Missing BRF Replies

Sleuthing saga

Robert Goodwin

Wed, Nov 2, 2005

Introduction

For quite some time, studies have been made of missing datagrams as seen by the Booster HLRF server `node06CF`. As a kind of bottleneck through which requests and replies pass from Acnet clients, the server node is in a position to monitor the success of network communications. For each active Acnet `RETDAT` request, it knows when to expect replies from all of the contributing nodes. When it detects a missing reply, it posts "`36 –7`" (stale data) status for those devices in the request that relate to that node and logs the occurrence in the `ACNETERR` data stream. The status errors are, of course, posted to the Acnet web error page. This note covers what has been learned so far from studies of such occurrences.

Preparing diagnostics

From previous studies of missing datagrams seen by the Linac server `node0600`, a local application called `MISS` was written to monitor error occurrences seen by a server node. This same LA is now being used to monitor errors seen by the BRF server. The application currently runs in test `node0509`, monitoring 15 Hz replies from the server for the most recent new records written into the `ACNETERR` data stream. As soon as it notices an error record, it makes one-shot requests to collect diagnostics that may help to analyze the cause of the error. These include especially the `NETFRAME` data stream records in both the missing node and in the server node. Along with recent clock events seen by the server node, these diagnostics are packaged into a 2K byte composite record that is logged internally, the log housing the last 8 such records accumulated in a circular buffer. Monitoring this log is done manually via the usual little console access to the various nodes. One notices the same `ACNETERR` error log in `node06CF` to know when new records have been written into the `MISS` circular buffer. The Print Memory page application `PMEM` is useful to print out the 2K byte records, as well as the other diagnostics, for analysis.

Server logic

A `RETDAT` request received by the server node is passed to the affected nodes usually via multicast in order to reach all front end nodes within the project. (The second word of each `SSDN` always indicates the real source node for that device-property.) Data structures are allocated to prepare for receiving the expected replies from the contributing nodes. As each reply is received, the local 15 Hz cycle number is captured as a means of later measuring the age of the reply, and the reply data is sprinkled into the composite reply buffer.

All nodes in a project operate synchronously, and at a time relatively early in the 15 Hz cycle when they have updated their own data pools, each delivers replies to its active requests (for which replies are due on that cycle) to the server node. At a "deadline" time that is normally 40 ms into the 66 ms cycle, the server node determines it will deliver a composite reply to the Acnet client, and it reviews the age of the replies from each contributing node.

For a periodic request, say, one specifying 1 Hz replies, a check is made that the current local cycle number minus the reply cycle number falls within the reply period. If not, a "`36 –7`" status is assigned to all device-properties from which reply data was expected but missed. Before the reply is actually queued to the network, a scan is made through the composite reply data to see whether any nonzero error status is found. If it is, a 16-byte record is written into the `ACNETERR` data stream log that includes the error status, the client node, the total size of the composite reply data, the missing node, and the time-of-day given to a resolution of a half millisecond within a 15 Hz cycle. (Note that the rest of the reply message is not scanned.)

On the next cycle, the reply from the server node received by the node running `MISS` will show that such a record has been written, and that will prompt issuing one-shot requests for more details. Replies to these one-shot requests are combined to form the complete 2K-byte diagnostic record that is written into the 8-element circular buffer.

GetDat status monitoring

While the `MISS` local application monitors replies about the `ACNETERR` data stream from `node06CF`, it checks for reliable replies. If it receives a "`7`" status from the `GetDat` call to collect the most recent reply data, it means that a reply was missed. A status of "`0`" means that it has obtained new reply data. An internal log is used to record any changes in this status value. A logged 8-byte entry includes the new status value and the time-of-day, indicated to a 15 Hz cycle. This measures the reliability of 15 Hz reply delivery from the server node to the test node that is running `MISS`. It doesn't have to do with the `ACNETERR` log in the server node, except that it is another monitor of successful network communications.

Lessons learned

There are occasional missed replies as seen by the server node from monitoring contributing node replies to active requests. But there is more than a single cause. The rest of this note describes what causes have been determined so far, recognizing that this is still a work in progress.

What is the typical activity seen from studying the `NETFRAME` records? There is a datalogger request for 1 Hz readings of the gap envelope from each of the 19 BRF stations. This request is not refreshed very often, and over time, it is possible that replies get somewhat out of step. Elapsed times of up to 12 days have been seen. At the time of this writing, this request is two and a half days old. All nodes but one are in sync, meaning that they all reply on the same cycle, the cycle on which the server node delivers the reply to the DAE client. (The reason for one node being out of sync will be explained later.) In addition to this active request (24/7), there are console initiated requests from page `B25`, the Booster HLRF status page. At this time, they ask for data to be returned from 18 of the RF stations at 1 Hz. These requests operate only as long as the console page is active. Note that the phasing of these requests with the normal datalogger request is arbitrary; it merely depends upon what cycle the request is received.

Late-recognized replies

An early characteristic from studying the detailed diagnostics was the apparent late arrival of the last of a flurry of replies. As all of the nodes perform essentially the same jobs every 15 Hz cycle, they are usually in a position to deliver their 1 Hz replies at about 3 ms into the 66 ms cycle; thus, they all arrive at the server node at nearly identical times, so close that the precise order is hard to predict. But when an error occurred, it was often the case that one reply was not recognized until just after the `Server` task returned the composite reply to the client. The cause for this is still unknown, though it may have to do with a race condition to which the ethernet driver code is vulnerable.

The ethernet interrupt code processes all received frames since the last interrupt, and it also processes at the same time "transmission complete" interrupts. It is not necessary to have a separate interrupt for each frame received. The interface chip is handed a circular queue of 340 full frame (1500 byte) buffers. The maximum number of received frames processed in a single interrupt in the server `node06CF` is 5, a record in that node (after being up for 100 days) that was set about 2 weeks ago. The main work done for each received frame is passing a short reference message via a message queue to awaken task level processing.

If the last frame of a flurry is missed by the interrupt handler, it may be that it will be seen only when another interrupt occurs. In the cases studied, another interrupt occurred as a result of completing the transmission of the composite reply at `Server` task time. It appears that the missed frame is finally recognized, and shortly processed completely at task level. This can partially explain why the frame was not seen by the deadline time.

Assuming something like the above is correct, and considering the request activity that is most common—that from the datalogger resulting in 86400 replies per day from all 19 stations—the timing of one of these stations, arbitrarily chosen as `node06B5`, was delayed by 4 ms so that the interrupt resulting from its lone reply can act as a "cleanup" mechanism for any lurking frame not properly recognized from the preceding flurry of replies. The final frame sent in the flurry might thereby be recognized in the interrupt first before the lone reply is recognized. With this pragmatically-driven modification, the "delayed" reply errors seem to have ceased.

Missing 15 Hz interrupt

An addition to this scheme was encountered after one of the nodes, `node06B6`, was moved to a different port on the ethernet switch. After the move, its phase in the normal flurry of replies was shifted to another cycle, the same one that `node0669` was replying on, and also the same one that `node06CE` might reply on if a reply to its event-based request is due. The problem could happen when a fourth node (`node06B8`) joined the trio. (Note that all this stuff nearly always works without error; we are only focusing here on those times in which an error occurred due to a missing reply.)

Studying another diagnostic that logs anomalous times measured between successive 15 Hz interrupts, it was learned that two nodes were occasionally missing a 15 Hz interrupt. One was `node06B8`, one of the participants in the flurry of replies at 1 Hz. (The other is `node06CB`, which does not cause a problem because it seldom shows up in requests.) This is important because a result of missing an interrupt, although it places the node into a mode in which it maintains a 12.5 Hz operating rate, is that one cycle of activity is missed, which means that its next reply to the 1 Hz request for data from all 19 nodes will be sent one cycle later than it was before the missing interrupt, placing it out of phase with the others. Every time an interrupt is missed, this will happen again, so that eventually it will "find" any phase at all. When it found the phase in which `node0669` and `node06B6` were regularly replying, and if `node06CE`, responding to a different event-based request, also replied on that same cycle, we saw that occasionally, we would get the "delayed" errors for one of these nodes. It was as if a flurry of 4 nodes was enough to sometimes cause the last one to be unrecognized. To get around this problem, at least temporarily, the reply for `node06B6` was moved later by 2 ms. This temporary ruse seemed to work.

Of course, we plan to fix the hardware that results in the missing 15 Hz interrupts. Of some 80 IRM nodes checked, only the two mentioned above logged missing 15 Hz interrupts. It is important that all nodes in a project operate synchronously.

Two second problem

Analysis of the diagnostics collected upon discovery of a "`36 –7`" error sometimes shows that replies from the missing node occurred not once but two or more times during an approximate 2 second interval. It is believed that the missing node sent the replies, but they were not received by the server node. The cause of this effect is still a mystery. There are times when more than one missing node exhibits this effect at the same time. Curiously, the server node will resend a request after about two seconds of missing replies from a contributing node, but this "2 second problem" does not normally provoke a resend.

Diagnostic example by MISS

The following simple example shows diagnostics gathered by the `MISS` local application (running in `node0509`) for the case of a single missing `RETDAT` reply from `node06B6` amidst the 1 Hz replies gathered by server `node06CF` for delivery to the DAE. This datalogger request is for 2 bytes from each of 19 Booster HLRF front end nodes at 1 Hz.

```
FILE<0509>             10/31/05 0943
:0007EDC2   0C5D 004C F924 06B6 0510 3020 0603 0551  ACNETERR record logged by node06CF
                                                      10/30/05 2006:03-05 missing node06B6 reply
:0007EDD2   1111 1111 1112 1213 1D1D 1D1D 1D1D 1D1D  recent pattern of Booster reset clock events

:0007EDE2   0C10 520B 1F1D 0C10 520B 1F1D 0C10 5222  recent EVTLOG entries in node06B6
:0007EDF2   0B1F 3C1D 0C10 520B 1F1D 0C10 521A 0B1F

:0007EE02   0A87 01E0 004B 0000 0343 3018 3722 036C  recent RETDLOG entries in node06B6
:0007EE12   0A87 0348 004B 0000 037E 3018 3722 076F
:0007EE22   0A87 036A 004B 0000 0383 3018 3722 090C
:0007EE32   0A87 03C2 004B 0000 0390 3018 3722 0984

:0007EE42   0018 78F2 002E A6CF C010 3020 0551 050B  recent NETFRAME records in node06B6
:0007EE52   0018 793A 002E A6CF C010 3020 0552 050A  (buffer, size, node, I/O, time)
:0007EE62   0018 7982 002E A6CF C010 3020 0553 050B  “C0” = output, “C1” = input
:0007EE72   0018 79CA 002E A6CF C010 3020 0554 050A
:0007EE82   0018 7A12 002E A6CF C010 3020 0555 050A
:0007EE92   0018 7A5A 002E A6CF C010 3020 0556 050B
:0007EEA2   0018 7AA2 002E A6CF C010 3020 0557 050A
:0007EEB2   0018 7AEA 002E A6CF C010 3020 0558 050B
:0007EEC2   0018 7B32 002E A6CF C010 3020 0559 050A
:0007EED2   0018 7B7A 002E A6CF C010 3020 0600 050A
:0007EEE2   002D 3622 002A C508 C110 3020 0600 113E
:0007EEF2   002D 3C22 002A C508 C110 3020 0600 1232
:0007EF02   0018 7BC2 002E A6CF C010 3020 0601 050C  :01-05 reply
:0007EF12   002D 4222 002A C508 C110 3020 0601 064E
:0007EF22   002D 4822 0018 E040 C110 3020 0601 0666
:0007EF32   0018 7C0A 0018 E0B0 C010 3020 0601 070B
:0007EF42   002D 4E22 0018 06B6 C110 3020 0601 070C
:0007EF52   002D 5422 002A C508 C110 3020 0601 115B
:0007EF62   0018 7C3C 002C A6CF C010 3020 0602 0006
:0007EF72   002D 5A22 002A A6CF C110 3020 0602 0014
:0007EF82   002D 6022 002A C508 C110 3020 0602 0144
:0007EF92   0018 7C82 002E A6CF C010 3020 0602 050A  :02-05 reply
:0007EFA2   0018 7CCA 002E A6CF C010 3020 0603 050B  :03-05 reply
:0007EFB2   002D 6622 004E C509 C110 3020 0603 070E

:0007EFC2   0019 3A98 002A A6C2 C010 3020 0602 0015  recent NETFRAME records in node06CF
:0007EFD2   0019 3ADC 002A A6B1 C010 3020 0602 0016
:0007EFE2   0019 3B20 002A A6B4 C010 3020 0602 0016
:0007EFF2   0019 3B64 002A A6BC C010 3020 0602 0016
:0007F002   0019 3BA8 002A A6B7 C010 3020 0602 0017
:0007F012   0019 3BEC 002A A6B5 C010 3020 0602 0017
:0007F022   0019 3C30 002A A6BE C010 3020 0602 0017
:0007F032   0019 3C74 002A A6BA C010 3020 0602 0018
:0007F042   0019 3CB8 002A A6C1 C010 3020 0602 0018
:0007F052   0019 3CFC 002A A6BF C010 3020 0602 0019
:0007F062   0019 3D40 002A A6BB C010 3020 0602 0019
:0007F072   0019 3D84 002A A669 C010 3020 0602 0019
:0007F082   0019 3DC8 002A A6C0 C010 3020 0602 001A
:0007F092   0019 3E0C 002A A6B3 C010 3020 0602 001A
:0007F0A2   0019 3E50 002A A6BD C010 3020 0602 001A
:0007F0B2   0019 3E94 0064 C509 C010 3020 0602 0106
:0007F0C2   002F 3A22 002A C508 C110 3020 0602 0144
:0007F0D2   0019 3F12 0064 C509 C010 3020 0602 0205
:0007F0E2   0019 3F90 0064 C509 C010 3020 0602 0305
:0007F0F2   0019 400E 0064 C509 C010 3020 0602 0404
:0007F102   0019 408C 0064 C509 C010 3020 0602 0505
:0007F112   002F 4022 002E A6BD C110 3020 0602 0509  18 replying nodes at :02-05
:0007F122   002F 4622 002E A6BF C110 3020 0602 0509  (2 bytes of data from each node)
:0007F132   002F 4C22 002E A6BB C110 3020 0602 0509  “
:0007F142   002F 5222 002E A6BA C110 3020 0602 050A  “
:0007F152   002F 5822 002E A6C2 C110 3020 0602 050A  “
:0007F162   002F 5E22 002E A6C0 C110 3020 0602 050A  “
:0007F172   002F 6422 002E A6BE C110 3020 0602 050A  “
:0007F182   002F 6A22 002E A6B2 C110 3020 0602 050A  “
:0007F192   002F 7022 002E A6B1 C110 3020 0602 050B  “
:0007F1A2   002F 7622 002E A6B3 C110 3020 0602 050B  “
:0007F1B2   002F 7C22 002E A6B4 C110 3020 0602 050B  “
:0007F1C2   002F 8222 002E A6BC C110 3020 0602 050B  “
:0007F1D2   002F 8822 002E A6B7 C110 3020 0602 050C  “
:0007F1E2   002F 8E22 002E A6C1 C110 3020 0602 050C  “
:0007F1F2   002F 9422 002E A6B9 C110 3020 0602 050C  “
:0007F202   002F 9A22 002E A669 C110 3020 0602 050C  “
:0007F212   002F A022 002E A6B6 C110 3020 0602 050C  (06B6 reply ok)
```

```
:0007F222   002F A622 002E A6B5 C110 3020 0602 0512  " (late cleanup reply from node06B5)
:0007F232   0019 410A 0076 AC5D C010 3020 0602 0551  :02-05 composite reply no error
:0007F242   0019 419A 0064 C509 C010 3020 0602 0606  (15 Hz monitoring by node0509 running MISS)
:0007F252   0019 4218 0064 C509 C010 3020 0602 0705
:0007F262   0019 4296 0064 C509 C010 3020 0602 0805
:0007F272   002F AC22 002E A6B8 C110 3020 0602 0807  (reply from node06B8 out of sync)
:0007F282   0019 4314 0064 C509 C010 3020 0602 0905
:0007F292   0019 4392 0064 C509 C010 3020 0602 1005
:0007F2A2   0019 4410 0064 C509 C010 3020 0602 1105
:0007F2B2   0019 448E 0064 C509 C010 3020 0602 1204
:0007F2C2   002F B222 002E A6B1 C110 3020 0602 1207
:0007F2D2   002F B822 002E A6CE C110 3020 0602 1207
:0007F2E2   0019 450C 0032 AC5D C010 3020 0602 1251
:0007F2F2   0019 4558 0064 C509 C010 3020 0602 1305
:0007F302   0019 45D6 0064 C509 C010 3020 0602 1405
:0007F312   002F BE22 002E A6CE C110 3020 0602 1408  (1F reply from node06CE)
:0007F322   0019 4654 002E AC5D C010 3020 0602 1451
:0007F332   0019 469C 0064 C509 C010 3020 0603 0006
:0007F342   002F C422 0048 A6CE C110 3020 0603 0009  (1F, 1D replies from node06CE)
:0007F352   0019 471A 0048 AC5D C010 3020 0603 0051  "
:0007F362   0019 477C 0064 C509 C010 3020 0603 0105
:0007F372   002F CA22 0048 A6CE C110 3020 0603 0108  (1F, 1D replies from node06CE)
:0007F382   0019 47FA 0048 AC5D C010 3020 0603 0151  "
:0007F392   0019 485C 0064 C509 C010 3020 0603 0205
:0007F3A2   002F D022 0048 A6CE C110 3020 0603 0208  (1F, 1D replies from node06CE)
:0007F3B2   0019 48DA 0048 AC5D C010 3020 0603 0252  "
:0007F3C2   0019 493C 0064 C509 C010 3020 0603 0305
:0007F3D2   002F D622 0048 A6CE C110 3020 0603 0308  (1F, 1D replies from node06CE)
:0007F3E2   0019 49BA 0048 AC5D C010 3020 0603 0351  "
:0007F3F2   0019 4A1C 0064 C509 C010 3020 0603 0405
:0007F402   002F DC22 0048 A6CE C110 3020 0603 0408  (1F, 1D replies from node06CE)
:0007F412   0019 4A9A 0048 AC5D C010 3020 0603 0451  "
:0007F422   0019 4AFC 0064 C509 C010 3020 0603 0505
:0007F432   002F E222 002E A6BD C110 3020 0603 050A  17 replying station nodes
:0007F442   002F E822 002E A6BF C110 3020 0603 050A  (but not node06B6)
:0007F452   002F EE22 002E A6BB C110 3020 0603 050A  "
:0007F462   002F F422 002E A6BA C110 3020 0603 050B  "
:0007F472   0028 0222 002E A6C0 C110 3020 0603 050B  "
:0007F482   0028 0822 002E A6C2 C110 3020 0603 050B  "
:0007F492   0028 0E22 002E A6BE C110 3020 0603 050B  "
:0007F4A2   0028 1422 002E A6B1 C110 3020 0603 050B  "
:0007F4B2   0028 1A22 002E A6C1 C110 3020 0603 050C  "
:0007F4C2   0028 2022 002E A6B4 C110 3020 0603 050C  "
:0007F4D2   0028 2622 002E A6B2 C110 3020 0603 050C  "
:0007F4E2   0028 2C22 002E A6BC C110 3020 0603 050C  "
:0007F4F2   0028 3222 002E A6B7 C110 3020 0603 050C  "
:0007F502   0028 3822 002E A6B3 C110 3020 0603 050D  "
:0007F512   0028 3E22 0048 A6CE C110 3020 0603 050D    (1F, 1D replies from node06CE)
:0007F522   0028 4422 002E A6B9 C110 3020 0603 050D  "
:0007F532   0028 4A22 002E A669 C110 3020 0603 050D  "
:0007F542   0028 5022 002E A6B5 C110 3020 0603 0513  " (late cleanup reply from node06B5)
:0007F552   0019 4B7A 00A6 AC5D C010 3020 0603 0552  :03-05 composite reply "36 –7" error
:0007F562   0019 4C3A 0064 C509 C010 3020 0603 0605
:0007F572   0028 5622 0048 A6CE C110 3020 0603 0608  (1F, 1D replies from node06CE)
:0007F582   0019 4CB8 0048 AC5D C010 3020 0603 0651  "
:0007F592   0019 4D1A 0064 C509 C010 3020 0603 0705
:0007F5A2   0028 5C22 0048 A6CE C110 3020 0603 0708  (1F, 1D replies from node06CE)
:0007F5B2   0028 6222 002A C509 C110 3020 0603 070B  request for this data from node0509
```

An example of the ACNETERR data stream records in node06CF:

```
:0000C9D0   0C5D 0004 F824 06CE 0510 3001 0020 0052
:0000C9E0   0C5D 004C F924 06B8 0510 3003 3524 1151  *
:0000C9F0   0C5D 004C F924 06B4 0510 3008 1329 0551  FTC
:0000CA00   0C5D 004C F924 06B8 0510 3016 2238 1051  *
:0000CA10   0C5D 004C F924 06B6 0510 3020 0603 0551  FTC
:0000CA20   0C5D 004C F924 06B8 0510 3107 1832 0250  *
:0000CA30   0C5D 004C F924 06B7 0510 3112 0101 0951  FTC (7)
:0000CA40   0C5D 004C F924 06B7 0510 3112 0102 0951  FTC (8)
:0000CA50   0C5D 004C F924 06B8 0510 3112 0103 0951  FTC
:0000CA60   0C5D 004C F924 06B8 0510 3114 1312 1050  FTC
:0000CA70   0C5D 004C F924 06B8 0510 3114 1313 1050  FTC
:0000CA80   0C5D 004C F924 06B8 0510 3116 4907 1151  (2 second problem?)
:0000CA90   0C5D 0004 F924 06CE 0510 3116 4907 1250
:0000CAA0   0C5D 0004 F924 06CE 0510 3116 4907 1251
:0000CAB0   0C5D 0004 F924 06CE 0510 3116 4907 1351
:0000CAC0   0C5D 0004 F924 06CE 0510 3116 4907 1351
:0000CAD0   0C5D 0004 F924 06CE 0510 3116 4907 1450
:0000CAE0   0C5D 0004 F924 06CE 0510 3116 4908 0051
:0000CAF0   0C5D 0004 F924 06CE 0510 3116 4908 0151
:0000CB00   0C5D 0004 F924 06CE 0510 3116 4908 0250
:0000CB10   0C5D 0004 F924 06CE 0510 3116 4908 0351
:0000CB20   0C5D 0004 F924 06CE 0510 3116 4908 0450
```

```
:0000CB30   0C5D 0008 F924 06CE 0510 3116 4908 0450
:0000CB40   0C5D 0004 F924 06CE 0510 3116 4908 0551
:0000CB50   0C5D 0008 F924 06CE 0510 3116 4908 0551
:0000CB60   0C5D 004C F924 06B8 0510 3116 4908 1150
:0000CB70   0C5D 004C F924 06B6 0510 3122 3613 1451
```

An example of `GetDat` status errors in monitoring 15 Hz replies from `node06CF`:

```
:00080F82   0007 1030 0545 0405 0000 1030 0545 0406
:00080F92   0007 1030 0545 0407 0000 1030 0545 0408
:00080FA2   0007 1030 0545 0409 0000 1030 0545 0410
:00080FB2   0007 1030 0545 0411 0000 1030 0545 0412
:00080FC2   0007 1030 1624 3504 0000 1030 1624 3505  (1624-1626 problem)
:00080FD2   0007 1030 1624 4909 0000 1030 1624 4910
:00080FE2   0007 1030 1624 5712 0000 1030 1624 5713
:00080FF2   0007 1030 1625 2306 0000 1030 1625 2307
:00081002   0007 1030 1625 2607 0000 1030 1625 2608
:00081012   0007 1030 1625 3109 0000 1030 1625 3110
:00081022   0007 1030 1625 3410 0000 1030 1625 3411
:00081032   0007 1030 1625 4213 0000 1030 1625 4214
:00081042   0007 1030 1625 4514 0000 1030 1625 4600
:00081052   0007 1030 1625 4900 0000 1030 1625 4901
:00081062   0007 1030 1625 5201 0000 1030 1625 5202
:00081072   0007 1030 1626 0004 0000 1030 1626 0005
:00081082   0007 1030 1626 0506 0000 1030 1626 0507
:00081092   0007 1030 1626 0807 0000 1030 1626 0808
:000810A2   0007 1030 1626 1108 0000 1030 1626 1109
:000810B2   0007 1030 1626 2212 0000 1030 1626 2213
:00080DD2   0007 1030 1626 2513 0000 1030 1626 2514
:00080DE2   0007 1030 1626 2713 0000 1030 1626 2714
:00080DF2   0007 1030 2152 3908 0000 1030 2152 3909
:00080E02   0007 1031 1024 0302 0000 1031 1024 0303
:00080E02   0007 1031 1532 0710 0000 1031 1532 0809   14 consecutive replies missing
:00080E02   0007 1031 1532 0913 0000 1031 1532 1010   12 "
```

An example of anomalous 15 Hz interrupt activity in `node06B8`:

```
:0000B350   0000 0000 0002 08F0 0510 3015 2234 101A  gap between interrupts = 0x208F0 μs
:0000B360   0000 0000 0002 08B1 0510 3016 0436 011A  (normal gap = 66666 μs = 0x1046A)
:0000B370   0000 0000 0002 08C9 0510 3016 1135 021B
:0000B380   0000 0000 0002 08F8 0510 3016 2044 101A
:0000B390   0000 0000 0002 08CB 0510 3016 2238 051B
:0000B3A0   0000 0000 0002 0905 0510 3017 0014 141B
:0000B3B0   0000 0000 0002 08A7 0510 3017 5447 021C
:0000B3C0   0000 0000 0002 0911 0510 3019 3428 031B
:0000B3D0   0000 0000 0002 08CE 0510 3019 4520 111B
:0000B3E0   0000 0000 0002 08AA 0510 3019 5036 121B
:0000B3F0   0000 0000 0002 08E5 0510 3020 1903 041A
:0000B030   0000 0000 0002 08CC 0510 3021 5412 091B
:0000B040   0000 0000 0002 08BF 0510 3022 3341 111B
:0000B050   0000 0000 0002 0921 0510 3023 1223 061B
:0000B060   0000 0000 0002 08F9 0510 3100 3909 121B
:0000B070   0000 0000 0002 090A 0510 3100 5719 031B
:0000B080   0000 0000 0002 091C 0510 3101 2037 041A
:0000B090   0000 0000 0002 090E 0510 3102 1925 141B
:0000B0A0   0000 0000 0002 091E 0510 3102 2757 021B
:0000B0B0   0000 0000 0002 08FE 0510 3102 3411 141B
:0000B0C0   0000 0000 0002 08EE 0510 3102 4748 041A
:0000B0D0   0000 0000 0002 08DB 0510 3104 3013 141B
:0000B0E0   0000 0000 0002 08F3 0510 3104 3627 071B
:0000B0F0   0000 0000 0002 08D1 0510 3105 4525 051B
:0000B100   0000 0000 0002 0903 0510 3106 0446 121B
:0000B110   0000 0000 0002 08B2 0510 3107 1831 101A
:0000B120   0000 0000 0002 08A2 0510 3108 2813 121B
:0000B130   0000 0000 0002 08D8 0510 3108 3543 121B
:0000B140   0000 0000 0002 08CB 0510 3109 3215 091B
:0000B150   0000 0000 0002 08C5 0510 3109 3401 061B
:0000B160   0000 0000 0002 08E0 0510 3110 5718 081B
:0000B170   0000 0000 0002 08DD 0510 3111 4340 141B
:0000B180   0000 0000 0002 090B 0510 3114 3031 101A
:0000B190   0000 0000 0002 08DE 0510 3114 3126 131A
:0000B1A0   0000 0000 0002 08DD 0510 3117 1340 141B
:0000B1B0   0000 0000 0002 08E9 0510 3118 1344 021B
:0000B1C0   0000 0000 0002 08A4 0510 3120 5707 131A
:0000B1D0   0000 0000 0002 08BB 0510 3122 1611 001B
:0000B1E0   0000 0000 0002 08C7 0510 3122 2427 021B
:0000B1F0   0000 0000 0002 08C2 0510 3122 2618 141B
:0000B200   0000 0000 0002 0892 0510 3122 5559 011A
:0000B210   0000 0000 0002 08C9 0510 3123 2038 011A
:0000B220   0000 0000 0002 08C9 0510 3123 2311 081B
:0000B230   0000 0000 0002 08CD 0510 3123 2426 121B
```