

# Missing Replies in IRMs

*Problem analysis*

Mon, Oct 8, 2001

A new problem has been uncovered that relates to a failure to receive replies to one-shot messages at times. It was noticed with the Acnet protocols because data logging at one-minute intervals would sometimes fail for many minutes at a time. The specific case was using node0588 as a server to reach data from node0584. Using node0509 as a client, one could find times when a similar one-shot example would fail, or not, every time it was attempted. Looking at the situation in more detail, it was found that the server node often did not even attempt a reply, in that the output counter in the Port# block (third word) did not increment.

The NETQFLG field in a type 9 Acnet message block contains 3 flag bits that are used to facilitate proper message delivery behavior. The sign bit (most-significant) of this 16-bit field is used to mark the message block "busy" upon successful queueing to the network via NETQUEUE. While it is busy, another update of the same message block cannot be done. If the request is to be canceled, a busy message block cannot be freed; instead, bit #7 in the MBLKTYPE field is set to signal to QMonitor that it should free the block once it becomes not busy. QMonitor code, when monitoring the entries in the OUTPQ table, clears the busy flag bit upon successfully delivering the reply message as a result of receiving the proper transmit interrupt. The transmit interrupt code clears the (different) busy bit in the USED byte-wide field of one or more consecutive OUTPQ entries whose messages comprised the datagram whose completion caused the transmit interrupt.

Bit 14 of the NETQFLG is used to indicate that QMonitor should not free the block of completion of transmission. It will be kept until the request is canceled. Bit 13 is used to mark a RETDAT request, as opposed to a DZero request. Since DZero is no longer supported, this flag bit is no longer required.

Enumeration of uses of NETQFLG field

## ACDELETE

Free the answers block, unless the NETQFLG busy bit is set, else set bit 7 in the MBLKTYPE field of the answers block. Similar logic is applied to the answers block for a server request.

## ACDELCHK

This routine is called by QMonitor when both bits 14 and 13 are set in NETQFLG. It calls ACDELETE in the case that the MLT (least-significant) bit in the first word of the Acnet header is clear, which would be the case for a one-shot request.

## ACUPDCHK

If bit 15 of the NETQFLG is set, do not update the request; simply return. If bit 15 is clear, perform any per-cycle logic required by this request. Then, if the FTDCNTR is positive, decrement it. If it is negative, implying that the request is clock event-based, look up the event in the bit-map. If the request is a one-shot, or if the FTDCNTR has counted down to zero, or if the clock event has occurred, update the request by calling ACUPDATE. Then, if the FTD is positive, reset the FTDCNTR to one-quarter of its value (in 60 Hz units), else set FTDCNTR = -1.

## ACUPDATE

This routine updates non-server requests. After successfully queueing the answers block to the network via NETQUEUE, set bit 15 of NETQFLG to show that the answers block is busy.

**ACUPSERV**

This routine updates server requests. If bit 15 of the `NETQFLG` in the answers block is set, do not do anything. After successfully delivering the answers via `NETQUEUE`, bit 15 of `NETQFLG` is set to show that the answers block is busy. If the `ACNERRX` field is non-negative, call the new routine `ERRORLOG` to log the answers delivery if any Acnet device error status is nonzero.

**RESENDX**

Re-issue data request to a single node from which replies are missing. If the original request was not for a set of devices all from the same node, allocate a new request message block, setting `NETQFLG = 0`. Build the request message using only those Acnet devices that come from the single node. After successful queueing of the re-issue request, set bit 15 of the `NETQFLG` field to show it is busy. Since no other bit is set, `QMonitor` will free this request message block as soon as it has been transmitted.

**ACSRSEND**

Re-issue data request using original saved request message block. Do nothing if the busy (bit 15) flag bit is set in `NETQFLG`; otherwise, queue the message to the network via `NETQUEUE`, and if successful, set bit 15 of the `NETQFLG` to show that the message block is busy. This is called by `RESENDX` when all the Acnet devices come from a single node.

**DOANSW**

Initialize `NETQFLG` to `0x6000`, setting both the reserve flag (bit 14) and the `ACRQ` flag (bit 13). The latter bit causes `QMonitor` to call `ACDELCHK` upon completion of transmission of the setting message. (If the `ACRQ` bit were clear, it implies a `DZero` protocol request, which is no longer supported.)

**DOXREQ**

Initialize `NETQFLG` to `0x4000`, setting only the reserve flag (bit 14).

**SETSERV**

Initialize `NETQFLG` to `0x6000`, setting both the reserve flag (bit 14) and the `ACRQ` flag (bit 13). The latter bit causes `QMonitor` to call `ACDELCHK` upon completion of transmission of the setting message. (If the `ACRQ` bit were clear, it implies a `DZero` protocol request, which is no longer supported.)

**USMSET**

Initialize `NETQFLG` to 0. But set bit 15 upon successful queueing of the setting message via `NETQUEUE`.

**SAALLOC**

Initialize `NETQFLG` to 0 in this message block used to carry the setting acknowledgment message, which consists of an array of Acnet status words, one for each device.

**SETACK**

Upon successfully queueing the setting acknowledgment message block via `NETQUEUE`, set bit 15 in `NETQFLG` to show that the block is busy.

**SAFREE**

If bit 15 of `NETQFLG` is clear, free the memory used by the setting acknowledgment message block. this routine is called from `SETNSERV` to free the setting ack block if it is not busy.

Since the logic associated with `NETQFLG` is intimately tied in with `QMonitor`, here are the uses

made of this flag word in `QMonitor`.

**FREEBLK**

When no more use is to be made with a type9 message block, clear the busy flag by clearing bit 15 of the `NETQFLG` field. If bit 7 of the `MBLKTYPE` field is set, it means the request has been canceled, so this message block can now be freed. If bit 7 of `MBLKTYPE` is clear, and bit 14 of `NETQFLG` is also clear, then free the message block, since the reserve flag was not set, meaning that the message block was to be used only once. If bit 7 of `MBLKTYPE` was clear, but bit 14 of `NETQFLG` was set, call `ACDELCHK` if the `MBLKTYPE` of the associated request block was 12, implying a `RETDAT` request, or else call `PROHAND` to get the supporting local application to handle it.