



**POLYTECHNIQUE
MONTRÉAL**

**LE GÉNIE
EN PREMIÈRE CLASSE**

Réseaux Informatiques
INF3405

Laboratoire 4

Présenté par:

Ibrahima Séga Sangaré (1788085)

et

Khalil Benani (1566707)

Soumis à:

Bilal Itani

Question 7

- 1) Quel filtre appliqueriez-vous afin d'afficher uniquement les échanges entre le client et le serveur? (1 point)
- le filtre est le : ip.addr==192.168.80.158 && ip.addr==192.168.80.164 && tcp.port==5000
- 2) À la lumière de vos observations, dites quel protocole de la couche 4 est utilisé pour la communication entre le client et le serveur. (0.5 point)

Le protocole est le TCP.

- 3) Combien de paquets et d'octets de données ont été envoyés du client vers le serveur et du serveur vers le client? (2 points)

nombre de paquets client vers serveurs: 21 paquet TCP
taille total des données client vers serveurs: 11 octets

nombre de paquets serveur vers client: 27 paquet
taille total des données serveur vers client: 17 octets

- 4) Normalement, le standard IEEE 802.3 limite la taille d'une trame Ethernet à 1518 octets. Dans votre capture Wireshark, existe-t-il des paquets ayant une taille supérieure à 1518 octets? Si oui, expliquez pourquoi et comment ce paquet réussit à transiger sur le réseau alors que sa taille est plus grande que celle spécifiée par le standard.

La plupart des réseaux imposent une limite physique à la taille des données qu'ils peuvent transporter. Un datagramme IP peut avoir une taille maximale de 65535 octets, ce qui est trop grand pour la plupart des réseaux. Le MTU (Maximum Transfert Unit) correspond à la taille maximale des données transportables par le réseau. Le datagramme IP (entête comprise) aura comme taille maximale le MTU du réseau. Mais comme il est cité précédemment cette règle est valide la plupart du temps mais pas toujours comme pour le cas d'un FDDI ou dans notre cas où la carte réseau le permet dans une connexion LAN.

- 5) Quel type d'information êtes-vous capables d'extraire de Wireshark en lien avec l'authentification au serveur de traitement d'images? (1 point)

0000	00	0c	29	09	65	b6	00	0c	29	ec	8c	45	08	00	45	00	..).	e...)..E..E.
0010	00	45	1d	b9	40	00	80	06	ba	53	c0	a8	50	aa	c0	a8	.E..@...	.S..P...	
0020	50	ab	c4	c2	13	88	8a	06	31	04	fa	4d	2e	75	50	18	P.....	1..M.uP.	
0030	01	00	48	62	00	00	78	70	00	00	00	02	77	04	00	00	..Hb...xpw...	
0040	00	02	74	00	06	6b	68	61	6c	69	6c	74	00	04	31	32	..t..kha	lilt..12	
0050	33	34	78														34x		

on peut voir dans la capture le nom d'utilisateur (khalil) suivi du mot de passe (1234).

- 6) Il est possible, avec Wireshark, d'extraire l'image envoyée par le client ou l'image traitée. Donnez les étapes à suivre, incluant des captures d'écran montrant chaque étape permettant l'extraction de l'image envoyée du client vers le serveur. Servez-vous des propriétés du fichier .jpg énoncées plus haut. Indice: utilisez le programme WinHex après avoir sauvegardé le flot de données en format "Raw" (2 points)

Oui, il est possible.

Les étapes sont:

1. On clique sur un paquet TCP et on sélectionne: (FollowTCPstream)
2. On choisit le flux qui prend départ de la machine client vers la machine serveur et on sélectionne (save as)
3. On choisi un nom pour la sauvegarde en format jpg
4. On ouvre le fichier créé avec WinHex
5. On trouve la valeur Hexadécimale
6. On choisi les valeur à supprimer jusqu'à la dernière valeur
7. On enregistre tous ça pour voir l'image a son etat transmit

9.4 Évaluer les risques et les incertitudes d'une situation Critère d'évaluation : Expliquer la relation étroite entre le développement technologique et le développement social, incluant les impacts de la technologie sur la société et vice versa. 7) Suite à toute cette analyse que pouvez-vous conclure quant à la sécurité de l'application de traitement d'images que vous avez développé lors du travail pratique no.2 (1 point)

Le remplacement de nos objets quotidiens par des objets numériques habitue progressivement le grand public aux usages d'Internet et l'adoption des services de téléphonie mobile étend ces usages à toutes les situations de la vie courante.

- Au cours de la dernière décennie, le nombre d'internautes est passé de 7% à 30%¹⁴ de la population mondiale.
- Le taux d'abonnement à la téléphonie mobile est passé de 10% à 73%¹⁵.^[2]

Et qui dit grand utilisateur dit grand risque de piratage de données personnel, et comme on vu dans le tp on peut intercepter une communication entre deux machine, un client et un serveur qui fournit le service, et ces informations interceptées peuvent être décryptées pour prendre leur forme original et être visualiser clairement et peut être même être utilisé à des fins mauvaises.

Question 8

Adresse IP du serveur avant de lancer les serveurs secrets:

```
C:\Users\GIGL\Downloads\Server_secret\Server>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : localdomain
    Link-local IPv6 Address . . . . . : fe80::9d1c:41e5:a946:cee%5
    IPv4 Address. . . . . : 192.168.80.155
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.80.2

Tunnel adapter isatap.localdomain:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : localdomain

Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2001:0:5ef5:79fb:87a:2d52:7b30:e297
    Link-local IPv6 Address . . . . . : fe80::87a:2d52:7b30:e297%4
    Default Gateway . . . . . : ::
```

```
Link-local IPv6 Address . . . . . : fe80::9d1c:41e5:a946:cee%5
IPv4 Address. . . . . : 192.168.80.155
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.80.2
```

Adresse IP du client avant de lancer le client secret:

```
C:\Users\GIGL\Downloads\Client_secret\Client>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : localdomain
    Link-local IPv6 Address . . . . . : fe80::ac8b:2b48:f0bb:c27a%5
    IPv4 Address. . . . . : 192.168.80.137
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.80.2

Tunnel adapter isatap.localdomain:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : localdomain

Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2001:0:5ef5:79fb:2002:1be0:7b30:e297
    Link-local IPv6 Address . . . . . : fe80::2002:1be0:7b30:e297%4
    Default Gateway . . . . . : ::
```

IPv4 Address. : 192.168.80.137

B) Mode secret (1, 2, 3 et 4) (2 points chaque)

Mode Secret 1

1) Le protocole utilisé est TCP. Le premier échange entre le client et le serveur se trouve ci-dessous. Il s'agit des lignes 57 et 58 (en jaune):

57	1.46352800	192.168.80.137	192.168.80.155	TCP	66	50361-5000	[SYN]	Seq=0 win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
58	1.46360600	192.168.80.155	192.168.80.137	TCP	66	5000-50361	[SYN, ACK]	Seq=0 Ack=1 win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
59	1.47674000	192.168.80.137	192.168.80.155	TCP	60	50361-5000	[ACK]	Seq=1 Ack=1 win=65536 Len=0
60	1.47842100	192.168.80.137	192.168.80.155	TCP	1514	50361-5000	[ACK]	Seq=1 Ack=1 win=65536 Len=1460
61	1.47842200	192.168.80.137	192.168.80.155	TCP	1514	50361-5000	[ACK]	Seq=1461 Ack=1 win=65536 Len=1460
62	1.47842300	192.168.80.137	192.168.80.155	TCP	1134	50361-5000	[PSH, ACK]	Seq=2921 Ack=1 win=65536 Len=1080
63	1.47847800	192.168.80.155	192.168.80.137	TCP	54	5000-50361	[ACK]	Seq=1 Ack=4001 win=65536 Len=0
64	1.47854100	192.168.80.137	192.168.80.155	TCP	60	50361-5000	[FIN, ACK]	Seq=4001 Ack=1 win=65536 Len=0
65	1.47855700	192.168.80.155	192.168.80.137	TCP	54	5000-50361	[ACK]	Seq=1 Ack=4002 win=65536 Len=0
66	1.47873400	192.168.80.155	192.168.80.137	TCP	54	5000-50361	[FIN, ACK]	Seq=1 Ack=4002 win=65536 Len=0
67	1.47908100	192.168.80.137	192.168.80.155	TCP	60	50361-5000	[ACK]	Seq=4002 Ack=2 win=65536 Len=0

L'échange s'appelle Synchronize (SYN). Il y a une synchronisation des Sequence Number et l'établissement d'une connexion. Il y a également une communication UDP. Il n'y a pas le même échange car, étant donné que SYN n'existe pas dans l'entête UDP, on ne peut donc pas faire l'échange. Il n'y a pas d'accusé de réception en UDP non plus.

2)

Pour TCP, le port du client est 50361 et celui du serveur est 5000, comme on peut le voir dans la capture précédente. Pour UDP, le port du client est 51969 et celui du serveur est 64003.

29	0.76717400	192.168.80.137	192.168.80.155	UDP	94	Source port: 51969	Destination port: 64003
----	------------	----------------	----------------	-----	----	--------------------	-------------------------

3)

Client-Serveur

Pour connaître le nombre de données envoyées par le client au serveur on peut appliquer le filtre suivant (couche TCP):

ip.src == 192.168.80.137 && ip.dst == 192.168.80.155 && tcp.len > 0

No.	Time	Source	Destination	Protocol	Length	Info
60	1.47842100	192.168.80.137	192.168.80.155	TCP	1514	50361-5000 [ACK] Seq=1 Ack=1 win=65536 Len=1460
61	1.47842200	192.168.80.137	192.168.80.155	TCP	1514	50361-5000 [ACK] Seq=1461 Ack=1 win=65536 Len=1460
62	1.47842300	192.168.80.137	192.168.80.155	TCP	1134	50361-5000 [PSH, ACK] Seq=2921 Ack=1 win=65536 Len=1080

Il y a donc trois paquets contenant des données envoyés par le client au serveur avec TCP et $1460 + 1460 + 1080 = 4000$ octets. Le champ "LEN" indique si des données ont été transmises ou non.

En ouvrant la fenêtre "Protocol Hierarchy" du menu "Statistics, on peut voir les informations sur les données envoyées par UDP et TCP.

Display filter: ip.src == 192.168.80.137 && ip.dst == 192.168.80.155							
Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes
Frame	100,00 %	7	100,00 %	4408	12,261	0	0
Ethernet	100,00 %	7	100,00 %	4408	12,261	0	0
Internet Protocol Version 4	100,00 %	7	100,00 %	4408	12,261	0	0
Transmission Control Protocol	100,00 %	7	100,00 %	4408	12,261	4	246
Data	42,86 %	3	94,42 %	4162	11,577	3	4162

On a 0 paquet UDP (contenant des données) et 3 Paquets TCP. En tout, il y a 4162 octets de données du client au serveur.

Serveur-Client

On applique le filtre “ip.src == 192.168.80.155 && ip.dst == 192.168.80.137” et on ouvre la fenêtre “Protocol Hierarchy” du menu “Statistics.

Display filter: ip.src == 192.168.80.155 && ip.dst == 192.168.80.137									
Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s	
Frame	100,00 %	4	100,00 %	308	0,000	0	0	0,000	
Ethernet	100,00 %	4	100,00 %	308	0,000	0	0	0,000	
Internet Protocol Version 4	100,00 %	4	100,00 %	308	0,000	0	0	0,000	
User Datagram Protocol	25,00 %	1	43,51 %	134	0,000	0	0	0,000	
Domain Name Service	25,00 %	1	43,51 %	134	0,000	1	134	0,000	
Transmission Control Protocol	75,00 %	3	56,49 %	174	0,000	3	174	0,000	

On remarque aucun octets contenant des données envoyés.

4)

Le mode secret 1 envoie le texte suivant en indiquant le protocole de transport le nom d'école polymtl et le sigle du cours INF3405.

- Follow TCP Stream (tcp.stream eq 10)

[illegible]

Stream Content

```

char peer0_0[] = {
0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3b, 0x15,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x3c, 0xc9, 0x11, 0xa4, 0x7b, 0x30, 0xe2, 0x97,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x28, 0x33, 0x11, 0x83, 0x7b, 0x30, 0xe2, 0x97,
0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x04, 0x04,
0x01, 0x00, 0x00, 0x00 };
char peer0_1[] = {
0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3b, 0x15,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x3c, 0xc9, 0x11, 0xa4, 0x7b, 0x30, 0xe2, 0x97,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x28, 0x33, 0x11, 0x83, 0x7b, 0x30, 0xe2, 0x97,
0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x04, 0x04,
0x00, 0x00, 0x00, 0x00 };
char peer0_2[] = {
0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3b, 0x15,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x3c, 0xc9, 0x11, 0xa4, 0x7b, 0x30, 0xe2, 0x97,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x28, 0x33, 0x11, 0x83, 0x7b, 0x30, 0xe2, 0x97,
0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x04, 0x04,
0x01, 0x00, 0x00, 0x00 };
char peer0_3[] = {
0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3b, 0x15,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x3c, 0xc9, 0x11, 0xa4, 0x7b, 0x30, 0xe2, 0x97,
0x20, 0x01, 0x00, 0x00, 0x5e, 0xf5, 0x79, 0xfb,
0x28, 0x33, 0x11, 0x83, 0x7b, 0x30, 0xe2, 0x97,
0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x04, 0x04,
0x01, 0x00, 0x00, 0x00 };

```

Également des tableaux similaires au contenu d'une image. Le client prend environ sept itérations pour faire ses transactions en se basant sur les captures précédentes.

Mode Secret 2

1)

Dans ce mode, on observe seulement, une connexion avec le protocole TCP. Le premier échange est le même que dans le premier mode avec un accusé de réception du serveur.

Filter:		ip.addr == 192.168.80.137 && ip.addr == 192.168.80.155		▼	Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info		
12	4.50256600	192.168.80.137	192.168.80.155	TCP	66	50439→5000 [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1		
13	4.50264200	192.168.80.155	192.168.80.137	TCP	66	5000→50439 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1		
14	4.50271800	192.168.80.137	192.168.80.155	TCP	60	50439→5000 [ACK] Seq=1 Ack=1 win=65536 Len=0		
15	4.50329600	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=1 Ack=1 win=65536 Len=40		
16	4.50334400	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=41 Ack=1 win=65536 Len=40		
17	4.50335700	192.168.80.155	192.168.80.137	TCP	54	5000→50439 [ACK] Seq=1 Ack=81 win=65536 Len=0		
18	4.50340500	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=81 Ack=1 win=65536 Len=40		
19	4.50340500	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=121 Ack=1 win=65536 Len=40		
20	4.50341900	192.168.80.155	192.168.80.137	TCP	54	5000→50439 [ACK] Seq=1 Ack=161 win=65536 Len=0		
21	4.50345100	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=161 Ack=1 win=65536 Len=40		
22	4.50352300	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=201 Ack=1 win=65536 Len=40		
23	4.50352300	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=241 Ack=1 win=65536 Len=40		
24	4.50353700	192.168.80.155	192.168.80.137	TCP	54	5000→50439 [ACK] Seq=1 Ack=281 win=65280 Len=0		
25	4.50357900	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=281 Ack=1 win=65536 Len=40		
26	4.50357900	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=321 Ack=1 win=65536 Len=40		
27	4.50359200	192.168.80.155	192.168.80.137	TCP	54	5000→50439 [ACK] Seq=1 Ack=361 win=65280 Len=0		
28	4.50362400	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=361 Ack=1 win=65536 Len=40		
29	4.50363400	192.168.80.155	192.168.80.137	TCP	54	5000→50439 [ACK] Seq=1 Ack=401 win=65280 Len=0		
30	4.50367000	192.168.80.137	192.168.80.155	TCP	94	50439→5000 [PSH, ACK] Seq=401 Ack=1 win=65536 Len=40		

2)

Les ports utilisés par TCP sont 50439 pour le client et 5000 pour le serveur comme on peut le voir sur la capture précédente.

3)

Client-Serveur

On observe 100 paquets contenant des données par TCP pour un total de 9400 octets envoyés. Aucune communication par UDP.

Serveur-Client

Display filter: ip.src == 192.168.80.155 && ip.dst == 192.168.80.137								
Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00 %	30	100,00 %	1632	2,997	0	0	0,000
Ethernet	100,00 %	30	100,00 %	1632	2,997	0	0	0,000
Internet Protocol Version 4	100,00 %	30	100,00 %	1632	2,997	0	0	0,000
Transmission Control Protocol	100,00 %	30	100,00 %	1632	2,997	30	1632	2,997

Il n'y a aucun paquets contenant des donnés donc pas d'octets de données envoyés par le serveur.

4)

[illegible]

Le mode secret 2 envoie le texte ci-dessus en indiquant le protocole de transport le nom d'école polymtl et le sigle du cours INF3405. Il prend environ plusieurs dizaines d'itération pour envoyer les données.

Mode Secret 3

1)

Dans ce mode, seul le protocole UDP est choisi. Pour les mêmes raisons que dans le mode 1, il n'y a pas d'échange "synchronize".

Filter: ip.addr == 192.168.80.137 && ip.addr == 192.168.80.155							Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info				
274	24.7346040	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=769d) [Reassembled in #282]				
275	24.7346050	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=769d) [Reassembled in #282]				
276	24.7346060	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=2960, ID=769d) [Reassembled in #282]				
277	24.7346060	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=4440, ID=769d) [Reassembled in #282]				
278	24.7346070	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=5920, ID=769d) [Reassembled in #282]				
279	24.7346070	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=7400, ID=769d) [Reassembled in #282]				
280	24.7346070	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=8880, ID=769d) [Reassembled in #282]				
281	24.7346080	192.168.80.137	192.168.80.155	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=10360, ID=769d) [Reassembled in #282]				
282	24.7346080	192.168.80.137	192.168.80.155	UDP	202	Source port: 64103 Destination port: 5010				
283	24.7346640	192.168.80.155	192.168.80.137	ICMP	590	Destination unreachable (Port unreachable)				
315	34.8541200	192.168.80.137	192.168.80.155	LLMNR	134	Standard query response 0x238a A 192.168.80.137 AAAA fe80::ac8b:2b48:f0bb:c27a				

2) Les ports utilisés par UDP sont 64103 pour le client et 5010 pour le serveur.

3) Client-serveur

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00 %	10	100,00 %	12904	1186,264	0	0	0,000
Ethernet	100,00 %	10	100,00 %	12904	1186,264	0	0	0,000
Internet Protocol Version 4	100,00 %	10	100,00 %	12904	1186,264	0	0	0,000
Data	80,00 %	8	93,86 %	12112	1113,456	8	12112	1113,456
User Datagram Protocol	10,00 %	1	1,57 %	202	18,570	0	0	0,000
Data	10,00 %	1	1,57 %	202	18,570	1	202	18,570
Internet Control Message Protocol	10,00 %	1	4,57 %	590	54,239	1	590	54,239

Un seul paquet de 202 octets a été envoyé par le client au serveur à travers UDP.

Serveur-client

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00 %	1	100,00 %	590	n.c.	0	0	n.c.
Ethernet	100,00 %	1	100,00 %	590	n.c.	0	0	n.c.
Internet Protocol Version 4	100,00 %	1	100,00 %	590	n.c.	0	0	n.c.
Internet Control Message Protocol	100,00 %	1	100,00 %	590	n.c.	1	590	n.c.

Aucun paquet de données du serveur n'a été envoyé par UDP ou TCP.

4)

Le mode secret 3 envoie le texte ci-dessus en indiquant le protocole de transport le nom d'école polymtl et le sigle du cours INF3405. Il le fait une itération principalement.

1)

Le mode 4 emploie uniquement le protocole UDP de la couche 4. Pour les mêmes raisons que dans le mode 1, il n'y a pas de synchronisation.

2) Les ports utilisés par UDP sont 63210 pour le client et 5010 pour le serveur.

3)

Client-Serveur

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00 %	301	100,00 %	24710	24,022	0	0	0,000
Ethernet	100,00 %	301	100,00 %	24710	24,022	0	0	0,000
Internet Protocol Version 4	100,00 %	301	100,00 %	24710	24,022	0	0	0,000
User Datagram Protocol	99,67 %	300	99,55 %	24600	23,915	0	0	0,000
Data	99,67 %	300	99,55 %	24600	23,915	300	24600	23,915
Internet Control Message Protocol	0,33 %	1	0,45 %	110	0,107	1	110	0,107

Serveur-Client

Il y a 3 paquets de données envoyés pour 282 octets de données.

4)

Le mode secret 4 envoie le texte ci-dessus en indiquant le protocole de transport le nom d'école polymtl et le sigle du cours INF3405. Cela s'effectue en quelques centaines d'itérations.

C) Analyse des performances et protocole TCP (2 points)

1) Comparez la performance des envois de données pour le mode 1 et le mode 2. Qu'est-ce qui diffère entre ces deux modes? Lequel est le plus performant selon vous et pourquoi? (0.5 point)

En comparant les deux modes, le mode 1 utilise UDP et TCP alors que le mode 2 n'utilise que TCP. En termes de performance, on peut observer que la vitesse d'envoi du mode 1 est supérieure (que ce soit du serveur au client ou du client au serveur) à celle du mode 2. On peut donc dire que mode 1 est plus performant.

2) Comparer la performance des envois de données pour le mode 3 et le mode 4. Qu'est-ce qui diffère entre ces deux modes? Lequel est le plus performant selon vous et pourquoi? (0.5 point)

Le plus performant est le mode 4, car il possède une vitesse de transmission supérieure et transmet plus de données.

3) Discutez de la fiabilité de chaque mode. Selon vous, quel(s) mode(s) est le plus fiable? (0.5 point)

Pour les classer en ordre, le mode le plus fiable est le mode 2 comme toute les communications se font par TCP. On peut savoir plus précisément si les paquets ont été reçus et les champs ne sont pas fixes comparativement à UDP. Ce mode est suivi par le mode 1 qui combine UDP et TCP. Ensuite, on pourrait dire que le mode 3 et 4 sont les moins fiables, car il n'utilisent que UDP (pas d'accusé de réception) et qui envoient un nombre conséquent de données avec ce seul protocole. Comme il s'agit de client et de serveur secrets, il est délicat de communiquer uniquement avec UDP.

4) Pour les modes secrets utilisant le protocole TCP, vous avez certainement remarqué à la fin de la communication un échange FIN, ACK. Expliquez en quoi consiste cet échange. (0.5 point)

Cet échange permet au client et au serveur d'indiquer que leur connexion est terminée. Le serveur confirme qu'il a bien reçu les données envoyées par le client et qu'il va terminer la connexion. Le client reçoit ce message et confirme qu'il a reçu le message de fin de la connexion et termine la communication.

référence:

- 1- <https://www.techwalla.com/articles/how-to-get-images-from-wireshark>
- 2- https://www.forum-avignon.org/sites/default/files/editeur/Interieur_HD.pdf, [en ligne], consulter le 13-04-2018