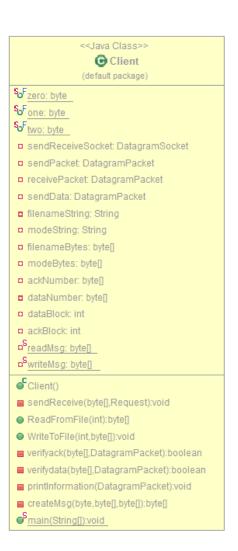
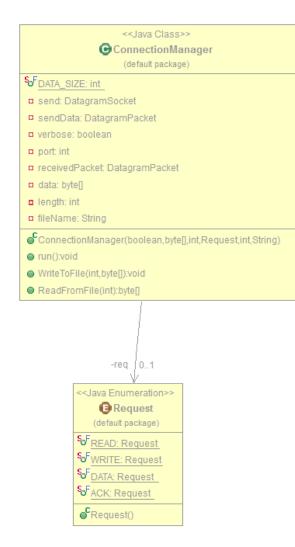
UML Class Diagram





<<Java Class>> ● ConnectionManagerESim (default package) SFDATA SIZE: int SERVER PORT: int □ sendReceiveSocket: DatagramSocket ■ sendSocket: DatagramSocket sendClientPacket: DatagramPacket □ receiveServerPacket: DatagramPacket ■ sendServerPacket: DatagramPacket verbose: boolean ■ data: byteП port: int □ length: int ConnectionManagerESim(boolean,byte[],int,int) printlnformation(DatagramPacket):void

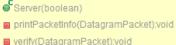
orun():void

(default package) SFDATA_SIZE: int □ receiveSocket: DatagramSocket □ receiveClientPacket: DatagramPacket ■ data: byte[] verbose: boolean ErrorSim(boolean) printPacketInfo(DatagramPacket):void sendReceive():void main(String[]):void <<Java Class>> G Server (default package)

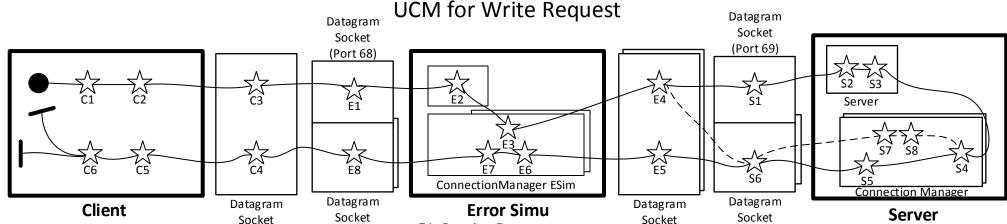
<<Java Class>>

⊕ErrorSim





- sendReceive():void
- Smain(String[]):void



C1: Create message C2: Form Datagram C3: Send Datagram C4: Receive Datagram C5: Extract message

C6: Verify Acknowledge

E1: Receive Datagram

E2: Extract Message

E3: Create Datagram

E4: Send Datagram

E5: Receive Datagram

E6: Extract Message

E7: Create Datagram

E8: Send Datagram

S1/S7: Receive Datagram S2: Extract Messages S3: Verify Write Request

S4: Write to file S5: Form Acknowledge

S6: Send Datagram

S7: Extract Message

S8: Verify data

S4: Read Data

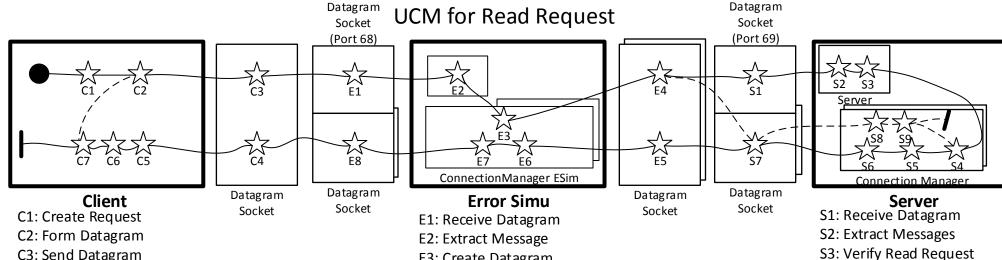
S5: Form Acknowledge

S6: Form Datagram

S7: Send Datagram

S8: Extract Messages

S9: Verify Acknowledge



C3: Send Datagram

C4: Receive Data

C5: Extract message

C6: Verify Data

C7: Send Acknowledge

- E3: Create Datagram
- E4: Send Datagram
- E5: Receive Datagram
- E6: Extract Message
- E7: Create Datagram
- E8: Send Datagram