

DØ Run II Data Distributor Requirements

S. Fuess, G. Guglielmo, C. Moore, L. Rasmussen & J. Yu

October 21, 1998

This document is to provide a set of requirements for the *event distributor* whose responsibility is to receive events from event collectors/routers (C/R) and requests from its client processes (EXAMINE executables), and to distribute events according to the requests by the clients. Since the EXAMINE must be able to provide maximal freedom for the users to select events in as unbiased as possible and the events are already streamed exclusively at the input stage to data loggers, the old design of attaching data distributors to the data loggers is obviously flawed (see the URL http://d0server1.fnal.gov/www/online_computing/documents/online_documents/DataLogger_19Jun98.ps for old design). The new design depicted in figures 1 and 2 would eliminate the flaw, provided each Level 3 node sees absolutely unbiased events. (*If the L3 nodes see any kind of bias in triggers, we must eliminate the bias.*)

1 Online Event Data Flow Architecture

The online event data flow architecture is extremely important in defining requirements to the event distributor, because the architecture has tremendous implications to all involved processes: Data Distributor (DD), EXAMINE, L3, Data Logger (DL). Figure 1 shows a logic diagram of the data flow and Fig. 2 shows the corresponding physical diagram. In this scheme, Level 3 nodes do not stream the events, while they still tags events with the triggers and the stream the events corresponds to, using **evpack**. The **evpack** allows the user process to examine basic information, such as triggers and streams, without doing any DSPACK unpacking (*Herb Greenlee is the author of the evpack*).

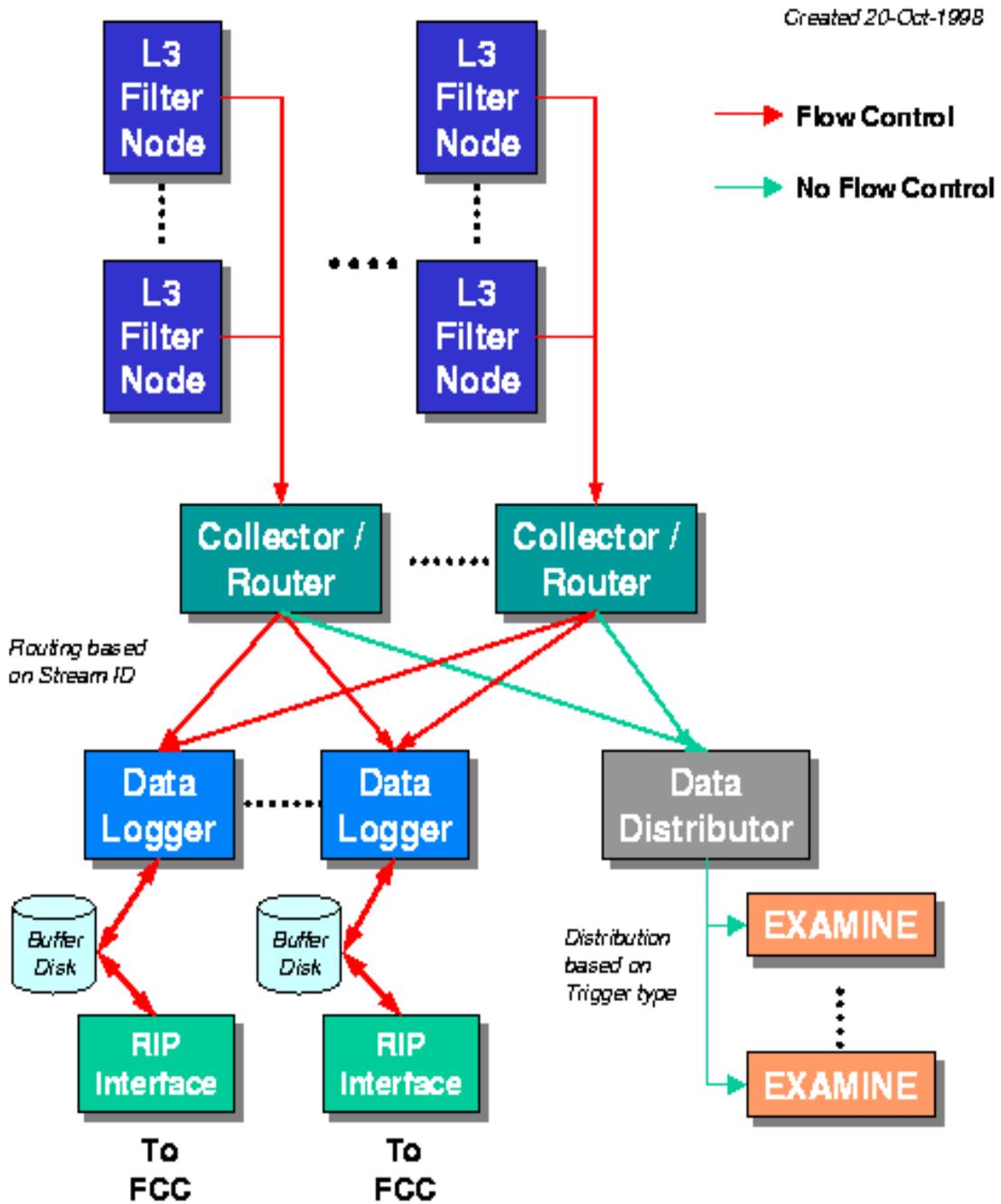


Figure 1: Logic diagram of the online data flow.

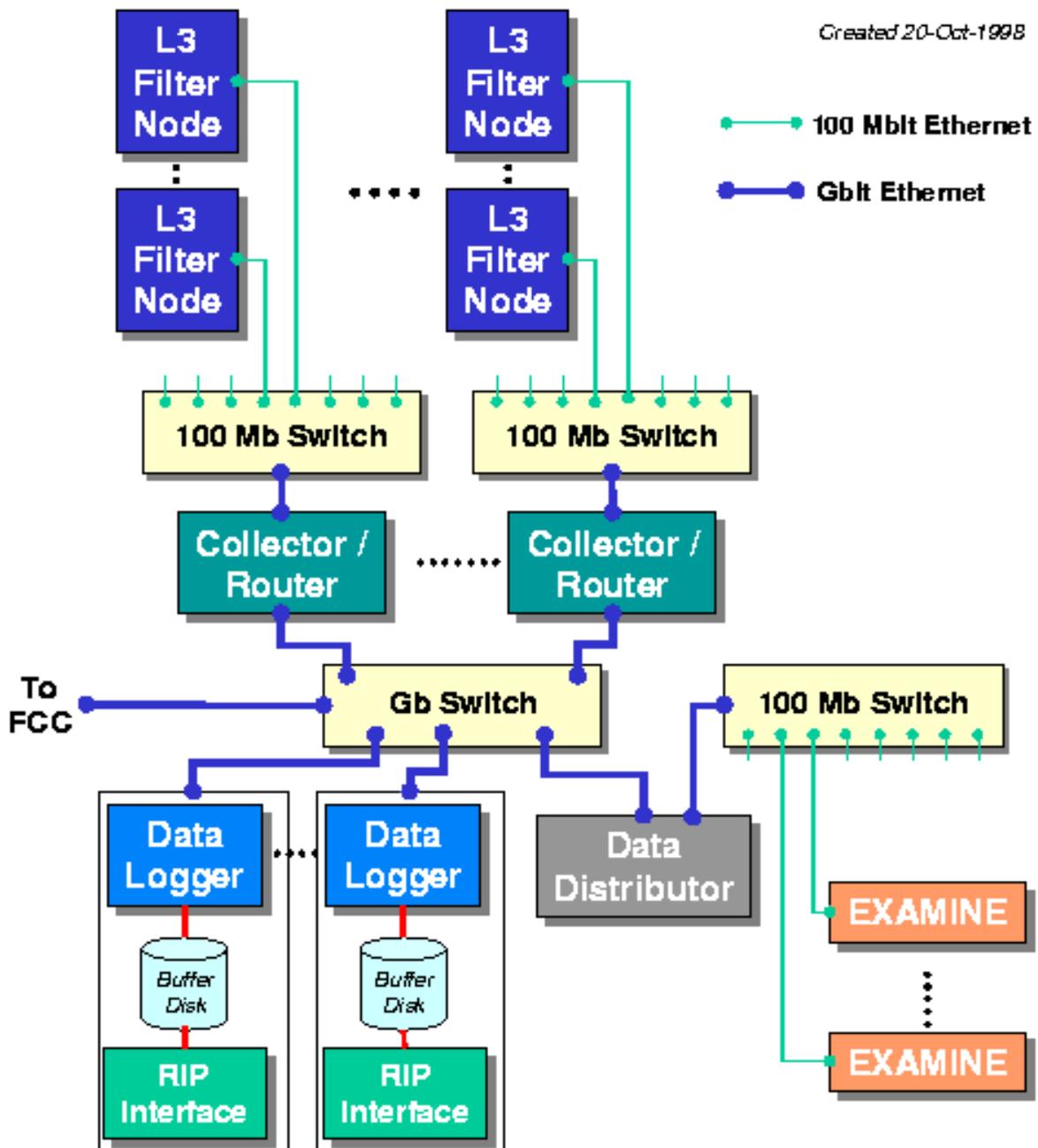


Figure 2: Physical diagram of the online data flow.

As can be seen in Fig 2, L3 nodes are connected to specific C/R nodes. The operating system and the machine as well as the total number of the C/R nodes must be determined depending on the bandwidths/rate input from L3 nodes or output to the DD node. It has not been clearly decided how much of hardware is needed, but the CPU and network capacity of the C/R nodes for sending the same event twice must be determined as early as possible for us to decide on the C/R architecture. However, one of the strengths of this design is that since the responsibilities are very much modular, we can upgrade them easily with minimal disruption.

The most important advantage of this design can be seen clearly from Fig. 1 that the event flow to the DD are **NOT FLOW CONTROLLED**. It has been pointed out by overwhelming number of people (essentially everyone in the online group) that **NO EXTERNAL PROCESS TO THE DATA TAKING SHOULD BE ALLOWED TO DISRUPT DATA TAKING**. This design would, in principle, prevent any disruption to data taking due to any of the EXAMINE or other auxiliary processes downstream of the DD.

2 The Data Distributor Requirements

Having designed an acceptable online data flow architecture, we now can provide more reasonable and stable requirements to the data distributor (DD). Although the C/R nodes to DD hardware connection would prevent any disruptions to data taking, the DD program itself must also follow this fundamental guideline. **There must be enough protection built into the DD program to prevent this from happening.**

In addition to this fundamental guideline, here are the list of requirements and conditions that the DD will need to meet. While this list will obviously grow, as we know more about the various EXAMINE functionalities, we would like to provide enough fundamental requirements to Gerry to start working on the DD to prepare for the January ICD.

- There will be one and only one DD under the currently proposed design as we have discussed in section 1. **Note that there is no flow control to the DD so that any blockage in the DD-EXAMINE end does not disturb the main data taking processes.**

- Should run on Linux/Digital unix (OSF1). It, of course, would be an additional good feature to have an ability to run on NT but not essential.
- I/O to Examine clients should be platform independent, though the OS of the EXAMINE work-horse nodes are most likely Linux under the current architecture.
- Although ideally the distributor should handle any size events, for practicality a parameterized upper limit is acceptable. Should handle the events with the typically 250kB to a maximum of 1 MB. The typical input rate is 50Hz to the DD with the typical 250kB event size. The DD should be able to handle burst input rate of 100Hz with possibly proportionately smaller events.
 - Of course the limitation here is the bandwidth of the switch in Fig 2 whose maximum bandwidth is 100MB/sec. The bandwidth needed to accept 50Hz of 250kB is 12.5MB/sec. Thus the hardware bandwidth limitation is more than sufficient to handle burst rate up to $8\times$ the average.
 - Typical output rate to clients will vary in client-by-client basis. The aggregate output bandwidth at the hardware level from DD to the clients is 100MB/sec. Since the functionalities and requested triggers are expected to differ depending on the EXAMINE's responsibility, the rate would be strongly correlated to the responsibility. However, what is important is to provide the maximum aggregated rate for DD to satisfy all the clients' requests.

Although the total sum of the requested trigger rate could exceed the hardware limitation of 100MB, in practice the clients would not be able to consume the events in such a high rate because the EXAMINEs are likely to be CPU limited. Current conservative estimate of aggregate of the DD output rate based on Run-I is at the most equal to the input rate. Of course this output rate has direct implication to buffering scheme of the DD.
- Should not limit number of attachable examine processes in software but limited by hardware (eg. size of buffered memory).

- The DD should assign a separate output buffer for each EXAMINE client at a depth sufficient to smooth out fluctuations in trigger rates. We suggest a depth of 10 events per client.
- Should, in principle, be independent of data format.

*(Scott Snyder points out that **evpack** written by Herb Greenlee allows attaching a user header to the event and this user header can be examined without doing any DSPACK unpacking. Currently the **evpack** need some work to build in NT. This feature must run on NT, because the most ideal place to attach the header would be at the L3 level while the events are composed.*

The DSPACK keeps the dictionary information separate from the data. Usually, the dictionary is written to the first record in a file; data records follow it. To handle this, the DD might have to be smart enough to cache the dictionary record it receives, and always send that to a client when a connection is established.

Another question is the transfer of histograms of the calibration runs from L3 to EXAMINE processes via DD. In Run-I, no EXAMINE process was run off of the calibration process. We will need to think about it, if we want to be able to monitor calibration runs. One of the main reasons why the calibration runs were run in L3 node was to maximally utilize the kHz input rate into L3, given the limited quiet time slots.)

- Should handle various trigger requirement configurations :
 - Single trigger
 - Multiple triggers
 - Event stream
 - Must allow EXAMINE level prescale of individual trigger in the requested trigger list.

⇒ The list of triggers should be passed from EXAMINE clients to the DD. The DD should keep the correspondence between requested list of triggers, the monitor prescale factor, and the run status EXAMINE client processes.

- Supports the query of configurations (eg. trigger selections and the corresponding EXAMINE prescale factor of the given EXAMINE process).
 - Get data event counts and transfer rates.
 - Get status of client processes.
- Should allow at least two data sources to the DD.
 - from L2 via C/R through the 1Gb (100MB) switch.
 - Reading from a file, mimicking online real time configuration. The read should occur only when another event is needed by one of the clients.
- We do not want to implement the functionality of accepting and sending every single events to EXAMINE processes, due to possible slow down of other processes. We want the users to run this kind of jobs on local files, instead of running online.
- Should be aware of Begin/End run and other run transitions by mean of a connection to the significant event (alarm) system. The DD is responsible for relaying the transition status to the EXAMINE clients.
- Diagnostics and error messages:
 - Supports queries of configuration, operation, statistics, and status, etc.
 - Generates alarms on :
 - * Buffered memory allocation
 - * Failed network connections between both the client and Collector/Router end.
 - * Buffered memory exhaustion.
 - Notify each EXAMINE client for establishment of a connection.
 - Check connections to clients (built in the Client/Server package).
 - If the connection to an EXAMINE client is lost the DD should clear all associated buffers and configurations for the client. It will be the EXAMINE clients' responsibility to re-establish the connection to the DD.