THE IMPLICATIONS FOR NETWORK RECORDER DESIGN IN A NETWORKED FLIGHT TEST INSTRUMENTATION DATA ACQUISITION SYSTEM

Nikki Cranley, Ph.D
ACRA CONTROL, Dublin, Ireland

ABSTRACT

The higher bandwidth capacities available with the adoption of Ethernet technology for networked FTI data acquisition systems enable more data to be acquired. However, this puts increased demands on the network recorder to be able to support such data rates. During any given flight, the network recorder may log hundreds of GigaBytes of data, which must be processed and analyzed in real-time or in post-flight. This paper describes several approaches that may be adopted to facilitate data-on-demand data mining and data reduction operations. In particular, the use of filtering and indexing techniques that may be adopted to address this challenge are described.

KEYWORDS

Network, Recording, Data-On-Demand, Data Reduction, Data Mining

1. INTRODUCTION

At the heart of many networked Flight Test Instrumentation (FTI) systems is the Network Recorder. The high data rates up to Gigabit speeds achievable in networked FTI systems put increased demands on the Network Recorder to support ever faster read and write rates. However, thanks to the developments in CompactFlash and SATA technologies, such recording rates are achievable. The primary challenge is to ensure that the recorder is designed to overcome any bottlenecks writing to and reading from the memory media whilst also optimizing memory capacity usage and minimizing data loss on power outages.

The key to the performance and success of the Network Recorder to meet high data rate recording speeds, particularly for Gigabit environments, is the format in which the data is recorded. Utilizing a simple lightweight network-centric recording file format, such as Packet CAPture (PCAP) [1], vastly reduces the per-packet processing overhead so as to enable fast writing to the recording media. For given applications there may be multiple recorders dedicated to recording certain subsets of data, for example, there may be a dedicated video recorder recording only video, dedicated PCM over IP (PCMoIP) recorder, and another recorder used for all other data. To allow for this requirement the recorder uses filters to enable the user to configure and control what is recorded.

Although recording is the primary function of the Network Recorder, it is also necessary to design the recorder for fast retrieval of the data from the media supporting read-after and read-while write functionality particularly as FTI networks move towards higher data rates and the SATA Solid State Disks (SSDs) support ever increasing recording capacities. Given that the recorder may have Terabytes of data recorded over months of testing, the challenge of data mining from the recorder is not trivial.

This paper describes some of the approaches that may be adopted to address this challenge. First, there is the data mining requesting protocols that are used to allow the user to specify their read-after or read-while write data mining requirements. Once the recorder has parsed the data mining request, the recorder must then retrieve the data in question. One solution is to use indexing techniques to provide fast internal memory address pointers to key locations, events, and times on the recording media.

2. NETWORK RECORDING OVERVIEW

The network recorder is an integral component in the network data acquisition system. In general, the network recorder serves to record all the acquired data while only a subset of that data is relayed to the ground over an RF link for real-time analysis and display as shown in Figure 1.

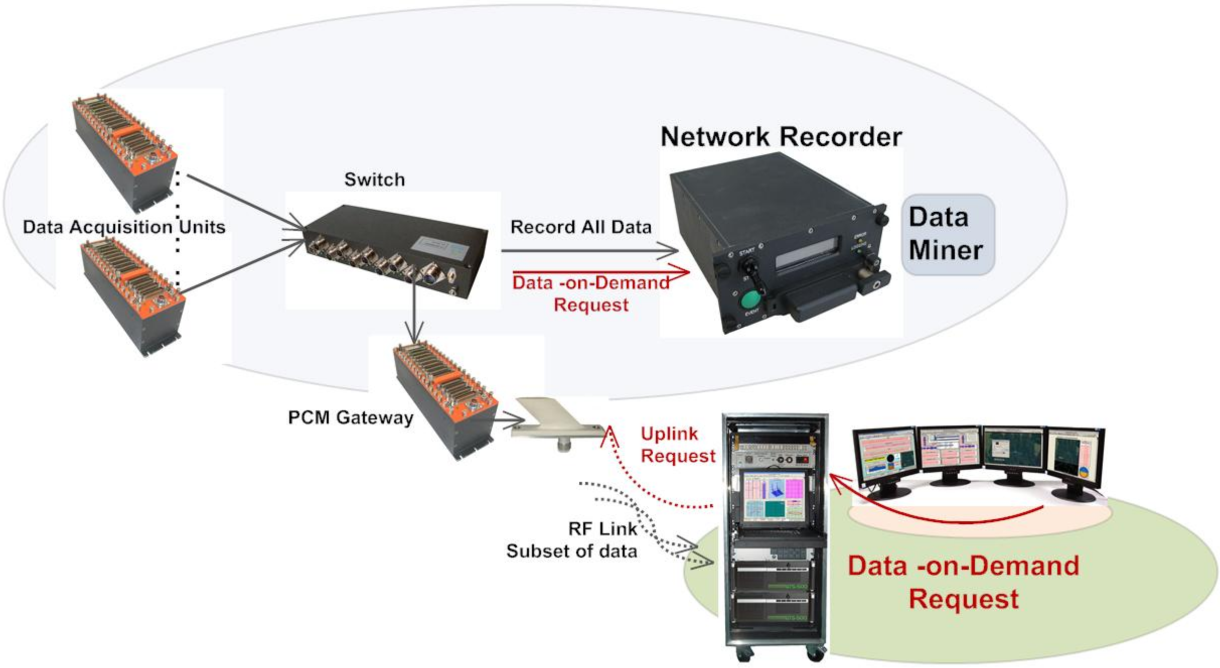

Figure 1: Typical Network Recording Architecture

The features of a FTI Network Recorder include:

- Support Fast Ethernet and Gigabit Ethernet speeds
- Support high storage capacities to the order of hundreds of GigaBytes particularly for Gigabit links where high capacity storage media is a necessity.
- Fast write rates to the storage media where incoming Ethernet frames are written to a Compact Flash (CF) card or SATA SSD with a standardized FAT32 file system.
- The recording media should be pre-formatted to eliminate the need to dynamically maintain FAT tables minimizing the effects of loss during brownouts.

- Be synchronized using either GPS, IEEE 1588 PTP or IRIG-B since packets may be transmitted asynchronously and may experience variable delays in the network, it is necessary to timestamp the arrival or capture time of the packet in order to facilitate playback of the packet stream.
- Control what is recorded either through I/O signals, multicast Internet Group Management Protocol (IGMP) subscription or data filtering.
- Support a low processing overhead and a network centric file format.
- Use standard protocols and methods to perform read-while-write and milking of the data from the recorder.

It is this latter point that is the topic of this paper. Network recorders can record large amounts of data. However, this opens a new challenge for the FTI engineers in terms of being able to perform data reduction and data mining. For clarity throughout the remainder of this paper the term dataset of interest is used to denote a packet stream(s) or iNET TmNS Message Definition ID payload(s) for a given time window or event window as requested by the user from the user during data reduction and data mining activities.

3. DATA REDUCTION AND FILTERING

Due to the high bandwidth links offered by Ethernet technology, TeraBytes of data may be recorded. The challenge now is for the FTI engineer is to be able to quickly access the data of interest for analysis. A simple solution to this is to use dedicated recorders that are designated only to record the data set of interest, for example having a dedicated video recorder, a dedicated hi-speed critical data recorder and so on. Typically the acquired data from the networked DAUs is transmitted as multicast and as such the recorder may "see" all the data in the network but needs a mechanism to be able filter the incoming data for the datasets of interest that need to be recorded. Another solution is to allow the recorder to parallelize the recording in real-time by filtering and classifying the incoming data, then directing it to the appropriate memory media or memory partition.

3.1. DEDICATED RECORDING

In general, the primary goal of the network recorder is to record only the data transmitted by the DAU's. However, there are many protocols in operation within a networked FTI system many of which do not need to be recorded. For example, routing protocols (STP, RSTP), time synchronization protocols (PTP), multicast subscription protocols (IGMP), and core protocols (ARP, ICMP, SNMP) to name but a few. There are two mechanisms by which filtering may be performed, through the use of the multicast IGMP, or through the use of the SNMP Bridge MIB filter.

IGMP is a network protocol that allows the end-nodes to signal their interest in certain multicast streams to the core network. The switches in the core network dynamically maintain and learn multicast subscriptions by periodically polling end nodes as shown in Figure 2. When the IGMP switch receives a multicast packet, it references its multicast subscription table to identify which of the connected network end nodes have subscribed to receive a copy of this multicast packet, the IGMP switch then forwards a copy of this multicast to only those devices that have subscribed to receive it. The network recorder can respond to IGMP polls issued by the IGMP switch to indicate its multicast status. This allows the recorder to promiscuously record the incoming Ethernet data streams having off-loaded the task of data reduction and data filtering to the switch through the exploit of IGMP multicast subscription.

Figure 2: Network-oriented Data Reduction

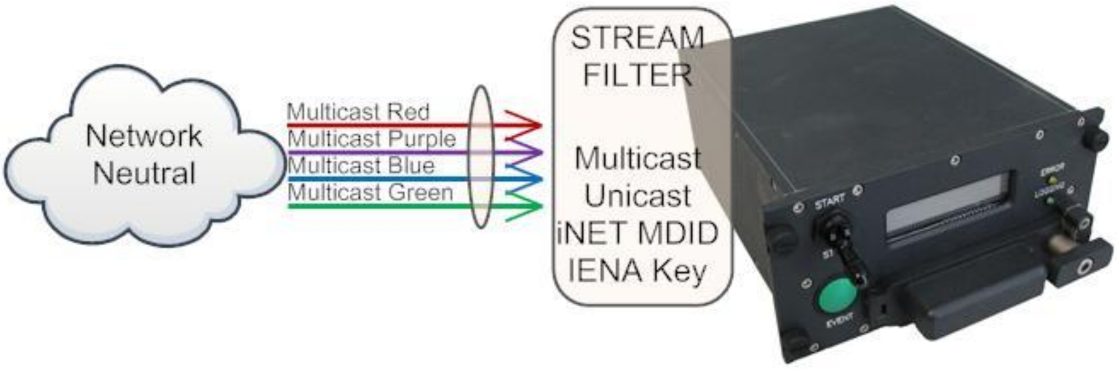

Figure 3: Network-independent Data Reduction

The downside of this approach is that it is restricted to multicast traffic, the IGMP subscription is dynamically maintained and polling intervals may be of the order of seconds. This means that the recorder may not receive the multicast stream until it has responded to an IGMP subscription poll request from an IGMP switch. Should there be any power outages in the network, the IGMP multicast subscription tables may need to be re-established which in turn exacerbates the potential data loss and recording latencies. It should be noted that the network recorder may be synchronized using the multicast protocol IEEE 1588 PTP. Since it has subscribed to the PTP multicast group, if the recorder promiscuously records its multicast subscription, the PTP traffic will be recorded unless there is an additional filtering mechanism to not write the received multicast data to the media.

A simpler, more reliable, and efficient mechanism of data reduction is to perform this filtering in the recorder itself. The recorder can be configured to filter the incoming data stream by specifying the network header fields upon which to filter as shown in Figure 3. This filter can be setup and configured [2] using the SNMP *dot1dStatic* subtree in the Bridge Management Information Base (MIB) [3, 4] as defined in RFC4188 ((Request For Comments) or the Extended Bridge MIB for bridges with traffic classes, multicast filtering and Virtual LAN extensions [5] as defined in RFC4363. Typical network header fields that are used to filter include: Source/Destination MAC address; Source/Destination IP address; IENA key or iNET Message Definition Identifier (MDID). If the Ethernet frame meets and passes the filter criteria, it is recorded to the memory media, else it is discarded. The advantages of this approach is that

- It is independent of the network and IGMP support throughout the network infrastructure

- It is not restricted to multicast; the filter can be applied to any unicast, broadcast, protocol etc.
- There is finer granularity of data filtering possible to the IENA key [6] or iNET MDID level [7]
- The filter can be reconfigured “on-the-fly” using the SNMP Bridge MIB
- The data reduction filter is live on power-up

3.2. PARALLEL RECORDING

In a similar vein to having dedicated recorders for specific datasets of interest, the recorder may support parallel recording utilizing a classification and filtering function. During the memory pre-formatting process the memory is partitioned to accommodate different datasets that correspond to the classifiers configuration. For example, consider a simple classifier with two classes of datasets of interest defined: high volume video data and the other to low volume data. The memory media is appropriately partitioned with two partitions one allocated and pre-formatting for its corresponding dataset. As the incoming Ethernet streams are received, they are parsed by the classifier-filter function, which directs the incoming data to the appropriate memory partition where they are recorded. This approach allows fast seeking to the dataset of interest particularly if there are few classifications. However, the process of parsing and classification of each incoming Ethernet frame is an additional processing overhead.

4. DATA-ON-DEMAND

Data-on-demand is the facility whereby an engineer can connect to the on-board recorder and request a specific dataset of interest to be retrieved from the recorder and played back. Data-on-demand can occur post-recording or while-recording. For example, post-recording data-on-demand is suited to mission debrief type scenarios whereby the subsets or all the data is played back from the recorder. While-recording data-on-demand is suited to scenarios where during the flight the data is lost over the RF link and needs to be retrieved for analysis, for example. To support these scenarios data-on-demand needs to be able to satisfy requests for a particular dataset of interest, for a given event window or time-window. Neither the data reduction or filtering solutions provide a means to locate the dataset of interest for a given event window or time window. The user must trawl through the filtered or partitioned recorded dataset which is time consuming and processor intensive.

The Real Time Streaming Protocol (RTSP) is an application layer network control protocol designed to control interaction with streaming data servers and recorders [8]. The protocol is used for establishing and controlling playback and real-time streaming sessions between the recorder and a client. Clients of the RTSP server issues VCR-like commands, such as PLAY and PAUSE, to facilitate real-time control of playback of either live streams, pre-recorded streams or pre-recorded files from the server. The transmission of streaming data itself is not part of the RTSP protocol. The RTSP server merely is the component, which receives and parses the data-on-demand requests and communicates these to the recorder data-mining component. The data-mining component is responsible for locating the dataset of interest in the memory media, retrieving the data and returning it back to the user. The next section discusses the challenges associated with data mining and possible approaches to support it.

4.1. DATA MINING

The challenge of data mining is to quickly locate the datasets of interest. Indexing is typically used to facilitate and speed up the location of the dataset of interest since searching through the index is faster than trawling through the raw recorded data – assuming the index is compact and efficient. The Indexer can be used in two ways. First, the FTI engineer may use the index directly in order to locate and retrieve a dataset of interest on the media for processing and analysis. Second, the FTI engineer issues a data on demand RTSP playback request and the index is used by the data-mining component to locate and retrieve the requested dataset. The support of an automatically generated and maintained index file is a secondary design goal to the reliable recording of the incoming data. The Indexer should be designed such that it

- Must not interfere with recording the incoming data in terms of slowing the ability to record data
- Act as a completely independent process to the FAT32 management and recording process
- Should be independent of and compatible with all application layer packetization protocols including iNET TmNS packets and IENA packets
- Should be interoperable with standard real-time data on demand RTSP and data download protocols (File Transfer Protocol and the Trivial File Transfer Protocol)
- Should satisfy not only data-on-demand requests but also be in a format that can be used by the FTI engineer directly for the purposes of processing and analysis on the recorded data
- Should allow for wraparound recording

4.2. INDEXING

The key component to meet this challenge is an Indexer that allows for high resolution seeking to specific datasets of interest on the memory media. However the Indexer should have minimal processing, be compact and efficient in order to facilitate high resolution, fast seeking to the dataset of interest, that is, to a specific file, specific time window, or specific event window. Reliability and performance are paramount and it is important to re-iterate that the support for data-on-demand should never interrupt or negatively affect the recorders ability to record the incoming data. Three indices have been identified to facilitate data mining by file, event window, and time-window based searching.

4.2.1. FILE AND EVENT INDICES

If the memory media is pre-formatted whereby a fixed number of empty space files are created in advance, the file index can be created in advance since the memory address of the start of each file is known, during recording the only thing that need to be updated are the associated file opening/closing timestamps. In this context, files have a fixed size but may contain an arbitrary timed duration of data as there the incoming data rate maybe variable i.e. slow/fast. The benefit of a file index for pre-formatted memory media is that since the number of files is pre-allocated, therefore the number of file index entries is also fixed which aids memory management and minimizes the processing required by the Indexer. Moreover, the file index is easier to manage when recording in wraparound mode. However, a file index only provides the granularity of seeking to a given file, there still remains the goal of being able to seek to specific event windows and time windows.

Events can be generated at any time during the recording and a single recording file may contain multiple events. However the frequency at which events are generated is significantly less than the incoming data rates and as such can be confined to small memory space where the Event Indexer provides a timestamped memory address pointer to recorded events.

4.2.2. TIME-BASED INDEX

By far the most complex task of compact and efficient indexing is being able to generate a Time-based Index. One solution is to use Timed Sequential Files (TSF). In TSF the memory is preformatted with a large number of sequential files where each file is allocated enough space to store N seconds of data. For example, on a 100BaseTX Ethernet link it can be assumed that the maximum amount of space required to store 1second of recording is 100Mbits. From this it would be known that the file contains 1second of data since it was opened. But since the recording may start and stop at arbitrary times, it is not true that files are always consecutive in time, therefore an index file would need to be updated indicating the opening/start time of the file of each file used. The advantage of this approach is that it is compatible with memory pre-formatting schemes, wraparound recording, and is deterministic. However, such an approach can be potentially wasteful in terms of memory management. Since if there is a low amount of traffic on the network and each file can hold up to 1second of data. For example, given an incoming data recording rate of 50Mbps for which and each file can store 1second of data (assuming a 100BaseTX Ethernet link), there is 50% unused space allocated per file. As the memory capacity increases, the number of files that would be held on the media increases dramatically and increases the workload on the FAT32 manager to frequently open and close files and update the corresponding file entry start time in the TSF Index. A 1TByte capacity SATA SSD could have ~8x10^4 files (for 100BaseTX where each file holds 100Mbits of data) or ~8x10^3 files (for 1000BaseT where each file holds 1000Mbits OR Gigabits of data). A corollary of having a large number of files is that the TSF index would also be large to hold the start time of each file opened during the recording. The larger the TSF Index, the longer it takes to search the TSF Index to satisfy an incoming time window based data-on-demand request. As before, the duration of the recording files directly affects the resolution of the indexing and seeking, that is if each file contains a maximum of 1second of data, the best resolution that the seeking is to within 1second of the requested time window. Finally, in post-analysis, often the recorded data files need to be merged, processed, and archived. Post-processing many small files may prove time consuming outweighing the utility of the TSF approach.

A variation on the time-based approach is a space-based approach. Since memory is effectively a space that is filled, timestamped address pointers can be generated every *N* Kilobytes. Since the size of the memory is fixed, address markers can be generated in advance and updated only with an associated time. Moreover, this approach is compatible with wraparound recording.

5. CONCLUSIONS

Ethernet technology offers many benefits to the FTI community including open standards-based technologies, greater vendor inter-operability, system design flexibility and simplicity. One of the primary advantages of Ethernet technology is that more data can be acquired and carried over the FTI network. Typically, the network recorder logs all data that is acquired in the networked data acquisition system whilst a subset of the data identified as mission critical data is transmitted over a PCM RF link to the ground for real-time analysis. However, the increased acquisition data rates pose fresh challenges for the network recorder. Not only must the data recorder be able to log/write to memory at increased rates but they must also have sufficient memory capacity to store the incoming data. Using state of the art technologies network recorders are capable of logging hundreds of GigaBytes of data. But the challenge now is to process and analyze the logged data. This paper describes several approaches that can be employed to satisfy data-on-demand requests. Of the techniques discussed include real-time data reduction for quick access by exploiting the Internet Group Management Protocol (IGMP) multicast subscription mechanism or data filtering within the recorder. Often data-mining operations involve the extraction and filtering of specific data streams for a specified event window or time window. To support this operation, an index file is required. This paper discusses the issues and requirements of a high-resolution compact and efficient indexing mechanism.

6. REFERENCES

[1] Wireshark PCAP, http://en.wikipedia.org/wiki/Pcap, Available online: 17th June 2011

[2] Cranley, N., "The Implications for Network Switch Design in a Networked FTI Data Acquisition System", To appear in Proc. ITC 2011, Las Vegas, NV, USA

[3] IETF RFC 4188, "Definitions of Managed Objects for Bridges"

[4] IEEE Std. 802.1D 2004, "Media Access Control (MAC) Bridges",

[5] IETF RFC 4363, "Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions"

[6] Caturla, J. P., "IENA Packet Format and Generic Control Tools", Proceedings of European Test and Telemetry Conference Toulouse, France, 2003

[7] Grace, T, Abbott, B, Moodie, M, ""iNET System Operational Flows", Proceedings of International Telemetry Conference 2010, San Diego, CA, USA

[8] IETF RFC 2326, "Real-Time Streaming Protocol"