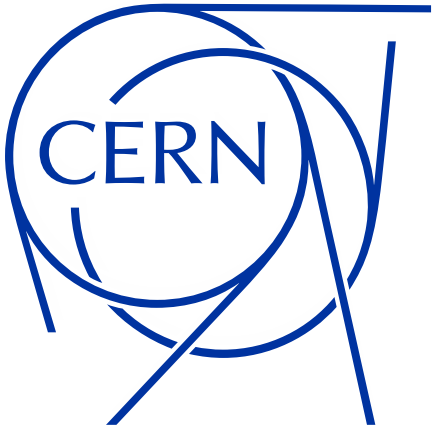


SY-EPC needs and proposals for µs-level synchronization

Electronics forum

Valerio Nappi for SY-EPC-CCE

October 31, 2024

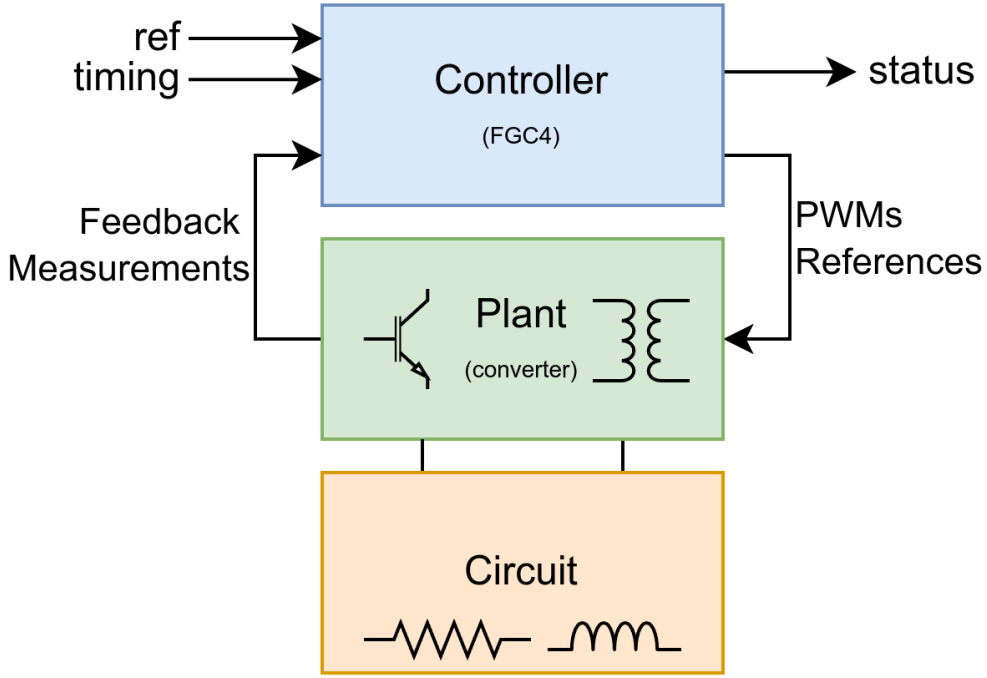
Tiniest recap possible on what power converter control does

About me

- I work in SY-EPC-CCE, where we develop power converter controls together with SY-EPC-CCS.
- I am working on gateware and SoC integration for FGC4, the next generation power converter controller

About power converter controllers

- The simplest power converter controller takes a reference value from a command and imposes it on a power converter
- A real power converter controller – such as FGC4 – is responsible for controlling a converter in much more complex scenarios (plus function generation, logging of signals, configurations...)
- The power converter controller interfaces with the technical infrastructure (TN, BIS, etc..) and the converter itself (power devices gates, voltages and currents...)



Tiniest recap possible on FGC4

Hardware

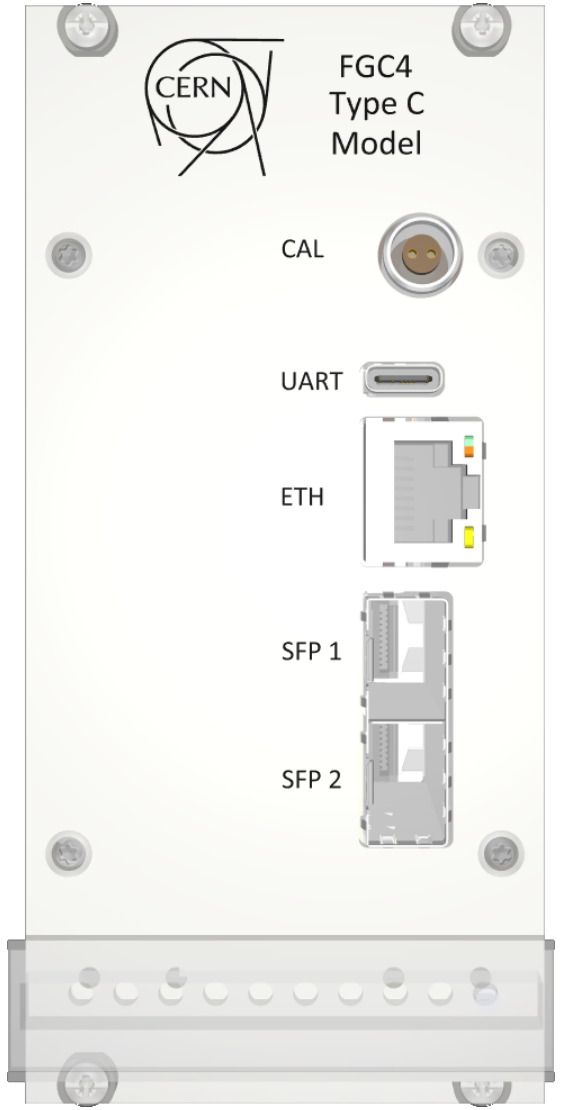
- Based on the shared SoM that we heard of today;
- Will be using the same SoC as the DI/OT.

Software

- FGC4 will act as it’s own FEC, so the FGC4 will be connected to technical network;
- FGC4 will receive UTC time from **PTP over technical network**.

Controlled converter systems

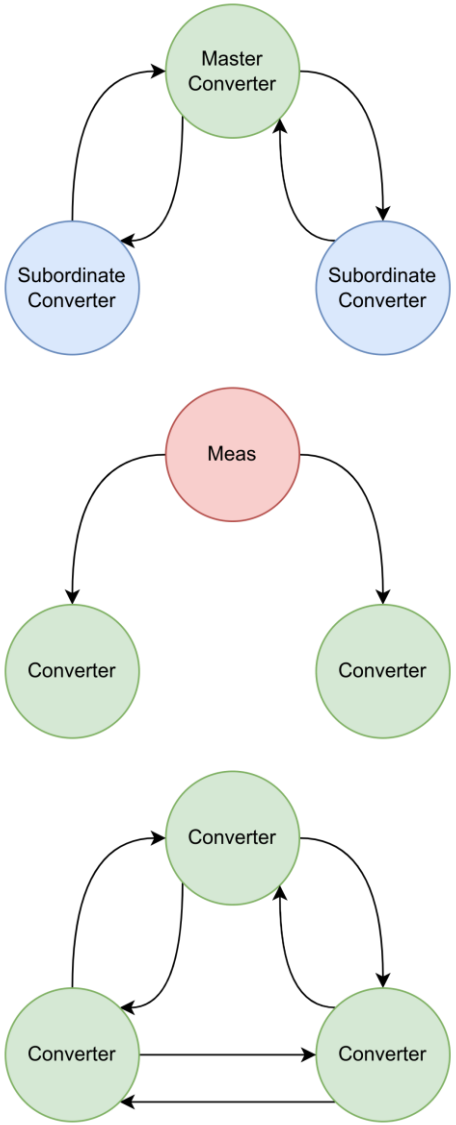
- FGC4 will be required to control a wide variety of power converter systems, including **multi-converter power systems**, where many FGCs will coordinate multiple converters to power a circuit;
- **Synchronization of data across converter and shared timing** is particularly critical;



Networking needs for power systems

Despite most converters operating independently, many cases exist where some real time communication is required for their operation:

- **Master-Slave converters** (parallel and series converters): A single coordinator controller gathers feedback data from all subordinates, and returns the actuation to all;
- **Magnetic measurements:** The B-field measurement is distributed in the accelerators on a White Rabbit Stream link, and the power converters can receive it to close the control loop with the B-field as feedback;
- **Decoupling** (symmetric and asymmetric): In some cases, a converter's magnetic circuit influences another converter's magnetic circuit. Nodes can exchange data to decouple themselves;
- **Orbit and tune correction:** a power converter can be used to apply orbit and tune corrections in an accelerator. A centralize system calculates the correct



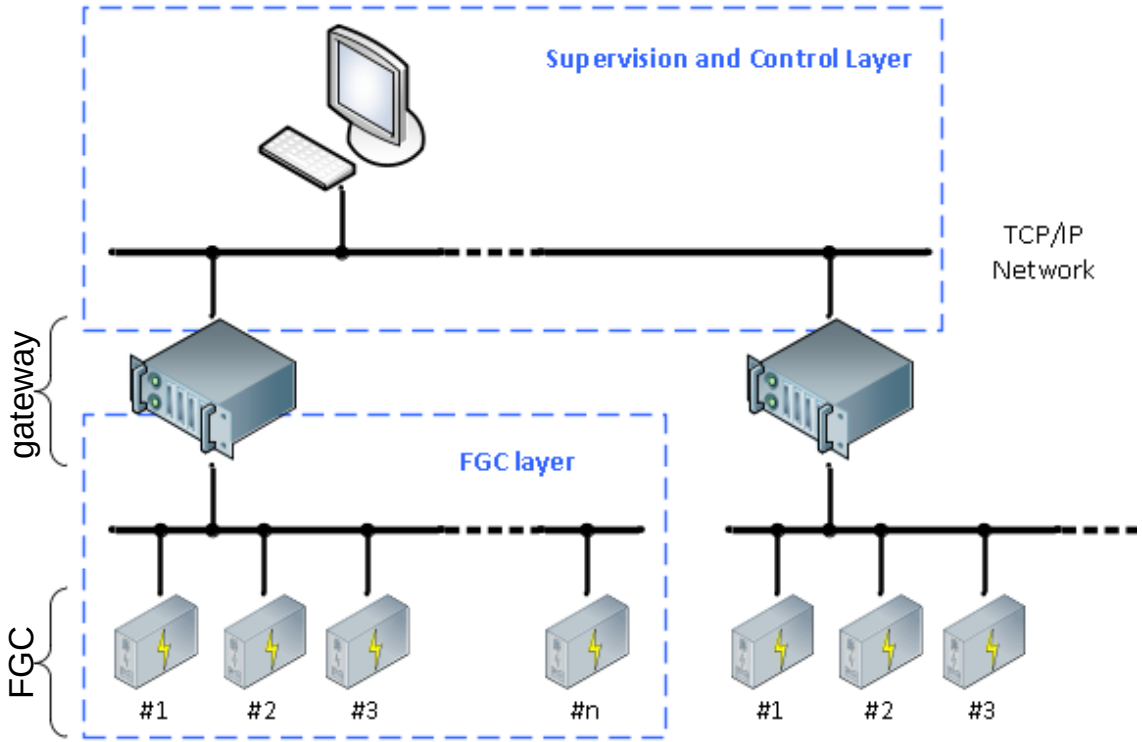
FGC_Ether: previous generation

In the older generations of power converter controllers in the LHC (FGC2, FGClite), the **WorldFip** fieldbus was used. For FGC3 in the injectors, this was replaced in the by **FGC_Ether**.

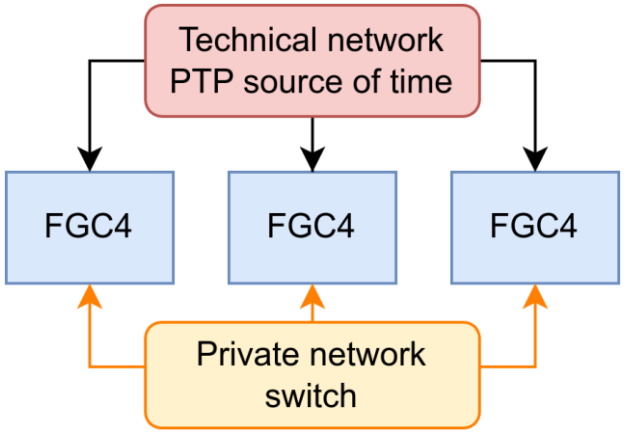
- FGC_Ether is a non-standard extension of 100 Mbps Ethernet, where one unused pair is used to distribute 50Hz synchronization pulses;

FGC_Ether has it's shortcomings:

- It was designed with the **FGC-Gateway** communication in mind, but not really for **inter-FGC** communication. Command data reception would slow down real-time data;
- It has a fixed **1ms period,** and each FGC can construct **only one packet per ms**, though it be broadcast or sent multiple times to different consumers**;**



FGC4 networking architecture



Given the experience with FGCEther, we wanted to decouple the **command traffic** from the **real time inter-FGC traffic**. An **FGC Private Network (PN)** was proposed.

Which protocol for PN? We need at least **HW timestamping** to measure latency, **ports** to differentiate information channels and **sequence numbers** for packets.

White rabbit?

- Has HW timestamps, has sequence numbers, no ports;
- Pros: We use it for Btrain anyways, centrally supported, readily available, cool;
- Cons: No ports or multiplexing of information, supports only one stream of data.

EtherCAT?

- Has ports (datagrams), no HW timestamps, no sequence numbers;
- Pros: Proven industry standard, open source (IEC)
- Cons: Proprietary implementations only for the PL, only one master, chain topology has little flexibility;

FGC4 networking architecture - 2

UDP:

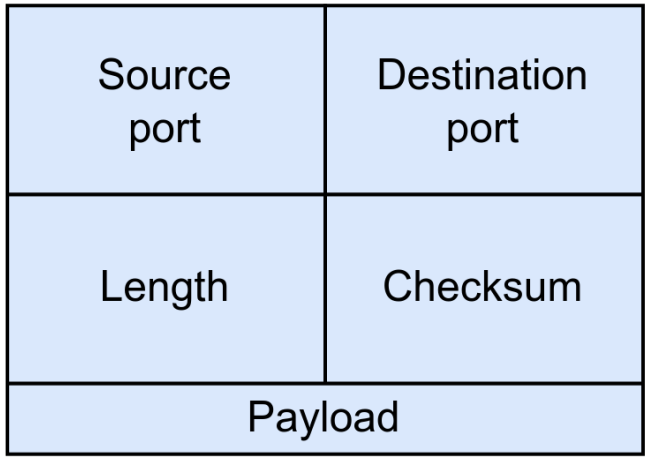
- Has ports, no timestamps, no sequence number;
- Pros: Very standard and widespread protocol, FPGA Implementations are very common;
- Cons: Needs IP, which adds a layer of complexity (and latency?) at no advantage, for such static networks. Doesn't have the features.

DIY protocol on raw Ethernet:

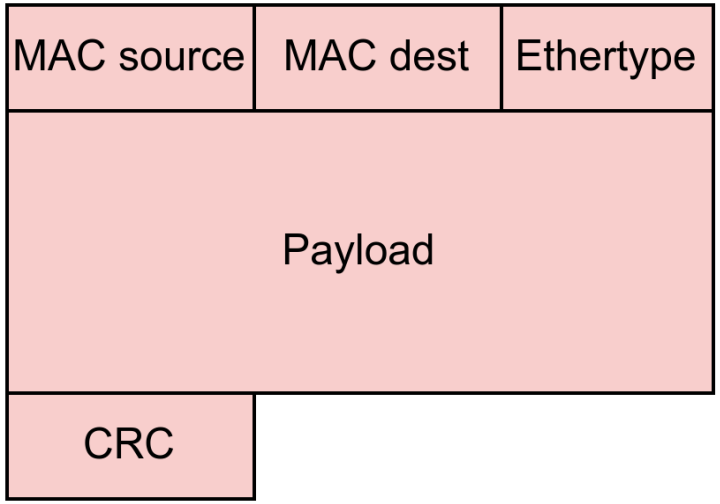
- Can have ports, timestamps, sequence numbers, at will
- Pros: Full flexibility, can exploit very cheap and fast Ethernet equipment;
- Cons: We have to implement it.

Plan: Propose a real time, low latency protocol that suits our needs (and somebody else's?), modular enough that it can be reused

UDP



Ethernet



FGC Private Network stack proposal

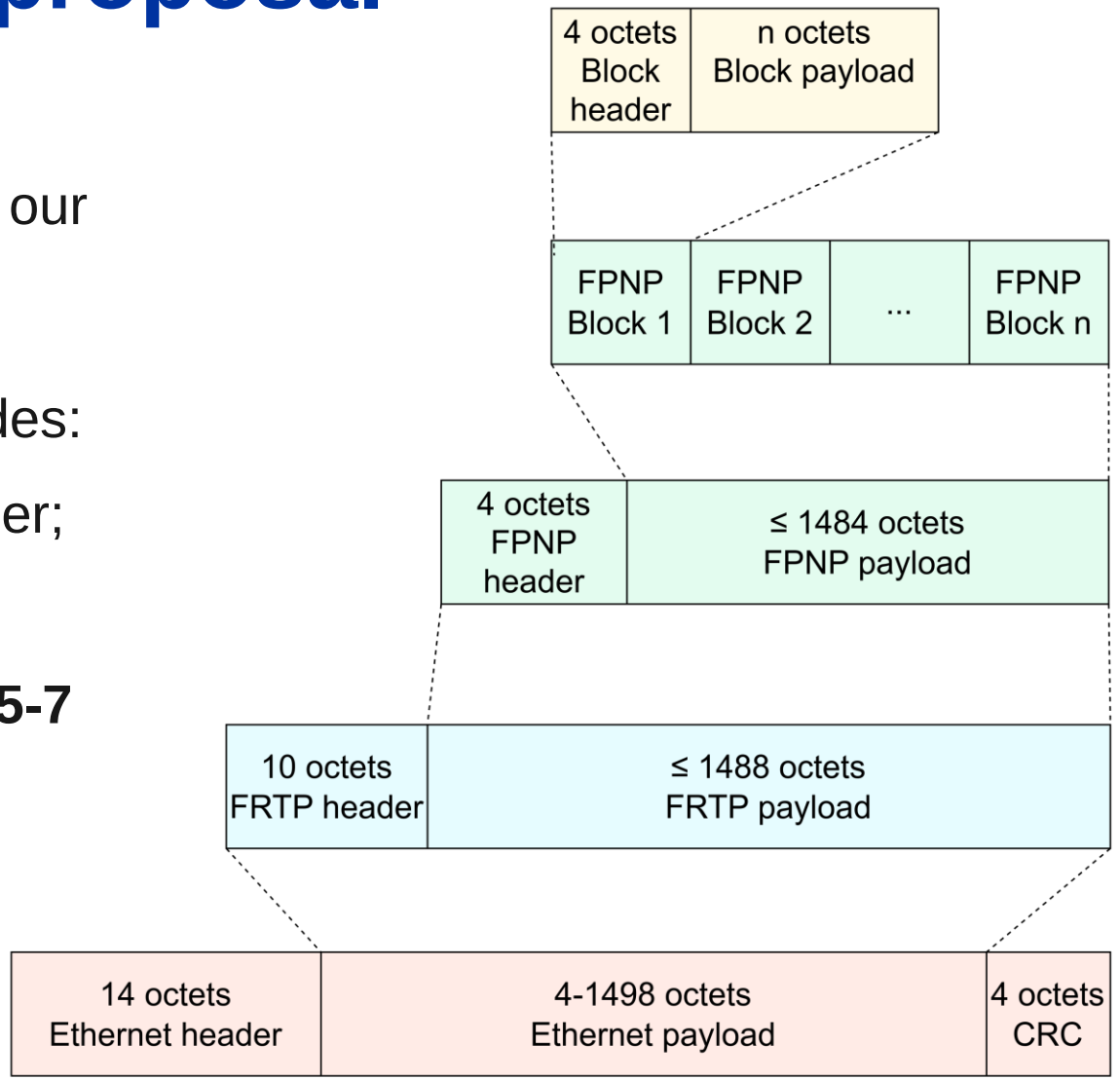
The protocol is heavily influenced by the fact that our control runs on a SoC. It is made of two layers:

FGC Real Time Protocol (FRTP) – OSI L4
This layer sits directly on top of Ethernet. It provides:

- The concept of **channel**. Similar to a port number;
- **Hardware timestamping.**

FGC Private Network Protocol (FPNP) – OSI L5-7
Sits on FRTP, provides:

- **Packet numbering** (OSI L5);
- **Blocks**, as organized units of data (OSI L6-7).



FRTP – PL layer protocol

Channel

A channel identifies a **stream** of information. A channel is linked to a **payload's programmable memory address**, both in the transmitter and in the receiver.

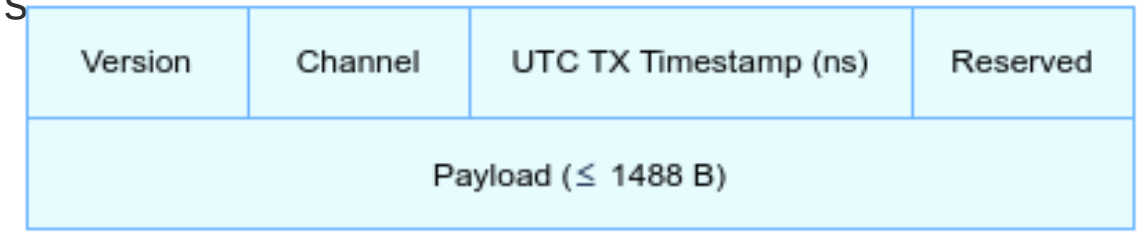
- On the **transmitter** side, a DMA fetches the payload and transmits it.
- On the **receiving** side, once received the payload is DMAed to a pre-programmed address for that channel.

Up to 256 channels are supported, depending on the implementation. The FRTP lives entirely in the PL. The FRTP IP generates latency statistics on a per-channel basis, exploiting the TX timestamp and the time of arrival.

Fast data extraction – planned

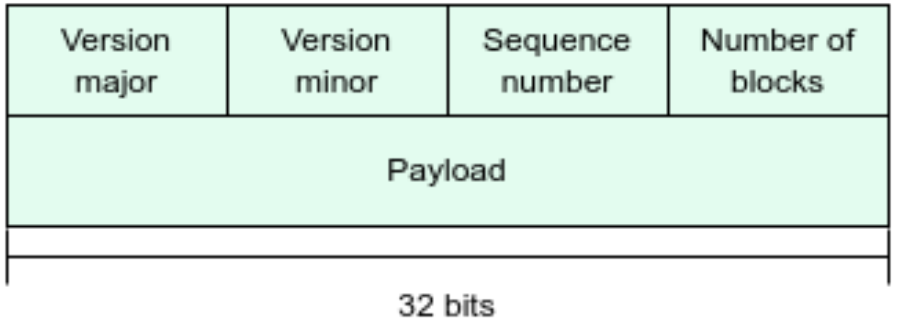
For each channel, it will be possible to specify a certain offset, at which a 32-bit word is extracted from the packet and made available to PL components on an AXI-Stream. This will enable very fast transfers to actuators, for example in case of Master-Slave converters

The FRTP protocol is not specific to power converters.



FPNP – PS layer protocol

FPNP structure

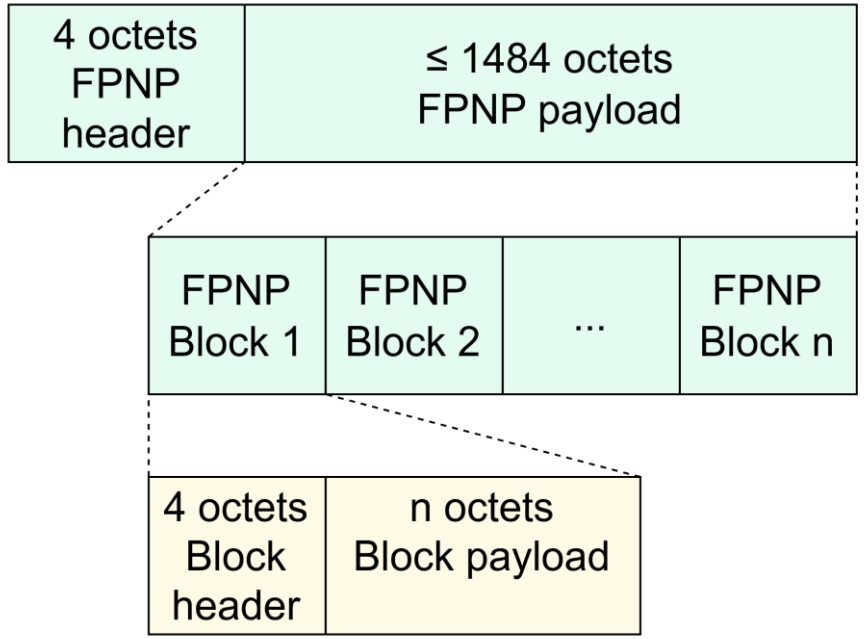


Sequence number

Each frame of the FPNP protocol contains a sequence number. This helps in the detection of missed packets, that can trigger warnings or faults.

Block number

The FPNP revolves around the concept of blocks. A block encapsulates a single unit of information, each frame can contain multiple blocks, up to 256 or until the frame is full.



Block header

Each block has a block header, which contains a block type and a length in bytes. The block type identifies a *struct* that can be used to parse the block. The block types are defined by Cheby-like memory maps.

Adopters can define their own blocks, although interoperatibility will not be guaranteed

home.cern