Introduction à Wireshark

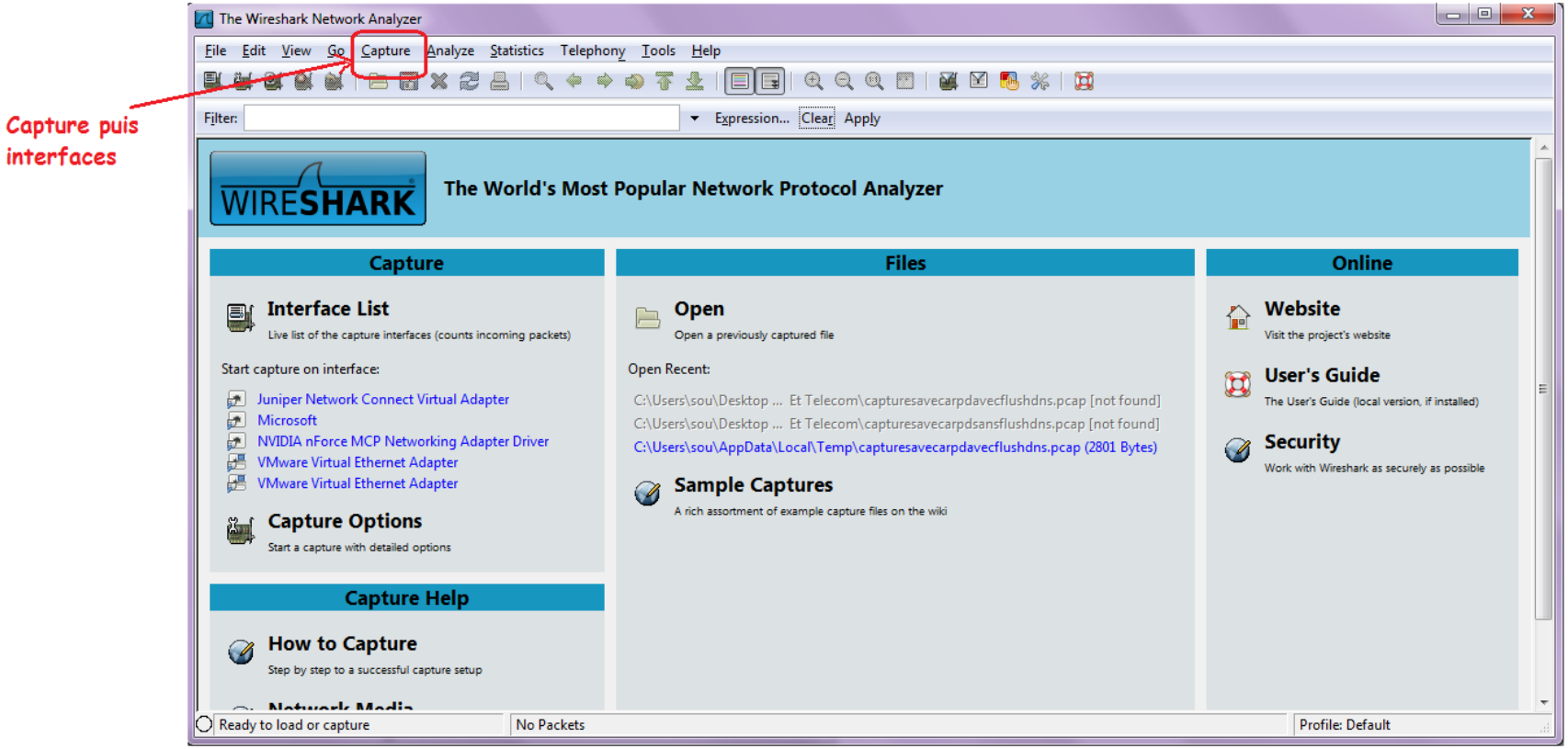
Souad Hadjres

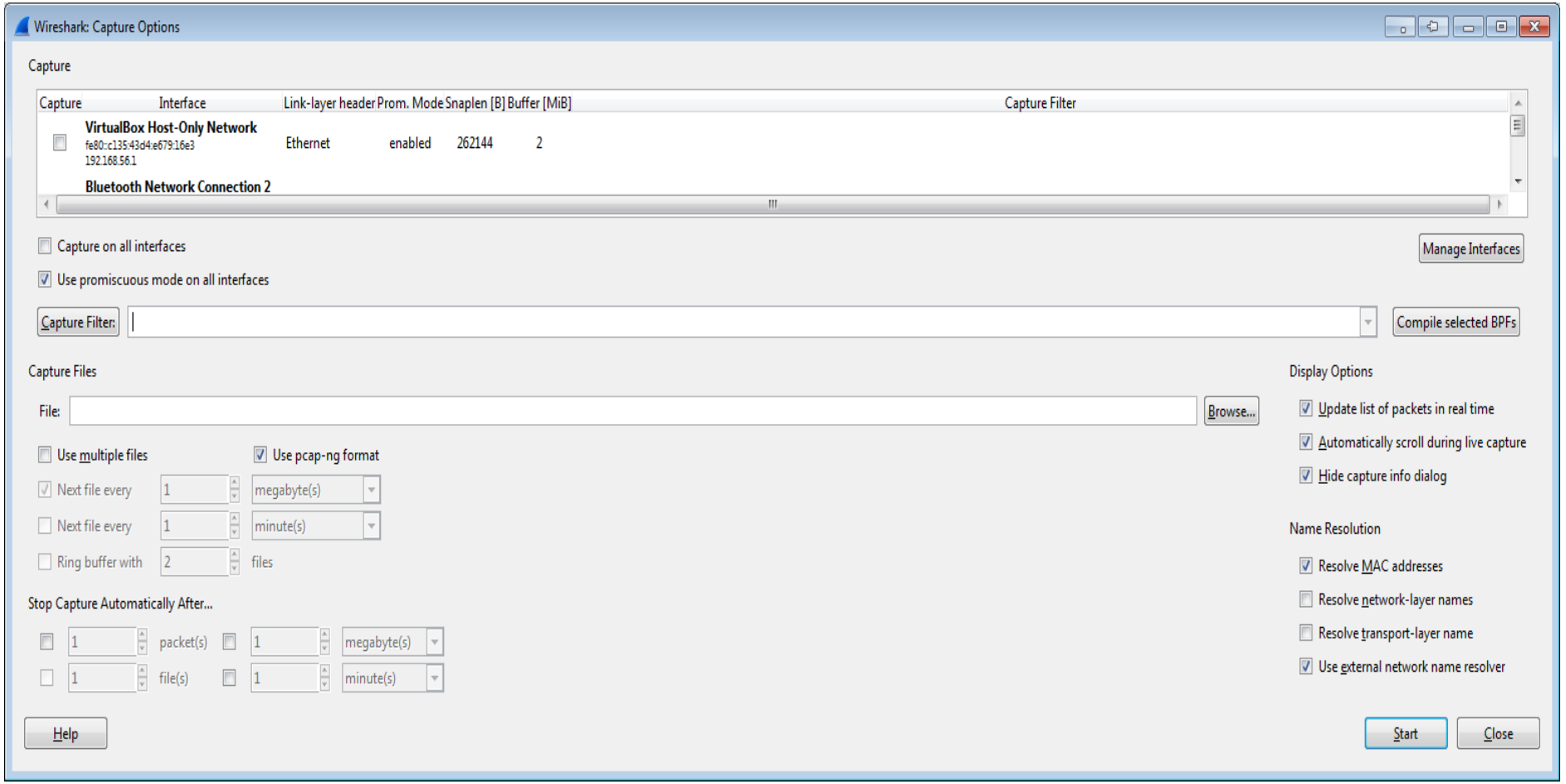
C’est quoi Wireshark?

- Un « renifleur » de paquets permettant de capturer et d’analyser les paquets reçus ou envoyés par une application ou un protocole s’éxécutant sur votre machine.
- Essentiel pour observer et comprendre les mécanismes de fonctionnement des protocoles de communication
- Utilisable sur presque toutes les technologies

- Logiciel libre téléchargeable à partir de :
  http://www.wireshark.org/

Pour démarrer une capture :





- Le « promiscuous » mode permet de capturer tous les paquets qui passent par l'interface choisie, même s'ils ne sont n'est émis ni reçus par ma machine. Donc tout trafic reçu par la carte réseau sera capturé!

- Il est parfois utile d'appliquer un filtre qui ne va capturer ou afficher que certains paquets. Wireshark comporte deux types de filtres, les filtres de capture et les filtres d'affichages.
  - **Les filtres de capture**: Utilisés pour sélectionner les données à capturer et à enregistrer. Ils sont définis avec le démarrage de la capture dans la fenêtre des options de capture.

    Exemples :

    ip src host 10.1.1.1

    affiche les paquets avec une adresse IP source égale à 10.1.1.1.

    host 192.168.10.129

    affiche les paquets avec une adresse source ou destination égal à 192.168.10.129

    tcp dst port 3128

    affiche les paquets avec un port destination TCP de 3128.

  - **Les filtres d'affichage**: Utilisés pour rechercher à l'intérieur des données capturées. Ils permettent de rechercher exactement les données souhaitées.

    Exemples :

    ip.addr == 192.168.10.129

    affiche les paquets avec une adresse source ou destination de 192.168.10.129

    ip.src != 10.1.2.3 or ip.dst != 10.4.5.6

    affiche les paquets avec une adresse IP source différente de 10.1.2.3 ou avec une adresse IP destination différente de 10.4.5.6.

    tcp.port == 25 : affiche les paquets dont le port TCP source ou destination est égal à 25.

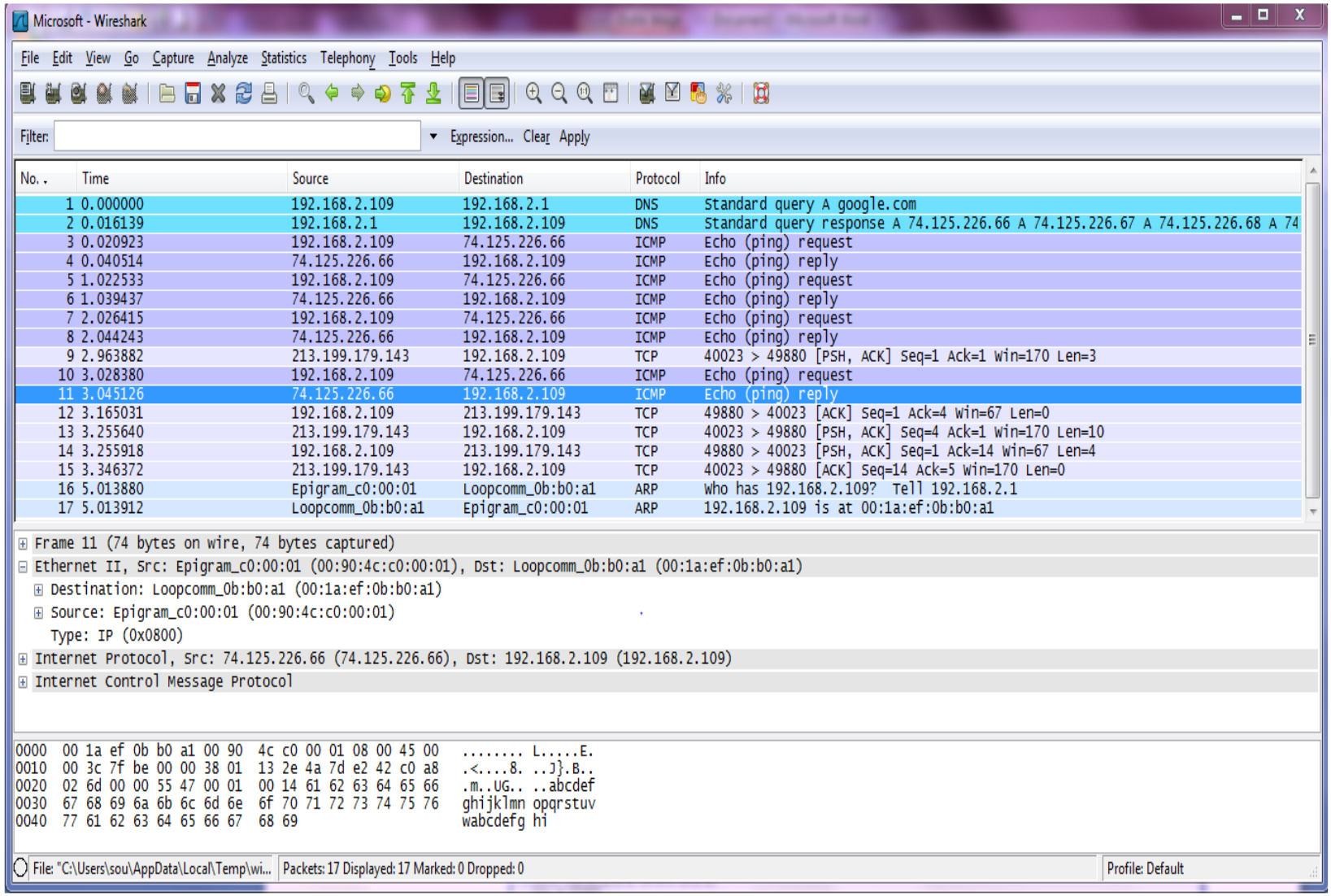
Menus

Filtrage

Liste des paquets capturés

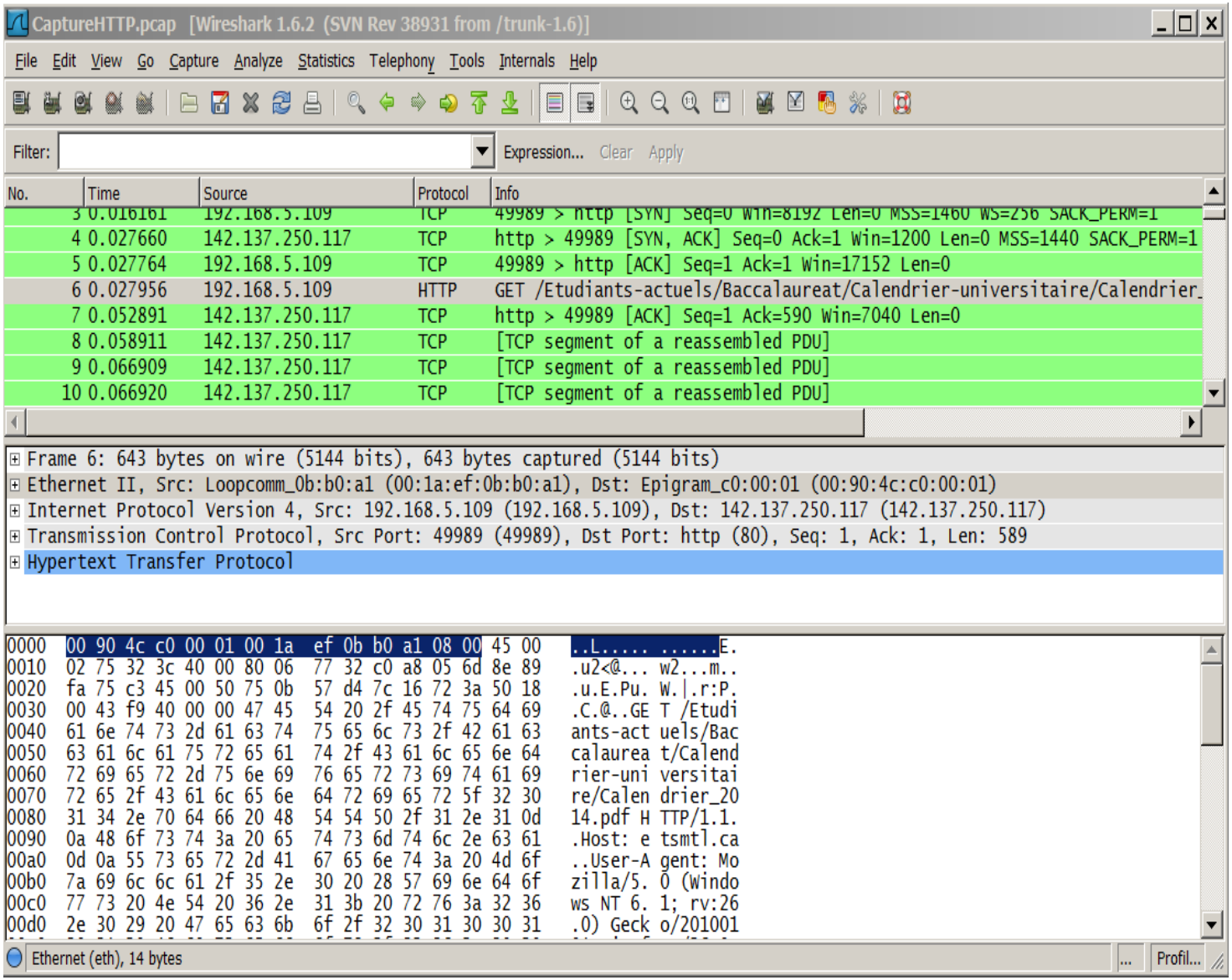
Détails du paquet selectionné

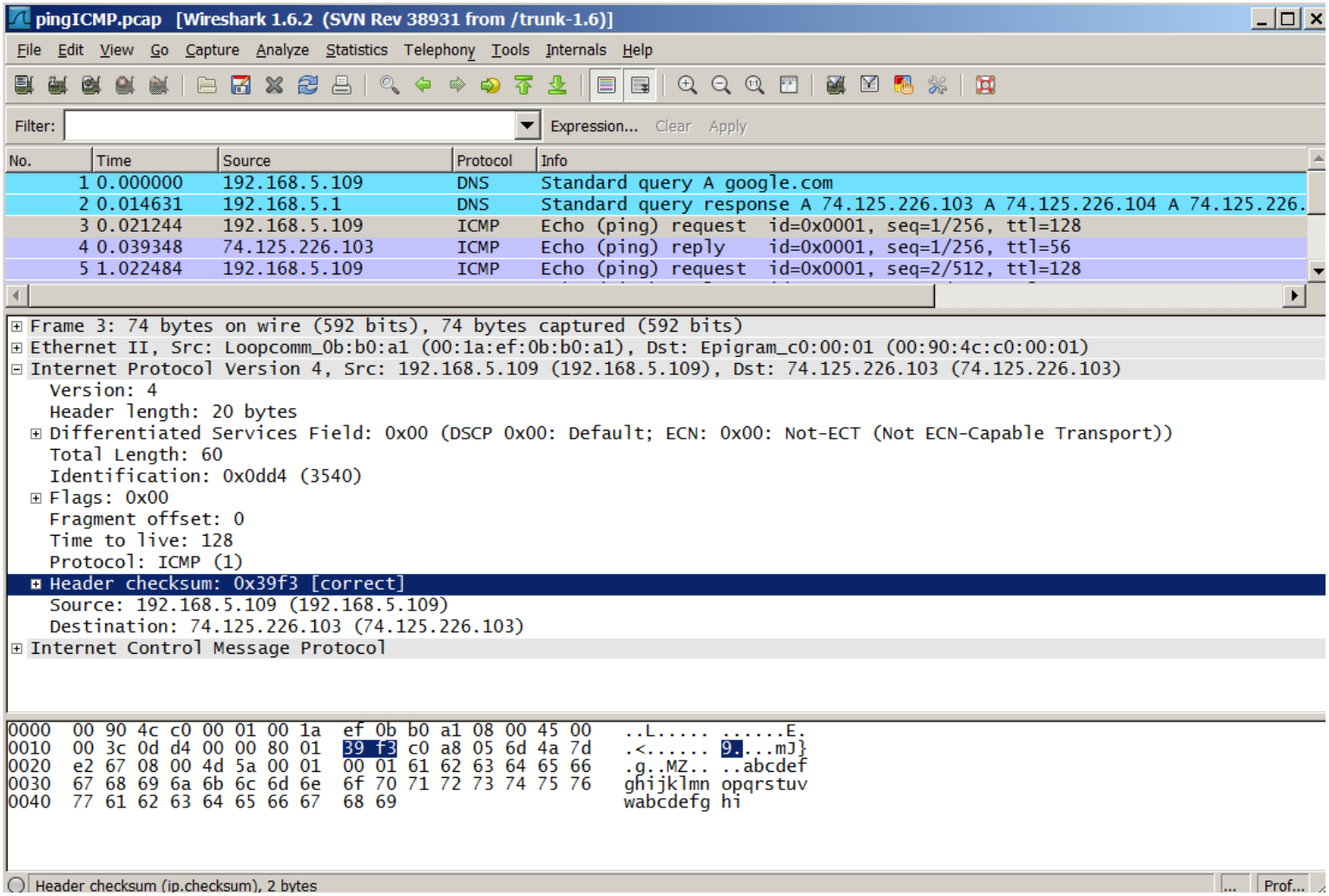
Contenu du paquet selectionné (en hexadécimal)

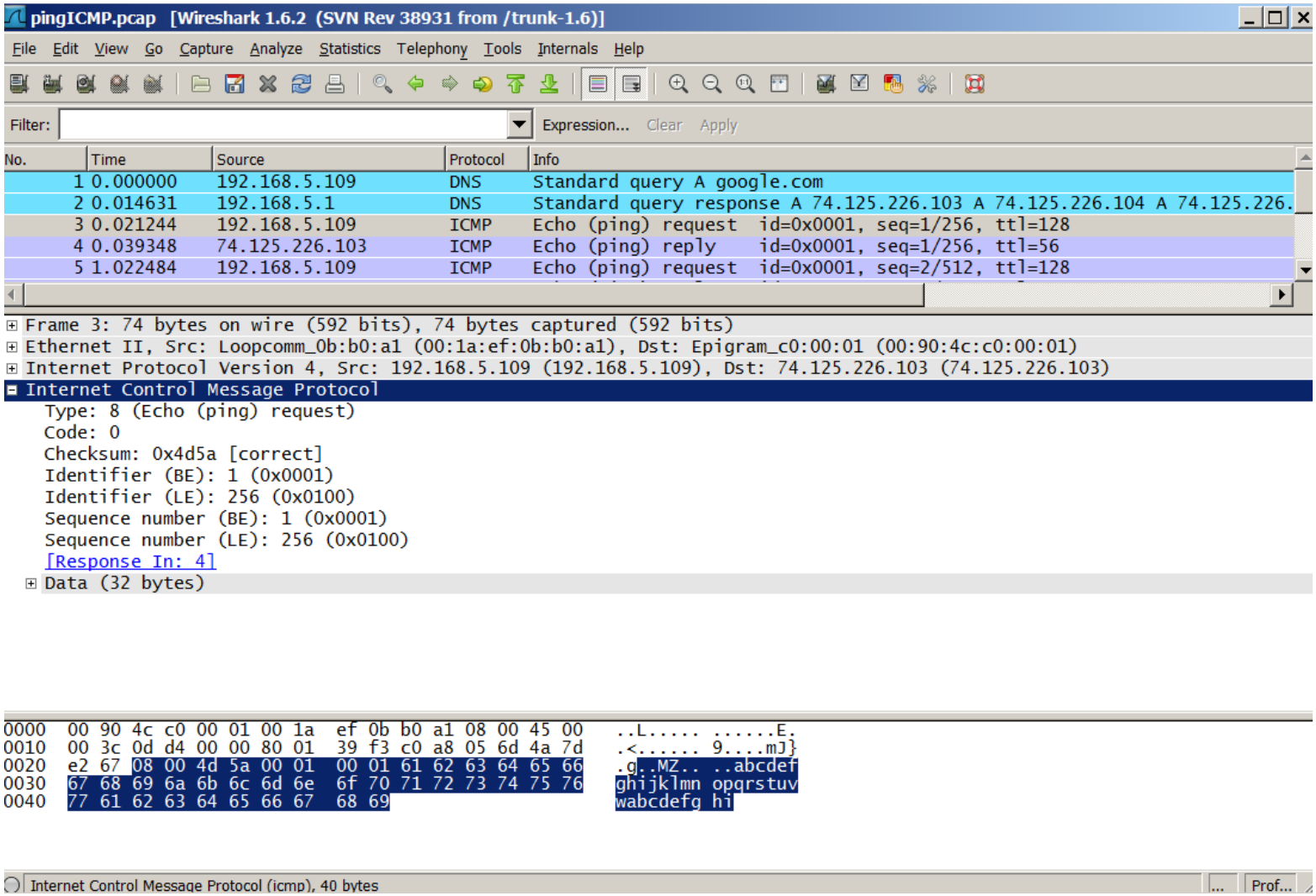


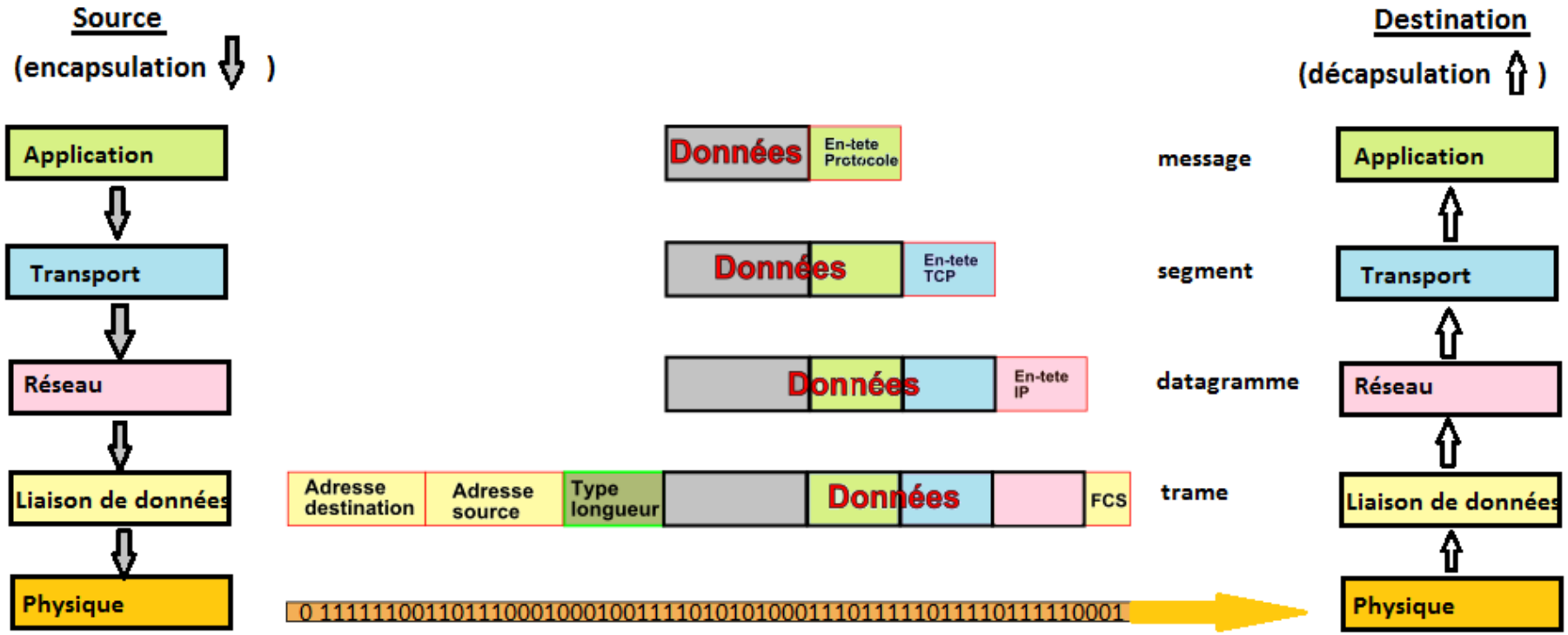
- La fenêtre « liste des paquets sélectionnés » affiche une ligne de résumé pour chaque paquet capturé.
  Cette ligne contient :
  - le numéro du paquet (attribué par Wireshark)
  - le temps de capture du paquet
  - les adresses source et destination du paquet : c'est des adresses MAC si le paquet est de couche 2, et des adresses IP si le paquet est de couche 3 ou plus.
  - le type de protocole véhiculé par le paquet, et
  - le résumé des champs caractéristiques de ce protocole.

- La fenêtre « détails du paquet sélectionné » donne la pile de protocoles décodés, pour le paquet sélectionné, allant du niveau physique (en haut de la fenêtre) jusqu'au niveau le plus haut reconnu. Ainsi, si le protocole analysé est un protocole de couche 3, on aura 3 lignes, une pour chaque couche, et s'il y' a lieu, une dernière ligne représentant la charge utile.
  Exemple : Si on considère un paquet véhiculant le protocole http, on aura :
  - Une première ligne correspondant à la couche physique => la quantité de bits capturés et la date de capture.
  - Une deuxième ligne correspondant à la couche liaison => le type et les champs de la trame et les adresses physiques.
  - Une troisième ligne correspondant à la couche réseau => les adresses logiques et les indicateurs d'état.
  - Une quatrième ligne correspondant à la couche transport => l'état de la connexion, numéros de ports utilisés et diverses options.
  - Une cinquième ligne correspondant à la couche application => les données utilisateur.


CaptureHTTP.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply
No. Time Source Protocol Info
3 0.016161 192.168.5.109 TCP 49989 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
4 0.027660 142.137.250.117 TCP http > 49989 [SYN, ACK] Seq=0 Ack=1 Win=1200 Len=0 MSS=1440 SACK_PERM=1
5 0.027764 192.168.5.109 TCP 49989 > http [ACK] Seq=1 Ack=1 Win=17152 Len=0
6 0.027956 192.168.5.109 HTTP GET /Etudiants-actuels/Baccalaureat/Calendrier-universitaire/Calendrier_
7 0.052891 142.137.250.117 TCP http > 49989 [ACK] Seq=1 Ack=590 Win=7040 Len=0
8 0.058911 142.137.250.117 TCP [TCP segment of a reassembled PDU]
9 0.066909 142.137.250.117 TCP [TCP segment of a reassembled PDU]
10 0.066920 142.137.250.117 TCP [TCP segment of a reassembled PDU]
Frame 6: 643 bytes on wire (5144 bits), 643 bytes captured (5144 bits)
Ethernet II, Src: Loopcomm_0b:b0:a1 (00:1a:ef:0b:b0:a1), Dst: Epigram_c0:00:01 (00:90:4c:c0:00:01)
Internet Protocol Version 4, Src: 192.168.5.109 (192.168.5.109), Dst: 142.137.250.117 (142.137.250.117)
Transmission Control Protocol, Src Port: 49989 (49989), Dst Port: http (80), Seq: 1, Ack: 1, Len: 589
Hypertext Transfer Protocol
0000 00 90 4c c0 00 01 00 1a ef 0b b0 a1 08 00 45 00 ..L..... ......E.
0010 02 75 32 3c 40 00 80 06 77 32 c0 a8 05 6d 8e 89 .u2<@... w2...m..
0020 fa 75 c3 45 00 50 75 0b 57 d4 7c 16 72 3a 50 18 .u.E.Pu. W.|.r:P.
0030 00 43 f9 40 00 00 47 45 54 20 2f 45 74 75 64 69 .C.@..GE T /Etudi
0040 61 6e 74 73 2d 61 63 74 75 65 6c 73 2f 42 61 63 ants-act uels/Bac
0050 63 61 6c 61 75 72 65 61 74 2f 43 61 6c 65 6e 64 calaurea t/Calend
0060 72 69 65 72 2d 75 6e 69 76 65 72 73 69 74 61 69 rier-uni versitai
0070 72 65 2f 43 61 6c 65 6e 64 72 69 65 72 5f 32 30 re/Calen drier_20
0080 31 34 2e 70 64 66 20 48 54 54 50 2f 31 2e 31 0d 14.pdf H TTP/1.1.
0090 0a 48 6f 73 74 3a 20 65 74 73 6d 74 6c 2e 63 61 .Host: e tsmtl.ca
00a0 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f ..User-A gent: Mo
00b0 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e 64 6f zilla/5. 0 (Windo
00c0 77 73 20 4e 54 20 36 2e 31 3b 20 72 76 3a 32 36 ws NT 6. 1; rv:26
00d0 2e 30 29 20 47 65 63 6b 6f 2f 32 30 31 30 30 31 .0) Geck o/201001
Ethernet (eth), 14 bytes
Profil...


pingICMP.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply
No. Time Source Protocol Info
1 0.000000 192.168.5.109 DNS Standard query A google.com
2 0.014631 192.168.5.1 DNS Standard query response A 74.125.226.103 A 74.125.226.104 A 74.125.226.
3 0.021244 192.168.5.109 ICMP Echo (ping) request id=0x0001, seq=1/256, ttl=128
4 0.039348 74.125.226.103 ICMP Echo (ping) reply id=0x0001, seq=1/256, ttl=56
5 1.022484 192.168.5.109 ICMP Echo (ping) request id=0x0001, seq=2/512, ttl=128
Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: Loopcomm_0b:b0:a1 (00:1a:ef:0b:b0:a1), Dst: Epigram_c0:00:01 (00:90:4c:c0:00:01)
Internet Protocol Version 4, Src: 192.168.5.109 (192.168.5.109), Dst: 74.125.226.103 (74.125.226.103)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 60
Identification: 0x0dd4 (3540)
Flags: 0x00
Fragment offset: 0
Time to live: 128
Protocol: ICMP (1)
Header checksum: 0x39f3 [correct]
Source: 192.168.5.109 (192.168.5.109)
Destination: 74.125.226.103 (74.125.226.103)
Internet Control Message Protocol
0000 00 90 4c c0 00 01 00 1a ef 0b b0 a1 08 00 45 00 ..L..... ......E.
0010 00 3c 0d d4 00 00 80 01 39 f3 c0 a8 05 6d 4a 7d .<...... 9....mJ}
0020 e2 67 08 00 4d 5a 00 01 00 01 61 62 63 64 65 66 .g..MZ.. ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi
Header checksum (ip.checksum), 2 bytes
Prof...


pingICMP.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply
No. Time Source Protocol Info
1 0.000000 192.168.5.109 DNS Standard query A google.com
2 0.014631 192.168.5.1 DNS Standard query response A 74.125.226.103 A 74.125.226.104 A 74.125.226.
3 0.021244 192.168.5.109 ICMP Echo (ping) request id=0x0001, seq=1/256, ttl=128
4 0.039348 74.125.226.103 ICMP Echo (ping) reply id=0x0001, seq=1/256, ttl=56
5 1.022484 192.168.5.109 ICMP Echo (ping) request id=0x0001, seq=2/512, ttl=128
Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: Loopcomm_0b:b0:a1 (00:1a:ef:0b:b0:a1), Dst: Epigram_c0:00:01 (00:90:4c:c0:00:01)
Internet Protocol Version 4, Src: 192.168.5.109 (192.168.5.109), Dst: 74.125.226.103 (74.125.226.103)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d5a [correct]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Response In: 4]
Data (32 bytes)
0000 00 90 4c c0 00 01 00 1a ef 0b b0 a1 08 00 45 00 ..L..... ......E.
0010 00 3c 0d d4 00 00 80 01 39 f3 c0 a8 05 6d 4a 7d .<...... 9....mJ}
0020 e2 67 08 00 4d 5a 00 01 00 01 61 62 63 64 65 66 .g..MZ.. ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi
Internet Control Message Protocol (icmp), 40 bytes
Prof...



Chaque couche rajoute son entête afin de dialoguer avec son vis à vis

Références :

http://www.inetdoc.net/travaux_pratiques/intro.analyse/wireshark.gui.html

http://openmaniak.com/fr/wireshark_use.php