

《计算机网络》作业

岳锦鹏

2023年10月6日——2023年12月31日

目录

第一章 计算机网络概述	5
第二章 应用层	9
第三章 传输层	13
第四章 网络层	23
第五章 数据链路层	29

第一章 计算机网络概述

P1. Consider an application that transmits data at a steady rate (for example, the sender generates an N -bit unit of data every k time units, where k is small and fixed). Also, when such an application starts, it will continue running for a relatively long period of time. Answer the following questions, briefly justifying your answer:

- a. Would a packet-switched network or a circuit-switched network be more appropriate for this application? Why?

电路交换网络更合适，一方面，该应用以稳定的速率传输数据，因此只需要分配固定的频宽或时隙等即可方便实现电路交换的多路复用，而不需要分组交换；另一方面，该应用开始后会持续运行相当长的一段时间，因此不需要分组交换来应对突发情况，此时使用电路交换，不需要分片，也不需要处理，能达到更高的传输效率。

- b. Suppose that a packet-switched network is used and the only traffic in this network comes from such applications as described above. Furthermore, assume that the sum of the application data rates is less than the capacities of each and every link. Is some form of congestion control needed? Why?

不需要拥塞控制，注意到“such applications”是复数，说明可能存在多个这样的应用，但由于应用传输速率小于每条连接的传输能力，而拥塞控制是用于在应用传输速率大于连接的传输能力时降低应用传输速率的，所以这时不需要拥塞控制。

P2. This elementary problem begins to explore propagation delay and transmission delay, two central concepts in data networking. Consider two hosts, A and B, connected by a single link of rate R bps. Suppose that the two hosts are separated by m meters, and suppose the propagation speed along the link is s meters/sec. Host A is to send a packet of size L bits to Host B.

- a. Express the propagation delay, d_{prop} , in terms of m and s .

$$d_{prop} = \frac{m}{s} \text{ second}$$

- b. Determine the transmission time of the packet, d_{trans} , in terms of L and R .

$$d_{trans} = \frac{L}{R} \text{ second}$$

- c. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.

$$d_{end_to_end} = d_{prop} + d_{trans} = \left(\frac{m}{s} + \frac{L}{R} \right) \text{ second}$$

d. Suppose Host A begins to transmit the packet at time $t = 0$. At time $t = d_{trans}$, where is the last bit of the packet?

最后一个 bit 刚刚被 Host A 发出, 但 Host B 还未收到。

e. Suppose d_{prop} is greater than d_{trans} . At time $t = d_{trans}$, where is the first bit of the packet?

第一个 bit 仍然在传播过程中, 未被 Host B 收到。

f. Suppose d_{prop} is less than d_{trans} . At time $t = d_{trans}$, where is the first bit of the packet?

第一个 bit 已经被 Host B 收到了。

g. Suppose $s = 2.5 \cdot 10^8$, $L = 100$ bits, and $R = 28$ kbps. Find the distance m so that d_{prop} equals d_{trans} .

解 :

$$\begin{aligned} d_{prop} &= d_{trans} \\ \frac{m}{s} &= \frac{L}{R} \\ \frac{m}{2.5 \cdot 10^8} &= \frac{100}{28 \cdot 10^3} \\ m &= 2.5 \cdot 10^8 \times \frac{100}{28 \cdot 10^3} = \frac{6250000}{7} \approx 892857.142857143 \quad (\text{meters}) \end{aligned}$$

□

P3. Consider the queuing delay in a router buffer (preceding an outbound link). Suppose all packets are L bits, the transmission rate is R bps, and that N packets simultaneously arrive at the buffer every LN/R seconds. Find the average queuing delay of a packet. (Hint: The queuing delay for the first packet is zero; for the second packet L/R ; for the third packet $2L/R$. The N th packet has already been transmitted when the second batch of packets arrives.)

解 :

第 i 个包的排队延迟为 $(i-1)\frac{L}{R}$, 每一批共 N 个包, 且各批之间互不影响

\therefore 一个包的平均排队延迟为:

$$\frac{1}{N} \sum_{i=1}^N (i-1) \frac{L}{R} = \frac{L}{NR} \sum_{i=1}^N (i-1) = \frac{L}{NR} \frac{N(N-1)}{2} = \frac{L(N-1)}{2R}$$

□

P4. Consider a packet of length L which begins at end system A and travels over one link to a packet switch, and travels from the packet switch over a second link to a destination end system. These two links are connected by one packet switch. Let d_i , s_i , and R_i denote the length, propagation speed, and the transmission rate of link i , for $i = 1, 2$. The packet switch delays each packet by d_{proc} . Assuming no queuing delays, in terms of d_i , s_i , and R_i , ($i = 1, 2$), and L , what is the total end-to-end delay for the packet?

$$d_{end_to_end} = d_{proc} + d_{prop} + d_{trans} = d_{proc} + \sum_{i=1}^2 \frac{d_i}{s_i} + \sum_{i=1}^2 \frac{L}{R_i}$$

Suppose now the packet is 1,000 bytes, the propagation speed on both links is $2.5 \cdot 10^8$ m/s, the transmission rates of all both links are 1 Mbps, the packet length is 1,000 bytes, the packet switch processing delay is 1 msec, the length of the first link is 4,000 km, and the length of the last link is 1,000 km. For these values, what is the end-to-end delay?

$$\begin{aligned}
 d_{end_to_end} &= d_{proc} + d_{prop} + d_{trans} = d_{proc} + \sum_{i=1}^2 \frac{d_i}{s_i} + \sum_{i=1}^2 \frac{L}{R_i} \\
 &= 1 \cdot 10^{-3} \text{s} + \frac{4000 \cdot 10^3 \text{m}}{2.5 \cdot 10^8 \text{m/s}} + \frac{1000 \cdot 10^3 \text{m}}{2.5 \cdot 10^8 \text{m/s}} + 2 \times \frac{1000 \times 8 \text{ bits}}{1 \cdot 10^6 \text{bits/s}} \\
 &= (1 \cdot 10^{-3} + \frac{4000 \cdot 10^3}{2.5 \cdot 10^8} + \frac{1000 \cdot 10^3}{2.5 \cdot 10^8} + 2 \times \frac{1000 \times 8}{1 \cdot 10^6}) \times 10^3 \text{ ms} \\
 &= 37 \text{ ms}
 \end{aligned}$$

- P5. Consider the throughput example corresponding to Figure 1.16(b). Now suppose that there are M client-server pairs rather than 10. Denote R_s , R_c , and R for the rates of the server links, client links, and network link. Assume all other links have abundant capacity and that there is no other traffic in the network besides the traffic generated by the M client-server pairs. Derive a general expression for throughput in terms of R_s , R_c , R , and M .

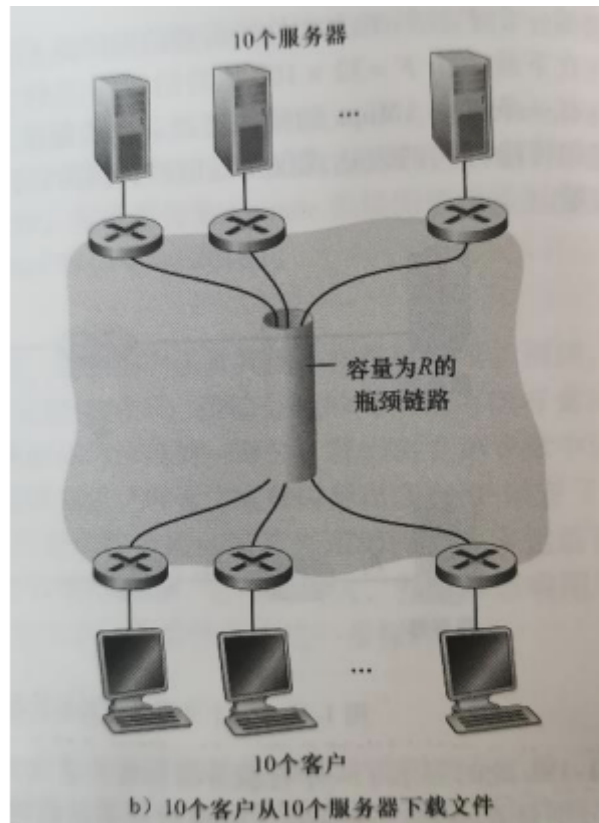


图 1.1: Figure 1.16(b)

吞吐量为:

$$\min \left\{ R_s, R_c, \frac{R}{M} \right\}$$

- P6. Consider sending a large file of F bits from Host A to Host B. There are two links (and one switch) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of S bits each and adds 40 bits of header to each segment, forming packets of $L = 40 + S$ bits. Each link has a transmission rate of R bps. Find the value of S that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay.

解：

设 $d_{all}(F, S, R)$ 表示将文件从主机A发送到主机B的全部延迟，即需要求解

$$\begin{aligned} & \arg \min_S d_{all}(F, S, R) \\ d_{all}(F, S, R) &= \frac{\lceil \frac{F}{S} \rceil \times (40 + S)}{R} + \frac{(40 + S)}{R} \\ &= \left(\left\lceil \frac{F}{S} \right\rceil + 1 \right) \times \frac{40 + S}{R} \\ &= \frac{1}{R} \left(40 \left\lceil \frac{F}{S} \right\rceil + S \left\lceil \frac{F}{S} \right\rceil + 40 + S \right) \end{aligned}$$

一般情况下， $S \ll F$ ，所以 $\left\lceil \frac{F}{S} \right\rceil \approx \frac{F}{S}$ ，所以

$$\begin{aligned} d_{all}(F, S, R) &\approx \frac{1}{R} \left(40 \times \frac{F}{S} + S \times \frac{F}{S} + 40 + S \right) \\ &= \frac{1}{R} \left(S + \frac{40F}{S} + F + 40 \right) \\ &\geq \frac{1}{R} \left(2\sqrt{40F} + F + 40 \right) \end{aligned}$$

当且仅当 $S = \sqrt{40F}$ 时，此不等式的等号成立

$\therefore S$ 取 $\sqrt{40F}$ 时，将文件从主机A发送到主机B的全部延迟最小。

□

第二章 应用层

P1. Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application-layer protocols besides HTTP are needed in this scenario?

首先，客户端需要根据 URL 中的域名找到 IP 地址，这需要 DNS 协议（这是应用层协议），而 DNS 协议是基于 UDP 协议的（这是传输层协议）。之后的 HTTP 协议如果不是 HTTP3.0，则是基于 TCP 协议的（这是传输层协议），如果是 HTTP3.0，则是基于 UDP 协议的（这是传输层协议）。

总结：如果是 HTTP3.0，则还需要应用层协议 DNS，传输层协议 UDP；如果不是 HTTP3.0，则还需要应用层协议 DNS，传输层协议 UDP、TCP。

P2. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that n DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT_1, \dots, RTT_n . Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

首先，需要查询 DNS，由于 DNS 基于 UDP，因此需要 $\sum_{i=1}^n RTT_i$ 的时间完成 DNS 查询。之后的网页应该是通过 HTTP 协议传输，假定这里使用的不是 HTTP 3.0，那么这里是基于 TCP 协议的 HTTP 协议，因此需要 RTT_0 的时间建立 TCP 连接，之后请求和响应还需要一个 RTT_0 的时间。

因此总的时间为：

$$\sum_{i=1}^n RTT_i + 2RTT_0$$

P3. Referring to Problem P8, suppose the HTML file references three very small objects on the same server. Neglecting transmission times, how much time elapses with

a. Non-persistent HTTP with parallel connections?

每次请求和响应都需要开启一个新的 TCP 连接，但这些连接可以同时进行，因此需要的时间为：

$$\sum_{i=1}^n RTT_i + 2RTT_0 + 2RTT_0 = \sum_{i=1}^n RTT_i + 4RTT_0$$

b. Non-persistent HTTP with no parallel TCP connections?

每次请求和响应都需要开启一个新的 TCP 连接，而且这些连接不能同时进行，因此需要的时间为：

$$\sum_{i=1}^n RTT_i + 2RTT_0 + 3 \times 2RTT_0 = \sum_{i=1}^n RTT_i + 8RTT_0$$

c. Persistent HTTP?

只开启一个 TCP 连接，并且在这个 TCP 连接中，HTTP 请求互相不阻塞，也就是说不需要等到前一个 HTTP 请求收到响应后再发送后一个 HTTP 请求，由于忽略传输时间，因此可以认为这三个请求同时发出，同时得到响应。但这三个请求还是会依赖于 HTML 文件的响应，因此在 DNS 查询后，需要建立一次 TCP 连接，之后在这个连接上进行两批 HTTP 的请求和响应，因此总的时间为：

$$\sum_{i=1}^n RTT_i + RTT_0 + 2RTT_0 = \sum_{i=1}^n RTT_i + 3RTT_0$$

P4. What is the difference between *MAIL FROM:* in SMTP and *From:* in the mail message itself?

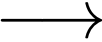
SMTP 中的 *MAIL FROM* 表示发信的 SMTP 服务器的地址与发信的用户名，而邮件信息中的 *From* 是用户自己填写的内容。

也就是说，一封邮件可能经过很多次转发，而 SMTP 中的 *MAIL FROM* 就是每次转发时直接发信的服务器和用户名，而邮件信息中的 *From* 是写信的人。

当然，由于 *From* 是可以由用户自己编辑的，所以用户也可以填写一个虚假的发信的邮箱地址，而用户就无法在 *MAIL FROM* 上造假了（除非用户用自己的 SMTP 服务器）。

P5. Consider accessing your e-mail with POP3.

a. Suppose you have configured your POP mail client to operate in the *download-and-keep* mode. Complete the following transaction:

<pre>C: list S: 1 498 S: 2 912 S: . C: retr 1 S: blah blah ... S:blah S: . ?</pre>		<pre>C: list S: 1 498 S: 2 912 S: . C: retr 1 S: blah blah ... S:blah S: . C: retr 2 S: blah blah ... S:blah S: . C: quit S: +OK POP3 server signing off</pre>
--	---	--

b. Suppose you have configured your POP mail client to operate in the *download-and-delete* mode. Complete the following transaction:

```

S: .....blah
S: .
C: dele 1
C: retr 2
S: blah blah ...
S: .....blah
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

```

→

```

S: .....blah
S: .
?
?

```

- c. Suppose you have configured your POP mail client to operate in the *download-and-keep* mode. Using your transcript in part (b), suppose you retrieve messages 1 and 2, exit POP, and then five minutes later you again access POP to retrieve new e-mail. Suppose that in the five-minute interval no new messages have been sent to you. Provide a transcript of this second POP session.

```

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: .....blah
S: .
C: retr 2
S: blah blah ...
S: .....blah
S: .
C: quit
S: +OK POP3 server signing off

```

- P6. Consider query flooding, as discussed in Section 2.6. Suppose that each peer is connected to at most N neighbors in the overlay network. Also suppose that the node-count field is initially set to K . Suppose Alice makes a query. Find an upper bound on the number of query messages that are sent into the overlay network.

每个节点最多向 N 个邻居节点发送查询信息，但是不应向给自己发送查询信息的服务器发送查询信息，因此除了 Alice 外，每个邻居最多向 $N-1$ 个邻居节点发送查询信息。而 Alice 因为是查询的发起者，所以最多向 N 个邻居节点发送查询信息。

因此，设 Alice 发起的查询为第 0 次查询，邻居发送给邻居的邻居为第 1 次查询，依此类推。设 a_i ($i \in \mathbb{N}$) 表示第 i 次查询时新收到查询的邻居数，则 $a_{i+1} = a_i \times (N-1)$ ，且 $a_0 = N$ 。这样的查询最多执行 K 轮，那么最多发送的查询信息数也就是所有收到查询的邻居数的总和，因此只需要将每次查询时新收到查询的邻居数求和即可。

因此最多发送的查询信息数为：

$$\underbrace{a_0 + a_1 + a_2 + \cdots}_{K \text{ 个}} = \underbrace{N + N(N-1) + N(N-1)^2 + \cdots}_{K \text{ 个}}$$

可以看到 $\{a_n\}_{n=1}^{\infty}$ 构成等比数列，所求的值为它的前 K 项求和，因此上式可化为：

$$\frac{N(1 - (N-1)^K)}{1 - (N-1)} = \frac{N(1 - (N-1)^K)}{2 - N}$$

第三章 传输层

P1. True or false?

- a. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is m , then the sequence number for the subsequent segment will necessarily be $m + 1$.

错误，下一次连接的序列号应为 m 加上这次发送的数据部分的字节数。

- b. Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send Host A. Host B will not send acknowledgments to Host A because Host B cannot piggyback the acknowledgments on data.

错误，即使没有数据要发送，也要发送 ACK 报文，否则就违背了可靠传输协议的原则。

- c. Suppose that the last SampleRTT in a TCP connection is equal to 1 sec. The current value of Timeout Interval for the connection will necessarily be ≥ 1 sec.

正确，

$$sRTT = 0.875sRTT + 0.125SampleRTT$$

$$DevRTT = 0.75DevRTT + 0.25|SampleRTT - sRTT|$$

$$Timeout = sRTT + 4DevRTT$$

若在某一时刻，假设 $sRTT$ 和 $DevRTT$ 都非常接近 0，而最后一个 $SampleRTT$ 为 1 秒，则在下一时刻可以计算得到：

$$sRTT = 0.125\text{秒} \quad DevRTT = 0.25\text{秒} \quad Timeout = 0.125 + 4 \times 0.25 = 1.125\text{秒}$$

若 $sRTT$ 增加或者 $DevRTT$ 增加，都会使 Timeout 增加，因此 $Timeout \geq 1.125\text{秒}$ ，所以 $Timeout \geq 1\text{秒}$ 正确。

- d. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.

正确。

e. The size of the TCP RcvWindow never changes throughout the duration of the connection.

错误，TCP RcvWindow 会根据接收端的接收能力变化。

f. The TCP segment has a field in its header for Rcvwindow.

正确。

g. Suppose Host A sends one segment with sequence number 38 and 4 bytes of data over a TCP connection to Host B. In this same segment the acknowledgment number is necessarily 42.

错误，确认序号为下一个希望对方发送给自己的段序号，而发送序号为自己发送给对方的段序号，即前者为 B 维护的序号，后者为 A 维护的序号，这两者互不影响。

P2. UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010101, 01110000, 01001100. What is the 1s complement of the sum of these 8-bit bytes?(Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work.

$$\begin{array}{r}
 01010101 \\
 01110000 \\
 + \quad 01001100 \\
 \hline
 \text{直接求和} \quad 100010001 \\
 \text{进位加到末尾} \quad 00010010 \\
 \text{取反} \quad 11101101
 \end{array}$$

所以 8 位校验和是 11101101。

Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum?

使用反码后，接收端校验时只需要把整个报文一起进行回卷求和，再取反，若结果为全 1，则通过校验；而不需要先把报文中的校验和字段置为 0 再进行求和回卷取反。

With the 1s complement scheme, how does the receiver detect errors?

接收端对整个报文和发送方进行相同的操作，包括校验和本身，即将报文分割成 16 位的字，若报文为奇数个字节（即分割到最后不足 16 位），则将最后一个字节在末尾增加 0 补充

到 16 位；之后将这些 16 位的字求和，将超过 16 位的高位部分（即进位）加到低 16 位的部分；再进行取反，此时结果如果为全 1，则说明校验正确，否则说明校验错误。

Is it possible that a 1-bit error will go undetected? How about a 2-bit error?

1 位的错误必定会被发现，因为 1 位的错误会导致最终的校验结果中的其中一位的 1 变为 0，而 2 位的错误可能不被发现。

P3. Consider our motivation for correcting protocol rdt2.1. Show that the receiver, shown in the following figure, when operating with the sender shown in Figure 3.11, can lead the sender and receiver to enter into a deadlock state, where each is waiting for an event that will never occur.

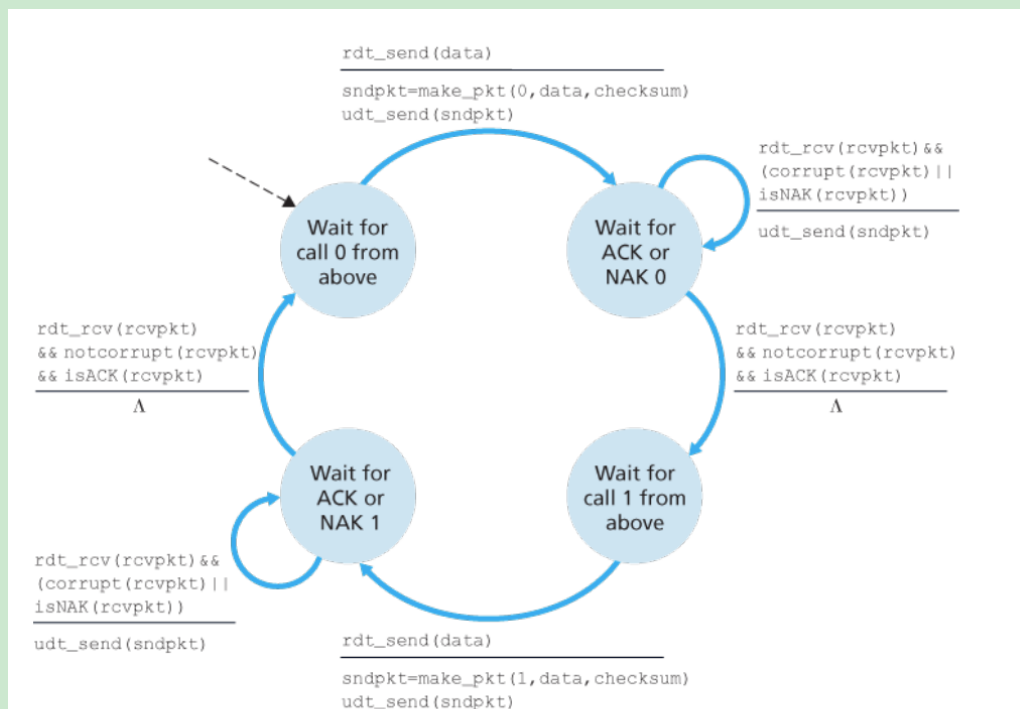
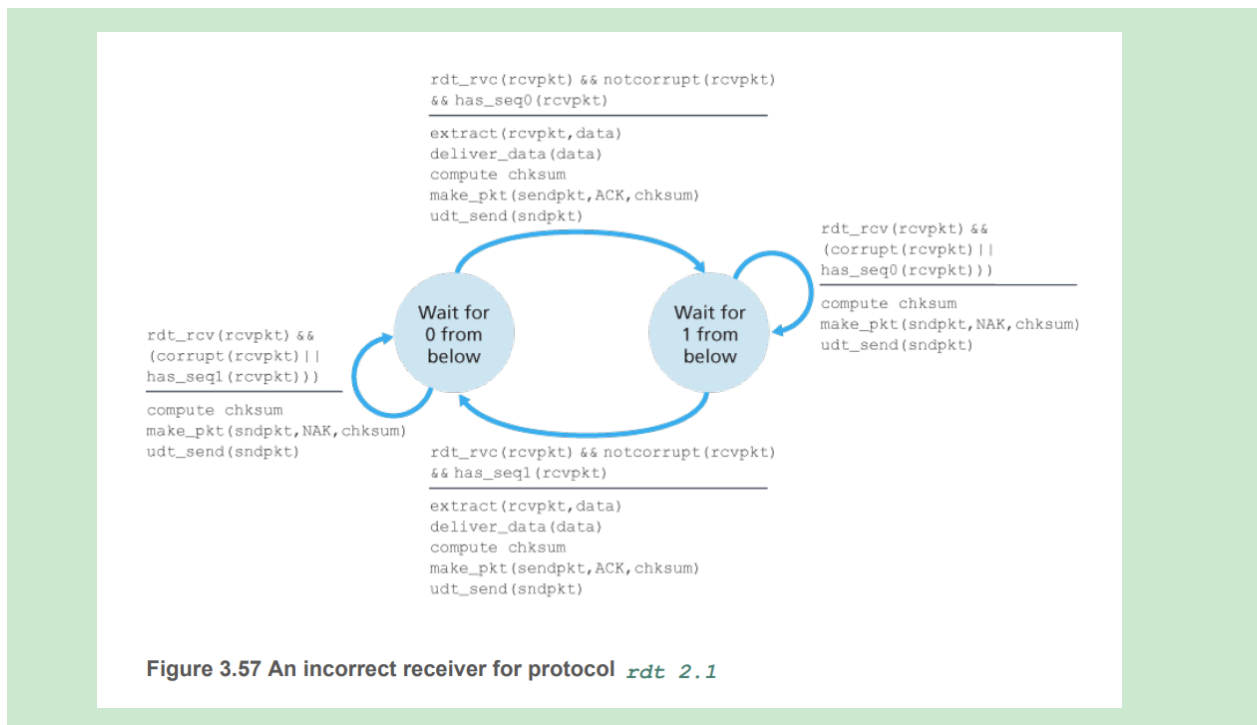


Figure 3.11 *rdt2.1* sender



首先，rdt2 考虑的是具有比特差错信道上的可靠数据传输（也就是不会丢包）。那么考虑如下情况，发送端发送了序号 0 的包，接收端接收，校验正确，返回 ACK 并转换状态，但 ACK 出错，发送端校验错误后重新发送序号 0 的包，此时接收端已经处于等待序号 1 的包的状态，它接收到序号 0 的包，会返回 NAK，发送端收到 NAK，又会重发序号 0 的包，这样就会导致死锁，发送端一直发送序号 0 的包，接收端一直返回 NAK。

- P4. Consider the GBN and SR protocols. Suppose the sequence number space is of size k . What is the largest allowable sender window that will avoid the occurrence of problems such as that in Figure 3.27 for each of these protocols?

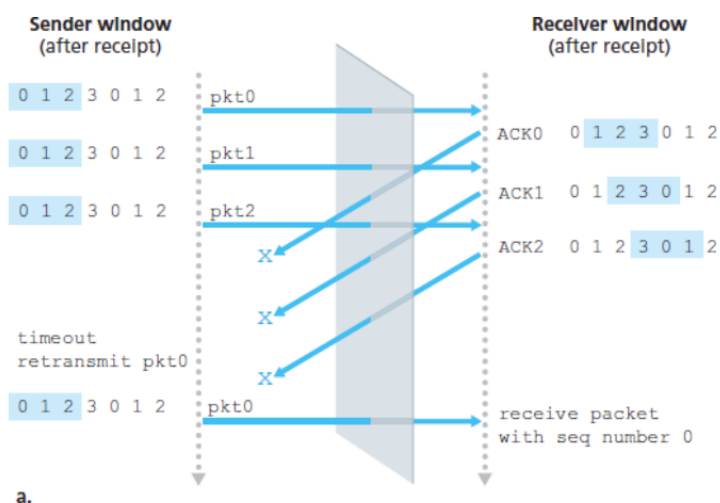
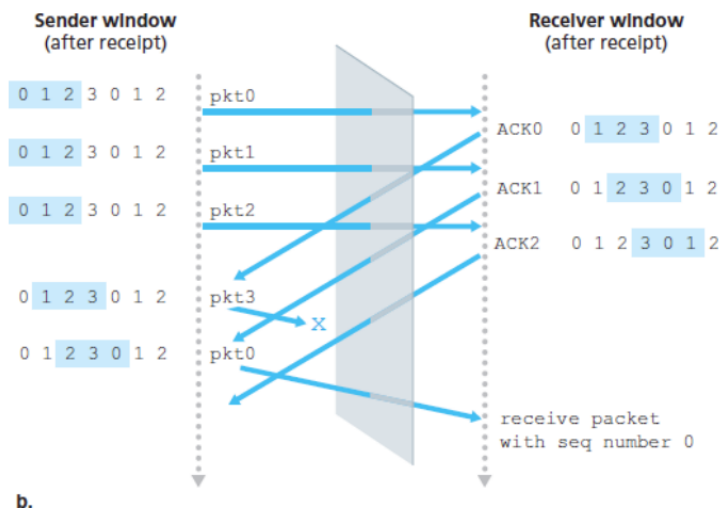


Figure 3.27 SR receiver dilemma with too-large windows: A new packet or a retransmission?



对于 GBN 协议，最大的发送端窗口大小为 $2^k - 1$ ；对于 SR 协议，最大的发送端窗口大小为 2^{k-1} 。

P5. P24. Consider transferring an enormous file of L bytes from Host A to Host B. (P24 MSS 为 536 字节)

a. What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.

由于 TCP 协议的序号有 4 个字节，那么序号范围就是 $0 \sim 2^{4 \times 8} - 1$ ，因此如果需要 TCP 的序号不耗尽， L 最大取 $2^{4 \times 8} = 4294967296 = 4\text{Gi}$ 。
(1Gi=1024Mi=1048576Ki=1073741824)

- b. For the L you obtain in (a.) , find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 10 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

一共要分 $\frac{L}{MSS} = \frac{2^{32}}{536}$ 个段，每段的长度为 $536 + 66 = 602$ (bytes)，于是需要传输的数据的总长度为 $(\frac{2^{32}}{536} \times 602)$ bytes。

所以传输文件需要的时间为

$$\frac{(\frac{2^{32}}{536} \times 602) \text{ bytes}}{10 \text{ Mbps}} = \frac{(\frac{2^{32}}{536} \times 602 \times 8) \text{ bit}}{(10 \times 2^{20}) \text{ bit/s}} = \frac{1232896}{335} \approx 3680.28656716418 \text{ 秒} \approx 61.338 \text{ 分钟}$$

P6. Consider Figure 3.58.

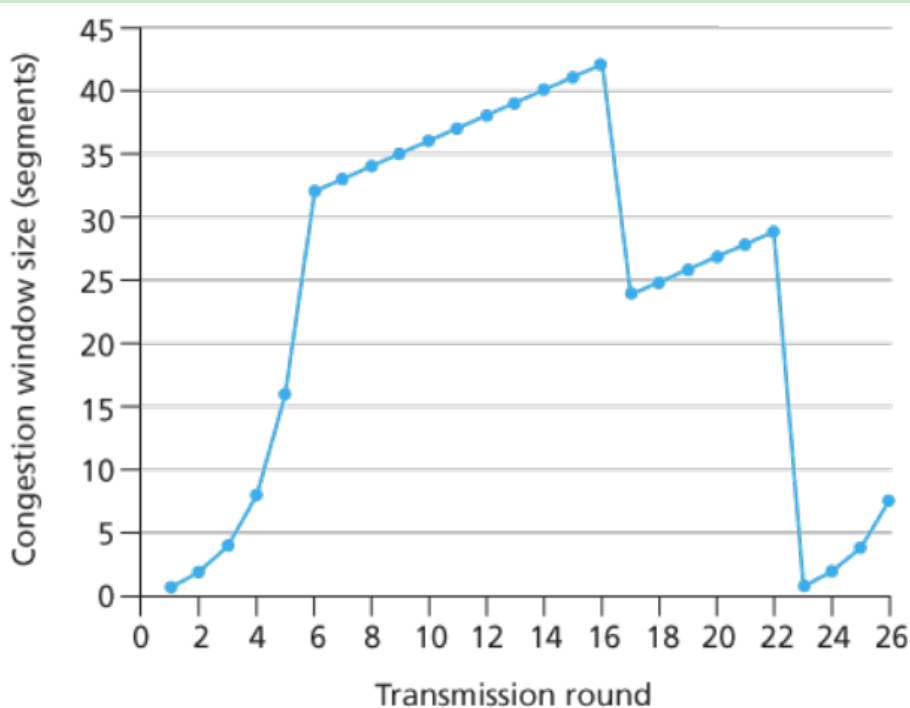


Figure 3.58 TCP window size as a function of time

Assuming TCP **Reno** is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.

- a. What is the value of Threshold $ssthresh$ at the 18th transmission round?

第 16 轮时的拥塞窗口为 42，第 17 轮拥塞窗口和 $ssthresh$ 都变为 42 的一半即 21，第 18 轮 $ssthresh$ 未改变，因此还是 21。

b. What is the value of Threshold *ssthresh* at the 24th transmission round?

第 22 轮时的拥塞窗口为 26，第 23 轮 *ssthresh* 变为 26 的一半即 13，第 24 轮 *ssthresh* 未改变，因此还是 13。

c. Identify the intervals of time when TCP slow start is operating.

即图中拥塞窗口大小指数增长的部分，即第 1 到 6 轮、第 23 到 26 轮。

d. Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of Threshold *ssthresh*?

第 26 轮原先的拥塞窗口大小为 8，*ssthresh* 为 13，收到三个重复 ACK 后拥塞窗口大小变为 8 的一半即 4，*ssthresh* 也变为 8 的一半即 4。

e. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

第 17 轮的拥塞窗口变为第 16 轮的一半，这说明发送端收到了三个重复的 ACK。

f. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

第 23 轮的拥塞窗口变为 1，这说明发送端通过超时检测到了丢包。

g. Identify the intervals of time when TCP congestion avoidance is operating.

即图中拥塞窗口大小线性增长的部分，即第 6 到 16 轮、第 17 到第 22 轮。

h. What is the initial value of Threshold *ssthresh* at the first transmission round?

在第 7 轮时由慢启动状态转换到拥塞避免状态，此时的拥塞窗口大小为 32，这说明第 1 轮初始的 *ssthresh* 为 32。

i. During what transmission round is the 70th segment sent?

每一轮的拥塞窗口大小为这一轮发的段的数量，将它们按顺序列出来：

$$1, 2, 4, 8, 16, 32, 33, \dots$$

则可以注意到前 6 轮发的段的数量为 $1 + 2 + 4 + 8 + 16 + 32 = 63$ ，前 7 轮发的段的数量为 $1 + 2 + 4 + 8 + 16 + 32 + 33 = 96$ ，而 $63 < 70 < 96$ ，因此第 70 个段在第 7 轮发出。

- j. Suppose TCP **Tahoe** is used (instead of TCP **Reno**), and assume that triple duplicate ACKs are received at the 16th round. What are the *ssthresh* and the congestion window size at the 19th round?

第 16 轮原先的拥塞窗口为 42，若使用 TCP **Tahoe**，则收到三个重复的 ACK 后，第 17 轮的 *ssthresh* 变为 42 的一半即 21，拥塞窗口大小变为 1。因此第 19 轮的 *ssthresh* 为 21，拥塞窗口大小为 4。

- k. Again suppose TCP **Tahoe** is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

第 17 轮到第 22 轮的拥塞窗口大小依次为 1, 2, 4, 8, 16, 21，每一轮的拥塞窗口大小为这一轮发的段的数量，第 22 轮的超时事件只会影响第 22 轮以后的发的包的数量而不会影响第 22 轮发的包的数量，因此从第 17 轮（包括）到 22 轮（包括）发的包的数量为：

$$1 + 2 + 4 + 8 + 16 + 21 = 52$$

【2010 统考真题】 主机甲和主机乙之间已建立一个 TCP 连接，TCP 最大段长为 1000B。若主机甲的当前拥塞窗口为 4000B，在主机甲向主机乙连续发送两个最大段后，成功收到主机乙发送的第一个段的确认段，确认段中通告的接收窗口大小为 2000B，则此时主机甲还可以向主机乙发送的最大字节数是 (B)。

- A. 1000 B. 2000 C. 3000 D. 4000

主机甲接收到主机乙发送的第一个段的确认段后，发现接收窗口小于当前拥塞窗口，因此窗口暂停滑动直到 2000B（接收窗口大小）以前的数据都已确认，而原先的发送窗口大小为 4000B，发送了两个最大段（即 2000B），因此还可以发 2000B 的字节数。

所以此题选 B。

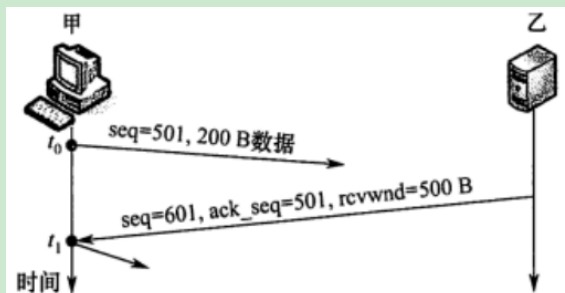
【2020 统考真题】 若主机甲与主机乙建立 TCP 连接时，发送的 SYN 段中的序号为 1000，在断开连接时，甲发送给乙的 FIN 段中的序号为 5001，则在无任何重传的情况下，甲向乙已经发送的应用层数据的字节数为 (C)。

- A. 4002 B. 4001 C. 4000 D. 3999

由题意可知，是主机甲主动发起连接，并且主机甲主动断开连接。主机甲发送的 SYN 段不应携带应用层数据，之后主机乙向主机甲发送 ACK+SYN，主机甲再发送 ACK 段时可以携带应用层数据，此时这个携带数据的 ACK 的序号为 1001。而甲发送给乙的 FIN 段（序号为 5001）也不应携带应用层数据，所以携带应用层数据的字节序号为 1001 到 5000，因此发送的应用层数据的字节数为 $5000 - 1001 + 1 = 4000$ 。

所以此题选 C。

【2021 统考真题】设主机甲通过 TCP 向主机乙发送数据，部分过程如下图所示。甲在 t_0 时刻发送一个序号 $seq=501$ 、封装 200B 数据的段，在 t_1 时刻收到乙发送的序号 $seq=601$ 、确认序号 $ack_seq=501$ 、接收窗口 $rcvwnd=500B$ 的段，则甲在未收到新的确认段之前，可以继续向乙发送的数据序号范围是 (A)。



A. 501 ~ 1000

B. 601 ~ 1100

C. 701 ~ 1000

D. 801 ~ 1100

此题中，主机甲初始的发送窗口大小未知，那么只能根据甲在 t_1 时刻收到乙发送的段来判断，此段中确认序号为 501，所以甲需要重发序号为 501 的段；接收窗口为 500B，则说明甲在收到新的确认段之前，最多只能发送 500B 的应用层数据，因此甲可以继续向乙发送的数据序号范围是 501 ~ 1000。

所以此题选 A。

第四章 网络层

R13. Suppose Host A sends Host B a TCP segment encapsulated in an IP datagram. When Host B receives the datagram, how does the network layer in Host B know it should pass the segment (that is, the payload of the datagram) to TCP rather than to UDP or to something else?

在 IP 协议的头部，有一个字段指示了高层协议的类型，主机 B 的网络层根据这个字段的值决定它应该把这个 IP 报文负载的段传递给哪个高层协议。

R16. Suppose an application generates chunks of 40 bytes of data every 20 msec, and each chunk gets encapsulated in a TCP segment and then an IP datagram. What percentage of each datagram will be overhead, and what percentage will be application data?

假设 TCP 层和 IP 层均没有额外的选项字段，那么 TCP 的头部为 20 字节，IP 的头部为 20 字节，应用层总长度也就是每个 chunk 为 40 字节，那么每个报文的头部占比为 $\frac{20+20}{20+20+40} \times 100\% = 50\%$ ，应用层占比为 $\frac{40}{20+20+40} \times 100\% = 50\%$ 。

R20. Suppose you purchase a wireless router and connect it to your cable modem. Also suppose that your ISP dynamically assigns your connected device (that is, your wireless router) one IP address. Also suppose that you have five PCs at home that use 802.11 to wirelessly connect to your wireless router. How are IP addresses assigned to the five PCs? Does the wireless router use NAT? Why or why not?

在没有进行特别设置的情况下，五台个人电脑的 IP 地址由无线路由器提供，并且是局域网 IP 地址，但为了使局域网中的电脑能访问互联网，路由器需要使用 NAT（网络地址转换），将局域网内其他设备的 IP 和端口映射到路由器的 IP 和端口，从而使局域网中的其他设备都能访问互联网。

R22. Compare and contrast link-state and distance-vector routing algorithms.

比较项目	链路状态 (Link-State) 算法	距离矢量 (Distance-Vector) 算法
原理	每个节点与所有其他节点广播交流, 只告知与其直接相连链路的费用。	每个节点只与邻居互相交流, 得到邻居的新费用, 并告知邻居自己的当前最低费用。
报文复杂性	知道网络每条链路的费用, 需发送 $O(nE)$ 个报文; 当一条链路的费用变化时, 必须通知所有节点。	迭代时, 在两个直接相连邻居之间交换报文; 收敛时间受许多因素影响; 当链路费用改变时, 只有该链路相连的节点的最低费用路径发生改变时, 才传播已改变的链路费用。
收敛速度	需要 $O(nE)$ 个报文和 $O(n^2)$ 的搜寻。	收敛较慢。可能会遇到选路回环, 或计数到无穷的问题。
健壮性	当一台路由器发生故障、操作错误或受到破坏时, 路由器向其连接的一条链路广播不正确费用。路由计算基本独立 (仅计算自己的转发表), 有一定健壮性。	当一台路由器发生故障、操作错误或受到破坏时, 一个节点可向任意或所有目的节点发布其不正确的最低费用路径。一个节点的计算值会传递给它的邻居, 并间接地传递给邻居的邻居。一个不正确的计算值会扩散到整个网络。

R24. Why are different inter-AS and intra-AS protocols used in the Internet?

策略:

自治系统之间: 管理员想要控制网络流量如何流动以及其他流量是否能通过它的网络。

自治系统内部: 单个管理员, 因此不需要这些决策。

规模:

层次化的路由减少了路由表的大小, 减少了更新路由表的流量。

性能:

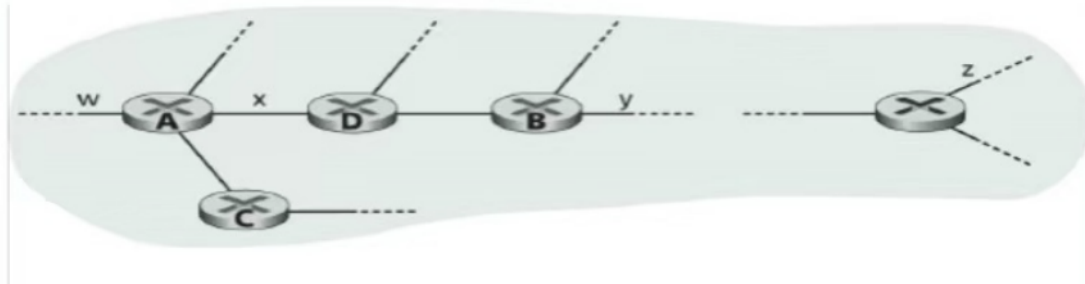
自治系统内部: 能够专注于性能。

自治系统之间: 策略可能比性能更重要。

R25. Consider Figure 4.33. Starting with the original table in D, suppose that D receives from A the following advertisement:

Destination Subnet	Next Router	Number of Hops to Destination
z	C	10
w	—	1
x	—	1
...

Figure 4.33 ♦ Routing table in router *D* before receiving advertisement from router *A*



Destination Subnet	Next Router	Number of Hops to Destination
w	A	2
y	B	2
z	B	7
x	—	1
....

Will the table in *D* change? If so how?

D 的路由表不会改变。

首先，可以看出 *A* 向 *D* 提供了 *A* 分别到 *z*, *w*, *x* 的下一跳和总跳数，那么之后 *D* 会查看自己的路由表。*D* 原先记录的到 *z* 的总跳数是 7，但如果经过 *A* 再到 *z*，那就会是 $1 + 10 = 11$ ，比原先的跳数还多，因此不会更新 *z* 这一行；*D* 原先记录的到 *w* 的总跳数是 2，从 *A* 收到的信息来看，经过 *A* 再到 *w* 的跳数就是 $1 + 1 = 2$ ，和原来一样，因此也不会更新 *w* 这一行；*D* 原先记录的到 *x* 的跳数是 1，已经最少了，不需要更新。

所以 *D* 收到的 *A* 的前三行信息都不会导致 *D* 更新，第三行以下的信息也没给出，因此可以认为 *D* 的路由表不会改变。

P11. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

Prefix Match	Interface
1	0
11	1
111	2
otherwise	3

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

接口	目的主机地址关联范围	目的主机地址数
0	1000 0000 - 1011 1111	$2^6 = 64$
1	1100 0000 - 1101 1111	$2^5 = 32$
2	1110 0000 - 1111 1111	$2^5 = 32$
3	0000 0000 - 0111 1111	$2^7 = 128$

检查一下，目的主机地址数总和为 $64 + 32 + 32 + 128 = 256 = 2^8$ ，正确。

P13. Consider a subnet with prefix 101.101.101.64/26. Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network.

直接把表示网络号的 IP 地址加一即可，即 101.101.101.65。

Suppose an ISP owns the block of addresses of the form 101.101.128/17. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

由于子网掩码为 $17 = 8 + 8 + 1$ 位，所以 IP 地址的前两个部分以及第三个部分的第一位表示网络号，因此 IP 地址第三个部分还有 $2^{8-1} = 128$ 个数字可以使用，要分成 4 个子网， $128 \div 4 = 32$ ， $17 + 2 = 19$

$128 + 0 \times 32 = 128$, $128 + 1 \times 32 = 160$, $128 + 2 \times 32 = 192$, $128 + 3 \times 32 = 224$ 。

所以四个子网的前缀为

101.101.128.0/19
 101.101.160.0/19
 101.101.192.0/19
 101.101.224.0/19

P14. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

Prefix Match	Interface
00	0
01	1
10	2
11	3

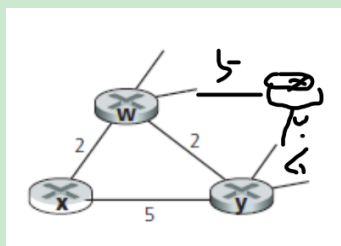
For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

接口	目的主机地址关联范围	目的主机地址数
0	0000 0000 - 0011 1111	$2^6 = 64$
1	0100 0000 - 0111 1111	$2^6 = 64$
2	1000 0000 - 1011 1111	$2^6 = 64$
3	1100 0000 - 1111 1111	$2^6 = 64$

P18. In this problem we'll explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without application-specific NAT configuration. If you have difficulty devising such a technique, discuss why.

由于 Arnold 和 Bernard 都在 NAT 后面，因此他们都只能发现对方所在子网的路由器的 IP 地址（公网 IP），而 TCP 连接是端到端的，因此如果没有对 NAT 进行特别配置，那么只能采用 TCP 打洞方式，TCP 打洞具体过程比较复杂，这里就简单描述一下，大致是通过一个具有公网 IP 的服务器对 TCP 三次握手的报文进行中继，使每一方与对方的路由器进行通信，但对于上层来说就好像是双方形成了 TCP 连接，而且 TCP 打洞还不一定能成功，例如对于对称 NAT 就无法成功，详细过程就不再叙述了，可以查找相关资料。

P20. Consider the network fragment shown below. x has only two attached neighbors, w and y . w has a minimum-cost path to destination u (not shown) of 5, and y has a minimum-cost path to u of 6. The complete paths from w and y to u (and between w and y) are not shown. All link costs in the network have strictly positive integer values.



a. Give x 's distance vector for destinations w , y , and u .

终点	下一跳	花费/代价/损失
w	—	2
y	w	4
u	w	7

b. Give a link-cost change for either $c(x, w)$ or $c(x, y)$ such that x will inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

若 $c(x, w) = 10$, 则 x 到 u 的最小代价的距离更新为 11, 且下一跳更新为 y , 此时 x 会通知 w 和 y 它的距离矢量发生了变化。

- c. Give a link-cost change for either $c(x, w)$ or $c(x, y)$ such that x will *not* inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

若 $c(x, y) = 6$, 则 x 到 u 的最小代价仍然是 7, 下一跳仍然是 w , 此时 x 不会通知它的邻居。

第五章 数据链路层

R1. What are some of the possible services that a link-layer protocol can offer to the network layer? Which of these link-layer services have corresponding services in IP? In TCP?

数据链路层可以提供流量控制，差错检测和纠正，共享广播信道-多路访问，链路层编址。其中流量控制、差错检测在 IP 和 TCP 中都有相应的服务，链路层编址在 IP 有类似的 IP 地址。

R2. If all the links in the Internet were to provide reliable delivery service, would the TCP reliable delivery service be redundant? Why or why not?

如果在因特网中的所有链路都能提供可靠收发服务，TCP 的可靠收发服务也不是多余的。这样虽然能保证到达的包没有差错，但是 IP 层可能把数据包转发到错误的地方导致丢包，或者各个数据包通过不同的链路到达目的，导致数据包乱序，TCP 的可靠收发服务需要解决丢包和乱序的问题。

R9. Why is an ARP query sent within a broadcast frame? Why is an ARP response sent within a frame with a specific destination MAC address?

ARP 查询使用广播帧是因为发送端此时还不知道接收端的 MAC 地址，无法单播，所以只能使用广播帧；而 ARP 响应帧理论上使用广播 MAC 地址也行，只要查询方能收到就行，但其他没有发送 ARP 查询帧的主机收到 ARP 响应帧也会丢弃，因此还不如不使用广播帧，其他主机收到目的 MAC 不是自己的帧就提早丢弃了。

R10. Suppose nodes A, B, and C each attach to the same broadcast LAN (through their adapters). If A sends thousands of IP datagrams to B with each encapsulating frame addressed to the MAC address of B, will C's adapter process these frames? If so, will C's adapter pass the IP datagrams in these frames to the network layer C? How would your answers change if A sends frames with the MAC broadcast address?

C 的适配器会将这些帧首部的目的 MAC 地址和 C 的 MAC 地址比较，发现不相同，就把这些帧丢弃了，所以 C 的适配器会处理这些帧，但是将这些帧丢弃了，所以不会将 IP 数据传递给 C 的网络层。但是如果 A 发送的帧使用了 MAC 广播地址，那么 C 的适配器会处理这些帧，并且将帧里的 IP 数据包传递给 C 的网络层（网络层还是有可能丢弃报文的）。

R11. In CSMA/CD, after the fifth collision, what is the probability that a node chooses $K = 4$? The result $K = 4$ corresponds to a delay of how many seconds on a 10 Mbps Ethernet?

第五次冲突后, K 的取值范围为 $[0, 2^5 - 1]$, 即 K 等可能地从 $[0, 31]$ 中选取一个整数, 因此 $P(K = 4) = \frac{1}{32}$ 。

而 $K = 4$ 就意味着延迟 $4 \times 512 \text{ bit}$ 的传输时间, 因此在 10 Mbps 的以太网中, 延时为

$$\text{Delay} = \frac{4 \times 512 \text{ bit}}{10 \text{ Mbps}} = \frac{2048 \text{ bit}}{10 \times 10^6 \text{ bit/s}} = 0.0002048 \text{ 秒}$$

R13. Suppose a 10 Mbps adapter sends into a channel an infinite stream of 1s using Manchester encoding. The signal emerging from the adapter has how many transitions per second?

曼彻斯特编码在每位信号的中间都有一个跳变, 由于数据全 1, 所以在每位信号的开始处还需要归零, 因此一位信号有 2 个跳变。而在 10 Mbps 的适配器上每秒能传输 10×10^6 位信号, 所以每秒的跳变数为 20×10^6 。

P1. Suppose the information content of a packet is the bit pattern 1010 1010 1010 1011 and an even parity scheme is being used. What would the value of the field containing the parity bits be for the case of a two-dimensional parity scheme? Your answer should be such that a minimum-length checksum field is used.

由于需要二维奇偶校验和长度最小, 并且此数据包的长度为 16 个字节, $\sqrt{16} = 4$, 所以应当一行 4 个字节, 共 4 行。校验过程可以画出二维表格如下:

1	0	1	0	0
1	0	1	0	0
1	0	1	0	0
1	0	1	1	1
0	0	0	1	1

于是带校验的数据为 10100 10100 10100 10111 00011。

P15. Suppose nodes A and B are on the same 10 Mbps Ethernet bus, and the propagation delay between the two nodes is 225 bit times. Suppose node A begins transmitting a frame and, before it finishes, node B begins transmitting a frame. Can A finish transmitting before it detects that B has transmitted? Why or why not? If the answer is yes, then A incorrectly believes that its frame was successfully transmitted without a collision. *Hint:* Suppose at time $t = 0$ bit times, A begins transmitting a frame. In the worst case, A transmits a minimum-sized frame of $512 + 64$ bit times. So A would finish transmitting the frame at $t = 512 + 64$ bit times. Thus, the answer is no, if B's signal reaches A before bit time $t = 512 + 64$ bits. In the worst case, when does B's signal reach A?

题目中已经给出了提示，在最差的情况下，A 传输的帧为 8 个字节的前同步码加上 64 个字节的帧的有效部分（数据字段长度最小为 46 字节），也就是 $64 + 512 = 576$ 位。A 的信号到达 B 需要 225 个比特时间，在最差的情况下，在 A 发出的帧到达 B 的一瞬间，B 发出了自己的帧，此时 B 的信号再到达 A 又需要 225 个比特时间，因此在第 $2 \times 225 = 450$ 个比特时间，B 的信号到达 A， $450 < 576$ ，因此 A 不会在检测到 B 传输之前结束传输。

P16. Suppose two nodes, A and B, are attached to opposite ends of an 900 m cable, and that they each have one frame of 1,000 bits (including all headers and preambles) to send to each other. Both nodes attempt to transmit at time $t = 0$. Suppose there are four repeaters between A and B, each inserting a 20-bit delay. Assume the transmission rate is 10 Mbps, and CSMA/CD with backoff intervals of multiples of 512 bits is used. After the first collision, A draws $K = 0$ and B draws $K = 1$ in the exponential backoff protocol. Ignore the jam signal and the 96-bit time delay.

a. What is the one-way propagation delay (including repeater delays) between A and B in seconds? Assume that the signal propagation speed is 2×10^8 m/sec.

这里说的包括中继器的延时是要把每个中继器 20 比特时间的延时也算进去？但这不是传输延时的范畴了？不过既然说了那就这样算吧。

信号传播速度是 2×10^8 m/sec，AB 之间使用 900 m 的电缆连接。因此在不考虑中继器的情况下传播延时为

$$\frac{900\text{m}}{2 \times 10^8\text{m/sec}} = 4.5 \times 10^{-6} \text{ 秒}$$

每个中继器有 20 比特时间的延时，传输率为 10 Mbps，因此每个中继器的延时为

$$\frac{20\text{bit}}{10\text{Mbps}} = \frac{20\text{bit}}{10 \times 10^6\text{bit/s}} = 2 \times 10^{-6} \text{ 秒}$$

所以在考虑中继器的情况下总的传播延时为

$$4.5 \times 10^{-6} + 4 \times 2 \times 10^{-6} = 12.5 \times 10^{-6} \text{ 秒}$$

b. At what time (in seconds) is A's packet completely delivered at B?

这里为什么用的是“packet”？“packet”不应该是指网络层的协议数据单元吗？这题不是关注数据链路层吗？而且题目中也说了传输了一帧（frame）。

首先将上一小题计算的传播延时转化为比特时间，数据传输率为 10 Mbps，所以

$$10\text{Mbps} \times (12.5 \times 10^{-6})\text{sec} = (10 \times 10^6)\text{bit/s} \times (12.5 \times 10^{-6})\text{sec} = 125\text{bit} \text{ (比特时间)}$$

A 和 B 都在 $t = 0$ 时刻发出帧，在 $t = 125$ 时刻，A 和 B 都检测到了冲突，忽略 48 比特的拥塞信号，直接停止发送。A 在指数退避中选择了 $K = 0$ ，因此会立刻重新准备发送，此时信道忙，A 继续侦听。B 在指数退避中选择了 $K = 1$ ，因此会在 512 比特时间后重新发送帧。

在 $t = 125 + 125 = 250$ 时刻, B 发送的帧已经全部到达了 A, 此时信道空闲, 忽略 96 比特的侦听时间, A 立刻重新发送帧。

在 $t = 250 + 125 = 375$ 时刻, B 开始接受帧。

在 $t = 125 + 512 + 737$ 时刻, B 准备重发帧, 但信道忙 (B 正在接收帧), 因此不会发送。

在 $t = 375 + 1000 = 1375$ 时刻, B 完整接收了帧。

再将比特时间转化为秒

$$\frac{1375\text{bit}}{10 \times 10^6\text{bit/s}} = 137.5 \times 10^{-6} \text{ 秒}$$

所以在 137.5×10^{-6} 秒, A 的帧完全到达 B。

c. Now suppose that only A has a packet to send and that the repeaters are replaced with switches. Suppose that each switch has a 20-bit processing delay in addition to a store-and-forward delay. At what time, in seconds, is A's packet delivered at B?

只有 A 发送包, 因此不考虑冲突以及排队延时。这时就不便于使用比特时间来计算了, 因此还是使用秒为单位来计算。

传播延时只需要整体考虑 A 到 B 之间的线缆即可

$$t_{\text{propagation}} = \frac{900\text{m}}{2 \times 10^8\text{m/sec}} = 4.5 \times 10^{-6} \text{ 秒}$$

4 个交换机, 每个交换机的处理延时为 20 个比特时间, 应该是处理的这段时间里能传输 20 比特的数据的意思吧, 那么根据数据传输率为 10 Mbps, 可以计算出处理延时

$$t_{\text{processing}} = 4 \times \frac{20\text{bit}}{10 \times 10^6\text{bit/s}} = 8 \times 10^{-6} \text{ 秒}$$

由于每个交换机都采用存储转发方式, 所以传输延时为 (4 个交换机把链路分成了 5 段)

$$t_{\text{transmission}} = 5 \times \frac{1000\text{bit}}{10 \times 10^6\text{bit/s}} = 500 \times 10^{-6} \text{ 秒}$$

所以总延时为

$$\begin{aligned} t_{\text{all}} &= t_{\text{propagation}} + t_{\text{processing}} + t_{\text{transmission}} \\ &= 4.5 \times 10^{-6} + 8 \times 10^{-6} + 500 \times 10^{-6} \\ &= 512.5 \times 10^{-6} \text{ 秒} \end{aligned}$$

所以在 512.5×10^{-6} 秒, A 的帧完全到达 B。

P17. Suppose nodes A and B are on the same 10 Mbps Ethernet bus, and the propagation delay between the two nodes is 225 bit times. Suppose A and B send frames at the same time, the frames collide, and then A and B choose different values of K in the CSMA/CD algorithm. Assuming no other nodes are active, can the retransmissions from A and B collide? For our purposes, it suffices to work out the following example. Suppose A and B begin transmission at $t = 0$ bit times. They both detect collisions at $t = 225$ bit times. They finish transmitting

a jam signal at $t = 225 + 48 = 273$ bit times. Suppose $K_A = 0$ and $K_B = 1$. At what time does B schedule its retransmission? At what time does A begin transmission? (*Note:* The nodes must wait for an idle channel after returning to Step 2—see protocol.) At what time does A's signal reach B? Does B refrain from transmitting at its scheduled time?

上一题不考虑 48 bit 的拥塞信号时间和 96 bit 的侦听时间，这一题就要考虑了。

根据题意，A 和 B 都在 $t = 225$ 时刻检测到了冲突，在 $t = 225 + 48 = 273$ 时刻结束发送拥塞信号。之后在指数退避过程中 A 选择了 $K = 0$ ，因此立刻进入侦听状态。由于 B 之前发送的帧还没有完全到达 A，因此这时 A 侦听到信道忙。

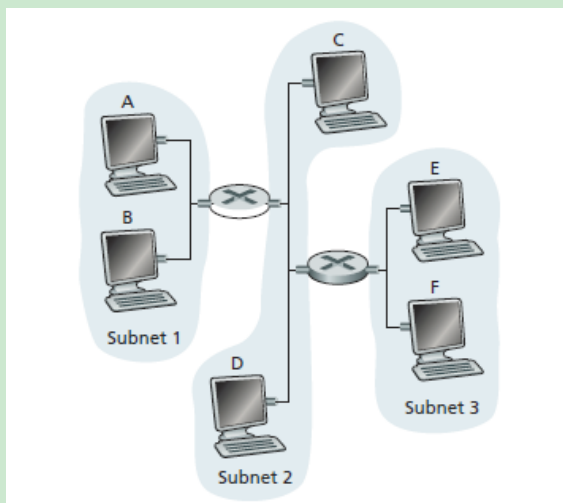
B 的信号在 225 个比特时间后完全到达 A，因此在 $t = 273 + 225 = 498$ 时刻，A 侦听到信道开始空闲，于是在 96 个比特时间后，也就是 $t = 498 + 96 = 594$ 时刻，A 开始重新发送帧。

在 225 个比特时间后，也就是 $t = 594 + 225 = 819$ 时刻，A 重新发送的帧开始到达 B。

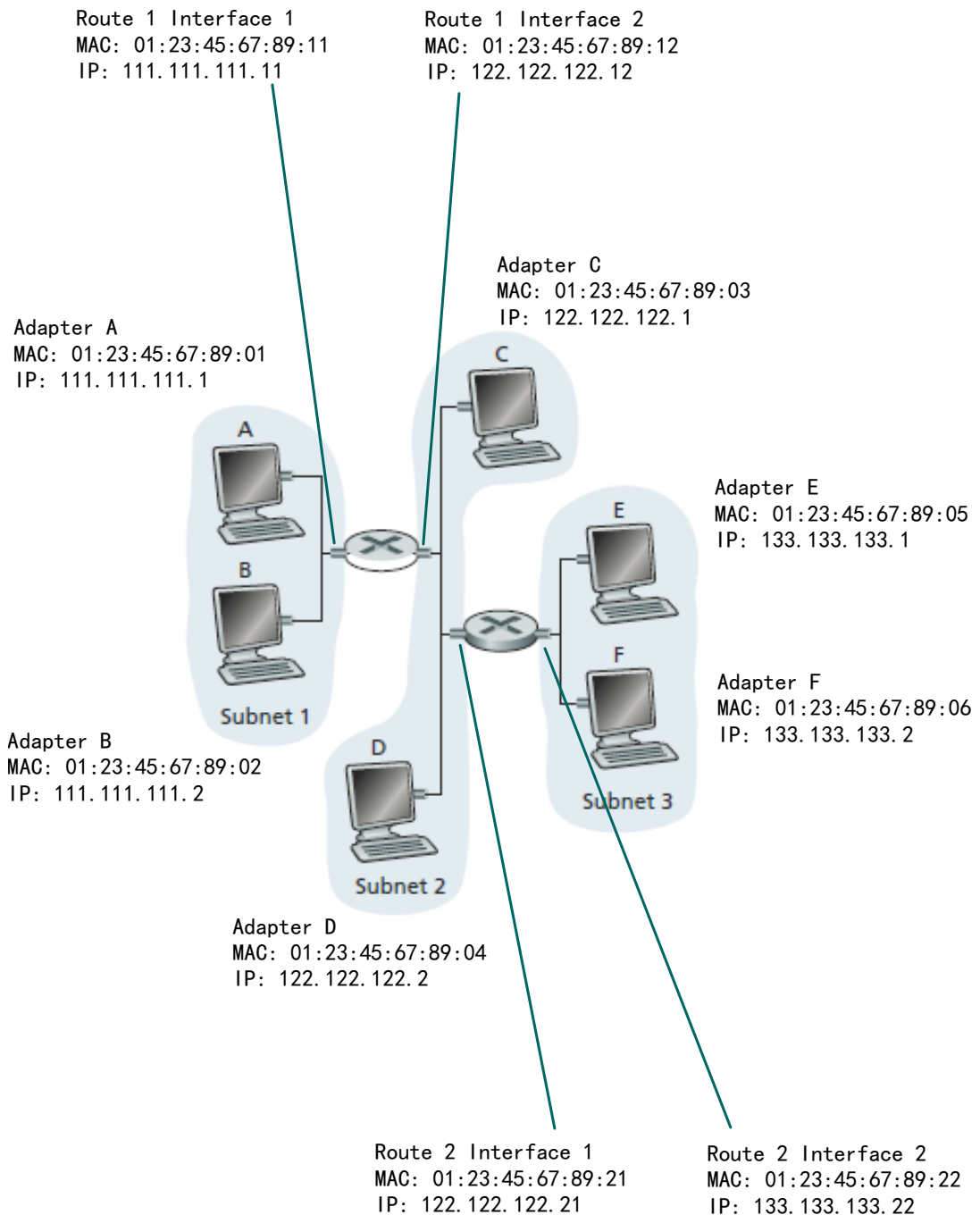
由于在指数退避过程中 B 选择了 $K = 1$ ，因此在 512 个比特时间后，也就是 $t = 273 + 512 = 785$ 时刻，B 进入侦听状态，此时信道空闲，但是在 $t = 819 = 785 + 34$ 时刻，B 开始接收 A 发送的帧，此时信道开始忙， $34 < 96$ ，也就是在 96 个比特时间内信道开始忙了，所以此时 B 不会发送帧。

因此，A 和 B 的重发帧不会冲突。

P19. Consider three LANs interconnected by two routers, as shown in the figure below.



- Redraw the diagram to include adapters.
- Assign IP addresses to all of the interfaces. For Subnet 1 use addresses of the form 111.111.111.xxx; for Subnet 2 uses addresses of the form 122.122.122.xxx; and for Subnet 3 use addresses of the form 133.133.133.xxx.
- Assign MAC addresses to all of the adapters.



- d. Consider sending an IP datagram from Host A to Host F. Suppose all of the ARP tables are up to date. Enumerate all the steps, as done for the single-router example in Section 5.4.2.

第一跳，主机 A 已知主机 F 的 IP 地址 (133.133.133.2)，于是封装了一个源 IP 为 111.111.111.1，目的 IP 为 133.133.133.2 的 IP 包，之后 A 发现目的 IP 和本机 IP 不在一个子网，于是 A 从自己的 ARP 表中查找网关 (路由器) 的 IP (111.111.111.11) 对应的 MAC (01:23:45:67:89:11)，将目的 MAC 为 01:23:45:67:89:11，源 MAC 为 01:23:45:67:89:01 放到数据链路层首部，形成帧并发送出去。

第二跳，路由器 1 的接口 1 收到该帧，比较该帧首部的 MAC 和自己的 MAC，确认是发送给自己的后，查看该帧内的 IP 层内容，发现目的 IP (133.133.133.2) 和接口 1、接口 2 都不在一个子网，于是查看自己的路由表，发现该子网的数据包应该发送给路由器 2 的接口 1 的 IP (122.122.122.21)，并且应该从接口 2 发出，之后查看自己的 ARP 表，找到此 IP 对应的 MAC (01:23:45:67:89:21)，将目的 MAC 为 01:23:45:67:89:21，源 MAC 为 01:23:45:67:89:12 (接口 2) 放到数据链路层首部，形成帧从接口 2 发出。

第三跳，路由器 2 的接口 1 收到帧，比较该帧首部的 MAC 和自己的 MAC，确认是发送给自己的后，查看该帧内的 IP 层内容，发现目的 IP (133.133.133.2) 和接口 2 在同一个子网，于是查看自己的 ARP 表，找到此 IP 对应的 MAC (01:23:45:67:89:06)，将目的 MAC 为 01:23:45:67:89:06，源 MAC 为 01:23:45:67:89:22 (接口 2) 放到数据链路层首部，形成帧从接口 2 发出。

最后，主机 F 收到帧，比较该帧首部的 MAC 和自己的 MAC，确认是发送给自己的后，查看该帧内的 IP 层内容，比较目的 IP 和自己的 IP，确认是发送给自己的后，将 IP 层的数据解包后交给上层处理。

- e. Repeat (d), now assuming that the ARP table in the sending host is empty (and the other tables are up to date).

首先，仍然是主机 A 已知主机 F 的 IP 地址 (133.133.133.2)，于是封装了一个源 IP 为 111.111.111.1，目的 IP 为 133.133.133.2 的 IP 包，之后 A 发现目的 IP 和本机 IP 不在一个子网，于是查找网关 (路由器的 IP) (已经配置为 111.111.111.11) 对应的 MAC，但 ARP 表为空，无法找到此 IP 对应的 MAC。于是 A 发送 ARP 查询广播，将目的 MAC 设为广播地址 (全 1)，源 MAC 为 A 的 MAC (01:23:45:67:89:01)，源 IP 为 A 的 IP (111.111.111.1)，目的 IP 设为 111.111.111.11。路由器 1 的接口 1 收到该帧，该帧为广播帧，因此不需要和自己的 MAC 比较，直接查看该帧封装的数据部分，发现是 ARP 查询报文，并且目的 IP 和自己的 IP 相同，于是准备发送 ARP 响应报文。路由器 1 在自己的 ARP 表中查询该报文中的源 (111.111.111.1) IP 对应的 MAC，查找到为 01:23:45:67:89:01，于是将目的 MAC 设为 01:23:45:67:89:01，源 MAC 为路由器 1 接口 1 的 MAC (01:23:45:67:89:11)，源 IP 为路由器 1 接口 1 的 IP (111.111.111.11)，目的 IP 为 111.111.111.1，将该帧从接口 1 发送出去。

A 收到该帧，比较该帧首部的 MAC 和自己的 MAC，确认是发送给自己的后，查看该帧内的 IP 层内容，比较目的 IP 和自己的 IP，确认是发送给自己的后，发现 IP 层内的数据为 ARP 响应报文，并且此响应报文中的源 IP (111.111.111.11) 正好是自己刚才查询的 IP，于是将此帧的源 IP 和源 MAC 放入自己的 ARP 表中，于是 A 就知道网关 (路由器的 IP) 对应的 MAC 为 01:23:45:67:89:11 了，之后的步骤就和上一小题相同了。