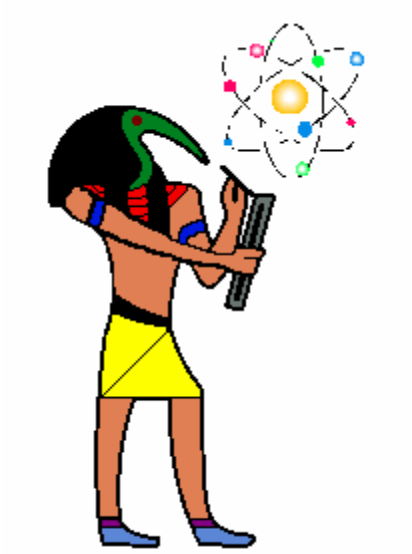
NYC Utility

DAS SCADA Network General Protocol and Traffic Analysis

September 2005



Applied Methodologies, Inc.

Table of Contents

Introduction

Applied Methodologies Inc. conducted a cursory general packet and traffic analysis against NYC Utility's Westchester DAS(Data Acquisition System) Supervisory Control And Data Acquisition(SCADA) network to obtain critical protocol and transaction functional behavior and to have a "before" snapshot of the network's traffic behavior before a major upgrade to the IP network portion of this SCADA system commences. A more detailed in-depth application specific per function analysis was not completed at the time of this writing due to project time constraints. The traces referenced in this document were general, no capture filters set, just an open capture of all traffic on the network. The traces were conducted at the NYC Utility Westchester Control Center location on the primary DAS switch and monitoring segment.

The goal was to obtain a "before" snapshot of the network protocol and traffic pattern behavior to compare to lab work conducted for an upgrade of this network. This activity is required to identify any issues seen in production and whether they will be seen or rectified in the lab. Also, this process is required so as to provide a comparison of the pre upgrade protocol and traffic behavior of the system to the post upgrade or migration stage protocol/traffic behavior of the system. This information can provide NYC Utility's support personnel a reference point to what is seen on the network, is normal, expected or unexpected during routine operation periods or during troubleshooting periods.

The traffic captured is between the Distributed Network Protocol(DNP) speaking application device D200 Master polling and control unit over an IP based LAN/WAN to several remote Master Radio sites which communicate to all of the RTUs via a separate non IP wireless network.

The intended audience of this document is for Network Systems Engineers and Control Center Engineers familiar with NYC Utility's current SCADA network. It is also understood that the reader is familiar with TCP and DNP v.3 protocol mechanics.

More information about the DNP protocol can be obtained here:

http://www.dnp.org/

A basic introduction to SCADA systems can be found here.

http://www.sandia.gov/scada/history.htm

http://ref.web.cern.ch/ref/CERN/CNL/2000/003/scada/

http://en.wikipedia.org/wiki/SCADA

Behavior observed

Some general points of what was observed in the trace files and can be reviewed in person with NYC Utility and or possibly GE personnel.

- DNP Data Link Layer Acknowledgements used, whether this is by design is an outstanding question to GE from NYC Utility.
- Master radio side Transit TCP upon 3 way handshake always sets its Maximum Segment Size(MSS) to 512 bytes
- D200 uses MSS of 1460 bytes. This can result in multiple packets for just one transactions needlessly in one direction if the packets are larger than the MSS.
- TCP 3 way handshake window size mismatch between the D200 and Transit radio.
    - Transit has a window size of 8196 bytes
    - D200 has a window size of 4096
- TCP connections from Master radio site to D200 seem to just reset thus causing periodic reestablishments of the TCP session between the master radio site and the D200.

The behavior appears to be as that of the following

1. A TCP session is active between a master radio and the D200
2. A few DNP transactions go through
3. The TCP connection is abruptly terminated by the master radio
4. The D200 will start a new TCP session with the master radio several seconds later
5. DNP traffic will resume
6. The Master radio will again terminate the TCP connection.
7. This happens repeatedly for each master radio site to the 192.168.144.4 D200 according to the trace files.

According to the GE design specification document and questions submitted to GE from Network Systems the TCP connection between a D200 and Master Radio site is supposed to stay up and never time out.

Note: *some older TCP stacks do use the RST method of terminating a TCP session without the use of a graceful 4 way FIN handshake. Currently unsure if this is the case with the Transits.*

Trace File excerpts to highlight the TCP reset behavior.

Trace file #1 a large several hour general capture of all traffic on the D200 side of the Westchester DAS production site. Note the delta times between packets in each sequence.

Sequence #1 Reset between 192.168.144.16 Master radio and 144.4 D200

No.	Time	Source	Destination	Protocol	Info
108	**3.573824**	192.168.144.16	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 4403 [RST] Seq=4368 Ack=1762194140 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
325	**9.986178**	192.168.144.4	192.168.144.16	TCP	4422 > 20000 [SYN] Seq=0 Ack=0 Win=4096 Len=0 MSS=1460

No.	Time	Source	Destination	Protocol	Info
327	**10.035635**	192.168.144.16	192.168.144.4	TCP	20000 > 4422 [SYN, ACK] Seq=0 Ack=1 Win=8196 Len=0 MSS=512

But then not much longer the master radio resets the connection yet again

No.	Time	Source	Destination	Protocol	Info
437	**13.726523**	192.168.144.16	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 4422 [RST] Seq=4140 Ack=1757586431 Win=0 Len=0

Sequence #2 Reset between 192.168.144.17 Master radio and 144.4 D200

No.	Time	Source	Destination	Protocol	Info
609	**19.481508**	192.168.144.17	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 4416 [RST] Seq=4452 Ack=1759378392 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
681	**23.346800**	192.168.144.4	192.168.144.17	TCP	4434 > 20000 [SYN] Seq=0 Ack=0 Win=4096 Len=0 MSS=1460

No.	Time	Source	Destination	Protocol	Info
684	23.396850	192.168.144.17	192.168.144.4	TCP	20000 > 4434 [SYN, ACK] Seq=0 Ack=1 Win=8196 Len=0 MSS=512

Sequence #3 Reset between 192.168.144.21 Master radio and 144.4 D200

No.	Time	Source	Destination	Protocol	Info
217	**5.513478**	192.168.144.21	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 4413 [RST] Seq=4139 Ack=1760146289 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
515	**16.964027**	192.168.144.4	192.168.144.21	TCP	4430 > 20000 [SYN] Seq=0 Ack=0 Win=4096 Len=0 MSS=1460

No.	Time	Source	Destination	Protocol	Info
518	**17.010913**	192.168.144.21	192.168.144.4	TCP	20000 > 4430 [SYN, ACK] Seq=0 Ack=1 Win=8196 Len=0 MSS=512

Trace file #2 A short capture for a couple of minutes of all production traffic on the D200 side segment.

The same behavior as above was observed below is the output of a display filter just showing all of the resets from just one master radio over a couple of minutes.

No.	Time	Source	Destination	Protocol	Info
370	**11.855852**	192.168.144.20	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 2004 [RST] Seq=4140 Ack=2300818431 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
1167	**43.101692**	192.168.144.20	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 2009 [RST] Seq=4400 Ack=2299666431 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
2113	**78.855385**	192.168.144.20	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 2034 [RST] Seq=4360 Ack=2293778431 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
2470	**90.596755**	192.168.144.20	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 2061 [RST] Seq=4140 Ack=2287634431 Win=0 Len=0

The following set of reset packets show a trend that could provide some clues as to why this behavior occurs. Notice how all the Sequence numbers of the reset packets are set to 4140. Also notice how the ACK number always ends with a 31. This was experienced with other sets of packets from Master Radios as well with other Sequence numbers such as 4160 etc.. These packets were from the large, several hour size trace file.

No.	Time	Source	Destination	Protocol	Info
216872	7858.155791	192.168.144.28	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 3544 [RST] Seq=**4140** Ack=322450**31** Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
217015	7863.955544	192.168.144.22	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 3560 [RST] Seq=**4140** Ack=319634**31** Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
217213	7873.465721	192.168.144.23	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 3564 [RST] Seq=**4140** Ack=318930**31** Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
217316	7876.743616	192.168.144.20	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 3569 [RST] Seq=**4140** Ack=318162**31** Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
217500	7883.666032	192.168.144.22	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 3588 [RST] Seq=**4140** Ack=315538**31** Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
217597	7887.151234	192.168.144.34	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 3579 [RST] Seq=**4140** Ack=316690**31** Win=0 Len=0

Other general observations about this behavior

There does not appear to be any specific packet event or transaction at the DNP level prior to any of these resets that is consistent across all of these examples. In other words there was no common, DNP error or TCP packet that was the same every time before each of the individual TCP session resets.

The Sequence number (SEQ#) noted on the reset packets above also show the reset from a sequence number not observed in the trace file. Meaning there were no valid packets with any of these sequence number. The sequence numbers noted in the reset packets are also much higher in their range than that of the valid packet transactions. The valid packet transactions used sequence numbers from the ones to hundreds yet the sequence numbers referenced and the ACKs in the reference packet are in the thousands. The low sequence numbers are a result from Ethereal's method to simplify the sequence numbers for easer TCP transaction analysis.

These resets could be causing issues in terms of response times and delay on the SCADA network due to the delay incurred with the radio and D200 having to re-establish a session periodically. Even if these sessions were not related to a valid session to transfer DNP traffic these constant TCP resets and establishments may incur a performance penalty on the D200 or Transit network processor and IP stack memory utilization.

Another variation of this behavior is outlined below with the first reset packet sent by the Transit the followed by a second

No.	Time	Source	Destination	Protocol	Info
212	9.535654	192.168.144.32	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 1585 [RST] Seq=4095 Ack=2389586382 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
213	9.563729	192.168.144.32	192.168.144.4	TCP	20000 > 1585 [RST] Seq=0 Ack=2389586382 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
738	30.590768	192.168.144.21	192.168.144.4	TCP	[TCP Previous segment lost] 20000 > 1606 [RST] Seq=4167 Ack=2384082431 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
739	30.634059	192.168.144.21	192.168.144.4	TCP	20000 > 1606 [RST] Seq=72 Ack=2384082431 Win=0 Len=0

DNP Data Link Layer Acknowledgements

.
It was disclosed that the system NYC Utility uses for its DAS network utilizes DNP Application layer acknowledgements and not DNP Data Link Layer acknowledgements.

Also, according to the following: ***Link Layer confirmations***
According to pg 169 Practical Modern SCADA protocols *"Link layer confirmations are unnecessary when using DNP3 over TCP/IP and are specifically not allowed. TCP provided a reliable delivery mechanism, and is backed up at the application layer by confirmations when required."*

Citing DNP3 over local and wide area networks Version 1.0 December 1998: *when using TCP/IP as the transport link layer confirmations shall be disabled?*

It appears that the DNP Data Link acknowledgement, normally used for noisy serial or radio links, is kept persistent across the IP network. This may be perfectly natural due to the application or the way the application's stack handles identifying when to send an ACK. However, this can result in duplicate packets, one just a pure TCP ACK packet and one TCP ACK packet with a DNP Data Link ACK piggybacked. The issue here is now there are two opportunities for an ACK to be missed. This leads to questions as to how the application recovers from a missed TCP ACK or a missed DNP ACK, or are both required? One or both ACKs can be missed during a network re-convergence or other condition. One consistent network layer set of acknowledgements should be present to ensure consistent operating behavior of the network and predictable responses in network or application failure situations.

The trace below shows a TCP and a DNP Data Link layer ACK. This was from a production trace in Westchester. Note the Sequence numbers in Blue Bold in the last two packets. These packets are the same, one with just a TCP ACK and another with a TCP and piggybacked DNP ACK.

No. Time Source Destination Protocol Info
100 3.395535 192.168.144.21 192.168.144.4 DNP 3.0 len=10, from 139 to 4, User Data

Frame 100 (71 bytes on wire, 71 bytes captured)
Ethernet II, Src: 192.168.144.1 (00:08:e3:9d:16:a0), Dst: 192.168.144.4 (00:00:c3:fe:f1:34)
Internet Protocol, Src: 192.168.144.21 (192.168.144.21), Dst: 192.168.144.4 (192.168.144.4)
Transmission Control Protocol, Src Port: 20000 (20000), Dst Port: 4413 (4413), Seq: 27, Ack: 38, Len: 17
Distributed Network Protocol 3.0
  Data Link Layer, Len: 10, From: 139, To: 4, PRM, FCB, FCV, User Data
    Start Bytes: 0x0564
    Length: 10
    Control: 0x73 (PRM, FCB, FCV, User Data)
      0... .... = Direction: Not set
      .1.. .... = Primary: Set
      ..1. .... = Frame Count Bit: Set
      ...1 .... = Frame Count Valid: Set
      .... 0011 = Control Function Code: User Data (3)
    Destination: 4
    Source: 139
    CRC: 0xd011 [correct]
  Transport Layer: 0xed (FIR, FIN, Sequence 45)
    1... .... = Final: Set
    .1.. .... = First: Set
    ..10 1101 = Sequence: 45
    Application data chunks
      Application Chunk 0 Len: 5 CRC 0x00d7
  Application Layer: (FIR, FIN, Sequence 3, Response)
    Control: 0xc3 (FIR, FIN, Sequence 3)
      1... .... = First: Set
      .1.. .... = Final: Set
      ..0. .... = Confirm: Not set
      ...0 0011 = Sequence: 3
    Function Code: Response (0x81)
    Internal Indications: (0x0000)
      0... .... = Device Restart: Not set
      .0.. .... = Device Trouble: Not set
      ..0. .... = Digital Outputs in Local: Not set
      ...0 .... = Time Sync Required: Not set
      .... 0... .... = Class 3 Data Available: Not set
      .... .0.. .... = Class 2 Data Available: Not set
      .... ..0. .... = Class 1 Data Available: Not set
      ....0 .... = Broadcast Msg Rx: Not set
      ....0. .... = Configuration Corrupt: Not set
      ....0 .... = Operation Already Executing: Not set
      .... 0... = Event Buffer Overflow: Not set
      .... .0.. = Parameters Invalid or Out of Range: Not set
      ....0. = Requested Objects Unknown: Not set

No. Time Source Destination Protocol Info
101 3.415621 192.168.144.4 192.168.144.21 TCP 4413 > 20000 [ACK] Seq=38 Ack=44 Win=4096 Len=0

Frame 101 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 192.168.144.4 (00:00:c3:fe:f1:34), Dst: 192.168.144.1 (00:08:e3:9d:16:a0)
Internet Protocol, Src: 192.168.144.4 (192.168.144.4), Dst: 192.168.144.21 (192.168.144.21)
Transmission Control Protocol, Src Port: 4413 (4413), Dst Port: 20000 (20000), **Seq: 38, Ack: 44,** Len: 0
Source port: 4413 (4413)
Destination port: 20000 (20000)
Sequence number: 38 (relative sequence number)
Acknowledgement number: 44 (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (ACK)
Window size: 4096
Checksum: 0x52d0 [correct]
SEQ/ACK analysis

Yet the same ACK packet with the same TCP sequence and ACK numbers are present again but with the DNP Data Link Layer ACK sent along…..

No. Time Source Destination Protocol Info
103 3.432128 192.168.144.4 192.168.144.21 DNP 3.0 len=5, from 4 to 139, ACK

Frame 103 (64 bytes on wire, 64 bytes captured)
Ethernet II, Src: 192.168.144.4 (00:00:c3:fe:f1:34), Dst: 192.168.144.1 (00:08:e3:9d:16:a0)
Internet Protocol, Src: 192.168.144.4 (192.168.144.4), Dst: 192.168.144.21 (192.168.144.21)
Transmission Control Protocol, Src Port: 4413 (4413), Dst Port: 20000 (20000), **Seq: 38, Ack: 44, L**en: 10
Distributed Network Protocol 3.0
Data Link Layer, Len: 5, From: 4, To: 139, DIR, ACK
Start Bytes: 0x0564
Length: 5
Control: 0x80 (DIR, ACK)
1... .... = Direction: Set
.0.. .... = Primary: Not set
...0 .... = Data Flow Control: Not set
.... 0000 = Control Function Code: ACK (0)
Destination: 139
Source: 4
CRC: 0xcebb [correct]

Summary

This protocol and traffic behavior observed and outlined in this document may be perfectly normal by design for this system from an application perspective. Or, a combination of the current system's network configuration with this application results in such behavior. Current testing in the SCADA lab on the new network is pending to determine under the new network configuration does such protocol behavior persist. Also, the D200 in the lab is upgraded and must also be compared to see if the application upgrade of the D200 shows a change in such protocol behavior.

It is recommended that the information in this document be reviewed by NYC Utility Control Center support personnel and followed up to ensure compliance of protocol and traffic behavior on the current system before any migration is to occur. Also the review should identify weather the behavior outlined in this document is strictly application related.

It is recommended that detailed protocol analysis be conducted against the most common and critical DNP based transactions prior to any migration and upgrade of the DAS network. This is required to obtain an insight on the behavior of the DNP transactions on the existing IP network and to outline their relative packet flows, required responses and unique protocol functions noted per transaction. This analysis can provide the support personnel the specific mechanics of the protocol and transaction so to assist them in quickly identifying and isolating an issue on the DAS network.

Having a reference of the most common transactions outlining for example, the number of packets expected per transaction, requires a TCP and or DNP ACK at different layers, expected response times and any other important information will help the support personnel become more efficient in solving issues related to the DAS network and identify quickly whether a problem is network related or application related. A pre and post migration comparison of analysis traces can provide insight as to any efficiency gained in the new network or if the same transaction behavior/traffic patterns/trends are still present from the old network configuration.