Trihedral KROHNE

Barry Baker
P. Eng.

Asset Data Integration

Field Device Communication Protocols

WELCOME !

Some housekeeping items before we get too far (and forget!)

Upon satisfactory completion of this course, you will receive CEU and/or CPE credits. Phoenix Contact is an authorized provider of CEUs licensed through the International Association for Continuing Education and Training (IACET).

Satisfactory completion requirements are as follows:

- Beginning of session – sign in sheet and pre-test
- End of session – post test and class evaluation

After completion of the course, the instructor will submit the class information to the corporate office training department.

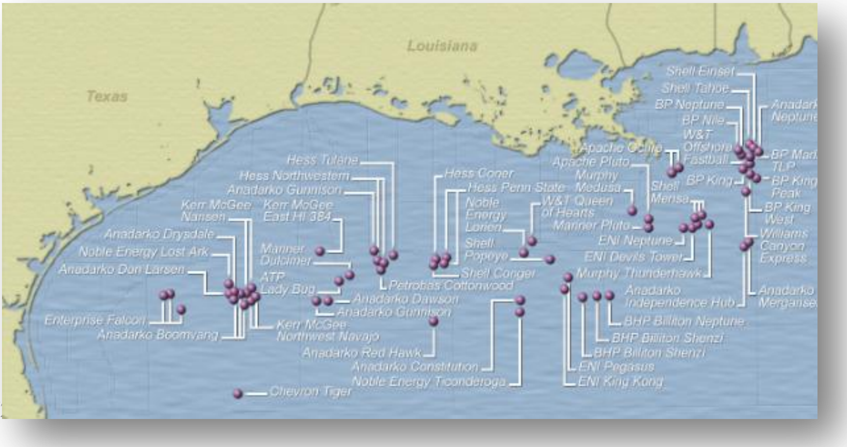
CEU certificates will be mailed to each participant that fulfills the course requirements (meaning we need your mailing address!)

Presentation Outline

1. Trihedral Introduction (Make sure you have completed Pre-test)
2. Overview
3. SCADA Protocol Types
4. SCADA Protocol Comparison
   - Report-by-Exception vs. Poll-for-Current methods
   - DNP3 vs. Modbus
   - IEC 61870-5-101 & -104
5. Protocol Tuning for Efficiency
   - Coalescing data
   - Controlling message size
6. Summary
7. Q & A
8. Post-test

Trihedral Introduction

- Trihedral was founded in 1986 as an HMI (Human Machine Interface) software provider.
- Our software, VTS™ and VTScada™, is installed in thousands of applications worldwide, via direct and indirect sales. The oil & gas sector is largely served via OEM's who re-label the software as their own to serve a variety of specific applications.
- Trihedral has an active engineering group that has written literally hundreds of communication drivers over the years. These range from custom developments for OEM equipment to implementations of standards and their evolution. Some of these activities take place using published protocols while others are created via reverse engineering for closed or obsolete hardware.
- While many communications drivers are specific to their equipment or niche system implementation, there is some commonality to designs. We would like to share with you the overall efficiency considerations and design possibilities that are possible.

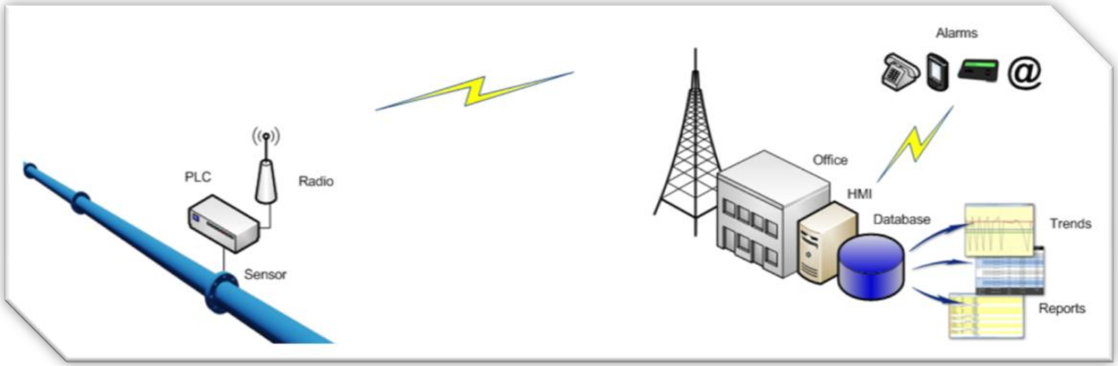


Overview

Selection of a SCADA protocol for communication with remote sites can have a large impact on overall system performance and operational integrity.

General Considerations

- Accuracy of information
- Ongoing cost of data retrieval
- Update rates for displays
- Bandwidth limitations



A typical SCADA communication scenario

- Part 1 -
SCADA Protocol Comparison

SCADA Protocol Classification

The majority of SCADA protocols can be classified as one of the two following types

- Poll-for-current-readings (e.g. Modbus, DF1)
- Report-by-exception or Change of State (e.g. BSAP, DNP3, IEC 60870-5-101).

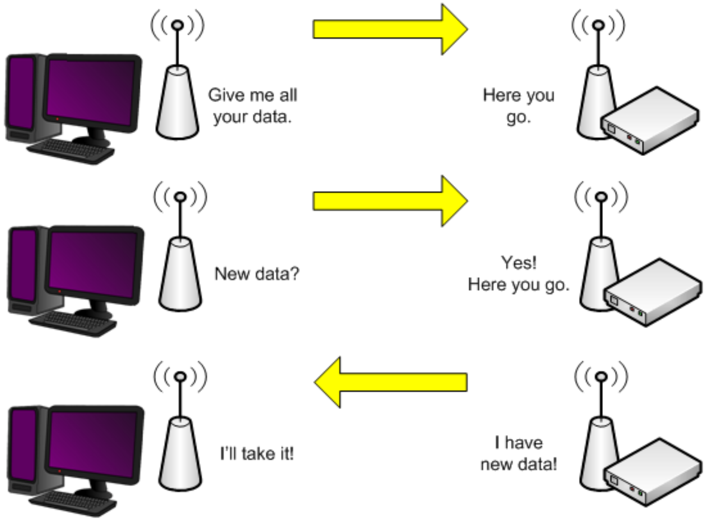
Selection of the correct protocol for a SCADA application will help to ensure

- Timely data updates
- Rapidly changing field conditions are not missed
- Events from several sites can be analyzed on a common time base
- Controlled costs for communication links

Poll for Current Readings

Report-by-exception

Report-by-exception



Poll-for-Current-Readings Protocols

Most originated in the PLC environment

Upside

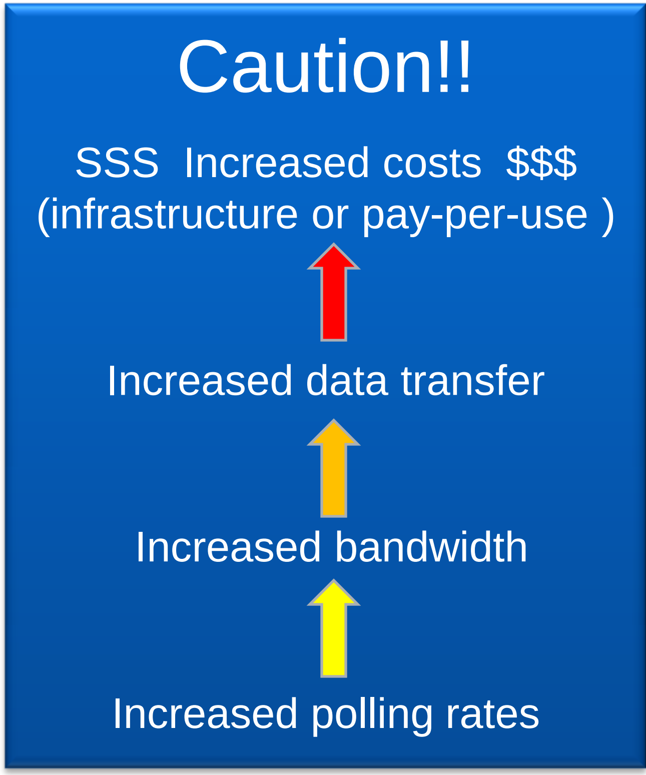
- Easier to configure communications

Downside

- Inefficient. Each poll retrieves all requested field device values.
- Most do not capture rapidly changing states of operation
- Most do not include time-stamping of data at the field device

Note: Poll rate must be equal or less than the minimal acceptable SCADA display update rates for

- Alarming
- Logging



Caution!!

SSS Increased costs $$$
(infrastructure or pay-per-use)

Increased data transfer

Increased bandwidth

Increased polling rates

Report-by-Exception Protocols

Two major variants

- Poll-for-exception
- Unsolicited-report-by-exception

DNP3 supports both variants

DNP3
IEC 60870/DNP3

Upside

- Standardized data types
- Millisecond duration event capturing (time stamped data)
- Data logging (SOE) in the field device
- Data quality attributes
- Two-pass control operations
- Reduced bandwidth requirement due to less data transfer

Downside

- Complicated to configure communications
- May be more expensive for hardware support

Bonus

Time-stamped event data storage has the added advantage of logging data locally.

This provides continuous data gathering during comm interruptions.

Sample Performance Comparison
Single Field Device

30 analog channels
Ave 5 values changing/30 sec

50 digital channels
Ave 1 value changing/30 sec

Comm link

30 sec display data updates

Min 5 sec resolution for logging changes

Modbus RTU Protocol Performance

Must read ALL data every 5 seconds to meet specification for logging, regardless of how many values actually changed

- 2 read messages every 5 seconds (one per each of 2 data types)*
- Total of 93 bytes every 5 seconds
- Total of 65.3k/hr

Display update specification of 30 seconds will be met by the 5 second polling rate

*Improvements to this will be covered later in the presentation

www.modbus.org

DNP3 Protocol Performance

Assuming that data will be classified by type

- Analogs as class 2
- Digitals as class 1

Device polled for class 1 and class 2 data every 30 seconds

All data time-stamped at the device and logged to the Historian with timestamp

- Meets the 5 sec logging accuracy requirements
- In practice, timestamps can have millisecond resolution

“Integrity Poll” requests present state of all values periodically. Ensures devices are OK

Total data requirements with hourly integrity poll is 20.7kB/hr

Modbus	DNP3
65.3kB/hr	20.7kB/hr
In this case DNP offers the same information with 68% less data transfer!	

www.dnp.org

Distributed
Network
Protocol

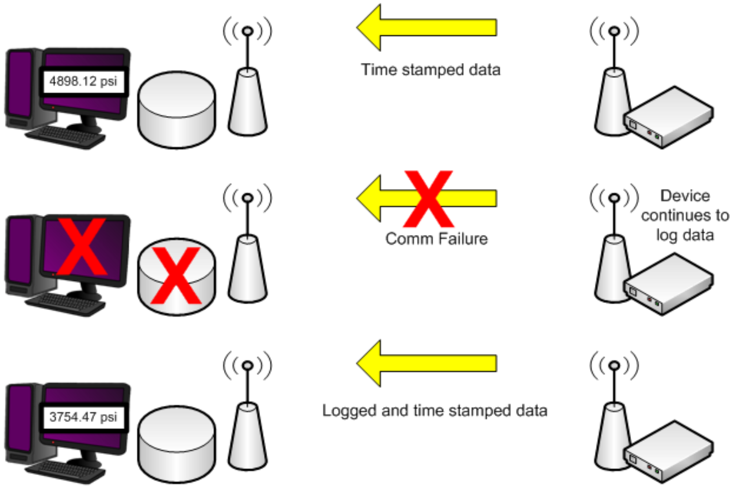
DNP3 (RBE) vs. Modbus (Poll for Current) Conclusions

DNP3 can use a lower bandwidth communication channel to accomplish the same time resolution as Modbus

Applications requiring time correlated data from multiple sites (e.g. leak detection) will benefit from DNP3’s time-stamped events

Communication outages to devices using Modbus will result in a loss of all readings for the duration of the outage

Communication outages to devices using DNP3 will result in loss of displayed data only for the duration of the outage. Logged events will accumulate in the device during the outage and be read & logged after the outage completes



Report by Exception Protocols comparison
DNP3 (North America) vs. IEC 60870-5 (Europe)

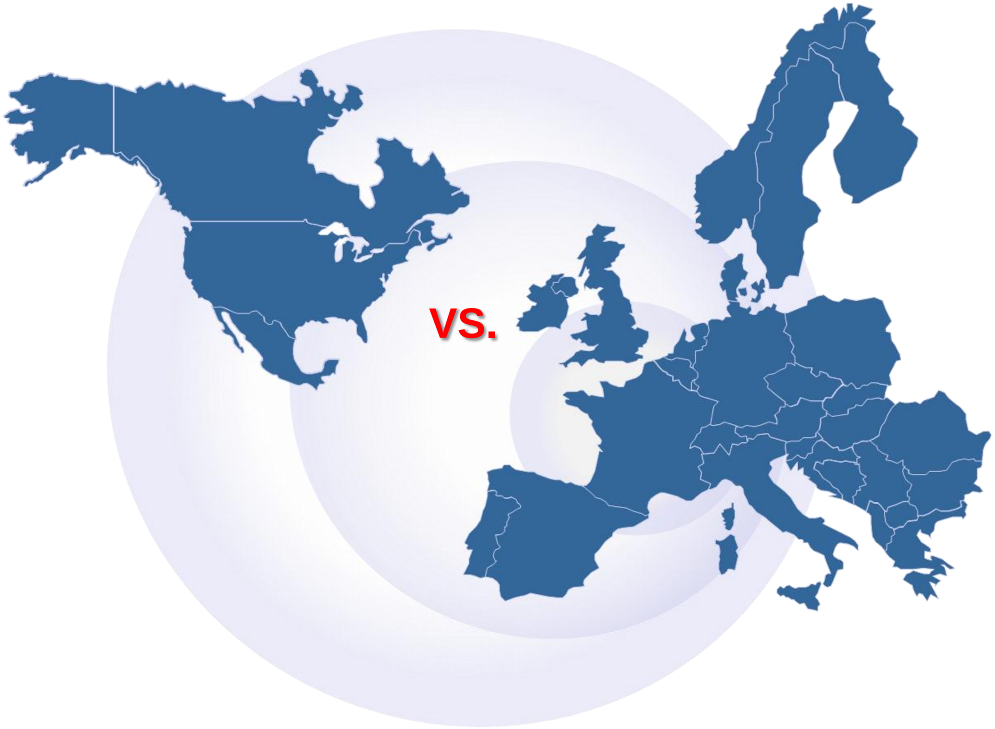
IEC 60870-5 available in two versions

- -101 for serial communication links
- -104 for TCP/IP communication links

IEC 60870-5 provides similar functionality to that described for DNP3

- Time stamped data in devices
- Poll for exceptions
- Data quality attributes
- Well defined data types

…similar performance as well



- Part 2 –
Protocol Tuning for Efficiency

Modbus Protocol Tuning

Modbus is the most widely used industrial & SCADA protocol in use today

Many new and legacy SCADA applications continue to use Modbus

Typical configuration scenarios can result in inefficient communication…

- Slower polling rates
- Higher data throughput requirements

The following examples will illustrate two scenarios where Modbus configuring tuning can be used to reduce poll rates and data throughput requirements

A Review – Modbus Variations

Modbus ASCII

- designed for serial implementations
- inefficient ASCII message format
- seldom used

Modbus designed for RTU

- serial implementations
- similar to ASCII message format but using efficient binary encoding
- **most common implementation of Modbus**

Modbus Plus

- proprietary Modicon industrial network
- seldom used outside of local control (in plant) configurations

Open Modbus TCP

- designed for TCP/IP networks
- enhancements for non deterministic network response times
- using same application layer message encoding as Modbus RTU

Dick Morley

Example #1 – Read of 4 Data Types using Modbus RTU

Premise – Multiple reads are typically required for multiple data types in SCADA to RTU communication. Simple changes can significantly reduce the number of poll messages.

Requirement - Read the following from a remote device using Modbus RTU protocol

- 40 Digital Status Outputs & Internal Status
- 30 Digital Status Inputs
- 20 Analog Outputs & Internal Registers
- 10 Analog Inputs

Typical messaging – Requires 4 query/response messages every poll interval:

- Read Coils 0x01 – query + response total 18 bytes
- Read Discrete Inputs 0x02 – query + response total 18 bytes
- Read Holding Registers 0x03 – query + response total 53 bytes
- Read Input Registers 0x04 – query + response total 33 bytes

Summary

122 bytes/poll interval

With 10 sec poll interval

30.1 MB/month

Example #1 - Re-configured for Efficiency

Re-configure remote device & SCADA to place all required poll data in contiguous holding registers in device

- Minimal RTU configuration required to realize savings
- SCADA need only support extraction of status values packed into words

The same data can now be read in a single query/response

- Read Input Registers 0x04 – query + response total 83 bytes

Result: Simple changes can realize:

- cost savings for pay per byte communication systems
- faster effective polling in shared communication channel systems (e.g. RF)

Before efficiency redesign	After efficiency redesign
122 bytes/poll	83 bytes/poll
Using contiguous registers in the device results in 32% less data transfer	

Example #1 Read of 4 Data Types using Open Modbus/TCP

TCP-based protocols require large overhead packets.

Data can be put into contiguous registers as with Modbus RTU. Same requirements…

- Minimal RTU configuration required to realize savings
- SCADA need only support extraction of status values packed into words

If we take into account the TCP/IP overhead:

- Four separate messages use a total of 473 bytes
- One single combined messages uses 171 bytes

Result: Minimizing the number of messages is far more important when using TCP/IP-based protocols.

Before efficiency redesign	After efficiency redesign
473 bytes/poll	171 bytes/poll

Using contiguous registers in the device results in 64% less data transfer!

However…

Sometimes reducing the number of messages is not always the best choice.

Let’s reconsider example 1

Example #2 – Reducing Message Size

Premise - If SCADA protocols automatically transfer the largest data blocks possible, inefficient communications can result if portions of data in the block are not required.

This example requires SCADA to read the following registers from a field device:

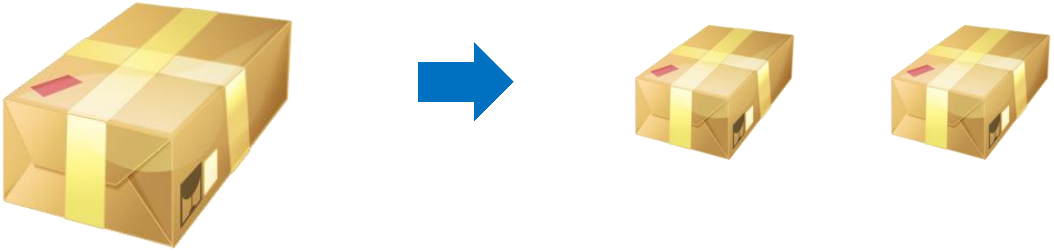
- Equipment 1 is represented by 30 registers starting at 40100
- Equipment 1 is represented by 20 registers starting at 40200

To read the entire set of registers in a single message, would result in a request + response size of 253 bytes.

Result: If two smaller read requests are used we can reduce the total data size per poll to 126 bytes.

- One message for 30 bytes starting at 40100 (73 bytes)
- One message for 20 bytes starting at 40200 (53 bytes)

Before efficiency redesign	After efficiency redesign
253 bytes/poll	126 bytes/poll
Using smaller message sizes results in 50% less data transfer	



Example #2 – …but with Open Modbus/TCP

Using Open Modbus/TCP the same scenario

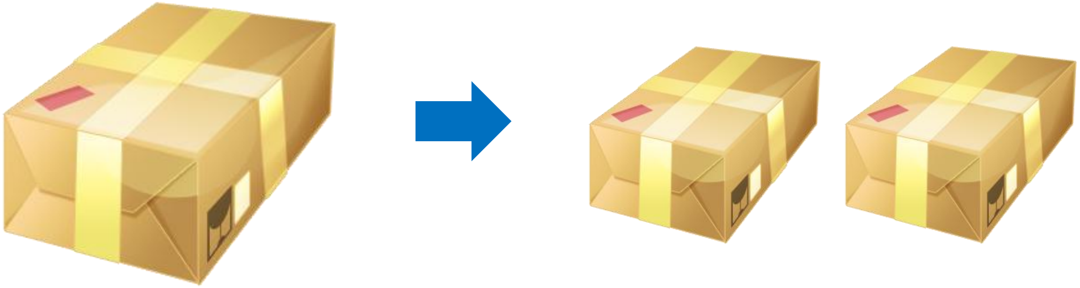
A total of 120 registers can be read in a single message resulting in a request + response size of 342 bytes

Breaking the message into two blocks results in:

- One message for 30 bytes starting at 40100 (162 bytes)
- One message for 20 bytes starting at 40200 (142 bytes)

Result: Note that in this example, the size of the TCP headers impacts on the overall message size so the reduction is not as pronounced as the Modbus RTU example

Before efficiency redesign	After efficiency redesign
342 bytes/poll	304 bytes/poll
Using smaller message sizes results in only 11% less data transfer	



Applicability of Examples to Other Protocols

The principals

- Using continuous register addressing in field devices
- Using appropriate block sizes

…can be applied to any other protocol that supports block memory reads

A few examples

- Allen Bradley (Rockwell) DF1
- GE SNP
- Omron Hostlink

Limitations are the same as when using Modbus

- Some extra programming of remote devices is required to “compact” the data
- SCADA must be able to extract status information from words
- SCADA must be able to define the maximum size of data blocks read from devices

Summary

Part 1

Report by exception protocols provide a number of distinct advantages over poll for readings protocols, including:

- Accurately time stamped data from remote devices
- Reduced telecom data throughput requirements
- No loss of event data during communication outages

Part 2

Legacy poll-for-exception protocols such as Modbus will continue to be used for some time due to their simplicity and entrenched user base.

Poll-for-exception systems can be ‘tuned’ to provide greater communication link efficiencies.

This can lead to lower cost for ‘pay-for-use’ communication links or higher poll frequencies.

Q&A

Your turn to speak!

Post-test

Did we teach you anything (useful)?