Routing Open Sound Control messages via vanilla JACK to build low-latency event translator/filter chains and map unconventional controller data to musical events

INTRODUCTION
We have run continuously into situations where it would have been of advantage to have a means to route OSC directly via JACK. There would be multiple benefits:
- Sample-accuracy, low-latency, less context switches, same routing infrastructure and session management for OSC and MIDI.
- Support for more complex real-time event data, e.g. for continuous controllers and motion or gesture driven sound synthesis.
- Tighter coupling between OSC and MIDI, possibility for on-the-fly translation of the two.
As it turns out, we can have all of this with vanilla JACK already, by piggybacking on top of JACK MIDI, one has to be cautious though not to interfere with the latter.

METHODS
Route raw OSC messages via JACK MIDI ports
Make use of JACK MIDI infrastructure, simply replace raw MIDI with raw OSC messages. JACK is indifferent to what the raw bytes encode and will handle routing and multiplexing for you. OSC types are not network encoded while inside the JACK graph. Each message has a timestamp, there are no bundles.

<table>
<thead>
<tr>
<th>C-Type</th>
<th>Description</th>
<th>MIDI</th>
<th>OSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint32_t</td>
<td>time</td>
<td>23</td>
<td>44</td>
</tr>
<tr>
<td>size_t</td>
<td>buffer</td>
<td>3</td>
<td>20</td>
</tr>
</tbody>
</table>

OSC message → JACK event
Inject single OSC messages at the beginning of the next sample period.

OSC packet (UDP/TCP)

-JACK MIDI

 OSC packet (UDP/TCP)

OSC nested bundles → JACK events
Unroll OSC bundles into single messages, translate between NTP timestamps and JACK frame times, add to sorting event queue for later dispatch into JACK graph.

OSC packet (UDP/TCP)

JACK event

| period (n) | /00 /hello 'world' |
| period (n+2) | /33 /tuio2/frm it 3459 d6ecd330.5800000 |
| period (n+3) | /33 /tuio2/tok iifff 129 0 0.89 0.23 0.0 |

JACK OSC events

| period (n) | /00 /new siisiis 'base' 129 0 0 'out' 0 'gate' 0 |
| period (n+2) | /33 /tuio2/frm it 3459 d6ecd330.5800000 |
| period (n+3) | /33 /tuio2/tok iifff 129 0 0.89 0.23 0.0 |

Discriminate between JACK MIDI and JACK OSC ports
Make use of the JACK metadata API to mark JACK MIDI ports used for OSC routing, makes JACK OSC clients aware of each other.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'<a href="http://jackaudio.org/event/type">http://jackaudio.org/event/type</a>'</td>
<td>'SuperCollider'</td>
</tr>
<tr>
<td>'<a href="http://jackaudio.org/event/OSC/type">http://jackaudio.org/event/OSC/type</a>'</td>
<td>'J100 2.0'</td>
</tr>
</tbody>
</table>

Tjost is JACKified Open Sound Control Transmission
Tjost is a reference implementation of a scriptable event translator, handles OSC via UDP/TCP/JACK. JACK MIDI, ALSA MIDI, JACK AUDIO, embeds LuaJIT tuned for realtime execution in a JACK process thread.

Case Study 2: MIDI + SuperCollider
Inject TUID from UDP into JACK graph, translate to general on/off/set events, route to downstream translators to steer SuperCollider or some JACK MIDI app.

DISCUSSION

Problems!
- Legacy clients cannot distinguish between JACK MIDI and JACK OSC.
- Only JACK clients which route OSC set the event type via the metadata API and thus are aware of each other.
- JACK MIDI and JACK OSC can be connected by jack_connect.
- As long as OSC stays in the JACK graph we are fine, but when one tries to export the event, e.g. to ALSA (a2jmidid) or NetJACK2, clients will expect MIDI data.

Fixes!
- Generalize JACK MIDI for arbitrary unidirectional events (MIDI, OSC, ...).
- Define jack_event_* and alias to jack_midi_* (downwards compatibility)?
- Use metadata API to set event type of a given port (MIDI, OSC, ...)?
- By default an event port would have type MIDI (downwards compatibility)?
- User can overwrite event port type, e.g. to route OSC?
- Modify jack_connect to only connect event ports with matching event types?
- Only allow MIDI to exit JACK graph, e.g. to/from ALSA, NetJACK2, ...?
- Include libjackosc into JACK, or provide it as an extension?