IPbus Performance and Robustness Tests for the 904 Test Stand in Summer 2013

version 2013.05.07

Marc Dobson, Marc Magrans de Abril, Tom Williams

Open Issues:

- **Kristian: How the firmware upload and load will work**
- **Marc: Is it possible to configure a 2 Gb/s link between bridge PC and switch**

1 Introduction

This document describes the IPbus integration tests that will take place during the Summer of 2013 in building 904.

1.1 Terminology

- ***Bandwidth (BW).*** The amount of data transferred or received per unit of time, typically expressed in bits per second (bit/s). In general, when this document refers to bandwidth it does not include the protocol overhead, but just the effective bandwidth.
- ***Latency.*** The time taken for μHAL client to perform an IPbus transaction. This should be measured in the μHAL client PC, starting from first corresponding `uhal::read/write` call, and stopping when the `dispatch()` method returns.

See also the terminology sections of the *IPbus Protocol* and *IPbus Network Architecture* documents.

1.2 References

CAEN (V2718) Bridge Performance Tests using the HAL library, http://cmsdoc.cern.ch/~cschwick/VME/html/VMEBridges_CAEN_Performance.html

IPbus Protocol version 2.0,
https://svnweb.cern.ch/trac/cactus/browser/trunk/doc/ipbus_protocol_v2_0.pdf
IPbus Network Architecture,
https://svnweb.cern.ch/trac/cactus/browser/trunk/doc/uHAL_network_addressing.pdf

1.3 Important points

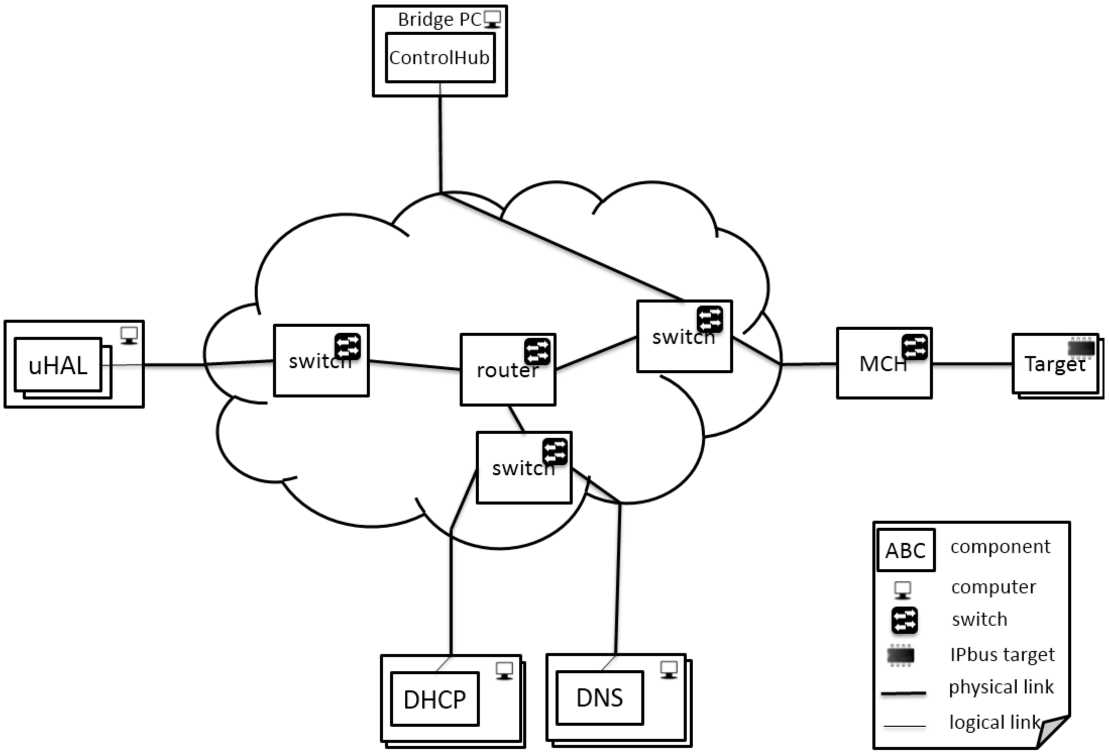
- The basic test executables used should be packaged up in the IPbus software suite release. This will allow the tests to be reproduced in the future.
- The Control Hub will be the component controlling the number of packets in flight (rather than making several different firmware images). Ideally, this will be a configurable parameter.
- The firmware used in the tests must correspond to an SVN tagged version.
- This document does not cover testing of IP address assignment using RARP, since the RARP firmware functionality will not be available until after a few weeks after the rest of the IPbus firmware functionality. However, this will of course be tested once the RARP functionality is available.

1.4 Document Structure

This document is divided into 6 sections. Section 2 describes the physical topology of the μTCA test stand and 904 network. Section 3 describes the basic performance measurements. Section 4 describes a block-write soak test of the system, along with measurements of the performance degradation from artificially-induced packet loss. Tests to simulate network congestion, and measure any resulting performance drop are then covered in section 5. Finally, section 6 lists tests to simulate system or component failures, and validate the corresponding recovery procedures.

2 904 Network Topology

The following diagram below shows the physical topology of the 904 network where the IPbus integration tests will take place.



It should be noted that the *Bridge PC* is connected to the same switch as the MCH, and the other PC is attached to a different switch, with each of these switches connected to one router (which also connects the 904 test stand PCs to the general 904 LAN). This setup is designed to replicate the eventual network topology at Point 5, as documented in the *IPbus Network Architecture* document.

Additionally, both PCs will be able to communicate directly with the boards via Ethernet. This will allow us to check that the IPbus-UDP traffic can travel through a router without any errors.

N.B. In order to allow 1Gb/s IPbus communication to/from the crate, the link from the Bridge PC to the switch must be 2Gb/s in each direction, and all other links between PCs and crate must have 1 Gb/s capacity.

N.B.2 According to current plans, there will be 4 GLIB boards in the μTCA crate.

3 Performance Measurements

3.1 Latency

Firstly, the average latency and the number of transactions per second should be measured for:

- The ping command
- Single-word reads and single-word writes via direct UDP
- Single-word reads and single-work writes via the ControlHub

These measurements could then be repeated for multiple uHAL clients, resulting in the following graphs:

- Latency as a function of the number of clients.
- Transactions per second as a function of the number of clients.

3.2 Bandwidth

Two sets of measurements need to be performed:

1. ***Single µHAL client talking to one board***

   Plot bandwidth as a function of depth for block writes. Several lines on the graph for direct UDP, and via ControlHub with *n* packets in-flight for n=1, 2, 4, 6, 8 and so on.

   From this graph, we'll establish the 'default' number of packets in-flight as the number that causes the 1Gb/s link to be saturated (for depths that result in all write-request UDP packets being full).

2. ***N µHAL clients talking to 1 or N boards***

   Plot the block write bandwidth (both total, and average per client) as a function of the number of clients for:

   i. Each client talking to the same board
   ii. Each client talking to a different board

   Here, we'll use the number of packets in-flight and write depth that maximizes the bandwidth in the previous "1 client to 1 board" measurements

   Error bars will be plotted for the average bandwidth per client showing the lowest and highest bandwidth experienced by the N clients, in order to show the full range of bandwidths seen by different clients (i.e. to check that each client slows down by the same amount).

   Additionally, the CPU and memory utilization of the ControlHub should also be plotted versus the number of clients.

N.B. Block writes (rather than reads) are used here in order to measure bandwidth of data flow to/from the boards under good conditions. This is because for writes the `return` stream of data back to the Bridge PC is much smaller than the dataflow to the boards, and hence there should not be any build-up of packets in the MCH switch queues. The bandwidth for block reads is plotted in the congestion tests section.

4 Reliability Tests

4.1 Soak test

Perform continuous block reads from 4 clients (via ControlHub), each talking to a different board, such that at least 10^9 UDP packets flow through the system. Check that system is 100% reliable from end-user perspective (i.e. µHAL applications don't crash) and record number of lost/malformed packets. The number of 10^9 packets was motivated by the bit error rate of modern Ethernet being at most the order of 10^{-11} [M. Magrans].

4.2 Artificially-induced UDP packet loss

Plot average latency for single-word reads and bandwidth for large block writes as a function of the IPbus UDP packet loss rate (from 0.01% up to 1%); this random packet loss is artificially-induced by the iptables command.

These measurements are intended to check that the packet-loss recovery mechanism (and its implementation) are properly understood. The measured latency should be compared with the prediction of:

$$L' = L + pR \quad \text{for} \quad p << 1$$

where p is the probability of UDP packet loss, L is the latency, and R is the time to detect a dropped packet (timeout) plus the IPbus status request-reply latency.

5 Congestion Tests

In the case of block reads from all boards in a μTCA crate, congestion-induced packet loss could occur on the board-side of the MCH switch since the reply packets that the boards send back to the PCs are much larger than the original request packets. Whether or not packet loss occurs in this scenario will depend on parameters such as the packet queue size in the MCH switch, the number of boards in the crate, and the number of packets in-flight to each board.

Therefore, to check for performance degradation in this scenario, we will repeat the "N clients to N boards" bandwidth measurements – i.e. plotting the total bandwidth versus N – but for block reads this time, and check that there isn't any sudden performance drop as the number of clients is increased. Here, the level of packet loss can also be checked from the controlhub_stats commands.

6 System Recovery Tests

The detection mechanism, and the recovery procedure and time of the following situation must be documented:

- Replacement of the MCH
- Replacement of an AMC board
- MCH not connected to the switch
- AMC board not plugged
- AMC firmware reload
- Check that the AMC is not reachable from the uHAL client PC