

基于5G的多路径数据传输系统设计

韩彦超¹, 李亮², 王成群¹

(1.浙江理工大学信息科学与工程学院, 浙江 杭州 310018;

2.浙江海利普电子科技有限公司, 浙江 海盐 314300)

✉1060163780@qq.com; li_316@163.com; cqwang@zstu.edu.cn



摘要: 针对目前电信运营商5G基站分布不均匀及5G SA模式基站覆盖效果有限的问题, 提出通过多路径数据传输协议(MultiPath TCP, MPTCP)与5G技术相结合的方案, 聚合不同运营商的5G通信链路, 提高5G在应用场景中的传输带宽、连接鲁棒性及网络服务质量。针对该方案, 设计基于5G的多路径数据传输系统; 该系统以I.MX6ULL作为主控制器, 通过两个5G模组完成数据传输, 采用局域网与外部终端完成数据交互, USB串口作为控制终端。测试结果表明: 系统使用MPTCP协议通过两条5G通信链路进行数据传输的传输速率比单条链路的传输速率提高了一倍左右, 并且系统稳定性也明显提高, 能够达到既定目标的链路聚合效果, 具有较高的实用价值。

关键词: 5G; MPTCP; I.MX6ULL; Linux

中图分类号: TP393 **文献标识码:** A

Design of Multipath Data Transmission System based on 5G

HAN Yanchao¹, LI Liang², WANG Chengqun¹

(1.Faculty of Information Science & Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China;

2.Zhejiang Holip Electronic Technology Co., Ltd., Haiyan 314300, China)

✉1060163780@qq.com; li_316@163.com; cqwang@zstu.edu.cn

Abstract: In view of the uneven distribution of 5G base stations of telecom operators and the limited coverage effect of 5G SA mode base stations, a scheme combining multipath TCP (MPTCP) with 5G technology is proposed to aggregate 5G communication links of different operators to improve the transmission bandwidth, connection robustness and network service quality of 5G in application scenarios. For this scheme, a multipath data transmission system based on 5G is designed. With I.MX6ULL as the main controller in this system, data transmission is completed through two 5G modules; data interaction between LAN (Local Area Network) and external terminals is completed with USB serial port as the control terminal. The test results show that the data transmission rate of the system using MPTCP protocol through two 5G communication links is about twice as high as that of a single link, and the system stability is also significantly improved, which can achieve the link aggregation effect of the established goal, and has high practical value.

Keywords: 5G; MPTCP; I.MX6ULL; Linux

1 引言(Introduction)

第五代移动通信^[1](5G)作为新一代蜂窝移动通信技术, 在4G技术的基础上集合了毫米波^[2], 大规模多输入多输出^[3], 以及波束赋形^[4]等多种通信领域的前沿技术, 具有传输速率高、低延时、高速度、广连接的特性^[5]。2019年, 5G技术在全球范围内正式投入商用部署, 极大地促进了工业、医疗、娱

乐、消费、交通等各行业数字化、网络化、智能化的发展。但是, 5G网络在农场直播带货、矿山无线远程遥控和深井作业等应用场景中会出现信号不稳定^[6-8]的问题, 导致数据传输速率严重下降和时延增大。这是因为5G商用化需要通过三大运营商建造的5G基站进行数据传输, 由于运营商建设的基站分布不均匀, 并且在农村及偏远地区建设的5G基站较少, 这

使得5G的传输性能受到极大的影响。针对上述问题，本文使用MPTCP^[9-11]技术设计了基于5G的多路径数据传输系统，目的是使用不同运营商的SIM卡将多条5G链路带宽聚合，提高5G在特殊应用场景中的传输带宽、连接鲁棒性及网络服务质量，有效解决当前5G应用中所面临的问题。

2 系统架构(System architecture)

系统总体架构(图1)包括主控模块、5G模组、数据交互和人机交互四个部分，其中主控模块使用ARM Cortex-A7内核^[12]的I.MX6ULL芯片作为主控制器控制系统的运行，完成数据的接收和发送；5G模组是实现远程数据传输的主要设备，本文系统中搭载了两个5G模块构成两条5G通信链路；数据交互采用以太网和无线Wi-Fi的方式与外部设备构成局域网，实现数据的采集。人机交互使用通用接口或设备实现与系统的交流，并对系统进行操作。

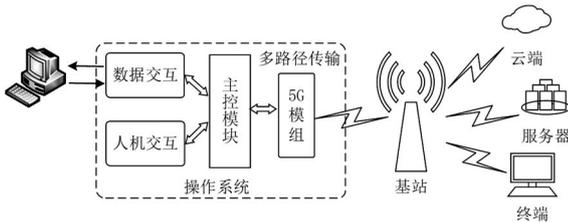


图1 系统总体架构图

Fig.1 Diagram of overall system architecture

3 系统硬件设计(System hardware design)

系统硬件设计框图如图2所示，输入DC12 V电压，经过BUCK转换电路后为系统平台的设备模块供电；核心控制系统由NXP公司的I.MX6ULL处理器芯片搭载FLASH和DDR3组成，为系统提供存储和运行内存空间。外围电路包括5G移动通信、有线以太网、无线网Wi-Fi、外部存储、USB串口、启动模式选择等设备，与核心控制系统连接完成系统硬件电路的搭建。

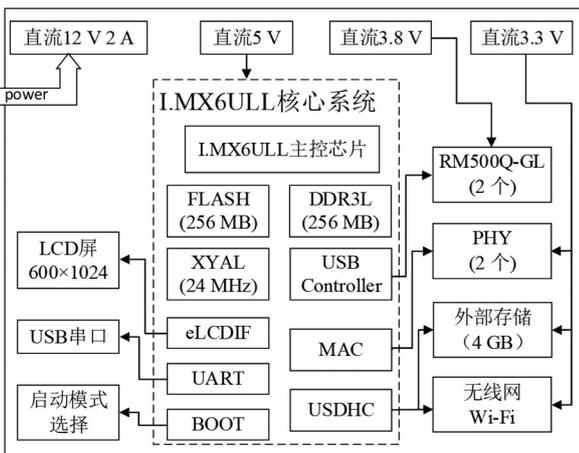


图2 系统硬件设计框图

Fig.2 Block diagram of system hardware design

3.1 I.MX6ULL核心控制电路

I.MX6ULL核心控制电路由内存扩展DDR3和存储空间扩展NAND FLASH接口电路组成。其中，将容量为512 MB的DDR3L芯片NT5CC256M16EP-EK用于系统内存空间扩展，为系统提供足够的运行空间；使用Micron公司512 MB的NAND FLASH芯片MT29F4G08作为存储扩展，用于存储系统镜像、设备树和根文件系统及相应的数据。

3.2 5G移动通信电路

本文使用Quectel公司的RM500Q-GL模组搭建5G移动通信电路。该模组支持5G NR-SA/NSA等多种网络制式；支持PCIe、USB3.0和USB2.0数据交互接口，是一款工业级5G通信模组。

5G移动通信硬件框图如图3所示，I.MX6ULL芯片内部集成UTMI接口^[13]连接控制器和USB PHY，其中TermSelect作为终端选择信号，通过控制上拉或下拉电阻的变化产生HS/FS状态下不同的线状态；LineState用于反映DP、DM线状态，根据该信号判断USB复位、挂起、恢复操作；TX/RX表示的信号为数据的收发信号；Data表示使用16 bit的数据接口。因此，模块直接通过USB2.0接口与处理器连接完成与控制器的数据交互；其中USB_DP和USB_DM是传输的差分数据线；开关机和复位引脚连接处理器的GPIO输出控制；模组提供SIM卡的接口。

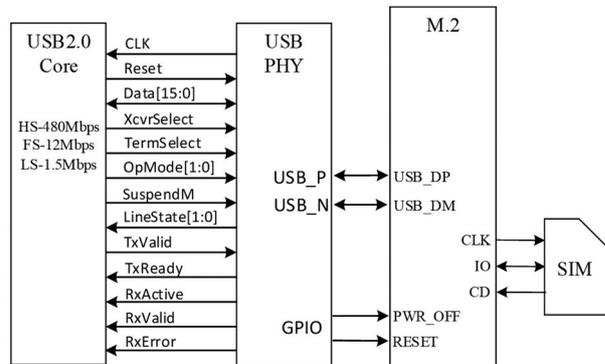


图3 5G移动通信硬件框图

Fig.3 Block diagram of 5G mobile communication hardware

3.3 有线和无线网电路

有线和无线网通信的硬件基础是网卡，网卡主要工作在数据链路层和物理层，其中数据链路层提供基于比特流的帧同步、数据流的流量控制、网络的链路控制等功能；物理层提供通信双方所需的物理信号。

以太网硬件框图如图4所示，以太网电路使用RMII接口连接处理器和PHY芯片LAN8720；处理器中数据发送和接收时钟的频率为50 MHz，它是通过专用Ethernet PLL倍频系统时钟生成的；TXP/TXN和RXP/RXN是两组差分信号，用

来传输物理层到数据链路层的电路信号；TXD和RXD是2 bit 数据传输接口；MDIO总线用于访问PHY芯片的任意寄存器。

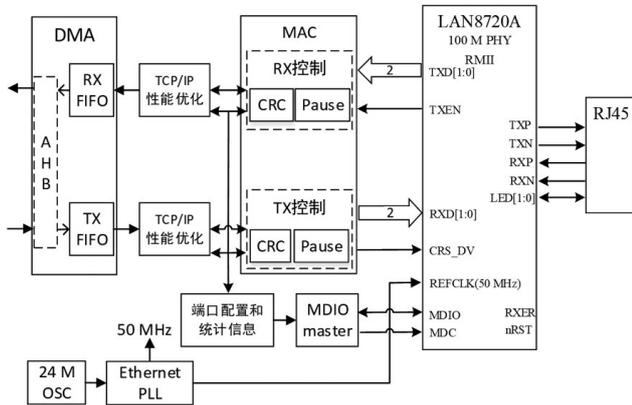


图4 以太网硬件框图

Fig.4 Block diagram of Ethernet hardware

无线网使用处理器I.MX6ULL内部IP核USDHC提供的SDIO接口连接RTL8189模块，该模块是集成MAC(Media Access Control)、PHY(Physical)和RF(Radio Frequency)于一体的802.11n无线局域网控制器，时钟频率为40 MHz时，传输速率最高可达150 Mbps。

4 系统软件设计(System software design)

在完成系统硬件设计的基础上，设计系统软件，实现系统设计的目标。系统软件总体架构如图5所示。在系统平台上移植Uboot引导程序和MPTCP_Linux0.95^[14]版本的内核，构建ubuntu-base根文件系统组成系统的运行环境。设备驱动文件的设计核心是根据板级硬件信息，使用Linux系统驱动框架实现硬件模块在系统运行环境中的全功能，为应用层提供读写数据功能。应用软件在用户空间使用系统API函数读写控制硬件设备。

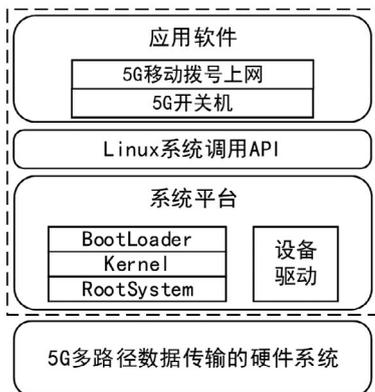


图5 系统软件总体架构图

Fig.5 Diagram of system software structure

4.1 系统运行环境

嵌入式硬件平台运行Linux系统需要完成Uboot、

MPTCP_Linux0.95和ubuntu-base的移植，三者之间的关系体现在Linux系统启动的过程：首先硬件上电芯片复位后，Uboot初始化部分硬件，完成DDR3、外部存储及以太网口等设备的初始化，然后加载引导存储器中的内核程序到DDR3中启动运行，Linux内核加载过程中会从ubuntu-base根文件系统中加载初始化脚本和服务程序，最后将根文件系统挂载执行相应的应用程序和系统进程。

在系统运行时，由于多个网卡接口上定义了不同的IP地址，如果没有指明路由所属的表，所有的路由默认放在主路由表里，所有网卡都会使用主路由表对应的网关进行路由，这不符合MPTCP的实际应用。因此，编写脚本文件使用ip rule和ip route命令配置路由，为每个网卡接口配置一个路由表，每个路由表有一个数字标识，系统会根据路由表编号由小到大进行检查，根据IP匹配对应的路由表，使所有网卡符合MPTCP的实际应用。

4.2 5G模组驱动

5G模组RM500Q-GL首先是一个USB设备，其次是网络设备。USB驱动是一种分层的总线结构，如图6所示，包括USB设备驱动和USB子系统驱动，其中USB设备驱动利用USB Core提供的API完成驱动的工作，通过管道(pipe)控制USB设备与主机控制器之间的通信；USB Core和USB主控制器驱动(HCD)构成USB子系统，隐藏了USB协议和硬件的复杂性；USB总线将设备和驱动关联起来。

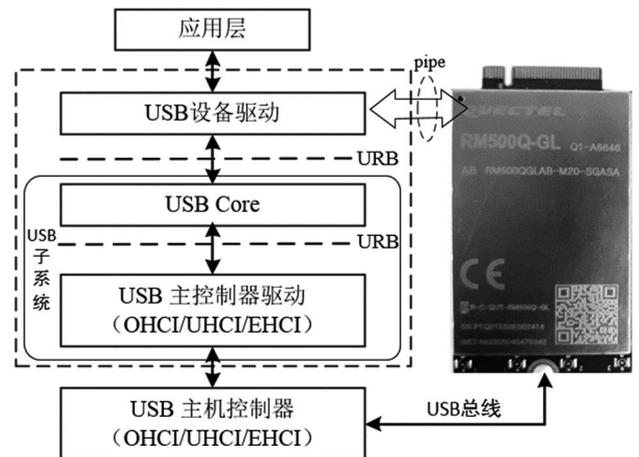


图6 USB驱动结构示意图

Fig.6 Diagram of USB drive structure

USB驱动的实现需要对Linux中USB子系统进行修改，实现与RM500Q-GL的匹配。首先添加设备供应商提供的VID=0X2C7C和PID=0X0800用于识别设备，加载相应的驱动程序；然后在struct usb_device_id option_ids数组中添加USB设备RM500Q-GL；最后增加零包处理机制，当端点传

输的数据包大小是wMaxPacketSize的整数倍时，需要发送一个长度为0的数据包结束这次传输。

USB设备驱动首先需要注册usb_driver，然后实现usb_device_id、probe、disconnect和name等成员变量，其中usb_device_id通过VID和PID匹配USB设备和驱动；probe函数完成设备的初始化，调用usbnet_probe函数将RM500Q-GL设置为USB网络设备。该驱动通过usb_control_msg函数发送USB的控制信息。

5G模组的网络驱动在usbnet_probe函数中绑定并初始化usb设备的网卡接口usb(x=0,1)，并向上层注册网卡设备，完成usb网络设备的配置，协议层就可以通过这个设备的tx、rx接口发送和接收数据。此外，实现5G模组拨号上网的功能，需要移植编译Quectel官方提供的GobiNet驱动实现模组的网络连接，在Linux系统下5G模组的USB接口被映射成ttyUSB 0-3这四个接口，接口的功能如表1所示。

表1 RM500Q-GL在系统中的接口

Tab.1 Interface of RM500Q-GL in the system

接口号	ttyUSB0	ttyUSB1	ttyUSB2	ttyUSB3
功能	DIAG	GNSS	AT Command	Modem

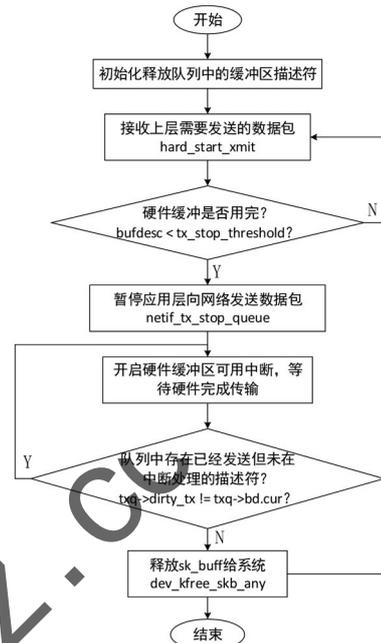
DIAG端口用于对网络参数的修改和平台日志信息的输出；GNSS是全球导航定位系统；ttyUSB2用于AT命令的交互；Modem端口用于拨号上网。因此，为防止5G模组RM500Q-GL的接口4被用作USB串口设备，需要在USB驱动文件中将接口4设置为USB网络接口。

4.3 网络驱动

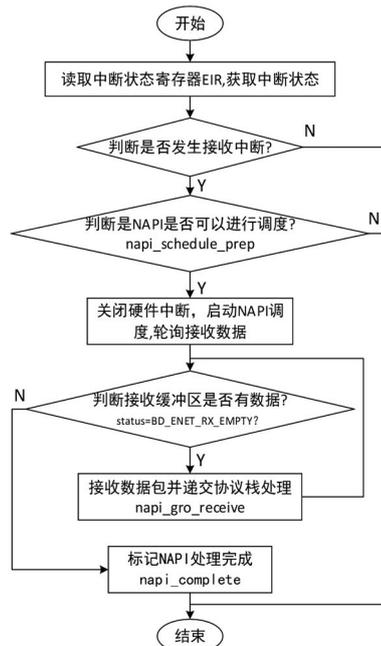
网络设备是用户数据包在网络媒介上发送和接收的设备。网络驱动架构中net_device结构体包含网络设备的属性描述和操作接口，以太网驱动的核心就是填充net_device中的具体成员并注册到Linux内核中。网络设备在初始化时首先使用alloc_netdev函数分配一个net_device结构体，然后配置相关的寄存器，设置net_device的结构成员，最后使用register_netdev函数将net_device注册到内核中。

以太网驱动程序中发送流程图如图7(a)所示，上层应用向驱动传入sk_buff结构体指针，接收来自上层传递的数据，使用环形缓冲区队列保存数据，如果数据发送队列剩余的描述符不足，需要暂停应用层向网络发送数据，并通过硬件中断将数据发送，直到队列中网络发送描述符与中断处理描述符指向地址相同，则结束发送并释放sk_buff空间给系统；接收流程图如图7(b)所示，该设计中使用中断和轮询的方式(NAPI)接收数据，首先创建sk_buff缓存接收数据，读取中断寄存器

的状态判断接收中断的触发，通过napi_schedule_prep的返回值判断NAPI是否可以调度，然后启动NAPI调度函数，在轮询调度函数中遍历接收描述符，并将数据递交到协议栈进行处理，直到缓冲区状态为空时停止接收数据。



(a)以太网发送流程图



(b)以太网接收流程图

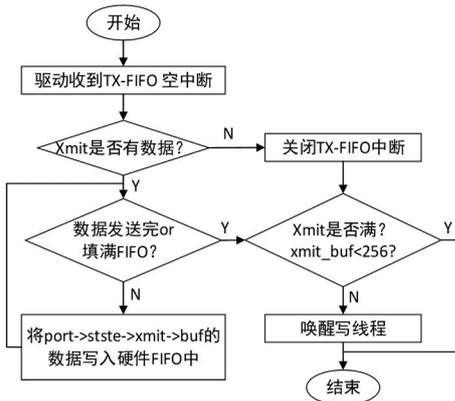
图7 以太网收发流程图

Fig.7 Ethernet transceiver flow chart

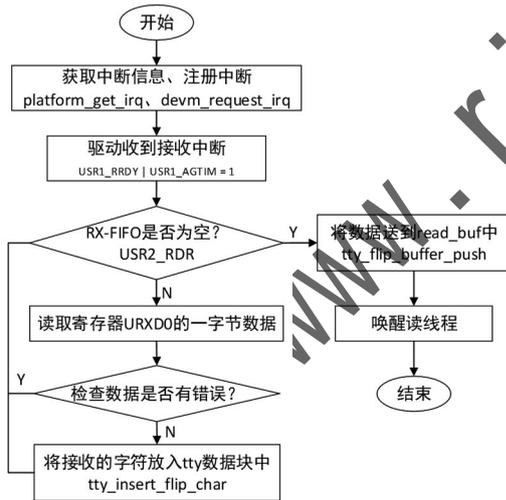
4.4 串口驱动

USB串口是常用的人机交互接口，串口驱动的设计使用platform总线结构，主要实现uart_port和uart_ops两个结构体中的函数。uart_port用于描述一个UART的I/O端口或I/O内

存地址、FIFO大小、端口类型等信息；uart_ops结构体是串口的操作函数集，包括串口的起始、发送、接收等函数，其中发送程序流程如图8(a)所示，用户进程执行写操作时会触发UART的发送中断，在中断处理函数中循环将上层驱动保存在xmit中的数据写入硬件FIFO，直到xmit中数据为空，关闭TX-FIFO中断；接收程序流程如图8(b)所示，在驱动匹配成功后获取设备树的中断信息，通过寄存器判断中断的触发信息，接着循环读取寄存器中的值，将接收的字符进行奇偶校验后存入tty数据块中，等待RX-FIFO中数据读完后将数据发送给tty核心层进行数据缓存。



(a)UART发送程序流程图



(b)UART接收程序流程图

图8 UART收发流程图

Fig.8 UART transceiver flow chart

5 实验验证(Experimental verification)

完成系统的软硬件设计后，测试系统的可运行性，系统实物连接图如图9所示，使用网线将系统平台连接路由器与PC机组成局域网；选择中国移动和中国电信的SIM卡插入卡槽，5G模组射频前端连接4根天线；系统启动模式设置为SD卡启动，方便调试；连接USB串口到PC端实现对系统的控制。

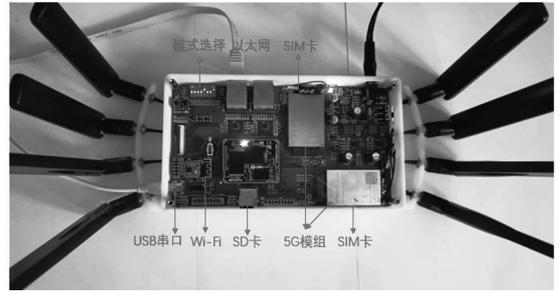


图9 系统实物连接图

Fig.9 Physical connection diagram of the system

通过“sysctl-w”命令对系统中MPTCP协议进行配置。如图10(a)所示，配置项包括：使用MPTCP协议，打开MPTCP的调试功能；配置MPTCP协议的路径管理算法为全路径(fullmesh)^[15]，调度算法为轮询调度(roundrobin)^[16]，拥塞控制算法为Cubic^[17]；支持重传次数为3次。运行mptcp_route.sh脚本文件完成对系统路由表的配置，如图10(b)所示。

```
net.mptcp.mptcp_binder_gateways =
net.mptcp.mptcp_checksum = 1
net.mptcp.mptcp_debug = 1
net.mptcp.mptcp_enabled = 1
net.mptcp.mptcp_path_manager = fullmesh
net.mptcp.mptcp_scheduler = roundrobin
net.mptcp.mptcp_syn_retries = 3
net.mptcp.mptcp_version = 0
```

(a)MPTCP配置结果

```
root@hyc_tm6ull:/home/hyc_app# ./mptcp_route.sh 运行路由配置脚本
网卡信息 usb0 10.133.247.147 255.255.255.248 10.133.247.148 10.133.247.144 1
网卡信息 usb1 10.68.101.39 255.255.255.240 10.68.101.40 10.68.101.32 2
root@hyc_tm6ull:/home/hyc_app# cd ../video/
root@hyc_tm6ull:/home/video# ip rule show
0: from all lookup local
32764: from 10.68.101.39 lookup 2
32765: from 10.133.247.147 lookup 1
32766: from all lookup main
32767: from all lookup default
```

(b)系统路由表配置结果

图10 系统多路径功能配置

Fig.10 System multipath function configuration

为验证本文设计的5G多路径数据传输系统的性能，在PC端使用FTP协议将数据传输到系统，系统接收到数据后分别使用一条和两条5G链路将数据上传到云端。测试结果如图11所示，在某一运营商信号不稳定时，使用单条路径传输数据的速率仅为1.1 MB/s；而使用两条运营商通信链路传输的速率为2.3 MB/s。由此可见，使用两个不同运营商的5G链路进行数据传输的速率明显高于单条路径数据传输的速率。

```
单路径:
mptcp_alloc_mpcb: created mpcb with token 0xf90808bd 单路径
mptcp_add_sock: [token 0xf90808bd pt 1, src_addr:10.134.175.145]59462 dst_addr:119.3
root@119.3.52.83's password:
109MB.m4
mptcp_close: Close of meta_sk with tok 0xf90808bd
多路径:
mptcp_alloc_mpcb: created mpcb with token 0x13ce7954 路径1
mptcp_add_sock: [token 0x13ce7954 pt 1, src_addr:10.133.247.147]58744 dst_addr:119.3
mptcp_add_sock: [token 0x13ce7954 pt 2, src_addr:0.0.0.0 dst_addr:0.0.0.0] 路径2
__mptcp_init4_subsockets:[token 0x13ce7954 pt 2 src_addr:10.68.101.39]0 dst_addr:11
root@119.3.52.83's password:
109MB.m4
mptcp_close: Close of meta_sk with tok 0x13ce7954
root@hyc_tm6ull:/home/video# mptcp_del_sock: Removing subsock tok 0x13ce7954 pt:2
mptcp_del_sock: Removing subsock tok 0x13ce7954 pt:1 state 7 is_meta? 0
mptcp_sock_destruct destroying meta-sk token 0x13ce7954
```

图11 系统单路径和多路径数据传输

Fig.11 System single path and multipath data transmission

为了进一步验证5G多路径数据传输系统可以有效实现带宽聚合,使用单路径和多路径分别上传10—150 MB的文件到云端,测试后的数据如图12所示,在传输不同大小的文件时,使用双路径传输的速率稳定在2.3 MB/s左右,单路径传输速率的速率稳定在1.2 MB/s左右,相较于使用单条5G链路,5G多路径数据传输系统实现了稳定的带宽聚合,验证了本文提出的MPTCP与5G技术相结合的多路径传输方案能够提高5G在特殊应用场景中的传输带宽、连接鲁棒性及网络服务质量。

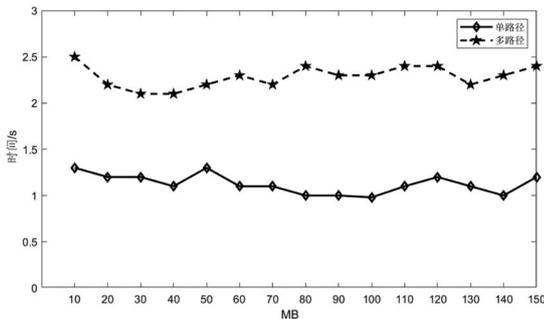


图12 单路径与多路径数据传输对比图

Fig.12 Comparison between single path and multipath data transmission

6 结论(Conclusion)

本文提出使用MPTCP协议与5G技术相结合的方案,并完成了5G多路径数据传输系统的设计。该系统实现将接收到的数据使用不同运营商SIM卡的5G链路发送到云端,实现多条5G通信链路带宽聚合。经过测试验证,本文设计的5G多路径数据传输系统在某一运营商5G信号不稳定情况下可以显著提高5G在特殊应用场景中的传输带宽、连接鲁棒性及网络服务质量。

参考文献(References)

[1] 张平,崔琪楣.第五代移动通信技术导论[M].北京:中国科学技术出版社,2021:38.

[2] PERDANA D, MUMTAZ N, BISONO Y G. Performance analysis of real-time services on 5G millimeter wave network[J]. Engineering Letters, 2021, 29(3):1151-1157.

[3] 陈园园.大规模MIMO中继传输技术的性能研究[D].南京:南京邮电大学,2018.

[4] 朱鹏飞.Massive MIMO下波束赋形技术研究[D].成都:电子科技大学,2022.

[5] 蔡玮.5G通信技术应用场景与关键技术研究[J].电子元器件与信息技术,2021,5(12):159-160.

[6] 李俊.5G 4K/8K超高清直播方案探究[J].电视技术,2022,46(1):139-141,149.

[7] 吴智广,栾桂勇,陈学辉.基于5G技术的井下铲运机远程控制设计与应用[J].现代矿业,2020,36(6):147-148,212.

[8] 刘佳乐.5G+工业互联网综述[J].物联网技术,2021,11(12):53-58.

[9] FORD A, RAICIU C, HANDLEY M, et al. TCP Extensions for Multipath Operation with Multiple Addresses[J]. Proceedings of the ACM on Programming Languages, 2013, 6824:1-64.

[10] 符发,周星,谭毓银,等.多场景的MPTCP协议性能分析研究[J].计算机工程与应用,2016,52(5):89-93,98.

[11] ALJUBAYRI M, PENG T, SHIKH-BAHAEI M. Reduce delay of multipath TCP in IoT networks[J]. Wireless Networks, 2021, 27(6):4189-4198.

[12] HUANG X, ZHANG T, DENG Z, et al. Design of moving target detection and tracking system based on cortex-A7 and openCV[J]. Traitement Du Signal, 2018, 35(1):61-73.

[13] GANESAN S K. A novel high speed serial interface in UTMI+transceivers for USB 2.0 Hub repeater operations: A proposal[J]. International Journal of Designed Objects, 2012, 6(3):73-91.

[14] PENG Q, WALID A, HWANG J, et al. Multipath TCP: Analysis, design, and implementation[J]. IEEE/ACM Transactions on networking, 2014, 24(1):596-609.

[15] 王琨,周星,周峰,等.基于MPTCP路径管理算法性能分析[J].海南大学学报(自然科学版),2017,35(3):211-218.

[16] BARRÉ S. Implementation and assessment of modern host-based multipath solutions[D]. Louvain-la-Neuve: Catholic University of Louvain, 2011.

[17] ATSUTA K, KOUYA S, SAKAMOTO N. Performance of TCP CUBIC and TCP BBR on IEEE802.11s mesh network[J]. Journal of Communications, 2020, 15(8):639-645.

作者简介:

韩彦超(1995-),男,硕士生.研究领域:嵌入式与物联网技术.

李亮(1981-),男,硕士,副高级工程师.研究领域:人工智能与智能制造,工业互联网,机电装备智能控制.

王成群(1981-),男,博士,讲师.研究领域:工业互联网, AI及其行业应用.