P4、

a)文档请求是`http://gaia.cs.umass.edu/cs453/index.html`。

host：字段表示服务器的名称，`/cs453/index.html`表示文件名。

b)浏览器正在运行HTTP1.1版本，就像第一对`<cr><lf>`之前所指出的那样。

c)浏览器正在请求持久连接，如连接所示：keep-alive。

d)这是刁钻的问题。此信息不包含在任何地方的HTTP消息中。因此，仅看HTTP消息的交换就无法区分这一点。您需要从IP数据报(承载承载HTTP GET请求的TCP段)中获得信息来回答这个问题。

e)Mozilla/5.0。服务器需要浏览器类型信息将同一对象的不同版本发送到不同类型的浏览器。

P7、获取IP地址的总时间为$\mathrm{RTT}_1 + \mathrm{RTT}_2 + \ldots + \mathrm{RTT}_n$。

一旦知道IP地址，RTT_0就会以建立TCP连接消逝，而另一个RTT_0则会以请求和接收小对象消逝。总响应时间为$2\mathrm{RTT}_0 + \mathrm{RTT}_1 + \mathrm{RTT}_2 + \ldots + \mathrm{RTT}_n$。

P8、

a)$\mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n + 2\mathrm{RTT}_0 + 8 \cdot 2\mathrm{RTT}_0$
$= 18\mathrm{RTT}_0 + \mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n$

b)$\mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n + 2\mathrm{RTT}_0 + 2 \cdot 2\mathrm{RTT}_0$
$= 6\mathrm{RTT}_0 + \mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n$

c)与流水线(pipelining)的持久连接。这是HTTP的默认模式。

$\mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n + 2\mathrm{RTT}_0 + \mathrm{RTT}_0$
$= 3\mathrm{RTT}_0 + \mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n$。

持续连接，没有流水线，没有并行连接。

$\mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n + 2\mathrm{RTT}_0 + 8\mathrm{RTT}_0$
$= 10\mathrm{RTT}_0 + \mathrm{RTT}_1 + \ldots + \mathrm{RTT}_n$

P13、SMTP中的`MAIL FROM:`的邮件是从SMTP客户端发送的标识发件人的邮件将邮件消息发送到SMTP服务器。

From：在邮件消息本身不是SMTP邮件，而不是邮件正文中的一行。

18 此题为开放性题目，没有标准答案，言之有理即可。

P18、

a)对于给定的域名输入(如`ccn.com`)、IP地址或网络管理员名称，whois数据库可用于定位相应的注册服务器、whois服务器、DNS服务器等。

f)攻击者可以使用whois数据库和nslookup工具来确定目标机构的IP地址范围、DNS服务器地址等。

g)通过分析攻击数据包的源地址，受害者可以使用whois获取攻击所来自的域的信息，并可能通知源域的管理员。

P23、

a)考虑一种分发方案，其中服务器以u_s/N的速率并行地向每个客户端发送文件，注意此速率低于客户端的每个下载速率，因为我们耗时$u_s/N \le d_{min}$。因此，每个客户端也可以以u_s/N的速率接收，因为每个客户端以u_s/N的速率接收，每个客户端接收整个文件的时间是$F/(u_s/N) = NF/u_s$。由于所有客户端都在NF/u_s中接收文件，所以总的分发时间也是NF/u_s。

b)考虑一种分发方案，其中服务器并行地向每个客户端发送文件，速率为d_{min}。注意，聚合速率$N\ d_{min}$小于服务器的链接速率我们，因为假设我们$u_s/N \ge d_{min}$。由于每个客户端以d_{min}的速率接收，每个客户端接收整个文件的时间是F/d_{min}。因为所有客户端都在这段时间内接收文件，所以整个分发时间也是F/d_{min}。

c)从2.6节我们知道

$D_{CS} >= \max\{NF/u_s, F/d_{min}\}$公式1

假设$u_s/N \le d_{min}$。那么我们可以从公式1得到 $D_{CS} \ge NF/u_s$。但是我们从a)可以得到$D_{CS} \le NF/u_s$。

结合两个公式：当$u_s/N <= d_{min}$公式2，$D_{CS} = NF/u_s$

那么当$u_s/N >= d_{min}$，可以得到$D_{CS} = F/d_{min}$公式3

结合公式2和3得到结果

P24、

a)定义$u = u_1 + u_2 + ... + u_N$。假设$u_s <= (u_s + u)/N$公式1

将文件划分为N个部分，第i部分具有大小$(u_i/u)F$。服务器传输第i部分至节点(peer)i的速率$r_i = (u_i/u)u_s$。请注意，$r_1 + r_2 + ... + r_N = u_s$，聚合服务器速率不超过服务器的链路速率。也具有每个节点i以速率r_i将其接收的比特转发到$N-1$个对等点中的每一个。最大节点i的转发速率是$(N-1)r_i$。我们有

$(N-1)r_i = (N-1)(u_s u_i)/u <= u_i$

其中最后一个不等式来自上一个方程。因此，节点i的总转发速率小于它的链路速率u_i。

在此分配方案中，节点i以合计速率接收比特

$r_i + \sum_{j<>i} r_j = u_s$

因此，每个节点以F/u_s接收文件。

b)再定义$u = u_1 + u_2 + ... + u_N$。假设$u_s >= (u_s + u)/N$ 公式2

定义$r_i = u_i/(N-1)$和$r_{N+1} = (u_s - u/(N-1))/N$

在此分发方案中，文件被分成N+1个部分。服务器发送位从第i部分到第i个节点(i=1，...N)在速率r_i。每个节点i以速率r_i转发该比特到达其它$N-1$节点中的每一个。此外，服务器以速率r_{N+1}发送位来自$(N+1)^{st}$部分到达N个节点中的每一个。节点不转发来自$(n+1)^{st}$部分。

服务器的聚合发送速率

$$r_1 + \ldots + r_N + N r_{N+1} = u/(N-1) + u_s - u/(N-1) = u_s$$

因此，服务器的发送速率不超过其链路速率。节点i的总发送速率为

$$(N-1)r_i = u_i$$

因此，每个对等点的发送速率不超过其链路速率。在此分配方案中，对等点i以合计速率接收比特。

$$r_i + r_{N+1} + \sum_{j<>i} rj = u/(N-1) + (u_s - u/(N-1))/N = (u_s + u)/N$$

因此，每个节点以$NF/(u_s + u)$速率接收文件。

（为了简单起见，我们忽略了为$i = 1, \ldots, n+1$文件部分的大小。我们现在在这里提供了。假设$\Delta = (u_s + u)/N$为分配时间。对于$i = 1, \ldots, N$，第i个文件部分是$F_i = r_i \Delta b$。$(N+1)^{st}$文件部分是$F_{N+1} = r_{N+1}\Delta b$。它是直接显示$F_1 + \ldots F_{N+1} = F$。）

c)本部的解决办法与17©的解决办法相似。我们从2.6节中得知

$$D_{P2P} >= \max\{F/u_s, NF/(u_s + u)\}$$

结合了a)和b)给出了预期的结果。

P26、是的。只要有足够的对等人留在蜂群中，他的第一个声明就可能是可能的。Bob可以总是通过乐观的非阻塞来接收数据对等端。

他的第二次声明也是真实的。他可以在每个主机上运行客户端，让每个客户端"免费乘坐，"并将收集的chunks从不同的主机合并为单个文件。他甚至可以编写一个小的调度程序，使不同的主机请求不同的chunks文件。这实际上是P2P网络中的一种Sybil攻击。

批改后作业链接：
https://web.ugreen.cloud/web/#/share/dd5acb14d9af42059cdfff7aec9613ab 提取码：5J65