

Requirements:

Lib aenum:

- <https://pypi.python.org/pypi/aenum>
- <http://stackoverflow.com/questions/36932/how-can-i-represent-an-enum-in-python>

Peut s'installer avec la commande suivante:

```
pip install aenum
```

Utilisation

Le client se lance de la manière suivante:

```
python src/client.py -get/-put host port filename
```

Le serveur se lance de la manière suivante:

```
python src/server.py -p port -d repertoireAUtiliser
```

Sortie

Exemple de sortie en lançant le serveur d'abord, puis en effectuant deux requêtes client d'affilée.

Serveur:

```
$ sudo python src/server.py -p 69 -d serverFiles/  
Nb packets lost: 1. Efficiency: 0.999935  
Nb packets lost: 0. Efficiency: 1.000000
```

Client:

```
$ python src/client.py -put localhost 69 test7777k  
Nb packets lost: 0. Efficiency: 1.000000  
$ python src/client.py -get localhost 69 test7777k  
Nb packets lost: 1. Efficiency: 0.999935
```

Test du client / serveur

Déroulement du test

Afin de tester les programmes, un script **test.sh** est lancé avec la commande:

```
sudo bash test.sh
```

Il peut être lancé sans les droits admin si l'on indique un autre numéro de port que 69 dans le script.

Le script effectue la chose suivante:

1. Téléchargement du fichier test7777k auprès de hepia.infolibre.ch avec client.py.

Début du téléchargement

31	30.61737000	192.168.0.17	86.119.33.160	TFTP	60 Read Request, File: test7777k, Transfer type: octet
32	30.63734300	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 1
35	30.68000600	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 1
36	30.69708900	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 2
37	30.69720100	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 2
38	30.71170200	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 3
39	30.71181500	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 3
40	30.72375300	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 4
41	30.72387300	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 4
42	30.73770600	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 5
43	30.73782600	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 5

Le téléchargement d'un grand fichier permet de vérifier la gestion des éventuelles pertes de packets.

22378	180.17181200	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 11160
22379	180.17186800	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 11160
22380	180.18357800	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 11161
22381	180.18369200	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 11161
22382	181.18478500	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 11161
22383	181.18483300	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 11161
22384	181.19759000	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 11162
22385	181.19769300	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 11162

Fin du téléchargement

30872	240.67444800	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 15399
30873	240.67451600	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 15399
30874	240.68827100	86.119.33.160	192.168.0.17	TFTP	558 Data Packet, Block: 15400
30875	240.68834600	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 15400
30876	240.70256400	86.119.33.160	192.168.0.17	TFTP	60 Data Packet, Block: 15401 (last)
30877	240.70264400	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 15401
30878	240.70266600	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 15401
30879	241.70374900	192.168.0.17	86.119.33.160	TFTP	46 Acknowledgement, Block: 15401

2. Vérification checksum du fichier téléchargé. Il doit correspondre à a60814a887e267e2412c268c549be5c0
3. Lancement du serveur local
4. Upload du fichier 7777k vers le serveur local.

32	211.17135000	127.0.0.1	127.0.0.1	TFTP	60 Write Request, File: test7777k, Transfer type: octet
33	212.17257800	127.0.0.1	127.0.0.1	TFTP	60 Write Request, File: test7777k, Transfer type: octet
34	212.17276500	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 0
35	212.17288400	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1
36	212.17296400	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 1
37	212.17302300	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 2
38	212.17309100	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 2
39	212.17313700	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 3
40	212.17320500	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 3
30831	212.61900400	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 15399
30832	212.61901900	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15399
30833	212.61903200	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 15400
30834	212.61904800	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15400
30835	212.61906200	127.0.0.1	127.0.0.1	TFTP	46 Data Packet, Block: 15401 (last)
30836	212.61909100	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
30837	212.61909700	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
30838	212.64258000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30839	213.62017300	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401

5. Suppression du fichier 7777k récupéré auprès de hepia.infolibre (pour être sûr de faire ensuite le checksum du bon fichier)
6. Téléchargement du fichier 7777k depuis le serveur local

30841	214.64496700	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30842	214.64512200	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1
30843	214.64523900	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 1
30844	214.64529100	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 2
30845	214.64530900	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 2
30846	214.64535700	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 3
30847	214.64541000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 3
30848	214.64546200	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 4
30849	214.64551600	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 4
61638	215.03642900	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 15399
61639	215.03644300	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15399
61640	215.03645600	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 15400
61641	215.03647500	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15400
61642	215.03648700	127.0.0.1	127.0.0.1	TFTP	46 Data Packet, Block: 15401 (last)
61643	215.03651400	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
61644	215.03652000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
61645	216.03709200	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401

- Vérification du checksum du fichier téléchargé depuis le serveur local. Il doit à nouveau correspondre à a60814a887e267e2412c268c549be5c0

Sortie:

```
$ sudo ./test.sh
downloading test7777k from hepia.infolibre.ch... This will take 5 minutes
unexpected block num 11161
unexpected block num 11577
Nb packets lost: 3. Efficiency: 0.999805
downloading finished.
md5 sum after downloading from hepia.infolibre.ch:
a60814a887e267e2412c268c549be5c0 test7777k
starting server
server started. PID : 4143
uploading file to local server...
Nb packets lost: 0. Efficiency: 1.000000
removing first downloaded file
downloading file from local server
Nb packets lost: 1. Efficiency: 0.999935
Nb packets lost: 0. Efficiency: 1.000000
Nb packets lost: 1. Efficiency: 0.999935
md5 sum after downloading from hepia.infolibre.ch:
a60814a887e267e2412c268c549be5c0 test7777k
stopping server
removing files
```

Vérification du changement de port

A travers wireshark surveillant l'interface locale, nous avons également vérifié que notre serveur notre client changent bien de port une fois la connexion initialisée:

Connexion avec PUT Request

Le client envoie sur le port 69.

29	208.805536000	127.0.0.1	127.0.0.1	TFTP	60 Write Request, File: test7777k, Transfer type: octet
30	208.805759000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 0
31	208.805901000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1

▶Frame 29: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 10315 (10315), Dst Port: tftp (69)
 ▼Trivial File Transfer Protocol
 [DESTINATION File: test7777k]
 Opcode: Write Request (2)
 DESTINATION File: test7777k
 Type: octet

Le serveur répond en ayant changé de port

29	208.805536000	127.0.0.1	127.0.0.1	TFTP	60 Write Request, File: test7777k, Transfer type: octet
30	208.805759000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 0
31	208.805901000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1

▶Frame 30: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 34562 (34562), Dst Port: 10315 (10315)
 ▼Trivial File Transfer Protocol
 [DESTINATION File: test7777k]
 Opcode: Acknowledgement (4)
 Block: 0

Le client continue la connexion sur le nouveau port

29	208.805536000	127.0.0.1	127.0.0.1	TFTP	60 Write Request, File: test7777k, Transfer type: octet
30	208.805759000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 0
31	208.805901000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1

▶Frame 31: 558 bytes on wire (4464 bits), 558 bytes captured (4464 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 10315 (10315), Dst Port: 34562 (34562)
 ▼Trivial File Transfer Protocol
 [DESTINATION File: test7777k]
 Opcode: Data Packet (3)
 Block: 1
 ▶Data (512 bytes)

Multiples GET request

Lorsque le serveur met un peu de temps à répondre, on peut confirmer que le client tente une requête avec un nouveau port à chaque tentative:

30835	210.188931000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
30836	210.211525000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30837	211.212716000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30838	212.213944000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30839	212.214137000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1
30840	212.214272000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 1
30841	212.214335000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 2

▶Frame 30836: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 19800 (19800), Dst Port: tftp (69)
 ▼Trivial File Transfer Protocol
 [Source File: test7777k]
 Opcode: Read Request (1)
 Source File: test7777k
 Type: octet

30835	210.188931000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
30836	210.211525000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30837	211.212716000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30838	212.213944000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30839	212.214137000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1
30840	212.214272000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 1
30841	212.214335000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 2

▶Frame 30837: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 17568 (17568), Dst Port: tftp (69)
 ▼Trivial File Transfer Protocol
 [Source File: test7777k]
 Opcode: Read Request (1)
 Source File: test7777k
 Type: octet

Connexion avec GET request

Enfin, nous vérifions que le changement de port se fasse correctement avec les Get Request.

30835	210.188931000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
30836	210.211525000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30837	211.212716000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30838	212.213944000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30839	212.214137000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1
30840	212.214272000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 1
30841	212.214335000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 2

▶Frame 30838: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 33413 (33413), Dst Port: tftp (69)
 ▼Trivial File Transfer Protocol
 [Source File: test7777k]
 Opcode: Read Request (1)
 Source File: test7777k
 Type: octet

30835	210.188931000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
30836	210.211525000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30837	211.212716000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30838	212.213944000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30839	212.214137000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1
30840	212.214272000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 1
30841	212.214335000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 2

▶Frame 30839: 558 bytes on wire (4464 bits), 558 bytes captured (4464 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 11799 (11799), Dst Port: 33413 (33413)
 ▼Trivial File Transfer Protocol
 [Source File: test7777k]
 Opcode: Data Packet (3)
 Block: 1
 ▶Data (512 bytes)

30835	210.188931000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 15401
30836	210.211525000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30837	211.212716000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30838	212.213944000	127.0.0.1	127.0.0.1	TFTP	60 Read Request, File: test7777k, Transfer type: octet
30839	212.214137000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 1
30840	212.214272000	127.0.0.1	127.0.0.1	TFTP	46 Acknowledgement, Block: 1
30841	212.214335000	127.0.0.1	127.0.0.1	TFTP	558 Data Packet, Block: 2

▶Frame 30840: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
 ▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 ▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
 ▶User Datagram Protocol, Src Port: 33413 (33413), Dst Port: 11799 (11799)
 ▼Trivial File Transfer Protocol
 [Source File: test7777k]
 Opcode: Acknowledgement (4)
 Block: 1

