交换机和路由器和集线器的相同、不同之处

设备	工作层次	主要功能	数据转发方式	网络效率	使用场景
交换机	数据链路层（Layer 2）	根据MAC地址转发数据，减少碰撞	智能转发，点对点通信	高效	局域网（LAN）
路由器	网络层（Layer 3）	根据IP地址转发数据，管理不同网络间的通信	路由计算，跨网段转发	较高	跨网段、互联网连接
集线器	物理层（Layer 1）	广播数据到所有端口，无数据处理	数据广播，半双工	低效	小型网络，已逐渐淘汰

相同之处：

- 都是网络设备，用于不同设备间的通信。
- 都会转发数据，但其转发方式和效率不同。

不同之处：

- 工作层次：集线器在物理层，交换机在数据链路层，路由器在网络层。
- 数据转发方式：集线器广播，交换机智能转发，路由器跨网段转发。
- 功能：交换机提高局域网效率，路由器连接不同网络，集线器简单转发数据。

请描述应用开发者在选择UDP协议或者TCP协议时的基本选择标准

主要根据应用的可靠性需求、延迟要求、流量控制需求、资源消耗和应用场景来做决定。TCP适合需要高可靠性、顺序保证的应用，UDP适合对实时性和速度要求较高、能容忍丢包的应用。

请列出网络拥塞的代价

延迟增加、数据丢失、吞吐量下降、重传增加、资源浪费、用户体验恶化等

CSMA/CD协议与传统的时分复用TDM相比优缺点如何?

- CSMA/CD适用于小规模、低流量的网络，尤其是在局域网（如传统以太网）中，它的简单性和低成本是优势。然而，随着网络流量增大，碰撞和重传会导致性能下降，不适合高带宽需求的场景。

- TDM适用于专用链路和需要高可靠性、高带宽保障的环境，尤其适用于电信、语音和实时应用。TDM的主要缺点在于其复杂性和带宽利用率在低负载时的浪费

CSMA/CD和CSMA/CA

5. 总结对比表

特性	CSMA/CD	CSMA/CA
碰撞处理	碰撞检测，发生碰撞时立即停止并重发	碰撞避免，使用 RTS/CTS 控制帧避免碰撞
适用场景	有线网络（如以太网）	无线网络（如 Wi-Fi）
碰撞检测	直接检测碰撞	无法检测碰撞，依靠避免
机制复杂度	相对简单，主要依靠碰撞检测	较复杂，需要 RTS/CTS 控制帧
延迟	较低，但碰撞发生时会增加延迟	较高，尤其在使用 RTS/CTS 时
退避机制	指数退避	随机退避，依赖信道空闲

总结：

- **CSMA/CD** 适用于有线网络，通过碰撞检测及时停止冲突并重发。简单高效，但在高负载下性能下降。
- **CSMA/CA** 适用于无线网络，避免碰撞发生，通过 RTS/CTS 等机制减少碰撞概率。机制较复杂，但可以更有效地避免无线环境中的冲突。

减小链路开销与无穷计数问题

当链路开销 **减小** 时，通常不会导致无穷计数问题。事实上，减小链路开销会导致网络中到达某个目标的路径变得更短，从而使相关路由器更新它们的路由表，使用新的更短的路径。这个过程是正常的，路由表更新会传播并逐渐稳定。

实现可靠数据传输的常用机制和技术

1. **确认与重传**：确保丢失的数据包能够重新传输。
2. **序列号**：确保数据包按正确顺序到达，并避免重复。
3. **流量控制**：调节发送和接收速度，避免接收方过载。
4. **差错检测与纠正**：确保传输过程中没有错误的数据包。
5. **拥塞控制**：防止网络拥堵，避免丢包。

五个层功能

1. 应用层：支持各类网络应用程序；
2. 传输层：提供端到端（进程到进程）的可靠数据传输；
3. 网络层：数据报路由和转发；
4. 数据链路层：保证数据帧在相邻节点的可靠传输；
5. 物理层：完成相邻节点链路上比特流的透明传输。

三种MAC协议

总结：

MAC协议	优点	缺点	适用场景
多路复用	高效利用信道、避免冲突、确定性强	同步复杂、带宽分配固定、适应性差	光纤通信、电话网络、卫星通信等
轮流接入	避免冲突、公平性好、确定性高	效率低、延迟大、控制开销大	集中式控制网络、Token Ring 网络等
随机接入	简单灵活、低延迟、高效（在低负载情况下）	冲突风险、效率降低、难以保证公平性	无线局域网（Wi-Fi）、以太网（CSMA/CD）等

Hub和router

总结对比表

特性	集线器（Hub）	交换机（Switch）
工作层次	物理层（Layer 1）	数据链路层（Layer 2）
传输方式	广播方式，数据包发送到所有设备	单播方式，数据包仅发送给目标设备
冲突域	所有设备共享同一个冲突域	每个端口形成独立的冲突域
带宽利用	所有设备共享带宽，随着设备增加带宽下降	每个端口独立带宽，不同端口之间互不干扰
网络性能	性能较差，容易发生冲突和网络拥塞	性能优越，减少冲突，提高网络效率
价格	较便宜	较贵
配置复杂度	简单，几乎不需要配置	可能需要配置，尤其是高级交换机
安全性	安全性差，所有设备都能接收到数据	安全性较高，数据只发给目标设备
应用场景	小型网络或低流量环境	中大型网络、需要高性能和高安全性的环境

结论：

- **集线器**适用于简单的、低负载的小型网络，但随着网络规模增大，性能和安全性问题会显现。
- **交换机**是现代网络中更常用的设备，能够提供更高的网络性能、减少冲突、增强安全性，适用于大多数中型和大型网络环境。

广播路由控制泛洪的三种方法

总结：三种控制泛洪的方法

方法	原理	优点	缺点	适用场景
限制广播次数（Hop Limit）	设置最大跳数限制，超过限制的广播包不再转发	防止广播包无限传播，减少网络负荷	可能导致信息无法传播到目标，特别是在复杂拓扑下	适用于希望控制广播范围和限制泛洪的简单网络或协议（如 RIP）
逆向路径转发（RPF）	检查广播包是否是从源节点的最佳路径转发来的，避免错误路径的转发	通过过滤错误路径广播包，减少冗余数据包，避免循环广播	需要维护路由信息，增加存储和计算开销	适用于多播路由协议（如 DVMRP、PIM-DM）
分层泛洪（Hierarchical Flooding）	网络划分为多个区域或组，广播只在局部区域内传播，跨区域通过边界节点转发	控制和减少广播的范围，适用于大规模网络	网络划分和层次结构复杂，增加边界节点的负担	适用于大规模网络和支持区域化的路由协议（如 OSPF、IS-IS）

这些方法都是为了减少网络中不必要的广播流量，提高网络性能和稳定性。不同的网络环境和需求决定了使用哪种方法最为合适。

区域间自治和区域内自治

- **区域内自治**是指单一自治系统内部的路由管理，强调自治系统内部的独立性和灵活性，通常使用 IGP（如 OSPF）来管理内部路由。它适用于小型网络、企业局域网、数据中心等场景。
- **区域间自治**则是多个自治系统之间的路由管理，通过 EGP（如 BGP）进行跨自治系统的路由交换。它适用于广域网（如互联网）以及不同组织或服务提供商之间的网络互联。

两者的关键区别在于自治系统的范围：区域内自治集中在单个自治系统内，而区域间自治处理的是跨多个自治系统的通信和路由管理。

三种switch fabric

bus、star、cross

IPv4向IPv6过渡的两种策略

From IPv4 To IPv6 提出的两种方法

- 双堆栈
    - 一些具有双栈（V6、V4）的路由器可以在格式之间“转换”

- 隧道
    - IPv6 在 IPv4 路由器之间作为 IPv4 数据报的负载传输

TCP（传输控制协议）是一个面向连接的、可靠的传输协议，它提供了流控制、拥塞控制、数据顺序保证等多种机制，确保数据在网络中的可靠传输。在这些机制中，**流控制**是TCP的一个关键特性，旨在防止发送方过快地发送数据，超出了接收方的处理能力。流控制的主要目标是避免接收方的缓冲区溢出。

TCP流控制流程概述

TCP的流控制主要通过**滑动窗口**（Sliding Window）机制来实现，它利用窗口大小来限制发送方在没有收到接收方确认的情况下能够发送的最大数据量。接收方通过告诉发送方它的缓冲区有多少可用空间，来动态调整窗口大小，从而实现流控制。

TCP流控制的关键元素

1. **接收窗口大小（rwnd）：**
    - **接收窗口（rwnd）**是接收方告诉发送方它可以接收的剩余数据量。这个值通过**TCP头部的窗口字段**（Window Size）进行传递，单位是字节。
    - 发送方必须确保它发送的数据不会超过接收方的缓冲区能力。
2. **滑动窗口（Sliding Window）：**
    - 滑动窗口是一个动态调整的概念，用于管理数据流的传输。它的大小由接收方控制并通过窗口大小（rwnd）传递给发送方。发送方必须保证自己发送的数据量不超过接收方的窗口大小。
    - 窗口大小会随着数据的接收和确认动态变化，窗口会滑动，允许发送方继续发送数据。
3. **ACK确认：**
    - 在TCP中，数据包的传输是通过确认（ACK）机制实现的。每当接收方接收到数据时，它会向发送方发送一个确认消息（ACK），确认已经收到的数据。
    - 在接收到ACK时，发送方知道哪些数据已经被成功接收，且可以移动窗口，从而发送更多的数据。

TCP流控制示意流程

1. **发送方发送数据**
    - 发送方根据接收方告知的接收窗口大小（rwnd），发送一定数量的数据。

- 每个数据包都会包含一个窗口字段，告诉接收方剩余的接收空间。

2. **接收方收到数据**
   - 接收方接收数据并通过ACK消息返回给发送方，告知接收到的数据序列号，并通告当前接收窗口大小。
3. **发送方调整发送窗口**
   - 发送方根据接收到的ACK和窗口大小调整其发送窗口，继续发送数据，直到接收窗口为0或所有数据传输完毕。
4. **接收方的缓冲区处理**
   - 如果接收方的缓冲区满了，它会在ACK中告诉发送方当前窗口大小为0，此时发送方会暂停发送数据，直到接收方缓冲区有足够空间。
5. **发送方的暂停与恢复**
   - 发送方暂停数据发送，直到接收方通过更新窗口大小（rwnd）允许继续发送更多数据。

流控制和拥塞控制

3. 流控制 vs 拥塞控制：

特性	流控制	拥塞控制
目标	确保接收方不被发送方过快的数据压垮	避免网络发生拥塞，确保网络能够正常工作
控制的对象	接收方的缓冲区大小（避免接收方缓冲区溢出）	网络的拥塞状况（避免路由器、交换机等网络设备过载）
窗口大小的来源	接收方通过ACK中的窗口大小告诉发送方	发送方根据网络拥塞情况调整拥塞窗口
窗口调整	通过接收方告知发送方接收窗口的大小	通过慢启动、拥塞避免、快速重传等机制动态调整发送速率
发送方行为	发送方根据接收方反馈的窗口大小控制发送数据量	发送方根据网络状况调整发送速率，避免过多数据占用网络带宽
影响因素	接收方的缓冲区能力	网络中的拥塞程度（如路由器的缓冲区、带宽等）
发生机制	由接收方控制，通过窗口大小反馈给发送方	由网络状况（如丢包、延迟等）控制，影响整个网络的传输效率
控制方式	滑动窗口机制	慢启动、拥塞避免、快速重传、快速恢复等机制