

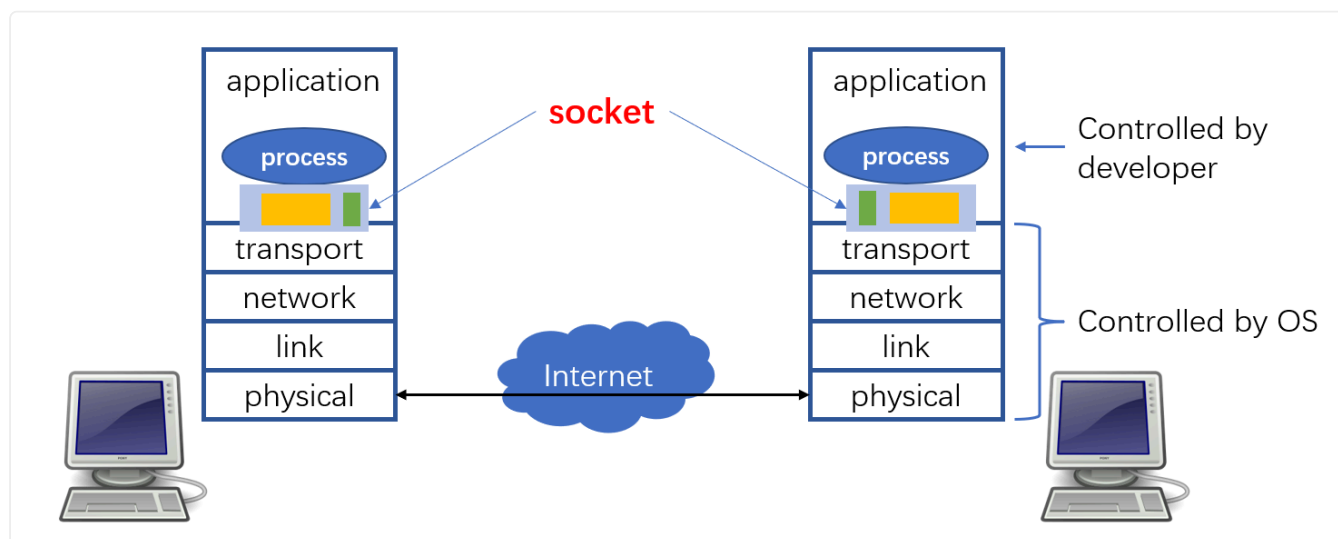
Lab 3: Socket编程

一、实验目标

1. 通过 Python socket 编程建立 TCP 连接，掌握使用 Socket 创建网络连接的基本方法；
2. 学习如何发送自定义的应用层协议报文，了解应用层协议的构建和传输过程；
3. 使用 Wireshark 插件识别和观察自定义的应用层协议报文，探索网络工具的更多用法；
4. 深化对TCP连接、应用层协议和网络通信过程的理解，提高网络编程和调试技能。

二、实验原理

套接字（Socket）是应用程序与网络之间的一种标准通信接口，用于建立连接、发送和接收数据等操作。从应用程序的角度看，可以将它视为使用网络服务的门户；从网络的角度看，可以将它视为一个通信端点。Socket 允许两个进程通过网络进行通信，无论它们运行在同一个计算机还是不同的计算机上。



socket示意图

使用 Socket 编程通常有以下步骤：

1. 创建 Socket：首先需要创建一个 Socket 对象，在创建之初就需要指定 Socket 使用的传输协议类型，例如 TCP、UDP等。

2. 绑定 Socket：将 Socket 对象绑定到一个特定的地址和端口，相当于告诉网络协议栈自己的“门牌号”，其他程序就可以通过该地址和端口与指定的 Socket 通信。
3. 监听端口：服务器端程序需要开始监听指定端口，等待响应客户端发送的连接请求或数据。
4. 建立连接：在有连接协议中，客户端程序需要尝试连接到服务器的 Socket 地址。
5. 发送和接收数据：客户端和服务端通过 Socket 发送和接收数据，可以是文本、文件、图像或任何其他类型的数据。
6. 关闭连接：通信结束后，有连接的协议应当关闭连接以释放资源。

Python 的 socket 库用法可参见[官方文档](#)。

三、实验内容

3.1 实验要求

1. 使用 Python TCP 套接字编程，实现客户端与服务端，建立 TCP 连接；
2. 通过建立起的 TCP 连接发送自定义的应用层协议 FDUnet 报文；
3. 使用 Wireshark 抓取自制客户端和服务端通信过程，并识别出 FDUnet 协议。
4. **提交内容：**提交相关代码。在报告中对代码进行简要解释，并包含以下运行效果截图：
 1. 服务器分别对不同输入返回 200 OK 和 501 Not Implemented；
 2. 一个服务器与一个客户端建立连接，并连续发送多条消息；
 3. 一个服务器同时与两个客户端建立连接，并发送消息；
 4. 任一客户端输入 #quit 后，双方关闭连接，但服务器继续服务其他客户端；
 5. 选择 Wireshark 抓取到的任意一对 FDUnet 协议请求和应答报文，展示解析出的报文内容。

3.2 运行逻辑

客户端：给服务端发送字符串，并打印收到的来自服务端的字符串。

服务端：监听端口，接收并打印客户端发来的字符串；将字符串中的大写字母转化为小写字母，小写字母转化为大写字母，其他字符不变，然后返回给客户端。

当客户端发送“#quit”时，双方关闭连接，客户端直接结束运行，服务器端继续服务其他连接。

编程时需要注意以下细节：

•

- 为了更好地观察 Socket 运行的过程，本实验要求使用 Python 的 socket 库，禁止使用抽象级别更高的 socketserver 库；
- 为了和下发的 Wireshark 自定义插件配合，服务器监听的 TCP 端口号需要使用 12000，否则要自行更改插件；
- 双方需要支持在一次连接中进行多次通信，不能每次通信都建立新的连接；
- 一个服务端需要支持同时与多个客户端建立连接并通信，为此你可能需要使用 Python 的 threading 等多线程库。

3.3 FDUnet 协议

FDUnet 是我们自定义的一个应用层协议。通过生成和发送 FDUnet 报文，同学们会发现，任意协议只要获得了通信双方的认可，就能被用来进行信息交流，甚至 Wireshark 等第三方工具也可以解析报文的内容。FDUnet 协议的报文具体格式如下：

客户端：

- 方法：POST
- URL：/
- 版本：1.0
- 首部行：Date: GMT时间（返回当前时间，格式仿照课本上 HTTP 报文时间格式：Thu, 15 Feb 2023 15:44:04 GMT）
- 实体：想要转化的字符串

方法	sp	URL	sp	版本	cr	lf
首部字段名:	sp	值	cr	lf		
首部字段名:	sp	值	cr	lf		
cr	lf					
实体内容						

客户端报文格式

服务器端：

- 版本：1.0

- 状态码和短语：
 - 如果客户端请求行正确，且实体内容中有字符串。则返回 200 OK，并在实体内容中返回转换后的字符串。
 - 否则，返回 501 Not Implemented，实体可以为空。
- 首部字段名: Date: GMT时间（返回当前时间，格式仿照课本上 HTTP 报文时间格式：Mon, 11 Dec 2024 15:44:04 GMT）

版本	sp	状态码	sp	短语	cr	lf
首部字段名:		sp	值	cr	lf	
首部字段名:		sp	值	cr	lf	
cr	lf					
实体内容						

服务器端报文格式

3.4 使用 Wireshark 解析 FDU-net

Wireshark 允许用户编写自定义的协议解析器，以便正确解析和显示自定义协议的数据包。解析器一般为 LUA 插件的形式，eLearning 已经上传了编写好的 FDU-net.lua 文件。请按以下步骤进行操作：

对 Wireshark 文件夹下没有 init.lua 的新版本：直接将 FDU-net.lua 复制到 Wireshark/plugins文件夹下，重启 Wireshark 即可从解析表里找到 FDU-net。

对 Wireshark 文件夹下有 init.lua 的旧版本：

在 Wireshark安装目录下，找到 init.lua 文件，并在最后一行添加：

```
if not running_superuser or run_user_scripts_when_superuser then
    dofile(DATA_DIR.."console.lua")
end
dofile(DATA_DIR.."FDU-net.lua")
--dofile(DATA_DIR.."dtd_gen.lua")
```

将 FDU-net.lua 文件复制到同一文件夹下，如下图：

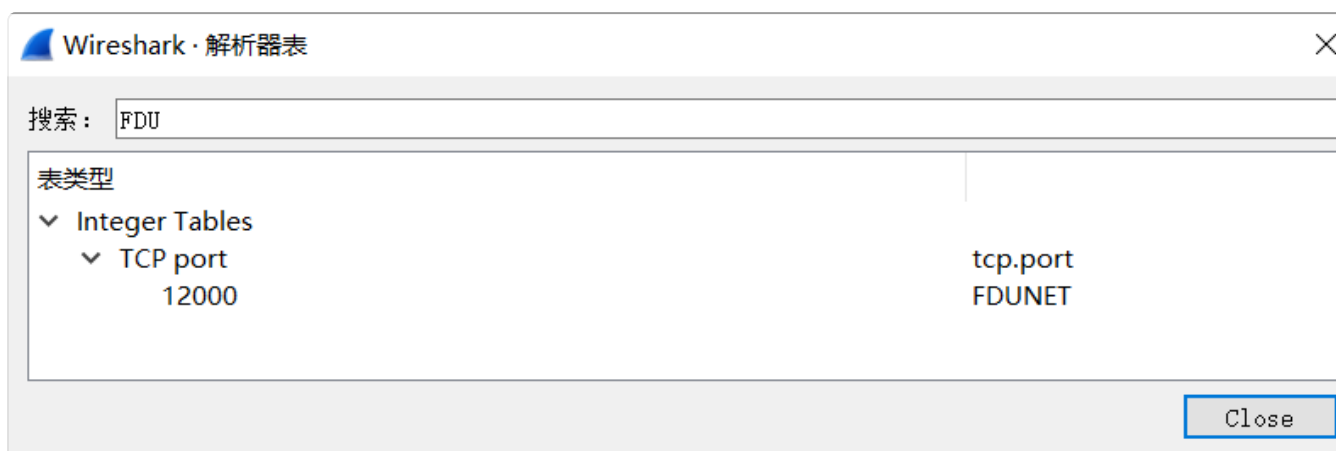
新加卷 (F:) > Wireshark

名称	修改日期	类型	大小
extcap.html	2023/8/24 3:03	Microsoft Edge ...	9 KB
faq.html	2023/8/24 3:08	Microsoft Edge ...	83 KB
FDU-net.lua	2021/10/27 18:52	LUA 文件	1 KB
glib-2.0-0.dll	2023/8/24 3:09	应用程序扩展	1,239 KB
gmodule-2.0-0.dll	2023/8/24 3:09	应用程序扩展	30 KB
gpl-2.0-standalone.html	2023/8/24 3:03	Microsoft Edge ...	21 KB
gthread-2.0-0.dll	2023/8/24 3:09	应用程序扩展	22 KB
iconv-2.dll	2023/8/24 3:09	应用程序扩展	1,062 KB
init.lua	2023/10/10 0:02	LUA 文件	18 KB
intl-8.dll	2023/8/24 3:09	应用程序扩展	96 KB

放置 FDU-net.lua 的位置

重启 Wireshark。

如果插件正确安装，可以在 Wireshark 的“视图-内部-解析器表 (Dissector Tables)”中查找到 “FDUnet” 协议，如下图。助教提供的插件使用的端口号是12000，如果想要使用其他端口可以自行修改插件。



安装插件后的解析器表

由于客户端和服务端都运行在本地，捕获分组时需要选择回环网口（Adapter for loopback traffic capture）。如果操作正确，Wireshark 应当能捕获并解析 FDUnet 协议的报文，效果如下图：

fdunet							
No.	Time	Source	Destination	Protocol	Length	Info	
635	2024-10-07 17:14:08.032716	127.0.0.1	127.0.0.1	FDUnet	107	62649 → 12000	[P
637	2024-10-07 17:14:08.033318	127.0.0.1	127.0.0.1	FDUnet	107	12000 → 62649	[P
<div>> Frame 635: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface \Device\NPF_Loopback</div> <div>> Null/Loopback</div> <div>> Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)</div> <div>> Transmission Control Protocol, Src Port: 62649, Dst Port: 12000, Seq: 1, Ack: 1, Len: 63</div> <div>▼ FDUnet protocol, used in computer net course</div> <div>FirstLine: POST / 1.0\r\n</div> <div>▼ [header]</div> <div>Date: Mon, 07 Oct 2024 09:14:08 GMT</div> <div>Data: Hello World!</div>							

Wireshark 捕获到的 FDUnet 分组

[Previous](#)

Lab 2: 应用层协议模拟与观察

[Next](#)

附录A: Wireshark 教程