**ΣΧΕΔΙΑΣΜΟΣ ΔΙΑΔΙΚΤΥΑΚΩΝ ΠΡΩΤΟΚΟΛΛΩΝ**

*Web Server*

Ο server που τροποποιήσαμε είναι o built-in HTTPServer (http.server.py) της python3. Τα sockets δημιουργούνται μέσω του module socketserver. Για τους servers των 2 πρωτοκόλλων (SCTP, DCCP) καλείται η κλάση TCPServer και το μόνο που αλλάζει είναι οι παράμετροι που καλούνται στη μέθοδο socket.socket που χρησιμοποιείται για την δημιουργία των sockets (στο παραδοτέο socketserver.py γραμμή: 455)

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

*Web Client*

Όσον αφορά τη μεριά του client, δημιουργήσαμε μια μέθοδο για το κάθε πρωτόκολλο, create\_dccp\_connection και create\_sctp\_connection στο αρχείο socket.py. Αυτές καλούνται στο αρχείο client.py για τη δημιουργία των sockets(συνδέσεων). Οι παράμετροι της συνάρτησης socket, παρακάτω, είναι :

af = socket family

socktype = SOCK\_STREAM, socket type for protocol

132 = protocol flag, protocol number for SCTP (IPPROTO\_SCTP)

Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Αντίστοιχα:

af = socket family

6 = SOCK\_DCCP, socket type for protocol

33 = protocol flag, protocol number for DCCP

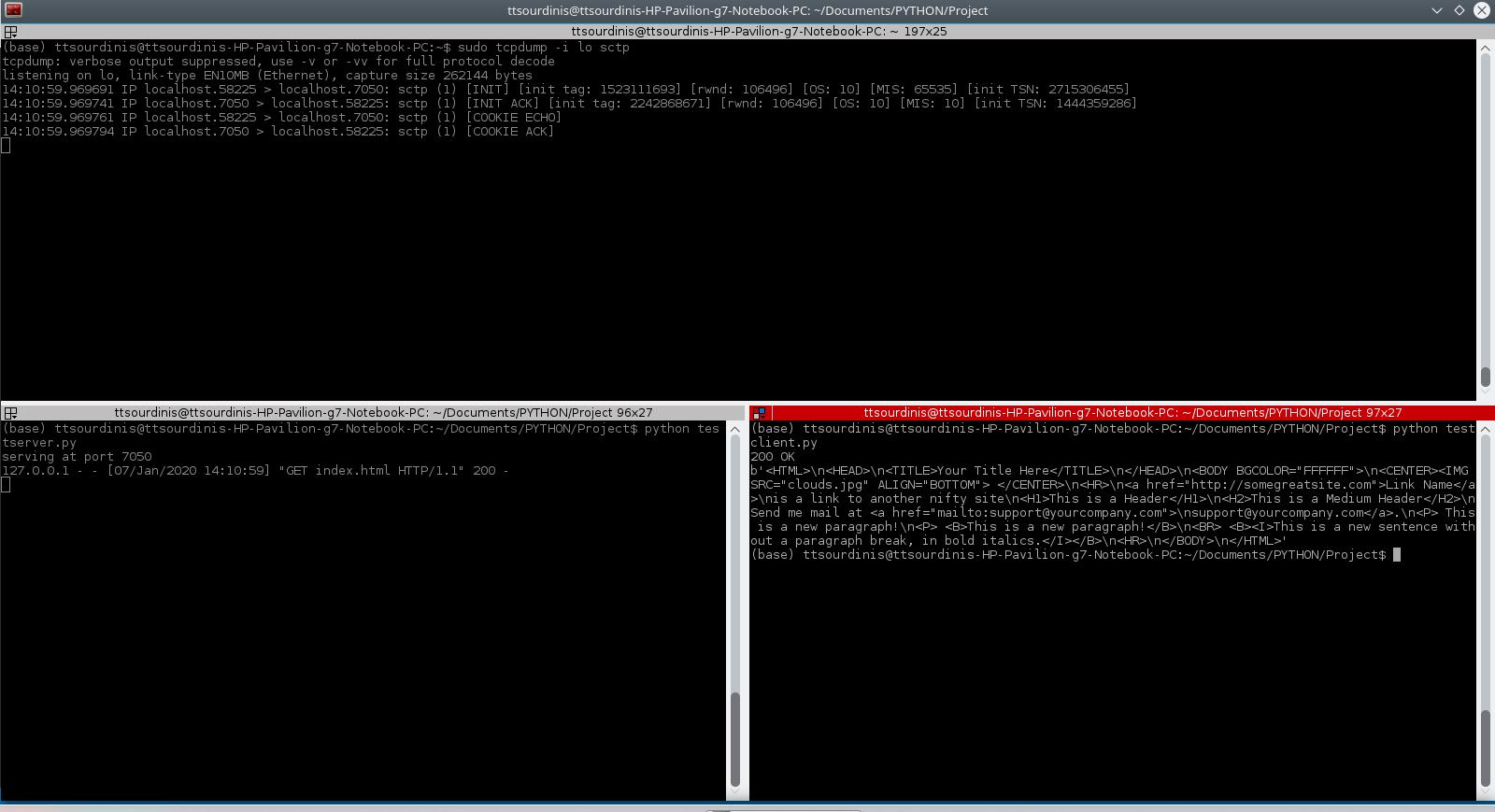
Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης

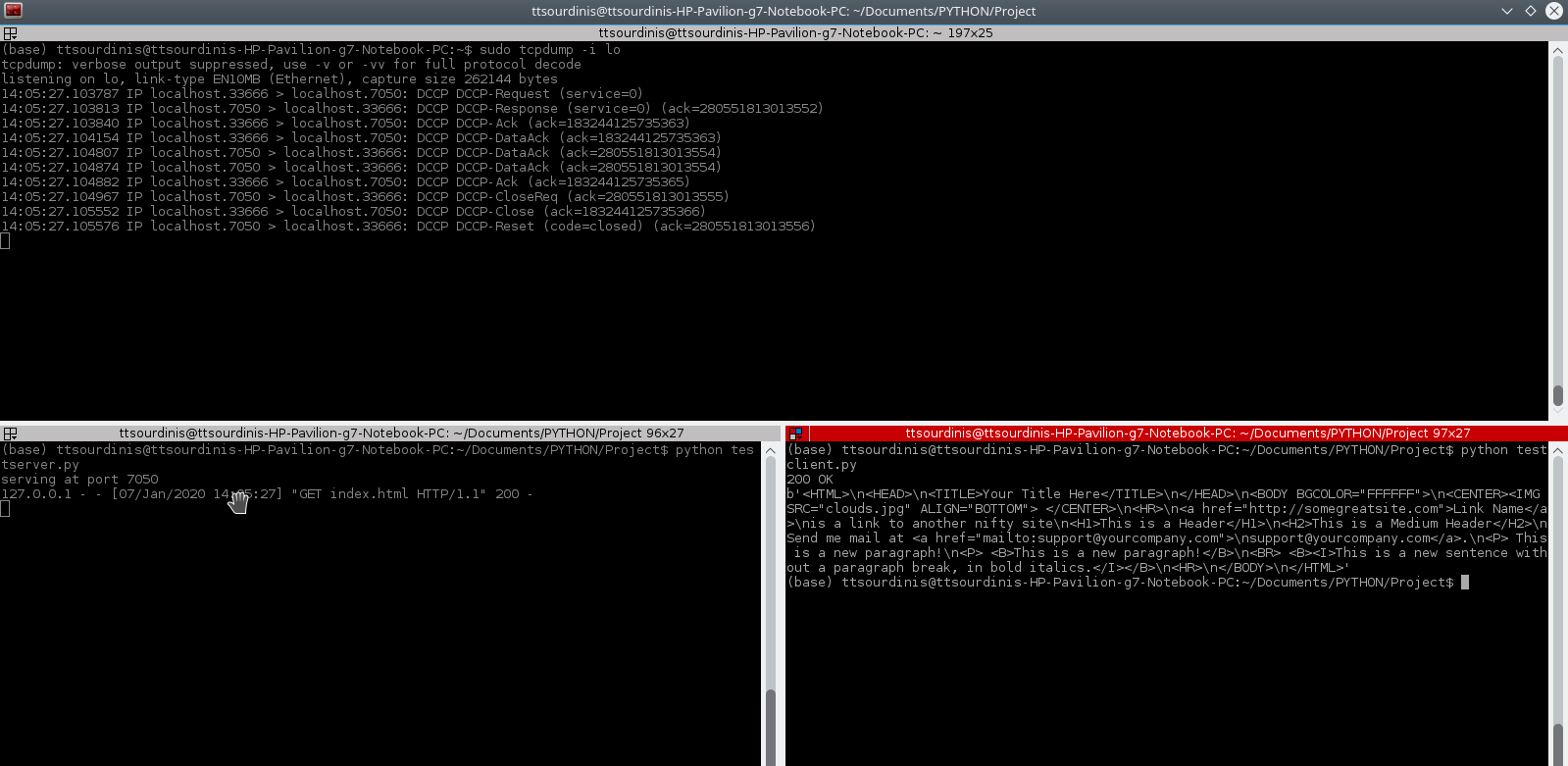
Περιγραφή που δημιουργήθηκε αυτόματα

*Execution Analysis*

Ο web server καλείται μέσω του αρχείου testserver.py ενώ ο web client μέσω του αρχείου testclient.py .

Για το test δημιουργήσαμε ένα html αρχείο (index.html) το οποίο γίνεται «ανεβαίνει» στον server και στη συνέχεια γίνεται serve στον client μέσω ενός get request από το testclient.py . Παρατίθενται τα στιγμιότυπα εκτέλεσης της επικοινωνίας από τα terminals αλλά και από το WireShark, αρχικά από SCTP και έπειτα από DCCP κίνηση.





*WireShark*

