Internet Engineering Task Force
INTERNET DRAFT

Yaron Y. Goland
Microsoft Corporation

June 21, 1999
Expires December 1999

Multicast and Unicast UDP HTTP Messages
<draft-goland-http-udp-00.txt>

Status of this Memo

Abstract

This document provides rules for encapsulating HTTP messages in Multicast and Unicast UDP packets to be sent within a single administrative scope. No provisions are made for associating requests with responses or for guaranteeing delivery beyond rebroadcasting.

1.   Introduction

This document provides rules for encapsulating HTTP messages in Multicast and Unicast UDP packets to be sent within a single administrative scope. No provisions are made for associating requests with responses or for guaranteeing delivery beyond rebroadcasting.

This technology is motivated by applications such as SSDP where it is expected that messages which are primarily transmitted over TCP HTTP need to be transmitted over Multicast or Unicast UDP in extreme circumstances.

2. Terminology

Since this document describes a set of extensions to the HTTP/1.1 protocol, the augmented BNF used herein to describe protocol elements is exactly the same as described in section 2.1 of [RFC2616]. Since this augmented BNF uses the basic production rules provided in section 2.2 of [RFC2616], these rules apply to this document as well.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. HTTPU URL

The HTTPU URL specifies that the HTTP request is to be sent over unicast UDP according to the rules laid out in this document.

```
httpu_URL = "httpu:" "//" host [ ":" port ] [ abs_path ]
```

The BNF productions host, port and abs_path are defined in [RFC2616].

The syntax of the HTTPU URL is to be processed identically to the HTTP URL with the exception of the transport.

One MUST NOT assume that if a HTTP, HTTPU or HTTPMU URL are identical in all ways save the protocol that they necessarily point to the same resource.

4. HTTPMU URL

The HTTPMU URL specifies that the HTTP request that HTTP request is to be sent over multicast UDP according to the rules laid out in this document.

```
httpmu_URL = "httpmu:" "//" host [ ":" port ] [ abs_path ]
```

The syntax of the HTTPMU URL is to be processed identically to the HTTP URL other than the absence of abs_path will result in the request-URI of the HTTPMU request being set to "*" rather than "/".

5. Unicast UDP HTTP Messages

HTTP messages sent over unicast UDP function identically to HTTP messages sent over TCP as defined in [RFC2616] except as specified below.

All messages sent over unicast UDP MUST fit entirely in a single UDP packet. If a message can not be fit into a single UDP packet then it MUST NOT be sent using unicast UDP. Incomplete messages SHOULD be ignored.

The request-URI of a HTTP message sent over unicast UDP MUST always be fully qualified.

A single unicast UDP packet MUST only contain a single HTTP message.

Replies to unicast UDP HTTP requests are sent to the IP address and port that sent the request.

6. Multicast UDP HTTP Requests

HTTP messages sent over multicast UDP MUST obey all the requirements for HTTP requests sent over unicast UDP in addition to the requirements provided below.

Resources that support receiving multicast UDP HTTP requests MUST honor the mm and mx headers if included in the request.

When used with a multicast UDP HTTP request the "*" request-URI means "to everyone who is listening to this IP address and port."

By default httpmu requests are not responded to. This default MAY be overridden on a method-by-method basis.

[Ed. Note: This one bugs me, I suspect we will end up putting in a flag so that any intermediaries such as proxies will know what's up without having to know anything about the particular method.]

7. Retrying Requests

UDP is an inherently unreliable transport and subject to routers dropping packets without notice. Applications requiring delivery guarantees SHOULD NOT use httpu or httpmu.

In order to increase the probability that a httpu or httpmu message is delivered the message may be repeated several times.

In order to prevent the network from being flooded a message SHOULD NOT be repeated more than MAX_RETRIES time. A random period of time between MIN_RETRY_INTERVAL and MAX_RETRY_INTERVAL SHOULD be selected between each retry to determine how long to wait before issuing the retry.

8. Caching UDP HTTP Requests

Caching of httpu and httpmu request/responses is certainly possible following the normal HTTP caching rules. However there is no mechanism provided in this specification to associated requests with responses. Therefore if a client sends multiple requests to a single resource there is no generic mechanism to tell the responses apart. This restriction has not proven to be a problem for the sorts of applications that intend to use httpu and httpmu. Therefore if there

is a strong desire to provide for generic association between requests and replies through the use of request Ids are similar mechanism this feature should be added in an extension specification, as it is not necessary for many applications and thus would prove to be a needless burden.

9. Proxying UDP HTTP Requests

Just as it is possible to cache a httpu or httpmu request/response pair so it is possible to proxy such requests. The same warnings provided in section .8 apply.

10. HTTP Headers

10.1. AL Header

AL = "AL" ":" 1*("<" AbsoluteURI ">") ; AbsoluteURI is defined in section 3.2.1 of [RFC2616]

The AL header is an extension of the Location header. The contents of an AL header are ordered. If both a Location header and an AL header are included in the same request then the URI in the location header is to be treated as if it were the first entry in the AL header. The AL header MAY be used by itself but implementers should be aware that existing systems will ignore the header.

10.2. mm Request Header

```
mm = "mm" ":" Integer
Integer = First_digit *More_digits
First_digit = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
More_digits = "0" | First_digit
```

Indicates the minimum number of seconds that a multicast UDP HTTP resource MUST wait before it sends a response stimulated by a multicast request.

10.3. mx Request Header

```
mx = "mx" ":" Integer
```

Indicates the maximum number of seconds that a multicast UDP HTTP resource MUST wait before it sends a response stimulated by a multicast request.

11. Security Considerations

[Ed. Note: Besides putting in a note that all the normal HTTP security considerations apply we need to put in a discussion of the problems associated with requests getting lost as well as over sized request problem. We also need to talk about the fact that requests can get randomly lost. We also need to discuss how one uses

authentication over UDP. Specifically, that one needs to assume the challenge and send the response as part of the request.]

12. Constants

MAX_RETRIES - 3

MIN_RETRY_INTERVAL - 0 second

MAX_RETRY_INTERVAL - 10 seconds

13. References

[RFC2119] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119, March 1997.

[RFC2616] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee. Hypertext Transfer Protocol - HTTP/1.1. RFC 2616, November 1998.

14. Author's Address

Yaron Y. Goland
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Email: yarong@microsoft.com

This document will expire in September 1999.