DXSpider PC Packet Cluster Protocol

Legacy Protocols

Topic	Protocol
Talk mode	PC10^from-user^to-user^msg^bell-flag^ ^from-node^~ PC10^from-user^route-via-node^msg^bell-flag^to-user^origin-node^~
DX info	PC11^DXfreq^DXcall^date^time^comment-txt^user-rprt^origin-node^hops^~
Announcement	PC12^from-user^route-to-node^msg^sysop-flg^origin-node^wx-flg^hops^~
Stn into CONF¹	PC13^user^hops^
Stn out of CONF¹	PC14^user^hops^
Conference Mode	PC15^from-user^msg^hops^~
PC user add	PC16^node^user talk-mode here^user talk-mode here^...^hops^
PC user delete	PC17^user^node^hops^
PC initialization: RequestInit	PC18^cluster info^ver^
PC initialization: NodeAdd	PC19^here^node^talk^ver^...^hops^
PC initialization: InitDone	PC20^
PC initialization: NodeDelete	PC21^node^reason^hops^
PC initialization: PCDone	PC22^
WWV info	PC23^date^hour^SFI^A^K^forecast^logger^origin-node^hops^~
Here status info	PC24^user^here^hops^
DX/WWV merge req	PC25^route-to-node^route-from-node^DX-cnt^WWV-cnt^
Merge DX info	PC26^DXfreq^DXcall^date^time^comment-txt^spotter^origin-node^ ^~
Merge WWV info	PC27^date^hour^SFI^A^K^forecast^logger^origin-node^ ^~
PC Mail: SendSubject	PC28^route-to-node^route-from-node^to-user^from-user^date^time^private-flg^subject^bbs^no-lines^rr-flg^via-node^origin-node^~
PC Mail: SendText	PC29^route-to-node^route-from-node^stream-no^text^~
PC Mail: AckSubject	PC30^route-to-node^route-from-node^stream-no^
PC Mail: AckText	PC31^route-to-node^route-from-node^stream-no^
PC Mail: CompleteText	PC32^route-to-node^route-from-node^stream-no^
PC Mail: AckCompleteText	PC33^route-to-node^route-from-node^stream-no^
Remote commands: Command	PC34^route-to-node^route-from-node^cmd^~
Remote commands: Response	PC35^route-to-node^route-from-node^cmd-resp^~
Remote commands: Show Command	PC36^route-to-node^route-from-node^cmd^~

Remote commands: Needs db update	PC37^route-to-node^route-from-node^user^stream-no^cmd^~
PC initialization: Connected nodes	PC38^node,node,...^~
NodeDelete w/Discon	PC39^node^reason^
PC file forward	PC40^route-to-node^route-from-node^filename^bulletin^linecnt^
User info	PC41^user^type^info^hops^~
Forwarding abort	PC42^route-to-node^route-from-node^stream-no^
Remote DB request	PC44^route-to-node^route-from-node^stream-no^qualifier^key^user^
Remote DB response	PC45^route-to-node^route-from-node^stream-no^info^~
Remote DB complete	PC46^route-to-node^route-from-node^stream-no^
Remote DB update	PC47^route-to-node^route-from-node^user^qualifier^key^stream-no^
Remote userDB req	PC48^route-to-node^route-from-node^stream-no^qualifier^key^user^
Bulletin delete	PC49^from-user^subject^hops^
Local User count	PC50^node^user-count^hops^
Ping	PC51^route-to-node^route-from-node^ping-flag^
WCY Info	PC73^date^hour^SFI^A^K^expK^R^SA^GMF^aurora^logger^origin-node^hops^
Unknown[2]	PC75
CLX Remote CMD Send	PC84^to-node^from-node^call^msg^~
CLX Remote CMD Reply	PC85^to-node^from-node^call^msg^~

[1] Not found in DXSpider source code.

[2] Found in DXProtHandle.pm - Line 1705: #dunno but route it

New protocols

Topic	Protocol	Description
DX Info	PC61^DXfreq^DXcall^date^time^comment-txt^user-rprt^origin-node^user-ip^hops^~	P61 is a replacement for PC11 originating with Lee Sawkins, VE7CC. PC61 differs by allowing several significant digits after the frequency decimal. All other fields are the same, except for an added originating user IP address. The user IP can be IPv4 or IPv6.
General Info	PC9x^<node call>^<seconds since last midnight>^...	<seconds since last midnight> is the number of seconds since the beginning of "today" UTC. It *MUST* be unique and increase with each sentence. It can therefore be a decimal number. The idea being that if the next sentence is sent in the same second as the last one, then you append (for instance) '.01' on the end. If there is an other one: .02 etc. But you add any number of decimal points (eg '.1' or '.001') you like. Look at gen_pc9x_t in DXProtHandle.pm for an example. At midnight, they all go back to 0.
Routing	PC92^GB7TLH^78031^C^5GB7TLH:5457^1G1TLH-2:XX.XX.XX.XX^5GB7DJK^H99^	See breakdown below
Talk/Announce/WX	PC93^<node call>^<timestamp>^<to>^<from>^<via>^<text>[^<onode>][^<IPaddr>]^H99	See breakdown below

PC92 - Routing

The biggest change is that nodes only (ever) send their configuration (using a PC92) and the configuration of any *directly connected* "traditionally routing" nodes (TNODES) (ie still using PC16,17,19,21). The config stored of any TNODES is limited only to the local users on those nodes. Anything else is kept as hints, but is not transmitted onward, neither to other PC9x nodes nor other TNODES. The only configuration that other TNODES will see are the composite of all the PC92 nodes's configs + any other locally connected TNODES.

PC92 Nodes periodically output their configuration. Failure to receive a config after 3 update periods will cause that node's config to be erased (and the changes to be propagated to any connected TNODEs). Think OSPF.

All PC9x sentences contain a timestamp and the originating node call. This allows any of these sentences to be deduplicated and deliberate loops (for routing and new other functions) will allow things to continue to work as only "new" PC9x sentences will be processed.

All PC9x sentences are passed onto neighbouring PC9x unaltered apart from a decremented hop count.

Breakdown

Ex. PC92^GB7TLH^78031^C^5GB7TLH:5457^1G1TLH-2:XX.XX.XX.XX^5GB7DJK^H99^

GB7TLH - Originating Node

78031 - Timestamp - seconds since midnight

'C' current configuration record

'A' records add nodes or users

'D' records delete nodes or users.

'K' keep alive records

5GB7TLH:5457

5 is a bit map which is made up like this:-

Bit 1 - here/not here

Bit 2 - external, traditional PC protocol node

Bit 4 - Node

1 - User/here

4 - Node/not here - here flag is unset

5 - Node/here

7 - Traditional PC node/here

GB7TLH is the callsign

:5457 is the version number, other fields may be tacked on (see K record below)

The first slot after the 'C', 'A' or 'D', is always reserved for the node call. This slot may be empty (as in ...^A^^1G1TLH^... as opposed to ..^A^5GB7TLH^1G1TLH^..) if the node is the same as the node call after PC92. Any software must be able to cope with this empty slot. The remaining payload...

The remaining payload...

Bitmap/Call/:IPaddress

Ex. 1G1TLH-2:XX.XX.XX.XX

*Note: the IP address may not always be present at this time.

H99 - hopcount

Note that DXSpider converts PC16/17/19/21 from *directly* connected PC only nodes into PC92 that look like:-

PC92^GB7TLH^78042^C^7GB7DJK-1:5453^1G1TLH-1^H99^

This means that GB7TLH has PC node GB7DJK-1 with user G1TLH-1 attached to it.

The principle being used here is similar to protocols like OSPF. A node *only* sends records that either come from it or PC nodes that it is "responsible" for. Nodes do *not* splurge out their local view of the network to cause even more confusion...

'K' Record Examples - Keep Alive

GB7DJK PC92^GB7TLH^82234^K^5GB7TLH:5457:568^3^1^H99^ *Note the build number attached to software version - 5GB7TLH:5457:568

WR3D PC92^GB7TLH^82744^K^7GB7DJK-1:5453^1^1^H99^

The first for the main PC9X handling node and the second for an attached Legacy PC protocol node. The two fields at the end are <no of nodes> and <no of users> respectively. You should note the similarity of this and the traditional PC50.

PC93 - Talk/Announce/WX

PC93^<node call>^<timestamp>^<to>^<from>^<via>^<text>[^<onode>][^<IPaddr>]^H99

NOTES:

* any ^ characters in <text> MUST be converted to %5E

* <to> can be a callsign (a talk), if I can route to it, I treat it as a talk.

WX (for weather).

SYSOP (sysop announcements).

<string> which is not recognised as a callsign by pattern recognition and is taken as a CHAT group (eg: LOGGER, #9000).

* (announce/full).

* <from> is the from users callsign

* <via> constrains whether it goes via a particular node set to * for now.

* <onode> is optional and is used when passing PC93s o.b.o non-PC9x handling nodes. Not currently used.

This is an announce:

PC93^IZ7AUH-6^79200^*^IZ7AUH-6^*^IZ7AUH-6 DX CLUSTER -> dx.iz7auh.net port 8000^^XX.XX.XX.XX^H97^

This a talk:

PC93^GB7TLH^81701^WR3D-2^G1TLH-2^*^wot?^H98^

This is a Chat message:

PC93^GB7TLH^81745^#9000^G1TLH-2^*^#48 anybody on?^H96^

Note that for backwards compatibility all chat <text> sections have a serial number #1->#999 + a space added to it. The first chat message sent after restart should be a random no in that range. This is designed to defeat any de-duping on announces.

Initial Connection Process

There is no "initial handshake" as such for nodes at this level. You are not a node until you have successfully logged in as a known entity to a node. Only at that point does PC protocol happen - **if you are defined as a node with that callsign** - on the node that you are attempting to connect to. Otherwise you will be treated as a user, given the contents of any MOTD file and sent a user prompt.

So if one connects to a PC92 capable node; you have successfully logged with user (+password if required); that username is a callsign that is recognized as a node by the receiving node; you are not locked out there (the default for some random node callsign) - then this happens (GB7TLH-2 connecting to GB7DJK) as shown from the receiving node's (GB7DJK) point of view:

11:31:10 (chan) <- A GB7TLH-2 telnet

11:31:10 (chan) -> B GB7TLH-2 0

11:31:10 (chan) -> E GB7TLH-2 0

11:31:10 (chan) -> D GB7TLH-2 PC18^DXSpider Version: 1.57 Build: 569 Git: test/83fc0019[r] pc9x^5457^

11:31:10 (state) GB7TLH-2 channel func state 0 -> init

11:31:10 (chan) <- I GB7TLH-2 PC92^GB7TLH-2^41469^A^^5GB7DJK:2001,bc8,3b8c,200,,2^H99^

11:31:10 (state) GB7TLH-2 channel func state init -> init92

```
11:31:10 (*) DXPROT: Do pc9x set on GB7TLH-2

11:31:10 (chan) <- I GB7TLH-2 PC92^GB7TLH-2^41469.01^K^5GB7TLH-2:5457:536^4^1^H99^

11:31:10 (chan) <- I GB7TLH-2 PC20^

11:31:10 (*) GB7TLH-2 send_local_config: doing pc9x

11:31:10 (chan) -> D GB7TLH-2 PC92^GB7DJK^41469^A^^5GB7TLH-2:2a01,4f8,1c1b,c95c,,1^H99^

11:31:10 (chan) -> D GB7TLH-2 PC92^GB7DJK^41469.01^K^5GB7DJK:5457:569^25^65^163.172.11.79^test/83fc0019[r]^H99^

11:31:10 (chan) -> D GB7TLH-2 PC22^

11:31:10 (state) GB7TLH-2 channel func  state init92 -> normal
```

shortly followed by:

```
11:31:13 (chan) <- I GB7TLH-2 PC51^GB7DJK^GB7TLH-2^1^

11:31:13 (chan) -> D GB7TLH-2 PC51^GB7TLH-2^GB7DJK^0^
```

Actual data transfer data green = GB7DJK (receiving node) red = GB7TLH-2 (connecting node).

The stuff in black is normal logging on the receiving node (GB7DJK) - and not passed across this connection.

Connecting to an old style PC protocol node is much more verbose but the PC18 -> PC20 -> PC22 pattern is the same. It just has a big flurry of PC19/PC17 records in between these three markers. The principle is the same.

Information initially compiled from dxcluster.org, DXSpider Support list serv, and DXSpider Technical list serv by Chris, WI3W.

Retrieved from "https://wiki.dxcluster.org/w/index.php?title=DXSpider_PC_Packet_Cluster_Protocol&oldid=274"

This page was last modified on 2 April 2025, at 01:07.