# Wireshark 实验: DNS

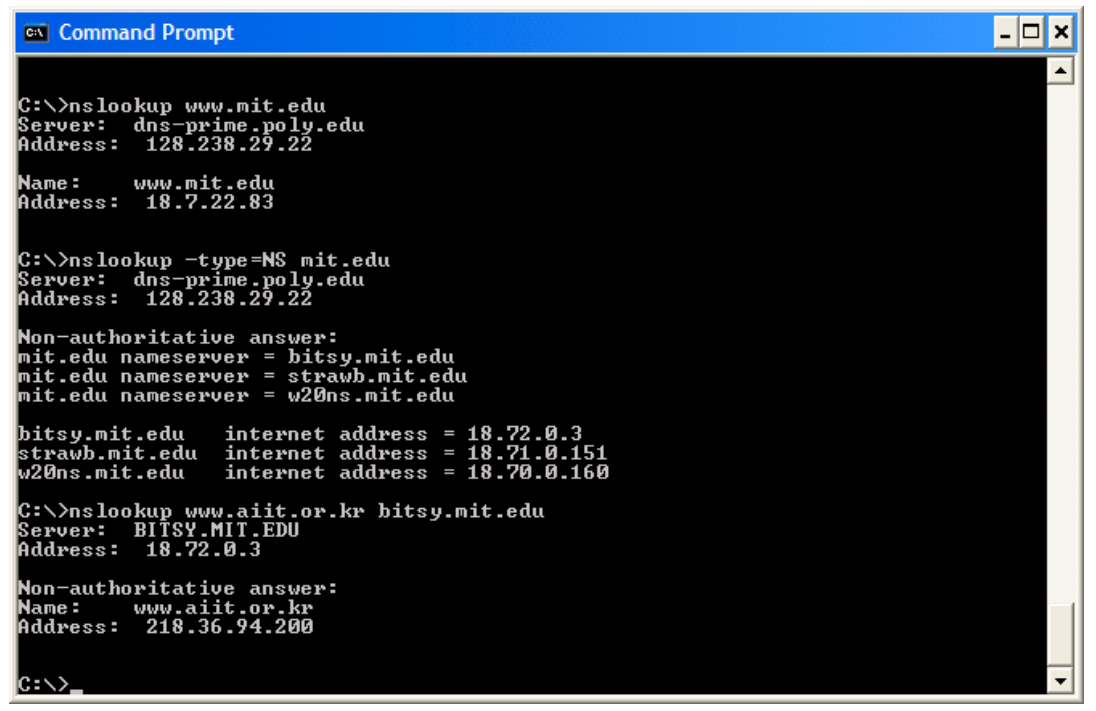
如书中第2.5节所述，域名系统(DNS)将主机名转换为IP地址，在互联网基础架构中发挥关键作用。在本实验中，我们将仔细查看DNS在客户端的细节。回想一下，客户端在DNS中的角色相对简单——客户端向其本地DNS服务器发送*请求*，并接收一个*响应*。如书中的图2.21和2.22所示，由于DNS分层服务器之间相互通信，可以递归地或迭代地解析客户端的DNS查询请求，而大多数操作是不可见的。然而，从DNS客户端的角度来看，协议非常简 ——将查询指向为本地DNS服务器，并从该服务器接收到响应。

在开始本实验之前，您可能需要阅读书中的第2.5节来了解DNS。另外，您可能需要查看关于**本地DNS服务器，DNS缓存，DNS记录和消息，以及DNS记录中的TYPE字段**的资料。

**1. nslookup**

在本实验中，我们将大量使用*nslookup*工具，这个工具在现在的大多数Linux/Unix和Microsoft平台中都有。要在Linux/Unix中运行*nslookup*，您只需在命令行中键入*nslookup*命令即可。要在Windows中运行，请打开命令提示符并在命令行上运行*nslookup*。

在这是最基本的操作，*nslookup*工具允许主机查询任何指定的DNS服务器的DNS记录。DNS服务器可以是根DNS服务器，顶级域DNS服务器，权威DNS服务器或中间DNS服务器（有关这些术语的定义，请参阅书本）。要完成此任务，*nslookup*将DNS查询发送到指定的DNS服务器，然后接收DNS回复，并显示结果。

[](https://github.com/moranzcw/Computer-Networking-A-Top-Down-Approach-NOTES/blob/master/WiresharkLab/Wireshark%E5%AE%9E%E9%AA%8C-DNS/image/%E5%9B%BE1.png)

上面的屏幕截图显示了三个不同*nslookup*命令的结果（显示在Windows命令提示符中）。在此示例中，客户端主机位于布鲁克林理工大学校园，默认本地DNS服务器为dns-prime.poly.edu。运行nslookup时，如果没有指定DNS服务器，则*nslookup*会将查询发送到默认的DNS服务器（在这种情况下为dnsprime.poly.edu）。来看第一个命令：

nslookup www.mit.edu

说这个命令是说，请告诉我主机 [www.mit.edu](http://www.mit.edu/) 的IP地址。如屏幕截图所示，此命令的响应提供两条信息：

（1）提供响应的DNS服务器的名称和IP地址；

（2）响应本身，即 [www.mit.edu](http://www.mit.edu/) 的主机名和IP地址。虽然响应来自理工大学的本地DNS服务器，但本地DNS服务器很可能会迭代地联系其他几个DNS服务器来获得结果，如书中第2.4节所述。

现在来看第二个命令：

nslookup -type=NS mit.edu

在这个例子中，我们添加了选项"-type=NS"和域名"mit.edu"。这将使得*nslookup*将NS记录发送到默认的本地DNS服务器。换句话说，“请给我发送mit.edu的权威DNS的主机名” （当不使用-type选项时，*nslookup*使用默认值，即查询A类记录。）上述屏幕截图中，首先显示了提供响应的DNS服务器（这是默认本地DNS服务器）以及三个MIT域名服务器。这些服务器中的每一个确实都是麻省理工学院校园主机的权威DNS服务器。然而，*nslookup*也表明该响应是非权威的，这意味着这个响应来自某个服务器的缓存，而不是来自权威MIT DNS服务器。最后，响应结果还显示了麻省理工学院权威DNS服务器的IP地址。 （即使*nslookup*生成的NS类型查询没有明确要求IP地址，本地DNS服务器依然”免费“返回了这些信息，然后被*nslookup*显示出来。）

最后来看第三个命令：

nslookup www.aiit.or.kr bitsy.mit.edu

在这个例子中，我们希望将查询请求发送到DNS服务器 bitsy.mit.edu ，而不是默认的DNS服务器（dns-prime.poly.edu）。因此，查询和响应事务直接发生在我们的主机和 bitsy.mit.edu 之间。在这个例子中，DNS服务器 bitsy.mit.edu 提供主机 [www.aiit.or.kr](http://www.aiit.or.kr/) 的IP地址，它是高级信息技术研究所（韩国）的Web服务器。

现在我们了解了一些示例， *nslookup*命令的一般语法是：

nslookup -option1 -option2 host-to-find dns-server

一般来说，*nslookup*可以不添加选项，或者添加一两个甚至更多选项。正如我们在上面的示例中看到的，dns-server也是可选的；如果这项没有提供，查询将发送到默认的本地DNS服务器。

现在我们提供了总览了*nslookup*，现在是你自己驾驭它的时候了。执行以下操作（并记下结果）：

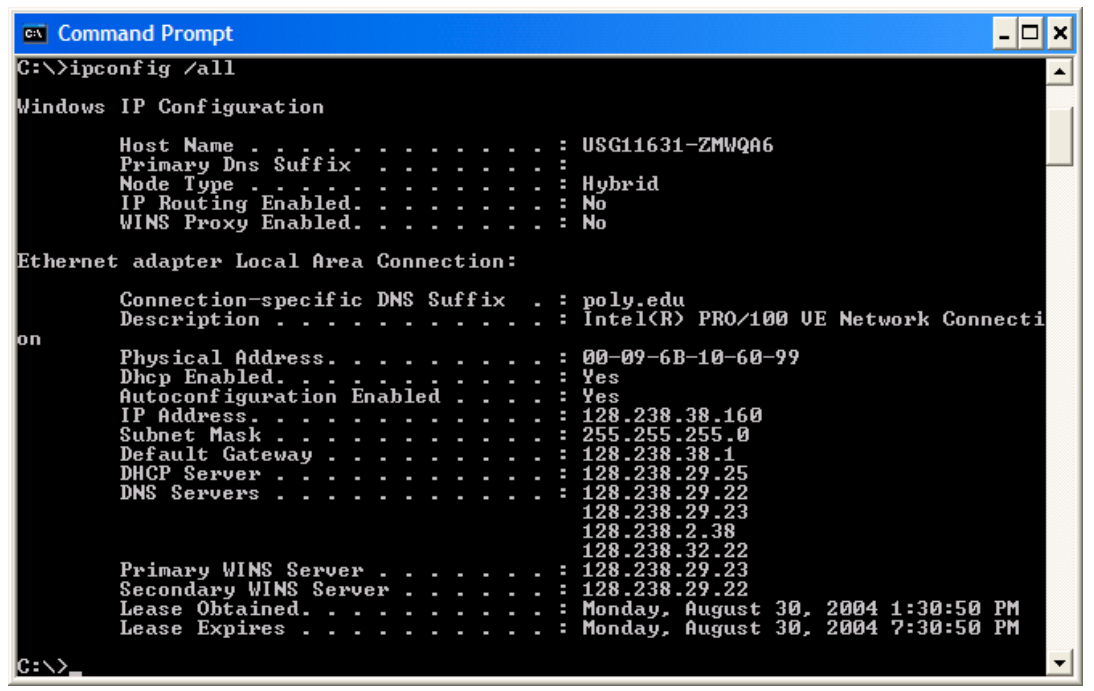
1. 运行*nslookup*以获取一个亚洲的Web服务器(xxb.ecust.edu.cn)的IP地址。该服务器的IP地址是什么？
2. 运行*nslookup*来确定一个中国的大学（www.ecust.edu.cn）的权威DNS服务器。
3. 运行*nslookup*，使用问题2中一个已获得的DNS服务器，来查询sina邮箱的邮件服务器。它的IP地址是什么？

**2. ipconfig**

*ipconfig*（对于Windows）和*ifconfig*（对于Linux / Unix）是主机中最实用的程序，尤其是用于调试网络问题时。这里我们只讨论*ipconfig*，尽管Linux / Unix的*ifconfig*与其非常相似。 *ipconfig*可用于显示您当前的TCP/IP信息，包括您的地址，DNS服务器地址，适配器类型等。例如，您只需进入命令提示符，输入

ipconfig /all

所有关于您的主机信息都类似如下面的屏幕截图所显示。

[](https://github.com/moranzcw/Computer-Networking-A-Top-Down-Approach-NOTES/blob/master/WiresharkLab/Wireshark%E5%AE%9E%E9%AA%8C-DNS/image/%E5%9B%BE2.png)

*ipconfig*对于管理主机中存储的DNS信息也非常有用。在第2.5节中，我们了解到主机可以缓存最近获得的DNS记录。要查看这些缓存记录，在 C:\> 提示符后输入以下命令：

ipconfig /displaydns

每个条目显示剩余的生存时间（TTL）（秒）。要清除缓存，请输入

ipconfig /flushdns

清除了所有条目并从hosts文件重新加载条目。

**3. 使用Wireshark追踪DNS**

现在，我们熟悉nslookup和ipconfig，我们准备好了一些正经的事情。首先让我们捕获一些由常规上网活动生成的DNS数据包。

* 使用ipconfig清空主机中的DNS缓存。
* 打开浏览器并清空浏览器缓存。 （若使用Internet Explorer，转到工具菜单并选择Internet选项；然后在常规选项卡中选择删除文件。）
* 打开Wireshark，然后在过滤器中输入“ip.addr==your\_IP\_address”，您可以先使用ipconfig获取你的IP地址。此过滤器将删除既从你主机不发出也不发往你主机的所有数据包。
* 在Wireshark中启动数据包捕获。
* 使用浏览器访问网页： [http://www.ietf.org](http://www.ietf.org/)
* 停止数据包捕获。

回答下列问题。

4、找到DNS查询和响应消息。它们是否通过UDP或TCP发送？

5、DNS查询消息的目标端口是什么？ DNS响应消息的源端口是什么？

6、DNS查询消息发送到哪个IP地址？使用ipconfig来确定本地DNS服务器的IP地址。这两个IP地址是否相同？

7、检查DNS查询消息。DNS查询是什么"Type"的？查询消息是否包含任何"answers"？

8、检查DNS响应消息。提供了多少个"answers"？这些答案具体包含什么？

\*9、考虑从您主机发送的后续TCP SYN数据包。 SYN数据包的目的IP地址是否与DNS响应消息中提供的任何IP地址相对应？

10、这个网页包含一些图片。在获取每个图片前，您的主机是否都发出了新的DNS查询？

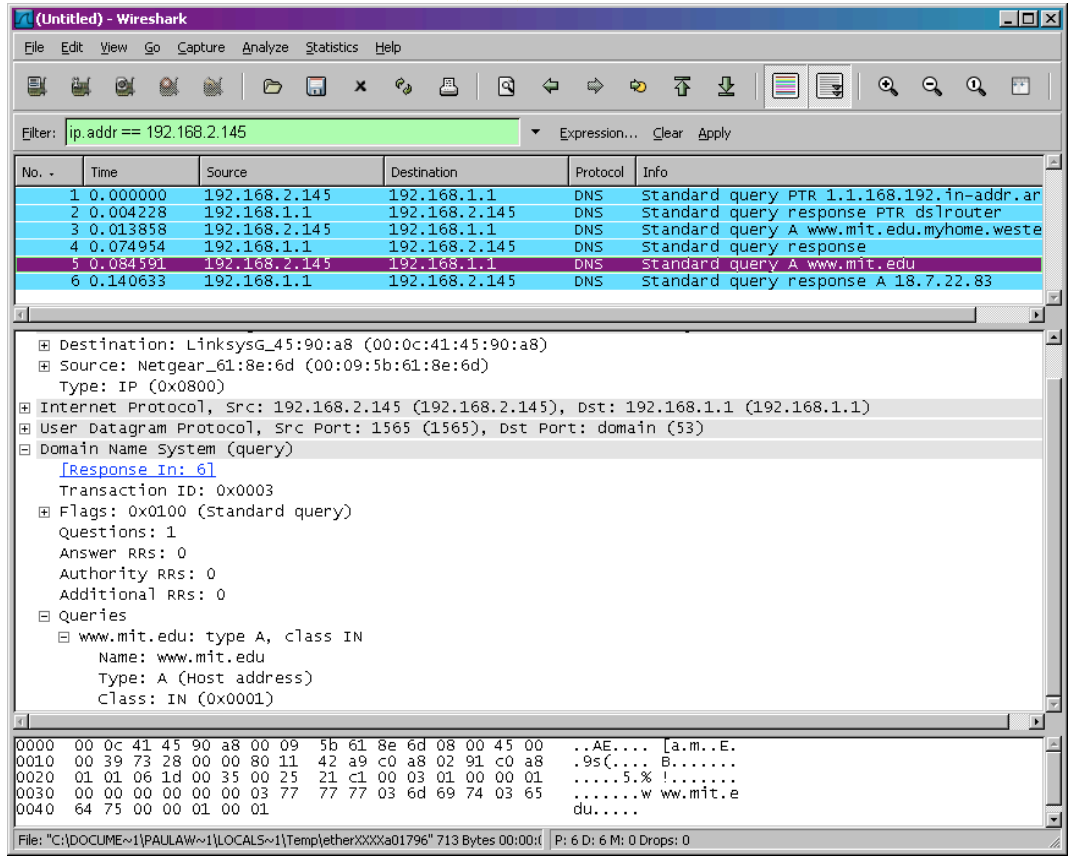
现在让我们玩玩*nslookup*

启动数据包捕获。

使用nslookup查询 [www.mit.edu](http://www.mit.edu/)

停止数据包捕获。

你应该得到类似下图所示的捕获结果：

[](https://github.com/moranzcw/Computer-Networking-A-Top-Down-Approach-NOTES/blob/master/WiresharkLab/Wireshark%E5%AE%9E%E9%AA%8C-DNS/image/%E5%9B%BE3.png)

我们从上面的屏幕截图看到，*nslookup*实际上发送了三个DNS查询，并收到了三个DNS响应。只考虑本次实验相关结果，在回答以下问题时，请忽略前两组查询/响应，因为*nslookup*的一些特殊性，这些查询通常不是由标准网络应用程序生成的。您应该专注于最后一个查询和响应消息。

1. DNS查询消息的目标端口是什么？ DNS响应消息的源端口是什么？
2. DNS查询消息的目标IP地址是什么？这是你的默认本地DNS服务器的IP地址吗？
3. 检查DNS查询消息。DNS查询是什么"Type"的？查询消息是否包含任何"answers"？
4. 检查DNS响应消息。提供了多少个"answers"？这些答案包含什么？
5. 提供屏幕截图。

现在重复上一个实验，但换成以下命令：

nslookup -type=NS mit.edu

回答下列问题：

1. DNS查询消息发送到的IP地址是什么？这是您的默认本地DNS服务器的IP地址吗？
2. 检查DNS查询消息。DNS查询是什么"Type"的？查询消息是否包含任何"answers"？
3. 检查DNS响应消息。响应消息提供的MIT域名服务器是什么？此响应消息还提供了MIT域名服务器的IP地址吗？
4. 提供屏幕截图。

现在重复上一个实验，但换成以下命令：

nslookup www.aiit.or.kr bitsy.mit.edu

回答下列问题：

1. DNS查询消息发送到的IP地址是什么？这是您的默认本地DNS服务器的IP地址吗？如果不是，这个IP地址是什么？
2. 检查DNS查询消息。DNS查询是什么"Type"的？查询消息是否包含任何"answers"？
3. 检查DNS响应消息。提供了多少个"answers"？这些答案包含什么？
4. 提供屏幕截图。

补充练习：

1. 对百度进行一次A记录查询  
   用8.8.8.8这个dns服务器查询 http://baidu.com 这个域名的ns记录
2. 用系统默认的DNS服务器进行查询（此时默认的DNS并不是http://baidu.com这个域名的默认DNS）
3. 用百度的默认dns（dns.baidu.com）进行查询

请描述步骤2和步骤3中wireshark捕捉到的DNS的应答报文有什么不一样