



WireShark/Ethereal

Bearbeiter: A. Lebedev, ET02wK1

© Professur Kommunikationstechnik, Prof. Dr.-Ing. habil. Lutz Winkler
Hochschule Mittweida (FH) – University of Applied Sciences, Fakultät Elektro- und Informationstechnik



win@hs-mittweida.de



<https://www.telecom.hs-mittweida.de>



WireShark ist ein leistungsfähiges Netzwerkanalysetool. Es ist aus **Ethereal**

hervorgegangen. Damit kann man folgende Aufgaben lösen:

- Mitlesen von Protokolldateneinheiten einer Netzwerk-Kollisionsdomäne und deren Speicherung,
- Auswertung mitgelesenen Protokolldateneinheiten bei verschiedener Detaillierung.

Ethereal/WireShark ist eine Open Source Software. Die aktuellste Programmversion

sowie Dokumentationen findet man unter <http://www.wireshark.org>

Diese Kurzbeschreibung und alle verwendete Bilder beziehen sich auf die Ethereal-Version 0.10.10.

Die WireShark-Oberfläche hat sich leicht verändert, ist aber im Wesentlichen unverändert. Insbesondere die zahlreichen Capture- und Anzeigefilter haben nach wie vor Gültigkeit.

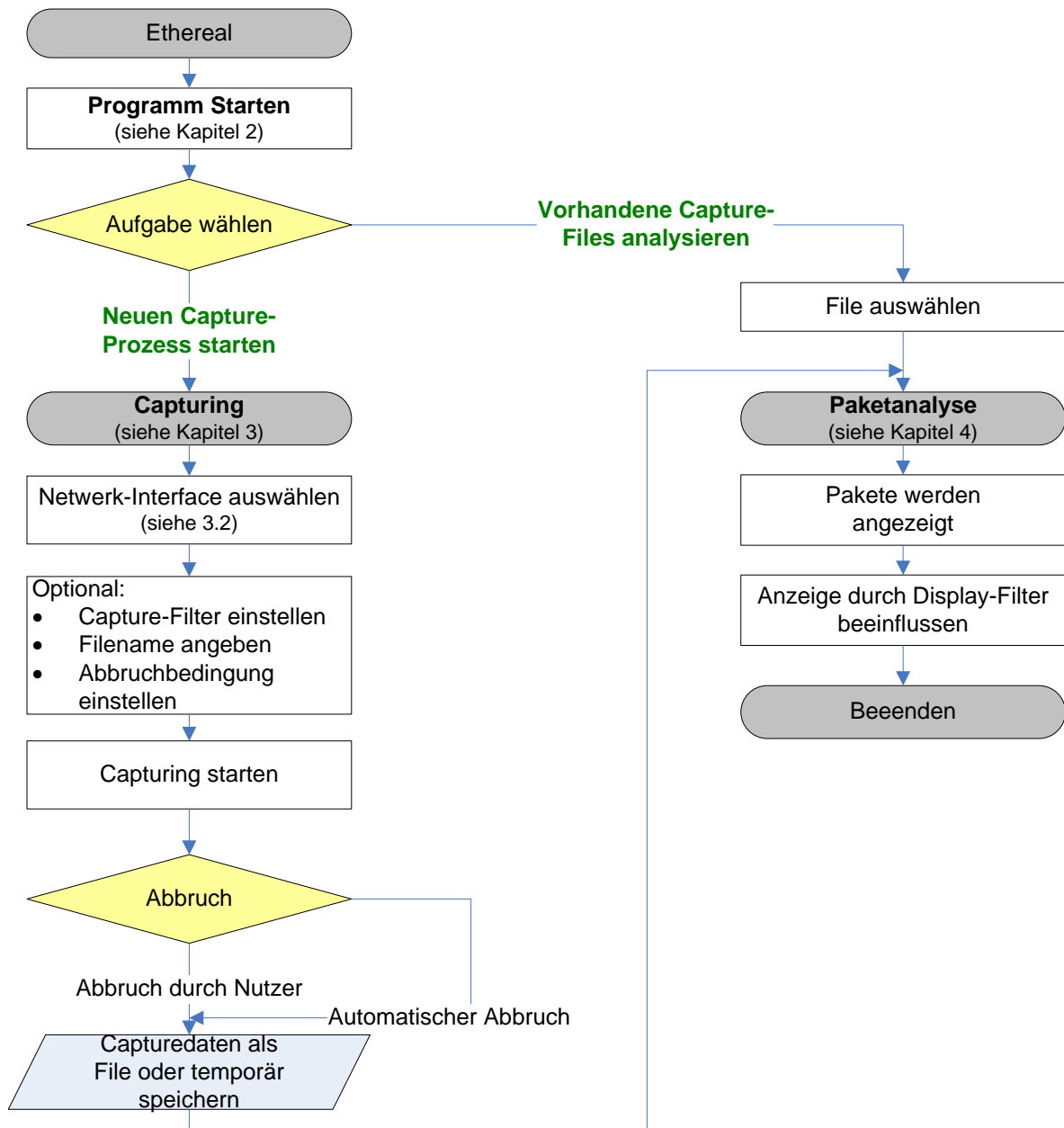
ACHTUNG: Mit **WireShark** kann man Netzwerkprotokolle kennen lernen, Fehler in Netzwerken aufspüren aber auch alle Daten einer Kollisionsdomäne mitschneiden. Der private Einsatz solcher Netzwerkanalysetools in Bus- oder Hubnetzen ist kriminell und deshalb zu unterlassen.

1 Programmablauf.....	2
2 Programmoberfläche	3
2.1 Das Hauptfenster	3
2.2 Menüleiste.....	3
2.3 Steuerleiste	4
3 Capturing.....	5
3.1 Start und Capture-Optionsassistent.....	5
3.2 Interfaceauswahl	5
3.3 Capturefilter.....	6
3.4 Start des Captureprozesses	9
4. Analyse.....	10
4.1 Auswahl der Analysedaten.....	10
4.2 Erster Schritt	10
4.3 Paketdarstellung	11
4.3.1 Die Paketliste	11
4.3.2 Paketdetailsliste	11
4.3.3 Paketinhaltsliste	12
4.4 Statusleiste	12
4.5 Displayfilter	12
4.5.1 Einführung.....	12
4.5.2 Displayfilterleiste.....	12
4.5.2.1 Displayfiltereingabe	13
4.5.3 Displayfiltersyntax	14
4.5.3.1 Einführung	14
4.5.3.2 Vergleichsoperatoren	14
4.5.3.3 Protokollelementtypen.....	14
4.5.3.4 Logische Verknüpfungen	15
4.5.3.5 Substring-Operator	15
4.5.3.6 Beispiele.....	15
4.5.4 Filter Expression - Assistent	16
4.5.5 Farbige Darstellung einzelner Pakete	16
Anlage: Spezielle Capturefilter	18
IP-bezogene Filter	18
Elemente-Indizierung des Arrays ip:.....	18
IP-ToS-Feld (Type of Service)	19
IP-Protokoll-Feld	19
TCP-bezogene Filter	20
Elemente-Indizierung des Arrays tcp:	20
UDP-bezogene Filter	20
Elemente-Indizierung des Arrays udp:	20
ICMP-bezogene Filter	21
Elemente-Indizierung des Arrays icmp:	21

1 Programmablauf

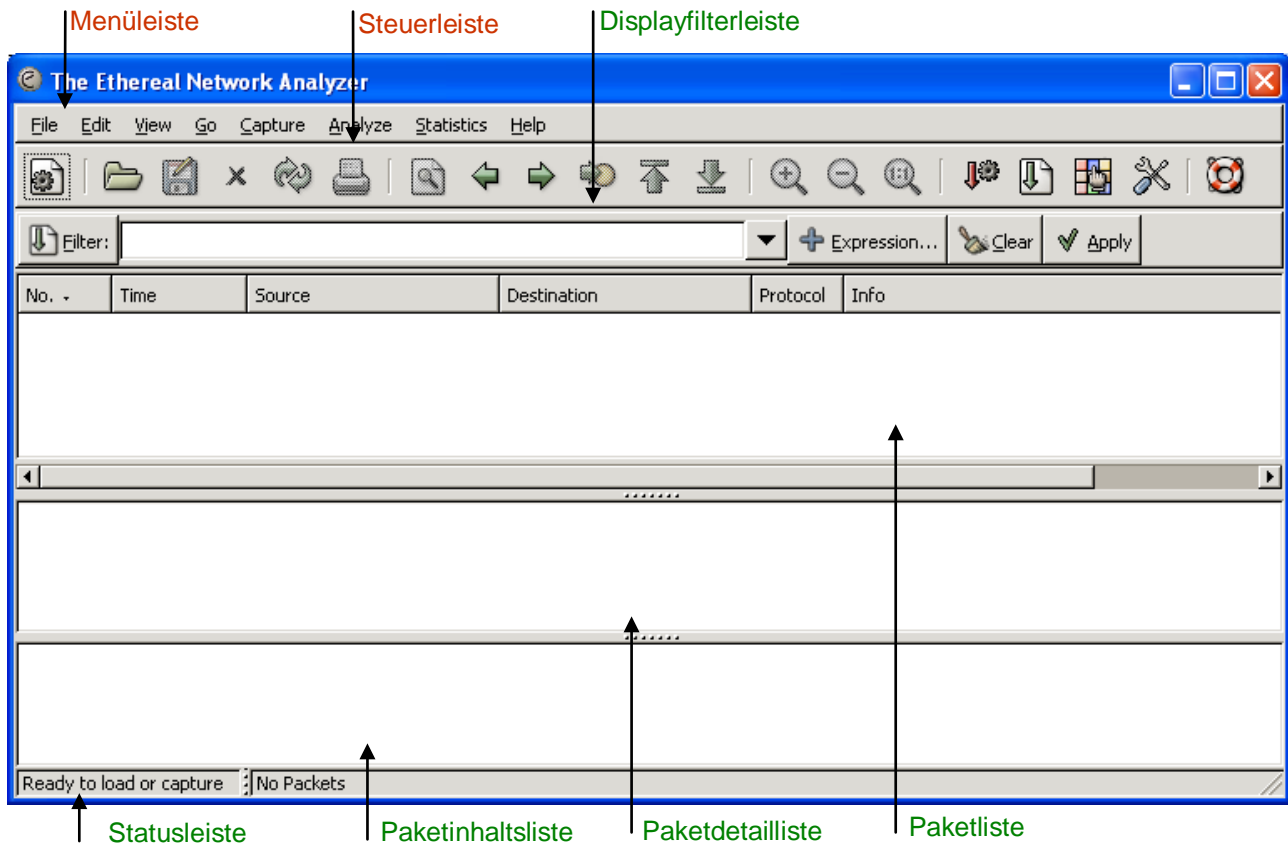
Mittels **Ethereal** kann man Protokollnachrichten aufzeichnen und aufgezeichnete Protokollnachrichten analysieren. Aus der Abbildung sind diese Hauptabläufe erkennbar:

- (1) Programm starten, neuen Capture-Prozess starten und ausführen, anschließend aufgenommene Paket-Daten analysieren.
- (2) Programm starten, vorhandene Capture-Files laden und anschließend Paket-Daten analysieren.



2 Programmoberfläche

2.1 Das Hauptfenster

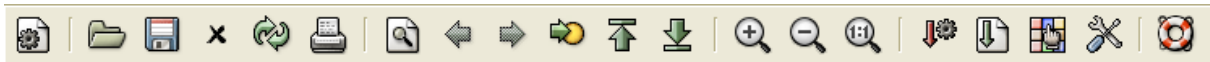


2.2 Menuleiste



Menü	Kurzbeschreibung
File	Öffnen und Vereinigen von Capture-Files, Speichern, Drucken, Export von Capture-Files im Ganzen oder teilweise, Verlassen des Programms
Edit	Suchen eines Paketes, Zeitbezugnahme oder Markierung eines oder mehrerer Pakete, Parametereinstellung
View	Steuerung von Darstellungsoptionen der Paketliste: Farben, Schriftart -größe, Anzeige eines Paketes in separatem Fenster, Zeigen/Verbergen von Details usw.
Go	Steuerung in der Paketliste
Capture	Start/Stop des Captureprozesses, Einstellen des Capturefilters
Analyze	Einstellungen des Anzeigefilters, Aktivieren/Deaktivieren der Protokolldekodierung, Konfiguration nutzerspezifischer Decoder usw.
Statistics	Darstellung statistischer Informationen wie: Anzahl der Pakete, Protokollhierarchiestatistik usw.
Help	Hilfe, Liste der unterstützten Protokolle, Online- Hilfe, Link zur Entwicklerseite und "about"-Information.

2.3 Steuerleiste




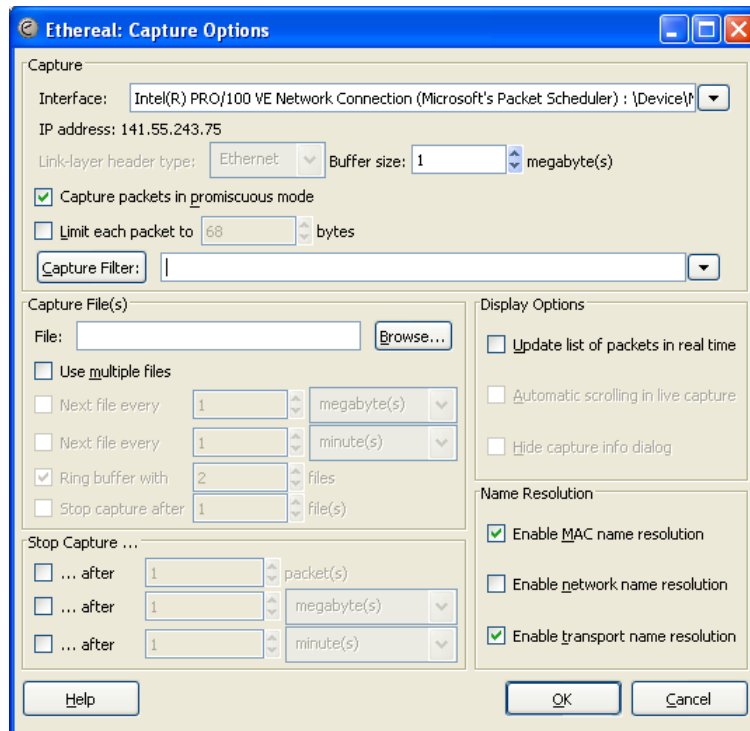
Button	Menü/Befehl	Funktion
	Capture/Start	Start eines neuen Captureprozesses
	File/Open	Öffnen eines gespeicherten Capturefiles um z.B. die Protokolldaten zu analysieren
	File/Save As...	Speichern der der Protokolldaten in ein Capturefile
	File/Close	Schließen des aktuellen Capturefiles
	View/Reload	Laden des aktuellen Capturefiles neu
	File/Print	Drucken aller oder ausgewählter Pakete in ein File, auf den Drucker
	Edit/Find Packet	Paketsuche nach Typ (z.B. tcp, http) oder Inhalt (Hexwert, String)
	Go/Back	In Pakethistorie rückwärts gehen
	Go/Forward	In Pakethistorie vorwärts gehen
	Go/Go to Packet	Sprung zu einer angegeben Paketnummer
	Go/First Packet	Sprung zum ersten Paket
	Go/Last Packet	Sprung zum letzten Paket
	View/Zoom In	Schriftvergrößerung
	View/Zoom Out	Schriftverkleinerung
	View/Normal Size	Normale Schriftgröße
	Capture/Capture Filters	Einstellung des Capturefilter
	Analyse/Display Filters	Displayfilter editieren, anwenden, speichern usw.
	View/Coloring Rules	Farbfestlegungen editieren, anwenden, speichern usw.
	Edit/Preferences	Ethereal-Basiseinstellungen
	Help/Contents	Ethereal-Hilfe

Displayfilterleiste, Paketliste, Paketdetailleiste, Paketinhaltsleiste und **Statusleiste** werden im Kapitel 4 Analyse beschrieben.

3 Capturing

3.1 Start und Capture-Optionsassistent

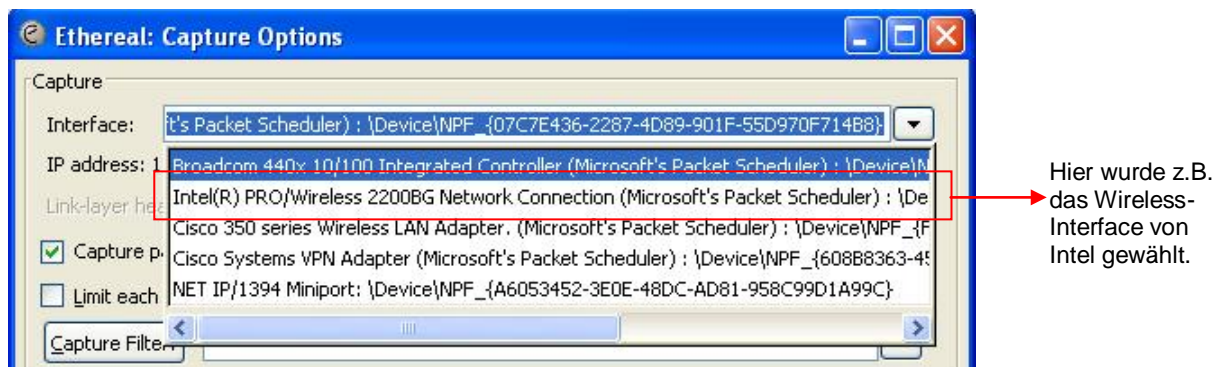
Button **Start a new live capture**  oder **Capture/Start** in der Menuleiste auswählen. Danach erscheint folgendes Dialogfenster.



Hier können verschiedene Optionen zum Captureprozess eingestellt werden, wie z.B.: das Interface von dem aufgenommen werden soll, Capturefilter zur Steuerung der Aufnahme, Capturefile zum Speichern, Displayoptionen und automatische Abbruchbedingungen. Weitere Details zu den einzelnen Optionen findet man in **Help/Contents/Capturing** oder unter: <http://www.ethereal.com/docs/>

3.2 Interfaceauswahl

Vor dem Start der Capturing muss ein Capture-Interface unbedingt eingestellt werden. Der Prozess kann sonst nicht gestartet werden.



3.3 Capturefilter

Nachfolgend werden oft benötigte Capturefilter dargestellt.

[src dst] host <ip-address host-name>	
Filtern von Paketen die von/zu einem Host kommen/gehen mittels der Schicht-3-Adresse (IP-Adresse). Der Host wird durch seine numerische Adresse oder seinen Namen adressiert. Mit src dst kann man einstellen, ob man alle kommenden gehenden Pakete aufnehmen will. Ist src dst nicht deklariert, werden alle kommenden und gehenden Pakete aufgenommen.	
src host 10.10.10.10	Pakete die von 10.10.10.10 kommen
dst host 141.55.192.70	Pakete die zu 141.55.192.70 gehen
host 141.55.192.70	Pakete die von/zu 141.55.192.70 kommen/gehen
src host www.htwm.de	Pakete die von www.htwm.de kommen

ether [src dst] host <ehost>	
Filtern von Paketen die von/zu einem Host kommen/gehen mittels der Schicht-2-Adresse (MAC-Adresse). Der Host wird durch seine MAC-Adresse adressiert. Mit src dst kann man einstellen, ob man alle kommenden oder alle gehenden Pakete aufnehmen will. Ist src dst nicht deklariert, werden alle kommenden und gehenden Pakete aufgenommen.	
ether src host 00:07:77:64:09:32	Pakete die von 00:07:77:64:09:32 kommen
ether dst host 00:07:77:64:09:32	Pakete die zu 00:07:77:64:09:32 gehen
ether host 00:07:77:64:09:32	Pakete die zu/von 00:07:77:64:09:32 gehen/kommen

[src dst] net <net> [mask <mask> len <len>]	
Filtern von Paketen die von/zu einem Netzwerk kommen/gehen mittels der Schicht-3-Netz-Adresse (IP-Netzadresse). Mit src dst kann man einstellen, ob man alle kommenden oder alle gehenden Pakete aufnehmen will. Ist src dst nicht deklariert, werden alle kommenden und gehenden Pakete aufgenommen.	
src net 10	Pakete die vom Netz 10 kommen
dst net 141.55	Pakete die zu Netz 141.55 gehen
net 141.55	Pakete die von/zu Netz 141.55 kommen/gehen
net 141.55 mask 255.255.0.50	Pakete die von/zu Netz 141.55 kommen/gehen aber nur jene, deren Adresse zusätzlich der Hostmaske genügt. Hier würden alle Pakete aufgenommen, die als Netzadresse 141.55. haben und zusätzlich an den markierten Maskenstellen 00000000.01010000 eine 1 in der Adresse haben.

[tcp udp] [src dst] port <port>	
Mit diesem Ausdruck kann man auf Ports ¹ der Schicht-4-Protokolle TCP und UDP , filtern. Beachte: [tcp udp] muss vor [src dst] stehen.	
port 80	Pakete die von/zu Port 80 kommen/gehen, egal ob UDP oder TCP
tcp dst port 80	Pakete, die zu TCP-Port 80 gehen
udp port 4987	Pakete, die zu UDP-Port 4987 gehen

¹ Alle Portnummer mit dazu gehörenden Diensten findet man in:
%WINDIR%\system32\drivers\etc\services oder unter:
<http://www.iana.org/assignments/port-numbers>

<protocol>	
Mit diesem Ausdruck kann nach Protokollen gefiltert werden. Solche Protokolle sind beispielsweise: icmp, igmp, igrp, pim, ah, esp, vrrp, udp, tcp usw.	
tcp	Alle TCP-Pakete
udp	Alle UDP-Pakete.
icmp icmp6 igmp igrp pim ah esp vrrp moprc mopdl lat sca decent atalk rarp arp ip ip6 aarp iso stp ipx netbeui	Alle nebenstehenden Protokolle, wobei die wichtigsten fett dargestellt sind.

proto \<protocol>	
Mit diesem Ausdruck kann nach Protokollen gefiltert werden. Solche Protokolle sind: tcp, udp, ip, icmp Beachte: Da tcp,udp, ip, icmp Schlüsselworte sind, müssen sie mit einem Escape-Zeichen, hier der Backslash, versehen werden (\tcp, \udp, \ip, \icmp).	
proto \tcp	Alle TCP-Pakete
proto \udp proto \ip proto \icmp	Alle UDP-Pakete Alle IP-Pakete Alle ICMP-Pakete

ip ip6 proto \<protocol>	
Mit diesem Ausdruck können Protokolle gefiltert werden, die direkt IP nutzen. Solche Protokolle sind: icmp, udp, tcp. Beachte: Da tcp,udp, ip, icmp Schlüsselwörter sind, müssen sie mit einem Escape-Zeichen, hier der Backslash, versehen werden (\tcp, \udp, \icmp).	
ip proto \tcp	Alle TCP-Pakete
ip proto \udp ip proto \icmp	Alle UDP-Pakete Alle ICMP-Pakete

ether proto \<protocol>	
Mit diesem Ausdruck können Protokolle gefiltert werden, die direkt Ethernet nutzen. Solche Protokolle sind: ip, ip6, arp, rarp atalk, decent sca, lat, mopdl, moprc. Beachte: Da ip, ip6, arp, rarp usw. Schlüsselworte sind, müssen sie mit einem Escape-Zeichen, hier der Backslash, versehen werden (\ip, \ip6, \arp, \rarp).	
ether proto \ip	Alle IP-Pakete
ether proto \ip6 ether proto \arp ether proto \rarp ether proto \atalk ether proto \decent ether proto \sca ether proto \lat usw.	Alle IP6-Pakete Alle ARP-Pakete Alle RARP-Pakete Alle ATALK-Pakete Alle DECNET-Pakete Alle SCA-Pakete Alle LAT-Pakete usw.

[ether ip] broadcast multicast	
Mit diesem Ausdruck können Pakete gefiltert werden, die multicast bzw. broadcast in der Schicht 3 oder Schicht 2 oder beiden Schichten gesendet werden.	
broadcast multicast	Alle Pakete die broadcast multicast gesendet/empfangen werden in Schicht 3 und 2
ip broadcast ip multicast	Alle IP-Pakete die broadcast multicast gesendet/empfangen werden
ether broadcast ether multicast	Alle Ethernet-Pakete die broadcast multicast gesendet/empfangen werden.

less greater <length> bzw. len <= >= <length>		
Dieser Ausdruck filtert Pakete dessen Länge kleiner oder gleich bzw. größer oder gleich <length> ist.		
less 80 len <= 80	Pakete die gleich oder kleiner als 80 Byte sind.	
greater 1024 len >= 1024	Pakete, die gleich oder größer 1024 Byte sind.	
Für Vergleichsausdrücke in den Längenangaben gilt folgende Notation:		
less 80	len <= 80	Pakete, die gleich oder kleiner 80 Byte sind
	len < 80	Pakete, die kleiner als 80 Byte sind
greater 80	len >= 80	Pakete, die gleich oder größer 80 Byte sind
	len > 80	Pakete, die größer 80 Byte sind
	len = 80	Pakete, die gleich 80 Byte sind

→ Capturefilter können zusätzlich logisch verknüpft werden.

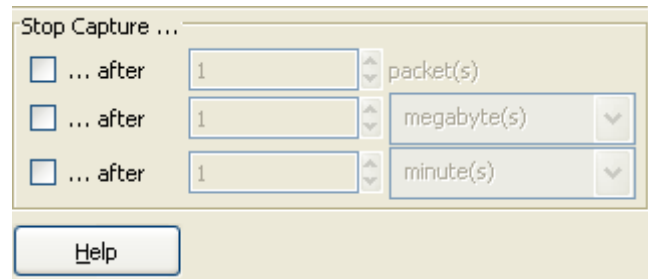
Beispiele für logische Verknüpfungen		
ip and less 80	IP-Pakete die gleich oder kleiner als 80 Byte sind.	
ether proto \ip && len>512	Ethernet-Pakete, die IP-Pakete transportieren und gleich oder größer 512 Byte sind.	
dst host 141.55.192.70 && port 80	Pakete, deren Ziel-IP-Adresse 141.55.192.70 ist und die zum Port 80 gehen.	
ip && ! src net 141.55	IP-Pakete deren Quelle nicht im Netz 141.55 liegt	
icmp[0]= 0 or icmp[0]= 8	ICMP-Pakete, deren Wert im Headerfeld [0] entweder 0 (d.h. Ein Ping-Replay) oder 8 ist (d.h. ein Ping-Request) ist.	
ip[0]&0x0f= 5 && !src net 141.55	IP-Pakete, deren Headerlänge 20 Byte groß ist (5*4Byte) und die nicht vom Netz 141.55. kommen.	
Für logische Verknüpfungen gibt es folgende Operatoren		
and &&	UND-Verknüpfung von Bedingungen	
or 	ODER-Verknüpfung von Bedingungen	
not !	Ausschluss von Bedingungen	
Allgemeine Deklaration logischer Ausdrücke für Capturefilter		
[not] primitive (and or) [not] primitive		
[!] primitive (&&) [!] primitive		

Bei einigen Protokollen, wie z.B. IP, TCP, UDP, ICMP ist es möglich nach einem bestimmten Wert in einer bestimmten Position zu filtern, z.B. nach der Headerlänge im IP-Protokoll oder einem Flag bei TCP. Der Protokollheader wird dabei als Array aufgefasst, beginnend mit [0].
Nachfolgend einige Beispiele, weitere Einzelheiten dazu → siehe **Anlage A**.

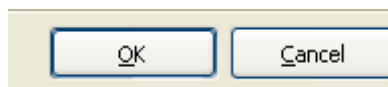
Capture-Filter-Aufgabe	Filterstring
IP-Pakete mit Header-Länge =5*4Byte=20 Byte	ip[0:1]&0x0f=0x05 ip[0:1]&0x0f=5
IP-Pakete, deren TTL < 128 sind	ip[8:1]<128 ip[8:1]<0x80
IP-Pakete, deren Protokollfeld den Wert 6 hat (TCP)	ip[9:1]=6 ip[9:1]=0x06
IP-Pakete mit Source-Adresse 141.55.192.70	ip[12:4]=0x8d37c046
IP-Pakete mit Destination-Adresse 141.55.193.21	ip[16:4]=0x8d37c115
TCP-Pakete, deren Source Port 80 ist	tcp[0:2]=0x0050 tcp[0:2]=80
TCP-Pakete, deren Destination Port 3700 ist	tcp[2:2]=0x0e74 tcp[2:2]=3700
TCP-Pakete mit Header größer/gleich 20 Byte	tcp[12:1]&0xf0>5 tcp[12:1]>0x50
TCP-Pakete vom Typ ACK	tcp[13:1]=0x10
TCP-Pakete mit Window-Feld größer/gleich 1000	tcp[14:2]>1000 tcp[14:2]>0x03e8
ICMP-Pakete vom Typ Echo Request (ping)	icmp[0:1]=8

3.4 Start des Captureprozesses

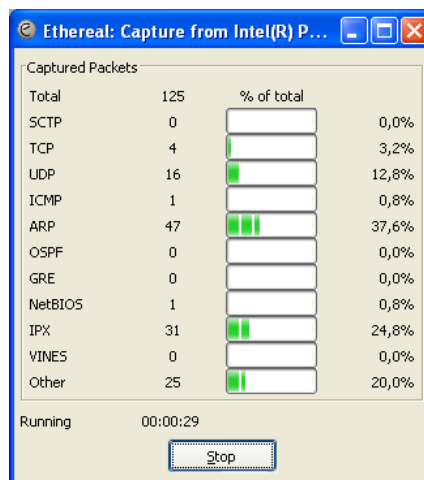
Nachdem ein Capture-Interface ausgewählt und ein Capture-Filter eingestellt wurde oder auch nicht, können zusätzlich Ende-Bedingungen für den Captureprozess eingestellt werden:



Mit **OK** werden alle vorgenommenen Einstellungen bestätigt und angewendet:



Nach dem **OK** werden in einem extra Fenster der Typ und die Anzahl der aufgenommenen Pakete sowie die Dauer der Aufzeichnung angezeigt:



Mit **Stop** kann der Captureprozess jederzeit manuell durch den Nutzer beendet werden, egal ob eine Ende-Bedingung programmiert wurde oder nicht.

Wenn die Aufzeichnung beendet ist, erscheint automatisch das Hauptfenster mit den aufgenommenen Paketen.

4. Analyse

4.1 Auswahl der Analysedaten

Ethereal ermöglicht die Datenanalyse gerade aufgenommener Pakete oder die Datenanalyse von gespeicherten Capturedaten. Weiterhin können auch Capturedaten von anderen Sniffer-Tools (z.B. Kismet, <http://www.kismetwireless.net>) analysiert werden. Kismet ist ein passiver WLAN-Sniffer.

4.2 Erster Schritt

Zur Datenanalyse besteht die Möglichkeit, die einzelnen Spalten der Paketliste nach **No.** (Nummer), **Time** (Zeit), **Source** (Quelladresse), **Destination** (Zieladresse), **Protocol**, **Info** aufsteigend oder absteigend zu sortieren. Hierfür muss lediglich der jeweilige Spaltenkopf angeklickt werden.

The screenshot shows the Ethereal network sniffer interface. The main window displays a list of captured packets. A filter is applied: `tcp.srcport == 80`. The packet list shows various TCP and HTTP traffic between 141.55.193.21 and 141.55.243.192. Packet 24 is selected, and its details are shown in the lower pane. The details pane shows the frame structure: Ethernet II, Internet Protocol, and Transmission Control Protocol. The hex dump at the bottom shows the raw bytes of the packet.

No.	Time	Source	Destination	Protocol	Info
24	3.099835	141.55.193.21	141.55.243.192	TCP	http > 2022 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
27	0.001705	141.55.193.21	141.55.243.192	TCP	http > 2022 [ACK] Seq=1 Ack=336 win=6432 Len=0
28	0.003401	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
31	0.022862	141.55.193.21	141.55.243.192	TCP	http > 2023 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
34	0.000734	141.55.193.21	141.55.243.192	TCP	http > 2023 [ACK] Seq=1 Ack=548 win=6564 Len=0
35	0.000730	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
36	0.001704	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
39	0.037200	141.55.192.70	141.55.243.192	TCP	http > 2024 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
42	0.000487	141.55.192.70	141.55.243.192	TCP	http > 2024 [ACK] Seq=1 Ack=442 win=6432 Len=0
45	0.013131	141.55.192.70	141.55.243.192	TCP	http > 2025 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
48	0.000486	141.55.192.70	141.55.243.192	TCP	http > 2025 [ACK] Seq=1 Ack=381 win=6432 Len=0
49	0.002683	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
52	0.006323	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
53	0.000232	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
54	0.064944	141.55.192.70	141.55.243.192	HTTP	HTTP/1.1 301 Moved Permanently (text/html)
55	0.000024	141.55.192.70	141.55.243.192	HTTP	Continuation or non-HTTP traffic
58	0.000936	141.55.192.70	141.55.243.192	TCP	http > 2025 [ACK] Seq=747 Ack=813 win=7504 Len=0
61	0.141544	141.55.192.70	141.55.243.192	HTTP	HTTP/1.1 200 OK[Unreassembled Packet]
62	0.000027	141.55.192.70	141.55.243.192	HTTP	Continuation or non-HTTP traffic
63	0.000005	141.55.192.70	141.55.243.192	HTTP	Continuation or non-HTTP traffic
66	0.015763	141.55.192.70	141.55.243.192	TCP	http > 2025 [ACK] Seq=2343 Ack=1166 win=8576 Len=0

Frame 24 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:e0:52:d8:9b:00, Dst: 00:30:05:75:42:85
Internet Protocol, Src Addr: 141.55.193.21 (141.55.193.21), Dst Addr: 141.55.243.192 (141.55.243.192)
Transmission Control Protocol, Src Port: http (80), Dst Port: 2022 (2022), Seq: 0, Ack: 1, Len: 0

```
0000  00 30 05 75 42 85 00 e0 52 d8 9b 00 08 00 45 00  .0.UB... R....E.
0010  00 30 00 00 40 00 3e 06 6d 83 8d 37 c1 15 8d 37  .0..@.>. m..7...7
0020  f3 c0 00 50 07 e6 2b 73 c8 42 03 e4 1b 84 70 12  ...P...+s .B....p.
0030  16 d0 81 a6 00 00 02 04 05 b4 01 01 04 02      ..... .....
```

4.3 Paketdarstellung

4.3.1 Die Paketliste

No.	Time	Source	Destination	Protocol	Info
24	3.099835	141.55.193.21	141.55.243.192	TCP	http > 2022 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
27	0.001705	141.55.193.21	141.55.243.192	TCP	http > 2022 [ACK] Seq=1 Ack=336 win=6432 Len=0
28	0.003401	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
31	0.022862	141.55.193.21	141.55.243.192	TCP	http > 2023 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
34	0.000734	141.55.193.21	141.55.243.192	TCP	http > 2023 [ACK] Seq=1 Ack=548 win=6564 Len=0
35	0.000730	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
36	0.001704	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
39	0.037200	141.55.192.70	141.55.243.192	TCP	http > 2024 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
42	0.000487	141.55.192.70	141.55.243.192	TCP	http > 2024 [ACK] Seq=1 Ack=442 win=6432 Len=0
45	0.013131	141.55.192.70	141.55.243.192	TCP	http > 2025 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0
48	0.000486	141.55.192.70	141.55.243.192	TCP	http > 2025 [ACK] Seq=1 Ack=381 win=6432 Len=0
49	0.002683	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
52	0.006323	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
53	0.000232	141.55.193.21	141.55.243.192	HTTP	HTTP/1.1 304 Not Modified
54	0.064944	141.55.192.70	141.55.243.192	HTTP	HTTP/1.1 301 Moved Permanently (text/html)
55	0.000024	141.55.192.70	141.55.243.192	HTTP	Continuation or non-HTTP traffic
58	0.000936	141.55.192.70	141.55.243.192	TCP	http > 2025 [ACK] Seq=747 Ack=813 win=7504 Len=0
61	0.141544	141.55.192.70	141.55.243.192	HTTP	HTTP/1.1 200 OK[Unresembled Packet]
62	0.000027	141.55.192.70	141.55.243.192	HTTP	Continuation or non-HTTP traffic
63	0.000005	141.55.192.70	141.55.243.192	HTTP	Continuation or non-HTTP traffic
66	0.015763	141.55.192.70	141.55.243.192	TCP	http > 2025 [ACK] Seq=2343 Ack=1166 win=8576 Len=0

No.	Fortlaufende Paketnummer
Time	Ankunftszeit eines Paketes z.B. in Sekunden
Source	Absenderadresse (Schicht-2- oder Schicht-3-Adresse).
Destination	Zieladresse (Schicht-2- oder Schicht-3-Adresse).
Protocol	Protokollname z.B. ARP, UDP, HTTP usw. Hier wird das oberste Protokoll der OSI-Hierarchie angezeigt was in dem Paket enthalten ist. Sendet ein Webbrowser z.B. ein HTTP-Request ist dieser wie folgt gekapselt (Ethernet II (IP (TCP (HTTP-Request)))). Im Protokollfeld wird HTTP angezeigt, die anderen Protokollinhalte kann man in der Paketdetailliste sehen.
Info	Information zum Hauptinhalt des Paketes


Jede Zeile in der Liste entspricht einem Paket im Capturefile. Bei der Markierung einer Zeile werden Details zum Paket in der Paketdetailliste und Paketinhaltsliste angezeigt.

4.3.2 Paketdetailliste

```

.....
|> Frame 24 (62 bytes on wire, 62 bytes captured)
|> Ethernet II, Src: 00:e0:52:d8:9b:00, Dst: 00:30:05:75:42:85
|> Internet Protocol, Src Addr: 141.55.193.21 (141.55.193.21), Dst Addr: 141.55.243.192 (141.55.243.192)
|> Transmission Control Protocol, Src Port: http (80), Dst Port: 2022 (2022), Seq: 0, Ack: 1, Len: 0
.....

```

Hier werden Details des ausgewählten Paketes, wie Protokolle und Protokollfelder ausführlicher dargestellt. Der Button  zeigt an, dass weitere Details vorhanden sind und bei Bedarf können durch Anklicken des Buttons eingeblendet werden.

4.3.3 Paketinhaltsliste

```

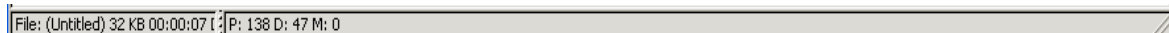
0000  00 30 05 75 42 85 00 e0 52 d8 9b 00 08 00 45 00  .0.uB... R.....E.
0010  00 30 00 00 40 00 3e 06 6d 83 8d 37 c1 15 8d 37  .0..@.>. m..7...7
0020  f3 c0 00 50 07 e6 2b 73 c8 42 03 e4 1b 84 70 12  ...P..+s .B...p.
0030  16 d0 81 a6 00 00 02 04 05 b4 01 01 04 02      .....

```

File: (Untitled) 32 KB 00:00:07 [P: 138 D: 47 M: 0]

Hier werden die Daten des in der Paketliste gewählten Paketes in hexadezimaler Form und als entsprechende ASCII-Zeichen dargestellt.

4.4 Statusleiste



Nach einem Captureprozess werden in der Statusleiste Informationen über das Capturefile: Filename (falls definiert), Filegröße usw. angezeigt.

Weiterhin werden Informationen zu den Paketen geliefert:

- **P** Anzahl der aufgenommenen Pakete.
- **D** Anzahl im Moment gezeigten Pakete.
- **M** Anzahl durch den Nutzer markierter Pakete.

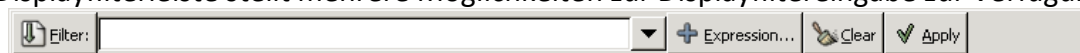
4.5 Displayfilter

4.5.1 Einführung

Displayfilter werden zur Auswahl bestimmter Pakete verwendet. Beispielsweise kann man sich im einfachsten Fall alle TCP-Pakete oder IP-Pakete oder HTTP-Pakete anzeigen lassen. Darüber hinaus gibt es sehr komplexe Selektionsmechanismen.

4.5.2 Displayfilterleiste

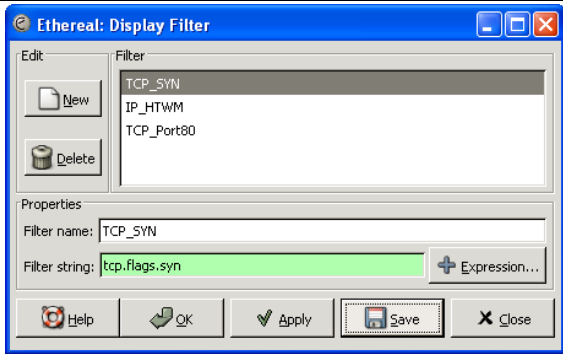

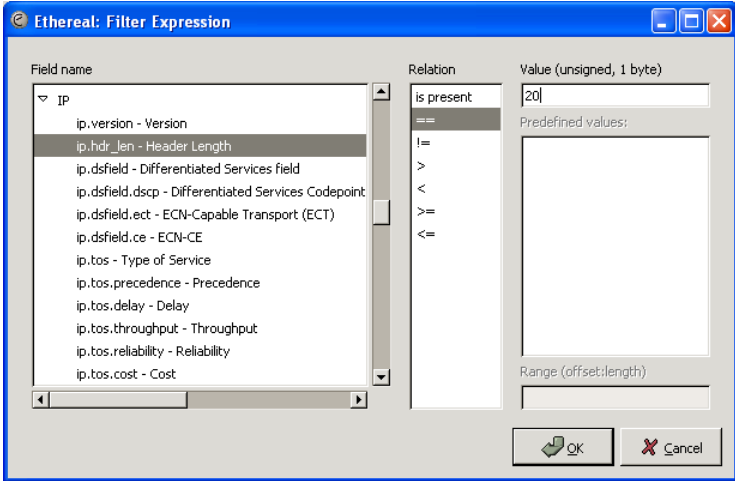
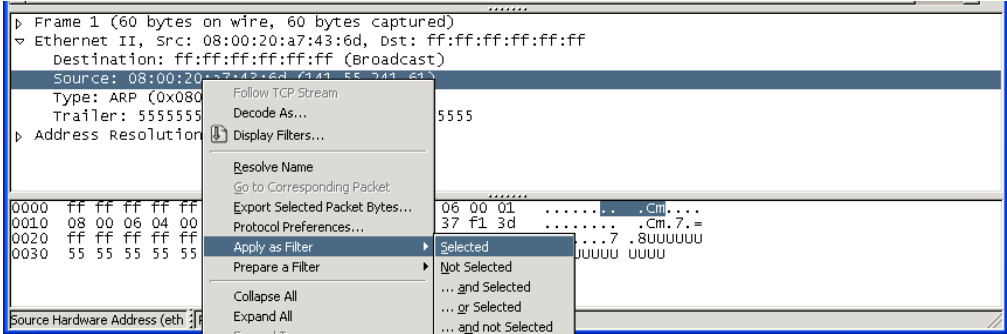
Die Displayfilterleiste stellt mehrere Möglichkeiten zur Displayfiltereingabe zur Verfügung.



In der nachfolgenden Tabelle wird ein kurzer Überblick zu den einzelnen Buttons und der dahinter liegenden Funktionalität gegeben. Nachfolgend wird dann etwas detaillierter auf die Möglichkeiten zur Displayfiltereingabe eingegangen.

Element	Funktion
	Öffnet Dialog zum Editieren, Laden, Speichern, Anwenden eines Displayfilters
	Möglichkeit zur direkten manuellen Eingabe eines Displayfilters
	Öffnet Assistenten zur Konstruktion eines Displayfilters. Dies ist sehr hilfreich zur Konstruktion komplexerer Ausdrücke
	Löschen des aktuellen Displayfilters
	Anwenden des eingestellten Displayfilters

4.5.2.1 Displayfiltereingabe

<p>Es gibt vier Möglichkeiten, Displayfilter zu setzen.</p>	
<p>1. Über den Button: <u>F</u>ilter:</p>	
<p>Dialog zum Editieren, Laden, Speichern, Anwenden eines Displayfilters</p>	
<p>2. Direkteingabe in das Eingabefeld</p>	
<p>Manuelle Eingabe eines Displayfilters²</p>	
<p>3. Über den Button +Expression</p>	
<p>Assistent zur Konstruktion eines Displayfilters. Sehr hilfreich zur Konstruktion komplexerer Ausdrücke (s. 4.5.4)</p>	
<p>4. Über die rechte Mousetaste in der Paketdetailliste</p>	
<p>Einsetzen der so gewählten Zeile als Displayfilter. Konstruktion komplexerer Ausdrücke mittels Prepare a Filter möglich.</p>	

Es ist zu beachten, dass sich die Syntax für die Displayfilter von denen der Capturfilter unterscheidet.

² Die Hintergrundfarbe grün → alles OK, rot → fehlerhafter Eintrag

4.5.3 Displayfiltersyntax

4.5.3.1 Einführung

Displayfilter haben folgenden allgemeinen Aufbau:

`<protocol>.<element>.<subelement><operator><wert>`

Elemente eines Protokolls sind mittels Punkt-Operator ansprechbar (z. B. `ip.version` → IP-Header-Element „Version“) (s. Displayfilteraufbau).

4.5.3.2 Vergleichsoperatoren

Protokollelemente können mittels folgender Operatoren oder ihrer “C”-ähnlichen Analoge verglichen werden:

Vergleichsoperator	Analog	Bedeutung (Englisch)	Bedeutung (Deutsch)
<code>eq</code>	<code>==</code>	Equal	gleich
<code>not</code>	<code>!</code>	Not	
<code>ne</code>	<code>!=</code>	Not Equal	nicht gleich
<code>gt</code>	<code>></code>	Greater Than	größer als
<code>lt</code>	<code><</code>	Less Than	kleiner als
<code>ge</code>	<code>>=</code>	Greater then or Equal to	größer gleich
<code>le</code>	<code><=</code>	Less than or Equal to	kleiner gleich

Eine Liste der unterstützten Protokolle mit Protokollfeldern, die als Displayfilter einsetzbar sind, findet man in **Help/Supported Protocols** oder unter <http://www.ethereal.com/docs/dfref/>.

4.5.3.3 Protokollelementtypen

Jedes Protokollelement ist von einem bestimmten Typ. Das sind:

- Ganzzahlig, vorzeichenlos (8-bit, 16-bit, 24-bit, 32-bit)
- Ganzzahlig, mit Vorzeichen (8-bit, 16-bit, 24-bit, 32-bit)
- Boolean (True, False)
- Ethernet-Adresse (6 byte)
- Bytefeld (byte array)
- IPv4-Adresse
- IPv6-Adresse
- IPX-Netzwerkdaten (network number)
- Text string
- Floating-Point-Zahlen mit doppelter Genauigkeit (Double-precision floating point number)

Integer-Zahlen sind in dezimaler, oktaler³ (z.B.: $(60)_8=074$) oder hexadezimaler Form (z.B.: $(60)_{16}=0x3C$) eingebbar.

Alle Angaben (Name und Typ) zu den einzelnen Elemente des jeweiligen Protokolls finden Sie in **Help/Manual Pages/Ethereal Filter** unter dem Überschrift **FILTER PROTOCOL REFERENCE**.

³ Die Angabe von Zahlen im oktalen Format ist auch bei Capturefilter möglich.

4.5.3.4 Logische Verknüpfungen

Einfache Ausdrücke sind mittels logischer Verknüpfungen **and** od. **&&**, **or** od. **||** und **not** od. **!** kombinierbar.

4.5.3.5 Substring-Operator

Mittels Substring-Operator lässt sich eine bestimmte Bytegruppe innerhalb eines Protokollelementes ansprechen.

Die gewünschte Bytegruppe wird nach dem Protokollfeldlabel in den eckigen Klammern mit Start-Offset und Länge bzw. End-Offset eingegeben. Es existieren folgende Varianten:

Befehl	Semantik
<code>fieldlabel[i:j]</code>	i = Start-Offset, j = Länge
<code>fieldlabel[i-j]</code>	i = Start-Offset, j = End-Offset, einschließlich
<code>fieldlabel[i:]</code>	i = Start-Offset, Länge = 1
<code>fieldlabel[:j]</code>	Start-Offset = 0, Länge = j
<code>fieldlabel[i]</code>	Start-Offset = i, End-Offset = Feldende
Auch möglich:	
<code>fieldlabel[i:j, k-1, m, :n, q:]</code>	

4.5.3.6 Beispiele

<code>ip.addr==10.0.0.5</code>	Alle Pakete mit der IP-Adresse 10.0.0.5 .
<code>ip.addr!=10.0.0.5</code>	Alle Pakete, für die IP-Adresse 10.0.0.5 als Zieladresse oder Quelladresse gilt.
<code>frame.pkt_len > 10</code>	Alle Pakete, dessen Länge größer als 10 Byte ist
<code>frame.pkt_len < 128</code>	Alle Pakete, dessen Länge kleiner als 128 Byte ist
<code>frame.pkt_len ge 0x100</code>	Alle Pakete, dessen Länge größer gleich 100 Hex ist
<code>frame.pkt_len <= 0x20</code>	Alle Pakete, dessen Länge kleiner gleich 20 Hex ist
<code>ip.addr==10.0.0.5 and tcp.flags.fin</code>	Alle Pakete, für die die IP-Adresse 10.0.0.5 Ziel- oder Quelladresse ist und bei denen FIN-Flag gesetzt ist
<code>ip.addr==10.0.0.5 or ip.addr==192.1.1.1</code>	Alle Pakete, für die die IP-Adresse 10.0.0.5 Ziel- oder Quelladresse ist oder die Pakete, für die die IP-Adresse 192.1.1.1 Ziel- oder Quelladresse ist
<code>not llc</code>	Alle Pakete außer LLC-Pakete
<code>eth.src[0:3] == 00:00:83</code>	Alle Pakete mit den Ethernet-Rahmen, dessen Feld Zieladresse die Bytefolge 00 00 83, die 3 Byte groß ist und beginnt ab der 0.Byteposition, enthält
<code>eth.src[1-2] == 00:83</code>	Alle Pakete mit den Ethernet-Rahmen, dessen Feld Zieladresse die Bytefolge 00 83 von der 1. bis einschließlich 2.Byteposition enthält
<code>eth.src[:4] == 00:00:83:00</code>	Alle Pakete mit den Ethernet-Rahmen, dessen Feld Zieladresse in der ersten vier (0.-3.) Bitpositionen die Bitfolge 00 00 83 00 enthält.
<code>eth.src[4:] == 20:20</code>	Alle Pakete mit den Ethernet-Rahmen, deren Inhalt 20 20 ab der 4.Bitposition beginnend bis zum Rahmenende ist.
<code>eth.src[2] == 83</code>	Alle Pakete, deren Ethernetrahmen im Feld Zieladresse in der 2. Bitposition 83 enthält.

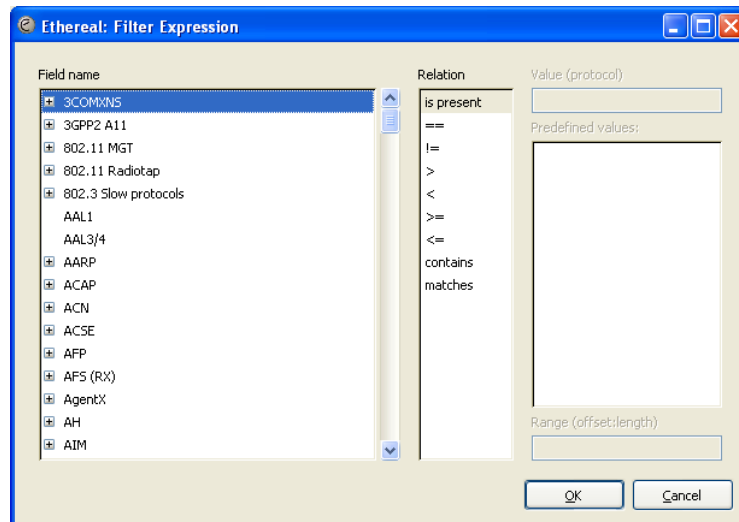
Beachte: Alle Pakete für die andere IP-Adressen außer 10.0.0.5 erhält man mit dem Ausdruck `!ip.addr==10.0.0.5` .

4.5.4 Filter Expression - Assistent

Wenn man sehr lange **Ethereal** nutzt, wird man die gebräuchlichsten Filterstrings im Kopf haben und diese direkt in das Eingabefeld eintragen.

Am Anfang weiß man aber oft nicht besonders viel über die Protokolle und die dort enthaltenen Elemente. Für den Ethereal-Anfänger ist deshalb die Verwendung des **Filter Expression - Assistent** besonders sinnvoll.

Dieser öffnet sich, wie bereits beschrieben, beim Klicken auf den Button **+Expression** in der Filterleiste:



Field Name	Hier kann man das Protokoll bzw. ein Protokollfeld auswählen, falls das Protokoll mehrere Felder hat.
Relation	Hier kann man die Relation auswählen.
Value	Nach der Auswahl eines Protokolls/Protokollfeldes und einer Relation kann man einen Wert eingeben. Der Wert muss dem Typ des Protokollfeldes entsprechen.
Predefined values	Bei einigen Protokollfeldern stehen vordefinierte Werte zur Verfügung.
Range (offset: length)	Hier kann man für die Werte einiger Protokollfelder Wertebereiche angeben.

Tipp: Sehr oft verwendete Filter können gespeichert werden.

4.5.5 Farbige Darstellung einzelner Pakete

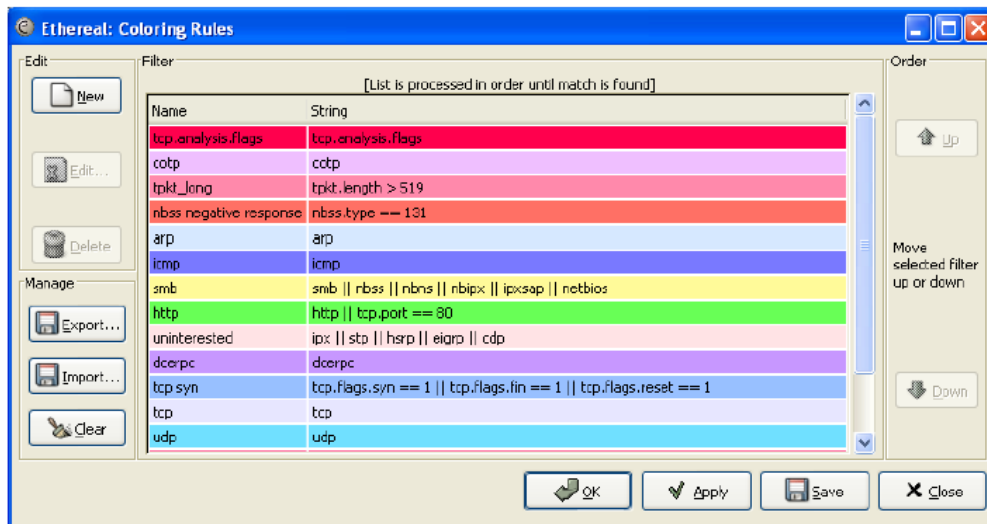
Bei sehr großer Anzahl aufgenommener Pakete kann es sehr nützlich sein, bestimmte Pakete farblich hervorzuheben. Auswahlkriterien können sein:

- **einzelne Protokolle oder ihre Kombination mittels logischer Verknüpfungen,**
- **einzelne Protokollfelder mit bestimmten Eigenschaften oder ihre Kombination mittels logischer Verknüpfungen.**

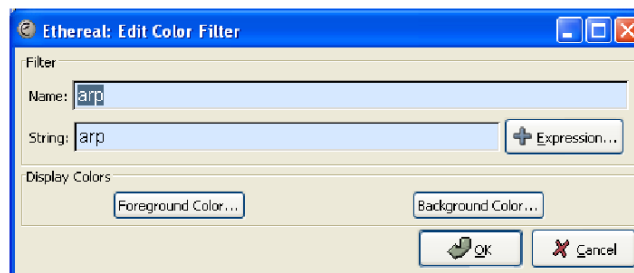
Alle eingegebene Werte werden in dem **Coloring Rules**-Dialogfenster, das man in der Menüleiste in **View/Coloring Rules**, oder in der Steuerleiste unter **Edit coloring Rules...**, oder

beim Anklicken eines Paketes in der Paketliste mit der rechten Maustaste, angezeigt. Bei der Eintragung der „Coloring rules“ muss man die Reihenfolge der Einträge beachten, wenn man sinnvolle Ergebnisse erreichen will.

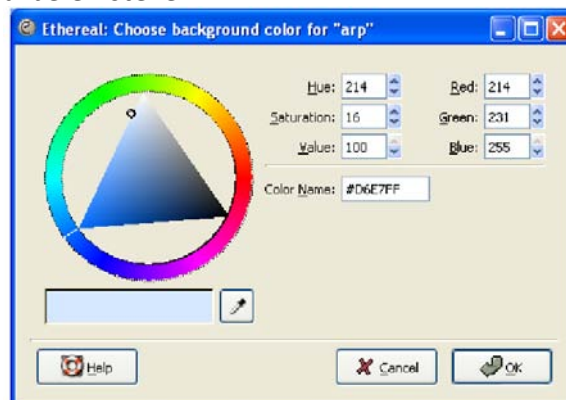
Die Farbfestlegungen für die einzelnen Protokolle sollten mit den Up-Down-Buttons in der Reihenfolge angeordnet sein, die der Ordnung der Protokolle im Kommunikationsstack entsprechen (von oben nach unten). Will man alle ARP-Pakete und alle Ethernet-Pakete farblich unterscheiden, muss der Farbeintrag für ARP über dem für Ethernet stehen. Steht der Eintrag für Ethernet vor dem Eintrag für ARP, nehmen alle Pakete die Farbe für Ethernet an.



Alle Eingaben werden im **Edit Color Filter-Assistent** gemacht, der nach dem Anklicken des **New**-Buttons erscheint. Die Syntax für das Feld **String** ist wie die für Displayfilter.



Mittels **Foreground Color**-Button kann man die Schriftfarbe und mit dem **Background Color**-Button die Hintergrundfarbe einstellen.



Anlage: Spezielle Capturefilter

Diese speziellen Capturefilter werden z.B. auf IP, TCP, UDP, ICMP usw. angewendet. Die Protokollheader werden dabei als Array, beginnend mit dem Index [0], aufgefasst. Die Elemente des Arrays werden wie folgt adressiert:

```
<protocol>[<offset0>:<length0>]&<mask> //mask ist optional
<protocol>[<offset1=offset0+length0>:<length1>]&<mask>
<protocol>[<offset2=offset1+length1>:<length2>]&<mask>.
...
<protocol>[<offsetn=offsetn-1+lengthn-1>:<lengthn>]&<mask>.
```

Beispiel:

```
ip[0:1]&0xf0 →Version, ip[0:1]&0x0f →IP-Header-Length
ip[1:1]→Type of Service
```

Dabei ist zu beachten, dass die Länge eines Elementes mindestens ein Byte beträgt. Steht die Information aber z.B. nur in einem Halbbyte, wird dies durch eine zusätzliche Maskierung angezeigt.

IP-bezogene Filter

Der Aufbau eines IP-Headers der Version 4 ([RFC 791](#)) und die Indizierung der einzelnen Elemente werden nachfolgend dargestellt.

0	8	16			32
Version	IHL	Type of service		Total length	
Identification			Flags	Fragment offset	
Time to live		Protocol		Header checksum	
Source address					
Destination address					
Options				Padding	
Data					

Elemente-Indizierung des Arrays ip:

Element	Adresse
Version ,Internet-Header-Länge	ip [0:1]&0xf0, ip [0:1]&0x0f
Type of service	ip [1:1]
Packet-Länge	ip [2:2]
Identification	ip [4:2]
Flags, Fragment offset	ip [6:2]&0xE000, ip [6:2]&0x1FFF
Time to live	ip [8:1]
Protocol	ip [9:1]
Header checksum	ip [10:2]
Source address	ip [12:4]
Destination address	ip [16:4]
Options	ip [20:4]

IP-ToS-Feld (Type of Service)

Mit diesem Feld kann man Qualitätsparameter für den Transport von IP-Paketen einstellen. Dafür werden 6 Bit verwendet. Diese haben folgende Bedeutung:

Bits 6-7	Bit 5: Sicherheit	Bit 4: Durchsatz	Bit 3: Verzögerung	Bits 0-2: Priorität			
Reserviert für die Zukunft	0	Normal	0	Normal	111	Network control.	
	1	Hoch	1	Hoch	110	Internetwork control.	
				1	Niedrig	101	CRITIC/ECP.
						100	Flash override.
						011	Flash.
						010	Immediate.
						001	Priority.
						000	Routine.

IP-Protokoll-Feld

IP-Protokoll-Feld zeigt den Nutzer des IP-Paketes an, z.B. TCP, UDP, ICMP. In der nachfolgenden Tabelle sind wichtige Werte, die dieses Feld annehmen kann, aufgelistet.

Decimal	Keyword	Protocol
1	ICMP	Internet Control Message
2	IGMP	Internet Group Management
6	TCP	Transmission Control
17	UDP	User Datagram
41	SIP	Simple Internet Protocol
55-60		Unassigned
61		any host internal protocol
63		any local network
68		any distributed file system
99		any private encryption scheme
114		any 0-hop protocol
138-252		Unassigned
253		Use for experimentation and testing
254		Use for experimentation and testing
255		Reserved

Weitere Protokolle finden Sie unter: <http://www.iana.org/assignments/protocol-numbers>

TCP-bezogene Filter

Nachfolgend werden TCP-Headers ([RFC 793](#)) und die Indizierung der Elemente gezeigt.

0	8	16	32
Source port		Destination port	
Sequence number			
Acknowledgment number			
Data Offset	Reserved	U R G	A P K
		P R H	S S T
		S Y N	F I N
Checksum		Window	
Options		Urgent pointer	
Options		Padding	
Data			

Elemente-Indizierung des Arrays tcp:

Element	Adresse
Source port	<code>tcp[0 : 2]</code>
Destination port	<code>tcp[2 : 2]</code>
Sequence number	<code>tcp[4 : 4]</code>
Acknowledgment number	<code>tcp[8 : 4]</code>
Header-Länge	<code>tcp[12 : 1] & 0xf0</code>
Flags	<code>tcp[13 : 1]</code>
Window size	<code>tcp[14 : 2]</code>
Checksum	<code>tcp[16 : 2]</code>
Urgent pointer	<code>tcp[18 : 2]</code>
Options	<code>tcp[20 : 4]</code>

UDP-bezogene Filter

Der Aufbau eines UDP-Headers ([RFC 768](#)) und die Indizierung der einzelnen Elemente werden nachfolgend dargestellt.

0	16	32
Source port	Destination port	
Length	Checksum	
Data		

Elemente-Indizierung des Arrays udp:

Feld	Zugriff
Source port	<code>udp[0 : 2]</code>
Destination port	<code>udp[2 : 2]</code>
Header-Länge	<code>udp[4 : 2]</code>
Checksum	<code>udp[6 : 2]</code>

ICMP-bezogene Filter

Der Aufbau eines ICMP-Headers ([RFC 792](#)) und die Indizierung der einzelnen Elemente werden nachfolgend dargestellt

0	8	16	32
Type	Code	Cheksum	
Identifier		Sequance number	
Address mask			

Elemente-Indizierung des Arrays icmp:

Feld	Zugriff
Type	<code>icmp[0:1]</code>
Code	<code>icmp[1:1]</code>
Checksum	<code>icmp[2:2]</code>
Identifier	<code>icmp[4:2]</code>
Sequence number	<code>icmp[6:2]</code>
Address mask	<code>icmp[8:4]</code>

Type	Code	Description
0		Echo reply
3		Destination unreachable
3	0	Net unreachable
3	1	Host unreachable
3	2	Protocol unreachable
3	3	Port unreachable
3	4	Fragmentation needed and DF set
3	5	Source route failed
4		Source quench
5		Redirect
5	0	Redirect datagrams for the network
5	1	Redirect datagrams for the host
5	2	Redirect datagrams for the type of service and network
5	3	Redirect datagrams for the type of service and host.
8		Echo
11		Time exceeded
11	0	Time to live exceeded in transit
11	1	Fragment reassemble time exceeded
12		Parameter problem
13		Timestamp
14		Timestamp reply
15		Information request
16		Information reply

Details zu anderen Protokollen finden Sie unter: <http://protocols.com/>

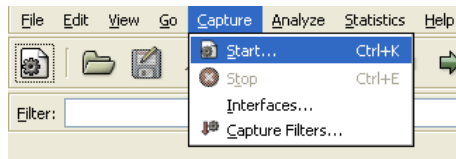
Ethereal: Quickstart

1. **Ethereal** starten
2. Capture-Prozess starten:

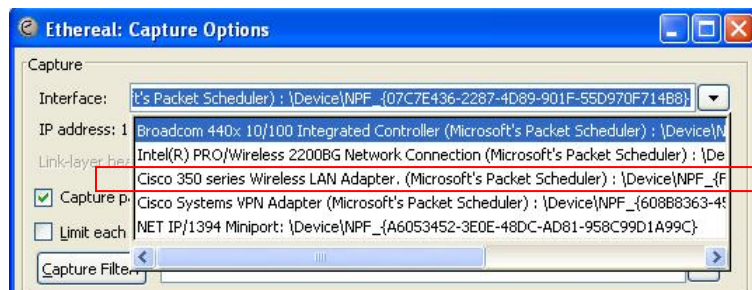
“Start a new live capture...”-Button drücken



oder **Capture/Start...** auswählen

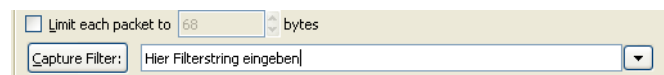


3. **Interface** auswählen (Netzwerkkarte), falls mehrere vorhanden sind.

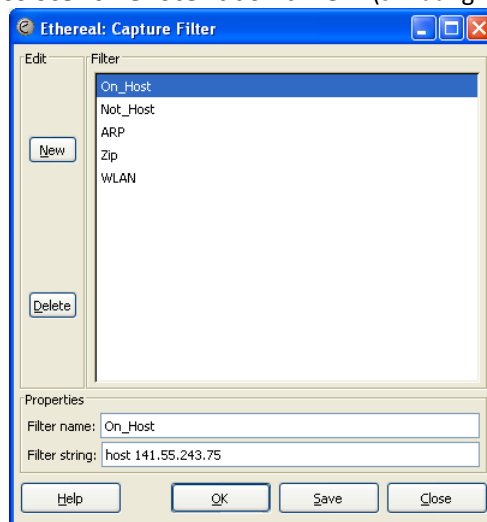


Wireless Interface

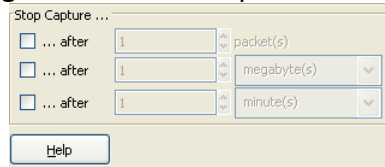
4. Filterstring eingeben, wenn nötig ist,



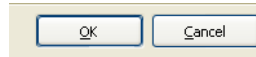
oder schon gespeicherte über “**Capture Filter**” - Button und dann im “**Capture Filter**” – Assistent-Fenster auswählen. (s. Häufig verwendete Capture-Filter)



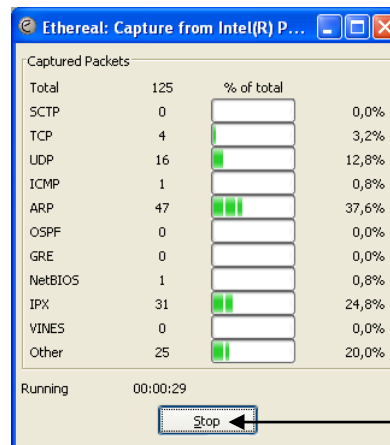
5. Optional Endebedingung einstellen oder später manueller Abbruch



6. Alle Einstellungen mit OK bestätigen und anwenden

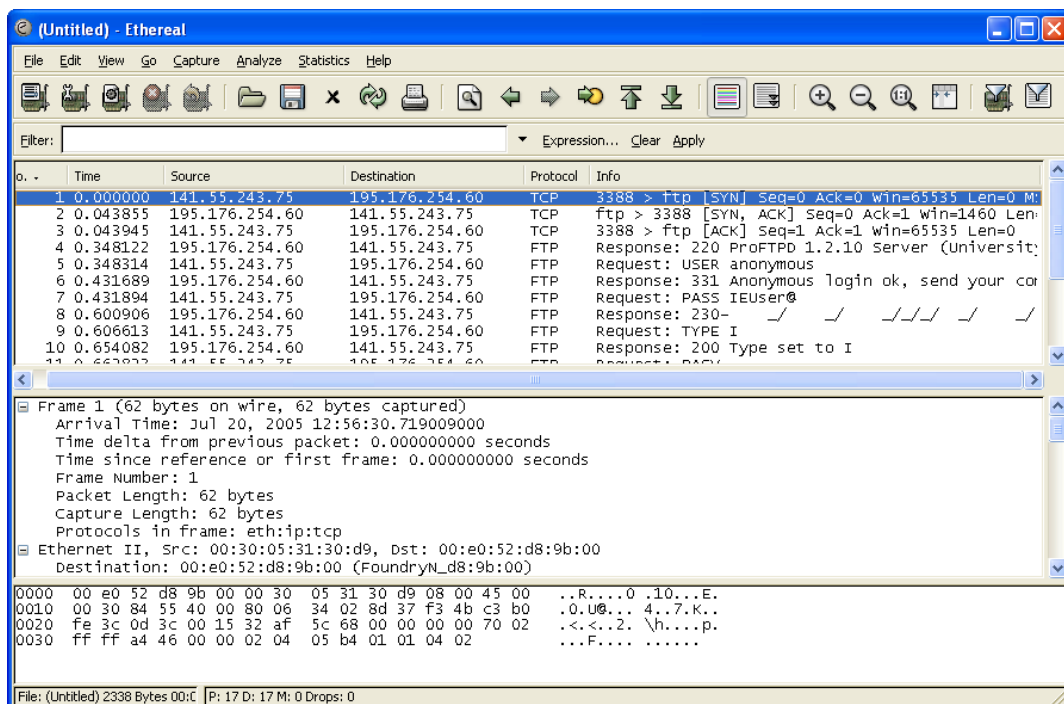


Jetzt läuft der Prozess...



Manueller Abbruch des Prozesses

Nach dem, das Prozess beendet oder abgebrochen wurde, kehrt das Programm automatisch zum Hauptfenster zurück. Es werden dabei die aufgenommenen Pakete angezeigt und man kann mit der Analyse anfangen



Häufig verwendete Capture-Filter

Ziel	Filter String
Alle HTTP-Pakete	<code>port 80</code>
Alle DNS-Pakete	<code>port 53</code>
Alle SMTP-Pakete	<code>port 25</code>
Alle FTP-Pakete	<code>port 21</code>
Alle TELNET-Pakete	<code>port 23</code>
Alle POP3-Pakete	<code>port 110</code>
Alle IP-Pakete	<code>ip</code>
Alle TCP-Pakete	<code>tcp</code>
Alle UDP-Pakete	<code>udp</code>
Alle ARP-Pakete	<code>arp</code>
Alle ICMP-Pakete	<code>icmp</code>
Alle Pakete mit ping request/-response	<code>icmp[0]= 0 or icmp[0]= 8</code>
Alle Pakete, die broadcast geschickt sind	<code>[ip ether] broadcast</code>
Alle Pakete, die multicast geschickt sind	<code>[ip ether] multicast</code>