



ICX TECHNOTE - PORT MIRROR V1.0

Ruckus ICX configuratie – Port mirror



Inhoud

1 Inleiding		2
1.1	DOELSTELLING	2
1.2	BEOOGD PUBLIEK	2
1.3	VOORKENNIS/BENODIGDHEDEN	2
2 Port mirr	oring	3
2.1	STANDAARD WAARDEN	4
2.2	ENKELE POORT MONITOREN	5
2.3	MEERDERE POORTEN MONITOREN	5
2.4	MONITOREN INPUT EN OUTPUT VERKEER	6
2.5	Monitoren van een LAG	6
3 Wireshar	·k	7
3.1	FILTERS	7
3.2	APPLY AS FILTER FUNCTIE	9
3.3	CONVERSATION	9
3.4	COLORING RULES1	.0
3.5	WIRESHARK TOOLS 1	1

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 wordt ingevoerd.

ICX7150(config)#show mirr	or
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 gebridged 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
                        : (U1/M1)
: (U1/M1)
  Input monitoring
                                         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
                    : (U1/M1) 6 12
: (U1/M1) 6 12
  Input monitoring
  Output monitoring
```

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
                     : None
: None
 Input monitoring
 Output monitoring
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
Input monitoring : (LAG) 1
Output monitoring : (U1/M3) 1
Output monitoring : (LAG) 1
                                                2
                                                2
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	dr 24.70.252	4.0.0.40		
etn.ad	ar == 24:79:2a2	4:0e:00		
49086 638.253378	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
49087 638.254248	172.18.0.17	172.17.216.174	SSHv2	130 Server: Encrypted packet (len=64)
49088 638.254249	172.17.216.174	172.18.0.17	TCP	66 54285 → 22 [ACK] Seq=2075 Ack=3606 Win=16128 Len=0 TSval=4294741108 TSecr=341650443
49091 638.559703	172.17.216.174	172.18.0.17	SSHv2	178 Client: Encrypted packet (len=112)
49092 638.561755	172.18.0.17	172.17.216.174	SSHv2	146 Server: Encrypted packet (len=80)
49093 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=341650751
49094 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 1715.626053	172.18.0.17	172.17.216.174	SSHv2	114 Server: Encrypted packet (len=48)		
95627 1715.627074	172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)		
95628 1715.628071	172.18.0.17	172.17.216.174	SSHv2	130 Server: Encrypted packet (len=64)		
95629 1715.629368	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:					
ancp					
- 48935 616.401838 0.0.0.0	0 255.25	5.255.255	DHCP	590 DHCP Discove	er - Transaction ID 0x13b5ee5b
48936 616.402941 0.0.0.0	0 255.25	5.255.255	DHCP	590 DHCP Discove	er - Transaction ID 0x13b5ee5b
48937 616.404685 172.17.	.216.1 172.17	.216.174	DHCP	341 DHCP Offer	- Transaction ID 0x13b5ee5b
48938 616.405803 0.0.0.0	0 255.25	5.255.255	DHCP	590 DHCP Request	t - Transaction ID 0x13b5ee5b
48939 616.405805 0.0.0.0	0 255.25	5.255.255	DHCP	590 DHCP Request	t - Transaction ID 0x13b5ee5b
48940 616.411391 172.17.	.216.1 172.17	.216.174	DHCP	341 DHCP ACK	- Transaction ID 0x13b5ee5b
1122 3227.645415 0.0.0.0	0 255.25	5.255.255	DHCP	590 DHCP Request	t - Transaction ID 0x1e30f7d1
1122 3227.645416 0.0.0.0	0 255.25	5.255.255	DHCP	590 DHCP Request	t - Transaction ID 0x1e30f7d1
filter: arp					
1118 3164 600122 Duck	niski 26•72•f0	Broadcast		ADD	60 Who has 172 17 216 1372 Toll 172 17 216 1
1118 3165 692269 Ruck	cusWi 2b:72:f0	Broadcast		ARP	60 Who has 172 17 216 183? Tell 172 17 216 1
1118 3165 602271 Buck	cushi 2h.72.fA	Broadcast		ADD	69 Who has 172 17 216 1932 Tall 172 17 216 1
1118 3165 692449 Head	at+D 3f-b5-70	Puckuski 2h:72:50		APD	42 172 17 216 183 ir at adva7:49:3f:b5:79
1118 2165 602407 Herei	lettp 26.65.70	Bushustii 26.72.10			42 172 17 216 182 is at c4.c7.40.26.b5.70
1110	uetti 2h.72.f0	Ruckuswi_20.72.10			42 1/2.1/.210.103 15 dt 04.0/.45.51.05.75
1110	(usw1_20.72.10 (usk1 2b.72.fo	Prophese			60 Libe has 172.17.216.174; TCL1 172.17.210.1
1110	uswi_20.72.10	Distances C		100	00 WHO HIDS 1/2.1/.210.1/4: 1011 1/2.1/.210.1
1110	uswi_24:00:00	RUCKUSWI_20:72:10		ARP	00 1/2.1/.210.1/4 15 dt 24:/9:2d:24:00:00
1110 51/0.442288 RUCK	uswi_24:00:00	RUCKUSWI_20:72:10		ARP	00 WHU Has 1/2.1/.210.11 Tell 1/2.1/.210.1/4
1118 31/0.443339 RUCK	CUSW1_2D:/2:T0	RUCKUSW1_24:00:00		AKP	60 1/2.1/.216.1 15 at 90:3a:/2:2D:/2:T0

filter:				
http				
1371 5236.329340	13.107.4.50	172.17.216.183	HTTP	305 HTTP/1.1 403 Forbidden
1371 5236.364266	2a01:111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	327 HTTP/1.1 403 Forbidden
1371 5236.370249	13.107.4.50	172.17.216.183	HTTP	305 HTTP/1.1 403 Forbidden
1371 5236.373556	2a01:111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	325 HTTP/1.1 403 Forbidden
1371 5236.375997	13.107.4.50	172.17.216.183	HTTP	305 HTTP/1.1 403 Forbidden
1376 5241.064064	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2a01:111:2003::50	HTTP	501 GET /filestreamingservice/files/fc989f39-6b38-4ee7-b13a-6f6b68435329?P1=1582566391&P2=4028
1376 5241.066116	172.17.216.183	13.107.4.50	HTTP	481 GET /filestreamingservice/files/fc989f39-6b38-4ee7-b13a-6f6b68435329?P1=1582566391&P2=4028
1376 5241.072872	2a01:111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	1088 HTTP/1.1 206 Partial Content
1376 5241.073970	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2a01:111:2003::50	HTTP	513 GET /filestreamingservice/files/fc989f39-6b38-4ee7-b13a-6f6b68435329?P1=1582566391&P2=4028
1376 5241.075257	13.107.4.50	172.17.216.183	HTTP	1072 HTTP/1.1 206 Partial Content
1380 5241.132553	2a01:111:2003::50	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	HTTP	792 HTTP/1.1 206 Partial Content
1380 5241.264576	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2001:4de0:ac19::1:b:3a	HTTP	385 GET /filestreamingservice/files/c7f2c5b1-feb7-4342-84ab-901aa15c012a/pieceshash HTTP/1.1

filter: tcp.port eq 22

5304 34522.969460 172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5304 34522.970365 172.18.0.17	172.17.216.174	SSHv2	130 Server: Encrypted packet (len=64)
5304 34522.972242 172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5304 34522.973170 172.18.0.17	172.17.216.174	SSHv2	162 Server: Encrypted packet (len=96)
5304 34522.974049 172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5305 34523.002859 172.17.216.174	172.18.0.17	SSHv2	130 Client: Encrypted packet (len=64)
5305 34523.003936 172.18.0.17	172.17.216.174	TCP	66 22 → 54285 [ACK] Seq=413910 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.po	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 1655.253706	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=790884 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.ar	alysis.flags			
94149 1666.082468	2a00:18c8:3e2e:10d8:3950:4ca4:d3b4:ec3e	2606:4700:20::681a:af0	TCP	75 [TCP Keep-Alive] 56758 → 443 [ACK] Seq=3931 Ack=289127 Win=131840 Len=1
94150 1666.088262				86 [TCP Keep-Alive ACK] 443 → 56758 [ACK] Seq-289127 Ack-3932 Win-70656 Len-0 SLE-3931 SRE-3932
94184 1676.166991				75 [TCP Keep-Alive] 56821 + 443 [ACK] Seq=1103 Ack=3602 Win=130816 Len=1
94185 1676.167965				75 [TCP Keep-Alive] 56804 → 443 [ACK] Seq=4006 Ack=26086 Win=130560 Len=1
94186 1676.172825				86 [TCP Keep-Alive ACK] 443 → 56821 [ACK] Seg=3602 Ack=1104 Win=68608 Len=0 SLE=1103 SRE=1104
94187 1676.173970				86 [TCP Keep-Alive ACK] 443 + 56804 [ACK] Seq=26086 Ack=4007 Win=83968 Len=0 SLE=4006 SRE=4007
94200 1679.392546				75 [TCP Keep-Alive] 56754 → 443 [ACK] Seq=11091 Ack=7755 Win=130816 Len=1
94291 1679 492497		2500119c912c2010d91205014c541d2b41cc2c		96 [TCD Keep-Alive ACK] 442 + 56754 [ACK] See-7755 Ack-11002 Min-98922 Len-8 CL5-11001 SEE-11002

3.2 Apply as filter functie

110113112333	ton concret riotocoty site		. 00, 50q, 1, ACK, 1,
> Hypertext	Transfer Protocol		
	Expand Subtrees	Shift+Right	
	Collapse Subtrees	Shift+Left	
	Expand All	Ctrl+Right	
	Collapse All	Ctrl+Left	
	Apply as Column	Ctrl+Shift+I	
	Apply as Filter	•	Selected
	Prepare a Filter	•	Not Selected
	Conversation Filter	•	and Selected
	Colorize with Filter	•	or Selected
	Follow	•	and not Selected
	Сору	•	or not Selected

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

Statistics	Telephony	Wireless	Tools	Help		
Capture File Properties Ctrl+Alt+Shift+C						
Resolved Addresses						
Protocol Hierarchy						
Conversations						
Endp	oints					

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.

anders gezegd het verkeer tussen twee specifieke

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

Ethernet · 620	IPv4 · 1015	IPv6 • 84	TCP · 8	56 UDP • 729							
Address A	Address B	Packets	Bytes	$Packets\:A\toB$	Bytes $A \rightarrow B$	$Packets\:B\toA$	Bytes B \rightarrow A	Rel Start	Duration	$Bits/s\:A\toB$	Bits/s B \rightarrow /
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	
00:0c:29:81:0e:e4	54:ec:2f:17:3c:d	0 53.702	65 M	45.299	63 M	8.403	2335 k	0.000000	36.3919	13 M	
00:0c:29:81:0e:e4	54:ec:2f:16:c6:90	0 63	7876	59	6908	4	968	1.632030	32.0918	1722	
00:0c:29:81:0e:e4	54:ec:2f:17:07:e0	0 63	7876	59	6908	4	968	1.632069	33.2297	1663	
00:0c:29:e7:45:2b	54:ec:2f:16:c6:90	0 30	3540	15	1722	15	1818	0.424834	31.2197	441	
00:0c:29:e7:45:2b	d4:c1:9e:8f:40:8	0 18	1774	9	744	9	1030	6.519076	27.8343	213	
00:0c:29:e7:45:2b	54:ec:2f:17:3c:d	0 18	2108	8	896	10	1212	7.481318	1.0738	6675	
00:0c:29:e7:45:2b	54:ec:2f:17:07:el) 18	2124	8	896	10	1228	13.521935	1.7594	4074	
00:0e:08:3a:af:cc	2c:44:fd:82:73:fd	: 7	2494	3	1169	4	1325	0.946711	30.0462	311	
00:0e:10:19:95:63	ff:ff:ff:ff:ff:ff	2	120	2	120	0	0	6.489233	0.0296	32 k	
00:0e:10:19:95:63	2c:44:fd:82:73:fd	: 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	21.468874	4.2744	1147	

Op een specifieke conversatie inzoomen kan op twee manieren:

00:0c:29:e7:45:	Apply as Filter			Colocted		ΔΡ
01:00:5e:00:00:	Apply as Fliter	- *		Selected	· ·	A⇔b
01:00:5e:00:00:	Prepare a Filter	•		Not Selected	•	$A \rightarrow B$
01:00:5e:00:00:	Find	•		and Selected	•	B → A
01:00:5e:00:00:	Colorize	•		or Selected	•	A ↔ Anv
01:00:5e:00:00:	COIOTIZC			mor beleeted		A ··· Any
01:00:5e:00:00:fc	e4:e7:49:3f:b5:79			and not Selecte	d 🕨	A → Any
01:00:5e:7f:ff:fa	e4:e7:49:3f:b5:79			or not Selected	•	Any → A
01:80:c2:00:00:00	18:e8:29:2a:85:f6		180	22 K		Any - R
01:80:c2:00:00:00	d4:c1:9e:8f:40:81		164	8692	0	Any + b
01:80:c2:00:00:03	e4:e7:49:3f:b5:79		8	152	0	Any → B
01:80:c2:00:00:0e	18:e8:29:2a:85:f6		12	732	0	B → Any
					-	

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.

19 0.001716	10.10.6.2	3		10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
20 0.001719	10.10.6.2	3		10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
21 0.001720	10.10.6.2	3		10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
22 0.001723	10.10.6.2	3		10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
23 0.001724	10.10.0	Mark/Unmark Packet	Ctrl+M	10.10.7.10	UDP	1458 23233 > 23233 Len=1412	
24 0.001726	10.10.0	Inners (University Destant	CHL D	10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
25 0.001728	10.10.	ignore/orignore Facket	Cuito	10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
26 0.001730	10.10.	Set/Unset Time Reference	Ctrl+T	10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
27 0.001731	10.10.0	Time Shift	Ctrl+Shift+T	10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
28 0.001733	10.10.0	Packet Comment	Ctrl+Alt+C	10.10.7.10	UDP	1458 23233 > 23233 Len=1412	
29 0.001775	10.10.0		current c	10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
30 0.001777	10.10.	Edit Resolved Name		10.10.7.10	UDP	1458 23233 → 23233 Len=1412	
				-			
Frame 22: 1458 by	tes on wir	Apply as Filter	•	64 bits) on interface 0			
Ethernet II. Src:	Vmware 81	Prepare a Filter	•	1 17:3c:d0 (54:ec:2f:17	:3c:d0)		
802.1Q Virtual LA	N, PRI: 0,	Conversation Filter	•	CIP Connection			
Internet Protocol	Version 4	Colorize Conversation		Ethernet			
User Datagram Pro	tocol, Sro	come conversation		56 700			
Data (1412 bytes)		SCIP	,	F5 ICP			
		Follow	•	F5 UDP			
				F5 IP			
		сору		IPv4			
		Protocol Preferences	•	IPv6			
		Decode As		TCD			
		CL D. L. C. M. MC. L		ICP			
		Show Packet in New Windo	w	UDP			
				PN-IO AR			
				PN-IO AR (with data)			
				D11 C04			
0000 54 ec 2f 17 3	c d0 00 0c 2	19 81 0e e4 81 00 00 0a	1.7)	PIN-CBA			

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.

🥖 Wireshark · Coloring Rules Def	ult	?	×
Name	Filter		
AP_X	ip.addr == 172.17.216.174		
DHCP	dhcp		

Effect:

48935 616.401838	0.0.0.0	255.255.255.255	DHCP	590 DHCP Discover - Transaction ID 0x13b5ee5b
48936 616.402941				590 DHCP Discover - Transaction ID 0x13b5ee5b
48937 616.484685			DHCP	341 DHCP Offer - Transaction ID 0x13bSee5b
48938 616.405803				590 DHCP Request - Transaction ID 0x13b5ee5b
48939 616.405805				590 DHCP Request - Transaction ID 0x13b5ee5b
48940 616.411391			DHCP	341 OHCP ACK - Transaction ID 0x13bSee5b
48941 616.588395	BrocadeC_2d:98:9c	LLDP_Multicast	LLDP	152 TTL = 120 SysName = ALCLAB-SER-SW007
48942 616.641558	RuckusWi_24:0e:d0	Broadcast	ARP	60 Gratuitous ARP for 172.17.216.174 (Request)
48943 616.641558	RuckusWi_24:0e:d0	Broadcast	ARP	60 Gratuitous ARP for 172.17.216.174 (Request)
48944 617.329354	RuckusWi_2b:72:fe	Spanning-tree-(for-bridges)_00		52 Conf. Root = 32768/0/00:e0:b1:d9:4a:12
48945 617.329354	RuckusWi_2b:72:fe	Spanning-tree-(for-bridges)_00	STP	52 Conf. Root = 32768/0/00:e0:b1:d9:4a:12
48946 617.588407	BrocadeC_2d:98:9c	LLDP_Multicast	LLDP	152 TTL = 120 SysName = ALCLAB-SER-SN007
48947 617.642199	RuckusWi_24:0e:d0	Broadcast	ARP	60 Gratuitous ARP for 172.17.216.174 (Request)
48948 617.642200	RuckusWi_24:0e:d0	Broadcast	ARP	60 Gratuitous ARP for 172.17.216.174 (Request)
48949 618.588430	BrocadeC_2d:98:9c	LLDP_Multicast	LLDP	152 TTL = 120 SysName = ALCLAB-SER-SN007
48950 618.643127	RuckusWi_24:0e:d0	Broadcast	ARP	60 Gratuitous ARP for 172.17.216.174 (Request)
48951 618.643127	RuckusWi_24:0e:d0	Broadcast	ARP	60 Gratuitous ARP for 172.17.216.174 (Request)
48952 618.811480	172.17.216.174	172.17.216.185		74 60054 + 1883 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=4294721666 TSecr=0 WS=128
48953 618.812424				74 1883 + 60084 [SYN, ACK] Seq=0 Ack=1 Win=5792 Lem=0 MSS=1460 SACK_PERM=1 TSval=342804033 TSecr=4294721666 WS=16
48954 618.812425				66 60004 → 1883 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=4294721666 TSecr=342804033
48955 618.822038				337 Connect Command[Malformed Packet], Reserved, Reserved, Connect Command[Malformed Packet], Reserved, Subscribe Request[Malformed
48956 618.822952				66 1883 → 60084 [ACK] Seq=1 Ack=272 Win=6864 Len=0 TSval=342804043 TSecr=4294721676
48957 618.822953				157 Connect Command[Malformed Packet], Reserved, Reserved, Connect Command[Malformed Packet], Reserved, Reserved, Connect Command[Palformed Packet], Reserved, Reserved, Connect Command[Pa
48958 618.822953				66 60004 → 1883 [ACK] Seq=272 Ack=92 Win=5888 Len=0 TSval=4294721677 TSecr=342804044
48959 618.830376				158 Connect Command[Malformed Packet], Connect Command[Malformed Packet], Reserved, Connect Command[Malformed Packet], Reserved
48960 618.831891				348 Connect Command[Malformed Packet], Reserved, Reserved, Publish Complete (id-13115), Unsubscribe Request (id-11334)[Malformed Pa
48961 618.837818				247 60004 + 1883 [PSH, ACK] Seq=364 Ack=374 Win=6912 Len=181 TSval=4294721692 TSecr=342804053 [TCP segment of a reassembled PDU]
48962 618 838956				119 1883 → 60084 [PSH, ACK] Seg=374 Ack=545 Win=7936 Len=51 TSval=342804059 TSecr=4294721692 [TCP segment of a reassembled PDU]

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.

tania.	/ 18.000	Court	A	After and	A.Less cond	Pate (mail	Barriet	Dt.enter	P		 -
opic	item	Count	Average	Min vai	Max vai	nate (ms)	Percent	Durst rate	burst start		
	V ICP	21002				0.0004	93.09%	0.1700	421,499		
	443	3340				0.0001	23.45%	0.1700	421,499		
	22	15662				0.0005	74.57%	0.1700	4189.297		
~	84.241.168.140	25848				0.0005	5.20%	0.1600	548.321		
	V ICP	23848				0.0005	100.00%	0.1600	548.321		
	443	25848				0.0005	100.00%	0.1600	548.321		
~	35.244.252.192	36897				0.0007	7.42%	0.7000	10666.818		
	~ ICP	36897				0.0007	100.00%	0.7000	16666.818		
	443	36897				0.0007	100.00%	0.7000	16666.818		
~	224.0.0.251	51228				0.0009	10.30%	0.5800	31841.544		
	~ NONE	902				0.0000	1.76%	0.0200	9.440		
	0	902				0.0000	100.00%	0.0200	9.440		
	✓ UDP	50326				0.0009	98.24%	0.5800	31841.544		
	5353	50326				0.0009	100.00%	0.5800	31841.544		
~	172.17.216.174	70036				0.0012	14.08%	0.8200	844,885		
	✓ UDP	2				0.0000	0.00%	0.0200	616.405		
	68	2				0.0000	100.00%	0.0200	616.405		
	✓ NONE	1560				0.0000	2.23%	0.0500	638.088		
	0	1560				0.0000	100.00%	0.0500	638.088		
	✓ TCP	68474				0.0012	97.77%	0.8200	844.885		
	48387	5				0.0000	0.01%	0.0200	318.529		
	Party Party and and	a dina								i	 _

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.

