



جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

السنة الخامسة

مشروع برمجة الشبكات بعنوان

TCP server and client to transfer files and
ensure the security of the link

إعداد الطلاب

ولاء محسن جبور

مايا سمير غانم

عمار بركات ملحم

بإشراف

د. مهند عيسى

الملخص

بروتوكول التحكم بالنقل أو بروتوكول (Transmission Control Protocol TCP) هو أحد البروتوكولات الأساسية في حزمة بروتوكولات الإنترنت، موصوف بالوثيقة (RFC 793) ، ويُؤمن نقلًا موثوقًا خاليًا من الأخطاء لدفق من البايتات بين مُضيفين يتصلان مع بعضهما البعض عبر شبكة تدعم بروتوكول الإنترنت في هذا المشروع سوف نطبق مخدم وزبون يستخدمان البروتوكول TCP للاتصال ونقل البيانات، يحاكي المخدم في حالتنا موقع تحميل كتب أو ملفات والزبون هو زائر للموقع ويريد تحميل أحد الكتب، ثم التأكد من أن الوصلة آمنة باستخدام التجزئة على محتوى الكتاب الواصل والتجزئة الذي وصلته من المخدم.

الكلمات المفتاحية

بروتوكول التحكم في الإرسال

مخدم

زبون

بايثون

التجزئة

بروتوكول TCP

يعد بروتوكول التحكم في الإرسال (TCP) أحد البروتوكولات الرئيسية لمجموعة بروتوكولات الإنترنت. نشأت في التنفيذ الأولي للشبكة التي استكملت بروتوكول الإنترنت (IP). لذلك ، يشار إلى المجموعة بأكملها باسم TCP / IP. يوفر بروتوكول TCP تسليمًا موثوقًا ومرتبًا ومدققًا للأخطاء لتيار من الثمانيات (البايتات) بين التطبيقات التي تعمل على الأجهزة المضيفة التي تتصل عبر شبكة IP. تعتمد تطبيقات الإنترنت الرئيسية مثل World Wide Web والبريد الإلكتروني والإدارة عن بُعد ونقل الملفات على TCP ، وهو جزء من طبقة النقل في مجموعة TCP / IP. غالبًا ما يتم تشغيل SSL / TLS أعلى TCP.

بروتوكول TCP موجه نحو الاتصال ، ويتم إنشاء اتصال بين العميل والخادم قبل إرسال البيانات. يجب أن يستمع الخادم (مفتوح غير نشط) لطلبات الاتصال من العملاء قبل إنشاء الاتصال. يضيف المصافحة ثلاثية الاتجاهات (المفتوحة النشطة) وإعادة الإرسال واكتشاف الأخطاء إلى الموثوقية ولكنها تطيل الكمون. قد تستخدم التطبيقات التي لا تتطلب خدمة دفع بيانات موثوقة بروتوكول مخطط بيانات المستخدم (UDP) ، الذي يوفر خدمة مخطط بيانات بدون اتصال تعطي الأولوية للوقت على الموثوقية. يستخدم TCP تجنب ازدحام الشبكة. ومع ذلك ، هناك نقاط ضعف لبرنامج التعاون الفني بما في ذلك رفض الخدمة ، واختراق الاتصال ، وفيتو TCP ، وإعادة ضبط الهجوم. بالنسبة لأمن الشبكة والمراقبة والتصحيح ، يمكن اعتراض حركة مرور TCP وتسجيلها باستخدام الشم.

على الرغم من أن بروتوكول TCP هو بروتوكول معقد ، إلا أن تشغيله الأساسي لم يتغير بشكل ملحوظ منذ تحديده لأول مرة. لا يزال بروتوكول TCP يستخدم في الغالب للويب ، أي لبروتوكول HTTP ، ولاحقًا 2 / HTTP ، بينما لا يتم استخدامه بواسطة أحدث معيار 3 / HTTP.

TCP هي خدمة توصيل دفع موثوق بها تضمن أن جميع البايتات المستلمة ستكون متطابقة وفي نفس الترتيب مثل تلك المرسل. نظرًا لأن نقل الحزم بواسطة العديد من الشبكات غير موثوق به ، فإن TCP يحقق ذلك باستخدام تقنية تعرف باسم الإقرار الإيجابي بإعادة الإرسال. يتطلب ذلك من جهاز الاستقبال الرد برسالة إقرار أثناء استقبال البيانات. يحتفظ المرسل بسجل لكل حزمة

يرسلها ويحافظ على جهاز ضبط الوقت من وقت إرسال الحزمة. يعيد المرسل إرسال حزمة إذا انتهى المؤقت قبل استلام الإشعار. المؤقت مطلوب في حالة ضياع الحزمة أو تلفها.

يتكون مقطع TCP من رأس مقطع وقسم بيانات. يحتوي رأس المقطع على ١٠ حقول إلزامية ، وحقل ملحق اختياري (خيارات ، خلفية وردية في الجدول). يتبع قسم البيانات العنوان وهو بيانات الحمولة المنقولة للتطبيق. طول مقطع البيانات غير محدد في رأس المقطع ؛ يمكن حسابها بطرح الطول المدمج لرأس المقطع ورأس IP من إجمالي طول مخطط بيانات IP المحدد في رأس IP.

TCP segment header																																	
Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0		N S	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size																
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if data offset > 5. Padded at the end with "0" bytes if necessary.)																															
...																															

صورة ١: مقطع TCP

منفذ المصدر (١٦ بت)

يحدد منفذ الإرسال.

منفذ الوجهة (١٦ بت)

يحدد منفذ الاستقبال.

الرقم التسلسلي (٣٢ بت)

له دور مزدوج:

إذا تم تعيين علامة SYN (١) ، فهذا هو رقم التسلسل الأولي. رقم التسلسل لبايت البيانات الأول الفعلي والرقم المعترف به في ACK المقابل هما رقم التسلسل زائد ١.

إذا كانت علامة SYN واضحة (٠) ، فهذا هو رقم التسلسل المتراكم لبايت البيانات الأول لهذا المقطع للدورة الحالية.

رقم الاعتراف (٣٢ بت)

إذا تم تعيين علامة ACK ، فإن قيمة هذا الحقل هي رقم التسلسل التالي الذي يتوقعه مرسل ACK. يقر هذا باستلام جميع البايتات السابقة (إن وجدت). يقر ACK الأول المرسل من كل طرف برقم التسلسل الأولي للطرف الآخر نفسه ، ولكن لا توجد بيانات.

إزاحة البيانات (٤ بتات)

يحدد حجم رأس TCP بكلمات ٣٢ بت. الحد الأدنى لحجم العنوان هو ٥ كلمات والحد الأقصى هو ١٥ كلمة ، مما يعطي الحد الأدنى للحجم ٢٠ بايت والحد الأقصى ٦٠ بايت ، مما يسمح بما يصل إلى ٤٠ بايت من الخيارات في الرأس. يحصل هذا الحقل على اسمه من حقيقة أنه هو أيضًا الإزاحة من بداية مقطع TCP إلى البيانات الفعلية.

محجوزة (٣ بتات)

للاستخدام المستقبلي ويجب ضبطه على صفر.

الأعلام (٩ بت)

يحتوي على ٩ أعلام ١ بت (بتات التحكم) على النحو التالي:

NS (١ بت): ECN-nonce - الحماية من الإخفاء [a]

CWR (١ بت): يتم تعيين إشارة الاختزال (CWR) لنافذة الازدحام من قبل المضيف المرسل للإشارة إلى أنها تلقت قطعة TCP مع مجموعة علامات ECE واستجابت في آلية التحكم في الازدحام. [b]

ECE (١ بت): ECN-Echo له دور مزدوج ، اعتمادًا على قيمة العلم SYN. يشير إلى:

إذا تم تعيين علامة SYN (١) ، فهذا يعني أن نظير TCP قادر على ECN.

إذا كان علم SYN واضحًا (٠) ، فإن حزمة مع مجموعة إشارة من ذوي الخبرة في الازدحام (ECN = 11) في رأس IP تم استلامها أثناء الإرسال العادي. [b] وهذا بمثابة إشارة إلى ازدحام الشبكة (أو الازدحام الوشيك) إلى مرسل TCP.

URG (١ بت): يشير إلى أن مجال المؤشر العاجل مهم

ACK (١ بت): يشير إلى أن حقل الإقرار مهم. يجب أن تحتوي كل الحزم بعد الحزمة SYN الأولية المرسل من قبل العميل على مجموعة العلم هذه.

PSH (١ بت): وظيفة الدفع. يطلب دفع البيانات المخزنة إلى التطبيق المتلقي.

RST (١ بت): إعادة تعيين الاتصال

SYN (١ بت): مزامنة أرقام التسلسل. يجب أن تكون مجموعة العلامات هذه هي الحزمة الأولى فقط المرسل من كل طرف. بعض العلامات والحقول الأخرى تغير المعنى بناءً على هذه العلامة ، وبعضها صالح فقط عندما يتم تعيينه ، والبعض الآخر عندما يكون واضحًا.

FIN (١ بت): الحزمة الأخيرة من المرسل

حجم النافذة (١٦ بت)

حجم نافذة الاستلام ، الذي يحدد عدد وحدات حجم النافذة [C] التي يرغب المرسل من هذا الجزء في تلقيها حاليًا. [د] (انظر § التحكم في التدفق و § قياس النافذة).

المجموع الاختباري (١٦ بت)

يتم استخدام حقل المجموع الاختباري ١٦ بت لفحص أخطاء رأس TCP والحمولة والرأس الزائف لعنوان IP. يتكون الرأس الزائف من عنوان IP المصدر وعنوان IP الوجهة ورقم بروتوكول بروتوكول TCP (٦) وطول رؤوس TCP وحمولة (بالبايت).

مؤشر عاجل (١٦ بت)

إذا تم تعيين علامة URG ، فإن هذا الحقل المكون من ١٦ بت هو إزاحة من رقم التسلسل الذي يشير إلى آخر بايت عاجل للبيانات.

الخيارات (متغير ٠-٣٢٠ بت ، بوحدة من ٣٢ بت)

تأسيس الاتصال

لإنشاء اتصال، يستخدم بروتوكول TCP مصافحة ثلاثية. قبل أن يحاول العميل الاتصال بخادم ، يجب أن يلتزم الخادم أولاً ويستمتع إلى منفذ لفتحه من أجل الاتصالات: وهذا ما يسمى بفتح سلبي. بمجرد إنشاء الفتح السلبي ، يمكن للعميل بدء فتح نشط. لإنشاء اتصال ، تحدث مصافحة ثلاثية.

الجزء العملي:

كود المخدم:

```
from socket import *
import socket
import threading
import hashlib

IP = "127.0.0.1"
PORT = 7788
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((IP, PORT))
server_socket.listen(5)
print("Server Running....")
list_of_book = {"1" : "file1.txt", "2" : "file2.txt", "3" : "file3.txt"}
def myThread(client_socket, client_address):
    while True:
        msg_to_client = "Hello! Please select the book you want to download: \n"
        msg_to_client += "(1) file1.txt\n"
        msg_to_client += "(2) file2.txt\n"
        msg_to_client += "(3) file3.txt\n"
        msg_to_client += "or \"exit\" to exit\n"
        print(msg_to_client)
        client_socket.send(msg_to_client.encode())
        book_number = client_socket.recv(1024).decode()
        print(book_number)
        words = ""
```

```

if book_number == "exit":
    break
try:
    file = open("books/" + list_of_book[book_number], "r")

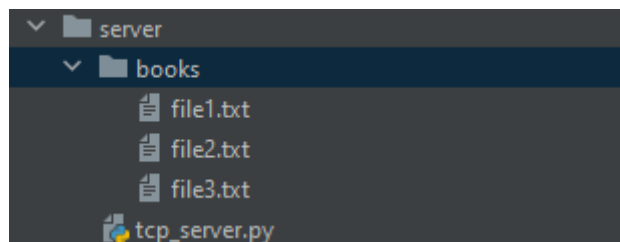
    for row in file:
        words_of_row = row.split()
        for word in words_of_row:
            words += word + " "
        words += "\n"
    print(list_of_book[book_number])
    print(words)
    md5_words = hashlib.md5(words.encode('utf-
8')).hexdigest()
    client_socket.send(words.encode())
    client_socket.send(md5_words.encode())
    is_secure = client_socket.recv(1024).decode()
    print(is_secure)
except socket.error as e:
    print(e)
    client_socket.send(str(e).encode())
except KeyError as e:
    client_socket.send("This file is not exist.".encode())

(client_socket , client_address) = server_socket.accept()
th =
threading.Thread(target=myThread, args=(client_socket, client_address))
th.start()

```

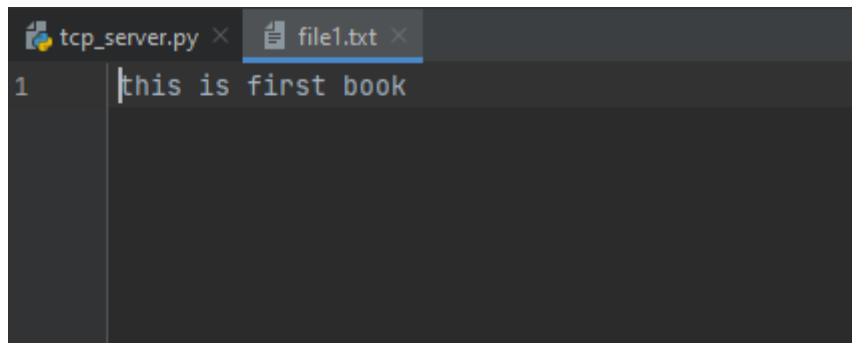
الكود السابق مكون من أربع أفكار رئيسية:

- الجزء الأول: تعريف المكتبات اللازمة وتهيئة الاتصال واستماع المخدم على المنفذ 7788 وتعريف أسماء الملفات الموجودة في المخدم حيث يكون مجلد المخدم كالتالي:



صورة ٢: محتوى مجلد المخدم

محتوى الكتاب الأول هو:



صورة ٣: محتوى الكتاب الأول

- الجزء الثاني (المكتوب بنهاية الكود): انتظار اتصال زبون وتهيئة الـ thread ليعمل مع أكثر من زبون بنفس الوقت حيث عند اتصال أي زبون سيخدمه عن طريق التابع `.myThread`.
- الجزء الثالث: بعد اتصال زبون سيخدمه المخدم بالتابع `myThread` بأن يطلب من المستخدم بإدخال رقم يدل على رقم الكتاب الذي يريد تحميله وينتظر الرد منه، بعد الحصول على رقم الكتاب يتم فتح الملف للقراءة، ثم تتم قراءته سطر سطر - كلمة كلمة، ويتم تخزين النتيجة (محتوى الكتاب) في المتحول `words`.
- الجزء الرابع: يقوم الكود بعمل التجزئة لمحتوى الكتاب وإرساله إلى المستخدم مع محتوى الكتاب، ثم يعيد الطلب بإدخال رقم كتاب جديد، وإذا كان الرد `exit` يقطع الاتصال مع هذا الزبون

كود الزبون:

```
from socket import *
import hashlib
import socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

IP = "127.0.0.1"
PORT = 7788
client_socket.connect((IP, PORT))
cnt = 1

while True :
    msg_from_server = client_socket.recv(1024).decode()
```

```

book_number = input(msg_from_server)

if book_number == "exit":
    break
client_socket.send(book_number.encode())
book_content = client_socket.recv(2048).decode()
hash_book_content_from_server =
client_socket.recv(1024).decode()
hash_book_content = md5_words =
hashlib.md5(book_content.encode('utf-8')).hexdigest()
print(book_content)
print("hash_book_content_from_server: "
+hash_book_content_from_server)
print("hash_book_content: " + hash_book_content)

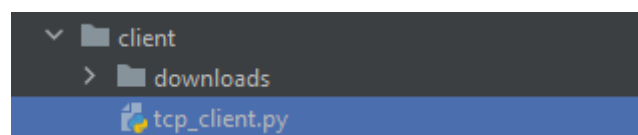
if(hash_book_content == hash_book_content_from_server):
    print("The connection is secure")
    client_socket.send("The connection is secure".encode())
else:
    print("The connection is not secure")
    client_socket.send("The connection is not secure".encode())
    break

try:
    file = open("downloads/download"+str(cnt)+".txt", "w")
    file.write(book_content)
    cnt += 1
    file.close()
except error as e:
    print(e)

client_socket.close()

```

محتوى مجلد الزبون:



صورة ٤: محتوى مجلد الزبون

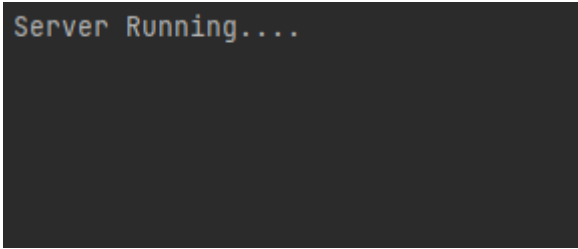
يمكن تقسيم الكود السابق إلى ثلاث أفكار رئيسية:

- الجزء الأول: لتهيئة الاتصال واستدعاء المكتبات الجاهزة.
- الجزء الثاني: استقبال رسالة من المخدم ليُدله ماذا عليه أن يفعل (يطلب منه إدخال رقم الكتاب)، ثم ندخل رقم الكتاب ونرسله للمخدم وننتظر الرد.

- الجزء الثالث: بعد استقبال الرد (محتوى الكتاب، التجزئة على محتوى الكتاب) نقوم بعمل تجزئة على محتوى الكتاب الوارد ونقارنه مع التجزئة الواردة من المستخدم.

التنفيذ:

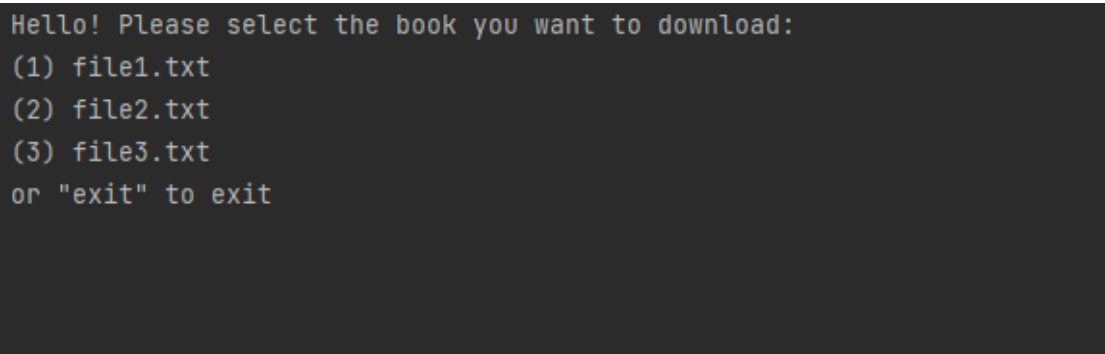
تنفيذ المخدم:



```
Server Running....
```

صورة ٥: نتيجة تنفيذ المخدم

تنفيذ الزبون:



```
Hello! Please select the book you want to download:  
(1) file1.txt  
(2) file2.txt  
(3) file3.txt  
or "exit" to exit
```

صورة ٦: نتيجة تنفيذ الزبون

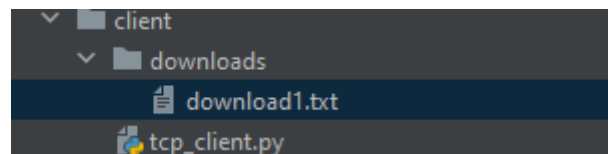
ندخل الرقم ١ مثلاً

```
Hello! Please select the book you want to download:
(1) file1.txt
(2) file2.txt
(3) file3.txt
or "exit" to exit
1
hash_book_content_from_server: 6e0e1590f308848454506f271fad5709
hash_book_content: 6e0e1590f308848454506f271fad5709
The connection is secure
Hello! Please select the book you want to download:
(1) file1.txt
(2) file2.txt
(3) file3.txt
or "exit" to exit
```

صورة ٧: نتيجة تنفيذ البرون بعد إدخال رقم الكتب

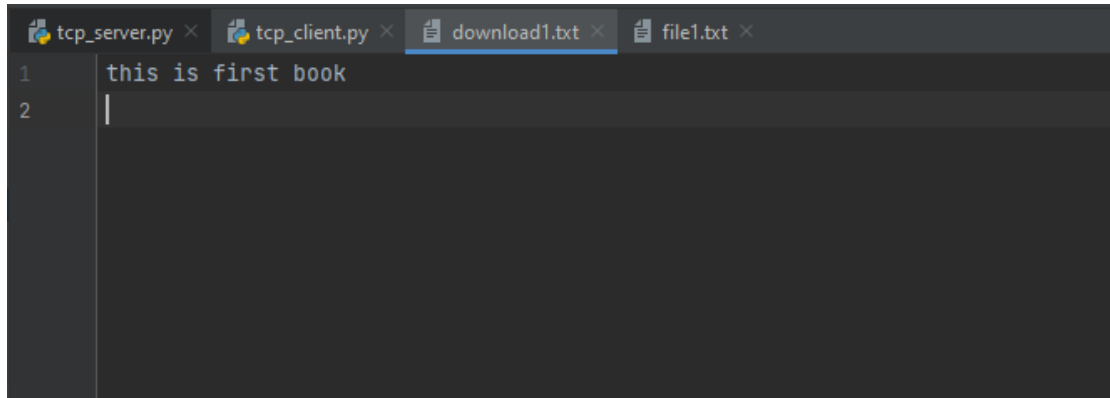
نلاحظ أن الاتصال موثوق وآمن لأننا حصلنا على نفس التجزئة

ونلاحظ بناء ملف جديد باسم download1 في مجلد الزبون:



صورة ٨: محتوى مجلد الزبون

ومحتواه:



```
1 this is first book
2 |
```

صورة ٩: محتوى الكتاب بعد التحميل

المراجع:

[1]: <https://searchnetworking.techtarget.com/definition/TCP>

[2]: <https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:transporting-packets/a/transmission-control-protocol--tcp>

[3]: <https://realpython.com/python-sockets/#tcp-sockets>

انتهى المشروع