**第四、五章练习及答案**

1. **路由器与链路层交换机根本区别。**

路由器基于IP地址转发分组，而链路层交换机基于MAC地址转发。

1. **网络层数据平面和控制平面的主要功能。**

数据平面的主要功能是分组转发，即将数据报从其输入链路转发到它们的输出链路。例如，数据平面的输入端口执行物理层功能，终止路由器上的传入物理链路，执行链路层功能以与传入链路另一侧的链路层互操作，并在输入端口执行查找功能。

控制平面的主要功能是路由，即确定数据包从其源到目的地的路径。控制平面负责执行路由协议，响应连接到上层或下层链路，与远程控制器通信以及执行管理功能。

1. **路由选择和分组转发主要区别。**

路由和转发之间的主要区别在于转发是路由器将数据包从其输入接口传输到其输出接口的本地操作，并且转发在非常短的时间尺度（通常为几纳秒）内进行，因此通常在硬件。路由是指网络范围的进程，用于确定数据包从源到目标的端到端路径。路由发生在更长的时间尺度（通常是几秒），并且通常在软件中实现。

1. **路由器中转发表的主要作用。**

转发表在路由器中的作用是保存条目以确定将通过交换结构转发到达的分组的输出链路接口。

1. **网络层的服务模型。**

互联网网络层的服务模式是尽力服务。使用此服务模型，无法保证按照发送顺序接收数据包，无法保证最终交付，不保证端到端延迟，也不保证最小带宽保证。

1. **路由器各个部分中哪些由硬件实现，哪些由软件实现。**

输入端口，交换结构和输出端口在硬件中实现，因为它们的数据报处理功能对于软件实现来说太快了。传统路由器内的路由处理器使用软件来执行路由协议，维护路由表和附加的链路状态信息，以及计算路由器的转发表。此外，SDN路由器中的路由处理器还依赖于与远程控制器通信的软件，以便接收转发表条目并将其安装在路由器的输入端口中。

由于需要快速处理，例如在纳秒时间尺度，数据平面通常在硬件中实现。控制平面通常以软件实现并且以毫秒或第二时间尺度操作，例如，用于执行路由协议，响应上行或下行的附加链路，与远程控制器通信以及执行管理功能。

1. **路由器基于目的地转发和通用转发有什么区别。**

基于目的地的转发意味着到达路由器的数据报将仅基于数据报的最终目的地被转发到输出接口。通用转发意味着当路由器确定数据报的输出接口时，除了最终目的地之外，还考虑与数据报相关联的其他因素。软件定义网络采用通用转发，例如，转发决策可以基于数据报的TCP / UDP源或目标端口号，以及其目的IP地址。

转发有两个主要操作：匹配和行动。

通过基于目的地的转发，路由器的匹配操作仅查找待转发数据报的目的IP地址，路由器的动作操作包括将数据包发送到交换结构到指定的输出端口。

通过通用转发，可以在与协议栈中不同层的不同协议相关联的多个报头字段上进行匹配，并且动作可以包括将分组转发到一个或多个输出端口，跨多个输出接口对数据包进行负载平衡，重写标头值（如在NAT中），有目的地阻止/丢弃数据包（如在防火墙中），将数据包发送到特殊服务器以进行进一步处理和操作等等。

1. **主机收到一个数据报，在网络层如何知道应当将该报文段（即数据报的有效载荷）交付给TCP还是UDP或其他。**

通过IP首部中的上层协议字段。

1. **在IP首部中，哪个字段可以确保一个分组的转发不超过N台路由器。**

TTL。

1. **当某个路由器发现一IP数据报的检验和有差错时，为什么采取丢弃的办法而不是要求源站重传此数据报？计算首部检验和为什么不采用CRC检验码？**

纠错控制由上层（传输层）执行。

IP首部中的源地址也可能出错，错误的源地址重传数据报是没有意义的。

不采用CRC简化解码计算量，提高路由器的吞吐量。

1. **为什么运输层和网络层都执行差错检验？**

IP层的校验只针对首部，而TCP/UDP的校验和针对整个报文段（包含数据部分）。另外，TCP/UDP与IP不一定属于同一个协议栈，TCP能够运行在不同的网络层协议上（如ATM），IP首部中的上层协议字段也暗示上层可能有除TCP/UDP外的其他协议。

1. **路由聚合用处。**

路由聚合意味着ISP使用单个前缀来通告多个网络。路由聚合很有用，因为ISP可以使用此技术向Internet的其余部分通告ISP具有的多个网络的单个前缀地址。

1. **专用网络地址（私有地址）特点。**

网络中设备的专用网络地址是指仅对该网络内的那些设备有意义的网络地址。

具有专用网络地址的数据报不应存在于较大的公共因特网中，因为专用网络地址可能被其自己的专用网络内的许多网络设备使用。

1. **IPv4和IPv6区别。**

IPv6具有固定长度的头部，其不包括IPv4头部可包括的大多数选项。即使IPv6头部包含两个128位地址（源和目标IP地址），整个标头也只有40字节的固定长度。其中一些领域的精神相似。 IPv6中的流量类，有效载荷长度，下一个报头和跳跃限制分别类似于IPv4中的服务类型，数据报长度，上层协议和生存时间。

1. **基于目的地转发与OpenFlow流表有什么差异。**

基于目的地转发的转发表中的每个条目仅包含IP报头字段值和要转发分组（与IP报头字段值匹配）的输出链路接口。

OpenFlow中流表的每个条目包括一组头字段值，传入的分组将与之匹配，一组计数器在分组与流表条目匹配时更新，以及一组在分组与流表条目匹配时采取的动作。

1. **“匹配加动作”含义。**

“匹配加动作”意味着路由器或交换机试图在分组的某些头部值与流表中的某些条目之间找到匹配，然后根据该匹配，路由器决定使用哪个接口将分组转发，甚至在分组上进行更多操作。

在基于目的地的转发分组交换的情况下，路由器仅尝试在具有到达分组的目的地IP地址的流表条目之间找到匹配，并且动作是确定该分组将是哪个接口转发。在SDN的情况下，可以匹配许多字段，例如，IP源地址，TCP源端口和源MAC地址，还可以采取许多动作，例如，转发，删除和修改字段值。

1. **在IP数据报中举出能够在OpenFlow 1.0通用转发中“匹配”的3个首部字段。**

“匹配”的三个字段是：IP源地址，TCP源端口和源MAC地址。

无法匹配的三个字段是：TTL字段，数据报长度字段，首部校验和（取决于TTL字段）

1. **IP地址分为几类？各如何表示？IP地址的主要特点是什么？**

答：分为ABCDE 5类;

每一类地址都由两个固定长度的字段组成，其中一个字段是网络号net-id，它标志主机（或路由器）所连接到的网络，而另一个字段则是主机号host-id，它标志该主机（或路由器）。

各类地址的网络号字段net-id分别为1，2，3，0，0字节；主机号字段host-id分别为3字节、2字节、1字节、4字节、4字节。

特点：

（1）IP地址是一种分等级的地址结构。分两个等级的好处是：

第一，IP地址管理机构在分配IP地址时只分配网络号，而剩下的主机号则由得到该网络号的单位自行分配。这样就方便了IP地址的管理。

第二，路由器仅根据目的主机所连接的网络号来转发分组（而不考虑目的主机号），这样就可以使路由表中的项目数大幅度减少，从而减小了路由表所占的存储空间。

（2）实际上IP地址是标志一个主机（或路由器）和一条链路的接口。

当一个主机同时连接到两个网络上时，该主机就必须同时具有两个相应的IP地址，其网络号net-id必须是不同的。这种主机称为多归属主机(multihomedhost)。

由于一个路由器至少应当连接到两个网络（这样它才能将IP数据报从一个网络转发到另一个网络），因此一个路由器至少应当有两个不同的IP地址。

(3)用转发器或网桥连接起来的若干个局域网仍为一个网络，因此这些局域网都具有同样的网络号net-id。

(4)所有分配到网络号net-id的网络，范围很小的局域网，还是可能覆盖很大地理范围的广域网，都是平等的。

1. **试说明IP地址与MAC（硬件）地址的区别，为什么要使用这两种不同的地址？**

IP地址就是给每个连接在因特网上的主机（或路由器）分配一个在全世界范围是唯一的32位的标识符。从而把整个因特网看成为一个单一的、抽象的网络。

在实际网络的链路上传送数据帧时，最终还是必须使用硬件地址。

MAC地址在一定程度上与硬件一致，基于物理、能够标识具体的链路通信对象、IP地址给予逻辑域的划分、不受硬件限制。

1. **IP数据报中的首部检验和并不检验数据报中的数据。这样做的最大好处是什么？坏处是什么？**

在首部中的错误比在数据中的错误更严重，例如，一个坏的地址可能导致分组被投寄到错误的主机。许多主机并不检查投递给他们的分组是否确实是要投递给它们，它们假定网络从来不会把本来是要前往另一主机的分组投递给它们。

数据不参与检验和的计算，因为这样做代价大，上层协议通常也做这种检验工作，从而引起重复和多余。

因此，这样做可以加快分组的转发，但是数据部分出现差错时不能及早发现。

1. **什么是最大传送单元MTU？它和IP数据报的首部中的哪个字段有关系？**

IP层下面数据链层所限定的帧格式中数据字段的最大长度，与IP数据报首部中的总长度字段有关系。

1. **在因特网中将IP数据报分片传送的数据报在最后的目的主机进行组装。还可以有另一种做法，即数据报片通过一个网络就进行一次组装。比较这两种方法的优劣。**

在目的地址而不是在中间的路由器进行组装是由于：

（1）路由器处理数据报更简单些；效率高，延迟小。

（2）数据报的各分片可能经过各自的路径。因此在每一个中间的路由器进行组装可能总会缺少几个数据报片；

（3）也许分组后面还要经过一个网络，它还要给这些数据报片划分成更小的片。如果在中间的路由器进行组装就可能会组装多次。（为适应路径上不同链路段所能许可的不同分片规模，可能要重新分片或组装）

1. **一个3200位长的TCP报文传到IP层，加上160位的首部后成为数据报。下面的互联网由两个局域网通过路由器连接起来。但第二个局域网所能传送的最长数据帧中的数据部分只有1200位。因此数据报在路由器必须进行分片。试问第二个局域网向其上层要传送多少比特的数据（这里的“数据”当然指的是局域网看见的数据）?**

第二个局域网所能传送的最长数据帧中的数据部分只有1200bit，即

每个IP数据片的数据部分 < 1200 – 160 = 1040 bit

由于片偏移是以8字节即64 bit为单位，所以IP数据片的数据部分最大不超过1024 bit （1024是小于1040的最大的能被64整除的数，1040/64 = 16.25，取整得到16就是第二片偏移量；16\*64 = 1024）。

这样3200bit的报文要分4个数据片，所以第二个局域网向上传送的比特数等于（3200+4×160），共3840 bit。

1. **主机A发送IP数据报给主机B，途中经过了5个路由器。试问在IP数据报的发送过程中总共使用了几次ARP？**

答：6次，主机用一次，每个路由器各使用一次。

1. **设某路由器建立了如下路由表：**

目的网络 子网掩码 下一跳

128.96.39.0 255.255.255.128 接口m0

128.96.39.128 255.255.255.128 接口m1

128.96.40.0 255.255.255.128 R2

192.4.153.0 255.255.255.192 R3

\*（默认） R4

现共收到5个分组，其目的地址分别为：

（1）128.96.39.10

（2）128.96.40.12

（3）128.96.40.151

（4）192.4.153.17

（5）192.4.153.90

问：以上分组分别从哪个接口转发。

（1）分组的目的站IP地址是：128.96.39.10。

先与子网掩码255.255.255.128相与，得128.96.39.0，可见该分组经接口0转发。

（2）分组的目的IP地址为：128.96.40.12。

与子网掩码255.255.255.128相与得128.96.40.0，经查路由表可知，该项分组经R2转发。

（3）分组的目的IP地址为：128.96.40.151，与子网掩码255.255.255.128相与后得128.96.40.128，与子网掩码255.255.255.192相与后得128.96.40.128，经查路由表知，该分组转发选择默认路由，经R4转发。

（4）分组的目的IP地址为：192.4.153.17。与子网掩码255.255.255.128相与后得192.4.153.0。与子网掩码255.255.255.192相与后得192.4.153.0，经查路由表知，该分组经R3转发。

（5）分组的目的IP地址为：192.4.153.90，与子网掩码255.255.255.128相与后得192.4.153.0。与子网掩码255.255.255.192相与后得192.4.153.64，经查路由表知，该分组经R3转发。

1. **某单位分配到一个B类IP地址，其net-id为129.250.0.0。该单位有4000台机器，分布在16个不同的地点。如选用子网掩码为255.255.255.0，试给每一个地点分配一个子网掩码号，并算出每个地点主机号码的最小值和最大值。**

4000/16 = 250，平均每个地点250台机器。如选255.255.255.0为掩码，则每个网络所连主机数 = 28 – 2 = 254 > 250，共有子网数 = 28 – 2 = 254 > 16，能满足实际需求。

可给每个地点分配如下子网号码：

地点：子网号（subnet-id） 子网网络号 主机IP的最小值和最大值

1：00000001 129.250.1.0 129.250.1.1---129.250.1.254

2：00000010 129.250.2.0 129.250.2.1---129.250.2.254

3：00000011 129.250.3.0 129.250.3.1---129.250.3.254

4：00000100 129.250.4.0 129.250.4.1---129.250.4.254

5：00000101 129.250.5.0 129.250.5.1---129.250.5.254

6：00000110 129.250.6.0 129.250.6.1---129.250.6.254

7：00000111 129.250.7.0 129.250.7.1---129.250.7.254

8：00001000 129.250.8.0 129.250.8.1---129.250.8.254

9：00001001 129.250.9.0 129.250.9.1---129.250.9.254

10：00001010 129.250.10.0 129.250.10.1---129.250.10.254

11：00001011 129.250.11.0 129.250.11.1---129.250.11.254

12：00001100 129.250.12.0 129.250.12.1---129.250.12.254

13：00001101 129.250.13.0 129.250.13.1---129.250.13.254

14：00001110 129.250.14.0 129.250.14.1---129.250.14.254

15：00001111 129.250.15.0 129.250.15.1---129.250.15.254

16：00010000 129.250.16.0 129.250.16.1---129.250.16.254

1. 

   **R2**

   **R1**

   L1

   130.11.120.1

   E1

   互联网

   L0

   202.118.2.2

   E0

   202.118.3.1

   202.118.3.2

   L0

   202.118.2.1

   局域网1

   E2

   局域网2

   域名服务器

   **某公司网络拓扑如下图所示，路由器R1通过接口E1、E2分别连接局域网1、局域网2， 通过接口L0连接路由器R2，并通过路由器R2连接域名服务器与互联网。R1的L0接口的IP地址是 202.118.2.1；R2的L0接口的IP地址是202.118.2.2，L1接口的IP地址是130.11.120.1，E0接口的 IP地址是202.118.3.1；域名服务器的IP地址是202.118.3.2。**

**R1和R2的路由表结构为：**

|  |  |  |
| --- | --- | --- |
| 目的网络IP地址 | 子网掩码 | 下一跳 |

1. **将IP地址空间202.118.1.0/24划分为两个子网，分配给局域网1、局域网2，每个局域网分配的IP地址数不少于120个。请给出子网划分结果，说明理由或给出必要的计算过程。**

子网掩码：255.255.255.128。

局域网1： 202.118.1.0/25；

有效IP地址范围为202.118.1.1~202.118.1.126

网络地址为202.118.1.0；广播地址为202.118.1.127。

局域网2： 202.118.1.128/25；

有效IP地址范围：202.118.1.129~202.118.1.254

网络地址为202.118.1.128；广播地址为202.118.1.255。

1. **请给出R1的路由表，使其明确包括到局域网1的路由、局域网2的路由、域名服务器的主机路由和互联网的路由。**

|  |  |  |
| --- | --- | --- |
| 目的网络IP地址 | 子网掩码 | 下一跳 |
| 202.118.1.0 | 255.255.255.128 | 接口E1 |
| 202.118.1.128 | 255.255.255.128 | 接口E2 |
| 202.118.3.2 | 255.255.255.255 | R2（202.118.2.2） |
| 0.0.0.0 | 0.0.0.0 | R2（202.118.2.2） |

1. **请采用路由聚合技术，给出R2到局域网1和局域网2的路由。**

|  |  |  |
| --- | --- | --- |
| 目的网络IP地址 | 子网掩码 | 下一跳 |
| 202.118.1.0 | 255.255.255.0 | R1（202.118.2.1） |

1. **.一个数据报长度为4000字节（固定首部长度）。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？各数据报片的数据字段长度、片偏移字段和MF标志应为何数值？**

答：

IP数据报固定首部长度为20字节

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |

1. **试找出可产生以下A类子网数目的子网掩码（采用连续掩码）。**

（1）2，（2）6，（3）30，（4）62，（5）122，（6）250.

答：（1）255.192.0.0，（2）255.224.0.0，（3）255.248.0.0，（4）255.252.0.0，（5）255.254.0.0，（6）255.255.0.0

1. **以下有4个子网掩码。哪些是不推荐使用的？为什么？**

（1）176.0.0.0，（2）96.0.0.0，（3）127.192.0.0，（4）255.128.0.0。

答：只有（4）是连续的1和连续的0的掩码，是推荐使用的

1. **有如下的4个/24地址块，试进行最大可能性的聚合。**

212.56.132.0/24

212.56.133.0/24

212.56.134.0/24

212.56.135.0/24

答：

212 =（11010100）2，56 =（00111000）2，

132 =（10000100）2，

133 =（10000101）2，

134 =（10000110）2，

135 =（10000111）2

所以共同的前缀有22位，即1101010000111000100001，聚合的CIDR地址块是：212.56.132.0/22

1. **有两个CIDR地址块208.128/11和208.130.28/22。是否有哪一个地址块包含了另一个地址？如果有，请指出，并说明理由。**

答：

208.128/11的前缀为：11010000 100

208.130.28/22的前缀为：11010000 10000010 000101，它的前11位与208.128/11的前缀是一致的，所以208.128/11地址块包含了208.130.28/22这一地址块。

1. **以下地址中的哪一个和90.32/12匹配：请说明理由。**

（1）90.33.192.123 （2）90.79.87.216 （3）90.58.224.74 （4）90.68.200.154。

90.32/12 -> 90.0010 0000，第二字节的前4位在前缀中。

给出的四个地址的第二字节的前4位分别为：0010，0100，0011和0100。因此只有（1）是匹配的。

1. **以下地址中的哪一个地址2.52.90.140匹配？请说明理由。**

（1）0/4；（2）32/4；（3）4/6；（4）152.0/11

答：

前缀（1）和地址2.52.90.140匹配。

2.52.90.140 -> 00000010.52.90.140

0/4 -> 00000000

32/4 -> 00100000

4/6 -> 00000100

80/4 -> 01010000

1. **下面的前缀中的哪一个和地址152.7.77.159及152.31.47.252都匹配？请说明理由。**

（1）152.40/13；（2）153.40/9；（3）152.64/12；（4）152.0/11。

答：前缀（4）和这两个地址都匹配。

1. **与下列掩码相对应的网络前缀各有多少位？**

（1）192.0.0.0；（2）240.0.0.0；（3）255.254.0.0；（4）255.255.255.252。

答：（1）/2; (2)/4; (3)/11; (4)/30。

1. **已知地址块中的一个地址是140.120.84.24/20。试求这个地址块中的最小地址和最大地址。地址掩码是什么？地址块中共有多少个地址？相当于多少个C类地址？**

答：

140.120.84.24 -> 140.120.(01010100).24

最小地址是140.120.(0101**0000**).0/20 -> 140.120.80.0

最大地址是140.120.(0101**1111**).255/20 -> 140.120.95.255

地址数是4096，相当于16个C类地址。

1. **已知地址块中的一个地址是190.87.140.202/29。试求这个地址块中的最小地址和最大地址。地址掩码是什么？地址块中共有多少个地址？相当于多少个C类地址？**

答：

190.87.140.202/29 -> 190.87.140.(11001010)/29

最小地址是190.87.140.(11001**000**)/29 190.87.140.200

最大地址是190.87.140.(11001**111**)/29 190.87.140.207

地址数是8，相当于1/32个C类地址。

1. **某单位分配到一个地址块136.23.12.64/26。现在需要进一步划分为4个一样大的子网。试问:**

**（1）每一个子网的网络前缀有多长？**

**（2）每一个子网中有多少个地址？**

**（3）每一个子网的地址是什么？**

**（4）每一个子网可分配给主机使用的最小地址和最大地址是什么？**

答：

（1）每个子网前缀28位。

（2）每个子网的地址中有4位留给主机用，因此共有16个地址。

（3）四个子网的地址块是：

第一个地址块136.23.12.64/28，可分配给主机使用的

最小地址：136.23.12.01000001＝136.23.12.65/28

最大地址：136.23.12.01001110＝136.23.12.78/28

第二个地址块136.23.12.80/28，可分配给主机使用的

最小地址：136.23.12.01010001＝136.23.12.81/28

最大地址：136.23.12.01011110＝136.23.12.94/28

第三个地址块136.23.12.96/28，可分配给主机使用的

最小地址：136.23.12.01100001＝136.23.12.97/28

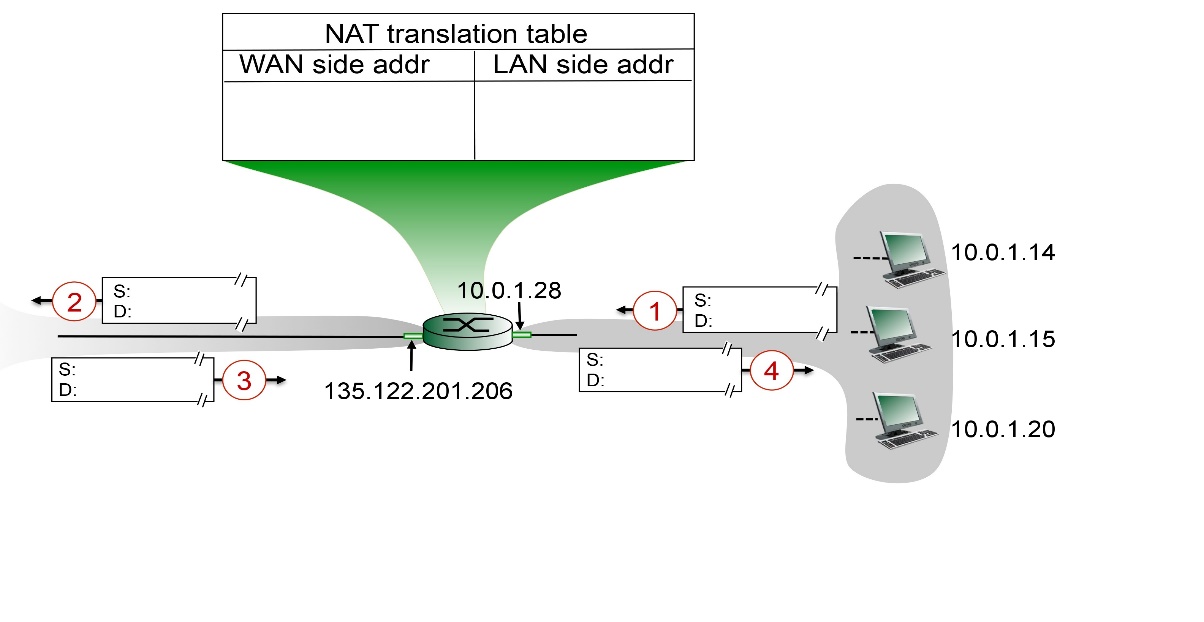
最大地址：136.23.12.01101110＝136.23.12.110/28

第四个地址块136.23.12.112/28，可分配给主机使用的

最小地址：136.23.12.01110001＝136.23.12.113/28

最大地址：136.23.12.01111110＝136.23.12.126/28

1. **考虑下面的场景，其中三个主机（具有专用IP地址10.0.1.14、10.0.1.15、10.0.1.20）位于本地网络中，位于这三个主机和更大的Internet之间的NATted路由器后面。从这三个主机发送或发送到这三个主机的IP数据报必须通过这个NAT路由器。局域网侧路由器接口的IP地址为10.0.1.28，而互联网侧路由器的IP地址为135.122.201.206。**



假设IP地址为10.0.1.14的主机向主机128.119.173.182发送一个IP数据报。源端口3419，目的端口80。

1. 在主机发送数据报之后，但在它到达NATted路由器之前，考虑步骤1中的数据报。此数据报的源和目标IP地址是什么？此IP数据报中TCP段的源端口号和目标端口号是什么？
2. 现在考虑步骤2中的数据报，在它被NATted路由器传输之后。此数据报的源和目标IP地址是什么？此IP数据报中TCP段的源端口号和目标端口号是什么？确定步骤1和步骤2之间数据报的IP地址和端口号的差异。指定在路由器的NAT表中所做的条目。
3. 现在考虑步骤3中的数据报，就在它被NATted路由器接收之前。此数据报的源和目标IP地址是什么？此IP数据报中TCP段的源端口号和目标端口号是什么？
4. 最后，考虑步骤4中的数据报，在它被NATted路由器传输之后，在它被主机接收之前。这个数据报的源和目标IP地址是什么？此IP数据报中TCP段的源端口号和目标端口号是什么？确定步骤3和步骤4之间数据报的IP地址和端口号的差异。路由器的NAT表中是否有新条目，或者从NAT表中删除了新条目？解释你的答案。

**解答：**

步骤1：

IP数据报的源地址：10.0.1.14

IP数据报的目标地址：128.119.173.182

IP数据报TCP段的源端口号：3419

IP数据报的TCP段的目标端口号：80

主机10.0.0.14已分配任意源端口号3419，并将数据报发送到LAN。数据报在NAT路由器的右端口接收。

步骤2：

IP数据报的源地址：135.122.201.206

IP数据报的目标地址：128.119.173.182

IP数据报TCP段的源端口号：5512

IP数据报的TCP段的目标端口号：80

在从主机10.0.0.14接收到数据报后，NAT路由器为数据报生成一个新的源端口号5512（在NAT表中尚未使用），并用新的源端口号5512替换原来的源端口号3419。离开主路由器前往更大互联网的流量具有NAT路由器的源IP，即135.122.201.206，因此数据报的源IP地址现在变为135.122.201.206。目标地址和端口号保持不变。在第2步之后的NAT表如下（参见文本中的图4.25）：

|  |  |
| --- | --- |
| **WAN-side address** | **LAN-side address** |
| 135.122.201.206, 5512 | 10.0.1.14, 3419 |

步骤3：

IP数据报的源地址：128.119.173.182

IP数据报的目标地址：135.122.201.206

IP数据报中TCP段的源端口号：80

IP数据报中TCP段的目标端口号：5512

此到达的数据报由远程主机128.119.173.182发送，以响应上述步骤2中此NAT路由器发送的数据报。

步骤4：

IP数据报的源地址：128.119.173.182

IP数据报的目标地址：10.0.1.14

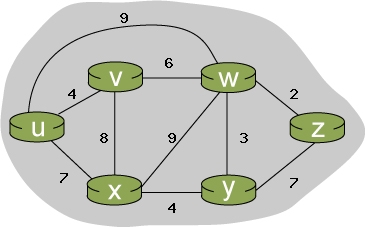
IP数据报的TCP段的源端口号：80

IP数据报的TCP段的目标端口号：3419

当此数据报从Internet到达NAT路由器的左端口时，路由器使用目标IP地址和目标端口号为NAT转换表编制索引，以获得家庭网络中目标主机的适当IP地址（10.0.1.14）和目标端口（3419）。然后，路由器重写数据报的目标地址和目标端口号，并将数据报转发到家庭网络。

1. **Dijkstra的链路状态算法（用于计算最小成本路径）**

**考虑下面所示的6节点网络，以及给定的链路成本。**



使用Dijkstra算法，找到从源节点**u**到所有其他目的地的最低成本路径。以表格格式作答。

**解答：**

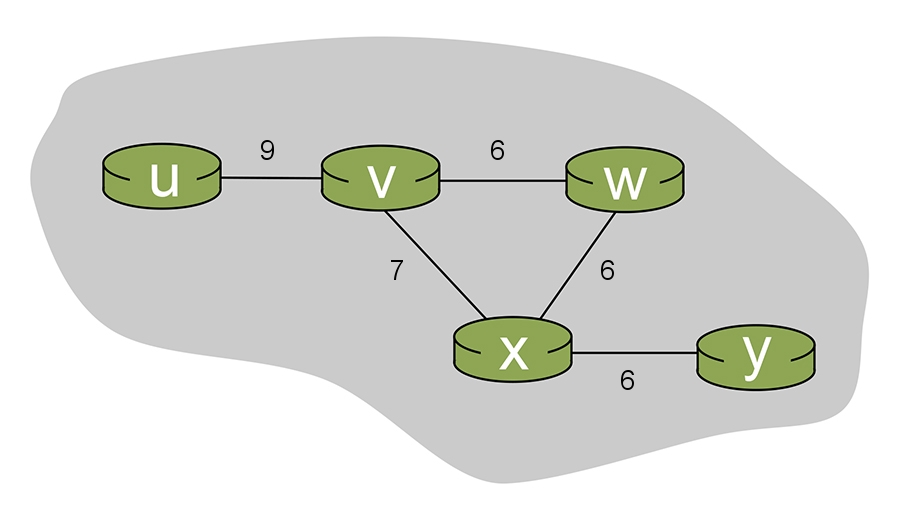
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***N*'** | ***D*(*u*), *p*(*u*)** | ***D*(*v*), *p*(*v*)** | ***D*(*w*), *p*(*w*)** | ***D*(*x*), *p*(*x*)** | ***D*(*y*), *p*(*y*)** | ***D*(*z*), *p*(*z*)** |
| **u** | **-** | **4, u** | **9, u** | **7,u** | **∞** | **∞** |
| **uv** | **-** | **-** | **9, u** | **7, u** | **∞** | **∞** |
| **uvx** | **-** | **-** | **9, u** | **-** | **11, x** | **∞** |
| **uvxw** | **-** | **-** | **-** | **-** | **11, x** | **11, w** |
| **uvxwy** | **-** | **-** | **-** | **-** | **-** | **11, w** |
| **uvxwyz** | **-** | **-** | **-** | **-** | **-** | **-** |
|  |  |  |  |  |  |  |

从节点u到所有目的地的最低成本路由树中的链接是：

u到v，u到w，u到x，x到y，w到z。

1. **Bellman-Ford距离矢量算法（用于计算最小成本路径）**

**考虑下面所示的6节点网络，以及给定的链路成本。**



回答以下问题。

（1）当算法收敛时，每个路由器中的距离向量是多少。请注意，这个网络很简单，你应该能够通过检查确定最终的距离向量（即，不必实际执行距离向量算法）。

（2）当每个节点只知道其直接连接邻居的单跳链路成本时，初始距离向量是什么？

（3）现在让我们考虑执行距离向量算法。我们将考虑一个迭代模型，其中所有路由器在同一时间和每次迭代时都将距离向量发送给它们的邻居。（然而，请记住，实际上，只有当路由器自己的DV由于从邻居接收到DV而发生变化时，路由器才会将其发送给邻居；这里我们将忽略这一点，并简单地假设所有节点在每次迭代时都与邻居交换其DV）

假设在第i次迭代之后，每个节点中的距离向量为：

du(u,v,x,w,y) = (0,\*,\*,\*,\*)

dv(u,v,x,w,y) = (\*,0,\*,\*,\*)

dw(u,v,x,w,y) = (\*,\*,0,\*,\*)

dx(u,v,x,w,y) = (\*,\*,\*,0,\*)

dy(u,v,x,w,y) = (\*,\*,\*,\*,0)

上面的每个\*都是指某个值。在每次迭代之后，在每个路由器上计算新的dv，假设在每次迭代中，一个路由器从它的每个邻居那里接收一个DV（在上一次迭代中计算），然后使用这些DV在当前迭代中计算它自己的新DV。对于每个迭代，显示该迭代中每个路由器的DV。

**解答：**

（1）节点u的距离向量为：

Du = [ Du(u),Du(v),Du(x),Du(w),Du(y) ]

Du = [ 0,9,16,15,22 ]

节点v的距离向量为：

Dv = [ Dv(u),Dv(v),Dv(x),Dv(w),Dv(y) ]

Dv = [ 9,0,7,6,13 ]

节点x的距离向量为：

Dx = [ Dx(u),Dx(v),Dx(x),Dx(w),Dx(y) ]

Dx = [ 16,7,0,6,6 ]

节点w的距离向量为：

Dw = [ Dw(u),Dw(v),Dw(x),Dw(w),Dw(y) ]

Dw = [ 15,6,6,0,12 ]

节点y的距离向量为：

Dy = [ Dy(u),Dy(v),Dy(x),Dy(w),Dy(y) ]

Dy = [ 22,13,6,12,0 ]

（2） 每个路由器的初始距离向量为：

Du = [0,9,∞,∞,∞]

Dv = [9,0,7,6,∞]

Dx = [∞,7,0,6,6]

Dw = [∞,6,6,0,∞]

Dy = [∞,∞,6,∞,0]

（3）第一次迭代后节点的距离向量：

Du = [0,9,16,∞,∞]

Dv = [9,0,7,6,13]

Dx = [16,7,0,6,6]

Dw = [∞,6,6,0,12]

Dy = [∞,13,6,12,0]

第二次迭代后节点的距离向量：

Du = [0,9,16,15,∞]

Dv = [9,0,7,6,13]

Dx = [16,7,0,6,6]

Dw = [15,6,6,0,12]

Dy = [∞,13,6,12,0]

第三次迭代后节点的距离向量：

Du = [0,9,16,15,22]

Dv = [9,0,7,6,13]

Dx = [16,7,0,6,6]

Dw = [15,6,6,0,12]

Dy = [22,13,6,12,0]

第四次迭代后节点的距离向量：

Du = [0,9,16,15,22]

Dv = [9,0,7,6,13]

Dx = [16,7,0,6,6]

Dw = [15,6,6,0,12]

Dy = [22,13,6,12,0]

[注]第4次迭代后，上面显示的距离向量停止，因为上面显示的最后两次迭代中计算的距离向量相同。回想一下，在实践中，路由器不会将它们的距离向量发送给邻居，除非它的dv发生了变化。

1. **教材P238，P5。**

a)

**前缀匹配 链路接口**

11100000 00 0

11100000 01000000 1

1110000 2

11100001 1 3

otherwise 3

b) 第1个地址的前缀匹配是第5个条目：链路接口3；

第2个地址的前缀匹配是第3个条目：链路接口2；

第3个地址的前缀匹配是第4个条目：链路接口3。

1. **教材P239，P6。**

**目的地址范围 链路接口**

00000000 - 00111111 0

01000000 - 01011111 1

01100000 – 01111111 2

10000000 – 10111111 2

11000000 - 11111111 3

接口0地址数 = 

接口1地址数= 

接口2地址数= 

接口3地址数= 

1. **教材P239，P8。**

223.1.17.0/26

223.1.17.128/25

223.1.17.192/28

1. **教材P239，P9。**

**Destination Address Link Interface**

200.23.16/21 0

200.23.24/24 1

200.23.24/21 2

otherwise 3

1. **教材P239，P10。**

**Destination Address Link Interface**

11100000 00 (224.0/10) 0

11100000 01000000 (224.64/16) 1

1110000 (224/8) 2

11100001 1 (225.128/9) 3

otherwise 3

1. **教材P239，P11。**

在地址范围128.119.40.128 到128.119.40.191 中的任何 IP 地址。

四个相同数量的子网：

128.119.40.64/28, 128.119.40.80/28, 128.119.40.96/28, 128.119.40.112/28

1. **教材P239，P12。**

从 214.97.254/23 中，可能的分配是：

a) Subnet A: 214.97.255/24 (256 addresses)

Subnet B: 214.97.254.0/25 - 214.97.254.0/29 (128-8 = 120 addresses)

Subnet C: 214.97.254.128/25 (128 addresses)

Subnet D: 214.97.254.0/31 (2 addresses)

Subnet E: 214.97.254.2/31 (2 addresses)

Subnet F: 214.97.254.4/30 (4 addresses)

b) 为简化方案，假设没有数据报将路由器接口作为最终目的地。另外，分别将右上，下和左上内部子网标记为D，E，F。

**Router 1**

**Longest Prefix Match Outgoing Interface**

11010110 01100001 11111111 Subnet A

11010110 01100001 11111110 0000000 Subnet D

11010110 01100001 11111110 000001 Subnet F

**Router 2**

**Longest Prefix Match Outgoing Interface**

11010110 01100001 11111111 0000000 Subnet D

11010110 01100001 11111110 0 Subnet B

11010110 01100001 11111110 0000001 Subnet E

**Router 3**

**Longest Prefix Match Outgoing Interface**

11010110 01100001 11111111 000001 Subnet F

11010110 01100001 11111110 0000001 Subnet E

11010110 01100001 11111110 1 Subnet C

1. **教材P239，P14。**

每个片段中数据字段的最大大小= 680（因为有20个字节的IP首部）。因此需要的片段数量 

每个片段将具有标识号422，除最后一个片段之外的每个片段将具有700字节的大小（包括IP头部）。最后一个数据报的大小为360字节（包括IP头）。 4个片段的偏移量为0,85,170,255。前3个片段中的每个片段都有flag = 1，最后一个片段将有flag = 0。

1. **教材P240，P15。**

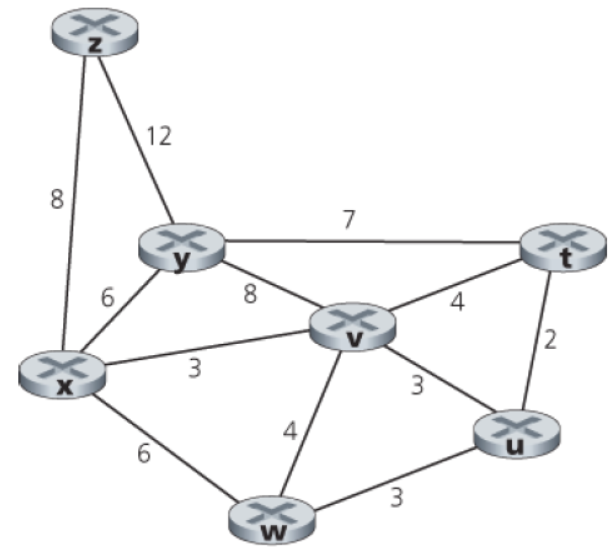
MP3文件大小= 500万字节。

假设数据在TCP段中携带，每个TCP段也具有20个字节的头。然后每个数据报可以携带1500-40 = 1460字节的MP3文件。

所需的数据报数量 。除最后一个数据报之外的所有数据报都是1,500字节;最后一个数据报将是960 + 40 = 1000字节。请注意，这里没有分片 - 源主机不会创建大于1500字节的数据报，并且这些数据报小于链路层的MTU。

1. **教材P279，P3。**

**用Dijkstra最短路径算法计算从 x 到所有节点的最短路径。**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Step*** | ***N’*** | ***D(t),p(t)*** | ***D(u),p(u)*** | ***D(v),p(v)*** | ***D(w),p(w)*** | ***D(y),p(y)*** | ***D(z),p(z)*** |
| 0 | **x** | ∞ | ∞ | 3,x | 6,x | 6,x | 8,x |
| 1 | **xv** | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 2 | **xvu** | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 3 | **xvuw** | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 4 | **xvuwy** | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 5 | **xvuwyt** | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 6 | **xvuwytz** | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |

1. **教材P279，P4。**

**考虑P3的网络（即上题）：**

1. **计算出从 t 到所有网络节点的最短路径。**
2. **计算出从 u 到所有网络节点的最短路径。**
3. **计算出从 v 到所有网络节点的最短路径。**
4. **计算出从 w 到所有网络节点的最短路径。**
5. **计算出从 y 到所有网络节点的最短路径。**
6. **计算出从 z 到所有网络节点的最短路径。**

a)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Step* | *N’* | *D(x), p(x)* | *D(u),p(u)* | *D(v),p(v)* | *D(w),p(w)* | *D(y),p(y)* | *D(z),p(z)* |
| 0 | t | ∞ | 2,t | 4,t | **∞** | 7,t | ∞ |
| 1 | tu | ∞ | 2,t | 4,t | 5,u | 7,t | ∞ |
| 2 | tuv | 7,v | 2,t | 4,t | 5,u | 7,t | ∞ |
| 3 | tuvw | 7,v | 2,t | 4,t | 5,u | 7,t | ∞ |
| 4 | tuvwx | 7,v | 2,t | 4,t | 5,u | 7,t | 15,x |
| 5 | tuvwxy | 7,v | 2,t | 4,t | 5,u | 7,t | 15,x |
| 6 | tuvwxyz | 7,v | 2,t | 4,t | 5,u | 7,t | 15,x |
|  |  |  |  |  |  |  |  |

b)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Step* | *N’* | *D(x), p(x)* | *D(t),p(t)* | *D(v),p(v)* | *D(w),p(w)* | *D(y),p(y)* | *D(z),p(z)* |
|  | u | ∞ | 2,u | 3,u | 3,u | ∞ | ∞ |
|  | ut | ∞ | 2,u | 3,u | 3,u | 9,t | ∞ |
|  | utv | 6,v | 2,u | 3,u | 3,u | 9,t | ∞ |
|  | utvw | 6,v | 2,u | 3,u | 3,u | 9,t | ∞ |
|  | utvwx | 6,v | 2,u | 3,u | 3,u | 9,t | 14,x |
|  | utvwxy | 6,v | 2,u | 3,u | 3,u | 9,t | 14,x |
|  | utvwxyz | 6,v | 2,u | 3,u | 3,u | 9,t | 14,x |
|  |  |  |  |  |  |  |  |

c)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Step* | *N’* | *D(x), p(x)* | *D(u),p(u)* | *D(t),pt)* | *D(w),p(w)* | *D(y),p(y)* | *D(z),p(z)* |
|  | v | 3,v | 3,v | 4,v | 4,v | 8,v | ∞ |
|  | vx | 3,v | 3,v | 4,v | 4,v | 8,v | 11,x |
|  | vxu | 3,v | 3,v | 4,v | 4,v | 8,v | 11,x |
|  | vxut | 3,v | 3,v | 4,v | 4,v | 8,v | 11,x |
|  | vxutw | 3,v | 3,v | 4,v | 4,v | 8,v | 11,x |
|  | vxutwy | 3,v | 3,v | 4,v | 4,v | 8,v | 11,x |
|  | vxutwyz | 3,v | 3,v | 4,v | 4,v | 8,v | 11,x |
|  |  |  |  |  |  |  |  |

d)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Step* | *N’* | *D(x), p(x)* | *D(u),p(u)* | *D(v),p(v)* | *D(t),p(t)* | *D(y),p(y)* | *D(z),p(z)* |
|  | w | 6,w | 3,w | 4,w | ∞ | ∞ | ∞ |
|  | wu | 6,w | 3,w | 4,w | 5,u | ∞ | ∞ |
|  | wuv | 6,w | 3,w | 4,w | 5,u | 12,v | ∞ |
|  | wuvt | 6,w | 3,w | 4,w | 5,u | 12,v | ∞ |
|  | wuvtx | 6,w | 3,w | 4,w | 5,u | 12,v | 14,x |
|  | wuvtxy | 6,w | 3,w | 4,w | 5,u | 12,v | 14,x |
|  | wuvtxyz | 6,w | 3,w | 4,w | 5,u | 12,v | 14,x |
|  |  |  |  |  |  |  |  |

e)

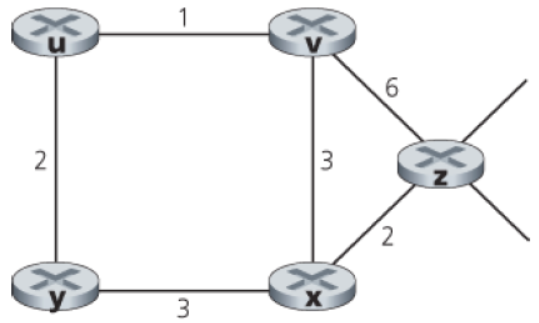
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Step* | *N’* | *D(x), p(x)* | *D(u),p(u)* | *D(v),p(v)* | *D(w),p(w)* | *D(t),p(t)* | *D(z),p(z)* |
|  | y | 6,y | ∞ | 8,y | ∞ | 7,y | 12,y |
|  | yx | 6,y | ∞ | 8,y | 12,x | 7,y | 12,y |
|  | yxt | 6,y | 9,t | 8,y | 12,x | 7,y | 12,y |
|  | yxtv | 6,y | 9,t | 8,y | 12,x | 7,y | 12,y |
|  | yxtvu | 6,y | **9,t** | 8,y | 12,x | 7,y | 12,y |
|  | yxtvuw | 6,y | 9,t | 8,y | 12,x | 7,y | 12,y |
|  | yxtvuwz | 6,y | 9,t | 8,y | 12,x | 7,y | 12,y |
|  |  |  |  |  |  |  |  |

f)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Step* | *N’* | *D(x), p(x)* | *D(u),p(u)* | *D(v),p(v)* | *D(w),p(w)* | *D(y),p(y)* | *D(t),p(t)* |
|  | z | 8,z | ∞ | ∞ | ∞ | 12,z | ∞ |
|  | zx | 8,z | ∞ | 11,x | 14,x | 12,z | ∞ |
|  | zxv | 8,z | 14,v | **11,x** | 14,x | 12,z | 15,v |
|  | zxvy | 8,z | 14,v | 11,x | 14,x | **12,z** | 15,v |
|  | zxvyu | 8,z | **14,v** | 11,x | 14,x | 12,z | 15,v |
|  | zxvyuw | 8,z | 14,v | 11,x | **14,x** | 12,z | 15,v |
|  | zxvyuwt | 8,z | 14,v | 11,x | **14,x** | 12,z | 15,v |
|  |  |  |  |  |  |  |  |

1. **教材P279，P5。**

**利用距离向量算法，写出节点 z 中的距离表项。**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Cost to** | | | | |
|  |  | ***u*** | ***v*** | ***x*** | ***y*** | ***z*** |
| **From** | ***v*** | **∞** | **∞** | **∞** | **∞** | **∞** |
| ***x*** | **∞** | **∞** | **∞** | **∞** | **∞** |
| ***z*** | ∞ | **6** | **2** | ∞ | **0** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Cost to** | | | | |
|  |  | ***u*** | ***v*** | ***x*** | ***y*** | ***z*** |
| **From** | ***v*** | **1** | **0** | **3** | **∞** | **6** |
| ***x*** | **∞** | **3** | **0** | **3** | **2** |
| ***z*** | **7** | **5** | **2** | **5** | **0** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Cost to** | | | | |
|  |  | ***u*** | ***v*** | ***x*** | ***y*** | ***z*** |
| **From** | ***v*** | **1** | **0** | **3** | **3** | **5** |
| ***x*** | **4** | **3** | **0** | **3** | **2** |
| ***z*** | **6** | **5** | **2** | **5** | **0** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Cost to** | | | | |
|  |  | ***u*** | ***v*** | ***x*** | ***y*** | ***z*** |
| **From** | ***v*** | **1** | **0** | **3** | **3** | **5** |
| ***x*** | **4** | **3** | **0** | **3** | **2** |
| ***z*** | **6** | **5** | **2** | **5** | **0** |