

In memory to Xiomara Reinoso Gómez

# The eXtended subsystem: Gateway

This module is inspired on NASA's moon gateway module.

This subsystem is intended to provide a full-duplex communication between the connected peers to the server. This is the closets TFProtocol provide a peer-to-peer comm without going to NAT hole pushing. The specific IPC -Interprocess communication- used by the Gateway subsystem is implementation dependent, however some guarantees must be meet.

The IPC is required to communicate the clients -peers-, through the server processes that represents them. However the data exchange between the server processes must take place after been decrypted because each one of them has their own session key that do not match with the other. It is not required that the IPC doesn't involve file storing operations because in some OS could be the only IPC available.

As the Gateway subsystem is intended to provide the most closest possible form of peer-to-peer comm, it is recommended, although not required, that the clients -peers- encrypts the data to be exchanged with they own cryptography system. This is a security provision to avoid a possible man in the middle done by a malicious process in the server that could read either the RAM or a possible file used for IPC.

As you can infer from the above, if the peer is no connected the message sent is lost forever.

## XS\_GATEWAY

XS\_GATEWAY makes the server enter the subsystem. Once inside the subsystem, the server remains there until the client explicitly exits the subsystem with the proper signal. The return status for this command could be:

OK

UNKNOWN

The UNKNOWN response is due the lack of implementation on the server.

The OK response means the server is inside the subsystem and is ready to be used.

This subsystem uses a 32bit integer normalized -converted to bigendian- to indicates the other end the size of the payload to be sent. 0 and the negative values are meant to signal the server. At present 0 is reserved and not used while only -1 is currently used which tells the server to exit the Gateway subsystem.

The maximum size of any transmission must not exceed the standard TFProtocol communication buffer size specified in the corresponding document.

Once inside the subsystem the first thing the server waits for is the peer self identity to be created. The identities are not persistent, instead they are created at the moment the peer enters the subsystem and send it. By identity we mean any string that works as the identifier to be used as destination by other peers. A header, as stated before, must be sent to the server with the size -in bytes- before sending the string with the identity.

When the server receives the identity it will respond one of the following status:

OK

FAILED 14 : Internal memory error.

FAILED 1 : Access denied to location. FAILED 1

FAILED 1 : Internal error occurred.

The server will precede the status with a header indicating the size of what coming next.

If a failure is received, then the server is back again waiting for commands in the main interface of TFProtocol. If OK is received, then the server is ready for communicating with other connected peers.

Once OK is received by the client, the server enters the next loop:

The server waits for a destination peer identity and a message for that peer.

The client must send a header containing the identity size of the destination peer. Then must send the identity. After that, again, must send a header with the message size, and right after to it, the message.

The identity size must be less than or equal to TFProtocol communication buffer size – 1. If that buffer is 512, for example, then the identity size must be less than or equal to 511.

There is no automatic acknowledgement sent back to the client for the message just sent. By contrary if the peers require it by the

nature of their communication, they must implement it. As far as is concern to the server, the received message is just forwarded to the destination peer, no matter if the peer is currently connected or not. The server only makes its best effort to deliver it. Any confirmation mechanism must be accorded by the peers. Note that if the destination peer is not connected at the moment in which the message is sent, there will be no delivery.

The window for the client to exit the Gateway subsystem and return to the TFProtocol main command interface is the moment to send the header that precedes the peer destination identity, by sending a header with value -1.