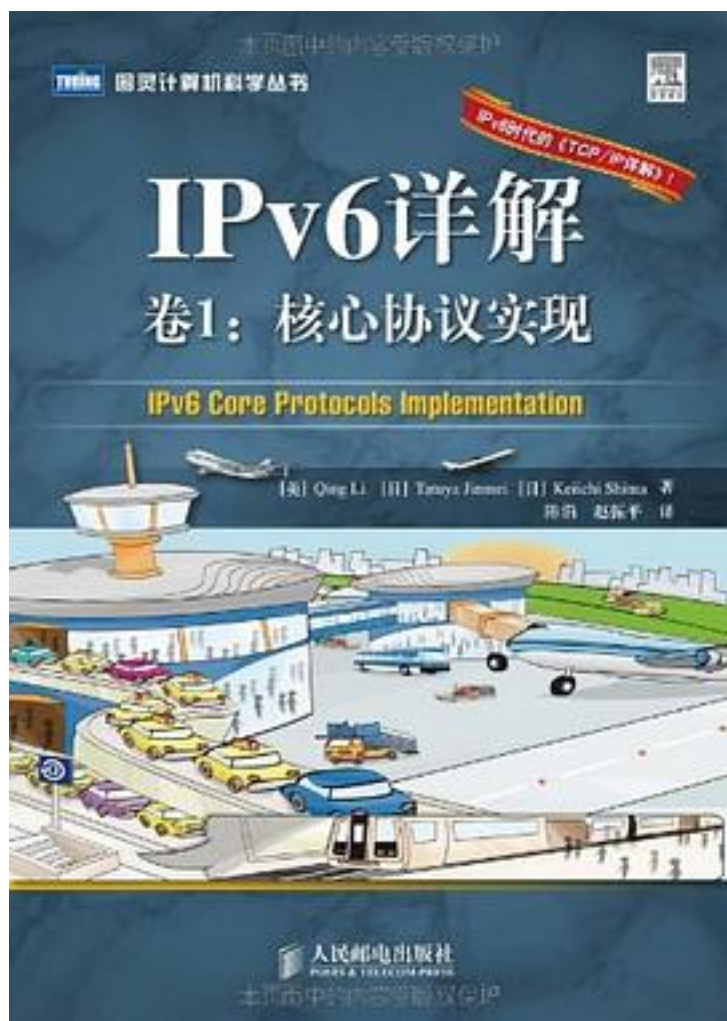


IPv6详解（卷1）



[IPv6详解（卷1）_下载链接1](#)

著者:

出版者:人民邮电出版社

出版时间:2009-1

装帧:平装

isbn:9787115189509

IPv6的时代即将到来！

本书由开源的IPv6标准参考实现KAME的核心开发人员撰写，沿袭了被奉为经典的Richard Stevens《TCP/IP详解》的写作方式和风格，覆盖了IPv6技术的全部内容，是毋庸置疑的IPv6权威参考书。书中详尽剖析了IPv6协议及其实现的技术细节，逐行诠释了KAME每一行代码的作用，并结合阐述了弥足珍贵的设计体会，对网络研究、设计和开发人员都有极高的参考价值。

全书分为两卷，第一卷主要介绍核心协议的实现，第二卷主要介绍高级协议的实现。本书适合网络设计和开发人员阅读，对于下一代网络产品研发人员尤其具有参考价值。

作者介绍:

Qing Li Blue

Coat系统公司资深架构师，负责领导下一代支持IPv6的安全代理应用系统的设计和开发工作。他曾在风河系统公司工作8年，是风河嵌入式IPv6产品的首席架构师。他拥有多项美国专利，并著有Real-Time Concepts for Embedded Systems等畅销书。他还是FreeBSD操作系统项目活跃的开发者的。

Tatuya Jinmei（神明达哉）

东芝公司研究与开发中心的科学家。KAME项目核心开发人员。2003年在日本庆应义塾大学获得博士学位。

Keiichi Shima（岛庆一） 日本Internet

Initiative公司的资深研究人员。他的研究领域是IPv6和IPv6移动性。KAME项目核心开发人员，开发了移动IPv6/NEMO基本支持协议栈。现在正致力于BSD操作系统中新的移动栈（SHISA栈）的研究。

目录: 第1章 绪论

1.1 引言

1.2 ipv6和kame的简史

1.3 kame发行版概述

1.3.1 源代码树结构

1.3.2 构建过程

1.4 bsd网络实现概述

1.5 源代码描述

1.5.1 排版约定

1.5.2 源代码描述示例

1.5.3 预处理器变量

1.5.4 网络设备与体系结构假设

1.6 mbuf与ipv6

1.6.1 常见的mbuf操纵宏和函数

1.6.2 mbuf标签

1.6.3 ipv6的mbuf要求

1.6.4 诊断mbuf链

第2章 ipv6编址体系结构

2.1 引言

2.2 ipv6地址

2.3 ipv6地址的文本表示

2.4 地址范围

2.4.1 范围区域

2.4.2 区域索引

2.4.3 范围受限地址的文本表示

- 2.4.4 单播站点本地地址的废弃
- 2.5 ipv6地址格式
 - 2.5.1 接口标识符的产生
 - 2.5.2 有关地址格式的注解
 - 2.5.3 多播地址格式
- 2.6 节点地址要求
- 2.7 ipv6地址空间管理
- 2.8 代码介绍
 - 2.8.1 ipv6地址结构：in6_addr{}与sockaddr_in6{}
 - 2.8.2 宏与变量
- 2.9 对范围区域的处理
 - 2.9.1 范围区域的初始化
 - 2.9.2 范围区域id
 - 2.9.3 地址结构中的区域id
 - 2.9.4 与范围相关的实用函数
- 2.10 接口地址结构
 - 2.10.1 ifaddr{}和in6_ifaddr{}结构
 - 2.10.2 in6_ifreq{}和in6_aliasreq{}结构
 - 2.10.3 多播地址结构
- 2.11 ipv6前缀结构
- 2.12 地址操纵程序概述
- 2.13 ipv6的接口初始化
 - 2.13.1 in6_if_up()函数
 - 2.13.2 in6_ifattach()函数
 - 2.13.3 in6_ifattach_loopback{}函数
 - 2.13.4 in6_ifattach_linklocal()函数
 - 2.13.5 get_ifid()函数
 - 2.13.6 get_hw_ifid()函数
 - 2.13.7 get_rand_ifid()函数
 - 2.13.8 in6if_do_dad()函数
- 2.14 ipv6接口地址配置
 - 2.14.1 in6_control()函数
 - 2.14.2 in6_update_ifa()函数
 - 2.14.3 in6_joingroup()和in6_leavegroup()函数
 - 2.14.4 in6_addmulti()和in6_delmulti()函数
 - 2.14.5 in6_ifinit()函数
 - 2.14.6 in6_ifaddloop()和in6_ifloop_request()函数
- 2.15 删除ipv6地址
 - 2.15.1 in6_purgeaddr()函数
 - 2.15.2 in6_ifremloop()函数
 - 2.15.3 in6_unlink_ifa()函数
- 2.16 用地址配置工具进行的操作

第3章 ipv6

- 3.1 引言
- 3.2 ipv6首部格式
- 3.3 ipv6扩展首部
 - 3.3.1 扩展首部的顺序
 - 3.3.2 逐跳选项首部
 - 3.3.3 目的选项首部
 - 3.3.4 路由首部
 - 3.3.5 片首部
 - 3.3.6 ipv6选项
- 3.4 源地址选择
 - 3.4.1 默认地址选择

- 3.4.2 源地址选择
- 3.4.3 目的地址选择
- 3.5 代码介绍
 - 3.5.1 统计数据
 - 3.5.2 首部结构
 - 3.5.3 ip6protosw{}结构
- 3.6 mbuf中的ipv6分组地址信息
 - 3.6.1 ip6_setdstifaddr()函数
 - 3.6.2 ip6_getdstifaddr()函数
 - 3.6.3 ip6_setpktaddrs()函数
 - 3.6.4 ip6_getpktaddrs()函数
- 3.7 输入处理: ip6_input()函数
- 3.8 逐跳选项首部的处理: ip6_hopopts_input()函数
 - 3.8.1 对每个选项进行处理: ip6_process_hopopts()函数
 - 3.8.2 处理未知选项: ip6_unknown_opt()函数
- 3.9 对目的选项首部的处理: dest6_input()函数
- 3.10 已分片分组的重装
 - 3.10.1 用于分组重装的结构
 - 3.10.2 frag6_input()函数
- 3.11 对路由首部的处理: route6_input()函数
- 3.12 转发: ip6_forward()函数
- 3.13 输出处理
 - 3.13.1 源地址选择: in6_selectsrc()函数
 - 3.13.2 路由选择: ip6_selectroute()函数
 - 3.13.3 ip6_output()函数
 - 3.13.4 构建扩展首部: ip6_copyexthdr()函数
 - 3.13.5 分割首部: ip6_splithdr()函数
 - 3.13.6 插入特大净荷选项: ip6_insert_jumboopt()函数
 - 3.13.7 分片: ip6_insertfraghdr()函数
 - 3.13.8 路径mtu判断: ip6_getpmtu()函数
 - 3.13.9 多播环回: ip6_mloopback()函数
- 第4章 icmpv6
 - 4.1 引言
 - 4.2 icmpv6报文
 - 4.2.1 目的不可达报文
 - 4.2.2 分组太长报文
 - 4.2.3 超时报文
 - 4.2.4 参数问题报文
 - 4.2.5 回送请求报文
 - 4.2.6 回送应答报文
 - 4.2.7 icmpv6报文处理规则
 - 4.3 pmtu发现机制
 - 4.4 节点信息查询
 - 4.4.1 节点信息报文的格式
 - 4.4.2 noop查询
 - 4.4.3 支持的qtype查询
 - 4.4.4 节点名查询
 - 4.4.5 节点地址查询
 - 4.4.6 ipv4地址查询
 - 4.5 代码介绍
 - 4.5.1 统计数据
 - 4.5.2 icmpv6首部
 - 4.6 icmpv6输入处理
 - 4.6.1 icmp6_input()函数

- 4.6.2 错误通知：icmp6_notify_error()函数
- 4.7 pmtu发现的实现
- 4.8 icmpv6输出处理
 - 4.8.1 发送错误：icmp6_error()函数
 - 4.8.2 错误速率限制：icmp6_ratelimit()函数
 - 4.8.3 icmp6_reflect()函数
- 4.9 节点信息查询的实现
 - 4.9.1 类型和变量
 - 4.9.2 ping6命令：发送查询
 - 4.9.3 ping6命令：接收应答
 - 4.9.4 ping6命令：打印支持的qtype
 - 4.9.5 ping6命令：打印节点地址
 - 4.9.6 查询处理：ni6_input()函数
 - 4.9.7 节点名的操作
 - 4.9.8 创建节点地址应答：ni6_store_addrs()函数
- 4.10 节点信息操作

第5章 邻居发现和无状态地址自动配置

- 5.1 引言
- 5.2 邻居发现协议概述
- 5.3 无状态地址自动配置概述
- 5.4 邻居发现协议报文
- 5.5 邻居发现协议报文的交换示例
- 5.6 邻居发现协议分组类型及格式
 - 5.6.1 路由器请求报文
 - 5.6.2 路由器广告报文
 - 5.6.3 邻居请求报文
 - 5.6.4 邻居广告报文
 - 5.6.5 重定向报文
- 5.7 邻居发现选项类型及格式
 - 5.7.1 链路层地址选项
 - 5.7.2 前缀信息选项
 - 5.7.3 重定向首部选项
 - 5.7.4 mtu选项
 - 5.7.5 路由信息选项
- 5.8 下一跳判定和地址解析
- 5.9 邻居不可达检测算法
- 5.10 无状态地址自动配置
 - 5.10.1 地址的构成和地址状态
 - 5.10.2 重复地址检测算法
 - 5.10.3 处理路由器广告
 - 5.10.4 隐私扩展
- 5.11 路由器特有的操作
 - 5.11.1 发送未经请求的路由器广告
 - 5.11.2 处理路由器请求
 - 5.11.3 处理路由器广告
- 5.12 主机特有的操作
 - 5.12.1 发送路由器请求
 - 5.12.2 处理路由器广告
 - 5.12.3 默认路由器选项
- 5.13 代码介绍
 - 5.13.1 邻居发现报文定义
 - 5.13.2 邻居缓存：llinfo_nd6{}结构
 - 5.13.3 操作变量：nd_ifinfo{}结构
 - 5.13.4 默认路由器：nd_defrouter{}结构

- 5.13.5 前缀：nd_prefix{}结构
- 5.13.6 前缀控制：nd_prefixctl{}结构
- 5.13.7 邻居发现报文选项：nd_opts{}结构
- 5.13.8 dad队列条目：dadq{}结构
- 5.13.9 ipv6地址：in6_ifaddr{}结构
- 5.13.10 目的地缓存
- 5.13.11 操作常量
- 5.14 初始化函数
 - 5.14.1 nd6_init()函数
 - 5.14.2 nd6_ifattach()函数
- 5.15 邻居缓存管理函数
 - 5.15.1 nd6_rtrequest()函数
 - 5.15.2 nd6_cache_lladdr()函数
 - 5.15.3 nd6_lookup()函数
 - 5.15.4 nd6_free()函数
 - 5.15.5 nd6_timer函数
- 5.16 邻居发现协议报文处理函数
 - 5.16.1 nd6_ns_output()函数
 - 5.16.2 nd6_ns_input()函数
 - 5.16.3 nd6_na_input()函数
 - 5.16.4 nd6_na_output()函数
 - 5.16.5 nd6_rs_input()函数
 - 5.16.6 nd6_ra_input()函数
 - 5.16.7 icmp6_redirect_input()函数
 - 5.16.8 icmp6_redirect_output()函数
- 5.17 邻居发现协议报文选项处理函数
 - 5.17.1 nd6_option_init()函数
 - 5.17.2 nd6_option()函数
 - 5.17.3 nd6_options()函数
- 5.18 默认路由器管理函数
 - 5.18.1 defrouter_addreq()函数
 - 5.18.2 defrouter_delreq()函数
 - 5.18.3 defrouter_addifreq()函数
 - 5.18.4 defrouter_delifreq()函数
 - 5.18.5 defrouter_lookup()函数
 - 5.18.6 defrouter_select()函数
 - 5.18.7 defrtrlist_del()函数
 - 5.18.8 defrtrlist_update()函数
- 5.19 前缀管理函数
 - 5.19.1 nd6_prelist_add()函数
 - 5.19.2 prelist_remove()函数
 - 5.19.3 prelist_update()函数
 - 5.19.4 find_pfxlist_reachable_router()函数
- 5.19.5 与在链条件有关的前缀和地址状态
- 5.19.6 pfxlist_onlink_check()函数
- 5.19.7 nd6_prefix_onlink()函数
- 5.19.8 nd6_prefix_offlink()函数
- 5.20 无状态地址自动配置函数
 - 5.20.1 in6_ifadd()函数
 - 5.20.2 in6_tmpifadd()函数
 - 5.20.3 regen_tmpaddr()函数
- 5.21 重复地址检测函数
 - 5.21.1 nd6_dad_find()函数
 - 5.21.2 nd6_dad_starttimer()函数

5.21.3 nd6_dad_stoptimer()函数

5.21.4 nd6_dad_start()函数

5.21.5 nd6_dad_stop()函数

5.21.6 nd6_dad_timer()函数

5.21.7 nd6_dad_duplicated()函数

5.21.8 nd6_dad_ns_output()函数

5.21.9 nd6_dad_ns_input()函数

5.21.10 nd6_dad_na_input()函数

5.22 其他函数

5.22.1 nd6_is_addr_neighbor()函数

5.22.2 nd6_output()函数

5.22.3 rt6_flush()函数

5.22.4 nd6_rtmsg()函数

第6章 传输层的实现

6.1 简介

6.2 ipv6上的tcp和udp

6.3 ipv6的伪首部

6.4 ipv4校验和与ipv6校验和的区别

6.5 ipv4映射的ipv6地址的用法

6.6 代码介绍

6.7 对pcb和套接字的一般操作

6.7.1 ipv6 pcb的分配: in_pcballoc()函数

6.7.2 绑定本地地址: in6_pcbbind()函数

6.7.3 固定远程地址: in6_pcbconnect()函数

6.7.4 in6_pcbladdr()函数

6.7.5 搜索pcb条目: in6_pcblookup_local()函数

6.7.6 搜索ipv4映射的pcb: in_pcblookup_local()函数

6.7.7 搜索pcb条目: in6_pcblookup_hash()函数

6.7.8 搜索ipv4映射的pcb: in_pcblookup_hash()函数

6.7.9 分离一个ipv6 pcb: in6_pcbdetach()函数

6.7.10 控制报文信令: in6_pcbnotify()函数

6.7.11 清空pcb缓存路由: in6_rtchange()函数

6.7.12 获取对等实体地址: in6_setpeeraddr()函数

6.7.13 获取本地地址: in6_setsockaddr()函数

6.8 ipv6上的tcp

6.8.1 ip6protosw{}的ipv6上的tcp实例

6.8.2 tcp输出

6.8.3 初始化首部: tcp_fillheaders()函数

6.8.4 tcp输入: tcp6_input()和tcp_input()函数

6.8.5 tcp控制输入: tcp6_ctlinput()函数

6.8.6 tcp用户请求

6.9 ipv6上的udp

6.9.1 ip6protosw{}的ipv6上的udp实例

6.9.2 udp输出: udp6_output()函数

6.9.3 udp输入: udp6_input()函数

6.9.4 udp控制输入: udp6_ctlinput()函数

6.9.5 udp用户请求的处理

6.10 原始ipv6

6.10.1 原始ipv6统计数据

6.10.2 原始ipv6输出: rip6_output()函数

6.10.3 原始ipv6输入: rip6_input()函数

6.10.4 icmpv6输入: icmp6_rip6_input()函数

6.10.5 原始ipv6控制输入: rip6_ctlinput()函数

6.10.6 原始ipv6控制输出: rip6_ctloutput()函数

- 6.10.7 原始ipv6用户请求处理
- 6.11 对ipv4映射的ipv6地址操作的总结
- 6.12 用netstat查看ipv6连接
- 6.13 配置ipv4映射的ipv6地址支持
- 第7章 套接字api扩展
 - 7.1 简介
 - 7.2 基本套接字api
 - 7.2.1 基本定义
 - 7.2.2 接口标识
 - 7.2.3 在af_inet6套接字上进行ipv4通信
 - 7.2.4 地址和名字转换函数
 - 7.2.5 基本套接字选项
 - 7.3 高级套接字api——[rfc3542]
 - 7.3.1 一些高级的定义
 - 7.3.2 ipv6原始套接字
 - 7.3.3 辅助数据介绍
 - 7.3.4 ipv6分组信息
 - 7.3.5 处理ipv6扩展首部
 - 7.3.6 路径mtu的api
 - 7.3.7 用于一批“r”命令的套接字扩展
 - 7.3.8 列表总结套接字选项
 - 7.4 ipv6套接字api的内核实现
 - 7.4.1 代码介绍
 - 7.4.2 ip6_pktopts{}结构
 - 7.4.3 ipv6套接字选项处理：ip6_ctloutput()函数
 - 7.4.4 获取套接字选项：ip6_getpcbopt()函数
 - 7.4.5 设置套接字选项与辅助数据
 - 7.4.6 清理：ip6_freepcbopts()函数
 - 7.4.7 ipv6多播套接字选项
 - 7.4.8 ipv6原始套接字选项：ip6_raw_ctloutput()函数
 - 7.4.9 icmpv6套接字选项：rip6_ctloutput()函数
 - 7.4.10 传送输入信息：ip6_savecontrol()函数
 - 7.5 套接字选项与辅助数据示例
 - 7.5.1 发送路径示例
 - 7.5.2 接收路径示例
 - 7.6 库函数的实现：libinet6
 - 7.6.1 inet_pton()函数和inet_pton6()函数
 - 7.6.2 inet_ntop()函数和inet_ntop6()函数
 - 7.6.3 getaddrinfo()函数
 - 7.6.4 地址排序示例
 - 7.6.5 freeaddrinfo()函数
 - 7.6.6 gai_strerror()函数
 - 7.6.7 getnameinfo()函数
 - 7.6.8 其他库函数
 - 参考文献
 - 索引
 - • • • • ([收起](#))

[IPv6详解（卷1）_下载链接1](#)

标签

IPv6

网络

卷1：核心协议实现

IPv6详解

计算机网络

计算机

内核

KAME

评论

苦逼的工科男。。。 Quick browse over, 2012年11月7日

快速瞄了两小时，太细了。适合对协议开发有兴趣的人。

当成手册书了。源码是结合kame项目，所以要看源码就会很头疼。理论部分到时可以借鉴。

与steven的卷二风格类似

算是IPv6领域的一本比较早的、比较权威的书，不仅仅是解释RFC。每章先讲概念、理论，然后讲BSD内核中的实现。

有点厚，只看了一部分，当时是为了实现IPv6组播看的，还不错

[IPv6详解（卷1）_下载链接1_](#)

书评

这本书是沿用Richard Stenves经典的《TCP/IP Illustrated Volume 2: The Implementation》，在看这本书之前，花了2个星期看来一遍Stenves的经典书（买了8年了，一直没有看完，这次终于看完了，很有收获）。总体来说还行，基本讲清楚了FreeBSD下的IPv6 KAME实现。KAME项目开...

[IPv6详解（卷1）_下载链接1_](#)