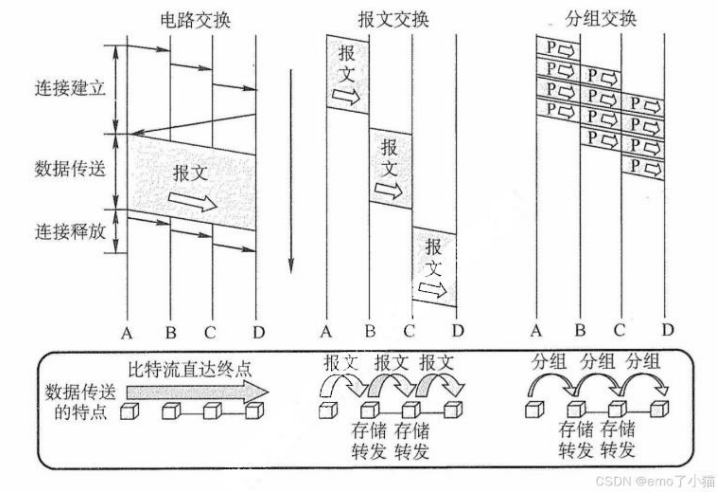
《高级计算机网络》期末复习

第 1 章 计算机网络概述

1、 简述电路交换、报文交换和分组交换的原理及优缺点。



(1) 电路交换（Circuit Switching）

原理：在数据传输前，首先在通信双方之间建立一条专用的物理连接（电路）。这条路径上的资源（如带宽）在连接期间被独占，直到通信结束，连接被释放。

优点：数据传输稳定，时延小，适用于需要持续、大量数据传输的场景。

缺点：建立连接时间长，线路利用率低，不适合突发性数据传输。

(2) 报文交换（Message Switching）

原理：数据以报文的形式被整个发送，每个报文独立寻址并经过网络中的节点存储转发至目的地。每个节点接收到报文后，先存储报文，然后根据目的地址选择下一个节点进行转发。

优点：不需要预先建立连接，灵活适应不同速率的链路，网络资源利用率高。

缺点：每个报文都需要经历存储转发过程，导致延迟较大，且对网络中的缓冲存储要求较高。

(3) 分组交换（Packet Switching）

原理：将数据分割成较小的单位，称为分组或包，每个分组包含目标地址信息。每个分组独立寻径并通过网络中的路由器进行转发，到达目的地后重新组装成原始数据。

优点：高效利用网络资源，减少了传播时延，支持动态路由选择，适应性强，非常适合互联网数据传输。

缺点：由于分组可能通过不同的路径到达，可能会出现乱序或丢失，需要额外的机制来保证数据的完整性和顺序。

2、 简述什么是发送时延、传播时延、处理时延、排队时延。

(1) 发送时延：发送数据时，数据块从结点进入到传输介质所需要的时间。

$$\text{发送时延} = \frac{\text{数据块长度}(b)}{\text{发送速率}\ (b/s)}$$

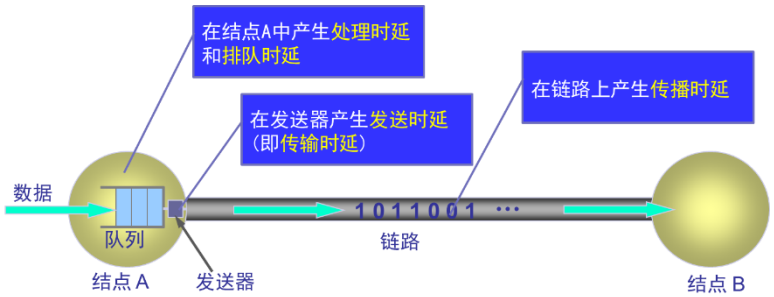
(2) 传播时延：电磁波在信道中需要传播一定的距离而花费的时间

$$\text{传播时延} = \frac{\text{信道长度}(m)}{\text{信号在信道上的传播速率}(m/s)}$$

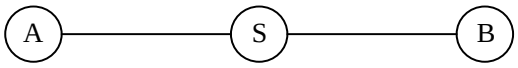
(3) 处理时延：主机或路由器在收到分组时进行一些必要的处理所花费的时间。

(4) 排队时延：结点缓存队列中分组排队所经历的时延。

(5) 总时延 = 发送时延 + 传播时延 + 处理时延 + 排队时延



3、 如下图所示，主机 A 和 B 都通过 10Mb/s 链路连接到交换机 S。



在每条链路上的传播延迟都是$20\mu s$。S 是一个存储转发设备，在它接收完一个分组$35\mu s$后开始转发收到的分组。试计算把 10000bit 从 A 发送到 B 所需要的总时间。

(1) 作为单个分组。

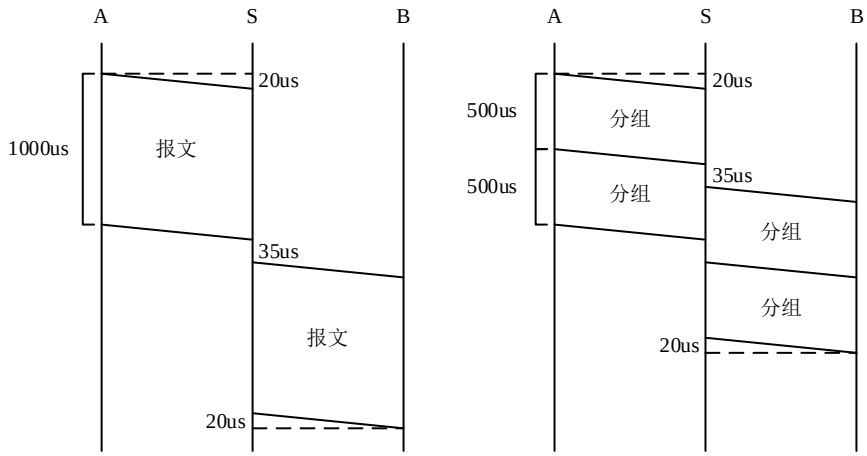
发送时延：$\frac{10000bit}{10Mb/s} = 1000\mu s$

总时间：$20 + 1000 + 35 + 1000 + 20 = 2075\mu s$

(2) 作为两个 5000bit 的分组一个紧接着另一个发送。

发送时延：$\frac{5000bit}{10Mb/s} = 500\mu s$

总时间：$20 + 500 + 35 + 500 \times 2 + 20 = 1575\mu s$



4、 假定有一个通信协议，每个分组都引入 100 字节的开销用于头和成帧。现在使用这个协议发送 1M 字节的数据，然而在传送的过程中有一个字节被破坏了，因而包含该字节的那个分组被丢弃。试对于 1000、5000、20000 和 40000 字节的分组数据大小分别计算“开销+丢失”字节的总数目？分组数据大小的最佳值是多少（提示：最佳值不是 1000、5000、20000 和 40000 字节中的任一个）？

分组数据大小	总分组数	开销+丢失
1000B	$\lceil 10^6 \div (1000 - 100) \rceil = 1112$	$1112 \times 100 + 1000 = 112200B$

5000B	$\lceil 10^6 \div (5000-100) \rceil = 205$	$205 \times 100 + 5000 = 25500$B
20000B	$\lceil 10^6 \div (20000-100) \rceil = 51$	$51 \times 100 + 20000 = 25100$B
40000B	$\lceil 10^6 \div (40000-100) \rceil = 26$	$26 \times 100 + 40000 = 42600$B

设分组数据大小为x，则“开销+丢失”字节的总数目为：$\lceil 10^6 \div (x-100) \rceil \times 100 + x$

对上式求导，得：$\lceil -10^8 \div (x-100)^2 \rceil + 1$

令上式为 0，解得$x = 10100$B，此为分组数据大小的最佳值。

5、 主机甲通过 1 个路由器（存储转发方式）与主机乙互联，两段链路的数据传输速率均为 10Mbps，主机甲分别采用报文交换和分组大小为 10Kb 的分组交换向主机乙发送 1 个大小为 8Mb（1M=10^6）的报文。若忽略链路传播延迟、分组头开销和分组拆装时间，请计算两种交换方式完成该报文传输所需的总时间。

(1) 采用报文交换时：

第 1 段链路的传输所需的时间为主机甲的发送时延 = 8Mb / 10Mbps = 0.8s = 800ms

第 2 段链路的传输所需的时间为路由器的发送时延 = 8Mb / 10Mbps = 0.8s = 800ms

采用报文交换时传输所需的总时间 = 800ms + 800ms = 1600ms

(2) 采用分组交换时：

传输所需的总时间 = 主机甲的发送时延 + 最后一个分组的路由器发送时延

（因为主机甲发送时延结束后最后一个分组从主机甲发出；最后这个分组还要经过路由器，直到最后一个分组到达主机乙才算传输结束，所以计算时除了全部数据在主机甲的发送时延，还要加上最后一个分组在路由器的发送时延。）

主机甲的发送时延 = 8Mb / 10Mbps = 0.8s = 800ms

最后一个分组的路由器发送时延 = 10Kb / 10Mbps = 0.001s = 1ms

采用分组交换时传输所需的总时间 = 800ms + 1ms = 801ms

6、 描述 OSI 参考模型和 TCP/IP 体系结构。

<table>
<tr><th>OSI 七层参考模型</th></tr>
<tr><td>应用层</td></tr>
<tr><td>表示层</td></tr>
<tr><td>会话层</td></tr>
<tr><td>传输层</td></tr>
<tr><td>网络层</td></tr>
<tr><td>数据链路层</td></tr>
<tr><td>物理层</td></tr>
</table>

<table>
<tr><th>TCP/IP 四层体系结构</th></tr>
<tr><td>应用层</td></tr>
<tr><td>传输层</td></tr>
<tr><td>网络层</td></tr>
<tr><td>网络接口层</td></tr>
</table>

<table>
<tr><th>TCP/IP 五层体系结构</th></tr>
<tr><td>应用层</td></tr>
<tr><td>传输层</td></tr>
<tr><td>网络层</td></tr>
<tr><td>数据链路层</td></tr>
<tr><td>物理层</td></tr>
</table>

7、 简述 OSI 参考模型各层主要功能及代表性协议。

(1) 物理层（Physical Layer）

功能：负责在物理媒介上传输原始的比特流。

代表协议：Ethernet（以太网）、WLAN（无线局域网）、HDLC（高级数据链路控制）。

(2) 数据链路层（Data Link Layer）

功能：负责在相邻的网络节点之间传输帧，进行帧同步、差错控制和流量控制。

代表协议：Ethernet（以太网）、PPP（点对点协议）、HDLC（高级数据链路控制）、LAPB（链路访问协议 B）。

(3) 网络层（Network Layer）

功能：负责在源节点和目的节点之间选择路由，确保数据包从源到目的地的传输。

代表协议：IP（网际互连协议）、ICMP（网际控制报文协议）、OSPF（开放最短路由优

先协议）、RIP（路由信息协议）、BGP（边界网关协议）。

(4) 传输层（Transport Layer）

功能：负责提供端到端的数据传输服务，确保数据的完整性和可靠性。

代表协议：TCP（传输控制协议）、UDP（用户数据报协议）、SCTP（流控制传输协议）。

(5) 会话层（Session Layer）

功能：负责建立、管理和终止应用程序之间的会话。

代表协议：RPC（远程过程调用）、SQL（结构化查询语言）、NFS（网络文件系统）。

(6) 表示层（Presentation Layer）

功能：负责数据的表示、安全和压缩，确保一个系统的应用层所发送的信息可以被另一个系统的应用层读取。

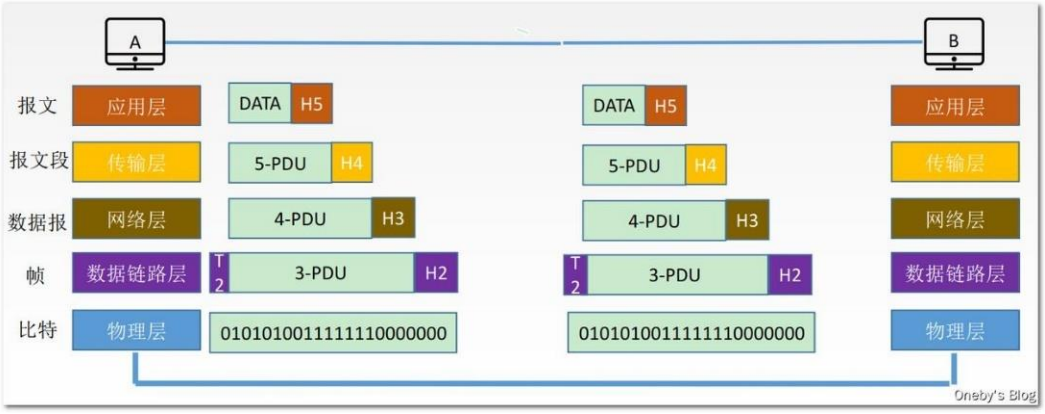
代表协议：SSL/TLS（安全套接层/传输层安全协议）、ASCII（美国信息交换标准代码）、JPEG（联合图像专家组）、MPEG（动态图像专家组）、ASN.1（抽象语法标记 1）。

(7) 应用层（Application Layer）

功能：为应用软件提供网络服务，如文件传输、电子邮件和远程登录。

代表协议：HTTP（超文本传输协议）、FTP（文件传输协议）、SMTP（简单邮件传输协议）、DNS（域名系统）、DHCP（动态主机配置协议）、SNMP（简单网络管理协议）。

8、 简述分层模型工作机制。



(1) 封装过程：

应用层：将数据转换为二进制，形成报文；

传输层：添加 TCP/UDP 头，标识端口号，形成报文段；

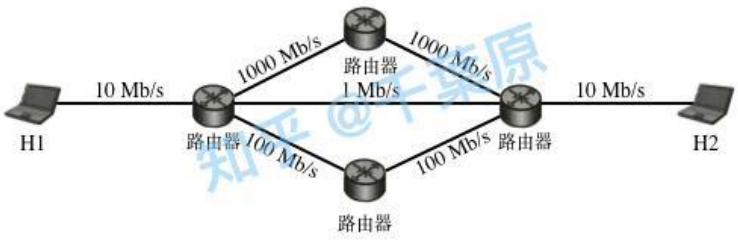
网络层：添加 IP 头，包含 IP 地址，形成数据报；

数据链路层：添加 MAC 头，注明 MAC 地址，形成帧；

物理层：转化为电信号传输。

(2) 解封装过程：与封装过程相反，逐步去除各层头部，恢复原始数据。

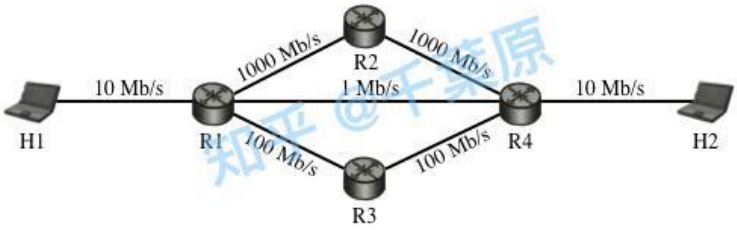
9、 若某分组交换网络及每段链路的带宽如下图所示，请计算 H1 到 H2 的最大吞吐量。



串连链路的最大吞吐量为该串连链路中最小带宽。

并连链路的最大吞吐量为该并连链路的所有分支链路的最大吞吐量总和。

为了方便讨论，将四个路由器分别命名为 R1、R2、R3 和 R4，如下图所示。结点 u 和 v 之间的带宽记为$c(u,v)$，例如，$c(H1,R1)=10Mb/s$。



H1 到 H2 的最大吞吐量

$= min\{c(H1,R1), min\{c(R1,R2),c(R2,R4)\} + c(R1,R4) + min\{c(R1,R3),c(R3,R4)\}, c(R4,H2)\}$

$= min\{10Mb/s, min\{1000Mb/s, 1000Mb/s\} + 1Mb/s + min\{100Mb/s, 100Mb/s\}, 10Mb/s\}$

$= min\{10Mb/s, 1101Mb/s, 10Mb/s\}$

$= 10Mb/s$

10、甲向乙发送数据$M=10011010$，采用 CRC 校验，生成多项式为$G(x)=x^3+x^2+1$（即$G=1101$，冗余比特数$k=3$），若在传输过程中没有发生错误，请计算乙接收到的比特串。

- x^k乘$M(x)$，即消息末尾加上k个 0，得到零扩展消息$T(x)=10011010000$
- $T(x)$除以$G(x)$，得到余数$S(x)=101$

```
        11111001
1101)10011010000
     1101
      1001
      1101
       1000
       1101
        1011
        1101
         1100
         1101
          0010
          0000
           0100
           0000
            1000
            1101
             101
```

- $P(x)=T(x)-S(x)=10011010000-101=10011010101$

第 2 章 直连和交换网络

1、什么是曼彻斯特码？什么是差分曼彻斯特码？优缺点？

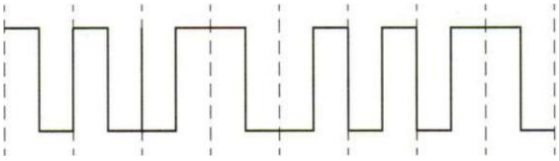
(1) 曼彻斯特码：一个周期的方波表示“1”，反向波形表示“0”。

(2) 差分曼彻斯特码：相邻周期的方波反相表示“1”，同相表示“0”；采用差分码的概念，不用绝对电平值，而用相对值表示，避免极性反转引起的解码错误。

(3) 优点：1) 码元周期的中间部分存在电平跳变，易于提取时钟，不受信源统计特性的影响；2) 方波周期内，正负电平各一半，不存在直流分量。

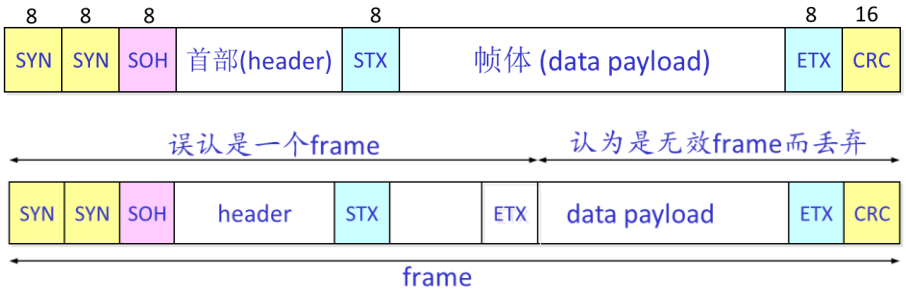
(4) 缺点：频带加倍，比特率为波特率（信号变化速率）的一半，编码效率仅 50%。

2、 **若下图为一段差分曼彻斯特编码信号波形，则其编码的二进制位串是（A）。**

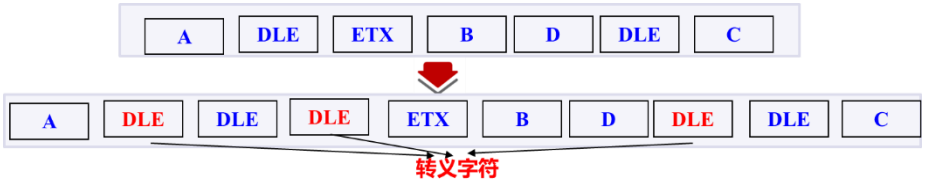


A. 1011 1001　　B. 1101 0001　　C. 0010 1110　　D. 1011 0110

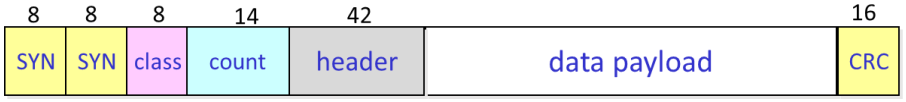
3、 **二进制同步通讯协议 BISYNC（面向字节的组帧：起始标记法）的帧结构如下图所示，其中 STX 代表正文开始符，ETX 代表正文结束符，如果数据负载中也包含 ETX 字符如何处理？**



字符填充法：引入转义符 DLE，将数据负载中所有的 ETX，用 DLE 进行转义。

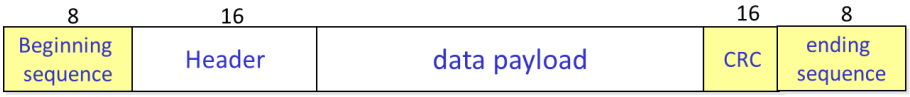


4、 **数字数据通信消息协议 DDCMP（面向字节的组帧：起始标记法）的帧结构如下图所示，它的缺点是什么？**



若 Count 出错，可能会产生累计多个错误。

5、 **由 IBM 开发的同步数据链路控制（SDLC）协议（面向比特的组帧）的帧结构如下图所示，其中 Beginning Sequence 和 Ending Sequence 为 01111110，如何解决 01111110 可能出现在帧的任何地方的问题？**



比特填充法：

- 发送方：当一串比特流尚未加上标志字段时，先用硬件扫描整个帧（用软件也可实现，但要慢一些），只要发现 5 个连续的 1，则立即填入一个 0
  例：发方发送 011111111110，则实际发送 0111110111110**0**
- 接收方：先找到开始点 01111110，然后再定结束点，在接收了连续的 5 个 1 之后，就可以根据下一位决定是填充位还是结束到达，下一位为：
  0：必为填充的 0，丢弃此位；
  1：再看下一位：若为 0，则为帧结束；若为 1，则出错（表示出现连续 7 个 1）。

6、 静态划分信道采用什么技术？其原理是什么？具体可分为哪些方式？

信道复用技术（Multiplexing）：为多个用户静态划分逻辑信道，相互不冲突。

频分复用、时分复用、波分复用、码分复用

7、 什么是载波侦听？什么是多点接入？载波侦听多点接入（Carrier Sense Multiple Access, CSMA）可分为哪几类？是否可以彻底解决碰撞问题，为什么？

(1) 载波侦听：

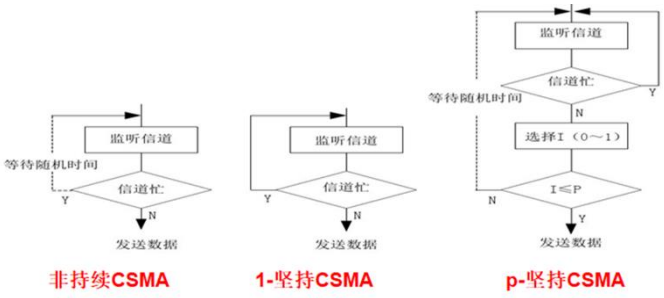
- 结点在发送前先检测信道，是否有其它结点也在发送，若有，则暂时不要发送数据，以免发生碰撞。
- 若结点具备同时收发数据的能力，在发送数据的同时，也要监听信道，判断是否发生碰撞，若发生，立即停止发送，并按一定策略重发数据。

(2) 多点接入：广播链路，多个结点以多点接入的方式连接在一链路，同一冲突域。

(3) 非持续 CSMA：可减少碰撞，会导致信道利用率降低，较长的延迟。

1-坚持 CSMA：会导致较多的碰撞，导致性能降低。

p-坚持 CSMA：通过调节p在减少碰撞和高信道利用率之间取得平衡。



(4) 无法彻底解决碰撞问题，因为信号传播时延导致。

8、 下列介质访问控制方法中，可能发生冲突的是（B）。

A. CDMA　　B. CSMA　　C. TDMA　　D. FDMA

ACD 分别是码分多路复用、时分多路复用、频分多路复用。这 3 种都属于静态媒体接入控制解决发送信息的次序问题，不会发生冲突。只有动态媒体接入控制会发生冲突，比如：ALOHA、时隙 ALOHA、CSMA、CSMA/CD 和 CSMA/CA 协议。

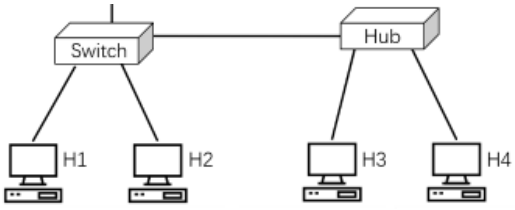
9、 集线器、交换机和路由器中哪些能够隔离碰撞域？哪些能够隔离广播域？

隔离碰撞域：交换机、路由器

隔离广播域：路由器

10、若主机 H2 向主机 H4 发送 1 个数据帧，主机 H4 向主机 H2 立即发送一个确认帧，则除 H4 外，从物理层上能够收到该确认帧的主机还有（D）。

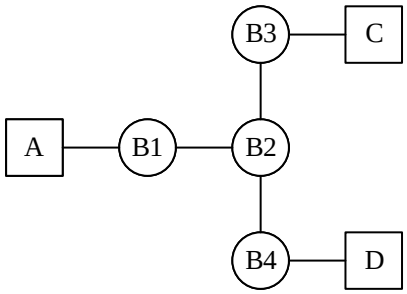
A. 仅 H2　　B. 仅 H3　　C. 仅 H1、H2　　D. 仅 H2、H3



11、考察下图中示出的透明桥接器的布局。假定开始时所有的转发表都是空的，试给出在下列的传输序列之后，桥接器 B1-B4 中的每一个的转发表的内容：

***A 给 C 发送　　*C 给 A 发送　　*D 给 C 发送**

要求在表中用可以从一个端口可以直接到达的那个邻居结点来标识该端口，例如，B1 的两个端口可标识为 B1 的 A 端口和 B1 的 B2 端口。



*A 给 C 发送

B1		B2		B3		B4	
主机	端口	主机	端口	主机	端口	主机	端口
A	A	A	B1	A	B2	A	B2

*C 给 A 发送

B1		B2		B3		B4	
主机	端口	主机	端口	主机	端口	主机	端口
A	A	A	B1	A	B2	A	B2
C	B2	C	B3	C	C		

*D 给 C 发送

B1		B2		B3		B4	
主机	端口	主机	端口	主机	端口	主机	端口
A	A	A	B1	A	B2	A	B2
C	B2	C	B3	C	C	D	D
		D	B4	D	B2		

注：在 FDB 表为空的时候，初始收到某个数据包且表中没有的时候，会选择广播给相邻的节点。而如果表中有的话，就是选择这条记录然后单播。

12、直连网络、交换网络、互联网络是三种不同的网络形态，请回答以下问题：(1) 描述三种网络形态的基本组网形式及所采用的典型网络设备；(2) 分析直连网络发展到交换网络、以及交换网络发展到互联网络的原因。

(1) 直连网络：

- 基本组网形式：设备之间采用一对一的直接物理连接，每个节点都通过单独的连接线缆与其他节点相连。
- 典型网络设备：网卡、网线等。

交换网络：

- 基本组网形式：所有设备通过交换机（Switch）相连。交换机负责根据数据帧的目标地址，将数据准确地转发给相应的设备。
- 典型网络设备：交换机、网卡、网线等。

互联网络：

- 基本组网形式：通过路由器（Router）将多个交换网络或子网互连，形成广域网（WAN）或更大规模的网络（如互联网）。
- 典型网络设备：路由器、交换机、防火墙、网关等。

(2) 直连网络适用于简单小规模互连，交换网络解决了局域网内设备拓展和高效通信的问题，而互联网络进一步满足了大规模、多样化网络之间的互联互通需求。

13、最小生成树机制能够在有环的物理网络中构造出一个无环的树状逻辑拓扑，在保证网络连通性的同时避免广播风暴。当网络是静态时，即不存在节点/链路的加入和删除，网络对应的最小。生成树拓扑是否是唯一的？请简述原因或举例说明。

构造的最小生成树拓扑可能不唯一，但是最小生成树的权值是唯一的。当无向图中存在相同权值的边，并且该边的权值小于最小生成树中边集的最大权值，则一定存在多个最小生成树。举例：一个正方形网络中，每个顶点各有一个节点，每条边的权重是 1，对于某个点，它的最小生成树拓扑有两个。

第 3 章 网络互联

1、 什么是 IP 分片？其过程是什么？

(1) IP 分片是在网络层（IP 层）进行的，它将大的 IP 数据报分割成多个较小的片段，以适应不同网络中的 MTU 限制。MTU（Maximum Transmission Unit，最大传输单元）是指在某个网络中能够传输的最大数据包大小。当一个 IP 数据报的大小超过了网络的 MTU 限制时，就需要进行分片。

(2) IP 分片的过程如下：

- 发送端的网络层检查待发送的 IP 数据报的大小是否超过了网络的 MTU 限制。
- 如果超过了 MTU 限制，发送端的网络层将原始的 IP 数据报分割成多个片段。
- 对于每个片段，发送端的网络层设置相应的标识、偏移量和标志位等字段。
- 发送端依次发送这些片段到网络中。
- 接收端的网络层接收到这些片段后，根据标识和偏移量等字段进行排序和组装。
- 接收端的网络层将组装好的 IP 数据报交给上层的协议进行处理。

2、 什么是 ARP 协议？其使用过程是什么？

(1) 数据链路层发送数据需要接收方 IP 地址和 MAC 地址，发送方 IP 地址和目的地址。但是，有时候仅有接收方的 IP 地址，而没有接收方的 MAC 地址，ARP（Address Resolution Protocol，地址解析协议）就是通过接收方的 IP 地址查询其 MAC 地址。

(2) ARP 协议使用过程：

网络中的设备都有一个 APR 缓存表，存储了其它网络设备 IP 地址与 MAC 地址的对应关系。当发送设备向目标设备发送信息时，首先会检索 APR 缓存表来查找目标设备的 MAC 地址。若缓存表中没有存储，发送设备会在本地网络上广播发送一个 ARP 请求，目标设备接受后以单播方式进行 ARP 应答。这样发送设备便获得了目标设备的 MAC 地址，可以发送信息，同时也会更新自己的 ARP 缓存表。

3、 举例几个特殊用途的 IPv4 地址。

(1) 主机号全 0，表示本网络的网络地址；

(2) 主机号全 1，表示本网络的广播地址；

(3) 32 位全 0，即 0.0.0.0，表示本网络上的本主机；

(4) 32 位全 1，即 255.255.255.255，表示整个网络的广播地址；

(5) 127.X.X.X，保留为环回自检地址，此地址表示任意主机本身。

4、 能不能将域内路由协议直接应用到域间路由？

不能。域内路由（Intra-Domain）是性能目标导向的，全域有统一目标，根据最小化某种链路度量值（开销），找到一条最有路径；域间路由（Inter-Domain）是策略和经济目标导向的，每个 AS 有自己的策略。

5、 给定一个路由转发表如表 1 所示，路由器收到数据包后，按照最长前缀匹配方式查找 IP 地址的相应转发端口。

表 1 路由转发表示意图

Entry NO.	Prefix	Interface
1	192.168.128.0/24 = 1100 0000.1010 1000.1000 0000	A
2	192.168.128.0/20 = 1100 0000.1010 1000.1000	B
3	192.168.192.0/18 = 1100 0000.1010 1000.11	C
4	10.1.0.0/16 = 0000 1010.0000 0001	D
5	* / Default	E

对于如下 IP 地址列表，请写出每个地址对应的转发接口。

(1) 10.1.1.1 = 0000 1010.0000 0001.0000 0001.1 → D

(2) 10.10.1.1 = 0000 1010.0000 1010.0000 0001.1 → E

(3) 192.168.240.1 = 1100 0000.1010 1000.1111 0000.1 → C

(4) 192.168.136.1 = 1100 0000.1010 1000.1000 1000.1 → B

(5) 192.168.224.1 = 1100 0000.1010 1000.1110 0000.1 → C

(6) 192.168.128.1 = 1100 0000.1010 1000.1000 0000.1 → A

6、 RIP、OSPF、BGP 三个协议的不同。

(1) RIP（Routing Information Protocol，路由信息协议）：距离向量算法

- 通过仅与相邻路由器交换路由信息实现。
- 交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
- 定期或者触发交换路由信息。

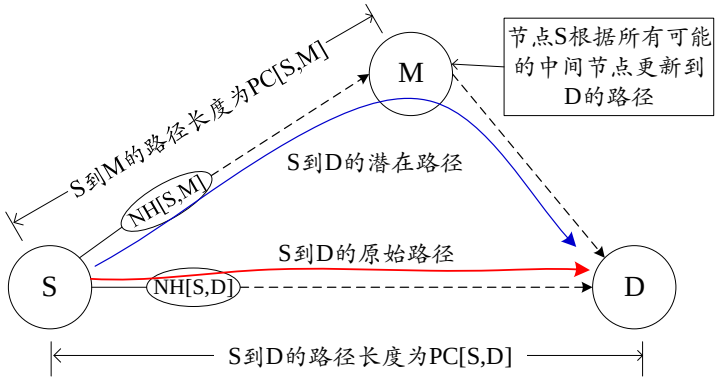
(2) OSPF（Open Shortest Path First，开放最短路由优先协议）：链路状态算法

- 向本自治系统中所有路由器发送信息。
- 发送的信息只是与本路由器相邻的路由器之间的链路状态（接口及其与邻近路由器之间关系的描述，如接口的 IP 地址、掩码、链路类型、到连接路由器的开销等）。
- 只有当链路状态发生变化时，路由器才用洪泛法向所有路由器发送此信息。
- 各结点根据链路状态数据库单独计算它到其它结点的最短路径，生成路由表。

(3) BGP（Border Gateway Protocol，边界网关协议）：路径向量算法

- 不同自治系统的路由器之间交换路由信息。
- 每个路由更新中携带整条路径信息（路径是指途经的 AS 的集合）。
- 当收到至某 AS X 的路由通告，根据本地策略，选择到 X 的较好路径，并决定向哪些 AS 通告该路由更新。

注：RIP 和 BGP 类似三角路由更新算法。



7、 OSPF 路由协议的核心是链路状态机制，请简述该协议的运行原理，拓扑变动后的收敛过程以及如何拓展到规模更大的网络环境的方法。

(1) 运行原理：

OSPF 设备向外发送 hello 包，与其他使用相同协议的设备建立邻居关系。

互相发送 LSA（链路状态通告）相互通告路由，形成 LSDB（链路状态数据库）。

基于 LSDB 通过 SPF 算法，计算通向每个目的网络的最佳路径后，放入路由表。

(2) 收敛过程：

进行 LSA 的泛洪。

收集泛洪的 LSA 集合，进行 LSDB 的组建。

基于 LSDB 用 SPF 算法得到一颗以自己为“根”覆盖全网的无环的树。

(3) 拓展方法：

将一个自治系统再划分为若干个更小的范围，叫区域，使更新过程收敛更快。

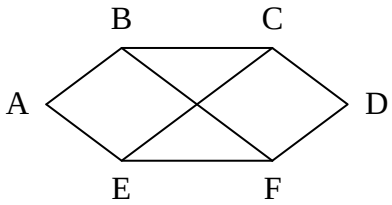
8、 考虑下图所示的子网。使用距离向量路由选择，下列向量刚刚被路由器 C 收到：

来自 B：(5,0,8,12,6,2)

来自 D：(16,12,6,0,9,10)

来自 E：(7,6,3,9,0,4)

路由器 C 测量得到的到达 B、D 和 E 的延时分别等于 6、3 和 5。试问路由器 C 的新的路由表是什么？请给出所使用的输出线路和所预期的延时。



此时链路中的距离向量矩阵如下（空格表示未知）：

	A	B	C	D	E	F
A	0					
B	5	0	8	12	6	2
C		6	0	3	5	
D	16	12	6	0	9	10
E	7	6	3	9	0	4
F						0

根据距离向量算法：

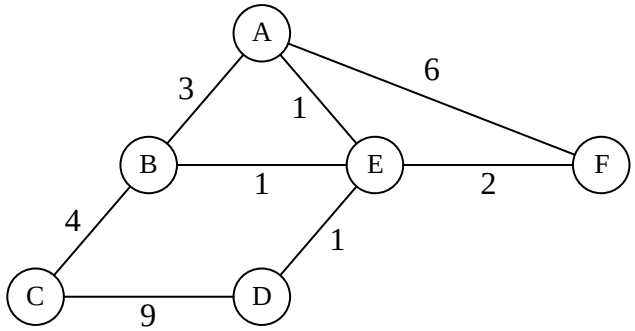
	A	B	C	D	E	F
C→B→	11	6	14	18	12	8
C→	∞	6	0	3	5	∞
C→D→	19	15	9	3	12	13
C→E→	12	11	8	14	5	9

因此，C 此时的路由表及输出线路为：

目的主机	代价	下一主机	路径
A	11	B	C→B→A
B	6	B	C→B

C	0	C	C
D	3	D	C→D
E	5	E	C→E
F	8	B	C→B→F

9、 考虑下图中的网络拓扑，每条边上的数字代表对应链路的代价，使用距离向量方法进行网络路由选择，当前节点 A 和 C 的路由表项分别如下两个子表所示，当两个节点分别收到节点 B 的距离向量信息（A:2, B:0, C:4, D:2, E:1, F:3）后，试更新两个节点的路由表项。



Dest	Cost	Next Hop
A	0	A
B	2	E
C	7	B
D	2	E
E	1	E
F	3	E

(a) 节点 A 的路由表项

Dest	Cost	Next Hop
A	7	B
B	4	B
C	0	C
D	9	D
E	5	B
F	∞	-

(b) 节点 C 的路由表项

(a) 对于节点 A：

Dest	Old Path	Old Cost	New Path	New Cost
C	A→B→C	7	A→E→B→C	2+4=6
D	A→E→D	2	A→E→B→E→D	2+2=4
E	A→E	1	A→E→B→E	2+1=3
F	A→E→F	3	A→E→B→E→F	2+3=5

更新后的节点 A 的路由表项：

Dest	Cost	Next Hop
A	0	A
B	2	E
C	6	E
D	2	E
E	1	E
F	3	E

(b) 对于节点 C：

Dest	Old Path	Old Cost	New Path	New Cost
A	C→B→A	7	C→B→E→A	4+2=6

D	C→D	9	C→B→E→D	4+2=6
E	C→B→E	5	C→B→E	4+1=5
F	-	∞	C→B→E→F	4+3=7

更新后的节点 C 的路由表项：

Dest	Cost	Next Hop
A	6	B
B	4	B
C	0	C
D	6	B
E	5	B
F	7	B

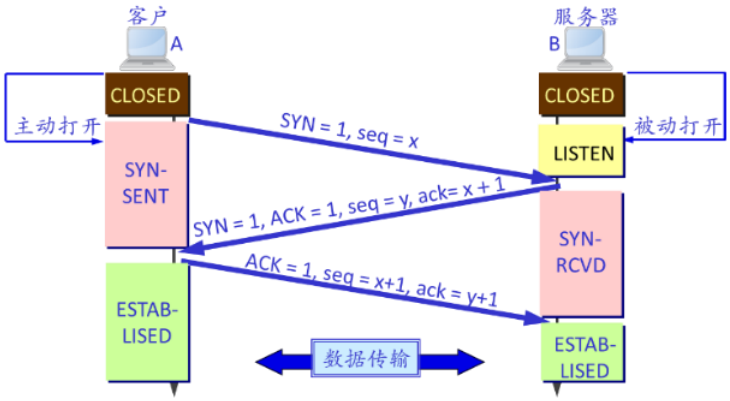
第 4 章 端到端传输

1、 简述建立 TCP 连接的过程。

[1] 第一次握手（SYN）：客户端向服务器发送一个 SYN 报文段，其中包含一个随机生成的初始序列号（seq=x），表示请求建立连接。客户端进入 SYN-SENT 状态。

[2] 第二次握手（SYN-ACK）：服务器收到客户端的 SYN 报文段后，确认收到连接请求并准备建立连接，回复一个 SYN-ACK 报文段。该报文包含服务器生成的序列号（seq=y）以及对客户端序列号的确认（ack=x+1）。服务器进入 SYN-RECEIVED 状态。

[3] 第三次握手（ACK）：客户端收到服务器的 SYN-ACK 报文后，再发送一个 ACK 报文，确认服务器的 SYN 报文（ack=y+1）。此时，客户端和服务器都进入 ESTABLISHED 状态，表示连接已建立，可以开始数据传输。



2、 为什么建立 TCP 连接需要第三次握手？

防止已失效的连接请求报文导致资源浪费：

假设只有两次握手，若客户端发送的第一个 SYN 包因网络延迟等问题，在长时间后才到达服务器，而此时客户端可能已经重新发送了新的 SYN 包并建立了连接，甚至可能已经完成数据传输并关闭了连接。如果按照两次握手的机制，服务器收到延迟到达的 SYN 包后会认为这是一个新的连接请求，从而分配资源建立连接并等待数据传输，但实际上客户端并不需要这个连接，这就导致服务器资源被无端浪费。

3、 对于滑动窗口协议，若分组序号采用 3 比特编号，发送窗口大小为 5，则接收窗口最大是（B）。

A. 2　　B. 3　　C. 4　　D. 5

发送窗口 + 接收窗口 $\leq 2^n$

4、 TCP 流量控制和拥塞控制的目的以及实现原理。

(1) 流量控制（Flow Control）：

目的：防止快发送方给慢接收方发数据造成接收崩溃，缓冲区溢出

实现原理：

- 接收方：根据缓存大小确定接收窗口 AdvertisedWindow 大小，并通知发送方
- 发送方：发送窗口上限值(MaxWindow) ≤ 接收窗口上限值(AdvertisedWindow)

MaxWindow = MIN (CongestionWindow, AdvertisedWindow)

(2) 拥塞控制（Congestion Control）：

目的：在网络不拥塞的情况下，尽可能快速发送，高效、公平利用网络资源

实现原理：发送端感知网络状况，计算 cwnd（拥塞窗口）与 pacing rate（发送速率）

5、 滑动窗口（sliding-window）算法相较停等（stop-and-wait）算法的优势是什么？滑动窗口算法中的发送窗口大小与可用序列号数需要满足什么关系？

(1) 增加了单位时间传输数据帧的数目，提升了传输速率；

允许多个在途传输（未收到 ACK）的数据帧。

(2) 发送窗口大小不能大于可用序列号数的一半。

6、 在标准 TCP 实现中，TCP 连接空闲多长时间就会在下次发送数据包时，触发慢启动（简称 SSAI，Slow Start After Idle）？请简述此时重新从慢启动开始的原因。是否可以把 SSAI 直接关闭掉，请简述原因。

(1) 1 个 RTO。

(2) 重启慢启动是需要重新探索可用带宽。

(3) 两种答案：1) 可以，因为这样可以节省吞吐率从 0 增长到可用带宽的时间；2) 不可以，因为在 1 个 RTO 时间内，有可能有新建的流占用了已有带宽，这时还用原来的窗口大小（发送速率）会导致拥塞。（回答任一一种即可）

第 5 章 下一代网络协议与应用

1、 下列关于 IPv6 和 IPv4 的叙述中，正确的是（D）。

I. IPv6 地址空间是 IPv4 地址空间的 96 倍

II. IPv4 和 IPv6 的基本首部的长度均可变

III. IPv4 向 IPv6 过渡可以采用双协议栈和隧道技术

IV. IPv6 首部的 Hop Limit 等价于 ipv4 首部的 TTL 字段

A. 仅I、II　　B. 仅I、IV　　C. 仅II、III　　D. 仅III、IV

I. 错误。IPv6 使用 128 位地址，而 IPv4 使用 32 位地址。因此，IPv6 地址空间是 IPv4 地址空间的$2^{128}/2^{32} = 2^{96}$倍。

II. 错误。IPv4 的基本首部长度是可变的，为 20~60 个字节，其中固定部分 20 字节，可变部分 0~40 字节，而 IPv6 的基本首部长度是固定的，为 40 个字节。

III. 正确。IPv4 向 IPv6 过渡可以采用双协议栈和隧道技术，这允许 IPv4 和 IPv6 网络共存，并允许在它们之间进行通信。

IV. 正确。IPv6 首部的 Hop Limit 字段被用来限制一个数据包可以经过的路由器数量，这类似于 IPv4 首部的 TTL 字段的作用。

2、 简述 IPv6 的主要特点。

(1) 更大的地址空间：IPv6 使用 128 位地址，相比于 IPv4 的 32 位，极大地扩展了地址空间。这意味着理论上可以分配几乎无限数量的独特地址。

(2) 改进的路由和数据包处理：IPv6 简化了数据包头部结构，使得路由器处理数据包更加高效。

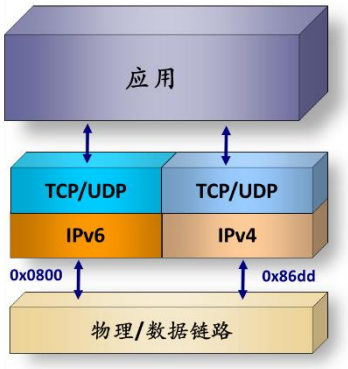
(3) 内置的安全功能：IPv6 设计时考虑了安全性，包括对 IPSec 的支持，这是一种网络层安全协议。

(4) 更好的移动性和多播支持：IPv6 提供了更好的支持以适应移动设备和多播应用。

3、 IPv4 到 IPv6 的过渡技术主要有双协议栈（Dual Stack）、隧道技术（Tunnel）和网络地址转换-协议地址转换技术（Network Address Translation-Protocol Translation, NAT-PT）三种技术方案，简述它们的实现方式。

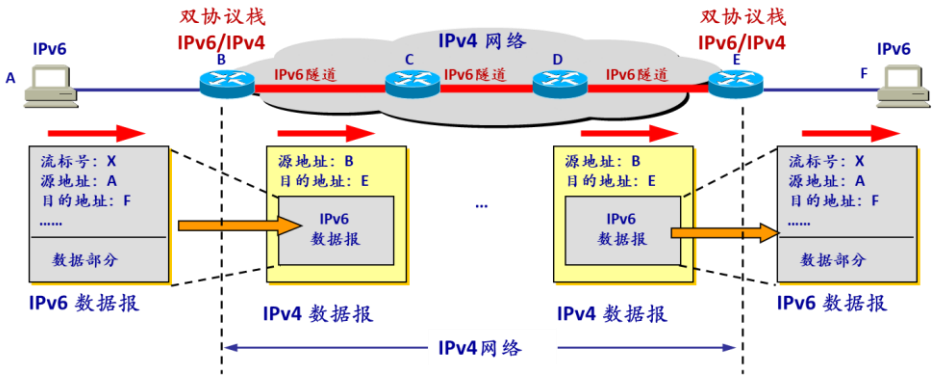
(1) 双协议栈

双栈方法就是主机和路由器在同一网络接口上运行 IPv4 栈和 IPv6 栈。



(2) 隧道技术

- 在 IPv6 数据报要进入 IPv4 网络时，把 IPv6 数据报封装成为 IPv4 数据报，整个的 IPv6 数据报变成了 IPv4 数据报的数据部分。
- 当 IPv4 数据报离开 IPv4 网络中的隧道时，再把数据部分（即原来的 IPv6 数据报）交给主机的 IPv6 协议栈。



(3) 网络地址转换-协议地址转换技术

IP 网络地址翻译器（NAT）的地址翻译机制和无状态 IP/ICMP 翻译器（SIIT）的 v6/v4 协议翻译机制的结合。

4、 简述将 48 位的以太网 MAC 地址转换为 128 位的 IPv6 地址的转换过程。

(1) 地址分割与插入

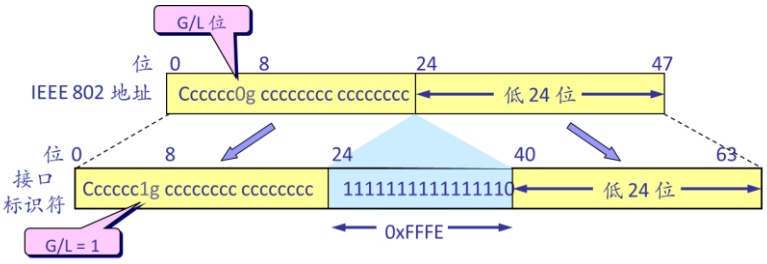
将 48 位的 MAC 地址从中间分开，例如 00-1A-2B-3C-4D-5E，然后插入 FFFE，形成 00-1A-2B-FF-FE-3C-4D-5E。

(2) 反转第 7 位比特位

对地址的第 7 位进行反转操作，即通常所说的“取反”。若该位原来为 1，则变为 0；若原来为 0，则变为 1。转换后，MAC 地址的局部（Local）或全局（Universal）属性改变，以匹配 EUI-64 标准。

(3) 附加网络前缀

将 64 位前缀（通常由 ISP 提供或是本地网络配置）加在此 64 位地址前面，组成了一个完整的 IPv6 地址。



IPv4地址构成

IPv6地址构成

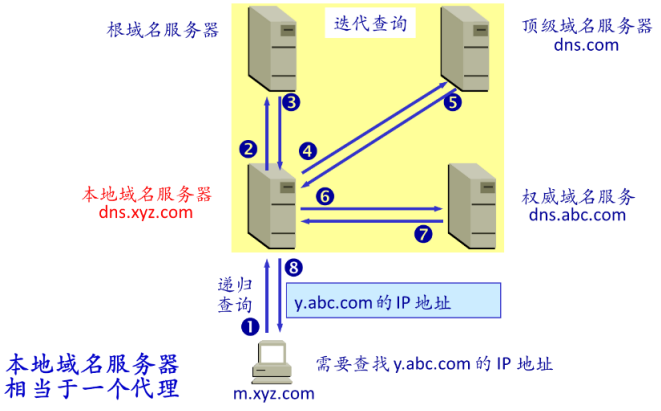
→没有类的概念

5、 根域名服务器的作用，递归查询和迭代查询的原理。

(1) 作用：根域名服务器是最高层次的域名服务器，也是最重要的域名服务器，所有的根域名服务器都知道所有的顶级域名服务器的域名和 IP 地址。

(2) 递归查询：当 DNS 客户端向本地 DNS 服务器发起查询请求时，如果本地 DNS 服务器无法解析该域名，它会代替客户端向上级 DNS 服务器（如根服务器、顶级域服务器等）发起查询，直到获得最终结果或查询失败。

(3) 迭代查询：当 DNS 客户端向本地 DNS 服务器发起查询请求时，如果本地 DNS 服务器无法解析该域名，它会告诉客户端下一个可能知道答案的 DNS 服务器的地址，让客户端自己去查询。这个过程会一直迭代下去，直到找到能够解析该域名的 DNS 服务器。



6、 请描述“在浏览器中输入网址，到取回网页”这段时间内发生的网络操作。

[1] 浏览器分析超链指向页面的 URL

[2] 浏览器向 DNS 请求解析域名的 IP 地址

[3] 域名系统 DNS 解析出域名服务器的 IP 地址

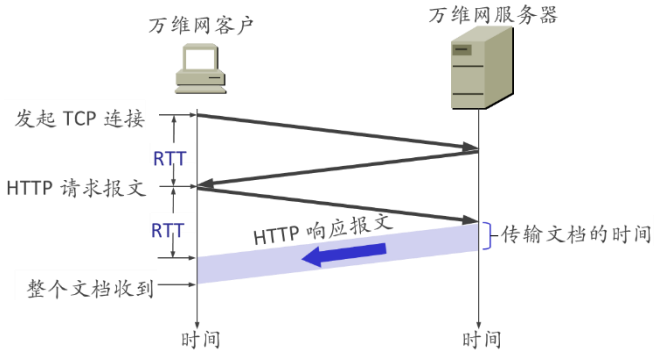
[4] 浏览器与服务器建立 TCP 连接

[5] 浏览器发出请求

[6] 服务器返回响应

[7] 释放 TCP 连接

[8] 浏览器显示获取的文本

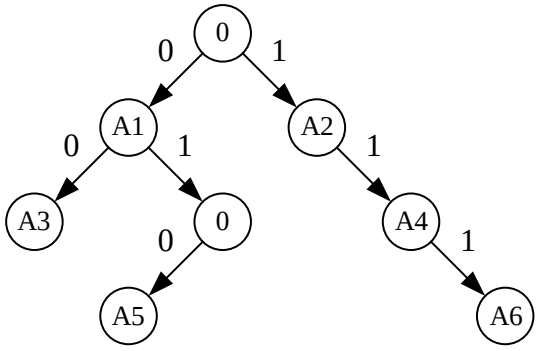


注：请求一个万维网文档所需的时间 ＝2 个 RTT＋ 文档传输时间

7、 基于路由转发表表项进行最长前缀，确定下一跳转发路径，是路由查找转发的基本动作。Trie 查找是主要路由查找算法之一，下图(a)是路由表示例，图(b)是对应的 Trie 数据结构。请描述 IPv4 路由查找中 Trie 查找过程，分析时间复杂度与空间复杂度，以及基于 IPv6 的路由表的查找算法的优化方法。

Prefix	Next
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

(a) 转发表表项



(b) 转发表对应的 Trie 数据结构

(1) 查找过程：

[1] 从根结点开始一次搜索；

[2] 取得要查找关键词的第一个字符，并根据该字符转到对应的子树进行检索；

[3] 在相应的子树上，取得要查找关键词的第二个字符，并进一步选择对应的子树进行检索；

[4] 迭代过程……

[5] 在某个结点处，关键词的所有字符已被取出，则读取附在该结点上的信息，即完成查找。

(2) 时间复杂度：Trie 树的时间复杂度取决于要查找的 IP 地址的长度，这是因为在最坏的情况下，需要遍历整个 IP 地址的长度来确定最长匹配前缀。由于 IPv4 地址的长度为 32，所以查找的时间复杂度为O(1)。

(3) 空间复杂度：Trie 树的空间复杂度取决于存储的路由前缀数量和每个前缀的长度，对于具有n个不同前缀的路由表（$n \leq 2^{32}$），每个前缀平均长度为l位（$l \leq 32$），查找的空间复杂度为$O(n \times l) = O(1)$。

(4) 基于 IPv6 的路由表的查找算法：基于前缀 hash 表的二分搜索
优化方法：1) 降低时间复杂度：Sail；2) 降低空间复杂度：DxR；3) 在时空复杂性上进行折衷考虑和优化：Poptrie。

第 6 章 传输层协议分析与优化

1、 阐述三种拥塞控制机制（基于丢包的拥塞控制、基于延迟的拥塞控制、基于模型的拥塞控制）的基本原理，并从性能、公平性、环境适应性等方面分析每种拥塞控制机制的优劣势。

	基于丢包的拥塞控制	基于延迟的拥塞控制	基于模型的拥塞控制
基本原理	通过检测数据包的丢失来判断网络拥塞。当网络出现拥塞时，路由器的队列缓冲区溢满，导致数据包丢失。发送方在接收到丢包信号时，会降低发送速率。	通过监测RTT的变化来判断网络拥塞。当网络流量增加时，队列长度增加，导致往返时延增大。发送方检测到RTT 增加后，会降低发送速率来缓解拥塞。	通过数学模型或机器学习算法来预测网络状态，并根据模型的预测结果动态调整发送速率。模型考虑网络参数（如带宽、时延、丢包率）来优化拥塞控制决策。
性能	高带宽、低延迟下表现好	低丢包网络中表现平稳	各种网络条件下表现稳定
公平性	同类算法间公平	与基于丢包的算法共存时不公平	可通过模型设计实现公平
环境适应性	适用于传统有线网络	适用于低丢包、高延迟网络	适应多种复杂网络环境
缺点	不适合高丢包或无线网络	无法有效处理丢包信号	依赖模型精度，复杂度高

2、 TCP 拥塞控制机制，包括慢启动、拥塞避免、快速重传、快速恢复等功能，这些功能共同完成了数据流的高可靠和高性能传输，请简述每种功能的原理和设计目标。

(1) 慢启动（Slow Start）

- 原理：慢启动是一种指数增长的拥塞控制算法。在 TCP 连接建立后，发送方以一个较小的窗口（通常是 1 个 MSS，即最大报文段长度）开始发送数据，每收到一个 ACK（确认），窗口大小就增加 1 个 MSS，直到达到慢启动阈值（ssthresh）。
- 设计目标：慢启动的目的是渐进地增加发送速率，以避免在连接建立初期就发送大量数据导致网络拥塞。

(2) 拥塞避免（Congestion Avoidance）

- 原理：当窗口大小超过慢启动阈值后，TCP 进入拥塞避免阶段。在这个阶段，窗口大小以线性方式增长，即每收到一个 ACK，窗口大小增加 1 个 MSS 的 1/拥塞窗口大小。
- 设计目标：拥塞避免旨在在不引起网络拥塞的情况下，尽可能高效地利用网络带宽。

(3) 快速重传（Fast Retransmit）

- 原理：快速重传是一种快速响应丢包的机制。当发送方连续收到三个相同的重复 ACK 时，它会立即重传丢失的数据包，而不是等待重传计时器超时。
- 设计目标：快速重传旨在减少因丢包导致的延迟，提高数据传输的实时性。

(4) 快速恢复（Fast Recovery）

- 原理：快速恢复是与快速重传配合使用的机制。当触发快速重传时，拥塞窗口设置为慢启动阈值+3，并重传丢失的数据包；如果再收到重复的 ACK，则拥塞窗口值+1；如果

收到新数据的非重复 ACK 后，表明丢失的数据包已被成功接收，网络拥塞状况有所缓解，此时把拥塞窗口值设置为慢启动阈值，结束快速恢复阶段，进入拥塞避免阶段。

- 设计目标：快速恢复的目的是减少因丢包导致的连接速率下降，快速恢复到较高的传输速率。

3、 Timeout Retransmission（超时重传）对 TCP 传输性能的影响体现在哪几个方面？为什么说超时重传对高大宽、高时延（RTT）的网络影响最大？是否可以直接减小 RT0（Recovery Time Objective，恢复时间目标）时间来减小超时重传的影响？

(1) RTO 可能是 RTT 的几个数量级以上，在 RTO 时间内不能传输数据，因此将会使发送端经较长时间等待才能发现报文段丢失，降低了连接数据传输的吞吐量。此外，超时重传会导致进入 slow start。

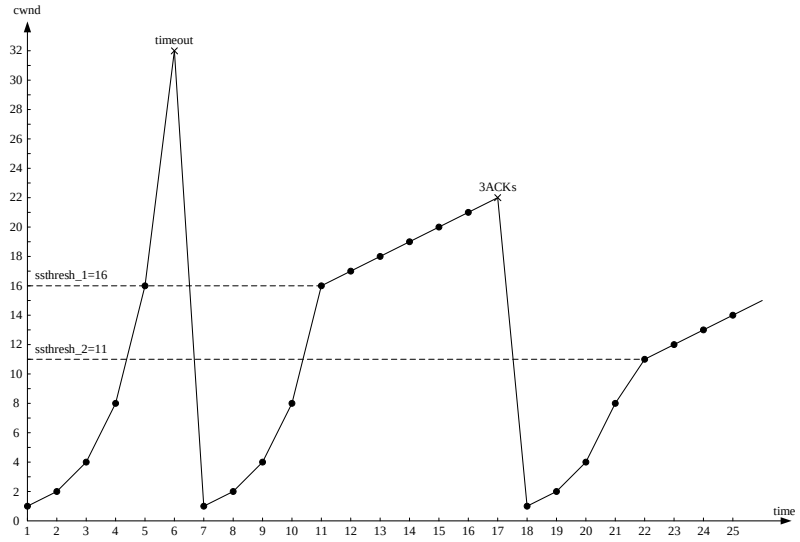
(2) TCP 根据得到的 RTT 值更新 RTO 值，发送端对每个发出的数据包进行计时，如果在 RTO 时间内没有收到所发出的数据包对应的 ACK，则认为数据包丢失，将重传数据，若 RTO 较大，则系统在长时间内无法发送数据包，此时若系统的带宽也很大，则造成了资源的大量浪费。

(3) 不能直接减小。因为若 RTO 过小，发送端尽管可以很快得检测出报文段的丢失，但也可能将一些延迟大的报文段误认为是丢失，造成不必要的重传，浪费了网络资源。

4、 假设一个 TCP 连接总是以 1KB 的最大段发送 TCP 报文段，且发送方有足够多的数据要发送，接收方有足够的接收能力（接收窗口足够大）。发送方以拥塞窗口为 1KB 开始发送，当拥塞窗口为 32KB 时发生了数据全部丢失而超时，在这之后的连续的 10 个 RTT（往返时间）时间内的 TCP 报文段的传输都是成功的，接着再往后因为一个 TCP 报文段传输时延过大而导致发送方接收到三个连续的重复确认。

(1) 采用 Tahoe、Reno 算法下，拥塞窗口、阈值、随着传输轮次（1 个 RTT 时间为 1 个轮次）的变化？

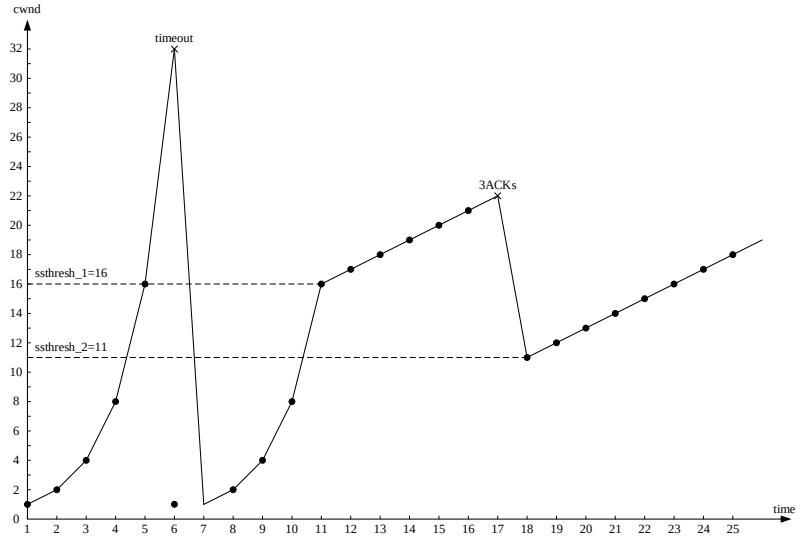
Tahoe 算法：



cwnd	1	2	4	8	16	32	1
ssthresh							16
n	1	2	3	4	5	6	7
cwnd	2	4	8	16	17	18	19
ssthresh							
n	8	9	10	11	12	13	14
cwnd	20	21	22	1	2	4	8
ssthresh				11			
n	15	16	17	18	19	20	21

拥塞避免

Reno 算法：



cwnd	1	2	4	8	16	32	1
ssthresh							16
n	1	2	3	4	5	6	7
cwnd	2	4	8	16	17	18	19
ssthresh							
n	8	9	10	11	12	13	14
cwnd	20	21	22	11	12	13	14
ssthresh				11			
n	15	16	17	18	19	20	21

拥塞避免

(2) 采用上述个算法，分别是第几轮次发送出第 168 个报文段？

Tahoe 算法：

- 第 20 轮次：1+2+4+8+16+1+2+4+8+16+17+18+19+20+21+1+2+4=164
- 第 21 轮次：1+2+4+8+16+1+2+4+8+16+17+18+19+20+21+1+2+4+8=172

Reno 算法：

- 第 16 轮次：1+2+4+8+16+1+2+4+8+16+17+18+19+20+21=157
- 第 18 轮次：1+2+4+8+16+1+2+4+8+16+17+18+19+20+21+11=168

5、 简述 TCP Reno 算法和 TCP NewReno 算法的区别。

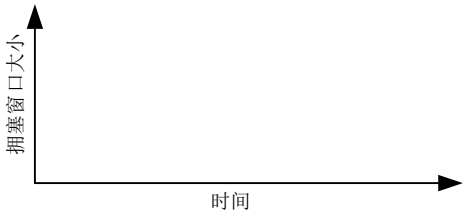
Reno 算法在快速恢复阶段收到一个新的数据包的 ACK 后就会退出快速恢复状态，进入拥塞避免阶段。这意味着如果在一个窗口内发生了多个数据包的丢失，Reno 算法可能需要多次进入和退出快速恢复状态，导致多次将 cwnd 和 ssthresh 减半，这会显著降低吞吐量。

NewReno 算法改进了这一点。在快速恢复阶段，NewReno 会持续重传丢失的数据包，直到所有丢失的数据包都被重传并收到确认后，才会退出快速恢复状态。这样，NewReno 可以更有效地处理单个窗口内多个数据包的丢失，避免了多次减半 cwnd 和 ssthresh 的问题。

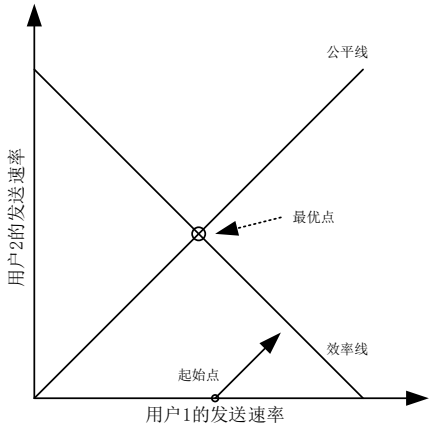
6、 什么是 SACK？相较于传统的 TCP 确认机制有何优势？

SACK（Selective Acknowledgment，选择性确认）是 TCP 协议的一个扩展，用于提高在高丢包率环境下的数据传输效率。在传统的 TCP 确认机制中，接收方只能确认已按序收到的最后一个字节（即累计确认）。当出现多个数据包丢失时，累计确认机制可能导致发送方重复发送已成功接收的数据包，降低传输效率。SACK 允许接收方通知发送方哪些数据块已成功接收，哪些数据块尚未收到，从而使发送方仅重传丢失的数据，提高网络带宽的利用率。

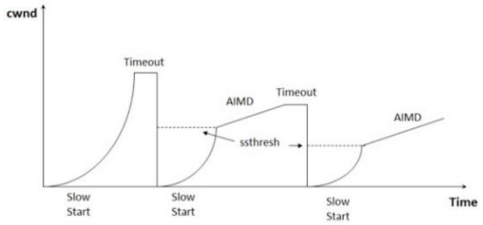
7、 TCP 通过 AIMD（Additive Increase Multiplicative Decrease，和性增，乘性减）机制来探测可用带宽和保障竞争流间的公平性。只考虑 AIMD 机制，试在如下图(a)中画出一个 TCP 流的拥塞窗口随时间变化的形状，并说明该形状的变化周期；对于两个竞争流，从如下图(b)中的起始点出发，在图中画出两个竞争流的发送速率收敛到最优点过程中的变化曲线。

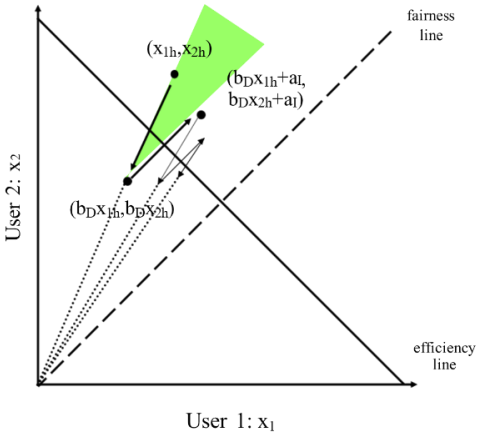


(a) AIMD 机制下拥塞窗口随时间的变化图



(b) 两条竞争流通过 AIMD 机制收敛到最优点





注：参考上图

8、 (1) 什么是主动测量，什么是被动测量，它们各自的优缺点？(2) 什么是链路带宽，什么是可用带宽？(3) 请简述一种测量可用带宽的经典算法的工作原理。(4) 打开一个视频网站发现在线视频的加载速度通常达不到家里宽带带宽，请分析各种可能的原因。

(1) 主动测量是主动发送探测数据进行测量，被动测量是被动捕获数据进行测量。前者更有针对性，但容易有偏采样，探测数据也可能对背景流量产生影响；后者不会对网络背景流量产生影响，但测量没有针对性。

(2) 链路带宽是指链路每秒钟传输的最大字节数；可用带宽是指网络在不降低其他业务流的传输速率的情况下，所能提供给一个业务流的最大传输速率。

(3) 1) 探测报文间隔模型：从发送端发送两个连续数据包，设数据包发送的时间间隔为 Tin，接收端接受这对数据包的时间间隔为 Tout，数据包大小为 L，网络最大带宽为 C，则可用带宽为 A=C*(1-(Tout-Tin)/Tin)；

2) 探测报文速率模型：当测试报文发送速率小于链路可用带宽时，传输时延相对固定，当测试报文发送速率大于链路可用带宽时，网络出现排队现象，传输时延增大，两种状态之间的那个点即为可用带宽。

(4) 1) 该视频网站和家里宽带不是一个运营商，受到限制；

2) 视频网站给用户做了限制以避免少数人占用了大多数人的资源；
3) 服务器的带宽不够了；
4) 其它进程占用了带宽；
5) 电脑硬件读取和解析的能力限制了加载速度。

9、 简述 Vegas 算法的优缺点。

(1) 优点：
- 精确测量 RTT，以 ms 为单位，颗粒度较细。
- 基于延迟变化，能够更早地感知拥塞，更加温和地调节速度，让速率靠近可用带宽。

(2) 缺点：
- 路径变化带来的 RTT 变化，误以为是拥塞，做出误动作。
- 在没有竞争时表现优异，几乎能完全利用带宽，但在有其他算法（CUBIC、Reno 等）协同工作时过于谦让（敏感且温和）。

10、BBR 是 Google 提出的一种拥塞控制算法，其核心思想是测量最小 RTT 和瓶颈链路的可用带宽，请简述为什么最小 RTT 和瓶颈链路的可用带宽不能同时测得，以及如何才能较为准确地测量这两个值。

(1) 原因：要测量最大带宽，就要把瓶颈链路填满，此时 buffer 中有一定的数据包，自然测不了最小 RTT；要测量最低延迟，就要保证 buffer 为空，网络中的数据包越少越好，这也与瓶颈链路带宽的测量条件相悖。

(2) 如何准确测量：针对测不准的问题，在论文中 BBR 算法采用的方案是，交替测量带宽和延迟，用一段时间内的带宽极大值和延迟极小值作为估计值，动态更新测量值，最终控制发送速率，避免网络拥塞。

11、简述 QUIC 算法的特点。

(1) 基于 UDP。

(2) 低延迟连接建立：通过将连接建立与加密握手结合，QUIC 能够显著减少连接建立的时间。通常情况下，QUIC 只需一次往返（1-RTT）即可完成握手。

(3) 内置加密：QUIC 默认使用 TLS 1.3 加密，提供与 HTTPS 同等的安全性，避免了传统 TCP 在加密层次上的额外开销。

(4) 多路复用：在一个 QUIC 连接中可以同时传输多个数据流，每个流都有独立的流量控制和错误恢复机制，避免了 TCP 中“队头阻塞”问题。

(5) 快速重传和恢复：QUIC 能够快速检测丢包并进行重传，改进了传统 TCP 的拥塞控制和丢包恢复机制。

(6) 更灵活的拥塞控制：QUIC 允许实现自定义的拥塞控制算法，提供更大的灵活性和优化空间。

第 7 章 应用层网络数据分发

1、 下列关于网络应用模型的叙述中，错误的是（B）。

A. 在 P2P 模型中，结点之间具有对等关系

B. 在客户/服务器 (C/S) 模型中，客户与客户之间可以直接通信

C. 在 C/S 模型中，主动发起通信的是客户，被动通信的是服务器

D. 在向多用户分发一个文件时，P2P 模型通常比 C/S 模型所需的时间短

A. 正确。P2P（Peer to Peer）网络，即对等网络，是一种在对等者（peer）之间分配任务和

工作负载的分布式应用架构，是对等计算模型在应用层形成的一种组网或网络形式，取消了服务器的中心地位，各个系统内计算机可以通过交换直接共享计算机资源和服务。在 P2P 模型中，结点之间具有对等关系，因此可以直接通信。

B. 错误。客户/服务器（Client/Server, C/S）模型，是以数据库服务器为中心、以客户机为网络基础、在信息系统软件支持下的两层结构模型。在 C/S 模型中，客户端与客户端无法直接通信，通信必须通过服务器进行。

C. 正确。在 C/S 模型中，通信一般是由客户发起的。

D. 正确。在 P2P 模型中，用户可以同时从多个其他用户那里下载文件，从而提高了下载速度。相比之下，在 C/S 模型中，用户只能从中心服务器下载文件，因此可能需要更长的时间来完成下载。

2、 CDN 依赖于 DNS 实现用户到服务器的映射，假设使用传统 DNS 协议，用户侧配置公用的 DNS 服务器（如 8.8.8.8）和使用运营商自动配置的本地 DNS 服务器会对这种映射造成什么影响？如何缓解这种影响？

(1) 影响：在 DNS 查询过程中，权威服务器接收请求的时候，只能得到本地 DNS 的 IP，并不知道客户端 IP。如果本地 DNS 设置不当，可能会误判用户的位置，将用户误导到错误的 CDN 缓存节点，造成加速效果差的问题。

(2) 缓解：可以利用 end-user mapping 的技术，通过客户端 IP 地址的前缀，来对客户端进行表示识别。

注：NS-based mapping 中，权威 NS 只知道 LDNS 的 IP；end-user mapping 中，权威 NS 还知道客户端的 IP

3、 CDN 通常根据客户端所使用的 DNS 服务器地址，来指定为客户端提供数据的 CDN 服务器。例如，根据国科大自动配置的 DNS 服务器，为国科大怀柔小区的客户端选择位于怀柔数据中心的服务器。请分析这种根据 DNS 服务器地址来为客户端选择 CDN 服务器的优劣势，以及如何改进来避免存在的劣势。

(1) 优势：当用户访问已经加入 CDN 服务的网站时，首先通过 DNS 重定向技术确定最接近用户的最佳 CDN 节点，同时将用户的请求指向该节点。

(2) 劣势：其实在 DNS 查询过程有一个这样的问题，权威服务器接收请求的时候，只能得到 Local DNS 的 IP，并不知道 client IP。一般如果 Local DNS 设置不当，例如没有使用当前 ISP 提供的 Local DNS 这种实现方式可能会误判用户的位置，从而将用户误导到错误的 CDN 缓存节点，造成加速效果差的问题。

(3) 改进：利用 end-user mapping 的技术，通过 client IP 地址的前缀，来对 client 进行识别。

4、 假设不使用 EDNS，当客户端使用 Google 8.8.8.8 等公共 DNS 解析服务器进行内容访问时，CDN 有可能会给用户分配的 CDN 服务器离客户端距离较远，原因是什么？EDNS 为什么能协助解决这个问题？Google 的 8.8.8.8 使用 Anycast 技术也能一定程度上缓解该问题，请问原理是什么？

(1) CDN 的加速资源是跟域名绑定的，公共 DNS 由于结点数有限，可能经常离用户很远。寻找 CDN 是由 DNS 服务器去查询的，因此寻找 CDN 的过程其实并不知道用户地址，如果是本地 DNS 则可以找到较近的 CDN 服务器，若使用公用 DNS 则可能不能获取到较近的 CDN 资源，影响用户感知体验。

(2) EDNS 根据用户请求来对同一域名作出相应的不同解析，允许 DNS 解析器传递用户的 IP 地址给权威 DNS 服务器以进行精确调度。

(3) Anycast 则允许客户端向一组目标服务器发送数据，而这组服务器中最终处理数据的服务器是由路由系统选择“最近”服务器。

5、 简述为了解决 CDN 存不下的问题可以采取什么方案。

(1) 部分缓存：如只存储视频开头的一部分

(2) 缓存替换：大部分内容（≈70%）只请求一次，请求大于 2 次的内容进入缓存

第 8 章 数据中心网络

1、 数据中心网络内部构成一个网络。请从网络管理、协议设计的角度，定性对比数据中心网络和互联网网络。

互联网网络（The Internet）	数据中心网络（DCN）
许多自治系统	一个管理域
分布式控制/路由	集中控制和路径选择
单个最短路径路由	多个从源地址到目的地址的路径
难以测量	容易测量，但数据量大
标准化传输（TCP 和 UDP）	多种传输（DCTCP, pFabric, …）
创新需要国际互联网工程任务组达成共识	单个公司可以创新
“网络的网络”	“大型超级计算机的基架”

2、 数据中心网络的目标之一是尽量让任意服务器对之间的吞吐仅受限于服务器网卡速率，请结合 Fat-Tree 以及 VL2，简述通过哪些手段可以帮助实现上述目标。

- 多路径设计：多个等价路径减少单一链路的拥塞。
- 负载均衡：如 ECMP 和随机路径选择，动态分散流量。
- 分层拓扑结构：确保网络资源可以水平拓展。
- 全双工链路：提高链路的利用效率。

3、 在数据中心网络中，多个发送端向一个接收端发送数据时，会带来 TCP Incast 问题。请简述 TCP Incast 问题发生的原因，以及可能的解决方案。

(1) TCP Incast 问题：当多条并发流到达同一交换机设备时，突发流量容易造成丢包。

(2) 解决思路：

- 增加设备缓冲区大小。
- 在中间设备增加通知功能，通过显式通知发送方哪些数据包丢失，尽快恢复丢包。

4、 交换机/路由器将待处理的数据包放到缓冲队列中，缓冲队列的大小对设备的转发性能有很大的影响，请简述队列过大或过小对传输流的性能影响。RED 机制可以缓解队列过大带来的性能问题，请简述该机制的运行过程。

(1) 队列大小的影响：

- 缓冲区太小：会导致丢包率过高，数据链路利用率低，TCP 传输效率低。
- 缓冲区太大：会导致网络拥塞时，数据包转发时延过大。

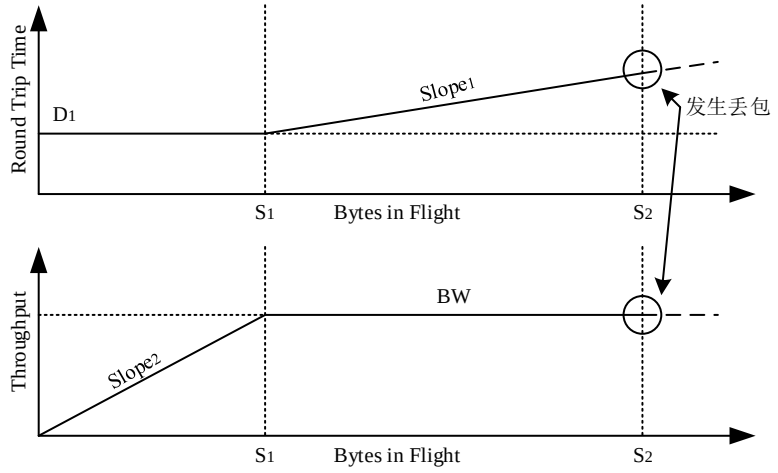
(2) RED（Random Early Detection）随机早期检测

在 RED 中，为缓存队列都设置了一个最小阈值和最大阈值，当收到数据包时：

- 若队列长度小于最小阈值，直接加入缓存队列；
- 若队列长度大于最大阈值，直接丢弃数据包；
- 若队列长度位于两者之间，开始按一定概率丢弃数据包。队列越长，丢弃概率越高。

5、 TCP 传输协议中，发送的字节数与吞吐（Throughput）以及 RTT 的关系如下图所示。假设，路径的传播时延为 RTT_{prop}，瓶颈链路带宽为 BW，瓶颈链路处的队列长度为 Q。请问 1) D_1，$Slope_1$，$Slope_2$，S_1，S_2 分别为多少？2) BBR 根据传播时延与瓶颈链路带

宽的乘积设置拥塞窗口，如何测得传播时延 RTT_{prop}，以及瓶颈链路带宽 BW？



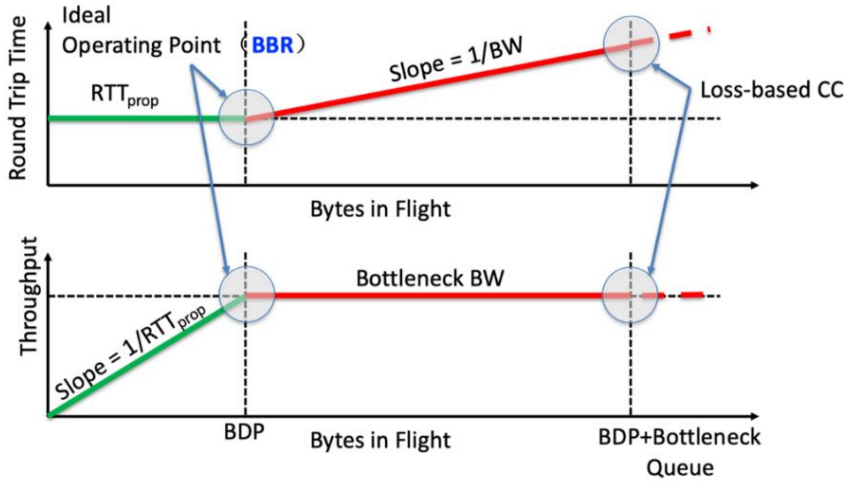
(1) $D_1 = RTT_{prop}$

$Slope_1 = 1 / BW$

$Slope_2 = 1 / RTT_{prop}$

$S_1 = BW * RTT_{prop}$

$S_2 = BW * RTT_{prop} + Q$



(2) 交替测量带宽和延迟，用一段时间内的带宽极大值和延迟极小值作为估计值，动态更新测量值，最终控制发送速率，避免网络拥塞。

6、 数据中心网络问题：1) ECMP（Equal Cost Multi Path，等价多路径路由）是如何实现流级别的负载均衡的？2) 负载均衡是数据中心网络中提升带宽利用效率的重要机制，请简述数据包级别的负载均衡和数据流（flow，五元组标识）级别的负载均衡的优劣势。3) 数据流级别的负载均衡和数据包级别的负载均衡相比，劣势是什么？4) 基于 ECN（Explicit Congestion Notification，显式拥塞通知）的拥塞控制在数据中心网络中广泛使用，简述其原理，以及为什么能缓解 TCP Incast 问题。

(1) 当路由器发现同一目的地址出现多个最优路径时，会更新路由表，为此目的地址添加多条规则，对应于多个下一跳，可同时利用这些路径转发数据，增加带宽。

(2) 数据包级别的负载均衡：

- 工作原理：每个数据包可以独立地根据网络负载情况分发到不同的路径，而无需考虑它们所属的流（Flow）。

- 优点：带宽利用率高、动态性强
- 缺点：数据包乱序问题、额外的开销、网络抖动增加

数据流级别的负载均衡：

- 工作原理：以数据流（五元组：源 IP、目的 IP、源端口、目的端口、协议）为单位进行负载均衡。同一数据流中的所有数据包都通过同一条路径转发，确保数据包顺序一致。
- 优点：保证数据包顺序、较低的处理开销、适合大流量传输
- 缺点：负载分布不均衡、静态性较强

(3) 数据流级别的负载均衡对于流大小相差不多的情况效果更好，而对于流大小差异较大，特别大的数据流无论选择哪条路径都容易造成拥塞。

(4) 原理：基于的丢包反馈、路径延时反馈、显示反馈等信息进行拥塞窗口调节。

原因：交换机作为集中控制器计算所有流的平均窗口值，通过显示反馈给发送端，发送端统一将发送窗口调整为平均窗口值大小来公平地分配带宽。

补充：具体来说，在 DCTCP 方法中，在交换机处有一个阈值 K，若数据包到达时，队列占用量大于 K，则标记该数据包。当接收端接收到带有标记的数据包后，返回带有 ECN（Explicit Congestion Notification，显式拥塞通知）标记的 ACK 给发送端，发送端以此调节拥塞窗口的大小。

7、 一个"客户-服务器"系统的性能受到两个网络因素的影响：网络带宽（每秒传输多少位）和延迟（第 1 位从客户传播到服务器花多少秒的时间）。(1) 带宽和延迟成反比关系吗？如果是，请阐述其关系；如果否，试给出一个具有高带宽高延迟的网络的例子，再给出一个具有低带宽低延迟的网络的例子。(2) 除了带宽和延迟，还需要什么其它的参数，才能很好地刻画一个用于视频传输网络所提供的服务质量？

(1) 带宽和延迟不成反比，高带宽高延迟：卫星链路；
低延迟带宽：56kbps 调制解调器、蓝牙。

(2) 启动时间、缓冲时间、卡顿率等。

第 9 章 未来网络体系架构

1、 简述 TCP/IP 体系结构对移动性支持不好的主要原因，以及可能的解决方案。

(1) 主要原因：

- IP 地址既表示地址，又标识主机。当移动后，IP 地址发生变化。
- 连接和 IP 地址绑定，当 IP 地址变化时，连接只能断开。

(2) 几种可能的方案：

- Mobile IP 技术。
- 连接和 IP 地址解绑定，当 IP 地址变化时，移动一方通告对方自己新的地址，两端的应用连接不断开。
- 使用 NDN（Named Data Networking，命名数据网络）等类似机制。

2、 TCP/IP 体系结构存在两个绑定：(1) 位置与身份的绑定；(2) 数据和位置的绑定。请回答两个问题：1) 解释这两种绑定的含义，以及它们带来的问题；2) NDN（Named Data Networking，命名数据网络）是如何解决这两个绑定的。

(1) 位置与身份的绑定

- 含义：IP 地址既表示设备的位置（在网络中的位置）又表示其身份（通信的终端）。
- 带来的问题：移动性问题、可拓展性问题、安全性问题。

数据和位置的绑定

- 含义：数据传输依赖于发送方和接收方的具体位置（IP 地址）。
- 带来的问题：内容分发效率低、可靠性问题、缓存利用不足。

(2) NDN 通过将通信基于数据名称而非网络位置，解除了这两种绑定，实现了高效的数据分发和灵活的移动性支持，提升了网络的可拓展性和可靠性。

3、 1) 未来互联网体系结构 NDN、SOFIA、MobilityFirst 等都强调名字（Name）与地址（Address）的分离，请问名字与地址分离解决什么问题？2) 与 TCP/IP 体系结构不同，NDN 是接收端驱动的，即接收端发送 Interest 数据包，发送端收到 Interest 数据包后才回复 Data，且 Data 沿着 Interest 数据包的反向路径发送给接收端。这样做的好处与劣势是什么？

(1) 名字与地址分离解决的问题：

- 支持节点和内容的移动性
- 提高内容分发效率
- 提升网络的可拓展性
- 增强多宿和路径选择能力
- 加强安全性和隐私保护

(2) 优势：

- 不存在身份与地址的绑定，支持移动性；
- 安全性和可拓展性增强；
- 减少冗余传输，节省带宽。

劣势：

- 名字不定长，在路由表中难查找；
- 网络设备的实现和维护复杂。

4、 NDN 等未来互联网体系结构试图改变 TCP/IP 协议的哪些方面，为什么这些新型的互联网体系结构部署比较困难？

(1) 改变：

- 可拓展性：在现有互联网体系结构中，只能通过不断地增加硬件设备进行拓展。
- 动态性：使网络中身份和位置分离，增加可移动性。
- 安全可控性：构建面向服务和数据的安全架构，从源头上限制网络攻击行为的发生。

(2) 部署困难原因：

网络规模太大，不易统一部署，且过去的网络设备在设计之初并没有考虑到这些问题，无法直接应用这些架构。

第 10 章 软件定义网络及网络功能虚拟化

1、 什么是 SDN？为什么需要 SDN？

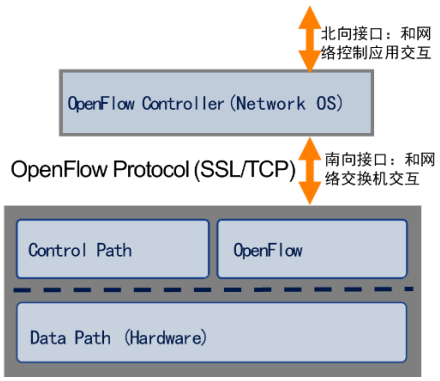
(1) SDN（Software Defined Network，软件定义网络）：数据面（data plane）与控制面（control plane）物理分离的网络，由（逻辑上）集中的控制面控制多个转发设备（如交换机）。

(2) 传统网络配置静态且难以改变、不灵活：
传统网络创新难、成本高；
传统网络配置和管理复杂。

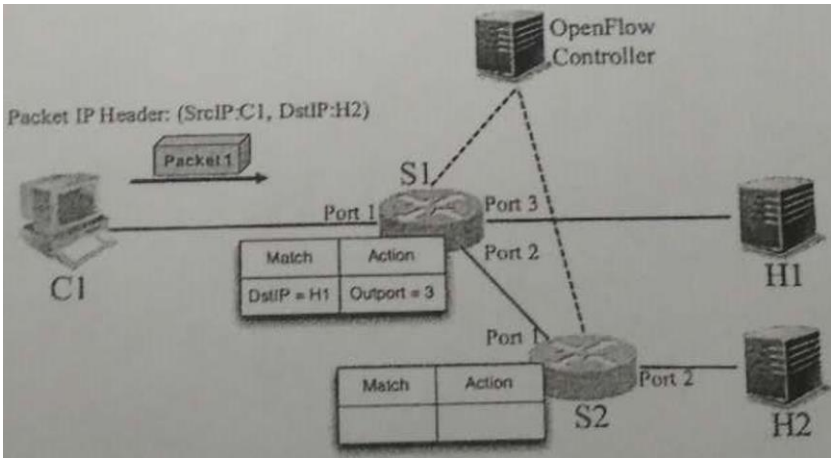
2、 在 SDN 网络体系结构中，SDN 控制器向数据平面的 SDN 交换机下发流表时所使用的接口是（B）。

A. 东向接口　　B. 南向接口　　C. 西向接口　　D. 北向接口

SDN 控制器有控制层面和数据层面上下两个层面，通过北向 API 控制网络控制应用层序，通过南向 API 控制分组交换机。综上所述，SDN 控制器向数据平面的 SDN 交换机下发流表所使用的接口是南向接口。



3、 下图是 OpenFlow 局域网络拓扑，S1 的流表包含一条转发规则，Controller 持有全局网络的拓扑信息。请描述 Packet1 从 C1 到 H2 的转发过程，包括流表查询、流表安装流程以及具体的流表转发规则。



[1] 首先，数据从 C1 唯一的端口转出，发到 S1。

[2] S1 查询流表，匹配不到对应条目，因此将该数据包缓存下，并查询 Controller。

[3] Controller 下发规则至 S1，规则内容为“DstIP = H2, Outport = 2”。

[4] S1 按照相应规则，将数据包从 Port 2 转出，至 S2。

[5] S2 收到数据包后，找不到对应规则，因此也将该数据包缓存下，并查询 Controller。

[6] Controller 下发规则至 S2，规则内容为“DstIP = H2, Outport = 2”。（注意，这一步可以与第 3 步合并，即 Controller 同时给 S1 和 S2 下发规则）

[7] S2 查询到相应规则后，将数据包从 Port 2 转出，至 H2。

4、 什么是 NFV？为什么需要 NFV？

(1) NFV（Network Functions Virtualization，网络功能虚拟化）：一种对于网络架构的概念，利用虚拟化技术，将网络节点阶层的功能分割成几个功能区块，分别以软件方式实现，不再局限于硬件架构。

(2) 网络包含大量中间盒子（middlebox），中间盒子维护成本高，易出错。NFV 可以减少投资成本、加快产品研发、提高灵活性和可扩展性。