

# 实验十六：网络测试工具

## 一、实验目的

- 1、了解常用的网络测试工具；
- 2、掌握使用 Ping、tracert、mtr 工具进行网络测试；
- 3、掌握使用 tcpdump、iperf3 工具进行网络测试。

## 二、实验学时

2 学时

## 三、实验类型

设计性

## 四、实验需求

### 1、硬件

每人配备计算机 1 台。

### 2、软件

Windows 操作系统，安装 Oracle VM VirtualBox 软件，安装 Mobaxterm 软件。

### 3、网络

本地主机与虚拟机能够访问互联网，不使用 DHCP 服务。

### 4、工具

无。

## 五、实验任务

- 1、完成使用 Ping、Tracert 进行网络连通测试；
- 2、完成使用 mtr 进行网络性能测试；
- 3、完成使用 tcpdump、iperf3 进行网络通信分析。

## 六、实验环境

- 1、本实验需要 VM 1 台；
- 2、本实验 VM 配置信息如下表所示；

虚拟机配置	操作系统配置
虚拟机名称：VM-Lab-16-Task-01-172.20.1.29 内存：1GB CPU：1 颗，1 核心	主机名：Lab-16-Task-02 IP 地址：172.20.1.29 子网掩码：255.255.255.0

虚拟磁盘：20GB 网卡：1 块，桥接	网关：172.20.1.1 DNS：8.8.8.8
------------------------	------------------------------

3、本实验拓扑图。

无

4、本实验操作演示视频。

无

## 七、实验内容及步骤

### 1、使用 Ping 进行连通性测试

ping 工具用于确定本地主机是否能与另一台主机成功交换(发送与接收)数据包，再根据返回的信息，就可以推断 TCP/IP 参数是否设置正确，以及运行是否正常、网络是否通畅等。

---

# 测试与 www.baidu.com 是否连通

```
[root@Lab-16-Task-01 ~]# ping www.baidu.com
```

# 使用 ping -c 测试对 www.baidu.com 接收包的次数,接收 2 次

```
[root@Lab-16-Task-01 ~]# ping -c 2 www.baidu.com
```

# 使用 ping -w 测试对 www.baidu.com 的连通性，测试时间 5 秒退出

```
[root@Lab-16-Task-01 ~]# ping -w 5 www.baidu.com
```

# 使用 ping -R 记录路由信息

```
[root@Lab-16-Task-01 ~]# ping -R www.baidu.com
```

# 使用 ping -r 命令来忽略路由表，直接发送包给主机地址

```
[root@Lab-16-Task-01 ~]# ping -r www.baidu.com
```

# 使用 ping -q 命令对 www.baidu.com 进行测试，只显示结果，忽略过程

```
[root@Lab-16-Task-01 ~]# ping -q www.baidu.com
```

# 使用 ping -i 测试对 www.baidu.coms 收发信息的时间间隔，每 3 秒进行一次

```
[root@Lab-16-Task-01 ~]# ping -i 3 www.baidu.com
```

# 使用 ping -s 测试对发送 ICMP 请求数据包的大小，数据包为 1024bytes

```
[root@Lab-16-Task-01 ~]# ping -s 1024 www.baidu.com
```

# 多参数测试

# 测试对 www.baidu.com 的连通性，接收 2 次，每次时间 3 秒

```
[root@Lab-16-Task-01 ~]# ping -c 2 -i 3 www.baidu.com
```

# 测试对 www.baidu.com 的连通性，极限测试模式，数据包为 1024bytes，测试 100 次

```
[root@Lab-16-Task-01 ~]# ping www.baidu.com -f -s 1024 -c 100 -q
```

---

### 2、使用 Traceroute 进行访问路由测试

traceroute 工具利用 ICMP 协议定位您的计算机和目标计算机之间的所有路由器。

---

```
# 使用 traceroute 进行访问路由路径
[root@Lab-16-Task-01 ~]# traceroute www.baidu.com

# 使用 traceroute -m 检测数据包的最大存活数值 TTL,数值 10
[root@Lab-16-Task-01 ~]# traceroute -m 10 www.baidu.com

# 使用 traceroute -n 用来只显示 IP 地址, 不显示主机名称
[root@Lab-16-Task-01 ~]# traceroute -n www.baidu.com

# 使用 traceroute -p 设置探测包使用的 UDP 传输协议的通信端口, 端口设置为 6888
[root@Lab-16-Task-01 ~]# traceroute -p 6888 www.baidu.com

# 使用 traceroute -r 用来绕过路由表直接发送到网络相连的主机
[root@Lab-16-Task-01 ~]# traceroute -r www.baidu.com

# 使用 traceroute -w 设置等待远程主机的响应时间, 设置时间为 3 秒
[root@Lab-16-Task-01 ~]# traceroute -w 3 www.baidu.com

# 使用 traceroute -q 设置探测包的次数, 设置为 4 次
[root@Lab-16-Task-01 ~]# traceroute -q 4 www.baudu.com
```

---

### 3、使用 mtr 进行网络稳定性测试

mtr 工具结合了 ping、traceroute、nslookup 的相关特性, 使管理员能够诊断和隔离网络错误, 并向上游提供商提供网络状态报告。

---

```
# 使用 mtr -r 命令以报告的模式显示测试结果
[root@Lab-16-Task-01 ~]# mtr -r www.baidu.com

# 使用 mtr -c 命令用来设置每秒发送数据包的数量, 设置为 30
[root@Lab-16-Task-01 ~]# mtr -c 30 www.baidu.com

# 使用 mtr -s 命令用来指定 ping 数据包的大小, 设置为 1024
[root@Lab-16-Task-01 ~]# mtr -s 1024 www.baidu.com

# 使用 mtr -n 命令用来设置输出结果, 只显示 IP 地址, 不显示域名
[root@Lab-16-Task-01 ~]# mtr -n www.baidu.com

# 使用 mtr -b 命令用来设置输出结果, 同时显示主机名与 IP 地址
[root@Lab-16-Task-01 ~]# mtr -b www.baidu.com

# 使用 mtr -i 命令用来设置 ICMP 返回之间的时间间隔, 设置为 2 秒
[root@Lab-16-Task-01 ~]# mtr -i 2 www.baidu.com

# 多参数测试
# 通过 udp 测试对 www.baidu.com 的连通性, 每秒发送 50 个数据包, 以报告模式显示
[root@Lab-16-Task-01 ~]# mtr --udp -rwc 50 www.baidu.com
```

```
# 通过 tcp 测试对 www.baidu.com 的连通性, 每秒发送 50 个数据包, 以报告模式显示  
[root@Lab-16-Task-01 ~]# mtr --tcp -rwc 50 www.baidu.com
```

---

#### 4、使用 tcpdump 进行通信报文分析

tcpdump 是一个运行在命令行下的抓包工具。它允许用户拦截和显示发送或收到过网络连接到的该计算机的 TCP/IP 和其他数据包。

---

```
# 使用 tcpdump 命令显示 tcp 包信息  
[root@Lab-16-Task-01 ~]# tcpdump  
  
# 使用 tcpdump -c 命令用来设置显示数据包的数量, 设置为  
[root@Lab-16-Task-01 ~]# tcpdump -c 20  
  
# 使用 tcpdump -w 命令用来将包指定到记录文件  
[root@Lab-16-Task-01 ~]# tcpdump -c 10 -w tcpdump_test.log  
  
# 使用 tcpdump -r 命令用来从指定的文件中读取包  
[root@Lab-16-Task-01 ~]# tcpdump -r tcpdump_test.log  
  
# 使用 tcpdump -i 命令用来设置指定监控的网卡  
[root@Lab-16-Task-01 ~]# tcpdump -i ens192  
  
# 使用 tcpdump -v 命令用来显示更加详细的抓包信息  
[root@Lab-16-Task-01 ~]# tcpdump -v  
  
# 使用 tcpdump -tttt 命令用来显示抓包结果中包含抓包日期  
[root@Lab-16-Task-01 ~]# tcpdump -t  
  
# 多参数测试  
# 监视所有送到主机 hostname 的数据包  
[root@Lab-16-Task-01 ~]# tcpdump -i ens192 dst host hostname  
  
# 获取主机 210.45.123.249 接收或者发出的 telnet 包, 23 为 telnet 的端口  
[root@Lab-16-Task-01 ~]# tcpdump tcp port 23 and host 210.45.123.249  
  
# 监视本机的 udp 123 端口, 123 为 ntp 的服务端口  
[root@Lab-16-Task-01 ~]# tcpdump udp port 123  
  
# 截获主机 172.16.1.29 和主机 210.45.123.249 或 210.45.123.248 的通信  
[root@Lab-16-Task-01 ~]# tcpdump -i enp0S3 -c 3 'host 172.16.1.29 and 172.20.1.  
79'
```

---

#### 5、使用 iperf3 进行网络性能测试

iPerf3 工具用于主动测试 IP 网络上最大可用带宽的工具。它支持时序、缓冲区、协议 (TCP, UDP, SCTP 与 IPv4 和 IPv6) 有关的各种参数。对于每次测试, 它都会详细的带宽报告, 延迟抖动和数据包丢失。

Server 参数:

- s, --server, 以 Server 模式运行;
- D, --daemon, 在后台以守护进程运行;
- I, --pidfile file, 指定 pid 文件;
- l, --one-off, 只接受 1 次来自 Client 端的测试, 然后退出。

Client 参数:

- c, --client, 以 Client 模式运行, 并指定 Server 端的地址;
- u, --udp, 以 UDP 协议进行测试;
- b, --bandwidth #[KMG][/#], 限制测试带宽。UDP 默认为 1Mbit/秒, TCP 默认无限制;
- t, --time #, 以时间为测试结束条件进行测试, 默认为 10 秒;
- n, --bytes #[KMG], 以数据传输大小为测试结束条件进行测试。

---

# 使用 iperf3 -s 命令服务器端运行, -p 可指定端口号

```
[root@Lab-16-Task-02 ~]# iperf3 -s
```

# 使用 iperf3 -c 命令在客户端, 开始进行带宽测试

```
[root@Lab-16-Task-01 ~]# iperf3 -c 172.20.1.82
```

# 使用 iperf3 -P 命令在客户端, 设置多个并行数据流

```
[root@Lab-16-Task-01 ~]# iperf3 -c 172.20.1.82 -P streams
```

# 使用 iperf3 -R 命令在客户端, 进行反向测试

```
[root@Lab-16-Task-01 ~]# iperf3 -c 172.20.1.82 -R
```

# 使用 iperf3 -t 命令在客户端, 设置测试时间为 20 秒

```
[root@Lab-16-Task-01 ~]# iperf3 -c 172.20.1.82 -t 20
```

# 使用 iperf3 -b 命令在客户端, 使用 UDP 协议, 并设置使用的测试带宽

```
[root@Lab-16-Task-01 ~]# iperf3 -c 172.20.1.82 -b 1000M -t 60 -d
```

---

**提醒:**

请根据开展本实验的本地主机所在网络实际情况, 选择合适的服务器地址进行测试。

本实验使用的 IP 地址是 172.20.1.82。

---