Software Engineering and Applications 软件工程与应用, 2025, 14(4), 928-937
Published Online August 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.144082

Hans汉斯

基于PyQt5的IP数据特征字节抓取分析软件设计与实现

陈宝海，骆利涛，刘雪峰，林　东，陈　炳

中国人民解放军63726部队，宁夏 银川

收稿日期：2025年7月21日；录用日期：2025年8月18日；发布日期：2025年8月27日

摘　要

随着网络技术的飞速发展，网络数据包分析在网络监控、故障排查、安全审计等领域的重要性日益凸显。本文设计并实现了一款集成数据包捕获、多条件过滤、特征提取与Excel存储功能的图形化分析工具。该系统以PyQt5为界面开发框架，结合tshark抓包引擎、pyshark解析库和openpyxl数据处理库，实现了从网络接口抓取特定IP数据包、提取自定义字节范围特征，并将结果自动化存储到Excel的完整流程。系统支持IP地址、端口、帧长等多维度过滤条件，提供直观的输入验证和操作反馈，降低了网络数据分析的技术门槛。测试结果表明，该系统功能完备、操作简便、性能稳定，能够有效满足网络管理和研究中的数据提取需求，为后续数据分析提供高质量的结构化数据支持。

关键词

网络数据包，特征提取，PyQt5，Tshark，数据可视化

IP Data Feature Byte Capture and Analysis Software Based on PyQt5 Design and Implementation

Baohai Chen, Litao Luo, Xuefeng Liu, Dong Lin, Bing Chen

Unit 63726 of People's Liberation Army of China, Yinchuan Ningxia

Received: Jul. 21st, 2025; accepted: Aug. 18th, 2025; published: Aug. 27th, 2025

Abstract

With the rapid development of network technology, the importance of network packet analysis in areas such as network monitoring, troubleshooting, and security auditing is becoming increasingly

文章引用: 陈宝海, 骆利涛, 刘雪峰, 林东, 陈炳. 基于 PyQt5 的 IP 数据特征字节抓取分析软件设计与实现[J]. 软件工程与应用, 2025, 14(4): 928-937. DOI: 10.12677/sea.2025.144082

prominent. This article designs and implements a graphical analysis tool that integrates packet capture, multi condition filtering, feature extraction, and Excel storage functions. The system is developed using PyQt5 as the interface framework, combined with tshark packet capture engine, pyshark parsing library, and openpyxl data processing library, to achieve the complete process of capturing specific IP packets from network interfaces, extracting custom byte range features, and automatically storing the results in Excel. The system supports multi-dimensional filtering conditions such as IP address, port, and frame length, providing intuitive input verification and operational feedback, reducing the technical threshold for network data analysis. The test results show that the system has complete functions, easy operation, and stable performance, which can effectively meet the data extraction needs in network management and research, and provide high-quality structured data support for subsequent data analysis.

Keywords

Network Packets, Feature Extraction, PyQt5, Tshark, Data Visualization

1. 引言

在当今信息化社会，网络已成为社会运转的核心基础设施，各类网络应用的普及使得网络流量呈现指数级增长。网络数据包作为网络通信的基本单元，承载着丰富的通信细节，包括源地址、目的地址、传输内容、协议类型等关键信息。对这些数据包进行深入分析，能够为网络故障诊断、流量监控、安全防护、行为分析等提供重要依据。

传统的网络数据包分析工具主要分为两类：一类是以 Wireshark 为代表的专业工具，支持上千种协议解析和复杂过滤规则，功能强大但操作复杂，用户界面面向专业人员，需要使用者具备扎实的网络协议知识，普通用户难以快速掌握其核心功能；另一类是 Tcpdump 等命令行工具，适合脚本自动化，虽然灵活但缺乏图形化交互，数据处理和导出能力有限。在实际应用中，许多用户(如网络管理员、科研人员)并不需要掌握复杂的协议解析技术，仅需快速抓取特定数据包并提取其中的特征字节(如特定位置的标识、状态码等)进行分析。因此，开发一款兼具易用性和针对性的数据包特征提取工具具有重要的现实意义。

本研究旨在通过集成 tshark 抓包引擎、利用 pyshark 读取 pcap 文件、加载 Excel 自动化处理模块、引入输入验证机制，设计一款基于 PyQt5 的 IP 数据特征字节抓取分析图形化工具，从而简化数据包抓取与特征提取流程，让用户能够通过简单配置即可完成复杂的数据分析任务，有效提高工作效率，降低技术门槛。

2. 相关技术与工具

2.1. 图形用户界面开发技术

PyQt5 是 Qt 应用框架的 Python 绑定库，提供了丰富的 GUI 组件和跨平台支持，能够快速开发出美观且功能完备的桌面应用[1]。本系统利用 PyQt5 的 QMainWindow、QGroupBox、QGridLayout 等组件构建主界面，通过 QPushButton、QLineEdit、QComboBox 等实现用户交互，使用样式表定制界面风格，确保操作的直观性和视觉的友好性。

2.2. 网络数据包捕获与解析技术

作为本系统开发设计最为关键的 tshark、pyshark 网络数据包捕获与解析技术，相比较 Scapy 等其他库具备 C 语言底层的高性能、Wireshark 生态的协议覆盖、成熟的跨平台部署等优势，尤其适合高效处理大规模数据包、解析复杂协议或快速开发等企业级应用场景。

2.2.1. Tshark 抓包引擎

tshark 是 Wireshark 的命令行版本，基于 libpcap/WinPcap 库，支持网络接口监听、数据包过滤和存储[2]。本系统通过 subprocess 模块调用 tshark 进程，动态构建包含接口、过滤条件、输出文件的命令参数，实现后台抓包并隐藏控制台窗口。

2.2.2. Pyshark 解析库

pyshark 是 tshark 的 Python 封装库，能够读取 pcap 文件并解析数据包的详细信息[3]。系统利用 pyshark 的 FileCapture 类读取 pcap 文件，获取原始帧数据后进行十六进制到二进制的转换，为特征提取奠定基础。

2.3. 数据存储与处理技术

openpyxl 是 Python 中用于操作 Excel 2010+ xlsx/xlsm 文件的库，支持工作表创建、单元格读写、格式设置等功能[4]。系统通过 openpyxl 实现 Excel 文件的动态管理，包括新建文件、打开现有文件、指定列写入数据、设置列名等操作，为分析结果提供结构化存储方案。

2.4. 系统辅助技术

psutil 库：跨平台系统监控库，用于获取网络接口信息，筛选物理网卡并排除虚拟接口；
正则表达式：通过 re 模块实现 IP 地址、端口号、字节范围等输入的格式验证；
进程管理：使用 subprocess 模块创建和控制 tshark 进程，实现抓包的启动与停止；
文件系统操作：通过 os 模块实现路径处理、文件存在性判断和文件打开操作。

3. 系统总体设计

系统采用分层架构设计，自下而上分为数据层、核心功能层和界面层，各层之间通过接口实现数据交互[5]。这种分层架构实现了关注点分离，便于各模块的独立开发、测试和维护，同时为系统的扩展提供了灵活性，具体架构如图 1 所示。

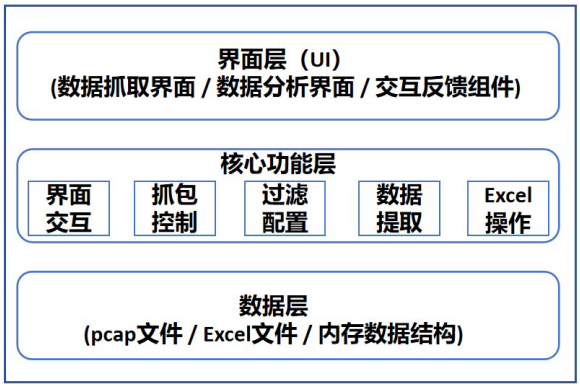

Figure 1. System architecture diagram
图 1. 系统架构图

3.1. 界面层

负责用户交互，包括数据抓取区域和数据分析区域的控件布局、事件响应和视觉反馈，软件界面如图 2 所示：

Figure 2. Software interface diagram
图 2. 软件界面图

3.2. 核心功能层

3.2.1. 界面交互模块

功能：提供图形化用户界面，实现用户操作的接收和结果的展示；

组成：数据抓取区域(网络接口选择、过滤条件输入、抓包按钮)和数据分析区域(文件选择、提取参数配置、分析按钮)；

接口：与抓包控制模块、数据提取模块和 Excel 操作模块交互，传递用户指令和参数。

3.2.2. 抓包控制模块

功能：控制数据包的捕获过程，包括启动抓包、停止抓包和管理抓包文件；

组成：进程管理单元(创建/终止 tshark 进程)、文件命名单元(生成带时间戳的 pcap 文件名)；

接口：接收界面模块的启动/停止指令，从过滤配置模块获取过滤条件，输出 pcap 文件路径。

3.2.3. 过滤配置模块

功能：解析用户输入的过滤参数，生成 tshark 可识别的过滤表达式；

组成：参数解析单元(提取 IP、端口、帧长)、表达式生成单元(组合过滤条件)；

接口：接收界面模块的过滤参数，向抓包控制模块输出过滤表达式。

3.2.4. 数据提取模块

功能：解析 pcap 文件，提取指定范围的字节特征并进行进制转换；

组成：文件读取单元(加载 pcap 文件)、数据转换单元(十六进制转二进制)、特征提取单元(截取指定范围)；

接口：接收界面模块的提取参数(文件路径、字节范围、进制)，向 Excel 操作模块输出提取结果。

3.2.5. Excel 操作模块

功能：管理 Excel 文件的创建、更新和打开；

组成：文件管理单元(创建/打开/保存文件)、数据写入单元(写入特征数据和列名)；

接口：接收数据提取模块的结果数据，接收界面模块的文件操作指令。

3.3. 数据层

负责数据的持久化存储和内存管理，包括 pcap 抓包文件、Excel 分析结果文件和内存中的临时数据结构。

4. 系统详细设计与实现

4.1. 界面交互模块实现

界面模块采用 PyQt5 的 QMainWindow 作为主窗口，通过 QHBoxLayout 实现左右分栏布局，左侧为数据抓取区域，右侧为数据分析区域，整体风格简洁直观，符合用户操作习惯。

4.1.1. 主界面布局

主窗口设置：设置窗口标题为“IP 数据特征字节抓取分析软件”，尺寸为 800 × 500 像素，背景色为浅灰色(#f5f5f5)，通过样式表定制组件外观；

分栏布局：使用 QHBoxLayout 作为主布局，左右各添加一个 QGroupBo 容器，比例为 1:1，间距 20 像素，边距 15 像素，实现两个功能区域的隔离与平衡。

4.1.2. 数据抓取区域设计

标题：QLabel 组件，文本为“IP 数据特征字节抓取”，居中对齐，字体加粗；

网络接口选择：QComboBox 组件，通过 get_physical_interfaces ()方法填充物理网络接口(排除虚拟接口如 VMware、Loopback)；

过滤条件输入：包含 5 个 QLineEdit 组件，分别用于输入帧长、源 IP、目的 IP、源端口、目的端口，并绑定输入验证函数；

控制按钮：QPushButton 组件“开始”和“停止”，分别绑定 start_capture ()和 stop_capture ()方法，初始状态“停止”按钮禁用；按钮样式通过样式表定制，绿色表示开始，红色表示停止，并设置悬停效果。

4.1.3. 数据分析区域设计

标题：QLabel 组件，文本为“IP 数据特征字节分析”，样式与左侧标题一致；

文件选择：QComboBox 组件，通过 update_file_list ()方法加载程序目录下的 pcap 文件，按修改时间排序；

列号选择：QComboBox 组件，包含 A-H 选项；

列名输入：QLineEdit 组件，用于自定义 Excel 列名；

字节范围输入：QLineEdit 组件，用于指定提取的字节范围(如“5~12”)；

数据帧数显示：QLineEdit 组件(只读)，显示提取的数据包数量；

显示进制选择：QComboBox 组件，包含“二进制”和“十进制”选项；

操作按钮：QPushButton 组件“分析”和“打开”，分别绑定 extract_data ()和 open_excel ()方法，样

式为蓝色(分析)和紫色(打开)。

4.1.4. 输入验证实现

IP 地址验证：使用正则表达式匹配 IPv4 地址格式(xxx.xxx.xxx.xxx，每个段 0~255)；

端口验证：检查输入是否为 0~65,535 的整数；

帧长验证：检查输入是否为 1~9600 的整数或空值(空表示无限制)；

字节范围验证：检查输入是否为单个 1~9600 的整数或“start-end”格式的范围(start ≤ end)。

输入验证确保了用户输入的合法性，减少了后续处理的错误概率，并通过边框颜色和背景色提供视觉反馈(绿色表示有效，红色表示无效)。

4.2. 抓包控制模块实现

抓包控制模块负责管理 tshark 进程的生命周期，实现数据包的捕获与存储，核心功能包括启动抓包、停止抓包和进程管理[6]。

4.2.1. 启动抓包

start_capture()方法实现抓包的启动。

输入验证：调用 validate_inputs ()方法检查所有过滤参数的合法性，若不合法则弹出警告对话框；

生成文件名：以当前时间戳(如“capture_20250725_200601.pcap”)生成 pcap 文件名，确保唯一性并便于追溯；

构建 tshark 命令：根据用户输入构建命令参数；

启动 tshark 进程：使用 subprocess. Popen ()启动 tshark 进程，设置启动参数隐藏控制台窗口(Windows 平台)；

更新界面状态：禁用“开始”按钮和接口/帧长输入框，启用“停止”按钮，弹出信息对话框提示抓包开始和文件路径。

4.2.2. 停止抓包

stop_capture ()方法实现抓包的停止。

终止进程：调用 terminate ()方法终止 tshark 进程，并等待进程结束：

恢复界面状态：启用“开始”按钮和接口/帧长输入框，禁用“停止”按钮；

更新文件列表：调用 update_file_list ()方法刷新 pcap 文件列表，使新捕获的文件可见；

提示信息：弹出信息对话框提示抓包停止和文件保存路径。

4.2.3. 进程管理

模块通过 self.capture_process 变量跟踪 tshark 进程状态，确保同一时间只有一个抓包进程运行。在启动新抓包前检查进程是否存在，在程序退出时确保进程已终止，避免资源泄漏。

4.3. 过滤配置模块实现

过滤配置模块负责解析用户输入的过滤参数，生成 tshark 兼容的过滤表达式，支持的过滤条件包括源 IP、目的 IP、源端口、目的端口。

4.4. 数据提取模块实现

数据提取模块负责解析 pcap 文件，提取指定范围的字节特征并进行格式转换，核心方法为 extract_data ()。

参数解析与验证

获取参数：从界面组件提取 pcap 文件路径、字节范围、列号、列名和显示进制；

验证字节范围：解析“start-end”格式的范围或单个数值，检查是否在 1~9600 范围内且 start ≤ end [7]。

解析 pcap 文件

使用 pyshark.FileCapture 加载 pcap 文件，遍历每个数据包并提取原始数据；原始数据通常以带分隔符的字符串形式存在(如“00:1a:2b:3c:...”)，需要处理为纯十六进制字符串后转换为字节流。

特征提取与转换

将字节流转换为二进制字符串，截取指定范围的比特，并根据用户选择转换为十进制(如需要) [8]。

4.5. Excel 操作模块实现

Excel 操作模块负责管理 Excel 文件的创建、更新和打开，使用 openpyxl 库实现。

文件管理

根据 pcap 文件名生成对应的 Excel 文件名(如“capture_20231001_153045.xlsx”)，检查文件是否存在并进行相应处理。

数据写入

将提取的特征数据写入指定列，并设置列名；处理完成后保存文件。

打开 Excel 文件

调用系统默认程序打开生成的 Excel 文件，兼容不同操作系统。

5. 系统测试与结果分析

5.1. 测试环境

5.1.1. 硬件环境

处理器：Intel Core i5-10400F 2.9 GHz

内存：16GB DDR4 2666MHz

网络接口：Realtek PCIe GBE Family Controller (千兆以太网)

存储：512 GB NVMe SSD

5.1.2. 软件环境

操作系统：Windows 10 专业版 21H2

Python 版本：3.8.10

依赖库：PyQt5 5.15.4、pyshark 0.4.3、openpyxl 3.0.9、psutil 5.9.0

tshark 版本：3.6.7 (Wireshark 3.6.7 捆绑版本)

测试网络：实验室局域网(192.168.1.0/24)，包含 1 台网关、10 台主机和 2 台网络打印机。

5.2. 功能测试

功能测试采用黑盒测试方法，验证系统各模块的功能正确性，主要测试用例及结果如下：

5.2.1. 数据抓取功能测试

如表 1，测试结果表明，数据抓取模块能够正确响应用户指令，根据过滤条件捕获符合要求的数据包，进程管理功能正常。

Table 1. Data capture function test
表 1. 数据抓取功能测试

测试用例	输入参数	预期结果	实际结果	测试结论
TC-G01	接口：以太网；无过滤条件	生成 pcap 文件，包含所有流经接口的数据包	符合预期，文件包含 ARP、ICMP、TCP、UDP 等多种数据包	通过
TC-G02	接口：以太网；源 IP：192.168.1.100	pcap 文件仅包含源 IP 为 192.168.1.100 的数据包	符合预期，过滤掉其他源 IP 的数据包	通过
TC-G03	接口：以太网；目的端口：80	pcap 文件仅包含目的端口为 80 的 TCP 数据包	符合预期，仅捕获 HTTP 流量	通过
TC-G04	接口：以太网；帧长：100	仅捕获长度 ≤100 字节的数据包	符合预期，长数据包被截断为 100 字节	通过
TC-G05	接口：以太网；源 IP：192.168.1.100；目的端口：443	pcap 文件包含源 IP 为 192.168.1.100 且目的端口为 443 的数据包	符合预期，多条件过滤有效	通过
TC-G06	点击“停止”按钮	抓包进程终止，pcap 文件可正常打开	符合预期，tshark 进程被成功终止，文件完整	通过

5.2.2. 数据分析功能测试

Table 2. Data analysis function test
表 2. 数据分析功能测试

测试用例	输入参数	预期结果	实际结果	测试结论
TC-A01	pcap 文件：TC-G01 生成；字节范围：1~16；进制：二进制	Excel 文件 A 列包含每个数据包前 16 位的二进制值	符合预期，数据准确无误	通过
TC-A02	pcap 文件：TC-G02 生成；字节范围：17~32；进制：十进制；列号：B；列名：特征值	Excel 文件 B 列第 1 行为“特征值”，其余行为 17~32 位的十进制值	符合预期，列名和数据正确	通过
TC-A03	选择不存在的 pcap 文件	弹出错误提示“没有可用的 pcap 文件”	符合预期，错误处理正确	通过
TC-A04	字节范围：9601~9602	弹出错误提示“位数范围必须在 1~9600 之间”	符合预期，输入验证有效	通过
TC-A05	点击“打开”按钮	自动启动 Excel 并打开生成的文件	符合预期，文件关联正确	通过

如表 2，测试结果表明，数据分析模块能够正确解析 pcap 文件，提取指定范围的特征并按要求写入 Excel，错误处理机制有效。

5.2.3. 输入验证功能测试

Table 3. Input verification function test
表 3. 输入验证功能测试

测试用例	输入内容	预期结果	实际结果	测试结论
TC-V01	源 IP：192.168.1.256	输入框显示红色边框，提示格式错误	符合预期，验证有效	通过
TC-V02	目的端口：65536	输入框显示红色边框，提示格式错误	符合预期，验证有效	通过
TC-V03	帧长：abc	输入框显示红色边框，提示格式错误	符合预期，验证有效	通过
TC-V04	字节范围：5~3	输入框显示红色边框，提示格式错误	符合预期，验证有效	通过

如表 3 所示，测试结果表明，输入验证模块能够实时检测并反馈不合法输入，有效防止错误参数进入后续处理流程。

5.3. 性能测试

性能测试评估系统在不同负载下的处理效率，主要测试指标为抓包速率和数据分析耗时。

5.3.1. 抓包速率测试

在无过滤条件下，连续抓包 10 分钟，记录捕获的数据包数量和文件大小，结果如表 4 所示：

Table 4. Packet capture rate test
表 4. 抓包速率测试

测试时长	捕获数据包数量	pcap 文件大小	平均速率(包/秒)	结论
10 分钟	28,560	45.2 MB	47.6	抓包过程稳定，无丢包现象

使用 Wireshark 同时抓包作为基准，对比发现本系统的抓包速率与 Wireshark 基本一致(偏差 < 2%)，表明 tshark 集成效果良好。

5.3.2. 数据分析耗时测试

选取不同大小的 pcap 文件，测试数据分析(提取特征并写入 Excel)的耗时，结果如表 5 所示：

Table 5. Time consumption testing for data analysis
表 5. 数据分析耗时测试

pcap 文件大小	数据包数量	字节范围	分析耗时	结论
1.2 MB	1000	1~32	2.8 秒	处理快速，无明显延迟
12.5 MB	10,000	1~32	27.6 秒	处理稳定，耗时与数据包数量近似线性相关
63.8 MB	50,000	1~32	135.2 秒	处理完成，内存占用稳定在 80~100 MB
127.5 MB	100,000	1~32	289.7 秒	处理完成，无崩溃或数据丢失

测试结果表明，系统处理效率随数据包数量增加近似线性增长，在处理 10 万个数据包时仍能保持稳定，满足中小规模网络分析需求。

6. 结论

本文设计并实现了一款基于 PyQt5 的 IP 数据特征字节抓取与分析系统，通过集成 tshark、pyshark 和 openpyxl 等工具，实现了从网络数据包捕获到特征提取、结构化存储的完整解决方案。

系统的主要成果包括设计了左右分栏的图形化界面，将数据抓取和数据分析功能有机整合，操作流程清晰；实现了基于 IP 地址、端口、帧长的多条件过滤，提高了数据包捕获的针对性；开发了自定义字节范围提取功能，支持二进制和十进制输出，满足不同分析需求；集成 Excel 自动化处理，实现了分析结果的结构化存储和便捷查看；通过输入验证和错误处理机制，提升了系统的健壮性和用户体验。

测试结果表明，系统功能完备、操作简便、性能稳定，能够有效提高网络数据包首部和载荷中地址、协议号、标志位、校验和等特征字节提取分析的效率。通过网络数据包原始、底层、基础、关键信息的提取分析，为网络管理、性能优化、安全监控和协议研究提供有力辅助支撑。

参考文献

[1] 陈涛. PyQt5 从入门到实践[M]. 北京: 清华大学出版社, 2020.
[2] tshark(1) Manual Page. https://www.wireshark.org/docs/man-pages/tshark.html
[3] Krammer, E. (2023) Pyshark Documentation. https://kiminewt.github.io/pyshark/
[4] Openpyxl Development Team (2023) Openpyxl Documentation. https://openpyxl.readthedocs.io/
[5] 李强, 刘伟. 基于 Python 的协议分析软件设计与实现[J]. 软件导刊, 2019, 18(5): 45-49.
[6] 王磊, 张建华. 基于 PyQt5 的网络数据包捕获与分析工具设计[J]. 计算机工程与应用, 2020, 56(12): 109-114.
[7] 陈明, 周涛. 基于机器学习的网络流量特征选择方法研究[J]. 通信学报, 2021, 42(3): 78-87.
[8] 赵阳. 网络协议逆向分析中的特征字段提取技术[J]. 信息安全研究, 2022, 8(6): 112-119.