Science & Technology Facilities Council
Rutherford Appleton Laboratory

A subset of ATA over Ethernet as the control protocol in xTCA

Particle Physics

David Sankey and Tim Durkin, ATLAS and CMS groups, Rutherford Appleton Laboratory

Introduction

Both ATLAS and CMS are investigating xTCA as a replacement for VME for use in upgrades to their triggers
- ATLAS L1Calo are considering ATCA
- CMS are considering double width μTCA

Both crate standards provide Gigabit Ethernet on the Base Fabric for module configuration and control

A generic solution for module control based on Ethernet could simplify configuration and control in the new environments

Conventional Ethernet solutions have to balance a trade-off between performance, complexity and reliability

A subset of the Storage Area Network protocol ATA over Ethernet promises simplicity, performance and reliability and could therefore be ideal

ATCA crate

μTCA crate

Ethernet on FPGA

Implementations of Ethernet on FPGA vary in complexity and functionality between TCP/IP and raw Ethernet sockets

Each have different advantages and disadvantages:
- TCP/IP is connection based and reliable, but incurs the highest overheads, both on the FPGA and on the host processor
- UDP/IP has reduced overheads but is not reliable
- raw Ethernet sockets have significantly reduced overheads, giving potentially twice the throughput compared to TCP/IP, but again are not reliable and moreover require root privileges on the host

Solutions in use tend to be based on UDP/IP, typical examples including:
- commercial QuiXstream protocol
- existing implementation in CMS IPbus protocol

Storage Area Networks

With the move away from dedicated networks such as Fibre Channel to Ethernet, Storage Area Networks have encountered the same compromise between performance and reliability

In general this has been resolved with custom hardware:
- 'Converged Ethernet' for Fibre Channel over Ethernet
- TCP/IP offload engines for iSCSI

One implementation has however achieved performance and reliability with commodity hardware and networks, namely ATA over Ethernet

Further reading:

ATA over Ethernet <http://www.coraid.com/TECHNOLOGY/What-is-AoE>
aoetools including vblade <http://aoetools.sourceforge.net/>
HDMC <http://cdsweb.cern.ch/record/479712/files/p464.pdf>
IPbus <http://projects.hepforge.org/cactus/trac/browser/trunk/doc/related/Simple_IP_uTCA_Protocol_r1_2.pdf>
QuiXstream <http://www.tekmicro.com/products/product.cfm?gid=5&id=51>

ATA over Ethernet

ATA over Ethernet (AoE) is an open-standards-based protocol that allows direct network access to disk drives by client hosts
- it is a Layer 2 protocol encapsulating ATA directly in Ethernet frames
- the source is released under the GNU GPL
- it is native in the Linux kernel since 2.6.11

It claims to "deliver a simple, high-performance, low-cost alternative to iSCSI and Fibre Channel for networked block storage by eliminating the processing overhead of TCP/IP"
- it achieves reliability by the host keeping track of packet acknowledgements leading to a particularly simple implementation on the disk server
- it achieves performance by avoiding the nested memory to memory copies implicit in the default implementation of TCP/IP

It is a stateless protocol which consists of request messages sent to the AoE disk server and reply messages returned to the client host, again leading to a particularly simple implementation on the disk server

AoE subset for xTCA

AoE messages have two formats:
- ATA messages
- Config/Query messages

AoE Config/Query messages are very interesting as a means of module control:
- they are inherently simple
- they are designed to be flexible in order to allow extra uses
- all the complexity involved in providing reliability is handled by the host
- user-mode tools exist avoiding the need for root privileges on the host

Encapsulating a protocol such as the CMS IPbus command set in an AoE Config/Query message can provide a simple and reliable way of implementing module control over the Base Fabric in ATCA and μTCA

Finally device discovery is supported by means of broadcast packets

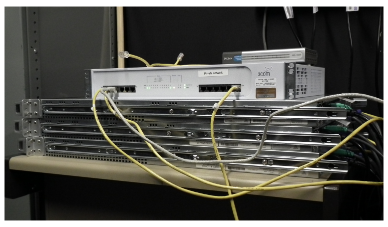
Work at RAL

The work at RAL is based on incrementally implementing this subset of the AoE protocol on a Xilinx development board as part of an AoE system on a private managed network:

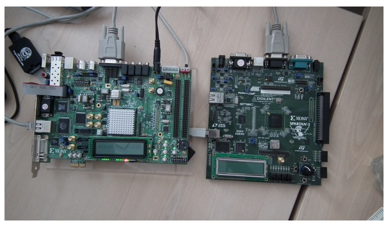
i) set up standard AoE using vblade on Linux
- model and sniff device discovery on Ethernet
- implement AoE device discovery on development board

ii) progress to AoE Config/Query
- model and sniff config/query with aoetools and vblade
- implement on development board

iii) implement AoE as a communication layer in existing protocols
- implement CMS IPbus command set as an alternative to the existing UDP/IP version
- investigate implementation compatible with ATLAS HDMC software layer



Linux nodes and managed switch



Xilinx development board