

# Arbeiten mit dem Packet-Sniffer Wireshark

Prof. Dr. Otto Parzhuber

Hochschule München FK 06

Version vom 22.02.2019

## Inhaltsverzeichnis

1. Einführung - Packet Sniffer .....	1
1.1. Beispielanwendung zum Protokoll HTTP .....	8
1.2. Beispielanwendung zum Protokoll HTTP (Browser-Cache).....	8
1.3. Beispielanwendung Wetterdaten von einem Server holen .....	9
Beispielanwendung zum Protokoll http (Download einer größeren Datei) .....	10
HTTP Authentifizierung .....	10
DNS.....	12
nslookup .....	12
ipconfig .....	15
DNS mit Wireshark.....	16
Aufgaben zu nslookup .....	17
TCP .....	19
Bulk TCP von Ihrem Computer zu einem Server .....	19
Überblick über den Trace.....	20
ICMP .....	23
IP .....	24
Capture von Paketen die durch das Programm traceroute ausgelöst werden .....	24
Untersuchung des Wireshark Capture.....	25

## 1. Einführung - Packet Sniffer

Das grundlegende Werkzeug für die Beobachtung von Daten zwischen Rechnern wird als „**packet sniffer**“ bezeichnet.

Wie der Name schon sagt, fängt dieses Werkzeug empfangene/gesendete Daten Ihres Rechners ab. Ein Sniffer ist immer passiv, das heißt er verschickt keine Daten, sondern speichert Kopien der Daten der Kommunikation auf Ihrem Rechner.

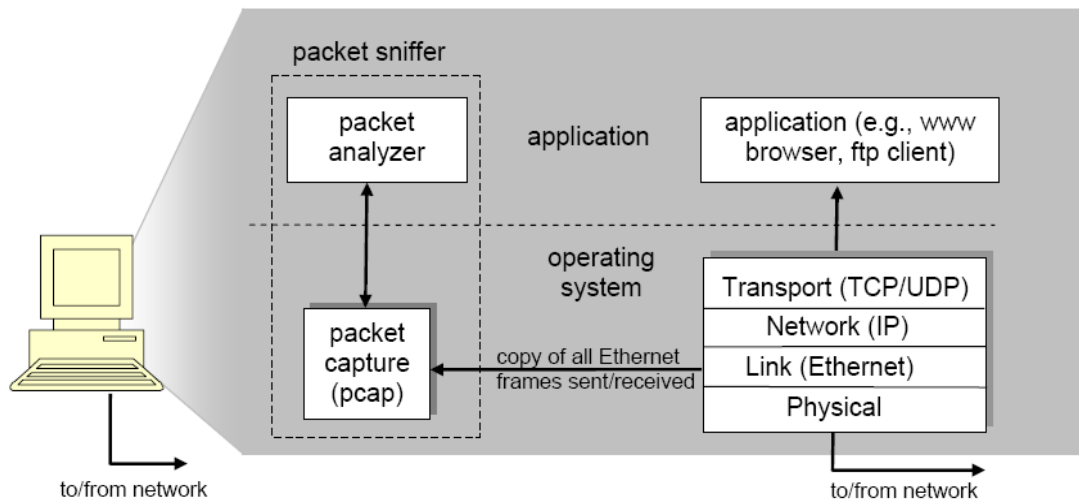


Figure 1: Packet sniffer structure

Das Bild 1 zeigt die Struktur eines „Sniffers“. Rechts unten im Bild sind die beteiligten Protokolle abgebildet, es werden also Protokolle der Layer 1 bis 4:

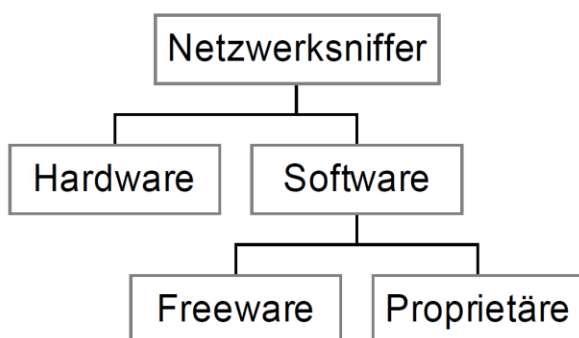
- oben ist die Applikation zu sehen, in unserem Fall ist dies der Webbrowser (Firefox, IE).
- die Blöcke die in den gestrichelten Linien eingerahmt sind gehören zu unserem „Sniffer“.

Am einfachsten lässt sich ein Netzwerk sniffen wenn Hubs als Verbindung der Netzwerkssegmente zwischengeschaltet sind. Bei anderen Verbindungen wie z.B. bei einem Switch bekommt man im Normalfall Probleme den Netzwerkverkehr abzuhören, da der Switch die Daten vom Sender nur an den tatsächlichen Empfänger weiterleitet. Somit würde man nur die eigenen Daten, sowie unwichtigeren Netzwerkverkehr wie z.B. Broadcasts aufzeichnen.

Für diesen Fall haben viele Hersteller dieser Komponenten Switches bzw. Router mit einer Monitorfunktion in Ihrem Portfolio. Damit ist es möglich Netzwerkdaten eines gewünschten Ports auf einen anderen zu spiegeln. Auf diesen gespiegelten Port kann man nun direkt zugreifen.

**Es ist zu beachten; dass das Abhören von Netzwerkverkehrsdaten ohne Einverständnis gemäß Telekommunikationsgesetz eine Straftat ist.**

Es gibt verschiedene Arten von Netzwerksniffen:



Die bekanntesten Netzwerksniffer sind in der Tabelle aufgelistet:

<b><i>Freeware</i></b>	<b><i>Proprietäre Produkte:</i></b>
<ul style="list-style-type: none"><li>- Wireshark (früher Ethereal)</li><li>- Ettercap</li><li>- NETCORtools (TCP Trace basierend)</li><li>- Tcpdump</li></ul>	<ul style="list-style-type: none"><li>- caplon (consistec)</li><li>- ClearSight Analyzer (ClearSight Networks)</li><li>- EtherPeek</li><li>- OmniPeek</li><li>- GigaPeek (Wild Packets)</li><li>- LABdecoder32 (Triticom)</li><li>- Microsoft Network Monitor</li><li>- NetSpector (INAT)</li></ul>

Im Praktikum werden wir mit dem Wireshark Sniffer arbeiten. Wireshark wörtlich aus dem Englischen übersetzt bedeutet „Kabelhai“ und ist ein kostenloses Programm zur Analyse von Netzwerkkommunikationsverbindungsdaten. Daher zählt es zur Programmgruppe der Netzwerksniffer. Es kann kostenlos unter: <http://www.wireshark.org/download.html> heruntergeladen werden.

### **Geschichte von Wireshark**

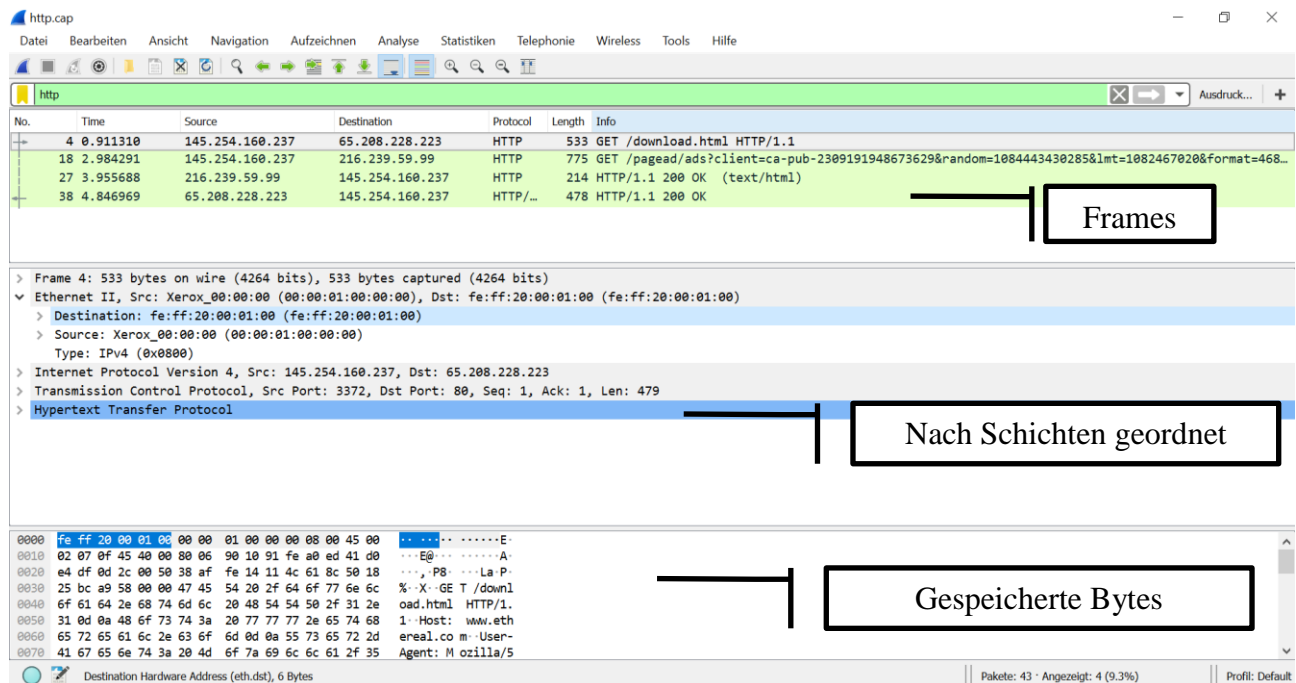
Ursprünglich war Wireshark unter den Namen „Ethereal“ bekannt. Ethereal wurde von der Firma Ethereal Software unter der Leitung von Gerald Combs als GNULizenz (General Public License) entwickelt. Als Gerald Combs im Jahre 2006 von Ethereal Software Inc. zu CACE Technologies wechselte, startete er gleich ein eigenes Folgeprojekt und nannte es Wireshark.

Wireshark verdrängte viele Netzwerkanalyseprogramme kommerzieller Hersteller vom Markt, da dieses Open-Source-Produkt von Bedienung, Erscheinungsbild und Wirkungsweise jedem dieser kommerziellen Programme vergleichbar und zudem kostenlos war!

Wireshark stellt nach der Aufzeichnung des Datenverkehrs einer Netzwerk Schnittstelle (meist eine Ethernet-Netzwerkkarte mit TCP/IP) die Daten in Form einzelner Pakete dar. Dabei werden die Daten übersichtlich und für den Benutzer nachvollziehbar zum Analysieren dargestellt. Der Inhalt der mitgeschnittenen Pakete kann betrachtet oder nach Inhalten gefiltert werden.

## Einführung in Wireshark

Das folgende Bild zeigt ein typisches Bild der Fenster des Wireshark Packet Sniffer:



Startbildschirm:

Willkommen bei Wireshark

### Öffnen

C:\Users\OPARZ\Documents\WS1819-IDEEN\Wireshark\_Samples\http.cap (nicht gefunden)  
 C:\Users\OPARZ\Documents\WS1819-IDEEN\Wireshark\_Samples\AliceInWonderland.pcapng (nicht gefunden)  
 C:\Users\OPARZ\Documents\WS1819-IDEEN\Wireshark\_Samples\dhcp.pcapng (nicht gefunden)  
 C:\Users\OPARZ\Documents\WS1819-IDEEN\Wireshark\_Samples\dns\_ietf.org.pcapng (nicht gefunden)  
 C:\Users\OPARZ\Documents\SS18-IDEEN\Wireshark\_Samples\http.cap (nicht gefunden)  
 C:\Users\OPARZ\Documents\SS18-IDEEN\Wireshark\_Samples\AliceInWonderland.pcapng (nicht gefunden)  
 C:\Users\OPARZ\Documents\SS18-IDEEN\Wireshark\_Samples\dhcp.pcapng (nicht gefunden)  
 C:\Users\OPARZ\Documents\SS18-IDEEN\Wireshark\_Samples\cc3100\_weather.pcapng (nicht gefunden)

### Aufzeichnen

...mit diesem Filter:

LAN-Verbindung\* 11

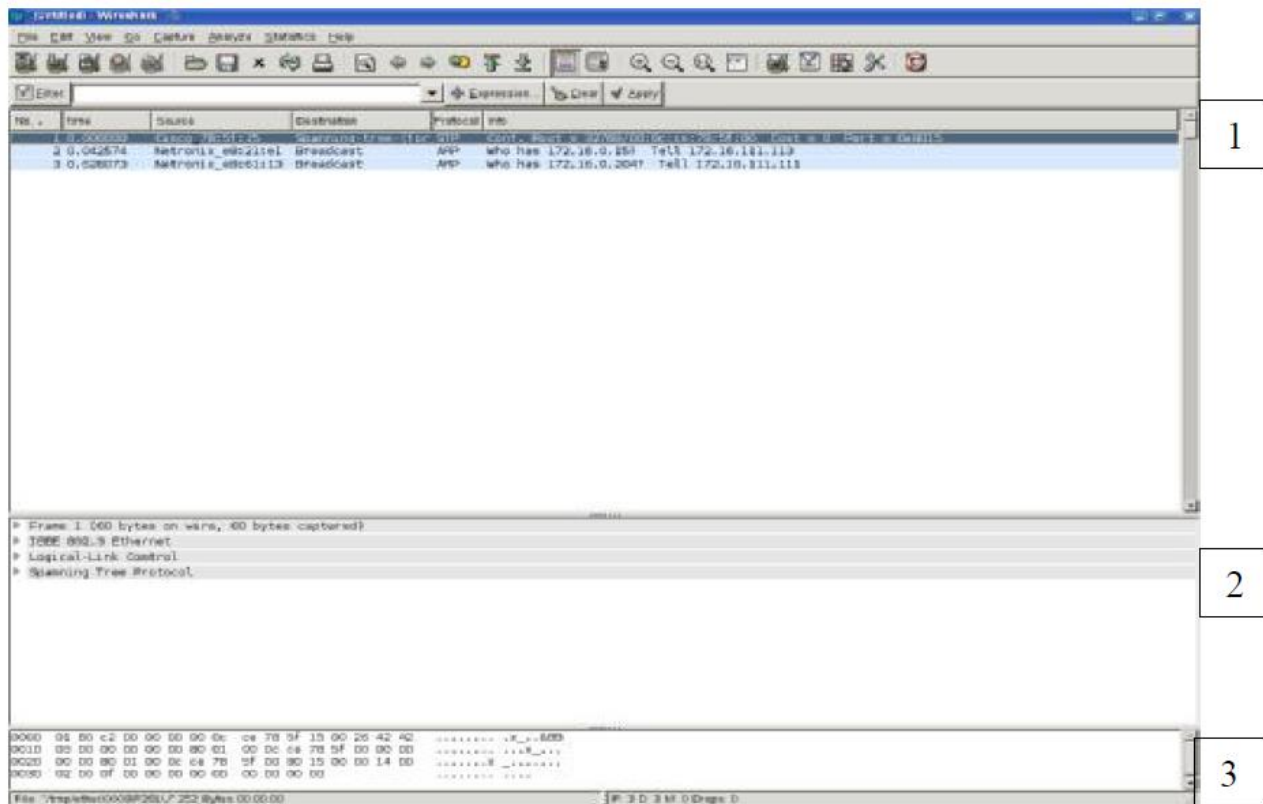
Ethernet

WLAN

Ethernet 3

USBPCap1

Daten des Aufzeichnungsmodus:



Der Bildschirm, in dem die aufgezeichneten Daten bearbeitet und analysiert werden, ist in 3 Bereiche aufgeteilt.

### 1) Paketliste

In der Paketliste, sieht man alle aufgezeichneten Frames. Die Spalten geben folgende Informationen preis:

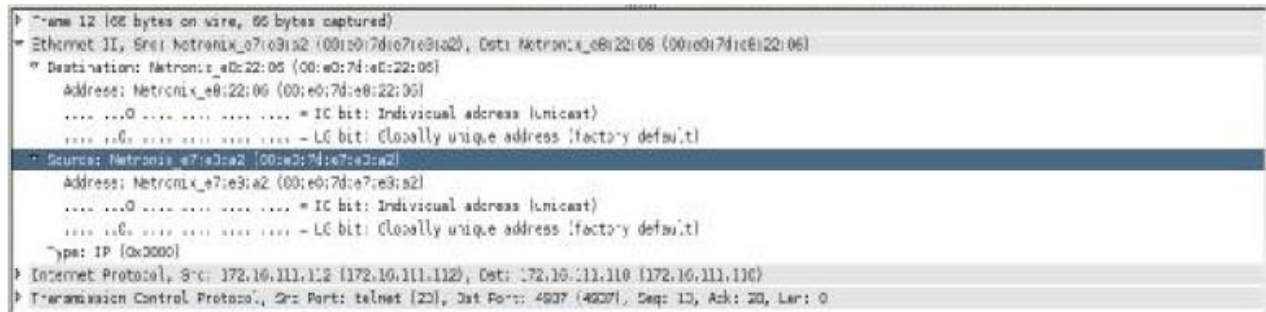
1. No. = ist eine fortlaufende Nummerierung der Frames
2. Time = zeigt den Zeitabschnitt der Aufzeichnung an
3. Source = zeigt den Absender eines Frames an (meist die IP)
4. Destination = zeigt den Empfänger des Frames an (meist die IP)
5. Protocol = zeigt das verwendete Protokoll des Frame an
6. Info = gibt zusätzliche Informationen zum Frame bekannt.

1	2	3	4	5	6
No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.105	192.168.0.105	Spanning Tree Protocol	STP: Root = 192.168.0.105, Cost = 0, Port = 2047
2	0.042574	Netronix_48c211e1	Broadcast	ARP	who has 172.16.0.105? Tell 172.16.101.113
3	0.026873	Netronix_48c61113	Broadcast	ARP	who has 172.16.0.2047? Tell 172.16.101.113

## 2) Paketdetails



In den Paketdetails werden die Layer (Schichten) des Datenframes angezeigt. Durch anklicken des Pfeil-Symbols kann der gewählte Layer erweitert werden.



Die Protokolldetails variieren von Protokoll zu Protokoll.

Die ersten zwei Layer sind immer gleich:

- Frame

Hier sind Informationen von Wireshark zum betreffenden Frame zusammengefasst. Unter anderem erkennt man hier die Größe des Frames, Zeit und die Zeitdifferenz zum vorrangegangenen Frame.

- Ethernet II

Zeigt Informationen zum OSI-Layer 2 (Sicherheitsschicht). Hier finden Sie die MAC-Adressen des Absenders und des Empfängers, im Normalfall wird die MAC-Adresse des Empfängers die des Default Routers sein.

---

Wie bereits oben erwähnt variieren die übrigen Paketdetails von Protokoll zu Protokoll, im Beispiel wird das TCP-Protokoll als Beispiel näher erläutert.

Beim TCP-Protokoll finden wir vier Paketdetails wieder (siehe oben gezeigte Abbildung). Auf die ersten beiden Layer die stets gleich sind wir schon näher eingegangen. Bei den anderen beiden Layer handelt es sich um:

- Internet Protokoll (IP)

Dieser Layer gibt Aufschluss zum OSI-Layer 3 und gibt Angaben wie die IP-Adresse des Absenders und Empfängers sowie die Lebenszeit des Pakets (TTL) preis.

- Transmission Control Protokoll (TCP)

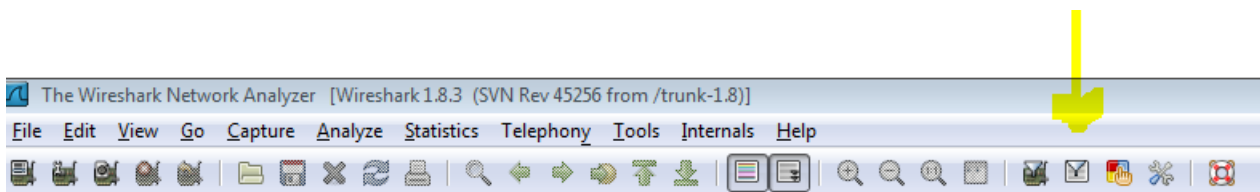
Das Layer zeigt Informationen sogar zum OSI-Layer 4 und zeigt alle Informationen zur ENDE-ZU-ENDE Verbindung an. Darüber hinaus werden die Ports preisgegeben.

### 3) Erläuterung zu Hexadezimale Paketanzeige

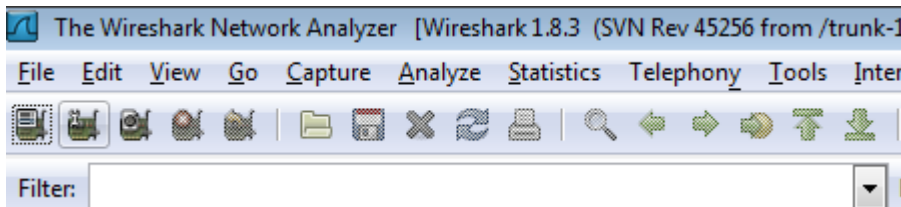


Hier sind die Daten einmal im Hexadezimalsystem (links) und nebendran im Klartext (rechts) bzw. in entschlüsselter Form angezeigt.

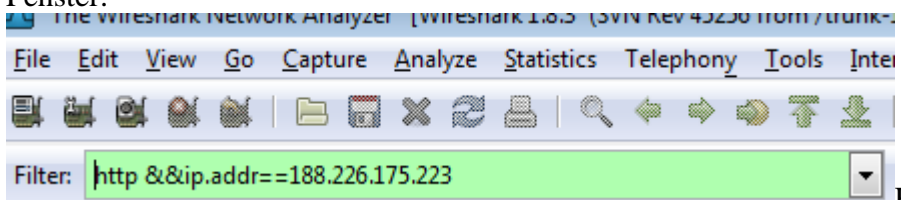
Die Filter:



Durch einen Klick auf das Filtersymbol öffnet sich ein Fenster in dem eine Vielzahl an Filterungsmöglichkeiten bereits voreingestellt sind. Damit kann man gezielt nach gewissen IP's oder Protokollen suchen.



In dem links gezeigten Textfeld können beliebig komplizierte Filter eingestellt werden. Ein Beispiel zeigt das folgende Fenster:



Hier werden nur http mit der IP-Adresse 188.226.175.223 angezeigt. Da http auf tcp/ip basiert sind natürlich auch die tcp-, ip- und Ethernet Daten angezeigt.



### 1.1. Beispielanwendung zum Protokoll HTTP

1. Starten Sie Firefox oder IE mit einer beliebigen Webseite
2. Starten Sie die Wireshark Software.
3. Wählen Sie unter Capture->Filter das Protokoll „http“ aus. Probieren Sie es ohne diese Filtereinstellung, werden Sie mit den Nachrichten des kompletten *TCP/IP-Stacks* überschwemmt.
4. Prüfen Sie, ob für den Capture die Schnittstelle bereits eingestellt ist. Wenn nicht müssen Sie das nun erledigen (in der Regel gibt es nur eine Ethernetkarte in Ihrem PC)
5. Starten Sie den Capture-Vorgang ( am besten etwa eine Minute warten vorher)
6. Geben Sie nun die folgende Zeile in den Browser ein: <http://192.168.10.113/hallo.html> .Nun müsste Ihr Browser eine einfache HTML-Datei anzeigen, die auf dem Server des Labors bereitgestellt wird.
7. Stoppen Sie nun den Wireshark Capture Vorgang.

Beantworten Sie die folgenden Fragen:

1. Welche Browser Version HTTP Version 1.0 oder 1.1 läuft? Welche Version läuft auf dem Server?
2. Welche Sprachen kann der Browser vom Server akzeptieren?
3. Wie lautet die IP-Adresse Ihres Rechners?
4. Wie lautet der Status-Code, der vom Server zum Browser geschickt wird?
5. Wann wurde die HTML Datei zuletzt geändert?
6. Wie viele Bytes wurden zu Ihrem Browser geschickt?
7. Ist die http Verbindung *nichtpersistent* oder *persistent*?
8. Welche Protokolle in den darunterliegenden Schichten werden für das http Protokoll verwendet

### 1.2. Beispielanwendung zum Protokoll HTTP (Browser-Cache)

Die meisten Web-Browser nutzen sogenanntes “object caching” und führen deshalb nur ein bedingtes GET aus, wenn auf ein http Objekt zugegriffen wird.

Für die nächste Aufgabe müssen Sie unbedingt zuerst den Cache des Browsers löschen.

1. Starten Sie den Browser
2. Starten Sie Wireshark und Capture
3. Als nächstes geben Sie nun die folgende URL ein: <http://192.168.10.113/hallo.html>
4. Geben Sie nun dieselbe URL noch mal ein (oder einfaches refresh).
5. Stoppen Sie nun den Wireshark Capture und betrachten nur die http Meldungen.



Beantworten Sie nun die folgenden Fragen:

1. Schauen Sie sich den Inhalt der ersten HTTP GET Anfrage von Ihrem Browser zum Server an. Sehen Sie eine "IF-MODIFIED-SINCE" Zeile im HTTP GET?
2. Schauen Sie auf die Antwort des Servers. Gibt der Server den Inhalt der Datei zurück?
3. Wie sieht die zweite Anfrage HTTP GET vom Browser zum Server aus? Gibt es eine "IF-MODIFIED-SINCE:" Zeile im HTTP GET? Wenn ja. Was kommt nach dem "IF-MODIFIED-SINCE:" header?
4. Wie sieht der HTTP Statuscode und die Antwort des Servers bei der zweiten HTTP GET Anfrage aus? Liefert der Server den Inhalt der Datei?
5. Welche Rolle spielt hier das Feld E-tag?
6. Je nach Browser werden bei der Anfrage an den Server zwei http GET Anfragen erstellt. Was ist der Grund dafür?

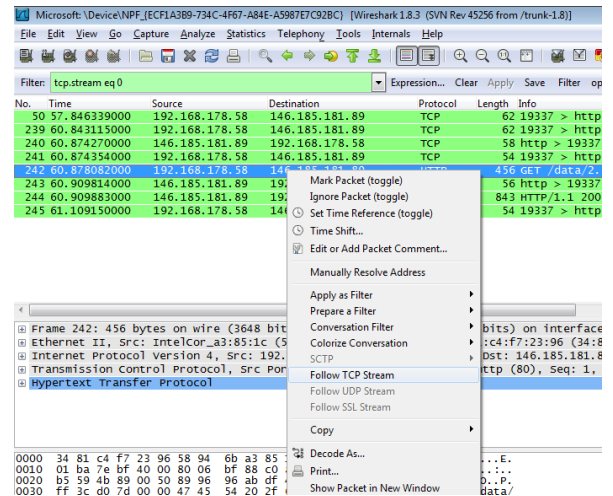
### 1.3. Beispielanwendung Wetterdaten von einem Server holen

In diesem Beispiel greifen Sie über einen API Befehl auf einen Wetter Server zu und holen die Daten ab.

1. Starten Sie den Browser, achten Sie darauf, dass der Cache leer ist.
2. Starten Sie Wireshark
3. Geben Sie nun in die Browserzeile folgendes Kommando ein:  
<http://api.openweathermap.org/data/2.5/weather?q=London,uk&appid=eaf1cdc1127bcd29e2ed8923f6f9a028> Als Ort können Sie selbstverständlich auch viele andere Städte auswählen.
4. Stoppen Sie nun Wireshark

Beantworten Sie die folgenden Fragen:

1. Welche http Methode wird verwendet?
2. Klicken Sie bei der Zeile mit den gesendeten http Befehl auf **Analyze->Follow TCP Stream**. Welche Information zeigt Ihnen das popup-Fenster?
3. Welche http Methoden erlaubt der Server



### Beispielanwendung zum Protokoll http (Download einer größeren Datei)

Bisher waren die Dateien einfache und kurze HTML-Dateien. Was geschieht beim herunterladen von größeren Dateien?

1. Starten Sie den Browser, achten Sie darauf, dass der Cache leer ist.
2. Starten Sie Wireshark
3. Geben Sie die folgende URL ein: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>
4. Stoppen Sie nun den Wireshark packet Capture.

Zuerst müssen Sie die IP-Adresse des Servers finden. Dazu können Sie entweder auf der Kommandozeile nslookup verwenden oder komfortabler ein grafisches Frontend über den Browser suchen.

Wenn Sie die IP-Adresse gefunden haben geben Sie diese als Displayfilter in Wireshark ein:

```
http && ip.addr== .....
```

In dem Anzeigefenster müssten Sie nun Ihre HTTP Requests sehen, gefolgt von mehreren Paketen als Antwort auf Ihre Anfrage.

Beantworten Sie die folgenden Fragen:

- Wie viele HTTP Requests wurden von Ihrem Browser geschickt? Welche Methoden wurden verwendet?
- Wie viele TCP Segmente mit Daten wurden für die http Antwort benötigt?
- Wie ist der Status-Code und die sonstigen Nachrichten verbunden mit der HTTP GET Anfrage?
- Gibt es irgendwelche HTTP Status Meldungen in den übertragenen Daten, die mit dem TCP “Continuation” verbunden sein könnten?

### HTTP Authentifizierung

Das Ziel dieser Aufgabe ist es eine passwortgeschützte Webseite zu öffnen und die http Nachricht dazu anzusehen.

Die URL

[http://gaia.cs.umass.edu/wireshark-labs/protected\\_pages/HTTP-wireshark-file5.html](http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html)

ist passwortgeschützt.

Der Benutzername ist “wireshark-students” (ohne Anführungszeichen) und das Passwort ist “network”. Um diese „sichere“ Seite anzusehen müssen Sie folgendes aufrufen:

- Browser Cache muss gelöscht sein, sicherheitshalber Browser schließen und wieder öffnen.
- Wireshark starten

- Geben Sie nun die oben genannte URL ein: [http://gaia.cs.umass.edu/wireshark-labs/protected\\_pages/HTTP-wiresharkfile5.html](http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html) .Ebenso natürlich Benutzername und Passwort.
- Stoppen Sie nun Wireshark Capture, und geben “http” als Displayfilter ein, so dass nur HTTP Nachrichten in dem Listing Fenster angezeigt wird.

Eine gute und vor allem einfache Beschreibung der http Authentifizierung ist hier zu finden  
[http://frontier.userland.com/stories/storyReader\\$2159](http://frontier.userland.com/stories/storyReader$2159)

Fragen zu dieser Aufgabe:

- Wie lautet die erste Antwort des Servers (Statuscode und Text) auf die HTTP GET Nachricht von Ihrem Browser?
- Wenn Ihr Browser die http Nachricht zum zweiten mal sendet, wie sieht das neue Feld aus, das in der HTTP GET Nachricht hinzugefügt worden ist.

Der Benutzername und das Passwort werden kodiert mit einem Feld aus Characters (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcm0=), hinter dem “Authorization: Basic” Header in der HTTP GET Nachricht des Clients.

Benutzername und Passwort sind nicht verschlüsselt!!!

<http://www.patshaping.de/projekte/kleinkram/base64.php>

Es handelt sich nur um eine Codierung im Format Base64.

## DNS

DNS wandelt Hostnamen in IP-Adressen um. Die Rolle des Client ist relativ einfach: der Client schickt eine Anfrage an den lokalen DNS Server und bekommt eine Antwort.

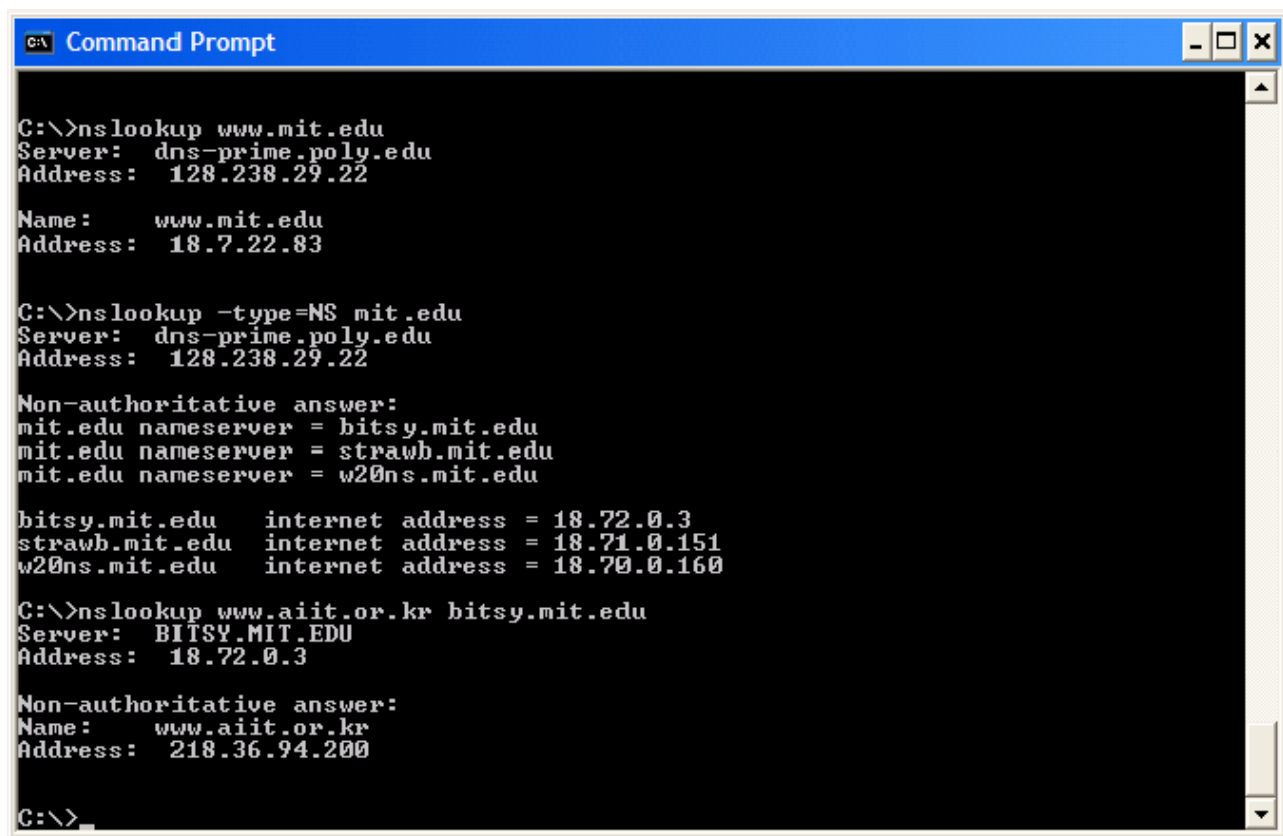
Unsichtbar für den Client geschieht natürlich viel Kommunikation zwischen den Servern, damit die Anfrage des Clients entweder rekursiv oder iterativ gelöst werden kann

### *nslookup*

Der Aufruf dieses Tools ist in Windows und in Linux identisch, um es zu starten müssen Sie *nslookup* auf der Kommandozeile eingeben.

Die primäre Aufgabe von *nslookup* ist es einen angegebenen DNS-Server nach einem DNS Record abzufragen. Dieser DNS-Server kann ein Root DNS Server, ein Top-Level-Domain Server oder ein autoritativer Server sein.

*nslookup* schickt eine Anfrage zu dem angegebenen DNS-Server, erhält eine Antwort und zeigt diese an.



```
C:\>nslookup www.mit.edu
Server:  dns-prime.poly.edu
Address:  128.238.29.22

Name:    www.mit.edu
Address:  18.7.22.83

C:\>nslookup -type=NS mit.edu
Server:  dns-prime.poly.edu
Address:  128.238.29.22

Non-authoritative answer:
mit.edu nameserver = bitsy.mit.edu
mit.edu nameserver = strawb.mit.edu
mit.edu nameserver = w20ns.mit.edu

bitsy.mit.edu    internet address = 18.72.0.3
strawb.mit.edu  internet address = 18.71.0.151
w20ns.mit.edu   internet address = 18.70.0.160

C:\>nslookup www.aiit.or.kr bitsy.mit.edu
Server:  BITSY.MIT.EDU
Address:  18.72.0.3

Non-authoritative answer:
Name:    www.aiit.or.kr
Address:  218.36.94.200

C:\>
```

Hier sind drei verschiedene unabhängige *nslookup* Befehle gezeigt:

- Wenn kein DNS-Server angegeben wird, wird der Default DNS-Server ausgewählt. In diesem Beispiel ist das der Server mit der Adresse ***dns-prime.poly.edu***.

### *nslookup* [www.mit.edu](http://www.mit.edu)

Diese Anfrage liefert zwei Informationen: (1) den Namen und die IP-Adresse des DNS-Servers, der die Antwort liefert und (2) die Antwort selber. Obwohl die Antwort offensichtlich vom Default DNS-Server kommt, ist es sehr wahrscheinlich, dass dieser lokale DNS-Server iterative mehrere andere anfragt.

- Im zweiten Fall wird die Option NS in der Kommandozeile angegeben.

```
nslookup -type=NS www.mit.edu
```

Die Aufgabe ist dieselbe: such mir die IP-Adresse zu der Domain [www.mit.edu](http://www.mit.edu). Die Anfrage geht wiederum an den Default DNS-Server, als Antwort wird aber eine autoritative gefordert. (ohne `-type` wird automatisch `-type=A` angefragt). Wenn trotzdem „nicht autorisierende Antwort“ vorkommt, bedeutet es, dass diese Anfrage vom Cache eines Servers kommt anstelle des autoritativen DNS Servers vom [www.mit.com](http://www.mit.com).

- Im dritten Fall werden die Anfragen direkt zu einem spezifizierten DNS-Server geschickt, in dem gezeigten Fall soll die IP-Adresse von [www.studyinkorea.go.kr](http://www.studyinkorea.go.kr) von dem DNS-Server der Fritzbox angefragt werden

```
nslookup www.studyinkorea.go.kr fritz.box
```

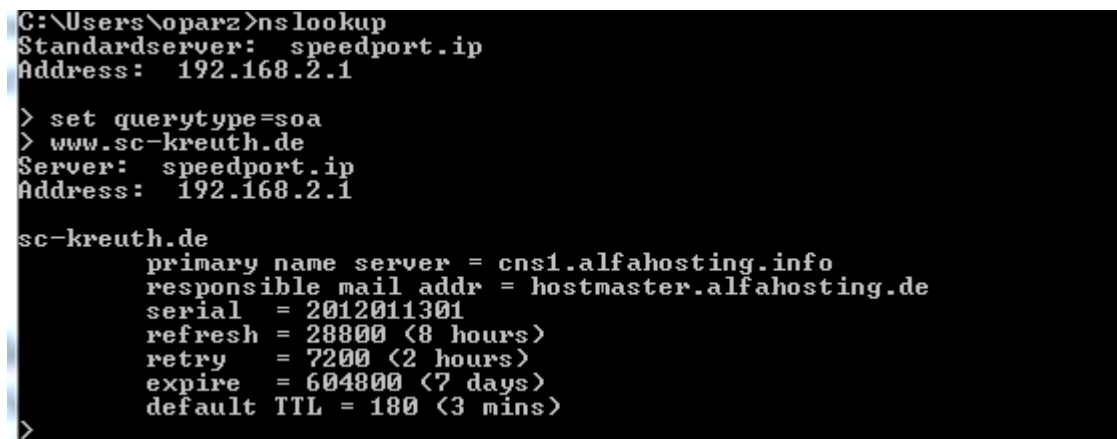
Dieser Aufruf mit dediziertem Nameserver ist in der Praxis weniger relevant.

Generell kann `nslookup` mit keiner, einer oder mehreren Optionen gestartet werden. Die Angabe des DNS-Servers ist optional, wenn kein Server angegeben ist, wird die Anfrage an den default Server geschickt.

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

Aufgaben:

1. Starten Sie `nslookup` um eine IP Adresse eines Web-Servers in Asien zu erhalten, z.B. [www.studyinkorea.go.kr](http://www.studyinkorea.go.kr).
2. Starten Sie `nslookup` um den DNS-Server einer Universität in Europa zu erhalten.
3. Starten Sie `nslookup` so, dass einer der Server, den Sie als Ergebnis aus 2 erhalten eine Anfrage an Mailserver für web.de Mail sendet.
4. Wie lauten die Nameserver von [www.google.de](http://www.google.de)? Handelt es sich um autorisierende Nameserver?
5. Wie viele Mailserver können Sie bei T-Online identifizieren?



```
C:\Users\oparz>nslookup
Standardserver: speedport.ip
Address: 192.168.2.1

> set querytype=soa
> www.sc-kreuth.de
Server: speedport.ip
Address: 192.168.2.1

sc-kreuth.de
primary name server = dns1.alfahosting.info
responsible mail addr = hostmaster.alfahosting.de
serial = 2012011301
refresh = 28800 (8 hours)
retry = 7200 (2 hours)
expire = 604800 (7 days)
default TTL = 180 (3 mins)
>
```

Der obige Screenshot liefert den Primary Nameserver der Domain [www.sc-kreuth.de](http://www.sc-kreuth.de).

(soa = start of authority)

Die Bedeutung der einzelnen Einträge liefert folgende Erläuterung (IBM):

What does serial / refresh / retry / expire / minimum / and TTL mean?

### Caching and time to live

Because of the huge volume of requests generated by a system like the DNS, the designers wished to provide a mechanism to reduce the load on individual DNS servers. The mechanism devised provided that when a DNS resolver (i.e. client) received a DNS response, it would cache that response for a given period of time. A value (set by the administrator of the DNS server handing out the response) called the time to live, or TTL defines that period of time. Once a response goes into cache, the resolver will consult its cached (stored) answer; only when the TTL expires (or when an administrator manually flushes the response from the resolver's memory) will the resolver contact the DNS server for the same information.

Generally, the time to live is specified in the Start of Authority (SOA) record. SOA parameters are:

**Serial** — The revision number of this zone file. Increment this number each time the zone file is changed so that the changes will be distributed to any secondary DNS servers.

**Refresh** — The amount of time in seconds that a secondary name server should wait to check for a new copy of a DNS zone from the domain's primary name server. If a zone file has changed then the secondary DNS server will update its copy of the zone to match the primary DNS server's zone.

**Retry** — The amount of time in seconds that a domain's primary name server (or servers) should wait if an attempt to refresh by a secondary name server failed before attempting to refresh a domain's zone with that secondary name server again.

**Expire** — The amount of time in seconds that a secondary name server (or servers) will hold a zone before it is no longer considered authoritative.

**Minimum** — The amount of time in seconds that a domain's resource records are valid. This is also known as a minimum TTL, and can be overridden by an individual resource record's TTL.

**TTL (time to live)** - The number of seconds a domain name is cached locally before expiration and return to authoritative nameservers for updated information.

*ipconfig*

*ipconfig* (für Windows) und *ifconfig* (für Linux/Unix) gehören zu den am nützlichsten kleinen Tools speziell für das Debuggen im Netzwerk. Diese Tools können verwendet werden um:

- TCP/IP Konfiguration
- IP-Adresse
- DNS\_Server Adressen
- Adapter Typ usw.

anzuzeigen. Wenn Sie die gesamte Information anzeigen wollen, geben Sie ein:

```
ipconfig /all
```

```

C:\>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : USG11631-ZMWQA6
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : poly.edu
    Description . . . . . : Intel(R) PRO/100 UE Network Connection
    Physical Address. . . . . : 00-09-6B-10-60-99
    Dhcp Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IP Address. . . . . : 128.238.38.160
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 128.238.38.1
    DHCP Server . . . . . : 128.238.29.25
    DNS Servers . . . . . : 128.238.29.22
                           128.238.29.23
                           128.238.2.38
                           128.238.32.22
    Primary WINS Server . . . . . : 128.238.29.23
    Secondary WINS Server . . . . . : 128.238.29.22
    Lease Obtained. . . . . : Monday, August 30, 2004 1:30:50 PM
    Lease Expires . . . . . : Monday, August 30, 2004 7:30:50 PM

C:\>

```

Was bedeuten die beiden folgenden wichtigen Aufrufoptionen?

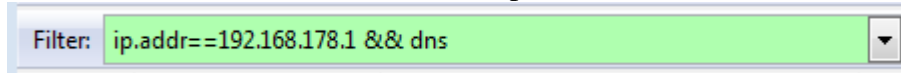
- `ipconfig /displaydns`
- `ipconfig /flushdns`



## DNS mit Wireshark

Als Aufgabe sollen nun DNS Pakete mit Wireshark untersucht werden.

- Verwenden Sie *ipconfig* um den DNS Cache in Ihrem Host zu löschen.
- Starten Sie Ihren Browser und löschen Sie den Cache.
- Öffnen Sie Wireshark und geben Sie ein passendes Displayfilter ein wie in dem Bild gezeigt ist. Die IP-Adresse müssen Sie entsprechend ändern.



Dieses Filter entfernt alle Pakete, die nicht zu Ihrem Host gehören und zeigt ausschließlich Pakete mit dem Protokoll dns an.

- Starten Sie den Capture.
- Besuchen Sie nun die Webseite <http://www.ietf.org> oder eine andere Ihrer Wahl.
- Stoppen Sie den Capture-Vorgang.

Beantworten Sie die folgenden Fragen:

- Suchen Sie die DNS Anfrage- und Antwort-Nachrichten (*Standard Query* und *Standard Query Response*). Werden die Anfragen über UDP oder über TCP gesendet?
- Was ist der Zielpport der DNS Anfrage? Was ist der Ausgangsport der DNS Antwort?
- An welche IP-Adresse wird die DNS Anfrage geschickt? Benützen Sie *nslookup* um die IP Adresse Ihres lokalen DNS Servers zu bekommen. Sind es dieselben Adressen?
- Untersuchen Sie die Nachricht der DNS Anfrage. Um welchen “Typ” einer DNS Anfrage handelt es sich (Iterativ oder rekursiv)?
- Untersuchen Sie die DNS Antwort. Wie viele „Antworten“ werden geliefert? Was enthält jede dieser Nachrichten?

DNS, **d**as **n**ie endende Chao**s**:

Es gibt verschiedene Protokolle, die in Wireshark immer wieder zu sehen sind:

NBNS:

NetBIOS Name Service (RFC1001/1002) ähnlich wie DNS

LLMNR:

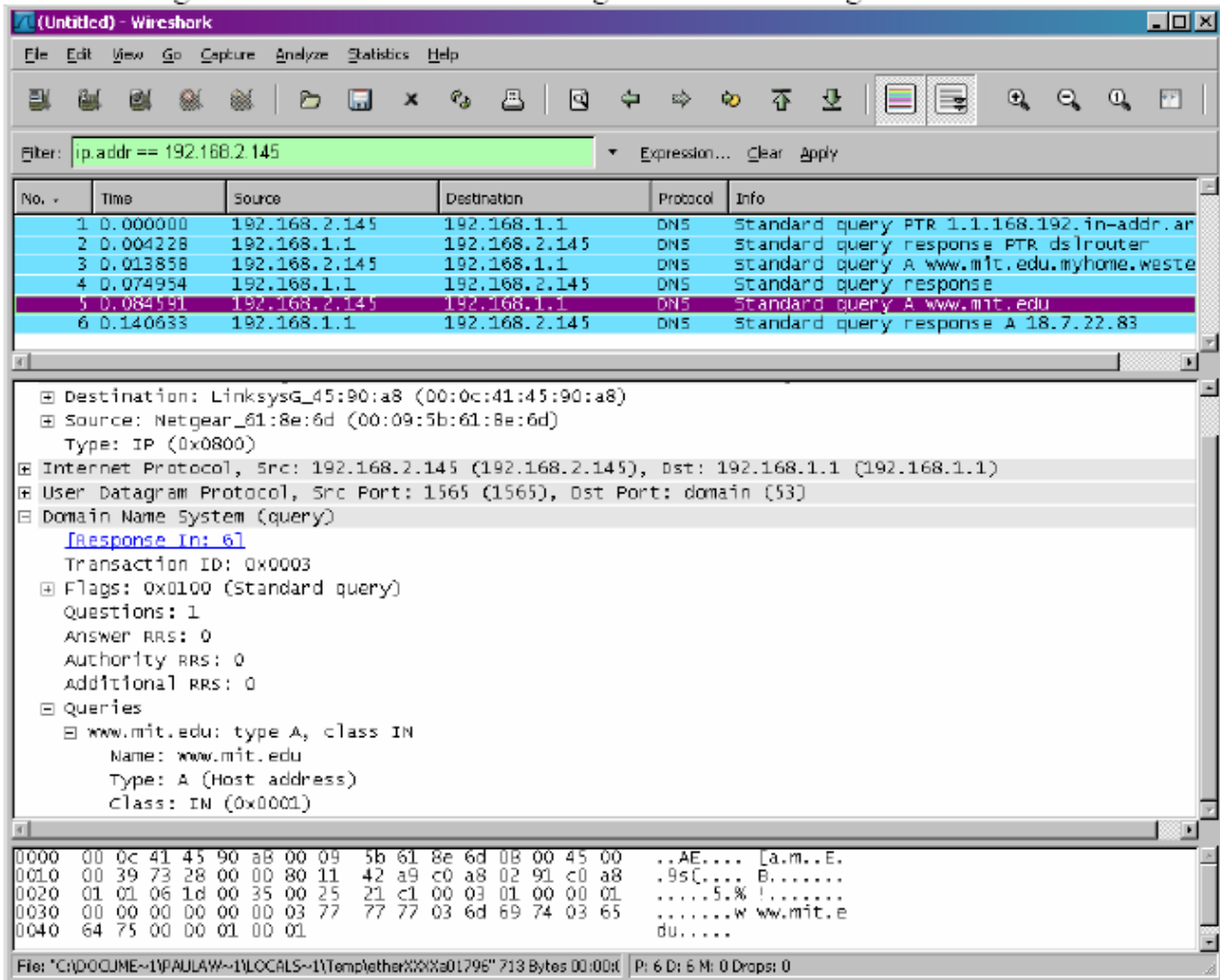
Link-Local Multicast Name Resolution (RFC4795) ähnlich wie DNS

Beides sind Windows spezifische Protokolle zur Namensauflösung!

## Aufgaben zu nslookup

- Starten Sie den Capture Vorgang.
- Rufen Sie *nslookup* mit einem Domainnamen Ihrer Wahl auf.
- Stoppen Sie den Capture Vorgang.
- Stellen Sie einen passenden Displayfilter ein. Wie lautet dieser?

Sie sollten einen Trace, ähnlich dem folgenden Bild erhalten:



*nslookup* hat drei DNS Anfragen geschickt und drei DNS Antworten erhalten. Die ersten beiden Teile sind spezifisch zu *nslookup* und für uns nicht interessant. Wichtig ist der letzte Teil

- Geben Sie den Zielpport der DNS Anfrage an. Wie lautet der Source Port der DNS Antwort?
- Zu welcher IP-Adresse wird die DNS Anfrage geschickt? Ist dies die IP- Adresse Ihres lokalen DNS Servers?
- Untersuchen Sie die Nachricht der DNS Anfrage. Welcher “Typ” von DNS Anfrage ist es? Welche Informationen können Sie aus der Anfrage lesen?
- Ist die Anfrage iterative oder rekursiv?
- Untersuchen Sie die DNS Antwort. Welche Informationen enthält die Antwort?

Wiederholen Sie nun den vorherigen Versuch, aber geben nun folgende Parameter in nslookup ein:

```
nslookup -type=NS mit.edu
```

Beantworten Sie die folgenden Fragen:

- Zu welcher IP-Adresse wird die DNS Anfrage geschickt? Ist dies die IP- Adresse Ihres lokalen DNS Servers?
- Untersuchen Sie die Nachricht der DNS Anfrage. Welcher “Typ” von DNS Anfrage ist es? Welche Informationen können Sie aus der Anfrage lesen?
- Untersuchen Sie die DNS Antwort. Wie lautet der Name Server, den die Antwort-Nachricht liefert? Enthält die Antwort auch die IP-Adressen der Nameserver?

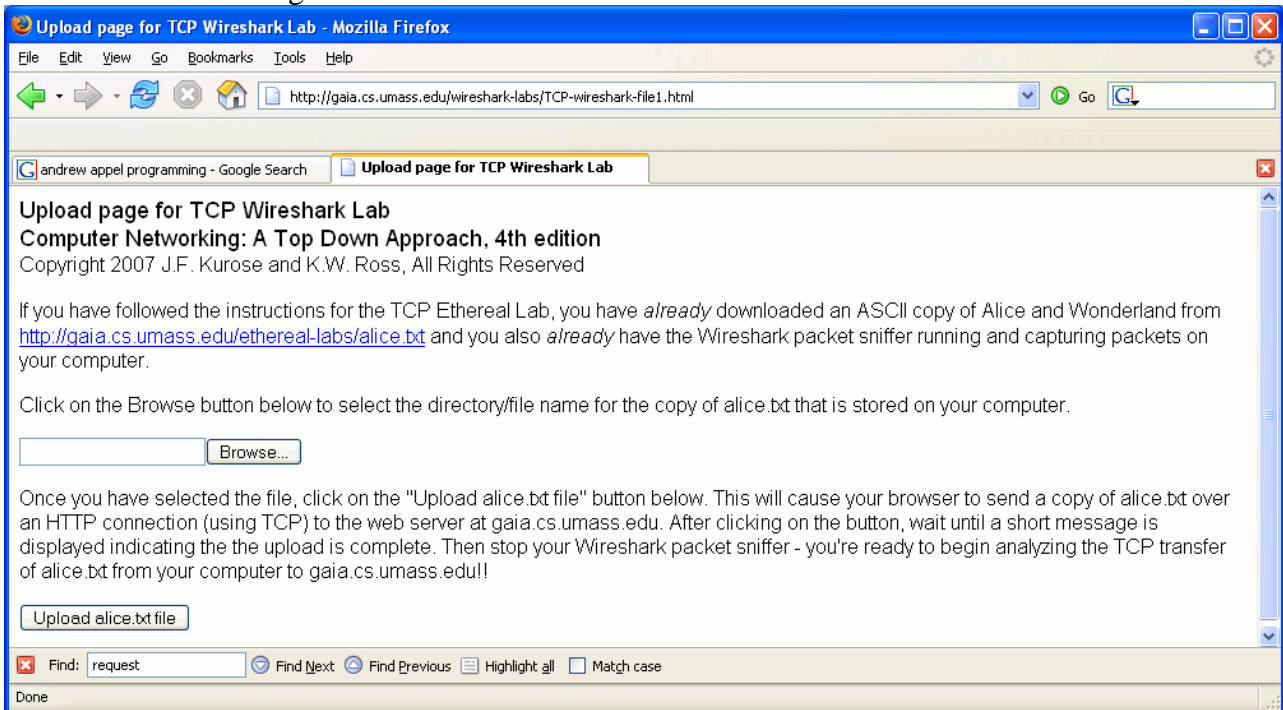
## TCP

In dieser Aufgabe wird TCP genauer untersucht. Als Beispiel dient eine 150 kB Datei (*Alice's Adventures in Wonderland*).

### Bulk TCP von Ihrem Computer zu einem Server

- Starten Sie Ihren Browser. Gehen Sie zur Seite <http://gaia.cs.umass.edu/wiresharklabs/alice.txt>. Sie erhalten eine ASCII Kopie von *Alice in Wonderland*. Speichern Sie diese Datei auf Ihrem Computer.
- Rufen Sie nun die Seite <http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html> auf.

Sie sollten nun das folgende Bild sehen:



- Benützen Sie nun den *Browse* Button dieses Formulars, um den Namen der Datei (mit vollständigem Pfadnamen) einzugeben. Starten Sie den *Upload* noch nicht!
- Starten Sie nun Wireshark und beginnen Sie den Capture (*Capture->Options*) mit *OK* im Packet Capture Options Bildschirm.
- Drücken Sie nun im Browser den Button für den upload "*Upload alice.txt file*" zum [gaia.cs.umass.edu](http://gaia.cs.umass.edu) Server. Wenn der Upload geklappt hat, wird in Ihrem Browser eine kurze Nachricht angezeigt.
- Stoppen Sie nun den Wireshark Capture. Ihr Wireshark Fenster sollte aussehen wie in dem folgenden Bild:

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.145	128.119.245.12	TCP	1250 > http [SYN] Seq=0 Len=0 MSS=1460
2	0.046402	128.119.245.12	192.168.2.145	TCP	http > 1250 [SYN, ACK] Seq=0 Ack=1 win=5840
3	0.046524	192.168.2.145	128.119.245.12	TCP	1250 > http [ACK] Seq=1 Ack=1 win=65535 [TC
4	0.046963	192.168.2.145	128.119.245.12	HTTP	POST /ethereal-labs/lab3-1-reply.htm HTTP/1
5	0.047339	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
6	0.128451	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=514 win=6432 Le
7	0.128619	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
8	0.128717	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
9	0.214161	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=1966 win=8712 L
10	0.214315	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
11	0.214415	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
12	0.298180	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=3418 win=11616
13	0.298326	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
14	0.381927	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=4870 win=14520
15	0.382241	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
16	0.382377	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
17	0.382459	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
18	0.421386	192.168.2.102	192.168.2.255	NBNS	Name query NB MSHOME<lb>
19	0.466467	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=6322 win=17424
20	0.552453	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=7774 win=20328
21	0.624375	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=8957 win=23232
22	0.624707	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
23	0.624857	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
24	0.624943	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
25	0.708403	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=10409 win=26136
26	0.794139	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=11861 win=29040
27	0.866343	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=13053 win=31944
28	0.868855	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
29	0.869431	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
30	0.869544	192.168.2.145	128.119.245.12	HTTP	Continuation or non-HTTP traffic
31	0.950346	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=14505 win=32767
32	1.036229	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=15957 win=32767
33	1.108269	128.119.245.12	192.168.2.145	TCP	http > 1250 [ACK] Seq=1 Ack=17149 win=32767

Frame 1 (62 bytes on wire, 62 bytes captured)

Ethernet II, Src: Netgear\_61:8e:6d (00:09:5b:61:8e:6d), Dst: LinksysG\_45:90:a8 (00:0c:41:45:90:a8)

Internet Protocol, Src: 192.168.2.145 (192.168.2.145), Dst: 128.119.245.12 (128.119.245.12)

Transmission Control Protocol, Src Port: 1250 (1250), Dst Port: http (80), Seq: 0, Len: 0

```

0000  00 0c 41 45 90 a8 00 09 5b 61 8e 6d 08 00 45 00  ..AE.... [a.m..E.
0010  00 30 2b 6b 40 00 80 06 96 9f c0 a8 02 91 80 77  .0+k@... ..w
0020  f5 0c 04 e2 00 50 c2 67 22 99 00 00 00 00 70 02  ....P.g .....p.
0030  ff ff 60 2f 00 00 02 04 05 b4 01 01 04 02      .../.....

```

File: "C:\DOCUME~1\PAULAW~1\LOCALS~1\Temp\etherXXXa03100" 165 KB 00:00:09 P: 214 D: 214 M: 0 Drops: 0

## Überblick über den Trace

Filtern Sie die angezeigten Pakete mit dem Filter "tcp".

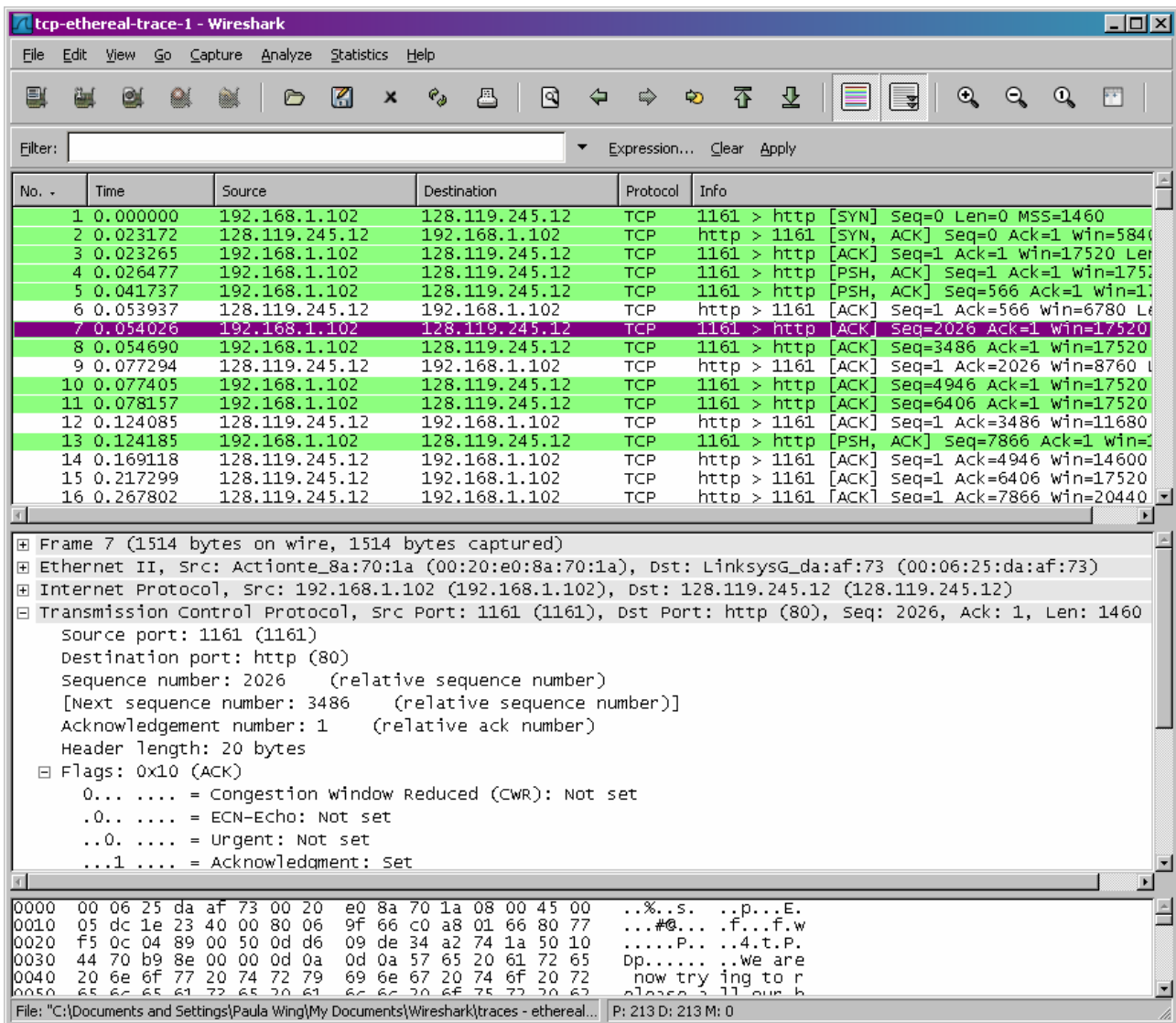
Sie sollten nun eine Reihe von TCP und HTTP Nachrichten sehen, die zwischen Ihrem Computer und gaia.cs.umass.edu ausgetauscht wurden.

- Sie sollten den Verbindungsaufbau mit dem 3 Wege-Handshake ( SYN Nachricht) sehen.
- Sie sollten eine HTTP POST Nachricht und eine Reihe von "HTTP Continuation" Nachrichten sehen, die von Ihrem Computer an gaia.cs.umass.edu geschickt werden. Wireshark zeigt damit an, dass mehrere TCP Segmente benötigt werden um ein einzige HTTP Nachricht zu senden.
- Sie sollten TCP ACK Segmente sehen, die von gaia.cs.umass.edu an Ihren Computer geschickt werden.

Beantworten Sie nun die folgenden Fragen:

- Wie lautet die IP Adresse und die TCP Port Nummer, die vom Client Computer genutzt wird? Am einfachsten ist dies über die Auswahl einer HTTP Nachricht zu finden (am einfachsten ist es eine http Nachricht zu suchen und darin die Details der TCP Pakete zur Übertragung dieser http Nachricht)
- Wie lautet die IP-Adresse von gaia.cs.umass.edu? Auf welchem Port sendet und empfängt der Server TCP-Segmente für diese Verbindung?

Filtern Sie nun nur die TCP-Nachrichten (http Nachrichten nicht anzeigen lassen, sonst wird es zu unübersichtlich).



Beantworten Sie die folgenden Fragen:

- Geben Sie die Sequenznummer des TCP SYN Segments an, welches die TCP Verbindung zwischen dem Client und gaia.cs.umass.edu initialisiert. Was ist in dem Segment enthalten, womit dieses Segment als SYN Segment gekennzeichnet wird?

- Geben Sie die Sequenznummer des SYN ACK Segments an, welches von gaia.cs.umass.edu als Antwort zu dem SYN Segment zum Client geschickt wird.
- Was für ein Wert steht im ACKnowledgement Feld des SYN ACK Segment? Wie hat gaia.cs.umass.edu diesen Wert festgelegt? Durch welchen Eintrag in dem Segment wird es als SYN ACK Segment identifiziert?
- Geben Sie Sequenznummer des TCP Segments an, das den HTTP POST Befehl enthält. (Um diese Nummer zu erhalten, müssen Sie in das „packet content field“ unten im Wireshark Fenster gehen und in dem DATA Feld ein Segment mit „POST“ suchen.
- Betrachten Sie nun das TCP Segment mit dem http POST als das erste Segment der TCB Verbindung.
  - Geben Sie die Sequenz Nummern der ersten 6 Segmente der TCP Verbindung an (inklusive des Segments mit POST)
  - Zu welcher Zeit wurde jedes dieser Segmente geschickt?
  - Wann wurde das ACK der einzelnen Segmente erhalten?
  - Geben Sie die RTT für jedes der 6 Segmente an. (Differenz zwischen Versenden und dem ACK ist bekannt!) Es gibt eine interessante option bei Wireshark: *Statistics->TCP Stream Graph->Round Trip Time Graph*.
  - Wie lang sind die ersten sechs TCP Segmente?



## ICMP

In diesem Versuch werden verschiedene Aspekte des ICMP-Protokoll untersucht:

- ICMP-Meldungen generiert durch das Ping-Programm;
- ICMP-Meldungen generiert durch das Traceroute-Programm;
- Format und Inhalt einer ICMP-Nachricht.

Am einfachsten kann das ICMP Protokoll anhand eines Wireshark Capture des bekannten ping Programms untersucht werden. Mit dem ping Programm kann ein Netzwerkadministrator überprüfen, ob ein Host lebt oder nicht.

Das Ping-Programm des Quellhosts sendet ein Paket an die Ziel-IP-Adresse. Wenn das Ziel erreichbar antwortet der Ziel-Host, indem er ein Paket zurück an den Quell-Host sendet. Wie leicht zu erraten ist, werden die Pakete mit ICMP verschickt.

- Starten Sie die Windows-Eingabeaufforderung (cmd).
- Starten Sie Wireshark und beginnen Sie die Paketerfassung.
- Geben Sie den Befehl: *ping -n 10 hostname* ein, wobei hostname z.B. [www.google](http://www.google) ist. (Was bedeutet das Argument -n10?)
- Wenn das Ping-Programm beendet ist, beenden Sie die Paketerfassung in Wireshark..

Am Ende dieses Versuchs sollte die Kommandozeile ähnlich wie die folgende Abbildung aussehen.

```
C:\Dokumente und Einstellungen\oparz>ping -n 10 www.yosemite.com

Ping yati1.yosemite.com [139.151.188.4] mit 32 Bytes Daten:

Antwort von 139.151.188.4: Bytes=32 Zeit=186ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=229ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=185ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=185ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=213ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=196ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=216ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=194ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=218ms TTL=120
Antwort von 139.151.188.4: Bytes=32 Zeit=190ms TTL=120

Ping-Statistik für 139.151.188.4:
    Pakete: Gesendet = 10, Empfangen = 10, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 185ms, Maximum = 229ms, Mittelwert = 201ms

C:\Dokumente und Einstellungen\oparz>_
```

Der Source-Ping stammt aus München und die Zieladresse ist in Nordamerika. Es wurden 10 Nachrichten versendet und auch wieder empfangen. Zusätzlich wird auch die RTT (Round Trip Time) mitberechnet und ausgegeben. Der Mittelwert ist bei diesem Beispiel 201ms.

- Weshalb sehen sie 20 Nachrichten, obwohl nur 10 Nachrichten verschickt wurden?
- Woran sehen Sie im IP Header, dass das Protokoll ICMP verwendet wird? Welche Nummer hat dieses Protokoll?
- Welchen Typ des ICMP Protokolls verwenden Sie hier?
- Wie sieht die *payload* des ICMP Protokolls aus?

- Verwendet das ICMP Protokoll Portnummern?
- Wieviele Bytes verwendet die Checksumme, die Identifikationsnummer und die Sequenznummer?

## IP

Dieses Kapitel befasst sich mit dem IP-Protokoll, und als Schwerpunkt sollen IP Datagramme untersucht werden.

Für die Untersuchung von gesendeten und empfangenen IP Datagrammen ist das Programm *traceroute* ein sehr gutes Hilfsmittel. Im einzelnen werden die verschiedenen Felder im IP Datagramm und die Fragmentierung untersucht.

Eine gute Einführung für das Programm *traceroute* liefert der RFC2151

(<http://www.rfc-editor.org/rfc/rfc2151.txt>). Die Beschreibung des IP Protokolls finden Sie im RFC791 (<http://www.ietf.org/rfc/rfc791.txt>).

### *Capture von Paketen die durch das Programm traceroute ausgelöst werden*

Um für diese Aufgabe IP Datagramme zu erzeugen wird das Programm *traceroute* verwendet. Damit werden Datagramme mit verschiedenen Größen an ein bestimmtes Ziel X gesendet.

*Traceroute* sendet dazu mehrfach Pakete mit einer veränderten und jeweils um 1 erhöhten Time-to-live (TTL), beginnend mit 1, an das Zielsystem. Jeder Host, der das Datenpaket in Folge empfängt, zählt den Wert der TTL um eins herunter. Empfängt ein Router ein Paket mit TTL=1 und müsste es vermitteln, verwirft er es und sendet die ICMP-Antwort Typ 11: Time-to-live exceeded und Code 0: „Time to live exceeded in transit“ an den Absender mit seiner Adresse zurück.

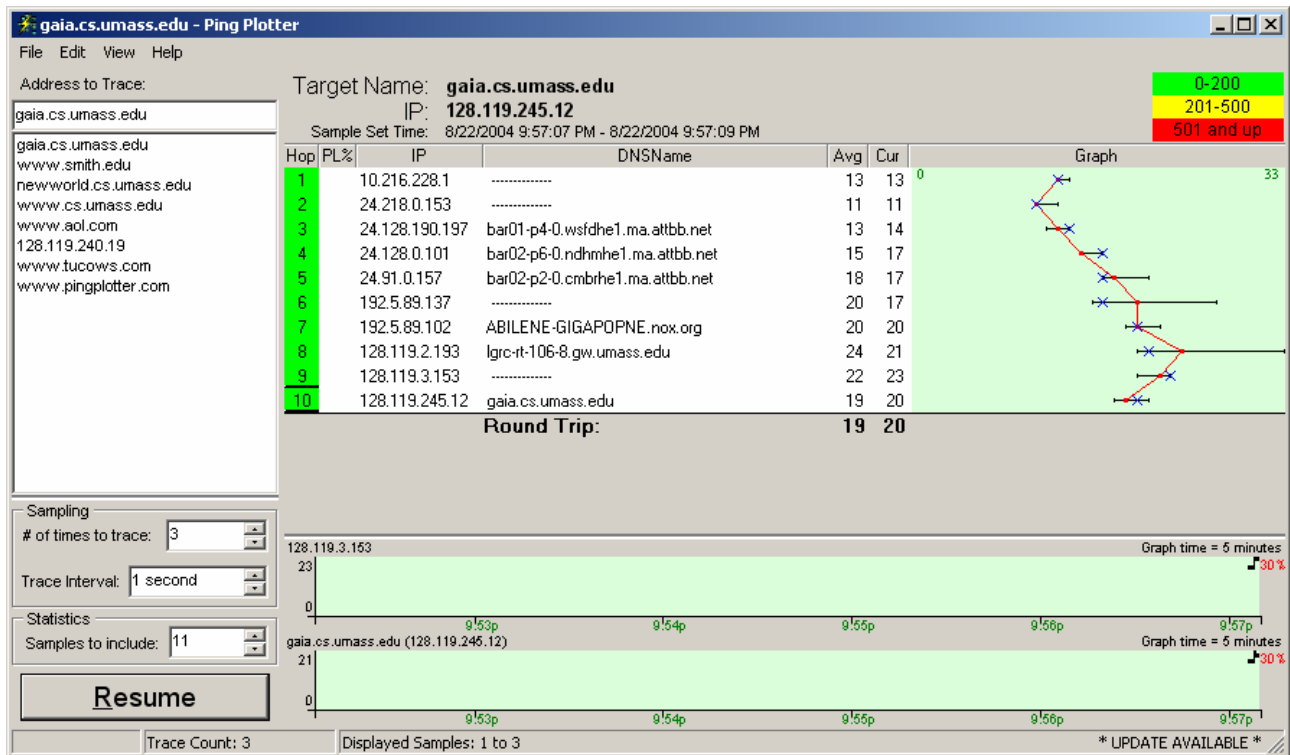
Der Zielhost verschickt dagegen die ICMP Antwort Typ 3: Destination Unreachable, Code 3 Port Unreachable (bei UDP-basiertem Traceroute) bzw. ICMP Echo Replies (bei ICMP-basiertem Traceroute). Die Sequenz der so gesammelten Adressen kennzeichnet den Weg zum Ziel durch das Netz. Der Rückweg ist in der Regel identisch, kann aber bei asymmetrischem Routing anders verlaufen.

Mit dem *tracert* Programm (Windows) ist es nicht möglich, die Größe des ICMP echo requests (ping) einzustellen. Ein komfortables Pendant zu *tracert* ist das Programm *pingplotter*, das mit einer grafischen Oberfläche ausgestattet ist und zudem mehrere Einstellungen bietet, u.a. auch der Größe des ICMP echo requests. Ebenfalls sehr gut geeignet ist die Freeware *FreeIPTools*.

Ablauf des Versuchs:

- Wireshark starten und packet capture (*Capture->Option*) ausführen. Drücken Sie den Button *OK* im Wireshark Packet Capture Options Bildschirm (es müssen hier keine Einstellungen vorgenommen werden).
- Starten Sie nun *pingplotter* und geben Sie den Namen eines Zielhosts im “Address to Trace Window” ein. Geben Sie 3 in das “# of times to Trace” Feld ein, ansonsten werden die Datenmengen eventuell zu groß.

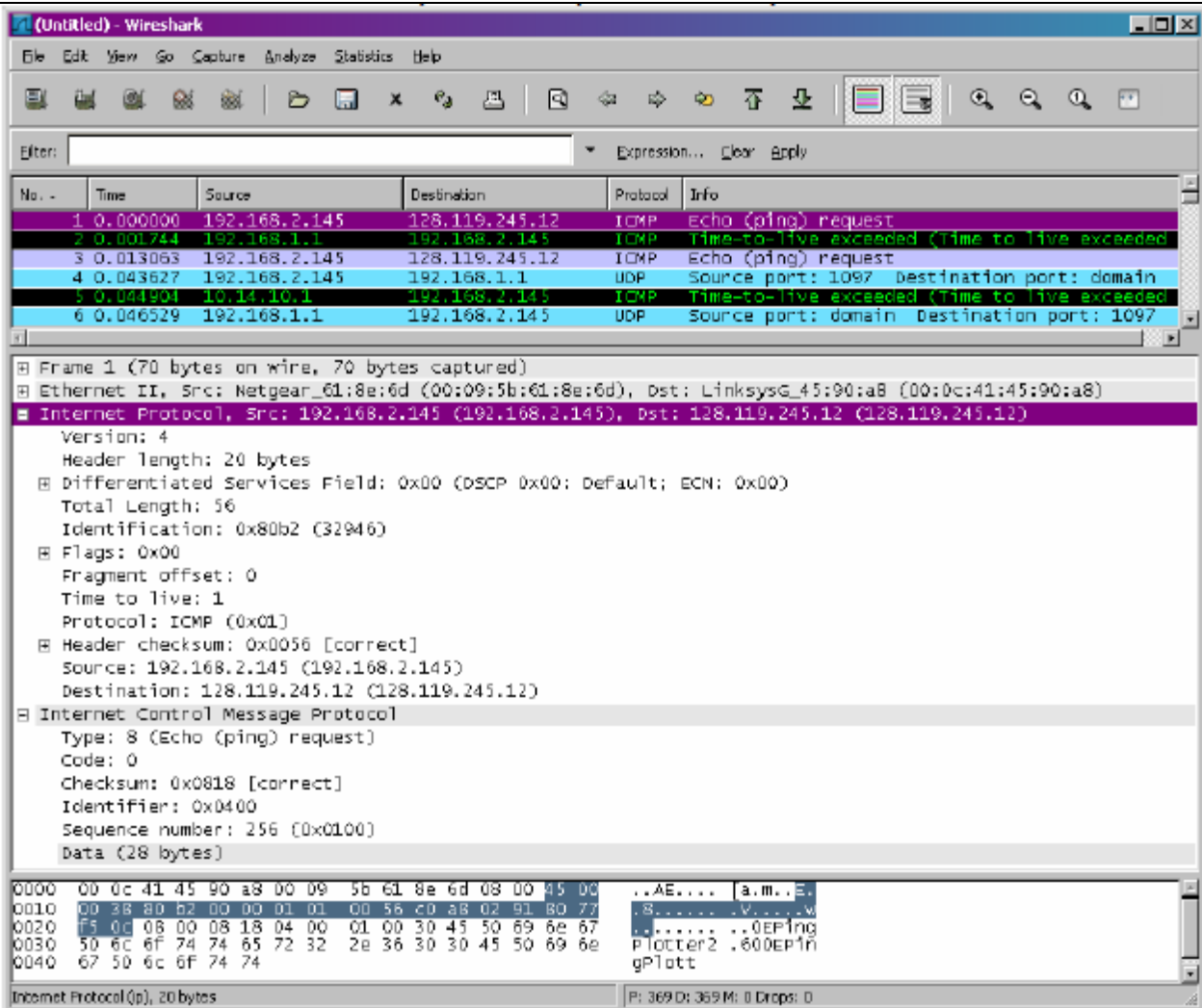
- Drücken Sie nun den Trace Button von *pingplotter* und starten den Wireshark Capture. Das *pingplotter* Fenster sollte etwa so aussehen:



### Untersuchung des Wireshark Capture

Im Trace finden Sie eine Reihe von ICMP echo requests, die von Ihrem Computer gesendet wurden und auch von den Routern zurückgeschickten ICMP TTL-exceeded Nachrichten

- Wählen Sie die erste ICMP Echo Request Nachricht aus, die von Ihrem Computer gesendet wurde und expandieren den Internet Protokoll Teil des Pakets :



- Wie lautet die IP Adresse Ihres Computers?
- Wie lautet der Wert für das Folgeprotokoll im IP-Header?
- Wie viele Bytes enthält der IP-Header? Wieviele Bytes sind in der payload des IP Datagramms? Erklären Sie wie Sie die Zahl dieser Bytes bestimmen?
- Ist dieses IP Datagramm fragmentiert? Woran können Sie sehen, ob ein Datagramm fragmentiert ist oder nicht?

Sortieren Sie nun die aufgezeichneten Pakete nach der IP Quell-Adresse (Klicken auf den Header der Source Spalte). Wählen Sie die erste ICMP echo request Nachricht die von Ihrem Computer gesendet wurde aus.

- Welches Feld im IP Datagramm ändert sich *immer* von einem Datagramm zum nächsten?
- Welche Felder bleiben konstant. Welche Felder müssen konstant bleiben? Welche Felder müssen sich ändern? Warum?
- Beschreiben Sie das Muster, das Sie in den Werten des Identifikationsfelds des IP Datagramms sehen.

Suchen Sie nun die Reihe der ICMP TTL exceeded Antworten die vom *first hop* Router zu Ihrem Computer geschickt wurden.

- Geben Sie den Wert im Identifikationsfeld und im TTL Feld an. Bleiben diese Werte für alle geschickten ICMP TTL-exceeded Antworten unverändert? Warum?

- Sind die ICMP Echo Requests fragmentiert? Welche Information im IP Header liefert diese Information?

Fragmentierung:

Stellen Sie nun die Paketgröße auf einen Wert größer als 1500 Bytes und starten einen neuen traceroute und einen neuen Wireshark Capture

- Sind die Nachrichten fragmentiert? Woran sehen sie das?
- Woran erkennen Sie das erste Fragment und die weiteren Fragmente?
- Welche Felder im IP Header ändern sich zwischen dem ersten und dem zweiten Segment?