

# Universidade Presbiteriana Mackenzie

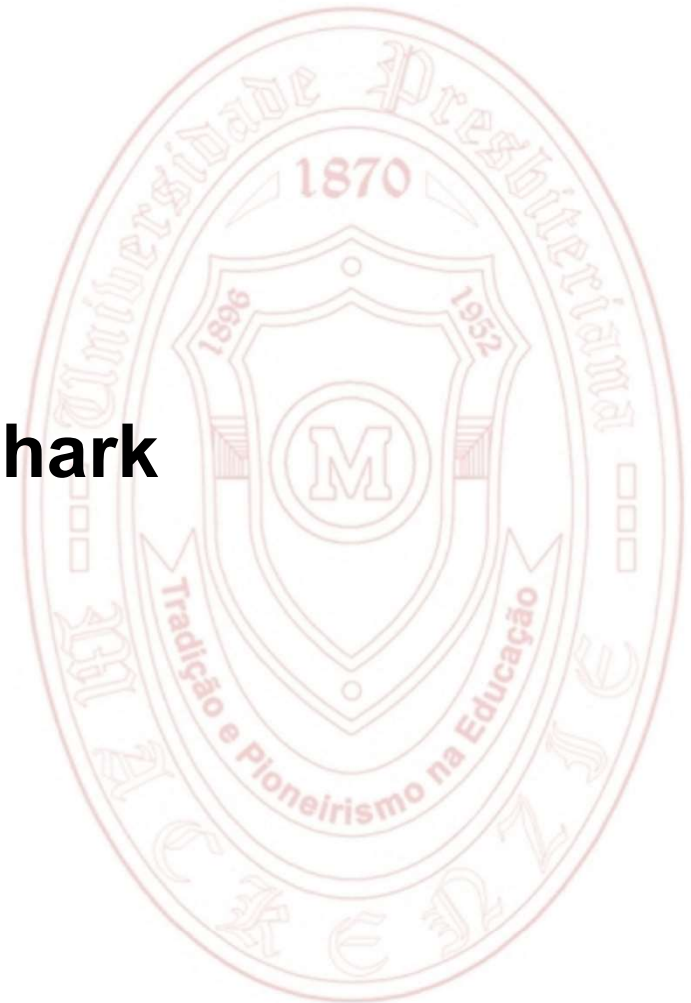
## FCI – Faculdade de Computação e Informática



## Camada de Aplicação





**Prof: Dr. Bruno Rodrigues**

- Monitoramento de redes
- Wireshark
- Experiências usando Wireshark

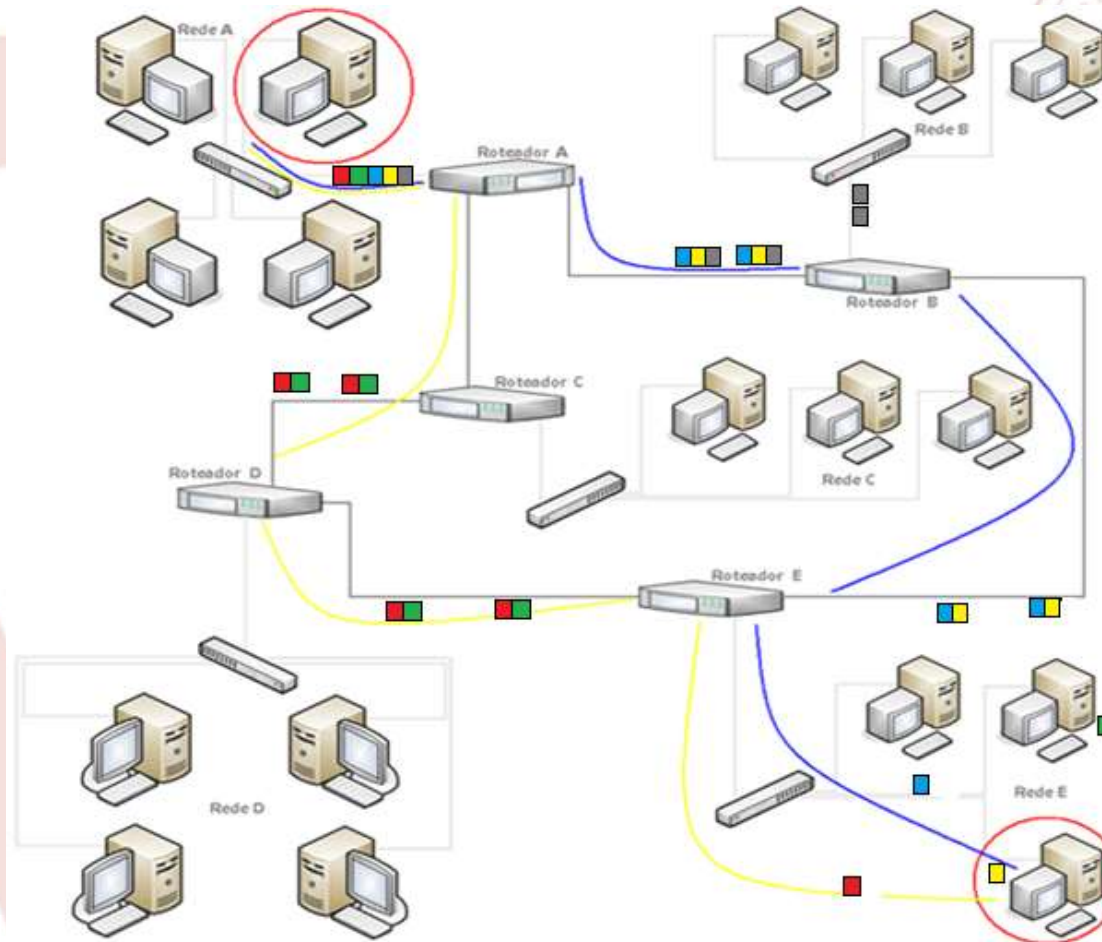


## ***Análise de pacotes de redes***

Análise de pacotes, muitas vezes referida como **packet sniffing** (farejamento de pacotes) ou **análise de protocolo**, descreve o processo de capturar e interpretar dados em tempo real à medida que flui através de uma rede

-  Entender melhor o que está acontecendo na rede.
-  Identificar quem está em uma rede
-  Identificar possíveis ataques ou atividades maliciosas
-  Encontrar aplicações inseguras.

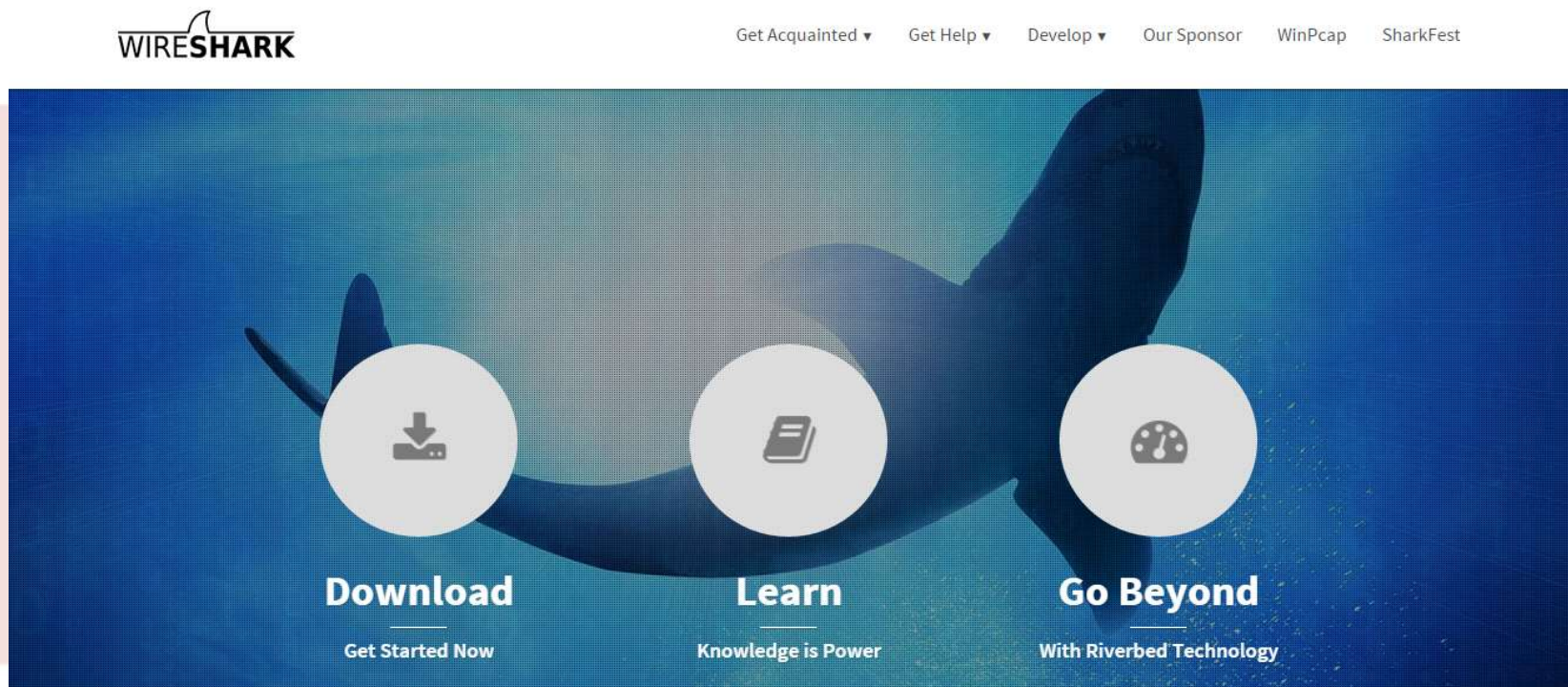
## Analise de pacotes de redes









## Analise de pacotes de redes

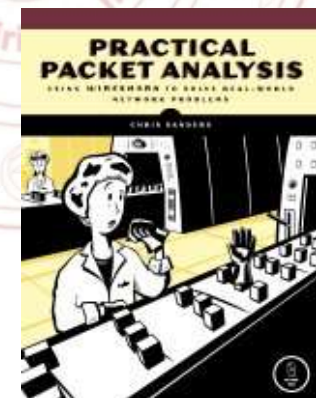
Existem diferentes farejadores (sniffing) de pacotes disponíveis para análise de desempenho de uma rede. Com intuito de aprimorar nosso conhecimento iremos adotar o **Wireshark**.



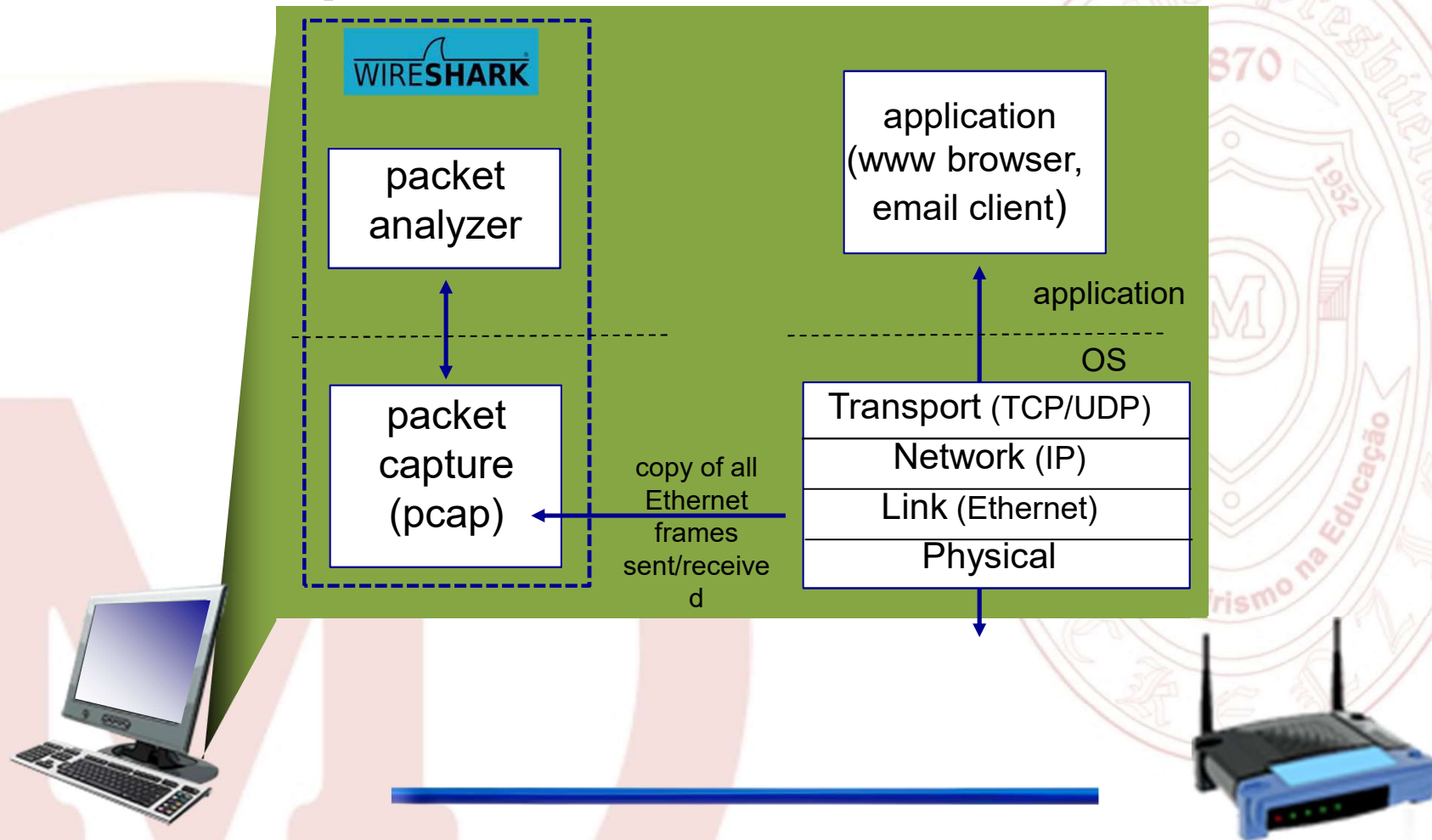
## Análise de pacotes de redes

Existem diferentes farejadores (sniffing) de pacotes disponíveis para análise de desempenho de uma rede. Com intuito de aprimorar nosso conhecimento iremos adotar o Wireshark pelas seguintes **razões**:

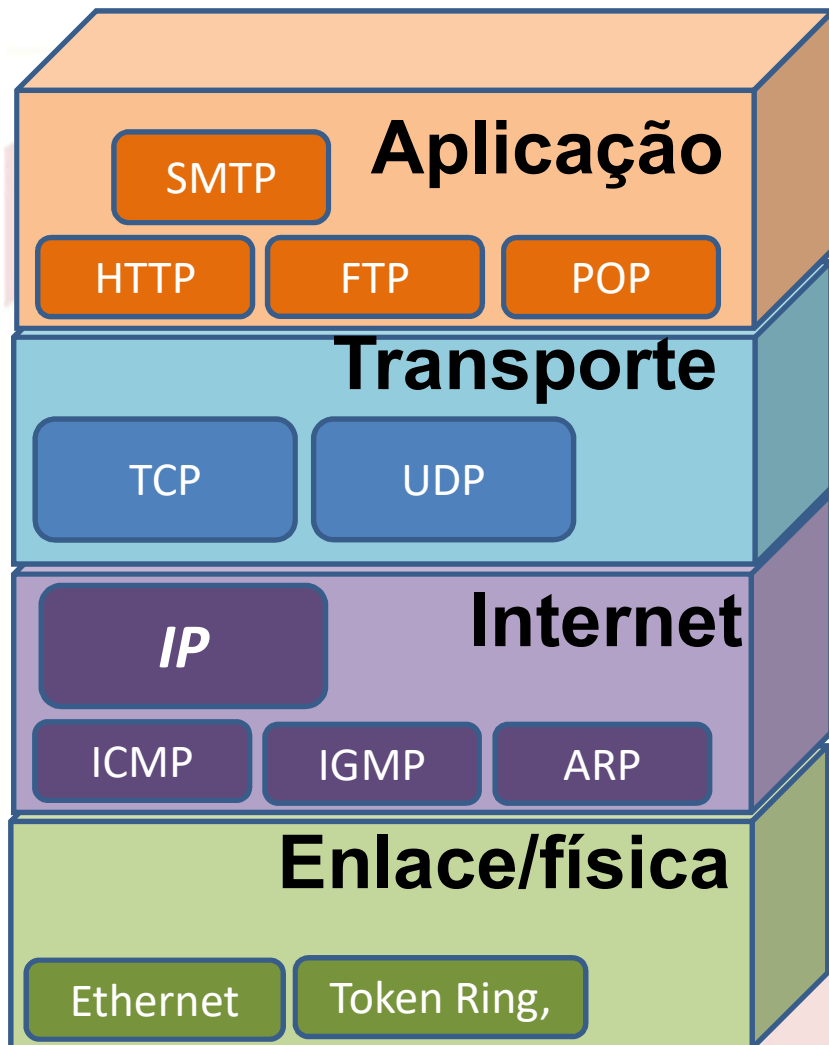
-  Interface amigável (GUI)
-  Custo – Ferramenta OpenSource
-  Suporte ao programa
-  Suporte ao SO



## Analise de pacotes de redes







A pilha de encapsulamento pode ser usada para atacar redes



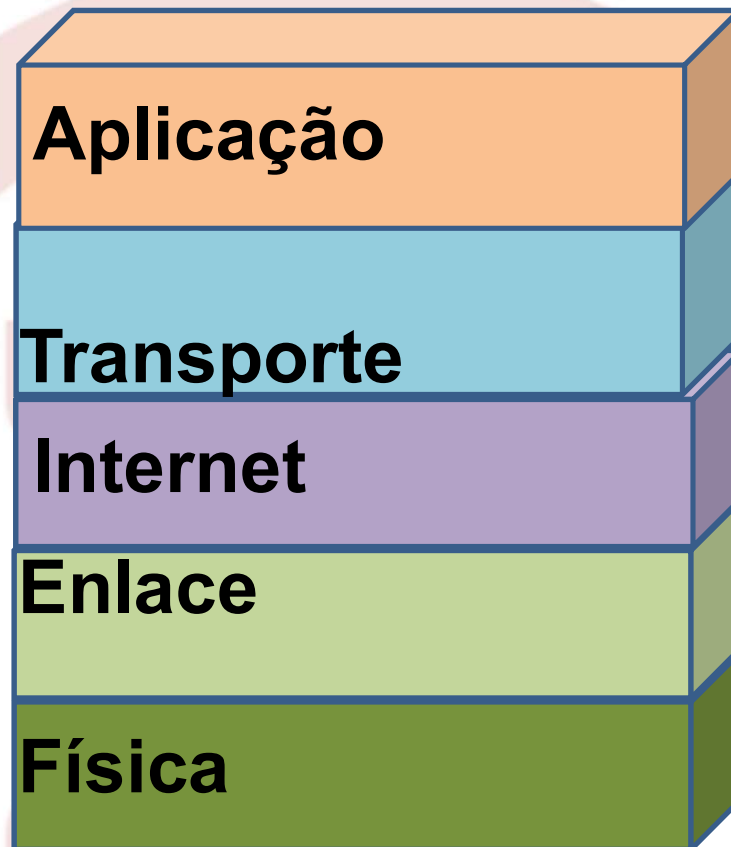
É necessário entender todos os elementos para criar proteções



Cada camada de rede tem ameaças específicas



## Analise de pacotes de redes



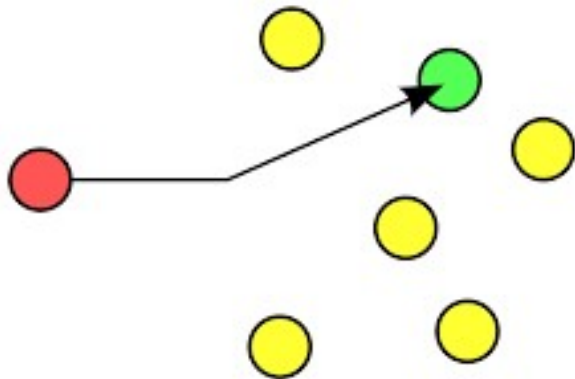
Endereços IP

Tabelas CAM

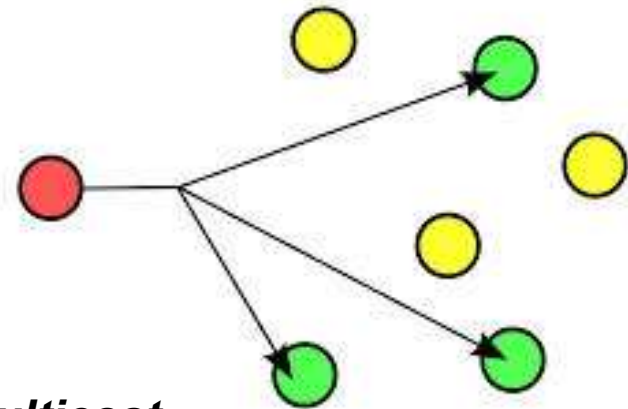
Broadcast

## Analise de pacotes de redes

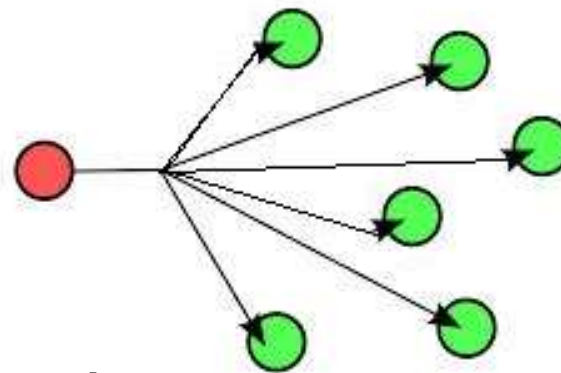
**Unicast**



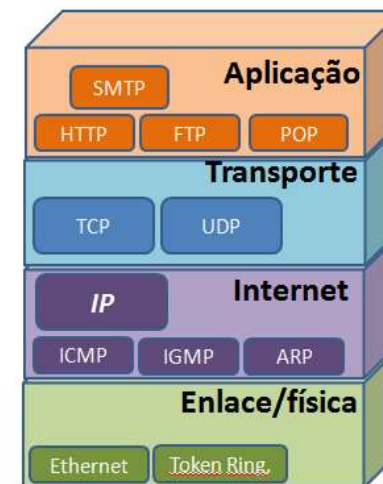
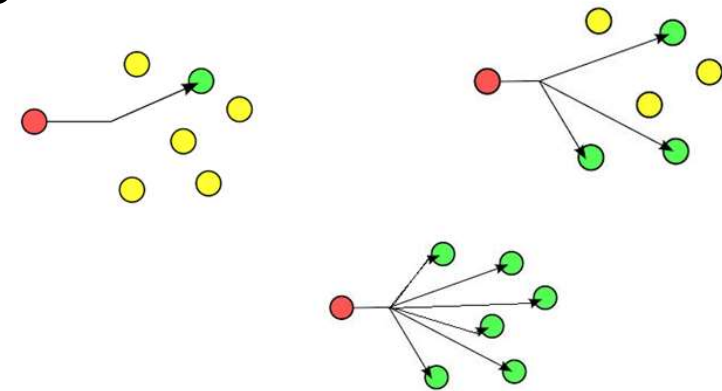
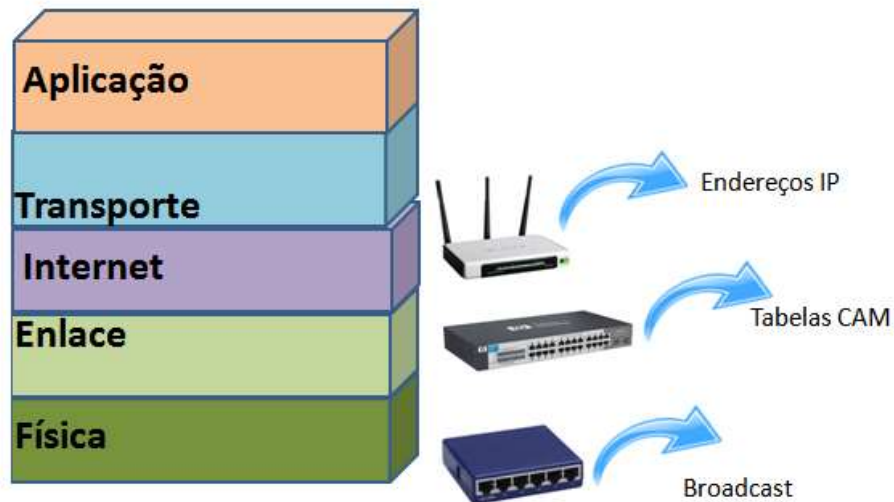
**Multicast**



**Broadcast**



## Analise de pacotes de redes



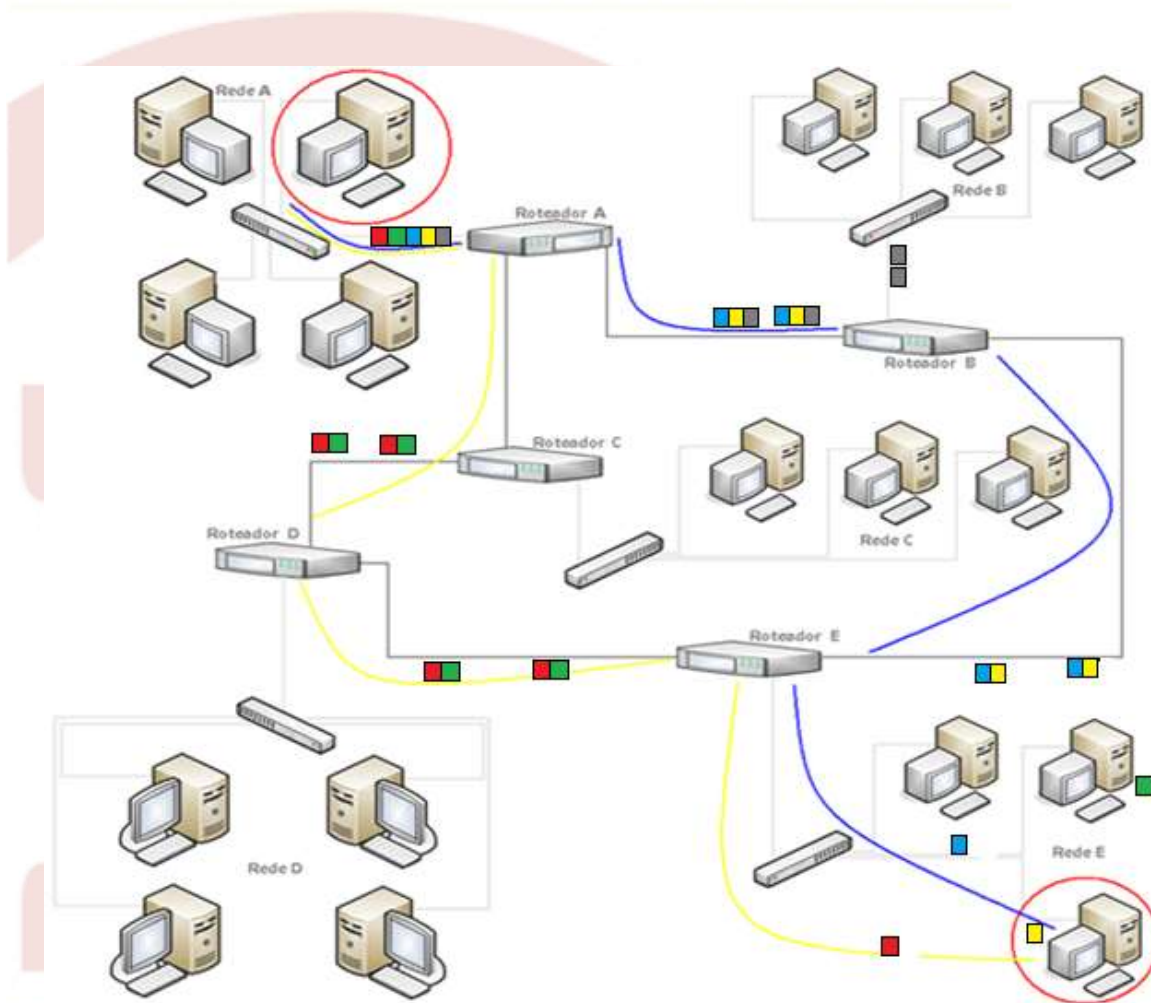
## Analise de pacotes de redes - Modo Promísquo

É o modo promísquo que permite uma NIC ver todos os pacotes que atravessam a rede!!!!





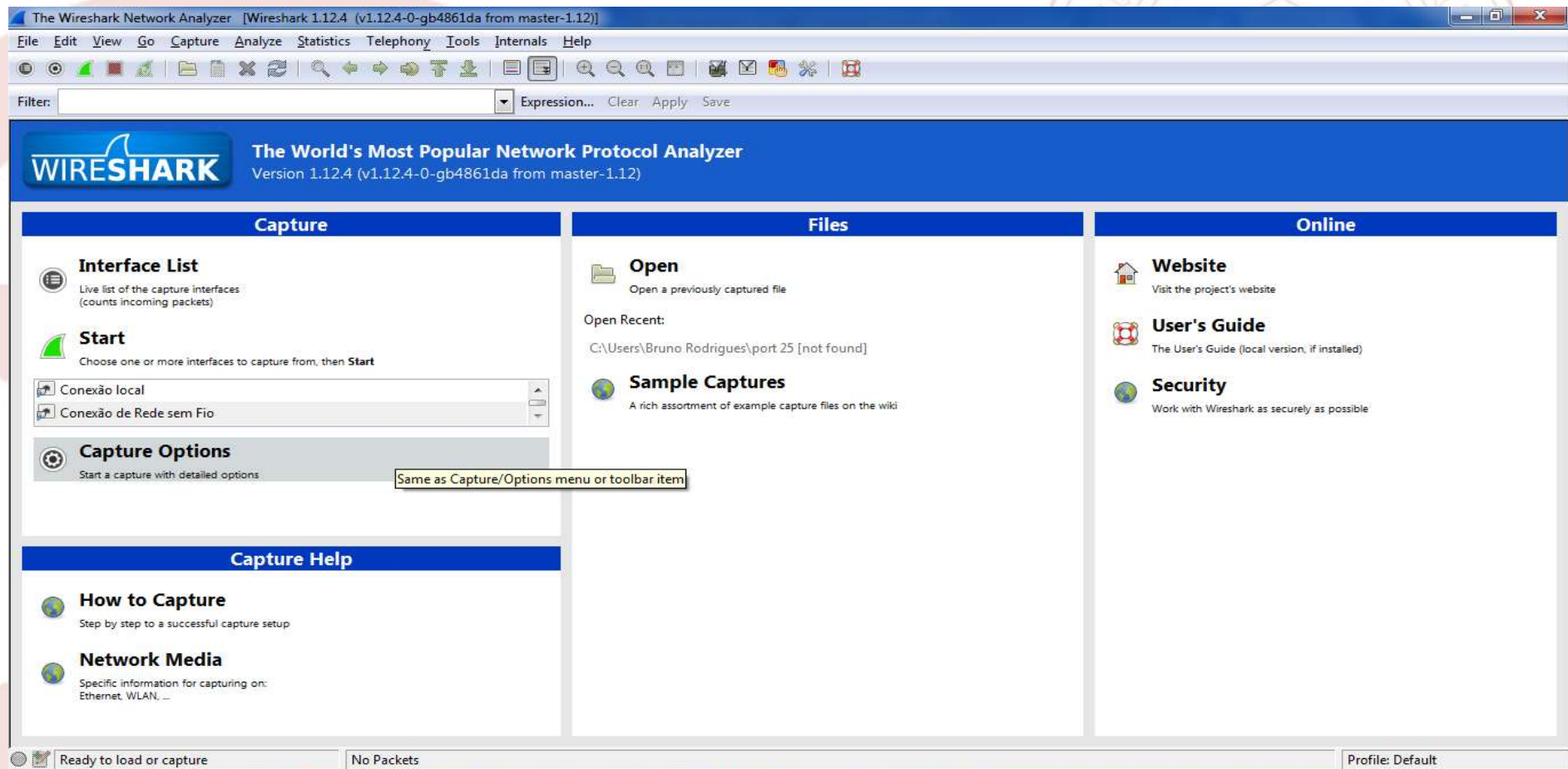
## Analise de pacotes de redes



# Iniciando e conhecendo o WireShark

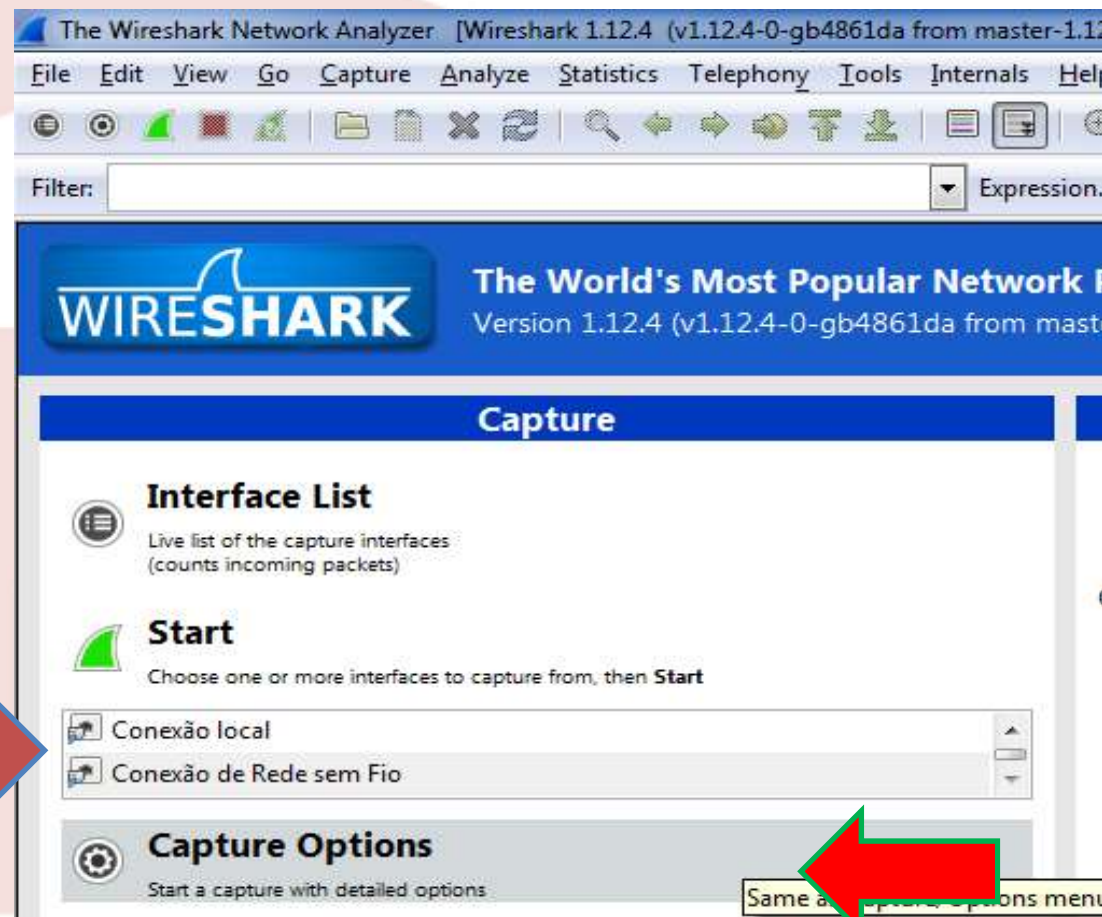
## Analise de pacotes de redes

### Conhecendo o Wireshark:



## Analise de pacotes de redes

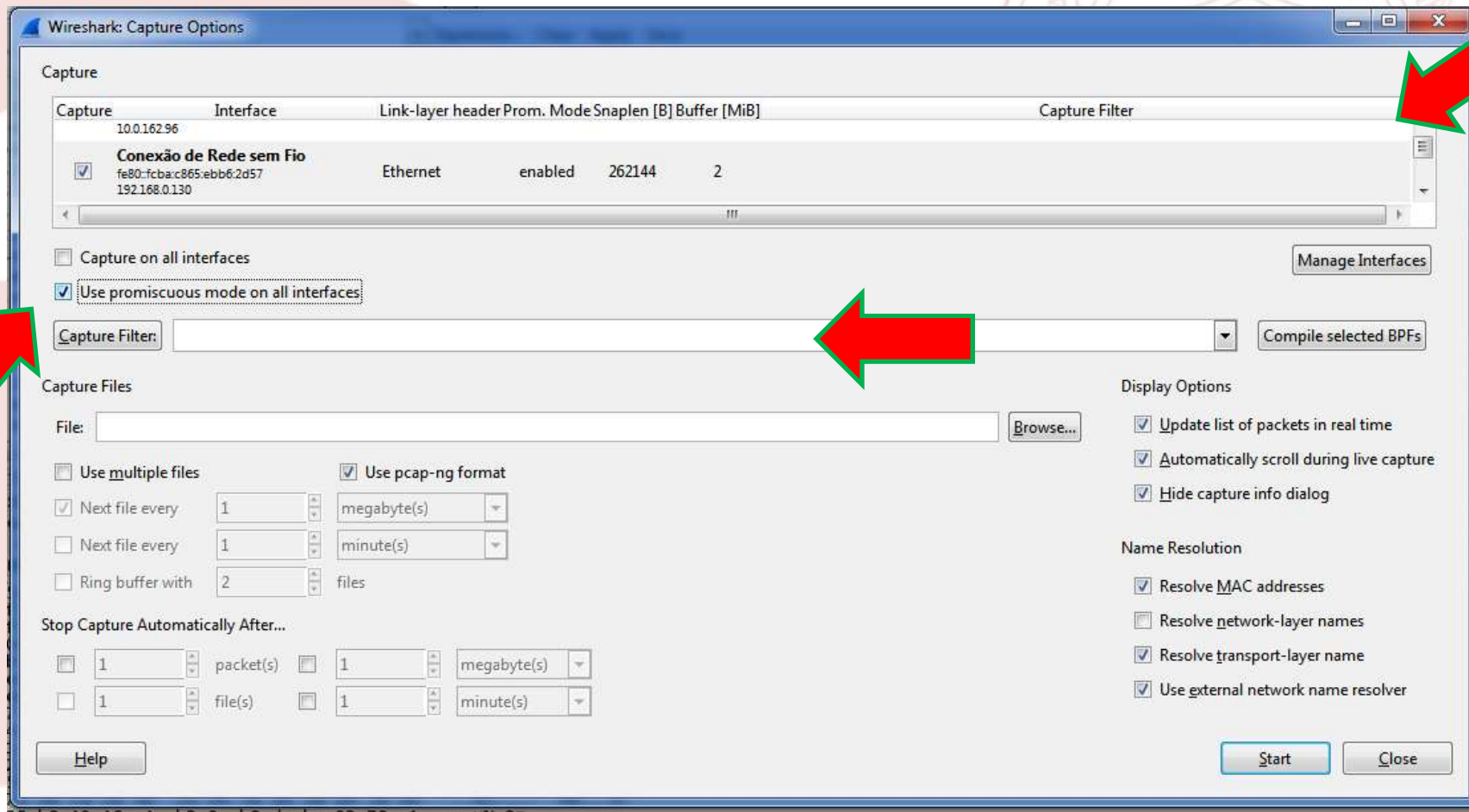
### Conhecendo o Wireshark:





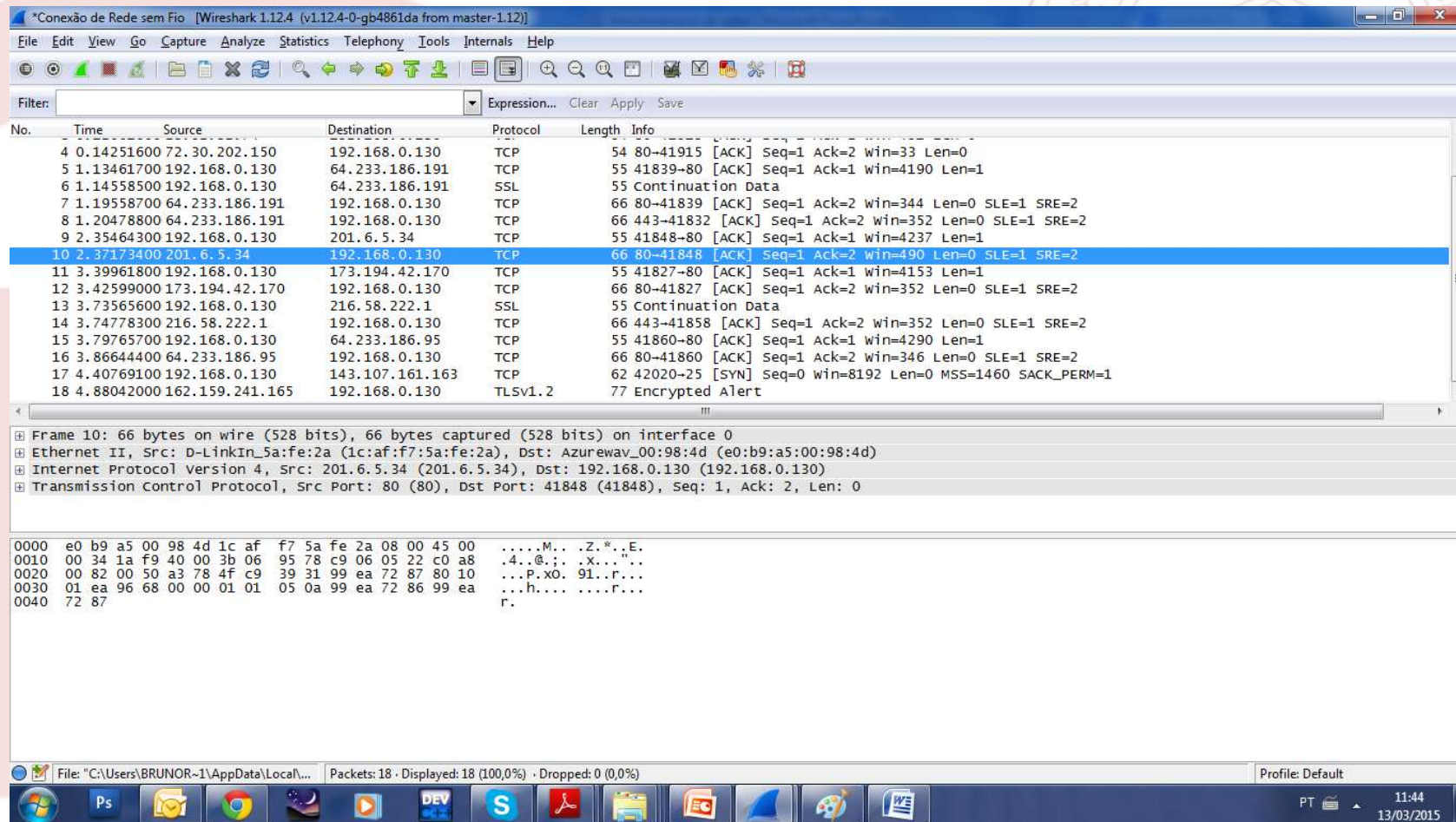
## Analise de pacotes de redes

### Conhecendo o Wireshark:



## Analise de pacotes de redes

### Conhecendo o Wireshark:



Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
4	0.14251600	72.30.202.150	192.168.0.130	TCP	54	80->41915 [ACK] Seq=1 Ack=2 win=33 Len=0
5	1.13461700	192.168.0.130	64.233.186.191	TCP	55	41839->80 [ACK] Seq=1 Ack=1 win=4190 Len=1
6	1.14558500	192.168.0.130	64.233.186.191	SSL	55	Continuation Data
7	1.19558700	64.233.186.191	192.168.0.130	TCP	66	80->41839 [ACK] Seq=1 Ack=2 win=344 Len=0 SLE=1 SRE=2
8	1.20478800	64.233.186.191	192.168.0.130	TCP	66	443->41832 [ACK] Seq=1 Ack=2 win=352 Len=0 SLE=1 SRE=2
9	2.35464300	192.168.0.130	201.6.5.34	TCP	55	41848->80 [ACK] Seq=1 Ack=1 win=4237 Len=1
10	2.37173400	201.6.5.34	192.168.0.130	TCP	66	80->41848 [ACK] Seq=1 Ack=2 win=490 Len=0 SLE=1 SRE=2
11	3.39961800	192.168.0.130	173.194.42.170	TCP	55	41827->80 [ACK] Seq=1 Ack=1 win=4153 Len=1
12	3.42599000	173.194.42.170	192.168.0.130	TCP	66	80->41827 [ACK] Seq=1 Ack=2 win=352 Len=0 SLE=1 SRE=2
13	3.73565600	192.168.0.130	216.58.222.1	SSL	55	Continuation Data
14	3.74778300	216.58.222.1	192.168.0.130	TCP	66	443->41858 [ACK] Seq=1 Ack=2 win=352 Len=0 SLE=1 SRE=2
15	3.79765700	192.168.0.130	64.233.186.95	TCP	55	41860->80 [ACK] Seq=1 Ack=1 win=4290 Len=1
16	3.86644400	64.233.186.95	192.168.0.130	TCP	66	80->41860 [ACK] Seq=1 Ack=2 win=346 Len=0 SLE=1 SRE=2
17	4.40769100	192.168.0.130	143.107.161.163	TCP	62	42020->25 [SYN] Seq=0 win=8192 Len=0 MSS=1460 SACK_PERM=1
18	4.88042000	162.159.241.165	192.168.0.130	TLSv1.2	77	Encrypted Alert

Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

Ethernet II, Src: D-LinkIn\_5a:fe:2a (1c:af:f7:5a:fe:2a), Dst: Azurewav\_00:98:4d (e0:b9:a5:00:98:4d)

Internet Protocol Version 4, Src: 201.6.5.34 (201.6.5.34), Dst: 192.168.0.130 (192.168.0.130)

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 41848 (41848), Seq: 1, Ack: 2, Len: 0

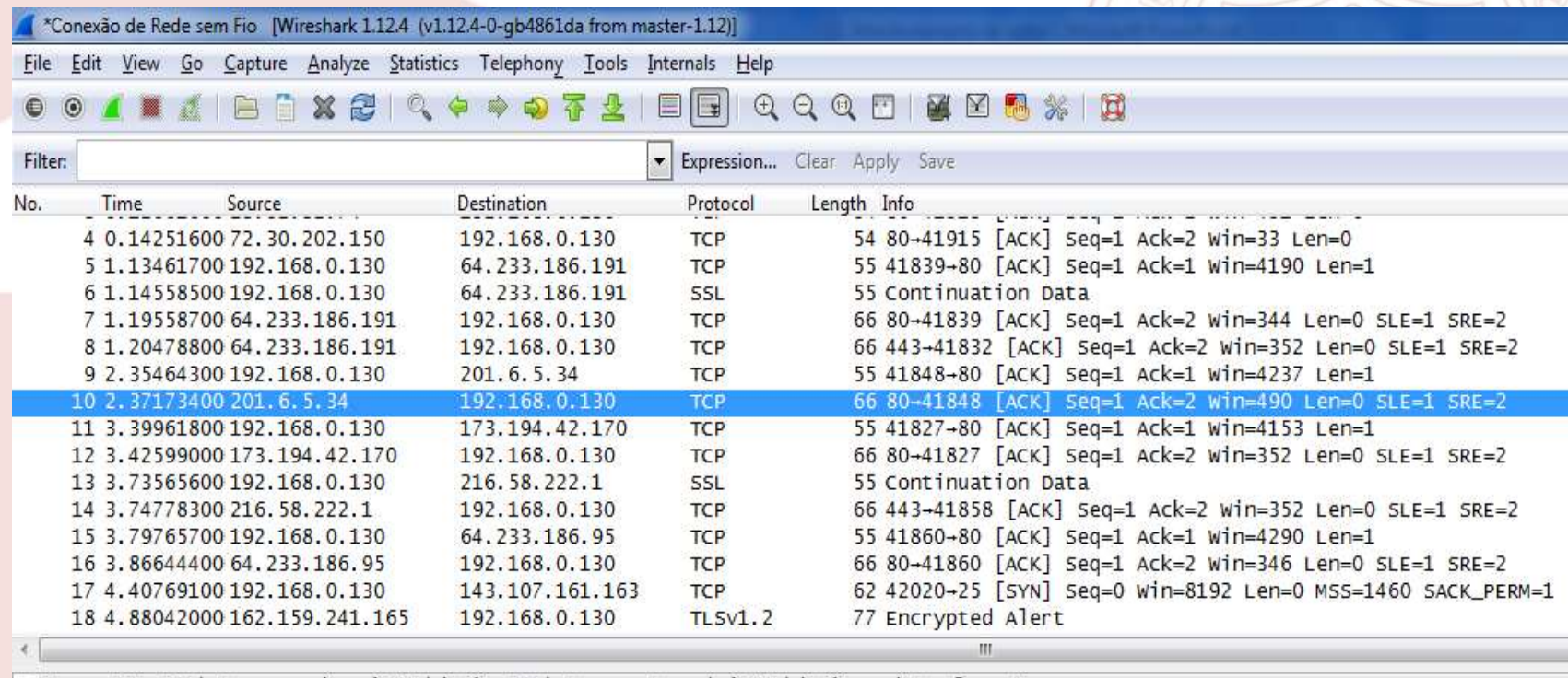
0000 e0 b9 a5 00 98 4d 1c af f7 5a fe 2a 08 00 45 00 .....M...Z.\*..E.  
0010 00 34 1a f9 40 00 3b 06 95 78 c9 06 05 22 c0 a8 .4.@.;.x..."  
0020 00 82 00 50 a3 78 4f c9 39 31 99 ea 72 87 80 10 ...P.x0. 91..r..  
0030 01 ea 96 68 00 00 01 01 05 0a 99 ea 72 86 99 ea ...h....r..  
0040 72 87 r.

File: "C:\Users\BRUNOR~1\AppData\Local\..." Packets: 18 · Displayed: 18 (100,0%) · Dropped: 0 (0,0%) Profile: Default

PT 11:44 13/03/2015

## Analise de pacotes de redes

### Conhecendo o Wireshark:

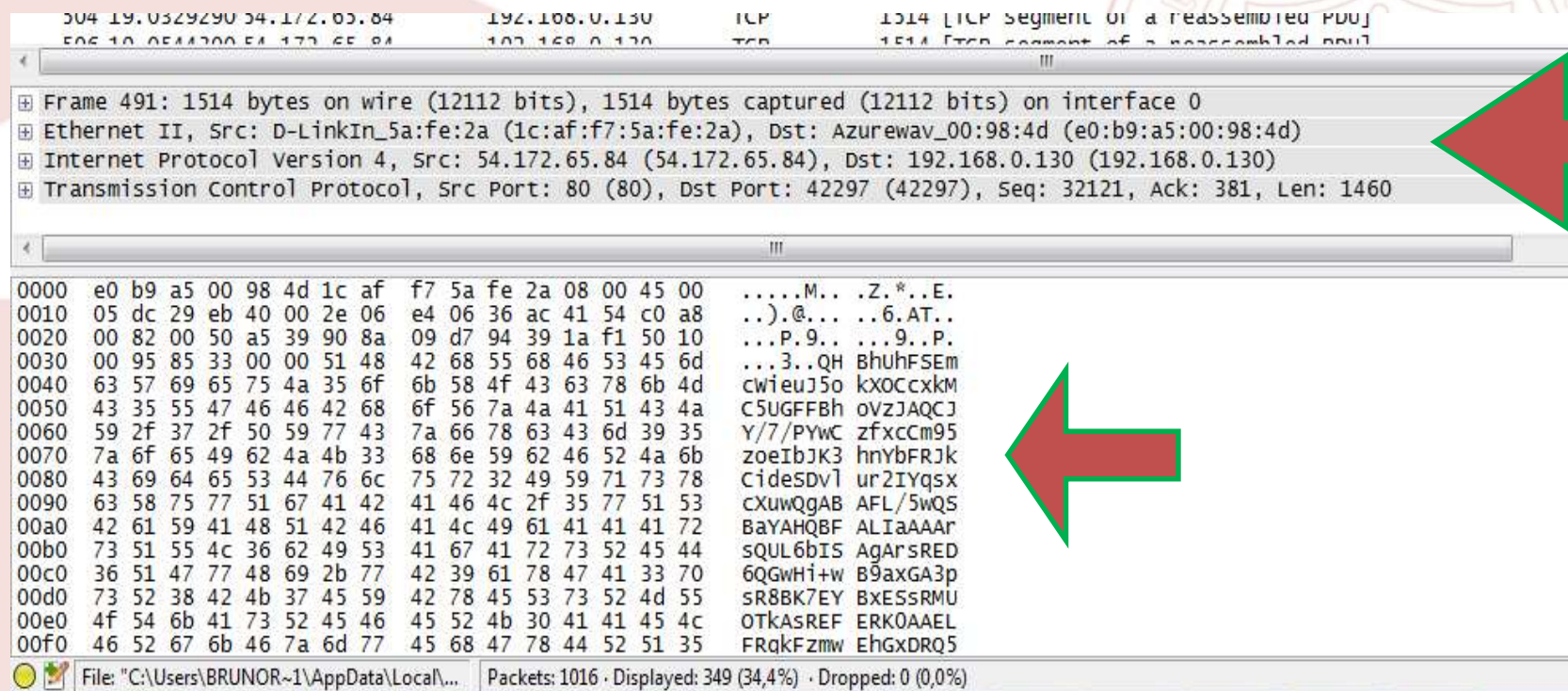


No.	Time	Source	Destination	Protocol	Length	Info
4	0.14251600	72.30.202.150	192.168.0.130	TCP	54	80->41915 [ACK] Seq=1 Ack=2 win=33 Len=0
5	1.13461700	192.168.0.130	64.233.186.191	TCP	55	41839->80 [ACK] Seq=1 Ack=1 win=4190 Len=1
6	1.14558500	192.168.0.130	64.233.186.191	SSL	55	Continuation Data
7	1.19558700	64.233.186.191	192.168.0.130	TCP	66	80->41839 [ACK] Seq=1 Ack=2 win=344 Len=0 SLE=1 SRE=2
8	1.20478800	64.233.186.191	192.168.0.130	TCP	66	443->41832 [ACK] Seq=1 Ack=2 win=352 Len=0 SLE=1 SRE=2
9	2.35464300	192.168.0.130	201.6.5.34	TCP	55	41848->80 [ACK] Seq=1 Ack=1 win=4237 Len=1
10	2.37173400	201.6.5.34	192.168.0.130	TCP	66	80->41848 [ACK] Seq=1 Ack=2 win=490 Len=0 SLE=1 SRE=2
11	3.39961800	192.168.0.130	173.194.42.170	TCP	55	41827->80 [ACK] Seq=1 Ack=1 win=4153 Len=1
12	3.42599000	173.194.42.170	192.168.0.130	TCP	66	80->41827 [ACK] Seq=1 Ack=2 win=352 Len=0 SLE=1 SRE=2
13	3.73565600	192.168.0.130	216.58.222.1	SSL	55	Continuation Data
14	3.74778300	216.58.222.1	192.168.0.130	TCP	66	443->41858 [ACK] Seq=1 Ack=2 win=352 Len=0 SLE=1 SRE=2
15	3.79765700	192.168.0.130	64.233.186.95	TCP	55	41860->80 [ACK] Seq=1 Ack=1 win=4290 Len=1
16	3.86644400	64.233.186.95	192.168.0.130	TCP	66	80->41860 [ACK] Seq=1 Ack=2 win=346 Len=0 SLE=1 SRE=2
17	4.40769100	192.168.0.130	143.107.161.163	TCP	62	42020->25 [SYN] Seq=0 win=8192 Len=0 MSS=1460 SACK_PERM=1
18	4.88042000	162.159.241.165	192.168.0.130	TLSv1.2	77	Encrypted Alert



## Analise de pacotes de redes

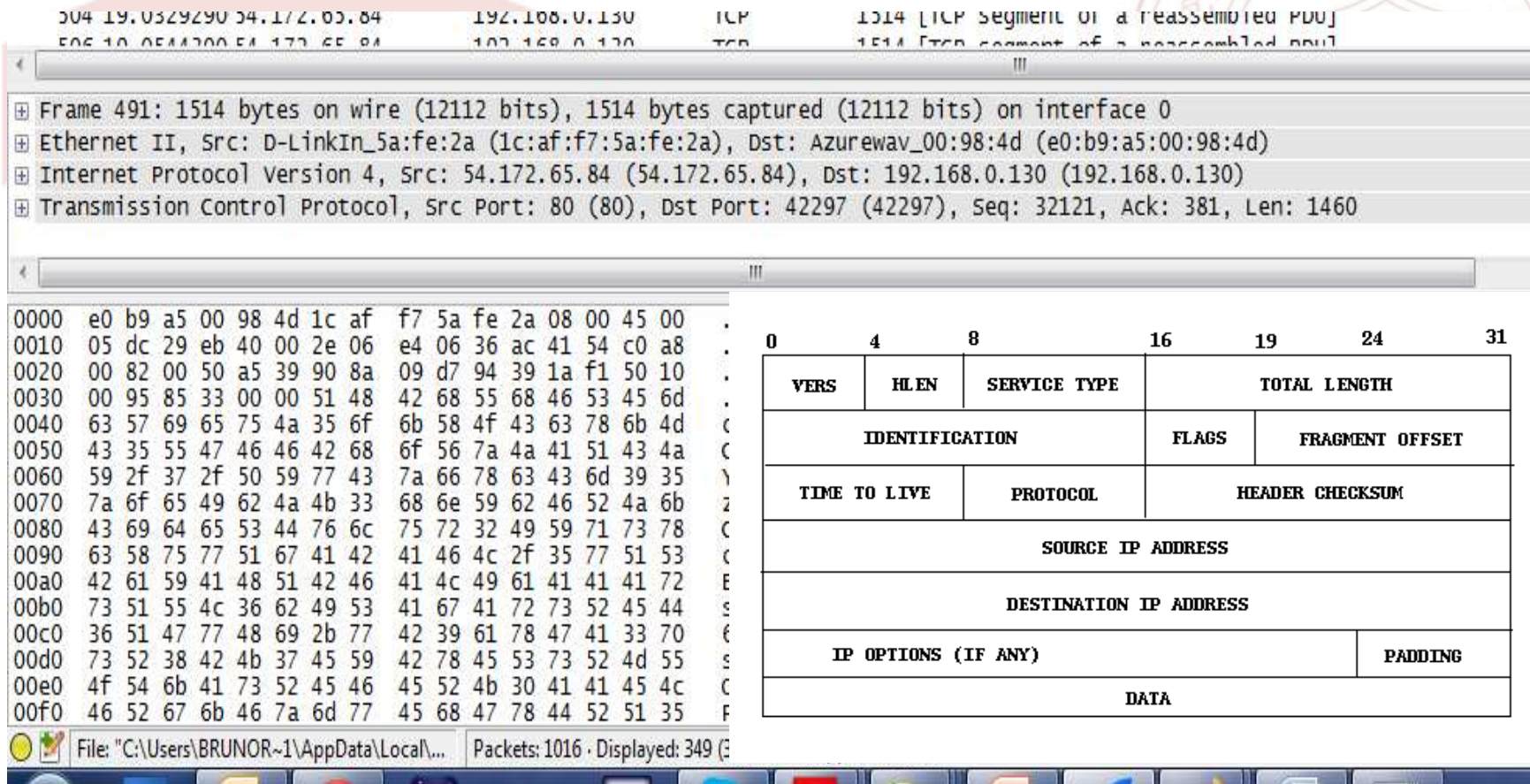
### Conhecendo o Wireshark:





## Analise de pacotes de redes

### Conhecendo o Wireshark:



The image shows the Wireshark network protocol analyzer interface. The top pane displays a list of captured packets, with packet 491 selected. The middle pane shows the details of the selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The bottom pane shows the raw data in hexadecimal and ASCII format.

Packet 491 details:

- Frame 491: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
- Ethernet II, Src: D-LinkIn\_5a:fe:2a (1c:af:f7:5a:fe:2a), Dst: Azurewav\_00:98:4d (e0:b9:a5:00:98:4d)
- Internet Protocol Version 4, Src: 54.172.65.84 (54.172.65.84), Dst: 192.168.0.130 (192.168.0.130)
- Transmission Control Protocol, Src Port: 80 (80), Dst Port: 42297 (42297), Seq: 32121, Ack: 381, Len: 1460

Raw data (hex and ASCII):

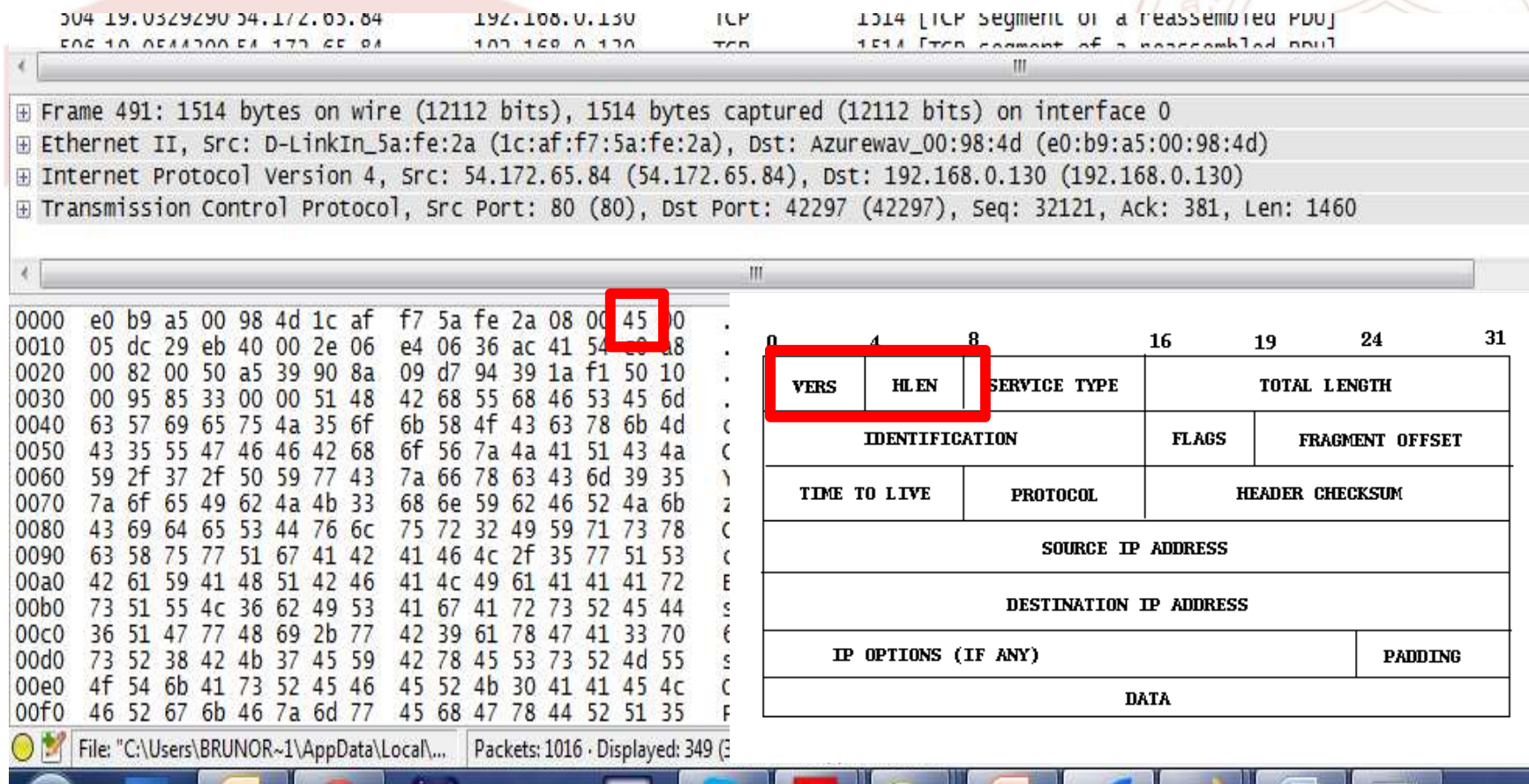
```
0000 e0 b9 a5 00 98 4d 1c af f7 5a fe 2a 08 00 45 00 .
0010 05 dc 29 eb 40 00 2e 06 e4 06 36 ac 41 54 c0 a8 .
0020 00 82 00 50 a5 39 90 8a 09 d7 94 39 1a f1 50 10 .
0030 00 95 85 33 00 00 51 48 42 68 55 68 46 53 45 6d .
0040 63 57 69 65 75 4a 35 6f 6b 58 4f 43 63 78 6b 4d c
0050 43 35 55 47 46 46 42 68 6f 56 7a 4a 41 51 43 4a c
0060 59 2f 37 2f 50 59 77 43 7a 66 78 63 43 6d 39 35 y
0070 7a 6f 65 49 62 4a 4b 33 68 6e 59 62 46 52 4a 6b z
0080 43 69 64 65 53 44 76 6c 75 72 32 49 59 71 73 78 C
0090 63 58 75 77 51 67 41 42 41 46 4c 2f 35 77 51 53 c
00a0 42 61 59 41 48 51 42 46 41 4c 49 61 41 41 41 72 E
00b0 73 51 55 4c 36 62 49 53 41 67 41 72 73 52 45 44 s
00c0 36 51 47 77 48 69 2b 77 42 39 61 78 47 41 33 70 e
00d0 73 52 38 42 4b 37 45 59 42 78 45 53 73 52 4d 55 s
00e0 4f 54 6b 41 73 52 45 46 45 52 4b 30 41 41 45 4c C
00f0 46 52 67 6b 46 7a 6d 77 45 68 47 78 44 52 51 35 f
```

0	4	8	16	19	24	31	
VERS		HLEN		SERVICE TYPE		TOTAL LENGTH	
IDENTIFICATION				FLAGS		FRAGMENT OFFSET	
TIME TO LIVE				PROTOCOL		HEADER CHECKSUM	
SOURCE IP ADDRESS							
DESTINATION IP ADDRESS							
IP OPTIONS (IF ANY)						PADDING	
DATA							

File: "C:\Users\BRUNOR~1\AppData\Local\... Packets: 1016 · Displayed: 349

## Analise de pacotes de redes

### Conhecendo o Wireshark:



Frame 491: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

Ethernet II, Src: D-LinkIn\_5a:fe:2a (1c:af:f7:5a:fe:2a), Dst: Azurewav\_00:98:4d (e0:b9:a5:00:98:4d)

Internet Protocol Version 4, Src: 54.172.65.84 (54.172.65.84), Dst: 192.168.0.130 (192.168.0.130)

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 42297 (42297), Seq: 32121, Ack: 381, Len: 1460

0000 e0 b9 a5 00 98 4d 1c af f7 5a fe 2a 08 00 45 00 .  
0010 05 dc 29 eb 40 00 2e 06 e4 06 36 ac 41 54 10 48 .  
0020 00 82 00 50 a5 39 90 8a 09 d7 94 39 1a f1 50 10 .  
0030 00 95 85 33 00 00 51 48 42 68 55 68 46 53 45 6d .  
0040 63 57 69 65 75 4a 35 6f 6b 58 4f 43 63 78 6b 4d .  
0050 43 35 55 47 46 46 42 68 6f 56 7a 4a 41 51 43 4a .  
0060 59 2f 37 2f 50 59 77 43 7a 66 78 63 43 6d 39 35 .  
0070 7a 6f 65 49 62 4a 4b 33 68 6e 59 62 46 52 4a 6b .  
0080 43 69 64 65 53 44 76 6c 75 72 32 49 59 71 73 78 .  
0090 63 58 75 77 51 67 41 42 41 46 4c 2f 35 77 51 53 .  
00a0 42 61 59 41 48 51 42 46 41 4c 49 61 41 41 41 72 .  
00b0 73 51 55 4c 36 62 49 53 41 67 41 72 73 52 45 44 .  
00c0 36 51 47 77 48 69 2b 77 42 39 61 78 47 41 33 70 .  
00d0 73 52 38 42 4b 37 45 59 42 78 45 53 73 52 4d 55 .  
00e0 4f 54 6b 41 73 52 45 46 45 52 4b 30 41 41 45 4c .  
00f0 46 52 67 6b 46 7a 6d 77 45 68 47 78 44 52 51 35 .

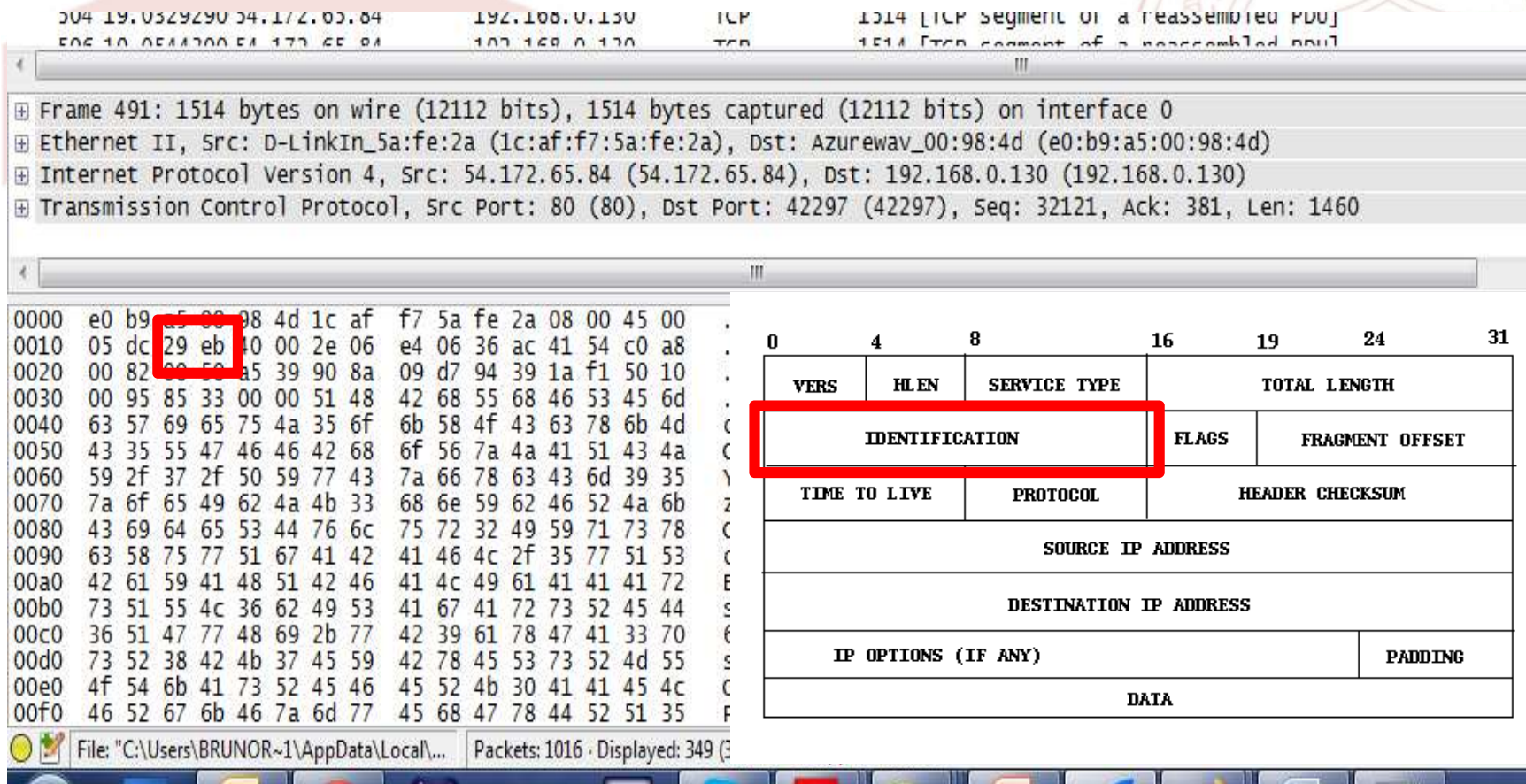
0	4	8	16	19	24	31	
VERS		HLEN		SERVICE TYPE		TOTAL LENGTH	
IDENTIFICATION				FLAGS		FRAGMENT OFFSET	
TIME TO LIVE		PROTOCOL		HEADER CHECKSUM			
SOURCE IP ADDRESS							
DESTINATION IP ADDRESS							
IP OPTIONS (IF ANY)					PADDING		
DATA							

File: "C:\Users\BRUNOR~1\AppData\Local\... Packets: 1016 · Displayed: 349



## Analise de pacotes de redes

### Conhecendo o Wireshark:



Frame 491: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

Ethernet II, Src: D-LinkIn\_5a:fe:2a (1c:af:f7:5a:fe:2a), Dst: Azurewav\_00:98:4d (e0:b9:a5:00:98:4d)

Internet Protocol Version 4, Src: 54.172.65.84 (54.172.65.84), Dst: 192.168.0.130 (192.168.0.130)

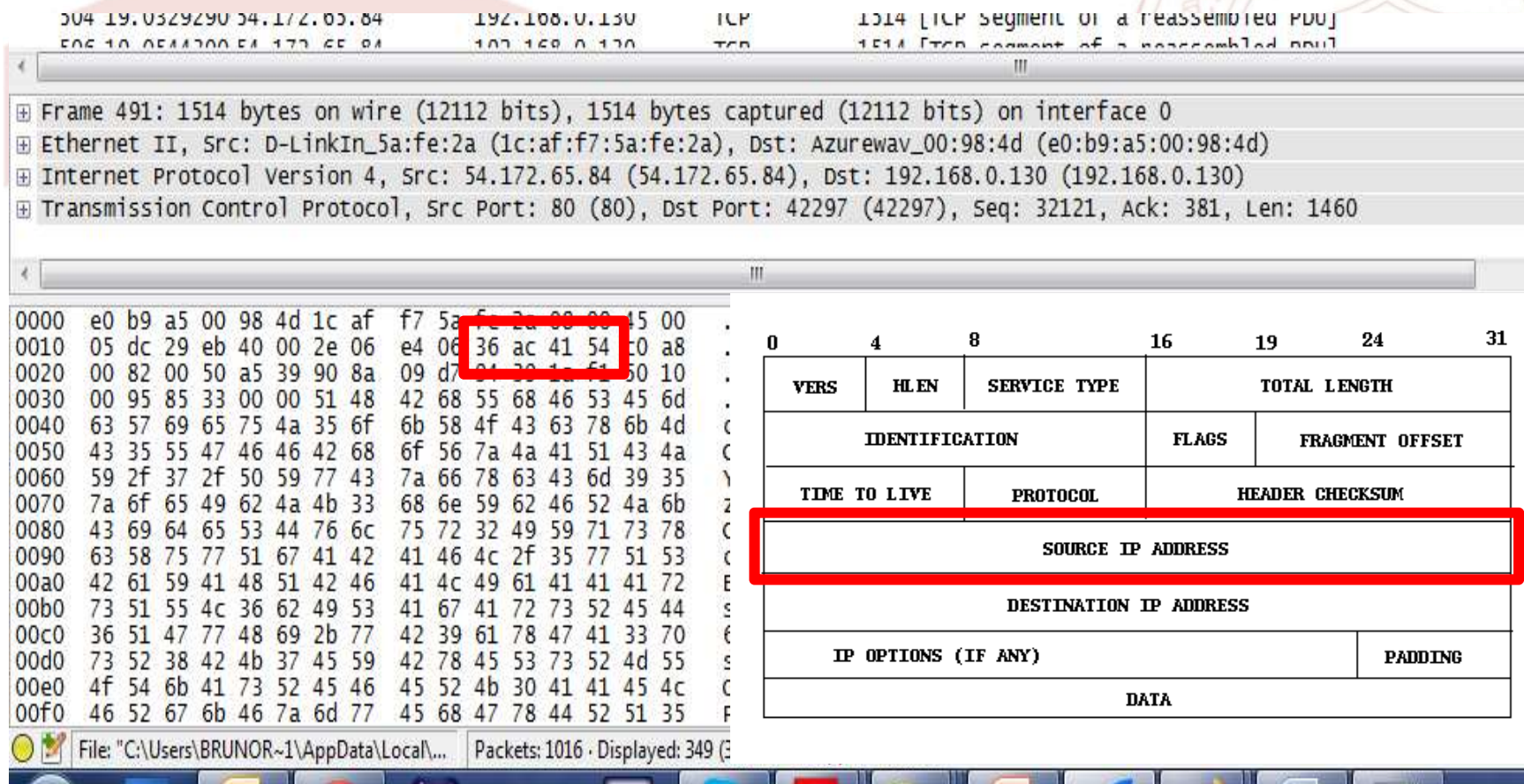
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 42297 (42297), Seq: 32121, Ack: 381, Len: 1460

0	4	8	16	19	24	31	
VERS		HLEN		SERVICE TYPE		TOTAL LENGTH	
IDENTIFICATION				FLAGS		FRAGMENT OFFSET	
TIME TO LIVE				PROTOCOL		HEADER CHECKSUM	
SOURCE IP ADDRESS							
DESTINATION IP ADDRESS							
IP OPTIONS (IF ANY)						PADDING	
DATA							

File: "C:\Users\BRUNOR~1\AppData\Local\..." Packets: 1016 · Displayed: 349

## Analise de pacotes de redes

### Conhecendo o Wireshark:



304 19.0329290 54.172.65.84 192.168.0.130 TCP 1514 [TCP segment of a reassembled PDU]

506 10.0511200 54.172.65.84 192.168.0.130 TCP 1514 [TCP segment of a reassembled PDU]

Frame 491: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

Ethernet II, Src: D-LinkIn\_5a:fe:2a (1c:af:f7:5a:fe:2a), Dst: Azurewav\_00:98:4d (e0:b9:a5:00:98:4d)

Internet Protocol Version 4, Src: 54.172.65.84 (54.172.65.84), Dst: 192.168.0.130 (192.168.0.130)

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 42297 (42297), Seq: 32121, Ack: 381, Len: 1460

0000 e0 b9 a5 00 98 4d 1c af f7 5a fe 2a 00 00 45 00 .  
0010 05 dc 29 eb 40 00 2e 06 e4 06 36 ac 41 54 e0 a8 .  
0020 00 82 00 50 a5 39 90 8a 09 d7 81 30 12 51 50 10 .  
0030 00 95 85 33 00 00 51 48 42 68 55 68 46 53 45 6d .  
0040 63 57 69 65 75 4a 35 6f 6b 58 4f 43 63 78 6b 4d .  
0050 43 35 55 47 46 46 42 68 6f 56 7a 4a 41 51 43 4a .  
0060 59 2f 37 2f 50 59 77 43 7a 66 78 63 43 6d 39 35 .  
0070 7a 6f 65 49 62 4a 4b 33 68 6e 59 62 46 52 4a 6b .  
0080 43 69 64 65 53 44 76 6c 75 72 32 49 59 71 73 78 .  
0090 63 58 75 77 51 67 41 42 41 46 4c 2f 35 77 51 53 .  
00a0 42 61 59 41 48 51 42 46 41 4c 49 61 41 41 41 72 .  
00b0 73 51 55 4c 36 62 49 53 41 67 41 72 73 52 45 44 .  
00c0 36 51 47 77 48 69 2b 77 42 39 61 78 47 41 33 70 .  
00d0 73 52 38 42 4b 37 45 59 42 78 45 53 73 52 4d 55 .  
00e0 4f 54 6b 41 73 52 45 46 45 52 4b 30 41 41 45 4c .  
00f0 46 52 67 6b 46 7a 6d 77 45 68 47 78 44 52 51 35 .

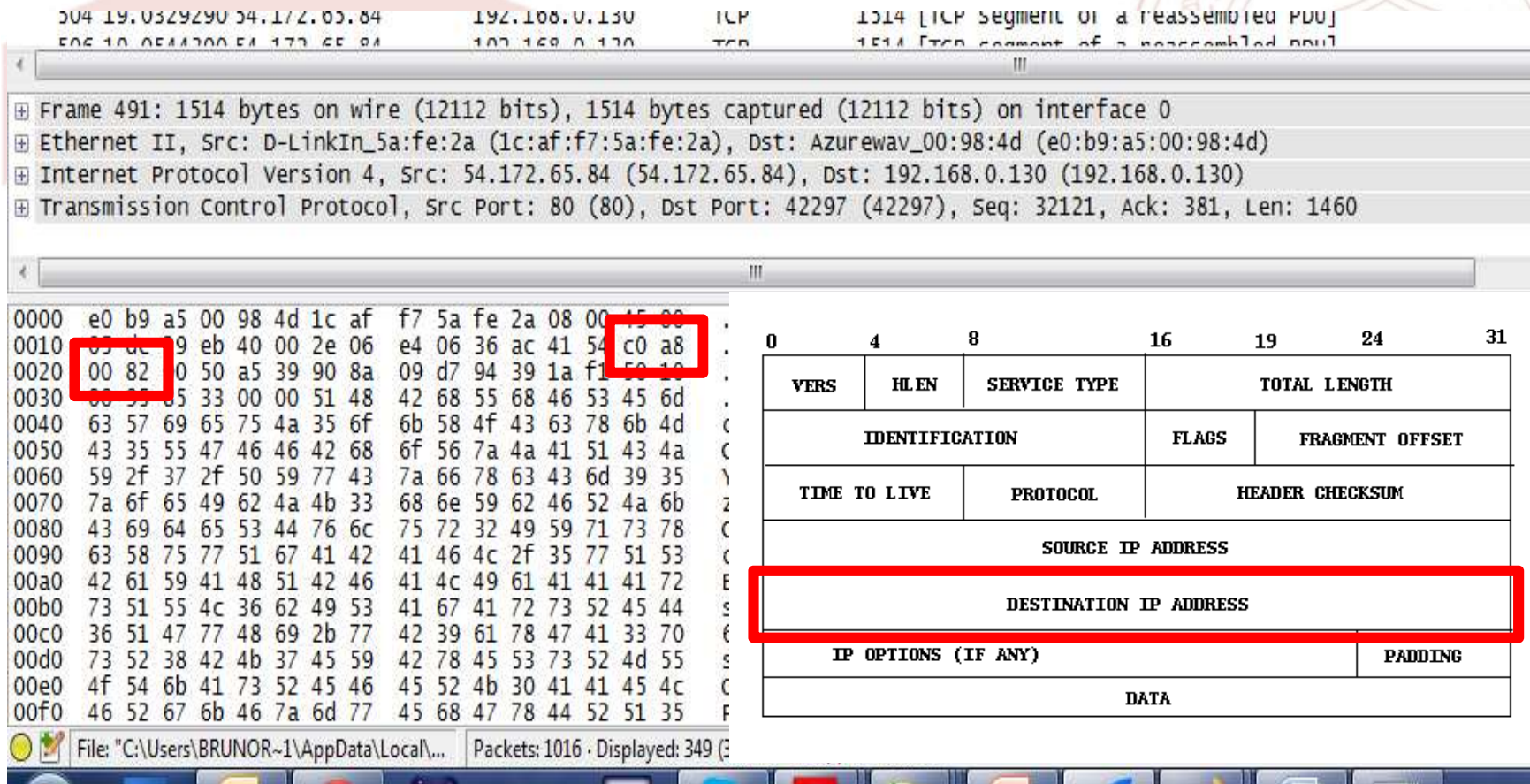
0	4	8	16	19	24	31	
VERS		HLEN		SERVICE TYPE		TOTAL LENGTH	
IDENTIFICATION				FLAGS		FRAGMENT OFFSET	
TIME TO LIVE		PROTOCOL		HEADER CHECKSUM			
SOURCE IP ADDRESS							
DESTINATION IP ADDRESS							
IP OPTIONS (IF ANY)						PADDING	
DATA							

File: "C:\Users\BRUNOR~1\AppData\Local\... Packets: 1016 · Displayed: 349



## Analise de pacotes de redes

### Conhecendo o Wireshark:



304 19.0329290 54.172.65.84 192.168.0.130 TCP 1514 [TCP segment of a reassembled PDU]

506 10.0511200 54.172.65.84 192.168.0.130 TCP 1514 [TCP segment of a reassembled PDU]

Frame 491: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

Ethernet II, Src: D-LinkIn\_5a:fe:2a (1c:af:f7:5a:fe:2a), Dst: Azurewav\_00:98:4d (e0:b9:a5:00:98:4d)

Internet Protocol Version 4, Src: 54.172.65.84 (54.172.65.84), Dst: 192.168.0.130 (192.168.0.130)

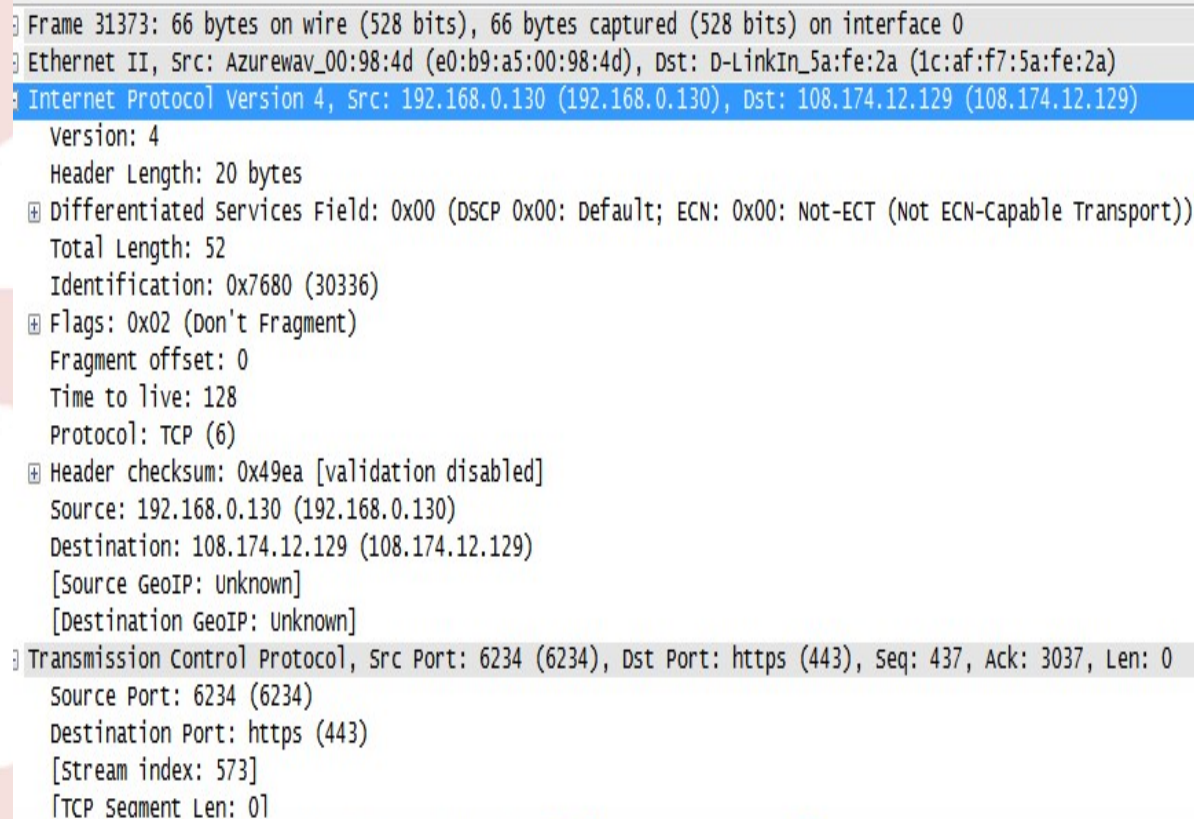
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 42297 (42297), Seq: 32121, Ack: 381, Len: 1460

0	4	8	16	19	24	31
VERS	HLLEN	SERVICE TYPE	TOTAL LENGTH			
IDENTIFICATION				FLAGS	FRAGMENT OFFSET	
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM			
SOURCE IP ADDRESS						
DESTINATION IP ADDRESS						
IP OPTIONS (IF ANY)					PADDING	
DATA						

File: "C:\Users\BRUNOR~1\AppData\Local\... Packets: 1016 · Displayed: 349

## Analise de pacotes de redes

### Conhecendo o Wireshark:



Frame 31373: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

Ethernet II, Src: Azurewav\_00:98:4d (e0:b9:a5:00:98:4d), Dst: D-LinkIn\_5a:fe:2a (1c:af:f7:5a:fe:2a)

Internet Protocol Version 4, Src: 192.168.0.130 (192.168.0.130), Dst: 108.174.12.129 (108.174.12.129)

Version: 4  
Header Length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

Total Length: 52  
Identification: 0x7680 (30336)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 128  
Protocol: TCP (6)

Header checksum: 0x49ea [validation disabled]  
Source: 192.168.0.130 (192.168.0.130)  
Destination: 108.174.12.129 (108.174.12.129)  
[Source GeoIP: Unknown]  
[Destination GeoIP: Unknown]

Transmission Control Protocol, Src Port: 6234 (6234), Dst Port: https (443), Seq: 437, Ack: 3037, Len: 0

Source Port: 6234 (6234)  
Destination Port: https (443)  
[Stream index: 573]  
[TCP segment Len: 0]

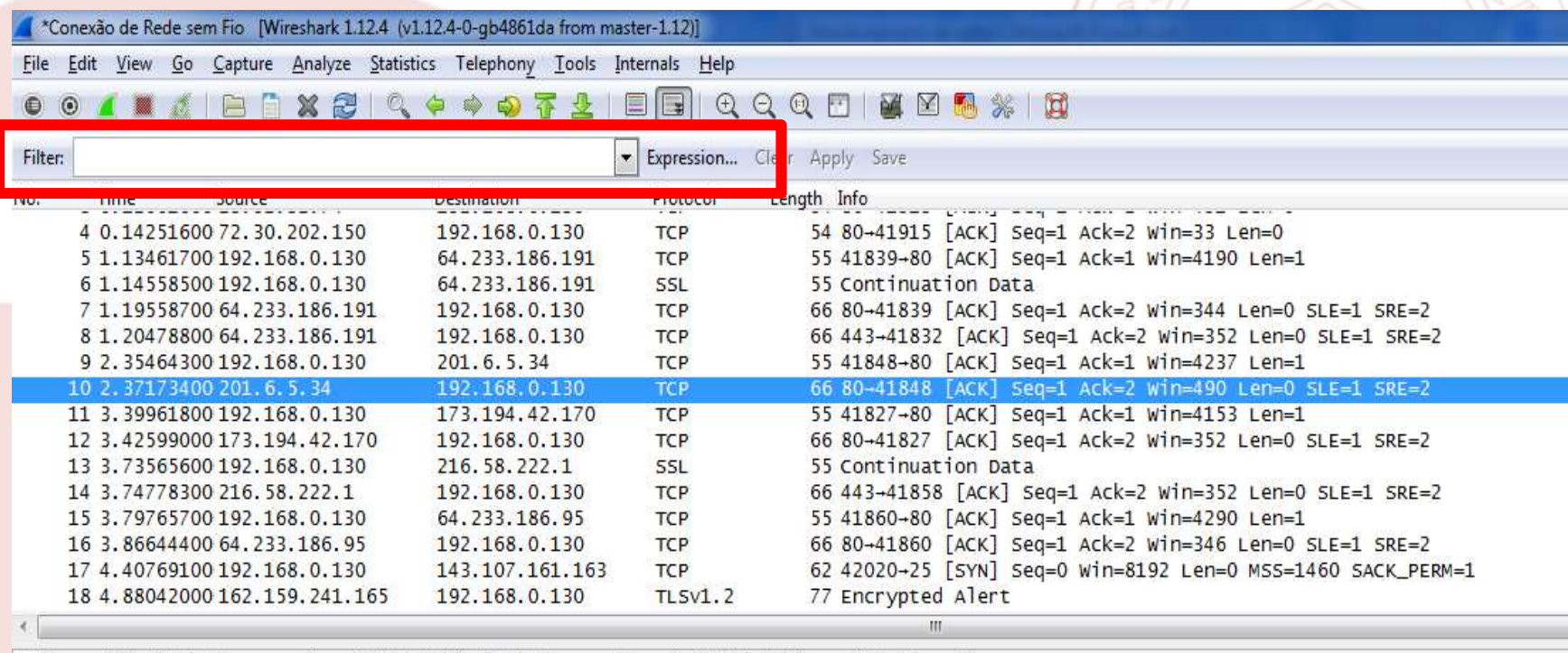
0000 1c af f7 5a fe 2a e0 b9 a5 00 98 4d 08 00 45 00 ...Z.\*.. ..M..E.  
0010 00 34 75 00 10 00 00 00 10 00 00 00 00 00 00 00 ...4.u.......

# Filtrando pacotes no Wireshark



## Analise de pacotes de redes

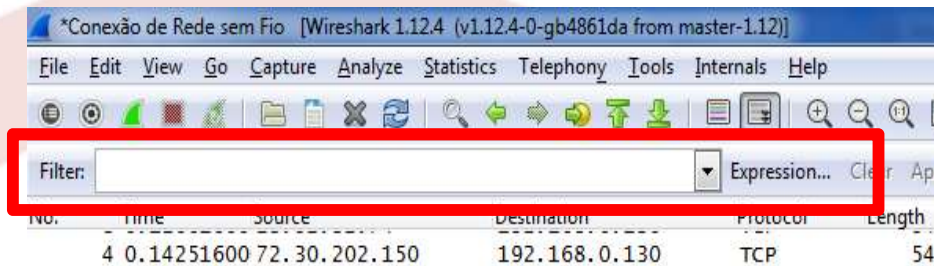
### Conhecendo o Wireshark:





## Analise de pacotes de redes

### Conhecendo o Wireshark:



### Monitorar pacotes com um IP específico:

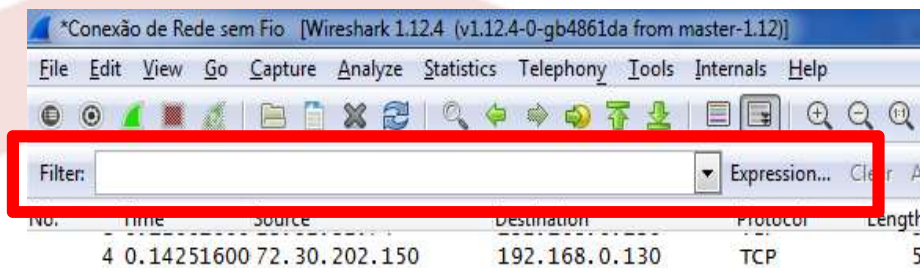
`ip.addr == 10.168.0.130`

`ip.src == 10.168.0.130`

`ip.dst == 10.10.10.10`

## Analise de pacotes de redes

### Conhecendo o Wireshark:



### Monitorar pacotes um Protocolo específico:

icmp, udp, tcp, dns

### Monitorar pacotes com um IP específico:

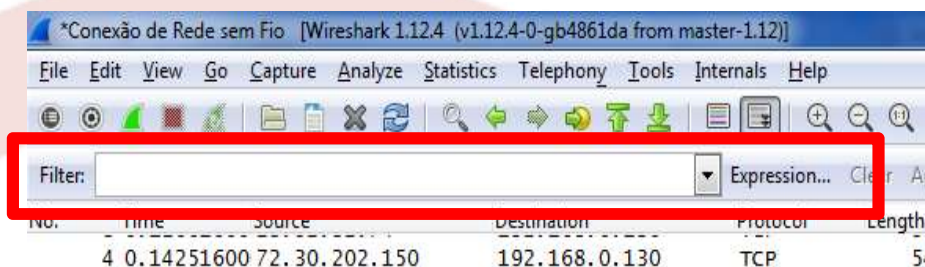
ip.addr == 10.168.0.130

ip.src == 10.168.0.130

ip.dst == 10.10.10.10

## Analise de pacotes de redes

### Conhecendo o Wireshark:



### Monitorar pacotes com um IP específico:

`ip.addr == 10.168.0.130`

`ip.src == 10.168.0.130`

`ip.dst == 10.10.10.10`

### Monitorar pacotes um Protocolo específico:

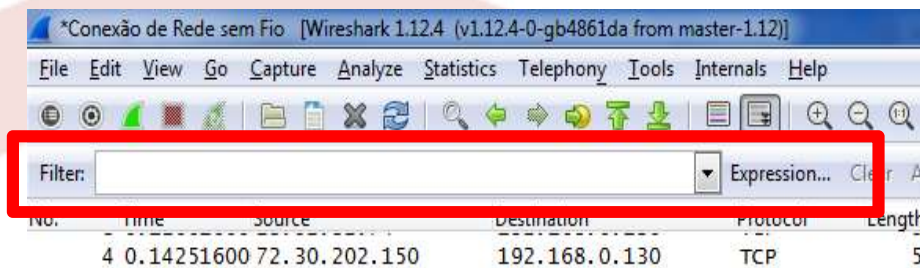
`icmp, udp, tcp, dns`

### Monitorar pacotes uma Porta específica:

`tcp.port == 20 (FTP)`

## Analise de pacotes de redes

### Conhecendo o Wireshark:



### Monitorar pacotes com um IP específico:

`ip.addr == 10.168.0.130`

`ip.src == 10.168.0.130`

`ip.dst == 10.10.10.10`

### Monitorar pacotes um Protocolo específico:

`icmp, udp, tcp, dns`

### Monitorar pacotes uma Porta específica:

`tcp.port == 20`

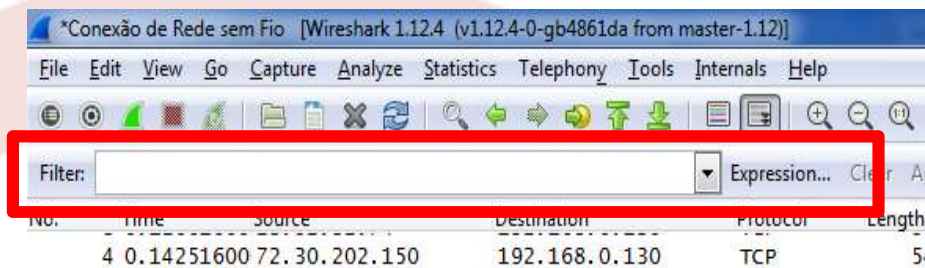
### Monitorar pacotes TCP com Flag SYN:

`tcp.flags.syn==1`



## Analise de pacotes de redes

### Conhecendo o Wireshark:



### Monitorar pacotes com um IP específico:

**Monitorar pacotes com um IP específico:**

`ip.addr == 10.168.0.130`

`ip.src == 10.168.0.130`

`ip.dst == 10.10.10.10`

### Monitorar pacotes um Protocolo específico:

`icmp, udp, tcp, dns`

### Monitorar pacotes uma Porta específica:

`tcp.port == 20`

### Monitorar pacotes TCP com Flag SYN:

`tcp.flags.syn==1`

### Monitorar pacotes com mais de um parâmetro:

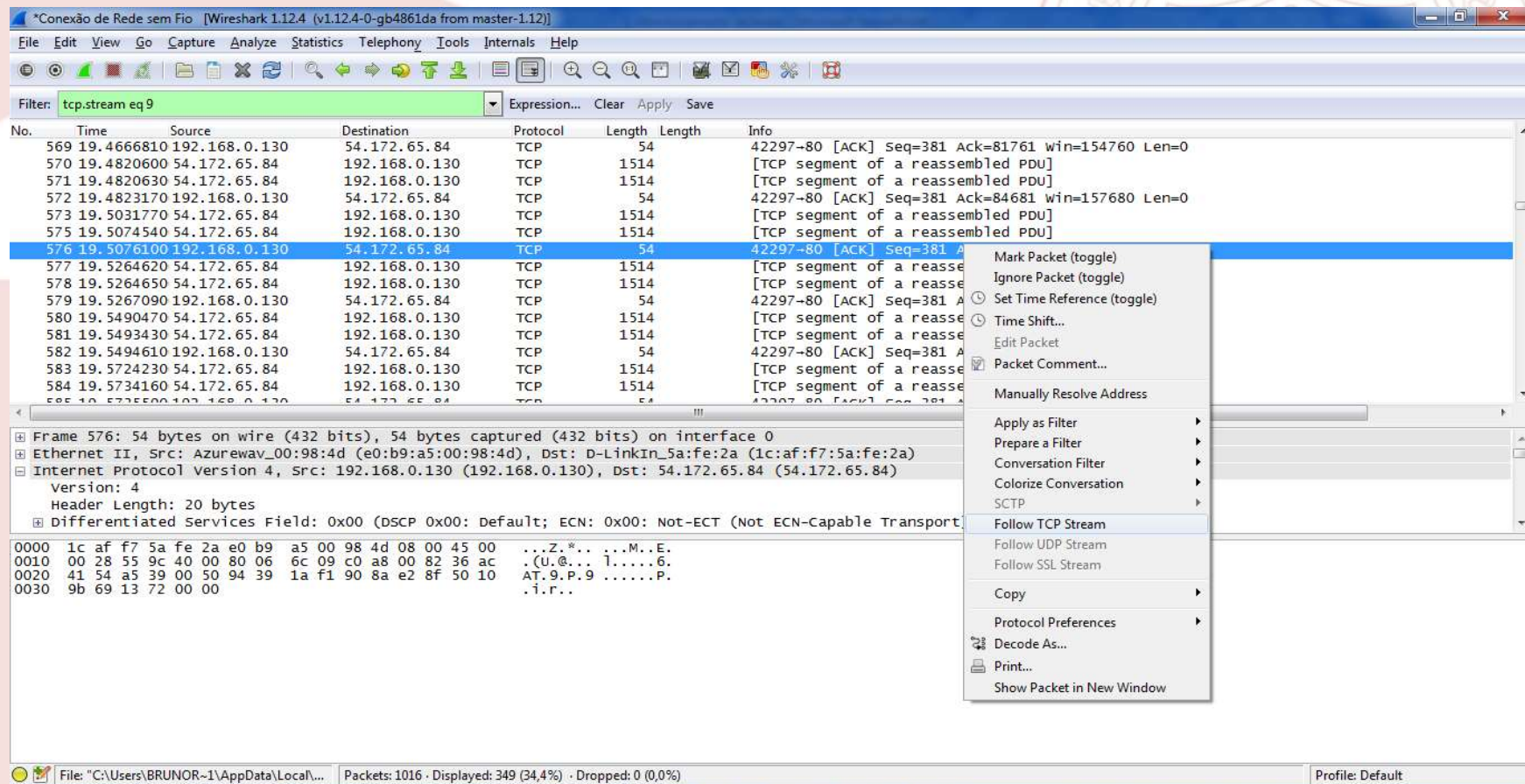
`ip.src == 192.168.0.130 && tcp.dstport==80`



**Reconstruído a fragmentação dos pacotes**

## Analise de pacotes de redes

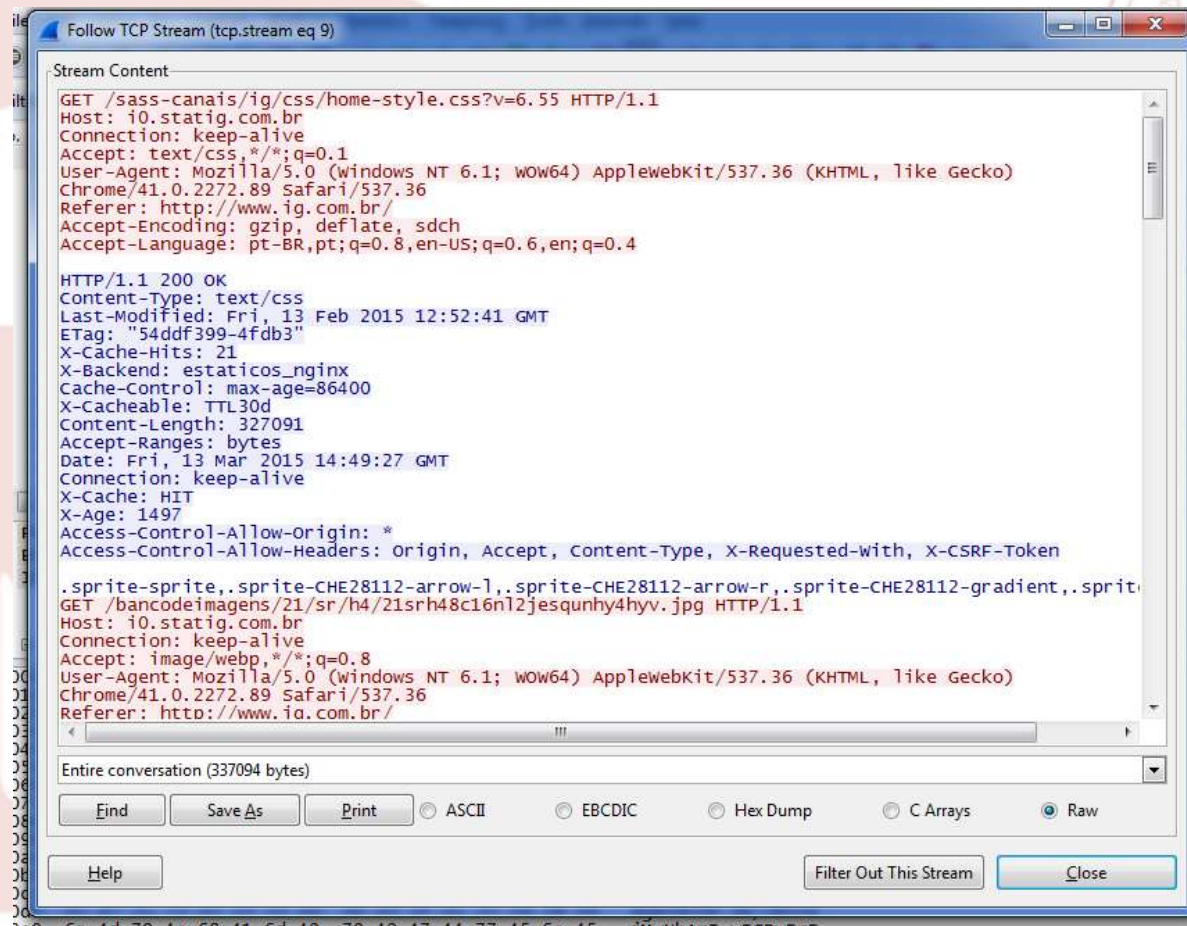
### Conhecendo o Wireshark:





## Analise de pacotes de redes

### Conhecendo o Wireshark:

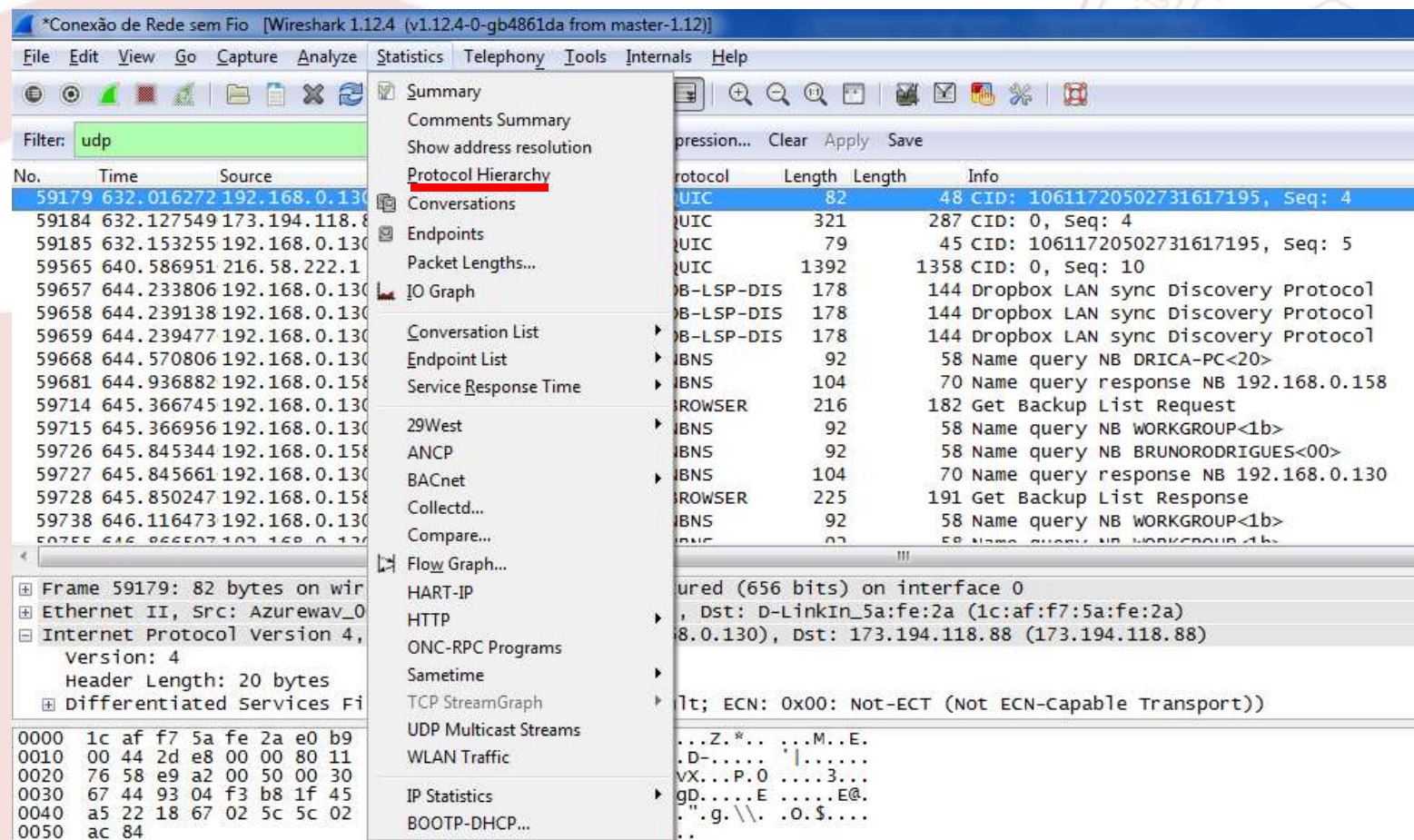




# Estatísticas usando WireShark

## Analise de pacotes de redes

### Conhecendo o Wireshark:



The image shows the Wireshark network protocol analyzer interface. The main window displays a list of captured packets. The first packet, No. 59179, is selected, showing its details in the left pane and its raw data in the bottom pane. The packet is an Ethernet II frame from Source 192.168.0.130 to Destination 173.194.118.88, encapsulating an Internet Protocol Version 4 packet from 192.168.0.130 to 173.194.118.88, which in turn encapsulates a User Datagram Protocol (UDP) packet from port 58 to port 58.

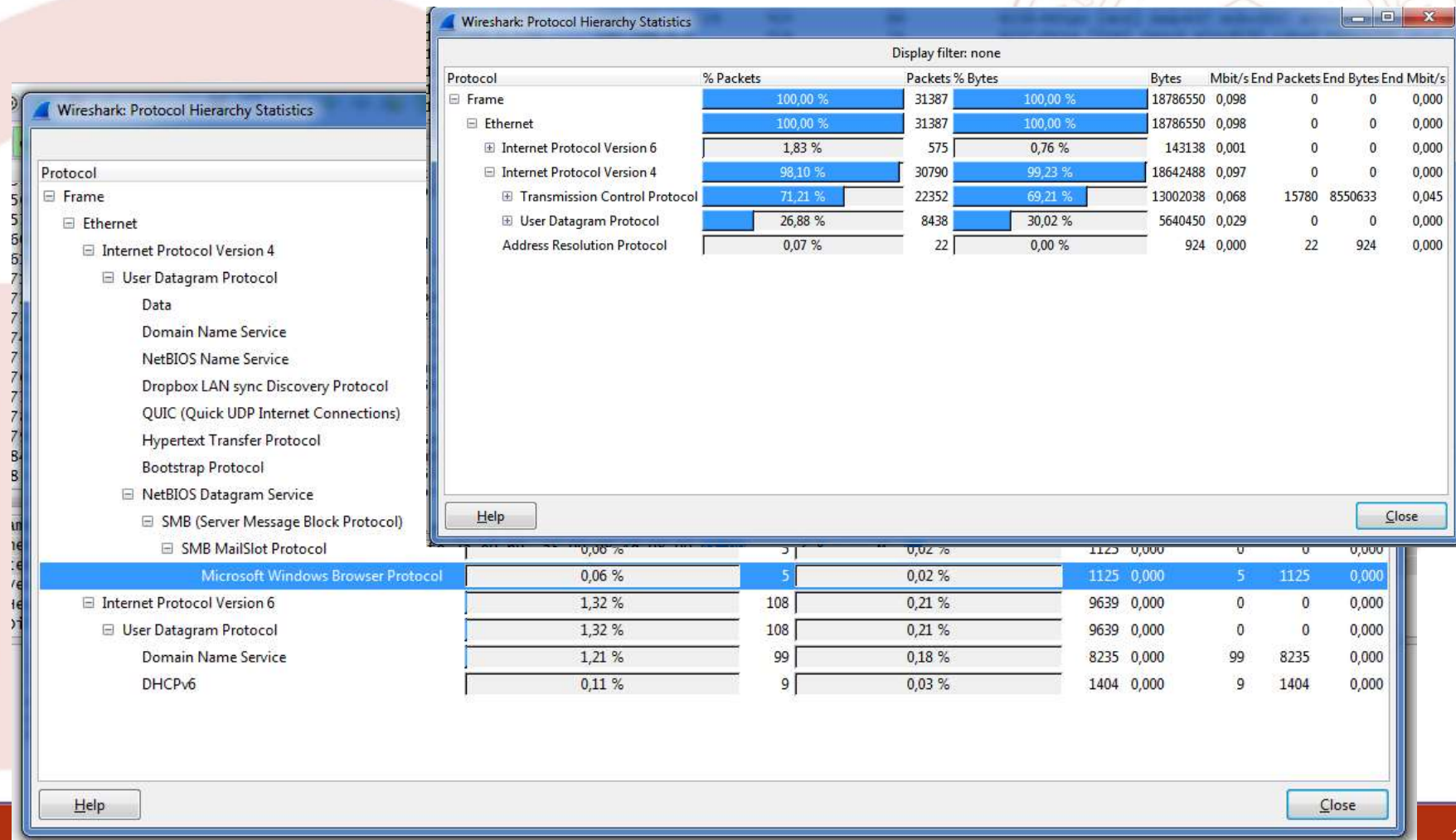
The left pane shows the packet list and the details of the selected packet. The details pane shows the following information:

- Frame 59179: 82 bytes on wire (656 bits) captured (656 bits) on interface 0
- Ethernet II, Src: Azurewav\_08:00:27:11:11:11, Dst: D-LinkIn\_5a:fe:2a:1c:af:f7:5a:fe:2a
- Internet Protocol Version 4, Src: 192.168.0.130, Dst: 173.194.118.88
- User Datagram Protocol, Src Port: 58, Dst Port: 58

The bottom pane shows the raw data of the packet in hexadecimal and ASCII format.

## Analise de pacotes de redes

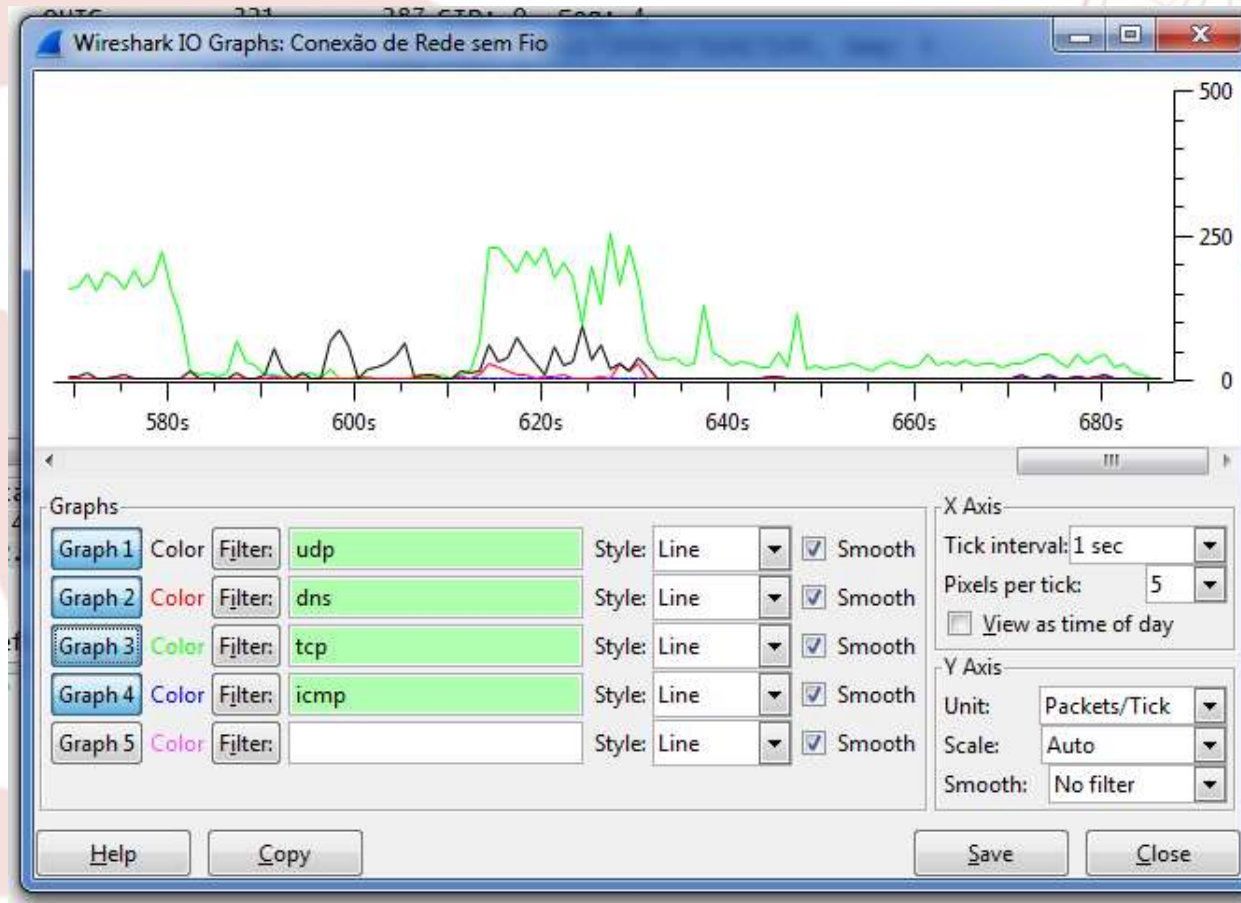
### Conhecendo o Wireshark:





## Analise de pacotes de redes

### Conhecendo o Wireshark:





**CHRIS SANDERS** – *Practical Packet Analysis using wireshark to solve real-world network problems* – 2ª edição - No Starch press, 2011

**KUROSE, J. F.; ROSS, K. W.** *Redes de Computadores e a Internet: uma nova abordagem*. Tradução de Arlete Simille Marques. São Paulo: Addison Wesley, 2003.

**STALLINGS, W.** *Criptografia e Segurança de Redes: princípios e práticas*. 4.ed. São Paulo: Pearson Prentice Hall, 2008.

**TANENBAUM, A. S.** *Redes de Computadores* . 4ª Ed., Editora Campus (Elsevier), 2003.

**Apostilas de segurança – CERT Br** - <http://cartilha.cert.br/>

**Internet Assigned Numbers Authority** - <http://www.iana.org/>



Obrigado!