Fusion Engineering and Design 127 (2018) 202–206

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Development of a real-time data archive system for a KSTAR real-time network

Giil Kwon*, Woongryol Lee, Taegu Lee, Jaesic Hong

Control Team, National Fusion Research Institute, Daejeon, South Korea

ARTICLE INFO

Keywords:
Real-time data translation
Network
Reflective memory (RFM)
MDSplus

ABSTRACT

This paper presents the Korea Superconducting Tokamak Advanced Research (KSTAR) Real-Time Data Archiving System (RTDAS) for a real-time network. In most fusion experiments, a stable and low latency real-time network is essential for achieving real-time controllability. Inspecting and supervising the operation of a real-time network is essential for ensuring its stability. Archiving the network data makes it possible to analyze the performance and status of a network. In the KSTAR, RTDAS has been developed to monitor and archive real-time network data that is used to control the plasma and send diagnostic data to Plasma Control System (PCS) in real-time to ensure the stable operation of the real-time network. The KSTAR adopts reflective memory (RFM) as a real-time network and this paper presents the algorithm of the real-time data archiving system and the experimental results of this system. The real-time data archiving system operates in two phases. During the experiment, the archiving system stores data from these real-time networks in real-time. After the experiment is finished the system sends this archived data to the MDSplus data server; this system stores all data in accordance with the KSTAR timing system. Therefore, all data is synchronized with the KSTAR timing system. This system is configured with Experimental Physics and Industrial Control System (EPICS) and Asynchronous Driver Support (AsynDriver). This system operates in real-time, was implemented on Messaging Realtime Grid -Real time (MRG-R) kernel 3.10, and uses Multi-Core Utilities (MCoreUtil). The system was tested to validate the performance of this archiving system; the test results show this system's real-time performance.

1. Introduction

Real-time control is essential in nuclear fusion experiments. In particular, in magnet confinement–based fusion systems, PCS connects to the Magnet Power Supply (MPS) in a real-time network to control the plasma parameters. The feedback control used in PCS requires both a fast response time ($< 50\,\mu s$) and stable operation. Achieving this response time requires that PCS and other related devices connect with a low-latency real-time network ($< 50\,\mu s$). The real-time network must be stable to obtain the stability of the feedback system. The stable operation of a real-time network requires regular inspection and supervision. KSTAR uses a RFM network as a real-time network to control MPS because RFM has a low-latency ($< 50\,\mu s$) and provides the deterministic behavior. Each connected node can share up-to-date data in predefined memory areas and each RFM node copies the data written in its shared local memory to all other RFM nodes at a transfer rate up to 170MB/s without CPU overheads. Because of these characteristics, many of tokamak adopt the RFM as real -time network [1–5]. However, an RFM network is vulnerable to data corruption and is difficult to monitor and administer. Because when any RFM node writes the data to RFM shared memory, the RFM node does not know which node wrote the data. Therefore, any RFM node in the network can change shared data without any other node knowing. This causes data corruption and it is often difficult to find which node caused a problem. RFM does not provide network status information; thus, it can be difficult to determine an RFM network's connection status. To prevent theses problem, many of tokamak archive RFM data. QUEST [4] and SST-1 [5] stores RFM data to local file system and share these data from the master node that controls entire system. However, dedicated RFM network archiving system is essential to monitor RFM network. To properly monitor the system, a third-party system must monitor the entire system. If the master node has fault, it is difficult to analyze the fault situation. In KSTAR, Real time monitoring for RFM interface (RTMON) was developed at 2012 [6]. RTMON convert RFM data to EPICS [7] PV. and archives RFM data to local file system. This system stores the data at low rate(< 1 kHz). However, in order to analyze the plasma phenomenon, a faster data archiving ability (> 1 kHz) is required. This system does not have an interface for sharing data.

* Corresponding author.
E-mail address: giilkwon@nfri.re.kr (G. Kwon).

https://doi.org/10.1016/j.fusengdes.2018.01.019
Received 15 June 2017; Received in revised form 22 December 2017; Accepted 3 January 2018

Therefore, it is hard to share and view this data. KSTAR has developed a real-time data archiving system (RTDAS) to store RFM data. The RTDAS stores RFM data in an MDSplus [8] server. These archived data can be viewed by jScope and python. jScope is a java based tool used to display data stored in mdsplus. In KSTAR, RFM data is stored to local memory during a shot. The RFM data archiving system stores RFM data at 2 kHz in local memory. The server synchronizes with a Time Communication Network (TCN) and runs at 2 kHz. TCN is an ITER Time Communication Network for absolute synchronization that is also synchronized with the KSTAR timing system [9]. Therefore, all archived data are synchronized with the KSTAR timing system. After a shot, the system transfers stored data to the KSTAR MDSplus server through the 1-GbE network.

In the current section, we wrote about an RFM data archiving system. We describe the RFM network in KSTAR in Section 2 and we describe the real-time network data archiving system (RTDAS) in Section 3. We describe performance test environment and the results of the test in Section 4 and Section 5 concludes this paper.

2. RFM network in KSTAR

KSTAR adopted RFM as a real-time network to control MPS and obtain plasma diagnostic information from a diagnostic system in a real-time manner. In the 2017 campaign, 35 RFM nodes were used to control the KSTAR system and PCS and MPS were connected to the RFM network. The fastest control cycle is set to 50 µs. The RFM card always updates the content of the RFM memory and each RFM node copies the data written in shared local memory to all other RFM nodes. RFM read/write throughputs are different; the RFM write throughputs is relatively lower than read throughput. Fig. 1. shows that the KSTAR RFM network has a tree topology; each RFM node is connected to its local RFM hub, and each RFM node and local hub is connected to the KSTAR main RFM hub.

3. Real-time data archive system (RTDAS)

The real-time data archive system (RTDAS) stores RFM data to KSTAR MDSplus server. RTDAS reads data from the RFM shared memory in accordance with the KSTAR timing system and stores data in local memory during a shot. After the shot finishes, the real-time data archive system sends this data to the MDSplus server. An analysis of the archived data permits measurement of the RFM network performance and detection of faults.

3.1. Real-time parallel data communication and processing software framework (RT-ParaPro)

RTDAS was developed using real time parallel data communication and a processing software framework (RT-ParaPro) [10]. This software framework provides real time data communication using a real-time network (such as SDN or RFM) and parallel data processing in a real-time manner. As shown in Fig. 2, RT-ParaPro provides three networks interfaces (SDN, RFM, and EPICS). Using the EPICS channel access interface means that system-required parameters can be set and the system status can be monitored from a remote location.

Synchronous Data Network (SDN) and RFM can be used as a real-time network [9]. RT-ParaPro uses multi thread pairs to process multiple data in parallel. And each thread pair implements production and consumption patterns; one pair consists of three elements: a producer thread, circular buffer, and consumer thread. The producer thread produces data by receiving data from network and pushing the data to the circular buffer. The consumer thread get data from the circular buffer and sent through a network interface. Since the two threads operate asynchronously, this system can efficiently process data from networks with different data rates. With this framework, RTDAS can efficiently store data coming from the network at a higher data transfer rate than writing data to local storage.

3.2. Architecture of a real-time data archive system

Fig. 3 describes the architecture of a RTDAS. This system was written on top of MRG-R [11] kernel–based 64-bit Scientific Linux 6.5 to provide a real-time performance RTDAS that consists of multiple

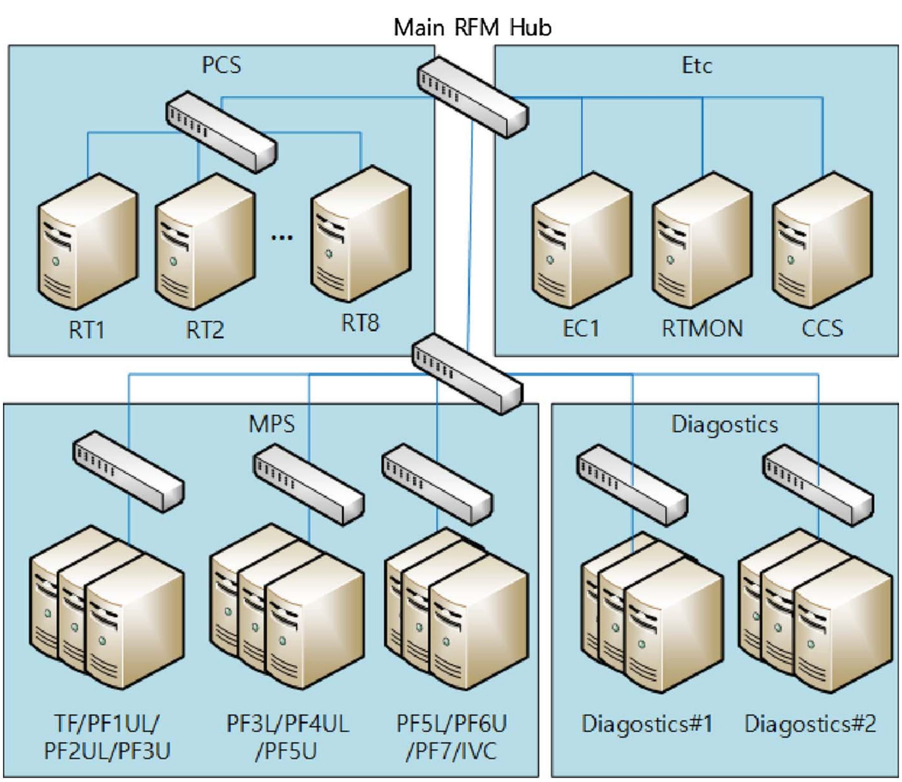

Fig. 1. KSTAR RFM network configuration.