

DNS und HTTP

DNS (Domain Name System) bezeichnet das System und auch Protokoll, mit dem Domainnamen in IP-Adressen übersetzt werden -- und mit dem noch mehr getan werden kann. HTTP (Hypertext Transfer Protocol) wird zur Übertragung von Webseiten verwendet.

Hintergrund: DNS

In einem typischen Netzwerk kontaktiert Ihr Computer einen lokalen DNS-Nameserver, um Domännennamen in IP-Adressen aufzulösen. Der lokale Nameserver kann ein anderer Computer in Ihrem Firmennetzwerk, ein Computer bei Ihrem ISP oder Ihr drahtloser Access Point sein. Es tauscht eine Reihe von Nachrichten mit entfernten DNS-Nameservern im gesamten Internet aus, um die Auflösung durchzuführen. Der Aufbau ist wie in der folgenden Abbildung dargestellt:

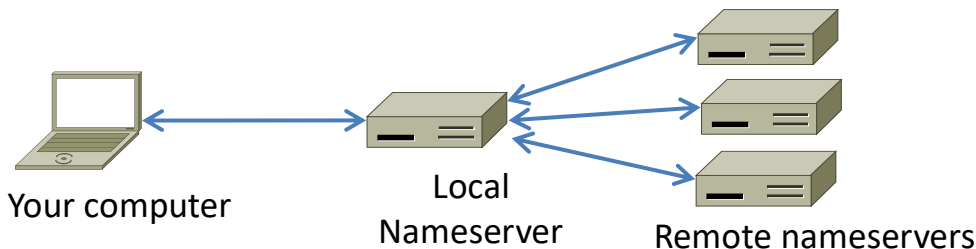


Abbildung 1: Typische Netzwerkconfiguration für DNS

Dies hat eine wichtige Implikation: der Trace, den wir an unserem Computer sammeln, zeigt den Austausch zwischen unserem Computer und dem lokalen Nameserver, aber nicht zwischen dem lokalen Nameserver und den entfernten Nameservern.

Aufgabe 1: Manuelle Namensauflösung

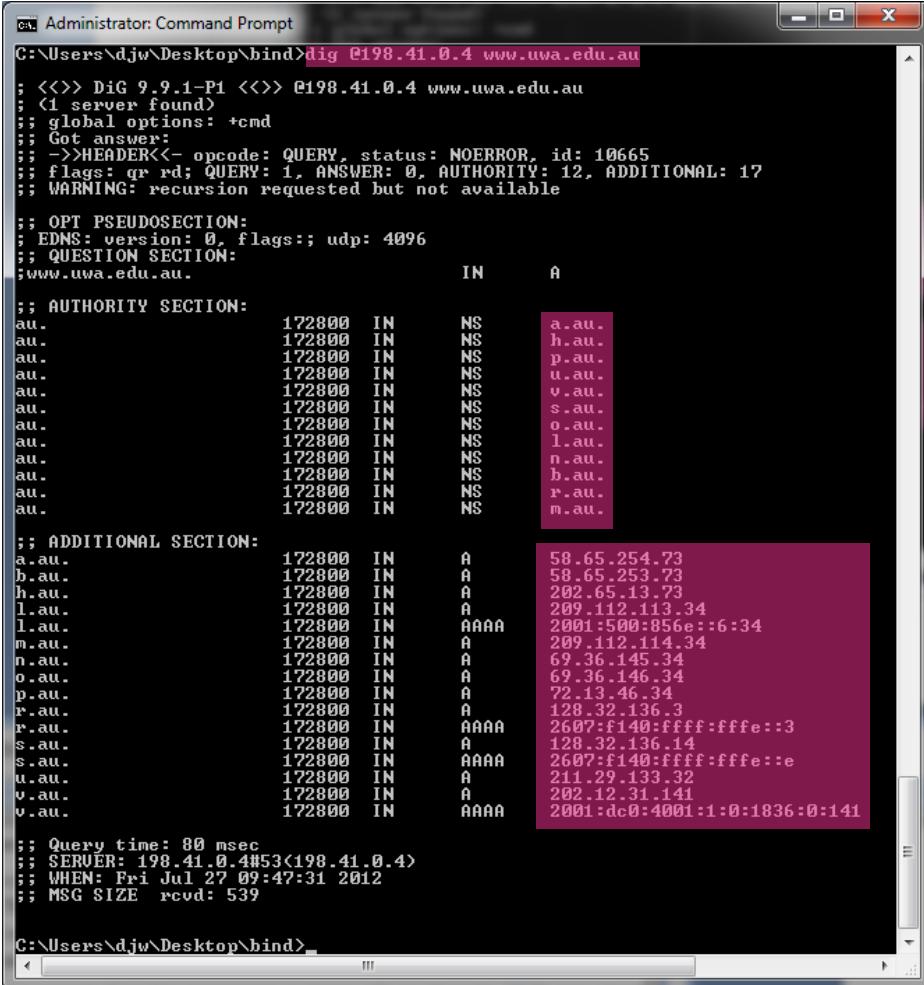
In dieser Übung tun Sie als wären Sie der lokale Nameserver und stellen Anfragen an entfernte Nameserver. Verwenden Sie für diese Aufgabe das Tool `dig` (normalerweise unter Linux vorinstalliert, für Windows als Download verfügbar) oder verwenden Sie eine Web-Version wie

<https://www.digwebinterface.com>.

Wählen Sie einen Domännennamen aus, den Sie auflösen möchten, z.B. den eines Webserver. Ein Beispiel aus unserer Fakultät, das Sie wählen können, ist `magik-demo.inf.unibz.it`.

Finden Sie die IP-Adresse von einem der Root-Nameserver, indem Sie im Web suchen. Senden sie als ersten Schritt der Auflösung eine Anfrage an einen dieser Root-Nameserver. Nehmen Sie an, dass Sie keine Informationen im Cache haben, mit denen Sie eine Auflösung unterhalb der Wurzel beginnen können.

Das Format eines `dig`-Befehls ist „`dig @aa.bb.cc.dd domain-name`“. Er weist `dig` an, eine Anfrage an den Nameserver `domain-name` unter der IP-Adresse (oder dem Namen) `aa.bb.cc.dd` nach dem angegebenen Domain-Namen zu senden. Die folgende Abbildung zeigt eine Antwort von `dig` auf die Anfrage „`dig @198.41.0.4 www.uwa.edu.au`“. Sie richtet sich an den Root-Name-Server `a.root-servers.net`. Die Antwort von der Wurzel des Namensraums liefert nicht die volle Namensauflösung, aber sie liefert uns die Nameserver, die näher an der Information sind, die wir suchen. Im Fall unseres Beispielproblems, der Adresse von `www.uwa.edu.au`, sind es Nameserver, die von der „.au“-Domäne wissen. Mehrere Nameserver werden als Alternative angegeben, und die Antwort enthält ihre IP-Adressen; wir können sowohl IPv6-Adressen als auch IPv4-Adressen sehen.



```
Administrator: Command Prompt
C:\Users\djw\Desktop\bind>dig @198.41.0.4 www.uwa.edu.au

;; <<>> DiG 9.9.1-P1 <<>> @198.41.0.4 www.uwa.edu.au
;; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10665
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 12, ADDITIONAL: 17
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 4096
;; QUESTION SECTION:
;www.uwa.edu.au.                        IN      A

;; AUTHORITY SECTION:
au.                172800 IN      NS      a.au.
au.                172800 IN      NS      h.au.
au.                172800 IN      NS      p.au.
au.                172800 IN      NS      u.au.
au.                172800 IN      NS      v.au.
au.                172800 IN      NS      s.au.
au.                172800 IN      NS      o.au.
au.                172800 IN      NS      l.au.
au.                172800 IN      NS      n.au.
au.                172800 IN      NS      b.au.
au.                172800 IN      NS      r.au.
au.                172800 IN      NS      m.au.

;; ADDITIONAL SECTION:
a.au.                172800 IN      A      58.65.254.73
b.au.                172800 IN      A      58.65.253.73
h.au.                172800 IN      A      202.65.13.73
l.au.                172800 IN      A      209.112.113.34
l.au.                172800 IN      AAAA   2001:500:856e::6:34
m.au.                172800 IN      A      209.112.114.34
n.au.                172800 IN      A      69.36.145.34
o.au.                172800 IN      A      69.36.146.34
p.au.                172800 IN      A      72.13.46.34
r.au.                172800 IN      A      128.32.136.3
r.au.                172800 IN      AAAA   2607:f140:ffff:fffe::3
s.au.                172800 IN      A      128.32.136.14
s.au.                172800 IN      AAAA   2607:f140:ffff:fffe::e
u.au.                172800 IN      A      211.29.133.32
v.au.                172800 IN      A      202.12.31.141
v.au.                172800 IN      AAAA   2001:dc0:4001:1:0:1836:0:141

;; Query time: 80 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Fri Jul 27 09:47:31 2012
;; MSG SIZE rcvd: 539

C:\Users\djw\Desktop\bind>
```

Setzen Sie den Auflösungsprozess mit `dig` fort, bis Sie die Auflösung abgeschlossen haben. Wenn Sie Alternativen zur Auswahl haben, wählen Sie IPv4-Nameserver und nehmen Sie den ersten in alphabetischer Reihenfolge. Wenn dieser Nameserver mehrere IP-Adressen hat, dann wählen Sie die numerisch kleinste IP-Adresse.

Aufgabe: Fertigen Sie eine Zeichnung an, die die Reihenfolge der von Ihnen kontaktierten Remote-Nameserver und die Domäne, für die sie verantwortlich sind, anzeigt.

Aufgabe 2: Aufzeichnen eines HTTP-Trace

Zeichnen Sie ein Trace Ihres Browsers auf, wie er HTTP-Anfragen stellt:

1. Verwenden Sie Ihren Browser, um zwei URLs zu finden, mit denen Sie experimentieren können. Beide sollten HTTP-URLs (nicht HTTPS-URLs) ohne speziellen Port sein. Die erste URL sollte die eines kleinen bis mittleren Bildes sein, da wir an einfachen statischen Daten interessiert sind. Die zweite URL sollte die Startseite einer großen Website sein, die Sie untersuchen möchten. Sie wird vergleichsweise komplex sein.
2. Bereiten Sie Ihren Rechner vor, indem Sie die HTTP-Aktivität zurückfahren und den Browser-Cache leeren. Abgesehen von einem neuen Tab, das Sie verwenden werden, schließen Sie alle anderen Tabs und Anwendungen, um den HTTP-Verkehr zu minimieren.
3. Starten Sie Wireshark und beginnen Sie ein Capture mit dem Filter „tcp port 80“. Wir verwenden diesen Filter, weil es für HTTP keinen speziellen Filter gibt, aber HTTP normalerweise über den TCP-Port 80 führt.
4. Laden Sie nacheinander Daten von den URLs in der gegebenen Reihenfolge herunter, nachdem Sie einen Moment gewartet haben, um zu sicherzustellen, dass kein HTTP-Verkehr mehr läuft. Wenn es HTTP-Verkehr gibt, finden Sie die Anwendung, die ihn verursacht, und schließen Sie sie. Andernfalls wird Ihr Trace zu viel HTTP-Verkehr enthalten, den Sie nicht verstehen können. Fügen Sie jede URL in die URL-Leiste des Browsers ein und drücken Sie die Eingabetaste, um sie herunterzuladen. Geben Sie die URL nicht manuell ein, da dies dazu führen kann, dass der Browser zusätzliche HTTP-Anfragen generiert, während er versucht, Ihre Eingabe automatisch zu vervollständigen.
 - a. Laden Sie die erste statische Bild-URL herunter, indem Sie die URL in die Browserleiste einfügen und „Enter“ drücken oder was auch immer erforderlich ist, um Ihren Browser zu starten.
 - b. Warten Sie 10 Sekunden und rufen Sie die statische Bild-URL erneut ab. Tun Sie dies auf die gleiche Weise, insbesondere ohne den „Reload“-Button Ihres Browsers, damit er kein anderes Verhalten bewirkt.
 - c. Warten Sie weitere 10 Sekunden und laden Sie die zweite Homepage-URL herunter.
 - d. Stoppen Sie die Aufzeichnung, nachdem die Abrufe abgeschlossen sind.

Aufgabe 3: Überprüfen des Trace

Damit Sie sich auf den HTTP-Verkehr konzentrieren können, geben Sie den Filter-Ausdruck „http“ ein und wenden ihn an. Dieser Filter zeigt HTTP-Requests und -Antworten an, nicht aber die einzelnen beteiligten Pakete. Erinnern Sie sich daran, dass eine HTTP-Antwort, die Inhalte enthält, normalerweise über mehrere Pakete verteilt ist. Wenn das letzte Paket in der Antwort ankommt, stellt Wireshark die komplette Antwort zusammen und markiert das Paket mit dem Protokoll HTTP. Die früheren Pakete sind einfach TCP-Segmente, die Daten enthalten; das letzte einem Paket wird mit dem Label HTTP versehen und enthält eine Liste aller früheren Pakete, die für die Antwort verwendet wurden. Ein ähnlicher Prozess findet für die Anfrage statt, aber hier ist es meistens der Fall, dass eine Anfrage in ein einzelnes Paket passt. Mit dem Filterausdruck von „http“ verstecken wir die zwischengeschalteten TCP-Pakete und sehen nur die HTTP-Anfrage und -Antwort.

Wählen Sie das erste GET im Trace aus und expandieren Sie dessen HTTP-Block. Dadurch können wir die Details einer HTTP-Anfrage überprüfen. Beachten Sie, dass der HTTP-Header dem TCP- und IP-Header folgt, da HTTP ein Anwendungsprotokoll ist, das über TCP/IP transportiert wird. Um es anzusehen, wählen Sie das Paket aus, suchen Sie den HTTP-Block in der Mitte und erweitern Sie ihn (mit dem "+" Expander oder Symbol).

Untersuchen Sie die Header, die mit der Anfrage gesendet werden. Zuerst sehen Sie die GET-Methode am Anfang der Anfrage, einschließlich Details wie dem Pfad. Dann sehen Sie eine Reihe von Überschriften in Form von getaggten Parametern. Es kann viele Header geben, und die Auswahl der Header und deren Werte variieren von Browser zu Browser. Sehen Sie nach, ob Sie eine dieser gemeinsamen Überschriften vor sich haben:

- Host. Ein Pflicht-Header, der den Namen (und Port) des Servers identifiziert.
- User-Agent. Die Art des Browsers und seine Funktionen.
- Accept, Accept-Encoding, Accept-Charset, Accept-Language. Beschreibungen der Formate, die in der Antwort akzeptiert werden, z.B. text/html, einschließlich ihrer Kodierung, z.B. gzip, und der Sprache.
- Cookie. Name und Wert der Cookies, die der Browser für die Website speichert.
- Cache-Control. Informationen darüber, wie die Antwort zwischengespeichert werden kann.

Die Anfrageinformationen werden in einem einfachen Text- und Zeilenformat gesendet. Wenn Sie in das untere Feld schauen, können Sie einen Großteil der Anfrage direkt aus dem Paket selbst lesen!

Wählen Sie die Antwort aus, die dem ersten GET im Trace entspricht, und expandieren Sie deren HTTP-Block. Die Info für dieses Paket sollte „200 OK“ sein. Sie werden sehen, dass die Antwort der Anfrage ähnlich ist, mit einer Reihe von Feldern, die dem Statuscode „200 OK“ folgen. Es werden jedoch verschiedene Felder verwendet, denen der gewünschte Inhalt folgt. Sehen Sie nach, ob Sie eine dieser gängigen Überschriften haben:

- Server Die Art des Servers und seine Funktionen.
- Date, Last-Modified. Die Zeit der Antwort und die Zeit der letzten Änderung des Inhalts.

- Cache-Control, Expires, Etag. Informationen darüber, wie die Antwort zwischengespeichert werden kann.

Beantworten Sie die folgenden Fragen:

- 1. Was ist das Format einer Headerzeile? Geben Sie eine einfache Beschreibung, die zu den angezeigten Headern passt.**
- 2. Welche Header werden verwendet, um die Art und Länge des Inhalts einer Antwort anzugeben?**

Aufgabe 4: Content Caching

Der zweite Abruf im Trace sollte ein Wiederabruf der ersten URL sein. Dieser Abruf ermöglicht uns, das Caching in Aktion zu beobachten, da es sehr wahrscheinlich ist, dass sich das Bild oder Dokument nicht verändert hat und daher nicht erneut heruntergeladen werden muss. HTTP-Caching-Mechanismen sollten diese Möglichkeit identifizieren. Wir werden nun sehen, wie sie funktionieren.

Wählen Sie das GET aus, das ein erneuter Abruf des ersten GET ist, und expandieren Sie dessen HTTP-Block. Finden Sie nun den Header, mit dem der Server herausfinden kann, ob er neue Inhalte senden muss. Der Server muss nur dann neue Inhalte senden, wenn sich der Inhalt seit dem letzten Herunterladen durch den Browser geändert hat. Um dies herauszufinden, fügt der Browser einen Zeitstempel ein, der vom vorherigen Download für den Inhalt, den er zwischengespeichert hat, abgelesen wurde. Dieses Feld war beim ersten GET nicht vorhanden, da wir den Browser-Cache geleert haben, so dass der Browser keinen vorherigen Download des Inhalts hatte, den er hätte verwenden können.

Wählen Sie schließlich die Antwort auf die Wiederabfrage und expandieren Sie deren HTTP-Block. Unter der Annahme, dass das Caching wie erwartet funktioniert hat, wird diese Antwort den Inhalt nicht enthalten. Stattdessen lautet der Statuscode der Antwort „304 Not Modified“. Dadurch wird dem Browser mitgeteilt, dass der Inhalt gegenüber der vorherigen Kopie unverändert ist und der zwischengespeicherte Inhalt angezeigt werden kann. Beantworten Sie die folgende Frage:

- 1. Wie lautet der Name des Header-Feldes, das der Browser sendet, damit der Server herausfinden kann, ob er neue Inhalte senden soll?**

Aufgabe 5: Komplexe Seiten

Betrachten wir nun den dritten Abruf am Ende des Trace. Dies war ein Abruf für eine komplexere Webseite, die wahrscheinlich eingebettete Ressourcen hat. Der Browser lädt also das ursprüngliche HTML plus alle eingebetteten Ressourcen, die zum Rendern der Seite benötigt werden, sowie weitere Ressourcen, die während der Ausführung von Page Scripts angefordert werden. Wie wir sehen werden, kann eine einzige Seite viele GETs enthalten!

Um die GETs für die dritte Seite zusammenzufassen, rufen Sie in Wireshark ein HTTP-Lastverteilungs-fenster auf. Sie finden dieses Fenster unter Statistics>HTTP.

In diesem Fenster sehen Sie, wie viele Anfragen an welche Server gestellt wurden. Es ist gut möglich, dass Ihr Abruf Inhalte von anderen Servern anfordert, von denen Sie nicht vermutet haben, dass sie die Seite erstellen. Zu diesen anderen Servern können auch Dritte wie Content-Distribution-Netzwerke, Werbenetzwerke und Analysenetze gehören.

Um die GETs auf eine andere Art zusammenzufassen, rufen Sie das HTTP Packet Counter-Fenster auf. Sie finden dieses Fenster auch unter *Statistics>HTTP*. Das Fenster informiert Sie über die Art der Anfragen und Antworten. Sie sind vielleicht neugierig, welche Inhalte von all diesen Anfragen heruntergeladen werden. Neben den URLs in der Spalte Info erhalten Sie eine Zusammenfassung der URLs in einem HTTP-Request-Panel unter „Statistics“ und „HTTP“. Jede der einzelnen Anfragen und Antworten hat die gleiche Form, die wir in einem früheren Schritt gesehen haben. Gemeinsam werden sie beim Abrufen einer kompletten Seite mit einer bestimmten URL ausgeführt.

Für einen detaillierteren Blick auf den gesamten Seitenladeprozess verwenden Sie die Analysefunktionen Ihres Browsers. In Firefox finden Sie sie unter *Tools>Web Developer>Toggle Tools>Network*. Diese zeigen die Reihenfolge der HTTP-Requests und -Antworten.

Aufgabe 6: Kontakt mit dem Webserver über Telnet

Versuchen Sie, sich über Telnet mit einem Webserver (z.B. `www.inf.unibz.it`) zu verbinden und eine Webseite (z.B. `/~nutt/index.html`) oder ein Foto (z.B. `/~nutt/Pictures/WernerNUTT-3.jpg`) herunterzuladen.

- Starten Sie dazu Telnet in einer Shell mit dem Kommando
`telnet <Zielrechner> 80`
Der Webserver auf dem Zielrechner wird (in der Regel) eine TCP-Verbindung annehmen. Sie können dann dem Webserver auf dem Zielrechner eine HTTP-Nachricht schicken.
- Tippen Sie den Text einer HTTP-GET-Anfrage ein, mit der Sie die HTML-Seite oder das Bild verlangen. (Oder, besser, schreiben Sie die Anfrage in einem Editor und kopieren Sie sie.) Wie lange bleibt die Verbindung offen?
- Versuchen Sie, die Anfrage über dieselbe Verbindung zu wiederholen (kopieren Sie die Anfrage in den Editpuffer Ihres Rechners und pasten Sie sie in die Shell, in der TELNET läuft). Wie oft können Sie dies wiederholen?
- Laden Sie auf die gleiche Art die Homepage von `www.mit.edu` herunter. Wie lange bleibt die Verbindung offen? Schauen Sie sich die Header der Antwort an. Teilt der Server etwas mit, das mit der Dauer der Verbindung zu tun hat?
- Laden Sie nun ein Bild herunter. Sie werden in der Antwort wahrscheinlich ein Header derfeld Art `ETag: "7b88c-440366719d340"` finden. Informieren Sie sich auf Wikipedia über die Bedeutung dieses Feldes und Stellen Sie die Anfrage erneut mit der Headerzeile `If-None-Match: <ETAG>`, wobei Sie das richtige Etag einsetzen. Was ist die Antwort des Servers?
- Stellen Sie nun eine Anfrage nach demselben Bild, diesmal mit dem Feld `If-Modified-Since`, nachdem Sie sich informiert haben über die Bedeutung dieses Feldes.

Finden Sie selbst heraus ...

- Finden Sie heraus, wie HTTP GETs auf TCP-Verbindungen abgebildet werden. Browser können eine TCP-Verbindung zu einem Server herstellen und mehrere HTTP-Requests senden. Wie lange bleibt diese Verbindung offen?
- Sehen Sie sich an, wie ein HTTP GET oder POST funktioniert. Können Sie Kennwörter aufzeichnen, die gesendet werden?
- Versuchen Sie, sich das AIMD-Verhalten von TCP anzusehen, indem Sie eine große Datei über FTP von <http://speedtest.tele2.net/> laden und sich dann

Statistics>TCP Stream Graph>Throughput
ansehen