道在这段时间之内都是忙的，不再去侦听，以免浪费功率，同样也可以起到避免冲突的作用。用下面的时序图进行表述的话，A 就相当于 Source 站点，B 就相当于 Destination 站点，而 C 和 D 就相当于 Other 站点。当 A 和 B 进行了 RTS/CTS 握手之后，C 和 D 都以收到的 RTS 帧和 CTS 帧上标注的 NAV 时延进行虚拟侦听，直到 NAV 倒计时结束才完成侦听过程，再次尝试争用信道[17][19]。该过程的时序图如图 1-5 所示[16]。

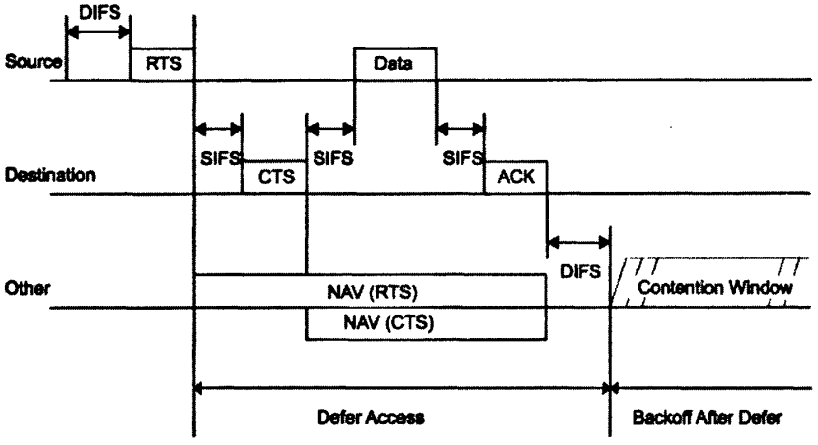


图 1-5　虚拟侦听时序

1.4.5 802.11 的多播模式

在 802.11 的标准中，多播帧与广播帧是以同样的形式进行传输的。当 AP 发送多播帧时，只能采用基本的接入模式。无论帧的长度多少，都不启用 RTS/CTS 握手过程，而且没有任何 ACK 确认帧会在多播数据帧发送之后进行回复。同时多播也没有任何 MAC 层的错误恢复机制，不会产生任何帧重传过程。但任何从 Station 发送出来的多播帧都必须设置 ToDS 域为标记状态，而且要遵守 RTS/CTS 握手过程的规则，而且将会有 ACK 回复的确认过程，同时也有相应的重传恢复

过程。这跟单播模式是类似的。之后AP将会为该Station重新以多播形式再次发送该帧，所以从链路层看起来整个过程就相当于从Station单播一次给AP，然后AP再为其多播到整个网络覆盖范围内的所有Station[16][17][18]。所以无论多播帧是由AP发出，还是由Station发出，其最后的一段由AP进行多播的过程是类似的，本文也是针对AP分发多播帧到Station这一段过程进行讨论，至于由Station发送多播帧的过程，将不再另行讨论。

第2章 ns-3 网络仿真系统

ns-3 网络仿真系统是一个基于离散事件模型的网络仿真系统，专门用于研究和教学用途。该项目从 2006 年开始运作，至今已发布多个版本的 ns-3 系统，是一个开源的国际项目，参与开发者遍布全球各地，有高校研究人员，也有企业的工程师。

ns-3 自从 2007 年 3 月发布第一版 ns-3.0.1 之后，陆续发布更新版本，截止至本文撰写的时候为止，最新的版本是 2010 年 3 月 17 日发布的 ns-3.7.1[20]。而由于各版本之间系统模块和一些实现接口随着版本的更新都会有所改变，所以本文所涉及的大部分工作都是基于较早期的 ns-3.4 版本，而在该版本中 ns-3 已经加入了对无线局域网链路层多播的支持，修复了更早版本 ns-3.2.1 的在开启多播模式进行发送数据帧出现系统崩溃的 bug，为本文的所涉及的研究工作提供了一个较为可靠的实验平台。

2.1 离散事件模型

离散事件模型是大部分的仿真系统的基础，包括 ns-3 网络仿真系统。该模型将连续的时间维度上的不断发生的事件，只关注其中的对仿真过程有用的关键事件，将其抽取出来并作出简化，只记录该事件的开始时间和结束时间，并映射到一个事件队列中，从而实现将连续的事件发生序列离散化，变成一个个时间点上发生的单个事件，而在这些事件与事件之间所发生的一切，该模型并不关心，直接将其忽略。

如图 2-1 所示，例如我们只关注一个大学生一天的生活中几大主要发生的事件，如起床、上课、吃午饭和打球运动等，那么可以简化成一个事件队列，上面记录了每一个事件所发生的时刻。那么只要将该队列上的事件按照时间的先后顺序逐一重演一遍，就可以把该大学生的一天的生活的主要经历再现出来。而基于离散事件模型的仿真系统原理也是类似的。

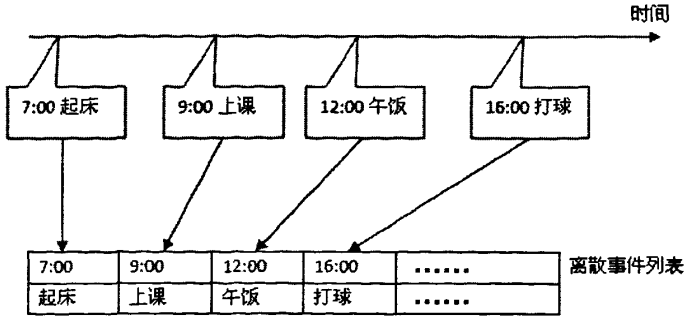


图 2-1　离散事件模型

2.2 ns-3 的无线 Wi-Fi 网络设备模型

在 ns-3 系统的节点模型中，可以包含一个或多个网络设备对象，类似真实计算机中的不同的网络接口卡，如以太网网络接口卡、无线 Wi-Fi 网络接口卡、蓝牙网络接口卡等。以下将重点介绍 ns-3 的无线 Wi-Fi 网络设备模型。

ns-3 支持有基础设施的无线网络拓扑（infrastructure-based）或无线自组网的网络拓扑（ad-hoc based）。本文的研究工作主要基于有基础设施的无线网络拓扑，所有网络仿真实验都是基于有线骨干网连接的单 AP 转发的无线网络环境。

2.2.1 无线网络设备模型 WifiNetDevice

ns-3 的无线 Wi-Fi 网络设备模型称为 WifiNetDevice[21][22]，该模型基于 IEEE 802.11 标准建立一个无线网络接口控制器模型。以下是该模型的几大特点：

- 基本的 802.11 DCF 实现，包括有基础设施的和无线自组网两种模式。
- 802.11a 和 802.11b 物理层实现。
- QoS-based EDCA 和 802.11e 的队列扩展特性。
- 多种信道传播损耗模型，包括 Nakagami、Rayleigh、Friis、LogDistance、FixedRss 模型和随机衰减模型等。
- 两个信道传播延迟模型，包括基于距离的模型和随机模型。
- 多速率控制算法，包括 Aarf、Arf、Cara、Onoe、Rraa、ConstantRate 和 Minstrel 模型等。
- 支持 802.11s（Mesh 网络）

ns-3 提供的 802.11 模型集合尝试尽可能地接近真实的符合 802.11 标准规范

的 MAC 层实现，并且提供一个不太慢（not-so-slow）的 802.11 物理层模型。

整个 WifiNetDevice 模型提供四大层次的建模：

- 物理层模型
- MAC 低层模型，实现 DCF 和 EDCAF
- MAC 高层模型，实现 MAC 层的信标帧机制、探测和关联状态机模型
- 速率控制算法，由 MAC 低层模型使用

其中 MAC 高层模型有 6 种不同的实现，分别有 3 种非 QoS 的和 3 种基于 QoS 的模型：

- AdhocWifiMac：

提供一个简单的 adhoc 状态机模型，没有任何的信标帧机制、探测机制或关联机制。

- NqstaWifiMac：

提供一个主动探测和关联状态机，在过多的信标帧丢失之后自动处理重关联的过程。通常用于实现无线局域网终端 Station 的建模。

- NqapWifiMac：

提供一个接入点模型，周期性产生信标帧，并且接受所有尝试的关联请求。通常用于无线局域网无线接入点 AP 的建模。

基于 QoS 的模型就是在非 QoS 的模型上增加了对 QoS 流量的管理功能，同样是对应的三种模型，用于三种不同的场合的建模。

MAC 低层模型又分为三个模块：

- MacLow：

负责 RTS/CTS/DATA/ACK 帧的发送和接收

- DcfManager 和 DcfState：

实现和管理 DCF 和 EDCAF 功能，保证模型严格按照 DCF 协议标准运行。

- DcaTxop 和 EdcaTxopN：

负责处理帧的排队、帧的分段以及帧的重传等过程。其中前者是用于非 QoS 的模型，而后者是用于带 QoS 的模型。

2.2.2 无线物理层模型 WifiPhy

WifiPhy 模型[21][22]是 WifiNetDevice 里面负责向信道发送比特流和从信道上

接收比特流的一个模块。缺省的ns-3的WifiPhy模型模拟的是802.11a的物理层标准，包括频率、调制方式和比特率等。由于在ns-3.4版本中还没引入对802.11b的支持，所以本文所涉及的所有研究实验都基于802.11a的物理层模型进行。

WifiPhy物理层模型也是一个状态机模型，其状态包括以下三种：

- TX：表示物理层正在传送一个物理信号，该信号来自于MAC层传递下来的帧。
- RX：表示物理层正在同步一个信号，并且等待其最后一个字节完成接收并传递给上层的MAC层。
- IDLE：表示物理层当前正处于空闲状态，既不在TX状态也不在RX状态。

当物理层接收到一个新的帧的第一个字节时，如果发现自身并不处于IDLE状态（比如说正处于前一帧的接收同步状态或处于发送数据帧的状态），则该帧会被直接丢弃掉。如果发现正处于IDLE状态，就会计算所接收帧的第一个比特的功率，并与功率检测门限进行对比。如果帧比特的功率更高，则物理层转移到RX状态并安排一个接收完成事件，事件发生时间设置为接收完该帧的最后一个比特的时间。如果帧比特的功率不高于门限值，则仍然是直接丢弃帧，不作接收处理。

2.2.3 无线信道模型 WifiChannel

WifiChannel模型[21][22]是模拟无线信道环境的一个模型，其建模包括两个子模型类别，一个是传播时延模型（PropagationDelayModel），另一个是传播损耗模型（PropagationLossModel）。

PropagationDelayModel又包括以下两种可选的建模方式：

- 固定速率传播延迟模型（ConstantSpeedPropagationDelayModel），缺省采用光速进行延迟值的计算。用户可以修改该速率值。
- 随机传播延迟模型（RandomPropagationDelayModel），信道延迟值采用随机方式计算。

ns-3默认是采用固定速率传播延迟模型，而本文的研究工作也是采用该默认值进行。

PropagationLossModel也包括以下几种可选的建模方式：

- 随机传播损耗模型（RandomPropagationLossModel）
- Friis 传播损耗模型（FriisPropagationLossModel）
- 对数距离传播损耗模型（LogDistancePropagationLossModel）
- Jakes 传播损耗模型（JakesPropagationLossModel）
- 混合传播损耗模型（CompositePropagationLossModel）

其中 ns-3 默认是采用对数距离传播损耗模型，而本文也是利用该模型进行信道的恶化模拟。以下将对该模型的细节进行展开陈述。

2.2.4 对数距离传播损耗模型

该模型[21][22]采用公式(2-1)进行接收功率的计算：

$$L = L_0 + 10n\log_{10}(\frac{d}{d_0}) \qquad \text{公式 (2-1)}$$

其中：

- n: 路径损耗距离指数，缺省为 3
- d0: 参考距离(单位：m)，缺省为 1m
- L0: 参考距离内的路径损耗(单位：dB)，缺省为 46.6777dB
- d: 实际距离值 (单位：m)
- L: 实际路径损耗 (单位：dB)

当实际距离小于参考距离时，路径损耗的计算将会直接返回发送功率值 TxPower。按照 ns-3 的缺省参数设定，绘制该函数图像如图 2-2 所示。

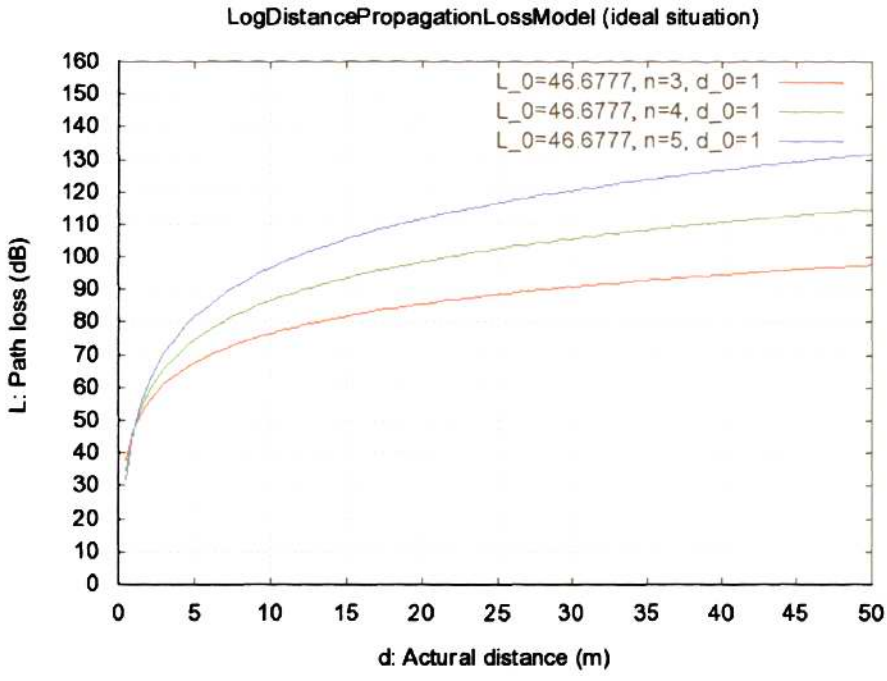


图 2-2　对数距离传播损耗模型函数图像

从图上可以看出，如果在理想情况下，在 n 为缺省值 3 时，在 25 米范围内，路径损耗会随着距离的增大而急剧上升，比特流的接收功率会因此而快速衰减，但到了 25 米以外的区域，损耗速度趋于平稳，此时接收功率已经下降到一个比较低的水平，容易出现比特误码，只有发射功率较大的发射机才能使其信号仍旧被识别出来。观察另外两条曲线，随着 n 值的增大，该曲线的上扬程度会提高，达到同样的路径损耗值的距离会逐渐缩短，例如当 n=3 在 25 米处的路径损耗值，在 n=4 时只需到 12 米左右就已经达到，而在 n=5 时，在 7 米左右就已经达到，可见此时衰减程度是非常高的。这在后文的实验场景部署中会提到。

2.3 ns-3 源代码修改

整个 ns-3 的 Wifi 模型如图 2-3[21]所示：

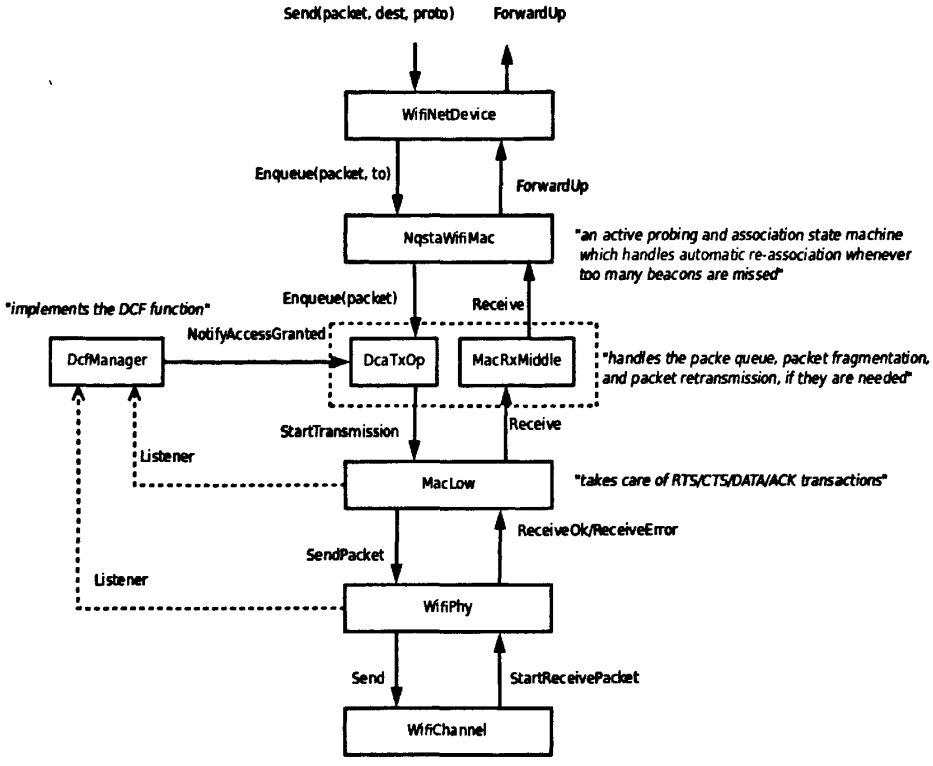


图 2-3　ns-3 无线设备模型体系结构图

可以从图上看出，一个数据包从上层协议传递下来之后，首先交给 WifiNetDevice 模块进行发送前的数据包类型识别等过程，然后交给 NqstaWifiMac，进行 LLC 帧头的插入，以及构造 MAC 帧头。然后就交给缓冲队列模块（WifiMacQueue，图上未标出），即将数据帧入队列，等待发送。发送的时机由 DcfManager 模块安排，当信道空闲可以安排发送帧之后，DcfManager 会通知 DcaTxOp 将缓冲队列中队首的帧取出，并检测是否需要发送 RTS 帧，是否需要分段等。做好一切发送前的准备工作之后，就送到 MacLow 模块进行发送。MacLow 模块会甄别不同帧的类别，按照不同的方式进行发送，将帧传递到物理层 WifiPhy 模块。WifiPhy 模块将帧视为比特流，送到目的终端的 WifiPhy 模块，通知其接收帧，并通过 WifiChannel 模块计算路径损耗和传播时延，模拟该帧在信道上传输的过程。

当帧到达目的站点的 WifiPhy 模块之后，WifiPhy 检测是否误码或者无法识别信号，从而判断是否需要丢弃该帧。当检测完毕后 WifiPhy 会将该帧上交给 MacLow 模块，MacLow 按照不同帧的类别进行不同的接收处理。并且去掉 MAC

帧头，上交给上层模块继续处理。最后通过WifiNetDevice上交给上层协议。整个链路层和物理层的帧处理过程就是如此。

而本文所设计的MLBP协议在实现时，增加了一个LBP协议管理模块LbpManager，通过该模块记录并统一管理每一个多播组的源组对信息以及多播组成员的信息，同时提供选定Leader、查询Leader等回调函数功能。另外还增加了一个计算和刷新重传概率的管理模块RetryManager，通过该模块动态地对丢帧情况进行计数和刷新当前丢帧率，并在一定采样频率后更新重传概率值。

在MacLow模块处理过程中，插入了与LbpManager沟通的步骤。每一个Station通过向LbpManager查询Leader的功能，来判断自己当前是否已成为Leader，进而判断是否需要进行确认帧的回复。同时每接收一个确认帧都调用RetryManager模块的丢帧率更新功能，刷新当前的重传概率值。

在DcaTxop模块的处理过程中，加入了与RetryManager模块的沟通步骤。每一次需要进行重传之前，都会调用RetryManager的重传概率判断功能，通过随机计算之后以一定概率进行重传。

另外MLBP协议在实现时还对ns-3的wifi模块中的缓冲队列模块WifiMacQueue进行了修改（在该流程图上未能体现），增加了多播资源预约表的管理模块ReservManager，并提供带有多播资源预留帧的识别功能的回调函数。WifiMacQueue模块通过调用该回调函数可以判断当前因拥塞不能进入缓冲队列的数据帧是否是资源预留帧，若是则进行优先调度，否则则按正常处理流程进行丢弃。

第3章 无线可靠多播协议

1999年Kuri, J.等[1]提出了LBP协议之后，该协议一直成为无线可靠多播领域的研究热点。但同时另外两种经典的无线可靠多播协议一样没有被人们遗忘，例如后来提出的RLBP协议[2]，实际上是部分借鉴了DBP协议的思想。以下将详细介绍这三种经典的无线可靠多播协议LBP、DBP和PBP。

3.1 Leader-Based Protocol（LBP）

该协议首先由Kuri, J.等[1]提出，是早期比较经典的无线局域网可靠多播协议的一种。该协议主要是为了解决可靠多播中确认帧容易发生冲突的问题。协议假设无线局域网内的其中一个Station已经被预先选定为Leader，之后所有的确认帧包括CTS帧和ACK帧都由该Leader STA单播回复给AP。AP收到Leader STA回复的确认帧，视为其他Station也正确接收之前的多播帧，继续下一帧的发送过程。所以Leader相当于代表了无线局域网内的所有Station。由于回复的确认帧的Station只有Leader一个，所以从根本上解决了确认帧冲突的问题。

该协议的执行流程如下所示：

[1] 基站 → 接收终端（Slot 1）

发送多播RTS帧，请求获取信道使用权。

[2] 接收终端 → 基站（Slot 2）

- Leader终端：如果准备好接收数据，发送CTS帧；否则，保持沉默。
- 非Leader终端：如果准备好接收数据，就保持沉默；否则，发送NCTS（Not Clear to Send）帧。

[3] 基站 → 接收终端（Slot 3）

- 如果在Slot 2接收到CTS帧，就开始传输多播Data帧。
- 如果在Slot 2没接收到任何CTS帧，或者接收到NCTS帧，则启动退避过程，回到步骤1重新开始。

[4] 接收终端 → 基站（Slot L+3，假设Data帧长为L）

- Leader终端：如果Data帧正确接收完毕，发送ACK帧；否则，发送NAK（Negative ACK）帧。

- 非 Leader 终端：如果 Data 帧正确接收完毕，保持沉默；否则，发送 NAK 帧。

[5] 基站

- 如果在 Slot L+3 之后收到 ACK 帧，则结束该次传输过程，等待或安排下一个 Data 帧的发送过程。
- 如果在 Slot L+3 之后没有收到 ACK 帧，或者收到 NAK 帧，则判断当前重传次数是否已达到重传上限，达到上限则结束该次传输过程，丢弃该 Data 帧，并通知上层协议发送失败。若未达到上限，则将已发送的 Data 帧打上重传标记，进入退避过程，等待下一次发送机会，重新竞争信道开始以上全过程。

在信道质量良好，所有 Station 都能正常收到 AP 多播的 m-RTS 帧和 m-DATA 帧时，协议的时序图如下图 3-1 所示。其他 Station 一直处于静默状态。当信道质量恶化，其他 Station 出现不能正常接收 m-RTS 帧或 m-DATA 帧时，就会出现如图 3-2 和图 3-3 所示的情况，其他 Station 发送 NCTS 帧或 NAK 帧，与 Leader 发送到 CTS 帧或 ACK 帧产生冲突，使得 AP 不能正常接收到 Leader 回复的 CTS 帧或 ACK 帧，从而中断后续的发送过程，或者安排重传恢复过程。当其他站点如果在接收到 AP 发来的 m-RTS 帧后，仍未准备好接收 m-DATA 帧，也会出现图 3-2 的情况，发送 NCTS 帧通知 AP 使其不能继续 m-DATA 帧的发送。此时 AP 也只能等待下一次的发送机会重新安排该 DATA 帧的发送。当然，以上所说的 NCTS 帧和 NAK 帧同样也可以由 Leader 发出，如果遇到同样的情况的话。

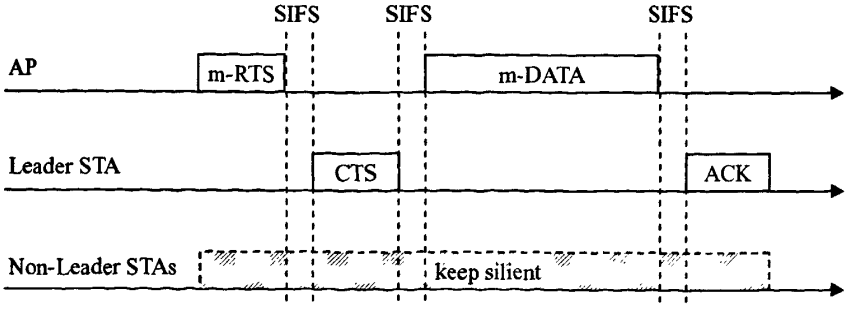


图 3-1　LBP 协议时序图 1

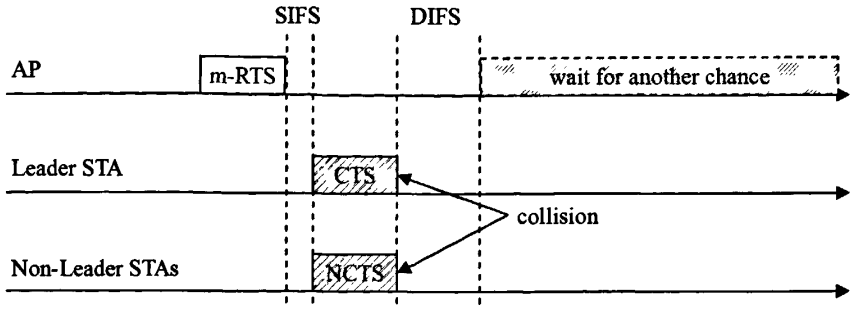


图 3-2 LBP 协议时序图 2

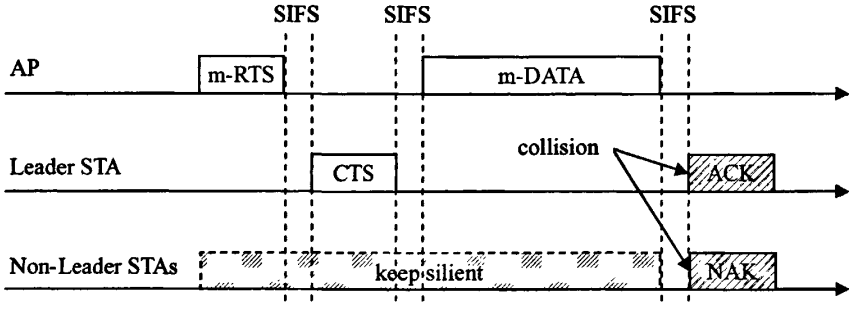


图 3-3 LBP 协议时序图 3

3.2 Probability-Based Protocol（PBP）

该协议假设已经预定好一个概率 P，所有 Station 以概率 P 回复 CTS 帧。每当 AP 发送 m-RTS 帧之后，每一个 Station 立即计算一个随机数 rand_num，若 rand_num 小于 P 时，则向 AP 发送 CTS 帧，若 rand_num 大于 P 时，则不发送 CTS 帧，保持静默状态。当 AP 完成 m-DATA 帧的发送之后，不必等待 ACK 帧的回复，继续进入下一次的发送过程，所以在 PBP 里面是没有 ACK 帧的。该协议只有在所有 Station 中只有一个 Station 计算概率后发送 CTS 帧，其他 Station 计算概率后保持静默状态时，才能顺利运行，如图 3-4 所示。否则，当出现两个以上的 Station 计算概率后选择发送 CTS 帧，则依然会出现帧冲突现象，AP 无法继续发送 m-DATA 帧。这在概率 P 较大的时候尤其明显。但当 P 太小的话，出现所有 Station 计算概率后都选择保持静默，则 AP 也是无法继续发送 m-DATA 帧，如图 3-5 所示。所以一般该协议比较少被使用。

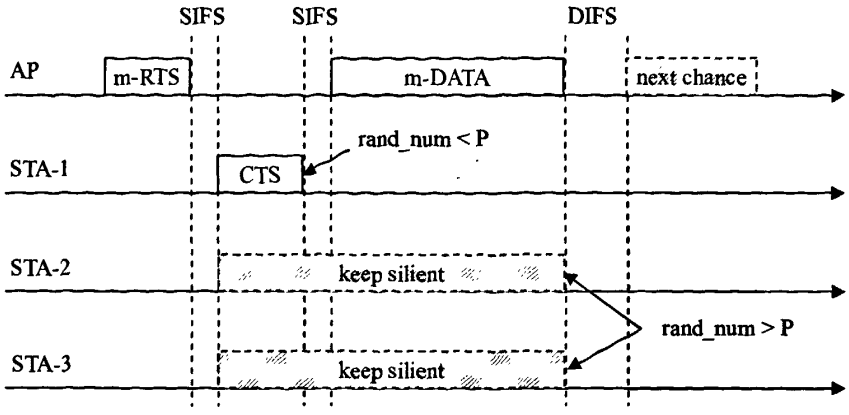


图 3-4　PBP 协议时序图 1

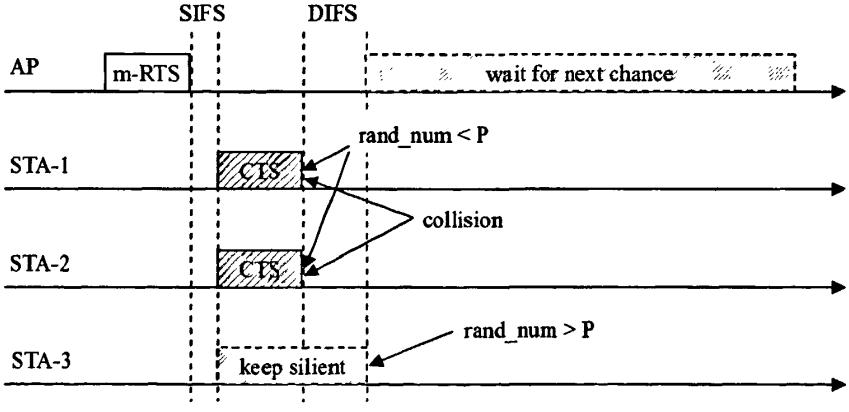


图 3-5　PBP 协议时序图 2

3.3 Delay-Based Protocol（DBP）

该协议采用随机延迟的策略安排 Station 发送 CTS 帧，每当 AP 发送 m-RTS 帧之后，所有 Station 都计算一个随机的 slot 时间，计算公式如下所示：

$$\text{BackoffTime} = \text{Random} \times \text{aSlotTime} \quad \text{公式 (3-1)}$$

并且按照这个时间进行退避，首先完成退避延迟的 Station 就发送 CTS 帧，其他 Station 侦听到该 CTS 帧之后停止退避，进入 CTS 抑制状态，保持静默。AP 在收到 CTS 帧之后就安排发送 m-DATA 帧，完成后同样不必等待 ACK 帧的

回复，直接进入下一次的发送过程。由于退避时间是随机计算的，所以每一次进行 CTS 帧发送的 Station 都可能会不一样。由于发送确认帧的同样只有一个 Station，所以该协议也是很好地解决了确认帧冲突的问题，但由于增加了一个随机退避时间，协议的时延可能会因此而增加，而且缺少 ACK 帧的回复过程，可靠性也会有一定的下降。该协议的时序如图 3-6 所示。

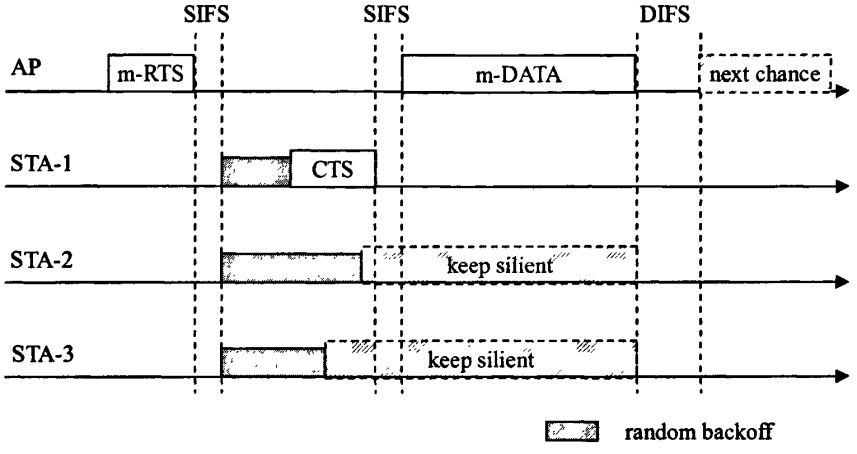


图 3-6　DBP 协议时序图

第4章 MLBP协议设计

本协议是基于标准 LBP 协议改进的，协议主要针对单发送者，即基站，及其所覆盖的无线信号覆盖范围内的单个无线局域网。基站上承担着一个无线多播组的流量转发。本文假设该无线基站支持链路层的多播机制，包括链路层的多播地址，都被基站和接收终端所支持。

协议主要包括两大部分，第一部分是基于概率的信道错误重传恢复，第二部分是基于优先级的 QoS 缓冲队列调度算法，第三部分则是多媒体多播流资源预约机制。以下分别展开陈述。

4.1 基于概率的信道错误重传恢复

对于无线信道来说，因许多因素干扰，如其他无线信号源的电磁干扰，空间中障碍物的阻挡，多径效应等等，都会造成信道的误码率增大，从而反映在链路层上则是丢帧率升高。当丢帧发生之后，发送端便会发生确认帧超时，于是就会安排对丢失帧的重传恢复。但当信道质量恶化到一定程度时，这种重传的次数就会增多，从而造成真正能顺利到达接收端的数据帧的帧间隔变大，而重传次数的不确定也使得这种帧间隔的变化会比较剧烈，最终导致业务流的抖动增大。这对多媒体流来说是非常不利的，所以 MLBP 协议设计基于概率的信道错误重传恢复，就是通过增加概率选择这一步骤，通过牺牲可靠性，缩小重传的平均次数，从而保证多媒体流能够以较低的帧接收间隔和较稳定的抖动值进行传送，保证终端用户体验的流畅性和稳定性。

协议的具体实现步骤如下：

[1] 基站 → 接收终端（Slot 1）

发送多播 RTS 帧，请求获取信道使用权。

[2] 接收终端 → 基站（Slot 2）

Leader 终端：如果准备好接收数据，发送 CTS 帧；否则，就保持沉默。

非 Leader 终端：如果准备好接收数据，就保持沉默；否则，发送 NCTS（Not Clear to Send）帧。

[3] 基站 → 接收终端（Slot 3）

如果在 Slot 2 接收到 CTS 帧，就开始传输多播 Data 帧。

如果在 Slot 2 没接收到任何 CTS 帧，或者接收到 NCTS 帧，则启动退避过程，回到步骤 1 重新开始。

[4] 接收终端 → 基站（Slot L+3，假设 Data 帧长为 L）

Leader 终端：如果 Data 帧正确接收完毕，发送 ACK 帧；否则，发送 NAK（Negative ACK）帧。

非 Leader 终端：如果 Data 帧正确接收完毕，保持沉默；否则，发送 NAK 帧。

[5] 基站

如果在 Slot L+3 之后收到 ACK 帧，则结束该次传输过程，并将成功传输计数器 STC（Successful Transmission Counter）加一。然后等待或安排下一个 Data 帧的发送过程。

如果在 Slot L+3 之后没有收到 ACK 帧，或者收到 NAK 帧，将成功传输计数器 FTC（Failure Transmission Counter）加一。然后计算一个[0,1]之间的随机数，并与重传概率 γ 比较，如果大于 γ，则放弃重传过程，结束该次传输过程，丢弃该 Data 帧，并通知上层协议发送失败。如果小于 γ，则将已发送的 Data 帧打上重传标记，进入退避过程，等待下一次发送机会，重新竞争信道开始以上全过程。

[6] 基站

重新计算当前丢帧率，将 STC 加上 FTC 的和再与丢帧采样频率比较（缺省为 100），如果相等，则按照以下公式重新计算丢帧率 p：

$$p=\frac{FTC}{STC+FTC} \qquad \text{公式 (4-1)}$$

并重新计算重传概率 γ：

$$\gamma=\begin{cases}1 & (0\le p\le p_0)\\ \frac{p_0}{p} & (p_0<p\le 1)\end{cases} \qquad \text{公式 (4-2)}$$

然后清零 STC 和 FTC 重新计数。

协议的前 5 步骤在 AP 上的实现流程，可以用图 4-1 的流程图表示，而第 6 步则用图 4-2 的流程图表示。

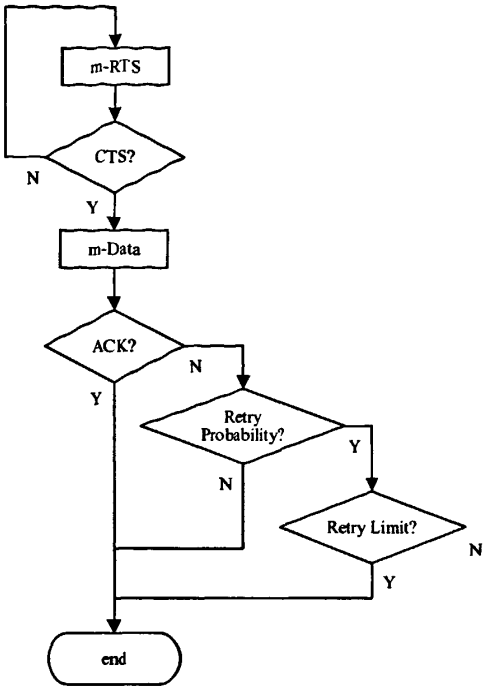


图 4-1　MLBP 协议流程图 1

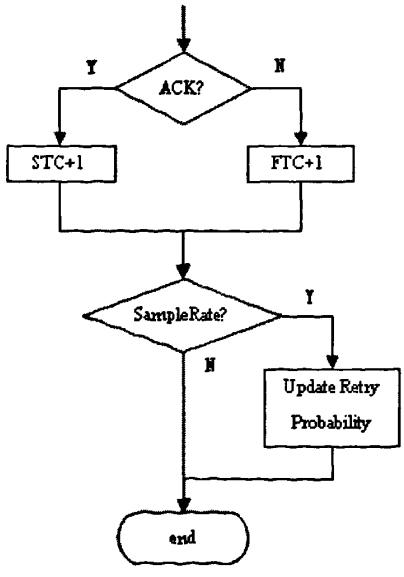


图 4-2　MLBP 协议流程图 2

4.2 基于优先级的QoS缓冲队列调度算法

通常一个AP就担任了其所覆盖范围的无线局域网的网关角色，承载在该局域网内所有无线终端的流量，包括上行和下行。随着局域网内无线终端的数量增多，AP上的流量负担越来越重，最终将因为AP的处理转发能力的限制而使其成为整个无线局域网的瓶颈。

对于上行流量而言，当AP收到这些数据帧之后就立即通过路由协议找到转发的地址，然后通过其上级端口所连接的骨干网络转发出去，而这些骨干网络通常是有线网，其带宽一般都能应付AP转发出来的流量，所以极少在AP的上游端口造成拥塞。

而对于下行流量而言，情况就不一样了。当从骨干网发送来的数据帧到达AP之后，由于无线局域网的信道竞争机制和无线信道的有限带宽，导致AP通常不能马上将数据帧分发到对应的无线终端上。此时，所有到达AP的下行数据帧都会先进入到链路层的缓冲队列进行排队，AP按先进先出（FIFO）的方式逐一为其服务。当下行数据流量比较小时，这种方式是不成问题的，但当下行数据流量增大到一定程度，使得AP不能及时处理掉缓冲队列的数据帧，就会出现缓冲队列满的现象，使得后续到达的数据帧不能进入缓冲队列，而此时AP通常的做法就是直接将其丢弃，只有在缓冲队列中的数据帧被处理掉一些，缓冲队列重新有空余容量的时候，后续到来的数据帧才能进入队列等待发送。

在这样的机制下，就意味着数据帧只要能够进入缓冲队列，就一定能被发送出去，而如果不能进入缓冲队列，就相当于发生了丢帧，此时该数据帧不会再有机会从AP发送出去，被直接丢弃，只有等待源端的上层协议发生超时后进行重传，才有可能得以恢复。而如果源端发送的是类似多媒体流这种通常采用UDP协议封装的业务流，就不会有任何恢复机制，源端也不可能获知数据帧已经在AP处被丢弃，不能到达目的端主机，此时，这种丢帧被视为拥塞丢帧。

在线多媒体的流量的特点通常都是流量连续但速率较低，占用带宽不多，但需要连续不间断的长时间占用。一旦发生断流或流速降低，将马上反映到终端的用户体验上，影响体验质量。而多媒体流的这种特点使得其在跟其他背景流量竞争无线信道时非常吃亏。当背景流量加大，或者产生一些突发的大流量时，瞬间

就可以将带宽全部占用，使得多媒体流被排挤到只有很小带宽的状态，出现流量很小甚至断流的状态。体现在微观层面上，就是链路层的缓冲队列的进入问题。虽然说在宏观上各种业务流都可能是以流的形式存在，但在微观层面上，仍然是以帧的方式进行传送，如此便有了帧间隔的概念。每一帧发送都有一定的间隔，按照发送速率的不同，速率越高，间隔越短，反之亦然。当背景流的流量增大时，自然其速率就提高，帧间隔就缩小，而多媒体流依然是维持原有的速率发送帧间隔不变，如此一来就可能造成每次多媒体帧到达缓冲队列时，缓冲队列都被背景流给占满了，于是多媒体帧只能被丢弃，而接下来的帧也可能会遇到同样的情况，当连续一定数量的多媒体流的帧都找不到一个间隙可以进入缓冲队列，就可能造成连续的拥塞丢帧，在宏观上就呈现断流的现象。此时多媒体流的帧间隔和抖动都会随之而增大，用户体验质量急剧下降。于是，为多媒体业务流预留一定资源，使其流量能够得到保证不会出现断流，就显得非常有必要了。而这也正是 QoS 的思想，即提升多媒体业务流的 QoS。

从另一方面来看，对于本文所研究的需要可靠多播支持的多媒体流，一般都是受众较多，相对其他背景流来说比较重要的业务流，例如视频会议、体育赛事实时直播等，在这种情况下，都是单一数据源分发数据到多个接收终端，而且要求保证一定的 QoS，所以为这一类的多媒体流预留一定的资源，也是非常合理的。

4.2.1 协议细节

本协议将所有经过 AP 的业务流分成两个优先级，分别为“资源预留级（Reservation）”和“自由争用级（Contention）”。当未开启资源预留机制时，所有数据流都视为自由争用级，所有数据帧到达 AP 的链路层缓冲队列时，都按照原有的缓冲队列调度算法进行调度，即当缓冲队列未满时将新帧插入队尾，当缓冲队列已满时就直接丢弃。当用户（通常由管理员操作）为某个流预订了资源预留之后，AP 将会通过源地址和目的地址，以及源端口和目的端口这个四元组对该流进行甄别，将其设置为资源预留级，其他所有流仍旧视为自由争用级。

当开启了资源预留机制之后，当出现有数据帧到达缓冲队列发现队列已满时，分析其上层协议的帧头，甄别是否为资源预留级的多媒体流，若是则从队尾开始扫描缓冲队列，对每一帧做同样的甄别，当发现一个不是该多媒体流的数据帧时，就将其提取出队列并直接丢弃。此时缓冲队列中便腾出一个数据帧的空间，

令新到达的资源预留级的多媒体数据帧可以顺利入队列。

该协议的流程图如图 4-3 所示：

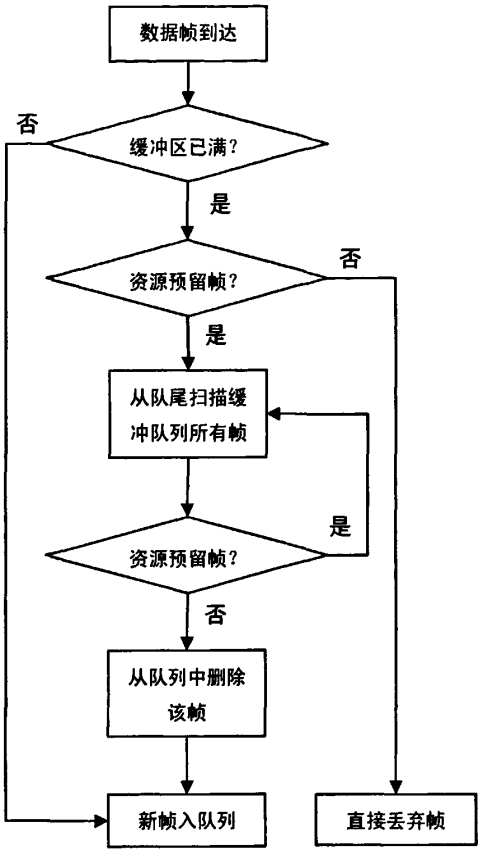


图 4-3　MLBP 协议流程图 3

4.3 多媒体多播流资源预约机制

该机制采用人工的预约模式，将需要为其保留资源的多媒体多播流的源组对信息通过上层应用层界面的操作，记录的到 AP 的资源预约表上（该应用层界面设计的问题已经超出本文的讨论范围，在此不展开详述）。该资源预约表的格式如图 4-4 所示：

Source Address	Destination Address
Server IP Address 1	Multicast Group Address 1
Server IP Address 2	Multicast Group Address 2
……	……
Server IP Address n	Multicast Group Address n

图 4-4　多播流资源预约表

每当 AP 上所承载的流量增大，链路层的缓冲队列出现队列满，造成拥塞丢帧时，AP 启动基于优先级的 QoS 缓冲队列调度算法，每次发现有新的数据帧到达链路层的但无法进入缓冲队列时，就分析其 IP 帧头（如果是 IP 帧），读取出源地址和目的地址字段值，如果源地址匹配资源预约表上某一源组对的源地址，而且目的地址是多播地址的同时也匹配资源预约表上该源组对的多播组地址，则 AP 对其实施优先调度，按照上文介绍的调度算法，该资源预留帧可以顺利进入队列。由于 IP 帧头的长度在头 20 字节的格式是固定的，所以在此分析帧头过程中并不需要将 IP 帧头移除，只需先辨别出是否 IP 协议帧，再进行固定偏移量和固定长度的数值读取，就可以获得相应的字段数值，不会因为要移除帧头而大幅增加 AP 的处理能力开销。同样的道理，本协议设计时不采用分析到传播层协议帧头的方法来获得业务流的源端口和目的端口，因为这样就需要移除 IP 帧头，增加 AP 的处理能力开销，使得本已负荷满盈的 AP 更加雪上加霜。所以本协议目前只支持针对主机到主机的资源预约，暂时不提供针对端到端的业务流区分的资源预约。

第5章 协议性能分析

以下分别针对基于概率的信道错误重传恢复和基于优先级的 QoS 缓冲队列调度算法进行数学建模与性能分析。

5.1 基于概率的信道错误重传恢复

因为 LBP 通过 Leader 节点和 AP 之间的帧交换过程代表多播组所有节点的接收情况，不妨假设当前 AP 与 Leader 节点之间的信道丢帧率为 p，当前设置的重传概率为 γ。如果每一次帧突发（packet burst）所需的时间 $T_{burst} = T_{DIFS} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}$，而且每一次重传都要经过这么一个时序过程，同样要产生这么多时间开销，所以每一次成功传送一个数据帧（包括重传）的时间 τ 的可能值为：T_{burst}，$2T_{burst}$，$3T_{burst}$ … $7T_{burst}$（标准 802.11 的 MAC 设置最大重传次数为 6）。各个可能值的概率分布如下所示：

- 标准 LBP 协议：

$P^S_{\tau = Tburst} = 1 - p$

$P^S_{\tau = 2Tburst} = p(1 - p)$

$P^S_{\tau = 3Tburst} = p^2(1 - p)$

$P^S_{\tau = 4Tburst} = p^3(1 - p)$

.....

$P^S_{\tau = (n+1)Tburst} = p^n(1 - p)$　（第 n 次重传）　　公式 (5-1)

不难算出，其期望均值$E(P^s_\tau)$为

$$E(P^s_\tau) = \left(\frac{1 - (n+2)p^{n+1} + (n+1)p^{n+2}}{1-p}\right) T_{burst} \qquad \text{公式 (5-2)}$$

$$\lim_{n \to \infty} E(P^s_\tau) = \left(\frac{1}{1-p}\right) T_{burst} \qquad \text{公式 (5-3)}$$

- MLBP 协议(每一个概率包括成功发送和重传抑制两种情况)：

$P^M_{\tau = Tburst} = (1 - p) + p(1 - \gamma) = 1 - p\gamma$

$P^M_{\tau = 2Tburst} = p\gamma(1 - p) + p^2\gamma(1 - \gamma) = p\gamma(1 - p\gamma)$

$P^{M}_{\tau=3Tburst} = p^2\gamma^2(1-p) + p^3\gamma^2(1-\gamma) = p^2\gamma^2(1-p\gamma)$

$P^{M}_{\tau=4Tburst} = p^3\gamma^3(1-p) + p^4\gamma^3(1-\gamma) = p^3\gamma^3(1-p\gamma)$

.....

$P^{M}_{\tau=(n+1)Tburst} = p^n\gamma^n(1-p) + p^{n+1}\gamma^n(1-\gamma) = p^n\gamma^n(1-p\gamma)$（第 n 次重传） 公式 (5-4)

同样不难算出，其期望均值$E(P_\tau^M)$为

$$E(P_\tau^M) = \left(\frac{1-(n+2)(p\gamma)^{n+1}+(n+1)(p\gamma)^{n+2}}{1-p\gamma}\right)T_{burst} \quad \text{公式 (5-5)}$$

$$\lim_{n\to\infty} E(P_\tau^M) = \left(\frac{1}{1-p\gamma}\right)T_{burst} \quad \text{公式 (5-6)}$$

将两个期望值相减得到：

$$\lim_{n\to\infty} E(P_\tau^s) - \lim_{n\to\infty} E(P_\tau^M) = \left(\frac{1}{1-p}\right)T_{burst} - \left(\frac{1}{1-p\gamma}\right)T_{burst}$$

$$= \frac{p(1-\gamma)}{(1-p)(1-p\gamma)}T_{burst} > 0 \quad \text{公式 (5-7)}$$

所以，从以上结果可以明显看出，只要 γ 处于小于 1 的情况下，对 MLBP 的帧突发时间的概率密度分布计算而得的期望均值，总比标准 LBP 的小，即更加集中在T_{burst}附近，说明从统计分析上来说，MLBP 协议的预期帧突发时间的变化抖动，会比标准 LBP 的变化抖动要小，相当于帧传输过程的抖动得到了抑制。而且，从分析过程来看，在 γ 越小的情况下，其计算出来的期望均值差就越大，即 MLBP 比标准 LBP 的抖动就更加小，但这个 γ 值并不是越小越好的，因为有另外一个因素制约着它的值的范围。那就是帧发送的成功率，也可以看成是协议的可靠度。

帧发送成功率 s，同样可以分成以下两种情况考虑：

- 标准 LBP 协议：

$s = (1-p) + p[(1-p) + p(1-p) + p2(1-p) + \cdots + pn\text{-}1(1-p)]$ (n 为重传次数)

$= 1 - pn+1$

$$= \begin{cases} 1 & (n\to\infty, \text{绝对可靠}) \\ 1-p^7 & (n=6, \text{802.11 标准设置}) \end{cases} \quad \text{公式 (5-8)}$$

- MLBP 协议：

$$s = (1-p) + p\gamma(1-p) + p2\gamma2(1-p) + \cdots + pn\gamma n(1-p)$$

$$= \frac{(1-p)[1-(p\gamma)^{n+1}]}{1-p\gamma}$$

$$= \begin{cases} \frac{1-p}{1-p\gamma} < 1 & (n \to \infty,\ \gamma < 1) \\ \frac{(1-p)[1-(p\gamma)^7]}{1-p\gamma} < 1-p^7 & (n=6,\ 802.11\text{标准设置}) \end{cases}$$ 公式 (5-9)

从上面的推导过程可以看出，在增加了 γ 参数以及 γ 小于 1 的情况下，帧发送的可靠度是有所下降的，而且随着 γ 值的降低，可靠度也随之降低。所以 γ 值不能无限制地变小，因为增加 γ 参数换来的抖动缩小，是以牺牲可靠度为代价的。

从 τ 的概率密度分布公式来看，只要假设原来的丢帧率为 p_0，改变后的丢帧率为 p_1，只要改变 γ，使得 $p_0 = p_1 \times \gamma$ 保持成立，则无论丢帧率 p 怎么变化，τ 的概率密度分布就不会改变，其方差也不会改变，相当于整个帧传送过程的抖动就不会发生改变。这实际上是通过 γ 来修正了丢帧率改变对抖动的影响。所以只要将 p_0 设置为一个比较合理的值，使得其对应的抖动对于当前承载的多媒体流是可以接受的，那么通过牺牲一点可靠性，将换来抖动的抑制，所承载的多媒体流将会更加平滑。

当然，γ 的改变范围也是有限的，只能在[0,1]之间浮动，所以 γ 因 p 而改变的函数如下：

$$\gamma = \begin{cases} 1 & (0 \le p \le p_0) \\ \frac{p_0}{p} & (p_0 < p \le 1) \end{cases}$$ 公式 (5-10)

只要 γ 按照以上规律变化，就可以使得整个帧传送的过程的抖动维持在一个比较稳定的水平。而可靠度就变成：

$$s = \begin{cases} 1-p^7 & (0 \le p \le p_0) \\ \frac{1-(p_0)^7}{1-p_0}(1-p) & (p_0 < p \le 1) \end{cases}$$ 公式 (5-11)

所以，当 p 仍处在比较小的范围内$(0 \le p \le p_0)$的时候，可靠度 s 的变化趋势是和标准 LBP 协议下的情况是一致的，而只有 p 增大到一定范围之后$(p_0 < p \le 1)$，s 才会改变为以一种线性的速度下降，而实际上比原来的 7 次方曲线的下降更加平稳一些，不会出现“雪崩式”的丢帧情况。

5.2 基于优先级的 QoS 缓冲队列调度算法

在 802.11 的 MAC 层中所采用的缓冲队列，一般是采用先进先出（First-in-First-Out，FIFO）的队列调度算法，而在 ns-3 中的实际实现也是采用这种方式。设队列长度为 L（ns-3 里 L 值缺省为 400，即最多可缓冲 400 个 packet）。进入队列的流量包括本文协议所承载的多媒体多播流，还有众多背景流，如 TCP 流等，所以数据帧进入队列的速度和间隔时间的分布是相当复杂的，要综合各种可能出现的流的特性及其符合的统计分布。而这些因素已经超出本文的讨论范围，为了简化以下的讨论过程，在此将简单地以 $T_{multicast}$ 表示多媒体多播帧到达队列尾的平均时间间隔，T_{bg} 表示其他所有背景流的帧到达队列尾的间隔时间。而同样，帧离开队列进入发送过程的时间间隔，也是根据当前无线局域网内的繁忙程度，各 Station 和 AP 之间争用信道的情况来综合决定的，在此也简单地以 $T_{avg\text{-}out}$ 表示帧离开队列的平均时间间隔。如图 5-1 所示。

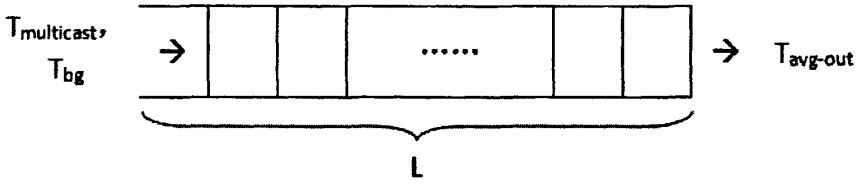


图 5-1　无线链路层缓冲队列

假设当前的无线局域网的繁忙程度很低，$T_{avg\text{-}out}$ 很短，足以保持该缓冲队列处于未满状态，所有新到的帧都能顺利进入队列等待发送，则此时在队列中多媒体多播帧的数量 $N_{multicast}$ 和背景流的帧的数量 N_{bg} 的比值为：

$$\frac{N_{multicast}}{N_{bg}} = \frac{T_{bg}}{T_{multicast}} \qquad \text{公式 (5-12)}$$

假设当前缓冲队列中的总缓冲帧数为 N_{total}，则有：

$$N_{total} = N_{multicast} + N_{bg} \qquad \text{公式 (5-13)}$$

$$N_{multicast} = N_{total} - N_{bg} = N_{total} - \frac{T_{multicast}}{T_{bg}} N_{multicast} \qquad \text{公式 (5-14)}$$

所以有：

$$N_{multicast} = \frac{T_{bg}}{T_{multicast}+T_{bg}} N_{total} \qquad 公式\ (5\text{-}15)$$

若 N_{total} 逐渐增大到等于队列长度 L，即队列处于满状态，而且此时该无线 AP 的处理帧的能力刚好能让每一次有新的帧到达之前，都能从队列中发送出一个帧，从而腾出一个空位给新到的帧入队，即整个队列处于动态的队列满状态，但又不至于造成丢帧，此时的 $N_{multicast}$ 为在队列中能容纳的多播帧的动态最大数量 $N_{multicast-max}$，即：

$$N_{multicast-max} = \frac{T_{bg}}{T_{multicast}+T_{bg}} L \qquad 公式\ (5\text{-}16)$$

则此时多媒体多播帧离开队列转发到无线局域网中的时间间隔 $t_{multicast}$ 为：

$$t_{multicast} = \frac{L}{N_{multicast-max}} T_{avg-out} \qquad 公式\ (5\text{-}17)$$

所以，此时的 $t_{multicast}$ 反映的是多媒体多播帧转发到无线局域网中的即时的帧间隔情况。当无线 AP 无法应付大量涌入的帧，发生在新帧到达缓冲队列时无法入队，被丢弃时，若只考虑多播帧被丢弃的情况，则 $N_{multicast}$ 就会减少，从而使得 $t_{multicast}$ 也随之增大。直到 $N_{multicast}$ 等于 0 时，$t_{multicast}$ 变为无穷大，即此时多媒体多播流出现断流。

而如果此时开启基于优先级的 QoS 缓冲队列调度算法，为多媒体多播流预留缓冲队列资源，即每当发生多媒体多播帧无法进入队列时，将队列中的一个背景流的帧移出队列让多播帧可以顺利进入队列，则此时相当于 $N_{multicast}$ 就会出现增大的情况。直到 $N_{multicast}$ 增大到等于 L 时，队列中全都是多媒体多播流的数据帧，则此时该多播流在无线局域网中的转发速度完全等于该 AP 此时的转发速度，即等于该多媒体流生成数据帧的速度。

假设 AP 的转发速度下降使得 $T_{avg-out}$ 增大到 $T'_{avg-out}$，在未启动 QoS 队列调度时发生了多播帧无法入队而被丢弃的情况，使得此时队列中的多播帧数量减少了 n（$0 < n < N_{multicast-max}$），即多播帧的分发间隔 $t_{multicast}$ 为：

$$t_{multicast} = \frac{L}{N_{multicast-max-n}} T'_{avg-out} \qquad 公式\ (5\text{-}18)$$

而开启了QoS队列调度策略之后，由于这n个帧都能进队列，所以队列中的多播帧的数量不但不会减少，还会增加n个，此时多播帧的分发间隔$t'_{multicast}$为：

$$t'_{multicast} = \frac{L}{N_{multicast-max}+n} T'_{avg-out} \qquad \text{公式 (5-19)}$$

两者相减可得：

$$t_{multicast} - t'_{multicast} = \frac{L}{N_{multicast-max} - n} T'_{avg-out} - \frac{L}{N_{multicast-max} + n} T'_{avg-out}$$
$$= \left(\frac{1}{N_{multicast-max} - n} - \frac{1}{N_{multicast-max} + n} \right) L T'_{avg-out} > 0$$

公式 (5-20)

即：

$$t_{multicast} > t'_{multicast} \qquad \text{公式 (5-21)}$$

从推演结果看出，开启了QoS队列调度策略之后，多播帧的分发间隔比未开启之前要小。

第6章 仿真实验

本文实验主要基于 ns-3 网络实验仿真系统。如前文所述，ns-3 对无线模块的建模是非常细致的，甚至可以认为，ns-3 是特别针对无线仿真进行了优化定制的。ns-3 对无线网络的实验也提供非常好的支持，大部分的实现非常忠于标准和实际产品的实现。

6.1 可控丢帧率信道测试

本文的研究工作主要是基于网络仿真平台上进行网络实验，所采用的模型都是较为理想的经验模型，如 WifiChannel 模块中的传播损耗模型（PropagationLossModel）缺省采用的对数距离模型（LogDistancePropagationLossModel），再加上其缺省的参数配置，相当于在一个完全空旷无障碍物的球体自由空间中进行仿真，在实际仿真实验中发现，信道质量非常优秀，基本不会出现因为信道质量恶化造成丢帧的情形，而这与实际环境是有很大差距的。在实际环境中，有可能存在复杂的障碍物环境造成折射、反射和衍射，造成多径效应，甚至有其他无线电信号的干扰，都有可能造成信道恶化和误码率的提升。为了尽可能模拟真实的无线信道环境中容易出现误码率不稳定甚至恶化最终导致丢帧率提升的情况，本文研究工作中包括了一系列的基于 ns-3 平台的无线信道恶化测试实验，同时也是为本文后面的协议仿真测试提供一个可调的信道环境，监测信道恶化时的协议性能表现。

ns-3 的信道模型中，传播损耗模型缺省采用对数距离模型。如前文所述，而该模型对传播距离是非常敏感的，会随着距离的增加，其信号的功率损耗值会呈对数式上升，到达某个距离之后就渐趋稳定，逐渐逼近一个恒定值。如果该渐近不足以使得比特流的接收功率低于功率检测门限的话，这些比特仍然是可以被正确检测出来的，但链路层就会视为该帧仍能正确接收。如果该渐进值已经超越了接收功率和检测门限的差值，使得接收功率已经损耗到无法被正确检测，则此时就会出现大量的比特错误，导致在链路层视为数据帧无法顺利接收，而如果这种情况持续较长时间的话，在 MAC 层甚至会导致关联的丢失，使得 Station 变成不能接入状态。

这在实际的实验场景中也有所体现。将 Station 均匀分布在从 11 米到 50 米的距离范围内，每 1 米部署一个，总共 40 个 Station。所有 Station 不发送任何数据帧，只是不停接收来自 AP 的广播数据帧（多播与广播类似），经过仿真时间 10 分钟的测试，在 AP 和每一个 Station 上截取数据帧接收情况，并用 Station 的接收数据量除与 AP 的发送数据量，再用 100%减，就可以得到每个 Station 的在其位置上的 10 分钟内的平均丢帧率。

通过调整 WifiChannel 中的对数距离传播损耗模型的 n 值，当 n 值增大到一定程度之后，例如 n=4，则在 44 米之外的 Station 都会变成不可接入状态，基本上不能收到任何来自 AP 转发的帧，而在距离 36 米以内的 Station 则能够正常接收大部分的数据帧，将所有 Station 的吞吐率作成图表的话，如图 6-1 所示，就可以看到在该距离 43 米处附近出现陡坡式的吞吐率下降，从较为平稳的正常值忽然下降到接近 0 点。但在 36 米到 43 米这段区间范围内，虽然这些 Station 也会出现丢帧，但并不至于恶化到不可接入的状态，本文称之为“过渡区域”。而在 n 值为其他值时也有类似的情况出现，只是该距离的区间的位置不同而已。如图所示。所以后文的所有仿真实验涉及信道丢帧率恶化控制的实验环境部署，都是采用以下这种方法，即将 Station 部署在所设置的 n 值对应的那段“过渡区域”内，使得每个 Station 的所处位置的平均丢帧率在一个可控的范围内，这样就可以比较准确地看出在不同丢帧率的信道情况下，本文所设计的协议的性能会受到什么影响。

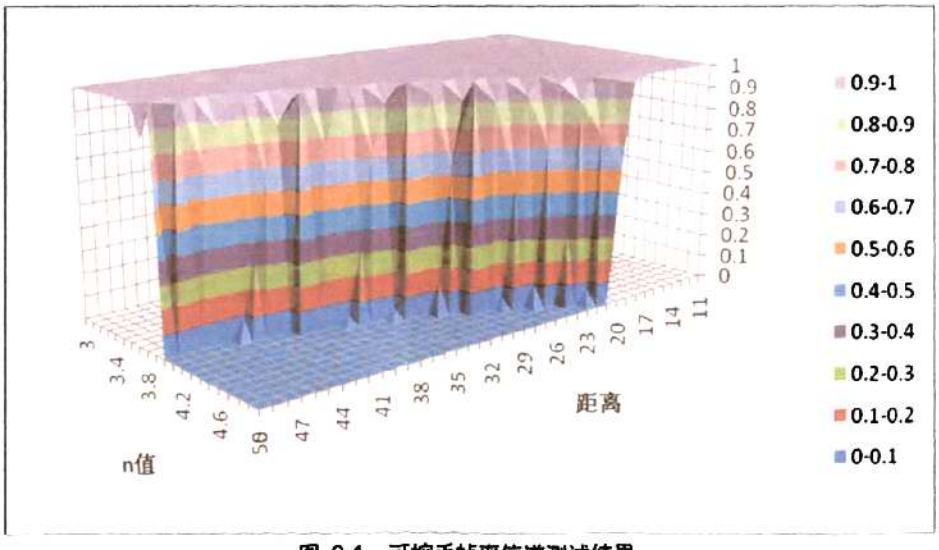


图 6-1　可控丢帧率信道测试结果

6.2 ns-3 仿真环境拓扑

本文所有的实验都是基于以下的网络拓扑结构（如图 6-2）。该拓扑是一个有基础设施的无线网络，只有一个 AP，而且兼任路由器的功能，负责将来自于有线骨干网的流量转发到其覆盖的无线局域网内，或者将来自无线局域网内的流量转发到有线骨干网上。在有线骨干网上有两组服务器。一组是单一多播源服务器，用于产生单个模拟多媒体流量的业务流，并以多播形式将该业务流通过 AP 发向位于无线局域网内的多播组成员站点。另一组是 N 个背景流服务器，用于产生 N 个 TCP 背景流或 N 个 UDP 背景流。为了避免由于服务器内部的流量拥塞对实验产生的影响，所以采用一个服务器负责产生一个背景流的做法。在无线局域网内，也有两组无线终端站点。一组是多播组成员站点，用于接收来自有线骨干网的多媒体多播服务器产生的业务流，但并不产生上传数据流量（除了控制帧之外）。另外一组背景流站点，同样也是 N 个站点，对应有线骨干网上的 N 个背景流服务器，每一对服务器和无线终端站点之间产生一条单向的 TCP 流或 UDP 流。这种设定主要是针对后文的拥塞测试，模拟下载流量在 AP 处产生拥塞而对多播流产生影响的情景。

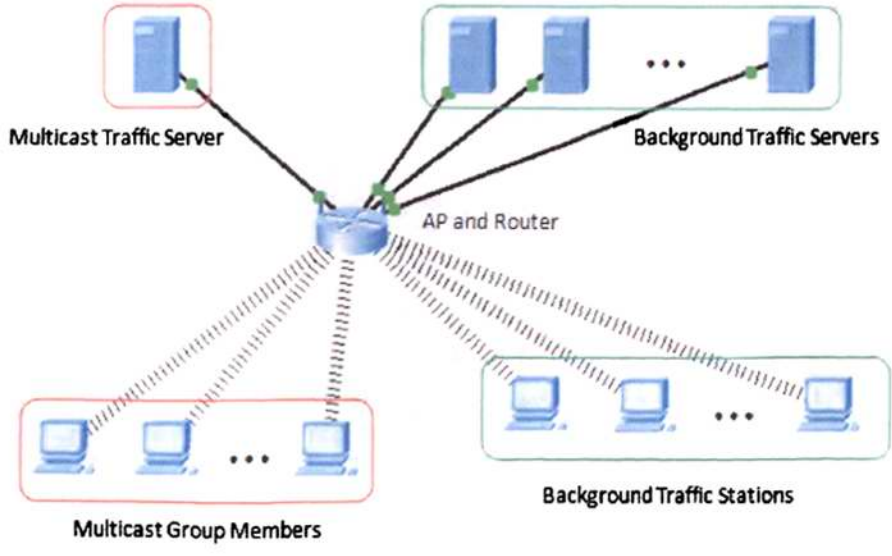


图 6-2　ns-3 仿真实验网络拓扑

6.3 ns-3 仿真参数设定

各实验都通用的 ns-3 仿真参数如表格 6-1 所示：

表格 6-1　ns-3 实验平台通用参数设定

参数名	参数值
仿真随机种子（Seed）	1
仿真随机运行序数（Run）	3
多播组成员终端数	10
终端移动	无
仿真时间	1800 秒
多播业务流数	1
多播数据流速率	1.5Mbps（Mpeg-1 速率）
多播帧帧长	128 字节
OnOff 模型	On = 1, Off = 0 （CBR）

6.3.1 仿真随机种子（Seed）和运行序数（Run）

ns-3 网络仿真系统有一个专门的随机数系统[21]，通过设定随机数所符合的分布规律，按照特定的数学模型生成随机数序列。在每一次仿真里面所有涉及到的随机数生成函数的调用，都需要一个统一的随机种子和运行序数作为起始值。当随机种子和运行序数确定之后，所有后续产生的随机数序列都是确定的。也就是说，只要给定相同的随机种子和运行序数，就可以产生相同的实验结果，反之，需要产生多次不同的实验结果，就要更改随机种子或运行序数。

6.3.2 多播成员终端数和多播业务流数

多播成员终端是指加入了同一个多播组的所有无线终端站点。虽然所有成员站点每次接收数据帧都是同一份数据拷贝，但由于所处位置距离和随机数生成结果的不同，每个站点最终的接收流量的情况都会有所不同。设置 10 个站点只是为了在统计时可以取平均值，或者分列 10 个受测站点的情况以观察变化趋势。而由于本文实验并不涉及 MLBP 协议多个流之间竞争的测试，所以只设置了单个多播流作为测试流。

6.3.3 仿真时间

仿真时间是指仿真实验里模拟实际环境中经历的时间长度，并不是指仿真实验实际运行的时间。该时间长度是根据所观察到的网络流量是否达到稳定或者以某种大致的周期进行重复而定。因为网络流量是时刻变化的，随机性相当大。就算是仿真器的伪随机性，也会因为随机数序列的回绕周期相当长，而使得实验的结果在相当长时间内都不会体现一定的周期性。所以本文根据具体的实验结果，发现 1800 秒内大部分的实验数据都渐趋稳定，所以只是截取到 1800 秒的仿真时长，确保足以能从实验数据中大致观察到网络流量的变化趋势。

6.3.4 多播流速率

该速率是指 ns-3 应用层产生数据帧的速率，通过该速率和帧长两个值，就可以求出每个帧产生的间隔时间。由于 802.11 标准中规定，在 AP 进行多播传送时只能采用基本速率模式[16]，而 ns-3 默认采用的是 802.11a 的物理层标准，所以其基本速率为 6Mbps，而这也使得我所能采用的多播速率不能超越该值。为了避免使信道过度饱和，影响测试结果，所以暂且取 1.5Mbps，足以满足 MPEG-1 的传输速率 1.2Mbps[30]，该速率足以承载 MPEG-1 的媒体，即标准的 VCD 画质。

6.3.5 多播帧帧长

ns-3 缺省的帧长是设置为 1024 字节，即由应用层数据源生成器产生数据帧的大小。由于帧长除与速率就可以得到帧之间的生成间隔，为了模拟更加频繁的数据帧产生效果，以满足下文的实验测试的需求，所以本文将帧长缩短为 128 字节。

6.3.6 终端移动

经过前期多次的测试实验发现，ns-3 中对于终端节点因移动所产生的一些反映在网络传输上的效应，建模似乎并不足够，类似多普勒效应等也没有模拟出来，所以实际上对网络传输真正起影响作用的是节点之间的空间距离。ns-3 的移动模型只是让空间距离按照一定的速率和方式进行改变而已。所以本文最后决定不采用 ns-3 的移动模型，而只是采用初始定位的改变来测试不同距离下的传输效果。

6.3.7 On-Off 模型

On-Off 模型是非常经典的网络数据流量生成模型。通过 On 和 Off 两个时间

长短以及在两个时间段内数据流量的设定，可以模拟许多复杂的网络流量模型，如泊松流等。本文主要模拟速率较为恒定的但流量较小的多媒体流，所以采用类似CBR的网络流设定方法作为代替，即On时长设为1，Off时长设为0，即没有Off的时间，数据源会不停地按照指定的速率和帧长生成数据帧，直到数据源被关闭或者仿真结束为止。

6.4 信道质量恶化测试实验

该项实验利用前文提到的可控丢帧率信道的构建方法，通过调节无线信道的传播损耗模型的参数，以及调节无线终端站点部署位置离AP的距离远近，从而获得不同丢帧率的无线信道。通过用不同质量的信道对MLBP协议中基于概率重传的算法进行对比测试，观察其应对信道质量恶化时的抖动抑制能力。由于该项实验主要是测试应对信道质量改变的能力，不涉及多流量拥塞的影响问题，所以不加入任何背景流，只保留单条多播流。

MLBP协议中基于概率的信道错误重传机制，涉及到一个可容忍丢帧率的设定，通常是提供给上层协议作为一个可调参数的接口。在实验中为了模拟实际多媒体业务体验的效果，将该参数设定为0.01，即1%的可容忍丢帧率。

6.4.1 实验方案

该实验采用以下四组测试方案：

- 无丢帧理想信道：对数距离传播损耗模型的n值设置为缺省值3，无线终端离AP距离在25米左右，模拟较为理性的无线信道环境，丢帧率接近0。
- 低丢帧率信道：设置对数距离传播损耗模型的n值为3.8，无线终端离AP距离在46米左右，根据上文的信道测试结果，此时的丢帧率大约在1%左右。
- 中丢帧率信道：设置对数距离传播损耗模型的n值为4.1，无线终端离AP距离在36米左右，根据上文的信道测试结果，此时的丢帧率大约在5%左右。
- 高丢帧率信道：设置对数距离传播损耗模型的n值为4.3，无线终端离AP距离在31米左右，根据上文的信道测试结果，此时的丢帧率大约在10%左右。

实验采用10个位置固定的受测站点，通过AP不断对其进行多播流的传输，观察在不同的信道情况下10个受测站点的业务流传输情况。尤其是观察其帧接

收间隔和抖动的指标，验证 MLBP 协议的基于概率的信道错误重传恢复算法是否能够抑制抖动，缩短帧间隔。

6.4.2 实验结果及分析

该实验主要测试 MLBP 协议在信道恶化，丢帧率上升的情况下，性能是否比原 LBP 协议更优。测试的性能指标包括帧间隔、抖动和吞吐率三项。通过计算帧间隔的均值，可以分析实验过程中业务流的顺畅度，是否出现较大的帧间等待甚至断流的情况。通过计算帧间隔的标准差，可以分析业务流的抖动情况，从而判断业务流的平滑度。而吞吐率指标主要是统计其均值，分析在牺牲了可靠度来满足抖动抑制的情况下，吞吐率受影响的程度。

帧间隔与抖动

帧间隔是指接收端链路层得到的非重复的 Data 数据帧之间的接收间隔，因为只有在物理层正确接收的数据帧才会送到链路层，所以链路层得到的数据帧都是完整的，但是否正确还要通过校验。控制帧诸如 ACK 是不进行统计的，因为控制帧并没有携带有效载荷（Payload），不会提取有效数据传递给上层协议，所以对于上层协议来说这些帧是不可见的。对于重复帧也会进行剔除，因为会出现重传的原因，造成收到重复帧，在统计业务流帧数据时，已将这些重复帧自动剔除，剔除方法是跟踪 MAC 帧头的序列控制字段值进行辨别。

抖动是指帧间隔的变化情况，帧间隔的变化越小，则视为抖动越小，此时的业务流就越平滑，反之亦然。所以对此指标最好的统计方法就是计算帧间隔的标准差，通过标准差的大小间接反映抖动情况，标准差越大，抖动越剧烈，标准差越小，抖动就越平缓。

图 6-3、图 6-4、图 6-5 和图 6-6 分别反映了在信道模拟丢帧率为 0%、1%、5%和 10%时业务流的帧间隔情况。横轴是 10 个被测试的无线终端 Station 的标号，每一个 Station 对应的点就是指该 Station 在测试期间，所有数据帧的平均帧间隔值。在进行绘图之前已经将 10 个 Station 的均值按从小到大进行升序排列，方便进行对比。

从四幅图上可以看到，当丢帧率为 0%时，10 个 Station 的接收帧间隔都非常稳定，LBP 协议对应的图线大约在 14.35ms 左右，而 MLBP 协议大概维持在 14.33ms 左右。而随着丢帧率的增大，从 1%开始到 5%，两个协议下的 Station

都出现了不同程度的帧间隔时延增大，LBP 协议下的帧间隔均值大部分都已经超过了 14.35ms，甚至在 5%时到达了 14.4ms 以上，而 MLBP 协议下则总体仍然维持在 14.35ms 以下的水平，差距逐渐拉开。到了 10%的高丢帧率时，Station 之间的帧间隔均值的差距就更加明显。LBP 协议下的帧间隔均值已经到达了 14.45ms 左右的范围，而 MLBP 协议下虽然总体的帧间隔均值也有大幅的提升，但依然维持在 14.4ms 以下的水平，比 LBP 协议在 5%丢帧率时的水平还低。两协议的帧间隔均值虽然都随着丢帧率的增大而有了显著的改变，但协议之间的差距是非常明显的，MLBP 协议对帧间隔的控制还是比较好的，即使在丢帧率较高的情况下。

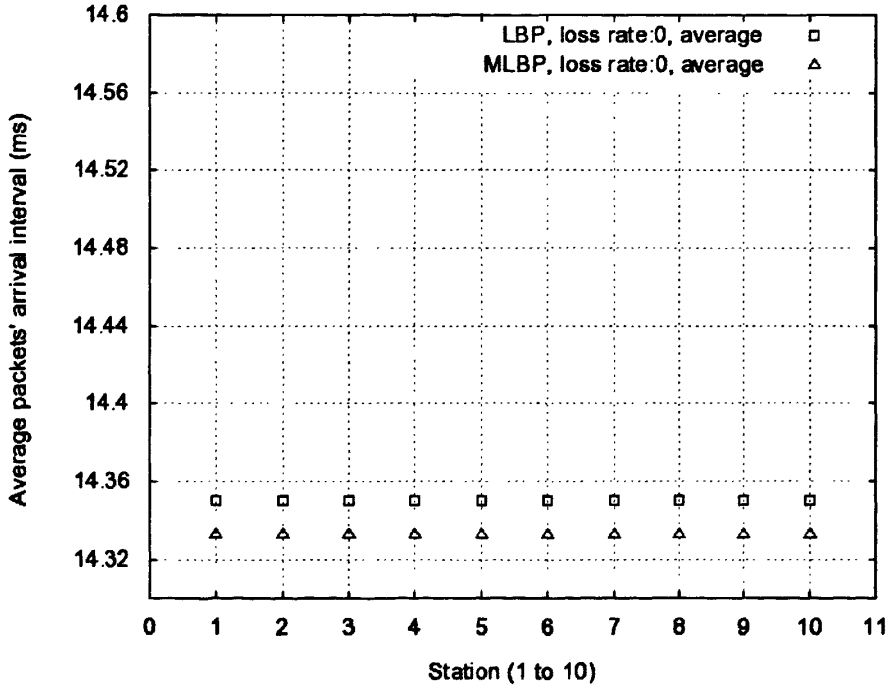


图 6-3 帧间隔均值对比，0%丢帧，10 个受测 Station

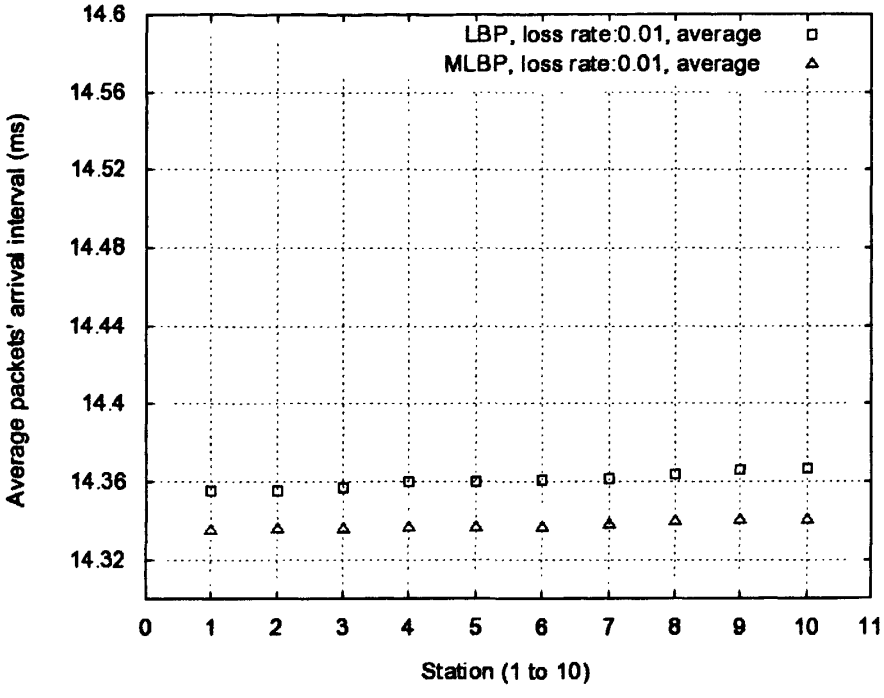


图 6-4 帧间隔均值对比，1%丢帧，10 个受测 Station

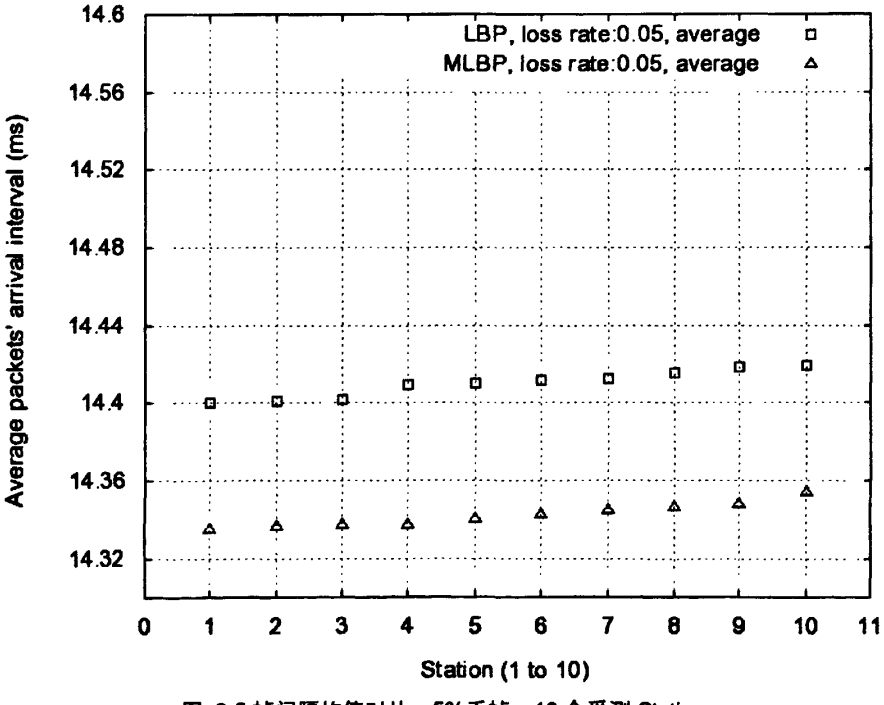


图 6-5 帧间隔均值对比，5%丢帧，10 个受测 Station

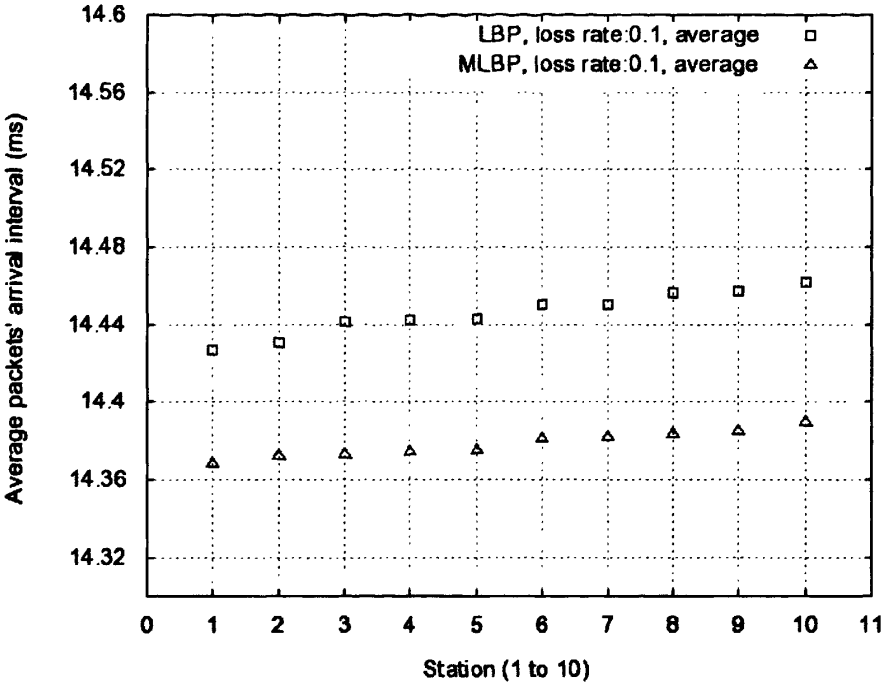


图 6-6 帧间隔均值对比，10%丢帧，10 个受测 Station

图 6-7、图 6-8、图 6-9 和图 6-10 则是对不同信道丢帧率 0%、1%、5%和 10%的情况下，业务流的抖动情况的反映。同样图上 10 对数据点是代表 10 个受测 Station 的各自统计值，每一个 Station 的帧间隔的统计标准差值就对应一个数据点。通过图线可以看出，与帧间隔均值的情况类似，当在低丢帧率的情况下，10 个 Station 的抖动都可以维持在相当平稳的水平上，LBP 协议下的抖动值大概在 18ms 左右，而 MLBP 协议也是维持在 17.95ms 和 18ms 之间。即使到了丢帧率为 1%时，两协议下的抖动情况实际上变化不大，只是 Station 之间出现了一点差距，某些 Station 的抖动值开始轻微上升。但当出现较高丢帧率时，如 5%，LBP 协议下的抖动值已经完全超过了 18ms，最大抖动的一个 Station 的值已经接近 18.05ms，而 MLBP 协议下的抖动情况则变化不大，依旧在 17.95ms 和 18ms 之间的位置稳定着。而到了 10%的丢帧率情况下，LBP 协议下的抖动值已经总体到达 18.05ms 的水平，而 MLBP 协议下的抖动值依然保持在 18ms 以下，变化依然不算明显，两协议的差距已经拉开。由此可以看出，无论是帧间隔和抖动，MLBP 协议都能有效地进行抑制和稳定，即使在信道恶化的情况下依然可以做到

这一点，使得所承载的业务流更加顺畅更加平滑，最终的用户体验的质量也更加优秀。

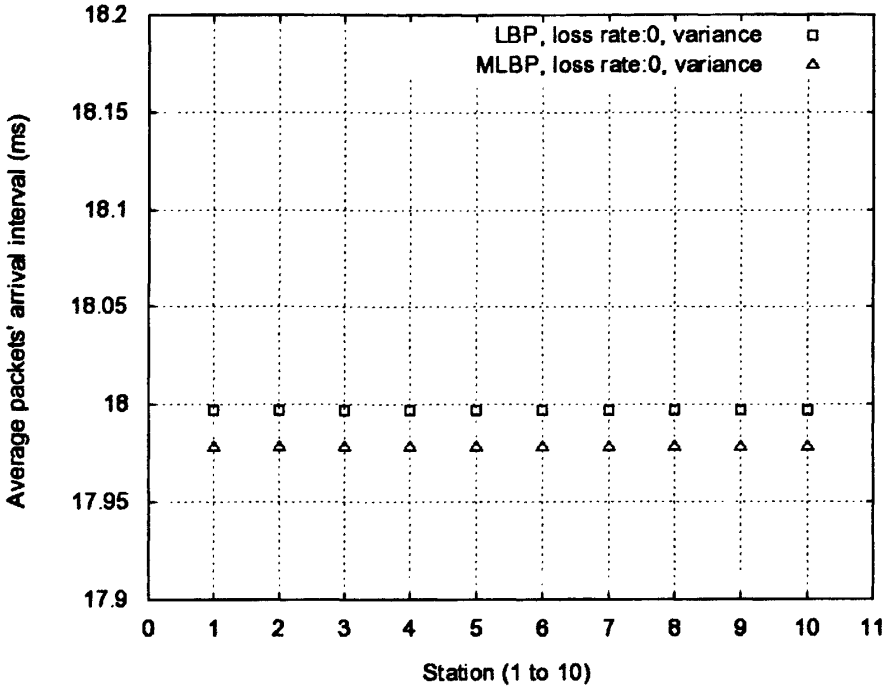


图 6-7 抖动对比，0%丢帧，10 个受测 Station

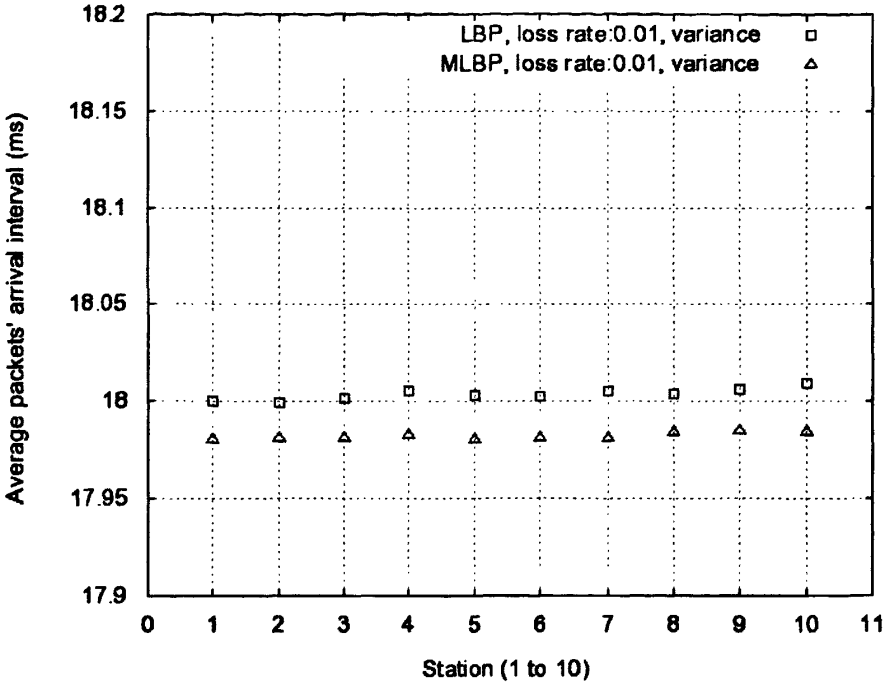


图 6-8 抖动对比，1%丢帧，10 个受测 Station

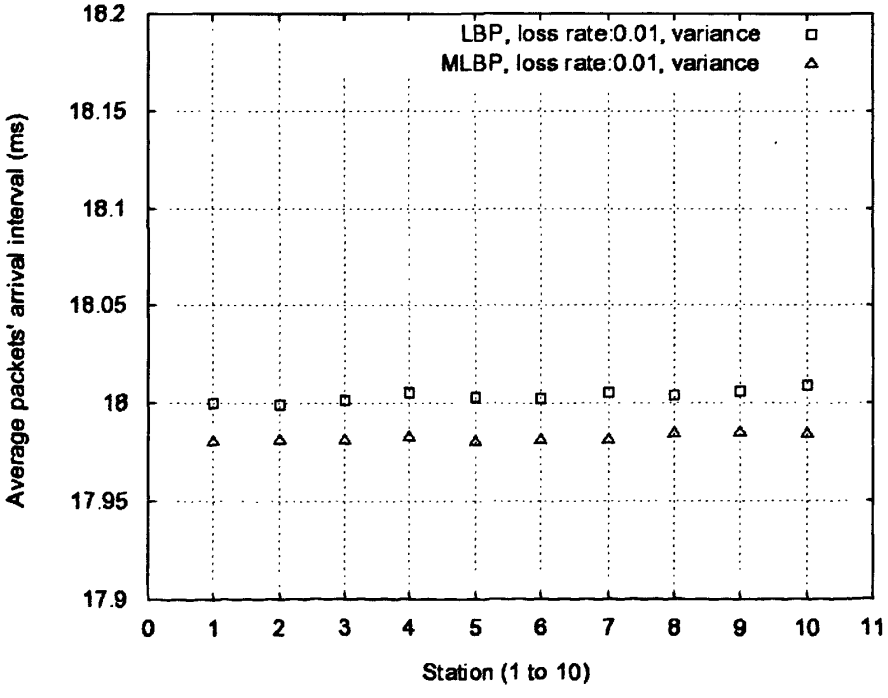


图 6-9 抖动对比，5%丢帧，10 个受测 Station

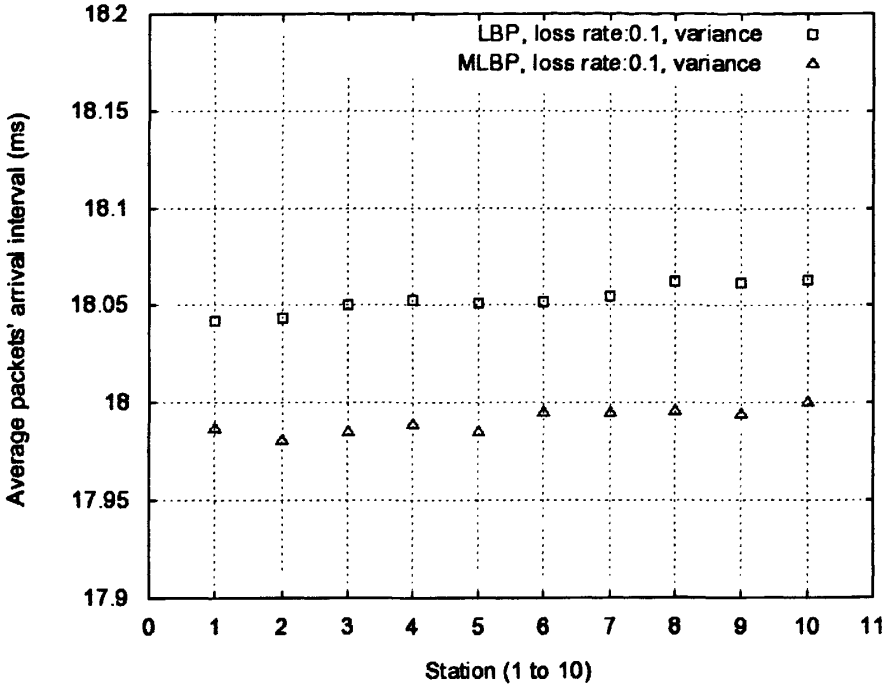


图 6-10 抖动对比，10%丢帧，10 个受测 Station

实测丢帧率

正如前文提到，MLBP 协议的基于概率的信道重传恢复机制，是通过牺牲可靠性来换取短帧间隔和低抖动的。不同丢帧率预设参数的信道情况下，实测的协议丢帧率将会随着协议的不同而有所差异。图 6-11、图 6-12、图 6-13 和图 6-14 分别了反映 0%、1%、5%和 10%预设信道丢帧率的情况下，协议丢帧率的实测结果。从四幅图上可以看出，在 LBP 协议下，实测丢帧率跟预设丢帧率基本符合，而 MLBP 协议下，实测丢帧率平均比预设丢帧率高出 1%，而且两协议的这个差距并没有随着预设丢帧率的提升而增大。同时可以看出，10 个 Station 的情况基本一致，不会出现很特别的个体。结合之前预设的可容忍丢帧率为 1%的设定值，本文判断该实测丢帧率的差距是跟这个参数的设定值是相关的，该值设定越大，自然 MLBP 协议的所牺牲的可靠性就更多。

由此看出，虽然 MLBP 协议的实测丢帧率有所提升，但恶化程度并不非常明显，基本在可接受的范围内。

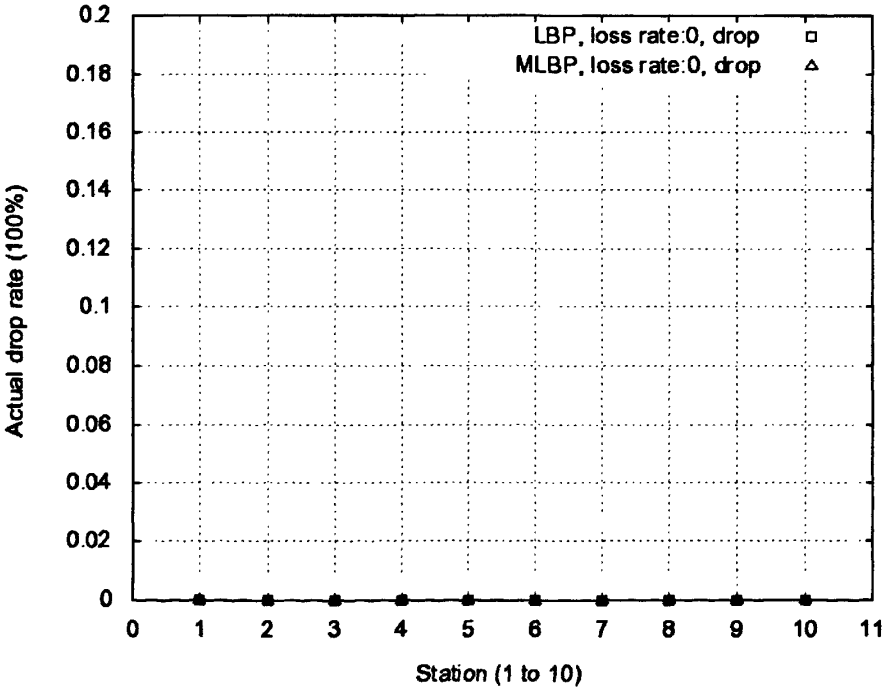


图 6-11　LBP 和 MLBP 协议在信道预设丢帧率为 0%时的实测丢帧率

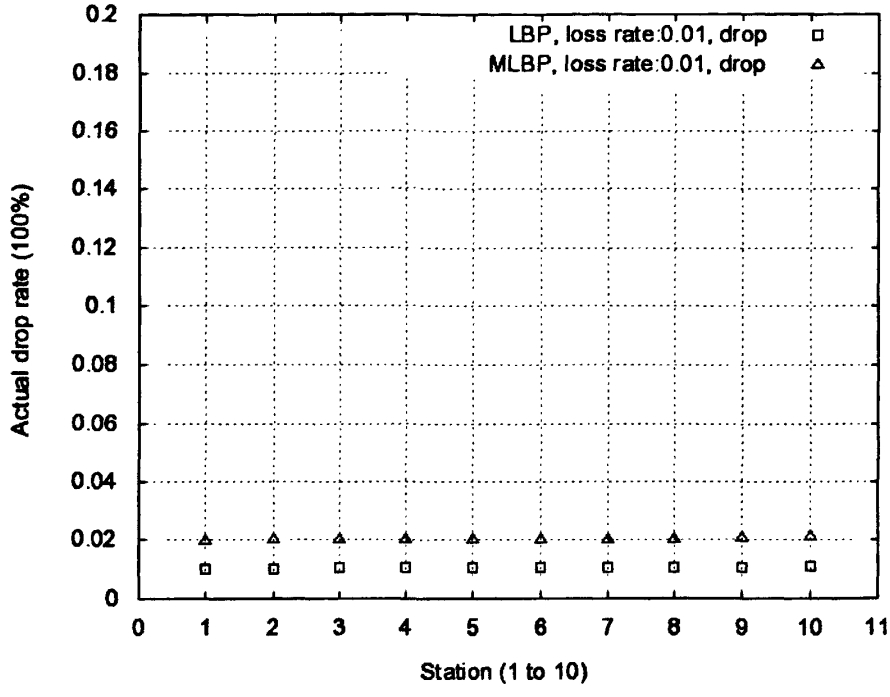


图 6-12　LBP 和 MLBP 协议在信道预设丢帧率为 1%时的实测丢帧率

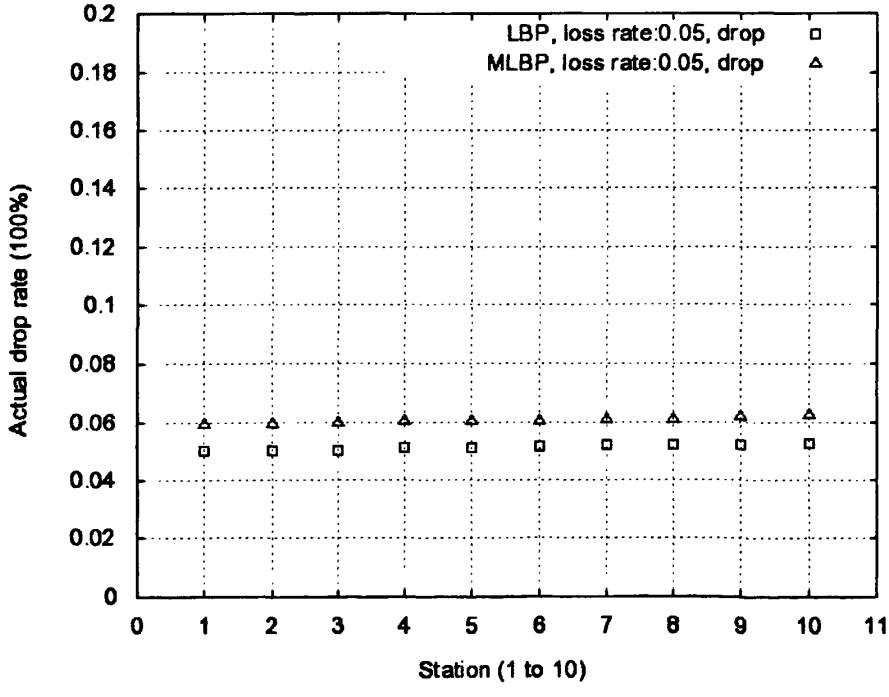


图 6-13　LBP 和 MLBP 协议在信道预设丢帧率为 5%时的实测丢帧率

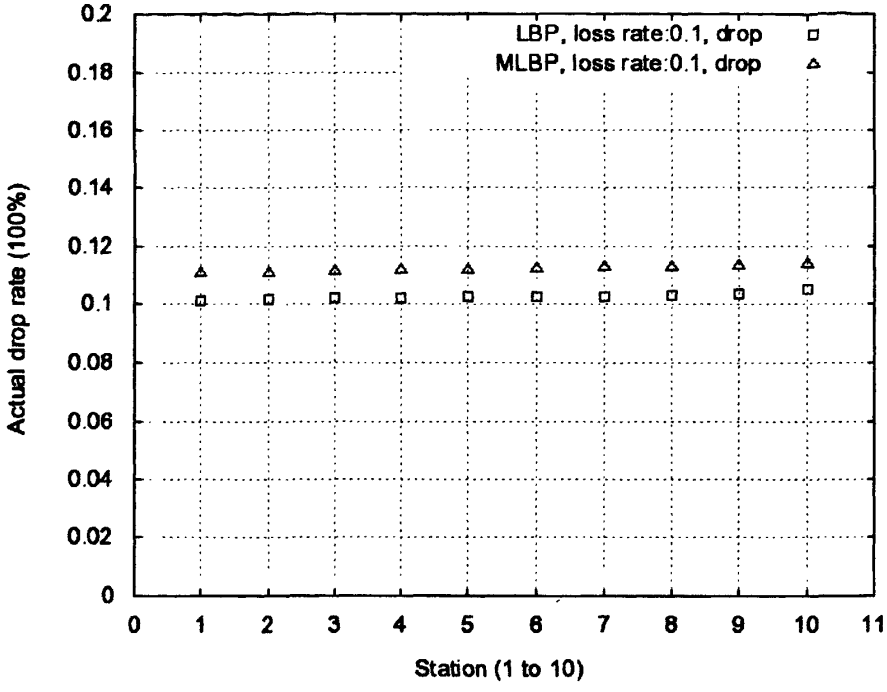


图 6-14　LBP 和 MLBP 协议在信道预设丢帧率为 10%时的实测丢帧率

吞吐率

吞吐率也是直接影响协议最终的用户体验的一个指标。MLBP 协议虽然牺牲了可靠性，在实测丢帧率的指标上有所下降，但在吞吐率指标上，并不一定比 LBP 协议差。

通过实际仿真实验后得出的结果，如图 6-15、图 6-16、图 6-17 和图 6-18 所示，在 0%、1%、5%和 10%四种丢帧率的信道情况下，对所有 10 个受测 Station 的吞吐率进行求平均值，发现 MLBP 协议下的吞吐率明显高于 LBP 协议，MLBP 协议在 0%丢帧的情况下平均吞吐率比 LBP 协议高出接近 0.45Mbps，而到了 10%的丢帧率情况下，这个差距更拉大到 0.9Mbps。而且明显看出 MLBP 协议对于丢帧率上升的情况下，吞吐率的变化并不大，而 LBP 协议的吞吐率则随着丢帧率的上升而下降明显。这说明在丢帧率改变时吞吐率的稳定性，MLBP 协议比 LBP 协议要更好。而出现该现象的原因，可以归结为 MLBP 协议的重传次数少了，帧间等待的时间缩短，使得更多的数据帧可以发送到无线链路上，最终使得在丢帧率高的情况下，反而吞吐率会有所上升。

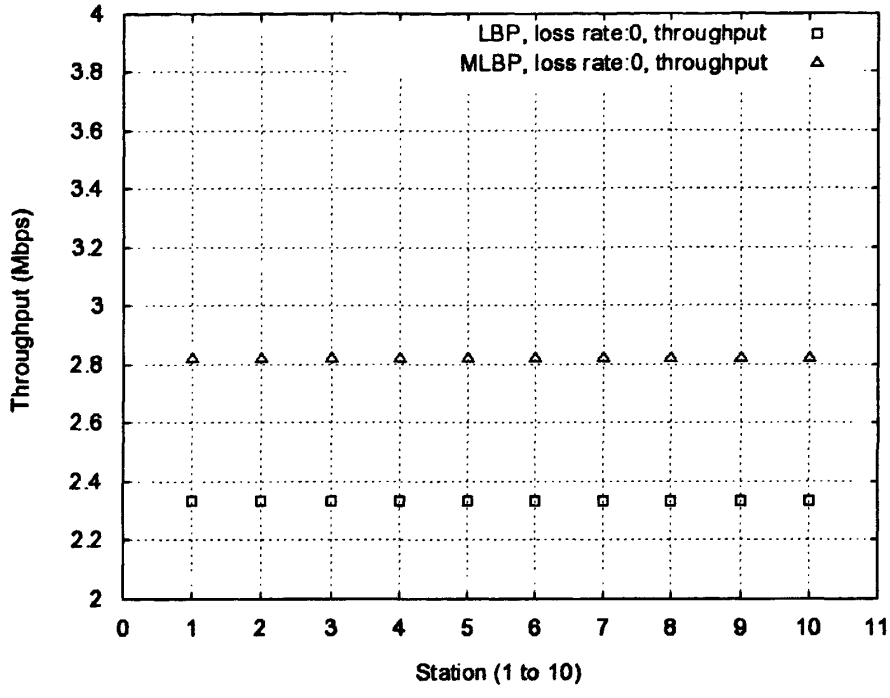


图 6-15 LBP 和 MLBP 协议在信道预设丢帧率为 0%时的实测吞吐率

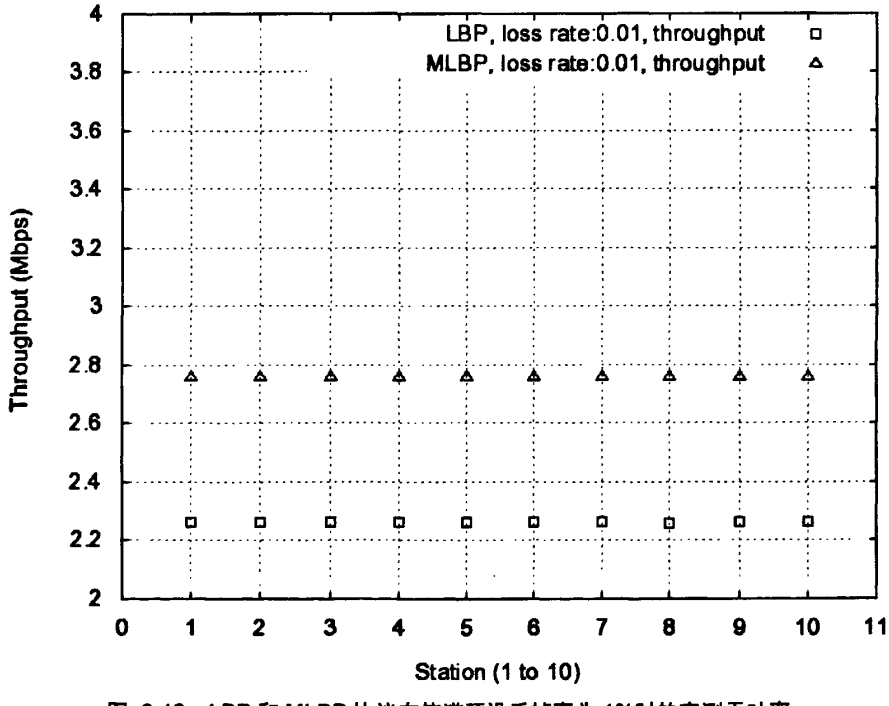


图 6-16 LBP 和 MLBP 协议在信道预设丢帧率为 1%时的实测吞吐率

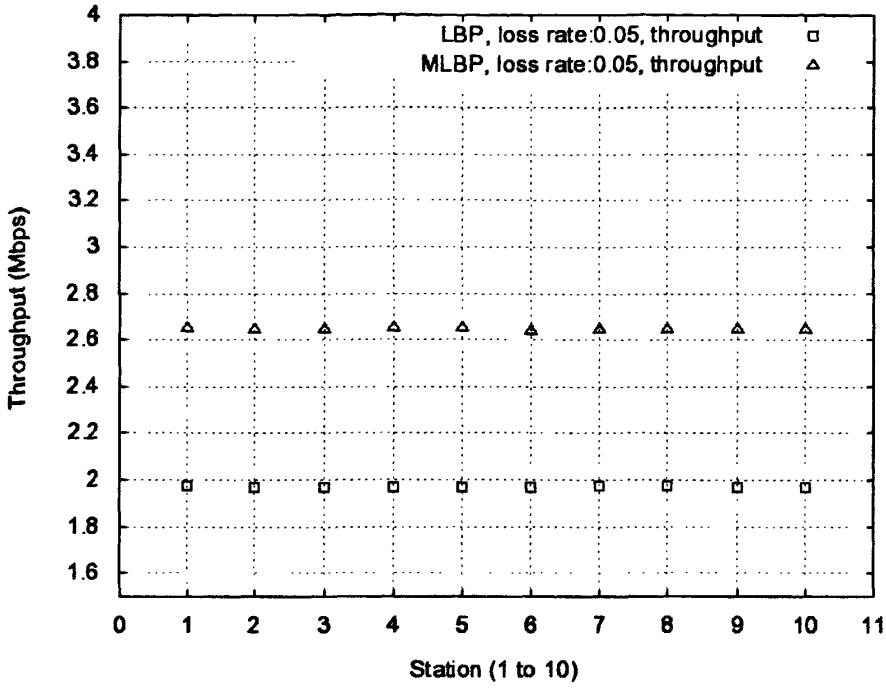


图 6-17　LBP 和 MLBP 协议在信道预设丢帧率为 5%时的实测吞吐率

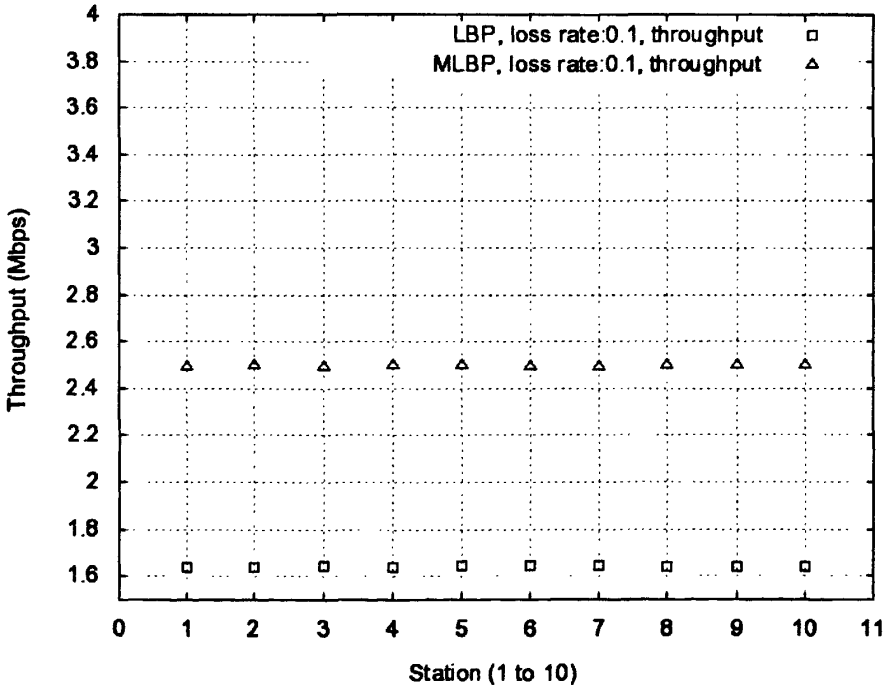


图 6-18　LBP 和 MLBP 协议在信道预设丢帧率为 10%时的实测吞吐率

6.5 高负载背景流拥塞测试实验

该项实验采用多种背景流的组合方式，针对 MLBP 协议中基于优先级的 QoS 缓冲队列调度算法，测试其应付队列拥塞时稳定多播业务流量的能力。实验主要模拟高负载的下载流量在到达 AP 处产生瓶颈效应，造成 AP 的缓冲队列拥塞，从而影响多播流的传送，所以所有测试用的背景流都是采用从 AP 到 Station 的下载流。该项实验由于只是测试 AP 的 MAC 层缓冲队列的调度能力，不涉及信道上的性能调节，所以不增加不同质量信道的对比测试，无线信道采用对数距离传播损耗模型，并且全部采用 ns-3 缺省参数设定，信道质量较为理想，一般不会出现因信道误码而造成丢帧的情况。

实验中涉及两种背景流，一种是 UDP 流，另一种是 TCP 流，而且都采用增加背景流的线程数来逼近信道的容量极限，以此测试 MLBP 协议的抗冲击能力，测试其是否能保持稳定的流量传输。以下分别对不同测试条件下的实验结果进行分析。

6.5.1 UDP 负载背景流测试

实验采用六组测试方案和两套测试设备：

- 2 组速率为 1Mbps 的 UDP 数据流
- 4 组速率为 1Mbps 的 UDP 数据流
- 6 组速率为 1Mbps 的 UDP 数据流
- 8 组速率为 1Mbps 的 UDP 数据流
- 10 组速率为 1Mbps 的 UDP 数据流

通过这 5 组测试方案的对比，测试不同协议不同负载的背景流的环境下，MLBP 协议和无 QoS 队列的 LBP 协议的性能差别。

该项实验通过在 AP 上捕捉转发到无线局域网内的所有帧，通过对结果进行一定的筛选，测试出多播流的吞吐率、帧间隔以及抖动等三个指标，同时提供 UDP 背景流受影响的程度的参考。

实验结果

UDP 流是非连接的协议上，不需要确认机制，没有任何速度调节和拥塞控制机制，属于一种比较霸道的流量，会不断用恒定的速率冲击信道，使得其他业

务流受很大影响甚至不能接入信道。这部分的实验是在 UDP 背景流的测试环境下，分别对帧间隔和吞吐率两大指标进行统计，其中帧间隔的图表中还包括了帧间隔的均值和标准差的统计情况展示。帧间隔的统计采用“分桶法”进行统计。先将所有接收端链路层收到的非重复数据帧的接收间隔时间计算出来，即带有有效载荷的 Data 帧，不包括控制帧，而且按照 MAC 帧头的序号控制字段值排除重复接收的帧。然后将所有间隔时间按照每桶 0.1ms 的精度进行归类统计，然后计算出每一桶的数量占总数量的比例，计算出百分比，作为该桶所对应的间隔值的统计指标。所以短间隔值所占比例越高，表明帧间隔的均值越小，间隔值的分布越集中，表明业务流的抖动越小。

图 6-19、图 6-20、图 6-21、图 6-22 和图 6-23 分别反映 UDP 背景流为 2 线程、4 线程、6 线程、8 线程和 10 线程时的帧间隔统计图。其中横轴为间隔时长的分布，纵轴为相应的百分比比例。右上角的图例中还包含了对应图线的统计均值（avg）和统计标准差（var）。

从图 6-19 可以看出，在 2 个 udp 线程的背景流情况下，LBP 协议和 MLBP 协议的下的帧间隔都集中在 0.5ms 和 0.9ms 两个值上，但很明显可以看出，MLBP 协议在 0.5ms 处更加集中，占总样本数的 46%的比例，即有将近一半的帧间隔是维持在 0.5ms 左右的水平，而 LBP 协议在 0.5ms 处只占总样本数的 18%，而更大比例的间隔值则分布在 0.9ms 左右，约占总样本数的 28%的比例，而且 LBP 图线的后延“尾巴”比 MLBP 的要更长，一直到 3ms 以后仍然有样本出现。为了使统计图显示得更加清晰，笔者在作图时已经将 3.2ms 后面的部分截去了，但实际上 LBP 协议在 3.2ms 间隔值的后面仍然有少量样本分布，说明某些帧之间的间隔值会变得比较大，从这方面也可以反映出 LBP 协议的抖动相比 MLBP 协议是较大的，在 2 个 udp 线程的背景流下就已经有此差距了。而且从右上角的统计值也可以看出，LBP 协议下的帧间隔均值为 0. 89924 ms，比 MLBP 协议的 0. 68269 ms 要大了 0.22ms 左右，而 LBP 协议的间隔标准差 0. 275843 ms，更是比 MLBP 协议的 0. 168163 ms 大了将近 0.21ms，抖动的差距已经非常明显。

同样从图 6-20、图 6-21、图 6-22 和图 6-23 也可以看出类似的差距。在图 6-20 上显示的 4 个 UDP 背景流的情况下，LBP 协议的间隔值在 0.5ms 处已经只有不到 10%的样本分布，更多的是分布在 0.9ms 左右和后面更大的间隔值的尾巴

上。其均值也超过了 1ms，达到了 1.03591ms，标准差也升到了 0.343905ms。到了图 6-21 所显示的 6 个 UDP 背景流的情况下，LBP 协议下的间隔值大部分都分布在 0.9ms 处，而且更多的样本出现在后面的“尾巴”，例如 1.2ms 处。均值也上升到了 1.28365ms，标准差为 0.595046ms。而到了图 6-22 的 8 个 UDP 背景流情况下，LBP 协议的性能更是下降明显，间隔值的峰值已经集中到了 0.9ms 和 1.9ms 之间，均值也达到了 1.69677ms，而标准差也突破了 1ms，达到了 1.02938ms，帧间隔和抖动的增大非常明显。到了图 6-23 所显示的 10 个 UDP 背景流的情况下，跟 8 个 UDP 背景流的情况已经差别不大，但依然有一定的恶化趋势，LBP 协议下多播流的帧间隔均值已经达到了 1.90207ms，标准差达到了 1.2773ms，仍然处于增大的趋势上。根据前文的协议性能分析章节的结论，出现这种情况的原因极大可能就是背景流对多播流的冲击使得多播流的帧无法顺利进入缓冲队列，使得有相当一段时间内出现断流，没有多播帧可以转发出去，从而造成前后两次顺利接收到多播帧的间隔增大。

另一方面从四幅图上观察 MLBP 协议的表现，会很容易发现，不论 UDP 背景流的数量增大多少，MLBP 协议所承载的多播流的帧间隔值都非常稳定，大部分样本基本上集中在 0.5ms 左右，没有发生很大的变化，五个图上的 MLBP 图线都几乎是重合的。而且看右上角的均值统计和标准差统计，从统计的角度来看也可以得出同样的结论。

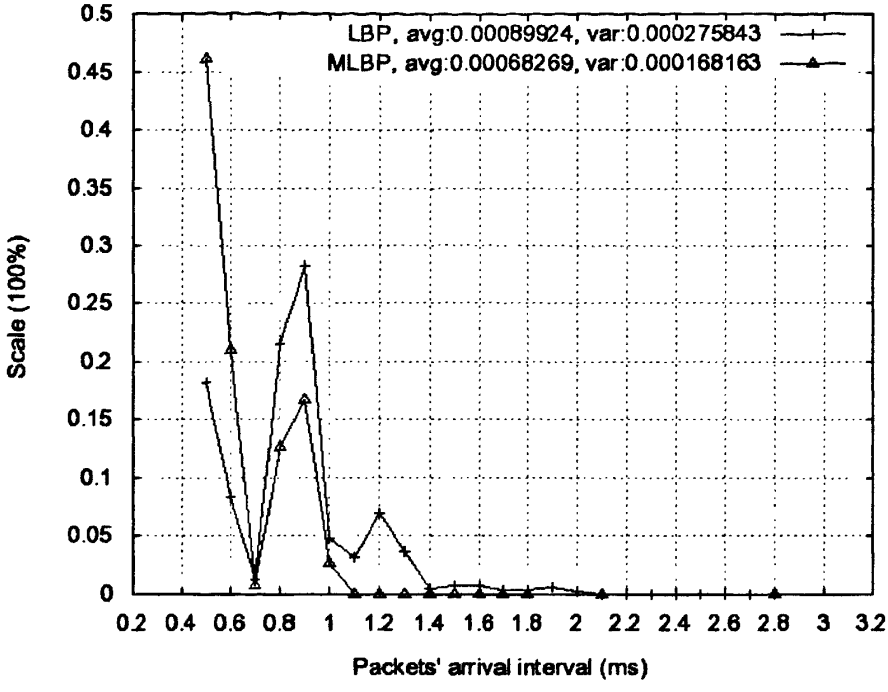


图 6-19 帧间隔对比，2 个 1Mbps 的 UDP 背景流

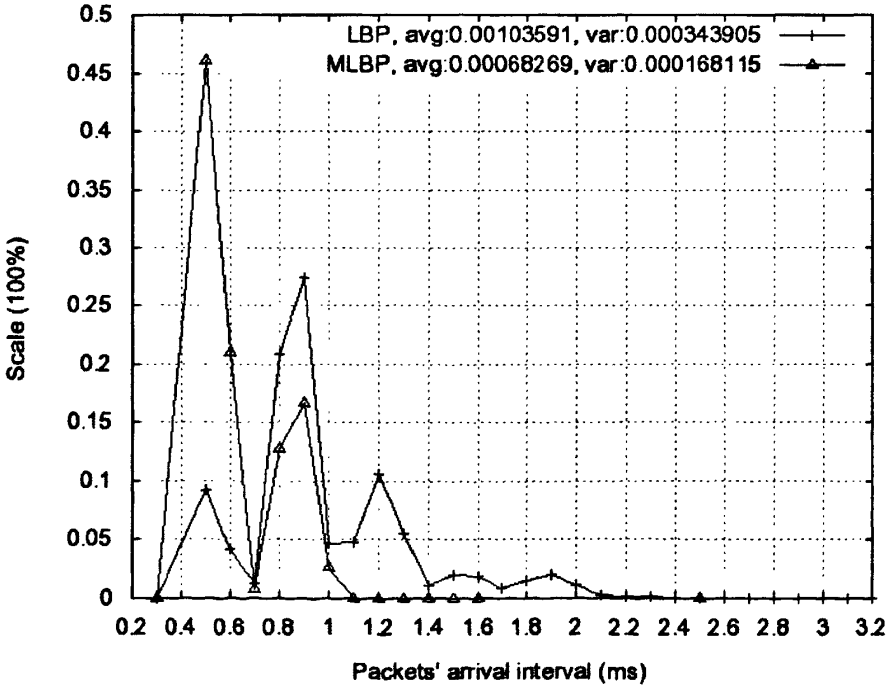


图 6-20 帧间隔对比，4 个 1Mbps 的 UDP 背景流

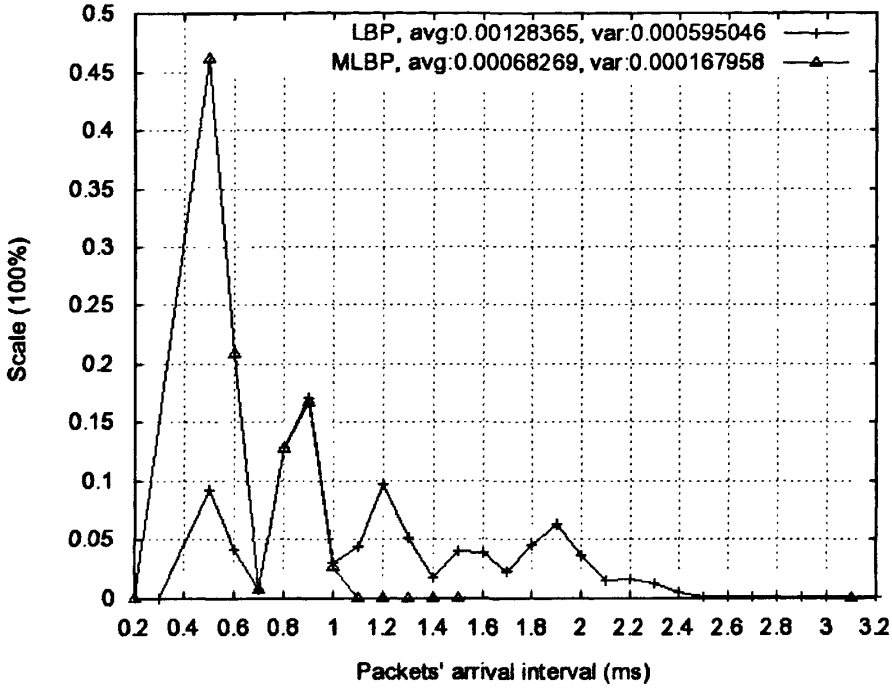


图 6-21 帧间隔对比，6 个 1Mbps 的 UDP 背景流

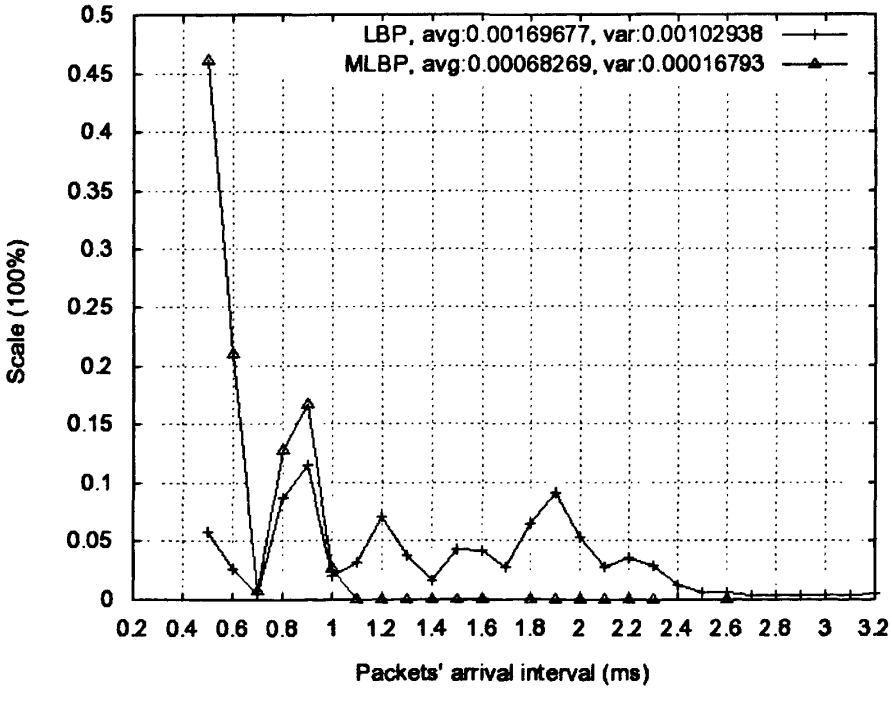


图 6-22 帧间隔对比，8 个 1Mbps 的 UDP 背景流

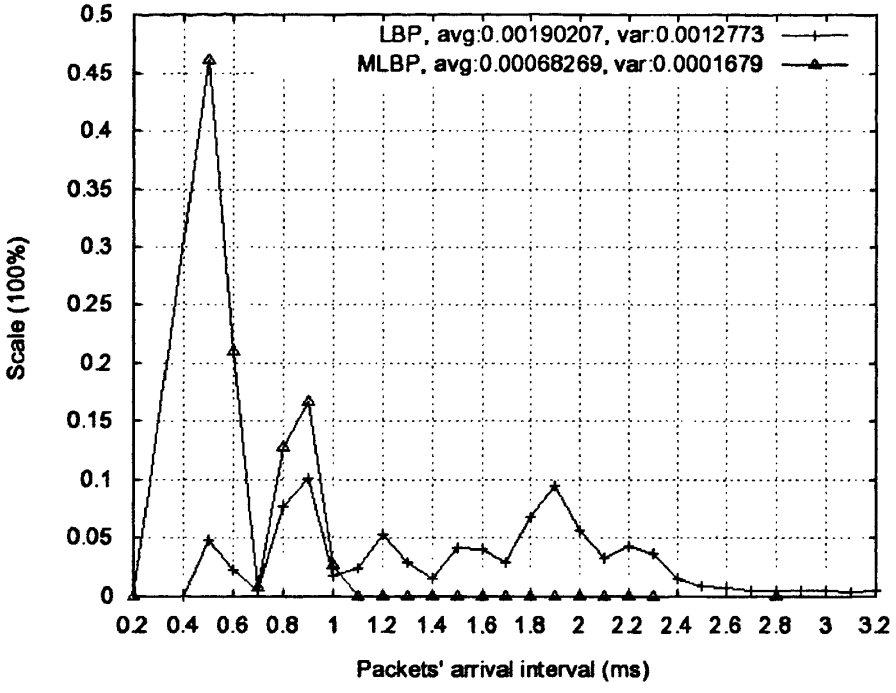


图 6-23　帧间隔对比，10 个 1Mbps 的 UDP 背景流

图 6-24、图 6-25、图 6-26、图 6-27 和图 6-28 反映的是 LBP 协议和 MLBP 协议下，协议所承载的多播业务流与背景 UDP 流的吞吐率情况，观察协议对其他流量的影响有多大，是否兼顾一定的流量间公平性。每幅图分别有四条图线，分别代表 LBP 协议所承载的多播 UDP 流、MLBP 协议所承载的多播 UDP 流、LBP 协议环境下的 UDP 背景流以及 MLBP 协议环境下的 UDP 背景流。四幅图分别反映 2 个、4 个、6 个、8 个和 10 个 UDP 背景流的情况。

从图 6-24 上可以看出，在 2 个 UDP 背景流时，LBP 协议所承载的多播流不断跟 UDP 背景流互相争抢带宽，其流量在 0.8Mbps 到 1.4Mbps 之间波动，每一次一但多播流争抢达到峰值 1.4Mbps 之后，速率马上下滑，又被背景流流量抢回来。而背景流则是在 1Mbps 和 1.2Mbps 之间波动。从这里可以看出，多播流始终无法保持一个稳定的速率传输数据。但另一方面可以看到，观察 MLBP 协议，其所承载的多播流稳定维持在 1.4Mbps 的水平，非常接近源端的发送速率 1.5Mbps，而此时背景流只获得了 0.4Mbps 左右的速率，虽然背景流被降速了，但并没有发生断流的情况，两者的速率都非常稳定，各自以自己所能争抢到的信道带宽进行稳定传输。

从图 6-25 上可以看到，当背景流数增加到 4 个后，LBP 协议下的背景流转为占了上峰，达到了 1.4Mbps 以上的峰值水平，而多播流的速率跌到了 1Mbps 以下徘徊，两者的速率依然是在互相争抢中进行着，但从图上可以看出，LBP 协议下的多播流，其平均吞吐率已经远远低于源端的发送速率 1.5Mbps。但观察 MLBP 协议，其所承载的多播流依然稳定维持在 1.4Mbps，同时背景流也一直稳定在 0.4Mbps。而在图 6-26 上 6 个 UDP 背景流线程的情况也是类似的，LBP 协议下的多播流的吞吐率已经完全跌到了 1Mbps 以下，更多的带宽被背景流所占用。而在 MLBP 协议下的多播流，完全不受背景流的线程数增加的影响，继续保持 1.4Mbps 的速率进行数据传输。

至于图 6-27 和图 6-28 所显示的 8 个和 10 个背景流的情况下，这种协议间的对比就更加明显。随着背景流线程数的增加，背景流的总流量对信道带宽的争抢更加剧烈。从图上可以看出，LBP 协议下 UDP 背景流的吞吐率已经完全在 1.4Mbps 的水平以上，而多播流则只能维持在 0.4Mbps 左右的水平徘徊，而且其争抢能力也逐渐显得软弱，无法以一定的规律和周期性进行信道的带宽的争抢，在 LBP 协议环境下，多播流已经完全显现不出任何信道带宽争抢优势。而 MLBP 协议下，多播流则依然稳定地维持在 1.4Mbps 的水平，背景流也没有因为其线程数的增加而获得更多的信道带宽，以此可以做出判断，从 2 个线程的 UDP 背景流开始，UDP 背景流已经完全占用了除保留给多播流的信道带宽之外的剩余的带宽资源，即 0.4Mbps，所以无论线程数增加多少，其所能获得带宽也并没有得到提升，由此可见，MLBP 协议虽然为多播流预留了带宽资源，但并没有表现出贪婪或者霸道的特性，剩余的带宽资源依然可以被其他背景流所占用。

综上所述，对于 UDP 这种霸道的背景流的冲击，MLBP 协议一样可以非常稳定地抵御住，保证其所承载的业务流平稳顺畅地传输。而且 MLBP 协议并没有表现出贪婪的特性，与其他流量可以和谐相处。

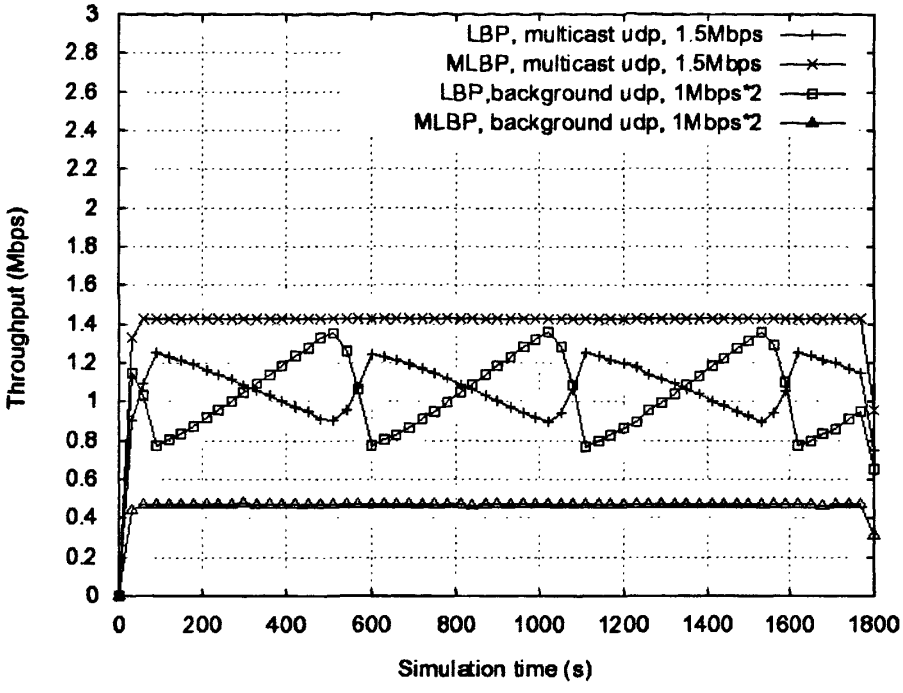


图 6-24 多播流与背景流吞吐率对比，2 个 1Mbps 的 UDP 背景流

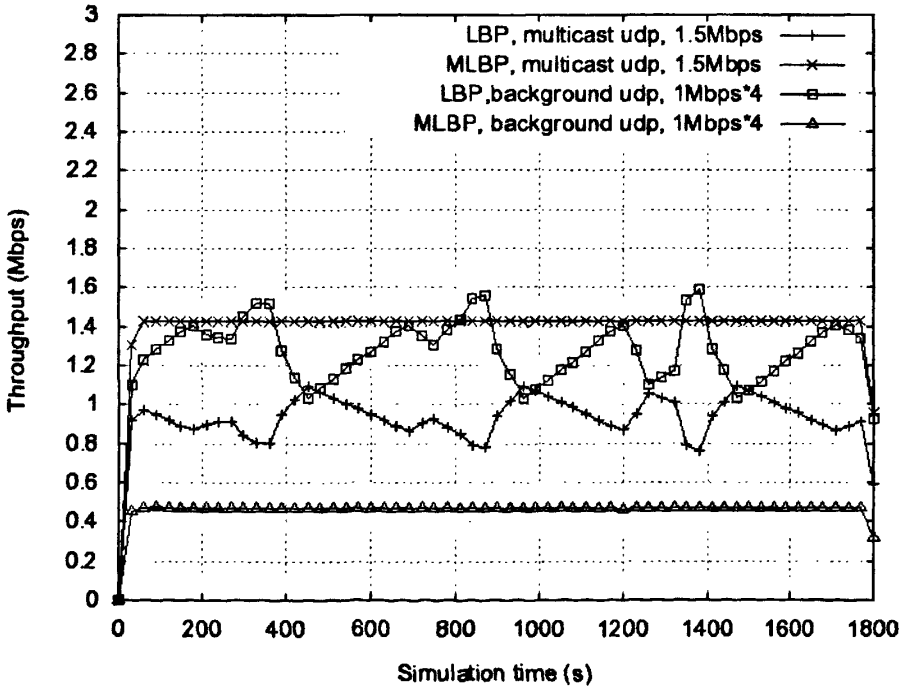


图 6-25 多播流与背景流吞吐率对比，4 个 1Mbps 的 UDP 背景流

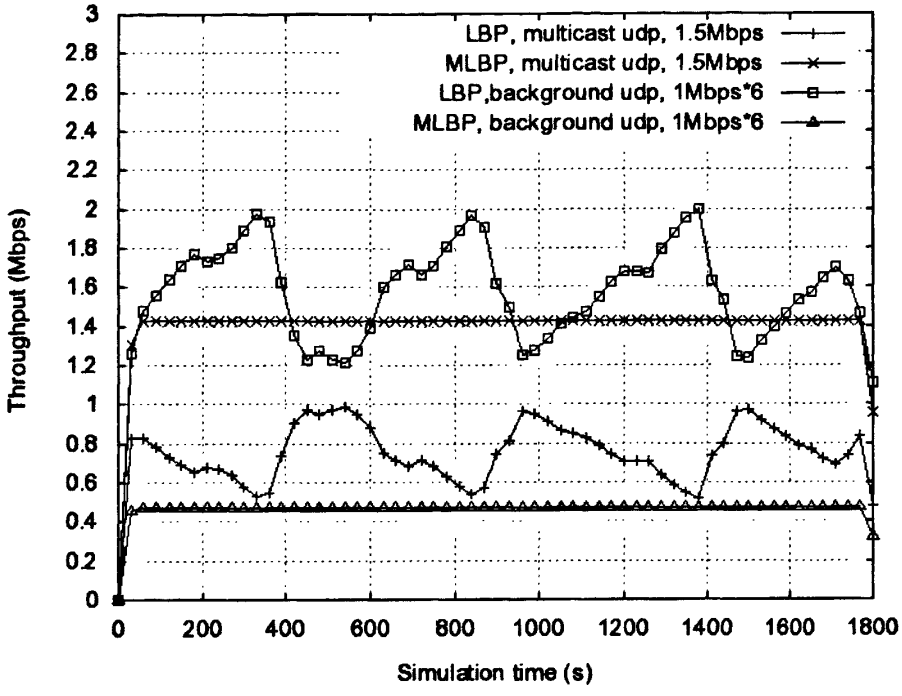


图 6-26 多播流与背景流吞吐率对比，6 个 1Mbps 的 UDP 背景流

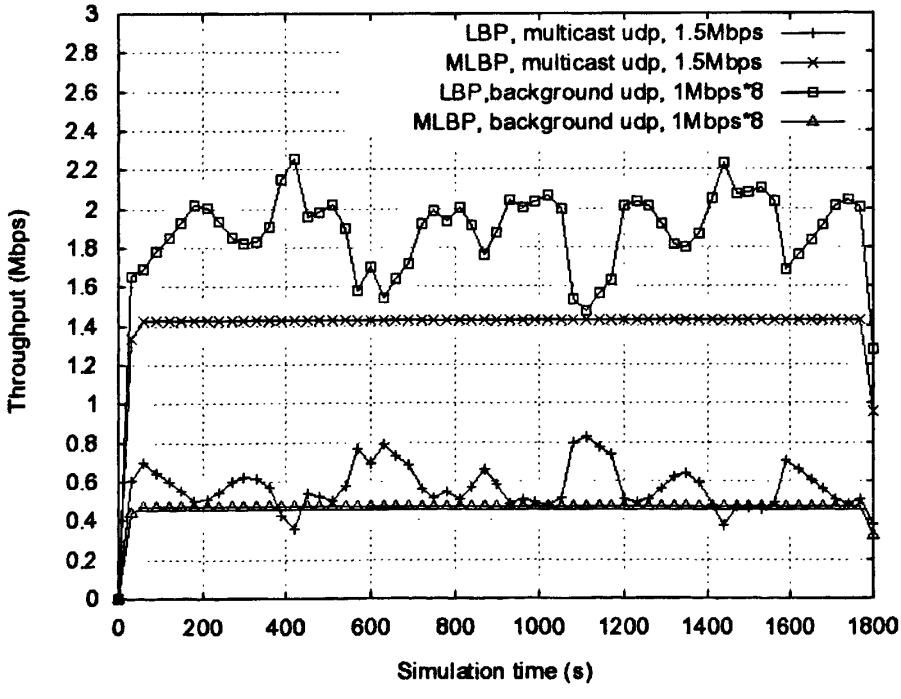


图 6-27 多播流与背景流吞吐率对比，8 个 1Mbps 的 UDP 背景流

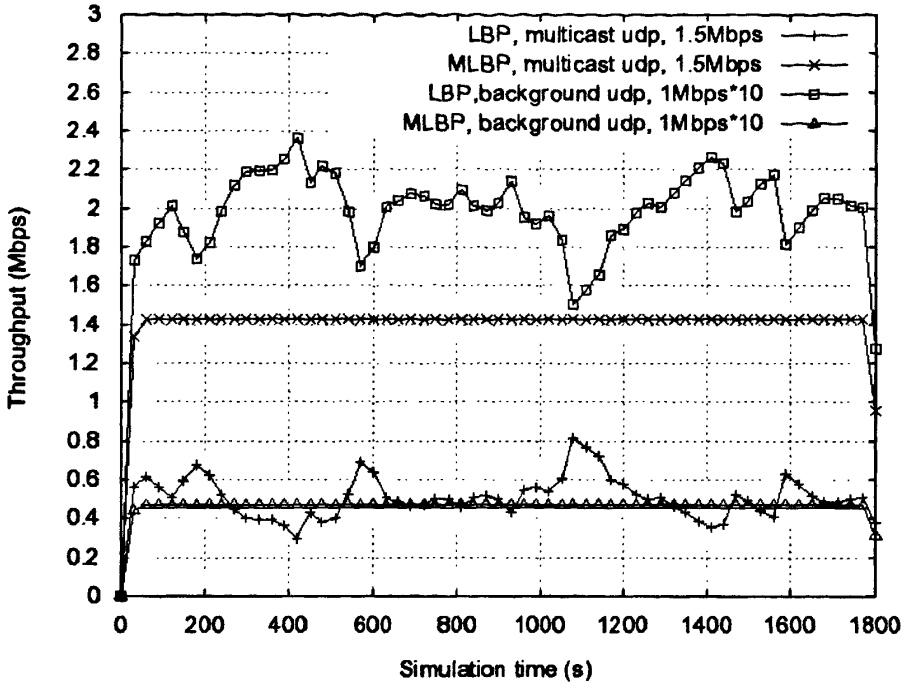


图 6-28 多播流与背景流吞吐率对比，10 个 1Mbps 的 UDP 背景流

6.5.2 TCP 负载背景流测试

相比起 UDP 流，TCP 流是基于连接的，而且带有拥塞控制机制和速率调节能力，能够根据当前信道的繁忙程度而自动调节发送速度。但 TCP 的慢启动算法以及滑动窗口等机制也使其能够在最短时间内重新占用空闲的信道，信道抢占能力也是非常强的。

与前文的 UDP 背景流下的实验结果统计方法类似，也是统计帧间隔的分布情况。同时观察多播流与背景流的吞吐率的变化关系，是否出现背景流被过度压制而造成不公平的情况。

实验结果

图 6-29、图 6-30、图 6-31 和图 6-32 反映的是在 2 个、4 个、6 个和 8 个 TCP 背景流的情况下多播流的帧间隔的分布情况。从图上可以看出，LBP 协议和 MLBP 协议相差并不明显，两协议的图线几乎重合。即使在图 6-32 上反映的增加到 8 个 TCP 背景流的情况下，LBP 协议与 MLBP 协议的性能也差别不大。这种结果只有一个可能，就是 TCP 背景流在与 UDP 多播流进行信道争抢时，并

不能造成拥塞，使得 UDP 多播流在进入缓冲队列时受到影响。即 UDP 多播流的帧总能顺利进入缓冲队列，不受 TCP 背景流的影响，所以 MLBP 协议的 QoS 缓冲队列调度算法并没有起到作用，或者说根本无“用武之地”。而且跟 LBP 协议对比也可以发现，在 LBP 协议下的情况也是类似的，所以这种现象的根源是因为 TCP 协议本身在和 UDP 协议争抢信道过程中就是弱者，UDP 没有自动速率调节机制，总是以恒定速率发送数据帧，并不管信道的拥塞程度如何。反而 TCP 会在拥塞时自动减速避让，在信道空闲时再重新抢占。当 TCP 这种避让机制遇到 UDP 流时，自然就只能一直处于避让状态，无法抢占信道，也就无法对 UDP 流造成影响。

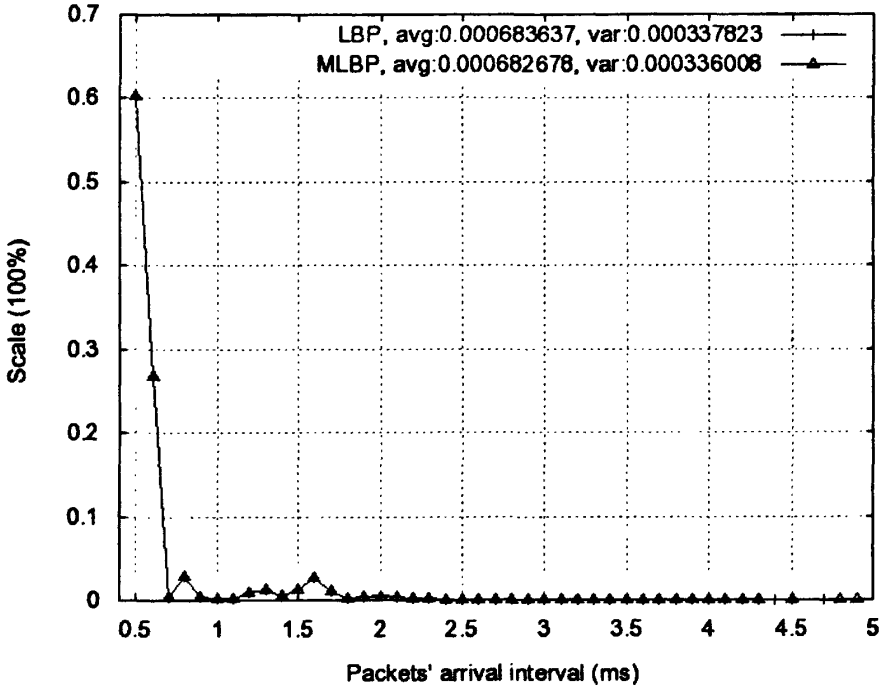


图 6-29 帧间隔对比，2 个 1Mbps 的 TCP 背景流

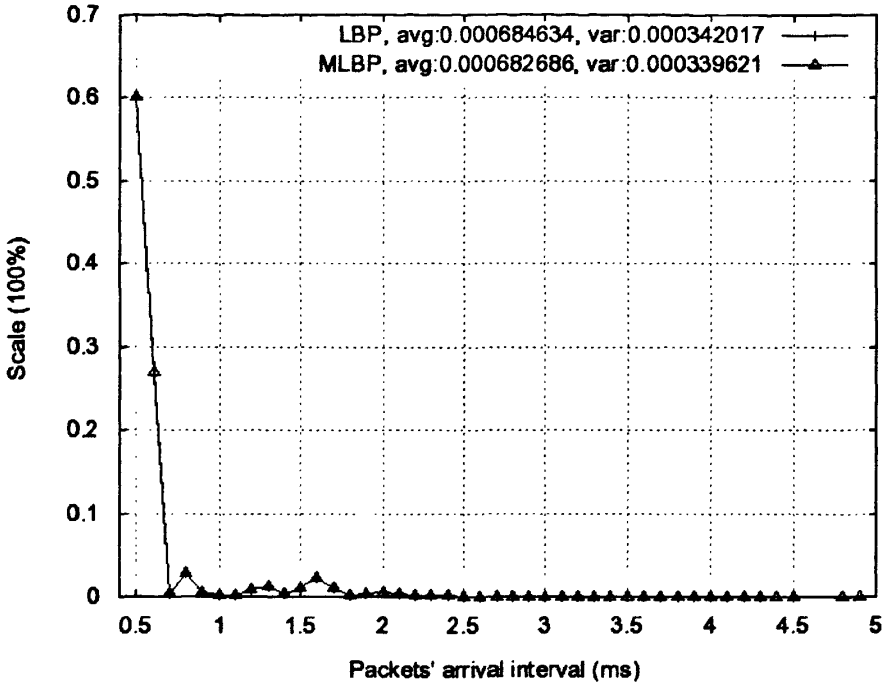


图 6-30 帧间隔对比，4 个 1Mbps 的 TCP 背景流

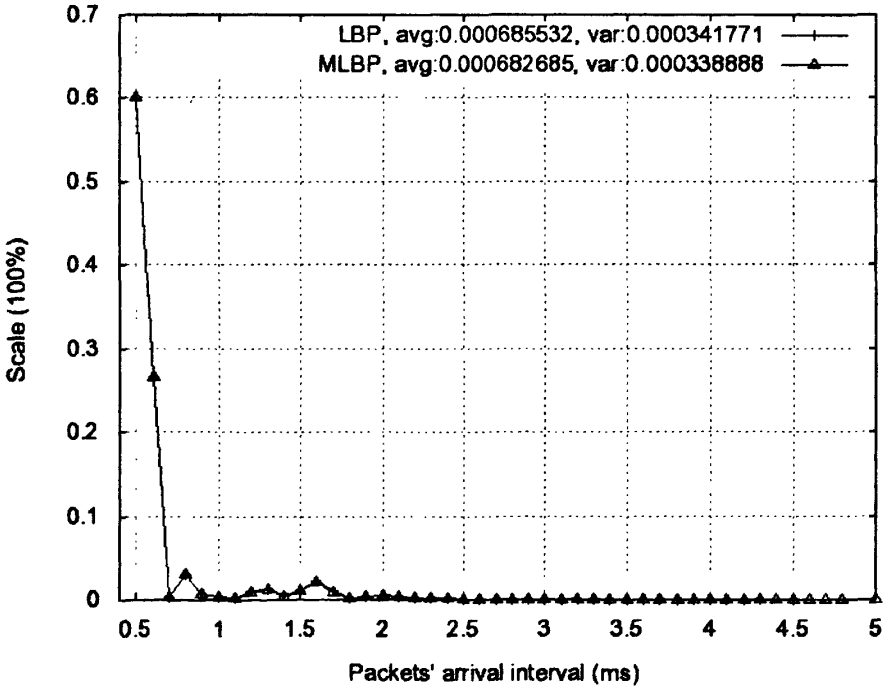


图 6-31 帧间隔对比，6 个 1Mbps 的 TCP 背景流

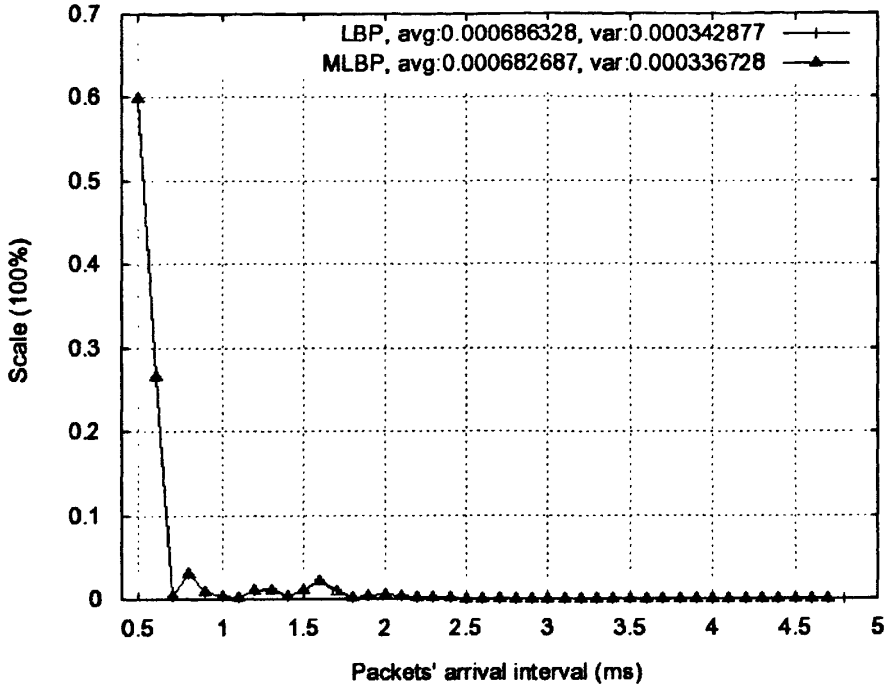


图 6-32 帧间隔对比，8 个 1Mbps 的 TCP 背景流

不但帧间隔没有受 TCP 流的线程数的影响，从吞吐率的角度来看该实验，也可以得到类似的结果。从图 6-33、图 6-34、图 6-35 和图 6-36 看到，在不同 2 个、4 个、6 个和 8 个线程的 TCP 背景流的情况下，无论是 LBP 协议还是 MLBP 协议，背景流对多播流的影响都没有体现出来，多播流在两种协议下一样可以保持 1.4Mbps 的稳定速率传输，两协议的图线基本重合。而究其原因，跟上文的分析的类似，可以归结为带拥塞控制算法的 TCP 流本身就很难跟 UDP 流竞争，导致出现 TCP 流抢不过 UDP 流的这种实验结果。

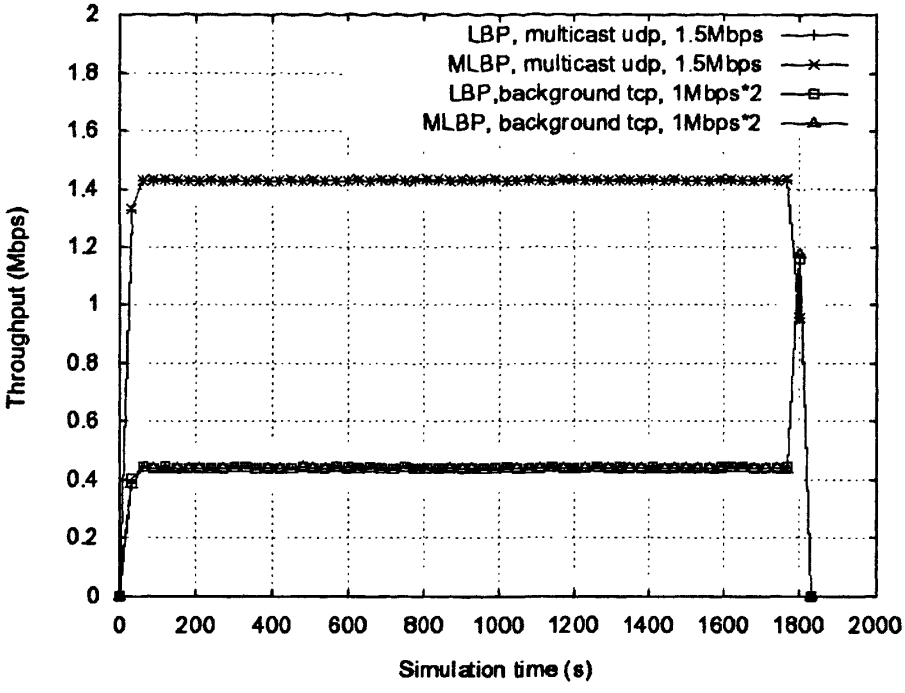


图 6-33 多播流与背景流吞吐率对比，2 个 1Mbps 的 TCP 背景流

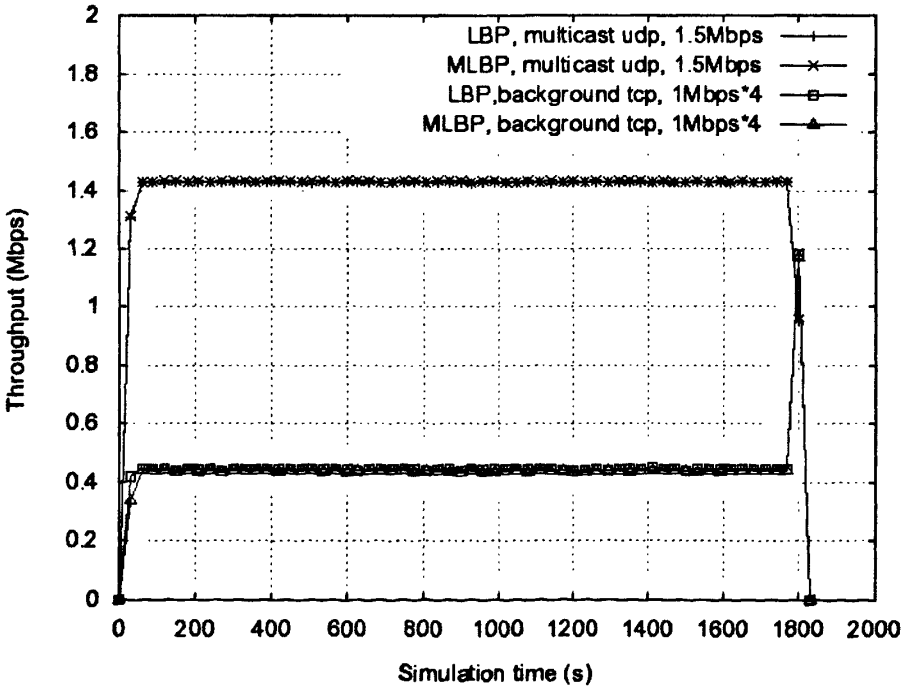


图 6-34 多播流与背景流吞吐率对比，4 个 1Mbps 的 TCP 背景流

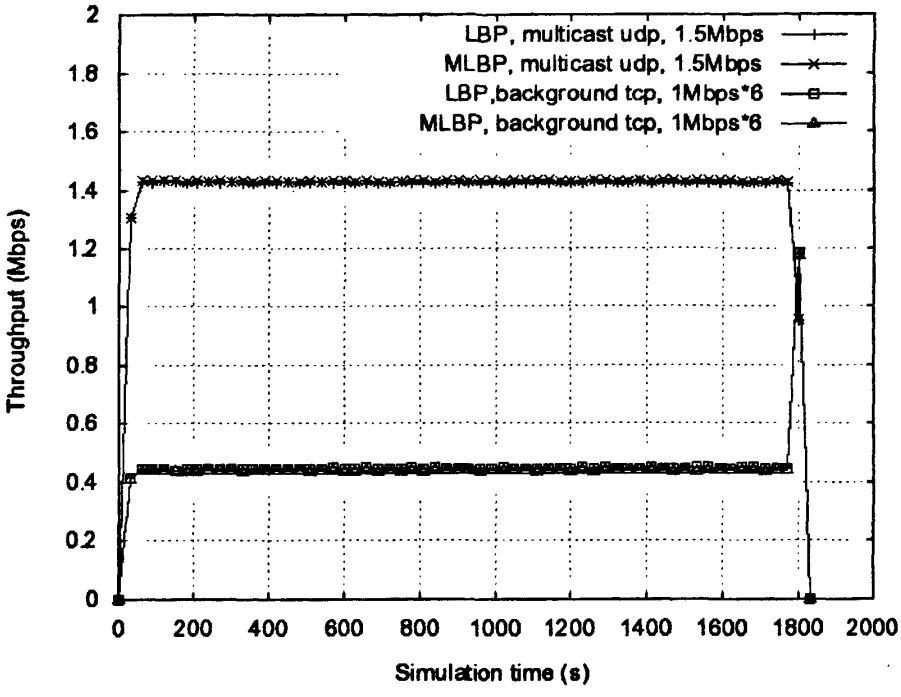


图 6-35　多播流与背景流吞吐率对比，6 个 1Mbps 的 TCP 背景流

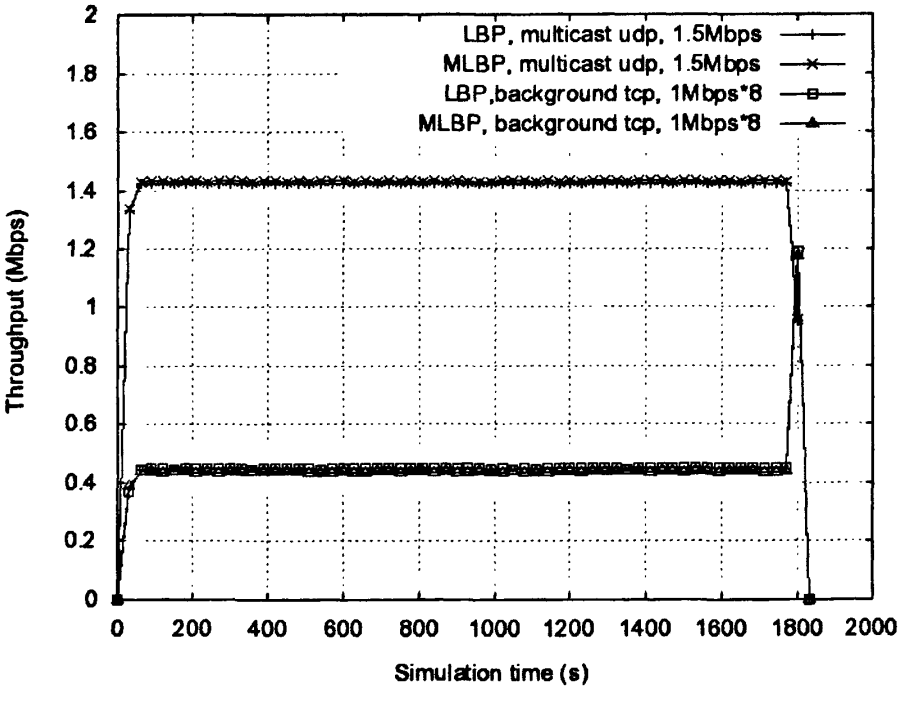


图 6-36　多播流与背景流吞吐率对比，8 个 1Mbps 的 TCP 背景流

6.6 实验小结

本文通过 ns-3 网络仿真实验平台搭建所需的无线网络仿真实验环境，并且采用不同的实验方案和实验参数组合，对比不同情况下 LBP 协议和 MLBP 协议的性能差别，从而获得了大量的实验结果及统计数据。通过信道质量恶化测试实验及结果分析，可以看到 MLBP 协议的基于概率的信道错误重传恢复机制，在信道丢帧率提升，信道传输质量恶化的情况下，能够使多媒体业务流以低时延和低抖动的状态进行传输，并且所牺牲的可靠性并不明显，反而在有效吞吐率上有所提升。通过高负载背景流拥塞测试实验及结果分析，可以得出结论，MLBP 协议的基于优先级的QoS缓冲队列调度算法，能够保证流经AP上的多媒体业务流，以一个恒定的带宽速率传输，不受背景流的负荷量的影响，从而保证了终端用户的多媒体体验质量，提供了较高的多媒体传输稳定性。

第7章 总结与展望

802.11无线局域网技术至今已成为一门成熟并被广泛应用的技术，无线局域网作为最后一公里的网络接入主要技术已是大势所趋。因此，大部分的终端应用都将通过无线局域网进行承载，尤其是越来越受大众欢迎的在线多媒体服务。所以，无线传输多媒体业务是一个重要的课题，而由于多播本身的特点，使其非常适合作为多媒体业务的传输模式。再加上承载更多业务需要有可靠性的支持，所以如何使得无线多播更为可靠，同时更能满足多媒体业务的需求，便成为一个非常热门的研究主题。

在无线可靠多播领域最为经典的协议之一就是LBP协议。但由于LBP协议主要针对的是可靠性问题而提出的，并没有针对多媒体这种具体的应用业务类型进行设计，所以本文在LBP协议的基础上，特别为多媒体业务作出优化改进，提出了一种新的无线可靠多播协议——MLBP。MLBP协议包括基于概率的信道错误重传恢复机制，在无线信道质量恶化时提供抑制抖动和缩短帧间隔的功能；还包括基于优先级的QoS缓冲队列调度算法，在高负载的背景流量对AP造成冲击时，仍能保证多播流可以稳定地传输；另外还包括非常人性化的无线信道资源预约机制，可以为特定的多媒体流预留无线信道资源，实现更高效的无线信道资源利用。

本文在ns-3网络仿真系统上构建无线局域网实验测试平台，通过不同的实验方案，验证了MLBP协议的性能。实验结果证明，在特定的情况下，如信道恶化丢帧率上升和高负载背景流对AP形成冲击时，相比原始的LBP协议，MLBP协议更加能够保持稳定的性能，所承载的多播业务流能够获得更短的帧间隔和更小的抖动，同时可以保持更稳定的吞吐率。

MLBP协议是比以往的无线可靠多播协议更适合传输多媒体的一种新型协议，其应用前景是非常广阔的。如果最终能实现成产品并投入使用，今后在更多的需要无线多播转发多媒体业务的场合中，用户将会获得更稳定质量更高的多媒体服务体验。

参考文献

[1] Kuri, J.; Kasera, S.K.; Reliable multicast in multi-access wireless LANs, IN FOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer a nd Communications Societies. Proceedings. IEEE Volume 2, 21-25 March 1999 Page(s):760 - 767 vol.2

[2] Han-Chieh Chao, S. W. Chang, J. L. Chen, Throughput Improvements Usin g the Random Leader Technique for the Reliable Multicast Wireless LANs, Proceedings of the First International Conference on Networking-Part 1, p. 708-719, July 09-13, 2001

[3] Dongkyun Kimt, Jaewoo Park, Yanghee Choir, An Efficient Reliable Multic asting Protocol for Micro-cellular Wireless Networks, Wireless and Optical Communications (WOC 2002), Track 356-147

[4] Zhao Li; Herfet, T.; HLBP: A Hybrid Leader Based Protocol for MAC La yer Multicast Error Control in Wireless LANs, Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, Nov. 30 2008-Dec. 4 2 008 Page(s): 1 – 6

[5] Diego Dujovne, PThierry Turletti, Multicast in 802.11 WLANs: an experim ental study, Oct. 2006, Proceedings of the 9th ACM international symposiu m on Modeling analysis and simulation of wireless and mobile systems

[6] Lyakhov, A.; Vishnevsky, V.; Yakimov, M.; Multicast QoS Support in IEEE 802.11 WLANs, Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on 8-11 Oct. 2007 Page(s):1 - 3

[7] Gupta, S.K.S.; Shankar, V.; Lalwani, S.; Reliable multicast MAC protocol f or wireless LANs, Communications, 2003. ICC '03. IEEE International Con ference on Volume 1, 11-15 May 2003 Page(s):93 - 97 vol.1

[8] Z.Li and Th. Herfet. "Beacon-driven Leader Based Protocol over a GE Ch

annel for MAC Layer Multicast Error Control", International Journal of Co mmunications, Network and System Science (IJCNS), 2008

[9] Xiaoli Wang; Lan Wang; Yingjie Wang; Yongsheng Zhang; A Reliable and Efficient MAC Layer Multicast Protocol in Wireless LANs, Vehicular Tec hnology Conference, 2009. VTC Spring 2009. IEEE 69th, 26-29 April 200 9 Page(s):1 – 5

[10]Ping Ding, Joanne Holliday, Aslihan Celik, MAC Layer Multicast Protocol to Increase Reliability in WLANs, Ping Ding Joanne, http://citeseerx.ist.psu. edu/viewdoc/summary?doi=10.1.1.58.7990, 2004

[11]Khayam, S.A.; Karande, S.S.; Ilyas, M.U.; Radha, H.; Header Detection to Improve Multimedia Quality Over Wireless Networks, Multimedia, IEEE T ransactions on Volume 9, Issue 2, Feb. 2007 Page(s):377 - 385

[12]Fu, F.; van der Schaar, M.; Noncollaborative Resource Management for Wi reless Multimedia Applications Using Mechanism Design, Multimedia, IEEE Transactions on Volume 9, Issue 4, June 2007 Page(s):851 – 868

[13]Li, Y.; Markopoulou, A.; Apostolopoulos, J.; Bambos, N.; Content-Aware P layout and Packet Scheduling for Video Streaming Over Wireless Links, M ultimedia, IEEE Transactions on Volume 10, Issue 5, Aug. 2008 Page(s): 885 – 895

[14]Lohier, S., Doudane, Y.G. (et al.), Cross-layer loss differentiation algorithms to improve TCP performance in WLANs, Telecommunication Systems, Spr inger US, 15.11.2007, vol. 36, no. 1, pp. 61-72

[15]Sungjoon Choi; Nakjung Choi; Yongho Seok; Taekyoung Kwon; Yanghee Choi; Leader-Based Rate Adaptive Multicasting for Wireless LANs, Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE 26-30 Nov. 2007 Page(s):3656 – 3660

[16]IEEE Computer Society, IEEE Std 802.11-2007, IEEE Standard for Informa tion technology, 12 June 2007

[17]Matthew Gast, 802.11 Wireless Networks—The Definitive Guide, Second E dition, O'Reilly Inc., 2005

[18]Andrew S.Tanenbaum., Computer Networks, Fourth Edition, Pearson Education, 2004

[19]Larry L.Peterson, Bruce S.Davie著，叶新铭，贾波等译，计算机网络－系统方法，第三版，机械工业出版社，2005年1月

[20]ns-3 official website, http://www.nsnam.org/, April, 10th, 2010

[21]ns-3 Reference Manual, ns-3 project, 28, January, 2010

[22]ns-3 Tutorial, ns-3 project, 28, January, 2010

[23]金纯等，IEEE 802.11无线局域网，电子工业出版社，2004年1月

[24]中文维基百科，http://zh.wikipedia.org/zh-cn/IEEE_802.11，2010年1月20日

[25]中国互联网络信息中心CNNIC，《中国互联网络发展状况统计报告》，2010，1：36-37，49

[26]WIKIPEDIA，http://en.wikipedia.org/wiki/Last_kilometer，6 March 2010

[27]中文维基百科，http://zh.wikipedia.org/zh-cn/CSMA/CA#CSMA.2FCA，2010年1月6日

[28]WIKIPEDIA, http://en.wikipedia.org/wiki/Distributed_Coordination_Function, 26 September 2009

[29]Cisco System Inc., Cisco Unified Wireless IP Phone 7920 Design and Deployment Guide, http://www.cisco.com/en/US/docs/voice_ip_comm/cuipph/7920/5_0/english/design/guide/wrlqos.html, October 2005

[30]中文维基百科，http://zh.wikipedia.org/wiki/MPEG-1，2010年4月4日

[31]刘君，新华网 http://news.xinhuanet.com/internet/2008-12/30/content_10579226.htm，2008年12月30日

[32]ugmbbc，cnBeta.com，http://www.cnbeta.com/articles/83992.htm，2009年5月12日

[33]孙燕飚，《第一财经日报》，2010年2月4日

[34]网易手机频道，http://tech.163.com/mobile/07/0110/08/34FB3LFG0011179O.html，2007年1月10日

[35]智渊，新浪科技，http://tech.sina.com.cn/it/2010-01-28/04283811510.shtml，2010年1月28日

[36]广州亚运会官方网站，http://www.gz2010.cn/09/1208/14/5Q1485IU0078000T.html，2009年12月3日

[37]广州亚运会官方网站，http://www.gz2010.cn/09/1030/15/5MSORA5C0078020N.html，2009年10月30日

[38]南方日报，广州亚运会官方网站，http://www.gz2010.cn/10/0408/10/63O9ND1E0078008P.html，2010年04月8日

作者简历

李钊文，2008年至今就读于中山大学信息科学与技术学院计算机系，攻读工学硕士学位。2004年至2008年大学本科期间在中山大学网络工程专业进修，获得工学学士学位。主要研究方向为无线局域网技术，熟悉ISO七层网络体系结构，熟悉802.11标准以及无线局域网技术的实现细节，对无线局域网链路层的协议有深入的研究，曾撰写相关领域的学士学位论文《802.11无线网络MAC层协议改进——动态分段与双分段并发机制》。熟悉ns-3网络仿真系统，是中山大学首先对ns-3系统进行研究并成功应用到实际仿真实验中的研究者之一。从2008年开始接触ns-3系统，至今已经过一年多的使用开发和经验积累，掌握了丰富的ns-3二次开发方法，以及进行网络仿真实验的技巧，曾为中山大学计算机系和电子系的学生进行ns-3系统的入门课程讲授。现已被百度公司录用，将于2010年8月入职，担任百度维护部系统维护工程师职位。

致 谢

由衷感谢我的导师尹冬生老师，本人两年研究生学习生涯中的所有研究工作，都是在他的悉心指导和耐心帮助下完成。本文的研究成果，离不开尹老师提供的充足的研究资源和良好的实验环境，同时也离不开老师的大力支持和不断鼓励。

感谢我的父母一直以来对我的培养和鞭策。同时感谢我的大学宿舍室友廖书正、张健翀、周景荣，感谢他们为我的研究工作所提供的支持和有益的建议。

感谢同实验室的李文俊同学和梁嘉男同学，感谢他们为我解决研究工作上的难题所提供的帮助，以及他们提供的研究资源和研究经验的分享。

李钊文

2010年4月8日