



# ICX TECHNOTE - PORT MIRROR V1.0

Ruckus ICX configuratie – Port mirror

Versie: 1.0  
Auteur: Herwin de Rijke  
Datum: 7 april 2020



# Inhoud

1	Inleiding .....	2
<b>1.1</b>	<b>DOELSTELLING .....</b>	<b>2</b>
<b>1.2</b>	<b>BEOOGD PUBLIEK.....</b>	<b>2</b>
<b>1.3</b>	<b>VOORKENNIS/BENODIGDHEDEN .....</b>	<b>2</b>
2	Port mirroring.....	3
<b>2.1</b>	<b>STANDAARD WAARDEN .....</b>	<b>4</b>
<b>2.2</b>	<b>ENKELE POORT MONITOREN .....</b>	<b>5</b>
<b>2.3</b>	<b>MEERDERE POORTEN MONITOREN .....</b>	<b>5</b>
<b>2.4</b>	<b>MONITOREN INPUT EN OUTPUT VERKEER .....</b>	<b>6</b>
<b>2.5</b>	<b>MONITOREN VAN EEN LAG.....</b>	<b>6</b>
3	Wireshark .....	7
<b>3.1</b>	<b>FILTERS.....</b>	<b>7</b>
<b>3.2</b>	<b>APPLY AS FILTER FUNCTIE.....</b>	<b>9</b>
<b>3.3</b>	<b>CONVERSATION .....</b>	<b>9</b>
<b>3.4</b>	<b>COLORING RULES.....</b>	<b>10</b>
<b>3.5</b>	<b>WIRESHARK TOOLS .....</b>	<b>11</b>

# 1 Inleiding

In dit document wordt beschreven op welke manier port mirroring geconfigureerd kan worden op een ICX switch. Daarnaast worden er enkele voorbeelden van analyse mogelijkheden in het programma Wireshark getoond.

De instructies die in dit document gegeven worden zijn op basis van firmware versie Version 08.0.90. Wij raden aan om uw switch te upgraden naar deze versie of hoger. Mogelijk zijn in andere versies als gebruikte versies bepaalde functies niet beschikbaar of is de werking anders.

## 1.1 Doelstelling

De doelstelling van dit document is het bekend maken met de manier waarop port mirroring op een Ruckus ICX switch kan worden geconfigureerd.

## 1.2 Beoogd publiek

Dit document is geschreven voor technisch personeel die een Ruckus ICX switch willen configureren om gebruik te maken van port mirroring.

## 1.3 Voorkennis/Benodigdheden

Om optimaal te kunnen profiteren van wat er in dit document beschreven staat is het van belang dat u basiskennis heeft van de volgende onderwerpen:

- Basiskennis van IPv4
- Basiskennis Ruckus ICX Command Line
- Basiskennis Wireshark

## 2 Port mirroring

Een switch stuurt standaard verkeer dat aan een client geadresseerd is alleen naar de switch poort waarop het MAC adres van deze client geleerd is. Om dit verkeer te kunnen bekijken vanaf een andere poort moet dit apart worden geconfigureerd.

Op de switch poort waar een device op aangesloten is zul je standaard alleen unicast verkeer zien gericht aan het device en broadcast verkeer.

Om op deze poort ook het verkeer dat gericht is op andere poorten te kunnen zien kan op de meeste managed switches port mirroring worden ingesteld.

Bij de Ruckus ICX switch zijn hiervoor twee typen poorten gedefinieerd.

Monitor poorten zijn poorten waarop devices zijn aangesloten waarvan het verkeer gekopieerd wordt. Mirror poorten zijn de poorten waar het verkeer naartoe wordt gekopieerd. Op deze poort wordt het capture device, bijvoorbeeld een laptop met wireshark aangesloten.

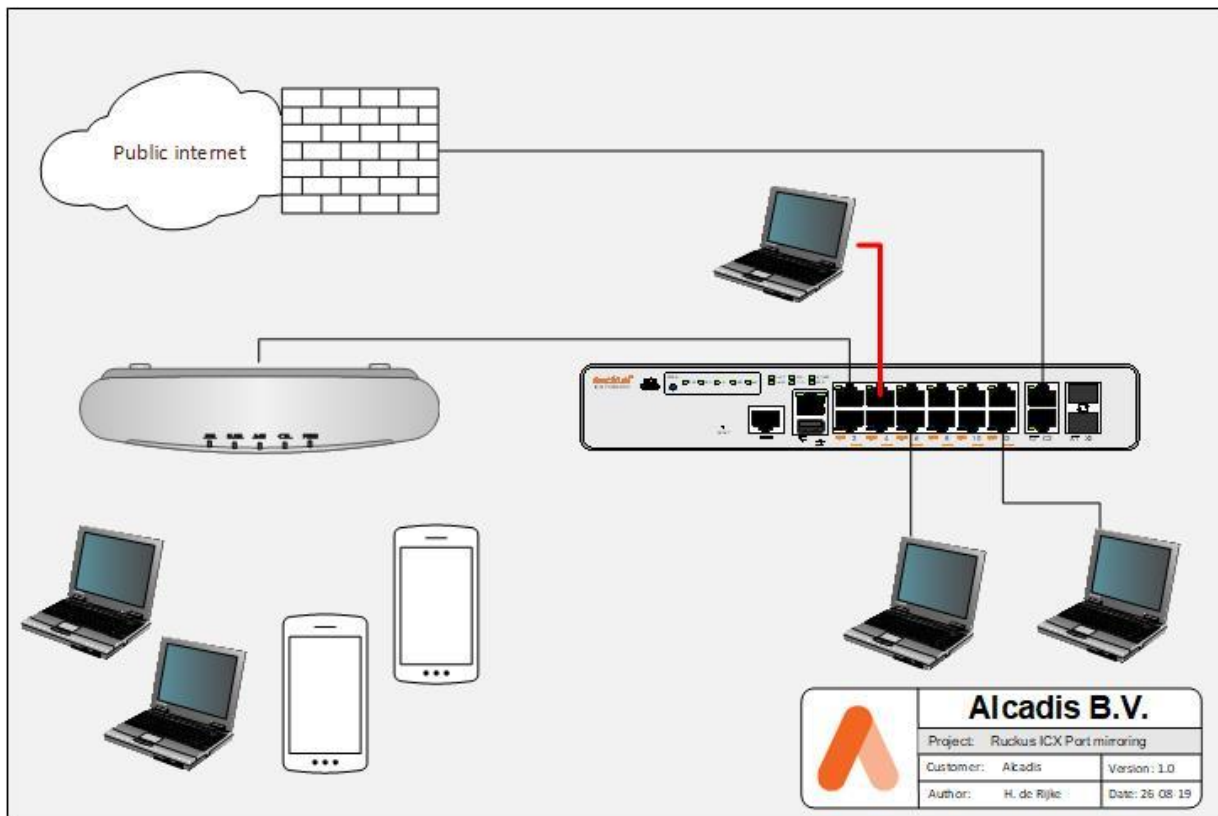
Daarnaast kan er onderscheid gemaakt worden tussen ingress en egress verkeer. Ingress verkeer is al het inkomende verkeer op een poort, egress is al het uitgaande verkeer van een poort. Er is voor elk type poort een beperking in het maximaal aantal beschikbare poorten per region.

Port type	Maximum supported
Ingress mirror port	1 per port region
Egress mirror port	1 per port region
Ingress monitoring port	No limit
Egress monitoring port	8 per port region

- Het is mogelijk meerdere monitoring poorten in te stellen in combinatie met 1 mirror poort
- Het is mogelijk om al het ingress verkeer te configureren op een andere mirror poort dan het egress verkeer
- Het is niet mogelijk meerdere mirror poorten in te stellen die zowel ingress als egress verkeer bevatten.

In onderstaand voorbeeld is aangegeven welke foutmelding wordt weergegeven als je probeert in dezelfde region meerdere poorten in te stellen voor ingress of egress verkeer.

```
ICX7150(config)#show mirror
Mirror port 1/1/11
  Input monitoring      : (U1/M1)  7  9
  Output monitoring    : (U1/M1)  7  9
ICX7150(config)#mirror ethernet 1/1/8
Error - Inbound mirror port 1/1/11 is already active on this region.
Error - Outbound mirror port 1/1/11 is already active on this region.
```



## 2.1 Standaard waarden

In een standaard configuratie staat port mirroring niet ingesteld. De poort waar de laptop met wireshark op zit aangesloten zit, gedraagt zich precies zoals elke andere poort. Mocht er toch een port mirror actief zijn en je wilt deze uitschakelen dan kan dit met onderstaand commando. Let op dat voor elke poort een apart commando moet worden ingevoerd.

```
ICX7150 (config) #show mirror
Mirror port 1/1/6
  Input monitoring      : None
  Output monitoring    : (U1/M2)  1
Mirror port 1/1/12
  Input monitoring      : (U1/M2)  1
  Output monitoring    : None
ICX7150 (config) #no mirror ethernet 1/1/12
ICX7150 (config) #no mirror ethernet 1/1/6
```

## 2.2 Enkele poort monitoren

In onderstaand voorbeeld is op poort 1/1/1 een laptop aangesloten en op poort 1/1/3 een accesspoint. Door al het verkeer dat van en naar de switchpoort waarop het accesspoint aangesloten is te kopiëren naar de switchpoort waarop een laptop aangesloten is; kan al het verkeer van de wireless clients dat door het accesspoint gebridget wordt, geanalyseerd worden.

```
ICX7150#conf t
ICX7150 (config)#mirror ethernet 1/1/3
ICX7150 (config)#show mirror
Mirror port 1/1/3
  Input monitoring      : None
  Output monitoring    : None
ICX7150 (config)#interface ethernet 1/1/1
ICX7150 (config-if-e1000-1/1/1)#monitor ?
  both                Both incoming and outgoing packets
  ethernet            Specify mirror port to use
  input               Incoming packets
  output              Outgoing packets
  profile              select monitor profile
ICX7150 (config-if-e1000-1/1/1)#monitor ethernet 1/1/3 both
ICX7150 (config-if-e1000-1/1/1)#exit
ICX7150 (config)#exit
ICX7150#show mirror
Mirror port 1/1/3
  Input monitoring      : (U1/M1)    1
  Output monitoring    : (U1/M1)    1
```

## 2.3 Meerdere poorten monitoren

In dit voorbeeld wordt al het input en output verkeer van poorten 1/1/6 en 1/1/12 naar poort 1/1/3 gekopieerd.

```
ICX7150#conf t
ICX7150 (config)#mirror ethernet 1/1/3
ICX7150 (config)#show mirror
Mirror port 1/1/3
  Input monitoring      : None
  Output monitoring    : None
ICX7150 (config)#interface ethernet 1/1/6
ICX7150 (config-if-e1000-1/1/1)#monitor ethernet 1/1/3 both
ICX7150 (config-if-e1000-1/1/6)#exit
ICX7150 (config)#interface ethernet 1/1/12
ICX7150 (config-if-e1000-1/1/12)#monitor ethernet 1/1/3 both
ICX7150 (config)#exit
ICX7150#show mirror
Mirror port 1/1/3
  Input monitoring      : (U1/M1)    6 12
  Output monitoring    : (U1/M1)    6 12
```

## 2.4 Monitoren input en output verkeer

In sommige situaties is het handig om alleen het verkeer dat naar een specifieke poort gestuurd wordt te onderzoeken of alleen het verkeer dat daarvan afkomstig is. Denk hierbij aan een uplink poort waarbij je bijvoorbeeld het DHCP verkeer richting server wilt onderzoeken of alleen het verkeer dat vanaf de server richting clients gaat. In onderstaand voorbeeld wordt het inkomende verkeer van uplink poort 1/2/1 naar een laptop met wireshark gestuurd die op poort 1/1/12 aangesloten is. Het uitgaande verkeer van de uplink poort wordt naar een andere laptop met wireshark op poort 1/1/6 gestuurd.

```
ICX7150#conf t
ICX7150 (config)#mirror ethernet 1/1/12 input
ICX7150 (config)#mirror ethernet 1/1/6 output
ICX7150 (config)#show mirror
Mirror port 1/1/6
  Input monitoring      : None
  Output monitoring    : None
Mirror port 1/1/12
  Input monitoring      : None
  Output monitoring    : None
ICX7150 (config)# interface ethernet 1/2/1
ICX7150 (config-if-e1000-1/2/1)#monitor ethernet 1/1/12 input
ICX7150 (config-if-e1000-1/2/1)#monitor ethernet 1/1/6 output
SSH@ALCLAB-SER-SW007 (config-if-e1000-1/2/1)#show mirror
Mirror port 1/1/6
  Input monitoring      : None
  Output monitoring    : (U1/M2)  1
Mirror port 1/1/12
  Input monitoring      : (U1/M2)  1
  Output monitoring    : None
```

## 2.5 Monitoren van een LAG

Het is ook mogelijk om het verkeer van een LAG te monitoren, bijvoorbeeld als je inzicht wilt krijgen in het verkeer tussen een MER en een SER. Hier moet uiteraard wel rekening worden gehouden met de capaciteit van de mirror en monitor poort. Als er meer verkeer over de LAG gaat dan de capaciteit van de mirror poort is dan zal de packet capture niet de juiste informatie weergeven.

```
ICX7150#conf t
ICX7150 (config)#mirror ethernet 1/1/1
ICX7150 (config)#show mirror
Mirror port 1/1/1
  Input monitoring      : None
  Output monitoring    : None
ICX7150 (config)#interface lag 1
ICX7150 (config-lag-if-lg1)#monitor ethernet 1/1/3 both
ICX7150 (config-lag-if-lg1)#exit
ICX7150 (config)#exit
ICX7150#show mirror
Mirror port 1/1/1
  Input monitoring      : (U1/M3)  1  2
  Input monitoring      : (LAG)      1
  Output monitoring     : (U1/M3)  1  2
  Output monitoring     : (LAG)      1
ICX7150#
```

## 3 Wireshark

Het doel van een mirror is het kopiëren van het verkeer dat over een verbinding gaat. Om dit verkeer vervolgens te analyseren maak je gebruik van het programma Wireshark. In dit hoofdstuk geven we een aantal voorbeelden van filters die gebruikt kunnen worden de verzamelde pakketten te filteren. Op deze manier kan op eenvoudige wijze verkeer van specifieke clients of van een specifiek type worden onderzocht. Daarnaast geven we nog een aantal voorbeelden van verdere gereedschappen die het programma biedt om de pakketten te onderzoeken

### 3.1 Filters

Als je al het verkeer afkomstig van en gericht aan een specifiek MAC adres wilt onderzoeken kun je dat doen met onderstaand filter:

```
filter
eth.addr == 24:79:2a24:0e:d0
```

49886	638.253378	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
49887	638.254248	172.18.0.17	172.17.216.174	SSHv2	130 Server: Encrypted packet (len=64)
49888	638.254249	172.17.216.174	172.18.0.17	TCP	66 54285 -> 22 [ACK] Seq=2875 Ack=3686 Win=16128 Len=0 TSval=4294741188 TSecr=341658443
49891	638.559783	172.17.216.174	172.18.0.17	SSHv2	178 Client: Encrypted packet (len=112)
49892	638.561755	172.18.0.17	172.17.216.174	SSHv2	146 Server: Encrypted packet (len=80)
49893	638.562660	172.17.216.174	172.18.0.17	TCP	66 54285 -> 22 [ACK] Seq=2187 Ack=3686 Win=16128 Len=0 TSval=4294741416 TSecr=341658751
49894	638.698437	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)

Om verkeer afkomstig van en gericht aan een specifiek IP adres te onderzoeken gebruik je onderstaand filter:

```
filter:
ip.addr == 172.17.216.174
```

95625	1715.624974	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
95626	1715.626853	172.18.0.17	172.17.216.174	SSHv2	114 Server: Encrypted packet (len=48)
95627	1715.627874	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
95628	1715.628871	172.18.0.17	172.17.216.174	SSHv2	130 Server: Encrypted packet (len=64)
95629	1715.629868	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
95630	1715.630427	172.18.0.17	172.17.216.174	SSHv2	162 Server: Encrypted packet (len=96)
95631	1715.631358	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)

Om verkeer van een bepaald type te onderzoeken (bijvoorbeeld DHCP, ARP, HTTP of alleen verkeer van TCP poort 22) dan kan dit door onderstaande filters toe te passen:

```
filter:
dhcp
```

48935	616.481838	0.0.0.0	255.255.255.255	DHCP	590 DHCP Discover - Transaction ID 8x13b5ee5b
48936	616.482941	0.0.0.0	255.255.255.255	DHCP	590 DHCP Discover - Transaction ID 8x13b5ee5b
48937	616.484685	172.17.216.1	172.17.216.174	DHCP	341 DHCP Offer - Transaction ID 8x13b5ee5b
48938	616.485883	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 8x13b5ee5b
48939	616.486885	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 8x13b5ee5b
48940	616.411391	172.17.216.1	172.17.216.174	DHCP	341 DHCP ACK - Transaction ID 8x13b5ee5b
1122.	3227.645415	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 8x1e30f7d1
1122.	3227.645416	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 8x1e30f7d1

```
filter:
arp
```

1118.	3164.608122	RuckusWi_2b:72:f0	Broadcast	ARP	60 Who has 172.17.216.133? Tell 172.17.216.1
1118.	3165.692269	RuckusWi_2b:72:f0	Broadcast	ARP	60 Who has 172.17.216.183? Tell 172.17.216.1
1118.	3165.692271	RuckusWi_2b:72:f0	Broadcast	ARP	60 Who has 172.17.216.183? Tell 172.17.216.1
1118.	3165.692440	HenlettP_3f:b5:79	RuckusWi_2b:72:f0	ARP	42 172.17.216.183 is at e4:e7:49:3f:b5:79
1118.	3165.692497	HenlettP_3f:b5:79	RuckusWi_2b:72:f0	ARP	42 172.17.216.183 is at e4:e7:49:3f:b5:79
1118.	3166.503178	RuckusWi_2b:72:f0	Broadcast	ARP	60 Who has 172.17.216.174? Tell 172.17.216.1
1118.	3166.503179	RuckusWi_2b:72:f0	Broadcast	ARP	60 Who has 172.17.216.174? Tell 172.17.216.1
1118.	3166.504358	RuckusWi_24:0e:d0	RuckusWi_2b:72:f0	ARP	60 172.17.216.174 is at 24:79:2a:24:0e:d0
1118.	3170.442288	RuckusWi_24:0e:d0	RuckusWi_2b:72:f0	ARP	60 Who has 172.17.216.1? Tell 172.17.216.174
1118.	3170.443339	RuckusWi_2b:72:f0	RuckusWi_24:0e:d0	ARP	60 172.17.216.1 is at 90:3a:72:2b:72:f0



filter:  
http

1371L. 5236.329340	13.107.4.50	172.17.216.183	HTTP	305 HTTP/1.1 403 Forbidden
1371L. 5236.364266	2a01:1111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	327 HTTP/1.1 403 Forbidden
1371L. 5236.370249	13.107.4.50	172.17.216.183	HTTP	305 HTTP/1.1 403 Forbidden
1371L. 5236.373556	2a01:1111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	325 HTTP/1.1 403 Forbidden
1371L. 5236.375997	13.107.4.50	172.17.216.183	HTTP	305 HTTP/1.1 403 Forbidden
1376L. 5241.064064	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2a01:1111:2003::50	HTTP	501 GET /filestreamingservice/files/fc989f39-6b38-4ee7-b13a-6f6b68435329?P1=15825663918P2=4028
1376L. 5241.066116	172.17.216.183	13.107.4.50	HTTP	481 GET /filestreamingservice/files/fc989f39-6b38-4ee7-b13a-6f6b68435329?P1=15825663918P2=4028
1376L. 5241.072872	2a01:1111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	1088 HTTP/1.1 206 Partial Content
1376L. 5241.073970	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2a01:1111:2003::50	HTTP	513 GET /filestreamingservice/files/fc989f39-6b38-4ee7-b13a-6f6b68435329?P1=15825663918P2=4028
1376L. 5241.075257	13.107.4.50	172.17.216.183	HTTP	1072 HTTP/1.1 206 Partial Content
1380L. 5241.132553	2a01:1111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	792 HTTP/1.1 206 Partial Content
1380L. 5241.264576	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2001:4de0:ac19::11:b:3a	HTTP	385 GET /filestreamingservice/files/c7f2c5b1-feb7-4342-84ab-901aa15c012a/pleceshash HTTP/1.1

filter:  
tcp.port eq 22

5304L. 34522.969468	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5304L. 34522.970865	172.18.0.17	172.17.216.174	SSHv2	130 Server: Encrypted packet (len=64)
5304L. 34522.972242	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5304L. 34522.973170	172.18.0.17	172.17.216.174	SSHv2	162 Server: Encrypted packet (len=96)
5304L. 34522.974849	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5305L. 34523.002059	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5305L. 34523.003936	172.18.0.17	172.17.216.174	TCP	66 22 - 54285 [ACK] Seq=4139310 Ack=538187 Win=195328 Len=0 TSval=375533702 TSecr=33658277

Het is ook mogelijk om filters te combineren, om bijvoorbeeld in te zoomen op TCP verkeer van poort 22 dat afkomstig is van een specifiek IP adres.

filter:  
tcp.port eq 22 && ip.src == 172.17.216.174

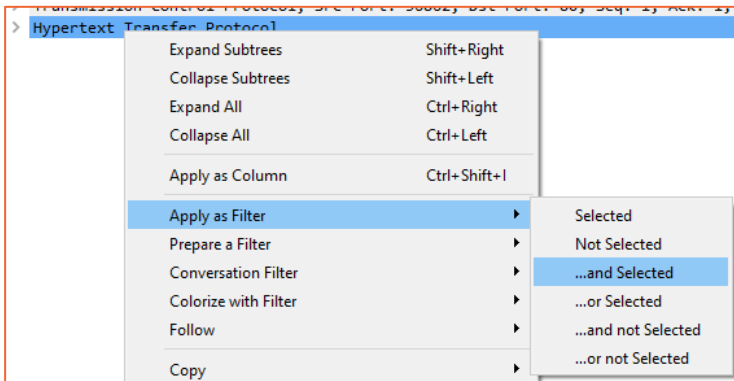
94051 1655.212608	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
94053 1655.222844	172.17.216.174	172.18.0.17	TCP	66 54285 - 22 [ACK] Seq=23787 Ack=24902 Win=44416 Len=0 TSval=790773 TSecr=342667367
94054 1655.251590	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
94056 1655.253786	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
94058 1655.257667	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
94060 1655.259768	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
94061 1655.288935	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
94066 1655.333105	172.17.216.174	172.18.0.17	TCP	66 54285 - 22 [ACK] Seq=24107 Ack=25174 Win=44416 Len=0 TSval=7908884 TSecr=342667438
94168 1669.823123	172.17.216.174	172.18.0.17	SSHv2	386 Client: Encrypted packet (len=320)

Een andere optie is bijvoorbeeld het zoeken naar pakketten met informatie of waarschuwingen zoals re transmissies, dubbele ACK's of out-of-orders.

filter:  
tcp.analysis.flags

94149 1666.082468	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2606:4700:201:681a:af0	TCP	75 [TCP Keep-Alive] 56758 - 443 [ACK] Seq=3931 Ack=289127 Win=131840 Len=1
94150 1666.088262	2606:4700:201:681a:af0	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	TCP	86 [TCP Keep-Alive ACK] 443 - 56758 [ACK] Seq=289127 Ack=3932 Win=70656 Len=0 SLE=3931 SRE=3932
94180 1676.165991	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2606:4700:201:681a:af0	TCP	75 [TCP Keep-Alive] 56821 - 443 [ACK] Seq=1183 Ack=3609 Win=130816 Len=1
94181 1676.167805	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2606:4700:3090:681b:8d72	TCP	75 [TCP Keep-Alive] 56804 - 443 [ACK] Seq=4086 Ack=26886 Win=130560 Len=1
94186 1676.172825	2606:4700:201:681a:af0	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	TCP	86 [TCP Keep-Alive ACK] 443 - 56821 [ACK] Seq=3602 Ack=1104 Win=68688 Len=0 SLE=1103 SRE=1104
94187 1676.173970	2606:4700:3090:681b:8d72	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	TCP	86 [TCP Keep-Alive ACK] 443 - 56804 [ACK] Seq=26886 Ack=4087 Win=83968 Len=0 SLE=4086 SRE=4087
94200 1679.392546	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2a00:1450:4013:c011:55	TCP	75 [TCP Keep-Alive] 56794 - 443 [ACK] Seq=11891 Ack=7755 Win=138016 Len=1
94201 1679.402497	2a00:1450:4013:c011:55	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	TCP	86 [TCP Keep-Alive ACK] 443 - 56794 [ACK] Seq=7755 Ack=11892 Win=88832 Len=0 SLE=11891 SRE=11892
94314 1694.438152	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2a00:1450:4013:c011:55	TCP	75 [TCP Keep-Alive] 56486 - 5228 [ACK] Seq=790 Ack=5018 Win=129792 Len=1
94315 1694.448172	2a00:1450:4013:c011:55	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	TCP	86 [TCP Keep-Alive ACK] 5228 - 56486 [ACK] Seq=5018 Ack=791 Win=64256 Len=0 SLE=790 SRE=791
94482 1696.641898	172.17.216.183	172.28.0.10	TCP	66 [TCP Retransmission] 56921 - 389 [SVN] Seq=0 Win=64240 Len=0 MSS=1460 US=256 SACK_PERM=1
94483 1696.642416	172.17.216.183	172.28.0.10	TCP	65 [TCP Retransmission] 56921 - 389 [SVN] Seq=0 Win=64240 Len=0 MSS=1460 US=256 SACK_PERM=1

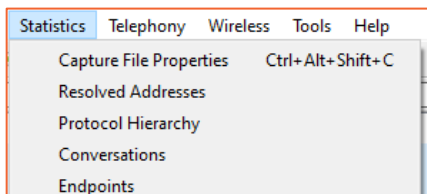
### 3.2 Apply as filter functie



Een andere manier om een filter toe te voegen of uit te breiden is door middel van een optie uit de "apply as filter" functie toe te passen op een selectie uit een pakket. In bovenstaande afbeelding is zichtbaar hoe uit een pakket het HTTP protocol aan het bestaande filter wordt toegevoegd.

### 3.3 Conversation

Wireshark biedt ook de mogelijkheid om specifiek in te zoomen op conversations of anders gezegd het verkeer tussen twee specifieke hosts.

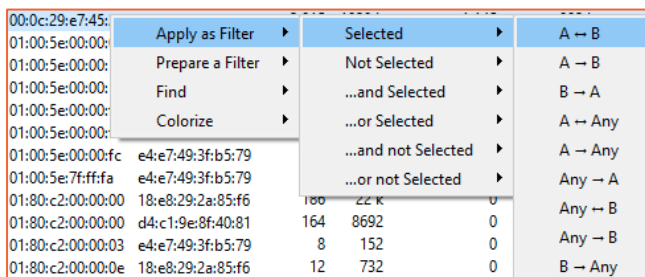


Via het menu op de taakbalk kan worden gekozen voor de optie **Statistics -> Conversation**. Er wordt vervolgens een scherm weergegeven die statistieken laat zien van alle conversations uit de packet capture.

Onderstaand screenshot toont een voorbeeld van deze statistieken. Op basis hiervan kan snel inzichtelijk worden gemaakt welke conversatie veel data verstuurd.

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
00:0c:29:3d:ef:6a	00:1a:fd:0a:af:fa	503	34 k	288	18 k	215	15 k	0.028744	36.5006	4134	13 M
00:0c:29:81:0e:e4	54:ec:2f:17:3c:d0	53.702	65 M	45.299	63 M	8.403	2335 k	0.000000	36.3919	1722	1663
00:0c:29:81:0e:e4	54:ec:2f:16:cb:90	63	7876	59	6908	4	968	1.632030	32.0918	441	213
00:0c:29:81:0e:e4	54:ec:2f:17:07:e0	63	7876	59	6908	4	968	1.632069	33.2297	6675	4074
00:0c:29:e7:45:2b	54:ec:2f:16:c6:90	30	3540	15	1722	15	1818	0.424834	31.2197	311	32 k
00:0c:29:e7:45:2b	d4:c1:9e:8f:a0:80	18	1774	9	744	9	1030	6.519076	27.8348	—	—
00:0c:29:e7:45:2b	54:ec:2f:17:3c:d0	18	2108	8	896	10	1212	7.481318	1.0738	—	—
00:0c:29:e7:45:2b	54:ec:2f:17:07:e0	18	2124	8	896	10	1228	13.521935	1.7594	—	—
00:0e:08:3a:af:cc	2c:44:fd:82:73:fc	7	2494	3	1169	4	1325	0.946711	30.0462	—	—
00:0e:10:19:95:63	ff:ff:ff:ff:ff:ff	2	120	2	120	0	0	6.489233	0.0296	—	—
00:0e:10:19:95:63	2c:44:fd:82:73:fc	1	60	0	0	1	60	6.489550	0.0000	—	—
00:11:32:9a:3d:43	ff:ff:ff:ff:ff:ff	5	613	5	613	0	0	21.466874	4.2744	—	—

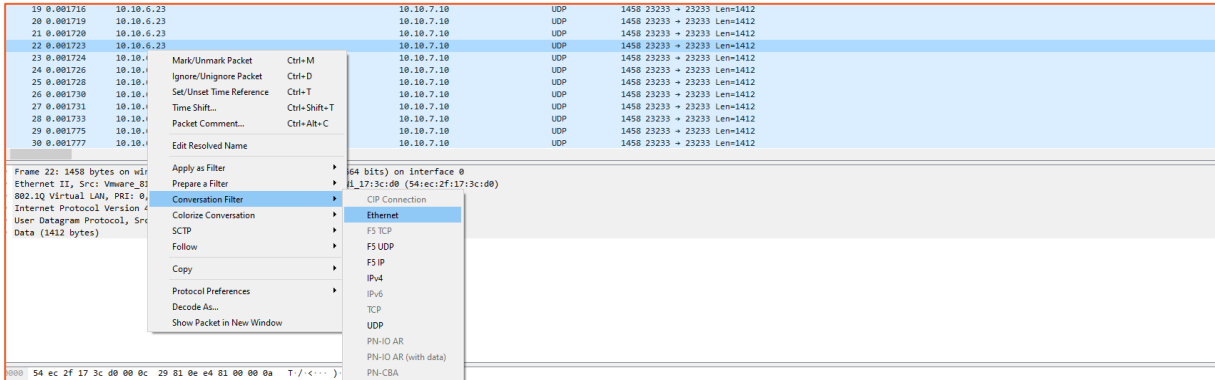
Op een specifieke conversatie inzoomen kan op twee manieren:



In vorig screenshot kan, door met de rechtermuisknop op een conversatie te klikken, deze als filter worden toegepast. Hierbij kan worden geselecteerd welk verkeer precies zichtbaar gemaakt wordt; bijvoorbeeld al het verkeer tussen A en B of alleen verkeer van A naar B. Dit resulteert in onderstaand filter:

filter  
**eth.addr==00:0c:29:e7:45:2b && eth.addr==e4:e7:49:3f:b5:79**

Een andere manier is het selecteren van een pakketje in het hoofdscherm om dan vervolgens op via de rechtermuisknop op conversation te klikken zoals wordt weergegeven in onderstaande afbeelding.



Dit resulteert weer in onderstaand filter die ook het verkeer tussen beide hosts laat zien.

```
filter
eth.addr eq 00:0c:29:81:0e:e4 and eth.addr eq 54:ec:2f:17:3c:d0
```

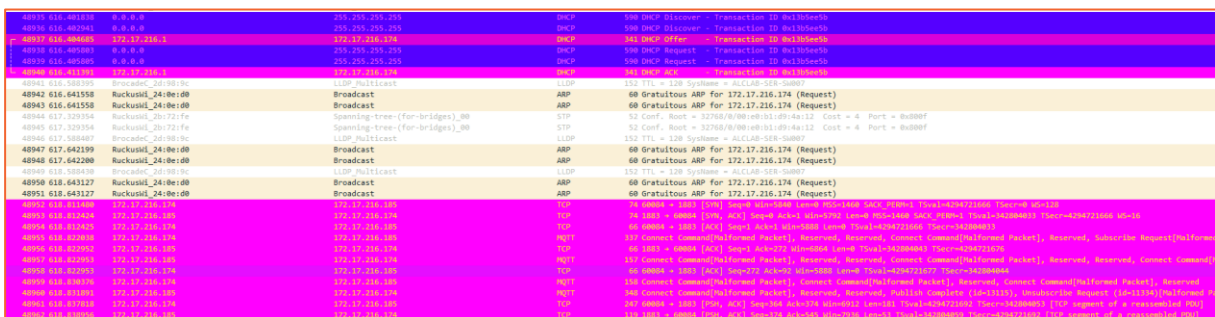
### 3.4 Coloring rules

Om in de wireshark output snel verkeer van een bepaald type te kunnen zien kan worden gewerkt met coloring rules. Deze coloring rules kunnen worden ingesteld op basis van display filters.

In onderstaand voorbeeld heeft alle verkeer van en naar IP adres 172.17.216.174 een magenta kleur gekregen en alle DHCP verkeer blauw.



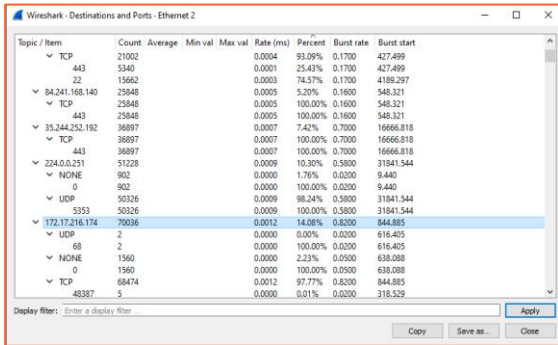
Effect:



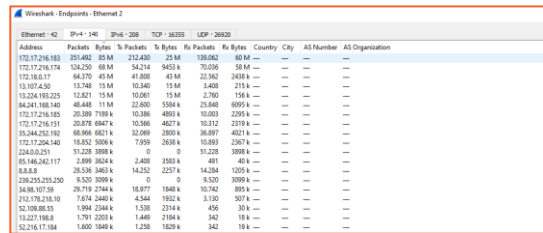
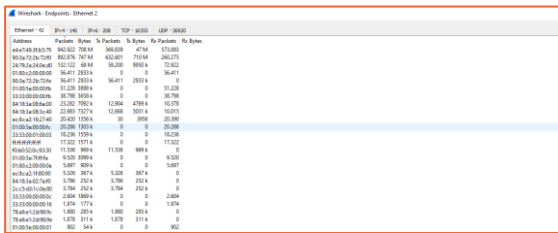
### 3.5 Wireshark tools

In wireshark zitten ook nog een aantal analyse tools ingebouwd die inzicht kunnen geven in de packet capture en het verkeer.

Er is bijvoorbeeld een tool om statistieken weer te geven van poorten en type verkeer gesorteerd op IP-adres.



Er is een tool die statistieken kan genereren op basis van ethernet of IP-adres, zodat je kan zien welke device welke hoeveelheid verkeer genereerd.



Er is een IO grafiek die verkeer visueel kan weergeven, ook bijvoorbeeld op basis van display filters. Hiermee zou je bijvoorbeeld in kaart kunnen brengen of een apparaat over tijd ineens veel meer of juist veel minder verkeer van een bepaald type genereerd.

