Revision of the Binary Floor Control Protocol (BFCP) for use over an unreliable transport (draft-sandbakken-dispatch-bfcp-udp-03)

Charles Eckel, Tom Kristensen, Mark Thompson, Geir Arne Sandbakken, Eoin McLeod

IETF 82 BFCPBIS WG Meeting
November 16, 2011

Motivation

- Existing deployments of SIP based videoconferencing typically:
  - Consist of RTP media streams for audio and video
  - Use ICE and/or other methods for NAT/firewall traversal
  - Found in enterprise networks
- When enhancing with support for content sharing, the BFCP connection often poses a problem
  - There may be a strong preference for UDP based signaling in general
  - Establishment/traversal of the TCP connection involving ephemeral ports, as is typically the case with BFCP over TCP, can be problematic
- This draft defines UDP as an alternate transport for BFCP, leveraging the mechanisms in place for RTP over UDP media streams for the BFCP communication

Approach

- Minor changes to transaction model
  - All requests now have a response to complete transaction
    - Added an explicit "Ack" primitive for each case in which none existed
  - Retransmission timer to ensure reliability
  - Transaction Initiator flag to indicate a primitive is a response to a previous request
  - One pending transaction per entity (ordering, congestion control)

Approach (cont)

- Goodbye/GoodbyeAck dissociate (TCP/BFCP close)
- New ERROR-CODEs for following cases:
  - Unable to parse message
  - Use DTLS
- DTLS MUST be supported
- ICE/STUN if applicable and needed

Open Items

1. DTLS connection establishment
   DTLS connection establishment:
   Request Specific ACK vs. Generic Ack
2. Request Specific ACK vs. Generic Ack
3. Large Message Considerations

DTLS Connection Establishment

Which party, the client or the floor control server, acts as the TLS/DTLS server depends on how the underlying TCP/DTLS connection is established. For example, when the TCP/DTLS connection is established using an SDP offer/answer exchange

DTLS Connection Establishment: Options

4583 (as currently defined)

2. The BFCP server always acts as the TLS/DTLS server
3. The offerer always offers setup:actpass and the answerer answers either setup:active or setup:passive, where

posted to bfcpbis mailer http://www.ietf.org/mail-archive/

Preferred option: (3)

- Adheres to RFC 5763, does not overload offer/answer semantics, works for offerless INVITE with B2BUAs

semantics, works for offerless INVITE with B2BUAs

Request Specific ACK vs. Generic ACK

GREEN ITALICS

- *[FloorRequestStatusAck]*

  FloorRelease / FloorRequestStatus / *[FloorRequestStatusAck]*
- FloorRequestStatus / *FloorRequestStatusAck*
- FloorRequestQuery / FloorRequestStatus / *[FloorRequestStatusAck]*
- UserQuery / UserStatus
- FloorQuery / FloorStatus / *[FloorStatusAck]*
- FloorStatus / *FloorStatusAck*
- ChairAction / ChairActionAck
- Hello / HelloAck
- Error /
- *Goodbye / GoodbyeAck*

Request Specific ACK vs. Generic ACK:

1. Always send request specific ACK (as currently defined)
Always send request specific ACK (as currently defined)

2. Send request specific ACK only if transaction initiator flag indicates message is initiating a new transaction

3. Send a generic ACK at the transport level for every message

(3) simplifies the existing transaction model as well as the adding of future BFCP primitives, but more chatty protocol

Large Message Considerations

COMMON-HEADER.

When using UDP, there is the added concern that a single BFCP message can be fragmented at the IP layer if its overall size exceeds the MTU threshold of the network.

The target use cases for BFCP via UDP typically involve relatively small BFCP messages ... BFCP entities SHOULD ensure that their messages are smaller than the recommended MTU size of 1300 bytes when encoded to minimize the likelihood of fragmentation in route to their peer entity.

1. Leave as out of scope (as currently defined)
   - additive messages
2. The mechanism defined for RELOAD in section 5.7 of [I-D.ietf-p2psip-base] has been identified as a good candidate.
3. Add an applicability statement on those BFCP messages and/or attributes deemed as inappropriate for use over transports where fragmentation is a concern
4. Define SIP event package to deliver information
   - Preferred option: (1)

BFCP/STUN Demultiplexing

> UDP entities are RECOMMENDED to use STUN [RFC5389] for keep-alives, as described for SIP [RFC5626].
> [RFC5389] for keep-alives, as described for SIP [RFC5626].

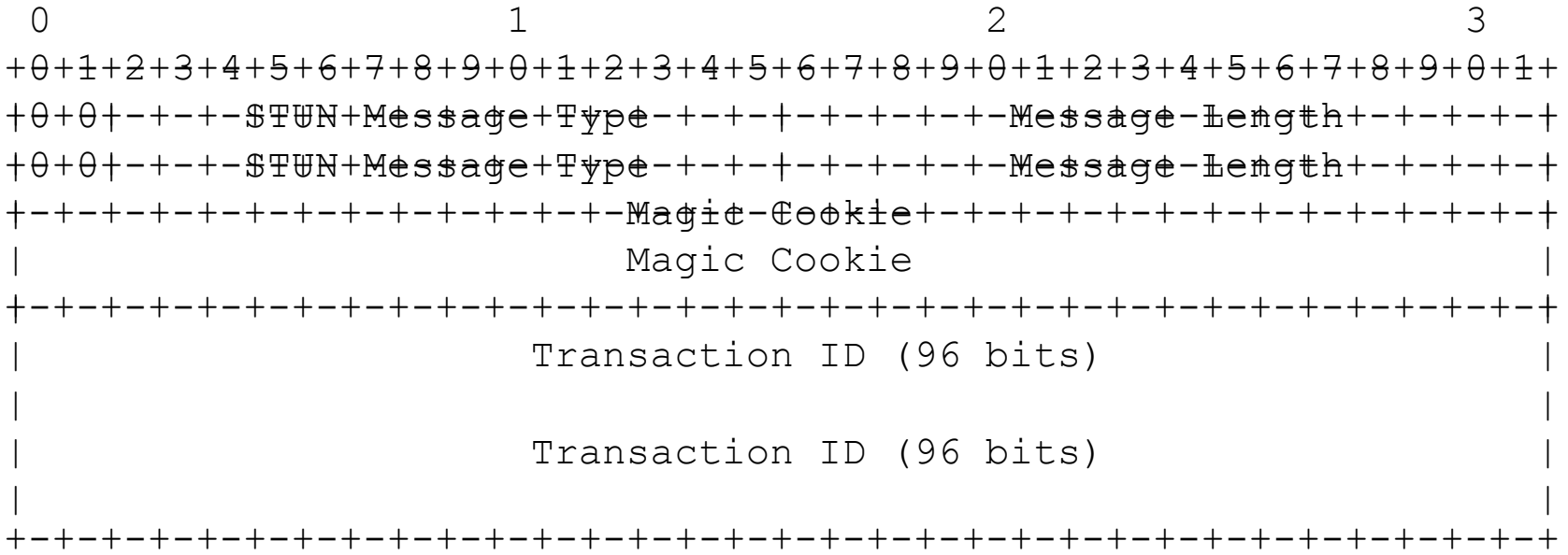
Consequently, implementations need to be able to demultiplex

Consequently, implementations need to be able to demultiplex incoming BFCP/STUN packets

BFCP/STUN Demultiplexing: STUN

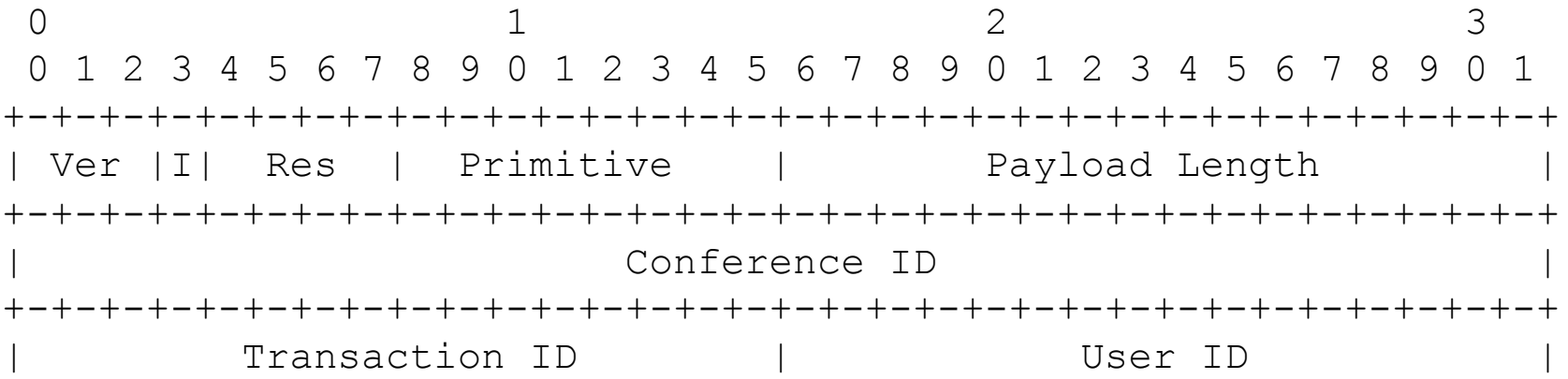
Format for STUN messages:

Format for STUN messages:



The most significant 2 bits of every STUN message MUST be zeroes. This can be used to differentiate STUN packets from

BFCP/STUN Demultiplexing: BFCP



The "I" field is added by draft-sandbakken-dispatch-bfcp-udp.

But per RFC 4582 as well as draft-sandbakken-dispatch-bfcp-udp:

Ver: The 3-bit version field MUST be set to one (i.e. 001) to indicate this version of BFCP.

Therefore, as with STUN, the first two bits are always zeroes.

BFCP/STUN Demultiplexing: Options

1. Leave as currently defined

   A reasonable STUN (RFC 5389) implementation will also check the magic cookie (0x2112A442) and check if the message length is sane (i.e. STUN messages are padded to a multiple of 4 bytes, the last 2 bits of this field is always zero)

2. Change the version number for BFPC [via UDP] to a value where one of the first two bits is one

Preferred option: (1)

future for other reasons, STUN demultiplexing is not viewed as a sufficient justification for such a change.

THANK YOU

Arne Sandbakken, Eoin McLeod

November 16, 2011