Acnet Request Timing Example

Network support efficiency

Mon, Jul 27, 1998

Introduction

This note describes some diagnostic results that illustrate internal IRM system processing activities associated with handling an Acnet data request for a few bytes of data from all 17 Booster HLRF stations. The server node receives the data request, forwards the same request to a multicast address so that all stations receive a copy of the request at once, then receives replies from all stations. At the time of this particular test, one HLRF station was off, which meant that one station could not respond. Because of that, the time of the first composite reply to the requesting Vax console is delayed.

Frame diagnostics

Here is the network frame diagnostics data for the period of interest:

```
E2C1 0140 R 16EC22 1427:35-04+37    Receive request from console
E072 0140 T 186940 1427:35-04+38    Forward request to multicast
E141 0040 R 16F222 1427:35-04+41    Receive 1st reply
E0B1 0040 R 16F822 1427:35-04+41    Receive 2nd reply
E0E1 0040 R 16FE22 1427:35-04+42    etc.
E161 0040 R 170422 1427:35-04+42
E101 0040 R 170A22 1427:35-04+42
E171 0040 R 171022 1427:35-04+42
E051 0040 R 171622 1427:35-04+42
E091 0040 R 171C22 1427:35-04+42
E0F1 0040 R 172222 1427:35-04+42
E0D1 0040 R 172822 1427:35-04+42
E191 0040 R 172E22 1427:35-04+43
E131 0040 R 173422 1427:35-04+43
E151 0040 R 173A22 1427:35-04+43
E111 0040 R 174022 1427:35-04+43
E0A1 0040 R 174622 1427:35-04+43
E0C1 0040 R 174C22 1427:35-04+43    Receive 16th reply
E1F1 0054 R 175222 1427:35-04+49    ? (unrelated frames)
E1F1 002C R 175822 1427:35-04+59    ?
E1D1 0220 T 186A9A 1427:35-05+ 3    ?
E2C1 01A0 T 186CD4 1427:35-06+41    Deliver composite reply
```

Task timing diagnostics

Here is the Task Timing diagnostic data for the period of interest:

```
IDLE   0000  15.71 1427:35-04+21
SNAP   0000    .26 1427:35-04+37    UDP/IP processing of request
ANet   0000    .1  1427:35-04+37    Acnet header processing
ACRq   0000    .63 1427:35-04+37    RETDAT processing
Updt   0000    .4  1427:35-04+38    Build forwarded request
QMon   0000    .07 1427:35-04+38    Clean up
IDLE   0000   1.48 1427:35-04+38
Cons   0000    .07 1427:35-04+40    (little console--unrelated)
Serv   0000    .09 1427:35-04+40    (server task--unrelated)
IDLE   0000    .38 1427:35-04+40
SNAP   0000   2.93 1427:35-04+40    UDP/IP processing for 16 replies
```

```
ANet   0000    .78 1427:35-04+43    Acnet header processing for 16 replies
Serl   0000    .59 1427:35-04+44    (serial task--unrelated)
ACRq   0000   1.12 1427:35-04+45    RETDAT processing of 16 replies
IDLE   0000   1.86 1427:35-04+46

(much later)
Updt   0000    .44 1427:35-06+41    Build composite reply, queue to net
QMon   0000    .07 1427:35-06+41    Clean up
```

Analysis

Note that the processing of network replies from the 16 responding nodes is handled by SNAP, which performs IP and UDP processing and is responsible for the time-stamps, and is completed in 3 ms. The Booster HLRF nodes run synchronously, so they transmit their replies at nearly the same time. (Ethernet switch hardware helps to limit degradation due to collision resolution.) They arrive at so nearly the same time, with each ethernet receive interrupt requiring about 40 microseconds, that SNAP remains active until all are processed. Each ethernet receive interrupt causes a reference to the frame to be written to the message queue awaited by SNAP. By the time SNAP had processed the first reply frame, it found another one waiting in the message queue to be processed. This continued for all 16 frames that were received. The result of such processing was that references to each datagram message were posted to the message queue upon which ANet, the task that processes Acnet headers, awaits. So ANet finds 16 reply messages to process, each of which in turn causes a reference to be placed into the message queue awaited by the ACReq task, which performs RETDAT processing of all 16 reply messages.

In rough summary, the original console request was completely processed in about 1.5 ms. After another 1.5 ms of idle time, the 16 replies were received and processed completely in about 5 ms. After being patient for about 2 more cycles hoping to receive the 17th reply that never arrived, the composite reply was built and queued to the network in about 0.5 ms. The total CPU processing time for all this activity seen at the server node was about 7 ms.

From prior studies, if all 17 stations had replied to the server, we would still have seen the request turned around by the server in less than 10 ms, measured from the time that the initial request was being processed by SNAP until the time that the composite reply was queued to be returned to the console.