Chapter 7

Data Acquisition

7.1 Introduction

The far detector (FD) data acquisition (DAQ) system is responsible for receiving, processing, and recording data from the DUNE FD. In doing so, it provides timing and synchronization for all detector modules and subdetectors; receives, synchronizes, compresses, and buffers streaming data from the subdetectors; extracts information from the data at a local level to subsequently make local, module, and cross-module data selection decisions; builds event records from selected space-time volumes and relays them to permanent storage; and carries out local data reduction and filtering of the data as needed.

This chapter provides a description for the design of the DUNE FD DAQ system developed by the DUNE FD DAQ consortium. This consortium brings together resources and expertise from CERN, Colombia, France, Japan, the Netherlands, the UK, and the USA. Its members bring considerable experience from ICARUS, MicroBooNE, SBND, and DUNE prototype LArTPCs, as well as from ATLAS at the LHC and other major HEP experiments across the world.

The system is designed to service all FD detector module designs indistinguishably. However, some aspects of the DAQ design are tailored to meet module-specific requirements, and those are documented in sections of this chapter which are unique to the detector module covered in this TDR volume; these sections are identifiable by their use of module-specific terms. In general, the DAQ services each FD detector module independently, but cross-module communication is facilitated at the trigger level.

The chapter begins with an overview of the DAQ design, including requirements that the design must meet, and specification of interfaces between the DAQ and other DUNE FD systems. Sub-
sequently, Section 7.4, which comprises the bulk of this chapter, describes the design of the FD DAQ in greater detail. Section 7.5.1 describes design validation efforts to date, and future design development and validation plans. At the center of these efforts is the ProtoDUNE DAQ system, which has served as a demonstrator of several key aspects of the DUNE DAQ design, and continues to serve as a platform for further design development and validation. Finally, the chapter finishes with two sections providing details on the management of the DAQ project, including schedule to completion of the design, production, and installation of the system, as well as cost, resources, and safety considerations.

7.2 Design Overview

An overview of the DUNE FD DAQ system servicing a single FD detector module is provided in Fig. 7.1. The system is physically located at the FD site, and it is split between the 4850 ft level and the ground level at SURF. Specifically, it occupies space and power both in the central utility cavern (CUC) and the on-surface DAQ room. The front-end part of the system, which is responsible for raw detector data reception and pre-processing, lives underground in the CUC, while the back-end part of the system, which is responsible for event-building as well as run control and monitoring, lives on the surface. Data flows through the DAQ from the front-end to the back-end of the system and to offline. The majority of raw data processing and buffering is performed underground, in the front-end part of the system, thus minimizing data bandwidth to the surface. A hierarchical data selection subsystem consumes minimally-processed information from the front-end readout, and constructs module-level trigger decisions. Upon such decision, a data flow orchestrator process is activated as part of the back-end part of the system to retrieve data to be built as part of an event record. At event building stage, optional down-selection of the data is possible via high-level filtering, prior to shipping the data to offline. The specifics of design implementation and data flow are described in Section 7.4.

Figure should perhaps be modified to indicate low level data selection. Some rewording on boxes is also needed to match subsystem definitions. Georgia

7.2.1 Requirements and Specifications

The DUNE FD DAQ system is designed to meet the DUNE top-level as well as DAQ-level requirements summarized in Table 7.2. The DAQ-level requirements are imposed to ensure that the system can record all necessary information for offline analysis of data that is associated with on- and off-beam physics events, as directed by the DUNE physics mission, and with minimal compromise to DUNE’s physics sensitivity. The requirements must be met by following the specifications provided in the same table. Those specifications are associated with trigger functionality, readout considerations, and operations considerations, and are motivated further in the following subsections.
Figure 7.1: DAQ design physical layout focusing on a single 10 kt module. Not shown in this figure are the system control paths.

7.2.1.1 How DUNE’s Physics Mission Drives the DAQ Design

The DUNE Far Detector has three main physics drivers: neutrino charge-parity symmetry violation (CPV) and related long baseline oscillation studies using the high intensity beam provided by Fermilab, off-beam measurements of atmospheric neutrinos and searches for rare processes such as baryon-number-violating decays, and detection of supernova neutrino burst (SNB) occurring within our galaxy. The DUNE FD DAQ system must facilitate data readout for delivering on these main physics drivers, while keeping within physical (space, power) and resource constraints for the system. In particular the off-beam measurements require the continuous readout of the detector, and the lack of external triggers for such events requires real-time or online data processing, and self-triggering capabilities. Since the continuous data rate of the far detector module reaches multiple terabytes per second, significant data buffering and processing resources are needed as part of the design.

The DUNE FD modules employ two active detector components from which the DAQ system must acquire data: time projection chamber (TPC) and photon detection system (PDS). The two components access the physics by sensing and collecting signals associated with very different sensing time scales. Ionization charge measurement by the time projection chamber (TPC) for any given localized activity in the detector requires a nominal recording of data over a time window of order 1 ms to 10 ms. This time scale is determined by the ionization electron drift speed in LAr and the detector dimension along the drift direction. On the other hand, the photon detection system (PDS) measures argon scintillation light emission, which occurs and is detected over a timescale of multiple nanoseconds to microseconds for any given event and/or subsequent subevent process. Unlike the TPC, the PDS data is zero-suppressed in the PDS electronics (see Chapter ??); therefore the total raw data volume received by the DAQ system is be dominated by the TPC data, which is sent out as a continuous stream.
Figure 7.2 provides the expected activity rates in a single far detector module as a function of true energy associated with given types of signal. At low energy (<10 MeV), activity is dominated by radiological backgrounds intrinsic to the detector, and low-energy solar neutrino interactions. Supernova burst neutrinos are would span the 10–30 MeV range, while at higher energies (generally above 100 MeV), rates are dominated by cosmic rays, beam neutrino interactions, and atmospheric neutrino interactions. With the exception of supernova burst neutrinos, the activity associated with any of these physics signals is localized in space and particularly in time. Supernova burst neutrinos on the other hand are characteristically different, as they arrive as multiple signals of localized activity that extend over the entirety of the detector and over multiple seconds.

The nature and rates of these signatures necessitates a data selection strategy which handles two distinct cases: a localized high energy activity trigger, prompting an event record readout for activity associated with a minimum of 100 MeV of deposited energy; and an extended low-energy activity trigger, prompting an event record readout when multiple localized low energy activity candidates with a minimum of 10 MeV of deposited energy each are found over a short (less than 10 seconds) time period and over the entirety of a 10 kton module. Because of the high granularity of the detector readout elements, a hierarchical data selection subsystem is employed to provide data processing and triggering, and facilitate optional data reduction and filtering. The DAQ system is required to yield >99% efficiency for localized high-energy activity triggers, and sufficient efficiency for low-energy activity trigger candidates as needed to achieve >90% galactic supernova burst trigger coverage. The galactic coverage is defined as supernova burst trigger efficiency weighted supernova burst probability.

By offline considerations, the steady state rate of localized triggers from the entire far detector is limited to 0.1 Hz, otherwise more than 30 PB of data (uncompressed) would be generated per year. This assumes (conservatively) that each localized trigger prompts 5.4 ms of losslessly compressed TPC data plus PDS data from the entire module to be read out as part of the event record. The average rate of extended triggers is limited to 1 per month, per similar considerations; this assumes that an extended trigger prompts 100 s of losslessly compressed data from the entire module to be read out as part of the event record. The capability of recording data losslessly is built into the design as a conservative measure; a particular concern is charge collection efficiency in the case of zero suppression. MicroBooNE is currently investigating the impact of zero suppression on reconstruction efficiency and energy resolution for low-energy events. Expected data rates from physics signals of interest, which fit the 30 PB yearly generated volume and trigger rate requirements, are summarized in Table 7.1.

Self-triggering on supernova neutrino burst (SNB) activity is a unique challenge for the DUNE FD, and an aspect of the design which has never been demonstrated in a LArTPC. The challenge with SNB triggering is two-fold. First, the activity of the individual SNB neutrino interactions is expected to be of relatively low energy (10 MeV to 30 MeV), often indistinguishable from radiological background activity in the detector. Triggering on an ensemble of O(100) events expected on average in the case of a galactic supernova burst is therefore advantageous; however, since this ensemble of events is expected to occur sparsely over the entire detector and over an extended period of O(10) s, sufficient buffering capability must be designed into the system. Furthermore, to assure high efficiency in collecting SNB interactions that individually are below individual interaction activity threshold, data from all channels will be recorded over an extended and contiguous period of time O(100) s around every SNB trigger.
Figure 7.2: Expected physics-related activity rates in a single 10 kt module.

Table 7.1: Summary of expected data rates. The rates assume no compression, and are given for a single 10 kt module. Trigger primitives are not kept permanently; they are temporarily stored for 1-2 months.

<table>
<thead>
<tr>
<th>Source</th>
<th>Annual Data Volume</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam interactions</td>
<td>27 TB</td>
<td>10 MeV threshold in coincidence with beam time, including cosmic coincidence; 5.4 ms readout</td>
</tr>
<tr>
<td>Cosmics and atmospheric neutrinos</td>
<td>10 PB</td>
<td>5.4 ms readout</td>
</tr>
<tr>
<td>Radiological backgrounds</td>
<td>&lt; 1 PB</td>
<td>&lt; 1 per month fake rate for SNB trigger</td>
</tr>
<tr>
<td>Cold Electronics calibration</td>
<td>200 TB</td>
<td></td>
</tr>
<tr>
<td>Radioactive source calibration</td>
<td>100 TB</td>
<td>&lt; 10 Hz source rate; single APA readout; 5.4 ms readout</td>
</tr>
<tr>
<td>Laser calibration</td>
<td>200 TB</td>
<td>$10^6$ total laser pulses; half the TPC channels illuminated per pulse; lossy compression (zero-suppression) on all channels</td>
</tr>
<tr>
<td>Random triggers</td>
<td>60 TB</td>
<td>45 per day</td>
</tr>
<tr>
<td>Trigger primitives</td>
<td>13 PB</td>
<td>Dominated by $^{39}$Ar (50 kHz per APA face); collection channels only; 20 bytes per trigger primitive</td>
</tr>
</tbody>
</table>
Table 7.2: Specifications for SP-DAQ

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>Specification (Goal)</th>
<th>Rationale</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP-FD-1</td>
<td>Minimum drift field</td>
<td>&gt; 250 V/cm ( &gt; 500 V/cm )</td>
<td>Lessens impacts of $e^-$-Ar recombination, $e^-$ lifetime, $e^-$ diffusion and space charge.</td>
<td>ProtoDUNE</td>
</tr>
<tr>
<td>SP-FD-2</td>
<td>System noise</td>
<td>&lt; 1000 $e^-$</td>
<td>Provides &gt;5:1 S/N on induction planes for pattern recognition and two-track separation.</td>
<td>ProtoDUNE and simulation</td>
</tr>
<tr>
<td>SP-FD-3</td>
<td>Light yield</td>
<td>$&gt; 20$ PE/MeV (avg), $&gt; 0.5$ PE/MeV (min)</td>
<td>Gives PDS energy resolution comparable to the TPC for 5-7 MeV SN vs, and allows tagging of &gt; 99% of nucleon decay backgrounds with light at all points in detector.</td>
<td>Supernova and nucleon decay events in the FD with full simulation and reconstruction.</td>
</tr>
<tr>
<td>SP-FD-4</td>
<td>Time resolution</td>
<td>&lt; 1 µs ( &lt; 100 ns )</td>
<td>Enables 1 mm position resolution for 10 MeV SNB candidate events for instantaneous rate &lt; 1 m$^{-3}$ms$^{-1}$.</td>
<td></td>
</tr>
<tr>
<td>SP-FD-5</td>
<td>Liquid argon purity</td>
<td>&lt; 100 ppt ( &lt; 30 ppt )</td>
<td>Provides &gt;5:1 S/N on induction planes for pattern recognition and two-track separation.</td>
<td>Purity monitors and cosmic ray tracks</td>
</tr>
<tr>
<td>SP-FD-12</td>
<td>Cathode HV power supply ripple contribution to system noise</td>
<td>&lt; 100 $e^-$</td>
<td>Maximize live time; maintain high S/N.</td>
<td>Engineering calculation, in situ measurement, ProtoDUNE</td>
</tr>
<tr>
<td>SP-FD-13</td>
<td>Front-end peaking time</td>
<td>1 µs  (Adjustable so as to see saturation in less than 10% of beam-produced events)</td>
<td>Vertex resolution; optimized for 5 mm wire spacing.</td>
<td>ProtoDUNE and simulation</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------</td>
<td>---------------------------------</td>
<td>--------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>SP-FD-16</td>
<td>Detector dead time</td>
<td>&lt; 0.5%</td>
<td>Meet physics goals in timely fashion.</td>
<td>ProtoDUNE</td>
</tr>
<tr>
<td>SP-FD-19</td>
<td>ADC sampling frequency</td>
<td>~ 2 MHz</td>
<td>Match 1 µs shaping time.</td>
<td>Nyquist requirement and design choice</td>
</tr>
<tr>
<td>SP-FD-20</td>
<td>Number of ADC bits</td>
<td>12 bits</td>
<td>ADC noise contribution negligible (low end); match signal saturation specification (high end).</td>
<td>Engineering calculation and design choice</td>
</tr>
<tr>
<td>SP-FD-22</td>
<td>Data rate to tape</td>
<td>&lt; 30 PB/year</td>
<td>Cost. Bandwidth.</td>
<td>ProtoDUNE</td>
</tr>
<tr>
<td>SP-FD-23</td>
<td>Supernova trigger</td>
<td>&gt; 90% efficiency for SNB within 100 kpc</td>
<td>&gt;90% efficiency for SNB within 100 kpc</td>
<td>Simulation and bench tests</td>
</tr>
<tr>
<td>SP-FD-25</td>
<td>Non-FE noise contribu-</td>
<td>&lt;&lt; 1000 e⁻</td>
<td>High S/N for high reconstruction efficiency.</td>
<td>Engineering calculation and Proto-DUNE</td>
</tr>
<tr>
<td></td>
<td>tions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP-FD-27</td>
<td>Introduced radioactivity</td>
<td>Less than that from $^{39}$Ar</td>
<td>Maintain low radiological backgrounds for SNB searches.</td>
<td>ProtoDUNE and assays during construction</td>
</tr>
<tr>
<td>SP-FD-28</td>
<td>Dead channels</td>
<td>&lt; 1%</td>
<td>Contingency for possible efficiency loss for &gt; 20 year operation.</td>
<td>ProtoDUNE</td>
</tr>
<tr>
<td>SP-DAQ-1</td>
<td>Off-beam High-energy Trigger</td>
<td>&gt;100 MeV</td>
<td>Driven by DUNE physics mission.</td>
<td>Simulations</td>
</tr>
<tr>
<td>SP-DAQ-2</td>
<td>Off-beam Low-energy Trigger</td>
<td>&gt;10 MeV</td>
<td>Driven by DUNE physics mission.</td>
<td>Simulations</td>
</tr>
<tr>
<td>SP-DAQ-3</td>
<td>Beam Trigger</td>
<td>&gt;100 MeV</td>
<td>Driven by DUNE physics mission.</td>
<td>Simulations, experience from past and ongoing experiments.</td>
</tr>
<tr>
<td>SP-DAQ-4</td>
<td>Calibration Trigger</td>
<td>Need to understand detector performance.</td>
<td>Experience from past and ongoing experiments.</td>
<td></td>
</tr>
</tbody>
</table>
7.2.1.2 Practical Considerations for Design

The DAQ system is designed as a single, scalable system which can service all FD modules. It is also designed on the principle that the system should be able to record and store full detector data with zero dead time; and that it should be evolutionary, taking advantage of the staged construction for the DUNE FD, and thus beginning very conservatively for the first DUNE FD module, and aggressively reducing the design conservatism as further experience is gained with detector operations. At the same time, it is designed to preserve the possibility of additional capacity as required. The bulk of processing and buffering of raw detector data is done underground, in the front-end part of the system (see Figure 7.1), in order to minimize data traffic to surface. Power, cooling, and space constraints in the CUC are limited to 600 kW total and 52 racks for all four FD modules.

There are three key challenges for the DUNE FD DAQ system:

- First, the system must accommodate a long (“permanent”) commissioning state for the far detector, and must therefore be a fully “partitionable” system.

  Given operational considerations, and in particular based on the need to minimize SNB dead time, partitioning the DAQ system allows a significant portion of the detector to remain physics-operational even if a fault interrupts data collection in some part of the detector. This partitionable operation mode also permits detector development and specialized runs (e.g., calibrations) to run in parallel with normal physics data taking, for small subsets of the detector.

- Secondly, the SNB physics requirements necessitate large buffering in the upstream DAQ and low fake supernova burst trigger rates.

  The implementation of a continuous storage element in the data flow architecture allows for the formation and capture of delayed, data-driven trigger decisions with minimal loss of physics information. The specification for this look-back buffer is set in consultation with physics groups. It is driven primarily by the need to record up to ten seconds of unbiased data preceding a SNB (with the neutronization time taken as the time of the burst), and it is specified to be greater than four seconds. This four-second buffering provision works in tandem with a trigger latency specification of less than one second. This aspect remains to
be validated with simulation, to ensure that high coverage (greater than 90%) for galactic SNBs is achieved by the SNB trigger.

The DAQ system is also designed so as to be able to apply lossless compression to these records, as well as filter them to remove unnecessary data regions in an intelligent way, i.e. without compromising physics performance.

A programmable trigger priority scheme ensures that the readout for the main physics triggers is never or rarely inhibited so as to enable easy determination of the live-time of these triggers. At the same time, generation of overlapping triggers will be possible, and ordering and prioritization will prevent data readout duplication.

- Finally, the difficult-to-access location requires that the DAQ operates with high reliability and fully remote operation.

Furthermore, to ensure minimal impact to overall detector live-time, the DAQ system is fully configurable, controllable, and operable from remote locations, with authentication implemented to allow exclusive control. It furthermore facilitates online monitoring of the detector and of itself.

### 7.2.2 Summary of Key Parameters

Table 7.3 summarizes the important parameters driving the DAQ design. These parameters set the scale of data buffering, processing, and transferring resources which must be built into the design of each FD module.

Table should use standard latex for numbers and units. Anne

### 7.3 Interfaces

The DAQ system scope begins at the optical fibers streaming raw digital data from the detector active components (TPC and PDS), and ends at a wide area network (WAN) interface that distributes the data from on site at SURF to offline centers off site. The DAQ also provides common computing and network services for other DUNE systems, although slow control and safety functions fall outside DAQ scope.

Consequently, the DUNE FD DAQ system interfaces with the TPC cold electronics (CE), PDS readout, computing, cryogenic instrumentation and slow controls (CISC), and calibration systems of the FD, as well as with facilities and underground installation. The interface agreements with
Table 7.3: Summary of important parameters driving the DAQ design.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPC Channel Count per Module</td>
<td>284,000</td>
</tr>
<tr>
<td>TPC Collection Channel Count per Subdetector (APA)</td>
<td>960</td>
</tr>
<tr>
<td>TPC Induction Channel Count per Subdetector (APA)</td>
<td>1,600</td>
</tr>
<tr>
<td>PDS Channel Count per Module</td>
<td>TBD</td>
</tr>
<tr>
<td>TPC analog-to-digital converter (ADC) Sampling Rate</td>
<td>2 MHz</td>
</tr>
<tr>
<td>TPC ADC Dynamic Rate</td>
<td>12 bits</td>
</tr>
<tr>
<td>Localized Event Record Window</td>
<td>5.4 ms</td>
</tr>
<tr>
<td>Extended Event Record Window</td>
<td>100 s</td>
</tr>
<tr>
<td>Full size of TPC Localized Event Record per Module</td>
<td>6.22 GB</td>
</tr>
<tr>
<td>Full size of TPC Extended Event Record per Module</td>
<td>115 TB</td>
</tr>
</tbody>
</table>

the FD systems are summarized in Table 7.4, and described briefly in the following subsections. Interface agreements with facilities and underground installation are described in Section 7.6.

Table 7.4: Data Acquisition System Interface Links

<table>
<thead>
<tr>
<th>Interfacing System</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPC CE</td>
<td>TPC CE DocDB 6742 [51]v6</td>
<td></td>
</tr>
<tr>
<td>PDS</td>
<td>PDS DocDB 6727 [73]v2</td>
<td></td>
</tr>
<tr>
<td>Integration Facility</td>
<td>Integration Facility DocDB 7042 [?]v0</td>
<td></td>
</tr>
<tr>
<td>Facilities</td>
<td>Facilities DocDB 6988 [?]v1</td>
<td></td>
</tr>
<tr>
<td>CISC</td>
<td>CISC DocDB 6790 [86]v1</td>
<td></td>
</tr>
<tr>
<td>Calibration</td>
<td>Calibration DocDB 7069 [?]</td>
<td></td>
</tr>
<tr>
<td>Computing</td>
<td>Computing DocDB 7123 [87]</td>
<td></td>
</tr>
<tr>
<td>Timing</td>
<td>Timing DocDB 11224 [?]</td>
<td></td>
</tr>
</tbody>
</table>

7.3.1 TPC Cold Electronics

The DAQ and TPC CE interface is described in DocDB 6742 [51]. The physical interface is at the central utility cavern (CUC), where optical links from the warm interface boards (WIBs) transfer the raw TPC data to the DAQ front-end (FE) readout (Front-End Link eXchange (FELIX); see Section ??). This ensures the DAQ is electrically decoupled from the detector cryostat. Ten 10 Gbps links are expected per anode plane assembly (APA), and have been specified as 300m OM4 multimode fibers from small form-factor pluggable (SFP) (SFP)+ at the WIB to miniature parallel optical device (MiniPOD) on FELIX. The data format has been specified to use no compression and custom communication protocol.
7.3.2 PDS Readout

The DAQ and PDS readout interface is described in DocDB 6727 [73]. It is anticipated to be of the form of 150 10 Gbps OM4 fibers from one FD module. This is similar to the interface to the TPC CE, except the overall data volume is lower by an order of magnitude. The data format has been specified to use compression (zero suppression) and custom communication protocol.

7.3.3 Computing

The DAQ and computing interface is described in DocDB 7123 [87].

The computing consortium is responsible for the online areas of WAN connection between SURF and Fermilab, while the DAQ consortium is responsible for disk buffering to handle any temporary WAN disconnects and the infrastructure needed for real-time data quality monitoring. The computing consortium is also responsible for the offline development and operation of the tools for data transfers to Fermilab. The primary constraint in defining the DAQ and offline computing interface is the requirement to produce less than 30 PB/year for transfer to Fermilab. DAQ and computing consortia are jointly responsible for data format definition and data access libraries, as well as real-time data quality monitoring software. The former is specified in the form of a data model documented in DocDB ?? [?].

7.3.4 CISC

The DAQ and CISC interface is described in DocDB 6790 [86]. The DAQ provides a network in the CUC for CISC, operation information and hardware monitoring information to CISC, and power distribution and rack status units in DAQ racks. The information from CISC feeds back into the DAQ for run control operations.

7.3.5 Calibration

The DAQ and calibration interface is described in DocDB 7069 [?]. Two calibration systems are envisioned for the FD: a laser calibration system and a neutron generator. Calibration pulses can be issued either by the DAQ or by the calibration systems themselves; the latter are to be distributed through the DAQ timing system.
7.3.6 Timing Subsystem

The timing system of the DUNE FD connects with almost all detector systems and with the calibration system and has a uniform interface to each of them. A single interface document, DocDB 11224 [?], describes all these timing interfaces.

Accuracy of timestamps delivered to detector endpoints will be ±500 ns with respect to UTC. Synchronization between any two endpoints in the detector will be better than 10 ns on average. Between detector modules, synchronization will be better than 25 ns on average. The timing system will also provide a synchronized clock source by which DAQ computer system clocks may be synchronized using standard network time protocols. System clocks are expected to be synchronized to within a ms using NTP and µs using PTP standards.

7.4 Data Acquisition System Design

This section begins with an overview of the DAQ design followed by brief descriptions of the subsystem design implementation specific. The implementation details are evolving rapidly; as such, more information is provided in technical notes as listed in Table 7.5.

Table 7.5: Summary of detailed DAQ technical notes.

<table>
<thead>
<tr>
<th>Title</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUNE FD Data Volumes</td>
<td>DocDB 9240 [?]</td>
</tr>
<tr>
<td>The DAQ for the DUNE prototype at CERN</td>
<td>DocDB 8708 [?]</td>
</tr>
<tr>
<td>A System for Communication Between DAQ Elements</td>
<td>DocDB 10482 [?]</td>
</tr>
<tr>
<td>Data Selection for DUNE Beam and Atmospheric Events</td>
<td>DocDB 11215 [?]</td>
</tr>
<tr>
<td>Data orchestrator and event building for DUNE FD DAQ</td>
<td>t.b.d.</td>
</tr>
<tr>
<td>DUNE Run Control, Configuration &amp; Monitoring (CCM)</td>
<td>t.b.d.</td>
</tr>
<tr>
<td>DUNE DAQ Readout</td>
<td>t.b.d.</td>
</tr>
<tr>
<td>DUNE FD Timing and Synchronization System</td>
<td>DocDB 11233 [42]</td>
</tr>
<tr>
<td>What are the DUNE FD DAQ Bottlenecks?</td>
<td>DocDB 11461 [?]</td>
</tr>
</tbody>
</table>

7.4.1 Overview

The DAQ system is composed of six distinct subsystems: (1) front-end readout, (2) data selection, (3) back-end DAQ, (4) inter-process communication (IPC), (5) control, configuration, and monitoring (CCM), and (6) timing and synchronization. Each of these subsystems is described in further detail in the following subsections. The physical extent of the DAQ subsystems, with the exception of the IPC and CCM, can be specified in reference to Fig. 7.1: the front-end readout and timing distribution live underground in the CUC; data selection occupies both underground and above-ground spaces; back-end DAQ is above-ground and includes event building and buffering.
before distribution of data to offline; and IPC and CCM extends throughout the entire physical layout of the system, supported on a private network throughout the DAQ system..

The overall system functionality is illustrated conceptually in Figure 7.3, while Figure 7.4 specifies the implementation. Front-end readout is carried out by custom data receiver and co-processing FPGA/CPU hardware, all of which is hosted in O(100) servers in the CUC. A similar number of additional servers is responsible for the execution of additional software-based low-level processing of trigger primitives generated in the front-end readout for the purposes of data selection; the collective low-level information (trigger primitives and trigger candidates constructed from trigger primitives) is propagated to a central server responsible for further processing and module-level triggering; the module level trigger also interfaces to a second server which is responsible for receiving and propagating cross-module and external trigger and timing information. The module level trigger considers trigger candidates and external trigger inputs in issuing a trigger command to the back-end DAQ subsystem. The back-end DAQ subsystem facilitates event building in O(10) servers and buffering for built events on non-volatile storage; upon receiving a trigger command, the back-end DAQ queries data from the front-end readout buffers and builds that into an “event record”, which is temporarily stored as (a number of) files. Event records can be optionally processed in a high-level filter/data reduction stage, which is part of overall data selection, for further down-selection, prior to becoming custody of the DUNE offline system. Pervasively, the DAQ control, configuration and monitoring sub-system (CCM) subsystem provides the central orchestration (Section 7.4.6), the inter-process communication (IPC) subsystem provides overall communication (Section 7.4.5), and the DAQ timing and synchronization sub-system (TSS) provides synchronization (Section 7.4.7).

![Figure 7.3: DAQ Conceptual Overview of DAQ System Functionality for a single 10 kton module](image-url)
Chapter 7: Data Acquisition

7.4.2 Front-end Readout

Front-end readout provides the first link in the data flow chain of the DAQ system and is where raw data from detector electronics is received by the DAQ. It implements a receiver, buffer, and a portion of low-level data selection (trigger primitive generation) as detailed in Figure 7.5. It is physically connected to the detector electronics via optical fiber(s) and buffers and serves data to other DAQ subsystems, namely the data selection and the event builder.

It would be nice to redraw this using DUNE colors.

Would be useful to add a figure showing functional blocks and implementation. Georgia

The readout system comprises many similar DAQ readout units (RU), each connected to a subset of electronics from a detector module and interfacing with the DAQ switched network. In the case of the TPC, 75 RU are each responsible for the readout of raw data from two APAs. In the case of the PDS, 6-8 RU are each responsible for the readout of raw data from a collection of PDS subdetectors, where each collection corresponds to an optically isolated region in the detector. Each RU encompasses a commercial off-the-shelf server that hosts a collection of custom hardware, firmware, and software that collectively form four functional blocks.
1. Data reception, facilitated by a FELIX card in the host server

2. Network based I/O, facilitated by a commercial off-the-shelf network card

3. Data processing, facilitated by FPGA resources on the FELIX card and/or on-host CPU resources, or, in the case of the TPC RU only, additional FPGA resources in the form of two dedicated co-processing boards (interfacing directly with the FELIX card)

4. Temporary data storage, facilitated by host RAM and SSD, or, in the case of the TPC RU only, RAM and SSD available on the co-processing boards

Each of these blocks is described below. In addition, and like all other DAQ subsystems, the readout participates in the common software framework for control, configuration, and monitoring as described in Section 7.4.6.

7.4.2.1 Data reception

The physical interface between the detector electronics and the DAQ to transmit data consists of 10 Gbps point-to-point serial optical links, running a simple (e.g., 8/10 bit encoded) protocol. The number of links per DUNE module varies from approximately 1000 to 2000, depending on the detector technology adopted.

To minimize the space and power consumption footprint of the DAQ, 10-20 links are aggregated into FELIX boards hosted in commercial, off-the-shelf computers. FELIX is a field programmable gate array (FPGA)-based PCIe board developed initially for ATLAS and now proposed or already used in several experiments, including ProtoDUNE. Existing firmware has been adopted and is being adapted to ensure decoding and format checking of incoming data and then to marshal the data to other blocks of the readout subsystem.
7.4.2.2 Network based I/O

The readout subsystem provides access to the data selection and DAQ back-end sub-system (BE) through a commercial, off-the-shelf switched network as illustrated in Figure 7.5. The network communication protocol is as described in Section 7.4.5. The network I/O is handled by the RUs via software; dedicated hardware or firmware development is not required.

7.4.2.3 Data processing

The data processing functional block resides either on the FELIX board FPGA, or on an additional, dedicated co-processor providing additional FPGA processing resources, or both. Data processing can also be carried out in the host server processor. This functional block is ultimately responsible for identifying regions of interest in the detector (in the TPC or PDS) as a function of time.

As a preliminary step, data is pre-processed, i.e., organized in a way that better suits subsequent data analysis. This implies, e.g., reorganizing data into different streams (collection plane vs. induction plane(s), or re-arranging time and channel order and aggregating samples into frames), applying noise filtering algorithms, and compressing or zero-suppressing data.

The readout system summarizes the identified regions of interest on a per-channel basis into information packets called trigger primitives. These are forwarded to the data selection system which makes correlations and ultimately decides whether and which data is to be saved.

This functional block may be implemented onto FPGAs, GPUs, CPUs, or a combination of these elements. Deciding on the implementation is premature at this stage; this is one of the main topics to explore and develop in the readout area. The DAQ design can facilitate either FPGA or CPU implementation interchangeably, which provides flexibility and adaptability to whatever the processing needs may ultimately be, depending, for example, on noise levels in the detector.

7.4.2.4 Buffering

In DUNE, the readout system is in charge of buffering all detector data until the DAQ data selection sub-system (DS) has formed a trigger decision (Section 7.4.3) and until subsequently the BE (Section 7.4.4) has requested and received that selected data. In addition, in the case of a SNB trigger, data received after the issuing of the trigger must be buffered for much longer to absorb the strongly punctuated bottlenecks that are expected to form downstream in that case.

The buffering time required to select data containing localized activity is dominated by processing speed, pipeline depths and network latency. Some studies must still be performed but initial estimates indicate that the time buffering time required is should not exceed a maximum of approximately one second. The duration of data that must be copied from buffering in order to capture the localized activity corresponds to 5.4ms. As the full stream of data must constantly be buffered, a RAM technology is selected based on providing sufficient throughput, endurance, and
The extended activity (e.g., due to a potential SNB) presents a far more challenging set of buffering requirements. Low-energy activity that is associated with a SNB trigger decision may exist for as much as 10 s prior to the issuing of SNB trigger (trigger time). A second challenge in recording data containing extended activity is that all channels must be recorded for 100 seconds around its trigger time, and requires extracting as much as 115 TB from the TPC readout. It is not cost effective to design the DAQ to accept this rate pulse in real time. Thus, additional buffering is provided to catch the temporary backlog of data that the SNB trigger will produce.

The technology and scale of this additional buffering must satisfy several requirements. It must accept the full data rate of the detector module (as much as 2 TB/s). The data must then reside on nonvolatile media. The media must have sufficient capacity and allow for sufficient extraction throughput so that it is vanishing unlikely to be allowed to become too occupied to accept another pulse of data. Assuming that, on average, a SNB trigger condition will be satisfied once per month, the most optimal technology is solid-state devices, which at the scale required to provide suitable input bandwidth, can provide a capacity to write the data from several extended activity triggers. Providing only a modest overhead to normal operations, this data can be extracted from such storage from the DAQ in under a day.

For both types of activity, the buffering requirements may be reduced by employing lossless compression to the data prior to it entering the buffer. A factor of at least two in reduction in buffer input rate is expected, based on MicroBooNE and protoDUNE experience. If expected noise levels are achieved, this compression would provide a factor of four to ten, depending on the detector module technology. Effort is currently underway to understand the costs and technology involved in exploiting this benefit.

### 7.4.3 Data Selection

The data selection subsystem is a hierarchical, online, primarily software-based system. It is responsible for immediate and continuous processing of a substantial fraction of the entire input data stream. This includes data from TPC and PDS subdetectors. From that input, as well as external inputs provided, for example, by the accelerator or detector calibration systems, the DS must form a trigger decision, which in turn produces a trigger command. This command summarizes the observed activity that led to the decision and provides addresses (in space and time) of the data in the FE buffers that capture information about the activity. This command is sent to and then consumed and executed by the BE as described in Section 7.4.4. It is also propagated to the external trigger logic (ETL) and from there it may be distribute it to other detector modules or other detector systems (e.g. calibration) for consideration.

The pipelines of processing required for data selection may execute in various stages and forms using different firmware and software implementations. Development is actively ongoing to demonstrate viability and performance of different implementations. In satisfying the philosophy and strategies of the DAQ design there is flexibility in defining whether each element of a pipeline executes on FPGA, CPU, GPU, or in principle, some other future hardware architecture.
The DS must select data associated with calibration signals, as well as beam interactions, atmospheric neutrinos, rare baryon-number-violating events, and cosmic ray events that deposit localized visible energy in excess of 100 MeV, with high efficiency (>99%). It must also select data associated with potential galactic SNBs, with galactic coverage\(^1\) of >90% efficiency. To meet the requirement that the DUNE FD maintain a <30 PB/year to permanent storage, the DS subsystem must make data selection decisions in a way that allows the DAQ system to reduce its input data by almost four orders of magnitude without jeopardizing the above efficiencies.

The DS subsystem design follows a hierarchical structure, where low-level decisions are fed forward into higher-level ones until a module-level trigger is activated. The hierarchy is illustrated in Figure 7.6. At the lowest level, trigger primitives are formed on a per-channel basis, and represent, for the baseline design, a “hit on a wire/channel” activity summary. Trigger primitives are aggregated into Trigger Candidates, which represent “clusters of hits”. Trigger Candidates are subsequently used to inform a module-level trigger decision, which generates a Trigger Command; this takes the form of either a localized high energy trigger or an extended SNB trigger, and each prompts readout of an event record. Post-event-building, further data selection is carried out in the form of down-selection of event records, through a high level filter.

The subsystem structure is illustrated in Figure 7.7. The structure has three levels of data selection: (1) low level trigger, which may consist of trigger primitive generation and subsequent trigger candidate generation; (2) module level trigger; and (3) high-level trigger (HLT). Each trigger level is described in subsequent sections. An additional subsystem component is the external trigger module, which serves as a common interface for the module level trigger of each of the FD detector modules and between the module level trigger and other systems (e.g., calibration, accelerator and timing system) within a single detector module. After sufficient confirmation of quality the ETL also sends SNB triggers to global coincidence trigger recipients such as SuperNova Early Warning System (SNEWS).

Figure 7.6: Data Selection strategy and hierarchy: TPC based.

\(^1\)Galactic coverage is defined as efficiency-weighted probability of galactic SNB.
Chapter 7: Data Acquisition

Figure 7.7: Block diagram of DUNE DAQ data selection subsystem, illustrating hierarchical structure of subsystem design, and subsystem functionality.

- Localized activity with a 100 MeV deposited energy threshold generates localized high energy trigger candidates for the module level trigger; any such candidate can be accepted as a localized high energy trigger.

- Localized activity with a 10 MeV deposited energy threshold generates localized low energy trigger candidates for the module level trigger; those are used as input to an extended low energy trigger for supernova neutrino bursts. Each trigger type prompts readout of the entire detector but over significantly different time ranges: localized triggers prompt readout of 5.4 ms event records; extended triggers prompt readout 100 s event records.

To facilitate partitioning, the data selection subsystem will be able to be instantiated multiple times, and multiple instances will be able to operate in parallel. Within any given partition, the data selection subsystem will also be informed and aware of current detector configuration and conditions and apply certain masks and mapping on subdetectors or their fragments in its decision making. This information is delivered to the DS from the DAQ control, configuration and monitoring sub-system (CCM) system.

Ultimately, each trigger decision culminates in a command sent to BE. This command contains all the logical detector addresses and time ranges required such that an event builder (EB) may properly query the front-end buffers and finally collect and output the corresponding detector data and the corresponding trigger data. To avoid duplication of data records associated with trigger commands that overlap in readout record “space”, the data selection system must also time-order and prioritize trigger decisions. The details for forming this command are described next and the operation of the BE is described in Section 7.4.4.
7.4.3.1 Low Level Trigger: Trigger Primitive Generation

A trigger primitive is defined nominally on a per-channel basis. In the case of the SP TPC, it is identified as a collection-channel signal rising above some noise-driven threshold for some minimum period of time (here called a “hit”). A trigger primitive takes the form of an information packet that summarizes the above-threshold waveform information in terms of its threshold crossing times and statistical measures of its ADC samples. In addition, these packets carry a flag indicating the occurrence of any failures or other exceptional behavior during trigger primitive processing.

Trigger primitives derived from TPC data are produced in the front-end readout part of the DAQ system, nominally in FPGA firmware or potentially in CPU or GPU software as described in Section 7.4.2. Any trigger primitives derived from PDS data are produced by the PDS system for consumption by the DAQ data selection subsystem.

Algorithms for generating trigger primitives are still under development [?]. One example algorithm[?] establishes a waveform baseline for a given channel, subtracts this baseline from each sample, maintains a measure of the noise, and searches for the waveform to cross a threshold defined in terms of the noise level. This algorithm has been validated using both Monte Carlo simulations and real data from ProtoDUNE. Its performance is summarized in Section 7.5.3.

The format and schema of trigger primitives require study and optimization and this may be tightly coupled with the formation of trigger candidates discussed next. Initial estimates show that 20 bytes provides a generous data representation of trigger primitive information. The trigger primitive rate will be dominated by the rate of decay of naturally occurring $^{39}$Ar, which is about 10 MHz per module. This leads to a detector module aggregate rate of 200 MB/s. The subsequent stage of the data selection must continuously absorb and process this rate providing trigger candidates as described next.

7.4.3.2 Low Level Trigger: Trigger Candidate Generation

Trigger primitives from individual, contiguous fragments of the detector module are consumed in order that cross-channel and -time clustering may be performed and possibly result in the output of trigger candidates. Once activity is localized in time and channel (“space”) it is possible to apply a rough energy-based threshold based on combining the statistical metrics carried by the input trigger primitives.

A trigger candidate packet carries information about all the trigger primitives that were used in its formation. In particular, it provides a measure of the total activity represented by these primitives. This measure will be used downstream to allow the final trigger decision, as described more in the next section.

Prior to output of, a candidate is subject to a selection criteria. While the selection applied in
the previous stage was driven by a measure of noise, here it is driven by background activity. In
particular, candidates which are consistent with activity from the very high rate, low energy $^{39}$Ar
decays will be strongly prescaled. The higher energy, lower rate but still numerous candidates
consistent with activity from the $^{42}$Ar decay chain will also be prescaled to an extent. Additional
studies are expected but nominally individual candidates, or groups of candidates nearby in detec-
tor space in time, with measures of energy greater than these two types of decays will be passed
with little or no prescaling.

### 7.4.3.3 Module Level Trigger

Trigger information is further aggregated as all candidates are consumed by the module level
trigger in order to form the final trigger decision. The channel and time extent as well as the
energy measure of the candidates are used at this stage to categorize the activity. The category
drives the algorithm applied to form a decision. For example, isolated, low energy candidates
arriving in coincidence over the period of a second across the full detector module may be used
toward satisfying a condition that indicates a potential SNB. Individual high energy candidates,
or lower energy candidates distributed over a localized region of detector space and time would be
applied toward satisfying a high-energy condition.

When a particular condition in a category is satisfied, the trigger decision is made and a trigger
command is formed. This packet includes information of the candidates (and primitives) that were
used to form it. The decision also provides direction as to what set of detector components are to
be read out and over what time period. As described at the start of this section, localized activity
will lead to the readout of the entire detector module for a period equal to 5.4 ms. Extended
activity triggers (SNB) will direct the readout of much longer period of 100 s.

The module level trigger will send its produced trigger commands to the DAQ back-end sub-
system (BE) for the detector module. The BE will dispatch the command to a EB for execution
as described in Section 7.4.4.

Trigger commands are also sent to the external trigger logic unit which forwards them to other
detector modules. Likewise, the module level trigger receives trigger commands from other mod-
ules, and considers this information in making its own trigger decision. This is particularly needed
in order to allow for cross-module coincidences to be formed and thus produce an overall lower
threshold for capturing potential SNB occurrences. The external trigger logic unit will also forward
SNB trigger commands, after suitable quality confirmation, to external recipients such as SNEWS.

In addition to accepting cross-module triggers via the external trigger logic unit, the module
level trigger also takes inputs from out-of-band sources such as needed for beam, calibration or
random triggering. If meaningful sources of triggering information that is external to DUNE can
be provided promptly enough so that the corresponding data still resides in the front-end buffers,
they may also be considered.
7.4.3.4 High Level Filter

The last processing stage in the data selection subsystem is the high-level trigger (HLT), which resides in the back-end part of the DAQ and is further referenced in Section 7.4.3.4. The HLT acts on triggered, read out, and aggregated data, produced by an EB. It therefore serves primarily to down-select and thus limit the total triggered data rate to offline, thereby keeping efficiency high in collecting information on activities of interest while minimizing selection and content bias, and reducing the output data rate. It may do so via further filtering, lossy data reduction, and/or further event classification. As it benefits from a longer latency (time between ~Hz-level built events), it can accommodate higher level of sophistication in algorithms for data selection decisions.

More specifically, the HLT may further reduce the rate of data saved to final output storage by applying refined selection criteria which may otherwise be prohibitive to apply to the pre-trigger data stream. For example, instrumentally-generated signals (e.g. correlated noise) may produce trigger candidates that can not be rejected by the module level trigger and if left unmitigated may lead to an undesired high output data rate. Post processing the triggered data may allow reducing this unwanted contamination. Furthermore, it can also reduce the triggered data set by further identifying and localizing interesting activity. A likely candidate hardware implementation of this level of data selection is a GPU-based system residing on surface at SURF.

To fully understand how much and what type of data reduction may be beneficial, simulation studies are ongoing DocDB xx [?] and will necessarily have to be validated with initial data analysis after first DUNE FD operation. Planned development efforts will also be carried out to determine the scale of processing required by the FD.

7.4.4 Back-end System

The DAQ back-end sub-system (BE) encompasses the output concept and interfaces to the buffer and trigger concepts shown in Figure 7.3. It accepts trigger commands produced by the DAQ data selection sub-system (DS) as described in Section 7.4.3. It queries the front-end buffer interfaces and accepts returned data as described in Section 7.4.2. Finally, it records trigger commands and the corresponding data to the output storage buffer, from which the data is transferred to the custody of DUNE offline.

7.4.4.1 Dataflow Orchestration

To minimize data extraction latency, the BE must not serially execute trigger commands to completion. This asynchronous execution governed by the dataflow orchestration (DFO) and operates as illustrated in Figure 7.8 and as discussed here:
• DFO accepts a time ordered stream of trigger commands and dispatches each for execution possibly by first splitting up each command into one or more contiguous segments.

• Each segment will then be dispatched to an EB process as described in Section 7.4.4.2 for execution.

• Execution entails interpreting the trigger command segment and querying the appropriate FE buffer interfaces to request data from the period of time.

• Requests and their replies may be sent synchronously, and replies are expected even if data has already been purged from the FE buffer.

• The data received may then undergo processing and aggregation until finally it is saved to one or more files on the output storage system before custody is transferred offline.

### 7.4.4.2 Event builder

The DAQ back-end subsystem will provide the instances of the event builder (EB). As described above, each will request selected data from the appropriate front-end buffer interface (FBI) as addressed by a consumed trigger command segment. An EB will aggregate the selected data and potentially apply processing and reduction (Section 7.4.3.4) as well as monitor its quality while in flight. Finally it will record the resulting data to the output storage system. The final output files shall use data schema and file formats as described in Section 7.4.4.4.
7.4.4.3 Data Quality Monitoring

Section 7.4.6.3 described a monitoring system for the CCM subsystems. Monitoring the quality of the information held in the detector data itself is critical to promptly responding to unexpected conditions and maximizing the quality of acquired data. A DAQ data quality monitoring (DQM) system will be developed to provide functionality for the infrastructure, visualization and algorithms required to continuously process a subset of the detector data so that prompt summaries may be provided for human consumption. This system will be designed to allow it to evolve as the detector and its data is understood during commissioning and early operation and to cope with any evolution of detector conditions. Many software modules will be developed offline, so the data quality monitoring (DQM) subsystem will facilitate their reuse when applied to samples of detector data.

7.4.4.4 Data Model

module-generic

Describe the data model. This isn’t a strict schema just things like how various parts of the detector readout map to files, etc.

7.4.4.5 Output Buffer

Describe the output buffer system, how it’s shared with offline, data hand-off protocols. Responsibility scope (eg, who handles transfer to FNAL).

The output buffer system is a hardware resource provided by DAQ and used offline by DUNE. It has two primary purposes. First, it decouples producing content in operating DAQ from transferring that content from the far site to archive storage units and offline processing. Second, it provides local storage sufficient for uninterrupted DAQ operation in the unlikely event that the connection between the FD and the Internet is lost. Based on very unusual losses of connectivity at major laboratories as well as FD sites of other long-baseline neutrino experiments, the output buffer must provide enough storage capacity to retain one week of output given nominal data production. The maximum data production rate for the FD is set at 30 PB/year. Thus, the output storage buffer must have a capacity of approximately 0.5 PB to service the entire FD.

How to reference the 30PB/yr requirement?
7.4.5 Inter-process Communication

The DUNE FD DAQ is an asynchronous, distributed data processing system. As such, it consists of loosely connected elements. The elements or “nodes” are connected through forms of inter-process communication (IPC). Connections include:

- CCM elements sending control and configuration messages to all DAQ nodes and receiving from them messages containing monitoring information. (Section 7.4.6)
- Passing trigger information through the Data Selection system (Section 7.4.3)
- Query based readout of upstream DAQ buffers. (Section 7.4.2)
- Delivery of trigger commands to the back-end and distributed back-end processing prior to writing final output. (Section 7.4.4)

The solutions for each form of IPC are currently being evaluated. Certain general requirements apply. For the most part, they must support transport mechanisms that are reliable and robust. Reliable that messages sent must be received fully unless critical failures of endpoints or the transport link occurs. Robust means that some classes of failures (temporary network congestion or temporary endpoint failure) that might otherwise interrupt communications may be overcome. The acceptable duration of failure is chosen based on the particular protocol in use. Related to reliability, the inter-process communication (IPC) system used by CCM will provide discovery and presence functionality as described in Section 7.4.6.

The ZeroMQ [88] smart network socket library is being evaluated for providing a basis for IPC covering CCM, Data Selection and readout queries. Section 7.5 contains description of some initial validation of this approach by adding a self-triggering feature to ProtoDUNE-SP which is close to what will be required for the DUNE FD DAQ. Also described initial prototyping efforts based on ZeroMQ of core elements required for the IPC to support CCM and readout queries.

The artDAQ system [89] utilizes IPC for connecting its various elements. It is full featured and well tested in production settings including in ProtoDUNE-SP. It is a natural choice for providing a basis for the DAQ back-end. It will also be considered for providing a basis for the Data Selection system. Upstream DAQ readout queries may also be implemented in terms of artDAQ elements.

Additional R&D and validation are needed to determine an optimal solution for each IPC domain. This optimization process will take into account requirements unique to DUNE FD (minimal downtime, data rates, variation in detector modules, long term use, available expertise, flexibility for future developmental improvements, etc).
### 7.4.6 Control, Configuration, and Monitoring

The DAQ control, configuration and monitoring sub-system (CCM) subsystem, illustrated in Figure 7.9, encompasses, as its name suggests, the software needed to control, configure, and monitor the rest of the DAQ, as well as itself. It provides a center for the highly distributed DAQ components, allowing them to be treated and managed as a single, coherent system. Figure 7.3 shows the central role of the CCM within the complete DAQ system. The following sections describe each of the three CCM subsystems.

**Figure 7.9: Main interaction among the three CCM subsystems.**

#### 7.4.6.1 Control

The DAQ control subsystem actively manages DAQ software process lifetimes, asserts access control policies, executes commands, initiates configuration changes, detects and handles exceptions, and provides an interface for human operators.

The control subsystem comprises a number of functional blocks of either global or partition scope as illustrated in Figure 7.10. In the figure, “partition” refers to a logical segmentation of the DAQ where each segment operates to some extent independently from others. The segmentation applies to the portion of detector electronics that provides input to the partition’s data selection and readout. The largest partition is that of one detector module. The functional blocks in the figure represent one or more semi-autonomous agents, each with defined roles, capabilities, and access. Although drawn as single blocks, they typically are implemented as multiple peer agents to assure redundancy and fail-over. The blocks at partition scope are first described.

- Partition naming service provides discovery and presence for the components of a DAQ partition. That is, it allows a component to be made aware of the creation and continued operation of other components (discovery) or that other components have recently become unresponsive (presence).

- Run control provides a central director; its creation is the first step in initiating a DAQ partition. The run control (RC) accepts, interprets, and validates input commands that may come either from a human via a user interface, or from other blocks. The commands describe...
Global scope controls DAQ components for all DAQ partitions across all detector modules. It consists of the following blocks:

- Global naming service aggregates the discovery and presence information across partitions. Like the partition naming service, this may be implemented as a centralized (but redundant) explicit service or may be provided by extending the IPC discovery and presence mechanism to the entire FD DAQ network.

- Process management allocates and may reclaim sets of processes on behalf of a requesting

---

Figure 7.10: Roles and services that compose the DAQ control subsystem.

---

1. a desired state of the DAQ partition. RC may query other blocks to validate commands and then execute the commands by allocating processes through process management. Once successfully allocated, their lifetimes are managed by RC. Throughout its lifetime, RC may reconfigure an existing process, destroy it, or allocate additional processes. RC may query the partition-naming service in order to resolve resource identifiers in commands into their corresponding network endpoint addresses.

2. Supervisor provides a locus of expert system automation. This block is initiated along with its RC peer and augments human commands with automated ones. For example, as certain “common exceptions” are encountered and understood so that a means to correct them can be developed, the supervisor can be extended to automatically issue the corrective actions that must otherwise be manually performed by a human operator.

3. DAQ instances at locations other than the FD cavern are expected to operate in a wholly distinct manner.
component (specifically RC and resource management). An allocation request includes a complete description of the processes and their desired initial configuration. A successful allocation occurs only after this role successfully initiates all requested processes and confirms their presence. Process management only allocates processes if their requester has appropriate access privileges as determined by access management and if resource management determines sufficient resources exist. Process management may support pre-allocation if sufficient access and resources are confirmed and reserved; if so, a token (aka “cookie”) is returned to the requester. This cookie may be subsequently be presented to complete the allocation and claim the processes. After a configured timeout, the cookie may be invalidated.3

- Resource management determines whether any process allocation can proceed and enacts a process garbage collection mechanism. An allocation is successful if sufficient process resources are available and if it is consistent with a set of configured constraints maintained by this component. Resource management monitors all allocated processes as well as the process initiating a request in order to perform “garbage collection”. This is performed in the event that the allocated processes outlive the requesting processes. Resource management notifies process management of any remaining processes from an allocation when it detects that the requester is no longer present. Resource management also supports pre-allocation validation queries. A validation response merely indicates current state and it represents no guarantee that the allocation will subsequently succeed.

- Access management is responsible for providing authentication and authorization for all DAQ functions that require access control. This block may be implemented as an explicit centralized (but redundant) service or through a distributed IPC mechanism or as some mixture of the two designs.

The last block in Figure 7.10 represents applications that provide user interfaces (UI) to the control subsystem. At least one UI will be developed to allow a trained operator to construct and issue commands required to initiate, configure, potentially reconfigure, and finally terminate DAQ partitions. The UI may validate user commands before issuing them to an RC by directly querying the process manager block. If sufficient resources are unavailable or if the user lacks appropriate access privileges, the UI will present a descriptive error message or otherwise disable corresponding functionality. If the user permission is valid the UI sends the use-initiated command to the RC for execution. Note that subsequent commands from the RC to other blocks are also subject to access management. Additional UI elements will be developed as described in sections 7.4.6.2 and 7.4.6.3.

7.4.6.2 Configuration

The DAQ configuration subsystem provides persistent data storage for all historic, current, and future configuration information applicable to the DAQ. It provides a singular point (via high-availability, redundant services) for the allocation of unique and monotonically increasing DAQ run numbers. The configuration data stores operate in an insert-only mode with no update nor deletion of records so as to keep a complete record of configuration actions. The configuration subsystem supports the following types of information:

3This will be required if the race condition between multiple UIs and RCs is a problem.
• Partition structure contains descriptions of the multiplicity and connectivity of DAQ components for any partition. Structure and connectivity is expressed in an abstract manner with logical addressing and not through concrete addressing (e.g., host computer network address and port numbers). This allows for identical structure to be reapplied to various collections of specific hardware.\(^4\)

• Component parameters comprise configuration information associated with any given DAQ component. This data is structured following a schema defined by the associated component and this schema is versioned to allow for schema evolution.

• Run number provides a monotonically increasing sequence of DAQ run numbers that are allocated upon request to assure that each is unique.

• Partition instances associate a DAQ run number and the set of component parameters that were used to initiate a DAQ partition or which are used to reconfigure an existing DAQ partition.

• Constraints define rules that must be held true by resource management servicing requests for process allocations. This information store also includes which constraints were used by resource management over time.

Access to configuration information is via a service that hides the choice of storage technology from any client queries. This interface will also be used by configuration editors utilized by human operators as well as any generators employed by expert systems.

### 7.4.6.3 Monitoring

The DAQ monitoring subsystem will help both humans and expert systems in detecting, diagnosing, and correcting anomalous activity, observing intended operation, and providing a historical record. This subsystem will accept required information produced by any DAQ component (here called status).

The precise implementation of the production, acceptance, store, post-processing, querying, and visualization of monitored status requires additional work. However, a publish-subscribe (PUB/SUB) network communication pattern is expected to be adopted for transport of monitoring messages. This will decouple production and consumption and facilitate development of a variety of status viewers, expert systems, debugging tools, etc. The types of messages include but are not limited to the following:

- Common to all will be a “header” holding a message type indicator, a sender address and the associated detector data time and the recent host computer time.

- Logging messages add an importance label (e.g., debug, info, warning, error) and a succinct,

\(^4\)It is the discovery and presence from naming services as described in Section 7.4.6.1 that allow mapping from abstract to concrete addressing.
human-readable information string providing an explanation of what occurred.

- Metrics will provide structured data carrying specific information about predefined aspects of the sender. This is similar to logging, but the messages support automated consumption and reaction by expert systems.

- Quality messages summarize information derived from the detector data (e.g., from waveforms) or its metadata (e.g., timestamps, error codes) while that data is “in flight” through the DAQ.

In general, the DAQ will retain all status records at least long enough to allow for any offline data quality validation procedures to be performed. However, some status feeds may be processed prior to storage if their raw form requires prohibitive amount of storage. In particular, the quality stream data rate may be too substantial for long term storage. Such streams will be summarized into histograms or other statistical representations prior to storing for longer term use.

In addition to this DAQ CCM monitoring subsystem, a separate system must be used to monitor in depth the quality of the detector data content itself. See Section 7.4.4.3 for the description of this data quality monitoring system.

### 7.4.6.4 Partition Lifetime

The partition lifetime is described here in a somewhat linear narrative, it should be noted that the components will be constructed through some suitable protocol that need not progress in the same linear order. In particular, the operation of the partition components shall be robust to the order in which peers are discovered.

After a process is executed via the allocation mechanism described in Section 7.4.6.1, it will apply its initial configuration. This information includes any personal identifiers the component will assert as part of discovery and presence as well as any identifiers required to find any other peer components which are needed for its own operation. In particular, a component is provided the identity of the partition’s RC instance so that it may receive control directives.

It is through a control directive enacted by each individual component that the overall partition structure and connectivity emerges. These directives must be issued prior to some activation criteria in order to enable the zero-downtime reconfiguration feature as described next. The control directive contains a CCC. A CCC provides, at a minimum, the following pieces of information:

- Run number is as described in Section 7.4.6.2. Here, it identifies a desired and collective partition state which will be constructed once all CCCs are enacted.

- Activation time stamp (ACT) states the data time (see Section 7.4.5) at which the CCC shall take effect.

- Configuration payload provides the component-specific configuration to be enacted and may
include actions must be performed prior to the ACT in order to assure a zero-downtime transition.

When a CCC is received by a component, that component initiates any new connections with peers and performs any other pre-ACT actions as directed by the CCC. The component then begins (or continues) to monitor the data time of received messages on its new (or previous) input sockets. If the component was operating as part of the prior partition it will continue to service its previous input and produce output to any previously connecting consumers. Data is not yet sent to any new connections. Upon receiving input with a data time after the ACT it will apply the new configuration specified in the CCC. In this reconfiguration process, any pre-ACT data that may still be buffered by the component shall be flushed to its (previously connected) output sockets. Any input or output connections no longer applicable to the new, post-ACT partition definition shall be dropped. Finally, the component shall renew operations, beginning with the held data which had satisfied the ACT criteria and which initiated the reconfiguration processes.

For this mechanism to truly provide zero-downtime, the partition components must receive reconfiguration messages from the CCC sufficiently in advance of input data passing the ACT. This means the human-UI-RC chain must select an ACT knowing the most recent data time as well as some lead time to apply. For any given reconfiguration an optimal lead time involves many independent factors but may be estimated by considering a maximum calculated over all components involved in the reconfiguration of the time difference between their required reconfiguration time and the latency for data to arrive at the component from the time of sampling. In a real system this maximum has some distribution. In practice it is expected that the lead time must be chosen in some manner or simply “long enough”. Even a generous choice is likely to satisfy human human impatience especially given the alternative is accepting data loss.

Although the lead time may need to be many seconds, it is important to note that the minimum time between subsequent ACTs is essentially zero. Multiple sets of CCCs may be issued over some short time span and queued by components. In principle, this allows zero-downtime sequencing of runs of arbitrarily short duration. Practically however, this may be limited due to performance issues. For example, if a new set of CCCs requires many duplicate readers of data streams this may cause bandwidth limits to be reached. To the extent this fast run sequencing is needed, these potential limitations require additional study.

Finally, after cycling through one or more run numbers, the partition may be terminated. A final round of CCCs is issued by the partition’s RC. Each CCC directs the termination procedure of its component. This procedure starts just as any zero-downtime reconfiguration. The CCC instructs the component to continue processing until receiving input data which satisfies the ACT criteria at which time any remaining buffered data is flushed to the component output connections. Then, unlike zero-downtime reconfiguration, the component simply destroys all connections and exits. The RC notifies the UI and process manager of the destruction of the partition (saving, for the moment, the destruction of the RC itself). The process manager notifies the resource manager that the resources have been released. The RC then terminates itself, and the partition is no more. The resource manager confirms partition processes have terminated through discovery and presence. In the odd case that the RC aborts without cleanly terminating the partition, its absence must be detected, and the remaining partition processes are reaped in the garbage collection mechanism described in Section 7.4.6.1.
7.4.6.5 Self-healing

The above zero-downtime reconfiguration mechanism is intentional and typically driven by human action or automated run sequencing algorithms. Similarly, the partition will be self-healing in the face of unexpected failures that render peers unresponsive or when unexpected information content is received by partition components. Extending the metaphor, self-healing involves these phases: detection of an injury to the partition, diagnosis of the scope of the injury, and intervention that executes an action on the partition.

Detection is performed by the DAQ control subsystem supervisor functional block using at least one of two methods. First, if a component in the partition becomes unresponsive (i.e., it crashes or hangs) the supervisor receives notification through DAQ discovery and presence. Second, if a component directly detects injury, such as may be the case when it receives data outside of expected norms, it reports this fact through IPC to the supervisor.

When injury is detected the supervisor diagnoses this using heuristics and other methods that are expected to evolve as failure modes are discovered or removed. It is thus crucial that the supervisor is designed and developed in a way that facilitates this ongoing evolution.

Finally, the supervisor responds to the diagnosis with some intervention. Response in all cases includes notifying the monitoring subsystem. Any additional response involves sending commands to the RC to initiate a reconfiguration, which may be to terminate the partition. When initiating a reconfiguration, as with any reconfiguration, the command to the RC must include information required by the RC to issue CCC messages. If the partition remains it is reconfigured and thus begins a new run number as described in Section 7.4.6.4.

7.4.7 Timing Distribution and Synchronization

All components of the FD use clocks derived from a single Global Positioning System (GPS) disciplined source, and all module components are synchronized to a 62.5 MHz clock. To make full use of the information from the PDS, the common clock must be aligned within a single detector module with an accuracy of $O(1 \text{ ns})$. For a common trigger for a SNB between modules, the timing must have an accuracy of $O(1 \text{ ms})$. However, a still tighter constraint is the need to calibrate the common clock to universal time derived from Global Positioning System (GPS) so the data selection algorithm can be adjusted inside an accelerator spill, which again requires an absolute accuracy of $O(1 \mu \text{s})$.

The DUNE FD uses a version of the ProtoDUNE timing system, where a design principle is to transmit synchronization messages over a serial data stream with the clock embedded in the data. The format is described in DocDB 1651 [41]. The timing system design is described in detail in DocDB 11233 [42].

Central to the timing system are four types of signals:
• a 10 MHz reference used to discipline a stable master clock,
• a one-pulse-per-second signal (1PPS signal) from the GPS,
• a Network Time Protocol (NTP) signal providing an absolute time for each one-pulse-per-second signal (1PPS signal), and
• an inter-range instrumentation group (IRIG) time code signal used to set the timing system 64-bit time stamp.

The timing system synchronization codes are distributed to the DAQ readout components in the central utility cavern (CUC) and the readout components on the cryostat via single mode fibers and passive splitters/combiners. All custom electronic components of the timing system are contained in two Micro Telecommunications Computing Architecture (µTCA) shelves; at any time, one is active while the other serves as a hot spare. The 10 MHz reference clock and the 1PPS signal signal are received through a single-width advanced mezzanine card (advanced mezzanine card (AMC)) at the center of the µTCA shelf. This master timing AMC is a custom board and produces the timing system signals, encoding them onto a serial data stream. This serial data stream is distributed over a backplane to a number of fanout AMCs. The fanout AMC is an off-the-self board with two custom FPGA mezzanine cards (FMCs). Each FMC has four SFP cages where fibers connect the timing system to each detector component (e.g., APA) or where direct attach cables connect to other systems in the CUC.

To provide redundancy, two independent GPS systems are used, one with an antenna at the surface of the Ross shaft, and the other with an antenna at the surface of the Yates shaft. Signals from either GPS are fed through optical single mode fibers to the CUC, where either GPS signal can act as a hot spare while the other is active. Differential delays between these two paths are resolved by a second pair of fibers, one running back from the timing system to each antenna.

7.5 Design Validation and Development Plans

The following strategy will be followed in order to validate and develop the DUNE FD DAQ design:

• Use of ProtoDUNE as a design demonstration and development platform.
• Use of vertical slice teststands for further development and testing of individual DAQ subsystems and for key aspects of the overall DAQ
• Use of horizontal slice tests to demonstrate scaling the design where the multiplicity of components in subsystem layers is important.
• Use of FD MonteCarlo simulations and emulations in order to augment actual hardware demonstrations at ProtoDUNE and teststands.

Single-Phase Far Detector Module

The DUNE Technical Design Report
• Benefit from developments and measurements from other ongoing LArTPC experiments, including MicroBooNE, SBND, and ICARUS.

This strategy reflects the current DAQ project schedule, which comprises several phases, including an intense development phase running through 2020 that culminates in an engineering design review (EDR) in Q1 of 2021. At this milestone, the system design will be finalized and shown to be capable of meeting the requirements of the final DAQ system. After the development phase, a pre-production phase will begin and will end with a production readiness review (PRR). By then, final designs of all components will be complete.

The following subsections summarize past, ongoing, and planned development and validation studies and identify how anticipated outcomes will be used to finalize the DAQ design.

7.5.1 Design Validation and Development at ProtoDUNE and Other LArTPC Experiments

Here we describe protodune, write what similarities and differences there are between ProtoDUNE and DUNE DAQ designs. Mention other related efforts at other experiments.

The FD DAQ consortium constructed and operated the DAQ system for ProtoDUNE, which included two DAQ readout architectures, one based on FELIX, developed by ATLAS [?], and the other on Reconfigurable Computing Element (RCE), developed at SLAC [?]. DAQ design and construction for ProtoDUNE began in Q3 of 2016, and the system became operational at the start of the beam data run in Q4 of 2018. The detector is continuing to run as of the writing of this document, recording cosmic ray activity, and providing further input for DAQ development toward DUNE.

Figure 7.11 depicts the ProtoDUNE DAQ system. The DAQ is split between the FELIX and RCE implementations. The two architectures share the same back-end and timing and trigger systems. Neither of these tested architectures exclusively represents the baseline design for the DUNE FD. Instead, each qualitatively maps into one of two data processing approaches: one in which the data is processed exclusively in custom-designed FPGA, and the other in which the data is processed primarily in commodity CPUs. The baseline system for a detector module instead merges elements of the two approaches. Specifically, it uses FELIX as the hardware platform for data receiving and handling, and an FPGA-based co-processor (analogous to the RCE platform) that interfaces with FELIX to provide additional, dedicated data processing resources. In that sense, ProtoDUNE has provided demonstration of the FELIX platform as FE readout data receiver, and demonstration of FPGA-based data reduction for TPC.

Figure 7.11: The ProtoDUNE DAQ system.
Besides overall readout architecture, the ProtoDUNE and DUNE DAQs exhibit two key differences. First, the ProtoDUNE DAQ is externally triggered (and at a trigger rate over an order of magnitude higher than that anticipated for DUNE). Because of this, the ProtoDUNE DAQ does not facilitate online data processing from the TPC or photon detector (PD) systems for self-triggering. Second, the ProtoDUNE system sits at the surface with a much higher data occupancy due to cosmic ray activity. Overcoming the first key difference to demonstrate data selection capability for the FD DAQ design is a main component of future DAQ development plans, described in Section ??.

Continuous self-triggering of the detector is also new with respect to other ongoing or planned near-term LArTPC experiments, including MicroBooNE, SBND, and ICARUS. Both MicroBooNE and ICARUS have demonstrated self-triggering in coincidence with external gates, which effectively limits both data and trigger rates, and is not a viable solution for DUNE’s off-beam physics program, as it would effectively limit exposure and therefore physics sensitivity. On the other hand, ICARUS has ... MicroBooNE has demonstrated successful continuous readout of a LArTPC, via use of dynamic and fixed-baseline zero-suppression implemented in firmware for both TPC and PDS readout. SBND, which utilizes the same readout system as MicroBooNE, will investigate trigger primitive generation and self-triggering based on TPC information in the timescale of 2021-2023.

Add references.

### 7.5.2 ProtoDUNE Outcomes

Despite its variant design with more limited scope, the successful operation of the ProtoDUNE DAQ has provided several key demonstrations for DUNE DAQ, in particular data flow architecture, run configuration and control, and back-end functionality.

More specifically, ProtoDUNE has demonstrated

- **Front-end Readout**: successful front-end readout hardware and data flow functionality for the readout of two out of the six APAs employed in protoDUNE. This was achieved with two TPC RU’s, without co-processor boards, and only one APA read out per FELIX board. The DUNE DAQ design will ultimately accommodate readout of two APAs per FELIX board. In addition to data flow functionality, ProtoDUNE Front-end readout also demonstrates interface to front-end electronics, and scalability to DUNE. It also supports host server requirements and specifications. Finally, it serves as platform for further development involving co-processor implementation and data selection aspects.

- **Back-end DAQ**: successful back-end DAQ implementation, including event builder farm, CCM machines, and disk buffering. This has allowed the development and exercising of system partitioning, and provides a basis for scalability to DUNE. The protoDUNE back-end also serves as a platform for further system development, in particular in the areas of CCM, IPC and data flow (orchestrator).

- **Data Selection/Timing**: successful operation of the timing distribution system, and external
trigger distribution to the front-end readout. Although protoDUNE was externally triggered, the system serves

Besides demonstrating end-to-end data flow, an important outcome of protodune daq has been the delineation of interfaces, i.e. understanding the exact DAQ scope and the interfaces to TPC, PDS, and offline. The use of commercial off-the-shelf solutions where possible, and leverage of professional support from CERN IT substantially expedited the development and success of the project, as did the strong on-site presence of experts from within the consortium during early installation and commissioning.

Outcomes specific to ProtoDUNE subsystems are discussed in greater detail in

\emph{talk from review}

\section{7.5.3 Ongoing Developments}

Subsystem development is ongoing at ProtoDUNE at the time of the writing of this document. A detailed schedule for 2019 is available in \emph{referencedocdb}. Major development plan milestones are:

\begin{itemize}
  \item optimization and tuning of the Front-end readout
  \item optimization and tuning of the artdaq based dataflow software
  \item enhancement of monitoring and troubleshooting capabilities
  \item introduction of CPU-based hit finding (necessary for PDS readout)
  \item introduction of FPGA-based hit finding (for TPC readout)
  \item implementation of online software data selection beyond trigger primitive stage (introduction of trigger candidate generation, and trigger command generation), and tests on well identified interaction topologies (e.g. long horizontal tracks, or Michel electrons from muon decay)
  \item integration of online trigger command and modified data flow to event builder to facilitate self-triggering of detector
  \item implementation of extended fpga based front-end functionality (e.g. compression)
  \item prototyping of fake SNB data flow in front-end and back-end
\end{itemize}
Below, we focus on ongoing developments related to data selection, which is a key challenge for DUNE and new with respect to ProtoDUNE (and other existing or planned LArTPC detectors), as well as IPC and CCM.

During early stages of design, significant effort has been dedicated on trigger primitive generation through MonteCarlo simulations. Specifically, charge collection efficiency and fake rates due to noise and radiologicals have been studied as a function of hitthreshold with MonteCarlo, demonstrating that requirements can be met, given sufficiently low electronics noise levels and radiological signal.

Ongoing efforts within DUNE’s Radiologicals Task force aim to validate or provide more accurate background predictions, upon which this performance will be validated. In addition, offline emulations of CPU trigger primitive generation on CPU (4 cores) have been carried out, demonstrating the ability of software algorithms in CPU to keep up with expected raw data rates, as shown in Figure ??.

Following the commissioning of ProtoDUNE, full-stream, single-APA, online trigger primitive generation on CPU (10 cores) was successfully demonstrated at ProtoDUNE. Trigger primitive rates were measured at ProtoDUNE in situ. Effort on understanding and removing contribution from cosmics/cosmogenics and (known) noisy channels is ongoing.

Trigger candidate generation, building on trigger primitives and defined as two consecutive trigger primitives in both channel and time space, with a minimum hit threshold, have also been studied with MonteCarlo simulations. Trigger candidates with sufficient energy can be accepted to generate corresponding Trigger Commands for localized high energy activity, such as for beam, atmospheric neutrinos, baryon number violating signatures, and cosmics. Simulation studies demonstrate that this scheme meets efficiency requirements for localized high energy triggers, as shown in Figure ??.

Specifically, simulations demonstrate that > 99% efficiency is achievable for > 100 MeV visible energy, and that the effective threshold for localized triggers for the system is at ~10 MeV.

Low-energy trigger candidates furthermore can serve as input to the SNB trigger. Simulation demonstrates that the trigger candidate efficiency for any individual SN neutrino interaction is on the order of 20-30%. Simulations have further demonstrated that a multiplicity-based SNB trigger decision which integrates low-energy trigger candidates over an up to 10 seconds integration window yields high (> 90%) galactic coverage while keeping fake SNB trigger rates to one per month, per system requirements. An energy-weighted multiplicity count scheme could be applied to further increase efficiency and minimize background. The dominant contributor to fake SNB triggers is radiological backgrounds from neutrons, followed by Radon. It is crucial to continue working closely with the Radiological Task force to validate radiological simulation assumptions.

Given that simulation studies support requirements and rate assumptions, the protoDUNE demonstration of ability to keep up with rates from 1/25th the size of a single DUNE FD SP module, for trigger rates up to 40 Hz and 3 ms readout window allows confident scaling of the protoDUNE back-end DAQ subsystem to that of DUNE.

Add HLT simulations

The following needs to be integrated in text above, with comparable level of detail

Single-Phase Far Detector Module

The DUNE Technical Design Report
A prototype inter-process communication (IPC) system is currently under development. Some of the goals of this prototype are:

- Evaluate raw data throughput, particularly via inter-thread communication transport.
- Evaluate packet rate limitations, particularly those relevant to the hierarchical trigger layers of the data selection system described in Section 7.4.3.
- Understand the required message schema and protocols.
- Prototype high-level functionality described in Section 7.4.5 such as zero-downtime reconfiguration, self healing and the patterns required CCM as described in Section 7.4.6.
- Investigate scaling in terms of performance and software complexity management.
- Provide functional support for the vertical and horizontal slice tests described above.

The prototype software development follows some design principles:

- The DAQ is modeled as a cyclic, data flow graph.
- Graph nodes in the graph perform transformations and may consume data input from and produce data output to other nodes.
- Graph edges connect nodes through ports associated with a network socket.
- Graph construction emerges by initiating connections locally.
- Executable processes provide one or more nodes that operate asynchronously from one-another governed only by the flow of messages across their shared edges.
- Construction of nodes in executable processes and larger graph construction is dynamic governed by initial user configuration and later by messages flowing in the graph itself.

The prototype development is based on the well-established, high-quality free software from the ZeroMQ group. Some of the reasons for selecting their technology include:

- ZeroMQ software follows a well layered set of software libraries that emphasize portability, high performance, fault tolerance, minimal software dependency and long-lived use and
support.

- A wide variety of language bindings exist, importantly C/C++ for high performance and Python for fast development are two of the best supported.

- ZeroMQ abstracts functionality in important ways such as concrete implementation of high level communication patterns (eg, pub/sub) and high-level independence from low-level transport mechanism (three are supported: inter-thread queues, inter-process Unix-domain sockets and inter-computer network sockets).

- It supports a truly decentralized system design (this is the “Zero” in “ZeroMQ”) critical to satisfying the requirements on robustness. In particular the ZeroMQ project Zyre provides a distributed, low-latency discovery and presence system.

- ZeroMQ has been evaluated favorably CERN [90] and has been used in various DAQ contexts in other experiments.

Initial prototype tests have demonstrated that the raw data throughput of ZeroMQ inter-thread transport is sufficient for use in even the highest rate DAQ context (input to trigger primitive production). Rate tests of small packets have been performed to gauge ZeroMQ ability to handle high packet rate trigger primitive or trigger candidate data. These tests have so far been performed merely on Gbps networks. Given reasonably linear scaling as well as benchmarks preformed on faster networks by others, ZeroMQ will perform adequately for DUNE FD DAQ purposes.

The ZeroMQ-based prototype software will continue to be developed with the goal to understand how to manage software and configuration complexity and to participate in the other demonstrators described in this section.

### 7.5.4 Additional Teststands

Concurrently with protodune operation and development, a number of “vertical slice” teststands will be built to allow development and testing of individual parts of the DAQ system as well as testing of key aspects of the design and overall scalability. A data selection subsystem vertical slice teststand will be constructed and operated on fake-generated data, to assist in the development of data selection, exercise the system for a variety of configurations, perform small-scale tests that stress the critical parts of the corresponding infrastructure, and identify likely failure points and/or bottlenecks. The subsystem will also be deployed and exercised on existing HPC clusters of comparable resources and specifications as planned for the final production system for “horizontal slice” tests of similar nature. The back-end DAQ subsystem will be developed and tested in a similar way.

In addition to dedicated vertical and horizontal slice teststands, a number of DAQ development kits will be available for the consortium for specific component testing, as well as to other detector and calibration consortia to support their own development, production, and quality assurance programs. The DAQ kit will also form the basis for testing at APA Construction sites beginning
in 2020.

7.6 Production, Assembly, Installation and Integration

(Not finished.)

7.6.1 Production and Assembly

Describe how hardware, firmware and software will be produced.

7.6.1.1 Computing Hardware

7.6.1.2 Custom Hardware Fabrication

7.6.1.3 Software and Firmware Development

Processes and practices.

7.6.2 Installation and Integration

Alec

Describe how we get stuff in place underground, how we will put it all together and make sure it works. What can we do to minimize the effort needed underground both in terms of physical work but also in working out the bugs both in individual processes and in emergent behavior of the system as a whole?
7.7 Organization and Project Management

7.7.1 Consortium Organization

The DAQ Consortium was formed in xx as a joint single and dual phase consortium, with a Consortium Leader and a Technical Leader. The organization of the consortium is shown in Figure 7.12. The DAQ consortium board currently comprises institutional representatives from 30 institutes as shown in Table 7.6. The consortium leader is the spokesperson for the consortium and responsible for the overall scientific program and management of the group. The technical leader of the consortium is responsible for managing the project for the group.

The consortium’s initial mandate has been the design, construction, and commissioning of the DUNE FD DAQ system. To realize this, the consortium was initially organized in the form of five working groups: (1) Architecture, (2) Hardware, (3) Data Selection, (4) Back-end DAQ, and (5) Installation and Infrastructure. This organization has seen the project through the conceptual design phase.

A new organizational structure has been adopted to see the project through engineering design and construction, and this structure is expected to evolve in order to meet the needs of the consortium. This is shown in Figure 7.12. Each working group has a designated working group leader. In addition to the working group leads, technical design report editors are responsible for the overall editing and delivery of the TDR document.

This is DRAFT/IN DEVELOPMENT. From Giovanna; needs consortium input.

Figure 7.12: Organizational chart for the DAQ Consortium

This is DRAFT. Needs consortium vetting.
Table 7.6: DAQ Consortium Board institutional members and countries.

<table>
<thead>
<tr>
<th>Member Institute</th>
<th>County</th>
</tr>
</thead>
<tbody>
<tr>
<td>CERN</td>
<td>CERN</td>
</tr>
<tr>
<td>Universidad Sergio Arboleda (USA)</td>
<td>Colombia</td>
</tr>
<tr>
<td>Lyon</td>
<td>France</td>
</tr>
<tr>
<td>Iwate</td>
<td>Japan</td>
</tr>
<tr>
<td>KEK</td>
<td>Japan</td>
</tr>
<tr>
<td>NIT Kure</td>
<td>Japan</td>
</tr>
<tr>
<td>NIKHEF</td>
<td>Netherlands</td>
</tr>
<tr>
<td>University of Birmingham</td>
<td>UK</td>
</tr>
<tr>
<td>Bristol University</td>
<td>UK</td>
</tr>
<tr>
<td>University of Edinburgh</td>
<td>UK</td>
</tr>
<tr>
<td>Imperial College London</td>
<td>UK</td>
</tr>
<tr>
<td>University of Liverpool</td>
<td>UK</td>
</tr>
<tr>
<td>Oxford University</td>
<td>UK</td>
</tr>
<tr>
<td>Rutherford Appleton Lab (RAL)</td>
<td>UK</td>
</tr>
<tr>
<td>University of Sussex Sussex</td>
<td>UK</td>
</tr>
<tr>
<td>University College London (UCL)</td>
<td>UK</td>
</tr>
<tr>
<td>University of Warwick</td>
<td>UK</td>
</tr>
<tr>
<td>Brookhaven National Lab (BNL)</td>
<td>USA</td>
</tr>
<tr>
<td>Colorado State University (CSU)</td>
<td>USA</td>
</tr>
<tr>
<td>Columbia University</td>
<td>USA</td>
</tr>
<tr>
<td>University of California, Davis (UCD)</td>
<td>USA</td>
</tr>
<tr>
<td>Duke University</td>
<td>USA</td>
</tr>
<tr>
<td>University of California, Irvine (UCI)</td>
<td>USA</td>
</tr>
<tr>
<td>Fermi National Lab (FNAL)</td>
<td>USA</td>
</tr>
<tr>
<td>Iowa State University</td>
<td>USA</td>
</tr>
<tr>
<td>University of Minnesota, Duluth (UMD)</td>
<td>USA</td>
</tr>
<tr>
<td>University of Notre Dame</td>
<td>USA</td>
</tr>
<tr>
<td>University of Pennsylvania (Penn)</td>
<td>USA</td>
</tr>
<tr>
<td>Pacific Northwest National Lab (PNNL)</td>
<td>USA</td>
</tr>
<tr>
<td>South Dakota School of Mines and Technology (SDSMT)</td>
<td>USA</td>
</tr>
<tr>
<td>Stanford Linear Accelerator Lab (SLAC)</td>
<td>USA</td>
</tr>
</tbody>
</table>
7.7.2 Cost and Labor

Table 7.7 shows the current cost estimates for the DAQ subsystems major components necessary to serve the first DUNE FD module. Costs are expected to be reduced for subsequent modules, since multiple components are common across modules. When appropriate, the quantities of components are shown, along with the total cost and a brief description of what is included in the cost estimate. The cost estimates include materials and supplies (M&S) for construction, and packing and shipping to SURF, but not labor and travel costs for construction, or spares.

Labor costs depend on personnel category (e.g., faculty, student, technician, post-doc, engineer), and vary by region and institution. As such, costs are quantified using labor hours needed to fulfil a given task. Table 7.8 provides estimates of labor hours for each subsystem. Significant physics and simulation effort is needed in particular for data selection related studies; those labor resources are listed separately.

Table 7.7: Cost estimates for different DAQ subsystems. All cost estimates include M&S for construction only. Packing and shipping costs are included; spares are not included.

<table>
<thead>
<tr>
<th>System</th>
<th>Quantity</th>
<th>Cost (under development) (k$ US)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPC Front-end Readout</td>
<td>75</td>
<td>-</td>
<td>Felix and co-processor, host server, and networking</td>
</tr>
<tr>
<td>PDS Front-end Readout</td>
<td>6-8</td>
<td>-</td>
<td>Felix, host server, and networking</td>
</tr>
<tr>
<td>Low-Level TPC Data Selection</td>
<td>75</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Low-Level PDS Data Selection</td>
<td>6-8</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>High-Level PDS Data Selection</td>
<td>1</td>
<td>-</td>
<td>MLT, EXT and interface boards, High-Level Filter networking</td>
</tr>
<tr>
<td>Back-end DAQ</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Following the funding model envisioned for the consortium, various responsibilities have been distributed across institutions within the consortium. At this stage of the project, these should
Table 7.8: Estimate of labor hours for each category of personnel for different DAQ subsystems.

<table>
<thead>
<tr>
<th>System</th>
<th>Faculty/Scientist (hours)</th>
<th>Post-doc (hours)</th>
<th>Student (hours)</th>
<th>Engineer (hours)</th>
<th>Technician (hours)</th>
<th>Total (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front-end Readout</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Data Selection</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Back-end DAQ</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IPC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CCM</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Physics &amp; Simulation</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

be considered as “aspirational” responsibilities until firm funding decisions are made. Table 7.9 shows the current institutional responsibilities for primary DAQ subsystems. Only lead institutes are listed in the table for a given effort. For physics and simulation studies, and validation efforts at ProtoDUNE, wider institutional effort is involved. A detailed list of tasks and institutional responsibilities are presented in

\[ WBSref \]

Table 7.9: Institutional responsibilities in the DAQ Consortium

<table>
<thead>
<tr>
<th>DAQ Sub-system</th>
<th>Institutional Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front-end Readout</td>
<td>Institutes</td>
</tr>
<tr>
<td>Data Selection</td>
<td>Institutes</td>
</tr>
<tr>
<td>Back-end DAQ</td>
<td>Institutes</td>
</tr>
<tr>
<td>IPC</td>
<td>Institutes</td>
</tr>
<tr>
<td>CCM</td>
<td>Institutes</td>
</tr>
<tr>
<td>Physics &amp; Simulation</td>
<td>Institutes</td>
</tr>
</tbody>
</table>

2.7.3 Schedule and Milestones

2.7.4 Safety and Risks
Table 7.10: Consortium X Schedule

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Date (Month YYYY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology Decision Dates</td>
<td></td>
</tr>
<tr>
<td>Final Design Review Dates</td>
<td></td>
</tr>
<tr>
<td>Start of module 0 component production for ProtoDUNE-II</td>
<td></td>
</tr>
<tr>
<td>End of module 0 component production for ProtoDUNE-II</td>
<td></td>
</tr>
<tr>
<td>Start of ProtoDUNE-SP-II installation</td>
<td>March 2021</td>
</tr>
<tr>
<td>Start of ProtoDUNE-DP-II installation</td>
<td>March 2022</td>
</tr>
<tr>
<td>Production readiness review (PRR) dates</td>
<td></td>
</tr>
<tr>
<td>Start of (component 1) production</td>
<td></td>
</tr>
<tr>
<td>Start of (component 2) production</td>
<td></td>
</tr>
<tr>
<td>Start of (component 3) production</td>
<td></td>
</tr>
<tr>
<td>South Dakota Logistics Warehouse available</td>
<td>April 2022</td>
</tr>
<tr>
<td>Beneficial occupancy of cavern 1 and CUC</td>
<td>October 2022</td>
</tr>
<tr>
<td>CUC counting room accessible</td>
<td>April 2023</td>
</tr>
<tr>
<td>Top of detector module #1 cryostat accessible</td>
<td>January 2024</td>
</tr>
<tr>
<td>End of (component 1) production</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Start of detector module #1 TPC installation</td>
<td>August 2024</td>
</tr>
<tr>
<td>End of detector module #1 TPC installation</td>
<td>May 2025</td>
</tr>
<tr>
<td>Top of detector module #2 accessible</td>
<td>January 2025</td>
</tr>
<tr>
<td>Start of detector module #2 TPC installation</td>
<td>August 2025</td>
</tr>
<tr>
<td>End of detector module #2 TPC installation</td>
<td>May 2026</td>
</tr>
<tr>
<td>last item</td>
<td></td>
</tr>
</tbody>
</table>