文章编号：1674－2869(2024)03－0310－07

基于WebSocket协议的消息安全推送系统设计与实现

徐智勇，戴祖旭*
武汉工程大学光电信息与能源工程学院、数理学院，湖北 武汉 430205

摘 要：为节约应用系统的带宽资源，提高客户端接收消息的可靠性和安全性，设计了一个独立的消息安全推送系统，提供开放的接口供应用系统调用。客户端向应用系统请求有时延风险的业务时，首先触发与推送系统的WebSocket连接，获取客户标识和密钥，然后携带标识再次与应用系统发起HTTP连接办理业务。在耗时业务处理过程中，客户端无需与应用系统保持长连接。应用系统完成业务后，将响应消息连同客户标识通过HTTP连接转发给推送系统缓存。推送系统将多位客户的响应消息打包加密成一条密文推送到客户端，客户端解密得到请求的业务结果。推送系统基于Golang语言和MySQL数据库实现，对外提供符合表述性状态转移（REST）架构的通用接口。功能测试结果表明，客户端能准确接收来自推送系统的消息，实现了预期功能。

关键词：消息推送；机密性；时延；WebSocket协议
中图分类号：TP317 **文献标志码**：A **DOI**：10.19843/j.cnki.CN42-1779/TQ.202306003

Design and implementation of a secure message feeding system based on WebSocket protocol

XU Zhiyong，*DAI Zuxu**
School of Optical Information and Energy Engineering，School of Mathematics and Physics，Wuhan Institute of Technology，Wuhan 430205，China

Abstract：To save the bandwidth resources of the application system and improve the reliability and security of messages received by clients，an independent message security feed system was designed to provide an open interface for the application system to call on. When a client requests a time-delayed risky business from the application system，the system first triggers a WebSocket connection with the feed system to obtain the client identification and key，and then carries the identification to initiate an hyper text transfer protocol connection with the application system again to handle the business. During the time-consuming business processing，the client does not need to maintain a long connection with the application system. Once the application system completes its business，it sends the response message，along with the client identification，to the feed system via an hyper text transfer protocol connection. The feed system packages and encrypts the response messages of multiple clients into a cipher text and feeds it to the clients，who decrypts it to get the requested business results. The feed system was implemented using the Golang language and MySQL database，and provides a generic interface that conforms to the representational state transfer（REST）architecture for external access. Functional tests results show that the client can accurately receive messages from the feed system and achieve the expected functions.

Keywords：message feeding；confidentiality；time delaying；WebSocket protocol

收稿日期：2023-06-06
基金项目：国家自然科学基金（61771353）
作者简介：徐智勇，硕士研究生。Email：907998253@qq.com
***通信作者**：戴祖旭，博士，教授。Email：zxdai@wit.edu.cn
引文格式：徐智勇，戴祖旭．基于WebSocket协议的消息安全推送系统设计与实现[J]．武汉工程大学学报，2024，46（3）：310-316.

消息异步传输目前有众多的应用场景，如支付场景[1-2]、协同编辑场景[3]、物联网[4-5]以及用途更广泛的各类实时监测管理系统[6-8]。以微信支付场景为例，微信支付结果必须通过回调地址通知服务端，由于支付过程存在时延现象，客户端与服务端之间基于超文本传输协议（hyper text transfer protocol，HTTP）的连接可能已断开，服务端无法主动将支付结果反馈给客户端，只能由客户端采取轮询等方式请求服务端来获取支付结果，导致出现无意义的HTTP连接，占用了系统资源[9-10]。为实现服务端主动向客户端推送消息的功能，可以在系统中增加基于传输控制协议（transmission control protocol，TCP）的WebSocket消息推送系统。

目前大多数应用将WebSocket协议集成在应用系统中，实现消息的主动推送功能，但是该协议属于长连接，过多的连接会占用系统内存以及服务器带宽，导致推送服务影响业务的正常运行[11-12]。

本文开发一套独立于应用系统的WebSocket消息推送系统，提供规范化的开发接口为各种应用系统提供消息推送服务，消息推送系统会维持与客户端的连接，应用系统在消息通知场景则不用再与客户端直接通信，而是将消息发给消息推送系统，由系统转发给客户端。此系统能提高消息实时性，降低开发和维护成本。

1　系统概要设计与需求分析

1.1　概要设计

总体设计思路是开发一个独立的消息推送系统，提供开放接口供应用系统调用。在需要消息推送的场景，客户端先与消息推送系统建立连接，获取唯一标识，再和应用系统建立与业务相关的HTTP连接并携带标识，应用系统对于同步消息直接响应，对于需主动推送的消息通过开放接口发送给消息推送系统，消息推送系统根据接口中标识发送到对应的客户端，即使客户端由于网络波动离线，也可在下次连接时补发消息。考虑到系统将长时间维护大量的连接，系统选用Golang语言来开发，其协程的特性[13]能以极小的内存维持单个连接。在推送系统中维护一个客户端连接池，用来管理所有连接到推送系统的客户端，对外提供标准的开放接口，供应用系统与客户端调用，用于消息推送功能。应用系统、客户端以及消息推送系统之间通信流程如图1所示。

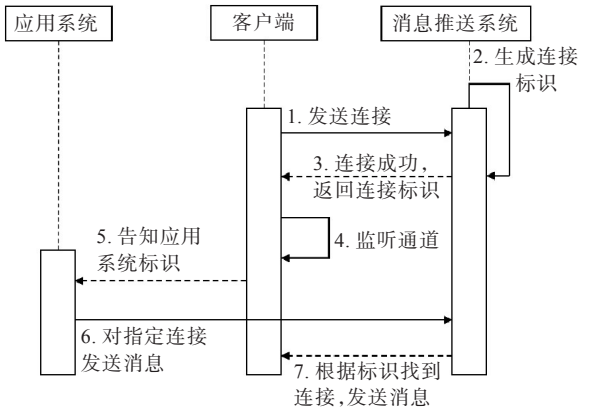

图1　应用系统、客户端以及消息推送系统通信流程

Fig. 1　Communication process of application system, client, and message push system

1.2　功能性需求分析

WebSocket消息推送系统的功能主要包括心跳检测功能、离线消息补发功能、安全加密功能和消息推送功能。(1)心跳检测功能。WebSocket消息推送系统与客户端之间使用长连接，需要定期发送报文告诉对方“我还活着”[14]。若双方心跳正常则保持连接，否则执行以下方案：(a)若服务端未检测到客户端心跳，主动关闭与客户端的连接；(b)若客户端未检测到服务端心跳，则向服务端发起新的请求。(2)离线消息补发功能。离线消息补发可以在用户重新连接系统时，将用户在离线期间错过的消息重新发送给用户，以保证用户获得完整的消息推送服务。离线消息补发的实现主要涉及消息的存储，以及消息的恢复和发送。当用户离线时，需推送的消息将被存储在关系型数据库MySQL中，并记录消息的相关信息，当客户端重新连接系统时发送给客户端。(3)安全加密功能。在消息推送系统推送一些敏感数据时，需对该数据进行加密处理，防止在传输链路中被窃取。为保障消息传输过程中数据机密性、数据完整性，系统将结合中国剩余定理进行打包加密处理。(4)消息推送功能。消息推送是该系统的核心功能。客户端需要和推送系统建立连接，并保持连接的状态。在建立连接后，推送系统将给客户端分发客户端标识或客户端自行携带标识在系统中注册，以便系统能够将消息推送给正确的客户端。系统将提供接口，用户可根据接口参数，对指定客户端标识发送消息，消息格式支持字符串、数字以及对象，消息是否成功发送以及客户端标识是否在系统中注册，系统都将返回对应状态码以及错误信息。

2 系统总体设计

2.1 架构设计

根据上述的功能性分析，消息推送系统分为公共模块以及业务模块。公共模块为系统提供基础功能，业务模块提供系统主体功能。PC端与移动端可以通过接口进行WebSocket连接，应用系统可通过接口做相应的业务处理，最后离线消息以及历史消息将存储在MySQL数据库中[15]。系统架构图如图2所示。

系统架构分为3层。访问层指的是用户可以通过什么设备接入系统。服务层由公共模块和业务模块组成，公共模块为服务提供基础功能，如日志采集模块引入日志存储，起业务溯源作用，业务模块则对应不同的功能。资源层是关系数据库，系统采用MySQL数据库存储离线消息和历史消息。

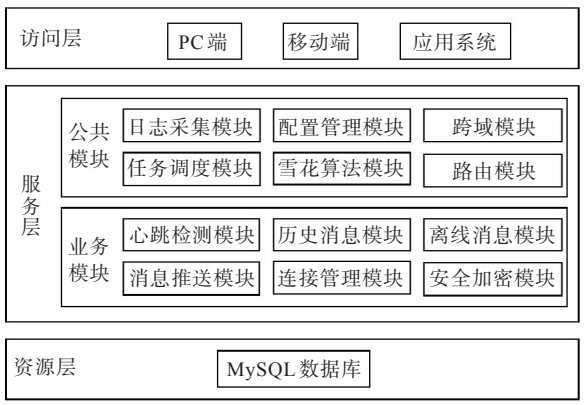

图2 系统架构图

Fig. 2 System architecture diagram

公共模块包括日志采集模块、配置管理模块、跨域模块、任务调度模块、雪花算法模块和路由模块。

(1)日志采集模块：系统引入日志库来实现便利的日志采集与管理，其完全兼容标准库记录器，将日志分为3个等级(info、warn以及error)，设置日志文件保存时间以及大小，格式化日志信息保存在文件夹中。日志采集将记录接口调用日志、数据库操作日志以及连接管理日志等。

(2)配置管理模块：支持配置统一管理，系统将解析配置文件保存在全局变量中，配置包括系统运行的端口、日志存储路径、雪花算法以及数据库配置。

(3)跨域模块：通过设置HTTP请求头中的AllowOrigins、AllowMethods、AllowHeaders以及AllowCredentials字段来快速地设置系统的跨域规则，并集成在路由中间件中。

(4)任务调度模块：为心跳检测模块提供基础功能，通过创建周期性定时器，系统堵塞地监听定时器的通知，当通道接收到定时器的通知时，便会告知心跳检测模块定时时间到。

(5)雪花算法模块：生成严格递增的随机数序列，用作客户端的唯一标识，在连接管理模块中使用。

(6)路由模块：采用Gin设计，Gin是一个HTTP Web框架，提供路由、参数校验、中间件服务等功能[16]。在路由模块中设置接口路由地址的具体方法在下文的接口设计中介绍。

2.2 数据库设计

在WebSocket消息推送系统中，离线消息和历史消息均存储到数据库中，以便用户在需要时进行查询和获取。

离线消息是指在用户离线期间，系统发送给用户的消息。离线消息需要存储在数据库中，以便用户在重新登录后可以查看。离线消息的数据库设计如表1所示。后续会根据具体的需求进行调整和优化，比如，添加更多的字段、消息类型、消息优先级、消息附件等。

表1 离线消息表字段

Tab. 1 Offline message table fields

字段	数据类型	是否主键	含义
id	bigint	是	主键
client_id	varchar (100)	否	消息接受者标识
msg	longtext	否	消息内容
send_time	datetime	否	消息发送时间
status	int (3)	否	消息的状态，已读

2.3 接口设计

合理的接口设计可以提高系统的易用性、可扩展性和可维护性，同时也可以降低开发成本和技术风险，规范的接口设计能大大减短开发者的开发周期。在WebSocket消息推送系统中，接口设计遵循表述性状态转移(representational state transfer，REST)架构，其优点是结构清晰[17]。消息推送系统接口如表2所示，其中WebSocket请求协议的接口由客户端调用，HTTP请求协议的接口由服务端调用。

消息推送系统接口参数如表3所示，客户端连接接口、连接关闭接口以及查询历史信息接口在请求路径中携带参数，发送消息接口的请求参数则以对象格式放在请求体中。

接口异常处理是保证系统稳定及可用性的重要环节。接口异常大致可分为3类：请求异常、服

表2　消息推送系统接口

Tab. 2　Message feeding system interface

请求协议	请求地址	请求方式	接口描述
WebSocket	/ws/clientID=?	GET	客户端连接接口，其请求参数clientID为选填项。 ①当客户端不携带标识进行连接请求时，系统会将其作为第一次连接的客户端，通过雪花算法生成唯一标识分配给它。 ②当客户端携带标识进行连接时，系统将查询离线消息数据库，在连接成功后，将离线消息按时间顺序补发给它。
HTTP	/api/closeClient?clientID=?	DELET	连接关闭接口，当业务完成时，应用系统可以通过该接口主动将客户端连接关闭。
HTTP	/api/sendMsgToClient	POST	发送消息接口，应用系统发起请求，请求体附带客户端标识以及需发送的消息。
HTTP	/api/queryMessage?clientID=?&startTime=?&endTime=?	GET	查询历史消息接口，可通过传递开始时间以及终止时间进行筛选，根据客户端标识查询发送给该客户端的所有消息。

表3　消息推送系统接口参数

Tab. 3　Message feeding system interface parameters

接口名称	参数名称	类型	是否必选	描述
客户端连接接口	clientID	String	可选	客户端标识
连接关闭接口	clientID	String	必选	客户端标识
发送消息接口	clientID	String	必选	客户端标识
	msg	String、Object	必选	消息内容
查询历史信息接口	clientID	String	必选	客户端标识
	startTime	String	可选	默认为当前时间的30天前
	endTime	String	可选	默认为当前时间

务器异常以及安全异常。请求异常包括请求超时、请求被拒绝、请求参数错误等，服务器异常包括业务错误、连接失败等，安全异常包括身份验证失败、权限不足等。

对于请求超时情况，客户端应设置超时时间，当服务端超时未响应时，做重发处理。

请求被拒绝情况一般发生在跨域请求场合，此时应联系管理员将域名添加进白名单内。

对于请求参数错误或业务错误情况，服务端以错误码的形式响应客户端并说明原因。常用的错误码包括请求参数错误、客户端不存在以及用户未登录等。

对于安全异常情况，要求用户发起请求时携带身份信息，若身份非法，则拒绝访问。

3　系统功能实现

3.1　心跳检测功能

心跳检测的目的是确保当前的连接都是在线的。在程序初始化时，系统创建一个协程，该协程内部创建一个周期性触发的定时器，当定时器触发时，定时器会往通道发送消息，系统持续监听该通道，当通道接收到消息时表明时间间隔周期到达，系统会给所有在线的连接发送心跳消息，以达到心跳检测的目的。其流程实现如图3所示。

3.2　安全加密功能

安全加密模块主要对信息进行加密处理，防止信息在传输链路上被窃取。系统将信息按时间分为一组，结合中国剩余定理的加密算法将一组信息打包加密成一个密文，各个用户使用独立的密钥解密出自己的明文，具有较高的效率。

定理1　质数有无穷多个。

定理2　（质数定理）设$\pi(\alpha)$为不超过α的质数个数，则

$$\lim_{\alpha\to\infty}\frac{\pi(\alpha)}{\frac{\alpha}{\lg(\alpha)}}=1$$

定理1和定理2说明质数数量无限但分布不均匀，质数分布稀疏，因此攻击者要搜索一个大质

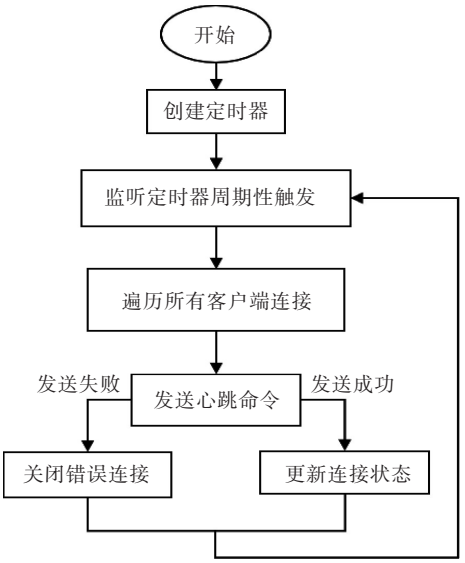

图3 心跳检测功能流程图

Fig. 3 Flowchart of heart rate detection function

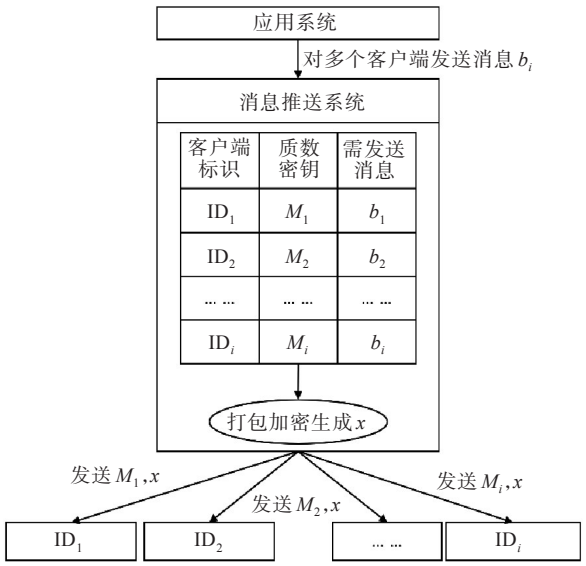

图4 加密流程图

Fig. 4 Encryption flowchart

数是很困难的。

定理3 （中国剩余定理）[18] $m_1,m_2,\cdots,m_k$ 是 k 个两两互质的正整数，$m = m_1m_2\cdots m_k$，$M_i = m/m_i$，$i = 1,2,\cdots,k$，则同余式组

$$x_1 \equiv b_1\left(\mathrm{mod}m_1\right), x_2 \equiv b_2\left(\mathrm{mod}m_2\right), \cdots, x_k \equiv b_k\left(\mathrm{mod}m_k\right)$$

的解是

$$x \equiv M_1^{-1}M_1b_1 + M_2^{-1}M_2b_2 + \cdots + M_k^{-1}M_kb_k\left(\mathrm{mod}m\right) \quad (1)$$

其中 $M_i^{-1}M_i \equiv 1\left(\mathrm{mod}m_i\right)$，$x$ 为加密消息。

根据上述定理，设计一个同时具备压缩功能和加密功能的方案。假设当前客户数量是 k，每个客户对应的推送消息是 b_i，系统为每个客户生成一个互不相同的质数 m_i 作为客户与推送服务器的共享密钥。推送服务器利用式(1)生成 x，并将 x 推送到 k 个客户端，客户端通过模密钥运算计算出对应消息，实现保密功能。推送服务器只需存储加密后的消息 x，实现压缩功能。其加密流程图如图4所示。

客户端第一次请求WebSocket消息推送系统时，系统生成一个新的质数，作为客户端的加密密钥登记保存，同时将该质数发送给客户端作为解密密钥。消息推送系统定期打包信息，利用式(1)计算出 x 发送给客户端。客户端接收到 x 后，可利用传来的密钥计算出原始信息。

3.3 消息推送功能

每当客户端发起连接时，系统将创建一个客户端对象来保存本次连接的相关信息，在创建客户端对象的同时，系统还创建一个协程监听来自该连接的消息，并对各种异常进行处理。客户端对象定义如表4所示。其中LastMsgTime属性以及LastHeartTime属性用于连接健康监测，对于长时间未收发的不活跃连接以及心跳检测未通过的连接做断开、销毁处理。

程序初始化阶段，消息推送系统创建全局客户端管理者，用于管理客户端对象。客户端管理者定义如表5所示，其中的ClientMap键为客户端标识，值为客户端对象，存储格式是哈希表。因为存储涉及多协程并发操作，所以需要给该哈希表添加读写锁。其中还有两个通道，分别处理连接、断开事件。客户端管理者所拥有的私有函数如表6所示。

表4 客户端对象定义

Tab. 4 Client object

属性名	数据类型	含义
ClientId	string	客户端标识
Socket	websocket.Conn	用户连接
ConnectTime	uint64	首次连接时间
LastMsgTime	uint64	最后一次收发消息时间
LastHeartTime	uint64	最后一次心跳检测时间

表5 客户端管理者定义

Tab. 5 Client management objects

属性名	数据类型	含义
ClientMap	map[string]Client	管理的全部连接
ClientMapLock	sync.RWMutex	读写锁
Connect	chan Client	连接通道
DisConnect	chan Client	断开通道

表6　客户端管理者的私有函数

Tab. 6　Private functions of client management object

函数名	传入参数	含义
Init	无	初始化客户端管理对象，开启一个协程用于处理心跳检测，并监听连接、断开两个通道
EventConnect	Client对象	用于处理客户端连接事件
EventDisconnect	Client对象	用于处理客户端断开事件
PingTimer	无	心跳检测函数，会开启一个定时器来对管理者哈希表中的客户端进行心跳检测
AddClient	Client对象	为clientMap添加一个客户端
GetAllClient	无	获取clientMap全部客户端
Count	无	统计clientMap中客户端的数量
DelClientByID	clientID	根据标识删除clientMap中连接
GetClientByID	clientID	根据标识获取clientMap中连接

当应用系统发起消息推送请求后，系统处理该请求的主流程如下：

（1）控制层接收到消息推送请求后，对该请求进行参数校验，并序列化为对象，若参数传递错误则直接返回错误码与错误信息。

（2）校验通过后，将该请求传递到相对应的业务逻辑层。

（3）在业务逻辑层中，系统从客户端管理对象中调用GetClientByID函数，获得相对应的客户端连接。

（4）若连接不存在，则将需推送的消息存入离线消息数据库中。若连接存在，则获取对应的客户端连接。

（5）获取到对应客户端连接后，通过内部属性Socket获取客户端地址，调用其私有函数，实现消息推送功能。

4　系统功能测试

将消息推送系统结合第三方支付系统以及商品端的应用系统进行全面的功能测试。主要测试用例为是否能完成正常的消息推送流程。

支付通知大体流程如下：客户端购买商品、应用系统对接第三方支付系统、完成订单支付、接收到订单信息以及商品信息的消息通知，如图5所示。

建立WebSocket连接后，可在浏览器页面按下F12打开检查台，在网络选项中查看连接状态以及交互信息。客户端能收到由消息推送系统推送的来自服务端的支付结果通知。

经测试，系统完成了消息推送功能，其规范的接口设计以及离线消息、心跳检测功能为系统带来了易用性以及可靠性。

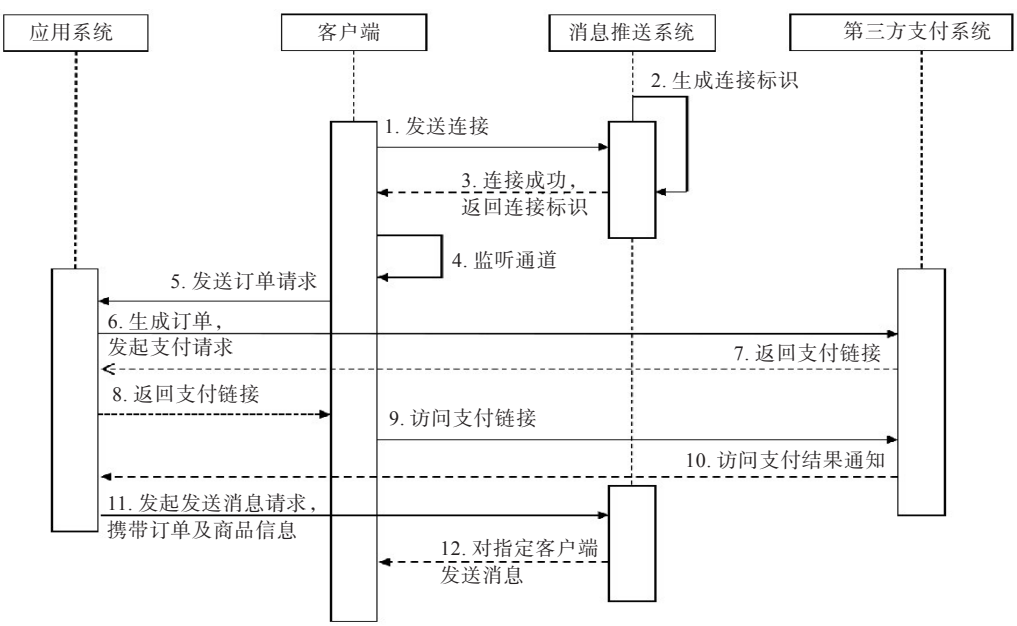

图5　支付通知流程图

Fig. 5　Flowchart of payment notification

5 结 论

本文开发了一套基于WebSocket协议且独立于应用系统的消息推送系统，可解决第三方服务通过回调地址接收的消息通知无法及时响应客户端的问题。应用系统将时延消息发送至消息推送系统，再由该系统推送至客户端，大大提高了消息推送的可靠性，保证了消息在传输链路中的安全性，同时也减少了应用系统资源的开销，并可帮助开发者缩短开发周期、降低开发和维护成本。

参考文献

[1] 陈诗雨，熊才权，董奕，等. 基于WebSocket的实时协同编辑方法及应用[J]. 软件导刊，2019，18(5)：79-83.

[2] 宇文涛，冉全. 数据传输改进方案在医疗设备监测系统的应用[J]. 武汉工程大学学报，2022，44(2)：203-208.

[3] 康庄. 灾害防御区域气象观测站服务数据监测系统设计[J]. 计算机测量与控制，2022，30(1)：86-91.

[4] 席佩瑶，陈森林，周子伟，等. 数据实时通信与可视化系统的设计与实现[J]. 自动化与仪表，2022，37(1)：55-59.

[5] 李鹏程，张文胜，郭栋，等. 基于物联网通信协议的车辆信息系统开发[J]. 计算机工程与设计，2022，43(3)：646-653.

[6] 杨帆，张辉寰. 无线楼宇安全监测系统的设计[J]. 武汉工程大学学报，2018，40(3)：310-314.

[7] 黄东栋，程莉，党晶晶. 基于物联网的环境在线监测系统研究[J]. 武汉工程大学学报，2018，40(3)：320-324.

[8] 陈思媛，易国洪，金旺. 智能停车系统中消息中间件的设计与实现[J]. 武汉工程大学学报，2020，42(2)：224-230.

[9] 曹文彬，谭新明，刘备，等. 基于事件驱动的高性能WebSocket服务器的设计与实现[J]. 计算机应用与软件，2018，35(1)：21-27.

[10] 石文涛. Html5中WebSocket协议关键技术的研究及基于WebSocket协议的实时Web通信系统的实现[D]. 南京：南京邮电大学，2014.

[11] 陈相余. 云平台下消息推送系统的设计与实现[D]. 成都：电子科技大学，2021.

[12] TIAN X L，WANG Y，ZHAO Q，et al. A miniaturized，low-cost and portable fiber Bragg grating interrogation system for remote monitoring [J]. Optik，2021，248：168054.

[13] 张嘉焰. 基于Beego框架的运动处方管理系统的设计与实现[D]. 武汉：华中科技大学，2020.

[14] 易俗，殷慧文，王闯，等. 多因素自适应心跳检测算法研究[J]. 计算机工程与应用，2017，53(24)：86-93.

[15] 周莉莉. 计算机应用软件开发技术探讨[J]. 中国设备工程，2023(4)：256-258.

[16] 肖睿. 基于Gin框架的营销活动公共类库的设计与应用[D]. 武汉：华中科技大学，2019.

[17] 肖文娟，王加胜. 基于Vue和Spring Boot的校园记录管理Web App的设计与实现[J]. 计算机应用与软件，2020，37(4)：25-30.

[18] 闵嗣鹤，严士健. 初等数论[M]. 3版. 北京：高等教育出版社，2003.

本文编辑：霍淑利

☆

(上接第309页)

[9] PARK K，PATTEN T，VINCZE M. Pix2Pose：pixel-wise coordinate regression of objects for 6D pose estimation [C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Piscataway：IEEE，2019：7668-7677.

[10] 李少飞，史泽林，庄春刚. 基于深度学习的物体点云六维位姿估计方法[J]. 计算机工程，2021，47(8)：216-223.

[11] FAN R Z，XU T B，WEI Z Z. Estimating 6D aircraft pose from keypoints and structures [J]. Remote Sensing，2021，13(4)：663.

[12] YANG X L，JIA X H，LIANG Y，et al. 6D object pose estimation in cluttered scenes from RGB images[J]. Journal of Computer Science and Technology，2022，37(3)：719-730.

[13] 王太勇，孙浩文. 基于关键点特征融合的六自由度位姿估计方法[J]. 天津大学学报(自然科学与工程技术版)，2022，55(5)：543-551.

[14] HAUGAARD R L，BUCH A G. Surfemb：dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings [C]// Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway：IEEE，2022：6749-6758.

[15] LIU S Q，WEI J S，LIU G，et al. Image classification model based on large kernel attention mechanism and relative position self-attention mechanism [J]. PeerJ Computer Science，2023，9：e1344.

[16] RONNEBERGER O，FISCHER P，BROX T. U-Net：convolutional networks for biomedical image segmentation [C]//Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015. Switzerland：Springer，2015：234-241.

本文编辑：霍淑利