

nsd1907_net_day01

张志刚：第五阶段Python，周末RHCA

DOD：国防部

ARPA：高级研究项目署 -> ARPAnet

Modem：猫 调制解调器 56kbps => 7kB/s = 》 拨号上网

描述速率的时候，用的都是bit，电脑上描述文件大小都是字节byte

1byte = 8 bit

ISDN：超级一线通，综合业务数字网

网络分类

按地域范围分类

- 局域网：LAN
- 广域网：WAN
- 城域网：MAN

按技术划分

- 对等网：网络中主机具有相同角色
- C/S：客户机/服务器
- B/S：浏览器/服务器

按安全划分

- intranet：内部网络
- extranet：外部网络
- Internet：国际互联网

巨大中华：巨龙集团、大唐电讯、中兴、华为

中华酷联：中兴、华为、酷派、联想

网络拓扑结构

- 星型：重要、常用的结构
- 点到点：互联网上有应用，LAN几乎没有
- 总线型：几乎没有
- 网状：几乎没有

OSI参考模型

- OSI：开放系统互联 参考模型。ISO制定
 - 自下向上是第一至第七层
 - 物理层：一层
 - 数据链路层：MAC地址、交换机
 - 网络层：IP地址、路由器
 - 传输层：TCP、UDP协议、端口号
 - 会话层
 - 表示层
 - 应用层：七层

印表会传往树屋

TCP/IP参考模型

- 共五层（四层）：前四层与OSI参考模型一样，OSI高三层合并为一个应用层

点到点：IP到IP

端到端：应用到应用

数据通信过程

1. 应用层程序准备好数据
2. 表示层负责将数据转换格式
3. 会话层负责建立会话连接，不是所有应用都有此步骤
4. 传输层负责把数据切分成一个个的片段，称作数据段segment。每个数据段加上编号，称作序列号。还需要指定使用的是TCP或UDP协议，以及端口号。
5. 网络层负责给数据加上IP地址，打成数据包package。
6. 数据链路层负责给数据加上MAC地址，打成数据帧frame。
7. 物理层将数据转换成二制的0和1，称作比特（流），在物理介质上传输。
8. 发送方，数据自上向下总是增加头部内容，称作封装过程。
9. 接收方，数据自下向上将发送方加入的头部移除，称作解封装过程。

华为设备

- 接口
 - console：控制台接口，连接控制线缆到PC机。相当于是为设备提供键盘、鼠标、显示器。
 - FastEthernet：快速以太网接口，速率100Mbps
 - GigabitEthernet：吉比特接口，速度1000Mbps
- 交换机配置
 - 命令需要在不同的模式下执行，不同模式有不同的指令
 - 用户视图：只能执行一些基本的查询指令：<主机名>
 - 系统视图：可以执行管理命令：[主机名]
 - 接口视图：只对某一接口生效的指令在这里执行：[主机名-接口]
 - 协议视图：配置相关协议时使用的视图：[主机名-协议]

```
<Huawei>?      # 显示该模式下所有的命令
<Huawei>l?      # 显示所有的以l开头的命令
<Huawei>la<tab>  # 因为以la开头的命令只有一条，所以此时按tab可补全
<Huawei>language-mode Chinese  # 改变为中文提示
<Huawei>system-view  # 进入系统视图
[Huawei]sysname sw1  # 修改主机名为sw1
# 在华为设备上，任何与显示相关的命令，都以display开头
[sw1]display interface brief  # 显示所有端口的简要信息
[sw1]int g0/0/1  # 进入接口
[sw1-GigabitEthernet0/0/1]quit  # 返回一步
[sw1-GigabitEthernet0/0/1]return  # 直接返回到用户视图
[sw1-GigabitEthernet0/0/1]ctrl+z  # 按组合键也会返回到用户视图
```

翻译软件：搜狗搜译。

交换机常用命令

```
# 华为操作系统名为VRP，即多样化路由平台
<sw1>display version  # 查看设备的软件版本号和运行时间
<sw1>display current-configuration  # 查看设备当前的生效配置

# 配置密码
# AAA模式，认证模式
# Authentication：认证，你是谁
# Athorization：授权，你能干什么
# Audit：审计，你干了什么
<sw1>system-view
[sw1]aaa
[sw1-aaa]local-user 用户名 password cipher 密码  # 设置加密密码
[sw1-aaa]quit
[sw1]user-interface console 0  # 进入控制台接口模式
[sw1-ui-console0]authentication-mode aaa  # 配置使用AAA认证
# 退出到最外面，再次回车，进入用户视图就需要用户名和密码了

<sw1>save  # 系统提示将会把系统原有配置覆盖，是否要继续，选择y，系统再次提示文件名，直接回车，采用默认的名字即可
<sw>reboot  # 重启

<sw1>reset saved-configuration  # 擦除保存的配置，系统提示，是否真的擦除，回答y。注意，擦除的是已保存的配置，不是当前内存中生效的配置。重启后，才会恢复到最初状态。
<sw1>reboot  # 系统首先比较内存中配置与保存的配置是否一致，发现不一致，提示是否要把当前配置保存，此时回答：n。系统再次提示，是否要继续重启，回答y
```

其他命令行小技巧

```
# 配置设备没有控制台超时时间，相当于配置主机、手机锁屏前的等待时间
[sw1]user-interface console 0
[sw1-ui-console0]idle-timeout 分钟数
```

交换机工作原理

- MAC地址：MAC是介质访问控制。
 - MAC地址也叫物理地址，因为设备在生产的时候，MAC地址已经烧到印制电路版中了
 - 网卡有MAC地址，交换机、路由器每个端口也有MAC地址
 - MAC地址共48位（48个2进制数）、6字节。
 - 表示的时候，往往是把2进制转换成16进制进行表示，每两个16进制数之间用冒号或减号分隔。一共是5个冒号或分号隔开6段。
 - MAC地址前24位是厂商的组织唯一标识符，即OUI；后24位厂商自行决定

2进制与16进制的转换：4位2进制转换成一个16进制数

```
0000 -> 0
0001 -> 1
0010 -> 2
0011 -> 3
0100 -> 4
0101 -> 5
0110 -> 6
0111 -> 7
1000 -> 8
1001 -> 9
1010 -> A
1011 -> B
1100 -> C
1101 -> D
1110 -> E
1111 -> F
```

- 交换机的工作原理
 - 交换机内部有一张MAC地址表，记录了每个端口连接设备的MAC地址
 - MAC地址表在开机的时候是空的，它要动态学习每个端口所连设备的MAC地址
 - 当某一终端发送数据帧到交换机时，交换机根据数据帧的源MAC地址构建MAC地址表
 - 交换机根据数据帧的目标MAC地址发送数据到指定端口
 - 如果交换机在MAC地址表中找不到目标MAC地址连在自己的哪个端口上，将会把数据帧向除接收端口以外的所有端口发出