BACnet Growing Pains

ABB has been shipping BACnet enabled products for over two years. We have several thousand ACH550s connected to various manufacturers DDC systems via BACnet. A partial list of manufacturers with working ACH550 BACnet connections includes: Alerton / Novar (Honeywell); American Automatrix; Andover Controls (TAC); Automated Logic Corp.; Cimetrics; Delta Controls; Honeywell; Invensys; Johnson Controls, Inc.; KMC Controls; Siemens Building Technologies; TAC; Teletrol; Trane; and York Controls.

BACnet is very powerful, extensible, and extendable and as such, BACnet is a memory intensive protocol. Following is a list of issues seen in installations and a discussion of potential solutions to the issues. Typical comments from the DDC contractor include:

**Question / statement -** When I add the ABB drives to my MS/TP LAN, traffic on that LAN slows considerably.

**Answer –** There is no technical reason that adding ABB drives to an existing MS/TP LAN should slow down network traffic. The BACnet protocol embedded in the ACH550 drive supports all BACnet MS/TP specified baud rates.

However, we have experienced the slowing traffic symptom when the Client (the DDC system) does not understand what services the Server (the ABB ACH550 drive) supports. All ACH550 drives with software version 5.03 and greater are BACnet Testing Labs Listed (BTL) as a BACnet Applications Specific Controller (B-ASC).

Regarding "expected" ACH550 MS/TP B-ASC behavior with Operators Work Station (OWS) or Building Controllers (BC) clients, here are a couple of issues we have encountered:

1. **Read Property Multiple** – the ACH550 does not support Read Property Multiple (DS-RPM-B), and we are NOT REQUIRED to support RPM according to the specification for a B-ASC. Read Property (DS-RP-B) is required by specification and is supported by the ACH550.

Therefore, the client (DDC System integrator) needs to re-write their code to query the drive using Read Property instead of Read Property Multiple. Once this occurs, the speed of the LAN will increase back to normal – unless number 2. below is also occurring.

**NOTE:** ACH550 supported services are published via a MANDATORY Device Object property, Protocol_Object_Types_Supported (97). It seems that OWS vendors do not query this property to determine if RPM is supported, but instead send a RPM request, wait for a NACK (non-acknowledgement), and then send RP. Of course, this slows down information exchange because multiple requests are sent. Unfortunately, we have no control over how the client uses server-published information, but it seems that
MANDATORY properties should be read and used to implement the most efficient method for information exchange. In other words, at the beginning of the implementation, ask the ACH550 what it can support, and then write your code accordingly!

2. Max APDU Size – the ACH550 has maximum APDU (Application Protocol Data Unit) of 100 (prior to version 5.04 the maximum was 50). This value is also published via MANDATORY Device Object property, Max_APDU_Length_Accepted (62). We have also seen an issue where a service request (e.g. RPM from #1) is sent with a length exceeding Max_APDU_Length_Accepted. This causes this message frame to be discarded, according to BACnet MS/TP Receive Frame Finite State Machine specification (9.5.4.4). Since no server response is generated, client waits some timeout interval and either declares the node offline, retries the service request, or sends a different request that is valid and information exchange continues. As with #1. above, if the client application queries MANDATORY device properties, useful information is available to improve the efficiency of information exchange.

In other words, the BACnet specification states that the ACH550 server can discard and ignore any message that exceeds our Max APDU Size, so that is what we do. Often, this causes the Client to send the request again; we drop the message (without telling them we dropped the message) and the cycle starts again. This obviously can slow down traffic on the wire where this loop is occurring.

Unfortunately, when these issues occur, some OWS vendors try to make it seem as if the ACH550 is somehow non-compliant, which is simply not true. We fully comply with the B-ASC device profile and support all MANDATORY Device Object properties. The DDC System integrators need to be educated as to what services and size of message we can support; and then write their code accordingly.

Other issues encountered include:

3. Auto baud detect – there is no defined method for implementing auto baud detection schemes in the BACnet protocol specification.

We believe that we have had a few job sites where the Client’s auto baud detect and the ACH550’s auto baud detect systems have interfered with each other. The symptom of this scenario is a prolonged delay in seeing the ACH550 drive on the DDC System after a power outage. The solution is to not use the auto baud detection feature in the ACH550 BACnet implementation. Your DDC System integrator should know the baud rate of their MS/TP network. Simply ask them for the baud rate and manually enter same into parameter 5303. Use the auto baud detection feature as a last resort.

4. Termination resistors – the ACH550 drive has “active terminators” built into our EIA-485 system. Active terminators include “pull-up / pull-down” bias resistors, not just 120 ohm termination resistors.
If the communications wire segment already includes bias resistors, adding additional bias can load down the network and cause intermittent communications problems. The termination resistors on-board the ACH550 drive seem to cause more problems than they solve. Use the on-board termination resistor jumpers as a last resort.

5. Two-wire hardware verses three-wire hardware – on a few problem projects noise has proven to be the issue. On these problem job sites, the integrator had pulled two wires and a shield instead of the recommended three wires and a shield.

Once the integrator pulled the third wire and landed same on our A-GND terminal, all of the communications problems disappeared. Unfortunately for AFD manufacturers, we arrived late on the horizon of digital building automation systems. The building automation system manufacturers utilize 24 VDC based controllers throughout their automation systems. Their low power controllers have been able to reliably operate in most cases with only two wires and a shield.

However, a 480 volt AFD has approximately 650 VDC on the power board immediately behind the logic board where the serial communications wires are connected. Because of this relatively high power, AFD manufacturers must “float” the logic board to isolate the low power control signals from the 650 V power. (Imposing 650 VDC on the serial communications wire would likely blow the temperature controllers off of the wall).

The lack of an earth ground on the AFD logic board necessitates the use of a third wire in serial communications systems to give the AFD logic a reference for the communications signal. This reference is used to distinguish a serial communications command from noise on the wire. All AFD manufacturers need the A (negative); B (positive); and reference wire in addition to the shield to have a solid communications network.

If the installing contractor pulls three conductors and a shield during the original installation, the cost difference between the three conductor and the two conductor wire they normally use is negligible. However, after the drywall is up and the automation controllers and AFDs are mounted on the walls, the cost of removing the two conductor and pulling three conductor wire can be substantial! Therefore – pull three conductor wire on all serial communications systems utilizing AFDs during the original installation.

6. Master / Slave MS/TP implementation – the ACH550 B-ASC implementation is a Master / Slave implementation. Having the Master capability is what allows the ACH550 drive to support Dynamic Device Binding (B-DDB).

Dynamic Device Binding means that an ACH550 drive can be added to an existing BACnet system automatically. No manual binding is required on the part of the automation systems integrator. This is a unique and powerful BACnet feature.
However, we occasionally get the question on why we developed a Master / Slave implementation rather than having the AFD be strictly a “Slave”. This is mostly a matter of terminology, and is implemented for the reason listed above.

In the "global" BACnet network context, the ACH550 is strictly a "server" (slave), while the building controller or operator workstation is the "client". This means that the ACH550 shall only produce responses to client requests. If you look at the BACnet Interoperability Building Blocks (BIBBs) we support, they all have -B suffix, which is server-side behavior. Client BIBBs have -A suffix.

In the "local" BACnet network context (i.e. MS/TP), the ACH550 participates in token passing, which means that it can accept the token, produce messages on the local network, poll for new masters, pass the token, etc. On the MS/TP network, devices with this capability are defined as "masters". The critical issue here is that only a MS/TP master can support dynamic device binding, where a new device is added and automatically discovered on the network. A slave-only MS/TP device must be manually configured into the BACnet network, which is prohibitively tedious.

So, on the global network, the ACH550 is a server. On the local network (i.e. MS/TP), the ACH550 is sometimes a master.

7. Shields landed per the ACH550 wiring diagram – from trial and error, it has been determined that in most cases it is best to tie the shields together inside of the drive, instead of landing the shields on the EIA 485 terminals as shown in our wiring diagrams.

We are presently changing the wiring diagrams in our manuals. In the mean time, please tie shields together using wire nuts or some other method inside of the AFD cabinet rather than landing the shields as shown. The shield should be terminated only at the EIA-485 source.

Basically, we have found that if there is substantial noise on the EIA-485 wire, and you land the shields on the AFD, our little EMI / RFI filter built into the ACH500 EIA-485 port will attempt to clean-up the noise on the entire building. We will either fail our RC circuit on the board or possibly set-up a ring on the MS/TP wire. It has proven most effective to not land the shields at the AFD most of the time.

**Conclusion** - From the above, one can see that – as in most things - applying AFDs on serial communications systems must be approached carefully. However, the author wishes to stress that we have over 100,000 AFDs connected to all types of serial communications systems (in the US HVAC market alone) with very few problems encountered. The purpose of this article is simply to get consulting engineers, building automation integrators and anyone else involved with serial communications and AFDs to understand the nuances of AFD serial communications when commissioning / designing projects.