



WWW.JAZZMUTANT.COM

This protocol is intended to transfer large data as a stream inside OSC packet. These OSC packets are usually transmitted using UDP, and therefore are limited in size to 1500 bytes.

Protocol of transmission

JazzMutant has defined a protocol for sending large data in several OSC packets. This protocol includes a transmission window and individual acknowledgment. See TCP literature for understanding 'transmit window' system and acknowledgment.

- 1) Transfer is initiated by sending the first packet of the stream. The receiver allocates the reception memory if it accepts the transfer and sends an ACK message to the sender. It also sends back an ACK message for each block successfully received or already received.
- 2) The sender sends any packet included in the transmission window, in any order, and is allowed to send them again if it does not receive the ACK in proper time to recover lost packets.
- 3) The transmission window is moved forward each time the sender receives an ACK that corresponds to the beginning of the window.

Compatibility

The current version of the Lemur only supports a block size of **1400 bytes** and a window size of **1 block**, later versions may support any block size and a larger window size.

OSC Stream packet format

Transmit packet

/interface **,iib** **size** **offset** **crc** **data**

size full size of the stream, it should be checked by the receiver.

offset current offset of the data block inside the stream.

crc checksum of the data block.

data is transmitted within an OSC ‘blob’ and therefore contain the data size. It is fixed to 1400 bytes in the current Lemur v1.3. Any data packet less than 1400 bytes in size must be padded with zeros so its length is a multiple of 4 bytes.

ACK packet

/interface ,**ii** **0** **offset**

size is set to 0 to identify an ACK packet

offset is the offset of the bloc that was successfully received, the sender could advance the window position according to this offset.

CRC is calculated by doing the 32 bits sum of the data in Little Indian order.

Data	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D	
CRC	0x04030201 + 0x08070605 + 0x0D0C0B0A	= 0x19161310