ADDRESSING THE CHALLENGES CREATED BY LARGE NETWORKED ETHERNET FTI

Pat Quinn[1]
[1] **Program Manager,**
Curtiss-Wright Defense Solutions, Avionics & Electronics,
Dublin, Ireland,
pquinn@curtisswright.com

ABSTRACT

As Flight Test Instrumentation (FTI) systems move away from traditional PCM towards Ethernet [1], a whole new set of system level considerations must be taken into account. This is particularly true when these systems consist of dozens of data acquisition systems (DAUs) and multiple layers of switches. This paper discusses the challenges presented by very large Ethernet based systems and the methodologies developed to address these during a recent application.

KEY WORDS:

INTRODUCTION

For some flight test programs, the move to a fully networked data acquisition system may seem daunting. However, given the sheer volume of data required to be captured and managed during flight test programs, the move to Ethernet based systems has become a reality.

While traditional PCM systems have over the years thrown up many challenges, network centric systems throw up their own set of unique challenges.

This paper attempts to discuss some of the issues encountered and how they were addressed when dealing with a very large networked system.

THE SYSTEM

The Flight Test System under discussion in this paper is a 45 chassis, redundant Ethernet system, with 3 layers of switches. Data from all 45 chassis and external Ethernet traffic needed to be transmitted across both networks and redundantly recorded, while simultaneously extracting a subset of the parameters out of the network traffic for inclusion in a coherent PCM telemetry stream.

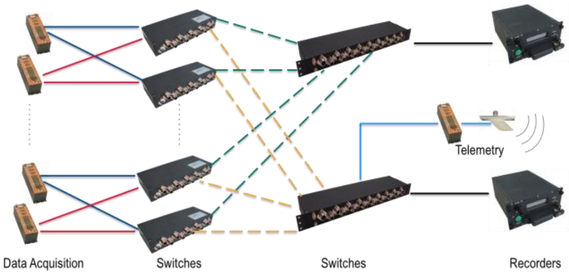
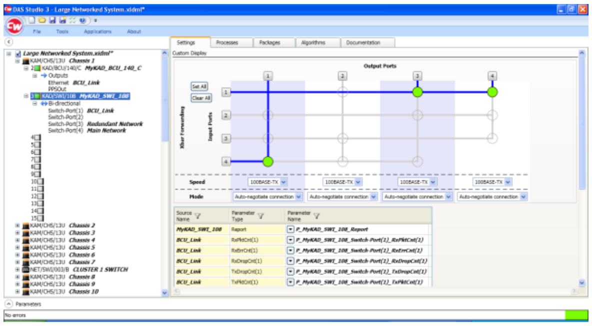


Figure 1: An outline of the system

The System in Figure 1 contains two mirror Ethernet networks, each of which has 3 layers of switches and is synchronized via the IEEE 1588 Precision Time Protocol (PTP). The first Switch Layer is contained in a single module sized, fully configurable, crossbar switch which takes the output from the controller in each chassis and mirrors it to two redundant networks.

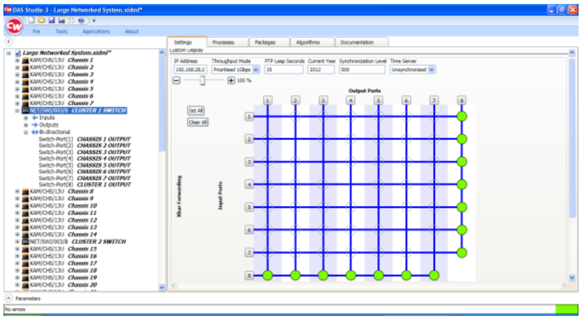


Large Networked System.xidml*
KAM/CHS/13U Chassis 1
2 KAD/BCU/140/C MyKAD_BCU_140_C
Outputs
Ethernet BCU_Link
PPSOut
3 KAD/SWI/108 MyKAD_SWI_108
Bi-directional
Switch-Port(1) BCU_Link
Switch-Port(2)
Switch-Port(3) Redundant Network
Switch-Port(4) Main Network
KAM/CHS/13U Chassis 2
KAM/CHS/13U Chassis 3
KAM/CHS/13U Chassis 4
KAM/CHS/13U Chassis 5
KAM/CHS/13U Chassis 6
KAM/CHS/13U Chassis 7
NET/SWI/003/B CLUSTER 1 SWITCH
KAM/CHS/13U Chassis 8
KAM/CHS/13U Chassis 9
KAM/CHS/13U Chassis 10

	1	2	3	4
Speed	100BASE-TX	100BASE-TX	100BASE-TX	100BASE-TX
Mode	Auto-negotiate connection	Auto-negotiate connection	Auto-negotiate connection	Auto-negotiate connection

Source Name	Parameter Type	Parameter Name
MyKAD_SWI_108	Report	P_MyKAD_SWI_108_Report
BCU_Link	RxPktCnt(1)	P_MyKAD_SWI_108_Switch-Port(1)_RxPktCnt(1)
BCU_Link	RxErrCnt(1)	P_MyKAD_SWI_108_Switch-Port(1)_RxErrCnt(1)
BCU_Link	RxDropCnt(1)	P_MyKAD_SWI_108_Switch-Port(1)_RxDropCnt(1)
BCU_Link	TxDropCnt(1)	P_MyKAD_SWI_108_Switch-Port(1)_TxDropCnt(1)
BCU_Link	TxPktCnt(1)	P_MyKAD_SWI_108_Switch-Port(1)_TxPktCnt(1)

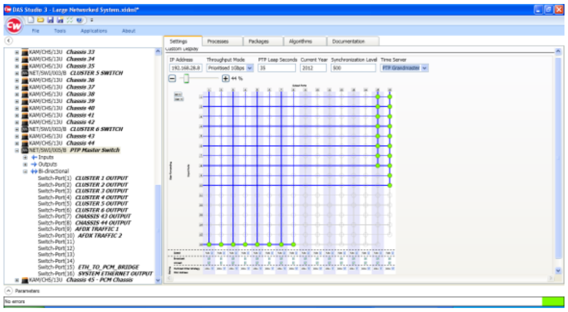
The second layer is a cluster level switch, where clusters of seven chassis are aggregated to a single 1000 Base T output, which is routed to the final switch layer. The switch used for this layer is an eight port, fully configurable, crossbar switch.



Large Networked System.xidml*
KAM/CHS/13U Chassis 1
KAM/CHS/13U Chassis 2
KAM/CHS/13U Chassis 3
KAM/CHS/13U Chassis 4
KAM/CHS/13U Chassis 5
KAM/CHS/13U Chassis 6
KAM/CHS/13U Chassis 7
NET/SWI/003/B CLUSTER 1 SWITCH
Inputs
Outputs
Bi-directional
Switch-Port(1) CHASSIS 1 OUTPUT
Switch-Port(2) CHASSIS 2 OUTPUT
Switch-Port(3) CHASSIS 3 OUTPUT
Switch-Port(4) CHASSIS 4 OUTPUT
Switch-Port(5) CHASSIS 5 OUTPUT
Switch-Port(6) CHASSIS 6 OUTPUT
Switch-Port(7) CHASSIS 7 OUTPUT
Switch-Port(8) CLUSTER 1 OUTPUT
KAM/CHS/13U Chassis 8
KAM/CHS/13U Chassis 9
KAM/CHS/13U Chassis 10
KAM/CHS/13U Chassis 11
KAM/CHS/13U Chassis 12
KAM/CHS/13U Chassis 13
KAM/CHS/13U Chassis 14
NET/SWI/003/B CLUSTER 2 SWITCH
KAM/CHS/13U Chassis 15
KAM/CHS/13U Chassis 16
KAM/CHS/13U Chassis 17
KAM/CHS/13U Chassis 18
KAM/CHS/13U Chassis 19
KAM/CHS/13U Chassis 20

IP Address	Throughput Mode	PTP Leap Seconds	Current Year	Synchronization Level	Time Server
192.168.28.2	Prioritised 1Gbps	35	2012	500	Unsynchronized

This final layer is the PTP Grandmaster level. Data from all sources is fed into this switch, where the data is managed and filtered such that all data goes to the redundant network recorders and a subset of the data is directed at the PCM transmission chassis for transmission to the ground station.



KAM/CHS/13U Chassis 33
KAM/CHS/13U Chassis 34
KAM/CHS/13U Chassis 35
NET/SWI/003/B CLUSTER 5 SWITCH
KAM/CHS/13U Chassis 36
KAM/CHS/13U Chassis 37
KAM/CHS/13U Chassis 38
KAM/CHS/13U Chassis 39
KAM/CHS/13U Chassis 40
KAM/CHS/13U Chassis 41
KAM/CHS/13U Chassis 42
NET/SWI/003/B CLUSTER 6 SWITCH
KAM/CHS/13U Chassis 43
KAM/CHS/13U Chassis 44
NET/SWI/005/B PTP Master Switch
Inputs
Outputs
Bi-directional
Switch-Port(1) CLUSTER 1 OUTPUT
Switch-Port(2) CLUSTER 2 OUTPUT
Switch-Port(3) CLUSTER 3 OUTPUT
Switch-Port(4) CLUSTER 4 OUTPUT
Switch-Port(5) CLUSTER 5 OUTPUT
Switch-Port(6) CLUSTER 6 OUTPUT
Switch-Port(7) CHASSIS 43 OUTPUT
Switch-Port(8) CHASSIS 44 OUTPUT
Switch-Port(9) AFDX TRAFFIC 1
Switch-Port(10) AFDX TRAFFIC 2
Switch-Port(11)
Switch-Port(12)
Switch-Port(13)
Switch-Port(14)
Switch-Port(15) ETH_TO_PCM_BRIDGE
Switch-Port(16) SYSTEM ETHERNET OUTPUT
KAM/CHS/13U Chassis 45 - PCM Chassis

IP Address: 192.168.28.8
Throughput Mode: Prioritised 1Gbps
PTP Leap Seconds: 35
Current Year: 2012
Synchronization Level: 500
Time Server: PTP Grandmaster

No errors

The data being generated by the above system contains a wide variety of traffic types, including over 1000 strain channels, over 1200 temperature channels, nearly 900 ICP channels and over 260 avionic bus channels of various protocols, including over 240 ARINC 429 channels.

CHALLENGES

Compiling Tasks

Before any data can be gathered by any system, regardless of size, the elements in the system must be told what is expected of them. This is achieved by creating a Task file, or XidML [3] file that describes:

- All elements in the system
- How they are connected
- What settings are applied to each element in the system
- What sample rates are required for the various parameters
- How data from all the chassis in the system is transported across the network to appear in the PCM frame for telemetry to the ground station.

To address these issues Curtiss-Wright's DAS Studio software has been developed, from the ground up, with ease of system configuration in mind. It features a number of tools to simplify and speed up configuration including;

- Hardware discovery
- Reading calibration data off the modules in the system
- Wizards to allow the user to instantly create hundreds of MIL-STD-1553, ARINC 429, CANbus (and more…) parameters
- Import for user-defined CSV files that contain definitions of 1000s of other avionic bus parameters
- Automated generation of all your outgoing packages, PCM and Ethernet, at user-defined sampling rates at the click of a single button

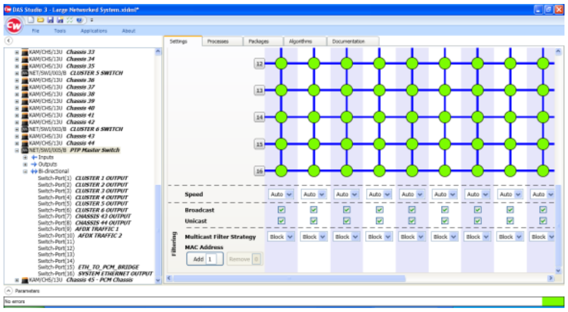
With the current DAS Studio software, a task describing such a system can be created in less than an hour and the binaries compiled, ready to program the system in a matter of minutes.

Programming the System

Once the system has been defined and the binaries created, the next challenge is to program the system. Unlike PCM systems, where there are dedicated programming links to every chassis in the system, Ethernet systems are programmed, synchronized and transmitted across the same wires.

The first step in programming the system is to clear the network of all traffic so that there is a clear path to all DAUs from the programming PC. Curtiss-Wright has developed a two-step process that is built into our hardware and software to ensure that a traffic-free network is created and maintained for the duration of the programming cycle.

First, the switches in the system are put in programming mode; an SNMP command is sent to the switches to block all multi-cast traffic out of every port and force a programming mode routing so that all ports can talk to all ports. This clears the path for the programming PC to talk freely to any DAU connected to any port to start the programming cycle.



KAM/CHS/13U Chassis 33
KAM/CHS/13U Chassis 34
KAM/CHS/13U Chassis 35
NET/SWI/003/B CLUSTER 5 SWITCH
KAM/CHS/13U Chassis 36
KAM/CHS/13U Chassis 37
KAM/CHS/13U Chassis 38
KAM/CHS/13U Chassis 39
KAM/CHS/13U Chassis 40
KAM/CHS/13U Chassis 41
KAM/CHS/13U Chassis 42
NET/SWI/003/B CLUSTER 6 SWITCH
KAM/CHS/13U Chassis 43
KAM/CHS/13U Chassis 44
NET/SWI/005/B PTP Master Switch
Inputs
Outputs
Bi-directional
Switch-Port(1) CLUSTER 1 OUTPUT
Switch-Port(2) CLUSTER 2 OUTPUT
Switch-Port(3) CLUSTER 3 OUTPUT
Switch-Port(4) CLUSTER 4 OUTPUT
Switch-Port(5) CLUSTER 5 OUTPUT
Switch-Port(6) CLUSTER 6 OUTPUT
Switch-Port(7) CHASSIS 43 OUTPUT
Switch-Port(8) CHASSIS 44 OUTPUT
Switch-Port(9) AFDX TRAFFIC 1
Switch-Port(10) AFDX TRAFFIC 2
Switch-Port(11)
Switch-Port(12)
Switch-Port(13)
Switch-Port(14)
Switch-Port(15) ETH_TO_PCM_BRIDGE
Switch-Port(16) SYSTEM ETHERNET OUTPUT
KAM/CHS/13U Chassis 45 - PCM Chassis

Speed	Auto	Auto	Auto	Auto	Auto	Auto	Auto	Auto	Auto
Broadcast	☑	☑	☑	☑	☑	☑	☑	☑	☑
Unicast	☑	☑	☑	☑	☑	☑	☑	☑	☑
Multicast Filter Strategy	Block	Block	Block	Block	Block	Block	Block	Block	Block
MAC Address									

No errors

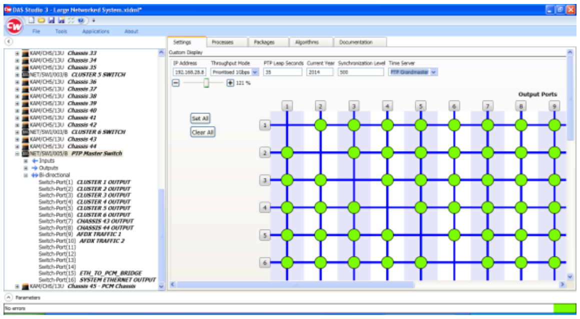
Second, the controllers in the chassis are placed in programming mode, forcing scheduled Ethernet packets to stop transmission. Once programming of a particular chassis is complete, the programming mode on both switches and chassis then times out and traffic flow recommences. To ensure that traffic flow from one programmed chassis does not interfere with the programming of another chassis, programming mode for both switches and chassis is maintained by the sending of a periodic “heart beat” packet that tells the chassis and switches to remain in programming mode until programming of all chassis is completed. Such an approach allows a system such as the one discussed in this paper to be programmed in less than 2 minutes.

Synchronizing the System

Ethernet based systems commonly use the IEEE 1588 Precision Time Protocol (PTP) [2] to ensure all chassis are synchronized to the same time base. This is an Ethernet based protocol whereby the path delay from any chassis to the grandmaster is constantly monitored and adjusted so all chassis are kept within around 100 ns of each other. Typical synchronization levels achieved in systems such as these are in the region of 80 – 130 ns time delta across the system. This is achieved through the exchange of delay requests and sync messages across the network, and it is therefore essential that all layers switches between the Grandmaster and the chassis controllers be PTP Transparent.

In the system under consideration, there are two redundant Ethernet networks and two grandmasters operating at the same time. The customer specified that the controllers in the chassis must only synchronize to one of the grandmasters. This was achieved in the four port switch module in each chassis which was programmed to let the traffic from the controllers be transmitted out of two ports but only one of those ports can talk back to the controller. This ensured that only PTP traffic from one side of the network was ever seen by the controller.

Having ensured that only one grandmaster was controlling the synchronization, we measured how fast the full system could be synchronized to <500ns delta from the PTP Grandmaster – this was found to be typically <45 seconds after power up.



KAM/CHS/13U Chassis 33
KAM/CHS/13U Chassis 34
KAM/CHS/13U Chassis 35
NET/SWI/003/B CLUSTER 5 SWITCH
KAM/CHS/13U Chassis 36
KAM/CHS/13U Chassis 37
KAM/CHS/13U Chassis 38
KAM/CHS/13U Chassis 39
KAM/CHS/13U Chassis 40
KAM/CHS/13U Chassis 41
KAM/CHS/13U Chassis 42
NET/SWI/003/B CLUSTER 6 SWITCH
KAM/CHS/13U Chassis 43
KAM/CHS/13U Chassis 44
NET/SWI/005/B PTP Master Switch
  Inputs
  Outputs
  Bi-directional
    Switch-Port(1) CLUSTER 1 OUTPUT
    Switch-Port(2) CLUSTER 2 OUTPUT
    Switch-Port(3) CLUSTER 3 OUTPUT
    Switch-Port(4) CLUSTER 4 OUTPUT
    Switch-Port(5) CLUSTER 5 OUTPUT
    Switch-Port(6) CLUSTER 6 OUTPUT
    Switch-Port(7) CHASSIS 43 OUTPUT
    Switch-Port(8) CHASSIS 44 OUTPUT
    Switch-Port(9) AFDX TRAFFIC 1
    Switch-Port(10) AFDX TRAFFIC 2
    Switch-Port(11)
    Switch-Port(12)
    Switch-Port(13)
    Switch-Port(14)
    Switch-Port(15) ETH_TO_PCM_BRIDGE
    Switch-Port(16) SYSTEM ETHERNET OUTPUT
KAM/CHS/13U Chassis 45 - PCM Chassis

IP Address	Throughput Mode	PTP Leap Seconds	Current Year	Synchronization Level	Time Server
192.168.28.8	Prioritised 1Gbps	35	2014	500	PTP Grandmaster

No errors

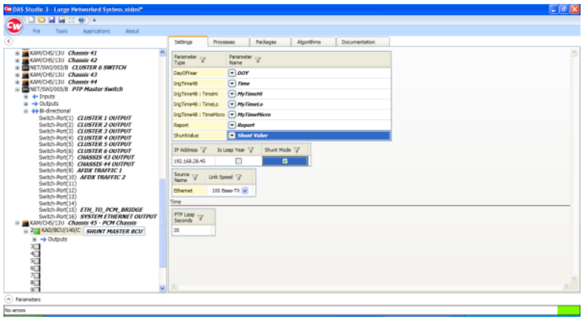
Strain Gauge calibration

The traditional shunt processes involves attaching a shunt resistor across one of the arms of the strain gauge bridge and checking that the desired deflection was achieved on the bridge output. However, this process can be time-consuming – in particular where there are a lot of strain gauges. Curtiss-Wright has developed a pseudo-shunt procedure whereby this same effect can be achieved by driving a known current across one arm of a bridge to replicate the effect of a large shunt resistor and checking the resulting deflection.

The shunt process is initiated by simply wiring a switch to the inputs of a discrete acquisition card. Flicking this switch is reflected in a status word which is read by the chassis controller across the backplane. This chassis is thus placed in shunt mode. In a multi-chassis system, this chassis will act as the shunt master chassis. The Shunt Master chassis then tells all other chassis in the system to also shunt at the same time via a series of specific Ethernet packets being sent out from this chassis across the network to all other chassis in the system.

However, this creates a dilemma; how do you create a path across the network from this Shunt Master chassis to talk to all other chassis in the network, while ensuring that this path is not overwhelmed with all the network traffic that could possibly flow on the system?

The answer to this was found by carefully choosing which chassis was the Shunt Master and then managing the traffic through crossbar switches and traffic filtering. Identifying the Shunt Master Chassis is simple – networked systems have a natural pyramid type shape, where all chassis transmit their data across the network towards one Ethernet to PCM bridge chassis, where the required PCM data is extracted out of the network traffic for inclusion in PCM. This chassis is essentially at the top of the pyramid and already has a path to receive traffic from all other chassis. A combination of traffic routing and packet filtering can be employed on the switch ports to ensure that this chassis controller, with its 100BaseT link, can talk to everyone else, while not being overwhelmed by the 1000BaseT traffic on the line.



KAM/CHS/13U Chassis 41
KAM/CHS/13U Chassis 42
NET/SW1/003/B CLUSTER 6 SWITCH
KAM/CHS/13U Chassis 43
KAM/CHS/13U Chassis 44
NET/SW1/005/B PTP Master Switch
Switch-Port(1) CLUSTER 1 OUTPUT
Switch-Port(2) CLUSTER 2 OUTPUT
Switch-Port(3) CLUSTER 3 OUTPUT
Switch-Port(4) CLUSTER 4 OUTPUT
Switch-Port(5) CLUSTER 5 OUTPUT
Switch-Port(6) CLUSTER 6 OUTPUT
Switch-Port(7) CHASSIS 43 OUTPUT
Switch-Port(8) CHASSIS 44 OUTPUT
Switch-Port(9) AFDX TRAFFIC 1
Switch-Port(10) AFDX TRAFFIC 2
Switch-Port(11)
Switch-Port(12)
Switch-Port(13)
Switch-Port(14)
Switch-Port(15) ETH_TO_PCM_BRIDGE
Switch-Port(16) SYSTEM ETHERNET OUTPUT
KAM/CHS/13U Chassis 45 - PCM Chassis
2 KAD/BCU/140/C SHUNT MASTER BCU

Parameter Type	Parameter Name
DayOfYear	DOY
IrigTime48	Time
IrigTime48 : TimeHi	MyTimeHi
IrigTime48 : TimeLo	MyTimeLo
IrigTime48 : TimeMicro	MyTimeMicro
Report	Report
ShuntValue	Shunt Value

IP Address	Is Leap Year	Shunt Mode
192.168.28.45	☐	☑

Source Name	Link Speed
Ethernet	100 Base-TX

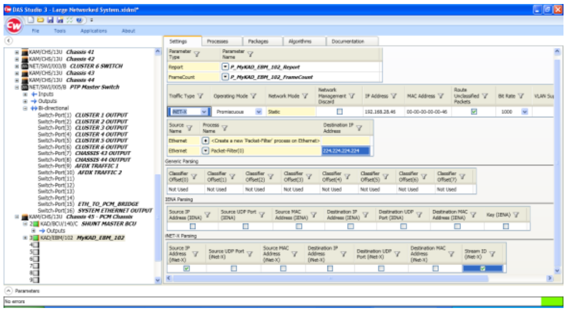
PTP Leap Seconds
35

The Ethernet to PCM Bridge

With the volume of data in FTI Systems ever increasing it has become impossible for Engineers to rely on solely PCM for analysis of flight data. Over time it has become the norm for PCM to be only a small subset of the whole data set that is captured and recorded during a flight. It has been mentioned several times in the paper that an Ethernet to PCM bridge gateway chassis is used to extract the required data out of the Ethernet traffic for transmission in PCM.

The process of how this works is one of the greatest challenges in achieving coherent PCM data for ground station analysis. In the DAS Studio software when constructing the PCM frame, the software lists all parameters available to the PCM stream, based on the chassis to switch routing topology as defined by the user. The user can then define the broad rules of how the PCM frame should be structured, choose the sampling rates for the required parameters – the software then takes care of the rest.

The software does this by identifying which parameters are required from which chassis and what sample rate they are required at. It then goes and builds transport packets to get the parameters out of these chassis at the required rate and across the network into the PCM bridge chassis. The bridge itself is simply a chassis with an Ethernet bus monitor that reads these transport packets off the network and makes them available to the PCM transmitter module in the same chassis. In order to differentiate this transport Ethernet traffic from all of the other traffic, the packets are grouped together into one multicast group, which enables easy traffic routing and filtering in the switch cores.



KAM/CHS/13U **Chassis 41**
KAM/CHS/13U **Chassis 42**
NET/SWI/003/B **CLUSTER 6 SWITCH**
KAM/CHS/13U **Chassis 43**
KAM/CHS/13U **Chassis 44**
NET/SWI/005/B **PTP Master Switch**
  Inputs
  Outputs
  Bi-directional
    Switch-Port(1) **CLUSTER 1 OUTPUT**
    Switch-Port(2) **CLUSTER 2 OUTPUT**
    Switch-Port(3) **CLUSTER 3 OUTPUT**
    Switch-Port(4) **CLUSTER 4 OUTPUT**
    Switch-Port(5) **CLUSTER 5 OUTPUT**
    Switch-Port(6) **CLUSTER 6 OUTPUT**
    Switch-Port(7) **CHASSIS 43 OUTPUT**
    Switch-Port(8) **CHASSIS 44 OUTPUT**
    Switch-Port(9) **AFDX TRAFFIC 1**
    Switch-Port(10) **AFDX TRAFFIC 2**
    Switch-Port(11)
    Switch-Port(12)
    Switch-Port(13)
    Switch-Port(14)
    Switch-Port(15) **ETH_TO_PCM_BRIDGE**
    Switch-Port(16) **SYSTEM ETHERNET OUTPUT**
KAM/CHS/13U **Chassis 45 - PCM Chassis**
  2 KAD/BCU/140/C **SHUNT MASTER BCU**
    Outputs
  3 KAD/EBM/102 **MyKAD_EBM_102**
  4
  5
  6
  7
  8
  9

Parameter Type	Parameter Name
Report	P_MyKAD_EBM_102_Report
FrameCount	P_MyKAD_EBM_102_FrameCount

Traffic Type	Operating Mode	Network Mode	Network Management Discard	IP Address	MAC Address	Route Unclassified Packets	Bit Rate	VLAN Su
iNET-X	Promiscuous	Static	☐	192.168.28.46	00-00-00-00-00-46	☑	1000	

Source Name	Process Name	Destination IP Address
Ethernet	<Create a new 'Packet-Filter' process on Ethernet>	
Ethernet	Packet-Filter(0)	224.224.224.224

Generic Parsing

Classifier Offset(0)	Classifier Offset(1)	Classifier Offset(2)	Classifier Offset(3)	Classifier Offset(4)	Classifier Offset(5)	Classifier Offset(6)	Classifier Offset(7)
Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used

IENA Parsing

Source IP Address (IENA)	Source UDP Port (IENA)	Source MAC Address (IENA)	Destination IP Address (IENA)	Destination UDP Port (IENA)	Destination MAC Address (IENA)	Key (IENA)
☐	☐	☐	☐	☐	☐	☐

iNET-X Parsing

Source IP Address (iNet-X)	Source UDP Port (iNet-X)	Source MAC Address (iNet-X)	Destination IP Address (iNet-X)	Destination UDP Port (iNet-X)	Destination MAC Address (iNet-X)	Stream ID (iNet-X)
☑	☐	☐	☐	☐	☐	☑

No errors

Quite often, these PCM parameters are also required to be sampled at higher rates for the “BIG Data” recorded on the aircraft. In these cases, these extra packets required for PCM transmission do not affect the sampling of this channel at the higher rate. The parameter will be sampled as required and the appropriate subset of these samples will be placed into the transport packets.

Network Latency

The network latency of these transport packets is critical to achieving coherent PCM data – however, when this was first implemented in a system of this scale it was discovered that the latency across the network was too unpredictable, with small fast transport packets getting stuck behind larger slower recorder packets as the traffic traversed the three layers of switches on the way to the Ethernet to PCM bridge.

The solution to this core issue resulted in what we refer to as the “clock-worked” solution. This is a process by which all the traffic transmitted by chassis was analyzed, taking into account packet size, line speeds and calculated latency figures for every single device in the path of each packet, from source chassis to Ethernet bus monitors and all switches in between. Every packet is assigned a transmit time to ensure no packet gets delayed by another packet in the network and therefore makes it to the Ethernet to PCM bridge in time for it to be read and inserted into PCM.

Then software calculates “receive-by” times which are essentially the moment in time it must receive a transport packet in order to get the parameters into their PCM location coherently. It then compares the initial transmit by time, plus the max. calculated path delay to see if this receive-by time can be achieved. If not, the transmit time is moved to the earliest possible moment, while keeping the required sampling windows for that parameter. If this still does not show to be achievable then the receive-by window is pushed enough to ensure all parameters make it into PCM on time to guarantee coherent data. The consequence of this may be that the start of the PCM frame is delayed.

Achieving Coherent PCM

Once network latency had been implemented the customer was invited over to test and confirm the full system operation. Modules of the same type, sampled at the same rate into PCM, from anywhere in the system, were checked for coherency, accuracy and latency against a specification of <1% of the sample frequency.

A sample of the results for some different module types is as follows for a 6.5Mbps PCM frame. The 139 strain channels were tested in the full system at 512Hz, 256Hz, 128Hz, 64Hz, 32Hz, 16Hz & 8Hz. The typical results were, comparing the same sine wave into any two modules at the same time, in the region of 50ns to 350ns at 512Hz. These were tested with both low input frequencies and full frequency sweeps up to the filter cut-off. Similarly, the 240 ARINC 429 channels were tested and found to be not more than 375ns apart.

External Traffic

Not all the traffic in the system originates from the same vendor. Non clock-worked traffic in such a system can be managed in the switches so as not to interfere with the Ethernet to PCM bridge through a combination of routing and filtering the traffic to the correct ports.

This was tested on the system using ADFX traffic which was introduced into one of the ports of the switch modules in one of the chassis. The switch module was programmed to aggregate both the AFDX traffic, along with the traffic from the controller in that chassis to the cluster level switch and on to the main switch where the transport packets were filtered out to the PCM Bridge module.

For testing, this was done while viewing a sine wave from a strain gauge module in the same chassis overlaid against the same sine wave input into another module in another part of the system. The AFDX traffic was increased until it started to affect the coherency of the sine waves. Along with the traffic from the controller in that chassis, AFDX traffic could be added to reach 93Mbps throughput on the 100BaseT link before any effect was observed.

CONCLUSIONS

This paper was written to highlight some of the issues that may arise when working with large Ethernet based FTI Systems, and what approaches were found to be effective in dealing with these issues. Using the solutions developed to meet these issues, whether your system is a simple two/three chassis configuration or a very large scale multi-chassis, multi-switch layer system, coherent PCM data can be guaranteed.

References:

[1] [1] Luiz Fernando de Souza, Domingos Henrique Beolchi Rios and Stephen Willis, 2012, “PCM to Ethernet: A Hybrid System Used to Certify the Next Generation of Data Transfer Technology”, Proceedings of the International Telemetry Conference 2012

[2] [2] IEEE Standards Committee, 2004, “Precision clock synchronization protocol for networked measurement and control systems”, IEEE Std. 1588

[3] [3] Cooke, Alan and Corry, Diarmuid, “XidML: A Global Standard for the Flight Test Community”, ETTC Proceedings, 2004