# Detector Readout Data Packing Format Draft Document

Steven Alcock, Detector Group
01/02/18

## 1. Introduction

This document describes an example readout format implemented in the Detector Group (DG) Control/Timing demonstrator. It is intended to be representative of the final readout format and is hence a useful starting point for discussions with the DMSC and Instrument teams. This is not a formal interface control document and its contents are liable to change.

## 2. Network Layer

All readout data for a given instrument is sent to the DMSC using standard Ethernet, IP and UDP protocols. These protocols are well understood and facilitate the use of commodity switches and interface cards. Although there is no routing requirement for the packets, the use of IP and UDP is expected to simplify the data acquisition software. The backend DG FPGA only transmits data to the DMSC – there is no receive capability. This prohibits the use of related IP functionality such as ARP and DHCP. The link is only used to send data – configuration and control information is send over a separate interface not described in this document.

### 2.1 Ethernet Frames

Standard Ethernet frames are used:

| **Name** | **Size (bytes)** | **Comments** |
|---|---|---|
| Destination MAC Address | 6 | User configurable. |
| Source MAC Address | 6 | User configurable, or can be hard-coded to a device-specific value (for example, each Xilinx VCU118 board is shipped with its own MAC address). |
| Ethertype | 2 | Hard-coded to 0x0800, because only IPv4 packets are to be sent. |

The MTU of the Ethernet frames has yet to be agreed and will depend on FPGA resources. It will not be smaller than 1500 bytes. It will not be greater than 9216 bytes (this is the MTU of the MSN2100 switch).

*SJA for discussion with the DMSC: what significance (if any) the MAC addresses have in terms of readout topology.*

*SJA: SDK has mentioned a preference for fixed-length Ethernet packets. It is not clear to me whether this will make the implementation easier or harder, and I would prefer not to specify at this stage.*

## 2.2 IP Packets

Standard IPv4 headers are used:

| Name | Size (bytes) | Comments |
|---|---|---|
| Version and IHL | 1 | Always 0x45, because only IPv4 is used with no option fields. |
| DSCP and ECN | 1 | Not used (set to zero). |
| Total Length | 2 | IP packets will not be fragmented, so the total length will always be less than the Ethernet MTU. |
| Fragmentation Fields | 1 | Not used (set to zero). |
| Time To Live | 1 | Arbitrarily set to 5 because lower values caused Wireshark to flag a warning at the receiver. |
| Protocol | 1 | Hard-coded to 0x11, because only UDP frames are to be sent. |
| Checksum | 2 | Implemented to prevent packets being dropped by the switch or interface card. |
| Source IP Address | 4 | User configurable. |
| Destination IP Address | 4 | User configurable. |

*SJA for discussion with DMSC: what (if any) significance the IP addresses have in terms of readout topology.*

## 2.3 UDP Datagrams

Standard UDP headers are used:

| Name | Size (bytes) | Comments |
|---|---|---|
| Source Port | 2 | User configurable. |
| Destination Port | 2 | User configurable. |
| Length | 2 | This will match that of the IP packet, minus the IP header length. |
| Checksum | 2 | Not implemented (set to zero). |

Depending on FPGA implementation, padding bytes may be appended after the underlying IP packet for word-alignment purposes.

*SJA for discussion with DMSC: how we might use the UDP ports for load-balancing or other advanced NIC-ingestion techniques.*

## 3. Readout Master Packet Structure

The Readout Master Header is applied at the DG FPGA Master and encapsulates all front end readout data. The fields are as follows:

| **Name** | **Size (bytes)** | **Purpose** |
|---|---|---|
| Type | 2 | Defines the encoding of the underlying data modules. |
| Length | 2 | The length of the readout packet, including the header, in bytes. |
| Sequence | 2 | Modulo-16 counter which increments on a per-packet basis. |
| Reserved | 2 | Not used (set to zero). |
| Trailer | 4 | Checksum of entire packet, including headers *(SJA: possibly not necessary, implementation to be decided).* |

The underlying data will be received from upstream instrument-specific sub-systems. Its protocol and format is not within the scope of this document. The specific encoding of the “Type” field is therefore undefined at this stage.

Depending on FPGA implementation, padding bytes may be added after the underlying data for word-alignment purposes.

*SJA: we should define the function of keep-alives and/or timestamp status packets as appropriate.*