-conversationID : MessageID

+send(envelope : Envelope)

+send(message : Message, recipient : InetSocketAddress)

+receive(clazz : Class<?>, timeout : long, timeUnit : TimeUnit) : Envelope

+waitForDataChannelAndConnect(timeout : long, timeUnit : TimeUnit) : TCPConnection

+openDataChannelAndWaitForConnection(recipient: InetSocketAddress, timeout: long, timeUnit: TimeUnit): TCPConnection

+connectToDataChannel(endpoint: InetSocketAddress, timeout: long, timeUnit: TimeUnit): TCPConnection

+receive(timeout : long, timeUnit : TimeUnit) : Envelope

+retry(attempts : int, operation : Runnable)

accesses

PostOfficeBox

add(e : Envelope)

-poll(): Message

Envelope

-remoteEndPoint : InetSocketAddress

+setMessage(message : Message) : void

Message
-messageID : MessageID
-conversationID : MessageID

-message : Message

+getMessage(): Message

Conversations and Conversation Factories <<abstract>> Conversation ConversationFactory -subsystem : CommSubsystem subsystem : CommSubsystem -conversationID : MessageID start(Envelope) : Conversation +send(envelope : Envelope) <<create>> ~start(Initiator) : Conversation +send(message : Message, recipient : InetSocketAddress) registerResponder(MessageType, ConvType) +receive(timeout : long, timeUnit : TimeUnit) : Envelope +receive(clazz : Class<?>, timeout : long, timeUnit : TimeUnit) : Envelope +openDataChannelAndWaitForConnection(recipient: InetSocketAddress, timeout: long, timeUnit: TimeUnit): TCPConnection +waitForDataChannelAndConnect(timeout : long, timeUnit : TimeUnit) : TCPConnection +connectToDataChannel(endpoint: InetSocketAddress, timeout: long, timeUnit: TimeUnit): TCPConnection +retry(attempts: int, operation: Runnable) **Note**: Initiators and Responders types of conversations that are meant to be subclassed at the application layer. Some implementations, such as RRWAInitiator and RRWAResponder, have been provided but these are not comprehensive to all use cases for conversations. <<abstract>> <<abstract>> Responder Initiator +respond(env : Envelope) +initiate() RRWAInitiator RRWAResponder timeout : long -timeout : long -timeUnit : TimeUnit -timeUnit : TimeUnit -attempts : int -attempts : int +getInitialMessage(): Message +onMessageReceived(msg : Message) : Messa... +getRecipient(): InetSocketAddre.. +onAckReceived(msg : AckMessage) Basic implementations for a Request-Reply with Acknowledge conversation pattern.

Shared Objects

MessageID

-left : int -right : short

+MessageID(left:int, right:short)

+MessageID(str : String)

+toString() : String +getLeft() : int +getRight() : short

TCPConnection

-socket : Socket

-serverSocket : ServerSocket

-in : DataInputStream-out : DataOutputStream

+readUTF(): String

+read(bytes : byte []) : int +readBoolean() : Boolean

+readByte(): byte

+readUnsignedByte(): int

+readShort(): short

+readUnsignedShort(): int

+readChar() : char +readInt() : int +readLong() : long +readFloat() : float +readDouble() : double

+read(): int

+write(i:int):TCPConnection

+write(bytes : byte []) : TCPConnection

+writeBoolean(b : boolean) : TCPConnection

+writeByte(b: int): TCPConnection +writeShort(s: int): TCPConnection +writeChar(c: int): TCPConnection +writeLong(I: long): TCPConnection +writeFloat(f: float): TCPConnection

+writeDouble(d : double) : TCPConnection +writeBytes(bytes : String) : TCPConnection +writeUTF(string : String) : TCPConnection

+flush(): TCPConnection +close(): TCPConnection

+isConnection(): boolean











