Migrating from PCM to Networked Data Acquisition Systems

Albert Berdugo and John Roach
Teletronics Technology Corporation
Newtown, PA USA

Abstract

Airborne PCM data acquisition systems have been in use for the past few decades by the flight test industry. They are relatively simple to use, require minimal configuration and operational software, easily found in very small units to very large distributed systems, and above all are well understood by the user community. On the other hand, Airborne Networked data acquisition systems are relatively new to the instrumentation community, and require new ways of thinking about the best way to apply them in this domain, and a fresh look at the old assumptions used to build our current systems.

This paper will examine and compare the basic assumptions, methodology and characteristics of the traditional PCM Data Acquisition system versus those of a Data Acquisition system built upon networking principals. In particular, this paper will pay special attention to the problems of time distribution and synchronization that arise in a networking paradigm.

Keywords

Data Acquisition, PCM, Networking, Internet Protocol, IEEE-1588

Introduction

Most major flight test programs around the world use PCM-based data acquisition systems. These systems are based on a Time Division Multiplexing (TDM) scheme and are highly deterministic in nature. PCM systems are relatively small and use a simple design to execute and generate a PCM data stream. The system may include a single unit or a group of units that form a distributed data acquisition system. In most cases, a distributed system is well synchronized by way of hardwired synchronization signals or through the use of a bus such as CAIS (Common Airborne Instrumentation Bus). The format that controls the parameter acquisition sequence is generally located in the PCM controller, and sometimes (based on the system manufacturer) a copy of that format or subset of that format may also reside in the remote acquisition units. The format is executed at a predetermined rate by simple state machines.

This paper will discuss key elements of a PCM-based data acquisition system, and then describe equivalent key elements related to a network-based data acquisition system. Within these elements, this paper will describe the basic assumptions, characteristics, control, and time distribution characteristics of both PCM and Network systems.

Elements of a PCM-Based Data Acquisition System

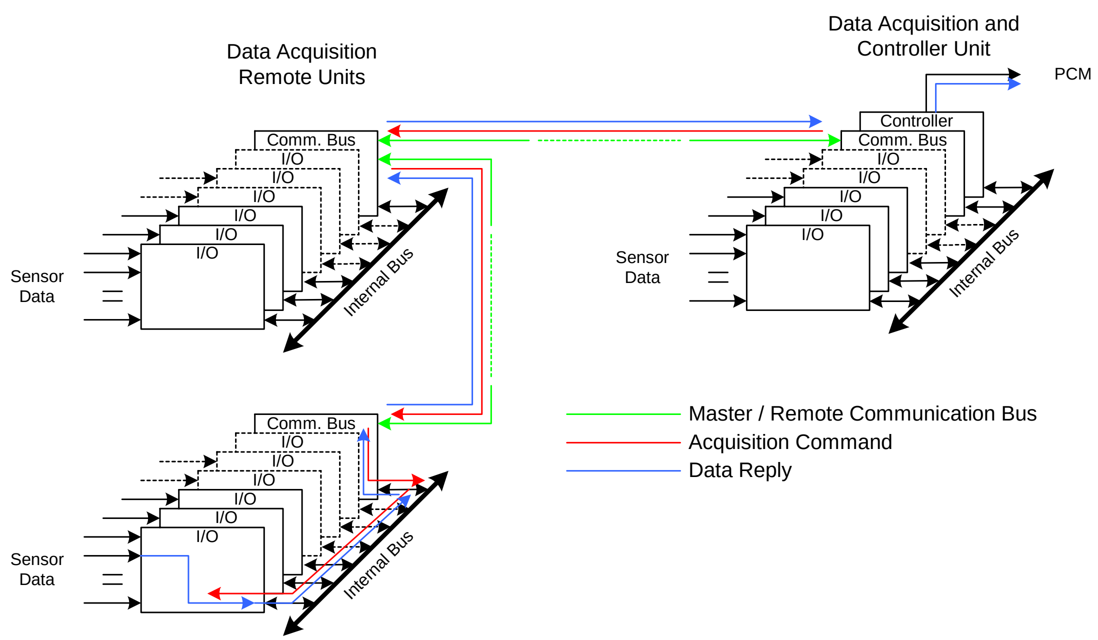
PCM-based data acquisition systems are so well understood by most flight test engineers and PCM manufacturers to the extent that the system basic characteristics are rarely discussed. It is important to review these basic characteristics in order to compare and challenge them when one migrates from the PCM paradigm into a networking paradigm.

A PCM system can be usually characterized as having:

- Unidirectional data movement
- Synchronous arrival of data
- Static quality of service parameters
- Strict separation of control and data
- Strict separation of time and data
- Accurate global timing synchronization

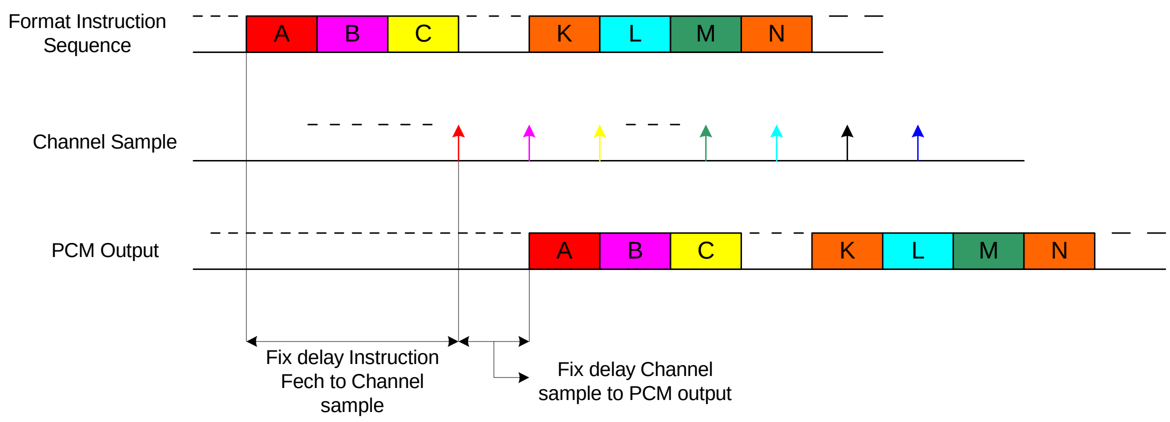
Unidirectional Data Movement

The unidirectional data movement is the result of a PCM controller executing command instructions from a format map to acquire data for insertion into the PCM stream. All or a subset of the data is transmitted to the ground, and all the data is recorded on an onboard recorder. The system and its wiring are rigid, and data cannot be redirected to other potential destinations within the acquisition system. Normally, a PCM data stream is transported within the aircraft through use of differential or singled-ended copper wires, carrying clock and data signals. When common data needs to be directed to other specialized airborne units (such as a cockpit display or processing element), it is done by way of additional unidirectional data wiring carrying the same information thru a duplicated clock and data signal.



Synchronous Arrival of Data

The synchronous arrival of data is due to the system architecture and customized wiring between units in the system. The traditional architecture is based upon the execution of fixed format instruction commands operating at a highly precise PCM word rate. Each command instruction within the format is transmitted in the system to the appropriate channel for data sampling. Data is sampled synchronously to the overall PCM word rate and format sequence. Data arrival at the PCM output is pipelined with a fixed delay for all channels in order to provide a PCM output channel sequence identical to the user input format sequence. System wiring is optimized to the architecture, and is mainly used for movement of instructions to the channel to be sampled, and movement of data toward the system controller's PCM output.



Static Quality of Service Parameters

PCM systems are constructed to have a static quality of service. The PCM system is well understood, and a known quality of service is assumed from the design. The quality of service for a given system, wiring, and format is static due to the following known factors:

- Format instruction movement is unidirectional and point-to-point.
- Data sample movement is unidirectional and point-to-point.
- Fixed system wiring with known bandwidth. (PCM rate is known)
- No retransmission of data.
- PCM systems are generally state machine-based with fixed sequencing and fixed pipelining.
- No data collision.
- Fix known delays based on bit rate, word rate, and frame rate.

Strict Separation of Control and Data

The control and data paths are strictly separated in a PCM system. Format execution is stopped when control operations are done via data wiring path. If the data path is not allowed to be interrupted, additional wiring is required to perform the controlling functions while collecting data. Control functions are used to reprogram the system, alter channel configuration, perform system audit, and collect status.

Strict Separation of Time and Data

Time distribution is done separately from the data path in a PCM data acquisition system. In general, units operating in a distributed data acquisition system require external time wiring. The time is used to time tag asynchronous avionic bus data and other acquired asynchronous events. Time is also used for time tagging each PCM minor frame. Time distribution units may be required when a very large data acquisition system is used. In most cases this may require two wires per acquisition unit to distribute IRIG-A/B/G modulated or DC time. IRIG time is further refined by the acquisition units to provide time tagging with a +/-1 microsecond of accuracy.

Accurate Global Timing Synchronization

PCM data acquisitions provide a high level of accuracy for global timing synchronization. This is due to the time division multiplex nature of the system, and the distribution of the PCM controller’s clocks to all units in the system. Although such a system is highly synchronized, designers and users make several synchronization assumptions that should be understood. These assumptions include:

- Zero time delay in system wiring between data acquisition units operating in a distributed architecture.
- All synchronization signals are part of the communication busses between data acquisition units.
- Data samples are controlled and sequenced by a single PCM controller or multiple synchronized controllers.

The zero-time-delay-in-wiring assumption should always be reevaluated when simultaneous sampling is used within a distributed system. This is because it is common to find 100 to 200 feet of wiring between remote data acquisition units. If one assumes an electrical signaling delay of 1.7 nanosecond/foot, the propagation delay between multiple units can be as much as 300 nanoseconds to 500 nanoseconds. Therefore, two units receiving simultaneous sample signals may see the signal at a slightly different time. This time can be as much as 500 nanoseconds.

Elements of a Networked-Based Data Acquisition System

In contrast to the proceeding, network-based data acquisition systems are so poorly understood by most flight test engineers and PCM manufacturers that the system basic characteristics are often undefined or in disagreement. We believe, however, that the basic characteristics of a networking architecture are well understood and understanding those basic principals allows one to easily distinguish some of the fundamental differences between this architecture and the one which best represents a PCM-based system. It is important to review these basic differences in order to validate that when one migrates a data acquisition system from the PCM paradigm into a networking paradigm, the basic requirements that made a PCM-based data acquisition system useful are preserved.

A networking-centric data acquisition system can be usually characterized as having:

- Packetization of data
- Bi-directional packet movement
- Asynchronous arrival of data
- Dynamic quality of service parameters
- Complete intermixing of control and data
- Low accuracy global timing synchronization

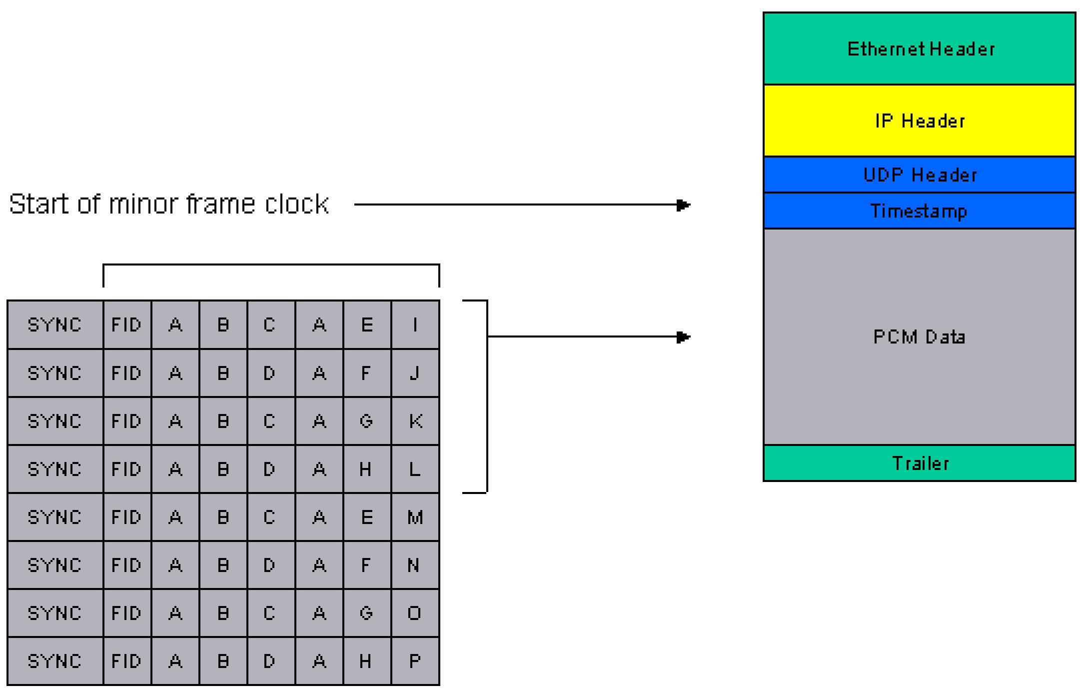
Packetization of Data

Packetization is defined as the transformation of a continuous stream of data into discrete chunks of data. Packets of data by definition have well-defined starting points and ending points. The utility of breaking up information into discrete units is best illustrated through an example. When you pick up your phone and dial a number, a point-to-point connection is constructed by the phone company between your handset and the handset at the other end. This has traditionally been accomplished using a circuit-switched procedure. The phone company pieces together a physical dedicated data path through the voice switches between the two endpoints and guarantees a 56 kilobit/sec PCM voice channel to carry information between the two parties. In many ways, the traditional phone system mirrors many of the basic operating principals of a PCM-based data acquisition system. This technique made sense when the physical bandwidth of the transport layer was comparable to the bandwidth needed to carry the information content needed. However, the technology used to transport information has grown at a far greater rate than the needs of the content being carried. From an economic point of view, it no longer makes sense for the phone company to give each subscriber a dedicated circuit through their equipment when phone calls are made. The tremendous bandwidth available on a single fibre makes it possible to carry millions of phone calls on a single wire. Packetization makes it possible to efficiently share bandwidth on a single transport connection.

Packetization is a fundamental difference between PCM-based data acquisition systems and Network-based systems. Its impact on the architecture needs to be fully understood to appreciate the pros and cons in migrating this way. Packetization requires accumulation of data in order to create packets. Inherent in accumulation is latency. The fixed latency command reception and parameter transmission found in the CAIS bus is lost through packetization. Minimizing the latency by keeping the packet size very small is counterproductive as it results in creating many small packets which waste bandwidth due to header overhead. They also waste processor overhead because of the need to switch and route lots of small packets to the same physical location. Choosing the right compromise in terms of latency and efficiency is a critical decision in the design of a network-based data acquisition system.

In order to illustrate some of the previous points, let us investigate one possible embedding of PCM data into UDP packets. Suppose we define the following PCM format for data acquisition:

- 16 bits per word
- 6 words per minor frame
- 8 minor frames per major frame
- Sub-frame ID word
- 16-bit Sync word
- 1 megabit/sec sample rate



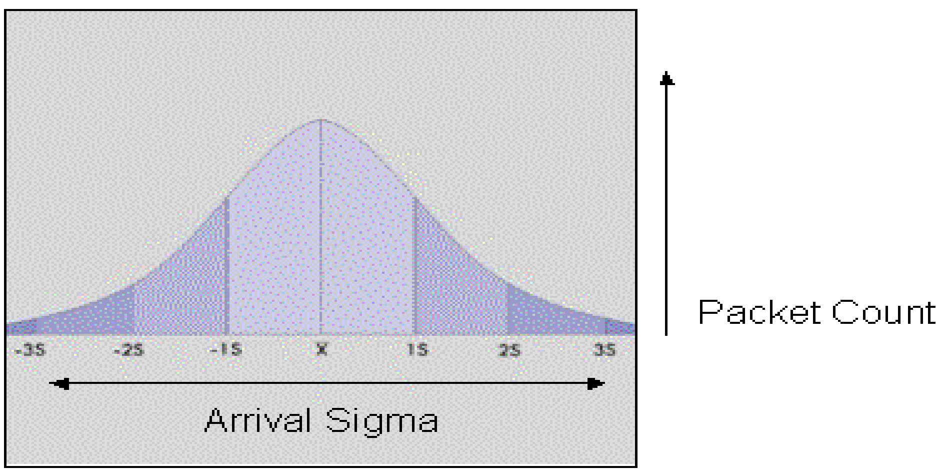
Let us choose to embed 4 minor frames into each UDP packet. That results in a latency of 448 microseconds to accumulate each packet. As part of this process, we would choose to ignore the sync word, as the packet headers already provide for data synchronization. The SFID word can be used to identify the minor frame block, and a single timestamp would be added to the beginning of each data payload to give us a timing reference point for all of the samples within the packet. This procedure results in 2232 packets per second, with a bandwidth overhead for the UDP protocol of about 100%. Each packet would require at least 448 microseconds to build; the actual transmission time (assuming 100 Mbit/sec link) would be about 9 microseconds.

Bi-Directional Packet Movement

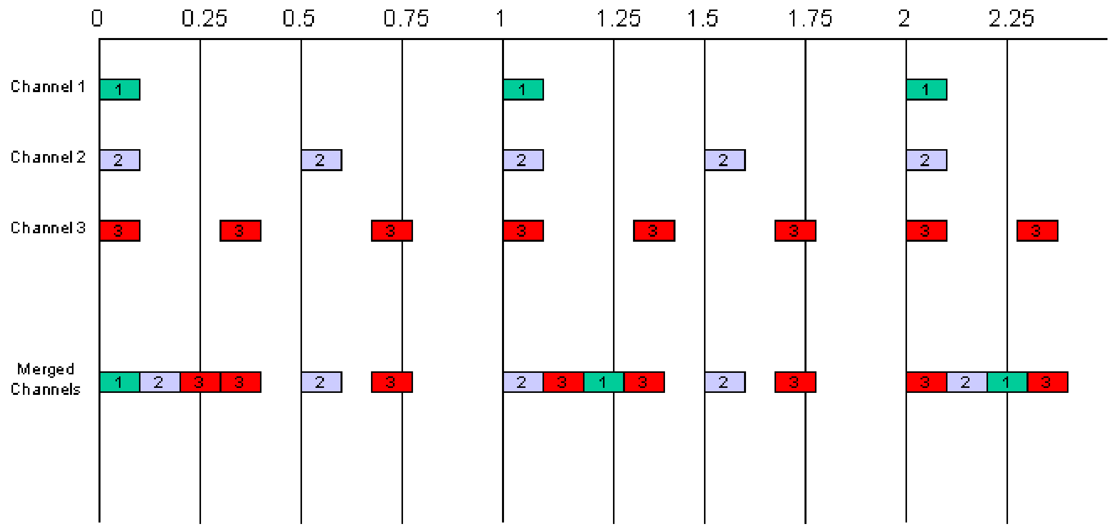
There is no inherent directionalism in a packet network. The media-access controller abstracts from a system the necessary signaling to indicate one particular node is a receiver or transmitter. Either half-duplex wiring with collision detection or a full-duplex link allow for a single node to both originate and terminate data transmissions. Thus the role that a single device plays in the data acquisition system is fluid, at any instance of time it can be originating acquired data into the network while at the next instance, be the target of an external command to readjust its internal programming.

Asynchronous Arrival of Data

While the sampling frequency of any particular parameter is predetermined, the packetization of a cluster of parameters and their traversal through the network result in an asynchronous arrival of packets at their destination. If we measured the arrival time of packets from a single source that collected its data in a synchronous fashion, we would find the packets spread out into a normal distribution centered on a mean value.



The primary cause of this phenomenon is due to the sharing of resources within the network infrastructure itself. Each packet from multiple sources may compete for a specific resource at different times, resulting in a random distribution of packets at different points in the network. For example, suppose we had 3 channels, one transmitting packets at 1 per second, one at 2 per second, and one at 3 per second. At aggregation points in the network, packets emitted at the same instance must be ordered for transmission, this ordering is done either thru collision detection, or thru insertion into a FIFO for transmission scheduling. In each case, the end result is a reordering of packets from instance to instance depending upon resource availability and statistical indeterminism. This behavior is illustrated below:



Dynamic Quality of Service Parameters

Almost all packet networks today carry some form of Internet protocol. IP packet flow is traditionally a best-effort process. Lost packets are retransmitted, packets are commonly reordered from their original transmission sequence, and any packet streams sensitive to timing suffer from possible congestion points within the network that are overloaded. Because data packets carry headers which not only define where they came from and where they going, but even how they are suppose to be sent; it is possible to define different levels of service with the network on a packet by packet basis. When these services include bandwidth, timing, or reliability, they are referred to as Quality of Service (QOS). Modern packet networks provide for dynamic control of quality of service parameters. While all acquired data is important, it may be that status data related to the operation of the data acquisition hardware has a lower priority than data being collected from a particular sensor. The network can be informed of this priority, such that when a sensor packet competes for the same resource as a status packet, the former packet will always get first usage.

Complete Intermixing of Control and Data

Packet networks have no need for physical separation of control and data information. The use of protocol stacks allows endpoints to differentiate bit streams that may contain either control or parameter content. MAC addresses allows for identification of a sender or receiver. TCP and UDP ports allow for separation of function. TCP retransmission allows for reliable delivery of control messages. UDP multicast provides for efficient use of bandwidth and delivery of the same data to multiple destinations. Any node within the network can control another and send/receive data.

Low Accuracy Global Timing Synchronization

Packet networks traditionally have not required a high degree of synchronization. Most systems implement an event-driven model of interaction. The common Internet applications such as FTP, Telnet, HTTP, web browsers, and the client/server model are all premised on an event-driven interaction. Recently, multimedia streaming has become an essential part of a packet network, but its primary concern is replicating stored time at a remote destination. It is primarily used as a one-to-many model, where stored audio/video is delivered to a remote destination in such a manner that its time sensitive content can be played correctly. It requires no coordination between destinations themselves, each viewer sees a synchronized presentation that remains unsynchronized with a different viewer seeing the same content at the same time. In addition, since the primary goal is to present information to a human, the synchronization accuracy required is on the order of milliseconds. The primary culprit here is flexible switching, it provides robustness in the event of failure and efficient use of shared resources, but complicates the ability to achieve tight synchronization between multiple nodes on the packet network.

Distribution of Time

Inherently, while packet networks offer many advantages over a traditional PCM architecture, they suffer from at least one significant shortcoming: timing accuracy. Distribution of highly accurate time within a packet network has been difficult to accomplish to date. Some of the possible approaches to achieving time synchronization in a packet network include those listed in the following table:

	SNTP	GPS	IRIG	IEEE 1588
Application Area	Global	Global	10s of Feet	100s of Feet
Communication	Internet	Satellite	Cabling	LAN
Accuracy	Few ms	< us	Few us	< us
Administration	Configured	N/A	Configured	Self
Special Hardware	No	Receiver	FPGA	FPGA

IEEE 1588

Of all the approaches described above, IEEE 1588 seems to hold the most promise in the near term. Version 1 of the standard was approved in 2002 and it has been demonstrated to be able to achieve accuracies in the range of 100-300 nanoseconds across real-world local area networks

Conclusions

A PCM-based data acquisition methodology is fundamentally different than any Network-based data acquisition methodology. The reasons for this have been discussed throughout this paper. Some of the differences make a network-centric data acquisition architecture attractive as a migration path for future avionics acquisition systems. However, in order for a network-centric architecture to succeed, communication of timing must be an intrinsic function of behavior as is communication of data. Any proposed system implementation will need to follow an evolutionary approach rather than a revolutionary one, as the need to maintain some form of compatibility with existing test infrastructure is required. Introduction of networking technology into flight test instrumentation opens the door for leveraging existing commercial technology, but in most cases, products will need to be ruggedized for most application scenarios.