

UDP Host Cache

From Gnutella Developers

Table of contents

[1 UDP Host Cache \(UHC\)](#)

[2 UDPHC ``UDP Host Cache``](#)

[3 IPP ``IP Port``](#)

[4 SCP ``Supports Cached Pongs``](#)

[5 PHC ``Packed Host Caches``](#)

[6 Usage](#)

[7 References](#)

UDP Host Cache (UHC)

What follows is an outline of how your client can make use of UDP Host Caches (UHCs). UHCs are designed to work both as a distributed bootstrapping system and a smaller GWebCache-like system, utilizing both volunteers and randomly network-elected nodes. The caches work by using Gnutella pings & pongs with additional GGEP fields.

There are four main GGEP fields that are essential for UHCs (although not all are strictly for use by UHCs).

UDPHC ``UDP Host Cache``

This GGEP extension is placed in a Pong by any client wishing to describe itself as a UHC. When a servent reads a pong that has this GGEP field, then it should remember the host as a UHC and use it for bootstrapping in future scenarios where it may need to repopulate its internal list of network hosts. A servent should NEVER contact a UHC via TCP. The UDPHC can contain an optional data value, which if it exists is interpreted to be the DNS name of this UHC. If this value exists, clients should remember the UHC via the DNS name instead of the IP contained in the pong. This allows UHCs to be reachable via DNS and thus change the ultimate IP it resolves to at any time. The port to contact the UHC is the same port as the pong's port. For example, a Pong from host 192.0.2.4, port 9999 containing the GGEP key UDPHC without any data will be remembered as a UHC at 192.0.2.4:9999. A pong from host 192.0.2.4, port 9999 containing the GGEP key UDPHC with the data of "uhc.udp-host-cache.com" will be remembered as a UHC at uhc.udp-host-cache.com:9999.

IPP ``IP Port``

This GGEP extension is placed in a Pong and contains a list of IP/Ports. The format is the same as other lists of IpPorts, the first four bytes representing the IP address and the last two representing the port. The length of the number of bytes of IPP must be divisible by 6. Note that any client can send IPP packed IpPorts in Pongs, which is particularly useful on initial entry into Gnutella. It is the Pong equivalent of "X-Try-Ultrapeers". IPP should only be sent in response to a Ping which contains the "SCP" field. It is RECOMMENDED that the IpPorts correspond to the Ultrapeer/Leaf preference contained in the data portion of the SCP field.

SCP ``Supports Cached Pongs``

This GGEP extension is placed in a Ping. Placing this in the ping indicates support for "cached pongs" -- the IPP field of Pongs. No data is required as a value for SCP, although data can be used to optimize how the ponger responds. Currently, it is possible to describe whether or not you would prefer hosts with free leaf slots of hosts with free ultrapeer slots. If data exists, the mask 0x1 will indicate which preference. If the data[0] & 0x1 == 0x1, the host prefers free ultrapeer slots. If the data[0] & 0x1 == 0x0, the host prefers free leaf slots. This leaves the other 7 bits of the first byte open for future use, and an endless supply of more bytes.

PHC ``Packed Host Caches``

This GGEP extension is placed in a Pong and contains a \n delimited list of alternate UHCs. The list is a string (and not bunch of IpPorts) so that it can transmit information about UHCs who are resolvable via DNS. It is possible to add optional key/value pairs to host caches, should the future ever need to describe properties about an individual cache. This is done by appending an ampersand and then key=value after the port. For example, a single cache can be described as "host.example.com:port&key=value&key2=value2". Multiple caches are simply appended with a \n between them. It is RECOMMENDED that this extension be compressed using normal GGEP compression.

Usage

Support for these four GGEP extensions to Pings & Pongs and remembering & utilizing UHCs are all that is needed for a client to ultimately use UHCs. The caches themselves can be built into servents (and flowed via normal ping/pong methods), or can be built separately and distribute themselves to other UHCs in an as-yet-uncreated method (probably using the PHC fields of Pongs). It is possible for clients to recommend that very long uptime users whose IP address does not change to create dyndns accounts, to be used as bootstrap hosts for brand new clients. Long uptime clients (likely leaves) can also be promoted to distributed UHCs, and spread their whereabouts around so that clients who at one point were on the network but now have a stale cached list of hosts can contact them.

Ultrapeers should add the [GGEP UP](#) and [GGEP DU](#) extensions when replying to a SCP Ping. This way the receiver can differ between ultrapeers and leaves as well as peers with long and short uptimes. The former is especially important for satisfying the preference parameter of a SCP Ping and also to recognize "full" peers. The GGEP DU ("Daily Uptime") extension is useful as peers with long uptimes can be considered more valuable.

References

[[Original GDF post](http://groups.yahoo.com/group/the_gdf/message/20840) (http://groups.yahoo.com/group/the_gdf/message/20840)]

Retrieved from "http://www.the-gdf.org/UDP_Host_Cache"

Views

- [Article](#)
- [Discussion](#)
- [View source](#)
- [History](#)

Personal tools

- [72.20.99.46](#)
- [Talk for this IP](#)