

2020网原课堂习题

Chiu & Phosphor - 2021.1.8

Chap3-5 数据通信、物理层、数据链路层

一、选择填空

1、实际应用最广泛的计算机网络体系结构是（TCP/IP），它的网络层是（基于无连接）的，同层对等实体进行信息交换时必须遵守的规则称为（协议）。

2、奈奎斯特定理：无噪声有限带宽信道的最大数据传输率 = $2H \log_2 V$ ，香农定理：带宽为 H 赫兹，信噪比为 S/N 的任意信道的最大数据传输率 = $H \log_2 (1 + S/N)$ 。二值信号在 3 kHz 的信道上传输，信噪比为 10dB，最大数据速率为（6kbps）

由题可知带宽 $H=3\text{kHz}$ ，信噪比 S/N 由其对数形式 $10 \log_{10} \frac{S}{N} = 10\text{dB}$ 可得 $S/N=10$ ；信号包含的离散等级 V 的计算由信号的二进制位宽为 n ，则离散等级 $V=2^n$ （chap2 PCM）；此处二值信号用 1 位二进制数表示，故 $V=2$ 。

由 Nyquist 定理计算的最大数据传输率 = $2H \log_2 V = 6\text{kbps}$

由香农定理计算的最大数据传输率 = $H \log_2 (1 + S/N) = 3 \log_2 (1 + 10)\text{kHz} \approx 10.38\text{kbps}$

最终得到的最大数据速率取 min，为 6kbps

尼奎斯特证明，如果一个任意信号通过了一个带宽为 B 的低通滤波器，那么只要进行每秒 $2B$ 次（确切）采样，就可以完全重构出被过滤的信号。由于通过样值能恢复出来的高频成分已经被滤掉了，所以高于每秒 $2B$ 次的采样毫无意义。如果信号包含了 V 个离散等级，则尼奎斯特的定理为：

$$\text{最大数据速率} = 2B \log_2 V \text{（比特/秒）} \quad (2-2)$$

例如，无噪声的 3 kHz 信道不可能以超过 6000 bps 的速率传输二进制（即只有两级的）信号。

到现在为止，我们只考虑了无噪声信道。如果存在随机噪声，情况会急剧地恶化。并且，由于系统中分子的运动，随机（热）噪声总是存在的。热噪声的数量可以用信号功率与噪声功率的比值来度量，这样的比值称为信噪比（SNR, Signal-to-Noise Ratio）。如果我们将信号功率记作 S ，噪声功率记作 N ，则信噪比为 S/N 。通常情况下为了适用很大的范围，该比率表示成对数形式 $10 \log_{10} S/N$ ，对数的取值单位称为分贝（dB, decibel）。decibel 意味着 10，而选择 bel 则是为了向发明了电话的贝尔（Alexander Graham Bell）致敬。10 的信噪比为 10 分贝，100 的信噪比为 20 分贝，1000 的信噪比为 30 分贝，等等。立体声放

3、差分曼彻斯特码的原理是：位前跳变表示（数据），位中间跳变表示（时钟）（即用于做时钟同步）。10M 802.3 LAN 使用曼彻斯特编码，它的波特率是（20Mbaud）。

波特率=电平变化的次数，比特率=比特变化的次数，题中 10M 即 10Mbps

对于曼彻斯特编码，传输 1 位需要电平信号 2 次跳变，故波特率=比特率×2，注意单位是 baud

4、物理层的四个重要特性：机械特性、电气特性、功能特性和（规程特性）。多路复用技术提高了线路利用率，SONET/SDH 采用了（TDM）即（时分复用）技术。

二、判断

- 1、面向连接的服务是可靠服务 (×) (例如打电话)
- 2、无连接的服务不能保证PDU (协议数据单元) 按顺序到达目的地 (√) (因为无连接服务中各个数据单元是按不同路径走的)
- 3、PDU (协议数据单元) 是由ICI (接口控制信息) 和SDU (服务数据单元) 组成的 (×)
($PDU = PCI + SDU$, PCI是协议控制信息)

三、各类交换方式延迟计算 (ch3 P53-69)

对各类通信子网定义下列参数:

N = 两个给定站点之间所经过的段数;
L = 报文长度 (L为分组大小P的整数倍), 单位: 位;
B = 所有线路上的数据传输速率, 单位: 位/秒;
P = 分组大小 ($P \leq L$), 单位: 位;
H = 每个分组的分组头, 单位: 位;
S1 = 线路交换的呼叫建立时间, 单位: 秒;
S2 = 虚电路的呼叫建立时间, 单位: 秒;
D = 各段内的传播延迟, 单位: 秒。

1、写出电路交换、报文交换、虚电路分组交换 (虚电路分组头可忽略) 和数据报分组交换的端到端延迟时间的表达式。

发送的都是一封报文 $L=kP$, 报文可以使用电路 / 报文交换完整发送, 也可以分成k个分组使用虚电路 / 数据报分组交换, 注意分组交换都采用**存储转发**的方式, 报文交换也是 (那么只有电路交换不是了)。下面分类说明。

【电路交换】

有**建立电路**、**传输数据**、拆除电路三个阶段, 端到端延迟涉及前两个阶段。

延迟时间=线路交换的呼叫建立时间 + 段数×各段传播延迟 + 报文传输时间= $S1 + ND + L/B$

【报文交换】

无需建立交换线路, 信息以报文 (逻辑上完整的信息段) 为单位进行**存储转发**。由于是存储转发, 每段都要重新发一次报文。

延迟时间=段数×各段传播延迟 + 段数×报文传输时间=段数×(各段传播延迟+报文传输时间) = $N(D + L/B)$

【虚电路分组交换】

有**建立虚电路**、**传输数据**、拆除虚电路三个阶段, 端到端延迟涉及前两个阶段。

题中表示虚电路分组头可忽略, 因此一个分组的传输时间为 P/B , 一共 L/P 个分组。注意虚电路分组交换比起电路交换只是变成一次发一个包了 (而不是一封报文), 包的传输仍然是按照指定路线顺序到达的, 因此**最后一个包**正要发出时, 之前所有包的数据传输用时为 $(L-P)/B$, 它从第一段发给第二段一直到第N段发到目的站点耗时 $N(D+P/B)$, 总共的传输时间+传播延时为 $(L-P)/B + N(D+P/B) = L/B + ND + (N-1)P/B$

也可以理解成最后一个包被发出, 即从第一段到达第二段时, 所有包的数据传输用时为 L/B 。最后一个包经历 ND 的传播延迟以及剩下 $N-1$ 段的存储转发, 总共为 $L/B + ND + (N-1)P/B$

延迟时间=虚电路的呼叫建立时间 + 传输时间+传播延迟= $S2 + L/B + ND + (N-1)P/B$

【数据报分组交换】

每个分组可以走不同路径，由分组头（带有源、目的地址）确定，因此一个分组的传输时间为 $(P+H)/B$ ，一共 L/P 个分组。

此时包仍然是一个一个发出，最后一个包从第一段发至第二段时，所有包的数据传输用时为 $\frac{L}{P} \frac{P+H}{B}$ ，传播延迟为 D ；虽然各个包可以是按不同路径到达的，但似乎题目表示了都是经过 N 段，所以最后一个包从第二段到达目的站点耗时 $(N-1)D + (N-1)(P+H)/B$ ，总共的传输时间+传播延迟为 $\frac{L}{P} \frac{P+H}{B} + D + (N-1)(D + \frac{P+H}{B})$

$$\text{延迟时间} = \text{传输时间} + \text{传播延迟} = \frac{L}{P} \frac{P+H}{B} + ND + (N-1) \frac{P+H}{B}$$

2、求 $N = 4$, $L = 30720$, $B = 9600$, $P = 1024$, $H = 16$, $S1 = 0.2$, $S2 = 0.1$, $D = 0.001$ 时, 上述四种交换方式的总延迟时间。

自行计算。（其实是因为懒得敲计算器了）

四、滑动窗口 (ch5 P43-72)

退后N帧重传 (go back N)

退后N帧重传 (go back N) 的基本过程:

- 发送窗口——大小 > 1 , $\text{max window size} = \text{MaxSeq}$ 即最大序号
 - 发出一个帧, 发送窗口上界+1
 - 若发送窗口满, 则不再发送 (即上界-下界=发送窗口尺寸)
 - 收到一个确认帧ack且确认帧序号在发送窗口之内, 发送窗口下界一直加到确认帧序号+1 (**累计确认**)
 - 某序号为x的帧迟迟没有得到确认, 出现超时 (此时发送窗口下界一定是x), 置发送窗口的上界=发送窗口的下界, 从待确认帧开始重发 (**退后N帧重传**)
- 接收窗口——大小 = 1【!】
 - 收到一个帧且该帧是希望收到的 (即帧序号==接收窗口下界), 接收窗口上界+1且下界+1 (**整体移动**), 并发送一个确认帧ack, 其序号为新的接收窗口下界

1、【给出事件画出窗口变化】在一个差错信道上使用滑动窗口协议进行数据传输。发送序号和接收序号的取值范围是0~7。其中一方发送窗口和接收窗口的初始状态如下表所示。请根据该方所发生的事件顺序，写出发送窗口和接收窗口上、下界的变化过程。采用“退回到N”的重传技术，允许捎带确认，其中 lx_y 和 ly_x 分别表示发送和接收一个帧， x 是发送序号， y 是希望接收的下一个帧的序号。

[illegible]

分析题目可以知道， lxy 表示发出序号为 x 的帧（发送窗口上界+1），此时接收窗口下界为 y ； lxy 表示收到序号为 x 的帧（接收窗口下界增加到 $x+1$ ），确认帧序号为 y （发送窗口下界增加到 y ，因为注意题目中说明了“ y 是希望接收的下一个帧的序号”，那么确认帧序号为 y 表示确认到了 $y-1$ ，则发送窗口下界增加到 $(y-1)+1=y$ ）。

【答案】

收的下一个帧的序号。

其实不用管

另一端仍在等待2

意思是这个帧超时退后n帧重传

上界：下一个要发的帧（还没发）
下界：等待确认的已发帧

某一方发生的事件	初始	100	110	120	101	112	132	142	122	132	142	120	125	153
发送窗口上界位置	0	1	2	3	3	3	4	5	5	5	5	5	3	3
发送窗口下界位置	0	0	0	0	1	2	2	2	2	2	2	2	2	3
接收窗口下界位置	0	0	0	0	1	2	2	2	3	4	5	5	5	6

2、【倒过来，由窗口变化反推事件】在一个差错信道上使用滑动窗口协议进行数据传输。发送序号和接收序号的取值范围是0 ~ 7。其中A方发送窗口和接收窗口的初始状态如下表所示。请根据A方发送窗口和接收窗口的变化过程，给出导致窗口变化的事件顺序。采用“退回到N”的重传技术，允许捎带确认，用 Axy 和 Bxy 分别表示A发送和接收一个帧， $Axy_timeout$ 表示 Axy 超时， x 是发送序号， y 是希望接收的下一个帧的序号。发送窗口上界为 $next_frame_to_send$ ，下界为 $ack_expected$ ；接收窗口下界为 $frame_expected$ 。

A方发生的事件	初始	A00														
发送窗口上界位置	0	1	2	3	3	3	4	5	5	5	5	2	3	3	4	5
发送窗口下界位置	0	0	0	0	1	2	2	2	2	2	2	2	2	3	3	3
接收窗口下界位置	0	0	0	0	1	2	2	2	3	4	5	5	5	6	6	6

【参考解答】（自己写的，没有找到标准答案，欢迎讨论）

A方发生的事件	初始	A00	A10	A20	B01	B12	A32	A42	B22	B32	B42	A20 timeout	A25	B53	A36	A46
发送窗口上界位置	0	1	2	3	3	3	4	5	5	5	5	2	3	3	4	5
发送窗口下界位置	0	0	0	0	1	2	2	2	2	2	2	2	2	3	3	3
接收窗口下界位置	0	0	0	0	1	2	2	2	3	4	5	5	5	6	6	6

A收到B确认，发送窗口下界+

A收到B信息帧

A收到B信息帧

选择重传

选择重传 (selective repeat) 的基本过程：

- 发送窗口——大小 > 1 ，最大为 MaxSeq ，即最大序号
 - 发出一个帧，发送窗口上界+1
 - 若发送窗口满，则不再发送（即上界-下界=发送窗口尺寸）
 - 收到一个确认帧ack且确认帧序号在发送窗口之内，发送窗口下界一直加到确认帧序号+1（**累计确认**）
 - 收到一个否定确认帧nak，则重发nak指定要求发送的那个帧，窗口尺寸不变
 - 某序号为x的帧迟迟没有得到确认，出现超时（此时发送窗口下界一定是x），重发一次待确认的最老帧（**选择重传**）
- 接收窗口——大小 > 1 ，整体要求发送+接收窗口尺寸 $\leq \text{MaxSeq}+1$ （注意课件上写的例子有错）
 - 收到一个帧，若不等于接收窗口下界则发送nak，若落在接收窗口之内则标记为收到 & 检测接收窗口是否增长。
 - 【注意】这两个判断是依次进行的，不是二选一的关系，因此会产生三种情况：
 - ①新帧序号=接收窗口下界，则标记该帧已收到，检查从下界开始的所有标记已收到的连续帧，设最大序号-最小序号的差为n，那么接收窗口上界+n+1，下界+n+1（**下界飞跃**），之后ACK超时的时候强制发送的确认帧ack序号是新的下界
 - ②新帧序号 \neq 下界但在接收窗口内（此时从下界开始没有连续帧，接收窗口无变化），则标记已收到，发送nak帧，要求对方发帧序号=接收窗口下界的帧来（下次很可能发生下界飞跃）
 - ③新帧序号在接收窗口外，丢弃并发送nak帧

3、在一个差错信道上A、B双方使用滑动窗口协议进行数据传输，双方采用“选择性”的重传技术，允许捎带确认，发送序号和接收序号的取值范围是0 ~ 7。请根据表1和表2中A方窗口初始状态和随后发生的事件，给出每个事件发生后发送窗口和接收窗口的变化过程，并给出表1中在S2_timeout，RNAK5事件之后A方按照协议接着发生的事件，表2中R41，ACKtimeout事件之后A方按照协议接着发生的事件。（请在表中对应事件下的空格里标出）

Sxy和Rxy分别表示A发送和接收一个数据帧，x是帧中携带的数据的序号，y是希望下一帧接收的数据的序号；Sz_timeout表示A方发送窗口内序号为z的数据超时，ACKtimeout表示ACK时钟超时；SACKz和RACKz分别表示A方发送和接收一个ACK确认帧，z是希望下一帧接收的数据的序号；SNAKz和RNAKz分别表示A方发送和接收一个NAK确认帧，z是希望下一帧接收的数据的序号。发送窗口上界为下一个将要发送的数据的序号（next_frame_to_send），下界为最早发送出去的但还没有被确认的数据的序号（ack_expected）；接收窗口下界为按照顺序希望最早接收到的数据的序号（frame_expected）。

A发生的事件		初始	S ₂₀	R ₀₁	R ₁₂	S ₃₂	S _{2_timeout}	R _{ACK4}	S ₄₂	S ₅₂	S ₆₂	R ₂₅	R _{NAK5}	R _{ACK7}
A方发送窗口	下界	0												
	上界	2												

分析题目可知：

- S_{xy} 使得发送窗口上界增长到 $x+1$ ，同时表明接收窗口下界= y
- R_{xy} 可能引起接收窗口变化，需要具体分析；同时如果 $x \neq$ 接收窗口下界，还会导致产生一个发送否定确认帧 S_{NAKz} 的事件，其序号 z =接收窗口下界（希望接收的下一帧）
- R_{ACKz} 一般都会引起发送窗口变化（除非 $z <$ 发送窗口下界），表明确认帧序号为 z （对方希望接收的下一帧是 z ），则发送窗口下界增长到 z （**累计确认**，实际上是 $(z-1)+1=z$ ）
- R_{NAKz} 会导致产生一个发送帧 S_{zy} 的事件，但窗口不变
- $ACK_timeout$ 会导致产生一个**发送确认帧 S_{ACKz}** 的事件，其序号 z =接收窗口下界（希望接收的下一帧）
- $Sz_timeout$ 会导致产生一个重发帧 z 的事件 S_{zy} ，其中 y =接收窗口下界（**选择重传**）

【答案】（注意这个题可以多画一行记录A接收窗口的下界，以便正确写出 S_{xy} 的 y ；它会被 R_{xy} 的 x 改变）

NAK后面的数是想收到的（要么它收到了错的5要么收到了5后面的，但4及之前都收到了）

表示下一个需要4，所以3及之前都收到啦

A发生的事件		初始	S_{20}	R_{01}	R_{12}	S_{32}	$S_{2_timeout}$	R_{ACK4}	S_{42}	S_{52}	S_{62}	R_{25}	R_{NAK5}	R_{ACK7}
A方发送窗口	下界	0	0	1	2	2	2	4	4	4	4	5	5	7
	上界	2	3	3	3	4	4	4	5	6	7	7	7	7

再发一遍2

那边收到6了！

4、题面同上，填一个新表

A发生的事件		初始	R_{20}	S_{03}	R_{41}	R_{31}	$ACK_{timeout}$	R_{51}
A方接收窗口	下界	2						
	上界	6						

【答案】（格外注意**接收窗口**变化是**上界和下界同时增加的**，而且是在**0~MaxSeq**之间循环增加，因此 $(7+2)\%8=1$ 。此外建议记录一个收到帧的序列，保证正确发现了一个 R_{xy} 导致接收窗口飞跃的情况）

A发生的事件	初始	S 是发给对方的 R_{41}			过久没发 S, 直接单独发一个确认帧 $\leftarrow \text{ACK}_{\text{timeout}}$		R_{51}
		R_{20}	S_{03}	S_{NAK3}	R_{31}	S_{ACK5}	
A方接收窗口	下界	2	3	3	5	5	6
	上界	6	7	7	1	1	2

只需要管对方发过来的是什
么, 也就是 R 的第一位 \leftarrow

因为之前已经收到 4 了, \leftarrow
所以想要的是 5 \leftarrow

接收窗口大小一直不变
上界跟着下界变

Chap6-9 局域网与介质访问子层、路由和网络层、端到端与传输层、网络应用

一、选择填空

1、局域网的IEEE 802系列标准中, 802.3定义了采用(CSMA/CD)技术的局域网标准(注: 而且是1-坚持型), 它所采用的介质访问控制技术确定冲突要花费(2)倍电缆长度的传输时间。

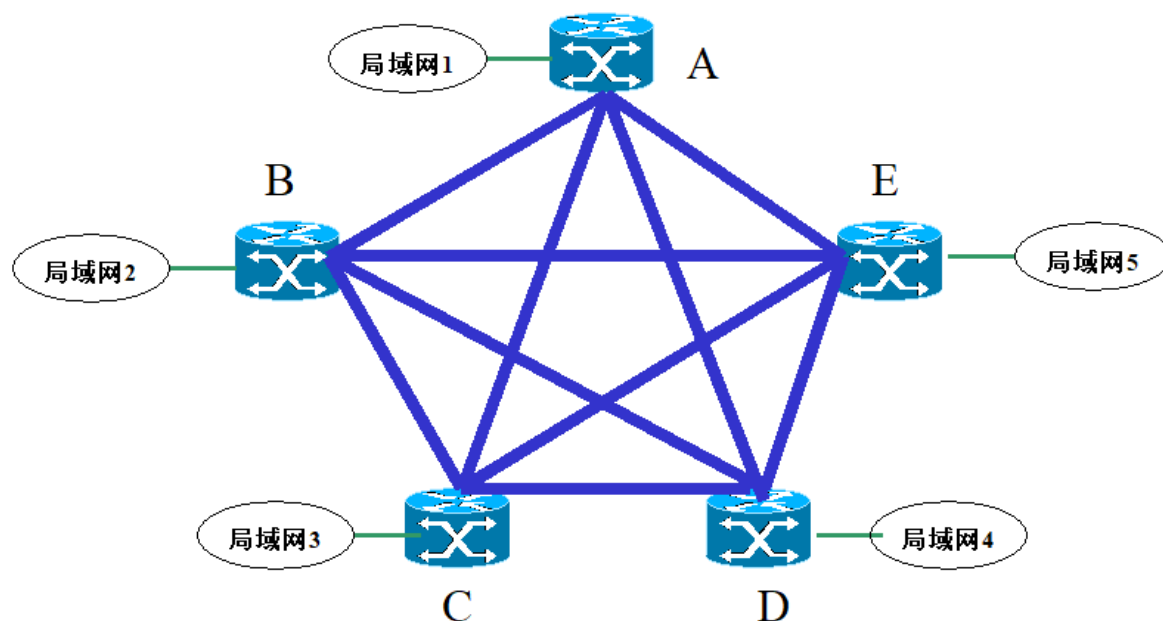
2、一个IPv4地址块的子网掩码为255.255.240.0, 则该子网可分配的最大IP地址数为(4094);

240(dec)=11110000(binary), 所以有4+8=12位二进制数为0, $2^{12}=4096$, **主机部分为全0和全1的不参与分配**, 所以是 $4096-2=4094$; 或者可以看 $256-240=16$ (dec), 因此第3个字节(8位)中后4位二进制数为0, 第4个字节(8位)全0, 共12位可以自由分配, 则为 $2^{12}-2=4094$

3、(OSPF)采用链路状态算法。

二、地址分配 (ch7 P83-92)

如图所示, 5台路由器组成全相连的网络, 每台路由器有5个接口, 分别连接其它4台路由器和1个局域网, 每个局域网最多连接20台计算机, 每台计算机分配1个IP地址。如果只有一个IPv4地址块202.112.10.0/24可供分配, 请给出一种合理的地址分配方案, 分别给出每个局域网的地址空间和路由器每个端口的地址以及它们的掩码。



路由器A：自身连接到局域网1里的端口×1+局域网1的计算机×20，一共需要21个IP地址，加上全0和全1一共23个。由于要求这些IP地址组成一个子网，与其他子网不相交，因此求对应的最小的2的幂次，即 $2^{\lceil \log_2 23 \rceil} = 2^5 = 32$ ，需要为这个子网分配32个IP地址。可以取202.112.10.0 ~ 202.112.10.31，子网掩码为/(32-5)=/27

路由器之间：注意连接路由器A和B的两个端口的链路也是一个子网，需要两个端口的2个IP地址，加上全0和全1一共4个，对应的最小的2的幂次（相当于补齐到2的整数次幂）为4。取202.112.10.32 ~ 202.112.10.34，子网掩码为/30

【答案】

具体的分配方案和分配顺序有关，因此答案不唯一。一个参考解答是先分配路由器之间 $C_5^2 = 10$ 条链路，对应0~4, 4~8, ..., 36~40这10段，掩码均为30位；再分配5个路由器端口所连局域网，注意由于掩码为27位，要保证不与之前的相交，要把0~40也补齐到 2^5 的整数倍，所以从64开始。

该参考分配方案如下：

5台路由器之间有10条链路，每个链路连接的两个端口构成一个子网，需要4个IP地址。

202.112.10. (0, 4, 8, 12, 16, 20, 24, 28, 32, 36) 255.255.255.252，注意252=256-4

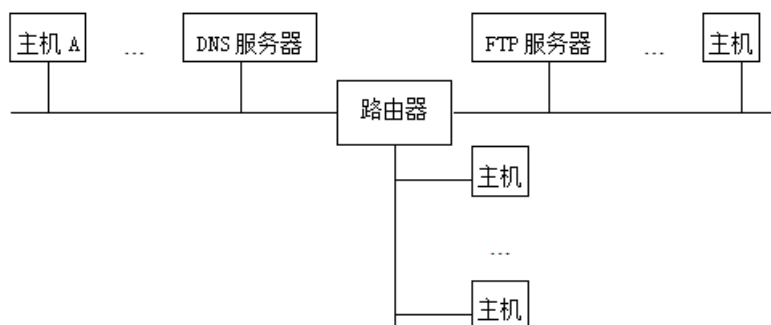
每个局域网分配32个地址——202.112.10. (64, 96, 128, 160, 192) , 255.255.255.224，注意224=256-32

三、综合大题

(这是一整道题，一般的考题不会出这么长，作为例题是为了说明其中的各个部分都代表了一类题型)

【地址分配】

一个网络的拓扑结构如下图所示，一台路由器连接三个IEEE 802.3 LAN，每个LAN连接60台计算机。现有一个IP地址块202.112.37.0/24，请合理化分子网，给出网络掩码和主机A、DNS服务器、FTP服务器的IP地址。



【参考解答】（自己写的，欢迎讨论）

注意到只有一台路由器，那么就没有不同路由器的端口之间连接而形成的子网，只有路由器某端口及其所连LAN形成的子网。

一个子网：端口×1+计算机×60=61个IP地址，加上全0和全1有63个IP地址，则需分配 $2^6 = 64$ 个IP地址，子网掩码为/(32-6)=/26，即32位中有26个1。按DNS、FTP、下面的LAN的顺序进行分配，得到：

202.112.37. (0, 64, 128) 255.255.255.192，注意256-64=192

则网络掩码为255.255.255.192，去除全0并将路由器端口设置为第一个IP地址，得到主机A的IP地址是202.112.37.3，DNA服务器是202.112.37.2，FTP服务器是202.112.37.66

【各层贯通】 (ch9 P16-26, P66-69, ch7 P94-96)

如果主机A知道DNS服务器和路由器的IP地址、路由器的MAC地址、和FTP服务器的域名，而不知道DNS服务器的MAC地址和FTP服务器的IP地址，路由器知道所有相关信息，请叙述主机A通过FTP操作从FTP服务器下载文件的工作流程和各层地址变换，包括应用层、传输层、网络层和数据链路层的操作。

注：以下所有带箭头的如 主机A→DNS服务器 表示对应层中对应源（如网络层则为源IP）是A，目的是DNS服务器

从FTP服务器下载文件，只知道FTP服务器的域名，要获得服务器的IP地址，故需要访问DNS服务器进行DNS域名解析：（已知DNS服务器的IP地址）

- 传输层：UDP，主机A → DNS服务器
- 网络层：IP，主机A → DNS服务器
- 数据链路层：
 - 不知道DNS服务器的MAC地址，故需发ARP请求，获得其MAC地址
 - 获得之后，MAC，主机A → DNS服务器
- DNS服务器进行域名解析，返回资源记录(UDP, IP, MAC)（【注意】是FTP服务器的资源记录）

解析完毕后，访问FTP服务器下载文件：

- 传输层：TCP，主机A → FTP服务器（端口号21），三次握手建立连接
- 网络层：IP，主机A → FTP服务器，经过路由器转发
- 数据链路层：MAC，主机A → 路由器，路由器 → FTP服务器（【注意】同一局域网的才能直接由MAC地址找到，不同局域网的必须发给路由器，让路由器转发从而在目的子网中找到目的MAC地址）
- FTP服务器收到主机A发来的连接请求后，给出确认，建立连接后传输数据（端口号20），最后拆除连接

【TCP拥塞控制】 (ch8 P44-52)

慢启动算法的基本过程：

1. 拥塞窗口大小初始化为TCP连接的最大发送段长MSS
2. 发出一个最大段长的TCP段，若得到确认，拥塞窗口大小+=一个最大段长
3. 每次发出 $\frac{\text{拥塞窗口大小}}{\text{最大段长}}$ 个最大段长的TCP段，凡有一个最大段长得到确认，拥塞窗口大小+=一个最大段长。【注意】存在累计确认的情况，一个ack帧可能确认了n个最大段长，那么拥塞窗口大小 += n个最大段长
4. 重复第3步，直到发生超时事件 或者 拥塞窗口大小>阈值

慢启动算法之后，进入拥塞避免算法，其基本过程为：（还会发包？）

- 若上一步没有超时，而是因拥塞窗口大小>阈值而退出慢启动，则拥塞窗口开始线性增长，直到发生超时
 - 凡有 $\frac{\text{拥塞窗口大小}}{\text{最大段长}}$ 个最大段长得到确认，拥塞窗口大小+=一个最大段长
- 若发生超时，则将阈值设置为当前拥塞窗口大小的 $\frac{1}{2}$ ，拥塞窗口重新设置为一个最大段长，重新进入慢启动算法

```
// slow start
initialize: Congwin = 1
for (each segment ACKed)
    Congwin++
until (loss event OR Congwin >= threshold)
```

```
// slow start is over (if caused by Congwin > threshold)
// congestion avoidance
until (loss event) {
    for (every w segments ACKed)
        Congwin++
}
threshold = Congwin/2
Congwin = 1
goto slowstart
```

主机C通过FTP向服务器S传送文件，双方建立TCP连接，采用慢启动算法进行拥塞控制。初始阈值threshold为5000字节，主机C发送的TCP段的数据长度固定为1000字节，初始拥塞窗口大小为1000字节。使用网络监听工具对主机C的FTP传输过程中的收发包进行监听。下表给出了该FTP数据通道的建立连接和数据传输的部分过程，请将相关内容填入空白处，并给出收发包事件发生后主机C的TCP实体的拥塞窗口congwin和阈值threshold的大小。“---”表示该空格不需要填写，确认序号等于希望接收的下一个字节的序号。

收发包序号	源IP地址	目的IP地址	源端口号	目的端口号	序号SEQ	确认序号ACK	控制位	备注	主机C的congwin	主机C的threshold
1	C	S	1077	20	1000	---	Syn=1	连接请求	---	---
2	S	C	20	1077	3000	1001	Syn=1 Ack=1	---	---	---
3	C	S	1077	20	1001	3001	Ack=1	---	---	---
4	C	S	1077	20	1001	3001	---	数据传输	1000	5000
5	S	C	20	1077	3001	2001	Ack=1	---		
6	C	S	1077	20	2001	3001	---	---		
7	C	S	1077	20	3001	3001	---	---		
8	S	C	20	1077	3001	3001	Ack=1	---		
9	C	S	1077	20	4001	3001	---	---		
10	C	S	1077	20	5001	3001	---	---		
11	S	C	20	1077	3001	5001	Ack=1	---		
12	C	S	1077	20	6001	3001	---	---		
13	C	S	1077	20	7001	3001	---	---		
14	C	S	1077	20	8001	3001	---	---		
15	C	S	1077	20	9001	3001	---	---		
16	S	C	20	1077	3001	10001	Ack=1	---		
17	C	S	1077	20	10001	3001	---	---		
18	C	S	1077	20	11001	3001	---	---		
19	C	S	1077	20	10001	3001	---	序号为10001的包超时重传		

由题可知，一个最大段长MSS=1000字节，序号4握手结束，同时开始数据传输。

序号5收到1个MSS的确认（1001~2000），因此congwin+=1000变为2000，threshold保持不变；

序号6、7继续发送，保持不变；

序号8收到1个MSS的确认（2001~3000），因此congwin+=1000变为3000，threshold保持不变；

序号9、10继续发送，保持不变；

序号11收到2个MSS的确认（3001~5000），因此congwin+=2000变为5000，threshold保持不变；

=====此时congwin >= threshold，进入拥塞避免=====

序号12-15继续发送，保持不变；

序号16收到5个MSS的确认（5001~10000），正好是congwin/MSS，因此congwin+=1000变为6000，threshold保持不变；

序号17-18继续发送，保持不变；

序号19发生超时事件，因此 $\text{threshold} = \text{congwin}/2 = 3000$ ， $\text{congwin} = 1000$

=====进入慢启动=====

【答案】

收发包序号	源IP地址	目的IP地址	源端口号	目的端口号	序号SEQ	确认序号ACK	控制位	备注	主机C的congwin	主机C的threshold
1	C	S	1077	20	1000	---	Syn=1	连接请求	---	---
2	S	C	20	1077	3000	1001	Syn=1 Ack=1	---	---	---
3	C	S	1077	20	1001	3001	Ack=1	---	---	---
4	C	S	1077	20	1001	3001	---	数据传输	1000	5000
5	S	C	20	1077	3001	2001	Ack=1	---	2000	5000
6	C	S	1077	20	2001	3001	---	---	2000	5000
7	C	S	1077	20	3001	3001	---	---	2000	5000
8	S	C	20	1077	3001	3001	Ack=1	---	3000	5000
9	C	S	1077	20	4001	3001	---	---	3000	5000
10	C	S	1077	20	5001	3001	---	---	3000	5000
11	S	C	20	1077	3001	5001	Ack=1	---	5000	5000
12	C	S	1077	20	6001	3001	---	---	5000	5000
13	C	S	1077	20	7001	3001	---	---	5000	5000
14	C	S	1077	20	8001	3001	---	---	5000	5000
15	C	S	1077	20	9001	3001	---	---	5000	5000
16	S	C	20	1077	3001	10001	Ack=1	---	6000	5000
17	C	S	1077	20	10001	3001	---	---	6000	5000
18	C	S	1077	20	11001	3001	---	---	6000	5000
19	C	S	1077	20	10001	3001	---	序号为10001的包超时重传	1000	3000

在TCP协议中，使用慢启动算法和拥塞避免算法进行拥塞控制。假设网络负载非常稳定，在一个TCP连接中，每当拥塞窗口大小达到W时就会丢包， $W = 2^n$ （n是正整数）。假设每个TCP段的长度都是最大发送段长MSS，段的发送和接收时间可以忽略，RTT（Round Trip Time）为常量，并且有足够多的数据要发送。求平均传输速率。

【解答】

从这个例子中感觉congwin是以MSS为单位的，因此认为初始值为1，丢包时值为W，实际大小都是 $1 \times \text{MSS}$ 或 $W \times \text{MSS}$

以及这道题看起来是认为到达阈值，从慢启动进入拥塞避免增长算法之后，发的第一次是阈值 $+1 \times \text{MSS}$ ，但是课件没有明确是怎么发。对于题目最终计算结果影响不大，只是一个比较容易使人疑惑的点。

第一次执行算法后， $\text{threshold} = W/2$ 并在之后保持稳定，由于有足够多的数据要发送，可以忽略第一次执行时间，以第二次及之后某次为例进行分析。

慢启动期间，congwin初始为MSS，结束为 $\text{threshold} = W/2$ ，每次翻倍增长。第*i*次发送（i从0开始），拥塞窗口大小 $\text{congwin}_i = 2^i$ ，因此当 $\text{congwin}_i = W/2$ 时， $i = \log_2(W/2) = n - 1$ ，发送次数=*n*，则总共发送的段数（单位为MSS）的计算为

$$\sum_{i=0}^{\log_2(W/2)} 2^i = 1 + 2 + \cdots + (W/2) = 1 + 2 + \cdots + 2^{n-1} = 2^n - 1$$

即 $2^n - 1$ 个MSS。发送了*n*次，故用时*n*个RTT。

拥塞避免期间, congwin 初始为 $\text{threshold} = W/2 = 2^{n-1}$, 结束为 $W = 2^n$, 每次线性增长, 需要发送 $\frac{\text{拥塞窗口大小}}{\text{最大段长}}$ 个MSS并得到确认才会增加1个MSS。第 i 次发送, 拥塞窗口大小 $\text{congwin}_i = \text{threshold} + i$ (i 从1开始), 因此当 $\text{congwin}_i = W$ 时, $i = W - \text{threshold} = 2^{n-1}$, 发送次数 $= 2^{n-1}$, 则总共发送的段数 (单位为MSS) 的计算为

$$\sum_{i=1}^{m=2^{n-1}} (W/2 + i) = m(W/2) + m(m+1)/2 = \frac{m}{2}(W + m + 1) = 2^{n-2}(2^n + 2^{n-1} + 1)$$

即 $2^{2n-2} + 2^{2n-3} + 2^{n-2}$ 个MSS, 用时 2^{n-1} 个RTT。

从而计算出平均传输速率 = 总MSS / 总RTT, 为

$$\begin{aligned} \text{平均传输速率} &= \frac{2^n - 1 + 2^{2n-2} + 2^{2n-3} + 2^{n-2}}{n + 2^{n-1}} \cdot \frac{\text{MSS}}{\text{RTT}} \\ &= \frac{1.25W + 0.375W^2 - 1}{\log_2 W + 0.5W} \cdot \frac{\text{MSS}}{\text{RTT}} \\ &= \frac{10W + 3W^2 - 8}{8 \log_2 W + 4W} \cdot \frac{\text{MSS}}{\text{RTT}} \\ &\approx 0.75W \cdot \frac{\text{MSS}}{\text{RTT}} \end{aligned}$$

更新日志

2021.1.8 v1修正Chap3-5 第一题选择填空的第2小题“奈奎斯特计算”, 灰色解答部分对于离散等级V的说明出现了错误; v2修正一些打字问题; v3修正了Chap3-5 第四题滑动窗口的选择重传部分, 对于发送窗口、接收窗口尺寸的限制 (老师订正了讲义)。

2021.1.2 修正Chap3-5 第一题选择填空的第2小题“奈奎斯特计算”, 灰色解答部分对于信噪比的计算出现了错误。

2020.12.30 基本完成所有题目的解答。