XEP-0017: Naive Packet Framing Protocol

Mike Lin
mailto:mikelin@mit.edu
xmpp:mlin@mlin.mit.edu

Julian Missig
mailto:julian@jabber.org
xmpp:julian@jabber.org

2002-02-19
Version 0.3

Status	Type	Short Name
Rejected	Informational	

An intermediate method for more efficient framing of the Jabber XML Stream.

Legal

Contents

1 Introduction

Framing is the mechanism by which a listener is able to separate a stream of information into discrete semantic units. In Jabber, each of these semantic units is a fragment of a <stream:stream> document consisting of a direct (depth=1) child element of the <stream:stream> document element, and all its attributes and child nodes. These semantic units are hereafter called packets. A Jabber session document, excluding the document element start tag and end tag which are handled as special cases, is implicitly encoded into these packets and transmitted across the network.
This document describes a framing method that may provide performance and code simplicity advantages over the framing method currently used.

2 Framing in Jabber

The current scheme for framing in Jabber relies on the inherent structure of XML to determine packet boundaries. Since each packet is a well-formed XML fragment, the start of the packet is unambiguously denoted by the element start tag at depth=1, and the end of the packet is unambiguously denoted by the corresponding close tag at depth=1.
This method of framing is elegant because all the necessary framing information is inherent in the XML; it makes framing independent of the underlying transport layer so long as that transport layer guarantees per-session FIFO delivery. However, it has significant performance and implementation disadvantages, as it requires any Jabber node (endpoint or router) to XML-parse the packet as it is still being received in order to determine the packet boundary. Many XML parsing suites are not designed to be used in this manner, and various hacks and workarounds have emerged in current Jabber software in order to adapt them for this purpose.

3 Byte Length Framing

Consider a simple method that prefixes the byte length of each packet as an XML text node at depth=1. Put more simply, the transmitting agent calculates the byte length of the element it wishes to transmit. It then transmits this integer length encoded as text, immediately followed by the element.
This technique has the following advantages:

- The receiving agent is able to anticipate the size of the incoming packet. It can then buffer the element without having to parse it while it is only partially received, which is significantly more straightforward and more efficient to implement given the design of most current XML parsing suites. For large packets, this also allows the receiving agent to reduce potentially expensive memory realocation and copying.

- A router needs not load the entire packet payload into a DOM; it has the option of parsing only the element start tag and retransmitting the rest of the packet verbatim after checking for well-formedness.
- Since the framing data are just XML text nodes, their addition should be largely backwards compatible with current Jabber software, which should be made to ignore the extraneous depth=1 text with few, or possibly no, modifications.
- Implementation of the protocol takes very little effort. Transmitting agents generally should know the size of the packets they are about to transmit. In comparison to the work they must currently do, receiving agents are only helped by the addition of framing data; in any case, interpretation of the framing data is optional.

This technique has the following disadvantages:

- The simple insertion of framing data into the streaming XML document does not pay attention to the logical distinction between the XML document and the framing transport over which it is being transmitted.
- Error-handling semantics must be arbitrarily defined in the event that a transmitting agent misframes a packet (that is, sends the incorrect packet size).

4 Implementation Notes

Framing data is included for the <stream:stream> and </stream:stream> tags as if they were their own packets, although they are not independently well-formed XML. These should be handled as special cases in a Jabber XML streams implementation.
The connecting agent (client) implicitly requests that the receiving agent (server) use XEP-0017 framing by transmitting XEP-0017 framing data in its own outgoing stream (the connecting agent always ”goes first”). Servers should detect the presence of framing data in the client’s stream (by testing whether the first character received is a digit or a <) and, if it is detected, activate outgoing framing for that session.
Regardless of the use of framing data, all nodes must verify the well-formedness of XML payloads in order to avoid propagating misframes.

5 Example

Listing 1: A Trivial And Nearly Meaningless Example

```
94<message to=”whomever@mlin.mit.edu” from=”mlin@mlin.mit.edu”>some
   opaque information</message>
```

6 Philosophical Notes and Conclusion

This framing method is not intended as a fully satisfactory or permanent solution to XML stream framing. In the ”distant” (longer than one year) time span, it may be desirable to consider more thorough systems such as BEEP. The intent of this document is to establish an intermediate solution that will provide code simplicity advantages to new implementations in the near term without requiring fundamental changes to the Jabber transport or protocol (as adopting BEEP would almost certainly require).