* **计算机网络：利用通信手段，把地理上分散的、能够以相互共享资源（硬件、软件和数据等）的方式有机地连接起来的、而各自又具备独立功能的自主计算机系统的集合。**

**TCP/IP的四层功能：1）应用层：应用层协议提供远程访问和资源共享及各种应用服务。2）传输层：提供端到端的数据传送服务；为应用层隐藏底层网络的细节。 3）网络层：处理来自传输层的报文发送请求；处理入境数据报；处理ICMP报文。4）网络接口层：包括用于物理连接、传输的所有功能。**

**在网络体系结构中，网络层的主要功能？**

1. **为不同的主机提供通信服务；**
2. **选择合适的路由。**

**物理层的主要任务：**

**利用传输介质为通信的网络结点之间建立、管理和释放物理链接，实现比特流的透明传输，为数据链路层提供数据传输服务。**

**为何分层:目的是把各种特定的功能分离开来，使其实现对其他层次来说是可见的。分层结构使各个层次的设计和测试相对独立。各层分别实现不同的功能，下层为上层提供服务，各层不必理会其他的服务是如何实现的，因此，层1实现方式的改变将不会影响层。**

**TCP/IP结构对应OSI结构**

|  |  |
| --- | --- |
| **TCP/IP** | **OSI** |
| 应用层(FTP、HTTP) | 应用层  表示层  会话层 |
| 主机到主机层（TCP、UDP）（又称传输层） | 传输层 |
| 网络层（IP） | 网络层 |
| 网络接口层(又称链路层）(PPP、ATM) | 数据链路层 |
| 物理层 |

TCP协议的特点:

1.  TCP是面向连接的运输层协议；

2. 每一条TCP连接只能有两个端点，只能是点对点的；

3. TCP提供可靠交付的服务；

4. TCP提供全双工通信；

 5. 面向字节流。

UDP协议的特点？

1. UDP是无连接的；
2. UDP使用尽最大努力交付；
3. UDP是面向报文的；
4. UDP没有拥塞控制；
5. UDP支持一对一、一对多、多对一、多对多的交互通信‘
6. UDP的首部开销小。

协议与服务区别和关系:

协议是水平的，服务是垂直的

网络协议是计算机网络中进行数据交换而建立的规则、标准或约定的集合。二者的区别在于：首先协议的实现保证了能够向上一层提供服务。本层的服务用户只能看到服务而不能看到下面的协议，下面的协议对上面的服务用户时刻透明的。其次，协议是控制实体之间通信的规则，而服务是由下层向上层通过层之间接口提供的

IP数据包中的首部检验和并不检验数据包中的数据，这样做的最大好处是什么？坏处是什么？

IP数据报对传输的数据不做检验，这样做的最大好处是可以减少IP数据报的处理复杂度，提高数据报的处理速度。坏处是，这样做实际上把检验的任务交给了上层协议（如传输层）,增加了上层协议的复杂性。

* 数据链路层

两种信道：广播信道、点对点信道。

数据链路层的功能：实现相邻结点间链路的可靠传输

链路层提供的服务：成帧（即（二）组帧）、链路访问（即（五）介质访问控制）、差错检测和纠错（即（三）差错控制）

（二）组帧：基本的模式为：帧首部+数据部分（大小不能超多mtu）+帧尾部。

（三）差错控制：CRC校验

发送方用一个特殊的多项式表达式（循环码生成多项式），去除信息码多项式所得的余式，附加在信息码之后，构成传输码；接收方用获得的传输码组成的多项式除以发送方使用的同一个多项式表达式，即生成多项式，若无余式，则表明传输码无错，若有余式，则表明有错

**（四）流量控制与可靠传输机制**

**滑动窗口协议的基本原理：数据单元编号不能太大，应循环利用。对发送方发出去的未经确认的帧的数目加以限制，这个受限制的数目称为发送窗口大小。为减少开销，接收端可在收到若干个正确的数据帧后一次性发送一次确认帧,类似于发送窗口，同时也规定一个接收窗口，只有当接收的帧号落在接收窗口内时才允许将该帧收下，否则将其丢弃。接收方每正确接收一个数据帧，就向网络层上交一个帧，接收窗口后移一帧，增加一个准备接收的新的帧号，并向发送方发回一个确认帧，发方接收到确认帧后，移动发送窗口，发送新的数据帧。**

**1.流量控制、可靠传输与滑轮窗口机制**

**a)滑轮窗口机制：发送方连续发送窗口内的所有分组。接收方可以累积确认（只对按序到达的最后一个分组发送确认。）**

**2.单帧滑动窗口与停止-等待协议：不论是否已有该分组，收到就确认、超时重传、分组和ack有编号。**

**3.多帧滑动窗口与后退N帧协议（GBN也称滑动窗口协议）：丢弃失序分组（使接受缓存简单）：序号、检查、累积确认、超时/重传。**

**4.多帧滑动窗口与选择重传协议（SR）：窗口长度必须小于或等于序号空间大小的一半。**

（五）介质访问控制

1.信道划分介质访问控制（信道划分协议） 频分多路复用（FDM）、时分多路复用（TDM）、波分多路复用（WDM，光的频分复用）、码分多路复用（CDMA）

2.随机访问介质访问控制（随机访问协议）

ALOHA协议：时隙ALOHA的效率公式，最大效率：1/e=0.37、0.37的概率空闲、0.26的概率碰撞；纯ALOHA的效率是时隙的一半。

 CSMA协议：载波侦听多址访问协议：

CSMA/CD协议：带碰撞检测的载波侦听多址访问协议。（以太网）

 CSMA/CA协议：避免碰撞的载波侦听多址访问协议。（无线局域网）

3.轮询访问介质访问控制：（轮流协议）

令牌传递协议要点：有帧传送时持有令牌、无帧传送时转发令牌、发送帧数超过最大帧数也要转发令牌；

令牌传递协议的注意点：一个节点崩溃可能导致整个信道崩溃，一个节点释放令牌出现故障时的恢复。

（六）局域网

1.局域网的基本概念与体系结构：（重点是以太网和无线局域网）

2.以太网与IEEE802.3：（mtu=1500字节）、提供不可靠的无连接服务、使用CSMA/CD。

a)以太网综述：使用802.3标准；使用CSMA/CD协议；物理上是星型拓扑结构，逻辑上是总线拓扑结构。

b)CSMA/CD：机制：

i.没有使用时隙

ii.载波侦听

iii.碰撞检测，有碰撞就停止发送。检测到冲突后继续等待的时间采用二进制指数退避算法决定

iv.等待重传 注：大致的工作流程也应该熟悉。

CSMA/CD协议的工作原理:  首先，每个站点发送数据之前必须侦听信道的忙、闲状态。如果信道空闲，立即发送数据，同时进行冲突检测；如果信道忙，站点继续侦听总线，直到信道变成空闲。如果在数据发送过程中检测到冲突，将立即停止发送数据并等待一段随机长的时间，然后重复上述过程。即：先听后发，边听边发；冲突检测，延时重发

CSMA/CD协议的要点为多点接入、载波监听、 碰撞检测

c)以太网帧结构：

i.数据字段46（任何小于46+6+6+2+4=64的以太网帧都是无效帧，会被丢弃，所以如果ip数据报的长度不足46字节，那么需要填充到46字节）到1500字节

ii.目的地址：6字节（mac地址）

iii.源地址：6字节（mac地址）

iv.类型字段：标志上层使用什么协议。2字节

v.循环冗余检测：4字节 注：前同步码并不是以太网帧的组成部分。

d)以太网的扩展：802.3规定任意两个站之间最多可经过三个电缆网段 。10BASE-T以太网中主机与集线器（或交换机等连接设备）的最大距离为100m，即同一网段的两主机的最大距离为200m。 注：此处倒是有可能出题。

3.IEEE 802.11：（Wi-Fi）

a)802.11b、a、g的链路层帧结构都相同，物理层有大区别。个人觉得注意相同点就好了。

b)802.11综述：使用CSMA/CA协议。基本构件为基本服务集BBS，中心称为接入点AP。

c)信道与关联： d)CSMA/CA协议：

i.工作原理，或者说流程。

ii.预约机制：RTS和CTS e)帧结构：

i.有效载荷与CRC字段。

ii.地址字段

iii.序号、持续期和帧控制字段

4.令牌环网的基本原理：lan的N个节点通过直接链路连接成一个环。令牌环的拓扑定义了令牌的传递次序。一个节点获得令牌另发送一个帧之后，帧沿整个环广播。目的节点接受帧，发送节点则删除帧。

（七）广域网

1.广域网的基本概念

2.PPP协议：点对点协议。面向字符的协议

a)特点：简单，只检错，不纠错，具有透明性

b)PPP帧格式：

i.标志字段：用01111110 的一字节字段来作开始和结束。

 ii.地址字段：唯一可能的值是：11111111

iii.控制字段：唯一可能的值：00000011（这两个字段以后可能有其他值，现在只有固定值，所以可以不发送）

iv.协议：上层网络层使用的协议。

 v.信息：最大1500字节。

 vi.检查和

c)PPP的字节填充问题：为了保持ppp的透明性，采用字节填充技术解决，其他字段出现标志字段的值。当其他字段出现标志字段的值时，发送方在前面填充控制转义字节01111101，接收方自行恢复。（一个转义字符，表示出现01111110，连续两个，表示出现01111101）

 d)PPP主要部分：成帧、链路控制协议、网络控制协议

 3.HDLC协议：高级数据链路控制

* 网络层

1. 网络层提供的两种服务

虚电路(VC):面向链接的，由网络确保提供可靠的服务。借鉴与电信网络。两个计算机通信前先建立链接。

数据报服务：网络层向上只提供简单灵活的，无连接的，尽最大努力交付数据报服务。

网络层不提供服务质量承诺。

 依据：计算机比电话机智能，有很强的差错处理能力。由于传输网络不提

供端到端的可靠服务，因此路由器可以设计的简单，价格低廉。