171026

Requirements for STP2

Hank Crawford, Jack Engelage, Eleanor Judd, John Nelson, Chris Perkins

The Star Trigger Pusher (STP) network is responsible for gathering all trigger data for each event at STAR. This document describes the Requirements that drive the design of the next generation of this network, the STP2. Our goals for this upgrade include increasing the trigger readout rate to >20 kHz and fabricating the new network with components expected to be available for at least the next 5 years.

Trigger data comes from ~200 9U VME boards distributed in 18 VME crates in the experiment hall. The STP network currently in use at STAR was installed in 2006 to replace the Myrinet network, a commercial network that suffered from indeterminate latencies. The STP network communicates bi-directionally with each crate CPU through a Peripheral Component Interface (PCI) Mezzanine Card (PMC) on the CPU which communicates via fiber optics with a port on an STP Concentrator (STPC) board. Each CPU receives the event trigger information from the controlling CPU (L0) through an STPC and sends the trigger via the VME backplane to all boards in its crate. The CPU then gathers information from each board via the crate backplane, whose bandwidth currently limits the trigger rate to a few kHz. The CPUs push this data to one of two STPCs in use which buffer the data and send it to the STPReceiver on the PCI bus of the L2 Linux processor where trigger events are built. L2 then returns the organizing token to the Trigger Control Unit (TCU) stack.

The STP2 upgrade will communicate directly with each VME board in the trigger, eliminating the bottleneck of the VME backplane and the older processors. A group of ~25 STP2Cs will connect the boards to the STP2Receiver (STP2R) which interfaces through the much faster PCIexpress (PCIe) to the L2 event builders.

1. **Requirement**: Receive BUILD_EVT commands from the controlling CPU on a dedicated channel and distribute these commands to all clients (VME boards).
Justification: This is how the readout of the trigger data from the VME boards is initiated.
Status: STP2C will fan out build evt to all boards.

2. **Requirement**: Receive tokens from L2 and pass them back to the controlling CPU on that dedicated channel.
Justification: This is how the controlling CPU knows that each BUILD_EVT command has been fully processed.
Status: STP2C will be able to communicate with individual clients and this is the path we will use for L2<->CPU.

3. **Requirement**: Accept variable-length data packets from ~200 clients and route them to one common destination.

Justification: We intend to upgrade all trigger VME boards to use point-to-point communication of data via a push-architecture to eliminate the VME backplane bottleneck. Each VME board will therefore respond to the BUILD_EVT command by extracting data from its local buffer and pushing it onto the STP2 network. The STP2 network must pass all of that data to L2, which will build the event and notify DAQ.
Status: The STP2 Concentrators (STP2C) are designed to accept data input from at least 8 ports. Multiple STP2Cs connected in a tree structure will therefore be used to gather data from the ~200 clients and funnel it into L2. The 200 clients implies we will need 200/8=25 STP2C boards. Each client will have a Readout Daughter Card (RDC) so we will need 200 RDCs. The tree will end in a single STP2R. Our current plan is to produce 30 STP2C, 250 RDC, and 5 STP2R board so that we have functioning spares for tests and expansion.

4. **Requirement**: Each piece of the STP2 network must EITHER be capable of transmitting output data at the maximum total rate of the input data OR contain enough elasticity buffer space to allow a burst of input data packets to be stored before being transmitted.
Justification: The instantaneous trigger rate at STAR varies randomly but the rate at which data can be funneled into L2 has a fixed upper limit. If the input data rate temporarily exceeds the maximum possible output data rate then data must be stored internally before it can be read out.
Status: The overall trigger rate and the maximum burst length will be throttled by restricting the number of available tokens. The buffer space at each step is therefore defined by the maximum incoming packet size and the maximum number of tokens.

5. **Requirement**: The STP2 network must have a small latency (<1μs)
Justification: The trigger operation depends on elasticity buffers in the VME boards as well as the STP2 network to maintain event integrity and latency sets the memory size limits.
Status: Trigger Readout Prototype (TRP) tests showed that the GTP transceiver solution works and did not lose any events.

6. **Requirement**: Accept input data on each channel at > 20Mb/s
Justification: Each QT board produces a maximum of 1kb per event:
32 chns * 32bits/chn = 1 kb/evt
1kb/evt * 20 kHz = 20 Mb/s
NOTE that each DSM produces just 128b per event, so it is the QT boards that define this Requirement.
Status: GTP transceivers on the Xilinx Artix-7 FPGAs operate at 6.6 Gb/s.

7. **Requirement**: Drive output data to L2 at > 2Gb/s
Justification: Each QT board produces a maximum of 1kb per event and each DSM produces 128b per event (see Req #6);
(100 QT bds x 1kb/bd) + (100 DSM bds * 128b/evt) = 113 kb/evt
113 kb/evt * 20 kHz = 2 Gb/s

Status: The TRP board tested throughput at 2.3Gb/s showing that the GTP links and the SDRAM buffering were robust. Note that QT data is 0 suppressed so typical events will be much smaller than this implying that events rates can be much larger.

8. **Requirement**: Work with STP PMC input
Justification: We need to operate the network during the transition from STP to STP2. After the planned upgrades to the VME boards there will still be one board (the TCU) that cannot connect directly to the STP2 network. It will continue to be read out over the VME backplane, by a CPU with an existing STP PMC card.
Status: We originally planned to design an STP2 Translator board (STP2T) to accept fiber input from the current STP PMCs and convert them to a data stream compatible with STP2 ports. Our current plan is to build GigaLink transceivers into the SPT2C to communicate with STP PMC cards.

9. **Requirement**: Use "standard" high-speed transceivers
Justification: These will be available in multiple FPGAs for many years
Status: The STP2 design uses Xilinx GTP transceivers on Artix-7 FPGAs which operate at 6.6 Gb/s.

10. **Requirement**: Use "standard" high-speed linux interface
Justification: These components are expected to be available for >5years
Status: Using PCIe interface. A four-lane PCIe device can transfer 1GB/s.

11. **Requirement**: Must establish the network in <10 seconds
Justification: This is a critical component of trigger operation and it needs to become operational without major delay in establishing data taking for efficient operation
Status: The design makes all links come up automatically and eliminates RBT programming - all configuration is accomplished through ethernet settable registers.

12**. Requirement**: The STP2 network must have a simple operator-initiated recovery sequence that completes in < 10 seconds
Justification: see 11 above
Status: The STP2 boards will be powered through a remotely controlled power supply to facilitate power cycling. In addition, each board will have a reset command response that reinitializes all control code on the board.

13. **Requirement**: Each piece of the STP2 network must have a communications path that is separate from all the data connections and allows the user to monitor the status of that piece.
Justification: It is necessary for the users to be able to monitor the performance of the STP2 network, and debug problems when they occur.
Status: Each board will have ConnectCore interface hardware for communication. This is what we tested on the TRP. We have developed VHDL-based communication with ConnectCore in a framework that is already set up. We want to add the

equivalent of m commands. Expect that each node will reside on trg.bnl.local even though we will add 200+ nodes.