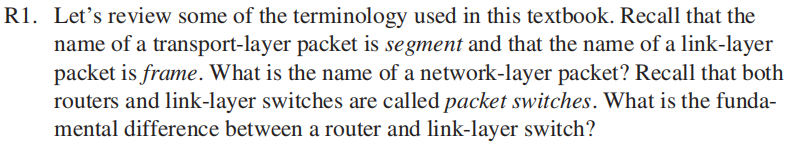
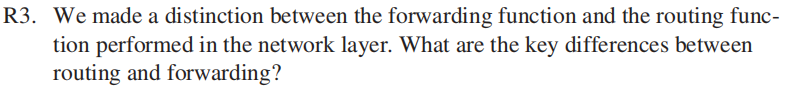
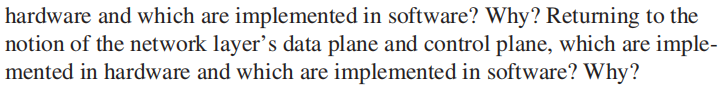
注：英文题目来自第八版教材，答案是根据第七版中文教材做的，所以可能会有所出入或者对不上号。未用第七版英文题目进行更正，因为第八版的新题也很有价值，可以日后研究。



解：网络层的分组名字是数据报。路由器和链路层交换机的根本区别是它们服务于不同的网络层协议。链路层交换机基于链路层帧中的字段值做出转发决定，服务于第二层链路层；路由器基于网络层数据报中的首部字段值做出转发决定，服务于第三层网络层。



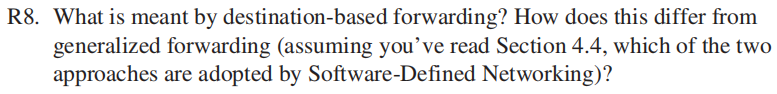
解：转发是指将分组从一个输入链路接口转移到适当的输出链路接口的路由本地动作，转发发生的时间很短，通常用硬件实现。路由选择是指确定分组从源到目的地所采取的端到端路径的网络范围处理过程，发生的时间较长，通常为几秒，因此用软件实现。



解：输入端口，输出端口，交换结构是用硬件实现的。路由选择处理器是用软件实现的。转到网络层的数据平面和控制平面的概念，数据平面是用硬件实现的，控制平面是用软件实现的。原因是在数据平面处理两个数据报之间的时间非常短，软件无法在这么短的时间内完成转发。控制平面则不同于数据平面的，它的主要功能是维护转发表，逻辑比数据平面复杂多了，而且它不需要短时间处理大量时间，只需要及时更新就行了，因此用软件实现。



解：首先一个路由器是有很多个输入端口的，转发行为的第一步是在转发表中查找输出接口。如果每个输入端口都并发地调用路由选择器查找输出接口必然会产生时延。所以要在高速路由器的每个输入端口都存储转发表的影子副本，使转发决策能在每个输入端口本地做出，避免了集中式处理的瓶颈。



解：软件定义网络SDN采用的是通用转发。基于目的地转发意味着：如果是根据目的地的IP地址转发的话，路由转发表中可能需要有40多亿项。当然可以通过匹配IP地址的最长前缀对所有IP地址进行分组。但是这样的路由器功能太单一，不具有普遍性。

通用转发延续了基于目的转发的“匹配+动作”模式，但不受限于通过源分组的IP地址匹配目的IP地址进行转发，而是通过富足首部字段值集合和计数器集合对动作集合进行匹配。要知道正因为有了通用转发这样对“匹配+动作”模式的高度抽象，才使得网络层变得丰富多彩。



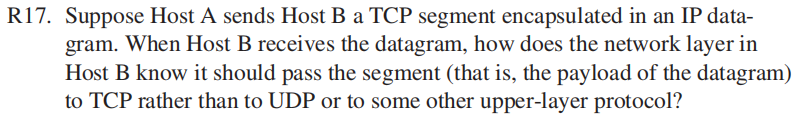
解：分组到达输入端口后需要经过交换结构传送到输出端口。如果有两个分组要同时传送到某个输出端口，那么在交换结构传输排在前面的分组时，排在后面的分组必须等待。这就形成了输入端口队列，若输入端口分组到达的速率大于交换结构的处理速率，队列就会越来越长，最后导致分组丢失。

由于只要输入链路上的分组到达速率达到其容量的58%，输入队列将无限增大导致大量丢包，要消除分组丢失需要控制分组的到达速率。



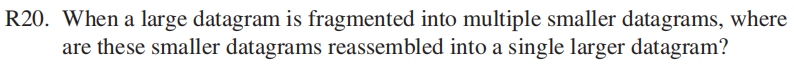
解：分组从输入端口经过交换结构到达输出端口，输出端口将分组推出到输出链路上，假如交换结构（纵横交换结构）同时将3个分组传输到输出端口上，而输出端口一次只能发送一个分组，那么将会有2个分组在输出端口排队。如果不断有大于1个的分组同时从交换结构传输到输出端口，输出端口的排队将会变长，当输出端口缓存耗尽后会出现分组丢失。

正是因为交换结构传输分组到输出端口的速率快于输出端口的传输速率导致输出端口出现排队，所以提高交换结构速率不能防止这种丢失。

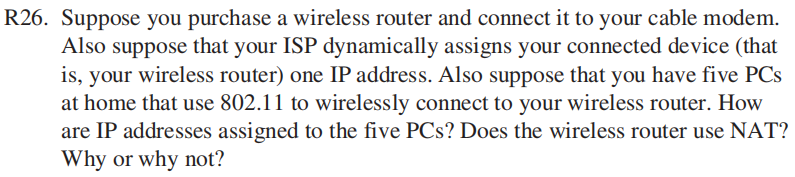


解：主机B中的网络层通过查看数据报首部字段中的协议字段得知应该把数据部分交给哪个特定的运输层协议。比如值为6表明交给TCP，值为17表明交给UDP。

协议字段的作用相当于运输层报文段首部中的目的端口号，指示将数据交给哪个套接字。链路层的链路层帧同样也有特殊字段用于将链路层和网络层绑定在一起。



解：网络层的数据报需要往下经过链路层封装成链路层帧才能放上链路开始传输。然而一个链路层帧的载荷大小根据链路层协议的不同而不同，数据报的大小不能超过一个链路层帧的最大传送单元（MaximumTransmissionUnit，MTU）。所以当链路层遇到大于自身MTU的数据报时需要把数据报分割成多个较小的数据报。这项工作将放到端系统中进行，因为如果在路由器中进行的话会大大降低路由器的性能。较小的数据报在目的端系统的网络层进行装配。

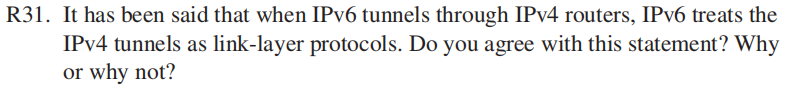


解：这5台PC的IP地址会通过请求DHCP服务器获得。该无线路由器会使用NAT，原因是：家庭网络可能会随时加入许多台联网设备，而最简单常用的管理这些设备IP地址的方法就是NAT。

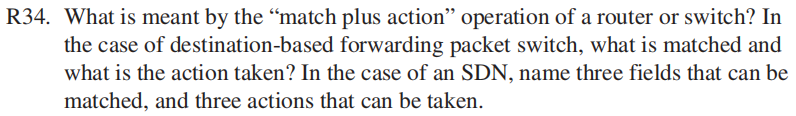


解：路由聚合意味着一个子网只通过接入IP地址的部分前缀和外部因特网相连，外界并不关心子网内还存在着多个组织，这种使用单个网络前缀通告多个网络的能力通常称为路由聚合。

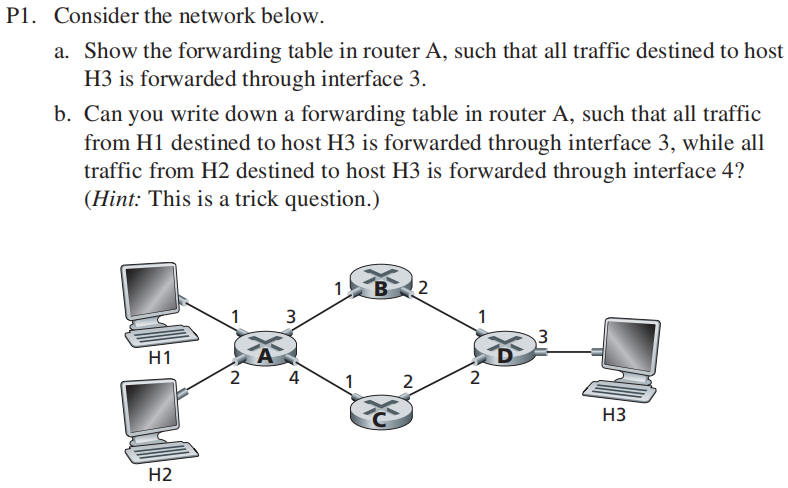
对于路由器而言，假设一台组织外的路由器要转发一个数据报，该数据报的目的地址在组织内部，那么该组织外的路由器转发到组织内的一台路由器时，只需要考虑地址的前x比特即可。这个做法相当大地减少了在这些路由器中转发表的长度。（使转发表不用记录每个具体目的IP地址，只需要记录一个类似223.1.1.0/24一样的带前缀的地址即可）



解：我不同意这种说法，IPv4的原理是用一个IPv4数据报把IPv6数据报封装起来，IPv6数据报相当于封装后数据报的有效载荷。这个隧道应该属于网络层协议。



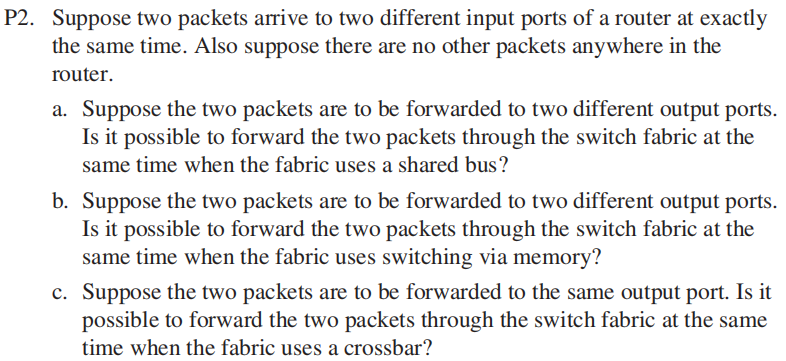
解：“匹配加动作”是指路由器或交换机试图在流表中有某些条目的包的一些报头值之间找到匹配，然后根据该匹配，路由器决定将该数据包转发到哪个接口，甚至决定对该分组进行更多的操作。在基于目的地的转发分组交换机的情况下，路由器只尝试在流表条目与到达数据包的目标P地址之间找到匹配，并且决定将该数据包转发到哪个接口。在SDN的情况下，可以匹配许多字段，例如IP源地址、TCP源端口和源MAC地址；还可以采取许多操作，例如转发、删除和修改字段值。



解：

a. 目的地：H3，接口：3。

b. 不存在这样的转发表，因为转发表只由目的地决定。

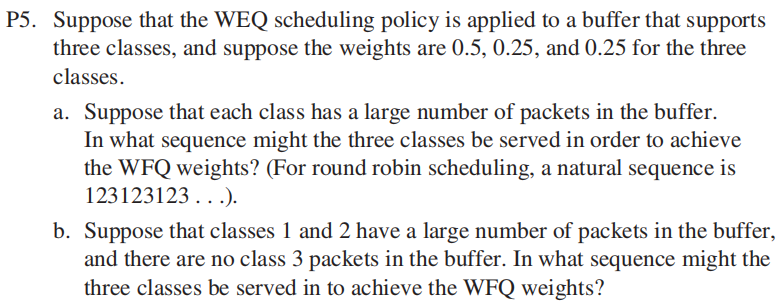


解：

a. 不能。

b. 不能。

c. 不能。

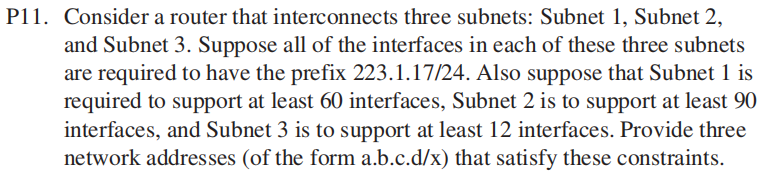


解：

a.

|  |  |
| --- | --- |
| 前缀 | 接口 |
| 11100000 00 | 0 |
| 11100000 01000000 | 1 |
| 1110000 | 2 |
| 11100001 1 | 3 |
| 其他 | 3 |

b. 3/2/3。



解：

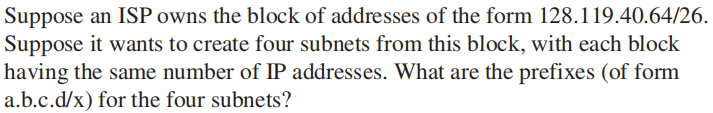
128.119.40.129

128.119.40.64/28

128.119.40.80/28

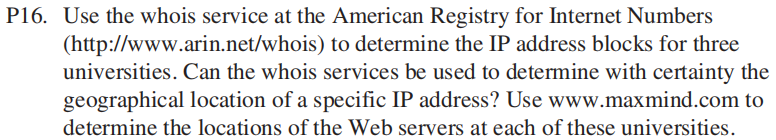
128.119.40.96/28

128.119.40.112/28



解：四个分片：

|  |  |  |
| --- | --- | --- |
| 标识 | 标志 | 片偏移 |
| 422 | 1 | 0 |
| 422 | 1 | 85 |
| 422 | 1 | 170 |
| 422 | 0 | 225 |

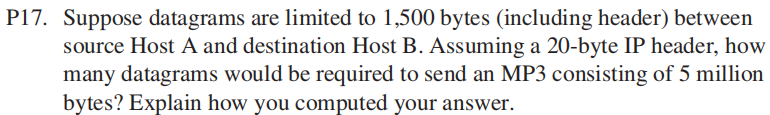


解：

a. 192.168.1.1；192.168.1.2；192.168.1.3；192.168.1.4。

b.

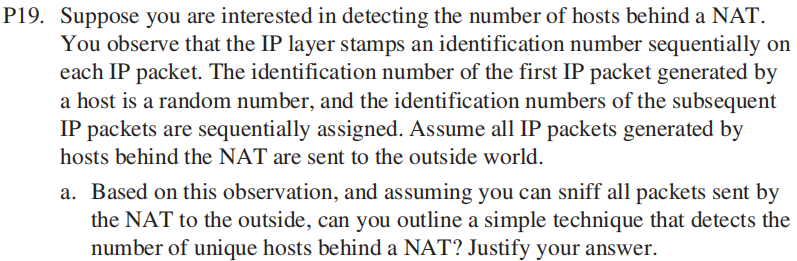
|  |  |
| --- | --- |
| WAN | LAN |
| 24.34.112.235,4000 | 192.168.1.1,3345 |
| 24.34.112.235,4001 | 192.168.1.1,3346 |
| 24.34.112.235,4002 | 192.168.1.2,3445 |
| 24.34.112.235,4003 | 192.168.1.2,3446 |
| 24.34.112.235,4004 | 192.168.1.3,3545 |
| 24.34.112.235,4005 | 192.168.1.3,3546 |



解：

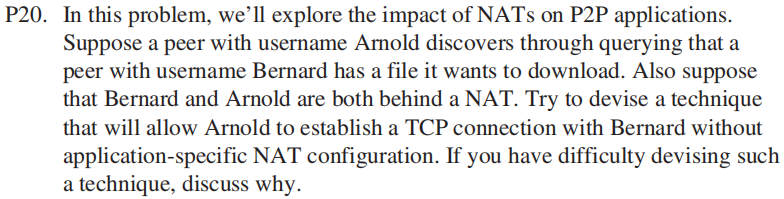
a. 在一段时间内收集从该NAT发出的所有IP分组，检查其中有多少分组的集合，满足在该集合内分组的序号是连续的。这些集合的数量即为主机的数量。

b. 不能正常工作。



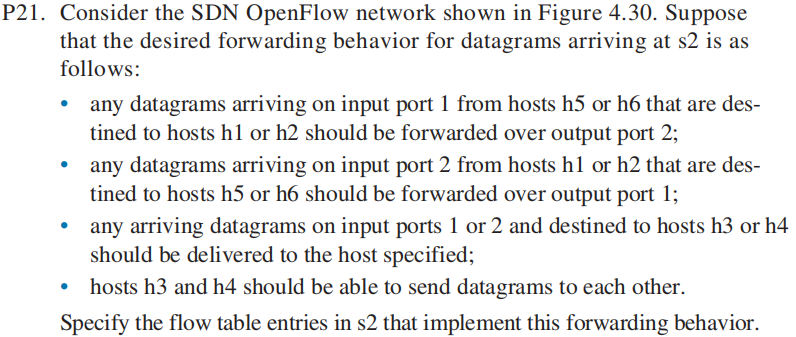
解：

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Ingress Port=1;Src=10.3.\*.\* ;Dst=10.1.\*.\*; | Forward(2) |
| Ingress Port=2;Src=10.1.\*.\*;Dst=10.3.\*.\*; | Forward(1) |
| Ingress Port=1;Dst=10.2.0.3; | Forward(3) |
| Ingress Port=2;Dst=10.2.0.3; | Forward(3) |
| Ingress Port=1;Dst=10.2.0.4; | Forward(4) |
| Ingress Port=2;Dst=10.2.0.4; | Forward(4) |
| Ingress Port=3 | Forward(4) |
| Ingress Port=4 | Forward(3) |



解：

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Ingress Port=3;Dst=10.1.\*.\* | Forward(2) |
| Ingress Port=3;Dst=10.3.\*.\* | Forward(2) |
| Ingress Port=4;Dst=10.1.\*.\* | Forward(1) |
| Ingress Port=4;Dst=10.3.\*.\* | Forward(1) |



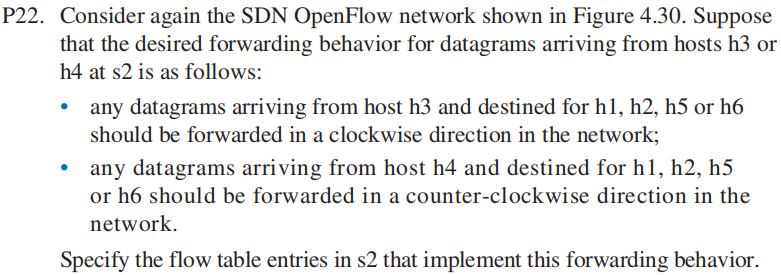
解：

s1：

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Src=10.2.\*.\*;Dst=10.1.0.1; | Forward(2) |
| Src=10.2.\*.\*;Dst=10.1.0.2; | Forward(3) |
| Src=10.2.\*.\*;Dst=10.3.\*.\*; | Forward(1) |

s3：

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Src=10.2.\*.\*;Dst=10.3.0.5; | Forward(2) |
| Src=10.2.\*.\*;Dst=10.3.0.6; | Forward(1) |
| Src=10.2.\*.\*;Dst=10.1.\*.\*; | Forward(3) |



解：

a.

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Src=10.1.0.1;Dst=10.2.0.3; | Forward(3) |
| Src=10.3.0.6;Dst=10.2.0.3; | Forward(3) |
| Src=10.1.0.1;Dst=10.2.0.4; | Forward(4) |
| Src=10.3.0.6;Dst=10.2.0.4; | Forward(4) |

b.

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Dst=10.2.0.3;port=TCP | Forward(3) |
| Dst=10.2.0.4;port=TCP | Forward(4) |

c.

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Dst=10.2.0.3 | Forward(3) |

d.

|  |  |
| --- | --- |
| 匹配 | 动作 |
| Src=10.1.0.1;Dst=10.2.0.3;port=UDP; | Forward(3) |