*版本：1.0*

# 《计算机网络》课程习题集

《计算机网络》课程组

华中科技大学

电信学院

2020.11

# 目　录

# 1. 概述

## 1.1. 基本概念

1. (**网络构成要素**)列举计算机网络的构成元素，分别举一个例子，并说明这些构成要素之间的关联。

答：(对于每一类构成元素，举一个正确的例子即可，可能不在下列参考答案之列)
计算机网络的构成元素包括(1)主机、(2)交换节点、(3)链路、(4)网络应用、(5)协议，对应的例子包括(1)台式机/服务器/智能手机等、(2)交换机/路由器等、(3)双绞线/光纤/无线链路等、(4)浏览器/QQ/微信等、(5)IP/TCP/HTTP 等。

2. (**网络连通性**)说明通过直接连通实现网络连通性的局限。

答：直接连通链路有点对点、多路接入两种。(1) 采用点对点链路实现 $N$ 台主机的网络连通，所需链路条数与 $N^2$ 成正比，开销随着网络规模的增大急剧增加，而且很难用于地理分布较广的网络，没有可扩展性。(2) 采用多路接入链路实现网络连通，发送冲突的可能性随着主机数量的增加而增大，因此浪费更多的带宽资源，同时多路接入(MAC)算法的效果和效率也受限于主机之间的距离，难以有效解决空间分布较广的主机竞争使用共享信道的冲突。

3. (**网络资源共享**, Ex 1.29, [PD12])假设共享介质 M 以循环方式向主机 $A_1$、$A_2$、…、$A_N$ 提供传输一个分组的机会，没有分组要传的主机立即放弃 M。它与 STDM 有何不同？与 STDM 相比，这种方式对网络的利用率如何？

答：In STDM the offered timeslices are always the same length, and are wasted if they are unused by the assigned station. The round-robin access mechanism would generally give each station only as much time as it needed to transmit, or none if the station had nothing to send, and so network utilization would be expected to be much higher.

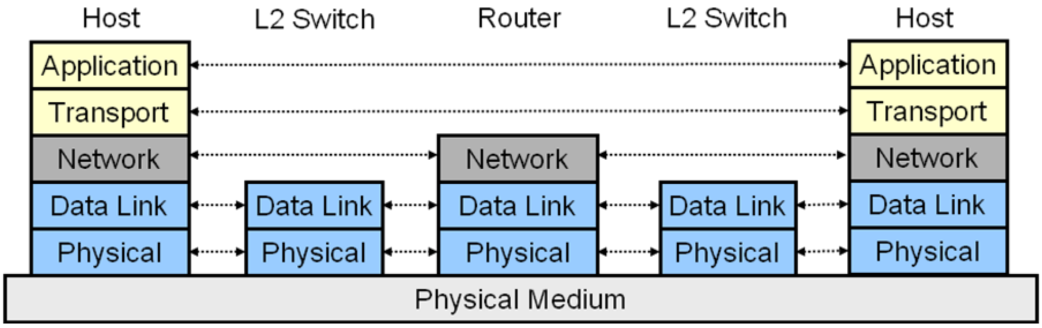
4. (**编址**，Ex. 1.7, [PD12]) 邮政地址与网络编址有何相似和不同之处？电话号码与网络编址又有何相似之处？

<u>Answer</u>: Postal addresses are strongly hierarchical (with a geographical hierarchy, which network addressing may or may not use). Addresses also provide embedded “routing information”. Unlike typical network addresses, postal addresses are long and of variable length and contain a certain amount of redundant information. This last attribute makes them more tolerant of minor errors and inconsistencies. Telephone numbers, at least those assigned to landlines, are more similar to network addresses: they are (geographically) hierarchical, fixed-length, administratively assigned, and in more-or-less one-to-one correspondence with nodes.

## 1.2. 网络架构

5. (**五层网络体系架构**) 对于合并 OSI 参考模型和 Internet 架构得到的五层网络体系架构，图解并说明每一层的功能。

Answer: It is shown below that there are 5 layers in the layered network architecture, as well as some sample nodes on a network.




The functionalities of 5 layers are respectively explained as follows:

- Physical: bit streaming — deal with the transmission of raw bits over a physical link
- Data link: framing — collect a stream of bits into a bit aggregate called frame
- Network: host-to-host communication — handle packet forwarding along the nodes within a network
- Transport: process-to-process communication — implement the communication between two processes running on two hosts
- Application: application-specific services — concern various types of application services provided to end users, e.g., ftp, telnet

## 1.3. 网络性能

6. (**文件传输时延与 RTT**, Ex. 1.3 rev., [PD12]) 设 RTT 为 50ms、分组长度为 1KB、数据发送前需要时长为 2RTT 的握手过程，计算下列情况下传输 1000KB 长度文件所需时间：
   (a) 速率为 2 Mbps，分组连续发送；
   (b) 速率为 2 Mbps，但发送完一个分组后需要等待一个 RTT 才能发送下一个分组；
   (c) 速率无限高，但每个 RTT 只能发送 20 个分组；
   (d) 速率无限高，但首个 RTT 只能发送 1 个分组，第二个 RTT 可以发送 2 个分组，第三个 RTT 可以发送 4 个分组，依此类推。

Answer: We will count the transfer as completed when the last data bit arrives at its destination. An alternative interpretation would be to count until the last ACK arrives back at the sender, in which case the time would be half an RTT (25ms) longer.

**(a) 2 initial RTT's (100ms) + 1000KB/2Mbps (transmit) + RTT/2 (propagation = 25ms)**
**≈ 0.125 + 8Mbit/2Mbps = 0.125 + 4 sec = 4.125 sec.**
If we pay more careful attention to **when a mega is $10^6$ versus $2^{20}$, we get 8,192,000 bits /2,000,000bps = 4.096 sec, for a total delay of 4.221 sec.**

(b) To the above we **add the time for 999 RTTs** (the number of RTTs between when packet 1 arrives and packet 1000 arrives), for a total of **4.221 + 49.95 = 54.171 sec.**

(c) This is **49.5 RTTs**, plus t**he initial 2**, for **2.575 seconds.**

(d) Right after the handshaking is done we send one packet. One RTT after the handshaking we send two packets. At $n$ RTTs past the initial handshaking we have sent $1 + 2 + 4 + ... + 2^n = 2^{n+1}-1$ packets. At $n$ = 9 we have thus been able to send all 1,000 packets; the last batch arrives 0.5 RTT later. Total time is **2+9.5 RTTs, or 0.575 sec.**

7. (**文件传输时延**)设长度为 F 的文件从源主机经过 M 台分组交换机传输至另一主机，分组长度固定为 L，封装开销可以忽略，每台交换机中分组连续转发，节点处理和排队时延均为零，每条链路速率、传播时延分别为 C、p。给出下列情况下的文件传输时延：(a) 交换机均为存储转发模式；(b) 交换机均为直通式，只需收到分组前 N 比特即可转发。

**Answer**: Considering (1) file transmission time at source, (2) propagation delays over $M$+1 links, and (3) packet transmission delays at $M$ switches, end-to-end file transmission latency can be achieved by:

(a) $D = F/C + (M+1)p + ML/C$

(b) $D = F/C + (M+1)p + MN/C$

8. (**电路交换与分组交换时延比较**)比较在一个电路交换网和在一个(负载轻的)分组交换网上将 x(bit)报文沿 k 个跳段传输的通路传输的延迟.假定电路建立时间是 s,每跨段上的传输延迟为 d,分组大小为 p(bit),数据传输速率是 b(b/s).在什么情况下,分组交换网的延迟更短?(忽略分组头的开销)

答：对于电路交换，t=s 时电路就会建立起来；t=s+x/b 时报文的最后一位发送完毕；t=s+x/b+kd 时报文到达目的地。而对于分组交换，最后一位在 t=x/b 时发送完毕。为到达最终的目的地，最后 1 个分组必须被中间的路由器重发 k-1 次，每次重发花时间 p/b（一个分组的所有比特都接收齐了，才能开始重发，因此最后 1 位在每个中间结点的停滞时间为最后一个分组的发送时间），所以总的延迟为：

$$x/b+(k-1)p/b+kd$$

为使分组交换比电路交换快，令:

$$x/b+(k-1)p/b+kd < s+x/b+kd$$

得: $s > (k-1)p/b$

当满足此条件时,分组交换网得延迟更短。

# 2. 物理/数据链路层基础

## 2.1 基础

1. (**物理层传输媒体**)简述物理层与传输媒体的关系。

答：传输媒体不属于物理层的内容，物理层只是规定了与传输媒体的接口，对高层用户屏蔽掉下面种类繁多的传输媒体的差异。

## 2.2 检错/纠错

2. (**CRC**, Ex. 2.18, [PD12]) 需传送数据信息为 1100011，采用多项式为 $x^3 + 1$ 的 CRC 检错。
   (a) 采用多项式长除法确定传输的信息。
   (b) 假设传输链路上的噪声使得传输消息最左边的比特翻转。接收方的 CRC 计算结果是什么？接收方如何知道发生了差错？

Answer:

(a) We take the message 11100011, append 000 to it, and divide by 1001 according to the method shown in Section 2.4.3. The remainder is 100; what we transmit is the original message with this remainder appended, or 1110 0011 100.

(b) Inverting the first bit of the transmission gives 0110 0011 100; dividing by 1001 ($x^3 + 1$) gives a remainder of 10; the fact that the remainder is nonzero tells us a bit error occurred.

## 2.3 滑动窗口/ARQ

3. (**滑动窗口与负面确认**, Ex. 2.22, [PD12]) 对于仅使用否定确认帧(NAK)但没有肯定确认帧(ACK)的 ARQ 协议，给出相应的超时方案，并说明 ARQ 中更愿意采用 ACK 而非 NAK 的原因。

Answer: Assume a NAK is sent only when an out-of-order packet arrives. The receiver must now maintain a RESEND NAK timer in case the NAK, or the packed it NAK’ed, is lost.

Unfortunately, if the sender sends a packet and is then idle for a while, and this packet is lost, the receiver has no way of noticing the loss. Either the sender must maintain a timeout anyway, requiring ACKs, or else some zero-data filler packets must be sent during idle times. Both are burdensome.

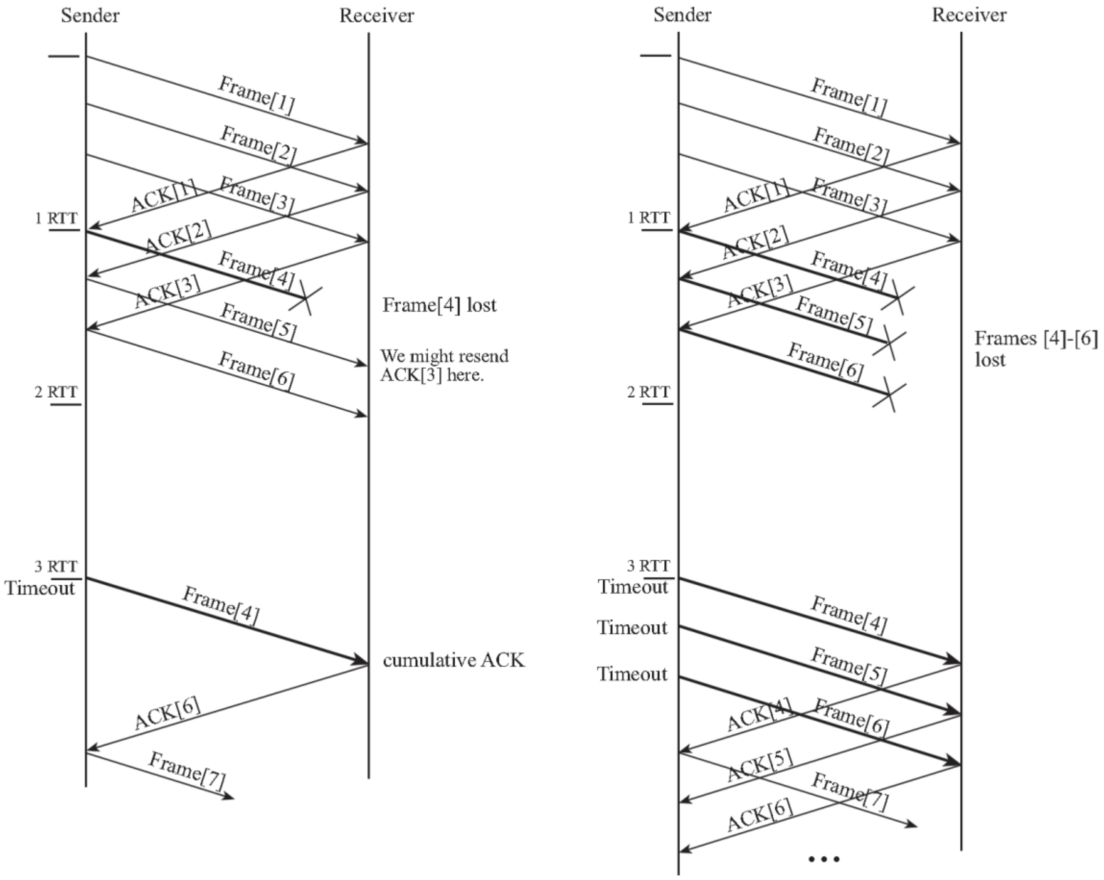
Finally, at the end of the transmission a strict NAK-only strategy would leave the sender unsure about whether any packets got through. A final out-of-order filler packet, however, might solve this.

4. (**滑动窗口序号**, Ex. 2.24 rev., [PD12]) 考虑为连接至月球的一条 2Mbps 点到点链路设计一个滑动窗口协议，其中单程时延是 1.25s，每帧携带 1KB 数据，最少需要多少比特作为序号？

Answer: Bandwidth×(roundtrip)delay is about 2Mbps × 2.5s ≈ 610.35 KB, or 611 packets. The window size should be this large; the sequence number space must cover twice this range, or up to 1222. 11 bits are needed.

5. (**滑动窗口时序过程**, Ex. 2.31, [PD12])对于下列两种情况，绘出 SWS = RWS = 3 的滑动窗口算法时序图，其中超时时间采用 2 × RTT。
   (a) 帧 4 丢失。
   (b) 帧 4 ~ 6 丢失。

Answer: The right diagram, for part (b), shows each of frames 4-6 timing out after a 2×RTT timeout interval; a more realistic implementation (e.g. TCP) would probably revert to SWS=1 after losing packets, to address both congestion control and the lack of ACK clocking.




6. (**滑动窗口大小**, Ex. 2.34, [PD12]) 考虑 SWS = RWS = 3 的滑动窗口算法，没有乱序到达，序号精度无限。
   (a) 说明如果 DATA[6]在接收窗口内，则 DATA[0] (更一般而言，任何更早的数据)不可能到达接收方(因此 MaxSeqNum = 6 足够)。
   (b) 说明如果能够发送 ACK[6] (更准确而言，DATA[5]在发送窗口内)，则不可能收到 ACK[2] (或更早的确认)。

   上述待说明结论等效于 2.5.2 节中公式的一个证明，特别是对于 SWS = 3 的情况。注意问题(b)隐含说明前一问题的情形并不能反过来包含不能区分 ACK[0] 和 ACK[5]的情形。

Answer: We first note that data below the sending window (that is, <LAR) is never sent again, and hence – because out-of-order arrival is disallowed – if DATA[N] arrives at the receiver then nothing at or before DATA[N-3] can arrive later. Similarly, for ACKs, if ACK[N] arrives then

(because ACKs are cumulative) no ACK before ACK[N] can arrive later. As before, we let ACK[N] denote the acknowledgment of all data packets less than N.

(a) If DATA[6] is in the receive window, then the earliest that window can be is DATA[4]-DATA[6]. This in turn implies ACK[4] was sent, and thus that DATA[1]-DATA[3] were received, and thus that DATA[0], by our initial remark, can no longer arrive.

(b) If ACK[6] may be sent, then the lowest the sending window can be is DATA[3]..DATA[5]. This means that ACK[3] must have been received. Once an ACK is received, no smaller ACK can ever be received later.

7. (**停止等待算法信道利用率**)信道的数据传输速率为 4Kbps,传播延迟时间 20ms（未注明往返则指单程），求帧长在什么范围内才使停等协议的信道利用率达到 50%以上，（利用反向信道发送确认信号）

解：$T_1$: 数据帧发送时间 =L/V

$T_2$: 信息上传播延时时间

$T_3$: 确认帧发送时间

$T_2'$: 确认帧传播延迟时间 = $T_2$

用“0”或“1”即1个bit帧长可忽略不计，但确认帧延迟时间 = $T_2$

$$效率 = \frac{t_1}{t_1 + t_2 + t_3 + t_2'} = \frac{\frac{l}{v}}{\frac{l}{V} + 4\times 10^{-2}} = \frac{L}{L+160} \geq 0.5$$

$$l \geq 0.5l + 80$$

$$l \geq 160bit$$

8. (**滑动窗口信道利用率**)在一个 1Mb/s 的卫星信道上发送 1000bit 长的帧。端到端的传输延迟是 270ms，确认总是捎带在数据帧中。帧头很短，使用 3 位序列号。对以下协议而言，可以取得的最大信道利用率是多少？

(a) 停止-等待协议

(b) 连续 ARQ

(c) 选择重传 ARQ

答：对应三种协议的窗口大小值分别是 1，7 和 4。

使用卫星信道端到端的传输延迟是 270ms。

以 1Mb/s 发送，1000bit 长的帧的发送时间是 1ms。

我们用 t = 0 表示传输开始时间，

那么在 t = 1ms 时，第一帧发送完毕。

t = 271ms，第一帧完全到达接收方。

t = 272ms 时，对第一个帧的确认帧发送完毕。

t = 542ms 时带有确认的帧完全到达发送方。

因此周期是 542ms。如果在 542ms 内可以发送 k 个帧，（每个帧发送用 1ms 时间），则信道的利用率是 k/542，因此：

（a）k = 1，最大信道利用率 = 1/542 = 0.18%

（b）k = 7，最大信道利用率 = 7/542 = 1.29%
（c）k = 4，最大信道利用率 = 4/542 = 0.74%

9. (**滑动窗口序号**)在使用连续 ARQ 协议时，采用 7 比特序号编码方式，发送窗口=118，设初始窗口后沿指向序号 24，试问：(1)前沿指向什么序号？(2)发送 80 帧，接收到 60 帧应答后，发送窗口前后沿各指向什么序号？（提示：前沿序号为发送窗口中发送的最后一个帧序号，后沿序号为发送窗口中发送后等待确认的帧序号。）（要求给出计算过程）

答：（1）（24 + 117）MOD 128 = 13
（2）前沿：（24 + 117 + 60）MOD 128 = 73
后沿：24 + 60 = 84

# 3. 局域网

## 3.1 基础

1. (**MAC 分类**)总结你学过的各种多路访问技术，概括其特点。

答：受控接入：轮询、 轮叫轮询、 令牌环、 令牌总线。

随机接入： ALOHA、 时隙 ALOHA、 CSMA、 CSMA/CD。

## 3.2 以太网

2. (**以太网帧长**)假定 1km 长的 CSMA/CD 网络的数据传输率为 1Gb/s。设信号在网络上传播速度为 200000km/s。求能够使用此协议的最短帧长。

解：网络端到端的传播时延为

T= 1km / (200000km/s)=5x10-6 s

CSMA/CD规定最短帧长不能小于2T。

因此最短帧长为：

1Gb/s * 2T = (1x109 b/s ) * ( 2x5x10-6 s )

=10000 (bit)

3. (**CSMA/CD 时序**)一个 CSMA／CD 基带总线网长度为 1000 米，信号传播速度为 200 米／微秒，假如位于总线两端的节点，在发送数据帧时发生了冲突，试问：（1）两节点间的信号传播延迟是多少？（2）最多经过多长时间才能检测到冲突？

答：（1）5us；（2）10us

4. (**Ethernet MAC**, Ex. 2.43, [PD12]) 一个共享式以太网中，A、B 两台主机均有一系列帧准备发送，A 的待发送帧编号依次为 $A_1$、$A_2$、…，B 的帧编号类似。令 $T = 51.2\mu s$ 表示指数回退的基本时间单元。设 A 和 B 同时想发送帧 1，首轮冲突后分别选择回退时间 $0\times T$、$1\times T$，即 A 赢得首轮回退竞争并传输 $A_1$，B 等待。
   (a) 首轮回退竞争传输结束后，B 试图再次发送 B1，而 A 试图发送 A2。第二次冲突后的第二轮回退竞争中，A 的回退时长随机选择 0×T 或 1×T 之一，B 的回退时长则随机选择 0× T, …, 3×T 之一。给出 A 在第二轮回退竞争中获胜的概率。
   (b) 设 A 在第二次回退竞争中获胜，随后 A 发送 A3，B 再次发送 B1，第三次发生冲突。给出 A 赢得第三轮回退竞争的概率。
   (c) 设 A 赢得首轮回退竞争，给出 A 赢得所有后续回退竞争的概率。
   (d) 设 A 赢得后续回退竞争，B 如何处理？

Answer:

(a) A can choose $k_A$=0 or 1; B can choose $k_B$ = 0,1,2,3. A wins outright if ($k_A$, $k_B$) is among (0,1), (0,2), (0,3), (1,2), (1,3); there is a 5/8 chance of this.

(b) Now we have $k_B$ among 0...7. If $k_A$ =0, there are 7 choices for $k_B$ that have A win; if $k_A$ = 1 then there are 6 choices. All told the probability of A's winning outright is 13/16.

(c) **<u>Original</u>**: P(winning race 1) = 5/8 > 1/2 and P(winning race 2) = 13/16 > 3/4; generalizing, we assume the odds of A winning the $i$th race exceed $(1 - 1/2^{i-1})$. We now have that

P(A wins every race given that it wins races 1-3)

$\geq (1 - 1/8)(1 - 1/16)(1 - 1/32)(1 - 1/64)....$

$\approx 3/4$.

**Alternative**: P(winning race 1) = 5/8 = (8-3)/8, and P(winning race 2) = 13/16 = (16-3)/16, … That is, the probability of A winning the $i$th race is exactly $(1 - 3/2^{i+2})$. We now have that

P(A wins every race given that it wins races 1-3) = $\prod_{i=1}^{\infty} (1 - 3/2^{i+2})$

(d) B gives up on it, and starts over with $B_2$.

## 3.3 无线局域网

5. (**802.11 MAC 与隐藏节点问题**, Ex. 2.55, [PD12]) 802.11 网络中，隐藏节点问题如何得到解决？

<u>Answer</u>: 802.11 uses the RTS-CTS mechanism to try to address hidden terminals. A node that has data to send begins by sending a short RTS packet indicating that it would like to send data, and the receiver responds with a CTS, which is also likely to be received by nodes that are in reach of the receiver but hidden from the sender. While this doesn't prevent collisions, the fact that RTS and CTS are short packets makes collisions less likely.

6. (**802.11 MAC**, P6.6, [KR12])参见讲义中关于 CSMA/CA 的介绍(讲义 5-DirectLink(II)第 35 页)，步骤 2.(b)中，一台终端成功地传送一帧后，如还有后续帧传送，返回到步骤 2.(a)，而不是步骤 1，也就是说，即使侦听到信道空闲，也不会立即传送下一帧。说明 CSMA/CA 这一设计的合理之处。

<u>Answer</u>: Suppose that wireless station H1 has 1000 long frames to transmit. (H1 may be an AP that is forwarding an MP3 to some other wireless station.) Suppose initially H1 is the only station that wants to transmit, but that while half-way through transmitting its first frame, H2 wants to transmit a frame. For simplicity, also suppose every station can hear every other station's signal (that is, no hidden terminals). Before transmitting, H2 will sense that the channel is busy, and therefore choose a random backoff value.

Now suppose that after sending its first frame, H1 returns to step 1; that is, it waits a short period of times (DIFS) and then starts to transmit the second frame. H1's second frame will then be transmitted while H2 is stuck in backoff, waiting for an idle channel. Thus, H1 should get to transmit all of its 1000 frames before H2 has a chance to access the channel. On the other hand, if H1 goes to step 2 after transmitting a frame, then it too chooses a random backoff value, thereby giving a fair chance to H2. **Thus, fairness was the rationale behind this design choice.**

# 4. 分组交换

## 4.1 数据报交换

1. (**数据报交换表**, Ex. 3.4, [PD12]) 给出下图所示网络中交换机 S1 ~ S4 的转发表。每台交换机的转发表中应该有一条默认条目，用于将目的地与其它条目都不匹配的分组转发至 OUT，同时去除输出端口与这条默认条目相同的所有条目以避免重复。

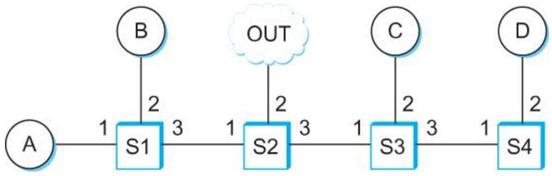



Figure 3.46

Answer: The forwarding tables are shown as follows.

| Switch | Destination | Port |
|---|---|---|
| S1 | A | 1 |
| | B | 2 |
| | default | 3 |
| S2 | A | 1 |
| | B | 1 |
| | C | 3 |
| | D | 3 |
| | default | 2 |
| S3 | C | 2 |
| | D | 3 |
| | default | 1 |
| S4 | D | 2 |
| | default | 1 |

## 4.2 虚电路交换

2. (**虚电路表征**, Ex. 3.5, [PD12]) 考虑下图所示虚电路交换网络，其后的表格列出了每个交换机的虚电路表。所有虚电路均为双向连接，为了简化形式，虚电路表中每条条目中左边和右边的<Port, VCI>对互为一条虚电路两个方向连接的输入/输出端口、VCI 组合。列出所有的端到端虚电路。

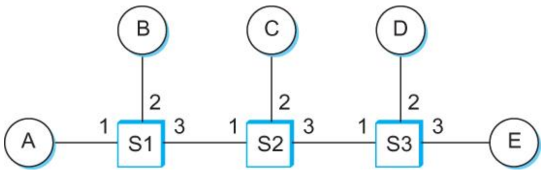


Figure 3.47

| Switch | Port | VCI | Port | VCI |
|---|---|---|---|---|
| S1 | 1 | 2 | 3 | 1 |
| | 1 | 1 | 2 | 3 |
| | 2 | 1 | 3 | 2 |
| S2 | 1 | 1 | 3 | 3 |
| | 1 | 2 | 3 | 2 |
| S3 | 1 | 3 | 2 | 1 |
| | 1 | 2 | 2 | 2 |

Table 3.15 VCI tables for Switches in Figure 3.47

Answer: In the following, Si[j] represents the jth entry (counting from 1 at the top) for switch Si.

A connects to D via S1[1]—S2[1]—S3[1]

A connects to B via S1[2]

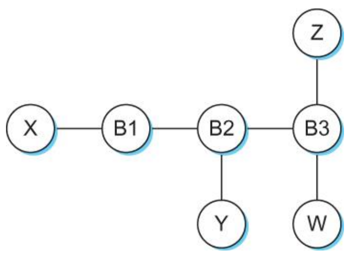
B connects to D via S1[3]—S2[2]—S3[2]

## 4.3 网桥与自学习

3. (**自学习网桥**)说明透明网桥工作原理，这种网桥在实际工作中会遇到什么问题。

答：透明网桥连接多个局域网，在局域网之间转发数据帧，以扩展局域网的覆盖范围。透明网桥转发数据帧，是根据数据帧目的地址，查找其转发表，选择相应端口输出；如转发表中无对应条目，则向除输入端口之外的所有其它端口转发(类似于广播)。网桥刚启动时，转发表为空；随后的过程中，采用自学习的方法累积构建转发条目，即每收到一个数据帧，根据其源地址和输入端口，构建一条以该地址为目的地址、以该端口为输出端口的转发条目。

4. (**自学习**, Ex. 3.17, [PD12]) 考虑下图中主机 X、Y、Z、W 和学习型网桥 B1、B2、B3，其中转发表初始化为空。
   (a) 设 X 发送到 W。哪个网桥学习到 X 的位置？Y 的网络接口能收到这个分组吗？
   (b) 设 Z 接着发送到 X。哪个网桥学习到 Z 的位置？Y 的网络接口能收到这个分组吗？
   (c) 设 Y 然后发送到 X。哪个网桥学习到 Y 的位置？Z 的网络接口能收到这个分组吗？
   (d) 设最后 W 发送到 Y。哪个网桥学习到 Z 的位置？W 的网络接口能收到这个分组吗？




Answer:

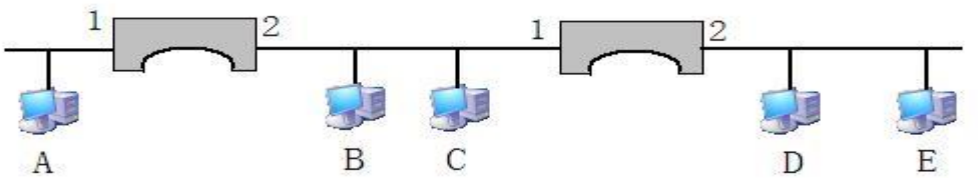
(a) When X sends to W the packet is forwarded on all links; all bridges learn where X is. Y's network interface would see this packet.

(b) When Z sends to X, all bridges already know where X is, so each bridge forwards the packet only on the link towards X, that is, B3→B2→B1→X. Since the packet traverses all bridges, all bridges learn where Z is. Y's network interface would not see the packet as B2 would only forward it on the B1 link.

(c) When Y sends to X, B2 would forward the packet to B1, which in turn forwards it to X. Bridges B2 and B1 thus learn where Y is. B3 and Z never see the packet.

(d) When W sends to Y, B3 does not know where Y is, and so retransmits on all links; Z's network interface would thus see the packet. When the packet arrives at B2, though, it is retransmitted only to Y (and not to B1) as B2 does know where Y is from step (c). B3 and B2 now know where W is, but B1 does not learn where W is.

5. (**自学习**)如图所示的网络中，设两个网桥的转发表都是空的。若各站向其他的站发送了数据帧的顺序是：A 发送给 E，C 发送给 B，D 发送给 C，B 发送给 A。试根据网桥工作原理填写在表中内容。

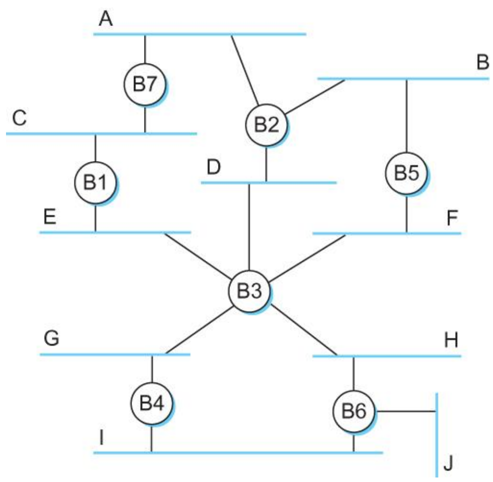



| 发送的帧 | 左网桥的转发表 | | 右网桥的转发表 | | 左网桥的处理 |
|---|---|---|---|---|---|
| | 地址 | 接口 | 地址 | 接口 | 转发？丢弃？登记？ |
| A→E | | | | | |
| C→B | | | | | |
| D→C | | | | | |
| B→A | | | | | |

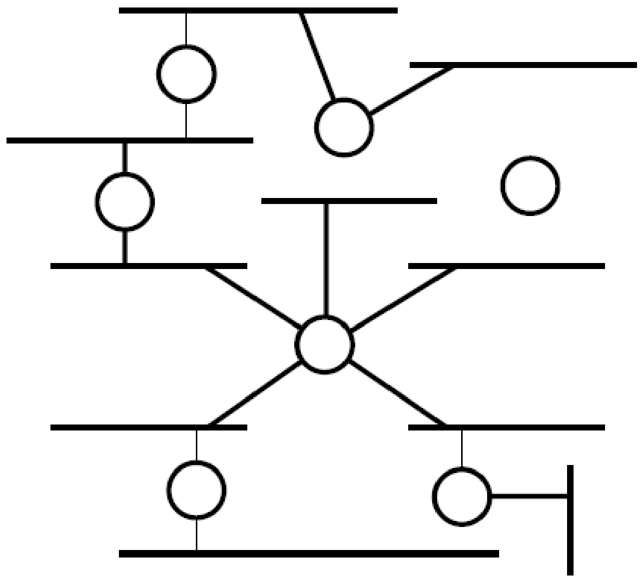
答：

| 发送的帧 | 左网桥的转发表 | | 右网桥的转发表 | | 左网桥的处理 |
|---|---|---|---|---|---|
| | 地址 | 接口 | 地址 | 接口 | 转发？丢弃？登记？ |
| A→E | A | 1 | A | A | 转发 写入转发表 |
| C→B | C | 2 | C | 1 | 转发 写入转发表 |
| D→C | D | 2 | D | 2 | 不转发 写入转发表 |
| B→A | B | 2 | B | 1 | 转发 写入转发表 |

6. (**生成树**, Ex. 3.13, [PD12])对于下图所示的网桥扩展局域网，图解运行生成树算法后哪些网桥端口被禁用。




Answer: The ports not selected are shown in the following figure.



7. (**网桥连接**)一个大学计算机系由 3 个以太子网用两个透明网桥连成一个线形网络。有一天，管理员有事临时由他人代替，这个新的管理员注意到网络两端没连，便用一个新的透明网桥进行连接组成一个环。请问会发生什么情况？

答：将不会发生什么特殊情况。新的桥接器在网上宣告自己的存在，生成树算法为新的配置计算一个生成树。新的拓扑会把其中的一个桥接器设置成备用方式，它将在其他桥接器失效的情况下投入工作。这种类型的配置以附加的代价提供附加的可靠性，但并非不正常。它不会引起任何的问题，因为无论你连接多少个桥接器，结果总是以生成树的形式运行网络。

## 4.4 局域网交换机

8. (**共享式、交换式以太网的吞吐量**) There are one server and *N* clients on an Ethernet network, which are connected via a 100-Mbps Ethernet hub. Suppose this Ethernet hub is replaced by 100-Mbps Ethernet switch. How does the share of link capacity between a client and the server change?

<u>Answer:</u> In the case of 100 Mbps Ethernet Hub, the share of link capacity between a client and the server is 100 / (N+1) Mbps; in the case of 100 Mbps Ethernet switch, this becomes 100 / N Mbps.

Thus, the gap is 100 / (N*(N+1)) Mbps.

9. (**以太网交换机组合的吞吐量**, [KR12])以太网交换机是一种能够并行转发多个分组(以太网帧)的网桥，为了方便起见，每个网络接口的输入和输出端口视为不同接口。设有两台 N×N 端口以太网交换机，N 很大，每台交换机都能够并行转发 3 个分组；两台交换机通过一对端口互相连接，成为一台组合交换机；连接两台成员交换机的链路每次只能处理一个分组，因而成为整个组合交换机系统的瓶颈。
   (a) 考虑只有两个分组流经过该组合交换机传输，其输入输出端口未知，可能是交换机任一端口。这两个分组流能够被该组合交换机并行转发的概率是多少？
   (b) 三个分组流又如何？

答：为了方便起见，分别用 A、B 表示组合的两台交换机。分组流的流向有 A 进 A 出、A 进 B 出、B 进 A 出、B 进 B 出四种可能，分别用(A, A)、(A, B)、(B, A)、(B, B)表示。连接 A 和 B 的两条单向链路每次只能处理一个分组，因此成为组合交换机的瓶颈：如有两个以上的分组流同为(A, B)或(B, A)，则不能并行转发。

(a) 两个输入输出端口未知的分组流，其流向共有 4x4=16 种组合，只有两个分组流同为(A, B)或(B, A)时，才不能被组合交换机并行转发。因此两个分组流可被并行转发的概率为：

$$1 - 2/16 = 7/8$$

(b) 三个输入输出端口未知的分组流，其流向共有 4x4x4=64 种组合。

首先考虑有且仅有两个分组流同为(A, B)或(B, A)的组合数目：两个分组流的相同流向取(A, B)、(B, A)两者之一，即可选项数为 2；两个分组流的相同流向一旦确定，另外一个分组流流向只能 3 选 1，即可选项数为 3；从三个分组流中挑出两个流向相同，可能的组合数为 $C_3^2$；因此总的组合数目为 $C_3^2 \times 3 \times 2 = 18$。

其次，三个分组流同为(A, B)或(B, A)的组合数目显然为 2。

因此，少于两个分组流为(A, B)或(B, A)即三个分组流能够被并行转发的概率为：

$$1 - (18+2)/64 = 11/16$$

# 5. 网络层

## 5.1 基础

1. (**IP 数据报分片**)一个数据报长度为 4000 字节（首部长度固定为 20 字节）。现在经过一个网络传送，但此网络能够传送的最大数据长度为 1500 字节。试问应当划分为几个短些的数据报片？根据各数据报片的数据字段长度、片偏移字段和 MF 标志的取值填写下表。

答：应当划分为 3 个短些的数据报片（2 分）

| | 总长度(字节) | 数据长度(字节) | MF | 片偏移 |
|---|---|---|---|---|
| 原始数据报 | 4000 | 3980 | 0 | 0 |
| 数据报片 1 | 1500 | 1480 | 1 | 0 |
| 数据报片 2 | 1500 | 1480 | 1 | 185 |
| 数据报片 3 | 1040 | 1020 | 0 | 370 |

2. (**IP 数据报分片**, Ex. 3.38, [PD12]) 设一个 IP 数据包分为 10 个分段，每个分段传输相互独立，丢失概率为 1%。合理地近似一下，据此可认为由于单个分段丢失而导致整个数据包丢弃(重组失败)的可能性为 10%。如果分段的数据包传输两个批次，整个数据包净丢失的概率是多少？
   (a) 假定重组时用到的所有分段必须是来自于同一批次传输。
   (b) 假定重组时用到的任一分段可来自于两个批次传输中的任何一次。
   (c) 解释 Ident 字段如何应用于此。

<u>Answer</u>:

(a) The probability of losing both transmissions of the packet would be $0.1 \times 0.1 = 0.01$.

(b) The probability of loss is now the probability that for some pair of identical fragments, both are lost. For any particular fragment the probability of losing both instances is $0.01 \times 0.01 = 10^{-4}$, and the probability that this happens at least once for the 10 different fragments is thus about 10 times this, or 0.001.

(c) An implementation might (though generally most do not) use the same value for **Ident** when a packet had to be retransmitted. If the retransmission timeout was less than the reassembly timeout, this might mean that case (b) applied and that a received packet might contain fragments from each transmission.

3. (**子网划分**)某机构下属 A、B、C、D、E 五个部门，分别有 157、108、51、24、16 台主机。现有 IP 地址块 192.12.6.0/23，需要将其划分用于五个部门的子网编址。给出一种满足上述要求的子网地址划分方案。

<u>Answer</u>: Two possible arrangements are shown as below.

| Subnet | #Host | #Bit | Net. Num.: Bytes 3 & 4 | Net Num | Network Mask |
|---|---|---|---|---|---|
| A | 157 | 8 | 0000 011<u>**0**</u>.0000 0000 | 192.12.6.0/24 | 255.255.255.<u>0</u> |
| B | 108 | 7 | 0000 011<u>**1.0**</u>000 0000 | 192.12.7.0/25 | 255.255.255.<u>128</u> |

| | | | | | |
|---|---|---|---|---|---|
| C | 51 | 6 | 0000 011**1.10**00 0000 | 192.12.7.128/26 | 255.255.255.192 |
| D | 24 | 5 | 0000 011**1.110**0 0000 | 192.12.7.192/27 | 255.255.255.224 |
| E | 16 | 5 | 0000 011**1.111**0 0000 | 192.12.7.224/27 | 255.255.255.224 |

or

| Subnet | #Host | #Bit | Net. Num.: Bytes 3 & 4 | Net Num | Network Mask |
|---|---|---|---|---|---|
| A | 157 | 8 | 0000 011**1**.0000 0000 | 192.12.7.0/24 | 255.255.255.0 |
| B | 108 | 7 | 0000 011**0.1**000 0000 | 192.12.6.128/25 | 255.255.255.128 |
| C | 51 | 6 | 0000 011**0.01**00 0000 | 192.12.6.64/26 | 255.255.255.192 |
| D | 24 | 5 | 0000 011**0.001**0 0000 | 192.12.6.32/27 | 255.255.255.224 |
| E | 16 | 5 | 0000 011**0.000**0 0000 | 192.12.6.0/27 | 255.255.255.224 |

4. (**子网划分**) 某机构拥有 C 类 IP 地址块 212.1.1/24，需要将其分为四个子网，分别用于 A、B、C、D 四个部门，相应的主机数分别为 75、35、20、18。
   (a) 给出一种子网划分方案。
   (b) 如部门 C 的主机数量增至 32，给出调整的子网划分方案。

答：

(a) 子网号及掩码如下：

| Subnet | #Host | #Bit | Byte 4 (binary) subnetted | Net. Num. | Network Mask |
|---|---|---|---|---|---|
| A | 75 | 7 | **0**0000000 | 212.1.1.0/25 | 255.255.255.128 |
| B | 35 | 6 | **10**000000 | 212.1.1.128/26 | 255.255.255.192 |
| C | 20 | 5 | **110**00000 | 212.1.1.192/27 | 255.255.255.224 |
| D | 18 | 5 | **111**00000 | 212.1.1.224/27 | 255.255.255.224 |

或

| Subnet | #Host | #Bit | Byte 4 (binary) subnetted | Net. Num. | Network Mask |
|---|---|---|---|---|---|
| A | 75 | 7 | **1**0000000 | 212.1.1.128/25 | 255.255.255.128 |
| B | 35 | 6 | **01**000000 | 212.1.1.64/26 | 255.255.255.192 |
| C | 20 | 5 | **001**00000 | 212.1.1.32/27 | 255.255.255.224 |
| D | 18 | 5 | **000**00000 | 212.1.1.0/27 | 255.255.255.224 |

(b) 鉴于子网大小的颗粒度限制，可将部门 A 的网络拆分成两个子网，一种划分方案如下：

| Subnet | #Host | #Bit | Byte 4 (binary) subnetted | Net. Num. | Network Mask |
|---|---|---|---|---|---|
| A1 | 50 | 6 | **00**000000 | 212.1.1.0/26 | 255.255.255.192 |
| A2 | 25 | 5 | **110**00000 | 212.1.1.192/27 | 255.255.255.224 |
| B | 35 | 6 | **01**000000 | 212.1.1.64/26 | 255.255.255.192 |
| C | 32 | 6 | **10**000000 | 212.1.1.128/26 | 255.255.255.192 |
| D | 18 | 5 | **111**00000 | 212.1.1.224/27 | 255.255.255.224 |

5. (**IP 与 MAC 地址**)网络中存在几种标识主机的方法，分别是什么？它们之间存在什么的区别与联系。

答：网络中主机的标识方式主要包括 IP 地址和 MAC 地址。

其中 IP 地址是主机的逻辑地址，由网络管理员进行配置管理；MAC 地址是主机的物理地址，由网络适配器的生产厂商写入。

IP 地址主要用于主机所在网络的寻址，MAC 地址用于主机的寻址。在整个网络的通信过程中，IP 地址保持不变，MAC 地址随着每一次跨子网的转发而改变。

两者之间通过 ARP 协议实现转换。

6. (**ARP**, Ex. 3.45, [PD12])设 IP 的具体实现中采用下列算法处理目的地 IP 地址为 D 的分组 P 的接收：

```
if (<Ethernet address for D is in ARP cache>)
  <send P>
else
  <send out an ARP query for D>
  <put P into a queue until the response comes back>
```

(a) 当 IP 层收到多个突发的、目的地为 D 的分组时，上述算法如何不必要地浪费资源？
(b) 简述一个改进版本。
(c) 假定当缓存查找失败时，发出一条查询后就简单地丢弃 P。这将导致什么结果？

Answer:

(a) If multiple packets after the first arrive at the IP layer for outbound delivery, but before the first ARP response comes back, then we send out multiple unnecessary ARP packets. Not only do these consume bandwidth, but, because they are broadcast, they interrupt every host and propagate across bridges.
(b) We should maintain a list of currently outstanding ARP queries. Before sending a query, we first check this list. We also might now retransmit queries on the list after a suitable timeout.
(c) This might, among other things, lead to frequent and excessive packet loss at the beginning of new connections.

## 5.2 路由算法

7. (**距离向量路由算法**) (a) 简述 RIP 协议所采用的距离向量路由(Distance Vector Routing)的设计思想，评价其优点和缺点。(b) 给定如下的网络拓扑图，A 到 F 表示网络节点，链路上数字表示该链路的权值，请给出节点 A 经过 1 次、2 次、3 次、4 次路由信息交换后，基于 Bellman-Ford 算法测算的到其它节点的距离向量信息，以及相应的路由表。

答：(a) 距离向量路由的核心思想是：每个节点构造一个包含到所有其他节点距离的一个向量，并将这个向量分发给他的邻节点；邻节点获得信息之后根据 Bellman-Ford 的方程计算出到网络中其他节点的最短路径。简单的说，就是网络中的节点将自己知道的全局拓扑信息告诉自己的邻居。

(b) 节点 A 到其它节点的距离向量

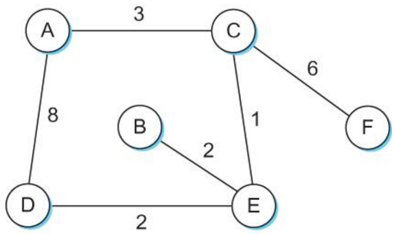
| 交换路由信息次数 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 0 | 0 | ∞ | 3 | 7 | ∞ | ∞ |
| 1 | 0 | ∞ | 3 | 7 | 6 | 5 |
| 2 | 0 | 7 | 3 | 7 | 6 | 5 |
| 3 | 0 | 7 | 3 | 7 | 6 | 5 |
| 4 | 0 | 7 | 3 | 7 | 6 | 5 |

节点 A 的路由表

| 序号 | 目的地址 | 路径权值 | 下一跳 |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | A | 0 | 直达 |
| 2 | B | 7 | C |
| 3 | C | 3 | 直达 |
| 4 | D | 7 | 直达 |
| 5 | E | 6 | C |
| 6 | F | 5 | C |

8. (**距离向量路由算法**, Ex. 3.46, [PD12])对于下图所示网络，给出下列情况下的全局距离向量表(形如教材第 130 页表 3-10)：
   (a) 每个节点只知道自己至直接邻居节点的距离。
   (b) 每个节点将上一步获知的信息告知其邻居节点。
   (c) 步骤(b)再次执行。




Answer:

(a)

| Info. Stored at Node | Distance to Reach Node | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| A | 0 | ∞ | 3 | 8 | ∞ | ∞ |
| B | ∞ | 0 | ∞ | ∞ | 2 | ∞ |
| C | 3 | ∞ | 0 | ∞ | 1 | 6 |
| D | 8 | ∞ | ∞ | 0 | 2 | ∞ |
| E | ∞ | 2 | 1 | 2 | 0 | ∞ |
| F | ∞ | ∞ | 6 | ∞ | ∞ | 0 |

(b)

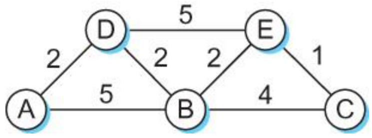
| Info. Stored at Node | Distance to Reach Node | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| A | 0 | ∞ | 3 | 8 | 4 | 9 |
| B | ∞ | 0 | 3 | 4 | 2 | ∞ |
| C | 3 | 3 | 0 | 3 | 1 | 6 |
| D | 8 | 4 | 3 | 0 | 2 | ∞ |
| E | 4 | 2 | 1 | 2 | 0 | 7 |
| F | 9 | ∞ | 6 | ∞ | 7 | 0 |

(c)

| Info. Stored at Node | Distance to Reach Node | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| A | 0 | 6 | 3 | 6 | 4 | 9 |
| B | 6 | 0 | 3 | 4 | 2 | 9 |
| C | 3 | 3 | 0 | 3 | 1 | 6 |

| | | | | | | |
|---|---|---|---|---|---|---|
| D | 6 | 4 | 3 | 0 | 2 | 9 |
| E | 4 | 2 | 1 | 2 | 0 | 7 |
| F | 9 | 9 | 6 | 9 | 7 | 0 |

9. (**链路状态路由算法**, Ex. 3.62, [PD12])对于下图所示网络，给出节点 A 运行前向搜索算法计算其路由表的步骤(形如教材第 137 页表 3-14)。




Answer: The steps are listed in the following table.

| **Step** | **Confirmed** | **Tentative** |
|---|---|---|
| 1 | (A, 0, -) | |
| 2 | (A, 0, -) | (D, 2, D) (B, 5, B) |
| 3 | (A, 0, -) (D, 2, D) | (B, 4, D) (E, 7, D) |
| 4 | (A, 0, -) (D, 2, D) (B, 4, D) | (E, 6, D) (C, 8, D) |
| 5 | (A, 0, -) (D, 2, D) (B, 4, D) (E, 6, D) | (C, 7, D) |
| 6 | (A, 0, -) (D, 2, D) (B, 4, D) (E, 6, D) (C, 7, D) | |

## 5.3 IP 网络路由

10. (**最长前缀匹配**, Ex. 3.72 rev.)根据如下路由器转发表，分别给出下列目的地 IP 地址数据包的下一跳。(注：网络号及地址均为十六进制形式)

(a) C4.5E.13.87; (b) C4.5E.22.09; (c) C3.41.80.02;
(d) 5E.43.91.12; (e) C4.6D.31.2E; (f) C4.6B.31.2E

| 网络号/掩码长度 | 下一跳 |
|---|---|
| C4.50.0.0/12 | A |
| C4.5E.10.0/20 | B |
| C4.60.0.0/12 | C |
| C4.68.0.0/14 | D |
| 80.0.0.0/1 | E |
| 40.0.0.0/2 | F |
| 0.0.0.0/2 | G |

答: (a) B; (b) A; (c) E; (d) F; (e) C; (f) D
(For the last one, note that the first 14 bits of C4.6B and C4.68 match.)

11. (**RIP**)简述 RIP 协议有哪些不足？
答：（1）坏消息传得慢
（2）只能用于小型网络
（3）路由只是根据经过的路由器数量决定最佳
（4）协议交换的是整个路由表，信息量太大

12. (**域间路由**, Ex. 4.11, [PD12])位于一个大型机构网络 A 中的网络 N，除了经由 A 的连接以外，还直接连接至一个 ISP 以连通至互联网。设 R1 为连接 N 至其 ISP 的路由器，R2 为连接 N 到 A 中其余部分的路由器。
    (a) 设 N 为 A 的子网，R1 和 R2 应如何配置？对于 N 使用自己单独的 ISP 连接，仍然存在什么限制？能否防止 A 使用 N 的 ISP 连接？假设可使用像 BGP 这样的机制，说明你的相关配置，包括 R1 和 R2 应通过什么路径通告何种路由信息。
    (b) 现在假设 N 采用自己的网络号，对于问题(a)的答案有何不同？
    (c) 说明当 A 自己的互联网连接失效后，允许其使用 N 的互联网连接的路由器配置。

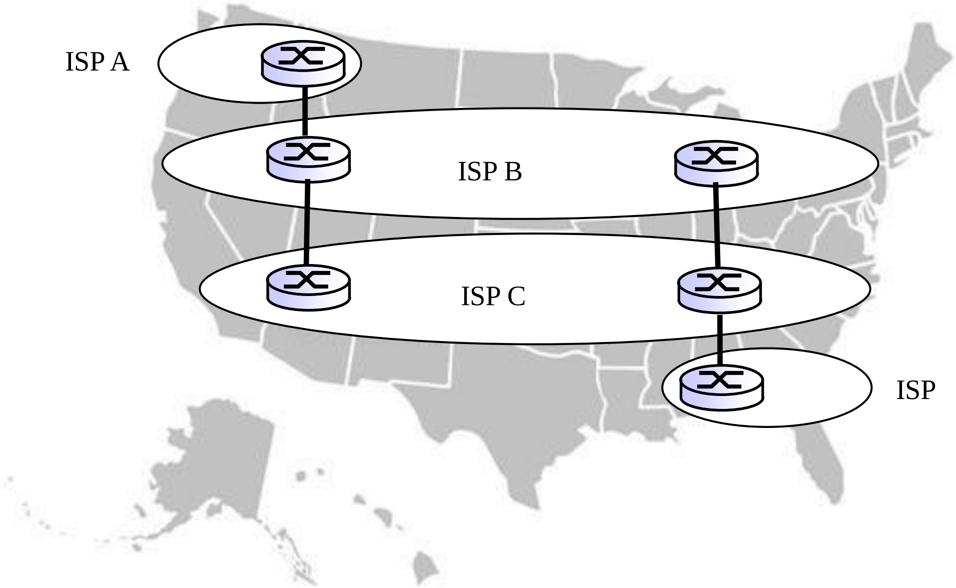
<u>Answer</u>:

(a) R1 should be configured to forward traffic destined for outside of A to the new ISP. R2 should route traffic destined for A to A as before. Note that if a host on N sends its outbound traffic to R2 by mistake, R2 will send it via the old link. R2 should continue to advertise N to the rest of A. N's outbound traffic would then take the new link, but inbound traffic would still travel via A. Subnets are not announced into the backbone routing tables, so R1 would not announce N to the world.

(b) If N has its own IP network number, then R1 does announce its route to N to the world. R1 would not necessarily announce its route to A, however. R2 would not change: it would still announce N into A. Assuming A does not announce its route to N into its provider, all external traffic to and from N now goes through the new link, and N-A traffic goes through R2.

(c) If A wants to use N's R1 link as a backup, then R1 needs to announce to the backbone that it has a route to A, but give this route a cost higher than that of A's original link (techniques for doing this via BGP include route preference indications and "padding" the report with extra ASes.)

13. (**BGP**, P5.16, [KR17]) 下图所示网络中，每个 ISP 网络都是一个独立的 AS，B 和 C 使用 BGP，为对等关系(peering)。B、C 分别为区域级 A、D 提供国内互联网主干服务。考虑 A 到 D 的流量，B 倾向于在西海岸将流量交给 C 传到东海岸(从而使得 C 承载跨越整个国家的流量开销)，而 C 倾向于通过在东海岸与 B 链接的节点接收这些流量(即由 B 承载跨越整个国家的流量)。C 可以采用什么样的 BGP 机制，使得 B 将通过东海岸对等链接传递 A 到 D 的流量？




Answer: One way for C to force B to hand over all of B's traffic to D on the east coast is for C to only advertise its route to D via its east coast peering point with C.

## 5.4 IP 进阶(多播、IPv6、MPLS)

14. (**多播**, Ex. 4.17, [PD12]) 设主机 A 向一个多播组发送消息，接收者是以 A 为根的一棵转发树上的叶子节点，树的深度为 $N$，每个非叶子节点有 $k$ 个子节点，即共有 $k^N$ 个接收者。
    a) 如果 A 向所有接收者发送一条多播消息，需要多少条次链路传输？
    b) 如果 A 向每个接收者发送单播消息，需要多少条次链路传输？
    c) 假设 A 发送的消息中，有一些因为丢失而需要重传。向多少比例接收者的单播重传链路条次等于向所有接收者的一次多播重传链路条次？

Answer:

(a) One multicast transmission involves all $k + k^2 + ... + k^{N-1} = (k^N - k)/(k - 1)$ links.

(b) One unicast retransmission involves N links; sending to everyone would require $N \times k^N$ links.

(c) Unicast transmission to $x$ fraction of the recipients uses $x \times N \times k^N$ links. Equating this to the answer in (a), we get $x = (k^N - k)/((k - 1) \times N \times k^N) \approx 1/(k - 1) \times N$.

15. (**IPv6**, Ex. 4.20, [PD12]) 判断下列 IPv6 地址的表示是否正确，并说明错误原因。
    a) ::0F53:6382:AB00:67DB:BB27:7332
    b) 7803:42F2:::88EC:D4BA:B75D:11CD
    c) ::4BA8:95CC::DB97:4EAB
    d) 74DC::02BA
    e) ::00FF:128.112.92.116

Answer:

(a) Correct
(b) Incorrect (::: is not defined as abbreviating notation)
(c) Incorrect (shorthand can only be used for one set of contiguous 0's)
(d) Correct
(e) Incorrect (an IPv4 address mapped to IPv6 should be preceded by “FFFF”).

16. (**MPLS**, Ex. 4.21, [PD12])MPLS 标签长度通常为 20 比特。解释这一长度在 MPLS 用于基于目的地 IP 地址的分组转发时仍然能够提供足够标签值的原因。

Answer: First, MPLS labels are of link-local scope — this means that the same label can be used on different links to mean different things. This in turn means that the number of labels needed on a link is just the number of forwarding equivalence classes (FECs) that are meaningful on that link. Thus, if each label is used to represent a prefix in the routing table, as described in Section 4.5.1, then up to a million prefixes could be handled with 20 bits.

17. (**移动 IP**, P7.15, [KR17]) 考虑在一个具有外部代理的外部网络中的两个移动节点。在移动 IP 中，这两个移动节点是否可能使用相同的转交地址？解释你的答案。

Answer: Two mobiles could certainly have the same care-of-address in the same visited network. Indeed, if the care-of-address is the address of the foreign agent, then this address would be the same. Once the foreign agent decapsulates the tunneled datagram and determines the address of the mobile, then separate addresses would need to be used to send the datagrams separately to their different destinations (mobiles) within the visited network.

# 6. 传输层

## 6.1 UDP

1. (**UDP**, Ex. 5.2, [PD12])考虑一个基于 UDP 的简单协议，类似于 TFTP (Trivial File Transport Protocol)，用于请求文件。客户机发送初始的文件请求，服务器回复首个数据分组，客户机和服务器运行停止等待机制继续通信。
   (a) 描述一种场景，其中客户机请求一个文件，但得到的却是另外一个文件。提示：允许客户机应用在意外退出后重启时，采用相同的 UDP 端口。
   (b) 提出一种协议改进方案，使得上述情况不太可能发生。

Answer:

(a) In the following, the client receives file “foo” when it thinks it has requested “bar”.
1. The client sends a request for file “foo”, and immediately aborts locally. The request, however, arrives at the server.
2. The client sends a new request, for file “bar”. It is lost.
3. The server responds with first data packet of “foo”, answering the only request it has actually seen.

(b) Requiring the client to use a new port number for each separate request would solve the problem. To do this, however, the client would have to trust the underlying operating system to assign a new port number each time a new socket was opened. Having the client attach a timestamp or random number to the file request, to be echoed back in each data packet from the server, would be another approach fully under the application’s control.

## 6.2 TCP

2. (**TCP 报文段序号**, Ex. 5.9, [PD12])假设需要设计一种采用滑动窗口的可靠字节流传输协议(类似于 TCP)，运行于链路速率为 1Gbps 的网络，RTT 为 100ms，报文段的最长生存期为 30 秒。
   (a) 协议首部的 AdverisedWindow 和 SequenceNum 字段应该包含多少比特？
   (b) 上述数值是如何确定的？其中哪些值可能并非那么肯定？

Answer:

(a) The advertised window should be large enough to keep the pipe full; delay (RTT) × bandwidth here is 100ms × 1Gbps = 100Mb = 12.5 MB of data. This requires 24 bits if we assume the window is measured in bytes ($2^{24}$ ≈= 16million) for the AdvertisedWindow field. The sequence number field must not wrap around in the maximum segment lifetime. In 30 seconds, 30 Gb = 3.75 GB can be transmitted. 32 bits allows a sequence space of about 4GB, and so will not wrap in 30 seconds. (If the maximum segment lifetime were not an issue, the sequence number field would still need to be large enough to support twice the maximum window size; see “Finite Sequence Numbers and Sliding Window” in Section 2.5.)

(b) The bandwidth is straightforward from the hardware; the RTT is also a precise measurement but will be affected by any future change in the size of the network. The MSL is perhaps the least certain value, depending as it does on such things as the size and complexity of the network, and on how long it takes routing loops to be resolved.

3. (**TCP 时序**, Ex. 5.39, [PD12])当 TCP 发出<**SYN**, **SeqNum** = *x*>或<**FIN**, **SeqNum** = *x*>报文段时，随后的 **ACK** 中 **Acknowledgment** = *x*+1，也就是说，**SYN** 和 **FIN** 报文段在序号空间中均占用一个单位。这样做是否有必要？如有必要，给出一个当 **Acknowledgment** 为 *x* (而非 *x* + 1)时会导致混乱的例子；如无必要，解释原因。

Answer: Incrementing the **ACK** number for a **FIN** is essential, so that the sender of the **FIN** can determine that the **FIN** was received and not just the preceding data.

For a **SYN**, any **ACK** of subsequent data would increment the acknowledgment number, and any such **ACK** would implicitly acknowledge the **SYN** as well (data cannot be ACKed until the connection is established). Thus, the incrementing of the sequence number here is a matter of convention and consistency rather than design necessity.

## 6.3 TCP 拥塞控制

4. (**TCP 慢启动**, Ex.6.16, [PD12])设有一种 TCP 的扩展版本，允许窗口大小远大于 64KB。这一 TCP 扩展版本运行于一条时延为 50ms、速率为 1Gbps 的链路，以传输一个 10MB 长度的文件，接收窗口为 1MB，报文段长度为 1KB，无拥塞和丢包。
   (a) 运行慢启动算法，需要多少个 RTT，才能将发送窗口增加到 1MB？
   (b) 发送该文件需要多少个 RTT？
   (c) 如果简单地取所需 RTT 数目与链路时延的乘积作为文件发送时间，传输的有效吞吐量为多少？链路带宽的利用率为多少？

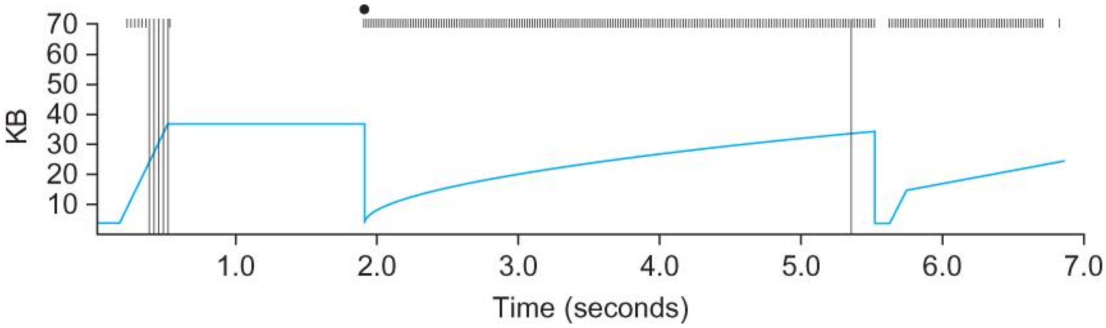
Answer:

(a) In slow start, the size of the window doubles every RTT. At the end of the *i*th RTT, the window size is $2^i$ KB. It will take 10 RTTs before the send window has reached $2^{10}$ KB = 1MB.

(b) After 10 RTTs, 1023KB = 1MB − 1KB has been transferred, and the window size is now 1MB. Since we have not yet reached the maximum capacity of the network, slow start continues to double the window each RTT, so it takes 4 more RTTs to transfer the remaining 9MB (the amounts transferred during each of these last 4 RTTs are 1MB, 2MB, 4MB, 1MB; these are all well below the maximum capacity of the link in one RTT of 12.5MB). Therefore, the file is transferred in 14 RTTs.

(c) It takes 0.7 seconds (14 RTTs) to send the file. The effective throughput is (10MB / 0.7s) = 14.3MBps = 114.3Mbps. This is only 11.4% of the available link bandwidth.

5. (**TCP 拥塞控制**, Ex. 6.27, [PD12])考虑下图所示 TCP 拥塞窗口值随时间变化的轨迹。分别确定连接起始时的慢启动、超时后的慢启动、避免拥塞的线性增加所对应的时间区间。解释 T = 0.5 ~ 1.9 期间发生了什么？产生下图轨迹的 TCP 版本，具有教材中图 6.11 轨迹对应 TCP 版本所不具备的一条特性，这条特性是什么？下图轨迹和教材中图 6.13 所示轨迹都不具备一条特性，这条特性是什么？




Answer: Slow start is active up to about 0.5 sec on startup. At that time a packet is sent that is lost; this loss results in a coarse-grained timeout at T=1.9.

At that point slow start is again invoked, but this time TCP changes to the linear-increase phase of congestion avoidance before the congestion window gets large enough to trigger losses. The exact transition time is difficult to see in the diagram; it occurs sometime around T=2.4.

At T=5.3 another packet is sent that is lost. This time the loss is detected at T=5.5 by fast retransmit; this TCP feature is the one not present in Figure 6.11 of the text, as all lost packets there result in timeouts. Because the congestion window size then drops to 1, we can infer that fast recovery was not in effect; instead, slow start opens the congestion window to half its previous value and then linear increase takes over. The transition between these two phases is shown more sharply here, at T=5.7.

# 7. 应用层

1. (**HTTP**, Ex. 9.12, [PD12])下列 HTTP GET 命令中，为什么还要包含所连接的 Web 服务器的名字？难道该 Web 服务器还不知道自己的名字吗？

```
GET http://www.cs.princeton.edu/index.html HTTP/1.1
```

Answer: One server may support multiple web sites with multiple hostnames, a technique known as *virtual hosting*. HTTP GET requests are referred by the server to the appropriate directory based on the hostname contained in the request.

2. (**DNS 和 HTTP**, P2.6, [KR17]) 假定在浏览器中点击一条链接以获得对应 Web 页面，其 URL 的 IP 地址及协议端口信息在本地没有缓存，因此需要 DNS 查询以获得相关地址信息。如果运行浏览器的本地主机从 DNS 得到地址信息之前已经访问了 $n$ 个 DNS 服务器，相继产生的 RTT 依次为 $RTT_1$、…、$RTT_n$。进一步假定该链接对应 Web 页面只包含一个对象，由少量 HTML 文本组成。令 $RTT_0$ 表示本地主机和包含请求对象的服务器之间的 RTT 值。假定该 Web 对象传输时间为零，从点击该链接到接收到该对象需要多长时间？

Answer: The total amount of time to get the IP address is $RTT_1 + RTT_2 + \ldots + RTT_n$.

Once the IP address is known, $RTT_0$ elapses to set up the TCP connection and another $RTT_0$ elapses to request and receive the small object. The total response time is

$$RTT_1 + RTT_2 + \ldots + RTT_n + 2RTT_0$$

# 8. 综合

1. (**计算机网络系统设计原则**)在计算机网络领域中有很多重要的系统设计原则，试结合案例论述其含义：(a) 沙漏(或称细腰型)的网络体系结构 (Hourglass/narrow waist network architecture model )；(b) 边缘复杂核心简单的网络功能划分(Complex edge and simple core)；(c) 尽力服务的服务模型 (Best-effort service model)。

答：

(1) 沙漏(或称细腰型)的网络体系结构 (Hourglass/narrow waist network architecture model )

沙漏模型表示网络体系结构的协议栈形如细腰，其中间部分采用了代表最小的、经过精心挑选的通用功能集，它允许高层应用和低层通信技术并存。

符合该模型的网络体系结构案例有 TCP/IP。IP 协议可以为各种各样的应用提供服务（所谓的 Everything over IP），同时也可以运行到各式各样的网络上（所谓的 IP over everything）。正因为如此，互联网才发展到今天的这种全球的规模。

(2) 边缘复杂核心简单的网络功能划分 (Complex edge and simple core)

“边缘复杂核心简单”的含义：在网络边缘的主机的实现较为复杂、支持很多功能，而网络的节点的实现则比较简单、只实现较为简单的功能。网络负责实现基础的连接功能，复杂的网络功能留待网络边缘的主机来解决。

符合该模型的例子有分组交换网络，其网络中的节点仅仅转发分组，实现尽可能简单，而终端主机承担更多的任务。让主机负责端到端的可靠性不但没有给主机增加更多的负担，反而能够使更多的应用在简单的网络上运行。不符合该模型的例子是电话交换网络，其终端设备电话的功能很简单，而网络中的节点电话交换机很复杂。

另外一个符合该模型的例子，但是教学内容没有涉及到的就是基于区分服务(DiffServ)的网络服务质量模型。

(3) 尽力服务的服务模型 (Best-effort service model)

“尽力服务”的含义：网络尽力把报文送到其目的地。如果出现分组错误、丢失等情况，网络什么也不做。尽力服务提供的是一种不可靠的传输服务。

符合该模型的案例是 IP 服务模型。IP 数据报在网络中以无连接方式发送分组，尽力的无连接的服务是互联网能够提供的最为简单的服务，有利于保持路由器设计的简单。

# 参考文献

**[PD12]** L. L. Peterson and B. S. Davie, Computer Networks: A Systems Approach (5th Edition), Morgan Kaufmann, 2012 (printed in China by China Machine Press)
**[KR12]** James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach (6th Edition), Pearson, 2012 (Chinese version printed in China by China Machine Press)
**[KR17]** James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach (7th Edition), Pearson, 2017
**[Wu11]** 吴功宜，计算机网络（第三版），清华大学出版社, 2011