

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 19, 2020

E. Sam
March 18, 2020

Updates to the NNTP Protocol
draft-sam-nntpupdates-00

Abstract

This Internet Draft proposes and suggests updates to the Network News Transfer protocol, or NNTP. The NNTP powers Usenet, one of the largest social media platforms in the world. It is primarily used for transferring text and binary posts. The aim of these updates is to improve efficiency between NNTP peers and NNTP clients.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 19, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Internet-Draft

Updates to the NNTP Protocol

March 2020

Table of Contents

1.	Background	2
2.	Conventions and Definitions	3
3.	Updates to Article Posting and Retrieval	3
4.	Command Updates/Additions	3
4.1.	XPAT	3
4.1.1.	Usage	3
4.1.2.	Description	4
4.1.3.	Examples	4
4.2.	WHOAMI	5
4.2.1.	Usage	5
4.2.2.	Description	5
4.2.3.	Examples	6
5.	Dynamic Feeds	6
5.1.	LIST CRITERIA command	7
5.2.	MAXARTSIZE attribute	7
5.3.	GROUPWILDMAT attribute	7
5.4.	MAXGROUPS attribute	8
5.5.	DIST attribute	8
5.6.	Example	8
6.	Header Related Commands	8
6.1.	IHAVEHDR	9
6.1.1.	Usage	9
6.1.2.	Description	9
6.1.3.	Examples	10
6.2.	Changes to Certain Commands with HEADER capability	11
7.	IANA Considerations	11
8.	Security Considerations	11
9.	References	11
9.1.	Normative References	11
9.2.	Informative References	12
	Acknowledgments	12
	Author's Address	12

[1.](#) Background

Usenet is one of the largest social media networks in the world, with many endpoints and users posting both text and binary content into the network. This draft proposes various changes and improvements to keep up with the demand for Usenet.

Today, many posts are transmitted through the Usenet network and this

can cause stress on a server. Header-only feeds, which are standardized in this document, help test servers capability to handle a Usenet feed with billions of articles.

Sam

Expires September 19, 2020

[Page 2]

Internet-Draft

Updates to the NNTP Protocol

March 2020

[2.](#) Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

[3.](#) Updates to Article Posting and Retrieval

News servers store and index articles with three primary keys, a Message ID (which is globally unique and non reusable), a article number (a unique key for an article), and an arrival timestamp. More information can be found in [[RFC3977](#)]. This draft aims to make changes to some of the rules set in [[RFC3977](#)].

Article numbers MUST lie between 1 and 9,223,372,036,854,775,807, inclusive. The client and server MAY use leading zeroes in specifying article numbers but MUST NOT use more than 19 digits.

[4.](#) Command Updates/Additions

This draft proposes additional commands and changes to existing commands put forth by previous NNTP RFCs.

[4.1.](#) XPAT

[4.1.1.](#) Usage

Syntax

```
XPAT [extended-wildmat] header range|<message-id> (pat  
[pat...])|wildmat
```

Responses

221 Header follows
430 no such article
502 no permission

Parameters

pat wildmat-pattern
message-id Article message ID
extended-wildmat Use [RFC 3977](#) extended wildmat

Sam

Expires September 19, 2020

[Page 3]

Internet-Draft

Updates to the NNTP Protocol

March 2020

[4.1.2.](#) Description

The XPAT command is used to retrieve specific headers from specific articles, based on pattern matching on the contents of the header. This command was first available in INN. This command is backward compatible with the XPAT specification in [[RFC2980](#)].

In this command, a pat (pattern) is equivalent to a wildmat-pattern in [[RFC3977](#)].

The required header parameter is the name of a header line (e.g. "subject") in a news group article. See [[RFC5536](#)] for a list of valid header lines. The required range argument may be any of the following:

- o an article number
- o an article number followed by a dash to indicate all following
- o an article number followed by a dash followed another article number

The required message-id argument indicates a specific article. The range and message-id arguments are mutually exclusive.

If the optional extended-wildmat parameter was specified in the command, then the last argument MUST be ONE [[RFC3977](#)] wildmat.

At least one single wildmat-pattern (as defined by [[RFC3977](#)]) must be

specified as well if the extended-wildmat parameter is not specified. If there are additional wildmat-patterns they are joined together separated by a single space to form a collection of wildmat-pattern(s). Successful responses start with a 221 response followed by the headers from all messages in which the pattern/wildmat matched the contents of the specified header line. This includes an empty list. Once the output is complete, a period is sent on a line by itself. If the optional argument is a message-id and no such article exists, the 430 error response is returned. A 502 response will be returned if the client only has permission to transfer articles.

[4.1.3.](#) Examples

Examples of a client using the PAT command to get the Path header for 2 posts:

```
[C] GROUP alt.example
[S] 211 80 1 80 alt.example
```

Sam

Expires September 19, 2020

[Page 4]

Internet-Draft

Updates to the NNTP Protocol

March 2020

```
[C] PAT path 79-80 *example*
[S] 221 Header information for path follows (from articles)
[S] 79 patton.com!example.com!not-for-mail
[S] 80 juju.org!kubellis.com!bowdoin.com!example.com!not-for-mail
[S] .
```

Examples of a client using the PAT command with extended-wildmat to filter out all path headers with "patton.com":

```
[C] GROUP alt.example
[S] 211 80 1 80 alt.example
[C] PAT path 79-80 *,!*patton.com*
[S] 221 Header information for path follows (from articles)
[S] 80 juju.org!kubellis.com!bowdoin.com!example.com!not-for-mail
[S] .
```

[4.2.](#) WHOAMI

[4.2.1.](#) Usage

Syntax

WHOAMI

Responses

102 username server-name Logged in
103 server-name Not logged in

Parameters

username The username of the current user (specified in AUTHINFO USER)
server-name The name of the NNTP server

4.2.2. Description

Upon receiving the WHOAMI command, the server will send the response code 102 along with the username and mnemonic name of the server if the user is authenticated. If the user is not logged in, then the response code 103 must be returned with the server-name.

Severs MUST only send the 102 response code when successful authentication has been achieved through the AUTHINFO command in [\[RFC4643\]](#).

This command MUST always return the response code 102 or 103, regardless of the clients authentication status.

Sam

Expires September 19, 2020

[Page 5]

Internet-Draft

Updates to the NNTP Protocol

March 2020

The server-name can be the hostname, FQDN, or mnemonic name of the server. It can also be the name that the news server injects into its path header. Multiple news servers can have the same server-name (server operators might due this for load balancing), but news servers MUST make sure servers with the same server-name have the same article numbering for newsgroups. News clients SHOULD assume a different server-name means the article numbering is different. News clients can be assured that receiving the same server-name on the invoking of the WHOAMI command means the article numbering is the same.

This command SHOULD NOT be used to confirm if posting/reading is allowed on the server. To see if those abilities are allowed, clients should use the specific commands (POST, GROUP, ARTICLE) to see if they can post and read articles.

[4.2.3.](#) Examples

Examples of a client using the WHOAMI command while unauthenticated:

```
[C] WHOAMI
[S] 103 example.com Not logged in
```

Examples of a client using the WHOAMI command while authenticated:

```
[C] AUTHINFO USER ama
[S] 381 PASS required
[C] AUTHINFO PASS dublin
[S] 281 Authentication accepted
[C] WHOAMI
[S] 102 ama example.com Logged in
```

[5.](#) Dynamic Feeds

A news server may like to filter incoming feeds to conserve space and reduce spam on their server. The LIST CRITERIA capability will aim to accomplish this by allowing clients to request from the server what types of articles it is looking for. This is not intended to be a complete replacement for out of band feed control, but should be good enough for temporary feed adjustment. On the absence of certain attributes in the LIST CRETERIA command or the absence of the LIST CRETERIA capability altogether, news servers and clients SHOULD revert to the out of band peering info they have stored or their default behavior.

[5.1.](#) LIST CRITERIA command

When the LIST CRITERIA command is sent to a server, the server will respond with one of the following attributes. Clients must send the LIST CRITERIA command ONLY if the server advertises the LIST CRITERIA capability.

MAXARTSIZE nnn

GROUPWILDMAT wildmat
MAXGROUPS nnn
DIST string[,string]

When it is done with all of its responses, the server should terminate the sequence with a single dot (".") on a line.

[5.2.](#) MAXARTSIZE attribute

MAXARTSIZE nnn

Argument: byte amount of the maximum size of desired articles.

The MAXARTSIZE attribute specifies the maximum size of the articles the server wishes to receive (in bytes). Clients SHOULD NOT send articles bigger than the size specified in MAXARTSIZE.

[5.3.](#) GROUPWILDMAT attribute

GROUPWILDMAT wildmat

Argument: [\[RFC3977\]](#) wildmat

The GROUPWILDMAT attribute specifies the newsgroups the server wishes to receive.

Using the extended wildmat format, servers can specify the articles they want and don't want in one attribute. Clients SHOULD NOT send articles if all of the newsgroups in the article are prohibited by the wildmat. Clients SHOULD send articles if at least one of the newsgroups in the article are allowed by the wildmat.

Some examples of some wildmats that might be used in GROUPWILDMAT:

- * Send all newsgroups
- *,!alt.usenet.kooks Send all newsgroups except "alt.usenet.kooks"
- alt.example Only send articles that were posted to "alt.example"
- *bin* Only send articles that were posted to newsgroups with the phrase "bin"

[5.4.](#) MAXGROUPS attribute

MAXGROUPS nnn

Argument: A positive integer specifying the maximum number of newsgroups to which an article may be posted for the site to be willing to receive it.

The MAXGROUPS attribute allows for a sever to inform the client that it should not send articles that have been cross posted to more newsgroups than the argument in MAXGROUPS.

[5.5.](#) DIST attribute

DIST string[,string]

Argument: A group of strings containing article distributions the server does not want to receive.

The DIST attribute allows for the server to inform the client that it should not send any articles that have the following distributions (determined by the Distribution: header).

[5.6.](#) Example

This is an example of a server who specifies and returns all attributes listed above after the client sends the LIST CRITERIA command.

```
[C] LIST CRITERIA
[S] MAXARTSIZE 256000
[S] GROUPWILDMAT *,!alt.example,!alt.test
[S] MAXGROUPS 7
[S] DIST local,internal
[S] .
```

[6.](#) Header Related Commands

The massive growth of Usenet has caused some peers to engage in header-only feeds. Header-only feeds exist to help test system reliability and test systems between two peers before a full binary feed is established.

In addition, caching software and proxy servers may wish to have a copy of headers and then request the actual article from a upstream server when requested from a user. This may be accomplished from a two servers: a faster one that only has headers, and a slower one that actually has the articles request.

These commands aim to standardize commands that can be used in the transmission of header only feeds. In order to use any of the commands listed below, a server MUST advertise the HEADERS capability and the client MUST be in MODE TRANSIT. More information about adding capabilities and modes can be found in [[RFC3977](#)].

[6.1.](#) I HAVEHDR

[6.1.1.](#) Usage

In order to use this command, the server MUST advertise the HEADER capability. This command MUST NOT be pipelined.

Syntax

IHAVEHDR message-id

Responses

Initial Responses

331 Send article headers to be transferred
431 Headers not wanted
432 Transfer not possible; try again later

Subsequent Responses

232 Header transfer successful
432 Transfer failed; try again later
433 Transfer failed; do not retry

Parameters

message-id Article message ID

[6.1.2.](#) Description

The I HAVEHDR command informs the server that the client has the headers of an article with the specified message-id. If the server wants a copy of the article headers, it MUST return a 331 response code. If the server does not want a copy from the client (example - the server already has a copy of the article headers or has the full article), the server MUST return a 431 response code.

If the server does not wish to receive the headers at the moment (example - if it is in the process of receiving a full article/

headers from another client), then the server MUST send a 432 response code.

If the server indicates it wants to receive the headers by a 331 response code, then the client will then send back a copy of the headers. The end of header transmission is marked by a single dot (".") on a line by itself. The actual content of the article MUST NOT be transmitted using IHAVEHDR, instead the client can use IHAVE to transmit both the headers and content of an article.

Once the transmission of the articles have been completed, a server MUST send a 232 response code if the header transfer is successful. Unless a 232 response code is sent, clients MUST NOT assume that the transfer of the headers was successful. A lack of response SHOULD BE treated the same as a 432 response. Once received, a server MUST add the article to the newsgroups and assign the article a article number, like it would with a complete article. However, certain commands must return a different response code if only headers are available.

[6.1.3.](#) Examples

Example of a successful header transfer to another site:

```
[C] IHAVEHDR <article.with.headers@example.com>
[S] 331 Send headers; end with <CR-LF>.<CR-LF>
[C] From: "A Usenet User" <usenet@example.com>
[C] Newsgroups: alt.test
[C] Path: example.com!not-for-mail
[C] Subject: Test Article
[C] Date: 20 Jan 2020 09:30:40 -0500
[C] Organization: Example Corp
[C] Message-ID: <article.with.headers@example.com>
[C] .
[S] 232 Header transfer successful
```

Example of a unsuccessful header transfer to another site:

```
[C] IHAVEHDR <article.with.headers@example.com>
[S] 331 Send headers; end with <CR-LF>.<CR-LF>
[C] From: "A Usenet User" <usenet@example.com>
[C] Newsgroups: alt.test
```

[C] Path: example.com!not-for-mail
[C] Subject: Test Article
[C] Date: 20 Jan 2020 09:30:40 -0500
[C] Organization: Example Corp
[C] Message-ID: <article.with.headers@example.com>
[C] .
[S] 433 Header transfer unsuccessful; don't try again

Sam

Expires September 19, 2020

[Page 10]

Internet-Draft

Updates to the NNTP Protocol

March 2020

[6.2.](#) Changes to Certain Commands with HEADER capability

Advertising the HEADER capability means that certain commands will return an extra response code when dealing with a header-only message ID. There are also certain commands that must return a successful response code only if a full article is available:

- o STAT SHOULD respond with code 226 (Header only article) along with the article number and message ID in that order.
- o ARTICLE SHOULD respond with code 227 for header only articles. It should then follow the conventional format (article number, message-id, content of article). To maintain compatibility with newsreaders, a empty line in between the headers and the dot terminator (".") MUST be present. A news server can also opt, but SHOULD NOT, to place a message in the body explaining it only has the headers for that article. If the server manages to get the full article, then it can respond normally with a 220 code and the article information.
- o When a server receives a IHAVE, CHECK or TAKETHIS command, it SHOULD act like the article does not exist if the server only has the headers. This will allow the server to get the full article.
- o IHAVE MUST only be used to send full articles. For sending headers only, use IHAVEHDR.
- o The header only article MUST be assigned an article number and the headers MUST be available in header commands like PAT, OVER, HDR, etc.

[7.](#) IANA Considerations

This document has no IANA actions.

8. Security Considerations

This document has no security considerations.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

Sam	Expires September 19, 2020	[Page 11]
-----	----------------------------	-----------

Internet-Draft	Updates to the NNTP Protocol	March 2020
----------------	------------------------------	------------

- [RFC2980] Barber, S., "Common NNTP Extensions", [RFC 2980](#), DOI 10.17487/RFC2980, October 2000, <<https://www.rfc-editor.org/info/rfc2980>>.
- [RFC3977] Feather, C., "Network News Transfer Protocol (NNTP)", [RFC 3977](#), DOI 10.17487/RFC3977, October 2006, <<https://www.rfc-editor.org/info/rfc3977>>.
- [RFC4643] Vinocur, J. and K. Murchison, "Network News Transfer Protocol (NNTP) Extension for Authentication", [RFC 4643](#), DOI 10.17487/RFC4643, October 2006, <<https://www.rfc-editor.org/info/rfc4643>>.
- [RFC4644] Vinocur, J. and K. Murchison, "Network News Transfer Protocol (NNTP) Extension for Streaming Feeds", [RFC 4644](#), DOI 10.17487/RFC4644, October 2006, <<https://www.rfc-editor.org/info/rfc4644>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [RFC5536] Murchison, K., Ed., Lindsey, C., and D. Kohn, "Netnews

Article Format", [RFC 5536](https://www.rfc-editor.org/info/rfc5536), DOI 10.17487/RFC5536, November 2009, <<https://www.rfc-editor.org/info/rfc5536>>.

Acknowledgments

Thanks to everyone who helped create the tools that let us use Markdown to create Internet Drafts.

Thanks to everyone who contributed to ideas for this draft.

Thanks to everyone in the DISPATCH WG.

Author's Address

Ekow Sam

Email: winshell64@gmail.com