

文章编号:1001-9081(2006)03-0534-03

多重抑制自动重传请求协议的实现与仿真

王绪国, 杨 杰

(武汉理工大学 信息工程学院, 湖北 武汉 430063)

(xuguo-wang@163.com)

摘 要:对采用多重抑制 ARQ 协议的宽带无线接入系统进行研究。实现了一种能为上层提供可控时延、高吞吐量、可控丢包率的链路传输结构。对影响协议算法仿真性能的因素进行了分析讨论。

关键词:自动选择重传; 宽带无线接入; 吞吐量

中图分类号: TP393.07 **文献标识码:** A

Implementation and simulation for multiple rejects ARQ protocol

WANG Xu-guo, YANG Jie

(School of Information Technology, Wuhan University of Technology, Wuhan Hubei 430063, China)

Abstract: Based on the research for a broadband wireless access system with multiple rejects protocol, a new data link transmission scheme was implemented. The simulation results show that this scheme can provide low delay, high throughput and low drop ratio service to the upper layer. And the factors influencing protocol performance were discussed.

Key words: ARQ; broadband wireless access; throughput

0 引言

自动重传请求 (ARQ) 协议是一种在通信中用来保证具有包数据格式媒体信息在链路端到端之间进行正确传递的网络协议。传统 ARQ 有三种典型技术: 停止等待算法、后退 N 步 GBN 算法以及选择重传 SR 算法^[1~3]。选择重传 ARQ 在具有无限大的缓冲区时可以达到传输的理论上限。GBN 及选择重传 ARQ 在有线网络环境下工作得相当好, 但它们并不适合无线系统。这是因为, 高误帧信道以及相对不足的控制带宽将不能满足过多确认帧的传输需求, 而且, 这两种技术中所使用的捎带技术同样不适合于无线信道。总而言之, 由于无线信道质量很差, 需要大量的链路控制带宽用于传输错误帧的确认信息, 导致确认效率很低。在无线数据通信系统中, 需要高确认效率的自动请求重传机制。为此, 3GPP 组织提出了一种多重抑制传输 (Multiple Rejects) ARQ 协议应用于无线数据通信的 RLC (Radio Link Control) 实体^[4]。在多重抑制 ARQ 协议中, 接受方通过使用 Super Fields (SUFI) 来发送状态协议数据单元 (STATUS PDU), 从而报告自身的状态。与标准的 ARQ 协议或改进的 ARQ 协议比较, 此协议采取的是资源消耗与传输效率之间的一种折中策略^[5]。

1 多重抑制 ARQ 的工作原理

标准的 ARQ 协议如后退 N 步 ARQ 或选择性重传 ARQ, 其应答 (ACK 或 NAK) 都是针对每一个接收数据包而言的, 这种方式在无线环境下, 接收端如果针对每一个数据包都回传一个应答包则势必造成大量的无线资源浪费及过大的延迟, 进而造成系统吞吐量下降。而多重抑制 ARQ 协议则基于这样一种方式: 存在一个固定的 ARQ 周期 T_{MR} , 发送方每个周期从发送窗口中发送若干数据 PDU, 对每个数据 PDU 按照网

络层的顺序标号, 同时每个周期收到接收方回传的一个 ACK 包, 每个 ACK 包会采用适当的组织方式表示出此点之前接收方收到的接收不正确包的位置、个数及 ACK 包当前传送位置等信息。而在发送方收到此 ACK 包后采取选择性重传方式将需要重传的数据包在当前位置逐次发送。接收方为了保证正确的向上提交顺序, 从接收窗口中收到此重传数据包后可将缓冲区中当前数据包按 PDU 头中的序号顺序向网络层提交。而发送方在等待 ACK 的过程中进行新一轮数据传输, 这样不断循环从而使数据能有效地从一个对等端传递到另一个对等端。

2 系统设计与实现

2.1 系统描述

本文研究的对象是链路层采用多重抑制 ARQ 的宽带无线接入系统。系统采用三层结构, AP (Access Point) 是用户接入点, 它可以通过网络侧的业务节点接口直接进入 PSTN 的交换机; CPE (Customer Premise Equipment) 是用户终端设备, 每个 CPE 可以接多个固定用户。AP 和 CPE 之间通过无线链路通信, CPE 和它的多个用户之间是有线连接。

该系统要实现的目标有: 在 AP 实现支持多个 CPE 平面 (每个 CPE 平面有自己的接收窗口、发送窗口) 的 ARQ 协议, 在 CPE 实现单个 CPE 平面的 ARQ 协议; 实现由以 AP 为中心的集中带宽控制, 由 AP 智能地、主动地为多个 CPE 分配相互不重叠的发送时间区间; 实现高吞吐量、低时延、低确认开销的可靠数据传输。

2.2 主要技术方案

吸取 3GPP 中绝对偏移确认及 HIPERLAN 中累积确认的思想^[6], 本文提出一种在发送方加入超时定时器, 且将相对偏移和累积确认合二为一的多重抑制 ARQ 机制, 实现高吞吐

收稿日期: 2005-09-22 修订日期: 2005-11-27

作者简介: 王绪国 (1969-), 男, 湖北松滋人, 讲师, 硕士, 主要研究方向: 多媒体网络通信、宽带接入技术; 杨杰 (1961-), 女, 湖北武汉人, 教授, 博士, 主要研究方向: 信号与信息处理。

率、低时延、低确认开销的可靠数据传输,同时屏蔽802.11 MAC层的ACK^[7],降低MAC层ACK造成的带宽浪费和附加的时延开销,保证802.11a无线带宽的高利用率,并避免CSMA/CA机制带来的时延不确定性。下面给出该机制的主要技术方案:

1) 数据发送方和数据接收方通过包含链路管理信息的链路控制PDU建立数据逻辑链路连接,为了保证业务时延在一定范围之内,在链路上采用尽力可靠模式RLC实体传输。

2) 数据发送方的数据链路层从上层获取SDU(服务数据单元),由封装模块封装成数据PDU送入合并模块,将多个数据PDU合并成固定大小的PDU,放入发送窗口缓冲区的空闲存储单元,等待发送。

3) 数据发送方的MAC子层通知数据发送模块可以发送数据PDU以及发送数据PDU的个数,数据发送模块首先根据发送窗口缓冲区中各数据PDU的发送状态确定本次发送哪些数据PDU,然后发送这些数据PDU,并为发送的每个数据PDU启动发送超时定时器,修改该数据PDU的发送状态为正在发送状态。

4) 当数据发送方有发送超时定时器超时,超时处理模块修改相应数据PDU的发送状态为等待发送状态,等待数据发送模块进行重传。

5) 当数据发送方收到包含确认信息的链路控制PDU时,确认信息处理模块首先分析该链路控制PDU包含对哪些数据PDU的确认,并对所有的否定确认进行否定有效性判断,再对每一个有效的确认,包括所有肯定确认和所有有效的否定确认,修改相应数据PDU的发送状态,并取消该数据PDU的发送超时定时器。

6) 当数据接收方收到数据PDU时,数据接收处理模块将接收的数据PDU和检测到丢失的数据PDU的PDU序号和到达时间记录到确认等待队列中,将正确接收的数据PDU放到接收窗口缓冲区的存储单元中,将其接收状态设为已正确接收状态,如果满足移动接收窗口的条件则通知移动接收窗口模块移动接收窗口。

7) 当移动接收窗口模块收到数据接收处理模块的移动接收窗口的通知时,将连续多个已正确接收的数据PDU移出接收窗口缓冲区,提交给解封模块,同时将移出的数据PDU原先占用的存储单元设为空闲,接收状态设为未正确接收状态。

8) 数据接收方的确认信息发送控制模块周期性地判断是否满足发送包含确认信息的链路控制PDU的条件,如果满足条件则通知确认信息发送模块发送包含确认信息的链路控制PDU。

9) 数据接收方的确认信息发送模块收到确认信息发送控制模块的发送确认信息通知后,利用确认等待队列,确定发送包含确认信息的链路控制PDU的FSN字段和确认类型字段,再生成Bitmap/Relative Offsets字段,构造出完整的固定大小的链路控制PDU,从确认等待队列中删除该链路控制PDU确认的数据PDU的记录节点,并发送该链路控制PDU。

PDU格式和定义可参照3GPP标准^[4],这里不作赘述。

2.3 算法前向发送流程

协议前向部分的结构如图1所示。

封装处理 封装模块对协议单元PU进行封装,形成数据PDU,送至合并处理模块。

合并处理 将长度较短的数据包合并成固定大小的PDU,放入数据PDU发送队列缓存。

发送处理 在MAC层信号的触发下,将ARQ发送窗口中的部分未发送PDU发送给MAC层,可以发送的PDU个数由MAC子层的前向速率分配算法和未发送的PDU总数决定。

超时处理 定时搜索ARQ发送窗口中的PDU,如果有已发送的PDU未接受确认的时间超过门限值,则判其发送超时,该PDU需要进行重发。

确认信息处理 收到移动终端发来包含前向ARQ确认信息的链路控制PDU时,提取其中的确认信息,根据确认信息修改前向ARQ发送窗口中的PDU状态,并移动发送窗口。

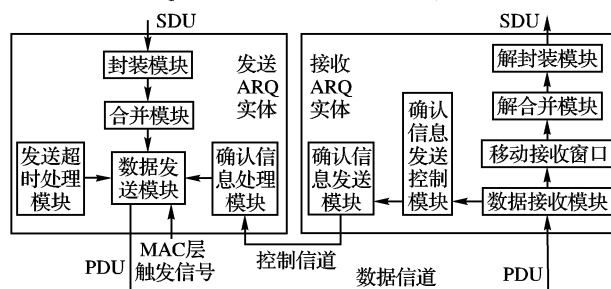


图1 协议的实现结构

对于多重抑制ARQ协议下行数据传输,包含确认信息的链路控制PDU通过反向控制信道逻辑通过映射到反向分组数据物理信道进行发送;对于多重抑制ARQ协议上行数据传输,包含确认信息的链路控制PDU通过公共控制PDU映射到前向分组数据物理信道进行发送,多个移动终端的确认信息,通过基站自适应确认信息发送时机控制算法在公共控制PDU的数据部分统计复用,将多个移动终端对上行数据发送总的确认带宽开销控制在48kbps之内。

3 仿真结果及分析

利用OPNET软件进行仿真,数据PDU的长度为定长,664个比特。控制信道采用(3,1,9)卷积编解码,数据信道则采用TPC(Turbo 乘积码)编解码方式。每个PDU数据所能允许的最大传输时延为120ms,数据信道发送PDU速率分别设为100个/时隙、250个/时隙和300个/时隙。使用到的控制信道和数据信道误帧模型都采用常见的二状态Markov模型以模拟无线衰落的Rayleigh信道,转换矩阵形如 $P = \begin{bmatrix} p & q \\ r & s \end{bmatrix}$ 。仿真了设定2%,5%两种误帧率的控制信道,以及5%,10%,20%和30%这4种误帧率的数据信道,对应使用到的转移矩阵参数如表1所示。

表1 两种信道误帧模型对应各种误帧率的转移矩阵

信道误帧模型	误帧率(%)	p	q	r	s
控制信道	2	0.995	0.005	0.245	0.755
	5	0.990	0.010	0.190	0.810
数据信道	5	0.997	0.003	0.057	0.943
	10	0.995	0.005	0.045	0.955
	20	0.988	0.013	0.050	0.950
	30	0.985	0.015	0.035	0.965

仿真1 链路平均时延性能

结果如图2所示,可以看出,一般情况下(数据误帧10%以下),协议时延控制在55ms以下;在恶劣情况下(数据误帧

20%以上),协议时延控制在 90ms 以内;而控制误帧对协议时延影响一般在 5ms 以下。从仿真结果来看,数据误帧率越大,意味着数据包的丢失就越多,因此更多的数据需要重传才能接收,链路重发时延必然增加;同时,数据误帧率越大,系统确认效率越低,链路重发等待时延也会增大。由于采用尽力可靠模式,平均时延对控制误帧并不敏感。

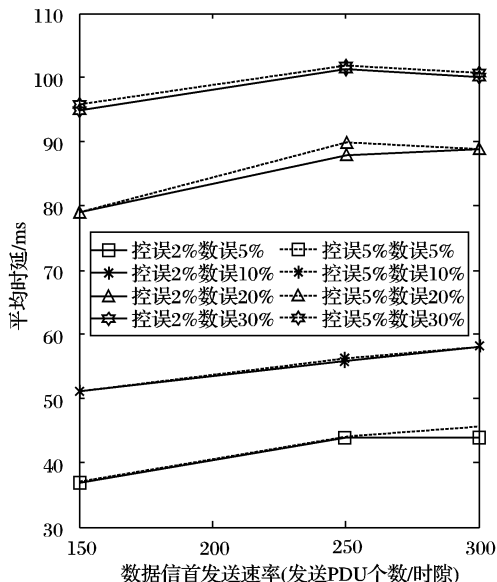


图2 在不同误帧下链路平均时延性能

仿真2 数据 PDU 丢包率性能

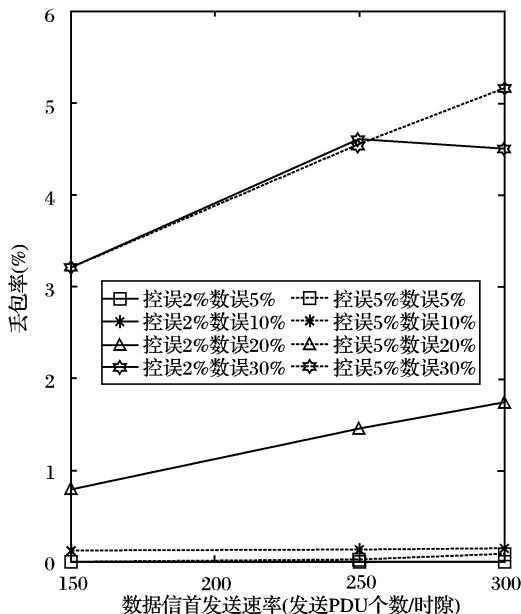


图3 在不同误帧下丢包率性能

结果如图 3 所示,可以看出,一般情况下(数据误帧 10%以下),协议丢包率低于 0.2%;在恶劣情况下(数据误帧 20%以上),协议丢包率控制在 5% 以下;而控制误帧对协议丢包率影响一般在 0.2% 以下。影响数据丢包率的主要因素包括数据误帧、发送速率和控制误帧,从图 3 分析,数据误帧增加会导致同一个数据 PDU 在信道中连续丢失的概率增加,从而使得数据因为超出发送次数而被丢弃,同时系统重传数据量增加,导致数据因超时而被丢弃的概率增大。同样,发送速率增大,系统负担增加,也导致丢包率增大。在本文的尽力可靠 RLC 实体仿真环境下,数据接收端每次确认完数据 PDU 后会

删掉其确认标识,即数据接收端认为已经完成对其的确认,而不管数据发送端是否正确收到这个控制 PDU,因此,控制误帧对丢包率的影响较小。

仿真3 归一化吞吐量性能

结果如图 4 所示,可以看出,一般情况下(数据误帧 10%以下),协议归一化吞吐量在 85% 以上;在恶劣情况下(数据误帧 20% 以上),协议归一化吞吐量仍然基本可以达到 50%;而控制误帧对协议归一化吞吐量影响在 5% 以下。

注意到每组具有相同数据 PDU 误帧率的曲线都随着发送速率的保持上升,控制 PDU 误帧对系统吞吐量的影响不断减小。分析如下:当发送速率仅为每时隙 150 个数据 PDU 时,每丢失一个控制 PDU,就相当于重传一个时隙的所有数据 PDU。因为数据 PDU 误帧率的存在,控制 PDU 误帧率的影响会相对减小,当数据 PDU 误帧率为 10% 时,如果数据 PDU 和控制 PDU 误帧分布均匀,随着发送速率的不断增大,特别是当发送速率达到或超过每时隙 250 个数据 PDU 时,接收端每时隙所发送的控制 PDU 已经不能全部确认一时隙内发送的所有数据 PDU,所以此时控制 PDU 的丢失对系统吞吐量的相对影响就变小了。可以看到,在发送速率每时隙 250 个数据 PDU,数据 PDU 误帧率为 30% 时,数据 PDU 的重传(包含误帧重传和非必要重传)已经将近 50%,此时,控制 PDU 误帧对系统吞吐量的影响就已经微乎其微了。

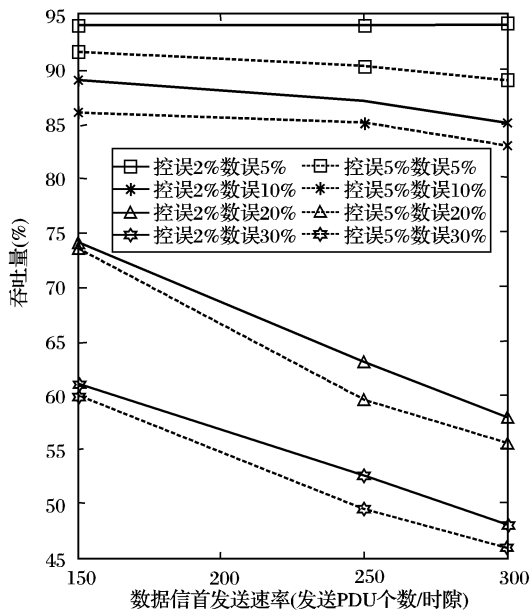


图4 在不同误帧下链路吞吐量性能

4 结语

ARQ 协议通过确认重传实现链路可靠性,3GPP 标准中提出的多重抑制 ARQ 协议可在无线信道条件下有效提高传输效率。基于此,本文提出了一种实现于宽带无线接入系统的完整技术方案,对此方案运用于尽力可靠模式 RLC 实体进行了仿真,该方案特点是对控制 PDU 的丢失不敏感,拥有较小的确认开销,较好的吞吐率性能,减少时延,更加适应未来的宽带非对称网络。

参考文献:

- [1] WELDON Jr EJ. An improved selective-repeat ARQ strategy[J]. IEEE Transactions on Communications, 1982, 30(3): 480-486.

为了验证上述估计算法的正确性,本文做了大量的 NS2^[8]仿真实验。仿真结构如图2。

上行方向为无线节点 WS 通过 AP 发送 TCP 流到固定节点 S,下行方向刚好相反。无线部分采用 IEEE802.11b 协议,使用 DCF 模式工作,最大最小竞争窗口为默认值 ($CW_{min} = 31, CW_{max} = 1023$),采用 short preamble,不使用 RTS/CTS。应用层使用固定包大小为 1000 字节的 CBR 流,速率为 11Mbps。TCP 接收窗口预设置为 42 packets,仿真时长为 100s。定义公平性指数 (Fairness Index) 为 $FI = (\sum_{i=1}^n x_i)^2 / n \sum_{i=1}^n x_i^2$,其中 x_i 为第 i 流的流量, n 为总的流数。 FI 越接近 1,各个流越公平。当 $FI = 1$,各个流的流量都相等,即实现了公平性。图3就是随着 AP 缓存区以及下行行流数目 N 的变化, FI 的变化情况。可见,AP 缓存区对公平性的影响确实较大,另外随着 N 的增大,为达到 $FI = 1$ 所需的 AP 缓存区也在增大。图4显示了当 $FI = 1$,所需的 AP 缓存区大小 B 与下行流数目 N 的关系图。图中 real 表示多次实际仿真得到的 B 与 N 的关系;rough 表示文献[5]中所提及的初步估计;computed 为本文提出的估计方法,即(7)式。从图中可以看出,根据本文提出的估计算法能较精确地反映实际测试结果。 N 越大,初步估计法与实际的误差也越大。

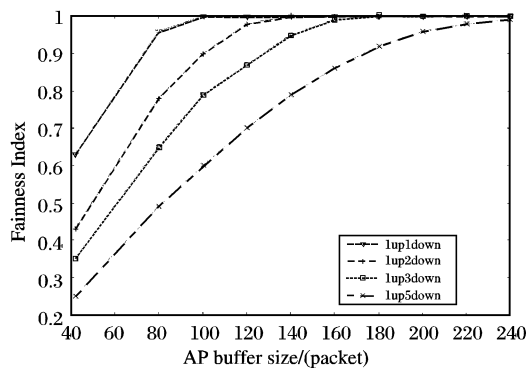


图3 不同的 N 随着 AP 缓存区变化时的 FI 值

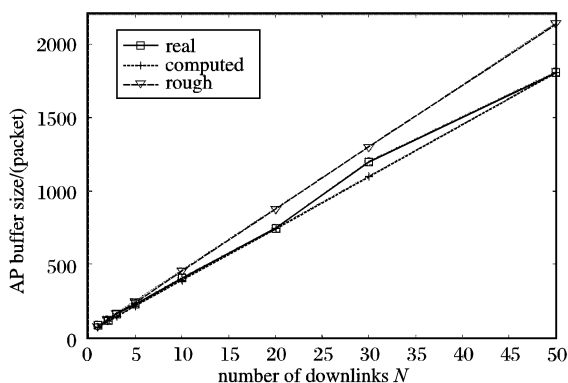


图4 $FI = 1$ 时不同所需的 B 对比

4 结语

本文针对无线局域网上 AP 缓存区对 TCP 公平性的影响进行了研究,并利用平均窗口的概念提出实现 TCP 上下行流公平时所需的 AP 缓存区大小与下行流数目的关系式。实验仿真结果表明,本文提出的估计算法能较准确地反映实际所需 AP 缓存区大小。同时,公式还能确定不同缓存区对 TCP 流公平性的影响程度,从而有助于更好地解决无线局域网中 TCP 流公平性问题。

但是 AP 缓存区不能随意设置,太大则加重队列时延,太小则影响到流的公平性,带来丢包、抖动以及总流量下降等问题。

参考文献:

- [1] LU S, BHARGHAVAN V, SRIKANT R. Fair scheduling in wireless packet networks [A]. ACM SIGCOMM'97 [C]. Cannes, France, 1997.
- [2] KWON Y, FANG Y, LATCHMAN H. A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs [A]. IEEE INFOCOM 2003 [C], 2003.
- [3] VAIDYA NH, BAHL P, GUPTA S. Distributed fair scheduling in a wireless LAN [A]. Mobile Computing and Networking [C], 2000. 167 - 168.
- [4] DENG J, VARSHNEY PK, HASS ZJ. A New Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function [A]. Proceedings of Communication Networks and Distributed Systems Modeling and Simulation (CNDST04) [C]. San Diego, CA, USA, 2004.
- [5] PILOSOF S, RAMJEE R, RAZ D, et al. Understanding TCP fairness over Wireless LAN [A]. IEEE INFOCOM 2003 [C], 2003.
- [6] BOTTIGLIENGO M, CASETTI C, CHIASSERINI C-F, et al. Smart Traffic Scheduling in 802.11 WLANs with Access Point [A]. IEEE VTC 2003-Fall [C]. Orlando, FL, USA, 2003.
- [7] WAHARTE S, XIAO J, BOUTABA R. Overlay Wireless Sensor Networks for Application-Adaptive Scheduling in WLAN [A]. Proceedings of the 7th IEEE International Conference on High Speed Networks and Multimedia Communications [C], 2004.
- [8] FALL K, VARADHAN K. ns Notes and documentation [R]. VINT Project, UC-Berkeley and LBNL, 1997.
- [9] KIM J. BLACK: A New Active Queue Management Scheme Using Arrival Rate for Reducing Queue Length Variation [EB/OL]. <http://cosmos.kaist.ac.kr/~hdyim/ez2000/system/db/cs540upload/upload/38/20023146-JonghwanKim.doc>, 2005.
- [10] MORRIS R. Scalable TCP Congestion Control [A]. IEEE INFOCOM 2000 [C], 2000.
- [11] TAN G, GUTTAG J. Time-based Fairness Improve performance in Multi-rate WLANs [A]. USENIX 2004 [C], 2004. 269 - 282.
- [12] HEUSSE M, ROUSSEAU F, BERGER-SABBATEL J, et al. Performance Anomaly of 802.11b [A]. IEEE INFOCOM 2003 [C], 2003.

(上接第 536 页)

- [2] CHANG Y, LEUNG C. On Weldon's ARQ strategy [J]. IEEE Transactions on Communications, 1984, 32(3): 297 - 300.
- [3] CL CH. The throughput efficiency of go-back-n ARQ scheme under Markov and related error structures [J]. IEEE Transactions on Communications, 1988, 36(2): 231 - 234.
- [4] 3G TS 25.322 version 3.0, 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, RLC Protocol Specification [S], 1999.
- [5] 杨涛, 谢剑英. 一种基于自适应 MR-ARQ 协议性能的研究 [J]. 计算机工程与应用, 2003, 39(26): 179 - 181, 214.
- [6] TS101761-2 Ver 1.1.1, Broadband Radio Access Networks (BRAN). HiperLAN Type2. Data link control (DLC) layer Part2. Radio link control (PLC) sublayer [S], 1999.
- [7] Institute of Electrical and Electronics Engineers. 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications [S], 1999.