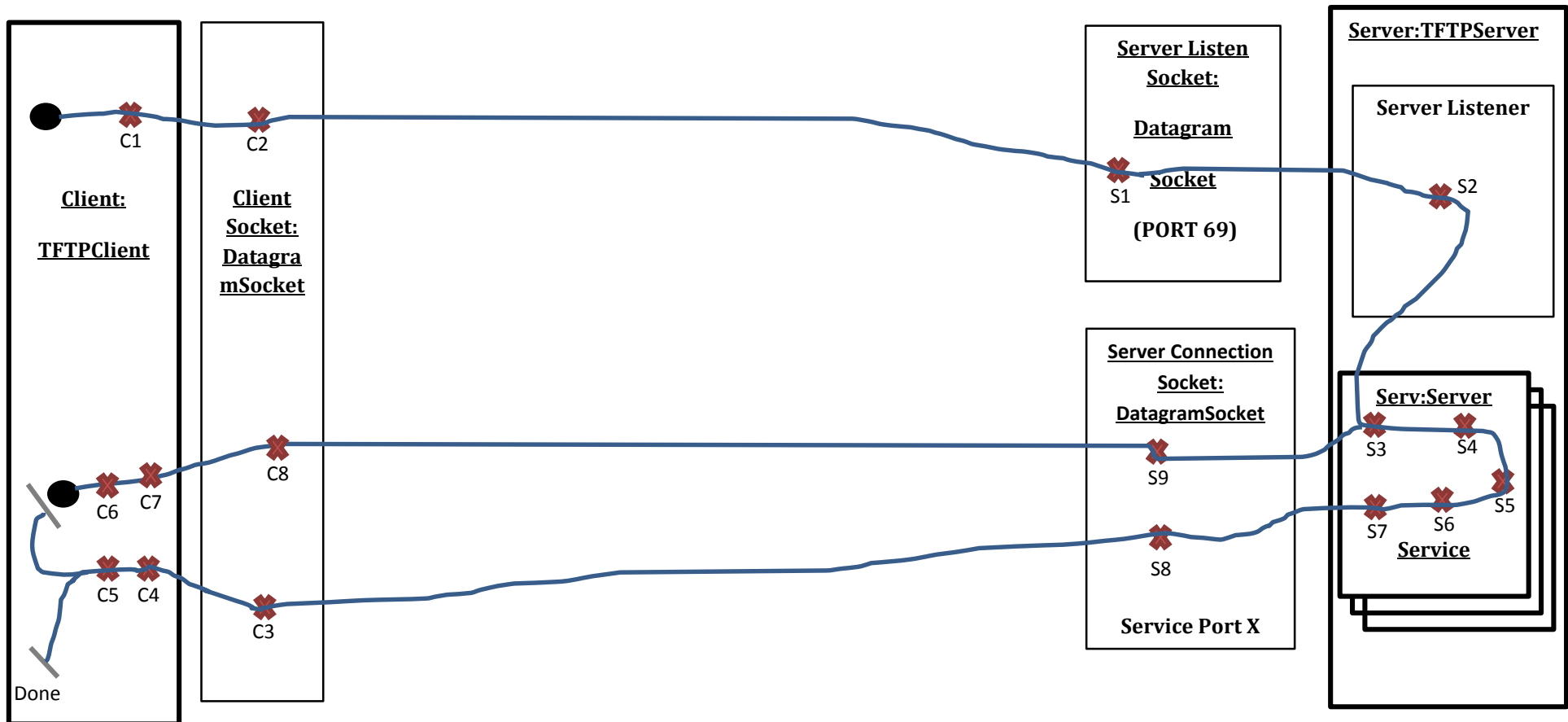


# UCM No Error-Sim Write request



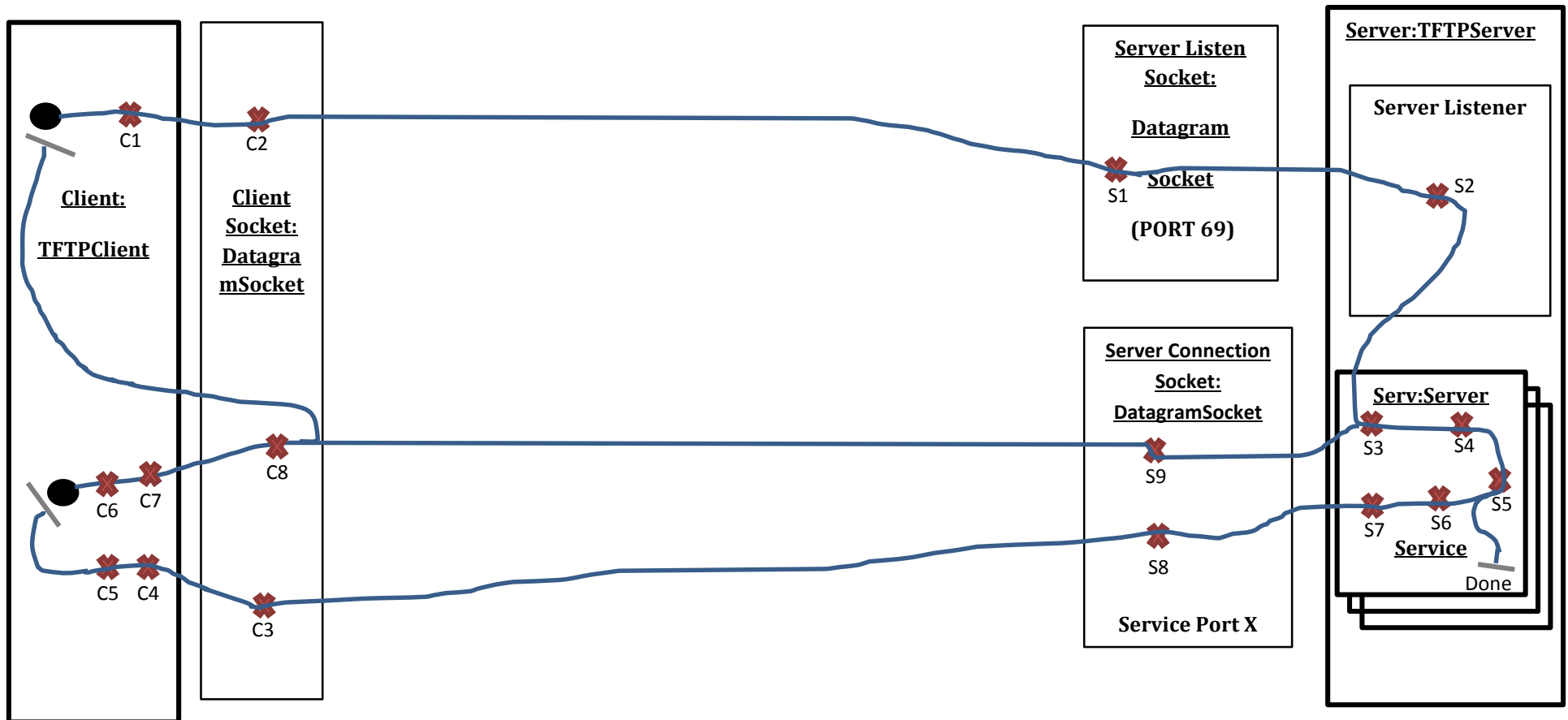
**Notes:**

- In between C5 and C6 the client makes sure if there are more blocks to send if so it goes to C6 otherwise it goes to Done

- C1** : Create write request packet
- C2**: Send write request packet
- C3**: Receive datagram
- C4**: extract message
- C5**: verify ACK
- READ NOTES**
- C6**: form DATA
- C7**: form datagram
- C8**: send datagram

<b>S1:</b> Receive write request packet
<b>S2:</b> New Server service thread is created and started
<b>S3:</b> extract message
<b>S4:</b> verify DATA
<b>S5:</b> write file
<b>S6:</b> form an ACK
<b>S7:</b> form datagram
<b>S8:</b> send datagram
<b>S9:</b> receive datagram

# UCM No Error-Sim Read request



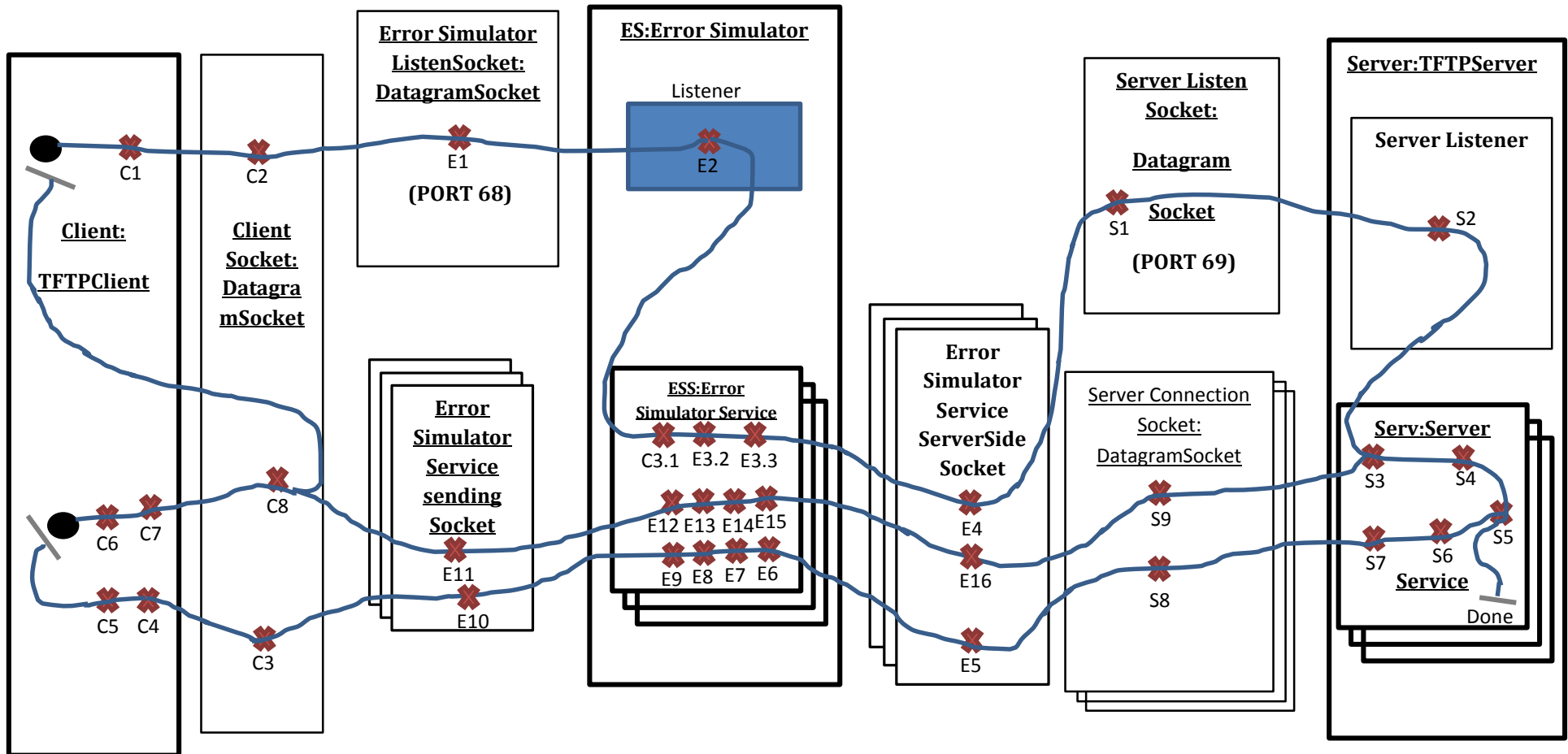
**Notes:**

- In between C5 and C6 the client checks if the datagram received has less than 512 bytes of data. If yes it creates the last ACK packet(S6) and sends it over to the server(S8) and then it finishes the transfer on its side

- C1** : Create read request packet
- C2**: Send read request packet
- C3**: Receive datagram
- C4**: extract message
- C5**: verify DATA & **write file**
- READ NOTES**
- C6**: form ACK
- C7**: form datagram
- C8**: send datagram

<b>S1:</b> Receive read request packet <b>S2:</b> New Server service thread is created and started <b>S3:</b> extract message <b>S4:</b> verify ACK <b>S5:</b> if last ACK go to Done and close service thread <b>S6:</b> form an DATA <b>S7:</b> form datagram <b>S8:</b> send datagram <b>S9:</b> receive datagram
--

# UCM with Error-Sim Read request



**Notes:**

- In between C5 and C6 the client checks if the datagram received has less than 512 bytes of data. If yes it creates the last ACK packet(**S6**) and sends it over to the server(**S8**) and then it finishes the transfer on its side
- packet
- C2:** Send read request packet
- C3:** Receive datagram
- C4:** extract message
- C5:** verify DATA & **write file**
- READ NOTES**
- C6:** form ACK
- C7:** form datagram
- C8:** send datagram

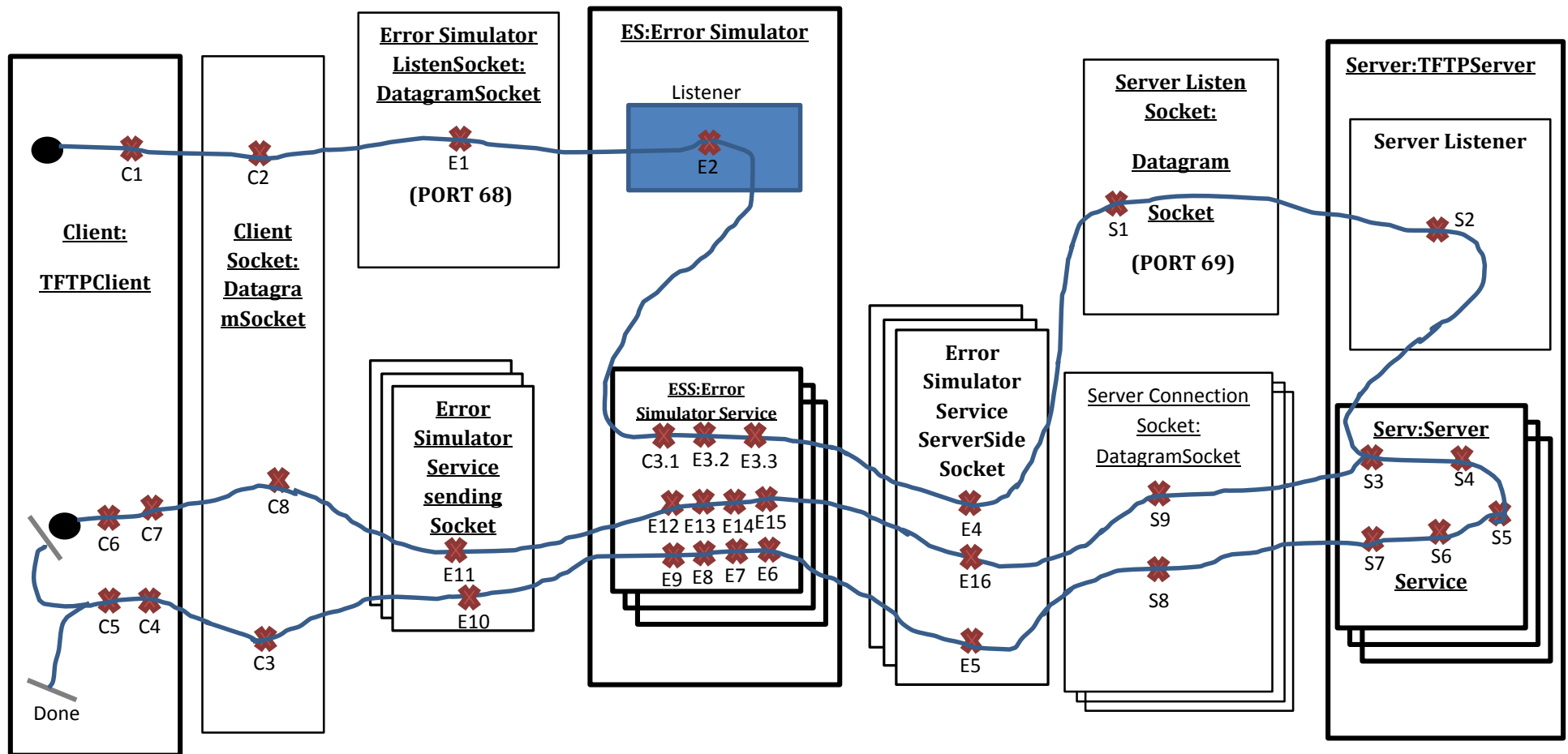
- C1 : Create read request packet
- C2: Send read request packet
- C3: Receive datagram
- C4: extract message
- C5: verify DATA & **write file**
- READ NOTES**
- C6: form ACK
- C7: form datagram
- C8: send datagram

- S1:** Receive read request packet
- S2:** New Server service thread is created and started
- S3:** extract message
- S4:** verify ACK
- S5:** if last ACK go to Done and close service thread
- S6:** form a DATA
- S7:** form datagram
- S8:** send datagram
- S9:** receive datagram

E1: Receive read request packet
E2: New Error simulator service thread is created and started
E3.1: extract RRQ
E3.2: copy RRQ content
E3.3: create new RRQ
E4: : Send read request
E5: Receive datagram
E6: extract message

E7: copy packet content
E8: form a Data using copied content
E9: form datagram
E10: send datagram
E11: receive datagram
E12: extract message
E13: copy packet content
E14: form ACK using copied content
E15: form datagram
E16: send datagram

# UCM with Error-Sim Write request



## Notes:

- In between C5 and C6 the client makes sure if there are more blocks to send if so it goes to C6 otherwise it goes to Done

**C1 :** Create write request packet  
**C2:** Send write request packet  
**C3:** Receive datagram  
**C4:** extract message  
**C5:** verify ACK  
**READ NOTES**  
**C6:** form DATA  
**C7:** form datagram  
**C8:** send datagram

**S1:** Receive write request packet  
**S2:** New Server service thread is created and started  
**S3:** extract message  
**S4:** verify DATA  
**S5:** write file  
**S6:** form an ACK  
**S7:** form datagram  
**S8:** send datagram  
**S9:** receive datagram

**E1:** Receive write request packet  
**E2:** New Error simulator service thread is created and started  
**E3.1:** extract WRQ  
**E3.2:** copy WRQ content  
**E3.3:** create new WRQ  
**E4:** Send write request  
**E5:** Receive datagram  
**E6:** extract message

**E7:** copy packet content  
**E8:** form a ACK using copied content  
**E9:** form datagram  
**E10:** send datagram  
**E11:** receive datagram  
**E12:** extract message  
**E13:** copy packet content  
**E14:** form DATA using copied content  
**E15:** form datagram  
**E16:** send datagram