

## Requirements for Unreliable Multicasting of Web Resources

### 1. Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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.''

To learn the current status of any Internet-Draft, please check the ``[1id-abstracts.txt](#)' listing contained in the Internet-Drafts Shadow Directories on [ds.internic.net](#) (US East Coast), [nic.nordu.net](#) (Europe), [ftp.isi.edu](#) (US West Coast), or [munnari.oz.au](#) (Pacific Rim).

The distribution of this memo is unlimited. It is filed as <[draft-aboba-unweb-00.txt](#)>, and expires October 1, 1997. Please send comments to the authors.

### 2. Abstract

This document discusses applications for unreliable multicasting of Web resources as well as requirements for an unreliable multicast protocol suitable for this use.

### 3. Introduction

#### 3.1. Purpose

Many Web sites that distribute popular, frequently updated content experience problems with "flash crowds": thousands of access per minute to the same Web page overload the Web server or the network link leading to the server. Moreover, with the increasing growth of "push" technology, it is likely that these problems will get worse, not better.

It is possible that IP multicasting of Web resources may be used both

to solve serious existing operational problems like "flash crowds", and to avoid creating new problems due to "server push" applications. Several applications appear to be naturally amenable to use of unreliable multicast.

INTERNET-DRAFT

26 March 1997

In this context, unreliable refers to applications where a repair mechanism is not required. These are typically applications where the material is of time value (stock tickers), so that it makes more sense to wait for the resource to be re-multicast than to attempt to repair it; applications in which an alternative means is available for retrieving the resource (cache filling); applications in which error correction is performed at the datalink layer; or applications in which a separate error correction stream is transmitted along with the data, typically on a separate group.

In a cache filling application, there is a relatively small probability that a particular missing resource will be hit, and so it is often more costly to request a repair than to leave an incompletely received resource in the cache, where it may never be requested. Cache hits when they do occur, will typically be spread out over time, and therefore not synchronized to the original transmission. As a result, such applications present no danger of a NAK implosion.

#### 4. Requirements

The requirements for unreliable multicast transmission of Web resources are as follows:

- Transport issues
  - Robustness
  - Prioritization
  - Low overhead
- Encapsulation issues
  - Source differentiation
  - Resource demultiplexing
  - Receiver reporting
  - Sender reporting
  - Layered encoding
  - Need for an encoded clock

## 4.1. Transport issues

### 4.1.1. Robustness

It is desirable that an unreliable Web multicast scheme be efficient in transmission of small as well as large resources.

Where transmission of large resources is required, unreliable multicasting techniques may be quite effective. For example, in a cache filling application where 80 percent of the file has been received prior to a cache miss, 80 percent of the bandwidth required to retrieve the resource would have been saved, which would greatly outweigh the cost of the connection setup for the subsequent cache miss.

However, most Web resources are in the 1 Kbyte to 10 Kbyte range, and so if the profile of resources transmitted via unreliable Web

multicasting were to more resemble the average, then an optimized encapsulation is likely to require from 2 to 20 packets. In this case unsophisticated unreliable multicast schemes are likely to be highly vulnerable to packet loss. The effect of packet loss may be mitigated by a number of techniques, including utilization of an error correction channel, as well as use of a data carousel, possibly operating with offset layering.

The use of an error correction channel is only likely to be effective for non-burst errors. Where errors are relatively uncorrelated, the effect of packet loss may be mitigated at the expense of additional overhead proportional to the loss rate. To allow receivers to subscribe to an error correction channel up to their estimated loss rate, it is necessary for the transmission rate to be sufficiently below the channel capacity so as to allow for the receiver to subscribe to the transmission as well as to the required error correction group.

Where burst errors are present, error correction channels will not be very effective and the resulting reception time will increase. If packet loss rates are substantial, then many revolutions of the data carousel will be required since on each revolution of the carousel the chance of completing the transmission will be small. Thus, after a certain number of carousel revolutions, it probably makes sense to stop listening. Thus in an unreliable multicast transmission there is

a distinct possibility that the transmission will not complete, and the application must be able to deal with this effectively.

#### [4.1.2.](#) Prioritization

In situations where "flash crowds" are anticipated, it is desirable to encourage use of unreliable Web multicasting, possibly at the expense of unicast retrievals of the same resource. As a result, it will be desirable for unreliable Web multicasts to reside on a high priority multicast port; in addition, it may be desirable for this traffic to be prioritized above unicast HTTP.

#### [4.1.3.](#) Low Overhead

Since resource multicasting will typically use a small MTU size (i.e. [536](#) octets), it is important that a low overhead encapsulation be chosen. In order to achieve this, the URL must not be sent in every packet. In addition, it may be desirable to support header compression.

### [4.2.](#) Encapsulation issues

#### [4.2.1.](#) Source differentiation

Since mixing is not useful for transmission of Web resources, and allowing multiple sources would make it difficult to maintain rate control, it is likely that only one source will be sending to a group at one time. Nevertheless, in the case where there is a handoff, it is necessary to be able to differentiate sources, since packets from the two sources may be intermingled. However, it appears that this source differentiation can be accomplished using the source IP address and port, without requiring a source ID in addition.

#### [4.2.2.](#) Resource demultiplexing

For multicast resource transmission, it is desirable to be able to reuse a group and port for the sending of multiple resources. Resources are typically of small size, and therefore the overhead of obtaining a group address and setting up the transmission would be excessive. Since it is possible for packets from one resource to become intermingled with another due to out of order delivery, it is necessary to be able to demultiplex resources sent from a single source. However, it is typically undesirable to have to include the resource URL in every packet, since packets will typically be small, and URLs may be large. Therefore a resource ID is likely to be required.

#### [4.2.3.](#) Receiver reporting

Even though we are discussing unreliable transmission, some kind of receiver feedback is likely to be desirable. Such feedback can be used to estimate listenership, packet loss rates, and receiver bandwidth availability. Jitter is not likely to be a useful metric for reception quality in this case.

Typically, receiver reporting information will be used both for engineering purposes (diagnosis of transmission problems) as well as for business purposes (listenership information). While receiver reports could be useful in allowing senders to adjust transmission parameters, typically it is more desirable to allow receiver-driven rate adaptation via layered encoding.

By transmitting a resource on several groups, each starting transmission with a different offset, the receiver may adjust their reception rate based on the available bandwidth. Typically, the group transmission rates will be tailored to commonly available bandwidths, i.e. 10 Kbps for 14.4 Kbps modems, 20 Kbps for 28.8 Kbps, 30 Kbps for single channel ISDN, etc.

Sender-driven transmission rate adjustment appears to be useful in only a limited number of circumstances. In cases where a small fraction of listeners are experiencing problems, it is undesirable to adjust the transmission rate; instead, the affected receivers should adjust their rate by leaving the higher bandwidth groups. If this does not work, they should stop listening to the transmission

altogether.

A circumstance in which sender-driven transmission rate adjustment appears useful is in the case where the majority of listeners are only subscribed to the lowest transmission rate group, yet appear to lack the bandwidth to also join an error correction group appropriate to their packet loss rate. In this case the sender should back off the transmission rate on the lowest group to allow for successful reception of the error correction information.

As there are applications in which receiver feedback may not be feasible or desirable (satellite transmission), it must be possible to turn off the receiver reporting mechanism if desired.

#### [4.2.4.](#) Sender reporting

Just as receivers may wish to provide feedback to senders, so may senders wish to provide instructions to receivers. This may include information about the particular resource being transmitted (such as the resource length, related keywords, or URL), or information on the status of the transmission itself, such as the highest offset transmitted.

#### [4.2.5.](#) Layered encoding

Layered multicasting appears to be essential for multicast transmission of resources, since it provides for receiver-driven rate adaptation, as well as for transmission of error-correction information. While layered encoding was originally proposed for use in audio/video transmission, it can also be applied to data carousel style transmissions. In this application, each of the layers transmits the same resource, beginning at a different offset. Receivers with sufficient bandwidth may then subscribe to multiple groups, and receive the resource more quickly.

Layered encoding may also provide for error correction, by allowing error correction information to be transmitted on separate groups. Receivers may then subscribe to these groups according to their average packet loss rate; receivers experiencing high loss rates will typically join a higher bandwidth error correction group. In order to allow for the additional bandwidth of an error correction group, the sender transmission rate should be set appropriately.

#### [4.2.6.](#) Need for an encoded clock

Most uses of unreliable resource multicasting do not have realtime

requirements, and as a result, it is not envisaged that unreliable multicasting of Web resources requires an encoded clock.

INTERNET-DRAFT

26 March 1997

## [5.](#) Acknowledgements

Thanks to Harald.T.Alvestrand, Jim Gemmell, Paul Leach and Thomas Pfenning for useful discussions of this problem space.

## [6.](#) References

- [1] R. Braden. "Requirements for Internet hosts - application and support." STD 3, [RFC 1123](#), IETF, October 1989.
- [2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. "RTP: A Transport Protocol for Real-Time Applications." [RFC 1889](#), GMD Fokus, January 1996.
- [3] H. Schulzrinne. "RTP Profile for Audio and Video Conferences with Minimal Control." [RFC 1890](#), GMD Fokus, January 1996.
- [4] M. F. Speer, S. McCanne. "RTP Usage with Layered Multimedia Streams." [draft-speer-avt-layered-video-01.txt](#), Sun Microsystems, LBNL, June 1996.
- [5] J. Palme, A. Hopmann. "MIME E-mail Encapsulation of Aggregate Documents, such as HTML (MHTML)." [draft-ietf-mhtml-spec-03.txt](#), Stockholm University/KTH, ResNova Software, August, 1996.
- [6] E. Levinson. "The MIME Multipart/Related Content-type." [draft-ietf-mhtml-related-00.txt](#), Xison, May, 1996.
- [7] J. Palme. "Sending HTML in E-mail, an informational supplement." [draft-ietf-mhtml-info-03.txt](#), Stockholm University/KTH, August, 1996.
- [8] R. Fielding, et al. "Hypertext Transfer Protocol - HTTP/1.1." [draft-ietf-http-v11-spec-07](#), UC Irvine, August, 1996.
- [9] J. Crowcroft, Z. Wang, A. Ghosh, C. Diot. "RMFP: A Reliable Mul-

ticast Framing Protocol." [draft-crowcroft-rmfp-00.txt](#), UCL, November, 1996.

[10] P. Parnes. "RTP Extension for Scalable Reliable Multicast." [draft-parnes-rtp-ext-srm-01.txt](#), LuTH/CDT, November, 1996.

## 7. Authors' Addresses

Bernard Aboba  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052

Phone: 206-936-6605  
EMail: [bernarda@microsoft.com](mailto:bernarda@microsoft.com)