Packet White Paper: 109
Paula Dowie
July 2001

NET/ROM DATA MULTIPLEXING

INTRODUCTION

In order to provide multiple "services" on a single Net/Rom node, each service (e.g. BBS, PMS, Chat server, DX Cluster, HTTP, game server etc) requires a different Secondary Station Identifier (SSID).

As there are only 16 possible SSIDs, and therefore services, per node callsign, this limitation could stifle future development of novel services.

Additionally, every service that is to be Net/Rom Layer 4 (L4) connectable must have an entry in the Net/Rom "Nodes table", and those entries must be propagated via nodes broadcasts. Just 20 nodes, each with eight services, would use a total of 160 entries. A table of that size is slow to propagate over RF links, making it difficult for the network to achieve stability. It is also cumbersome for users to view.

Finally, there is no agreed standard for SSID numbers, and the Net/Rom "Alias" often gives no clue to what service is associated with it.

This document proposes a fairly trivial solution to the above problems. If implemented, it would allow vastly more services to be carried over Net/Rom, with standardised service addresses, and zero cost to the nodes table size. In fact, table sizes could be reduced, because a single SSID could support all services.

MOTIVATION

The primary motivation for this proposal was a desire to allow novel services, such as HTTP, SMS, APRS, gaming and file transfer, to name but a few, to be transported over the Packet Radio network. It may however also help alleviate other problems, as explained below.

Over recent years, thanks to better connectivity and new software which allows larger nodes table sizes, the amount of routing information being moved around the network is increasing rapidly.

This is consuming valuable bandwidth, and if the trend continues, modelling suggests that the network will fail to converge to a stable, accurate state after a perturbation. By the time the last "nodes broadcast" packet has been sent, the table is already out of date.

If a node user issues a N(odes) command, it may take several minutes to transmit the whole nodes table over congested RF links, and it has been observed that many users simply disconnect rather than wait for the whole table to download.

A typical nodes table may contain multiple entries for the same node, each with a different alias and SSID, one for each service hosted by that node. There is no consistent standard for these addresses, leading to confusion, and wasted connections.

As more services are added, this situation can only get worse, but this proposal offers a mitigation.

DISCUSSION

Although Net/Rom may not be the BEST networking protocol, it has stood the test of time and has, despite the lack of a formal protocol specification, become the de-facto standard for AX25 networking. Any attempt to replace Net/Rom would probably be doomed to failure.

In order to expedite the development of novel services, a way must therefore be found to transport them over Net/Rom, without breaking anything.

In contrast with Net/Rom, which only allows one service per address (i.e. callsign-SSID), with only 16 possible addresses per node, TCP and UDP allow up to 65536 different services per IP address. Furthermore, the important service numbers are well known and standardised.

It is unlikely that Packet Radio would ever need to carry as many as 65536 different services per node, but there is certainly a need for more than 16.

Using TCP-like "service numbers" rather than SSIDs would allow both simplification of the nodes table, and standardisation of the service numbers. For example, service 0 could be a normal connection to the node's command line, service 1 might be an information server, service 2 could be the Sysop's PMS, service 21 might be file transfer, and service 80 could be an HTTP server.

So why not simply use TCP/IP over AX25?

TCP/IP is a verbose protocol, which does not work well over low bandwidth, unpredictable RF links. In addition, it requires coordination, and knowledge that many sysops do not posses, nor wish to posses. It has its niche, but will never displace Net/Rom.

Net/Rom needs to be more like TCP/IP, without being TCP/IP.

All that is required to achieve this is for the connection initiator to specify a service port number, and for the connectee to initiate different types of service depending on that number.

The service port number is only required during circuit set up. After that, the regular Net/Rom circuit numbers (along with callsigns) are sufficient to identify the circuit.

THE PROPOSAL

The proposed solution is to add an "extended" connect request to the list of Net/Rom L4 opcodes. This request would include the target service number. The suggested mnemonic is CREQX.

Such a packet should pass unmolested through intermediate Layer 3 (L3) routers, because the L4 header is of no concern to L3. It should in theory be ignored by legacy L4 endpoints, if they are correctly written according to the "Robustness Principle".

Why Add Another Opcode?

There is a "reserved" flag in the opcode/flags byte, but the Net/Rom designers may have earmarked that for future use, so we can't use it.

Additional bytes could be appended to a normal CREQ, but that might break other people's software.

So the safest option is to add another opcode.

The proposed opcode for CREQX is 8, and the unused sequence number fields of a "regular" CREQ could carry the requested service number, as shown in the following comparison between CREQ and CREQX:

Normal CREQ:

```
   ----------------------------------------------------------
   | id | ndx |   unused    | 1 | win | usercall | nodecall |
   ----------------------------------------------------------
     1     1        2         1    1      7           7  Bytes
```

Extended CREQ:

```
   ----------------------------------------------------------
   | id | ndx | svch | svcl | 8 | win | usercall | nodecall |
   ----------------------------------------------------------
     1     1     1      1     1    1      7           7  Bytes
```

Where "id" and "ndx" are the senders circuit ID and index as usual, "svch" and "svcl" are the high and low bytes of the target service number, 8 is the new opcode, and the remainder is as normal.

RECEIVE PROCESSING

Upon receipt of CREQX, the receiving node checks whether or not the requested service is available. If it is, a CACK (Connection Acknowledgement) is sent in the usual way, and the connection is identical to "normal" Net/Rom.

If the requested service is not available, not implemented or busy, the connection is rejected in the usual way.