**Abstract**

The TINE control system evolved in part to meet the needs of controlling a large accelerator, the size of HERA, where not only the size of the machine and efficient online data display and analysis were determining criteria, but also the seamless integration of many different platforms and programming languages. Although there has been continuous development and improvement during the operation of PETRA, it has now been 10 years since the last major release (version 4). Introducing a new major release necessarily implies a restructuring of the protocol headers and a tacit guarantee that it be compatible with its predecessors, as any logical deployment and upgrade strategy will entail operating in a mixed environment. We report here on the newest features of TINE Release 5.0 and an initial deployment.

**TINE Features**

TINE Release 4 and 5 both support the following features:

- Multi-Protocol
  - UDP, TCP, Shared Memory
  - (IPX is optionally available)
- Multi-Platform (clients and servers)
  - Windows, MAC, Linux, /C/C++ or Java
- VxWorks, Solaris, other Unix Systems
- DOS, VMS, other legacy systems
- Multi-Architecture
- Transactions (client-server)
- Publish Subscribe
- Event Scheduling
- Producer Consumer
- Publish-Consumer
- Multi-Language
  - C/C++, Java, .NET, LabView, Python, Matlab
- Multi-Threaded
  - (Single-threaded builds also possible)
- Multi-cast
- Producer Data is multi-casted
  - (can also be broadcasted)
- Address Restriction
- Compound Data Types
- Various tuples in addition to the primitives
- Structured Data
- User-defined structures for atomic transfers
- Contract Coercion
- Servers can coerce client requests into efficient transfers
- Save-and-Restore
- Initial property values use the last stored setting
- Local Histories
- of designated properties
- Local Alarms
- Specific or Automatic alarms
- Local Statistics
- Extensive Central Services
- Extensive Diagnostics

**Release 4 Issues**

- Protocol
  - Provide full support for IPv6
- Headers
  - Update message-size-in-bytes to unsigned 32-bit integer (was 16-bit)
  - aids in reassembly for jumbo transfers
  - Include process ID (client-side request header)
  - Include client type (client-side request header)
  - Include message-size-in-elements (server-side response header)
  - aids in generic call response interpretation
  - Include contract data size (server-side response header)
  - aids in generic call response interpretation
  - Include endianness and character encoding
- API
  - No major API issues
  - Some refactoring needed in C library to avoid name collisions in STL or MFC.

**Release 5 Solutions**

- Protocol
  - IPv6 fully supported (as of 4.6.3)
  - Use dual stack when possible.
  - Fewer headaches with an evaporating address space
- Headers
  - Real jumbo-grams (4,294,967,295 bytes)
  - BUT: IPv6 still very common ...
- API
  - message-size-in-bytes now unsigned 32-bit integer
  - process ID available (client-side request header)
  - client type available (client-side request header)
  - message-size-in-elements available (server-side response header)
  - contract data size available (server-side response header)
  - Include endianness and character encoding
  - still fixed as LITTLE ENDIAN and ASCII
  - Data Stamps still 4-byte unsigned integers
  - timstamp is an 8-byte double
  - System stamp (event/cycle number) is 4-byte integer
  - Cycle/Event (will wrap in 14 years at 10 Hz)
- API
  - Many macro definitions are now enums.
  - C++ and STL or MFC.
  - can optionally make use of namespace wrapper around time.h or not:
  - API unchanged as to the library calls themselves.