

Revision of the Binary Floor Control
Protocol (BFCP)
for use over an unreliable transport
(draft-sandbakken-dispatch-bfcp-udp-02)

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

IETF-81, Quebec City, July 25-29, 2011

Changes from Previous Version

- ▶ Switched from standard track to informational
- ▶ Added section on motivation, including alternatives considered
- ▶ Changed semantics of the Transaction Initiator (I) flag-bit
- ▶ Expanded transport section with considerations for:
 - ▶ congestion control
 - ▶ ICMP errors
 - ▶ large messages/fragmentation
- ▶ Updated security related sections
 - ▶ DTLS when using UDP
- ▶ Added section on NAT Traversal



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 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 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
- ▶ **DTLS MUST be supported**
- ▶ **ICE/STUN if applicable and needed**



Request/Response

RFC 4852 primitives in **BLUE**, new primitives in **GREEN ITALICS**

- ▶ FloorRequest / FloorRequestStatus
- ▶ FloorRelease / FloorRequestStatus
- ▶ FloorRequestStatus / *FloorRequestStatusAck*
- ▶ FloorRequestQuery / FloorRequestStatus
- ▶ UserQuery / UserStatus
- ▶ FloorQuery / FloorStatus
- ▶ FloorStatus / *FloorStatusAck*
- ▶ ChairAction / ChairActionAck
- ▶ Hello / HelloAck
- ▶ Error / *ErrorAck*
- ▶ *Goodbye / GoodbyeAck*
- ▶ UserStatus / *UserStatusAck* [QUESTION: DO WE NEED TO ADD THIS]



Transaction Initiator (the problem)

- ▶ Some primitives can be either a request or a response

FloorRequest / *FloorRequestStatus*

but then

FloorRequestStatus / FloorRequestStatusAck

FloorQuery / *FloorStatus*

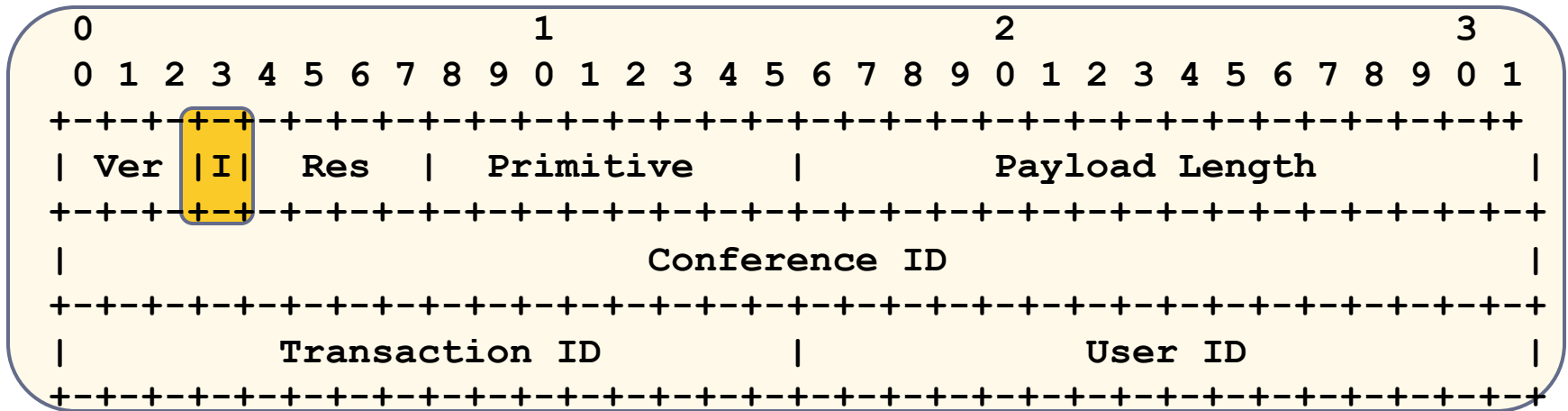
but then

FloorStatus / FloorStatusAck

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



Transaction Initiator (the solution)



Has relevance only for use of BFCP over unreliable transport.

When clear, 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.

Security

- ▶ **Mandate support for DTLS for transport over UDP:**
 - ▶ *BFCP floor control servers and clients (which include both floor participants and floor chairs) **MUST support TLS for transport over TCP and MUST support DTLS for transport over UDP [RFC5246].***
 - ▶ *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 [RFC4583], the answerer (which may be the client or the floor control server) always acts as the TLS/DTLS server.*
- ▶ **TLS → TLS/DTLS for mutual authentication and security considerations**
- ▶ **Adopt strategy similar to DTLS-SRTP specifications (RFC5763 and RFC5764) for negotiation**



NAT Traversal

- ▶ Leverage existing NAT traversal infrastructure and strategies deployed to facilitate transport of the media associated with video conferencing sessions
- ▶ Typically includes some subset of ICE [RFC5245]
- ▶ RECOMMEND STUN [RFC5389] for keep-alives
 - ▶ As described for SIP [RFC5626]
 - ▶ Facilitate establishment and maintenance of NAT bindings
- ▶ RECOMMEND symmetric ports for sending and receiving BFCP packets
 - ▶ As recommended for RTP/RTCP [RFC4961]



Future Work

- ▶ **Adapting DTLS usage to BFCP**
 - ▶ DTLS-SRTP specifications RFC 5763 and RFC 5764 referenced as example, need to add details
- ▶ **Example signaling exchanges over unreliable transport**
 - ▶ updated transactions
 - ▶ message retransmission
 - ▶ DTLS during call setup
- ▶ **IANA Considerations**
 - ▶ Now that draft is informational, it can no longer add attributes to the registries defined for BFCP per the requirements stated in RFC 4582
 - ▶ Sections related to IANA considerations will need to be reworked

Thank You

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

IETF-81, Quebec City, July 25-29, 2011