BACNET SUPPORT FOR EPICS

R. Nussbaumer, G. Waters

TRIUMF, Vancouver, Canada

ABSTRACT

The beam transport systems of the ISAC radioactive beam facility at TRIUMF are operated and controlled with an EPICS based control system. The buildings that house the ISAC project are independently controlled with a commercial system [1] based on the BACnet [2] standard. For reasons of convenience and enhanced capabilities, the building control system was integrated with the physics controls using EPICS. This paper describes the methods used for integrating BACnet data points into an EPICS based environment. The task was accomplished by integrating a collection of open-source software packages to produce a standalone BACnet stack/driver. The stack is co-hosted on a Linux platform that also supports an EPICS IO controller, and provides the necessary hooks to allow the EPICS IOC to access on-line BACnet data in real time. The various packages and integration techniques are discussed.

INTRODUCTION

BACnet, described in ANSI/ASHRAE standard 135-2001, is an open standard network protocol for interconnection of building control system components. The standard includes the use of industry standard ethernet for its physical and datalink layer components. Runtime access to the BACnet data can be accomplished by communicating with the appropriate physical and logical components of the network. Data thus extracted from the BACnet system can be gatewayed into other control system formats, such as EPICS.

BACnet vs. EPICS Model

BACnet systems are structured as a hierarchy of devices, objects and properties. The nature of BACnet objects and their respective properties closely parallels the EPICS schema of records with their respective fields. Many BACnet object types have a directly corresponding standard EPICS record type, and there is a similar parallel between BACnet object properties and EPICS record fields. This creates an obvious method of representing BACnet data in a corresponding EPICS database. Table 1 compares some common EPICS record types and BACnet Object types.

BACnet Object Type	**EPICS Record Type**
Analog Input	Analog Input
Analog Output	Analog Output
Analog Value	Analog Input / Analog Output
Binary Input	Binary Input
Binary Output	Binary Output
Binary Value	Binary Input / Binary Output
Multi-state Input	Multibit Binary Input
Multi-state Output	Multibit Binary Output

Table 1: Some common BACnet Object types compared to EPICS Record types

OPEN-SOURCE UNDERPINNINGS

BACnet for Linux

In order to acquire BACnet data for subsequent export to an EPICS database, the open-source software package 'bacnet4linux'[3] was built and modified. The distribution is composed of approximately 30 C language source files, and related Makefile and sparse documentation. Access to runtime data acquired by the bacnet4linux package may be achieved by attaching an ordinary web browser to bacnet4linux's built-in HTTP server. A user can then navigate through the BACnet runtime database using familiar browsing techniques.

Bacnet4linux acquires a complete copy of the runtime data by broadcasting queries on the BACnet ethernet, and building up a database of BACnet objects from the resultant replies. Real-time data is subsequently acquired on a subscription basis, where the BACnet objects report new data as it occurs. The base distribution of bacnet4linux was augmented by providing for the export and import of the overall BACnet database hierarchy in XML formatted disk files. Using the BACnet runtime discovery process, the TRIUMF installation was taking more than 10 minutes to fully discover all process variables that constitute the system. The XML import method reduces this start-up delay to a few seconds, and also reduces the burden on the rest of the network, which results from the start-up discovery process.

EPICS Soft IOC

The EPICS IO Controller is hosted on the same Linux host as the BACnet stack/driver. As of this writing, all BACnet data points are acquired in EPICS records as binary-input and analog input records.

EPICS Device Support

In order to translate BACnet data into EPICS records, a special BACnet device support layer was written for EPICS. Device support communicates with the BACnet stack using standard Unix style IPC message queues. This has been found to be an efficient way of transferring data into the EPICS database. Data updates are relatively infrequent, but come from a large number of sources on an asynchronous basis. An early attempt to poll all BACnet objects for new data was found to be too slow, and consumed far too much CPU resources.

FUSE Virtual Filesystem

At IOC start-up, all EPICS PVs are initialized from a virtual filesystem that has been retrofitted to the bacnet4linux package. This allows all PVs to be initialized quickly using a single-pass poll of all respective BACnet objects. The open-source 'fuse: filesystem-in-userspace'[4] package was used to build in this functionality. This facility can also be used as an alternative access method to run-time data, providing instant compatibility with any tool capable of reading from files. It is a useful diagnostic and debugging aid. The virtual filesystem creates read-only virtual files, which are named with BACnet device/object-type/object-instance names, and whose contents are the runtime values extracted from the BACnet system.

TWO DATABASES, ONE ORIGIN

The runtime database buried within the BACnet stack and the EPICS IOC database are two representations of the same data. As such, their underlying structures must be kept in synchronization when changes are made to the BACnet control system. In addition, there needs to be some way of correlating the BACnet objects with the respective EPICS database records. The EPICS runtime database contains addressing information that is unique for each record, and is used to create the one-to-one correspondence with each record's respective BACnet data object.

Database Translation Tools

A special utility program was written in perl, that generates the EPICS '.db' formatted database file from an exported XML formatted BACnet database. Addressing information is placed into the EPICS database, and is used at runtime to relate the EPICS database records to their respective BACnet objects. BACnet object types are translated into the corresponding EPICS record types, as well. This guarantees that the BACnet database objects are in correct synchronization with the EPICS database. In addition, a tool has been written to generate EPICS 'Display Manager' display list files which can be used for operator access to the runtime data.

Events & Alarms

BACnet alarm and event management is more complex than the EPICS model. BACnet has object types devoted to management of events & alarms, and the way they are handled by the system and human operators. In contrast, EPICS alarms are simpler components of each individual record. The translation tool that generates the EPICS '.db' files, migrates the BACnet alarm protocol out of the BACnet 'Event Enrollment' objects, and into the respective alarm threshold fields of the EPICS records.

INITIAL INSTALLATION & OPERATION

Physical Embodiment

At TRIUMF, the system operates in parallel with the existing Delta Controls Operator Console. It is hosted on a modestly powered desktop PC running Scientific Linux version 4.0. The BACnet ethernet exists as a dedicated building controls network. BACnet/ethernet is the physical and datalink layer interface used at TRIUMF, although most standard BACnet network layers are supported in bacnet4linux. The BACnet for EPICS system attaches to the network using one ethernet interface, and all other communication with the linux host is conducted on a second ethernet interface. Figure 1 illustrates the overall system as it presently exists.

Read-only functionality

The initial implementation operates as a 'read-only' arrangement, where no system control is performed; building status is only monitored. Although the intention was to be completely non-intrusive within the BACnet system, the BACnet protocol does require query messages to be sent to specific devices attached to the network. Future enhancements may be made to allow for the EPICS IOC to perform write-oriented access to the building control system.

Maintenance

As a matter of procedure, the BACnet driver/stack must be used to perform a discovery of the network configuration whenever changes to the BACnet system are made. The resulting XML database can then be used to generate a corresponding EPICS runtime database and associated DM screens.

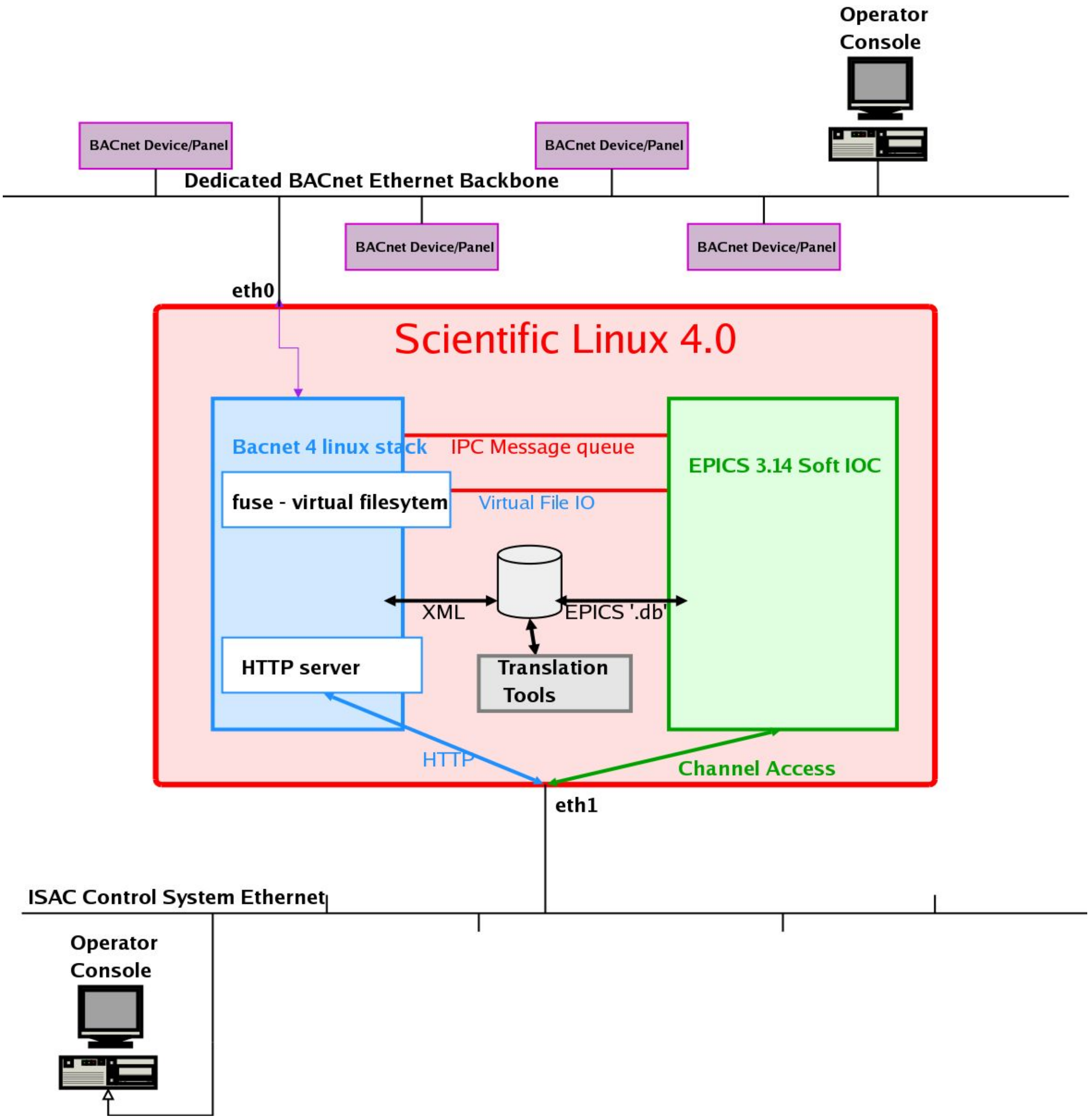

Figure 1: Overall Architecture

REFERENCES

[1] Delta Controls, http://www.deltacontrols.com

[2] ANSI/ASHRAE Standard 135-2001

[3] Sourceforge.net, http://sourceforge.net/projects/bacnet4linux

[4] Sourceforge.net, http://sourceforge.net/projects/fuse