# An In-Network Event Builder for the Mu2e TDAQ System

Sean Cummings[▽], Hyunsuk Bang[▽], Nishanth Shyamkumar[▽],
Michael H L S Wang[◇], James B Kowalkowski[◇], Ryan Rivera[◇], Nik Sultana[▽]

[▽] ILLINOIS INSTITUTE OF TECHNOLOGY

[◇] Fermilab

# Programmable Networking

- Switches and network cards (NICs) can apply programs to each packet while packets are being forwarded.
- It's not just the end-hosts (e.g., clients, servers) that are programmable.
- Various COTS products implement this. For example, in our lab we use Intel/Barefoot Tofino switches and AMD/Xilinx Alveo NICs.


Intel/Barefoot Tofino2

AMD/Xilinx Alveo NIC

# In-Network Computing through Programmable Networking

- Enables the rapid implementation of new features in network devices
  - Lower the cost and complexity of new feature development
  - Centralize custom processing in high performance programmable devices
- Provides flexibility in building distributed systems
  - System visibility – e.g., telemetry and monitoring.
  - Centralized control – e.g., for traffic engineering.
  - Operate at great speeds – e.g., 100s of Gbps up to 10s of Tbps.
  - Low latency – e.g., can support real-time systems.

# Programmable Networking in Data Acquisition (DAQ)

- **How DAQ can benefit from Programmable Networking:**
  - High-performance implementation – "if it compiles then it runs at line rate".
  - Simplified development and deployment loop – e.g., centralized logic, simpler/no synchronization.
  - Heterogeneous targets – ASICs, FPGAs, microprocessors, etc.

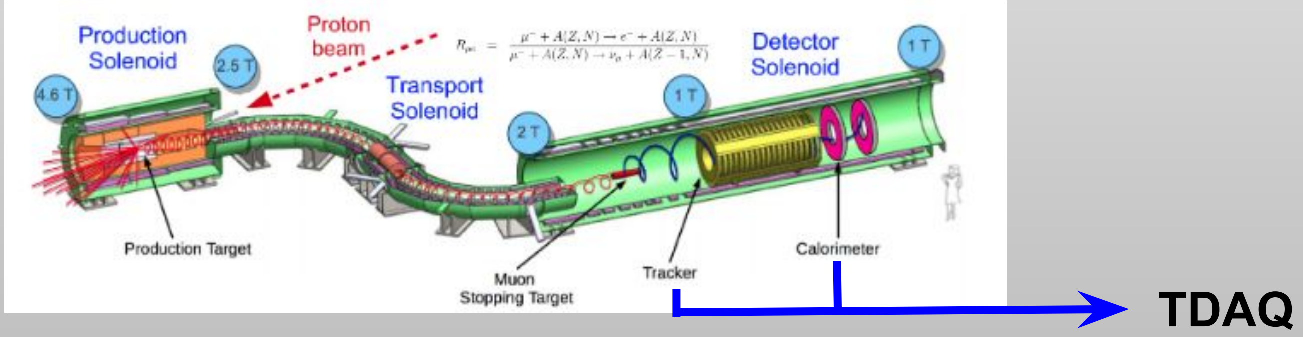
  Especially data flow-oriented systems.
- **Our project (this talk):** Event Builder + Diagnostics on programmable hardware.

  We implemented this using the P4 programming language.
- **Results to date:** prototype was evaluated (1) in simulation and

  (2) in a physical testbed.

# Mu2e Experiment

- **Why this experiment?** The observation or non-observation of muon-to-electron conversion would represent a very significant result.
  - This experiment could confirm or rule out various theories formulated to account for the shortcomings in our current understanding of sub-atomic particle physics.
- **What does it consist of?** Various systems that measure and convert signals, including the TDAQ (Trigger & Data Acquisition) subsystem.






Gioiosa, A., Donati, S., Flumerfelt, E., Horton-Smith, G., Morescalchi, L., O'Dell, V., Pedreschi, E., Pezzullo, G., Spinella, F., Uplegger, L., and Rivera, R. A.. Trigger-DAQ and Slow Controls Systems in the Mu2e Experiment. United States: N. p., 2020. Web.

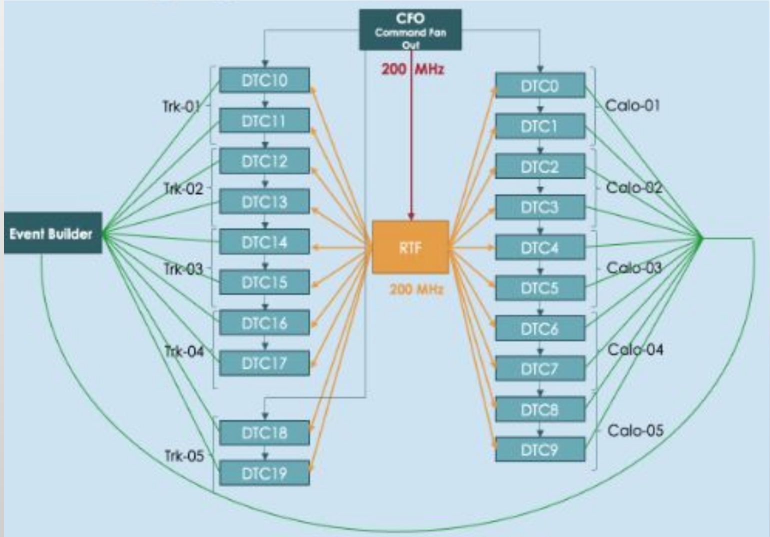
# About the Trigger & Data Acquisition (TDAQ) system

- Our prototype is located within this system.

- TDAQ:
  - Used to collect digitized data.
  - Collected data is processed online or offline.
  - Coordinates and monitors data sources.

- In Mu2e it also provides time synchronization mechanism.

- Within the TDAQ, the **Event Builder** (see diagram) composes fragments of readings from different parts of the detector.




*Partial production installation Mu2e DAQ to date*

Gioiosa, A., Donati, S., Flumerfelt, E., Horton-Smith, G., Morescalchi, L., O'Dell, V., Pedreschi, E., Pezzullo, G., Spinella, F., Uplegger, L., and Rivera, R. A.. Trigger-DAQ and Slow Controls Systems in the Mu2e Experiment. United States: N. p., 2020. Web.

# Prototype Design

- It implements 2 in-network functions on a single switch:
  - Event Building
  - Error detection
- For Event Building:
  - Programmable load balancing to different back-end servers.
  - In current prototype, using round-robin for phased load distribution among back-end servers.
- For Error Detection:
  - Per-link error tracking is implemented using registers
  - This gives per link granularity on the 7 types of specified errors
  - Registers are provided to enable system operators to quickly detect and respond to errors

# Event Header

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Event Tag Low four bytes (63-32) | | | | Res(31-24) | Inclusive Sub Event Byte Count(24-0) | | |
| Event Mode (five bytes) (63-24) | | | | | Num ROCs (23-16) | Event Tag high two bytes (15-0) | |
| Reserved (63-32) | | | | Data Source DTC ID (31-24) | EVB Mode (23-16) | DTC MAC Byte 2 (Part ID) (15-8) | DTC MAC Byte 0 (DTC Add) (7-0) |
| EMTDC (63-56) | Reserved (55-48) | Link 5 Err/stat (47-40) | Link 4 Err/stat (39-32) | Link 3 Err/stat (31-24) | Link 2 Err/stat (23-16) | Link 1 Err/stat (15-8) | Link 0 Err/stat (7-0) |

Figure 5 Sub Event Header

***(Advantage of using Programmable Networking: Support for Custom Packet Formats.)***

# Error detection

| | |
|---|---|
| Error (7) | Status (6-0) |

Figure 6 Link Error / Status Register

Status bits indicate non-error status when the Error bit = 0, and error conditions when the Error bit = 1.

| | |
|---|---|
| 6 | Fatal Error |
| 5 | Reserved |
| 4 | Reserved |
| 3 | Packet CRC Error |
| 2 | Packet Sequence Number Error |
| 1 | Reserved |
| 0 | ROC Timeout Error |

Figure 7 Link Error Flags

Stateful registers used to tabulate errors per link

# Prototype Topology

- One source traffic node generating high rates of traffic using DPDK (More than Mu2e’s current requirements).
- Two destination nodes.
- Packet from source is processed and distributed to destination nodes connected to programmable switch .

# Results

- Successful proof of concept for switch based round-robin load balancing
- Successful proof of concept for error detection
- Prototype can detect errors using stateful processing
    - Each link has an array of registers which count different types of errors that can occur on the link

# Next steps

- Investigate more context-sensitive load balancing schemes (as a function of the contents of the Mu2e custom header).
- Provide monitoring on various stats:
    - Packet count for source/destination pairs
    - Packet count by detector type
    - Packet count per link
- This information can be used to determine load balancing strategies and identify anomalies at the link layer.

# Thank you!

Any questions can be sent to Sean Cummings at: **scummings@hawk.iit.edu**