**#17871** - <https://redmine.vnc.biz/issues/17871>

1. **Description**:

The zimlet should identify common errors, like:

* One or both users go offline
* The receiver takes too long to deliver an acknowledgment
* The receiver does not accept the file transfer within 30 minutes.

After identifying an error, the file transfer should be aborted and the corresponding error message should be showed.

1. **Analytic**
   1. **Root cause**:
   2. **Solutions**:

**Handle error**:

* One or both users go offline

When user offline (offline chat or sign out) , disconnect function also send session-terminated and message “gone”.

Create messages “gone” display.

* The receiver takes too long to deliver an acknowledgment

When Sender send slices , create var Timer: (240 sec - default TCP)

* when timer = 240 sec, send session-terminated and message “gone”.
* when Sender receives message “ACK”, clear timer.
* The receiver does not accept the file transfer within 30 minutes.

When Sender send request transfer file, create var Timer: (30 min)

* when timer = 30 mins, send session-terminated and message “timeout”.
* when Sender receives message “accept”, clear timer.

Create messages display when session timeout.

1. **Implementation**
   1. **Code:**

**Handle error**:

* One or both users go offline:

When user offline (offline chat or sign out) , disconnect function also send session-terminated and message “gone”.

prosody-packaging/mini/javascripts/mini.js

|  |
| --- |
| self.disconnect = function() {  ...  //Send session-terminated: 'gone'  if (typeof receiverSession != 'undefined') {  receiverSession.end('gone',false);  } ;  if (typeof transferSession != 'undefined') {  transferSession.end('gone',false);  };  ...  } |

Create messages “gone” display:

prosody-packaging/mini/javascripts/mini.js

|  |
| --- |
| self.displayFileTransferStatus = function(session,action) {  …  switch(action) {  case 'gone':  body = 'User offline or not response';  break;  ...  } |

Fix the error when receiver display message “gone” although the Filetransfer disappeared:

|  |
| --- |
| receiverSession.on('terminated',function(receiverSession,reason){  ...  if (reason.condition == 'gone') {  return;  }  self.displayFileTransferStatus(session,reason.condition);  }); |

* The receiver takes too long to deliver an acknowledgment:

When Sender send slices , create var Timer: (240 sec - default TCP)

* when timer = 240 sec, send session-terminated and message “gone”.

prosody-packaging/mini/javascripts/standzaio.bundle.js

|  |
| --- |
| this.processingQueue = async.queue(function (task, next) {  ...  //Set time out:  timer = window.setTimeout(function(){  reader.abort();  //Send session-terminated: 'gone'  if (typeof transferSession != 'undefined') {  transferSession.end('gone',false);  };  },  240000 // 240 sec  );  };  })(task.file); |

* when Sender receives message “ACK”, clear timer:

|  |
| --- |
| self.channel.onmessage = function() {  // Handle ack  if(ab2str(event.data) == 'ACK') {  ...  //Clear time out:  window.clearTimeout(timer);  ...  }  } |

* The receiver does not accept the file transfer within 30 minutes:

When Sender send request transfer file, create var Timer: (30 min)

* when timer = 30 mins, send session-terminated and message “timeout”:

|  |
| --- |
| self.startFileTransfer = function(buddy){  ...  //Set TimeOut:  setTimeout(function(){  console.log("Session timeout");  transferSession.end('timeout',false);  },  1800000 // 30 min  ); }, |

* when Sender receives message “accept”, clear timer:

|  |
| --- |
| self.startFileTransfer = function(buddy){  …  //Clear timer  transferSession.on('accepted',function(){  console.log("accepted");  clearTimeout(timer);  });  } |

Create messages display when session timeout:

|  |
| --- |
| self.displayFileTransferStatus = function(session,action) {  …  switch(action) {  case 'timeout':  body = 'Session timeout';  break;  ...  } |