POETS Orchestrator: Packet Format

16/12/2020

All packets sent over the POETS network, whether travelling between two devices hosted on the POETS compute fabric or between the Mothership and a device, follow a consistent format.

1. Background

Packets are distinct to the MPI-based messages used in the rest of the Orchestrator and are used for all communication[1] within the POETS compute fabric. Packets are have a smaller and less complex header that is design for efficient processing by the embedded RISC-V cores.

Tinsel supports packets that are up to 64-bytes long. At a hardware level, Tinsel divides packets into multiple units (flits) that are transmitted and received in sequence. In the current version of Tinsel, each flit is 16-bytes long and each full packet consists of four flits. It should be noted that Tinsel does not have to send all four flits if a packet's payload will fit in fewer, which results in improved small-packet performance.

2. General Format

Packets sent over the POETS network by the Mothership and Softswitches follow the general layout shown in Table I. The first 64-bits of the Tinsel packet are used for headers. This leaves 56 bytes of payload that are usable for application data, enough to send 18 uint32_t/single-precision float values or nine double-precision float values.

Table I: General packet format

Flit	0				1				2				3			
Word	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bytes	0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31	32-35	36-39	40-43	44-47	48-51	52-55	56-59	60-63
Bits	0 - 31	32 - 63	64 - 95	96 - 127	128 - 159	160 - 191	192 - 223	224 - 255	256 - 287	288 - 319	320 - 351	352 - 383	384 - 415	416 - 447	448 - 479	480 - 511
Use	Header		Payload													

The format does not directly encapsulate the source or destination hardware addresses – the destination address is not part of the Tinsel payload and the sending address can be inferred from the header contents.

3. Header

The header is split into two sub-headers: the Software address and the Pin Address. The Software address, when combined with the 32-bit hardware address, uniquely identifies a device within the POETS compute fabric. The Pin address identifies the destination pin an edge of a packet[2].

3.1. Software Address

The first 32-bits of the header (`swAddr`) contain the software address as described in the Addresses documentation. The layout of `swAddr` is reproduced in Table II for convenience. Reserved device addresses are shown in Table III.

[1] With the exception of serial-based Debuglink output.

[2] N.B. this means that a sender needs to know about the pin and edge assignments of the receiver.

Table II: swAddr bit layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Device Address																	Opcode								Task					CNC	MS

Table III: Reserved device addresses

Address	Name	Mothership -> Softswitch	Softswitch -> Mothership
0xFFFF	P_ADDR_BROADCAST	Sends to all devices hosted on the target Softswitch.	*UNUSED*

3.1.1. Opcodes

Opcodes are only set when the packet is a control packet and indicate the purpose of the packet. An Opcode must only be set when the CNC bit is set in the software address. Opcodes above 0xEF are reserved for internal Orchestrator functionality and cannot be used by an application.

In future, opcodes up to 0xEF may be used within an application; a packet to a device with the CNC bit set and an opcode up to and including 0xEF will trigger a device’s OnCtl handler. Table IV lists the reserved Opcodes and details their functionality.

Table IV: List of opcodes

Opcode	Name	Mothership -> Softswitch	Softswitch -> Mothership
0x00	No-op		
0x01-0xEF	App-ops	Intended for future implementation of Application-specific Opcodes.	
0xF0-0xF9	***RESERVED***		
0xFA	`P_CNC_IMPL`	Indicates that the packet should be handled by a device’s implicit Supervisor receive handler.	Indicates that the packet is bound for the Supervisor’s implicit receive handler.
0xFB	`P_CNC_INSTR`	Requests instrumentation from the target Softswitch.	Indicates that the packet contains instrumentation.
0xFC	`P_CNC_LOG`	*UNUSED*	Log packet from the Softswitch.
0xFD	`P_CNC_BARRIER`	Indicates that the Softswitch should progress through the barrier and start the app.	Indicates that the Softswitch has reached the initialisation barrier.
0xFE	`P_CNC_STOP`	Indicates that the Softswitch should stop execution gracefully.	*UNUSED*
0xFF	`P_CNC_KILL`	*UNUSED*	Indicates that an Assert has failed and/or that execution cannot continue.

3.2. Pin Address

The interpretation of the last 32-bits of the header (`pinAddr`) depends on the context of the packet.

For device-to-device packets, `pinAddr` contains the target pin index and destination edge index as described in the Addresses documentation and reproduced in Table V for convenience. Target Pin (`TGTPIN`) indicates the index of the destination pin in the device type’s input pin list. This is used by the receiving softswitch to select the correct receive handler to use to process the packet. Destination Edge Index (`DESTEDGEINDEX`) indicates the index of the edge in the input pin’s edge list and is used to select the correct set of properties and state for the edge.

Table V: pinAddr bit layout for device-to-device packets

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Target Pin									Destination Edge Index																						

For device-to-supervisor packets sent via the implicit supervisor output pin or emitted by handler_log(), pinAddr contains an index that uniquely identifies the device within the box. The Supervisor contains a lookup table to convert this index into a 64-bit full symbolic address. This limits the number of devices serviced by a single supervisor to 2^{32} (~699,000 per thread with the current hardware assuming one supervisor per box), which is significantly more than will ever be realised for other practical reasons.

For certain control packets to the Mothership (e.g. instrumentation), pinAddr is used to send the hardware address of the sending Softswitch[3].

4. Implementation

Helper structs, masks and definitions for the packet format are implemented in common/poets_pkt.h.

Table VI: Packet helper structs

Short Name	Long Name	Description
P_Pkt_Hdr_t	poets_packet_header	Standard packet header. Includes the software address (swAddr) and pin address (pinAddr).
P_Pkt_t	poets_packet	Standard packet format. Contains a standard header (header) and an unformatted payload (payload) realised as an array of uint8_ts. The size of the payload is defined as P_PKT_MAX_SIZE-sizeof(P_Pkt_Hdr_t)
P_Addr_Pkt_t	poets_address_packet	Includes a hardware address (hwAddr) and a standard packet (packet). Used within the Mothership for outbound packets.
P_Debug_Pkt_t	poets_debug_packet	Includes a source hardware address (origin) and a byte of data (payload). Used within the Mothership to hold Debuglink/UART data.
P_Log_Pkt_Pyld_t	poets_log_packet_payload	Payload formatting for packets emitted by handler_log(). Includes a sequence number (seq) and an unformatted payload realised as an array of uint8_ts
P_Instr_Pkt_Pyld_t	poets_instr_packet_payload	Payload formatting for instrumentation packets. See the Softswitch documentation for more details.

[3] In due course, this will change for device-unique control packets (e.g. log packets) to use the same index as implicit device-to-supervisor packets.

poets_pkt.h includes two inline methods to assist with packet formation:

are not included in the packet payload. The basic packet layout is shown in Table I and a packed struct (`P_Pkt_t`) is provided in `poets_pkt.h`.

Each POETS packet uses a 64-bit header. A packed-struct (`P_Pkt_Hdr_t`) for the header is provided in `poets_pkt.h`.

The Device Address is used as the index into the array of `devInst_t*` stored in the `ThreadContext`. The Softswitch checks whether the specified address is one of the reserved addresses listed in Table III or that the address is in range, logging any occurrences of an out-of-range address.

The Softswitch checks to make sure that the specified pin and edge indices are in range and logs any occurrences of out-of-range indices.