

A Few Observations on NCP Statistics

The NCP in use at HARV-10, CMU-10A, and CMU-10B collects a number of operating and error statistics, which may be typed out on demand by any user by means of the 'IMP ERROR' command, as shown on the sample typescript.

The figures shown cover the period since the system was last restarted. They are not logged or recorded in any more permanent form due to extremely limited on-line storage at HARV-10. where the software was implemented. However, due to the small size of the system and infrequent monitor development work, HARV-10 tends to stay up for periods approaching the interval between hardware maintenance, which is one week. The attached output was obtained after 168 hours system uptime.

There are a few things I would like to point out that may be of interest to NCP implementers.

First, note that the number of discarded (unexpected) RFNMs is equal to the number of simulated (timed out) RFNMs. This has been the case almost every time I have looked at these statistics. It suggests that the RFNMs are not being lost but are rather delayed beyond the NCP timeout interval, which I believe is 30 seconds.

I have heard talk among a few people in the Network community about "lost RFNMs", and would like to suggest this as a possible alternative explanation. Perhaps longer timeouts are in order.

Second, the observed ratio of received allocates to transmitted allocates (on the order of two to one) is also fairly typical. I believe this reflects differences in allocation strategies among various hosts.

Many hosts appear to send out an allocate for every data message received. While this is reasonable for connections such as FTP data transfer connections, it imposes considerable extra traffic in the case of the single character messages that seem to be the most common on the network.

The strategy used by the Harvard NCP is to assign a "desired level of allocation" figure to each socket (typically quite small for Telnet connections and large for FTP data connections; it is a user program settable parameter). When the actual allocation for the socket falls below 50% of this level, enough additional allocation is sent to bring it up to the full "desired level".

The effect of this strategy is to significantly reduce the number of allocates returned for a given number of small messages received. This reduces both network traffic and control message overhead at the other end. The strategy has no effect on FTP data messages, since each message is usually large enough to reduce outstanding allocation by at least half at a single blow.

Finally, I should remark on the appallingly large number of NOPs received (typically 25% of all control messages). Most of these seem to be piggy-backed onto other control messages, so the situation is not as awful as the figures would indicate. Nevertheless, I am forced to wonder why anyone would want to send so many.

TELNET typescript file started at THU 31 JAN 74 428:05

#harv-10 (settings loaded) is complete.#

Harvard 5.06A-18 7:28:38

Type "HELP" if you need it.

.login 62,#

JOB 2 Harvard 5.06A-18 TTY25

Your name please (last name first): Taft

You are logged in as 62,404000

0728 31-Jan-74 Thur

SCHEDULED PM ON THURSDAYS, 0830-1200 EOT

.imp error

NCP version 1573.1604 operating statistics

07:29:02 31-JAN-74

NCP (link 0) message errors:

Socket not found: 2184

Improper state: 323
Illegal message type: 2
Last discarded allocation from PARC-MAXC (XEROX) link 12
Timed-out exec ICPs: 3

NCP messages:

Type Received Sent

NOP 81850 0
RTS 3688 2507
STR 2388 3562
CLS 6055 6059
ALL 183050 101442
GVB 772 0
RET 0 772
INS 109 0
ECO 7472 15426
ERP 15065 7472
ERR 2 0
RST 2782 226
RRP 162 2782

Received NCP error messages:

Type Count

4 2

Most recent error: type 4 from UCLA-CCN

Data (octal) 4 74 0 10 0 0 74 254 0 200

(decimal) 4 60 0 8 0 0 60 172 0 128

IMP data message faults:

Hardware fault: 2

Link not found: 8

Discarded RFNMs: 10

Simulated (timed out) RFNMs: 10

Received IMP messages:

Regular 590812

Err w/o id 3

NOP 4

RFNM 490095

Dest dead 366

Inc trans 52

IMP reset 2

Histogram of received data message sizes

Bits Count

<1 3

<16 146834

<32 39751

<64 7044
<128 196983
<256 46099
<512 147609
<1024 534
<2048 1820
<4096 1152
<8192 2979

72 free buffers
7% average buffer utilization

.kjob/k
Job 2, User [62,404000] Logged off TTY25 0729 31-Jan-74
Runtime 0 Min, 03.29 Sec