R2.05 – Les protocoles reseaux

RESPONSABLE : CRISTINA ONETE

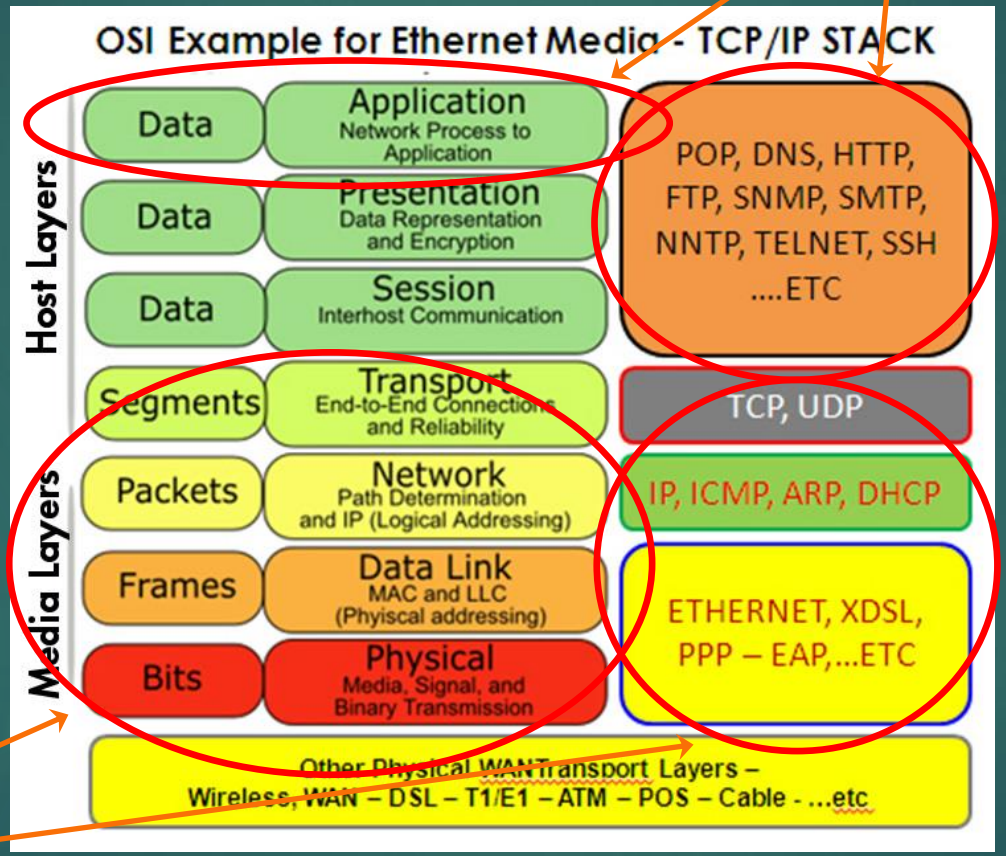
MATERIEL : HTTPS://WWW.ONETE.NET/TEACHING.HTML

EMAIL : CRISTINA.ONETE@GMAIL.COM

R2.04, R2.05

R2.05

Les couches architecturales des réseaux



Les protocoles des réseaux

R2.04

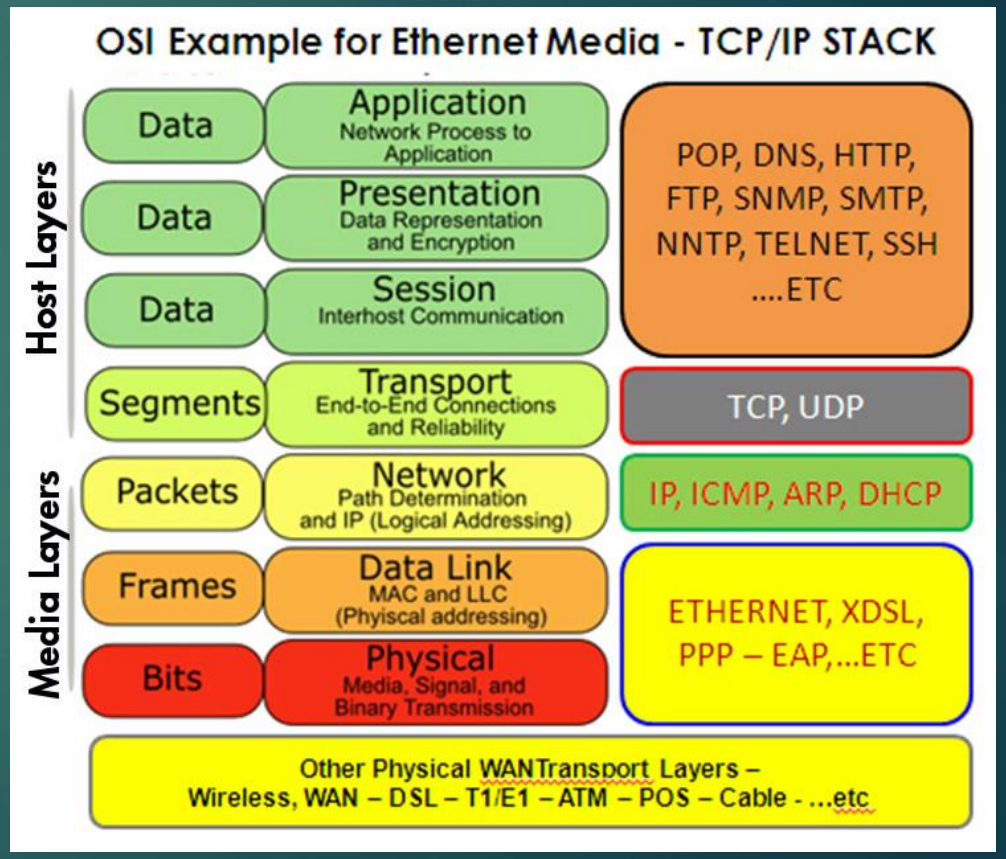
Source : ipv6.com

Aujourd'hui : des applications

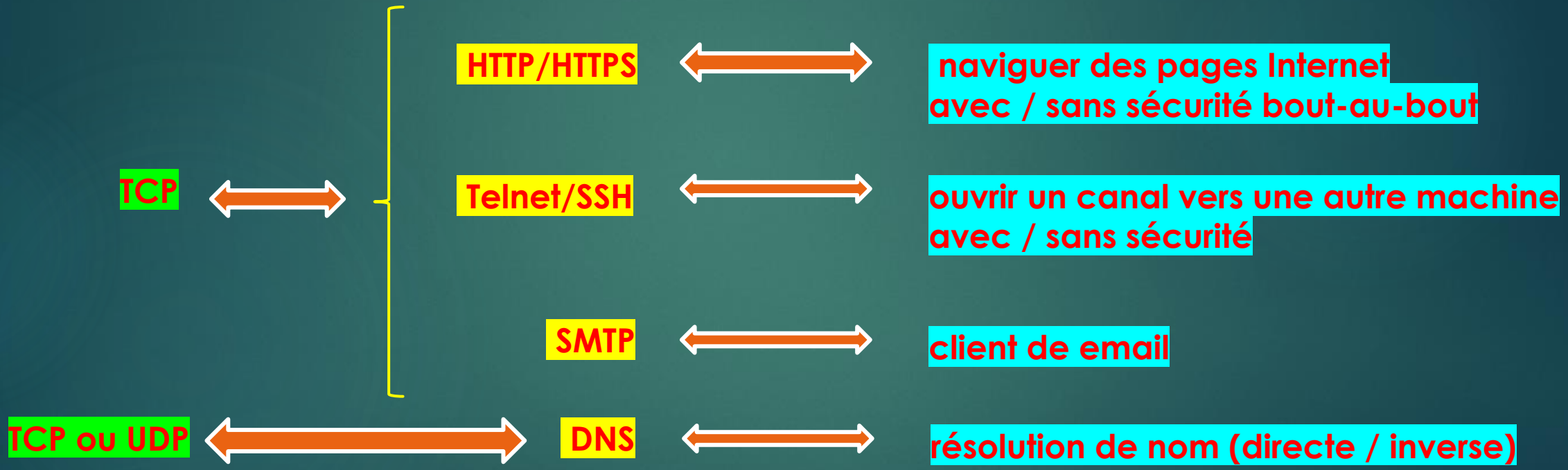
données des applications

messages pour plusieurs applications

séparation : chaque application ←→ port



Quelques protocoles typiques



HTTP(S) : la navigation Internet

- Une connexion Internet :

Modèle client-serveur

Client :

- application envoie des requêtes
- initialise le contact
- acrif

Serveur :

- application répond aux requêtes
- à l'attente
- sur Apache, Windows Server

Côté client

Processus

```
root      2094     1  0 10:03 ?        Ss     0:00 /usr/sbin/apache2 -k
www-data  2162  2094  0 10:03 ?        S      0:00  \_ /usr/sbin/apache2
www-data  2163  2094  0 10:03 ?        S      0:00  \_ /usr/sbin/apache2
www-data  2164  2094  0 10:03 ?        S      0:00  \_ /usr/sbin/apache2
www-data  2165  2094  0 10:03 ?        S      0:00  \_ /usr/sbin/apache2
www-data  2166  2094  0 10:03 ?        S      0:00  \_ /usr/sbin/apache2
```

6 processus pour traiter des requêtes

commande ss, options lntp (remplace netstat)

Réseau

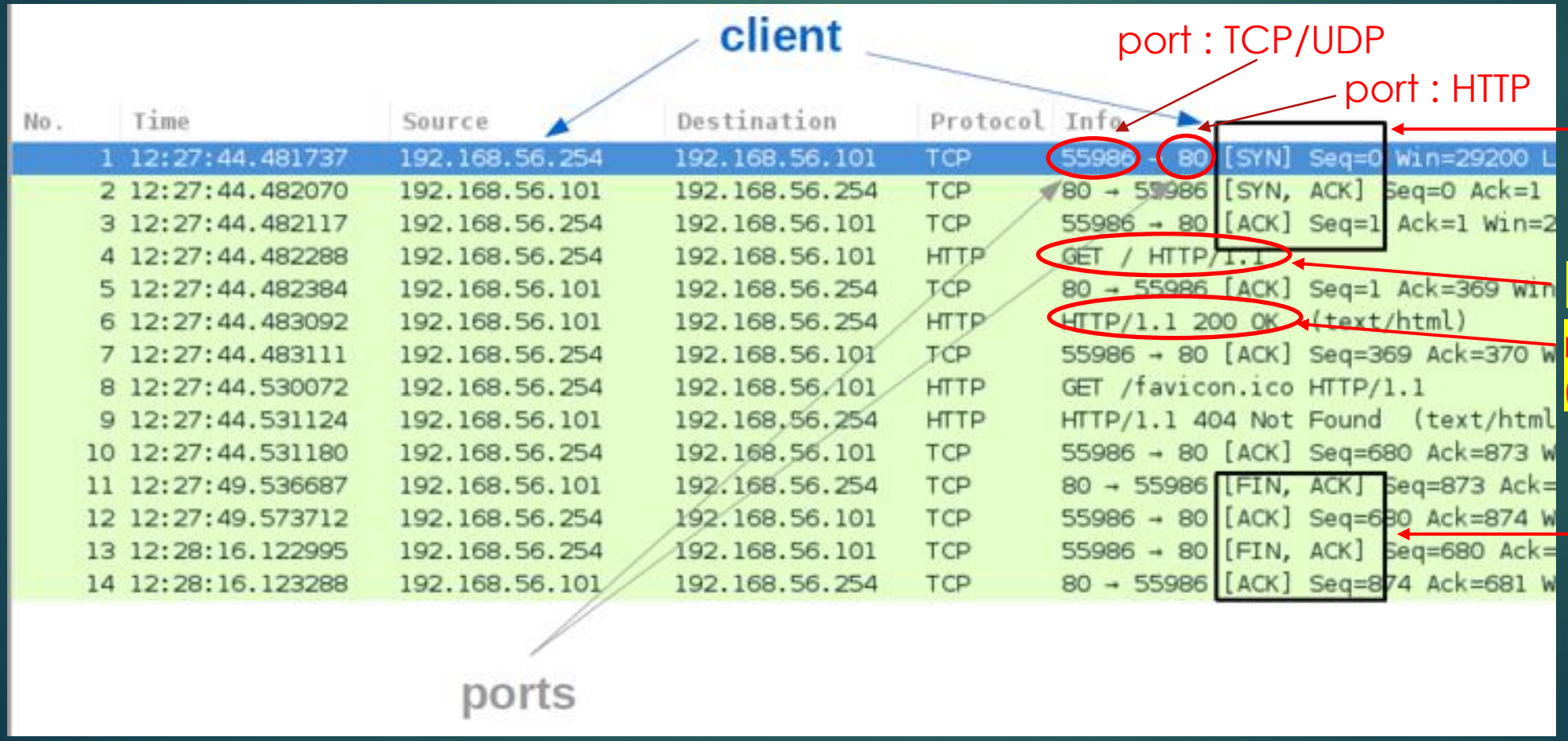
```
root@debian7-cli:~# netstat -lntp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
3389/apache2
```

Connexion se fait sur TCP

Requête attendee, port 80

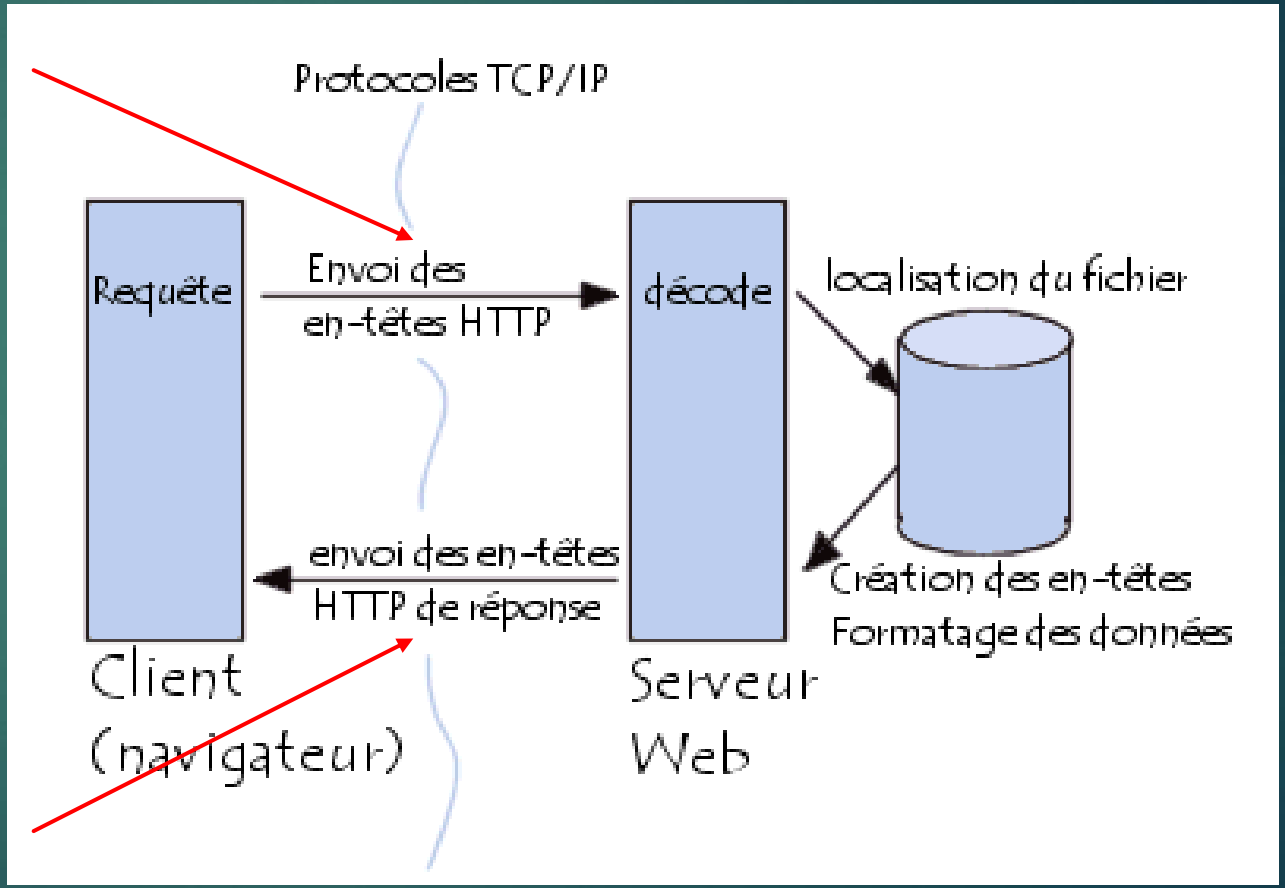
État d'écoute : LISTEN

HTTP sur Wireshark



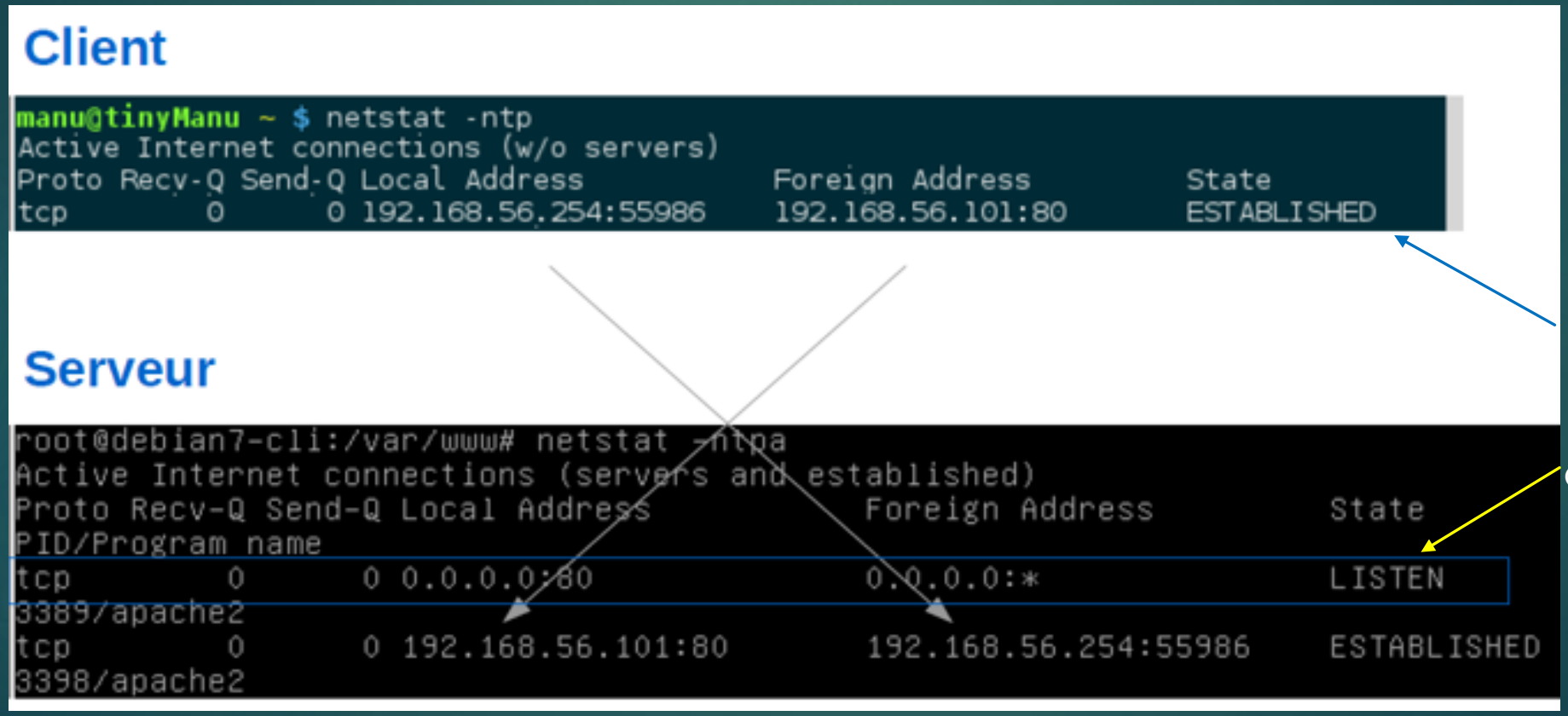
Le protocole HTTP


```
GET /fichier.html HTTP 1.1
Host: www.monsite.fr
User-Agent : Mozilla...
```

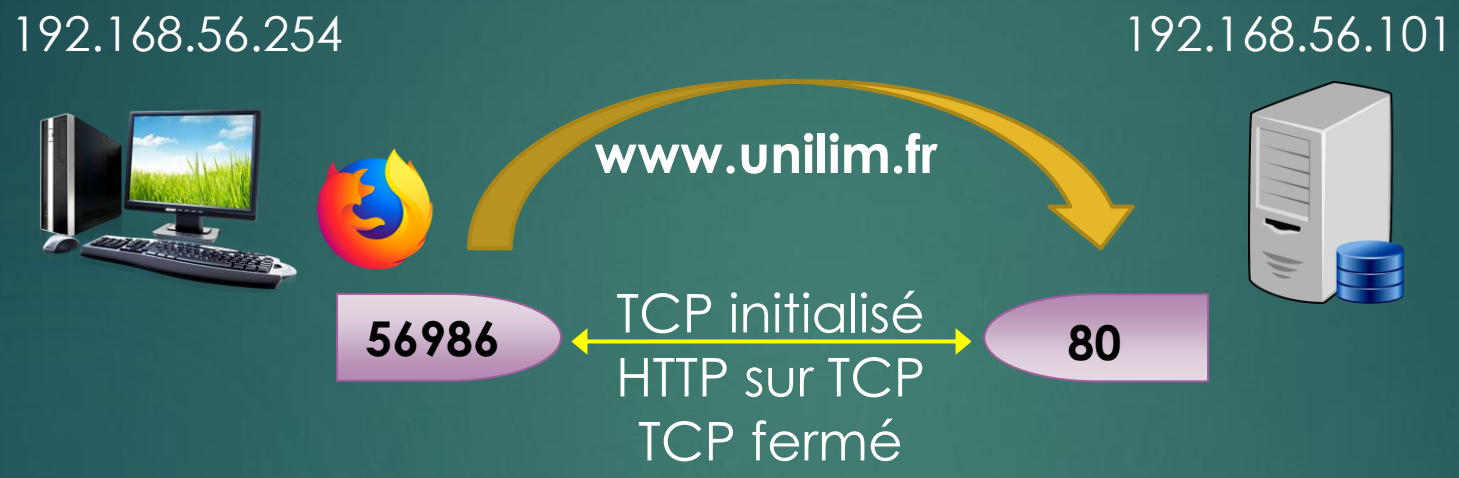



```
HTTP 1.1   200 OK
...
Server : Apache/2.2.3
Content-Length: <taille>
Content-type: text/html
```

Avec la commande ss



Pour résumer



- 1. Le client cherche la machine ciblée par adresse IP
- 2. Le service Web est à trouver sur le port 80 (HTTP)
- 3. La communication HTTP se fait dans l'encapsulation HTTP

HTTP vs HTTPS

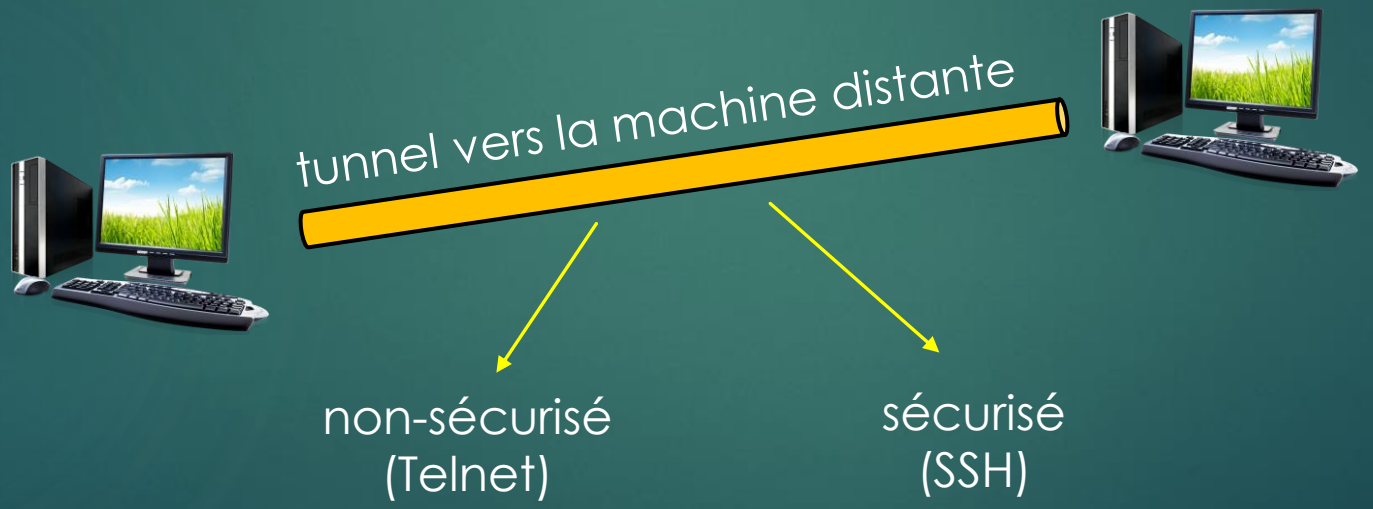
- Port HTTP : 80 vs. port HTTPS : 443
- Les communications sur HTTP sont non-sécurisées
  - Par contre HTTPS sert à donner une sécurité de bout au bout
  - De plus en plus, les connexions aujourd'hui sont sécurisées

- HTTPS = HTTP sur TLS (sur TCP)
  - Une première étape : un échange de messages, standard RFC 5246
  - A la fin : le client + le serveur partagent des clés de chiffrement /authentification
  - Deuxième étape : communication HTTP chiffrée + authentifiée

Telnet & SSH

Problématique

- Utilisateur d'une machine veut manipuler une machine à distance
  - Pour modifier ses fichiers, par exemple
  - Ou pour contrôler l'état de cette machine (pour un serveur, par exemple)



Telnet

- Protocole qui permet d'émuler un terminal vers une machine ciblée
- Les commandes tapées dans le terminal s'exécuteront sur l'autre machine

- Environnement client-serveur
  - Machine cible = serveur
  - Machine sur laquelle on se connecte = client
- Canal non-sécurisé (information susceptible à être interceptée)
  - Mais accès limité aux personnes fournissant un nom d'utilisateur & un mot de passe

Quelques commandes

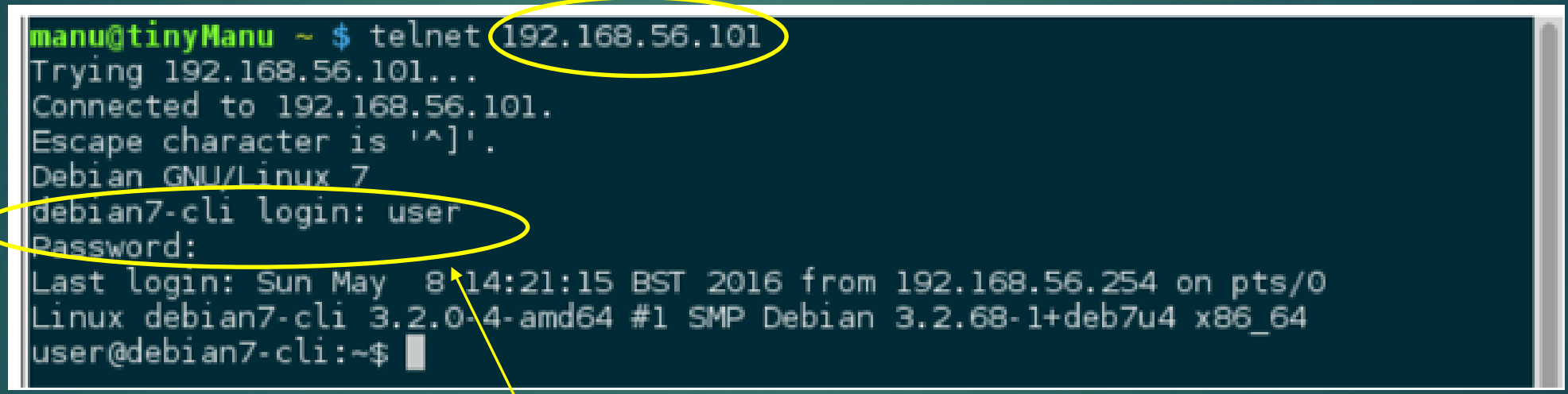
- Pour se connecter : `telnet <adresse IP de la machine ciblée>`

Commande	Description
?	Affiche l'aide
close	Termine la session Telnet
display	Affiche à l'écran les paramètres de la connexion (type de terminal, port)
environ	Permet de définir les variables d'environnement du système d'exploitation
logout	Permet de se déconnecter
mode	Bascule entre les modes de transfert ASCII (transfert d'un fichier en mode texte) et BINARY (transfert d'un fichier en binaire)
open	Permet de lancer une autre connexion à partir de la connexion en cours
quit	Quitte l'application Telnet
set	Modifie les paramètres de la connexion
unset	Charge les paramètres de connexion par défaut

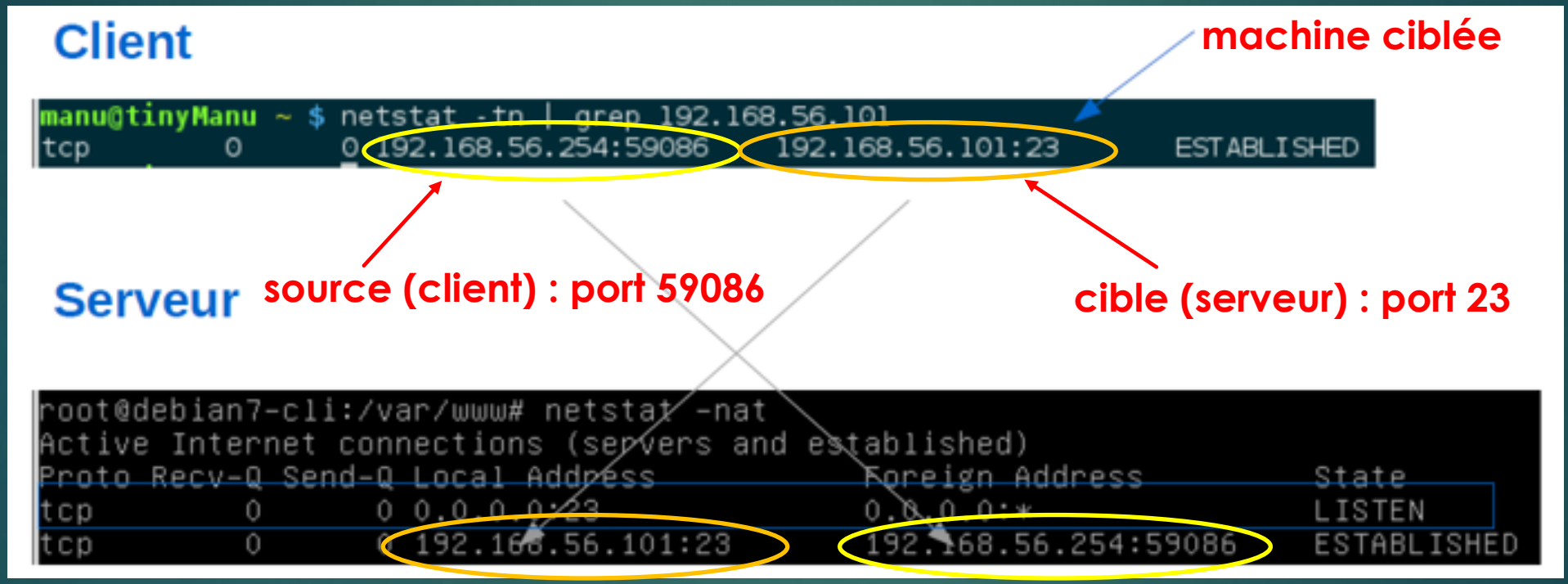
Source : commentcamarche.com

Se connecter à Telnet

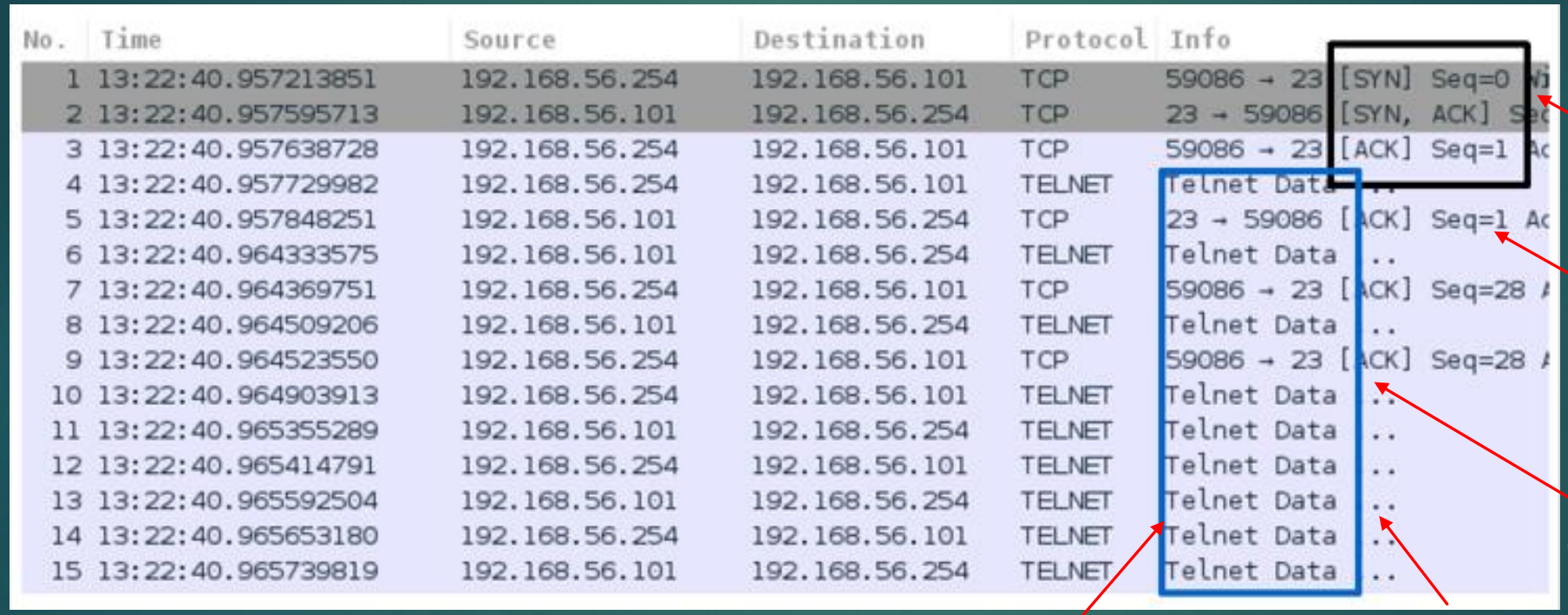
- Port 23 sur TCP



Telnet sur netstat



Sur Wireshark



No.	Time	Source	Destination	Protocol	Info
1	13:22:40.957213851	192.168.56.254	192.168.56.101	TCP	59086 → 23 [SYN] Seq=0
2	13:22:40.957595713	192.168.56.101	192.168.56.254	TCP	23 → 59086 [SYN, ACK]
3	13:22:40.957638728	192.168.56.254	192.168.56.101	TCP	59086 → 23 [ACK] Seq=1
4	13:22:40.957729982	192.168.56.254	192.168.56.101	TELNET	Telnet Data ..
5	13:22:40.957848251	192.168.56.101	192.168.56.254	TCP	23 → 59086 [ACK] Seq=1
6	13:22:40.964333575	192.168.56.101	192.168.56.254	TELNET	Telnet Data ..
7	13:22:40.964369751	192.168.56.254	192.168.56.101	TCP	59086 → 23 [ACK] Seq=28
8	13:22:40.964509206	192.168.56.101	192.168.56.254	TELNET	Telnet Data ..
9	13:22:40.964523550	192.168.56.254	192.168.56.101	TCP	59086 → 23 [ACK] Seq=28
10	13:22:40.964903913	192.168.56.254	192.168.56.101	TELNET	Telnet Data ..
11	13:22:40.965355289	192.168.56.101	192.168.56.254	TELNET	Telnet Data ..
12	13:22:40.965414791	192.168.56.254	192.168.56.101	TELNET	Telnet Data ..
13	13:22:40.965592504	192.168.56.101	192.168.56.254	TELNET	Telnet Data ..
14	13:22:40.965653180	192.168.56.254	192.168.56.101	TELNET	Telnet Data ..
15	13:22:40.965739819	192.168.56.101	192.168.56.254	TELNET	Telnet Data ..

TCP initialisé

ACK TCP à part

données envoyées par Telnet

ACK dans message Telnet

données non-chiffrées

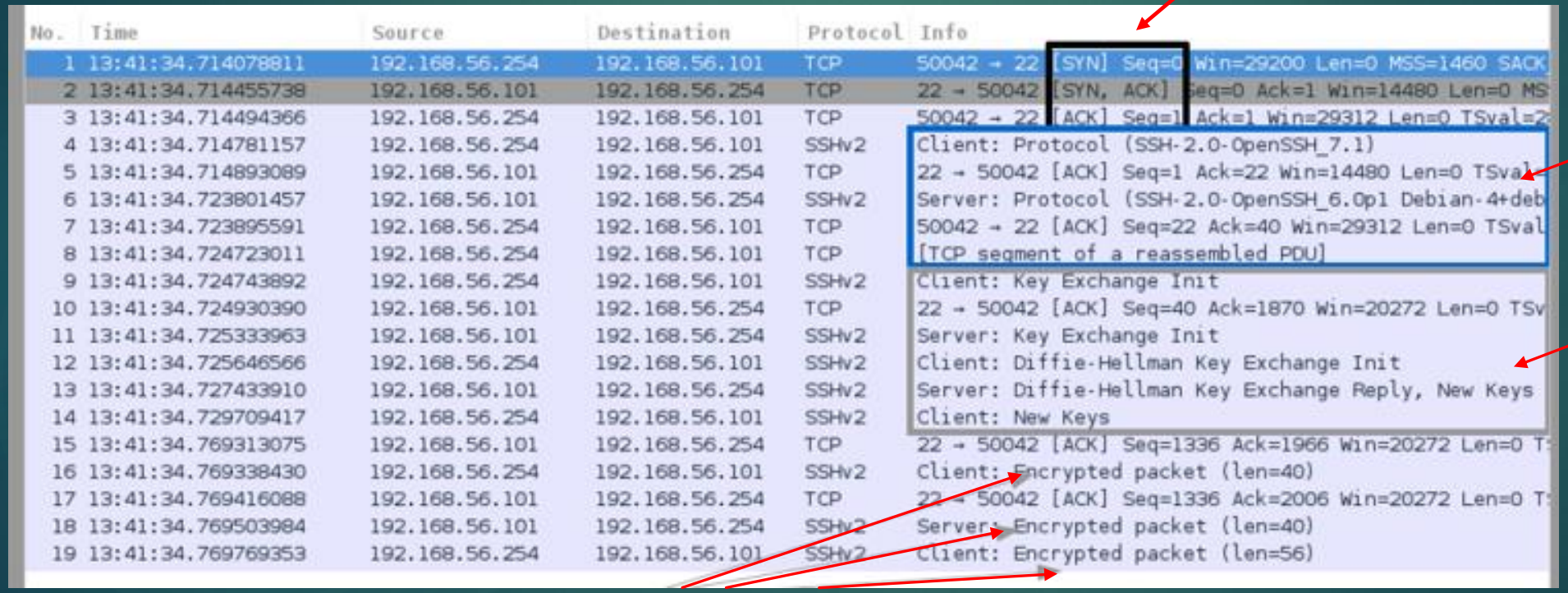
Telnet vs. SSH

- La communication sur Telnet n'est pas sécurisée
  - Si on veut avoir une sécurité de bout-au-bout, il faut utiliser SSH

- SSH sert à authentifier les deux parties et à établir des clés partagées pour faire du chiffrement et l'authentification des messages
  - Même rôle que TLS, mais un protocole different
  - Port 22 sur TCP

```
manu@tinyManu ~ $ ssh user@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ECDSA key fingerprint is SHA256:NJEV9ZQ1xJNNF/ut67o059+nV1ZxaaNNv5lqrkJE9WY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.101' (ECDSA) to the list of known hosts.
user@192.168.56.101's password:
Linux debian7-cli 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u4 x86_64
Last login: Sun May  8 14:38:49 2016 from 192.168.56.254
user@debian7-cli:~$ pwd
/home/user
user@debian7-cli:~$ exit
logout
Connection to 192.168.56.101 closed.
```

SSH sur Wireshark

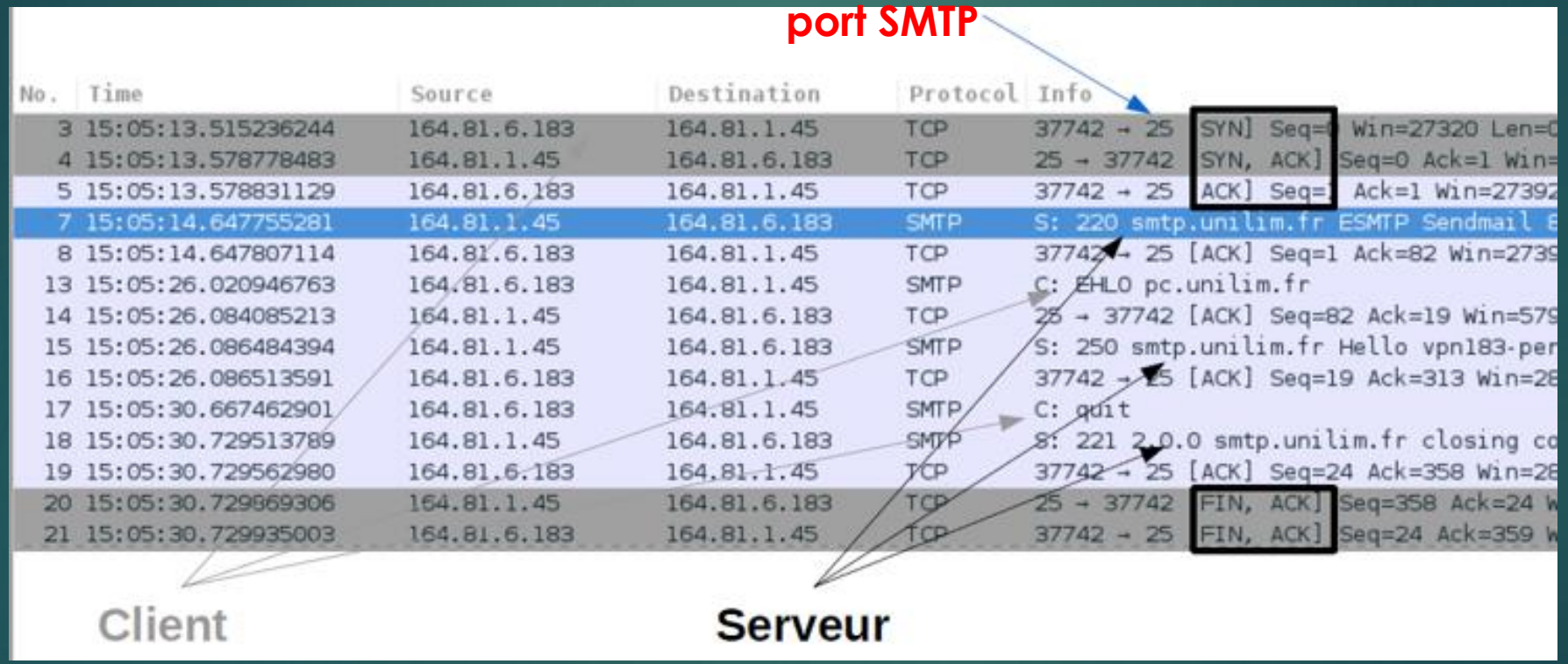


SMTP: Simple mail transfer protocol

Les serveurs mail

- Un utilisateur (client mail) consulte ses mails sur un serveur mail
- Les emails écrites par le client sont stockés par le serveur, et livres à un autre serveur mail
- Le recepteur consulte son propre serveur pour retrouver ses messages

SMTP sur Wireshark



SMTP -- le protocole

- Pour Outlook (ou d'autres clients WebMail), d'autres protocoles sont possibles