

Packet Switching Networks For Adaptive Optics Systems

Robert J. Eager

Boeing – LTS, Starfire Optical Range, AFRL/DE, KAFB, NM.

ABSTRACT

High Performance and Extremely Large Adaptive Optics systems place great demands on the data distribution and electronic control sub-systems. These servo-loop sub-systems acquire, process, and drive electro-optical/mechanical devices with large degrees of freedom. Two of the key parameters that affect AO loop performance are latency and jitter. To minimize these parameters, data must be efficiently acquired from sensors, processed into servo commands and then distributed to actuators. In addition to these critical path activities, there is also a requirement to “tap” into many points in the processing pipeline to monitor, characterize and perform high-level corrections. One answer to this communications dilemma is the use of a low-latency serial packet switched network. This paper illustrates the actual performance and flexibility of such a network by describing the high-performance system used at the Starfire Optical Range. The paper will then demonstrate the adaptability of this network architecture to accommodate the more complex communication requirements of extremely large AO systems.

1. INTRODUCTION

High Performance and Extremely Large Adaptive Optics systems place great demands on the data distribution and electronic control sub-systems. These systems operate with large (1K+) degrees of freedom and high (1-10 KHz+) sampling rates to achieve optimal loop performance. These characteristics place heavy demands on the electronic communication’s infrastructure to convey data efficiently between the sensors, processing nodes and actuator elements, with minimal latency and jitter. One emerging technology that best addresses these demanding communication requirements is packet switched networks. The objective of this paper is to present the benefits of this technology for AO applications and to provide details on a system implemented using a packet switched network architecture developed at the Starfire Optical Range.

2. PARALLEL BUS ARCHITECTURE

Parallel bus architectures enable a set of clients, connected via a shared set of signals to communicate with each other. This architecture supports communication at the System, Board, and Chip levels of integration. In the beginning, this bus architecture was created to provide the simplest way to communicate between multiple devices. As the demand for higher bandwidth to support faster devices grew, the bus paradigm was evolved and restructured to utilize a cascade of specialized bus architectures. Faster bus architectures were used to connect IC’s while progressively slower architectures were used to connect boards and peripherals. [Ex: Memory (DDR) → Peripherals (PCI) → Hard-Drives (SCSI)]. This push for higher bandwidth also resulted in the need for high clock rates, and more data signals. The downside to this increase in bus bandwidth was the reduction in the number of devices that could share the bus. This shrinkage in connectivity was due to signal integrity issues associated with shared electrical signals and temporal alignment (skew) problems between these many parallel signals (Ex: PCI-X → 110 signal wires/slot).

3. PACKET SWITCHED NETWORK ARCHITECTURE

The packet switched network architecture utilizes a formatted block of information (packet), via high-speed serial data links, to route data between multiple end-node devices. This architecture supports communication at the System, Board, and Chip levels of integration. Currently, there are three classes of network architectures: dumb, intelligent, and context aware. Dumb networks utilize intelligent end-node devices (Ex: PCs) at its periphery to make use of the network and do not interfere with the end node's operations. Intelligent networks manage the communications between relatively dumb end-nodes and provide centralized control of the application. Context aware networks are a blend of dumb and intelligent architectures. In AO, the context aware architecture utilizes dumb I/O end-node devices (Timing, Camera, and RS422) and intelligent end-node devices (CPU, DSP and FPGA).

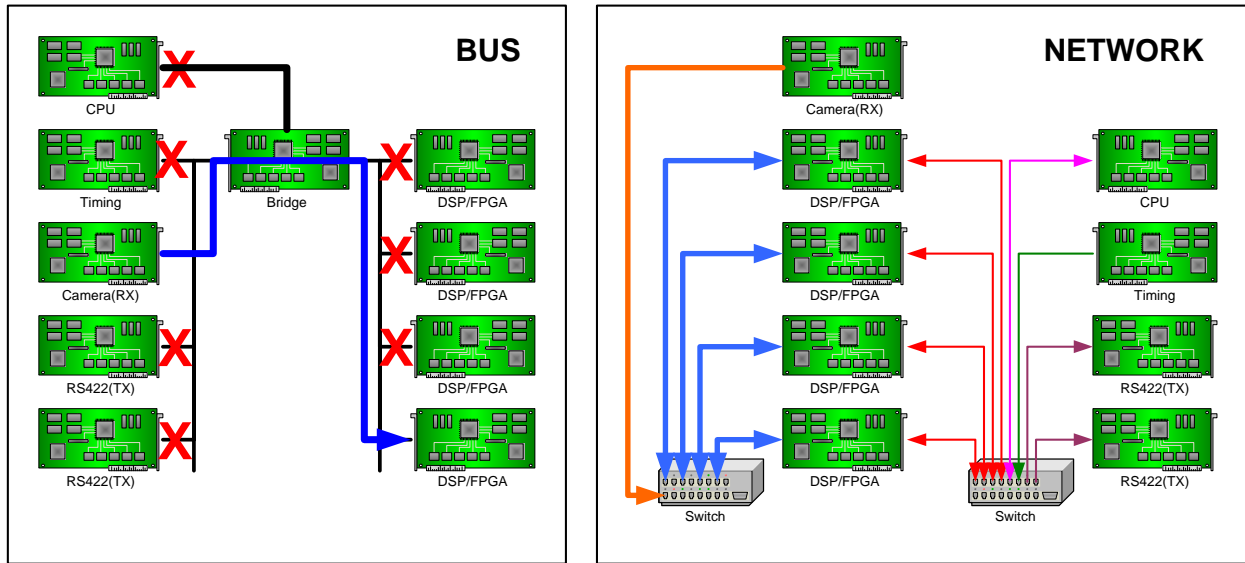


Fig. 1 –Network vs. Bus Architectures

4. COMPARISON BETWEEN BUS AND NETWORK ARCHITECTURES

Devices per System: Bus architectures use shared electrical signals that degrade in integrity with each new device added. Networks use point-to-point connections and multi-port switches to create multiple data paths between devices. The result is that there is no physical limit on the number of devices linked together in a network. The Internet with its millions of Ethernet connections is a testament to this capability. In Fig. 1, eight devices make up a system. In the network example, these devices are linked together via two switches. In the bus example, we see that the bus architecture requires that the eight devices be hosted on two electrically isolated local buses due to the connection limit of that bus technology.

Topology: Bus architectures are limited to either a single bus or tiered-bus (tree) communication topologies. Network architectures are virtually unlimited in their interconnect topologies (see section 10). In Fig. 1, the Bus example is a tiered-bus topology with two simple-bus structures bridged together to the head node where the CPU resides. This structure is commonly used in computer systems to increase the number of board slots available. The Network architecture implemented in Fig. 1 is an example of a dual star topology, with each switch implementing one star. One benefit of this particular network topology is its ability to support two simultaneous sets of data transfers between all DSP/FPGA boards.

Data Transfer Rate per Bus/Link (Bandwidth): The data transfer rate of Bus architectures is roughly a function of its clock speed and the width of its data bus. For example, a PCI bus operating at 66MHz and 64bits achieves a peak transfer rate of 503 Mega Bytes/sec (MB/s). Comparatively, the High Speed Serial (HSS) connections, used in network architectures, are typically serial (1-bit) data sequences which have the clock embedded into the data stream via 8B/10B encoding. This translates into a maximum data rate, in bytes, as 1/10th the bit rate of the link. To match the peak data rate of the PCI bus with the TX side of a data link, we would need to clock the link at 5.0GHz. Network

data transfer efficiency (actual/maximum) is a function of payload size per packet, and packet overhead. Efficiencies of 96% or better are achieved with ~ 1KB payloads for most network protocols.

Aggregate Data Transfer Rate: Bus architectures like VME and PCI are limited to one transaction at a time between any two devices residing on the same bus. Thus, the available bandwidth for N devices with pending bus requests scales as function of $[1/N]$. Networks avoid this bottleneck by supporting multiple connections per device and allowing simultaneous transfers to occur on all data links within the network. The only real constraint on a network's bandwidth is the number of active data links and the bandwidth utilization of each. In Fig. 1, we see the network example with multiple data flows occurring simultaneously (ex: orange arrow to all blue arrows), whereas the bus example shows that the transfer from the camera board to one of the DSP/FPGA boards (ex: blue arrow) blocks access to the bus for all other devices.

Data Transfer Flow Management: Bus architectures require at least one CPU to interact with each device to maintain the flow of data. This and low-level bus arbitration overhead, are the reasons for bus protocols never achieving sustained data rates comparable to their peak values. Networks, on the other hand, are “configured once and forget” architectures. No oversight manager is required to “maintain the flow” between devices. Thus, network links can operate, as close to maximum as the protocol and application will allow.

5. NETWORK ARCHITECTURES - CPU VS. DATA CENTRIC

Currently there are a number of industry standard packet switched network architectures competing for market share (Ex: Ethernet, Infiniband, Rapid-IO, ASI). The one thing that they all share in common is that they are optimized for communicating between computers. Thus, these architectures can be considered “computer-centric”. That is to say, the protocol assumes that the end-nodes are computers and that the data flow is managed to maximize the end-node’s CPU utilization. Applications best served by this centrality are those that are computationally bound. Super computer clusters and database servers are examples of systems that benefit the most from these architectures.

There is, however, a class of applications that would benefit from a “data-centric” network architecture. That is to say, the underlining protocol minimizes transfer latency between end-nodes and has the “processors in the system wait on the availability of the data” rather than the “data waiting on the availability of the processors”. It is this architecture, herein referred to as the “Fabric”, which was chosen to implement the High Performance Adaptive Optics System at the Starfire Optical Range, described in section 13.

6. KEY TECHNOLOGIES

The hardware used in packet switched networks, are board assemblies (devices) which utilize three key technologies:

- **FPGA+SERDES** – Are Field Programmable Gate Arrays (FPGAs) that have multiple High Speed Serial communication ports integrated with their programmable logic core. This component is used to host the Fabric’s low-level protocol and support the embedded Digital Signal Processing (DSP) algorithms.
- **Fiber-Optic Transceivers** – Enables the use of Fiber-Optic (FO) cables to convey HSS data streams between devices. This data transport medium offers the best in Size (3mm dia.), Weight (21g/m), Data Rates (1-10 GHz), and Range (1m to 5Km) as well as Electro Magnetic Interference (EMI) and lighting protection.
- **Multi-GHz class Printed Circuit Board (PCB) Materials** – Are structures used to electrically and mechanically host ICs and copper interconnects. Because HSS signals operate at multiple GHz, a new class of materials was developed to support transmission from chip-to-chip/transceivers with minimal signal degradation.

7. NETWORK DEVICES

The Fabric uses devices to create the target infrastructure. These devices come in three functional forms:

- **Switches** Perform packet-level routing between its full-duplex data link ports.
- **Adapters** Translate data streams between the Fabric and the target protocols.
- **Processors** Performs embedded Digital Signal Processing (DSP) of messages within the network.

These devices in turn are housed and connected via these components:

- **Enclosures** Used to provide mechanical, electrical and thermal management housing for devices.
- **Cables** Used to physically connect devices via their data link ports and provide external I/O.

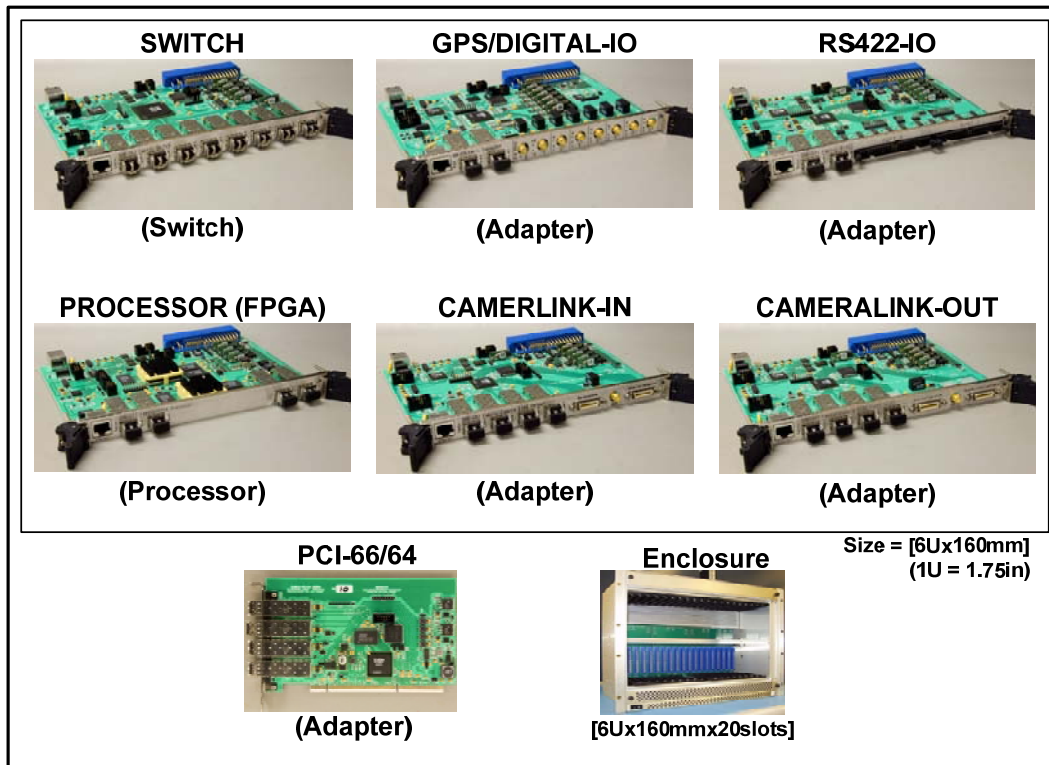


Fig. 2 – Fabric Hardware

Fig. 2 shows pictures of devices developed at the SOR to implement the Fabric. In this figure, we have:

- (1) **Switch** – Provides an eight 2.5GHz Full-Duplex (TX+RX) HSS ports for routing packets.
- (2) **GPS/Digital-IO** – Provides GPS correlated time stamp and programmable trigger event messages.
- (3) **RS422-I/O** – Provides Encoding/Decoding of source synchronous parallel data streams that comply with the RS422 electrical standard. This board is used to interface to legacy HW and DM Drivers (HVAs).
- (4) **Processor (FPGA)** – Provides a reconfigurable DSP node for hosting embedded AO algorithms. The board provides two Xilinx Virtex2 Pro (VP40) FPGA's and six banks of 256Kx72bit of external SRAM.
- (5) **CameraLink-IN** – Converts Camera-Link compliant data streams into Fabric messages. Also provides a TTL trigger output that can be used to control frame rate from the data source (i.e. Camera).
- (6) **CameraLink-OUT** – Converts Fabric messages into Camera-Link compliant data streams.
- (7) **PCI-64bit/66MHz** – Bridges between the fabric and computers with bus slots that comply with the PCI (v2.2) specification.
- (8) **Enclosure** – Provides power and cooling for up to 20 fabric boards that use the Euro-card 6Ux160mm standard form factor.

8. MESSAGE ENCODING/DECODING

For information to traverse the Fabric, it must be translated from its original form to the standardized format used by the Fabric. This process is referred to as encoding. Conversely, translating from the Fabric format to an external format is referred to as decoding. By this process, the Fabric also functions as protocol translator between external I/O. The format used by the Fabric to transport a data frame is called a message. Messages in themselves are made up of one or more packets. Packets are used to encapsulate a portion of the associated data frame. Source and destination devices operate at the message level while switch devices operate at the packet level. The key point of segmenting the data frame into packets is to allow for efficient data management and processing on the network. When there is no data to be transferred across a data link, three or more IDLE symbols are sent in order to maintain the clock alignment of the receiver. Fig. 3 illustrates the transcoding relationship between messages, packets and the associated data frame.

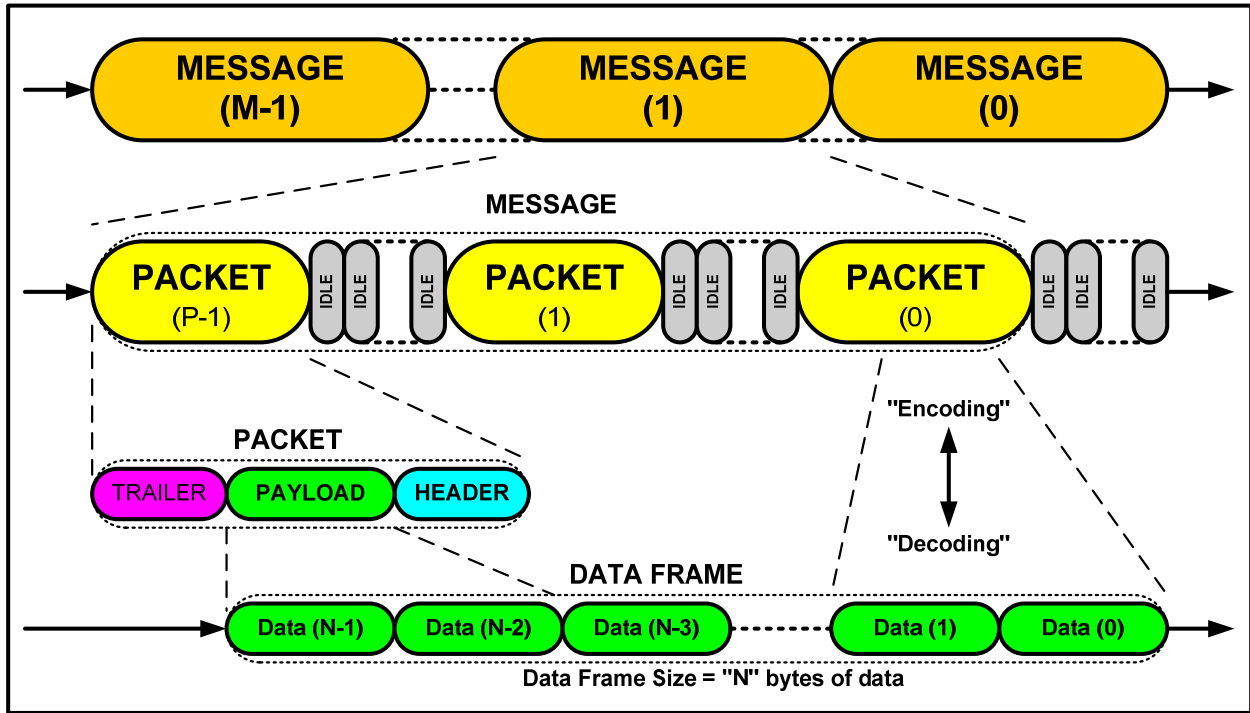


Fig. 3 – Message Encoding/Decoding Process

9. PACKET SWITCHING OPERATIONS

The heart of any network is the switch. Its purpose is to manage the transfer of multiple asynchronous message streams by routing their associated packets between its ports. Fig. 4 shows the runtime characteristics of a switch with several incoming Fabric messages entering on the left side and the resulting output message streams exiting on the right side. The messages illustrate a switch device performing the various routing methods [Direct, Multicast, Multiplex, and Demultiplex]. The ability of the switch to perform these routing methods, based on the nature of each packet, is the key to the network's communication efficiency and flexibility.

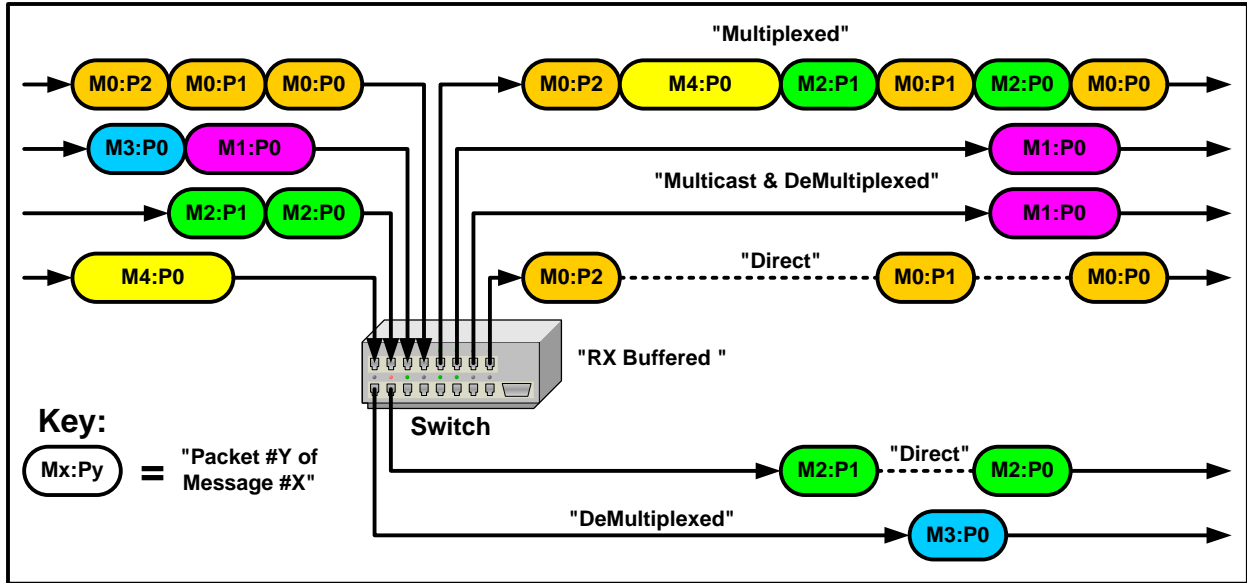


Fig. 4 – Packet Switching Operations

10. NETWORK TOPOLOGIES

When the end-node devices of a system cannot provide the connectivity required for the target application, then one or more switch devices are interconnected into a topology to achieve the required connectivity. Fig. 5 illustrates several fundamental network topologies that represent building blocks for more complex topologies. The key to optimal performance of a network is to utilize a topology that closely matches the data flow of the target application.

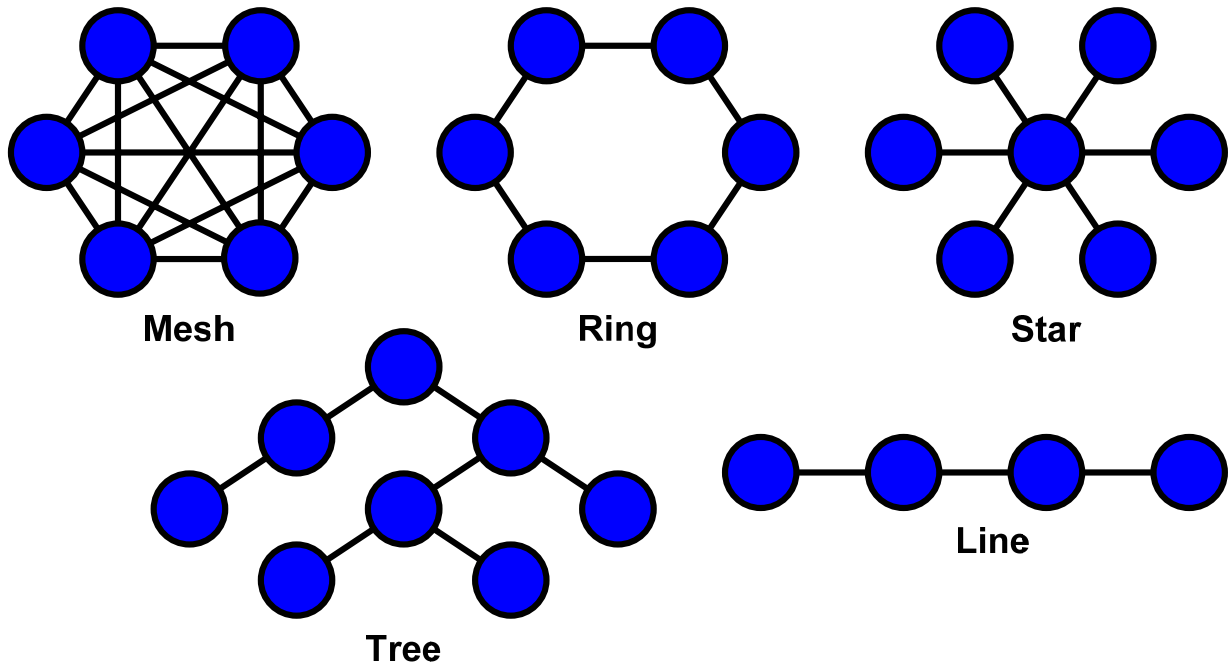


Fig. 5 – Network Topologies

11. DATA TRANSFER RATE VS. PAYLOAD

Fig. 6 shows the half-duplex data transfer performance of the Fabric protocol through a data link operating at 2.5GHz. The top curve shows the data transfer performance as a function of payload size when the packet overhead is 24bytes and packet separation is four IDLE symbols (4 bytes/symbol). The bottom curve shows the ratio (percentage) between the realized data throughput versus the theoretical maximum. As you can see by the graph, the transfer performance rapidly improves with payload size and effectively tops out for payloads in excess of 768Bytes.

Let us look at transporting a 128x128 camera image with 14-bit pixels across the Fabric. For efficient data processing, the image is encoded on the fly, during camera readout, into a message stream comprised of multiple packets. In the first case, let us have each packet contain one row = 128 pixels * 2Bytes/pixel = 256Bytes. From the graph, we see that the maximum data transfer rate achieved, per data link, would be 216MB/sec (86% efficiency). In the second case, let us have each packet contain two rows per packet (512Bytes). This gives us a max data transfer rate of 232MB/sec (93% efficiency). The result is, if the camera's readout data rate is equal to or less than these values, you will need only one data link to transfer the image with no transfer latency penalty due to local queuing of the image. Alternatively, if the camera's readout data rate is greater, then two or more data links are required to provide an aggregate bandwidth that meets or exceeds the camera's readout rate.

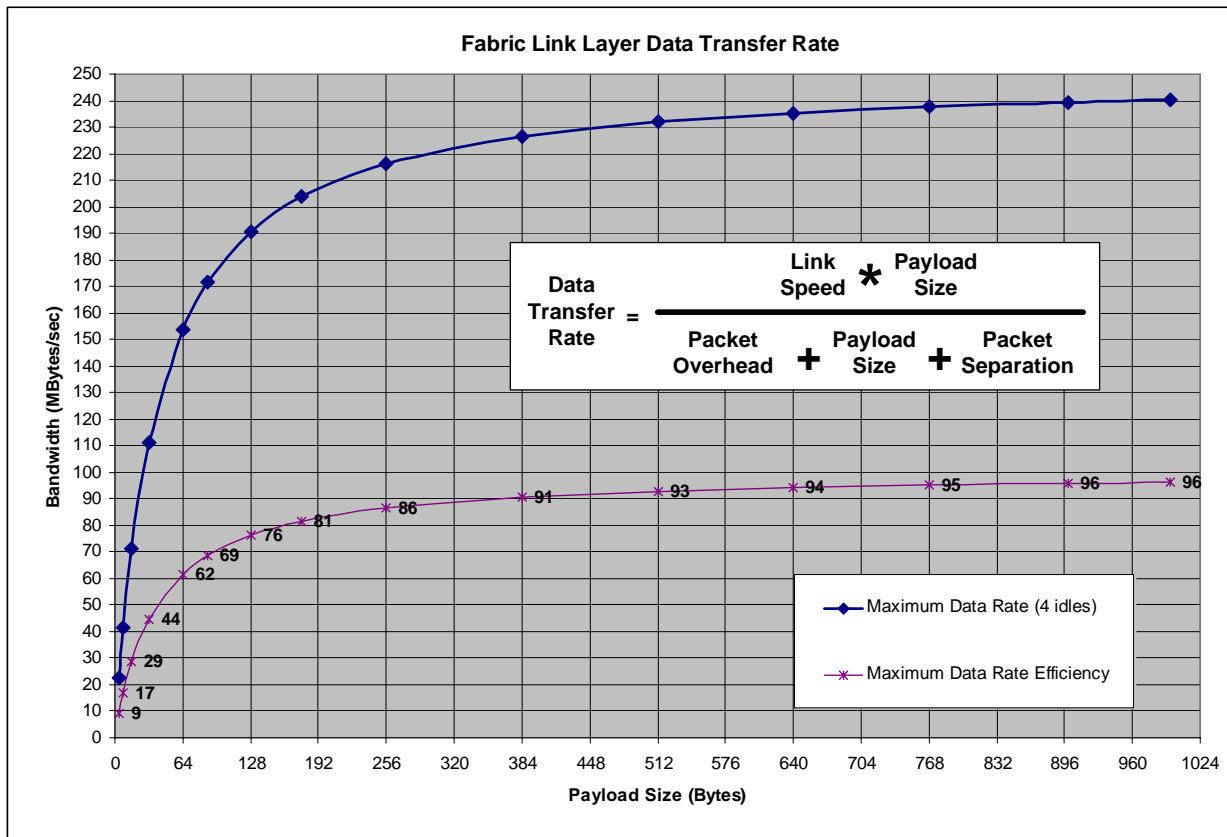


Fig. 6 – Fabric Data Transfer Rate vs. Payload Size

The method for choosing between transferring one or more pixel rows per packet is a function of what is the smallest unit of information (payload) on which the downstream processing nodes can do useful work. The more pixel rows packed together, the greater the time before the downstream device can begin processing the pixels.

12. DATA TRANSFER LATENCY & JITTER

In the example described above, the first word (pixel) transfer latency for the one row per packet case would be 1.1 microseconds (us), while the two rows per packet case would be 2.1us. The result is that, the larger payload increased the effective data transfer rate by 7%, at the cost of a 91% increase in first word transfer latency. Thus, for latency sensitive transfers, it is better to use the smallest payload size that meets your data processing requirements.

Of all of the routing methods used in a packet switched network, multiplexing is the only one that can create significant network-related latency and jitter in a system. Multiplexing is the process of routing multiple input message streams to the same output data port. For example: Message streams X, Y, and Z are received on corresponding ports A, B, C, and they are transmitted out port D. If X, Y or Z never arrives at the same time then no multiplexing latency occurs. If they do, then the packets are locally buffered (blocked) and scheduled for transfer to the port D. An optimal system topology would avoid multiplexing for critical path message flows to minimize latency and jitter, while utilizing multiplexing for diagnostic and/or low-bandwidth message flows, to save network resources.

Table 1 shows the measured data transfer latency and jitter overhead associated with message encoding, decoding, or routing on each of the Fabric devices developed. The numbers in green represent the lowest recorded values while the numbers in red represent the highest recorded values. As we can see in the table, the Fabric protocol introduces less than 1us of latency and less than 100 nanoseconds (ns) of jitter for each device that a message traverses.

Table 1 – Fabric Data Transfer Latency & Jitter

Fabric Timing	Min (ns)	Max (ns)	Jitter (ns)	Notes
Switch (8port)	416	512	96	Non-Blocked Transfers
RS422(OUT)	640	672	32	Packet Decoding
RS422(IN)	192	208	16	Packet Encoding
Camera Link (IN)	208	256	48	Packet Encoding
Camera Link (OUT)	672	720	48	Packet Decoding
Digital IO (OUT)	608	640	32	Packet Decoding
Digital IO (IN)	192	208	16	Packet Encoding
Processor (FPGA)	640	720	80	Packet Decoding->Encoding

13. AO SYSTEM DESCRIPTION

Fig. 7 shows the High Performance AO system implemented at the Starfire Optical Range (SOR), which utilizes the Fabric Architecture. This AO system was designed to advance the research and development of adaptive optics in strong scintillation. The system, utilizes a Self Referencing Interferometer (SRI) as its wave front sensor (WFS) and a Woofer-Tweeter (W-T) Deformable Mirror (DM) as its wave front corrector. The SRI WFS was chosen for its superior error rejection of scintillation-induced errors in the wave front measurements that would normally occur using a Shack-Hartmann WFS. The W-T DM geometry was selected to provide the required spatial resolution (Tweeter) and stroke (Woofer) to reconstruct the phase conjugate of a wave front with branch-point constructs.

The SRI utilizes an IR camera capable of acquiring a Region Of Interest (ROI) of 160x32 at 10 KHz with 31us of readout latency. The pixel dynamic range is 14-bits. The Camera's ROI provides four 32x32 spatially phase shifted (0, 90, 180, 270 degrees) snapshots of the wave front fringe pattern (one pixel/sub-aperture). These four phase shifted images are used to compute the phase error as a 2pi modulo phase value. The Tweeter DM has 1396 actuators, of which 705 are masters, 236 are slaves, and 455 hardwired to mid-scale. The Woofer DM has 577 actuators, of which 177 are masters, 399 are slaves, and one is hardwired to mid-scale. The Tweeter-Woofer actuator mapping is 2:1. The WFS required one CameraLink (IN) adapter to provide acquisition of a 400MB/s pixel stream. The DMs required nine RS422 adapters to provide 18 parallel output ports (16bits/port) with an aggregate DM command transfer rate of 720MB/s.

The data processing operations performed per camera frame are:

- 1) Calibrate Camera pixels (Gain, Offset, and Threshold) [5,120 pixels].
- 2) Calculate 2pi-phase from phase-shifted images [1024 2pi-phases].
- 3) Calculate Tip/Tilt/Piston correction and correct 2pi-phase elements [1024 2pi-phases].
- 4) Exponential Servo Calc (first order leaky integrator – modulo) [1024 2pi-phases].

- 5) Calculate 2pi-gradients from 2pi-phase elements [2048 2pi-gradients].
- 6) Calculate unwrapped-phase using a Least-Squares-Reconstructor (LSR) [1024x2048]*[2048x1].
- 7) Calculate Tweeter Tip/Tilt correction of unwrapped-phases [705 phases].
- 8) Perform Tweeter Tip/Tilt correction of unwrapped-phases [705 phases].
- 9) Calculate and Add Branch-points to unwrapped-phases [705 phases].
- 10) Tweeter DM calibration (Gain and Offset) [941 commands].
- 11) Woofer Servo Calc (first order leaky-integrator – standard) [177 phases].
- 12) Woofer Tip/Tilt correction [177 phases].
- 13) Woofer DM calibration (Gain and Offset) [576 commands].

Processing stages (1) through (4) are performed in the SRI Processor Node (PN). Processing stages (5), (6), (8), (9) and (10) are distributed across PN (RTR-0) through (RTR-7). Processing stages (7), (11), (12) and (13) are performed on PN (AUX).

Data acquisition and simulation is handled by six server class computers (DAS) which are directly connected to the Fabric (one data link per server). These computers are used to capture data streams for post-analysis and live displays. Additional standard desktop PCs (Console) are used to monitor and control the AO system over Ethernet. All data is synchronously time tagged via time stamp messages generated by the GPS+TTL device. This device also generates synchronous and phase aligned trigger messages used to initiate camera integration and readout every 100us (10 KHz).

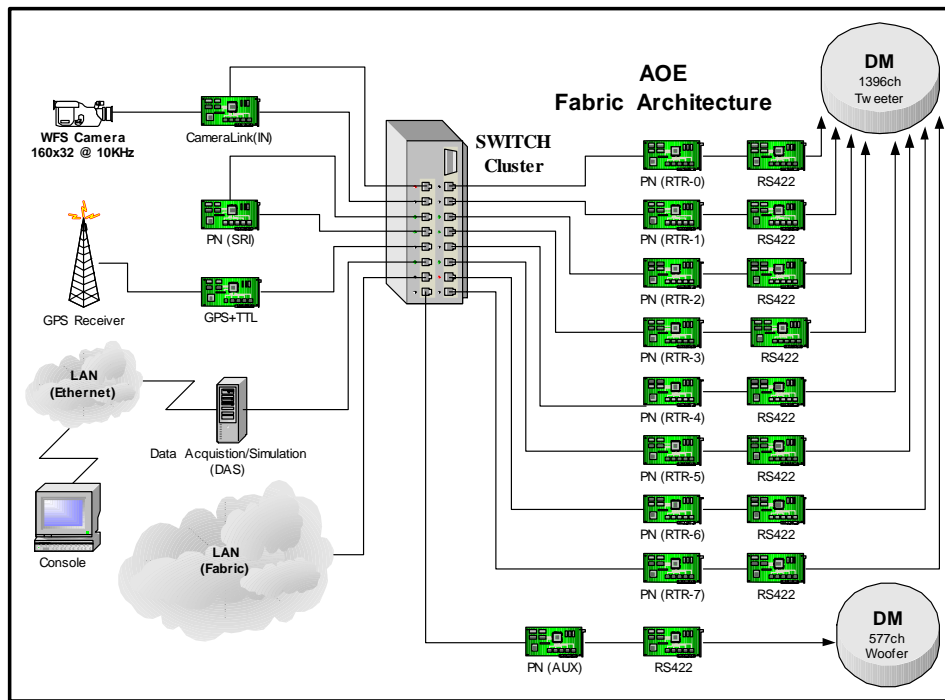


Fig. 7 – SRI-WT AO System using the Fabric Architecture

14. AO SYSTEM PERFORMANCE – LATENCY & JITTER

Fig. 8 shows the measured worst-case latencies for each process performed in the system as it executes the AO algorithm. The top line shows that the target latency budget for this system is 100 microseconds (us). The WFS Camera (30.7us) and DM drivers (HVA-G1 @ 33.5us) dominate the latency budget for the Tweeter portion of the servo-loop. The Tweeter’s algorithmic overhead was 21.1us while the Woofer’s algorithmic overhead was 33.8us. Measured Tweeter latency was 85.3us with a jitter of 0.5us for a worst-case latency of 85.8us. Measured Woofer latency was 69.3us with a jitter of 4.7us for a worst-case latency of 74.1us. Measured Woofer jitter was due to the multiplexing of eight unwrapped phase message streams from the RTR-(0..7) devices into the AUX device for computing Woofer DM commands and Tweeter Tip/Tilt correction. Thus, the measurements show that the system successfully meets the latency budget requirement of 100us with a 14.2us margin.

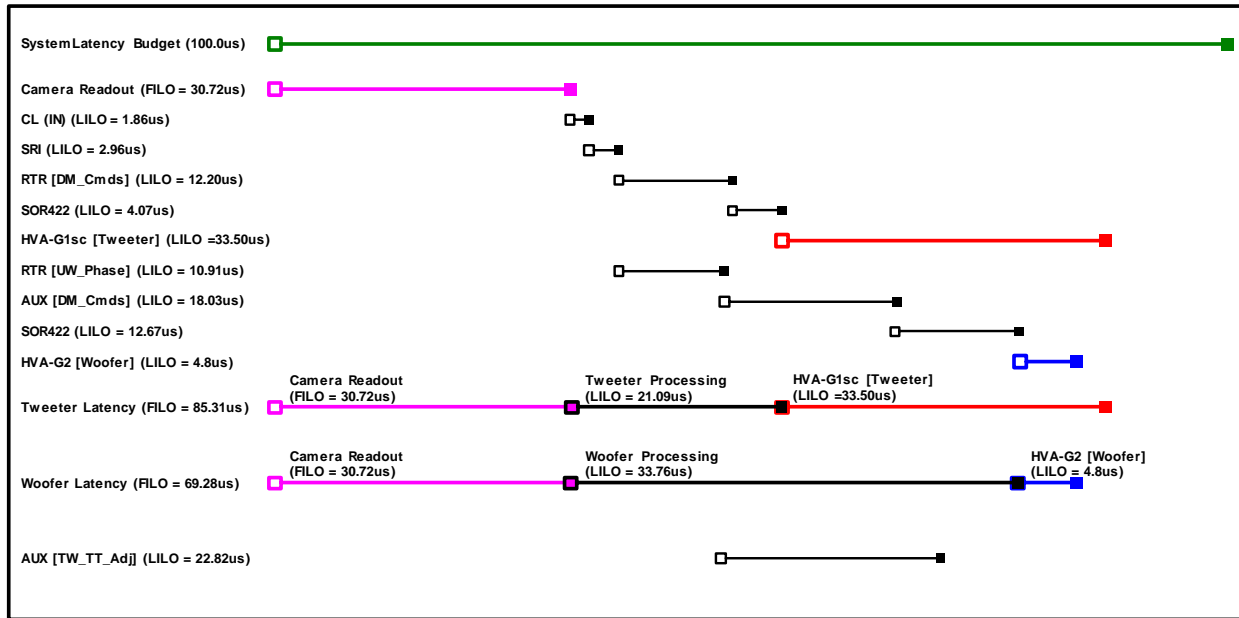


Fig. 8 – SRI-WT AO System Latency Chart

15. SUMMARY

The results of the SRI-WT AO system development demonstrates that a high performance AO system can be successfully created using a Fabric architecture that utilizes a topology optimized for the application.

The inherent nature of packet switched networks benefit systems engineering by providing a highly deterministic and low latency communication environment that is easy to model, configure, build and evolve as requirements change. Moreover, because of the inherent extensibility of this technology, multiple AO sub-systems can be integrated into a larger network to form a Multi-Conjugate AO (MCAO) system spanning one or more telescope mounts.

16. REFERENCE

1. PCI-SIG, PCI, Rev 2.2, 12/1998: Peripheral Component Interconnect (PCI) Local Bus Specification.
2. PICMG 2.0, R3.0, 8/1999: CompactPCI (cPCI) Specification.
3. ANSI/VITA 1-1994 (R2002): Virtual Machine Environment (VME) bus Standard.
4. ANSI INCITS 230-1994/AM2-1999: Fibre Channel Physical and Signaling Interface (FC-PH).
5. ANSI/VITA 17.1-2003: Serial Front Panel Data Port Specification.
6. IEEE Std 802.3-2002: Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.
7. RapidIO™, Rev 1.3, 6/2005: RapidIO™ Interconnect Specification, Part 6: 1x/4x LP-Serial Physical Layer.
8. IBA-v1 & v2, Rel 1.2, 8/2004: InfiniBand™ Architecture Specification Volume 1 & 2.