

文章编号: 2095-2163(2023)06-0007-06

中图分类号: TP393

文献标志码: A

# 基于 FPGA 的 UDP\_IP 协议栈的设计与实现

商艳娟, 徐卓汀, 王成群

(浙江理工大学 信息学院, 杭州 310018)

**摘要:** 工业控制现场复杂性的提升增加了对数据传输性能的要求, 为了提高工业控制现场设备之间数据传输的实时性和确定性, 本文对 TCP/IP 协议的基础架构进行研究, 设计了一种 UDP\_IP 网络通信协议栈, 支持 ARP 协议、ICMP 协议以及 UDP 协议通信; 通过使用 Cyclone IV 系列 FPGA 芯片和以太网 PHY 芯片 RTL8201CP 实现了各种网络协议通信。经过测试, 验证本文设计的 UDP\_IP 协议栈能够正确的发送和接收各种协议报文数据, 设计灵活, 具有较好的稳定性和较强的可移植性。

**关键词:** FPGA; UDP\_IP 协议栈; UDP 协议; 以太网

## Design and implementation of UDP\_IP protocol stack based on FPGA

SHANG Yanjuan, XU Zhuoting, WANG Chengqun

(School of Information, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**【Abstract】** The improvement of the complexity of the industrial control field increases the requirements for data transmission performance. To improve the efficiency and certainty of data transmission among devices in industrial control field, we conduct detailed study on the basic structure of the TCP/IP protocol and propose a new UDP\_IP network communication protocol stack which supports ARP protocol, ICMP protocol and UDP protocol communication. Various network protocol communication is realized by using Cyclone IV series FPGA chip and Ethernet PHY chip RTL8201CP. The designed protocol stack is flexible with good stability and strong portability and the test experiments verify that the designed UDP\_IP protocol stack can correctly send and receive various protocol message data.

**【Key words】** FPGA; UDP\_IP protocol stack; UDP protocol; ethernet

## 0 引言

网络通信技术的快速发展, 为大量数据的传输提供了可能<sup>[1]</sup>。总线按照传输数据的方式可以分为串行总线和并行总线。并行总线控制复杂, 传输距离有限, 串行总线设计简单, 但是传输速率低<sup>[2]</sup>。以太网技术的发展克服了总线传输的缺点, 接口速度快, 传输的距离远, 在复杂的工业控制系统中受到广泛应用<sup>[3]</sup>。传统的以太网在主控芯片内部编写以太网协议或者使用嵌入式处理器运行操作系统, 在操作系统下完成协议的通信<sup>[4]</sup>。使用主控芯片和嵌入式处理器来实现以太网通信协议无法保证较高的数据传输实时性和确定性。使用现场可编程门阵列(FPGA, Field Programmable Gate Array)实现数

据的通信, 因为通过硬件固化协议的处理逻辑, 可以有效地提高数据传输系统的性能, FPGA 芯片与物理层(PHY, Physical)芯片相连, 连接结构简单, 不需要外部设备驱动与大量的外围控制电路, 可移植性强, 系统集成度高<sup>[5]</sup>。基于综上的一些研究, 本文设计了一款 UDP\_IP 协议栈, 将其移植到 FPGA 芯片中, 使用以太网线将包含 FPGA 芯片和 PHY 芯片的电路板与电脑相连, 完成数据的传输。本设计具有以下功能:

- (1) 实现 ARP 请求和应答报文的发送, 完成 IP 地址和 MAC 地址的映射;
- (2) 支持 ICMP 报文的 ping 操作;
- (3) 实现 UDP 回环通信。

**基金项目:** 浙江省级重点研发计划(2021C01047); 流程工业综合自动化国家重点实验室开放课题基金(2021-KF-21-06, 2020-KF-21-09)。

**作者简介:** 商艳娟(1997-), 女, 硕士研究生, 主要研究方向: 物联网、工业互联网; 徐卓汀(1998-), 男, 硕士研究生, 主要研究方向: 嵌入式与物联网技术; 王成群(1981-), 男, 博士, 讲师, 主要研究方向: 工业互联网、AI 及其在行业中的应用。

**通讯作者:** 王成群 Email: cqwang@zstu.edu.cn

收稿日期: 2022-06-21

## 1 UDP\_IP 协议栈结构和功能

UDP\_IP 协议栈模型是将开放式系统互联通信参考模型 (Open System Interconnection Reference Model, OSI) 7 层结构简化成 4 层, 从上往下分别是应用层、传输层、网络层、网络接口层, 如图 1 所示。应用层支持 SMTP、HTTP、TFTP 等协议传输<sup>[6]</sup>; 传输层包含 TCP 协议和 UDP 协议, TCP 协议是一种基于 3 次握手机制的可靠传输协议, UDP 是一种不可靠的传输协议, 该协议实现简单, 适用于低时延、数据量大的快速传输场景; 网络层可以实现 IP 协议、ICMP 协议通信, IP 协议负责将数据报通过互联网发送出去, ICMP 协议主要是采用错误侦测和反馈机制, 为通信环境提供可能发生的反馈信息; 网络接口层支持 ARP 协议传输, 将已知的 IP 地址解析成 MAC 地址, 使得交换机可以通过 MAC 地址进行相应的通信。本文设计的 UDP\_IP 协议栈可以实现 ARP 请求、ARP 响应、ICMP 响应、UDP 回环等多种协议传输。



图 1 UDP\_IP 协议栈参考模型

Fig. 1 UDP\_IP protocol stack model

## 2 UDP\_IP 协议栈设计

本文采用模块化的思想设计 UDP\_IP 协议栈, 整个 UDP\_IP 协议栈由 6 个模块组成, 分别是 MAC 数据解析模块、IP 数据解析模块、ARP 处理模块、ICMP 处理模块、UDP 处理模块、MAC 数据封装模块, UDP\_IP 协议栈总体架构如图 2 所示。

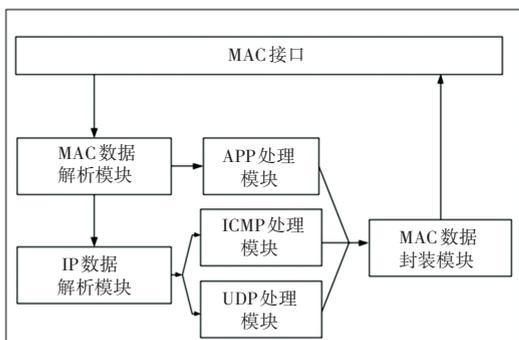


图 2 UDP\_IP 协议栈总体架构

Fig. 2 Overall architecture of the UDP\_IP protocol stack

### 2.1 MAC 数据解析模块设计

MAC 数据解析模块的作用是分离出 ARP 报文和 IP 报文, 在这个模块中使用异步先入先出队列 (First Input First Output, FIFO) 进行数据的跨时钟域处理, MAC 数据解析模块处理报文的流程图如图 3 所示。首先, 调用异步 FIFO 的 IP 核, 将接收的数据以及接收使能信号全部缓存进 FIFO 中; 根据 FIFO 中存储数据的个数产生 FIFO 读使能信号, 将数据和接收使能信号读出; 根据读出的使能信号驱动计数器计数, 当计数到相应的值时, 解析在该值处接收的数据, 该数据即是以以太网协议类型; 判断协议的类型是否为 IP 或 ARP 协议类型, 如果不是, 直接丢弃该数据帧, 如果类型值为 0x0800, 则判断该报文属于 IP 报文, 如果类型值为 0x0806, 则该报文属于 ARP 报文, 根据不同类型报文的判断结果, 将数据送入不同的后级模块。

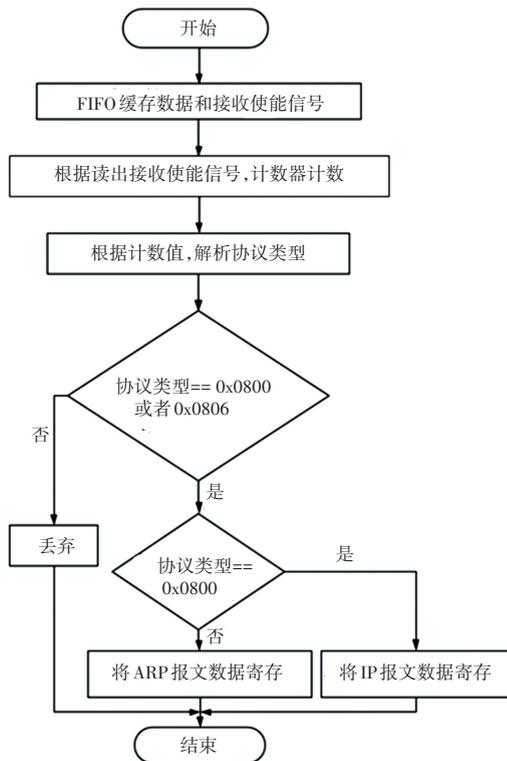


图 3 MAC 数据解析模块流程图

Fig. 3 Flow chart of the MAC data parsing module

### 2.2 ARP 处理模块设计

ARP 处理模块的主要功能是对接收的 ARP 报文进行分析, 更新 ARP 缓存表, 具体实现过程如图 4 所示。当上级 MAC 数据解析模块将 ARP 报文发送到 ARP 处理模块后, 对 ARP 接收报文进行解析, 得到 ARP 报文中的操作码、发送方 IP 地址、发送方 MAC 地址、接收方 IP 地址。当接收的操作码为 1,

同时报文中的目的 IP 等于本地设置的 IP,说明接收的是 ARP 请求报文,将该报文中的发送方 IP 地址和 MAC 地址寄存,用作 ARP 缓存列表的更新值,同时还用于发送 ARP 应答报文中目的 IP 地址和目的 MAC 地址。当接收的操作码为 2,同时报文中的目的 IP 等于本地设置的 IP,判断接收的是 ARP 应答报文,应答报文只需将报文中发送方 IP 地址和 MAC 地址寄存,用来 ARP 缓存列表信息的更新。图 4 虚线部分实现了 ARP 列表的更新,本设计定义了一个数组存储 ARP 信息,每当接收一个 ARP 请求或者是 ARP 应答报文时,定义的数组单元会更新 IP 地址和 MAC 地址,每 8 条 ARP 缓存满时,会自动覆盖之前的信息,进行下一轮的更新。另外,本设计还自定义了一个固定的目的 IP 地址,协议栈会自动地向这个自定义的目的 IP 地址主机定时发送 ARP 请求报文。

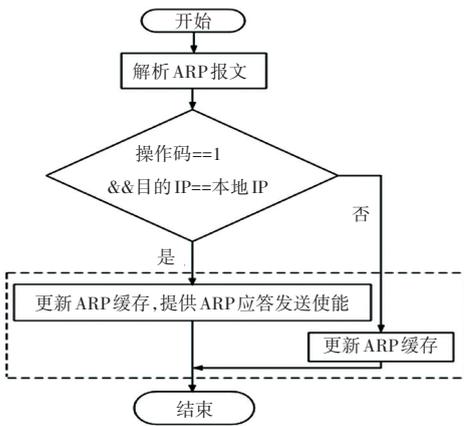


图 4 ARP 处理模块流程图

Fig. 4 Flow chart of ARP processing module

### 2.3 IP 数据解析模块设计

IP 数据解析模块有两个作用:将 UDP 报文和 ICMP 报文分离;进行 IP 层校验和计算,检测接收报文的 IP 层是否正确,IP 数据解析模块流程图如图 5 所示。由 MAC 分离出的 IP 报文被送入 IP 数据解析模块,对 IP 报文的头部进行解析,得到上层协议类型,目的 IP 地址。在报文解析的同时,对 IP 的报头信息进行校验和计算,得到的正确值为 16' hffff,如果校验和的计算不是该值,说明接收到的 IP 报文是错误的,丢弃该报文。根据报文解析的上层协议类型、目的 IP 地址、IP 校验和来判断该报文是 UDP 报文还是 ICMP 报文,将不同类型的报文数据使用存储器寄存,传输到接下来的 ICMP 处理模块或者是 UDP 处理模块。

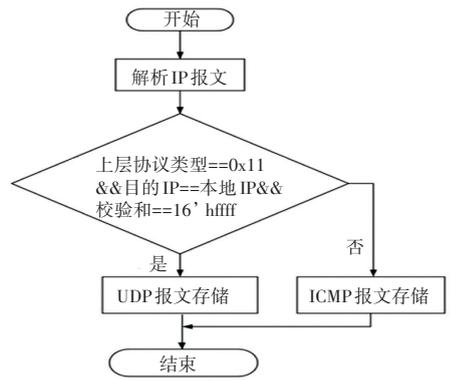


图 5 IP 数据解析模块流程图

Fig. 5 Flow chart of IP data analysis module

### 2.4 ICMP 处理模块设计

ICMP 处理模块能够对 ICMP 请求报文作出响应,同时将接收的 ICMP 报文数据部分缓存,然后转发。在该模块中整个报文的处理过程经历 3 个不同的状态,如图 6 所示。当 IP 数据解析模块没有发送任何数据给 ICMP 处理模块时,该模块一直处于空闲状态,一旦接收到上级模块发送 ICMP 报文,ICMP 报文接收开始使能信号跳变为高电平,状态跳转到接收状态;接收状态下,对接收的 ICMP 报文进行解析,得到 ICMP 报文的类型和代码值分别为 0x08 和 0x00,则接收的是 ICMP 请求报文,直到 ICMP 报文接收结束使能信号变为高电平时,状态跳转到发送状态,如果接收的是 ICMP 应答报文时,状态一直保持接收状态;接收状态下,另一个操作是使用同步随机存取存储器(Random Access Memory, RAM)存储报文,等到发送状态下取出报文数据部分内容,发送回电脑。等到发送的数据正好等于接收的数据长度时,ICMP 报文发送结束信号跳变为高电平,由发送状态跳转到空闲状态,等待 IP 数据解析模块发送下一个 ICMP 报文。

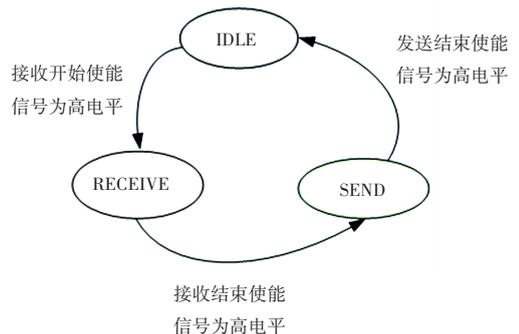


图 6 ICMP 处理模块状态机图

Fig. 6 State of ICMP processing module

### 2.5 UDP 处理模块设计

UDP 处理模块完成 UDP 报文的解析并且为发

送 UDP 回包封装 UDP 首部信息,整个 UDP 处理模块状态跳转如图 7 所示。首先,如果 IP 数据解析模块没有发送任何 UDP 报文,状态机一直处于空闲状态,直到接收到有效的 UDP 报文,UDP 报文接收开始使能信号跳为高电平,状态机随后跳转到接收状态;等到所有数据报文接收完成后,UDP 报文接收结束使能信号变为高电平,状态机跳转到计算 UDP 校验和状态,等到校验和的计数器计数到 9 时,状态会跳转到发送状态;当发送的数据长度正好等于接收的数据长度时,UDP 报文发送结束使能信号变为高电平,表示发送完成,状态机跳转到空闲状态,等待 IP 数据解析模块传输下一个 UDP 报文数据。在整个状态机跳转过程中,数据处于接收状态时,解析 UDP 报文,得到目的端口号、源端口号、UDP 数据长度、发送方 IP 地址,为 UDP 回环的实现提供目的端口号和源端口号。数据需要返回给发送端,必须将接收的 UDP 数据缓存下来,调用一个同步 RAM 的 IP 核进行数据的缓存。在 UDP 报头校验和计算状态下,计算校验和为发送 UDP 回包提供正确的 UDP 校验和。

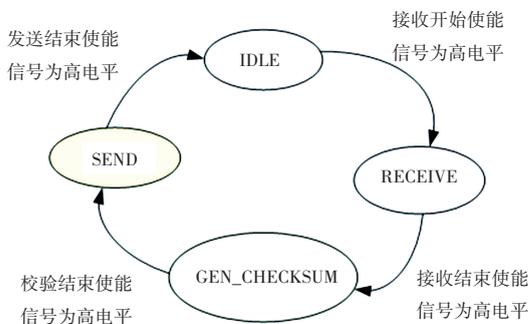


图 7 UDP 处理模块状态机

Fig. 7 State of UDP processing module

## 2.6 MAC 数据封装模块设计

ARP 处理模块提取的目的 IP 和目的 MAC 地址,ICMP 处理模块重新打包的 ICMP 应答报文以及 UDP 处理模块重新封装的 UDP 报文都传入 MAC 数据封装模块合流。具体选择返回哪一种协议报文在 MAC 数据封装模块需要进行仲裁,具体的 MAC 数据封装模块流程图如图 8 所示。首先,判断发送使能信号,如果是 ARP 报文发送使能信号,还需要判断发送的是 ARP 应答报文还是 ARP 请求报文。如果 ARP 应答报文发送使能信号为高电平,按照 ARP 数据格式组帧整个 ARP 报文内容;如果是 ARP 请求发送使能信号为高电平,依照同样的组帧方式设置 ARP 请求报文内容,传输到 MAC 数据封装模块。ARP 报文前面再封装以太网首部信息,最

终发送给目标主机。如果判断是 IP 报文发送使能信号为高电平,根据发送的上层协议类型,得到发送的报文是 ICMP 报文还是 UDP 报文,如果协议类型是 0x11,则表示发送 UDP 报文,将 UDP 处理模块传输过来的内容作为整个以太网帧的数据部分,如果协议类型是 0x01,则表示发送 ICMP 报文,将 ICMP 处理模块传输过来的内容作为整个以太网帧的数据部分。以太网帧其他部分使用以太网首部以及 IP 首部信息填充。数据按照标准以太网协议帧封装完成后,可以将这些数据直接传输给 MAC 接口,MAC 将这些数据传回给发送端。

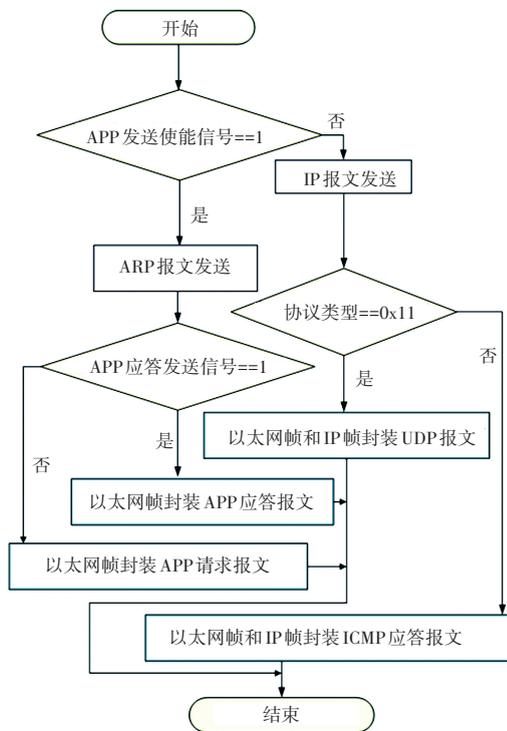


图 8 MAC 数据封装模块流程图

Fig. 8 Flow chart of MAC data encapsulation module

## 3 设计验证

本设计在 Quartus II 13.0 软件中使用 Verilog 语言编写整个 UDP 协议栈代码,设计完成后使用 modelsim 仿真软件对整个 UDP\_IP 协议栈进行功能验证。ICMP 请求报文和应答报文的仿真时序图如图 9 和图 10 所示。从图 9 可以看到两个报文发送的以太网协议类型都是 0x0800, ICMP 报文的首部符合协议数据的设定值,接收数据与发送数据是相同的,报文的结尾都是使用 FCS 校验码封装,验证了本协议栈可以正确接收 ICMP 请求报文和发送 ICMP 应答报文;此外,关于 ARP 协议和 UDP 协议的接收和发送功能仿真验证也是正确的。

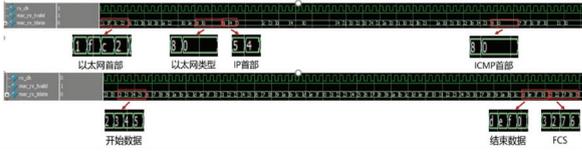


图 9 ICMP 请求报文接收图  
Fig. 9 ICMP reception diagram

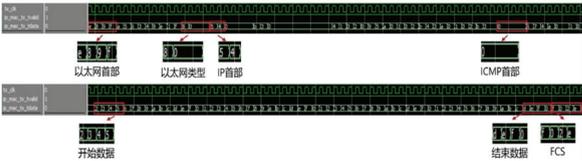


图 10 ICMP 应答报文发送图  
Fig. 10 ICMP message sending diagram

315	28.283486	2c:16:db:a8:22:e0	Broadcast	ARP	60 Who has 10.1.178.52? Tell 10.1.178.51
316	28.284126	IntelCor_43:e9:f1	2c:16:db:a8:22:e0	ARP	60 10.1.178.52 is at 98:af:65:43:e9:f1
317	29.284001	2c:16:db:a8:22:e0	Broadcast	ARP	60 Who has 10.1.178.52? Tell 10.1.178.51
318	29.284647	IntelCor_43:e9:f1	2c:16:db:a8:22:e0	ARP	60 10.1.178.52 is at 98:af:65:43:e9:f1
319	30.284116	2c:16:db:a8:22:e0	Broadcast	ARP	60 Who has 10.1.178.52? Tell 10.1.178.51
320	30.284748	IntelCor_43:e9:f1	2c:16:db:a8:22:e0	ARP	60 10.1.178.52 is at 98:af:65:43:e9:f1
32	31.000327	2c:16:db:a8:22:e0	IntelCor_43:e9:f1	ARP	60 10.1.178.51 is at 2c:16:db:a8:22:e0
33	32.000096	2c:16:db:a8:22:e0	IntelCor_43:e9:f1	ARP	60 10.1.178.51 is at 2c:16:db:a8:22:e0
34	32.384259	IntelCor_43:e9:f1	Broadcast	ARP	60 Who has 10.1.178.51? Tell 10.1.178.52
35	32.384291	2c:16:db:a8:22:e0	IntelCor_43:e9:f1	ARP	42 10.1.178.51 is at 2c:16:db:a8:22:e0
36	33.000332	2c:16:db:a8:22:e0	IntelCor_43:e9:f1	ARP	60 10.1.178.51 is at 2c:16:db:a8:22:e0

图 11 ARP 报文测试结果图  
Fig. 11 ARP packet test result

```

0000 98 af 65 43 e9 f1 2c 16 db a8 22 e0 08 00 45 00  ..eC... ..E-
0010 00 3c 94 78 00 00 00 01 00 00 0a 01 b2 33 0a 01  <-X-@...3-
0020 b2 34 08 00 4a fe 00 01 02 5d 61 62 63 64 65 66  -4-J...]abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmn opqrstv
0040 77 61 62 63 64 65 66 67 68 69  wabdefgh hi

0000 2c 16 db a8 22 e0 98 af 65 43 e9 f1 08 00 45 00  ..eC... ..E-
0010 00 3c 00 03 00 00 00 01 02 55 0a 01 b2 34 0a 01  <-...@...U...4-
0020 b2 33 00 00 52 fe 00 01 02 5d 61 62 63 64 65 66  -3-R...]abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmn opqrstv
0040 77 61 62 63 64 65 66 67 68 69  wabdefgh hi
  
```

图 12 ICMP 报文测试结果图  
Fig. 12 ICMP packet test result diagram

为了评估本设计 UDP\_IP 协议栈发送 UDP 循环报文的性能,使用网络调试助手每 1 ms 发送一次 UDP 报文,测试的结果如图 13 所示。发送 UDP 报文次数达到 10 万次,每一次的发送都能正确接收,说明本文设计的 UDP\_IP 协议栈具有可靠稳定的传输性能,传回电脑的数据与电脑发送出去的数据是完全一致的,验证了数据能够在该协议栈内完成正确的数据接收和发送。

仿真验证后进行板级验证,使用 Cyclone IV 系列 FPGA 芯片电路板。通过电脑发送不同的协议报文,然后使用 wireshark 软件抓取电脑发送的报文以及接收的报文,验证 UDP\_IP 协议栈设计的功能。当使用小兵以太网发送 ARP 请求报文和 ARP 应答报文时,ARP 报文测试结果如图 11 所示。电脑每发送一个 ARP 广播报文,板子会返回一个 ARP 应答报文给电脑。当电脑发送 ARP 应答报文给板子时,板子不会立即返回请求报文,但是本设计设置了一个电脑 IP 地址存储在 ARP 处理模块,会定时发送 ARP 请求报文给电脑。使用 ping 命令后,ICMP 报文测试结果图如图 12 所示,可以看到当发送一个 ICMP 请求报文,会传回一个 ICMP 应答报文。接收报文数据和发送报文数据内容完全相同,证明了接收的数据在协议栈中能够正确缓存并转发。

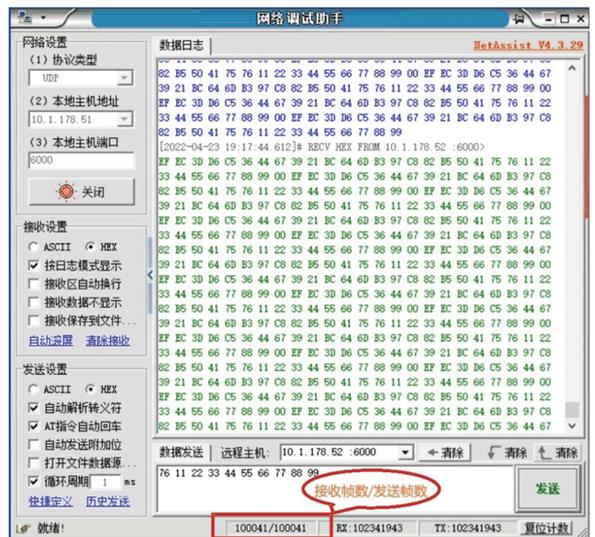


图 13 UDP 报文测试结果图  
Fig. 13 UDP packet test result diagram

## 4 结束语

本文设计了一种基于 FPGA 的 UDP\_IP 协议栈, ICMP 报文和 UDP 报文数据部分在该协议栈内完成了正确的缓存和转发, 当发送 ARP 报文时, 在该协议栈内能够及时进行 ARP 列表的更新。本设计具有良好的稳定性和可靠性, 为高速以太网通信提供了一个很好的解决方案, 有着广泛的应用前景。

## 参考文献

[1] 武朋, 黄虎, 王兴. 基于 FPGA 的 TCP/IP 协议设计与实现[J].

(上接第6页)

文研究了基于该环境下的服务迁移问题, 并对服务迁移过程中产生的计算成本、迁移成本和能耗建模, 将迁移决策规划为部分迁移而非二元迁移, 同时考虑了车辆移动性带来的问题。本文将服务迁移建模为 MDP 问题, 并提出了基于深度强化学习的迁移算法来降低服务的平均时延和能耗。实验结果表明, 本文提出的算法在学习率为 0.005 时能够较快达到收敛, 并且与其他算法相比在 20 km/h 和 60 km/h 时分别有 8.6% 和 15.3% 的性能提升, 同时当移动速度处于 60 km/h 时, 在能耗方面有 14.4% 的优势。在未来工作中, 将车辆加速度引入以预测用户驾驶车辆的移动意图, 从而进行更精确的迁移决策。

## 参考文献

- [1] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: A survey [J]. *IEEE Internet of Things Journal*, 2018, 5(1): 450-465.
- [2] BI J, YUAN H, DUANMU S, et al. Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization [J]. *IEEE Internet of Things Journal*, 2020, 8(5): 3774-3785.
- [3] ZHANG Q, ZHENG H, ZHONG Z, et al. Energy minimization for fog computing-enabled hierarchical networks with dynamic TDD [C]//2019 IEEE/CIC International Conference on Communications in China (ICCC). IEEE, 2019: 590-595.
- [4] WANG K, HU Z, AI Q, et al. Joint offloading and charge cost minimization in mobile edge computing [J]. *IEEE Open Journal of the Communications Society*, 2020, 1: 205-216.
- [5] WU F, LI X, LI H, et al. Energy-time efficient task offloading for mobile edge computing in hot-spot scenarios[C]//ICC 2021-IEEE International Conference on Communications. IEEE, 2021: 1-6.

软件导刊, 2013, 12(5): 22-23.

- [2] 朱明辉, 司斌, 张从霞, 等. FPGA 与 88E1111 的千兆以太网接口设计[J]. *单片机与嵌入式系统应用*, 2017, 17(3): 60-63, 66.
- [3] 李勋, 刘文怡. 基于 FPGA 的以太网接口设计与实现[J]. *自动化与仪表*, 2014, 29(5): 57-60.
- [4] 董永吉, 王钰, 袁征. 基于 FPGA 的万兆以太网 UDP\_IP 硬件协议栈设计与实现[J/OL]. *计算机应用研究*: 1-4[2022-04-25].
- [5] 韩剑南, 胡辽林. 基于 FPGA 和 UDP/IP 协议的千兆网络图传系统[J]. *计算机系统应用*, 2018, 27(3): 99-104.
- [6] 刘源, 张刚. 可靠 UDP 协议栈的 FPGA 实现[J]. *火力与指挥控制*, 2017, 42(7): 139-143.
- [6] SINGH S, KIM D H. Profit optimization for mobile edge computing using genetic algorithm [C]//2021 IEEE Region 10 Symposium (TENSYP). IEEE, 2021: 1-6.
- [7] KIM S. Bargaining game based offloading service algorithm for edge-assisted distributed computing model [J]. *IEEE Access*, 2022, 10: 63648-63657.
- [8] OUYANG T, ZHOU Z, CHEN X. Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing [J]. *IEEE Journal on Selected Areas in Communications*, 2018, 36(10): 2333-2345.
- [9] VU T T, NGUYEN D N, HOANG D T, et al. Optimal energy efficiency with delay constraints for multi-layer cooperative fog computing networks [J]. *IEEE Transactions on Communications*, 2021, 69(6): 3911-3929.
- [10] ALBONDA H D R, PÉREZ-ROMERO J. An efficient RAN slicing strategy for a heterogeneous network with eMBB and V2X services [J]. *IEEE Access*, 2019, 7: 44771-44782.
- [11] KSENTINI A, TALEB T, CHEN M. A Markov decision process-based service migration procedure for follow me cloud [C]//2014 IEEE International Conference on Communications (ICC). IEEE, 2014: 1350-1354.
- [12] TANG Z, ZHOU X, ZHANG F, et al. Migration modeling and learning algorithms for containers in fog computing [J]. *IEEE Transactions on Services Computing*, 2019, 12(5): 712-725.
- [13] WILDMAN J, NARDELLI P H J, LATVA-AHO M, et al. On the joint impact of beamwidth and orientation error on throughput in directional wireless poisson networks [J]. *IEEE Transactions on Wireless Communications*, 2014, 13(12): 7072-7085.
- [14] LI Z, XIANG L, GE X, et al. Latency and Reliability of mmWave Multi-Hop V2V Communications Under Relay Selections [J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(9): 9807-9821.
- [15] BELLMAN R. A markovian decision process [J]. *Journal of Mathematics and Mechanics*, 1957, 6(5): 679-684.
- [16] SUTTON R S, BARTO A G. Reinforcement learning: An introduction [M]. MIT press, 2018: 472-475.