AN IP-BASED RECORDING SYSTEM

John Roach
VP Network Products Division

John Hildin
Director of Network Systems

Teletronics Technology Corporation
Newtown, PA USA

ABSTRACT

Traditionally, acquired instrumentation data on a non-destructive test article is recorded to a non-volatile memory recorder. The data acquisition system usually samples and formats its inputs before transmitting the data to the recorder (also known in this paper as a data sink) via a PCM serial data stream (i.e., clock and data). In a network-based data acquisition architecture, the inclusion of an IP-based recorder adds a new dimension to the data acquisition process. Any IP network inherently allows for the bi-directional exchange of data. In this environment, the IP-based recorder can be treated as both a data sink for parameter recording and a data source for parameter extraction, data rate statistics, and recorder status reporting. The network model recasts the data recorder's function as a file server to which multiple clients could be simultaneously requesting services. Those clients that represent the data acquisition nodes are requesting storage of their acquired parameters. Clients, such as transmitters or test engineers, are requesting access to archived data or status information for further processing.

This paper presents the advantages of using an IP-based recorder in a network-based data acquisition system. The availability of an IP interface along with the intelligence built into the recorder expands its capabilities beyond that of a conventional PCM recorder. These capabilities include real-time health monitoring, support for the Simple Network Management Protocol (SNMP), data mining, reporting of real-time performance and network statistics.

KEYWORDS

Network, IP-based Recorder, SNMP, Data Acquisition, Data Mining

INTRODUCTION

Flight instrumentation recorders, whether magnetic tape, disk or silicon-based, have been typically accessed via a serial (typically PCM) simplex interface. The recorder's purpose has been to accept serialized data and transfer the data to the recording media and later, retrieve the data from the media for archiving and/or analysis purposes. The level of computing intelligence needed in the recorder has been relatively low. Not too many years ago, recording systems were

available that included the data multiplexing function, either in the same physical chassis or as a separate unit.

Today, as the acceptance of network-based data acquisition systems becomes more popular, there is a benefit to extending the network directly to the recorder and making the recorder effectively another network node. In a more traditional sense, the network-attached recorder inherently becomes a file server. As a file server, the recorder is not only capable of recording data, but it can also concurrently retrieve data. The ability to retrieve recorded data during a test mission is a significant benefit. One reason is being able to access data related to safety-of-flight measurements that are not otherwise telemetered in real-time.

Although an IP (Internet Protocol) Recorder can be an integral part of an IP-based data acquisition system, a complete treatment of such a system is beyond the scope of this paper. The reader is encouraged to refer to other documents, including those cited in the references section, to gain an overall understanding of elements of an IP-based data acquisition system.

IP RECORDER BASICS

The fundamental components of an IP recorder are the network interface, an IP stack, a processing element, and the storage media. Each one of these elements determines the overall performance, features and usability of the IP recorder. The network interface is that portion of the recorder that is used to attach to the network transport. The type of interface used would normally be determined by the overall system network that carries the acquisition data from the source nodes. Some possible examples would be Ethernet, Fibre Channel, and IEEE 1394. The choice of network transport can determine the reliability of packet delivery, the jitter and latency experienced by the packets before they arrive at the recorder, and the size of the packets.

A data recorder capable of communicating on an IP network is not much different from other network nodes in that it will run a software stack that will terminate the IP protocol. The lower layer is not necessarily tied to the stack as the IP protocol can be implemented over Ethernet, Fiber Channel, and other physical layers. The services available within the IP stack determine what form the packets take when they arrive and the complexity of sorting thru the different streams to determine how to place them onto the media. It's possible that data streams from different data sources might arrive as a mixture of TCP and UDP packets. Those packets sent via UDP could be sent as unicast or multicast streams. At a higher level, the data may arrive at the IP stack as iSCSI protocol. This is a data and command transport protocol based on SCSI, a long time industry standard for interfacing with storage media. iSCSI is commonly used in Storage Area Networks (SANs) in commercial applications for data storage, but requires a higher level of software complexity and computing resources in the data sources. A tradeoff could be made here, higher complexity in all network elements (in return for flexibility (iSCSI) and transparency of data on the wire) or lower overall system complexity (in return for slightly less flexibility (TCP/UDP) and data transparency).

Directly related to the complexity of the data transport is the use of a processing element to receive, interpret, and route data received from the IP stack. The complexity of the software

needed to manage this process; at the speeds that data is expected to arrive at the recorder; determines the needed performance of the processing element. Given that hundreds of thousands, if not millions of packets per second may be arriving at the recorder, it may be necessary to support multiple processing elements to direct all that data. The choice of the physical network transport might influence the choice of processing element, for example, Ethernet is commonly found integrated within many system processors while Fibre Channel is not.

Ultimately, all of the packets that arrive at the network interface find themselves placed on the media for permanent storage. The pathway from the network interface to the media needs to be able to sustain the data rates expected from the network. Typically, industry standard media interfaces include Fibre Channel, IDE, SCSI, and Serial ATA. Each technology has its own limitations regarding performance, in addition to the complexity of translating data arriving over the network into a form suitable for each media interface. Another performance limitation is the actual speed of the media. Writing performance to a single instance of standard media available today (either hard disk of solid state) normally tops out at 40 megabytes per second. In order to achieve higher recording throughput, it is necessary to use multiple instances of media and overlap the storage of data packets with multiple simultaneous writes. To do this efficiently requires the use of large packet buffers or memory caches, these buffers smooth out the jitter introduced by the network and smooth out the delays in writing out packets caused by mechanical movement of assemblies in hard drive media.

ORGANIZATION OF MEDIA

A critical decision within the IP recorder is the way the media is organized into storage. Given that it is often necessary to use multiple media instances to achieve the necessary recording performance, the way that data is laid out on the media array will determine the complexity of extracting the data after the test flight. The technology commonly used to manage this complexity is RAID (Redundant Array of Independent Disks). RAID technology commonly appears in Storage Area Networks (SANs) and supports numerous options regarding performance and redundancy. The type of RAID being used is typically identified by its level, with each level offering different combination of capabilities. The following table highlights some of the commonly found configurations:

Level	Striping	Mirroring	Parity	Storage Penalty
RAID 0	Yes	No	No	None
RAID 1	No	Yes	No	100%
RAID 3	Yes	No	Yes	1 Drive
RAID 5	No	No	Yes	1 Drive
RAID 10	Yes	Yes	No	100%

The choice of how to organize media on an IP recorder depends on the needs of the mission. In most cases, RAID 0 would be sufficient, as the primary concern is simply to handle the incoming data rate. In some cases, perhaps when using hard disks on a flight that has higher reliability

concerns, RAID 10 might be more appropriate in order to provide some measure of data reliability in the event of failure of one of the drives.

IP RECORDER – DATA SINK AND SOURCE

Traditionally, a data recorder has acted as a simplex device. Data flows in one direction at a time. While data is acquired on a test article, the data is collected, formatted and sent to the recorder. The recorder acts as a data "sink". An IP Recorder, however, is also able to retrieve data from the recording media and provide that data to the requesting node, thus also acting as a data "source". This ability comes as a result of terminating the IP protocol, in this case, and making use of the processing power built into the recorder. Additionally, it depends upon the ability of the recorder to index the data being recorded in real time, which allows for external queries to locate a random packet on the media, based on time. In order for this function to be practical in a flight test environment, it is necessary to be able to read data from the media without interfering in the recording of the flight data. This is where the use of RAID technology in an IP recorder can dramatically increase the flexibility of the system. A recorder that uses multiple drives and stripes the recorded data across the drives provides ample opportunities for a read operation to occur on an idle drive while many parallel writes may be occurring. This way any idle bandwidth under the peak bandwidth of the array can be exploited for reads while writing.

INTELLIGENT RECORDING

Adding a microprocessor, and associated software, to a recording system adds a great deal of flexibility and allows for a rich feature set. Part of this system's flexibility comes from the ability to upgrade the installed software in the field. For example, a simple software download could be all that is required to provide a new feature. This also helps to future-proof the customer's investment and prolong the usable life of the product. The power of the microprocessor can be harnessed to provide useful features to the customer. For example, the processor can monitor data traffic into and out of the recorder and report the statistics to a client elsewhere on the network. Similarly, the processor can monitor the health of the recorder itself by reporting on the internal temperature, bit errors, internal voltage levels, etc.

In addition to the features above, the processor is responsible for terminating the IP protocol and providing other services. Another useful service within an IP-based device is device management. This management service allows other network clients (e.g., management agents) to retrieve parameters from the recorder and push programming to the recorder. These operations take place while data is being recorded to and extracted from the media. Protocols exist that implement the management function. One popular protocol is the Simple Network Management Protocol or SNMP. SNMP is used widely on many IP networks.

Figure 1 below shows a basic block diagram of a PCM-based recorder. PCM clock and data are driven to the recorder's PCM interface. The serial PCM data is captured and formatted for recording. The formatted data is then sent via the media interface logic to the media. It is not

uncommon for one or more of the functional blocks below to be implemented in a programmable logic device, e.g., an FPGA.

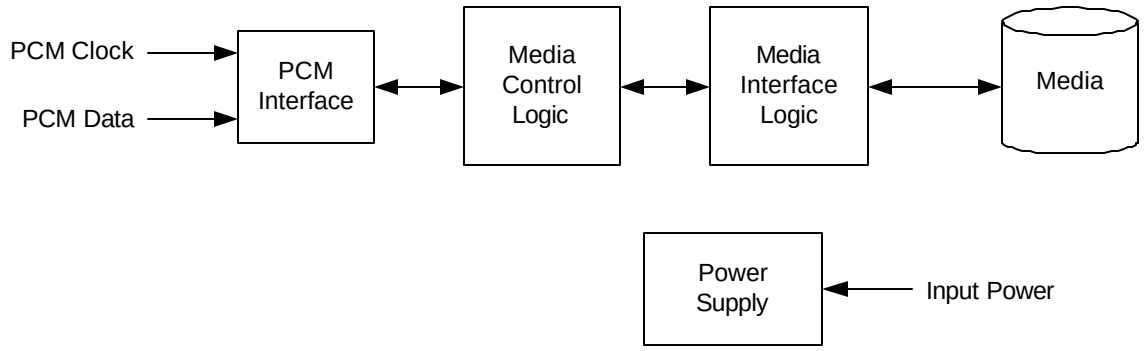

Figure 1 PCM Recorder High-Level Block Diagram

Figure 2 below shows a basic block diagram of an intelligent IP recorder. Here, data is delivered to or taken from the recorder's network interface. Received network data packets are then sent to the processor and stored in local volatile memory. Data packets are processed (i.e., the IP Protocol is terminated, error checking is performed, etc.) and sent to the media control logic. The media controller performs additional conditioning of the data in preparation for transfer to the recording media. This is where specific RAID, data formatting, CRC, and other functions may be performed, and recording algorithms implemented. Lastly, data is delivered to the media via the media interface logic. The media interface logic provides the physical interface to the recording media.

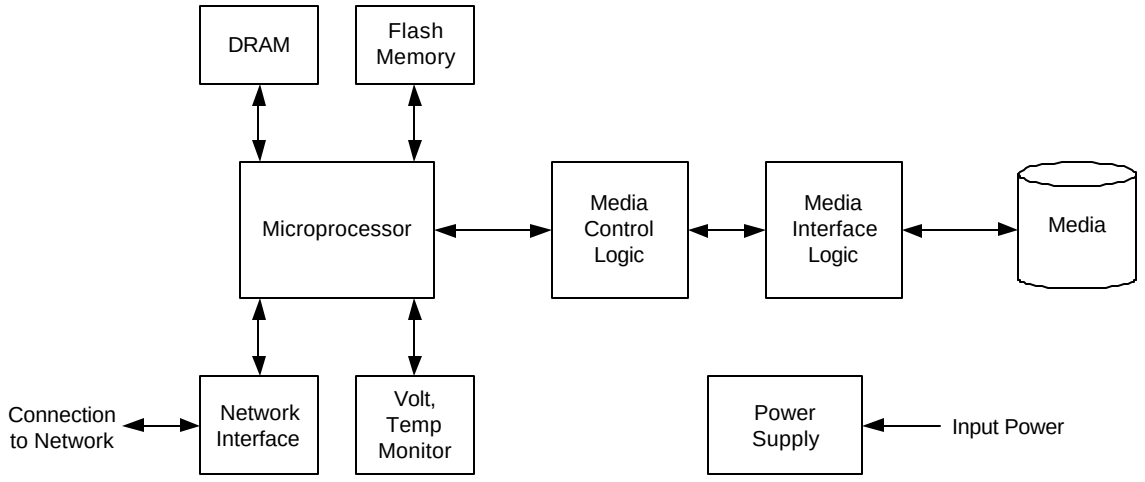

Figure 2 IP Recorder High-Level Block Diagram

CONCLUSION

With the addition of network connectivity and some processing power, the role of the data recorder can be expanded. An IP Recorder can be used as more than a non-volatile data storage platform. The recorder can now be used as a real-time, bi-directional data resource to other clients on the network. And that network can extend beyond the test article, giving access to clients on the ground or within other test articles. Bringing the data acquisition network to the recorder provides added functionality and value to the customer and is a natural extension of an overall networked data acquisition system.

REFERENCES

1. Berdugo, Albert A., "A Multiplexer/Recorder Architecture For Use With Conventional Media Technology", Proceedings of the International Telemetering Conference (ITC 2005), Las Vegas, NV USA, October 24-27, 2005.

2. Berdugo, Albert A., Hildin, John J. "A System Approach to a Network Centric Airborne Data Acquisition System", Proceedings of the International Telemetering Conference (ITC 2006), San Diego, CA USA, October 23-26, 2006.