

网络编程

Java Platform Standard Edition

Java教学部

课程目标

CONTENTS

ITEMS **1** 什么是网络

ITEMS **2** 计算机网络

ITEMS **3** 网络模型

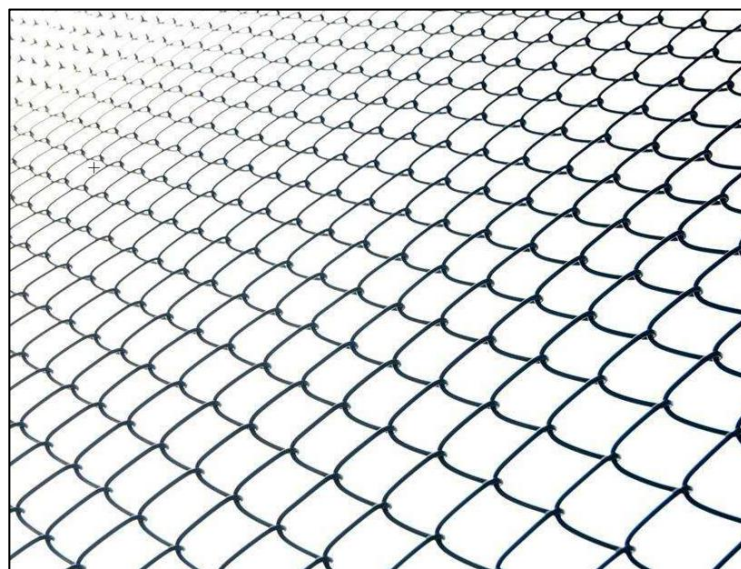
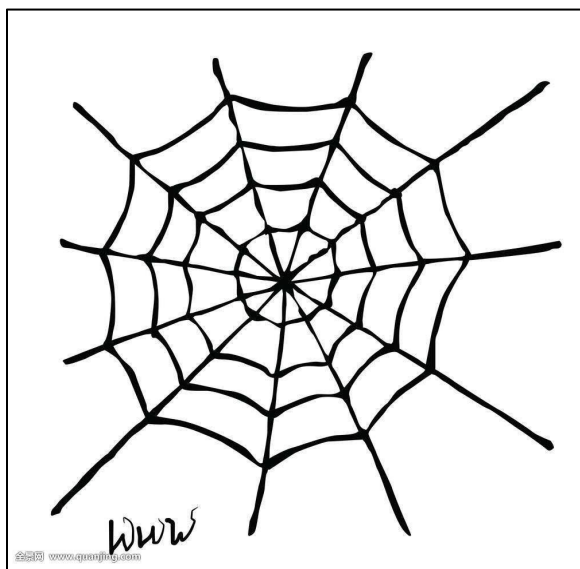
ITEMS **4** 通信协议

ITEMS **5** IP与端口

ITEMS **6** 网络编程

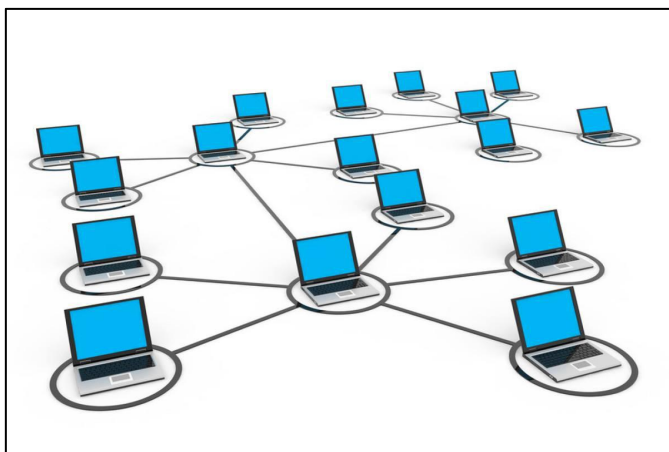
什么是网络

- 由点和线构成，表示诸多对象间的相互联系。



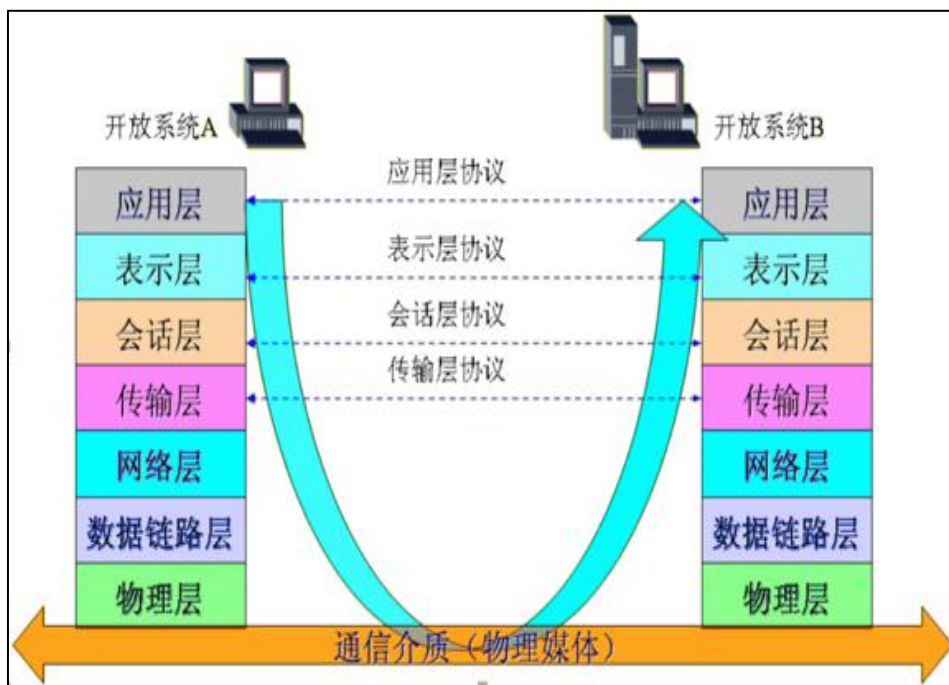
什么是计算机网络

- 为实现资源共享和信息传递，通过通信线路连接起来的若干主机（Host）。



- 互联网：（Internet）点与点相连
- 万维网：（WWW – World Wide Web）端与端相连
- 物联网：（IoT - Internet of things）物与物相连
- 网络编程：让计算机与计算机之间[建立连接](#)、[进行通信](#)。

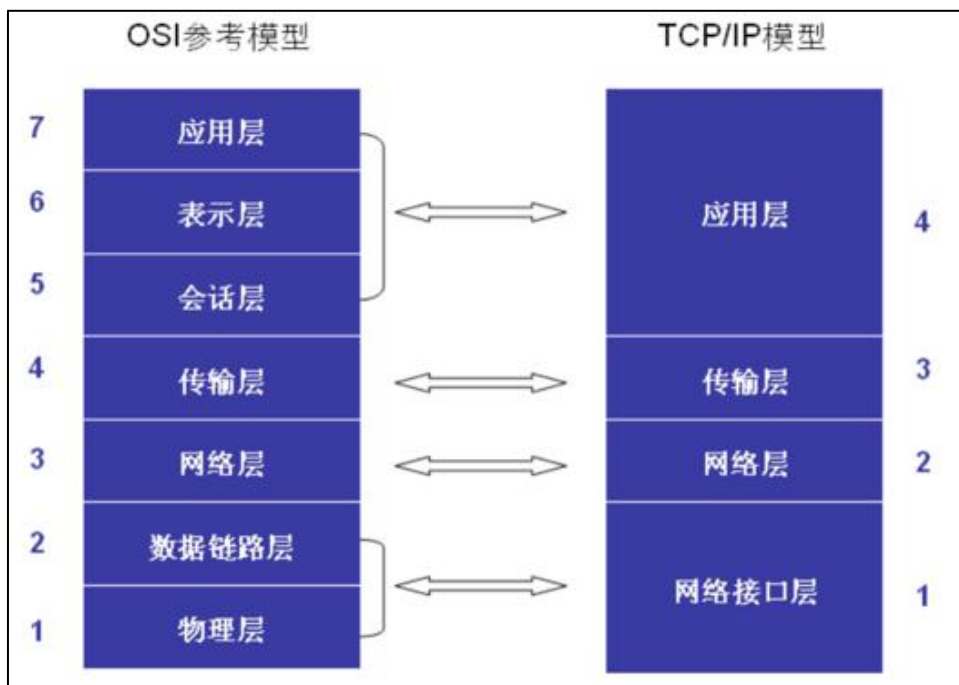
- OSI（Open System Interconnection）开放式系统互联。



- 第七层：应用层负责文件访问和管理、可靠运输服务、远程操作服务。（HTTP、FTP、SMTP）
- 第六层：表示层负责定义转换数据格式及加密，允许选择以二进制或ASCII格式传输。
- 第五层：会话层负责使应用建立和维持会话，使通信在失效时继续恢复通信。（断点续传）
- 第四层：传输层负责是否选择差错恢复协议、数据流重用、错误顺序重排。（TCP、UDP）
- 第三层：网络层负责定义了能够标识所有网络节点的逻辑地址。（IP地址）
- 第二层：链路层在物理层上，通过规程或协议（差错控制）来控制传输数据的正确性。（MAC）
- 第一层：物理层为设备之间的数据通信提供传输信号和物理介质。（双绞线、光导纤维）

TCP/IP模型

- 一组用于实现网络互连的通信协议，将协议分成四个层次。



- 第四层：应用层负责传送各种最终形态的数据，是直接为用户打交道的层，典型协议是HTTP、FTP等。
- 第三层：传输层负责传送文本数据，主要协议是TCP、UDP协议。
- 第二层：网络层负责分配地址和传送二进制数据，主要协议是IP协议。
- 第一层：接口层负责建立电路连接，是整个网络的物理基础，典型的协议包括以太网、ADSL等等。

- TCP协议：Transmission Control Protocol 传输控制协议
 - 是一种面向连接的、可靠的、基于字节流的传输层通信协议。数据大小无限制。建立连接的过程需要三次握手，断开连接的过程需要四次挥手。
- UDP协议：User Datagram Protocol 用户数据报协议
 - 是一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务，每个包的大小64KB。

- IP协议：Internet Protocol Address 互联网协议地址/网际协议地址
 - 分配给互联网设备的数字标签（唯一标识）。
- IP地址分为两种：
 - IPV4：4字节32位整数，并分成4段8位的二进制数，每8位之间用圆点隔开，每8位整数可以转换为一个0~255的十进制整数。
格式：D.D.D.D 例如：255.255.255.255
 - IPV6：16字节128位整数，并分成8段十六进制数，每16位之间用圆点隔开，每16位整数可以转换为一个0~65535的十进制数。
格式：X.X.X.X.X.X.X.X 例如：FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF

IPV4的应用分类

- A类：政府机构，1.0.0.1 ~ 126.255.255.254
- B类：中型企业，128.0.0.1 ~ 191.255.255.254
- C类：个人用户，192.0.0.1 ~ 223.255.255.254
- D类：用于组播，224.0.0.1 ~ 239.255.255.254
- E类：用于实验，240.0.0.1 ~ 255.255.255.254
- 回环地址：127.0.0.1，指本机，一般用于测试使用。
- 查看IP命令：ipconfig
- 测试IP命令：ping D.D.D.D

- 端口号：在通信实体上进行网络通讯的程序的唯一标识。
- 端口分类：
 - 公认端口：0~1023
 - 注册端口：1024~49151
 - 动态或私有端口：49152~65535
- 常用端口：
 - MySQL: 3306
 - Oracle: 1521
 - Tomcat: 8080
 - SMTP: 25
 - Web服务器: 80
 - FTP服务器: 21

- 概念：表示互联网协议（IP）地址对象，封装了与该IP地址相关的所有信息，并提供获取信息的常用方法。
- 方法：
 - `public static InetAddress getLocalHost()` 获得本地主机地址对象
 - `public static InetAddress getByName(String host)` 根据主机名称获得地址对象
 - `public static InetAddress[] getAllByName(String host)` 获得所有相关地址对象
 - `public String getHostAddress()` 获取IP地址字符串
 - `public String getHostName()` 获得IP地址主机名

基于TCP的网络编程

- Socket编程：
 - Socket（套接字）是网络中的一个通信节点。
 - 分为客户端Socket与服务器ServerSocket。
 - 通信要求：IP地址 + 端口号。



- 建立通信连接（会话）：
 - 创建ServerSocket，指定端口号
 - 调用accept等待客户端接入
- 客户端请求服务器：
 - 创建Socket，指定服务器IP + 端口号
 - 使用输出流，发送请求数据给服务器
 - 使用输入流，接收响应数据到客户端（等待）
- 服务器响应客户端：
 - 使用输入流，接收请求数据到服务器（等待）
 - 使用输出流，发送响应数据给客户端

- 使用Scoket编程实现服务器端注册：
 - 注册信息保存在properties文件中。
 - 封装格式：
 - `id = {id : "1001", name : "tom", pwd : "123", age : 20 , score : 98.5}`
 - 注册成功后返回字符串“注册成功”。
- 使用Scoket编程实现服务器端登录：
 - 获取properties文件中的用户信息，进行用户名与密码的校验。
 - 校验成功后返回字符串“登录成功”。

- 计算机网络：
 - 为实现资源共享和信息传递，通过通信线路连接起来的若干主机。
- TCP协议：
 - 是一种面向连接的、可靠的、基于字节流的传输层通信协议。数据大小无限制。
- IP：
 - 分配给互联网设备的数字标签（唯一标识）。
- Port：
 - 在通信实体上进行网络通讯的程序的唯一标识。
- Socket编程：
 - 建立连接、接收请求、发送响应。