

1 Introduction

2 Requirements

- monitoring should not be intrusive
- errors should be reported quickly

3 System Architecture

3.1 T.E.D.

3.2 L2 Processors

Figure 1 depicts the division of tasks on the L2 processor nodes. Components shown in the diagram are described below.

- **NodeAlgo** : This process deals with the evaluation of the trigger algorithms as well as S-LINK I/O.
- **NodeMonitor** : The NM gathers hardware information from the system and monitoring data from NodeAlgo, packages it and sends it to T.E.D.
- **NodeController** : The NC receives configuration, processing and monitoring commands from the L2NC on T.E.D. The interpreted commands guide NC's control of NodeAlgo and NodeMonitor.

4 T.E.D. Flow Control

5 Node Flow Control

The evaluation of L2 trigger algorithms is the central process on the L2 nodes. The control and monitoring of this process is directed by T.E.D. through its L2 Node Controller and L2 Node Monitor interfaces. Corresponding interfaces exist on the L2 nodes (NodeControl & NodeMonitor) and establish a communication channel with T.E.D. External communication with T.E.D. and error conditons on the nodes occur asynchronously. The flow of control initiated by these asynchronous events may be classified as etither synchronous or asynchronous. Sections 5.1 and 5.2 describe and categorize the interface functions used in the communication between processes on the nodes.

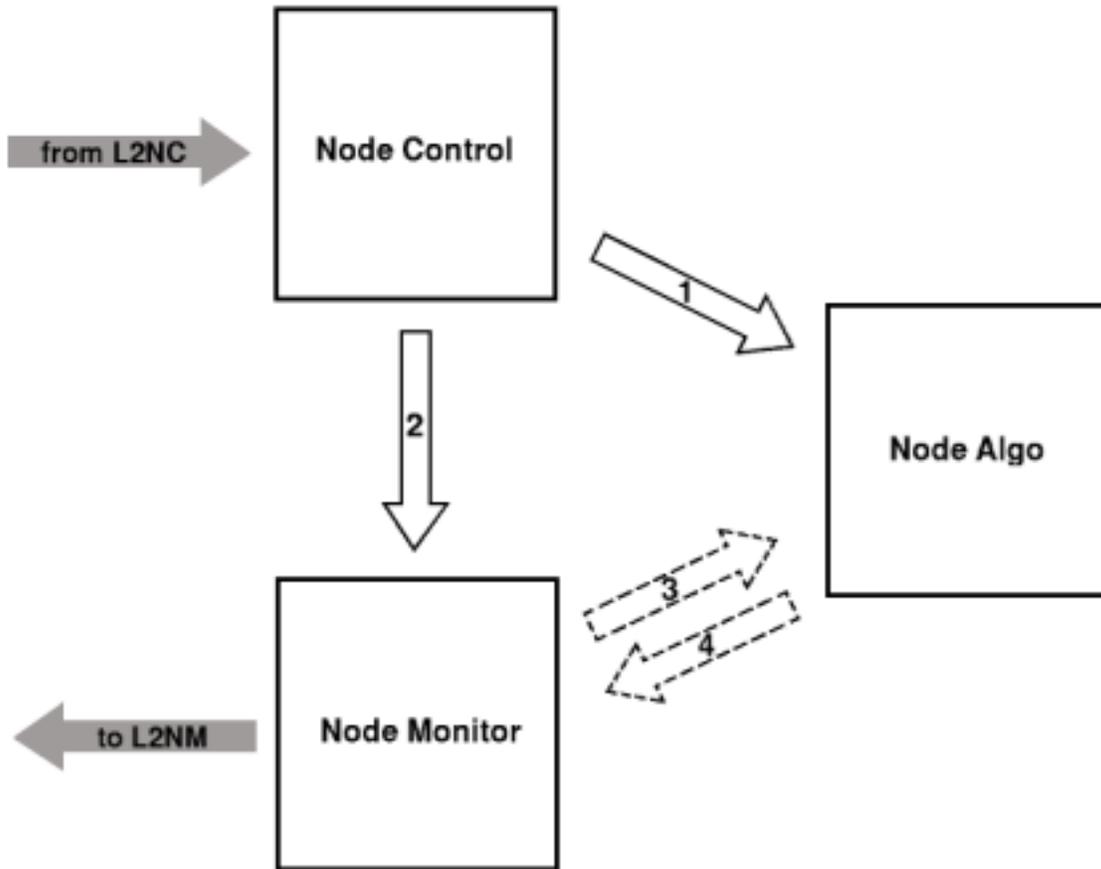


Figure 1: **L2 node Flow Control** This figure depicts the functional blocks that operate on the L2 nodes. Solid and hashed bordered arrows represent synchronous and asynchronous operations, respectively. Filled grey arrows imply external communication with T.E.D. The single-headed arrows shown point from initiator to target but information, such as acknowledgements and requested data, will also flow in the counter direction.

5.1 Synchronous Operations

NodeController

The synchronous operation of the L2NC on T.E.D. allows a node's NC to respond to single instructions from the L2NC before considering the next. The interface functions available to the NC mirror the commands used by the L2NC. Return values are sent back to the L2NC to indicate the successful or unsuccessful completion of the command.

- **bool NodeAlgo::configureAlgo(AlgoConfigSpec), (1)**
Configuration data, such as prescales and trigger table information, is passed to NodeAlgo.
- **bool NodeAlgo::setAlgoThirsty(), (1)**
NodeAlgo is told to prepare for the arrival of data.
- **bool NodeAlgo::setAlgoDrunk(), (1)**
NodeAlgo is told to ignore further data.
- **bool NodeAlgo::flush(), (1)**
NodeAlgo is told to discard the current event and clear any queued information.
- **bool NodeMonitor::configureMonitoring(MonConfigSpec), (2)**
Configuration data, such as sampling rate and ROI, is passed to NodeMonitor.
- **bool NodeMonitor::requestMonitoringData(), (2)**
Request that monitoring data from NodeMonitor be sent right away.

5.2 Asynchronous Operations

NodeMonitor

One of NodeMonitor's tasks is to collect profiling information from NodeAlgo. This operation is asynchronous so that NodeMonitor can perform its other duties (fetching HW statistics, formatting data, etc.) while it waits for information from NodeAlgo to arrive.

- **bool NodeAlgo::getAlgoInfo(MonSpec), (3)**
Request monitoring information from NodeAlgo. The information arrives when it becomes available.

NodeAlgo

NodeAlgo passes profiling data to NodeMonitor as it becomes available. Information pertaining to errors is sent as they occur.

- **bool NodeMonitor::setAlgoInfo(MonData), (3)**
This routine returns monitoring data to NodeMonitor in response to a getAlgoInfo call.
- **bool NodeMonitor::sendError(ErrData), (4)**
This routine delivers error information to NodeMonitor for delivery to T.E.D