

****

**Vorlesung Rechnernetze**

**AIN 5**

**Laborübung**

**Paketanalyse mit Wireshark**

**Prof. Dr. Dirk Staehle**

Die Abgabe erfolgt durch Hochladen der bearbeiteten Word-Datei in Moodle.

**Bearbeitung in Zweier-Teams**

**Team-Mitglied 1: Begüm Peker**

**Team-Mitglied 2: Ellen Peppmüller**

# Einleitung

WireShark ist ein Werkzeug, um Pakete, die über die Netzwerkkarte eines Rechners laufen, aufzuzeichnen und zu analysieren. In dieser Übung sollen Sie ein erstes Gefühl für WireShark bekommen und die Grundzüge der Datenübertragung im Internet auf den verschiedenen Protokollschichten zu verstehen.

WireShark ist auf den Rechnern im Labor bereits installiert, steht aber auch auf der Seite <https://www.wireshark.org/download.html> zum Download zur Verfügung.

Wenn Sie WireShark unter Linux nutzen wollen, können Sie diesen Anleitungen folgen:

* <https://wiki.wireshark.org/CaptureSetup/CapturePrivileges> 🡺 Limiting capture permission to only one group
* <https://wiki.archlinux.org/title/Wireshark>
* <https://itsfoss.com/install-wireshark-ubuntu/>

# WireShark-Umgebung

Um mit WireShark vertraut zu werden, folgen wir dem Einstiegsversuch, wie er in der Laborübung zum Buch „Computer Networks“ von J.F. Kurose und K.W. Ross spezifiziert ist. Die Beschreibung finden Sie auf Moodle oder hier:   
 <http://www-net.cs.umass.edu/wireshark-labs/Wireshark_Intro_v8.0.pdf> .

Änderungen im Vergleich zur Angabe aufgrund der Situation im Labor:

* wenn Sie als „Display-Filter“ http eingeben, werden weiterhin Protokolle wie z.B. SSDP (Simple Service Discovery Protocol) angezeigt. SSDP können Sie abschalten, indem Sie den Filter auf   
   http && !(udp.port==1900)  
  erweitern.
* SSDP wird übrigens von UPnP (Universal Plug and Play) zur Erkennung von Geräten in IP-Netzen genutzt.

# Fragen

Nachdem wir einen ersten Einblick in WireShark gewonnen haben, ein paar Fragen:

1. Nennen Sie mindestens 5 Protokolle, die WireShark erkannt hat.

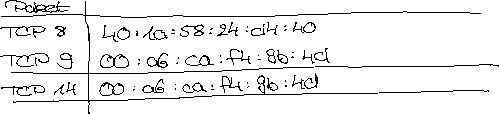
* QUIC, TCP, DNS, ARP, TLSv1.2, UDP/XML, NTP, http, MDNS

1. Wie lange hat es vom Senden des HTTP Requests (<http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html>) bis zum Erhalt der HTTP Response gedauert?

* [Time since request: 0.100624000 seconds]

1. Was ist die Internet-Adresse ihres Rechners? Was ist die Ethernet-Adresse (MAC-Adresse, physikalische Adresse) ihres Rechners? Welches ist die Ziel-MAC-Adresse, zu der ihr Rechner Pakete sendet? Vergleichen Sie die Ziel-MAC-Adresse für verschiedene Ziel-IP-Adressen. Welchem Netzknoten könnten Sie die Ziel-MAC-Adresse wohl zuordnen?

* Ellen: Src: 172.20.138.60; Source: 60:a5:e2:dd:1a:62; Destination: 00:a6:ca:f4:9b:4d



wenn MAC gleich ist beides im gleichen Netz, wenn unterschiedlich dann außerhalb des gleichen Netz also Router oder so

1. Betrachten Sie ein HTTP Paket. Welche weiteren Protokolle werden genutzt, um ein http Paket zu übertragen? Welchen Schichten des ISO/OSI Schichtenmodells können Sie die Pakete zuordnen?

* TCP, Transport (http) und Application (tcp) Layer

# Analyse eines Pakets

Pakete werden von mehreren Protokollschichten verarbeitet und jede Protokollschicht dem Paket, das sie von der höheren Protokollschicht erhält, einen Header hinzu. WireShark greift Pakete an der Netzwerkkarte ab. Wir sehen also alle Protokolle, die dem Paket einen Header hinzugefügt haben.

Das folgende HTTP Paket wurde von Wireshark aufgezeichnet und ist wie im unteren Bereich von WireShark als Hexadezimal-Code und ASCII-Zeichen dargestellt.



0000 38 22 d6 67 19 00 00 21 cc 63 82 2c 08 00 45 00 8".g...!.c.,..E.

0010 02 9c 02 ed 40 00 80 06 40 66 8d 25 1d 5d 5b c6 ....@...@f.%.][.

0020 ae c0 e2 26 00 50 4f 4c 29 24 72 ce 3c d4 50 18 ...&.POL)$r.<.P.

0030 40 b0 62 e7 00 00 47 45 54 20 2f 77 69 6b 69 2f @.b...GET /wiki/

0040 53 69 6d 70 6c 65 5f 53 65 72 76 69 63 65 5f 44 Simple\_Service\_D

0050 69 73 63 6f 76 65 72 79 5f 50 72 6f 74 6f 63 6f iscovery\_Protoco

0060 6c 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 l HTTP/1.1..Host

0070 3a 20 64 65 2e 77 69 6b 69 70 65 64 69 61 2e 6f : de.wikipedia.o

0080 72 67 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 rg..User-Agent:

0090 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e Mozilla/5.0 (Win

00a0 64 6f 77 73 20 4e 54 20 36 2e 31 3b 20 57 4f 57 dows NT 6.1; WOW

00b0 36 34 3b 20 72 76 3a 33 32 2e 30 29 20 47 65 63 64; rv:32.0) Gec

Analysieren sie dieses Paket, indem Sie es mit einem anderen http Paket vergleichen, das in WireShark dargestellt ist. Wenn Sie in WireShark zu einem ausgewählten Paket, Header und Header-Felder im Fenster "details of selected packet headers" markieren, so werden die entsprechenden Bytes des Pakets ebenfalls markiert.

1. Markieren Sie im obigen Paket Ethernet, IP und TCP Header
2. Was sind die Quell- und Ziel-MAC-Adressen des dargestellten Pakets?
3. Was sind die Quell- und Ziel-IP-Adressen des dargestellten Pakets?
4. Was sind die verwendeten TCP-Ports des dargestellten Pakets?

# Filter

Als nächsten Schritt wollen wir Paketfilter kennenlernen. Es gibt Capture-filter und Display-Filter. Capture-Filter legen fest, welche Pakete WireShark sammelt. Display-Filter legen fest, welche Pakete dargestellt werden. Wir verwenden Display-Filter, um der Vielzahl von Paketen Herr zu werden. Einen Filter haben wir bereits kennengelernt: http. Dadurch werden nur Pakete nach dem HTTP Protokoll dargestellt.

Pakete können generell nach den verwendeten Protokollen und nach dem Inhalt der Felder in den verschiedenen Protokoll-Headern gefiltert werden. Wollen wir beispielweise die Pakete betrachten, bei denen der die Ziel- oder Sendeadresse die 192.140.168.15 ist, dann wäre der Filter  
ip.addr==192.140.168.15. Wenn Sie alle Pakete betrachten wollen, die den UDP-Port 1500 auf Sender- oder Empfängerseite verwenden, dann ist der Filter udp.port==1500.

Versuchen Sie, allen HTTPS Verkehr über den verwendeten TCP Port (443) zu filtern.

Mögliche Filteroptionen finden Sie

* bei Eingabe von tcp. im Displayfeld
* indem Sie auf Expression klicken (eher umständlich)
* in der Manpage zu "WireShark Filter" und der Display Filter Übersicht: https://www.wireshark.org/docs/dfref/

Fragen:

1. Wie lautet der Filter, mit dem Sie über den TCP Port https Verkehr filtern können?

* tcp.port == 443

1. Vergleichen Sie die Ergebnisse, wenn Sie http Verkehr direkt filtern (Filter: http) oder http Verkehr über den TCP Port (80) filtern. Können Sie sich die Unterschiede erklären

* OSI Schichtmodell: http nutzt nur Dienste der Application Layer, mit tcp wird auch die transport Layer genutzt, man bekommt mehr Ergebnisse (http und tcp)

1. Es gibt einen Filter http aber keinen Filter https. Haben Sie eine Idee warum?

* Secure: ist verschlüsselte Version von http

1. Welcher Filter bewirkt, dass nur Pakete angezeigt, werden, die ihre eigene IP-Adresse als Ziel-Adresse haben?

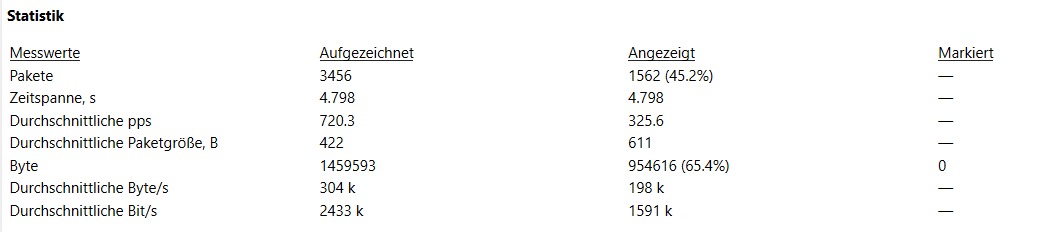
* ip.dst == [IP-Adresse]

# Analyse einer Web-Seite

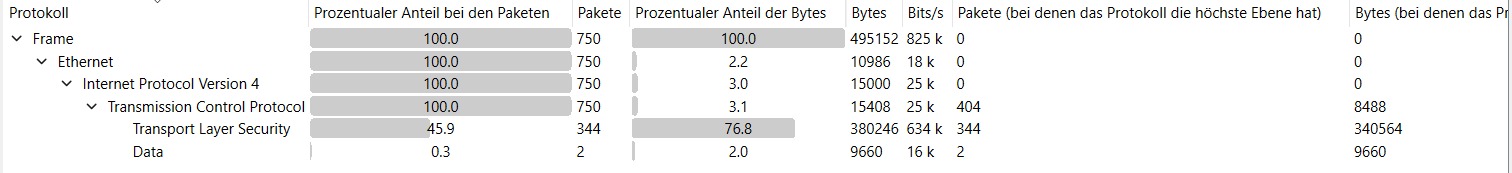
Laden sie z.B. die Web-Seite [www.spiegel.de](http://www.spiegel.de) (oder eine andere populäre Web-Seite) in ihrem Browser und analysieren Sie den https Verkehr. Für diese Aufgabe genügt es, den https Verkehr vom Aufruf der Web-Seite an für weniger Sekunden zu schneiden. Sie können das Ende ungefähr erkennen, wenn Sie z.B. in Firefox parallel das Netzwerkanalyse-Tool nutzen. Versuchen Sie anderen https Verkehr zu vermeiden, d.h. keine anderen Browser und Tabs offen zu haben. Auch andere Anwendungen können https Verkehr produzieren, wenn sie nach Hause funken.

1. Wie viele Pakete wurden insgesamt übertragen? Unterscheiden Sie Upstream- und Downstream Pakete. Upstream- oder Uplink-Pakete sind Pakete, die von ihrem Rechner ins Internet gesendet werden und Downstream- oder Downlink-Pakete sind Pakete, die aus dem Internet zu ihrem Rechner übertragen werden.

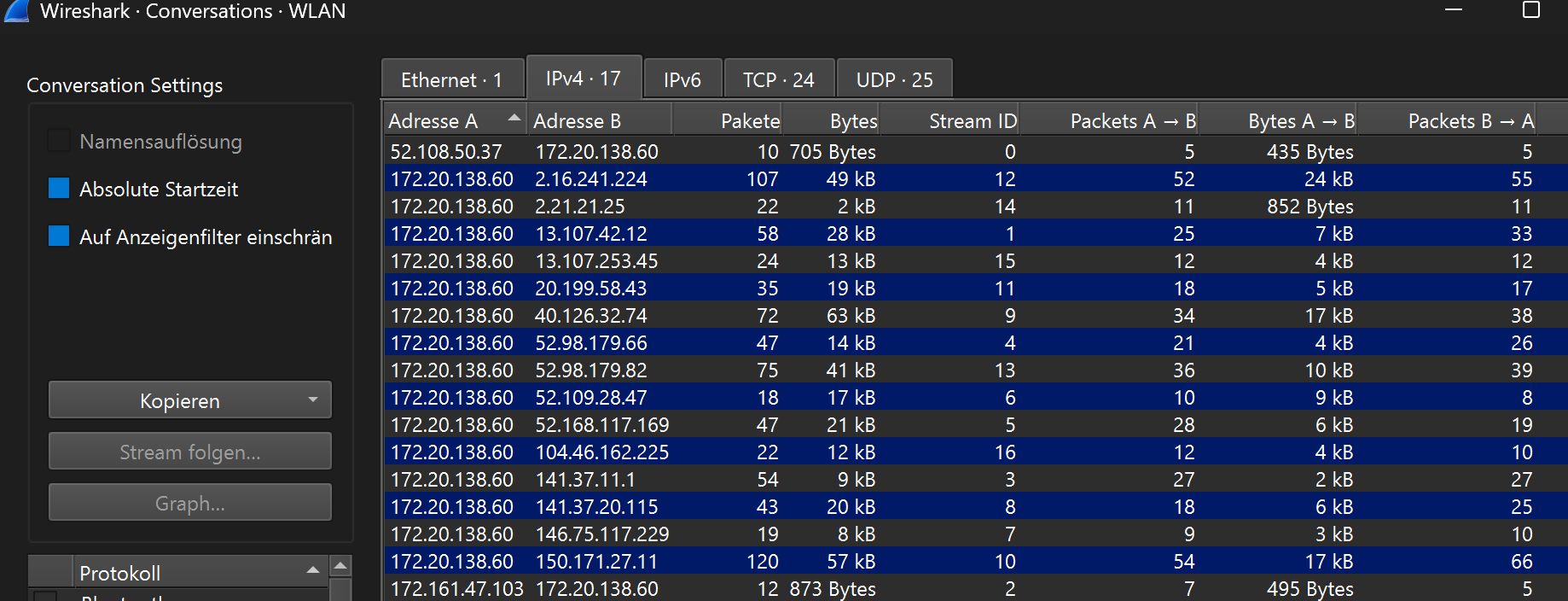
* downlink: (filter: ip.src == IP ADR and tcp.port == 443) 750 Pakete uplink: (filter: ip.dst == … ) 812 Pakete



1. Wie viele Bytes an Daten wurden insgesamt auf dem Uplink und Downlink übertragen?

* Uplink: 16200, Downlink: 15000 (Statistiken > Protokollhierarchie)

1. Von wie vielen IP-Adressen hat ihr Rechner Daten empfangen?



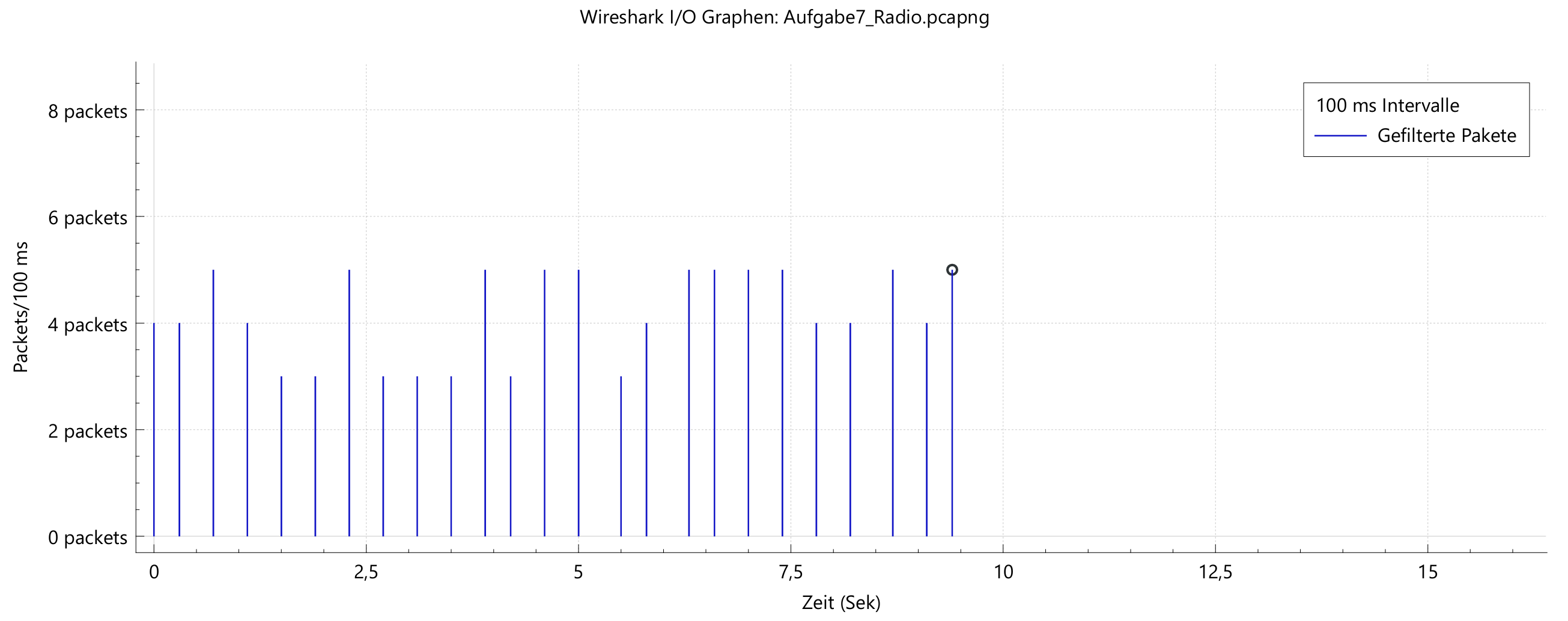
Statistiken > Verbindungen > Adresse B sind externe IP-Adressen

1. Über wie viele TCP Sockets hat ihr Rechner die Daten empfangen? Ein Socket wird über Quell-IP-Adresse, Quell-Port sowie Ziel-IP-Adresse und Ziel-Port identifiziert. -> 1

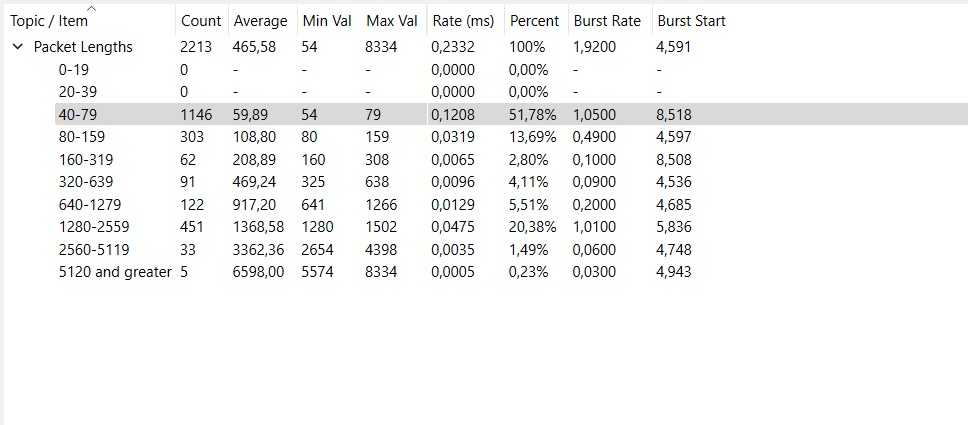
Nutzen Sie dazu die in WireShark angebotenen Tools (Statistiken/Endpunkten, Statistiken/Verbindungen, Statistiken/IPv4 Statistics, Statistiken/IPv6 Statistics)

# Aufzeichnen eines Audio-Streams

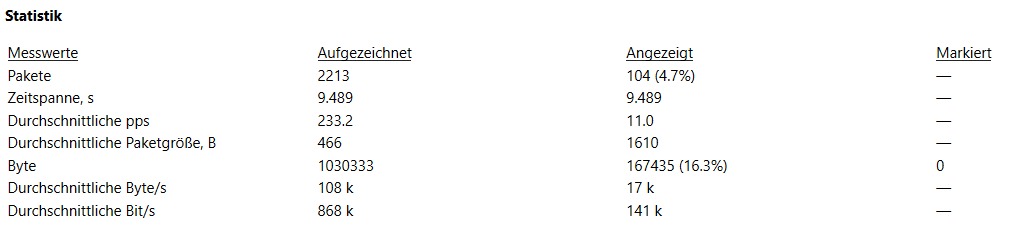
Starten Sie das Webradio von Radio Seefunk (oder einen anderen Stream ihrer Wahl, gerne auch Webex) und schneiden Sie die übertragenen Pakete mit. Speichern Sie das Ergebnis ab und stellen Sie den Paketstrom von Radio Seefunk zu ihrem Rechner grafisch mit Hilfe des Tools I/O-Graph (auch unter Statistiken) dar (x-Achse Ankunftszeitpunkt, y-Achse Paketgröße in Bit). Sie können nur die gefilterten Pakete anzeigen, die Einheiten der y-Achse durch Klicken in der Liste und auch die zeitliche Skalierung einstellen.



Beschreiben Sie, nach welchem Muster die Pakete übertragen werden. Analysieren Sie die Pakete, die zwischen ihrer Anwendung und dem Server ausgetauscht werden (Upstream und Downstream) hinsichtlich Bandbreite, Paketgröße und Zeit zwischen Paketen. Können Sie Regelmäßigkeiten feststellen?

* 55.84kb / 400ms, ungefähr 400ms ab und zu Ausschweifungen 300ms oder 500ms
* Bei höheren Paketen (5) sind die zeitlichen Ausschweifungen zu sehen (je mehr Pakete desto größer ist die Zeit zwischen den Paketen)
* Paketgrößen:

Statistik > Paketgrößen



Durchschnittliche Paketgrößen

Nutzen Sie dazu die in WireShark angebotenen Tools.

Alternativ zu dem Audio-Stream von Radio-Seefunk können sie auch eine andere Anwendung wählen, z.B. ein YouTube-Video, eine Counterstrike-Session oder eine Video-Konferenz.