IEEE1588 – A solution for synchronization of networked data acquisition systems?

Diarmuid Corry, M. Eng
ACRA CONTROL INC

ABSTRACT

One of the problems for manufacturers and users of flight test data acquisition equipment, is to guarantee synchronization between multiple units acquiring data on the vehicle. Past solutions have involved proprietary interconnects and multiple wire installations increasing weight and complexity and reducing inter-operation of units. This problem has become particularly important given the trend towards commercial busses, especially Ethernet, as a system interconnect.

The IEEE1588 standard offers a way to transmitting time accurately over Ethernet. This paper discusses the standard, how it might be implemented, and examines the issues involved in adopting this standard for flight test data acquisition. A particular implementation that results in a synchronized four-wire Ethernet based distributed data acquisition system is discussed in section 3.

KEYWORDS

Time Synchronization, Data Acquisition, PTP, Ethernet, Synchronous sampling

1 INTRODUCTION

In a synchronized data acquisition system all data sampling occurs at a known time relative to absolute time, or to some external or internal known event. (Modern data acquisition systems go a step further and provide **isochronous** sampling, meaning that all data sampling occurs at the **same** time and concurrent with absolute time or some external or internal known event). Synchronously sampled data is easier to analyze. The need to link data to absolute time becomes important when we are comparing data sets from multiple independent systems, or even multiple test articles.

To achieve synchronous sampling the data acquisition system(s) need two things:

1. A common concept of what the time is
2. A way of synchronizing operations either to each other or to this common time

In the past, various approaches have been taken to solving this problem. Most data acquisition units (DAU) in flight test have some way of capturing and holding IRIG time. This is the most common way of creating a common concept of time (although sometimes it involves physically connecting an IRIG time source to each system individually). In addition, there is some way of synchronizing the operation of the multiple systems through a proprietary method. Although effective, these solutions add wiring with some systems requiring 10-wires between individual DAUs.

Two trends in flight test instrumentation have started to put pressure on these proprietary solutions:

1. The need to reduce wiring for weight and complexity purposes – especially with the large systems that are becoming more common today.

2. The move towards commercial solutions for data transfer, e.g. Ethernet, Firewire and Fibre-Channel

A solution would provide accurate time (better than 1μs) distributed over commercial data networks without the need for additional wiring or proprietary protocols.

In late 2002 the IEEE standards committee approved IEEE1588 as a protocol for distributing time over a local area network with high accuracy[1]. Although targeted at Ethernet, IEEE1588 is a high level protocol that is physical layer independent. It offers a potential solution for the issue of managing time distribution over synchronized DAUs. The rest of this paper introduces this standard and discusses how it is applied in the flight test instrumentation environment.

2 AN OVERVIEW OF IEEE1588

2.1 What it is

IEEE1588 is formally titled: "Precision clock synchronization protocol for networked measurement and control systems" [1]. It describes a high level protocol (also referred to as Precision Time Protocol or PTP – see Figure 1) that permits one node on a network to become a time master (called a Grand master clock), and describes how that node can then distribute time accurately to all other nodes. The protocol addresses transmission delay issues and transmission of time through network switches.

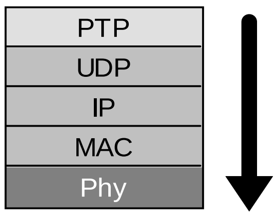

Figure 1: Protocol stack for IEEE1588

2.2 How it works

2.2.1 Overview

The basic concept is that each node on the network has a local clock, and the time of these clocks is synchronized to the network grand-master. There are three phases to this operation:

1. All nodes identify and designate the grand-master clock

2. Grand master distributes its current time to all nodes

3. Nodes calculate the transmission delay between themselves and the grand-master and adjust accordingly

---

[1] A revised standard was issued in 2004. The next revision is due no sooner than 2007.

These operations are repeated frequently – in a time known as the Sync Interval. The Sync Interval is administratively chosen to provide sufficient frequency to maintain accurate enough time, but not so frequently as to represent a significant overhead on the bus traffic. The Sync interval is taken from the set 1, 2, 8, 16 and 64 seconds

2.2.2 Management

The management parts of the protocol control configuration of the clock parameters (sync interval etc.) and designation of the master clock. There is nothing inherently “special” about the master clock – but an algorithm is executed to ensure that the “best” clock in the domain is assigned the master.

Clocks use fields in the sync messages to determine who is most accurate. The management protocol executes frequently so changes in the network topology and/or clock status (e.g. the time source loses visibility of GPS satellites and starts free-wheeling, with possibly less accuracy then when locked to GPS) can be detected and adapted for.

2.2.3 Timing

The main point of the protocol is to distribute time. The grand master broadcasts two messages to all nodes:

1. “This is the time I think I’m sending this message” – the SYNC message.
2. “This is the time I actually sent the message” – the FOLLOW_UP message.

The FOLLOW_UP message is used to account for any delay that might occur in the protocol stack during transmission and is sent only if the master clock implementation warrants it. The SYNC message is sent at regular intervals – the sync interval.

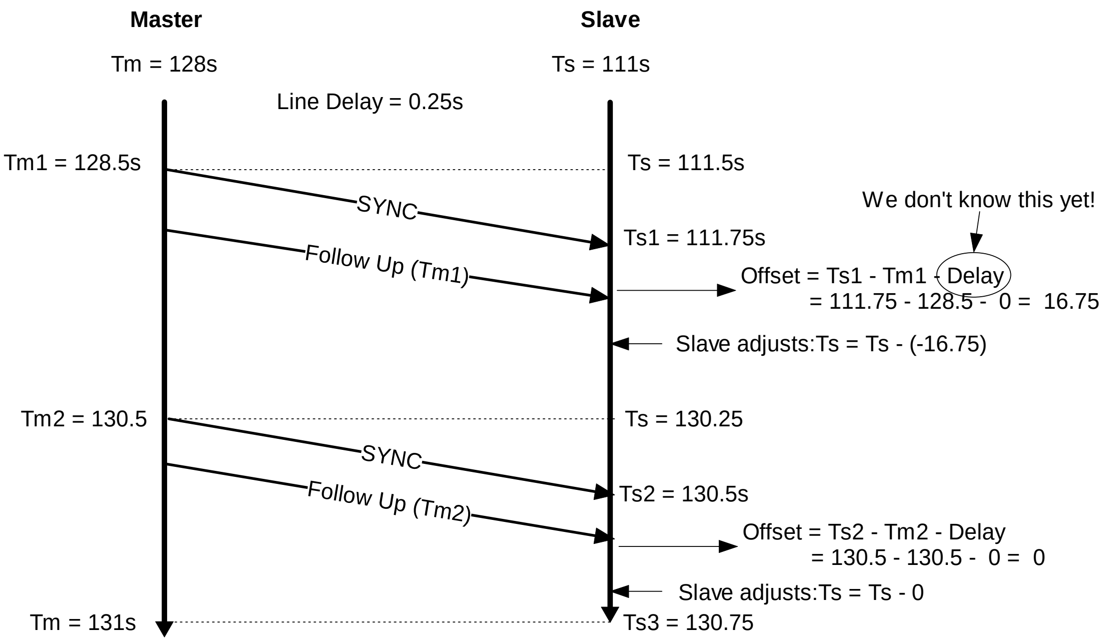

Figure 2: SYNC message operation

Figure 2 shows the operation of the SYNC and FOLLOW_UP messages. Clearly, the slave can easily synchronize to the master time with a constant error due to the transmission delay of the communication path. Also, any drift during the sync interval is corrected with the next occurrence of the SYNC message.

An additional element of the protocol permits each node to adjust its local time for transmission delays. The transmission delay is determined by exchanging messages with the master as follows:

1. Slave: "I am requesting the delay at this time" – the DELAY_REQUEST message.
2. Master: "I got your delay request at this time" – the DELAY_RESPONSE message.

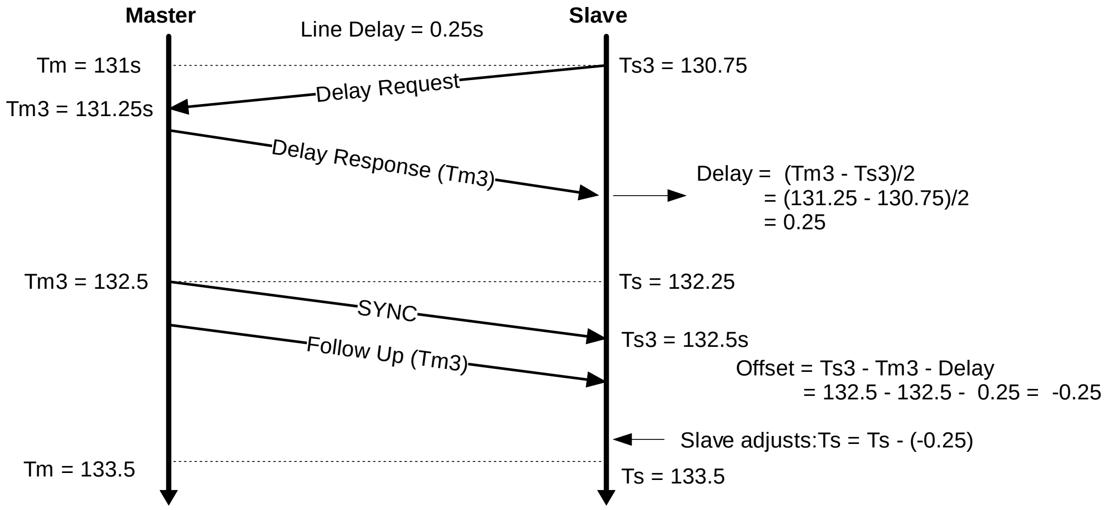

Figure 3: Delay calculation

Figure 3 shows how the delay request message is used to correct for the transmission delay. Delay request messages are sent at random intervals, but at least once every 64 seconds, to continuously adjust the delay.

2.3 Assumptions

The IEEE1588 standard makes several assumptions about the network being used (that it supports multi-cast for example) but the key assumptions that affect accuracy are:

1. Transmission delays are constant over time (or at least change slowly)
2. Transmission delays are symmetrical (i.e. time to travel from master to slave is the same as from slave to master).

Neither assumption is necessarily true in a general network. However, it must be remembered that the scope of IEEE1588 is for local networks, and indeed the standard was originally targeted at command and control networks used for data acquisition. Within these networks it is usually possible to control these parameters to acceptable levels.

The IEEE1588 protocol will adjust for consistent delays in the transmission network, any remaining error is due to jitter. Jitter comes from fluctuations in transmission time due to protocol stack fluctuations, switch buffering and so on. Jitter tends to increase (in both time and variation) as network load increases.

The next section looks at some ways these errors can be reduced in an "IEEE1588 enabled" network.

2.4 An IEEE enabled network

2.4.1 Minimizing Network Delays

An IEEE1588 enabled network is a network where key elements of the network infrastructure are IEEE1588 aware – primarily the switches. Standard switches contain buffers and sometimes some intelligence to implement prioritization and so on. Even with minimal network traffic, if a PTP message arrives at a switch just after it has begun to transmit a packet, it must wait until that packet is transmitted (theoretically, this could be as long as 122µs – the time taken to transmit a MAC packet). In cases of heavy load the delay can run into 10s of milliseconds due to buffering.

An IEEE1588 switch isolates its internal delays by operating as a boundary clock. A boundary clock acts as a slave clock at the port that connects to the grand master, and as a master to all other ports. It thus isolates the "down stream" clocks from any delays and jitter in the switch itself.

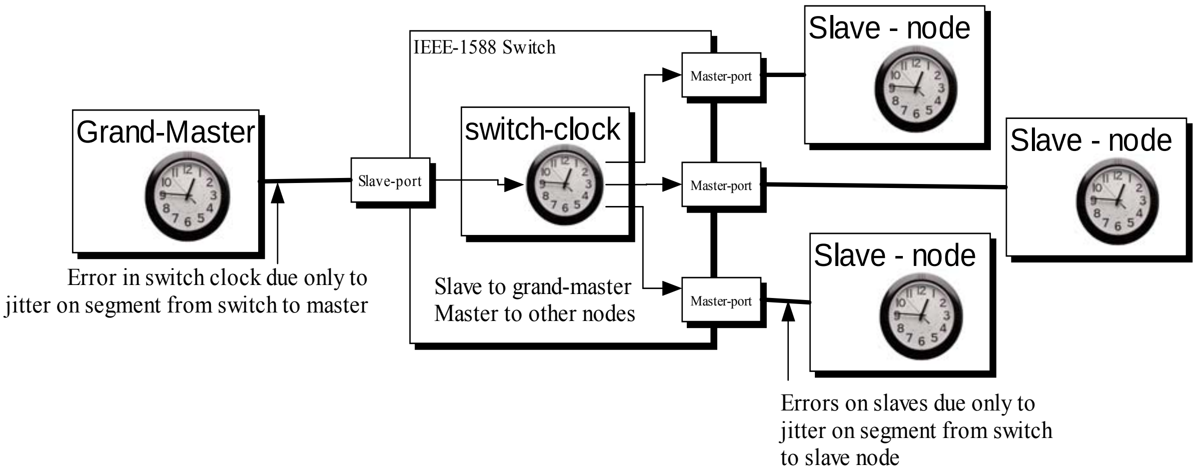

Figure 4: Boundary clocks in an IEEE1588 Network

An alternative, and simpler, implementation is an IEEE1588 "transparent" switch [2]. In this implementation, the switch modifies the time stamps in the time packets to adjust for delays introduced by the switch itself. This is a simpler and cheaper solution as the switch does not have to participate fully in PTP to implement it.

Apart from switch jitter, variations in transmission time can arise due to fluctuations in the protocol stack traversal. These can be minimized by placing the time-stamp mechanism as close to the physical layer (the PHY) as possible. Another factor is the number of nodes between the clock and its grand master – the less nodes the better.

While it is possible to implement PTP on any network, greater accuracy can be achieved with an IEEE1588 enabled infrastructure.

2.5 Achievable accuracy

Ultimately, the accuracy within which any two clocks can be maintained depends on many elements:

- inherent stability of the local clock
- sync interval
- whether network infrastructure is IEEE1588 aware
- complexity of network infrastructure
- network load
- symmetry of network load
- temperature variance
- Hardware implementation (packets time-stamped by hardware very close to the PHY) or software implementation (packets time-stamped in software interrupt routines)

Interpreting published results is difficult due to this large number of influencing factors.

Examples of results reported in the field are:

Topology	**Time-stamping**	**Traffic**	**Error**	**Std. Deviation**	**Source**
Point-to-point	hardware implementation	high	-4.248ns	23.95ns	[3]
single switch	hardware implementation	low	-174ns	500ns	[4]
single switch	hardware implementation	low	-49ns.	99ns	[5]
single hub	software implementation	low	-630ns.	2µs	[4]

Other results show that using boundary clocks or transparent switches can improve the performance to a standard deviation of less than 50ns. [6]

These results show that there is no straight answer to the question of how well accuracy can be maintained – but that we can safely say that accuracies better than 1µs can be achieved with a hardware implementation of the time-stamping.

2.6 Enhancements

Certain steps can be taken to try to improve performance in particular circumstances.

To try to compensate for jitter in the transmission path, the measured delays could be averaged over time. This improves accuracy at the cost of responsiveness – should the network dynamics or topology change it would take some time for the clocks to adjust to this.

Asymmetric loads introduce an important error into the system. This is a common scenario in flight test instrumentation networks where the data flow tends to be mostly one-way (from a DAU to a recorder). This introduces asymmetric delays into the PTP message transfers and results in a constant error offset [4]. This can be removed by characterizing it before application and adjusting the time accordingly.

3 AN IEEE ENABLED DATA ACQUISITION UNIT

3.1 Controller unit

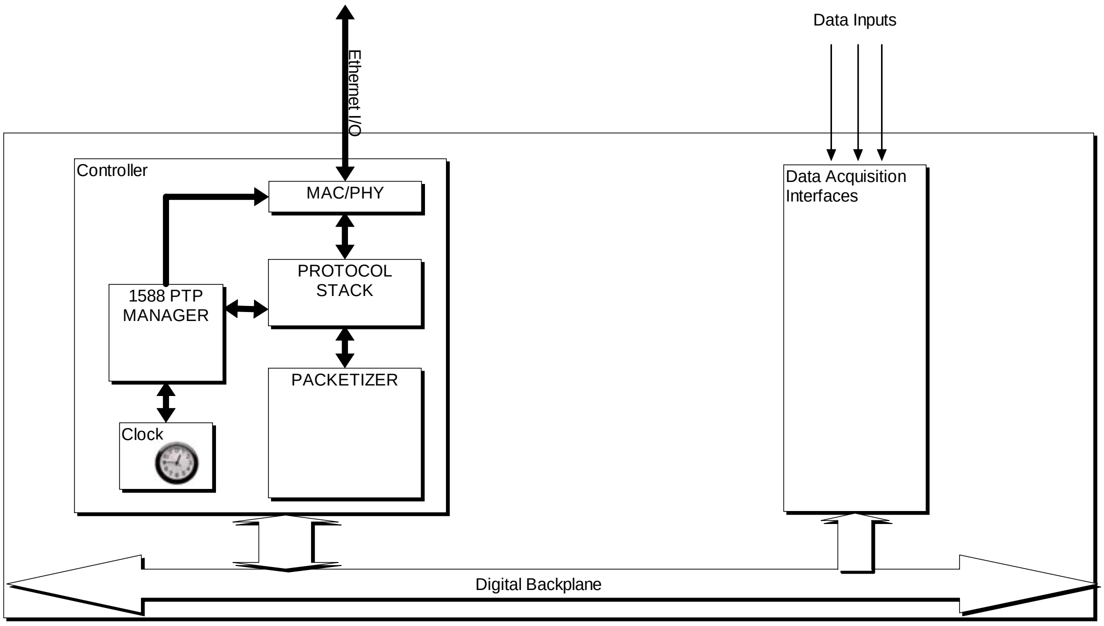

Figure 5: Block Diagram of IEEE1588 enabled digital data acquisition system

Figure 5 shows how the KAM-500 Data Acquisition System from ACRA CONTROL acts as an IEEE1588 enabled data acquisition system. In this implementation, the core building block is a modular chassis with a digital back-plane. Data acquisition modules for various data sources (sensors, busses etc.) can be added to the chassis for a particular applications needs. Traditionally, the chassis contained at least one internal clock maintaining time to 1µs resolution. All sampling within the chassis is isochronous and managed by the chassis controller over the backplane. Time-stamping of recorded and/or transmitted data, as well as of any asynchronous bus traffic that is acquired, is carried out with reference to the internal clock and is aligned to the second with micro-second resolution. Therefore, if the controller time in two chassis is synchronized, then the sampling of those two chassis is also synchronized.

Adapting this data acquisition unit to operate in an IEEE1588 world involved adding a block to manage the IEEE1588 protocol to the controller. The controller already had an Ethernet interface for programming and transmission of data. The PTP management block is implemented in hardware using FPGAs and message time-stamping is applied at the PHY layer. These result in minimal protocol stack jitter and improved accuracy.

With no other elements in the network, just KAM-500 chassis and a switch, this controller can be used to build a synchronized distributed data acquisition system using just four-wires to interconnect units. In this implementation (without an external time reference) the IEEE1588 interface acts just as a way of synchronizing time across the chassis.

3.2 GPS bridge module

In practical applications we need a way of aligning the internal clock of the chassis with absolute time. Traditionally this was done using an IRIG time input. With this implementation we also have the option of using a grand master somewhere on the network that is synchronized to IRIG or GPS.

A cost-effective and powerful way to do this is to install a GPS time-card directly into the chassis (Figure 6). The GPS time card synchronizes to GPS time (or IRIG), and locks the controller clock to this. Now the chassis can act as a grand master on the data acquisition network with time locked to GPS time. In the event of GPS lock being lost the time module will free run with a drift of less than 3ppm. In addition, the time module can generate IRIG-B time for legacy systems that need it.

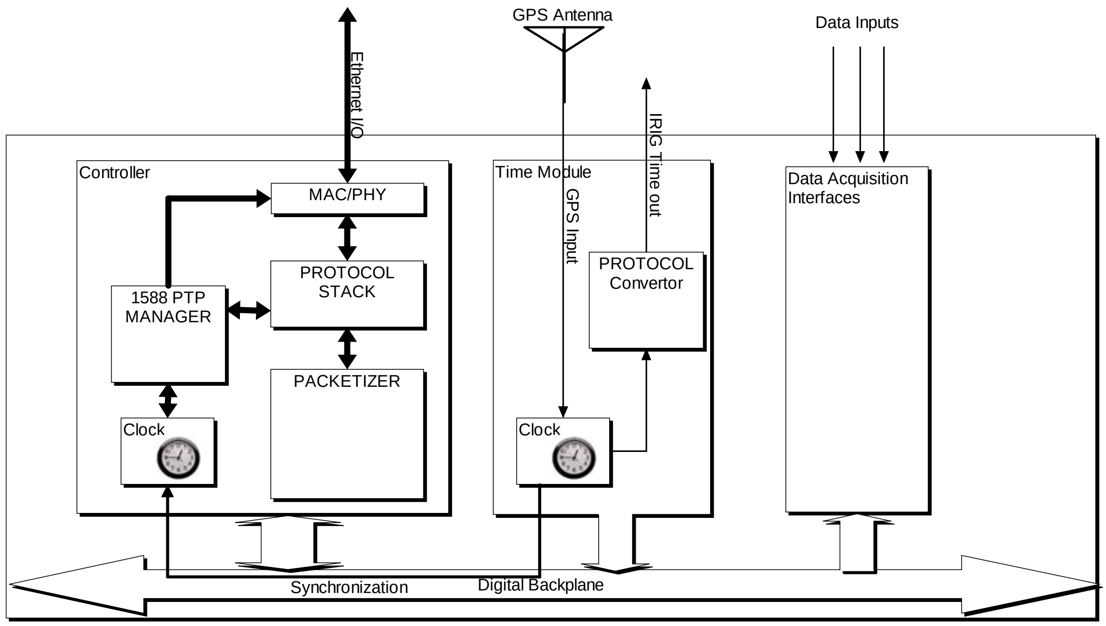

Figure 6: IEEE1588 DAU with GPS

An interesting an important side effect of using the GPS time module is that data acquisition is synchronized not only to itself and across all DAUs on the network, but it is also synchronized to absolute time. This permits data across multiple test articles to be synchronized as long as each one has a GPS receiver on board. [7]

3.3 Topologies

3.3.1 Flight Data Acquisition Networks

Flight test data acquisition networks have some peculiarities that both support and detract from the use of IEEE1588 as a synchronization mechanism. This section discusses generalities and of course any given application may differ widely. However, in general, flight data acquisition networks have the following characteristics:

- Stable topology. The topology is known and unlikely to change. In some case a dual redundant architecture based on ARINC-664 is used. This benefits PTP since it means that message transfer times do not change over time and are consistent.

- Known network load: In general the maximum load on the network is known in advance. In normal data acquisition mode the network dynamics are well known and well understood, and typically is cyclic. This also benefits PTP since it reduces the chances of fluctuations in the delay response message times.

- Asymmetric data transfer: In general, DAUs transmit a lot of data and receive little or none. Data flows within the network are asymmetric. This is a problem for PTP as it assumes symmetry. However, tying in with the point above, the network load is generally known and it is possible to characterize and compensate for the errors introduced by asymmetric data flows.

- Small size: By network standards even a large aircraft has a pretty small network. This makes it feasible to ensure that each node has only one or at most two switches between it and the grand master reducing the effect of switch jitter.

- Environment: The environment for a flight data acquisition network is harsh. This presents two types of problems for PTP. Firstly, IEEE1588 enabled switches are not very common at this time and ruggedized versions are very limited. Secondly, different elements of the network are exposed to very different temperatures. This means that good quality clocks are required to ensure that drift between time updates is acceptable.

3.3.2 Simple System

In its simplest form, we can use IEEE1588 as a way of synchronizing DAUs in a distributed data acquisition system. With no switches, we assign one unit to be the grand master (optionally with IRIG or GPS time input) and use it to set the other controller clocks. This gives us a four-wire distributed unit with up to 64Mbps data throughput.

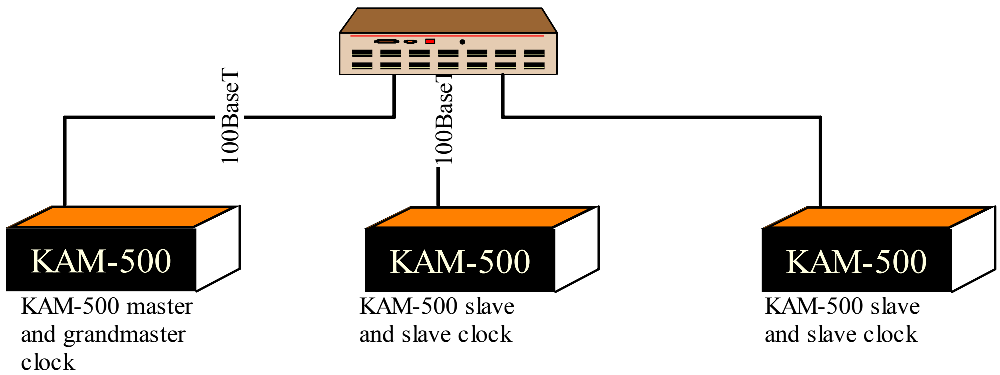

Figure 7: Simple distributed system

3.3.3 Star network

As the size and number of channels increases, and we need to interconnect various types of device, the network becomes more complex. However, we can still maintain synchronization. With more complex networks it may be necessary to optimize the network topology to reduce the number of nodes between any slave and the grand master and/or to use IEEE1588 enabled or transparent switches (Figure 8).

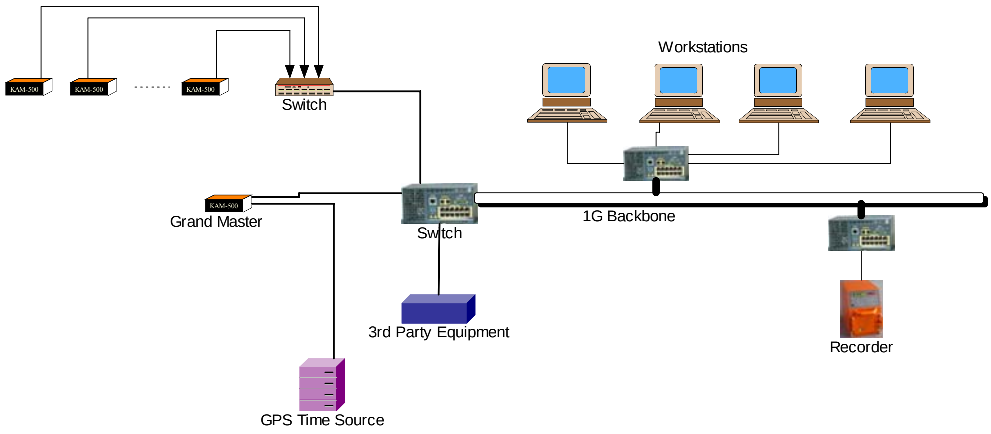

Figure 8: Complex network with extra elements

4 CONCLUSION

IEEE1588 is a strong candidate for a solution for the problem of synchronizing data acquisition systems in flight test – both within a test article and across multiple test articles. Further work is needed to characterize the inaccuracies that will exist across networks used in flight test. Particular attention must be paid to errors introduced by the peculiarities of flight test networks.

Introducing PTP into existing modular data acquisition designs is straightforward and details of an implementation given show that legacy data acquisition system can easily be upgraded to adopt the new technology, enabling networked data acquisition without obsoleting existing inventory.

REFERENCES

[1] IEEE Standards Committee, "Precision clock synchronization protocol for networked measurement and control systems", IEEE Std. 1588, 2004

[2] Sven Nylund and Øyvind Holmeide, "IEEE1588 Switch Transparency", Proceedings of NIST Conference on IEEE1588, Gaithersburg, Maryland, 27-29 Sept, 2004

[3] Dirk S. Mohl, "IEEE 1588 - Precise Time Synchronization as the Basis for Real Time Applications in Automation", http://www.industrialnetworking.com/support/general_faqs_info/Precise_Time_Sync.pdf

[4] Hans Weibel and Dominic Béchaz, "IEEE1588 – Implementation and Performance of Time Stamping Techniques", Proceedings of NIST Conference on IEEE1588, Gaithersburg, Maryland, 27-29 Sept, 2004

[5] John Eidson and Kang Lee, "Sharing a common sense of time", IEEE Instrumentation and Measurement Magazine, March 2003

[6] Symmetricom, "IEEE 1588 Precise Time Protocol: The new standard in time synchronization", White paper, www.symmetricom.com

[7] iNET Program, "iNET Needs Discernment Report", May 2004, Appendix 8.Scenario 34