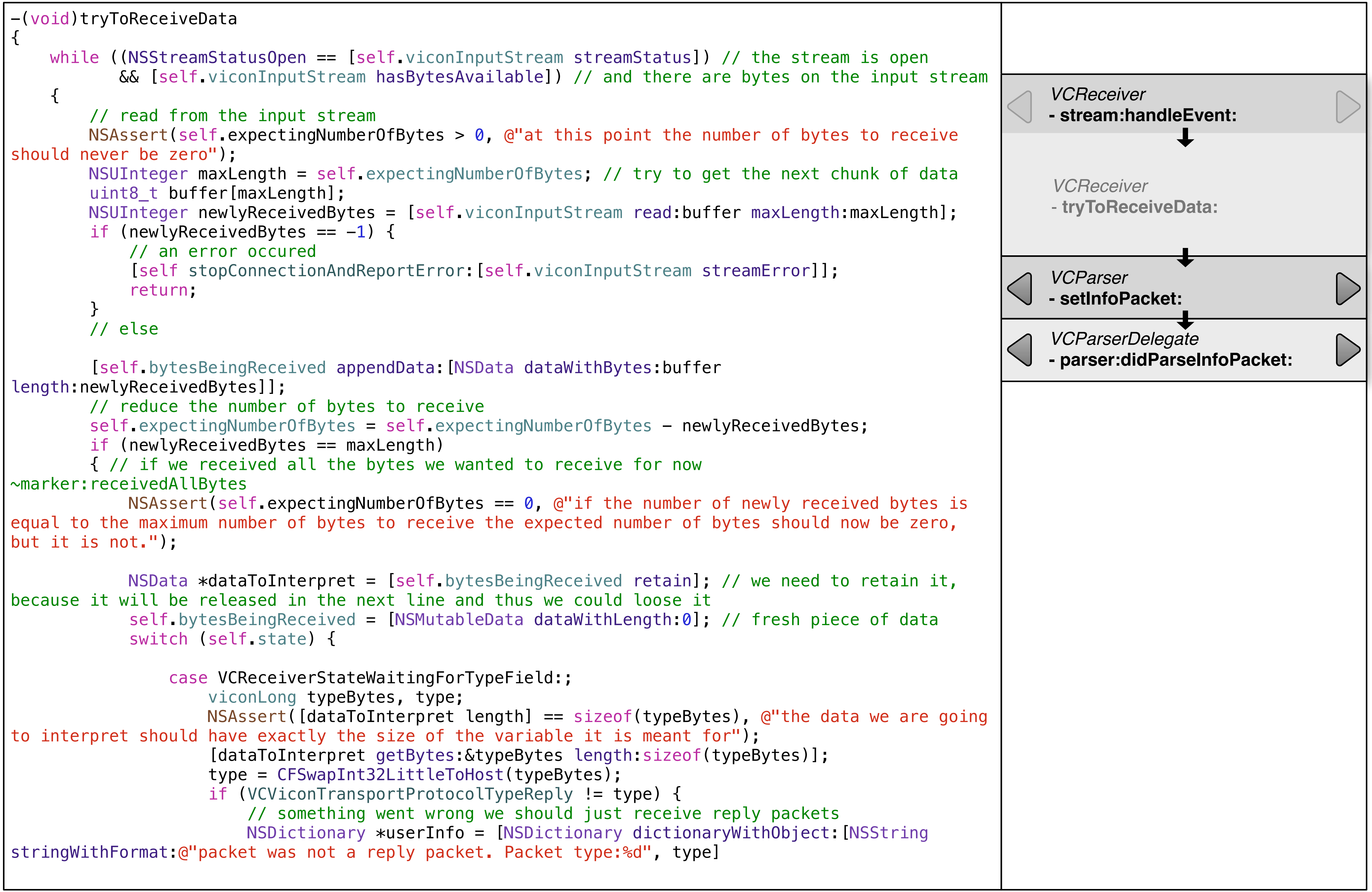

```
-(void)tryToReceiveData
{
    while ((NSStreamStatusOpen == [self.viconInputStream streamStatus]) // the stream is open
           && [self.viconInputStream hasBytesAvailable]) // and there are bytes on the input stream
    {
        // read from the input stream
        NSAssert(self.expectingNumberOfBytes > 0, @"at this point the number of bytes to receive
should never be zero");
        NSUInteger maxLength = self.expectingNumberOfBytes; // try to get the next chunk of data
        uint8_t buffer[maxLength];
        NSUInteger newlyReceivedBytes = [self.viconInputStream read:buffer maxLength:maxLength];
        if (newlyReceivedBytes == -1) {
            // an error occured
            [self stopConnectionAndReportError:[self.viconInputStream streamError]];
            return;
        }
        // else

        [self.bytesBeingReceived appendData:[NSData dataWithBytes:buffer
length:newlyReceivedBytes]];
        // reduce the number of bytes to receive
        self.expectingNumberOfBytes = self.expectingNumberOfBytes - newlyReceivedBytes;
        if (newlyReceivedBytes == maxLength)
        { // if we received all the bytes we wanted to receive for now
~marker:receivedAllBytes
            NSAssert(self.expectingNumberOfBytes == 0, @"if the number of newly received bytes is
equal to the maximum number of bytes to receive the expected number of bytes should now be zero,
but it is not.");

            NSData *dataToInterpret = [self.bytesBeingReceived retain]; // we need to retain it,
because it will be released in the next line and thus we could loose it
            self.bytesBeingReceived = [NSMutableData dataWithLength:0]; // fresh piece of data
            switch (self.state) {

                case VCReceiverStateWaitingForTypeField:;
                    viconLong typeBytes, type;
                    NSAssert([dataToInterpret length] == sizeof(typeBytes), @"the data we are going
to interpret should have exactly the size of the variable it is meant for");
                    [dataToInterpret getBytes:&typeBytes length:sizeof(typeBytes)];
                    type = CFSwapInt32LittleToHost(typeBytes);
                    if (VCViconTransportProtocolTypeReply != type) {
                        // something went wrong we should just receive reply packets
                        NSDictionary *userInfo = [NSDictionary dictionaryWithObject:[NSString
stringWithFormat:@"packet was not a reply packet. Packet type:%d", type]
```