

Revision of the Binary Floor Control Protocol (BFCP) for use over an unreliable transport

draft-ietf-bfcpbis-rfc4582bis-01

Tom Kristensen, Charles Eckel, Alfred Heggstad, Geir Arne Sandbakken

IETF BFCPbis virtual interim February 21th 2012

Use Case - SIP Videoconferencing

- BFCP used for floor control of content sharing video stream
- BFCP server and participant roles negotiated via SDP offer/answer exchange
- Responsibility for opening the BFCP channel negotiated via SDP offer/answer exchange
- A “normal user endpoint” might need to act as a BFCP server and/or initiate the connection for the BFCP channel
 - Point-to-point/Peer-to-Peer/Business-to-business
 - Point-to-multipoint (potential internal MCU)
- One/Both BFCP entities behind NATs/firewalls

Motivation

- Existing deployments of SIP 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, 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 channel

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 distinguish request vs. response
 - One pending request per entity (ordering, congestion control)
- Goodbye/GoodbyeAck dissociate (TCP/BFCP close)
- New ERROR-CODEs for following cases:
 - Unable to parse message
 - Use DTLS
- Fragmentation handling scheme
- DTLS MUST be supported
- ICE/STUN if applicable and needed
- Increment version number when using these extensions over unreliable transport

Request/Response

RFC 4582 primitives in **blue**, new primitives in *red italics*

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

Version RFC 4582 vs. extensions

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Ver  | I|F| Res | Primitive | Payload Length |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

- Aim is to clearly identify new extensions from existing
- Set to 1 when using BFCP over reliable transport, i.e. as in RFC 4582
- Set to 2 when using BFCP over unreliable transport, with the extensions specified in this document

Transaction Initiator (*the problem*)

- Some primitives can be either a request or a response

Request	Response
FloorRequest FloorRequestStatus	FloorRequestStatus FloorRequestStatusAck
FloorQuery FloorStatus	FloorStatus FloorStatusAck

- Distinguishing between the two requires some state information beyond that desirable at the stack level

Transaction Initiator (*the solution*)

```

+++++
| Ver | I | F | Res | Primitive | Payload Length |
+++++
| Conference ID |
+++++
| Transaction ID | User ID |
+++++

```

- Has relevance only for BFCP over unreliable transport
 - When cleared; it indicates that this message is a request initiating a new transaction, and the Transaction ID that follows has been generated for this transaction.
 - When set; it indicates that this message is a response to a previous request, and the Transaction ID that follows is the one associated with that request.
- When BFCP is used over reliable transports, the flag has no significance and SHOULD be cleared.

Fragmentation Handling (*header fields*)

Ver	I	F	Res	Primitive	Payload Length
Transaction ID			User ID		
Fragment Offset			Fragment Length		

- Has relevance only for BFCP over unreliable transport
 - When cleared; the message is not fragmented.
 - When set; it indicates that the message is a fragment of a large fragmented BFCP message.
- The fields *Fragment Offset* and *Fragment Length* are present only if *F* is set.
- When BFCP is used over reliable transports, the flag has no significance and SHOULD be cleared.

Fragmentation Handling (*mechanism*)

- For BFCP messages with size greater than MTU; fragment into N contiguous data ranges
- The sender creates N BFCP fragment messages
 - *F* flag set
 - Same *Transaction ID*
 - *Fragment Offset* denotes number of bytes in the previous fragments
 - *Fragment Length* contains length of the fragment itself
 - (As before, *Payload Length* contains the length of the entire, unfragmented message)
- The receiver buffers the fragments until entire BFCP message is received
 - Deliver assembled message to BFCP state machine
 - Can discard an incomplete buffer, starting the *Response Retransmission Timer* after the first fragment

Fragmentation Handling (*message loss*)

- If a fragment is lost, the sender will not receive an “Ack”
 - Sender will retransmit with same *Transaction ID*
- If the “Ack” sent by the receiver is lost
 - Then the entire message will be resent by the sender
 - The receiver must then retransmit the “Ack”

Security

- Mandate support for DTLS for transport over UDP
 - “[...] *MUST support TLS for transport over TCP and MUST support DTLS for transport over UDP*”
 - Utilizing the setup attribute defined in RFC5763 in SDP offer/answer exchange to determine DTLS server or client role
- TLS \Rightarrow TLS/DTLS for mutual authentication and security considerations
- Adopt a strategy similar to DTLS-SRTP (RFC 5763 and RFC 5764) for negotiation

NAT Traversal

- Leverage existing NAT traversal infrastructure and strategies deployed
- Typically includes some subset of ICE (RFC 5245)
- Recommend STUN (RFC 5389) for keep-alives
 - As described for SIP (RFC 5626)
 - Facilitate establishment and maintenance of NAT bindings
- Recommend symmetric ports for sending and receiving BFCP packets
 - As recommended for RTP/RTCP (RFC 4961)

Remaining Work

- Example signalling flows
- More popping up today?
- Reformat and merge to proper RFC4582bis (BFCPbis WG) and RFC4583bis (MMUSIC WG)
- Fix errata and other known issues (not related to unreliable transport)