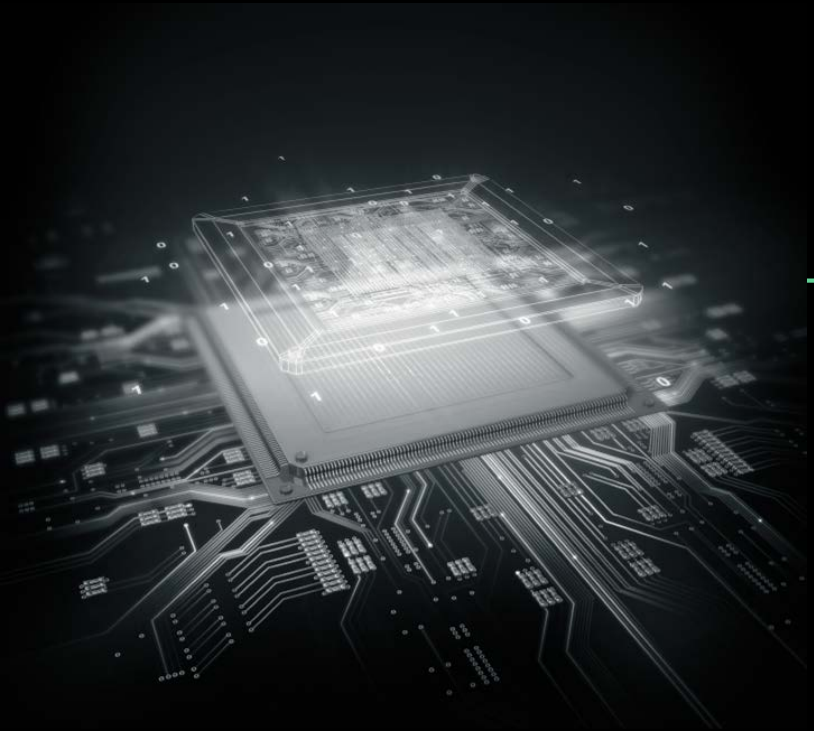
AXIoE

Advanced eXtensible Interface over Ethernet



Dr William Kamp
Senior Research Officer (Engineer)
High Performance Computing Research Lab
AUT

MID Central Beam Former

World’s largest - high performance - low power - distributed - embedded system.

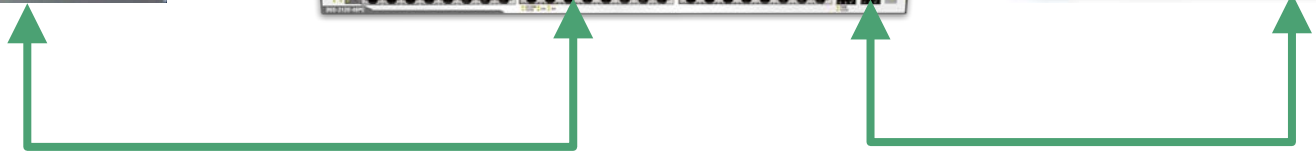
Thousands of FPGA supported by hundreds of microcontrollers.

Need a way to configure the FPGA datapath for the various bands/modes.

Need to monitor the health of the subsystems.

FPGA Application Monitor and Control (AMC)

Extending the memory bus
(AMBA AXI) on an FPGA,
via Gigabit Ethernet,
to the Monitor and Control Application.



Design Goals

Simplify Application Programming

Resource efficient implementation in FPGA

Efficient bulk data transport

Reliable and in-order access to the FPGA memory space.

Server (FPGA side) is kept simple.

Client (Application side) holds the complexity.

Traditional Transport Protocol Options

	Simplify AMC Software (Reliable Transport)	Server-(FPGA)-Side Resource Efficiency (Simplicity)	Network Efficiency (Low Overhead)
TCP	 Error recovery built in.	 Complex sliding window algorithm	 16 byte header
UDP	 No error recovery	 Simple packet based flow	 8 byte header

Introducing AXIoE

A new transport protocol, to meet my design requirements.

AXI - Advanced eXtensible Interface

- ➢ a memory bus specification from ARM.
- ➢ supported by both Xilinx and Altera (via Qsys).

o - over

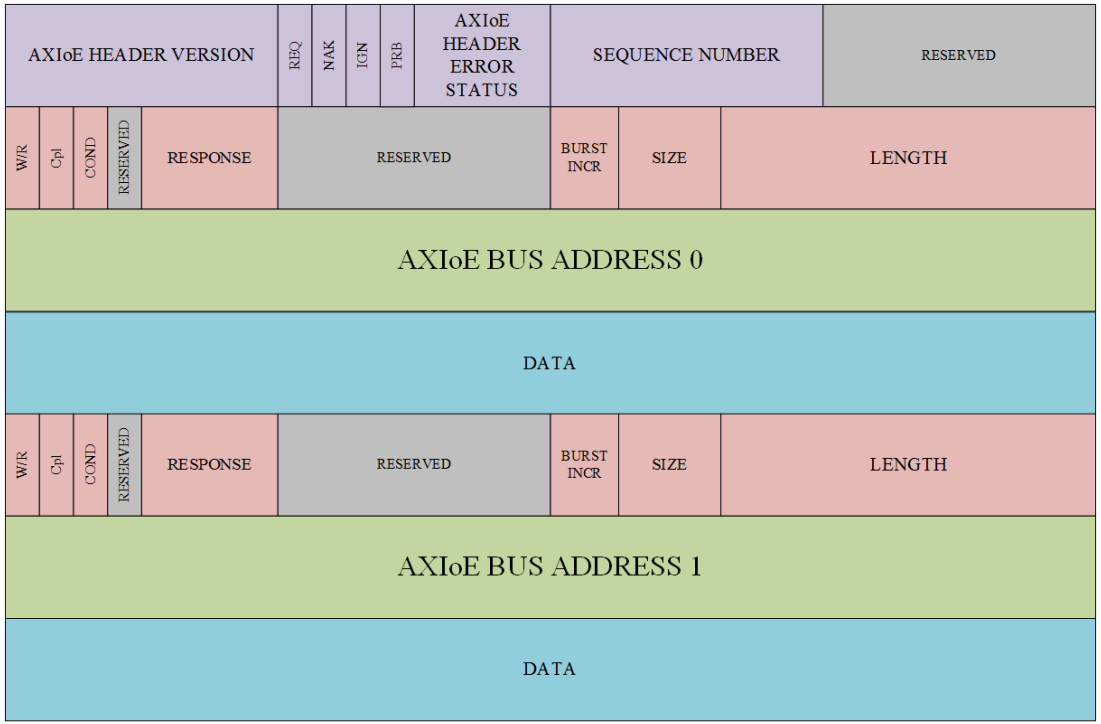
E - Ethernet

Introducing AXIoE

Simplicity is key.

1. Built on UDP.
2. Adds a simple “sliding window algorithm”.
3. Designed for low resource usage at the embedded device (server).
4. Encapsulates *AMBA AXI* bus transactions (Read/Write blocks of memory).
5. Can stack multiple *AXI* transactions into each packet.

Packet Format (UDP Payload)



Global Packet Header

Transaction Header

Transaction Address

Transaction Data

Another Transaction

Server - Designed to be cheap.

Simple behaviour:

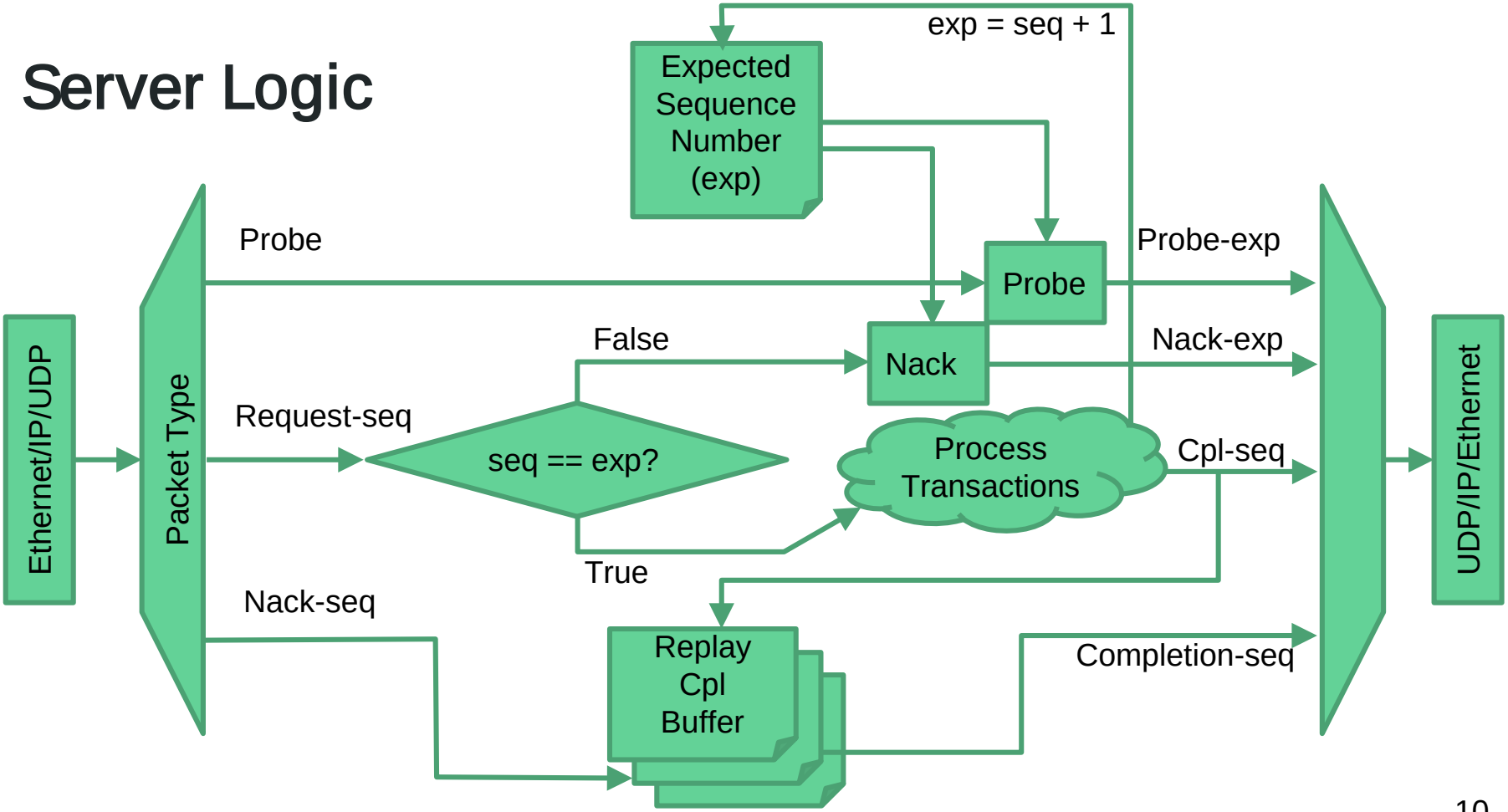
- One packet in - One packet out.
- One decision fork.

Requires memory to store only a single packet.

- More memory = bigger window = better performance.
- Half the memory of TCP.

Client responsible for enacting all **error recovery** behaviour.

Server Logic



AXIoE Protocol

Client - Server protocol

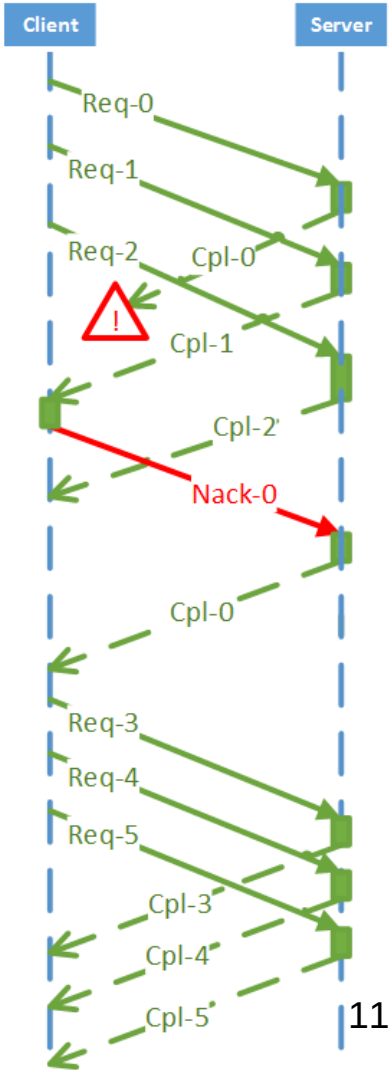
Sliding window algorithm for performance and error recovery.

Uses an 8-bit packet sequence number.

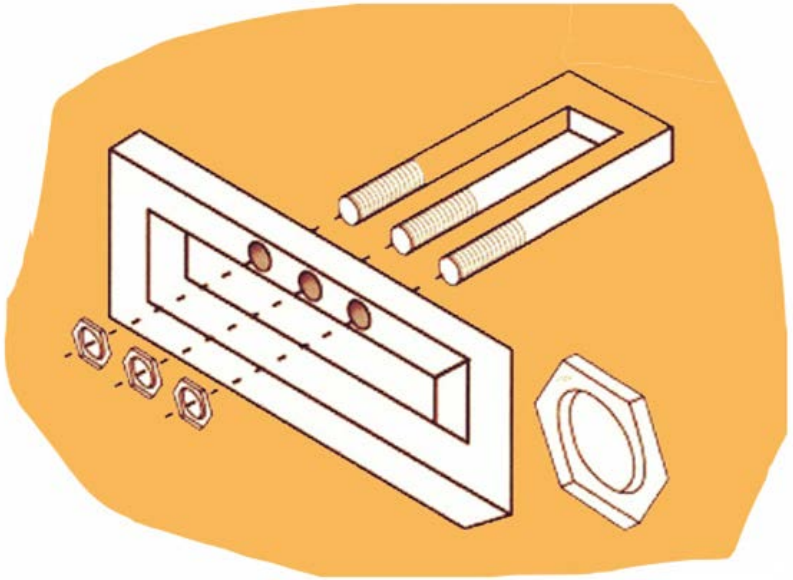
3 types of packets

- with a common header format for easy parsing.

Client to Server	**Server to Client**
Request (Req)	Completion (Cpl)
Negative Acknowledge (Nack)	Negative Acknowledge (Nack)
Probe (Prb)	Probe (Prb)



Where we are at ...



AXIoE base specification is in draft.
Prescibes the

1. **Packet formats**,
2. **Server behaviour**, and
3. **Client responsibilities**.

The client's error recovery behaviour is not formally unconstrained, but a recommended procedure is detailed.

AXIoE Remote Procedure Call (RPC) specification is in draft.

- Details a method to map function calls onto AXIoE transactions.

Where we are at ...

Server implementations written in

Python	William Kamp	AUT
Java	David Del Rizzo	NRC
C *(for ARM)*	Robert Chapman III	NRC
VHDL	William Kamp	AUT

Client implementations written in

Python	William Kamp	AUT
Java	David Del Rizzo	NRC
C	Emanuele La Rosa	Selex-ES

OSI Model

AMC / FPGA

AXIoE-transactions

UDP + AXIoE-protocol

IP

Ethernet

OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example		Central Device/ Protocols			DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP	**GATEWAY** Can be used on all layers		Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT			
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names			
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP			Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP			Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch Bridge WAP** PPP/SLIP	Land Based Layers		Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**			

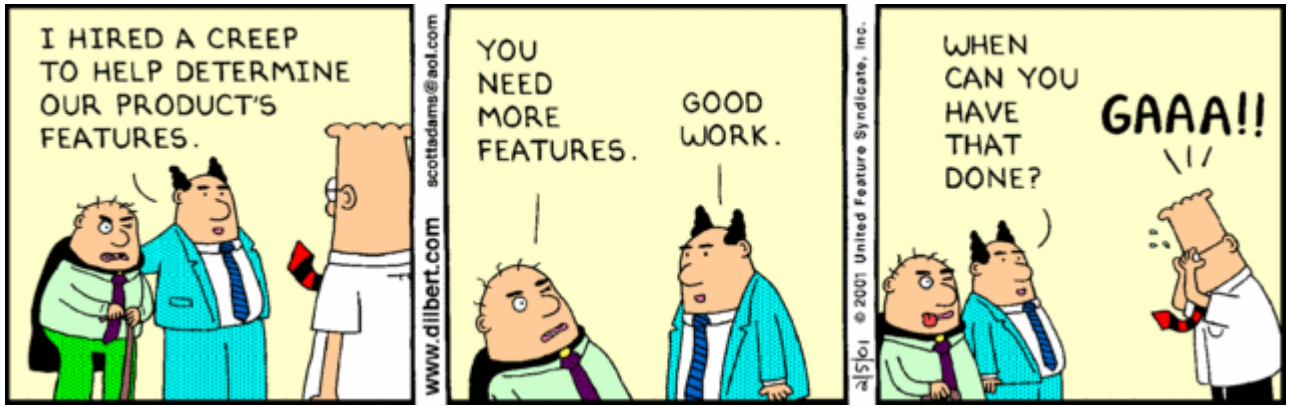
Feature Creep

IGN bit - disables the packet sequence number check. Reverts error recovery to UDP.

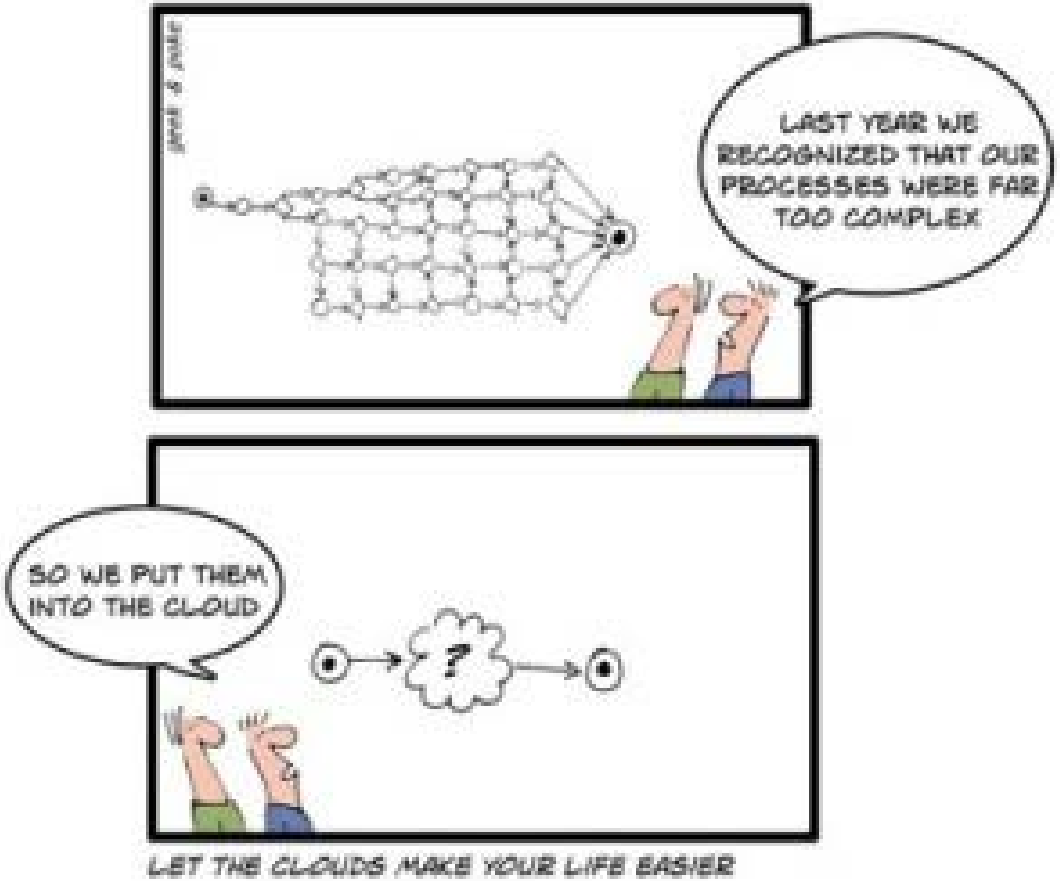
Useful for implementing MSI interrupts, doing other bad things, …

COND bit - conditionally execute a transaction based on the success of the previous.

If an error occured, don't try with this transaction either!



Questions?


LAST YEAR WE RECOGNIZED THAT OUR PROCESSES WERE FAR TOO COMPLEX
SO WE PUT THEM INTO THE CLOUD
?
LET THE CLOUDS MAKE YOUR LIFE EASIER