Revision of the Binary Floor Control Protocol (BFCP) for use over an unreliable transport
draft-sandbakken-xcon-bfcp-udp-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 26, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This memo extends the Binary Floor Control Protocol (BFCP) for use over an unreliable transport. It details a set of revisions to the protocol definition document and the specification of BFCP streams in the Session Description Protocol (SDP).

Table of Contents

1. Introduction .................................................. 4
2. Revision of RFC4582 ..................................... 4
   2.1. Overview of Operation (4) ......................... 4
   2.2. Floor Participant to Floor Control Server Interface (4.1) ...................... 5
   2.3. COMMON-HEADER Format (5.1) ......................... 5
   2.4. Hello (5.3.11) ........................................ 5
   2.5. FloorRequestStatusAck (5.3.14) ................... 6
   2.6. ErrorAck (5.3.15) .................................. 6
   2.7. FloorStatusAck (5.3.16) .......................... 6
   2.8. Transport (6) ........................................ 6
     2.8.1. Unreliable transport (6.2) .................... 7
   2.9. Lower-Layer Security (7) .......................... 7
   2.10. Protocol Transactions (8) ......................... 7
     2.10.1. Server Behavior (8.2) ....................... 7
     2.10.2. Timers (8.3) .................................. 8
   2.11. Receiving a response [to a FloorRequest Message] (10.1.2) ...................... 8
   2.12. Receiving a response [to a FloorRelease Message] (10.2.2) ..................... 8
   2.13. Receiving a response [to a ChairAction Message] (11.2) ...................... 9
   2.14. Receiving a response [to a FloorQuery Message] (12.1.2) .................... 9
   2.15. Receiving a response [to a FloorRequestQuery Message] (12.2.2) ............ 9
   2.16. Receiving a response [to a UserQuery Message] (12.3.2) ..................... 9
   2.17. Receiving a response [to a Hello Message] (12.4.2) ...................... 9
   2.18. Reception of a FloorRequestStatus Message (13.1.3) ...................... 10
   2.19. Reception of a FloorStatus Message (13.5.3) ...................... 10
   2.20. Reception of an Error Message (13.8.1) ...................... 10
   2.21. Security Considerations (14) .................... 10
   2.22. IANA Considerations - Primitive Subregistry (15.2) ...................... 10
3. Revision of RFC4583 ..................................... 11
   3.1. Fields in the ’m’ Line (3) ......................... 11
   3.2. Security Considerations (10) ...................... 11
   3.3. Registration of SDP ’proto’ values (11.1) ...................... 11
4. Future work .................................................. 11
5. Acknowledgements ........................................... 13
6. Normative References ...................................... 13
1. Introduction

The motivation for using unreliable transports for BFCP [RFC4582] messages is fuelled by network deployments where RTP proxies are present for NAT and firewall traversal. In these deployments, TCP may neither be applicable nor appropriate, for example, due to lack of support for TCP media relay or ICE-TCP [I-D.ietf-mmusic-ice-tcp].

This memo extends the BFCP protocol to support unreliable transport. Minor changes to the transaction model are introduced in that all requests now have an appropriate response to complete the transaction. The requests are sent with a retransmit timer associated with the response to achieve reliability. The draft also defines a mechanism to ensure bi-directional opening of pin-holes with appropriate keep-alive mechanism needed to keep those bindings available.

The intension is not to change the semantics of BFCP, but to present a trivial and workable extension that permits UDP as a transport. Existing implementations in the spirit of the approach detailed in -00 of this draft have demonstrated the approach to be feasible. The purpose of this document is to formalise the deviations from the baseline specification enabling interoperability between implementations.

The content of this draft relates to the BFCP protocol specification [RFC4582] and the format for the specification of BFCP streams in the SDP [RFC4583]. This memo is written with the goal of being incorporated into an upcoming revision of those documents without requiring additional protocol and stream specification documents.

This draft is not recommended for adoption as an XCON working group item at this time owing to the outstanding work detailed in Section 4, but is submitted for information and discussion within the XCON community.

2. Revision of RFC4582

This section details revisions to [RFC4582], the base protocol specification of BFCP. The section number to which updates apply are indicated in parentheses in the titles of the sub-sections below.

2.1. Overview of Operation (4)

Fourth paragraph, final sentence change:
Over reliable transports, server-initiated transactions consist of a single message, whose Transaction ID is 0, from the floor control server to a client.

Over unreliable transports, the Transaction ID must be non-zero and unique in the context of outstanding Transaction ID numbers used toward that particular client from the floor control server. Clients shall respond with the appropriate transaction-closing response message with the same Transaction ID.

2.2. Floor Participant to Floor Control Server Interface (4.1)

Ninth paragraph (page 11), replace "ID is 0" in first clause with:

... ID is 0 when communicating over reliable transports, non-zero and unique in the context of outstanding floor control server to client transactions otherwise.

Caption of Figure 3 on page 13 should be updated to reflect that this call flow is over reliable transport, hence the Transaction ID of 0.

2.3. COMMON-HEADER Format (5.1)

The values below should be appended to the end of Table 1: BFCP primitives:

| Value | Primitive | Direction |
|-------+-----------+-----------|
| 14    | FloorRequestStatusAck | P -> S ; Ch -> S |
| 15    | ErrorAck | P -> S ; Ch -> S |
| 16    | FloorStatusAck | P -> S ; Ch -> S |

Table 1: BFCP primitives

Further, the description of Transaction ID should have the final clause deleted with the reference to Section 8 remaining. The value used for server-initiated transactions shall be non-zero when BFCP is used over unreliable transports, and this qualification shall be described in the updated Section 8.

2.4. Hello (5.3.11)

This sentence below should be inserted between the two sentences that open this subsection:
Over unreliable transports, the floor control servers shall also use the Hello message to check the liveness and open any NAT pinholes as a direct consequence on the path between themselves and the floor participants and floor chairs (See Section 2.8.1).

2.5. FloorRequestStatusAck (5.3.14)

This new subsection should be added to specify the normative ABNF for the new primitive, FloorRequestStatusAck.

FloorRequestStatusAck = (COMMON-HEADER) *[EXTENSION-ATTRIBUTE]

Figure 1: FloorRequestStatusAck format

2.6. ErrorAck (5.3.15)

This new subsection should be added to specify the normative ABNF for the new primitive, ErrorAck.

ErrorAck = (COMMON-HEADER) *[EXTENSION-ATTRIBUTE]

Figure 2: ErrorAck format

2.7. FloorStatusAck (5.3.16)

This new subsection should be added to specify the normative ABNF for the new primitive, FloorStatusAck.

FloorStatusAck = (COMMON-HEADER) *[EXTENSION-ATTRIBUTE]

Figure 3: FloorStatusAck format

2.8. Transport (6)

The existing text should be demoted to become subsection 6.1 re-titled appropriately with suitable text added as introduction to section 6 (to follow in a later revision of this draft).

The discussion around identification of flows remains valid over both
transport types and there is no proposal to change the failure behavior of TCP connections over which errors have been received.

A new subsection 6.2 should be added with details of the use of BFCP over unreliable transport, thus:

2.8.1. Unreliable transport (6.2)

TBD. This section shall detail characteristics and consequences of transmission of BFCP messages using UDP flows, including use of Transaction ID to track request/response pairs, how to handle ICMP port unreachable failures mid-flow, and further issues as needed.

This section should also detail how NAT pinholes can be opened to cater for scenarios where either or both participants are behind restrictive NATs and therefore not port-reachable. It is proposed that the Hello message be re-appropriated for use by the floor control server as well as the existing use by floor participants and floor chairs. An initial Hello should be sent by all participants using the Timer A as defined in Section 2.10.2. After a HelloAck has been received by all participants, the client must continue to send Hello every 30 seconds for keep alive (Timer B). The floor control server should not send further Hello messages after the initial exchange.

2.9. Lower-Layer Security (7)

For review in future revisions of this draft, per Section 4.

2.10. Protocol Transactions (8)

The final clause of the introduction to section 8 shall be changed to read:

Since they do not trigger any response, their Transaction ID is set to 0 when used over reliable transports, but must be non-zero and unique in the context of outstanding transactions over unreliable transports.

When using BFCP over unreliable transports, all requests will use retransmit timer A (see Section 2.10.2) until the transaction is completed.

2.10.1. Server Behavior (8.2)

The final clause of this section shall be changed to read:
Server-initiated transactions MUST contain a Transaction ID equal to 0 when BFCP is used over reliable transports. Over unreliable transport, the Transaction ID shall have the same properties as for client-initiated transactions: the server MUST set the Transaction ID value in the common header to a number that is different from 0 and that MUST NOT be reused in another message from the server until the appropriate response from the client is received for the transaction. The server uses the Transaction ID value to match this message with the response from the floor participant or floor chair.

2.10.2. Timers (8.3)

When an unreliable transport is used, Timer A shall be used as a retransmit timer by which a request is retransmitted until an appropriate response is received or until the maximum number of retransmissions have occurred. Timer A doubles on each retransmit, starting at 500ms and failing after three unsuccessful retransmission attempts. That is, intervals of 500ms, 1s and 2s, with a total transaction window of 7.5s (The cumulation of retry periods plus double the value of Timer A after the last permitted retransmission).

If a valid response is not received for a client or server initiated transaction, the implementation MUST consider the BFCP stream as failed and follow the connection reestablishment as described in section 6 (e.g. initiate a new offer/answer [RFC3264] exchange).

2.11. Receiving a response [to a FloorRequest Message] (10.1.2)

Prepend the sentence below at the start of this subsection:

When communicating over unreliable transport and upon receiving a FloorRequest from a participant, the floor control server MUST respond with a FloorStatus message within the transaction failure window to complete the transaction.

2.12. Receiving a response [to a FloorRelease Message] (10.2.2)

Prepend the sentence below at the start of this subsection:

When communicating over unreliable transport and upon receiving a FloorRelease from a participant, the floor control server MUST respond with a FloorStatus message within the transaction failure window to complete the transaction.
2.13. Receiving a response [to a ChairAction Message] (11.2)

Prepend the sentence below at the start of this subsection:

When communicating over unreliable transport and upon receiving a ChairAction from a participant, the floor control server MUST respond with a ChairActionAck message within the transaction failure window to complete the transaction.

2.14. Receiving a response [to a FloorQuery Message] (12.1.2)

Prepend the sentence below at the start of this subsection:

When communicating over unreliable transport and upon receiving a FloorQuery from a participant, the floor control server MUST respond with a FloorStatus message within the transaction failure window to complete the transaction.

2.15. Receiving a response [to a FloorRequestQuery Message] (12.2.2)

Prepend the sentence below at the start of this subsection:

When communicating over unreliable transport and upon receiving a FloorRequestQuery from a participant, the floor control server MUST respond with a FloorRequestStatus message within the transaction failure window to complete the transaction.

2.16. Receiving a response [to a UserQuery Message] (12.3.2)

Prepend the sentence below at the start of this subsection:

When communicating over unreliable transport and upon receiving a UserQuery from a participant, the floor control server MUST respond with a UserStatus message within the transaction failure window to complete the transaction.

2.17. Receiving a response [to a Hello Message] (12.4.2)

Prepend the sentence below at the start of this subsection:

When communicating over unreliable transport and upon receiving a Hello from a participant or floor control server (in the case of initial exchanges), the recipient MUST respond with a HelloAck message within the transaction failure window to complete the transaction.
2.18. Reception of a FloorRequestStatus Message (13.1.3)

The sentence below shall appear as a new subsection:

When communicating over unreliable transport and upon receiving a FloorRequestStatus message from a floor control server, the participant MUST respond with a FloorRequestStatusAck message within the transaction failure window to complete the transaction.

2.19. Reception of a FloorStatus Message (13.5.3)

The sentence below shall appear as a new subsection:

When communicating over unreliable transport and upon receiving a FloorStatus message from a floor control server, the participant MUST respond with a FloorStatusAck message within the transaction failure window to complete the transaction.

2.20. Reception of an Error Message (13.8.1)

The sentence below shall appear as a new subsection:

When communicating over unreliable transport and upon receiving an Error message from a floor control server, the participant MUST respond with a ErrorAck message within the transaction failure window to complete the transaction.

2.21. Security Considerations (14)

TBD. It is a requirement that the extension of BFCP for unreliable transports shall not introduce any new threats.

2.22. IANA Considerations - Primitive Subregistry (15.2)

This section instructs the IANA to register the following new values for the BFCP primitive subregistry.

<table>
<thead>
<tr>
<th>Value</th>
<th>Primitive</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>FloorRequestStatusAck</td>
<td>RFC[XXXX]</td>
</tr>
<tr>
<td>15</td>
<td>ErrorAck</td>
<td>RFC[XXXX]</td>
</tr>
<tr>
<td>16</td>
<td>FloorStatusAck</td>
<td>RFC[XXXX]</td>
</tr>
</tbody>
</table>

Table 2: BFCP primitive subregistry
3. Revision of RFC4583

This section details revisions to [RFC4583], the format for specifying BFCP streams. The section number to which updates apply are indicated in parentheses in the titles of the sub-sections below.

3.1. Fields in the ‘m’ Line (3)

The section shall be re-written to remove reference to the exclusivity of TCP as a transport for BFCP streams.

1. In paragraph four, "... will initiate its TCP connection ..." becomes "... will direct BFCP messages ..."

2. In paragraph four, delete "Since BFCP only runs on top of TCP, the port is always a TCP port."

3. In paragraph five, we now define three new values for the transport field, adding "UDP/BFCP" as the third symbol, changing "former" for "first", "latter" for "second", and adding a final clause defining the use of UDP/BFCP as being for when BFCP runs on top of UDP

3.2. Security Considerations (10)

At this time, see Section 4.

3.3. Registration of SDP ‘proto’ values (11.1)

This section should be renamed now that there are more values to register in the SDP parameters registry, with the following added to the table:

<table>
<thead>
<tr>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP/BFCP</td>
<td>RFC[XXXX]</td>
</tr>
</tbody>
</table>

Table 3: Value for the SDP ‘proto’ field

4. Future work

This draft reflects a work in progress, with at least the following items to be documented and/or revised before soliciting adoption by the XCON working group:
Secured transport  This revision of the draft has not addressed the security of the transport of the BFCP stream. It is likely that the recommendation will be to specify UDP/DTLS/BFCP with the same cipher set as specified in [RFC4583]. However, no such recommendation shall be made until further study is carried out regarding the impact on such an approach where participant reachability may be hindered (e.g. opening and maintenance of NAT pin-holes and the directionality of the BFCP stream).

Protocol revision  Certain aspects of this draft require different behaviors depending on whether a reliable or unreliable transport is being used, e.g. server-initiated transactions having Transaction ID 0 over reliable without response versus non-zero and active-unique with a confirmation message over unreliable transports. This inconsistency may be unsuitable for implementors. We may look to keeping the behavior consistent across both types of transport, which may in turn require a change of version field in the COMMON-HEADER (Section 5.1 of [RFC4582]) such that implementations can discern which behavior to use.

Fragmentation  It has been observed that BFCP message structures can grow to be sufficiently large that they exceed the typical MTU threshold for local area networks (assumed here as 1500 octets). For example, a FloorStatus message with multiple FLOOR-REQUEST-INFORMATION attributes that contain detailed STATUS-INFO in the OVERALL-REQUEST-STATUS and FLOOR-REQUEST-STATUS attributes. A strategy for coping with such fragmented messages is required. This may be as simple as an applicability statement on those BFCP messages and/or attributes deemed as inappropriate for use over transports where fragmentation is a concern, or it may require further protocol specification to eradicate fragmentation as an issue.

UDP encapsulation of TCP  An alternative suggestion for conveying BFCP streams in such network deployments as has previously motivated this work has been to investigate encapsulating the TCP stream in a UDP flow [I-D.denis-udp-transport].

Example signaling flows  Future versions of this draft should include at least one example of a signaling exchange over unreliable transport showing updated transactions and message retransmission as a visual aid and reference for implementors.
5. Acknowledgements

The team working on this draft are: Trond G. Andersen, Tom Kristensen, Eoin McLeod, Geir A. Sandbakken and Mark K. Thompson at TANDBERG; Alfred E. Heggestad at Telio Telecom.

6. Normative References

[I-D.denis-udp-transport]

[I-D.ietf-mmusic-ice-tcp]


Editorial Comments

[c_Version] MKT: The version field may need revising in this table, also, depending on the outcome of future work.
Authors’ Addresses

Mark K. Thompson (editor)
TANDBERG
Philip Pedersens vei 22
N-1366 Lysaker
Norway

Phone: +44-118-934-8711
Email: mark.thompson@tandberg.com
URI: http://www.tandberg.com/

Tom Kristensen
TANDBERG

Email: tom.kristensen@tandberg.com, tomkri@ifi.uio.no

Geir A. Sandbakken
TANDBERG

Email: geir.sandbakken@tandberg.com

Trond G. Andersen
TANDBERG

Email: trond.andersen@tandberg.com