

Praktikumsbericht - Einführung 1 (Protokollanalyse)

Inhaltsverzeichnis

1. Einleitung
2. Erste Schritte
3. Kaskadierung von Hubs
4. Weitere Messungen
5. Schlussfolgerung

1. Einleitung

Unsere Gruppe setzt sich zusammen aus Alain Benninger (Multimediaelektroniker) und Stefan Aebischer (Informatik EMF). Wir haben uns für diese Konstellation entschieden, da wir beide sehr unterschiedliche Kenntnisse in der Teleinformatik haben und uns gegenseitig sehr gut helfen können.

Im ersten Teleinformatik-Praktikum, welches am **02.10.2009** statt gefunden hat, lernten wir das Netzwerkanalyseprogramm Wireshark kennen und haben damit diverse Netzwerkverbindungen aufgezeichnet und analysiert.

Wichtig: Sätze mit *[Px]* enthalten die Antworten auf die gestellten Fragen.

2. Erste Schritte

Als erstes fällt uns der moderne Arbeitsplatz für jede Gruppe im Teleinformatikzimmer auf:

- ein Rack mit Netzwerkkomponenten und Analysegeräten
- Mehrere Netzwerk-Anschlussmöglichkeiten, u.a. LAN1(Geswitchtes Netzwerk) und LAN2(Hub)
- Ein Computer mit einer Onboard-Netzwerkkarte und einer zusätzlich PCI (GigaStore)Netzwerkkarte, welche mit einem violetten Kabel am LAN1 verbunden ist *[P1]*

Die Netzwerkgrundeinstellungen *[P2]* des PCs unter Windows XP können via **ipconfig** im Kommandozeilenprogramm (CMD) eingesehen werden. Es werden Informationen wie Netzwerkadapter(-karten), IP Adresse(n), MAC Adresse(n), Gateways und Subnetzmaske angezeigt.

Hier einige relevante Parameter von ipconfig:

ipconfig /all	Zeigt ALLE Netzwerkeinstellungen für alle Netzwerkadapter an
ipconfig /release	Löscht die erhaltene IP Adresse
ipconfig /renew	Erfragt eine neue IP Adresse (zuerst releasen)

Um die Netzwerkeinstellungen zu verändern, z.B. um eine IP Adresse manuell zu vergeben, finden sich in → Netzwerkverbindungen → [Netzwerkadapter] die nötigen Einstellungen.

Wireshark dient dazu den Netzwerkverkehr aufzuzeichnen und zu analysieren.

Webseite: <http://www.wireshark.org/>

Wireshark ist in drei Bereiche aufgeteilt [P3]:

- Oben: Liste mit allen Rahmen (Frames)
- Mitte: Detailinformationen, Rohdaten-Interpretation von Wireshark vom ausgewählten Rahmen
- Unten: Rohdaten (Binär...) des ausgewählten Rahmen

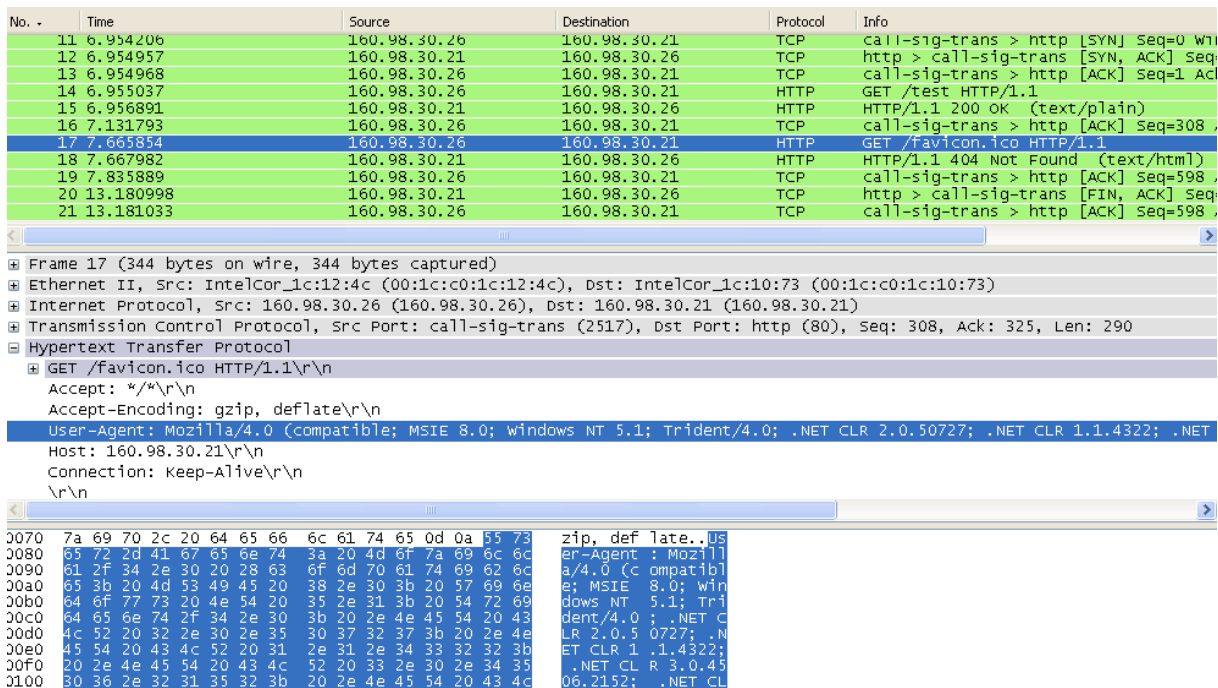


Abbildung 1: Aufbau Wireshark

Doch bevor man überhaupt Rahmen angezeigt bekommt, muss man natürlich den Netzwerkverkehr aufzeichnen. Im Menüpunkt „capture“ kann der Netzwerkadapter mit den entsprechenden Optionen (z.B. Paketfilter anwenden) ausgewählt werden.

Damit wir uns auf die relevanten Informationen konzentrieren können, bietet uns Wireshark unter der Menüleiste eine Filtereingabe an. Wir tippen TCP ein und sehen danach nur noch alle TCP Rahmen. Unter Expressions finden sich weitere Filter.

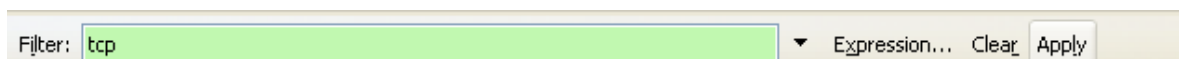


Abbildung 2: Wireshark Filter

Um den aufgezeichneten Netzwerkverkehr zu speichern [P4], klicken wir auf File → save as und können dort verschiedene Formate und Optionen auswählen wie z.B.:

- pcap als Standardformat mit allen Rahmen
- oder nur die ausgewählten Rahmen speichern (selected Packets)
- oder von welchem Rahmen bis zu welchem Rahmen speichern (Range...)

Unter File → export können wir die Rahmen auch in einer Druckerfreundlichen Form speichern. Das Resultat befindet sich in der Anhangdatei „ErsteMessungDruck.txt“.

3. Kaskadierung von Hubs

Unser erstes „Experiment“ sieht folgendermassen aus:

- Wir verbinden alle Laborcomputer via Hub zu einem Netzwerk.
- Unser PC öffnet eine HTML-Seite auf dem Computer des Professors.
- Der dadurch „verursachte“ Netzwerkverkehr wird durch die anderen am Hub angeschlossenen Computer aufgezeichnet. Die ist möglich [P5], weil der Hub in der OSI-Schicht 1 agiert; konkreter: Er leitet die Rahmen (Rohdaten) einfach an alle Teilnehmer weiter, weil er sie nicht „öffnen“ kann (wird auf Schicht 2 gemacht).

Folgend das Schema des erstellten Netzwerkes [P10]:

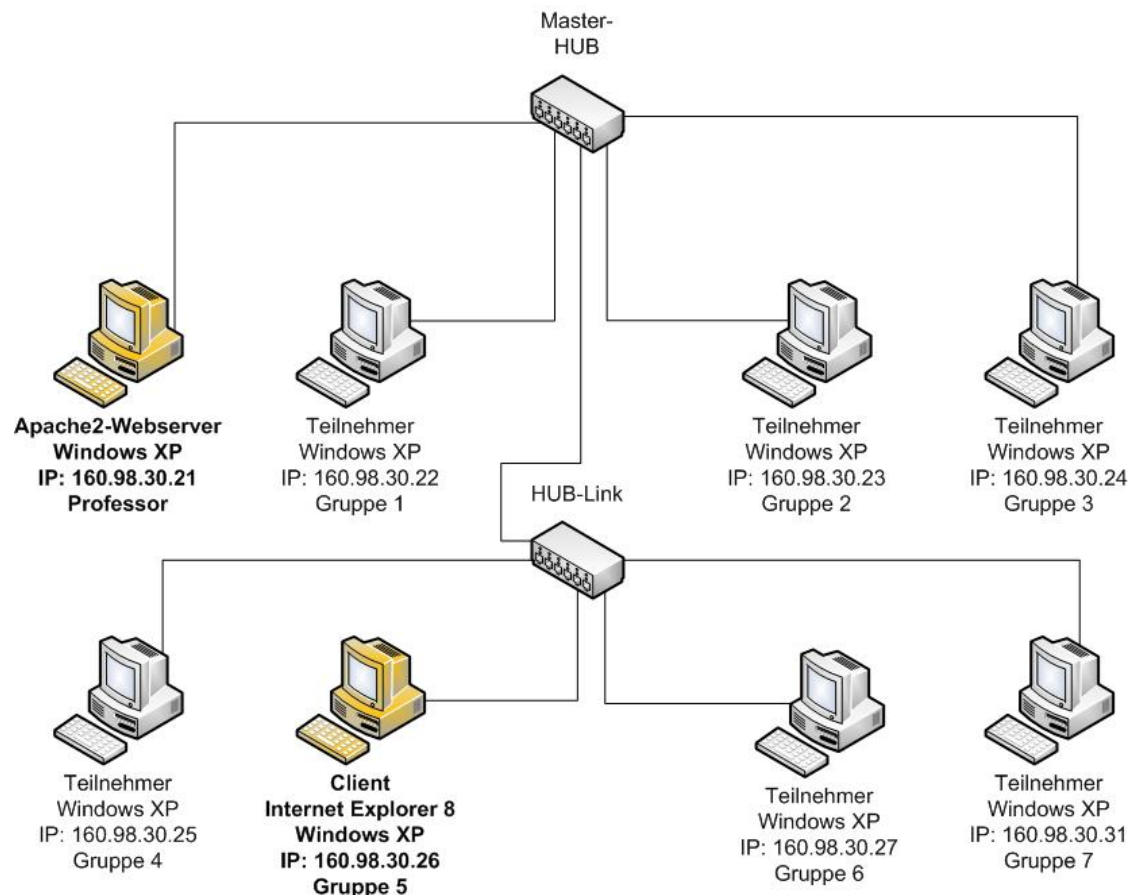


Abbildung 3: Verwendetes Schema

Da unser Hub nur 5 Ports besitzt und wir 7 Computer haben, müssen wir einen zweiten Hub als Unterhub (Baumstruktur) einsetzen. Achtung [P6]: Der zweite Hub kann nicht einfach über seine normalen Ports an den RootHub angeschlossen werden, da die Rx-Tx (empfangen, versenden) Adern der beiden Hubs gleich wären. Entweder benützt man hier ein Crossover-Kabel oder den dafür vorgesehenen Hublink-Port.

Nun starten wir die Messung des HTTP Zugriffs. Es ergibt sich folgende Kommunikation [P11, P12]:

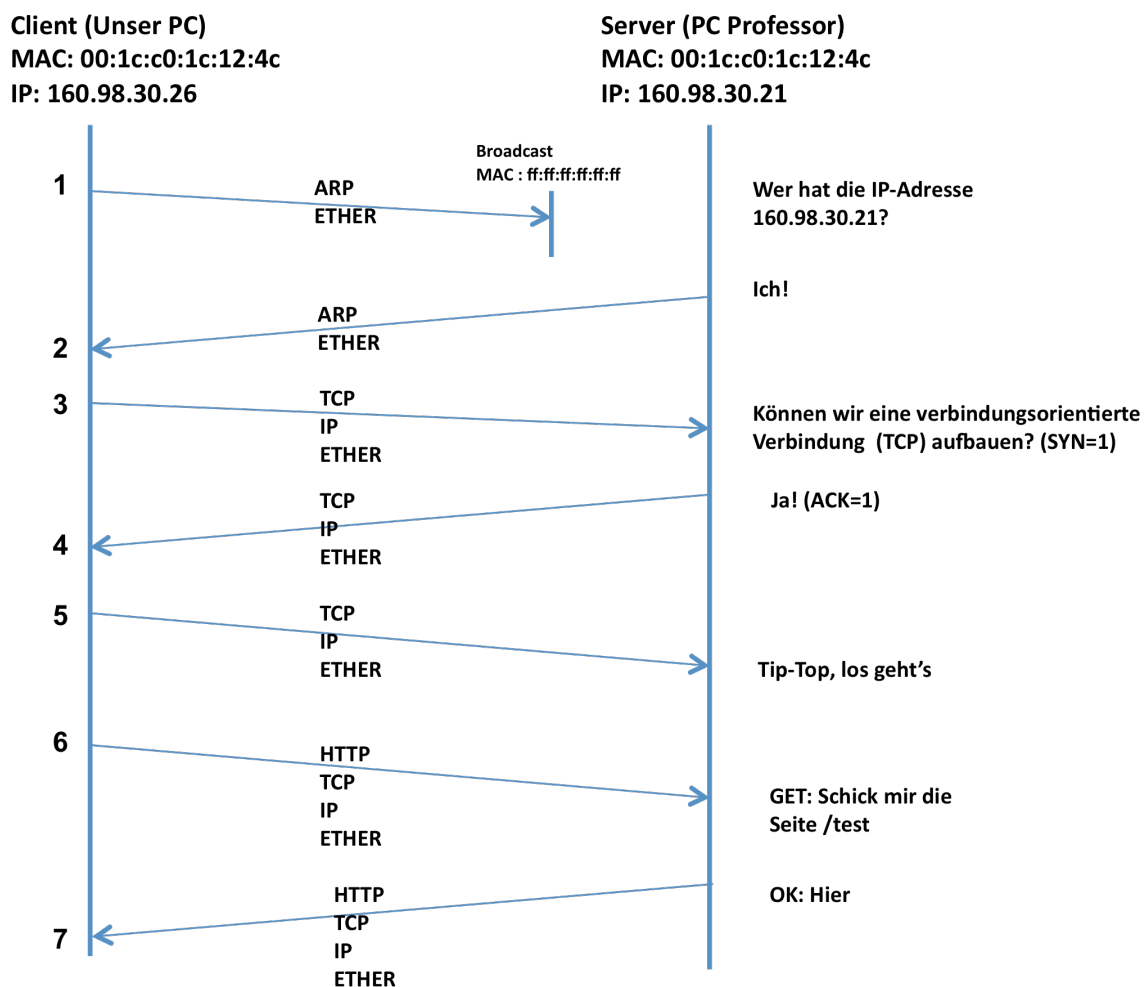


Abbildung 4: Pfeildiagramm

Wenn wir nun das Frame Nr 7 (Die Antwort auf die GET Anfrage) mit Wireshark genauer analysieren, finden wir im HTTP Payload den Text „Welcome to this first Lab!“ [P7], also die „HTML-Seite“. Der Statuscode ist 200 was *OK - Seite existiert und wird übertragen* bedeutet.

Aufbau des Rahmens:

Diesem Text (26 Bytes lang) wird zuerst ein HTTP Header (298 Bytes) hinzugefügt [P9] und dann der TCP Header (20 Bytes). Anschliessend wird es mit dem IP Header (20 Bytes) „verpackt“ und als Ethernet Rahmen verschickt (Ethernet header: 14 Bytes). Das ganze Ethernet Frame (Rahmen) beträgt also 378 Bytes. Will man den Buchstaben W [P8] ganz „unten“ im Rahmen suchen gehen, nimmt man einfach das Byte 353 (298 + 20 + 20 + 14 + 1) zur Hand.

4. Weitere Messungen

Wie schon erwähnt kann man unter „Expressions“ weitere Filter suchen und anwenden. [P13]:

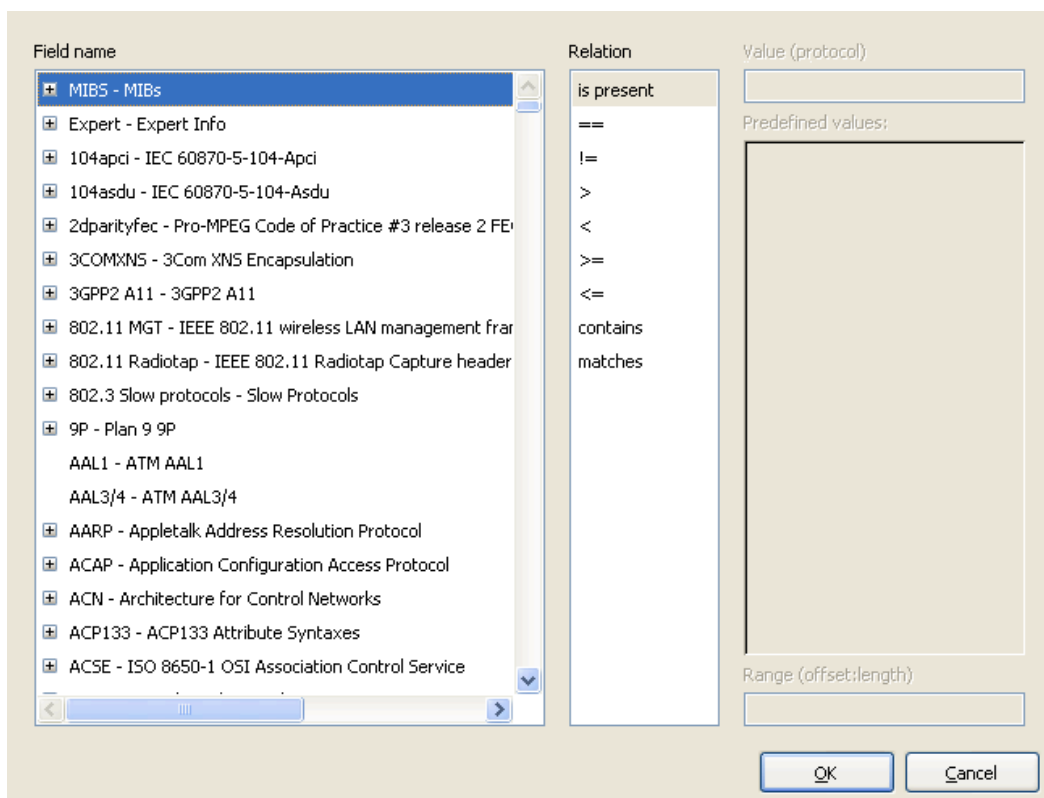


Abbildung 5: Wireshark Filterpalette

Hier könnten wir z.B.

```
http.request.uri contains "/test"
```

als Filter eingeben und wir würden nur den GET HTTP Rahmen (Pfeildiagramm Pfeil Nr 6) angezeigt bekommen.

In unserem Experiment haben wir die Seite mit dem Internet Explorer geöffnet. Dies ist im HTTP GET Rahmen unter User-Agent deutlich zu erkennen.

```
Accept-Language: fr-ch\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .\r\n
Accept-Encoding: gzip, deflate\r\n
Host: 160.98.30.21\r\n
Connection: Keep-Alive\r\n
\r\n
```

Würden wir mit einem anderem Browser [P14] zugreifen, ändert sich dort auch der Eintrag:

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.0.10) Gecko/2009042316
Firefox/3.0.10
```

5. Schlussfolgerung

Dieses erste Praktikum hat uns sehr geholfen, das OSI Modell praktisch zu begreifen und anhand von Wireshark zu analysieren.

Es gab keine grossen Probleme.