**1. 网络层实验：深入探索网络层工作原理：**

1. 实验的目标：

使用工具深入理解网络层的工作原理。

掌握traceroute, ping, arp, netstat, ipconfig/ifconfig, nslookup, route, wireshark等工具的基本使用。

通过实际操作了解MAC地址、IP地址、网络配置、数据包传输、DNS解析、路由选择等。

**2. 实验的内容：**

2.1. 获取和理解网络信息：

**操作**: ipconfig (Windows) 或 ifconfig (macOS/Linux)。

**预期输出**: 显示 IP 地址、子网掩码、默认网关等信息。

**学生理解点**: 理解网络配置的基础元素，如 IP 地址、子网掩码、默认网关。

2.2. 使用 Ping 命令

**操作**: ping 8.8.8.8。

**预期输出**: 显示到 8.8.8.8 的 ping 回应时间。

**学生理解点**: 利用 ping 命令检测网络连通性和网络延迟。

2.3. 使用 ARP 命令

**操作**: arp -a。

**预期输出**: 显示本地网络内设备的 IP 和 MAC 地址。

**学生理解点:** 理解 ARP 协议及其在本地网络中映射 IP 地址到物理地址的作用。

2.4. 使用 Traceroute/Tracert 命令

**操作:** tracert 8.8.8.8 (Windows) 或 traceroute 8.8.8.8 (macOS/Linux)。

**预期输出**: 显示到目的地址的所有路径节点。

**学生理解点**: 了解网络数据包传输路径及其所经过的路由器。

2.5 使用 Netstat 命令

**操作:** netstat -an，netstat -nr

**预期输出:** 显示网络连接、路由表等信息。

**学生理解点:** 理解网络端口的使用情况和网络连接的状态

2.6. NSLookup 的应用

**操作:** nslookup www.guet.edu.cn。

**预期输出:** 桂林电子科技大学网站的 IP 地址及其他 DNS 记录信息。

**学生理解点:** 了解如何通过域名查找 IP 地址和 DNS 解析过程。

2.7 使用 Wireshark 监控网络流量

**目标:** 使用 Wireshark 捕捉和分析网络流量。

**操作:** 打开 Wireshark，选择一个网络接口开始捕捉数据，进行网络活动，停止捕捉，分析数据包。

**预期输出:** 捕捉到的网络数据包和相关信息。

**学生理解点:**学会如何使用 Wireshark 捕捉网络流量。理解数据包结构和网络活动分析。

2.8. Python 脚本执行网络命令

**目标:** 使用 Python 脚本来调用网络命令（如 ping, nslookup）。

**操作:** 编写 Python 脚本，调用网络命令，并处理输出结果。

**预期输出:** 脚本执行命令的输出结果。

**学生理解点:** 学会如何使用 Python 进行网络操作。了解编程语言在网络任务自动化中的应用。

**3. 实验用到的工具：**

计算机： 安装有操作系统，可以运行上述命令和工具。

网络连接： 用于连接到互联网和局域网。

笔记本或文档编辑软件： 用于记录和分析实验结果。