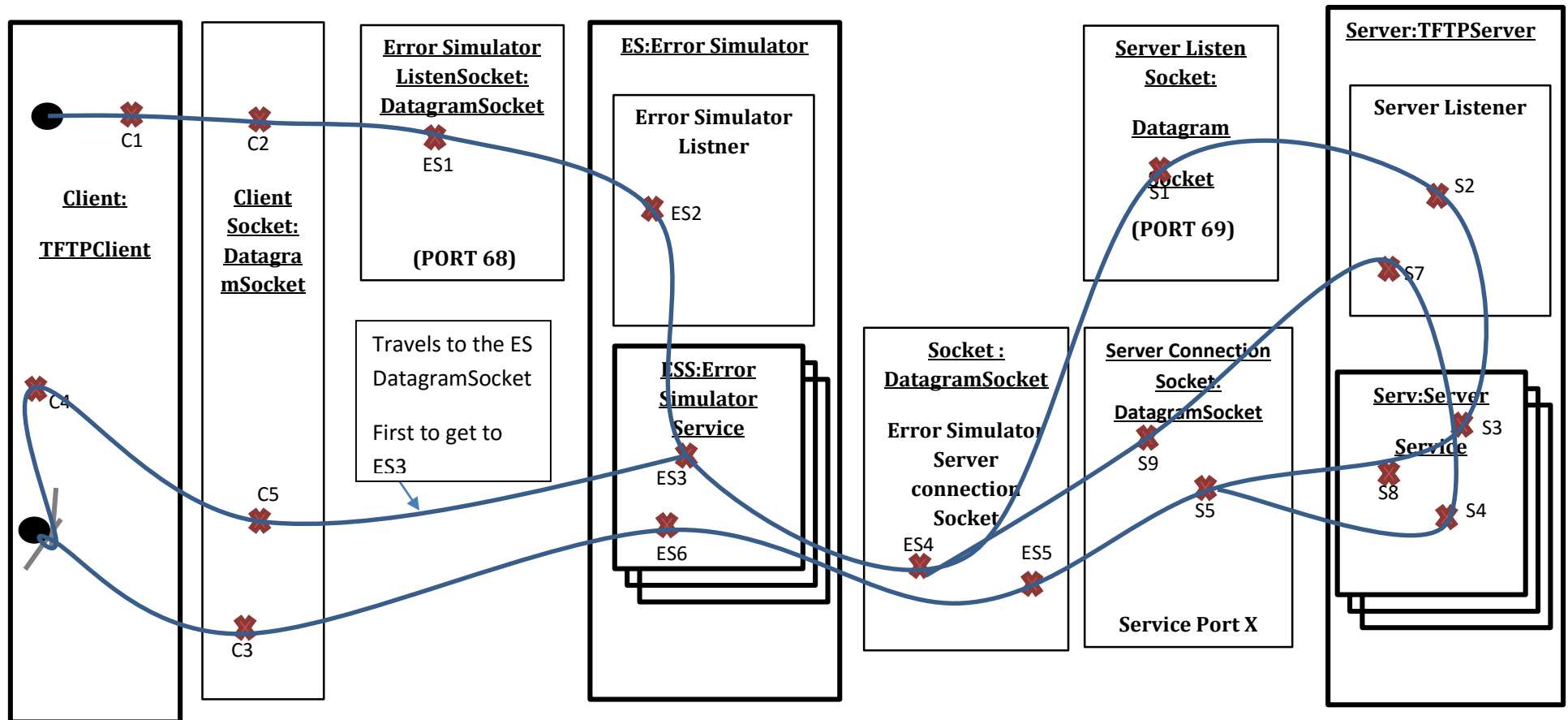


UCM: Error Free Write request



Notes:

- Here, C4->C5 send preceding packets to the server
- Once ES is up, all proceeding packets for the current file transfer go toward the same port

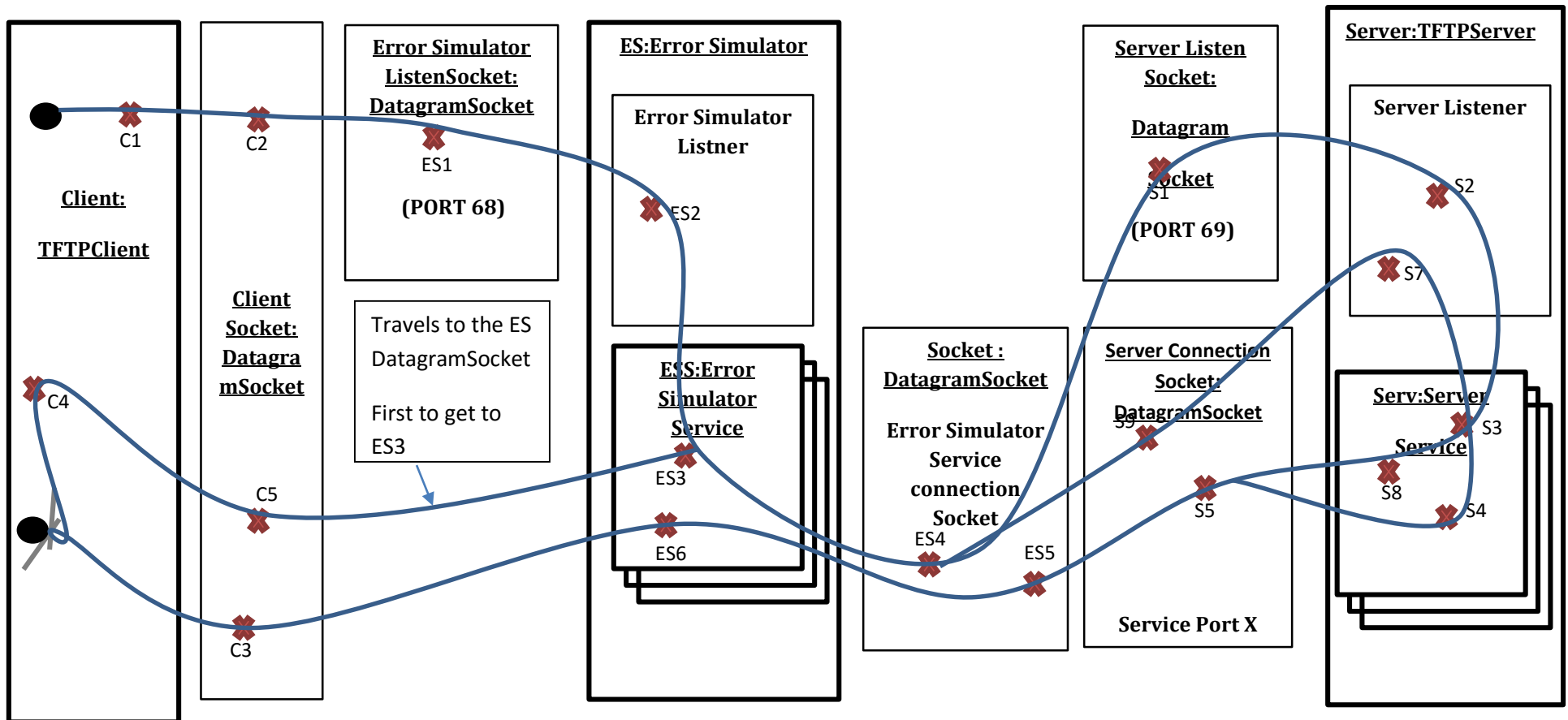
C1 : Create write request packet
C2: Send write request packet
C3: Receive acknowledge packet
C4: Prepare Data blocks
C5: Send data block

ES1: Receive write request Packet
ES2: New Error service thread is created and started
ES3: Rewrite packet address for server . (Port 69 on first packet, then forward to Server Service socket X)
ES4: Send off packet
ES5: Receive server response packet
ES6: Rewrite address for client

S1: Receive write request packet
S2: New Server service thread is created and started
S3: Create Acknowledge packet acknowledging that the write request packet was received
S4: Check for packet formatting errors
S5: Send Acknowledge packet and started

S6: Receive data block
S7: New Server service thread is created
S8: Checks if the received data block is the correct data block
S9: Send acknowledgement packet acknowledging that the data block was received

UCM: Error Free Read request



Notes:

- Here, C4->C5 send preceding packets to the server
- Once ES is up, all proceeding packets for the current file transfer go toward the same port

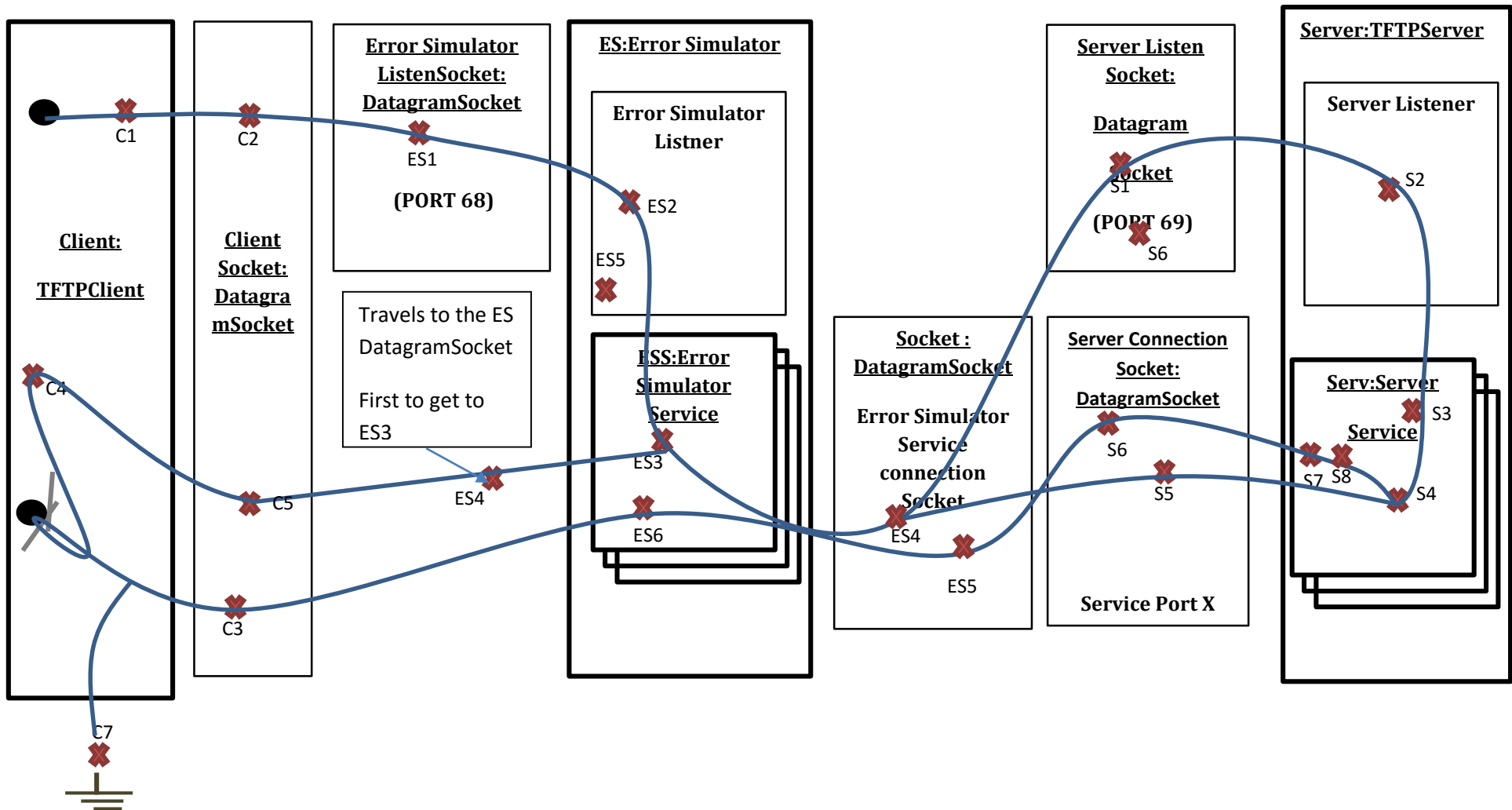
C1 : Create read request packet
C2: Send read request packet
C3: Receive acknowledge packet
C4: Prepare Data blocks
C5: Send data block

ES1: Receive write request Packet
ES2: New Error service thread is created and started
ES3: Rewrite packet address for server . (Port 69 on first packet, then forward to Server Service socket X)
ES4: Send off packet
ES5: Receive server response packet
ES6: Rewrite address for client

S1: Receive read request packet
S2: New Server service thread is created and started
S3: Create Acknowledge packet acknowledging that the read request packet was received
S4: Check for packet formatting errors
S5: Send Acknowledge packet and started

S6: Receive data block
S7: New Server service thread is created
S8: Checks if the received data block is the correct data block
S9: Send acknowledgement packet acknowledging that the data block was received

UCM: Error code 4-Any request



Notes:

- Here EC3 doesn't do anything because it is not a data block packet
- The second path from C4 to C7 repeats until an error packet is received from the server unless an error packet is received

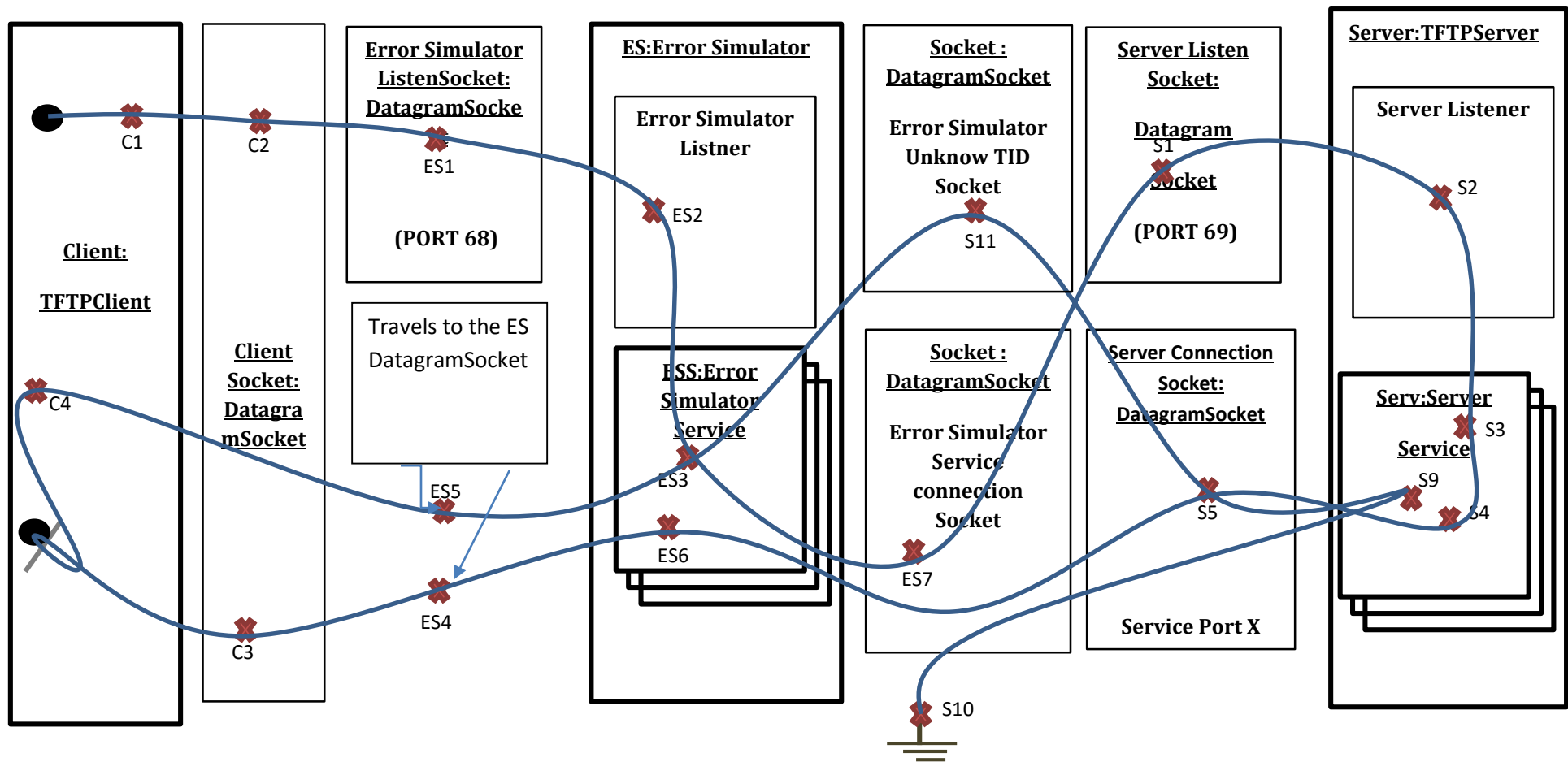
C1: Create write request packet
C2: Send write request packet
C3: Receive acknowledge packet
C4: Prepare Data blocks
C5: Send data block
C6: Receive error packet
C7: stop data block transfer and print error message

<p>ES1: Receive write request Packet</p> <p>ES2: New Error service thread is created and started</p> <p>ES3: Pass on or manipulates packets received if set to corrupt server packets</p> <p>ES4: Send packet out to server</p> <p>ES5: Receive server packet from reply</p> <p>ES6: Pass or manipulate packets received if set to corrupt client packets</p>

- S1:** Receive write request packet
- S2:** New Server service thread is created and started
- S3:** Create Acknowledge packet acknowledging that the write request packet was received
- S4:** Check for packet formatting errors
- S5:** Send Acknowledge packet and started

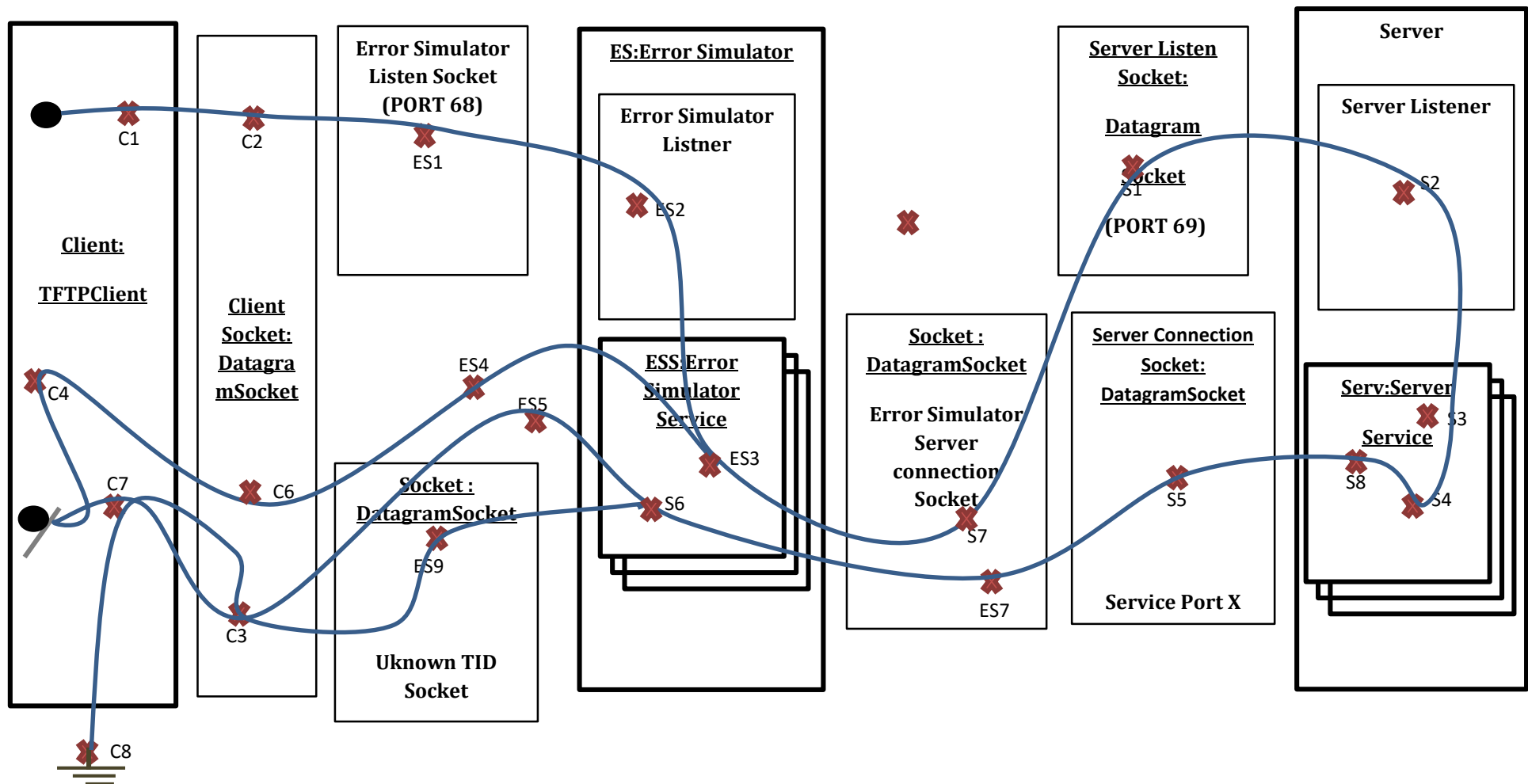
S6: Receive data block
S7: Send error packet stating that the packet received was corrupted
S8: Checks if the received data block is the correct data block

UCM: Error code 5- Server case - Write request



<p>Notes:</p> <ul style="list-style-type: none"> ES5 and ES4 travel through ES Service socket S10 is the end of the path for the Unknown host, but the transfer still continues for the correct TIDs 	<p>C1 : Create write request packet C2: Send write request packet C3: Receive acknowledge packet C4: Prepare Data blocks</p>	<p>ES1: Receive write request Packet ES2: New Error service thread is created and started ES3: Forwards the packet, on every 3 packets, create a new socket Unknown TID Socket and forward thru that port ES4: Forward packet back to Client ES5: Forward packet to ES ES6: Change address to client ES7: Sends the copied packet through an error socket</p>	<p>S1: Receive write request packet S2: New Server service thread is created and started S3: Create Acknowledge packet acknowledging that the write request packet was received S4: Check for packet formatting errors S5: Send Acknowledge packet and started</p>	<p>S6: Receive data block S7: New Server service thread is created S8: Send acknowledgement packet acknowledging that the data block was received S9: Signals an error since the received packet is not from the expected port number S10: Prints an illegal transfer ID error message S11: Send packet from different port</p>
---	---	--	---	--

UCM: Error code 5- Client case - Write request



Notes:	C1 : Create write request packet C2: Send write request packet C3: Receive acknowledge packet C4: Prepare Data blocks C5: Send data block C6: Pass packet to ES C7: Signals an error since the packet received is not from the port expected C8: Send message back to TID in unknown Host	ES1: Receive write request Packet ES2: New Error service thread is created and started ES3: Manipulates and corrupts packets received ES4: Forward packet back to Client ES5: Forward packet to ES ES6: The packet received is not corrupted and passed on ES7: Accept packet from server ES8: Makes a copy of the acknowledgement packet ES9: Sends copied packet through an error socket	S1: Receive write request packet S2: New Server service thread is created and started S3: Create Acknowledge packet acknowledging that the write request packet was received S4: Check for packet formatting errors S5: Send Acknowledge packet and started	S6: Receive data block S7: Recevie Packet from service socket S8: Checks if the received data block is the correct data block S9: Send error packet stating that the packet received was corrupted
---------------	--	---	--	---

