1. Give four key points that summarize the advantages of using a layered protocol architecture in your design.  *[4 marks]*
2. *复杂问题分为很多块*
3. *问题可以在层内被定位*
4. *模块化问题，厂商可以各自发力*
5. *有更大的市场*
6. List four key points of the process of how layered protocols architectures operate.分层模型如何运行
   1. 1. 逐层数据处理：每一层在向下传递的数据中添加其特定信息，如页眉或页脚。 在接收过程中，每一层都会处理并删除其各自的信息。
   2. 2. 封装和解封装：数据在各层中下降时，被封装有特定于层的控制信息。 随着数据的上升，这种封装会被反转，其中每一层都会剥离其各自的控制信息。
   3. 3. 独立层功能：架构中的每一层都有特定的职责，例如错误检查、加密或路由。 每层独立运行，确保处理的模块化。
   4. 4. 层与层之间的标准化接口：层之间通过标准化接口进行交互，使层之间能够有效地通信和传输数据，而无需了解彼此的内部工作原理。
7. OSI模型分层架构介绍
   1. 1. 物理层：
      1. • 通过物理介质传输原始比特。
      2. • 处理设备之间的物理连接以及信号的传输和接收。
   2. 2.数据链路层：
      1. • 管理物理传输的物理寻址和帧。
      2. • 处理物理层的错误检测和纠正。
   3. 3.网络层：
      1. • 管理数据包的寻址和路由。
      2. • 确保数据包发送到正确的目的地。
   4. 4.传输层：
      1. • 确保端到端通信和错误恢复。
      2. • 管理流量控制，确保数据以适合发送方和接收方的速率发送。
   5. 5.会话层：
      1. • 管理应用程序之间的会话。
      2. • 控制计算机之间的对话（连接）。
   6. 6.表示层：
      1. • 在应用层和网络之间转换数据。
      2. • 管理数据加密和解密。
   7. 7、应用层：
      1. • 最接近最终用户。
      2. • 管理基于网络的服务和应用程序
8. NAT协议的目的及原理
   1. 1、NAT的目的：
      1. 节省 IP 地址：NAT 允许专用网络上的多个设备共享单个公共 IP 地址。 鉴于 IPv4 地址的可用性有限，这一点至关重要。
      2. 增强安全性：通过对公共互联网隐藏私有 IP 地址，NAT 增加了一层安全性，使外部主机更难以直接访问内部设备。
   2. 2. NAT 的工作原理：
      1. 转换过程：当内部设备（例如家庭网络中的计算机）从互联网请求数据时，NAT 会将私有 IP 地址转换为分配给路由器或 NAT 设备的公共 IP 地址。 这种转换通常在将专用网络连接到互联网的路由器上完成。
      2. 端口映射：NAT 为每个传出请求分配唯一的端口号。 这样，当响应返回到路由器时，NAT 使用端口号来确定响应是针对哪个内部设备的，然后将响应转发到适当的设备。
   3. NAT 类型：
      1. 静态 NAT：将特定的公共 IP 地址分配给私有 IP 地址。
      2. 动态 NAT：从池中分配任何可用的公共 IP 地址。
      3. 端口地址转换 (PAT)：允许多个设备通过使用唯一的端口号进行区分来共享单个公共 IP 地址。
9. NAT是否违反OSI
   1. 1. 跨层：ISO/OSI 模型清楚地描述了从物理层到应用层的不同层的功能。 然而，NAT 在网络层（第 3 层 - IP）运行，但也利用传输层（第 4 层 - TCP/UDP）信息，特别是端口号。
   2. 2. 协议信息的使用： NAT 涉及修改数据包标头中的 IP 地址（网络层功能），并且还经常更改 TCP/UDP 端口号（传输层功能）。 这种不同层功能的混合违背了 OSI 模型保持各层独立且不同的原则。
   3. 3.对端到端连接的影响：OSI模型主张维护端到端原则，其中通信应由终端设备直接管理。 NAT 通过充当中介、修改 IP 地址并可能影响连接的直接性质来改变这一点。
   4. 从本质上讲，虽然 NAT 为 IP 地址稀缺和网络安全提供了实用的解决方案，但它的操作方式并不完全符合 OSI 模型中概述的严格关注点分离。
10. 不可靠服务提供了什么
    1. 1. 定义和产品：不可靠的无连接服务提供基本级别的数据传输，但不保证交付、订单或纠错。 它的典型特点是简单和快速。
    2. 2. 提供方式：
       1. • UDP（用户数据报协议）：这是不可靠的无连接协议的常见示例。 UDP 在不建立连接的情况下发送数据包（“数据报”），使其适合速度比可靠性更重要的应用程序。
       2. • 无确认：发送或接收的数据包没有确认机制，这意味着发送方不知道数据是否到达目的地。
       3. • 最小开销：由于缺少连接设置、维护和拆卸过程，该协议引入的开销最小，有助于加快数据传输速度。
       4. • 尽力交付：该服务提供“尽力而为”交付，表明它尝试交付数据，但不为丢失或损坏的数据包提供任何保证或恢复方法。
11. 可靠连接是什么服务
    1. “可靠的面向连接的服务”是一种网络通信类型，其中在传输任何数据之前在发送者和接收者之间建立专用连接。 该服务确保数据包的有序、无差错传送。
    2. [无错误 [1]，所有数据 [1] 已按顺序交付 [1]。 ]
    3. 可靠的面向连接的服务可保证所有数据均按发送顺序正确传送。
12. 可靠连接能提供什么服务
    1. 当使用不可靠的无连接网络时，可以提供可靠的面向连接的服务的机制包括错误检查、确认、序列号、定时器和流量控制。
    2. 1. 错误检查：确保传输的数据没有错误。
    3. 2. 确认：接收方确认收到数据，确保发送方知道传输成功。
    4. 3、序列号：用于对数据包进行排序，保证数据的顺序传输。
    5. 4. 定时器：监控数据传输和响应的时间，帮助检测丢失的数据包。
    6. 5. 流量控制：管理数据传输速率，防止网络拥塞。