DESIGN CONSIDERATIONS FOR NETWORKED DATA ACQUISITION SYSTEMS

Nikki Cranley, Ph.D, Diarmuid Corry, M.Sc.
ACRA CONTROL INC., Maryland, USA

ABSTRACT

Ethernet technology offers numerous benefits for networked Flight Test Instrumentation (FTI) systems such as increased data rates, flexibility, scalability and most importantly interoperability owing to the inherent interface, protocol and technological standardization. However, the best effort nature of Ethernet is in sharp contrast to the intrinsic determinism of tradition FTI systems. The challenge for network designers is to optimize the configuration of the Ethernet network to meet the data processing demands in terms of reliability and latency. This paper discusses the necessary planning and design phases to investigate, analyze, fine-tune and optimize the networks performance.

I. INTRODUCTION

In recent years there has been a shift from proprietary and closed solutions for Flight Test Instrumentation (FTI) networks towards more open standards-based systems using Ethernet technology [1, 2, 3]. The trend towards Ethernet is further driven by the CTEIP Integrated Network Enhanced Telemetry (iNET) initiative that is pushing the adoption of Ethernet technology for the future of FTI [4]. iNET is addressing and bringing about a standardized approach to meet the needs and requirements of FTI in terms of systems management, time synchronization, Data Acquisition Unit (DAU) configuration, data transmission etc. Using open standard technologies offers the FTI community with greater flexibility and scalability in system design and more choice for multi-vendor interoperable systems. As more ranges move towards Ethernet and together with the momentum of iNET, the number of published networked solutions both in small and large systems will increase. However, Ethernet technology is best effort: Ethernet does not make any guarantees with respect to data reliability, in-order delivery, latency, or determinism. Therefore it is crucial to design the FTI network such that these negating aspects of data transmission may be mitigated.

During the design and planning phases of a networked FTI system, extensive modelling, simulation and emulation of the intended topology and data traffic are required in order to ensure the Quality of Service (QoS) targets can be achieved. QoS targets are typically quantified by the network-related metrics: end-to-end latency, jitter, throughput, and packet loss probability. It cannot be assumed that the network is homogenous since although networked devices may contain the same networking functions and interfaces, the internal operations in terms of switching, queuing and buffering may differ rendering accurate modelling difficult. Furthermore, modelling the characteristics of the projected data traffic is particularly arduous for ad-hoc, asynchronous, and adaptive traffic such as those transmitted as Transmission Control Protocol (TCP). There are numerous QoS techniques categorised as Differentiated Services (DiffServ) or Integrated Services (IntServ) that can be employed to minimize end-to-end latency, jitter, and packet loss probability. However, these techniques are only truly effective when there is congestion in the network and the links are running close to capacity. Often it is favoured to simply adequately dimension or over-provision the network by allocating the necessary resources to service the projected data rates. In this paper, the components and characteristics of a heterogeneous networked FTI system are presented. In particular this paper discusses techniques to optimize and ensure the networked FTI QoS performance targets are met.

This paper assumes that the reader is familiar with Ethernet networking technologies, an overview of networked data acquisition systems can be found in [5]. The remainder of this paper is structured as follows: Section 2 provides an overview of the key components that comprise a networked data acquisition system. Section 3 discusses the various aspects of the networks' operation that must be considered during the preliminary network design and planning phases. In particular, an important aspect of the networks deployment is the definition of QoS. This involves the quantification of acceptable performance metrics and ensuring that the network is adequately designed and provisioned ensuring that these QoS performance targets can be met. Network simulations and emulations can be used to fine-tune and optimize the networks' configuration. Finally, some simple considerations and techniques to optimize the networks' design are discussed.

II. NETWORK DATA ACQUISITION

In a networked FTI system there are a number inter-connected networked nodes, as shown in Figure 1, these include: Ethernet links, DAUs, network switches, the 1588 Precision Time Protocol (PTP) Grandmaster, and the network recorder.

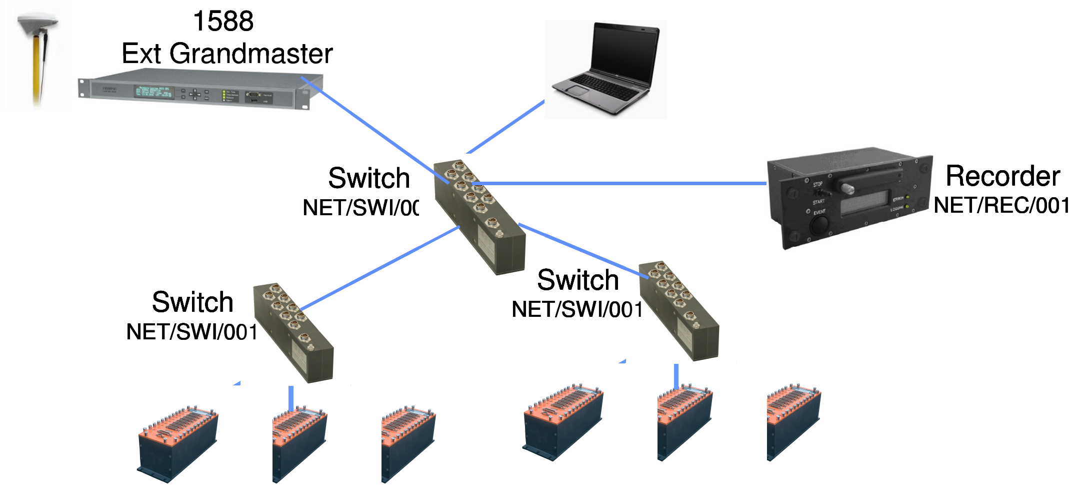


Figure 1. Network Tree Topology

A. *Ethernet Links and Speeds*

There are a number of Ethernet standards that define the physical properties of packet transmission across the network. Ethernet links may be either Fast Ethernet (FE) or Gigabit Ethernet (GbE), which determines the capacity of the link to carry data. FE is a collective term for a number of Ethernet standards that carry traffic at the nominal rate of 100Mbps. FE may be transmitted over twisted-pair copper or fiber links i.e. 100BaseTx (100Mbps over two-pair Cat5 or better cable) or 100BASE-FX is a version of Fast Ethernet over optical fiber. As the name implies, GbE is the term for the family of technologies that govern the transmission of Ethernet frames at a rate of a 1Gbps. This too, may be transmitted over copper, 1000BaseCx, or fiber, 1000BaseX. By far, 100BaseTx is the most common Ethernet standard used today for data transmission while GbE technologies are more suited for backhauling the aggregate data to a on-board data processing device or network recorder.

B. The Data Acquisition Unit

As previously described, the DAU must be capable of being synchronized using the IEEE 1588 PTP protocol [6, 7] and begin the transmission of the telemetered data reliably and without interruption in real-time within a very short boot-up cycle (<10s). The telemetry data is packetized according to the chosen Application Layer protocol, which is then typically subsequently encapsulated as a UDP/IP packet. The Application layer protocol prepends additional descriptive metadata to facilitate the decoding and processing of the data, for example, through the inclusion of sampling timestamps, sequence numbers and so on. There are a number of Application Layer protocols that have been developed for FTI telemetry data such as the IENA protocol from Airbus or the iNET data message format [8, 9, 10].

For a DAU to be able to transmit data, the DAU must be assigned a system-wide unique IP address. There are two main protocols to assign IP addresses, dynamically using the Dynamic Host Configuration Protocol (DHCP) or statically using the Bootstrap Protocol (BootP). The main disadvantage in using DHCP configured DAUs is that the IP address assigned to the DAU is not known in advance. (Although static IP address assignments are possible with DHCP). More crucially, DHCP is more complex with in-built mechanisms for IP address lease renewal, rebinding, and expiration that may not be suitable for a networked FTI system. As a consequence DHCP may be slow and that in turn prolongs the delay between power-up and the start of data transmission. On the other hand, BootP is a simple and fast method of assigning IP addresses. On power-on the DAU can configure itself dynamically and without user supervision. BootP provides a means to notify a DAU of its assigned IP address, the IP address of a boot server, and the name of a file to be loaded into memory and executed. For DAUs that do not know their IP addresses, the DAU contacts the BootP server requesting its IP address. The server has a database that maps hardware addresses to IP addresses which is relayed back to the DAU in the bootreply from the server. The advantage of static IP address assignment is that the network can be configured and optimized with prioritization and routing. Since the DAU supports standard Ethernet technologies, networking tools such as Ping, trace route, and packet sniffers may be used to debug and analyze the performance of the DAU. Management protocols such as Simple Network Management Protocol (SNMP) may be used to issue SET and GET commands on the DAU allowing for configuration and state changes to be remotely invoked [11].

C. The Switch

In a network of distributed peer DAU's, the switch is a key component that allows data to be transmitted to and from different nodes in the network. Switches comprise a number of ports to which DAU's may be connected or inter-connect switches. In general, connections in the switch utilize point-to-point full-duplex Ethernet links. By far, the most important task of the switch is to reliably and quickly forward packets to their destination. The packets received by the switch are stored in a buffer until they reach the head of the queue. Once at the head of the queue, the switch examines the packets' destination and through a lookup mechanism determines how to forward the packet to its intended destination. This process is known as Store-and-Forward, the packets are stored in a queue until they are switched or forwarded. Switching introduces delay and jitter. There are two components to the delay in the switch, the queuing delay and the switching delay. Queuing delay is the buffering of waiting time of the packets to reach the head of the queue, whilst the switching delay occurs during the lookup mechanism. Moreover, since buffers have a finite capacity, by ensuring that the packet arrival rate does not exceed the service or switching rate, no packets will be lost in the switch due to buffer overflow and furthermore the average queuing delay for the packets can be minimized.

Switches vary in complexity: unmanaged switches have no configurable options while managed switches have a configuration interface and additional management features. Although the available configuration features is largely vendor-specific, typical managed switch options include:

- Prioritization of packets based on IP address, IP priority flags using the Differentiated Service Code-Point value (DSCP) or MAC layer prioritization using VLAN 802.1q tags;
- SNMP Agents to report network-related statistics e.g. packets received per second, packet loss etc.;
- Sniff and Mirror ports allowing the packets on a given interface to be copied to another interface for analysis.

The user should be cognizant that the internal operations in the switch are not standardized and that this in turns affects how the packets are switched which may encumber deterministic behavior, consume processing/switching resources, and typically have a longer start-up delay. The switch should have a short boot-up time comparable to the boot-up time of the DAU's (<10sec).

1) PTP Support

The unusual aspect of FTI networks is that the traffic load in the network is heavily asymmetric particularly at medium to heavy data rates. This asymmetry arises from the fact that the DAU's acquire and transmit data at a much faster rate than they receive data. This affects the efficacy of the network to carry PTP messages between the grandmaster and the DAU's since one of the underlying assumptions of 1588 PTPv1 is end-to-end network delay symmetry. To overcome this asymmetry and improve synchronization accuracy the switch should support 1588 compensation mechanisms. The two compensation mechanisms include 1588 Boundary Clock and 1588 Transparency.

In Boundary Clock mode the switch itself is synchronized in slave mode to the attached master clock. Once synchronized the switch then acts as the master clock to all the attached slaves. In this way, delay asymmetries and internal delays are compensated. The disadvantage of Boundary Clock mode is that synchronization occurs at each switch hop between the grandmaster and the DAU resulting in an accumulation of non-linear time offsets degrading the synchronization accuracy. This is problematic in highly cascaded network topologies.

Transparency is not part of the IEEE 1588 PTPv1 standard and is therefore implemented in a vendor-specific way for PTPv1. However, there are significant commonalities in the various transparency implementations [12]. In Transparency mode, the switch intercepts the incoming PTP packets and calculates the time in residence (i.e. the buffering, queuing, and switching delay). Before the PTP packet is forwarded on, the timestamp in the PTP packet is modified to compensate for this residence time in the switch. The Transparency mode results in more accurate end-to-end synchronization between Grandmaster and DAU than can be achieved using Boundary Clock operations.

D. The Network Recorder

The network recorder allows for packets to be captured on the network link. As with all other nodes in the network, the network recorder is synchronized using the 1588 PTP protocol. 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.

An important aspect of recording data is choosing a suitable file format that facilitates fast read and write functions in real-time. At high data rates, for example on GbE links, complex processing may potentially exhaust the recorders' resources and result in recording gaps. For example, there is a significant processing overhead to examine each packet to locate and identify a specific parameter to be recorded. Furthermore, in order to extract the data from the packets, the network recorder would need to know the structure and placement of each parameter in each stream.

The simplest solution to this is to record the data in their packetised form, that is, timestamp and record the IP packets as they arrive and write them quickly and reliably to file. Wireshark is one of the most popular network analyzer and packet sniffing utilities available. Wireshark allows packets on the network to be recorded to file using the Packet CAPture (PCAP) file format [13, 14]. The PCAP file format is a simple, lightweight file format that supports the recording of packetised data on a variety of networking technologies including Ethernet, WLAN, Bluetooth, Token Ring, IrDA, USB and so on. From a user perspective, an open-standard, simple, file format allows custom applications and software to be developed to post-process the recorded data.

III. DESIGNING AND OPTIMIZING THE NETWORK

During the preliminary planning and design phases of networked data acquisition system, there are two core aspects of the systems configuration that must be addressed: physical network node limitations and intended target data routing [15].

Physical considerations:

- The number and types of networked nodes in the system needs to be determined (e.g. DAU's, network recorders, 1588 grandmaster, on-board data processing stations and PCM RF link requirements).
- The network topology is constructed by determining the number of interconnecting network switches and the number of physical interfaces on these devices. For networked FTI data acquisition systems, typically a cascading switch tree-type topology is adopted with full-duplex DAU-switch and inter-switch connections.
- The aggregate data rates generated by each DAU impact on the choice of link speeds and physical media, i.e. fiber or copper Fast Ethernet or Gigabit Ethernet. These options are naturally balanced against the availability of such physical interfaces in the switches.
- The functional requirements of the interconnecting switches in terms of managed versus unmanaged features. Such features may include the ability to pre-configure the switch with static routing tables, prioritized queuing mechanisms and so on.

Routing and Data Flow:

- For each DAU it is necessary to identify the number of data flows and the intended destinations for each flow. Furthermore, per-flow optimization is required in terms of minimizing the packetization latency and maximizing the payload utility such that the packet rate is reduced.
- The transmission paths of the data, choosing between unicast as opposed to multicast routing, and configuring routing tables. Similarly choices with regard the protocols used for the data

flows including the Transport Layer protocol (i.e. TCP, UDP) and Application Layer protocol (i.e. IENA, iNET etc.).

- Ancillary network services, such as network management and dynamic reconfiguration using SNMP, should be identified.

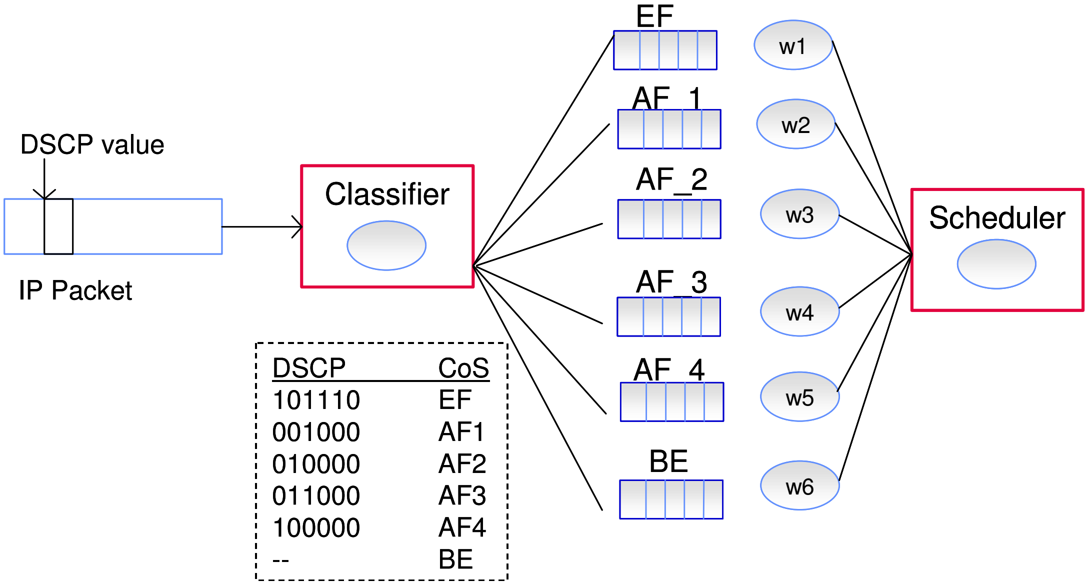
Concurrent to the design of the physical network, network performance acceptance targets must be defined. The network design and topology have a direct impact on the transmission reliability and end-to-end latency of the acquired data. To improve upon the best-effort nature of data delivery over Ethernet, reliability and delivery enhancing mechanisms (such as link redundancy, bandwidth over-provisioning, traffic flow bandwidth reservation and traffic prioritization) can be integrated and/or enabled in the network.

A. Quality of Service

Quality of Service (QoS) is the measure of performance for a networked FTI data acquisition system that reflects the data transmission quality and service availability. QoS is captured by the key metrics: latency, jitter, loss, throughput and determinism. During the network design, the upper bounds of acceptable QoS are defined to ensure real-time data integrity, reliability and availability. Moreover the QoS definitions need not be homogenous for all data types nor for all applications, for example, data destined for a network recorder does not have the same real-time latency constraints as those mission critical data destined to be transmitted over PCM RF links and processed in real-time. Similarly, certain multimedia flows such as video and voice have a certain degree of packet loss tolerance that does not impede data decoding and display.

There are two classes of QoS provisioning mechanisms, Integrated Services (IntServ) or Differentiated Services (DiffServ). IntServ is configured on a network-wide basis whereas DiffServ is deployed on a per-node and per-switch basis implementing what is known as Per Hop Behaviour (PHB). Integrated Service (IntServ) is a QoS enabling mechanism that provides fine-grained QoS. Each router and switch in the network must be configured with IntServ and traffic flows may request and be reserved bandwidth across the network path. For example, the ReSource reserVation Protocol (RSVP) provides mechanisms to request and reserve resources through a network.

The more popular QoS approach is Differentiated Service (DiffServ) that implements a Per Hop Behavior (PHB) whereby there is coarse behavioral differentiation between the individual traffic flows. In DiffServ, traffic flows are differentiated by setting the Differentiated Service Code Point (DSCP) field in the packet IP header to a specific value that is subsequently interpreted by the switch to provide a particular service to the traffic flow. The intermediate switches supporting DiffServ contains multiple queues that are configured with different behaviors. There are three main queue Classes of Service (CoS): the Assured Forwarding (AF) queue for reliability critical traffic; the Expedited Forwarding (EF) queue for latency critical traffic; the Best-Effort (BE) queue that is used for all other traffic. The switch is configured with a DiffServ policy whereby specific DSCP values are mapped to a corresponding queue. As packets arrive in the switch, the DSCP value is examined and the packet is buffered in the appropriate queue. Each queue is serviced according to some scheduling policy that achieves the desired properties of the queue as shown in Figure 2. Furthermore, each queue may be configured with a specific buffer management mechanism to handle queue overflow for example.



DSCP	CoS
101110	EF
001000	AF1
010000	AF2
011000	AF3
100000	AF4
--	BE

Figure 2. DiffServ Per Hop Behavior Operation

An important caveat, DiffServ QoS is most effective when the network is homogenous and that all intermediate nodes and switches are configured with the same QoS policy and use the same internal scheduling policy. Moreover, it is neither trivial nor always possible to adequately partition and prioritize the traffic flows mapping them to a given queue such that DiffServ mechanism provides optimal efficacy in a diverse range of network conditions. For example, if a majority of the traffic is mapped to an EF queue, this yields little benefit over a pure best-effort solution. QoS mechanisms yield maximum efficacy in the intervals preceding the onset and during periods of congestion, however, in an unsaturated over-provisioned network, the benefits of QoS may not be observable or significant.

Investigating and exploring the numerous possibilities for configuring the network for optimal performance requires significant testing prior to deploying the network. Performing this analysis and investigation with a real network is time-consuming and from a practical standpoint, only a fraction of the myriad of possible configurations may be covered.

B. Network Simulation and Emulation

For small scale and relatively simple network configurations, analytical techniques such as the use of network queuing theory can provide an insight to the systems' performance. The application of analytical techniques becomes complex and less accurate for example, as the number of nodes increases; "unknown" and complex internal operations in the hardware to optimize the nodes throughput, possibly integrating proprietary prioritization mechanisms; protocol sophistication; interactivity and ad hoc externally generated events.

To overcome these issues, simulation is beneficial for performing network research and development and more specifically for analyzing realistic scenarios of network operations and deployments. Network simulation allows for behavioral models of the network topology, protocols, and characteristics of the network nodes (e.g. DAU's and switches) to be assessed such the systems' performance can be observed under a variety of controlled test conditions. In essence a network simulator is a software surrogate for a real network. During the network planning and design phase, real networks, particularly large network configurations, are difficult to instantiate (purchase, install and configure) in order to experiment with scenarios. Moreover, it is difficult to experimentally generate or recreate specific network conditions of interest. In this way simulation acts as a software surrogate enabling the network designer to easily and quickly evaluate different configurations in order to avoid congestion and optimize the network design. Some of the more popular simulators include OPNET [16], QualNet [17], ns-2 [18], NCTUns [19], and Glomosim [20].

It is important to note that a simulation is only as good as the accuracy and realism of models within the package. For example, not all network switches operate in the same manner; there may still be certain proprietary optimization mechanisms that are not captured by the simulator. To improve upon the realism and accuracy of a network simulator, emulation is a technique that is used in conjunction with simulations that allow for the integration of live applications or live network nodes in a simulation. For example, in FTI this would include the ability to attach a live DAU into a network simulation so that the end-to-end transmission of the packets generated by the DAU to be assessed. Traffic generators are devices that can be used to emulate the end-nodes or DAU's. For black box testing, a traffic generator can be used to create traffic with known and specific statistic properties through a 'live' switch allowing for behavioral operations of the switch to be observed and for models to be derived, which can be subsequently plugged into a simulation package.

C. Getting the most out of the network

One of the most important caveats associated with the use of Ethernet technology is that Ethernet is best effort. That means there is no guarantee that packets will delivered in time, in order or arrive at the destination at all. There are a number of simple techniques that may be employed in the design of a networked FTI system to achieve better determinism, reliability, throughput, and reduced latency.

Achieving Determinism:

- Limit the use of dynamic traffic and adaptive protocols on the network during data acquisition i.e. TCP, SNMP, FTP data transfers etc.
- Avoid dynamic multicast joins (if IGMP is enabled).
- Configure switches with static routes to avoid MAC address learning, table timeouts and dynamic routing algorithms. Moreover it should ensured that these configurations be retained on power-up.
- Understanding of the switch behaviour in terms of queuing and switching algorithms. For example some switches implement Head of Line Blocking Prevention, Virtual Output Queues, Random Early Discard, Expedited Forwarding and so on.
- The network topology should be kept static, nodes should not be dynamically added/connected to the network nor should they transmit data sporadically during data acquisition.
- Assuming stable and constant traffic conditions, transmitting data in a deterministic manner results in quasi-deterministic received data**.**

Achieving Reliability:

- Use Ethernet IEEE 802.3 full-duplex point-to-point collision free links between the DAU and the switch, ensures that no packets are lost due to line contention or collisions.
- The network topology is known in advance, stable and does not change during a test.
- Over-provision the network. The aggregate peak data rate on each link never exceeds the capacity of that link.
- Switch load balancing. Queues in the switch have a finite capacity. The packet arrival rate in the queue should be less than the switching rate in order to prevent overflow and lost packets.
- Distribute the packet load from each DAU as uniformly as possible across the switch interfaces.

- Distribute the data load as uniformly as possible across the network.

Maximizing Throughput:

- Load balancing to avoid saturating any one single link.
- Use GbE and aggregate links if necessary.
- Make effective use of multicast. By using IGMP only receivers interested in the data, receive the data.
- Optimize data packetization strategies.
  - For each packet that is transmitted there is an associated network header overhead. By transmitting data in larger packets, fewer packets need to be transmitted for the same quantity of data, which in turn reduces the bandwidth used by network header overhead.
  - Furthermore, by transmitting data in larger packets, this reduces the number of packets that need to be switched, which improves the switch throughput.

Minimizing Start-up Delays and End-to-End Latency:

- Minimize the Start-up delay by ensuring that all DAUs and switches in the system boot-up quickly (<10s). 30seconds or more is too long, for example avoid DHCP to assign IP addresses on start-up.
- Minimize the number of switches through which packets must pass to reach their destination. There is an associated queuing and switching delay.
- Maximize the amount of data in a packet by using larger packets. By reducing the number of packets switched this reduces the queuing time for each packet to be switched.

Keep It Simple and Stupid (KISS):

- Design simplicity and reliability should be a key goal. Unnecessary complexity and sophisticated features that could potentially interfere with FTI data transmission should be avoided.

IV. CONCLUSIONS

Networked FTI solutions are becoming more and more prevalent. Ethernet technology offers many benefits to the FTI community including open standards-based technologies, greater vendor inter-operability, system design flexibility and simplicity. Naturally there is a challenging transition from deterministic “legacy PCM” towards a non-deterministic packet-switched system. The adoption of Ethernet requires a fundamental change in how data acquisition is performed, planned, and designed. However, with a good understanding of Ethernet and the behavior of the components in the network, in particular the switches, the network can be designed and optimized to meet the requirements and challenges of FTI. INET is actively driving the adoption of Ethernet, designing platforms of interoperability, and developing solutions for the short and long-term future of FTI.

This paper provides the reader with an overview of components in an Ethernet network in terms of their operation and functional requirements. The networked components discussed include the networked-DAU, switches, IP recorders and the time reference Grandmaster. There are a number of core protocols that provide system-wide time synchronization; management services; and data transfer between these components across the network. Finally this paper described some ways the network can be designed, planned, and optimized. By far the best approach is to keep the network as simple as possible to achieve predictable and reliable behavior and performance.

REFERENCES

[1] Eccles, Lee, “Network Based data Acquisition”, Proceedings of European Telemetry Conference (ETC), Munich, Germany, 2008

[2] Revaux, Nathalie and Abadie, Frédéric, “A380 Flight Test Architecture: Switched IENA from Pulse Code Modulation to Ethernet LAN”, Proceedings of European Test and Telemetry Conference (ETTC), Toulouse, France, 2003

[3] Doyle, David and Canizares, Ruben, “Review of a successful distributed networked and modular FTI implementation”, to be published in Proceedings of Society of Flight Test Engineers Symposium 39, 2008

[4] Grace, Thomas and Hodack, David, “Vehicle Network Technology Demonstration”, Proceedings of International Telemetry Conference, 2007.

[5] Cranley, Nikki, Fielding, Richard, “Key Components and Trends in Airborne Networked Data Acquisition Systems”, In Proc. of European Test and Telemetry Conference (ETTC) 2009, Toulouse, France, June 2009

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

[7] Corry, Diarmuid, “System Wide Synchronization in Distributed Data Acquisition Networks”, Proceedings European Telemetry Conference, Munich, Germany, 2008

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

[9] Cranley, Nikki, “Real-time Transport Protocols for Telemetry Data and Signaling”, In Proc. of International Telemetry Conference, San Diego, CA, USA, 2008

[10] iNET Test Article Standards Working Group, “Test Article Standard – Proposed version 0.6.1”, March 2009

[11] IETF, RFC 3411, “An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks”, Dec 2002

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

[13] Wireshark, “LibPcap File Format”, http://wiki.wireshark.org/Development/LibpcapFileFormat, Available online: May 2009

[14] Wireshark, http://www.wireshark.org/, Available online: May 2009

[15] Cisco Systems, “Internetworking Design Basics”, http://www.cisco.com/en/US/docs/internetworking/design/guide/nd2002.html**,** Available online: May 2009

[16] OpNet, http://www.**opnet**.com/, Available online: May 2009

[17] QualNet Scalable Networks, http:// www.scalable-networks.com/, Available online: May 2009

[18] NS-2 Network Simulator, http://www.isi.edu/nsnam/**ns/**, Available online: May 2009

[19] NCTUns Network Simulator and Emulator, http://nsl.csie.nctu.edu.tw/**nctuns**.html, Available online: May 2009

[20] Glomosim Global Mobile Information Systems Simulation Library, http://pcl.cs.ucla.edu/projects/glomosim/, Available online: May 2009