IPbus Network Architecture for µTCA Hardware

version 16th April 2013

Marc Dobson, Carlos Ghabrous, Marc Magrans de Abril, Dave Newbold, Christoph Schwick, Tom Williams

1 Introduction

1.1 Terminology

- ***MicroTCA Carrier Hub (MCH)***. The central management and data-switching device in a MicroTCA (µTCA) system.
- ***IPbus protocol.*** IP-based protocol for controlling hardware devices.
- ***IPbus target.*** The hardware device that responds to – and is controlled by – IPbus protocol requests from an IPbus client. The IPbus target is an endpoint of an IP network.
- ***IPbus client***. The software client that generates IPbus transaction requests to control an IPbus target device.
- ***µTCA Hardware Access Library (*****µHAL*****) client***. The µHAL library implements the IPbus protocol on different transport protocols (i.e. UDP, TCP, and TCP to the Control Hub).
- ***Control Hub.*** An IPbus client that serializes the concurrent access to each IPbus target from one or more µHAL clients. The Control Hub receives requests from µHAL clients, and forwards the request to the IPbus target; there can only be one Control Hub communicating with any given IPbus target at any time.
- ***Experiment Control Network (ECN).*** The local area network used for control and monitoring of the experiment.
- ***Bridge Computer.*** A computer that connects the Experiment Control Network with the MCH (and thereby with the IPbus target). A Bridge Computer can host the Control Hub, the RARP daemon, or the IPMI daemon. A bridge computer can also host an IPMI based service to provide slow control access to the Detector Control System (DCS).

- ***System Administrator***. The person or persons responsible for the administration of the network, its basic services (i.e. DNS, DHCP, RARP, Kerberos, etc.), and the computers attached.
- ***Hardware Expert.*** The person or persons maintaining a specific hardware or firmware.
- ***On-call Expert.*** The on-call person or persons responsible for immediate response to any incidences on a specific IPbus target. These persons should be able to diagnose a problem, take all possible corrective actions, and then if needed, forward the incidence to the system administrators, hardware experts or the IPbus protocol experts.

1.2 References

[IPBUS2.0] IPbus protocol version 2.0, https://svnweb.cern.ch/trac/cactus/browser/trunk/doc/IPbus_protocol_v2_0.pdf

[uHAL] µHAL Library Quick Tutorial, https://svnweb.cern.ch/trac/cactus/wiki/uhalQuickTutorial

1.3 Overview

This document describes the network topology, addressing scheme, and IP configuration procedure for a network using the IPbus protocol on a µTCA architecture.

The IPbus network architecture is based on known and proven technologies. The following differences can be observed with respect to typical network architecture:

1. ***The IPbus targets are only reachable through a Bridge Computer.***
   The existence of Bridge Computers is a consequence of the IPbus protocol requiring a unique client (Control Hub) per IPbus target to achieve perfect reliability. The Control Hub serializes the concurrent access of several µHAL clients to a single IPbus target in a reliable way.

2. ***The default IP configuration protocol for the IPbus targets is RARP. However, the document leaves open the possibility of configuring the IP addresses using IPMI.***
   The use of RARP is the consequence of a trade-off between the additional complexity required to implement DHCP in the IPbus firmware, the simplification of the system administration procedures (with respect to manually configuring the IP addresses consistently), and the cost of developing a non-standard IP configuration protocol.

1.4 Document Structure

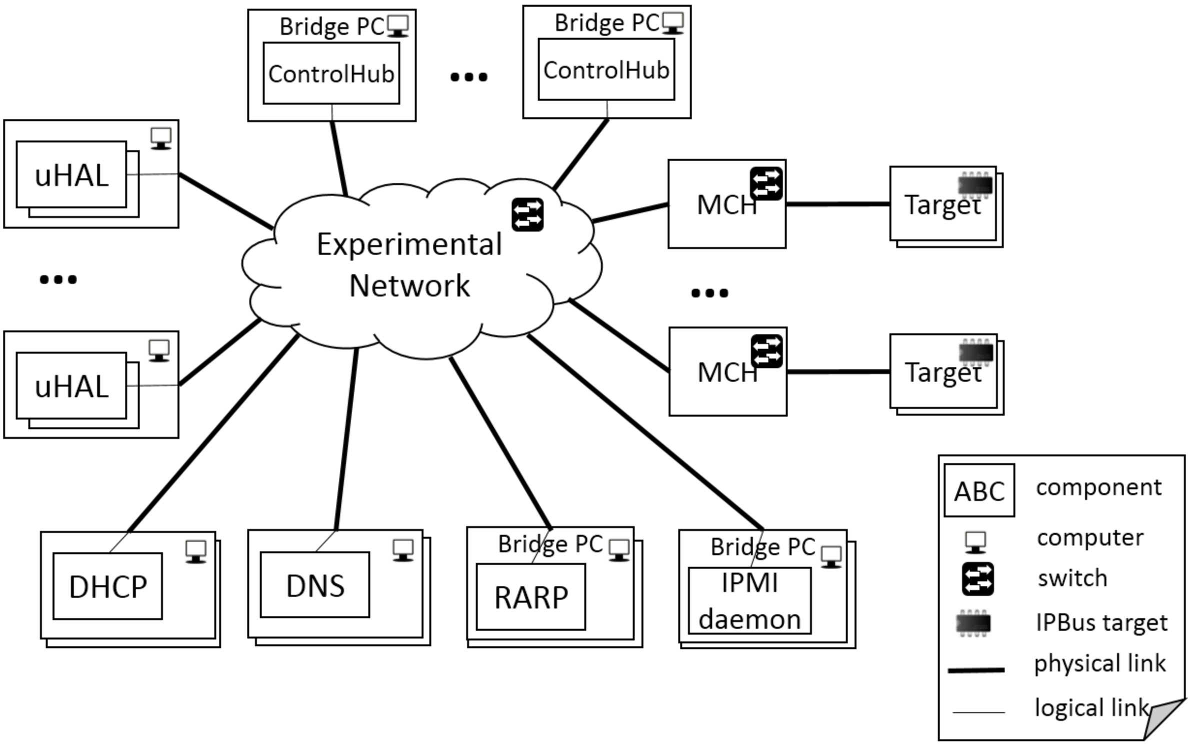
The document is divided in five sections. Section 2 describes the overall network topology. Section 3 briefly describes the naming and addressing scheme. Sections 4 and 5 describe the IP configuration mechanism for the MCH and the IPbus targets, respectively. Finally, section 6 describes some typical fault-free and fault-recovery scenarios.

2 Network Topology

The proposed network topology has been designed to simplify the installation and the recovery of physical components (i.e. PCs, MCHs, and IPbus targets) and logical components (i.e. µHAL clients, and Control Hub instances). From a physical point of view, all the components interacting in the IPbus network are equivalent – they are all plugged to the ECN.

However, from the logical point of view the network is segmented. The IPbus targets and MCHs can only be reached from a subset of computers – referred to as the Bridge PCs. Therefore, the Control Hub instances, the RARP daemons, the IPMI daemon, and the DCS-IPMI service should be hosted in one of the Bridge PCs.

The System Administrators are responsible for setting up the correct segmentation of the network.



3 Addressing and Naming Scheme

3.1 MAC Addresses

Each device (i.e. computers, MCHs, and IPbus targets) must have a globally unique MAC address, and must be labelled with this MAC address. It is the responsibility of the hardware expert to set the device MAC address on non-volatile storage before installation.

3.2 IP Addresses

All the devices in the network must have different IP addresses. All the IP addresses are public within the domain of the ECN. This means that from any point within the network the DNS name of any Bridge PC, MCH, or IPbus target can be translated to its corresponding IP address.

Notice that this does not imply that the IPbus targets are reachable from any node of the ECN. On the contrary, as explained earlier the ECN is segmented, and the IPbus targets can only be reached from the Bridge PCs.

The selection and assignment of IP addresses to DNS names is the responsibility of the System Administrators.

3.3 DNS Names

The MCH (1 or 2) located in the rack number X and rack position Y (in rack units, U) must have a primary DNS name `mch-X-Y-{1,2}` (e.g. `mch1-s1c3-1`).

This IPbus target installed in rack number X, rack position Y, and slot number Z (where Z={1,...,12}) must have a primary DNS name `amc-X-Y-Z` (e.g. amc-`s1c3-2-1`).

The System Administrators will determine the primary DNS name of the Bridge PCs. In addition, the Bridge PCs must have as many DNS names as MCHs being accessed through it. For example, if the Bridge PC `bridgeABC` is accessing the IPbus targets on the crate number 2 of rack number 1, and the crate number 4 of rack number 3, then it should have the following DNS names: `bridge-rack1-2`, `bridge-rack3-4`. This naming convention between Bridge PCs and crates is needed to ensure exclusive access of a Control Hub to any given IPbus Target.

4 IP Address Configuration of the MCH

The MCH must be configured using DHCP. It is the responsibility of the on-call expert to setup the MCH accordingly, and to provide the MCH MAC address to the System Administrator (or execute an script according to their instructions).

The System Administrator (or the script) will bind the MCH MAC address to a unique DNS name, and IP address, and it will make the information available through the DNS and DHCP services.

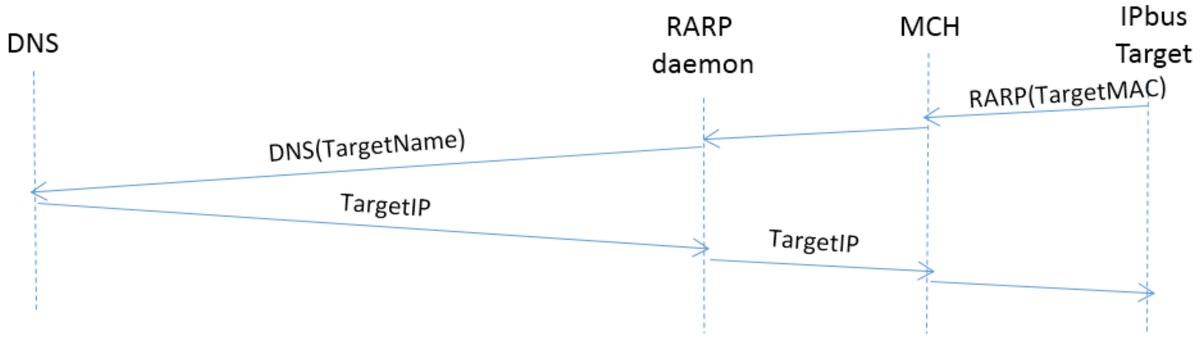
5 IP Address Configuration of the IPbus Target

There are two different ways to configure the IPbus target. The default method provided by the IPbus suite is using RARP. However, it must also be possible to disable this mechanism and allow an external daemon to set the IPbus target IP address directly using IPMI.

5.1 IP Address Configuration using RARP

This is the default IP address configuration mechanism used in the IPbus protocol suite. The System Administrators will setup one or more RARP daemons to serve the periodic RARP requests of the IPbus targets.

Each RARP daemon will first use the information stored in the DHCP and DNS services to resolve the IP address for the MAC address contained in the RARP request, and will then communicate this IP address back to the IPbus target via the RARP response.



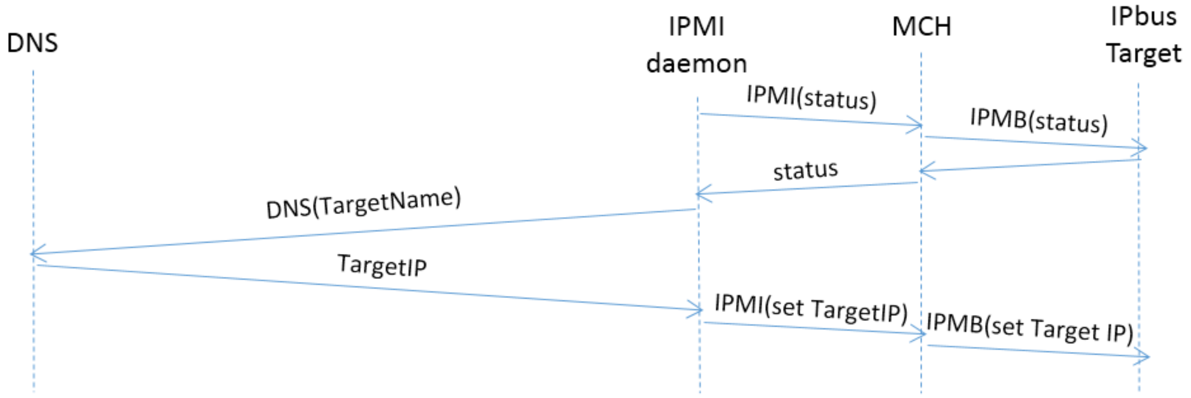
5.2 IP Address Configuration using IPMI

It is possible to access a limited number of functionalities of the IPbus target using the IPMI protocol as follows:

- Given that the MCH DNS name and IP address are known, it is possible to reach the MCH using IPMI commands.
- The IPbus target microcontroller can then be accessed using the IPMI Bus between the MCH and the IPbus target microcontroller.

In this way, an IPMI daemon could poll periodically all the board status for all the IPbus targets attached to an MCH. If one of the targets does not have the IP address set correctly, then it could be set using the following algorithm:

1) Execute an IPMI request to retrieve all the pairs of board slot number and IP address for a given MCH (e.g. `mch-X-Y-1`).
2) For each retrieved pair `<slot_number, IP>`, check that the IP address set on the board is the same than the IP address expected by the DNS for the DNS name `amc-X-Y-<slot_number>`.
3) If the IP addresses do not match, then use the DNS information, and configure the IP address through the IPMI.

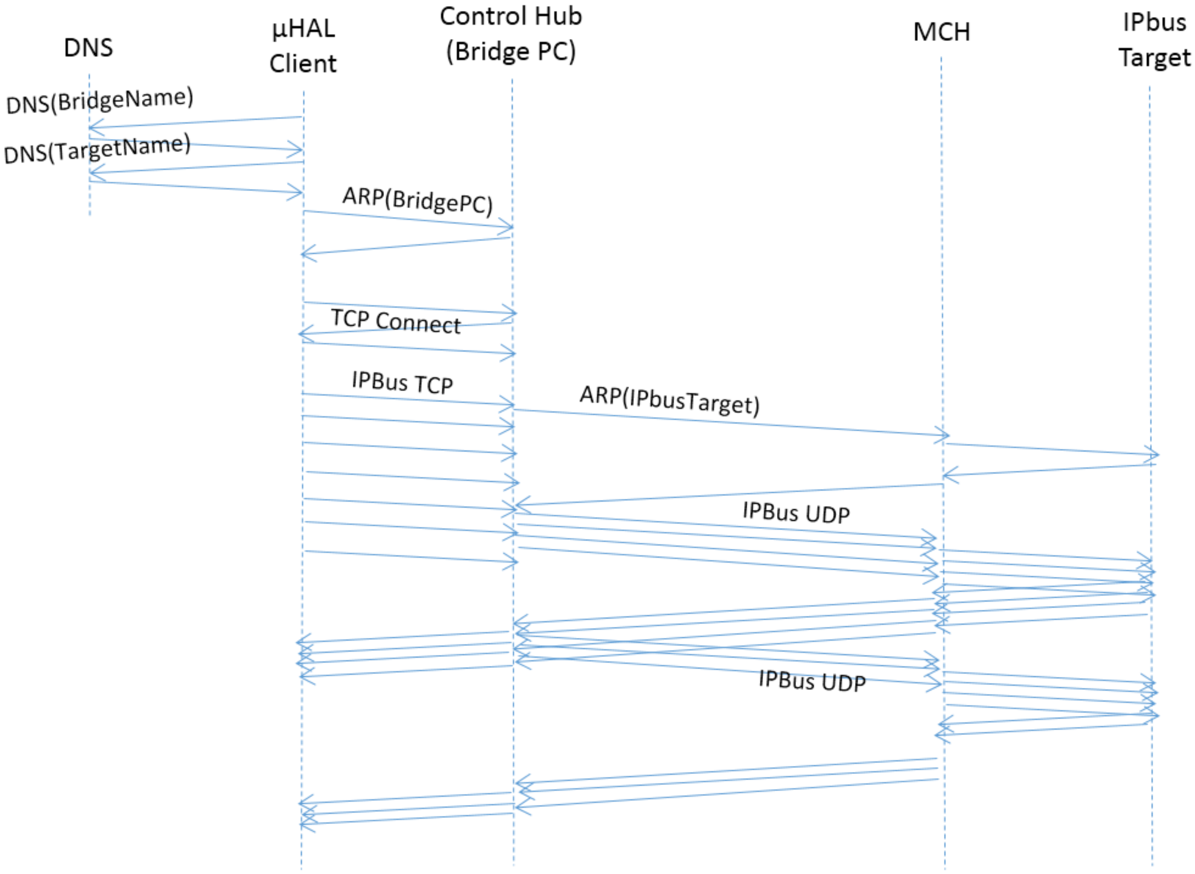


6 Typical Use Cases

6.1 Fault-free Behaviour

The following is a typical sequence of fault-free interactions between the various components of an IPbus network:

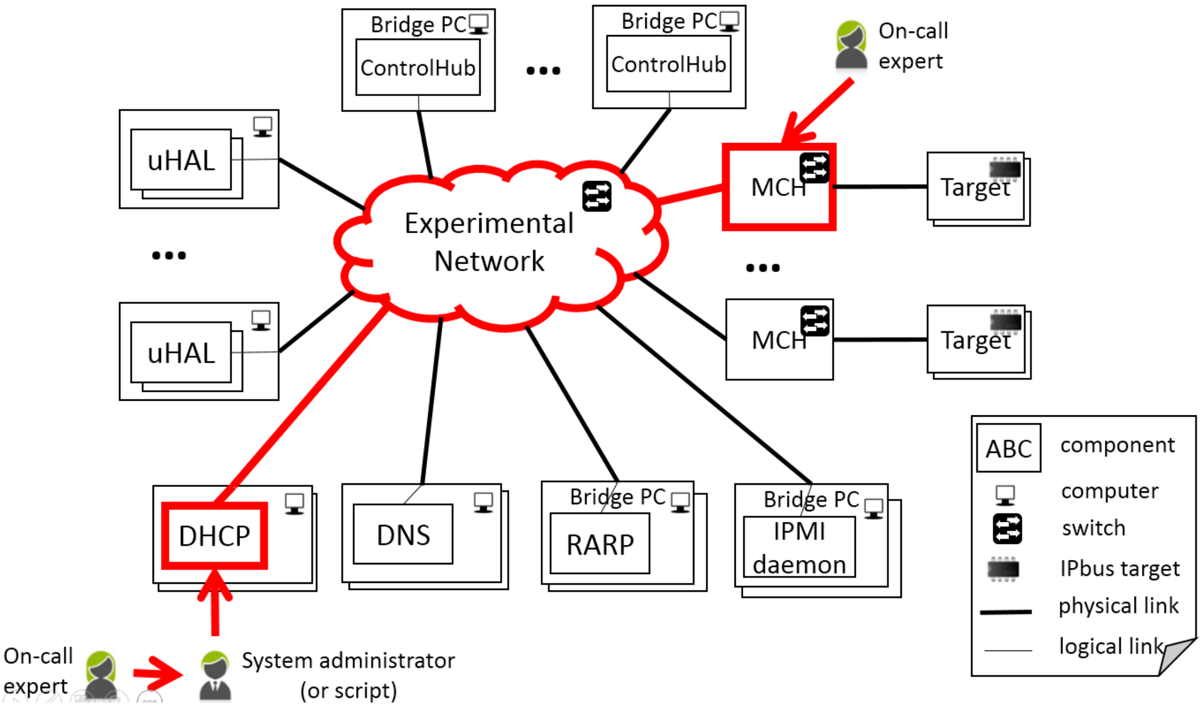
1) *(once per DNS cache renewal period)* The µHAL client resolves the Bridge PC and IPbus Taget IP addresses from the names in the connection URI using the DNS (e.g. `chtcp-2.0://bridgename.cms:10203?target=targetname.cms:50001`).
2) *(once per ARP cache renewal period)* The network stack of the µHAL client PC resolves the Bridge PC MAC address.
3) *(once per `uhal::HwInterface` instance)* The µHAL client connects to the Control Hub instance on the Bridge PC.
4) *(once per `uhal::HwInterface` instance)* The Control Hub receives the IPbus target's IP address from the header of the uHAL-ControlHub IPbus TCP stream.
5) *(once per ARP cache renewal period)* The network stack of the Bridge PC resolves the IPbus Target MAC Address.
6) The IPbus transactions are forwarded to the IPbus target via UDP using the IPbus protocol. In this example, we suppose that the firmware allows 4 packets in flight (see the IPbus protocol document for more details).
7) The IPbus UDP replies are immediately forwarded to the µHAL client using TCP.



6.2 Replacing a Malfunctioning MCH

The following procedure must be followed to replace a malfunctioning MCH:

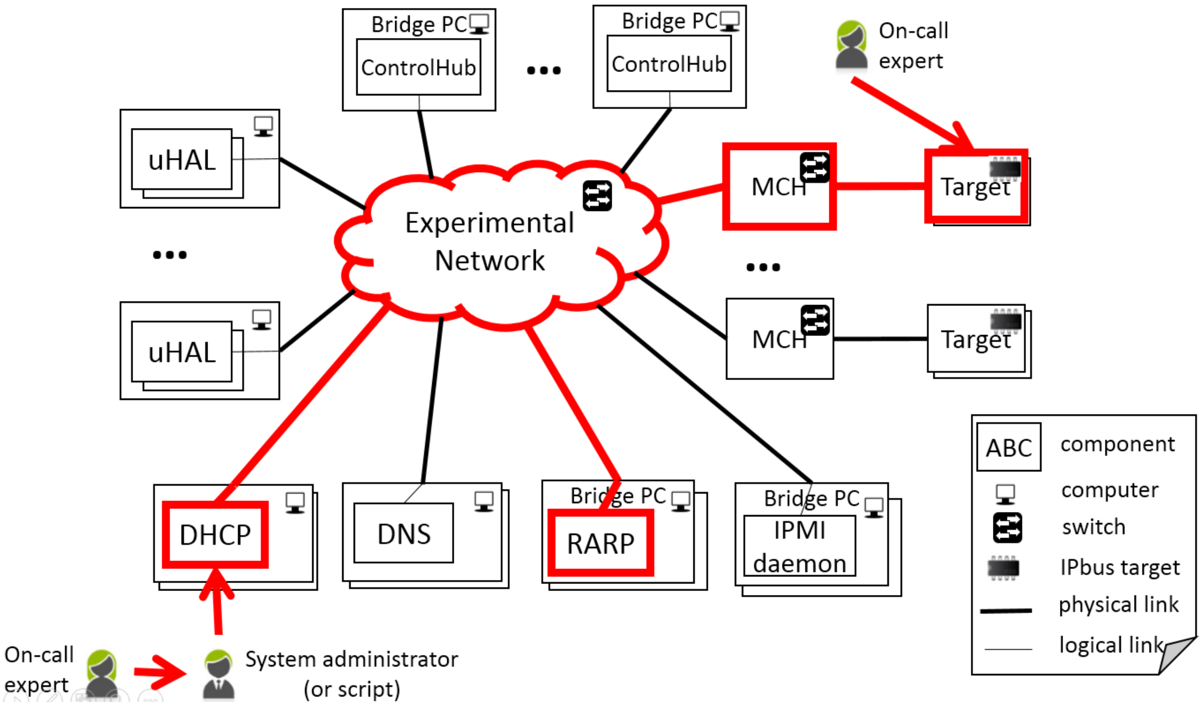
1) The spare MCHs have to be setup in advance by the on-call expert to configure its IP using DHCP.
2) The failing MCH is detected.
3) The old MAC address of the MCH is replaced on the DHCP database (either by the System Administrator or by the on-call experts).
4) The MCH is installed, and the IP is automatically set using DHCP.



6.3 Replacing a Malfunctioning IPbus Target Using RARP

The following procedure must be followed to replace a malfunctioning IPbus target that uses RARP to configure its IP address:

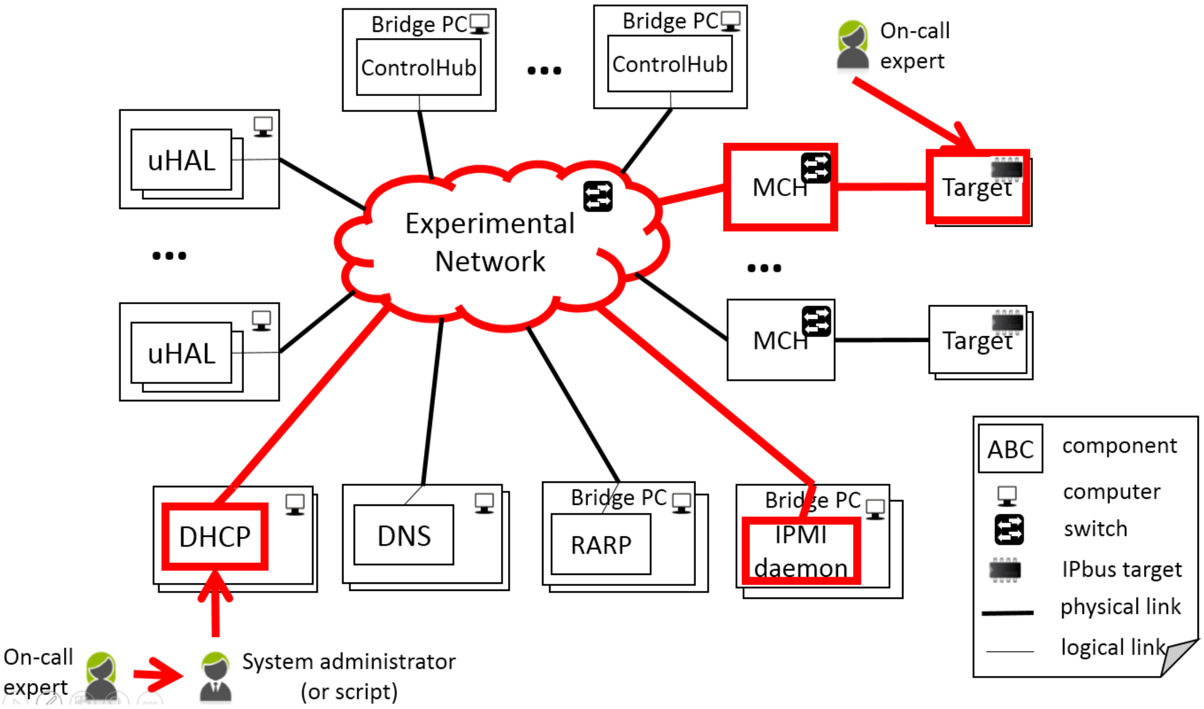
1) The failing IPbus target is detected
2) The MAC address of the old IPbus target is replaced in the DHCP database by the System Administrator or by the on-call expert using a script.
3) The new IPbus target is installed.
4) The IPbus target IP address is configured using RARP.



6.4 Replacing a Malfunctioning IPbus Target Using IPMI

The following procedure must be followed to replace a malfunctioning IPbus target that uses IPMI to configure its IP address:

1) The failing IPbus target is detected
2) The new IPbus target is installed.
3) If the MAC address of the IPbus target is stored in the DHCP database, then it must be replaced.
4) The IPMI detects the new board and configures the IP address.



6.5 Trying to Access to an IPbus Target from a non-Bridge PC

If an IPbus client tries to access to an IPbus target from a non-Bridge PC, then the network stack will return an error (typical a “no route to host” error). The reason is that the ECN is segmented, and the IPbus targets can only be reached from the Bridge PCs.

In addition, if the Control Hubs are configured to only communicate with a whitelist of end points, then each Control Hub will be restricted to just a subset of IPbus targets. This allows System Administrators to stop multiple Control Hubs from attempting to communicate with the same IPbus target.