NEXT GENERATION EPICS COMMUNICATION PROTOCOLS

J. Hill, LANL, USA

Abstract

The communications protocols utilized in the Experimental Physics and Industrial Control System (EPICS) have been in use, essentially unchanged, for more than 10 years. With more than 100 EPICS sites worldwide, there is increasing justification for performance and functional upgrades, but with a paramount requirement to remain backward compatible with the existing user base. The future viability of EPICS requires that it keep up with advances in computer and communication technology. To this end, a number of improvements have been proposed for next generation EPICS protocols. This paper describes the proposed changes and their benefits, as well as a strategy for implementing these changes, while allowing uneventful upgrades for the large number of installed systems.

1 MOTIVATION

The EPICS "Channel Access" Internet communications protocols have remained close to their original design since 1988[1][2]. Enhancements have been developed, but these changes were evolutionary rather than revolutionary. During this time period the EPICS collaboration has grown, EPICS is increasingly used to integrate dissimilar systems, and this experience has clearly illuminated some current limitations. Likewise, during this same period, the capability of computer technology has advanced substantially, and it is now possible to design and integrate computationally intensive functionality without impacting the performance of a typical installation.

All EPICS communication is standardized around an exclusive endpoint: the EPICS process variable. While this communication model has provided a strong foundation for a tools-based approach we also see that it unnaturally restricts innovation in certain applications. The primary focus of EPICS is distributed process control, yet there are many experimental physics control applications that could benefit from features which are traditionally associated with data acquisition systems. An early advantage of EPICS communications protocols was their simplified configuration. The EPICS system is increasing used in large projects where wide area networks (WANs) predominate. It is desirable to reduce the expertise that is currently required to configure and maintain WAN-based EPICS systems. Substantial gains in efficiency are possible after modest organizational changes in the protocol, and in certain situations algorithmic compression techniques could also be put to practical use. Opportunities exist for substantial improvements in the utility of EPICS if we are willing to make fundamental changes in its protocol.

2 SUBSTANTIAL IMPROVEMENTS

A more comprehensive process entity paradigm is needed. Currently, the EPICS name space is broken up into individual process variables, each with a fixed set of associated attributes. For example these attributes include the process variable's current value, alarm condition, time stamp, limits, units, and display precision. Since the set of attributes is fixed, software components interfacing with EPICS systems can only communicate between themselves in predefined ways. This approach encourages standardization, but it severely limits innovation. If software tools interfacing with EPICS could define novel attribute types such as "mesons-per-second" then new information flow standards could be established without modifying the EPICS core distribution. Likewise it is desirable for software components to invent named attributes that are aggregates of several other attributes so that container types can be defined. Container types would facilitate higher-level and object-oriented process entity interactions such as message passing and command completion synchronization. The process variable paradigm should be extended to allow components to address multi-dimensional matrix data. Existing string and vector size limits must also be eliminated.

Rudimentary support for data acquisition would greatly improve the utility of EPICS. When clients of EPICS need to monitor the state of a process variable they enter subscriptions with a server publishing the current state of the variable. This server sends state change update messages whenever the event condition specified in the subscription occurs. Currently only three event condition types are built into EPICS. Allowing software components to extend this fixed event set with new named event types such as "RF-arc-down" would substantially improve the flexibility of EPICS. If clients could specify run time evaluated event triggering conditions such as ""RF-arc-down" events WHEN *power* > 10kW" this would introduce a quantum leap in data acquisition flexibility. It is also desirable to allow clients to adjust the server's event queue length.

The EPICS system is now fully capable of extending operations from a local area network (LAN) to a wide area network (WAN). However, currently this requires considerable knowledge on the part of persons who are configuring the system. Introduction of a plug-compatible directory service interface into the core components of EPICS would allow wildcard queries, resource location monitoring, detection of name space collisions during resource installation, and substantially reduced WAN configuration complexity. These changes would extend the minimalist LAN-based EPICS system configuration transparently to WAN-based systems. The EPICS access security protocols could be extended to include support for hardened Internet security protocols such as the Secure Shell Tunnel or Kerberos. This would allow restricted operator access from anywhere within the interconnected Internet. While access of this type might have been dismissed as unnecessary, even irresponsible, in the past it is emerging as an operational necessity in large multi-laboratory collaborative projects, or where midnight access is required from our homes. Since the Internet multicast enhancements are now ubiquitous in today's operating systems, and IP multicast addresses route better through the Internet than IP broadcast addresses, it seems that they are a logical replacement in the protocol. Complete elimination of the confusion prone EPICS repeater daemon would be a likely enduring benefit from this change.

Considerable room for increasing the information density in the existing EPICS protocol exists. With modest effort large reductions could be obtained by reorganizing the protocol. With a higher level of effort on the part of the programmer, and also the processor, command and resource identifiers could be compressed using adaptive Huffman encoding, or some other scheme. Clients might also be provided with fidelity controls for analog event streams that could be used to adjust "lossy" compression (spectrum suppression). The level of compression desired would likely depend on the bandwidth of the communication path and the speed of the processors. Plug-compatible interfaces might allow for a symbiotic relationship of these compression techniques with the EPICS archiving tools.

EPICS IO controllers (IOCs) degrade in a controlled fashion by design. When CPU saturation occurs, high priority tasks continue and low priority tasks are starved. Unfortunately, due to limitations in the current generation of Internet protocols, when the network saturates it is not possible to specify that certain high priority messages should break through, and other low priority messages should starve. The next generation Internet protocols allow a higher degree of control over these choices that could be used to good advantage by specialized routers and LANs. Ideally EPICS clients would specify the "quality-of-service" desired from the network and the IOC. This topic is of particular interest to projects on a large geographic scale such as the Next Linear Collider. The complexity and the expense of the WAN that must be deployed makes it desirable to route low priority process control messages and higher priority feedback control messages on the same network.

Finally, a number of improvements could be made in the client side application programmer's interface (API). The client API needs to export the full functionality available today in the EPICS server level API, and also the changes described in the proceeding paragraphs. A simplified C++ based API would be easier to extend and use. It is also desirable to support multiple threads without relying on the "task variables" feature available in the vxWorks real time operating system.

3 UNEVENTFUL UPGRADE

Design of any modifications to the protocol must be approached carefully because there is a large installed base of EPICS users. Due to the increased risk involved, large sites will not even remotely consider a simultaneous upgrade of all of their applications to a new version during one of their limited maintenance intervals. Therefore, full backward compatibility must be maintained with all previous versions. An initial exchange of minor version numbers between client and server, and a replaceable protocol dispatch table in the protocol engine is envisioned to be a solution that will allow full interoperability between versions, and uneventful upgrades.

4 CONCLUSIONS

The proposed changes to the EPICS communications protocols are crucial facilitators for several new applications. They are not trivial optimizations. The EPICS system is increasingly used to integrate dissimilar systems and therefore careful attention to its process entity paradigm is warranted. Expected benefits include integration with a wider range of related systems, less restriction on the innovation of tool developers, simplified configuration in WAN environments, improved control over degradation under load, improved performance, and a simplified client side application programmers interface. The increasing size of the EPICS user community increases the practicality of devoting resource to this effort. With care, changes of this scale can be made while preserving backward compatibility.

REFERENCES

[1] J. Hill, ``Channel Access: A Software Bus for the LAACS", ICALEPCS'89, Vancouver, 1989.
[2] J. Hill, ``EPICS Communication Loss Management", ICALEPCS'93, Berlin, 1993, pp 218-220.