Questions on ESS Readout Protocol

# Introduction

This document serves as a starting point for the discussion on defining an interface between the ESS readout system and DMSC. The basis for the protocol specification is the suggestion by teh Detector Group shown in Figure 1.



Figure 1: Proposal for the ESS readout protocol. Image curtesey of the Detector Group.

# Readout Protocol

The protocol is based on UDP/IP over Ethernet with the following parameters

## Data only

The protocol consists of UDP data transmitted in the direction from readout system to the event formation system. Since data is not transmitted in the opposite direction there is no acknowledgements or retransmissions.

## Ethernet

### Packet sizes

Legacy Ethernet packet sizes up to 1518 bytes MUST be supported. Jumbo frames up to 9018 bytes MAY be supported. Sizes will depend on FPGA resources

### Addressing

Source mac addresses (smac) MUST be locally administered mac addresses with a common prefix (suggested 02:0E:55) and a configurable three byte device id.

Destination mac addresses (dmac) must be configurable if they are not resolved using arp. Unicast mac addresses MUST have have the multicast bit set to 0.

All addresses must be predefined via the control system.

Example:

The source mac addresses may reflect the readout topology by encoding ids of readout systems and individual ports in it. This is useful for the development and bring-up phase. No other meaning will be attributed to the mac address.

readout device 1 port 1 smac: 02:0E:55:00:01:01

readout device 7 port 2 smac: 02:0E:55:00:07:02

## IP

Only IPv4 is supported.

The correct operation of IP requires configuration of the source and destination ip addresses.

For each source and destination mac address there MUST be a corresponding source and destination ip address.

Both the source and destination ip addresses MUST be configurable.Yes

## UDP

The udp checksum is not used and is set to zero. Yes

The source and destination udp ports MUST be configurable. Yes

## Packet Structure

The packet structure is shown in Figure 2. It consists of a common packet header, zero or more readout specific data blocks and a common packet trailer.

Add protocol version field ?

The data is volatile in that the no data is archived in this format. We don’t anticipate changing the protocol once in use, so did not provide for this.



Figure 2: Readout Packet Structure

All multi byte values are respresented as little endian in order to be compatible with Intel CPUs. This means that no CPU cycles will be spent on byte swapping on 16, 32 or 64 byte fields.

This is OK

The fields of the packet header and footer will be described below. The data formats for the individual data blocks may be described in this document or in a detector specific Interface Control Document (ICD). This is OK

### type

field size: 2 bytes

Type is a field identifying the source (and thereby the encoding) of the readout data. It is encoded as two bytes. Parsing the data blocks will be specific to **type**. See table Table 1 for an initial suggestion. We will take this suggestion on board, but some different partitioning may be needed

Table 1: Reserved packet types and ranges.

|  |  |
| --- | --- |
| **packet type** | **description** |
| 0x0000 | invalid, must not be used |
| 0x0001 | reserved for test |
| 0x0002 - 0x0fff | *unallocated* |
| 0x1000 - 0x4fff | detector readout |
| 0x5000 - 0x8fff | monitor readout |
| 0x9000 - 0xcfff | sample data |
| 0xd000 - 0xfffe | *unallocated* |
| 0xffff | reserved |

### word count

field size: 2 bytes

Number of (32 bit?) words in the data blocks? Why not just length in bytes?

The basic quantum of data is 32 bits , but If you prefer bytes we agree

You missed the sequence field.

### reserved

field size: 2 bytes

This field is reserved and must be zero for the initial version of this protocol.

### length

field size: 2 bytes

how is this related to word count - is this reduntant?

There is no length field.

### PAD

field size: 0 - 63 bytes

Padding will be applied to ensure that ethernet packets are a multiple of 64 bytes. The padding consists of zeroes.

Yes, and in cases where fixed length packets are desired, the pad can also be used for that.

### checksum

field size: 4 bytes

Add text

The exact nature of the checksum is to be defined. It could be dropped for Ethernet as the CRC already provides this.

# Type specific data blocks

Add text All to be defined

## 0x0001 - test packet

The test packet has valid packet headers and trailers but may consist of a custom and configurable payload.

## xxxxx Status packet

Can we have the occasionally sent packet with 64 bit transmission counters allowing us to detect packet loss more accurately?

We will send a keep-alive packet at a rate of ~1per second. The sequence counter tells you if there is packet loss. What else do you think would help?

## 0xXXXX - Multigrid readout

Just added as an example. We should choose a suitable target here.

Steven’s ADC design could serve as the most trivial example, if required.

# References