bacnet experts

PolarSoft® Inc.
914 South Aiken Ave
Pittsburgh PA 15232-2212
412-683-2018 voice
info@polarsoft.com
www.polarsoft.com

Broadcast Storms in BACnet

David Fisher

4-Mar-2017

TUTORIAL

Contents

Introduction 3
Common Use Cases for Broadcasts 3
Change of Value (COV) Notifications 5
Best Practices 6

Broadcast Storms in BACnet

4-Mar-2017

David Fisher

Introduction

Most BACnet messages are sent as *unicast* meaning that they have a specific destination device that is intended to be the recipient of the message. However, there are other kinds of BACnet messages that are sent as *broadcast* messages, meaning that they are intended for all of the devices in a particular BACnet *network segment*, or across all BACnet network segments which is called a *global broadcast*.

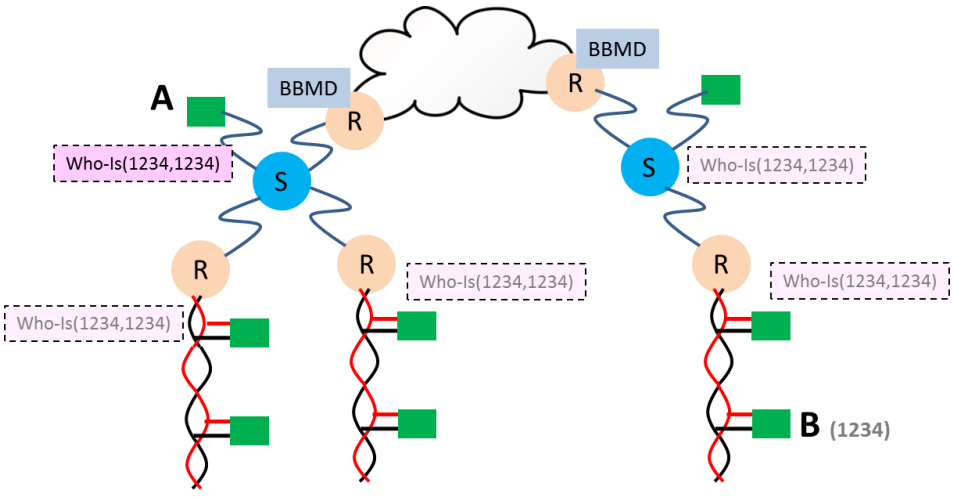
As a rule, broadcasts, particularly global broadcasts, should be used sparingly because they place a higher burden on all devices that are compelled to look at them.

Abuses in terms of the number of, and frequency of, broadcasts can lead to a situation where a large percentage of network message traffic is this type of message. This is often called a "broadcast storm".

The side effects of broadcast storms can vary from simply reduced performance, to actually choking some systems with unnecessary traffic. How these situations occur and how the effects can be mitigated are the subject of this paper.

Common Use Cases for Broadcasts

The most common uses for broadcasts are in device discovery and dynamic binding. The usual scenario is that some BACnet client device A is trying to locate a server device B that it wishes to send messages to. The A device knows the BACnet *device instance*, that is to say the object instance of the device's Device object. Device A typically sends a broadcast Who-Is message using B's device instance for the lower and upper limits of the Who-Is range. Although Who-Is can be used as a local network segment broadcast, more typically it is sent as a global broadcast. In this case the Who-Is message is passed through BACnet routers so that it reaches all of the network segments in a BACnet internetwork.



In this example, device A is looking for device B (instance 1234) so it transmits a global broadcast Who-Is(1234,1234). BACnet routers (R) retransmit copies of this message on each network segment, even across IP subnets when BBMDs are used.

This is a common and reasonable usage for broadcasts, even global ones. Issues occur in two circumstances.

In this example, device A restricts the range presented in the Who-Is to a single device instance, which is the normal and expected usage. Device A is allowed to use a wider range, for example Who-Is(120,130). In this case, any device within that range is expected to respond. This could be useful in circumstances when device A knows that a range of instances are in use, and it wants to locate all of those devices in the range. The operative word here is that it wants all of those devices. However, some BACnet clients abuse the flexibility of Who-Is and send a very large range, such as Who-Is(0,100000), or simply omit the range. In those cases potentially a very large number of devices will reply with I-Am messages.

The second circumstance occurs when device A resends its Who-Is too often. The standard doesn't say what "too often" is, but in practice a client shouldn't send a Who-Is for the same range of target devices any more often than say 15 seconds to every minute as a maximum frequency. Once a given target device does reply with an I-Am message, the device A client should retain the relevant information about the target device, and stop asking for it again and again. If device A had sent Who-Is(1234,1234) and sometime later 1234 replies with an I-Am(1234…), then device A sends various other messages to 1234, then no further Who-Is(1234…) requests are required or desirable. If for some reason 1234 stops replying to requests, then after some number of retries of those requests there is still no response, then it's reasonable for device A to perhaps resend a Who-Is on the theory that device 1234 has been reconfigured, for example to a different (network number,MAC address). However it may be difficult or impossible for device A to distinguish between transient unresponsiveness of 1234, for example due to network congestion, power failure, etc., and actual reconfiguration which is much less common.

Of course the worst possible situation is when device A does both of these bad things, i.e. sending large range Who-Is messages frequently.

In and of themselves, even frequent Who-Is broadcasts do not cause a lot of broadcast traffic. What causes big issues is the manner in which individual devices within the Who-Is range respond. The response is always an I-Am message which may be delayed when the I-Am responding device is on an MS/TP segment because it needs to receive the token before replying as Who-Is is an unconfirmed service. The standard uses this unfortunately broad language:

> "…the I-Am shall be sent in such a manner that the BACnet-user that sent the Who-Is will receive the resulting I-Am."

What is bad about this language is that some BACnet device implementations interpret this to mean that it's OK to send the I-Am as a global broadcast. This doctrine allows the responding device(s) to ignore where the source of the Who-Is message came from, i.e. the source network number and MAC address. That may be convenient in some circumstances, but causes a serious side effect which is the generation of global broadcast traffic. This affects all of the devices in the BACnet internetwork.

The correct and best practice is for devices, that send I-Ams in reply to Who-Is, to unicast the I-Am so that it is targeted directly to the Who-Is sending device. That eliminates the broadcast I-Am traffic.

In situations when Who-Is is sent with a wide range, clearly many devices will respond. If any substantive number of them use the global broadcast I-Am doctrine, this will create a large amount of global broadcast traffic. Obviously if the frequency of Who-Is is often, say more than once per minute, then it will exacerbate the problem by prompting frequent global broadcasts from numerous devices.

Change of Value (COV) Notifications

Another common use case for broadcasts is in COV notification. Normally COV is based on a subscription, where a specific client subscribes to an object or (object,property) in order to receive notifications about changes. Even when the subscription specifies unconfirmed notifications, these are sent as unicast messages to the subscriber.

However it is also possible for devices to send so-called *unsubscribed* COV notifications. In this case, some device has object(s) whose value may be of interest to other related devices. Rather than having subscriptions, the device just sends out UnconfirmedCOVnotifications whenever the value changes. Typically devices that do this have some mechanism for enabling the sending of unsubscribed COV, as well as a mechanism for adjusting the potential frequency of sending. Although a device could target another specific device, BACnet doesn't define a standard way to do this. Typically the UnconfirmedCOVNotification is just broadcast. Note that the ProcessIdentifier in the COVnotification is always zero for unsubscribed notifications.

The standard does not rule on what kind of broadcast should be used, or on the frequency issue.

From a philosophical perspective, it would be generally bad practice to issue global broadcasts for unsubscribed COV notifications. There are few use cases when one could argue that there will be a large number of interested devices for any given value, if any. Overwhelmingly, when there is a value that is of "general interest" it is likely to be confined to a group of related devices, nearly always on the same network segment and in the case of BACnet/IP on the same IP subnet. So the best practice for broadcasting unsubscribed COV notifications is to restrict them to local segment broadcasts. Although the standard doesn't say this explicitly, it should.

As with I-Ams, unsubscribed unconfirmedCOVNotifications have two potential issues: scope of the broadcast and frequency of broadcasts. The scope, as already mentioned, should always be the local segment, and as a rule should never be global broadcasts. Frequency is more subtle. Obviously if we are interested in changes of value, and the value changes frequently, then we will get frequent notifications. For COV this is normally mitigated by a COV_Increment property that sets the amount by which the value must change since the last notification, before another notification can be sent. The standard doesn't say this, but really any implementation of unsubscribed COV should also have a setting that limits the absolute frequency of sending regardless of whether COV_Increment has been exceeded. This frequency and what value is reasonable for it of course varies by application.

Unsubscribed COVnotifications can cause broadcast storms under several circumstances. Most common is setting the COV_Increment too low, without having the ability to also set a maximum limit on frequency of broadcast. In a dynamic environment this can easily cause floods of unnecessary notifications. If the notifications are globally broadcast that would clearly cause issues, which is why in general COV notification should never be globally broadcast.

But even when the COV_Increment is a reasonable value it's possible for environment to play a key role in triggering broadcast storms. A good example are spaces adjacent to a loading dock. In winter, whenever the loading dock doors are opened it can create an incremental dip in

temperature for example. While short lived, this can easily cause corresponding dips in adjacent areas, triggering many COV notifications.

Best Practices

In order to reduce or eliminate broadcast storms, follow these best practices:

- Configure all BACnet client devices to use minimal ranges for Who-Is discovery, ideally single device only, e.g. Who-Is(1234,1234)
- Identify devices that make use of wide range or unranged Who-Is, and remove or replace those devices with more socially responsible ones. Product developers take note: Don't Do This!
- Identify all devices that respond to Who-Is with globally broadcast or locally broadcast I-Am and replace them with devices that only unicast I-Am. Product developers take note: Don't Do This!
- When using unsubscribed COV notification, make sure that all devices ONLY broadcast locally. Remove and replace any devices that make use of global broadcast COVNotifications.
- Any devices that initiate unsubscribed COVNotification should also have a means to disable COVNotification completely, as well as limiting the frequency that any unsubscribed COVNotification local broadcast may be sent.