

Binary Floor Control Protocol (BFCP)

Gonzalo.Camarillo@ericsson.com



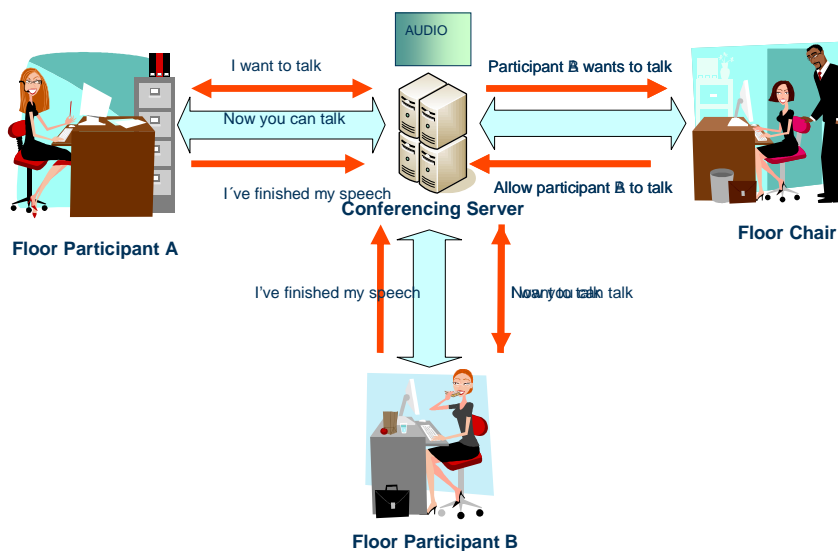
Contents

- What is Floor Control?
- BFCP: Example of an audio conference
- Packet Format
- Message Format
- Operations
- Connection Establishment in BFCP
 - Offer/Answer exchange mechanism
 - General mechanism

What is Floor Control?

- Floor Control: A mechanism that enables applications or users to gain safe and mutually exclusive or non-exclusive input access to the shared object or resource.
- Binary Floor Control Protocol: BFCP is a protocol to coordinate access to shared resources in a conference.

BFCP: Example of an audio conference



Terminology

- Floor : A permission to temporarily access or manipulate a specific shared resource or set of resources.
- Floor Chair: A logical entity that manages one floor (grants, denies, or revokes a floor).
- Floor Participant: A logical entity that requests floors from a floor control server.
- Floor Control Server: A logical entity that maintains the state of the floor(s) including which floors exists, who the floor chairs are, who holds a floor, etc.



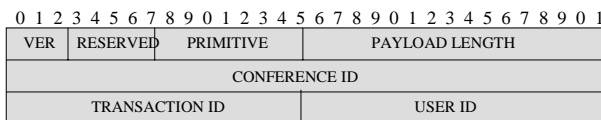
Packet Format

- Two types of attribute format:
 - Normal attribute format: encoded in TLV (Type-Length-Value) format.
 - Group attribute format: a sequence of normal attributes
 - Example of Group attribute format:
 - BENEFICIARY-INFORMATION
 - FLOOR-REQUEST-INFORMATION
 - REQUESTED-BY-INFORMATION



Packet Format (1)

- Common Header Format

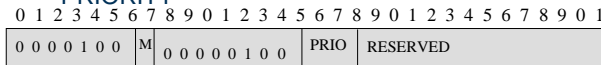


- Attribute Format

– BENEFICIARY-ID, FLOOR-ID, FLOOR-REQUEST-ID



– PRIORITY

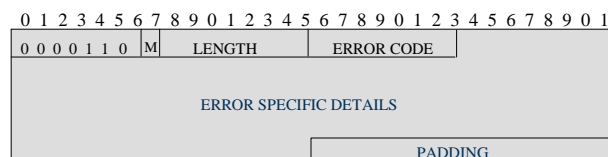


– REQUEST-STATUS

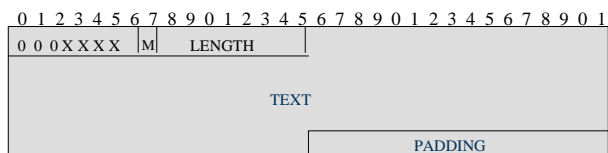


Packet Format (2)

– ERROR-CODE



– ERROR-INFO, PARTICIPANT-PROVIDED-INFO, STATUS-INFO, USER-DISPLAY-TEXT, USER-URI



Packet Format (3)

- SUPPORTED-ATTRIBUTES, SUPPORTED-PRIMITIVES

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
0	0	0	1	0	1	X	M				LENGTH										PRIMITIVE / ATTRIBUTE / PRIMITIVE
																					PRIMITIVE / ATTRIBUTE / PRIMITIVE
.....											PADDING										

- BENEFICIARY-INFORMATION

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
0	0	0	1	1	1	0	M				LENGTH										BENEFICIARY ID

BENEFICIARY INFORMATION = (BENEFICIARY-INFORMATION-HEADER)
 [USER-DISPLAY-NAME]
 [USER-URI]
 *[EXTENSION-ATTRIBUTE]

- REQUESTED-BY-INFORMATION

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
0	0	1	0	0	0	0	M				LENGTH										REQUESTED-BY ID

BENEFICIARY INFORMATION = (REQUESTED-BY-INFORMATION-HEADER)
 [USER-DISPLAY-NAME]
 [USER-URI]
 *[EXTENSION-ATTRIBUTE]



Packet Format (4)

- FLOOR-REQUEST-INFORMATION

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
0	0	0	1	1	1	1	M				LENGTH										FLOOR REQUEST ID

FLOOR-REQUEST-INFORMATION = (FLOOR-REQUEST-INFORMATION-HEADER)
 (REQUEST-STATUS)
 1*(FLOOR-ID)
 [BENEFICIARY-INFORMATION]
 [REQUESTED-BY-INFORMATION]
 [PRIORITY]
 [PARTICIPANT-PROVIDED-INFO]
 [STATUS-INFO]
 *[EXTENSION-ATTRIBUTE]



Message Format

FloorRequest: Floor participants request a floor by sending this message to the floor control server.

```
FloorRequest = (COMMON-HEADER )
               *(FLOOR-ID)
               [BENEFICIARY-ID]
               [PARTICIPANT-PROVIDED-INFO]
               [PRIORITY]
               *[EXTENSION-ATTRIBUTE]
```

FloorRelease: Floor participants release a floor by sending this message to the floor control server.

```
FloorRelease = (COMMON-HEADER )
               (FLOOR-REQUEST-ID)
               *[EXTENSION-ATTRIBUTE]
```

FloorRequestQuery: Floor participants and floor chairs request information about a floor request.

```
FloorRequestQuery = (COMMON-HEADER )
                    (FLOOR-REQUEST-ID)
                    *[EXTENSION-ATTRIBUTE]
```



Message Format (1)

FloorRequestStatus: Floor control server informs floor participant and floor chair about the status of their floor request.

```
FloorRequestStatus = (COMMON-HEADER )
                    [FLOOR-REQUEST-INFORMATION]
                    *[EXTENSION-ATTRIBUTE]
```

UserQuery: Floor participants and floor chairs request information about a user.

```
UserQuery = (COMMON-HEADER )
            [BENEFICIARY-ID]
            *[EXTENSION-ATTRIBUTE]
```

UserStatus: Floor control server informs floor participant and floor chair about the status of their user request.

```
UserStatus= (COMMON-HEADER )
            [BENEFICIARY-INFORMATION]
            1 *(FLOOR-REQUEST-INFORMATION)
            *[EXTENSION-ATTRIBUTE]
```



Message Format (2)

FloorQuery: Floor participants and floor chairs request information about a floor.

```
FloorQuery = (COMMON-HEADER )
              *(FLOOR-ID)
              *[EXTENSION-ATTRIBUTE]
```

FloorStatus: Floor control server informs floor participant and floor chair about the status of a floor.

```
FloorStatus = (COMMON-HEADER )
              (FLOOR-ID)
              *[FLOOR-REQUEST-INFORMATION]
              *[EXTENSION-ATTRIBUTE]
```

ChairAction: Floor chairs send instruction to floor control servers.

```
ChairAction = (COMMON-HEADER )
              1*(FLOOR-ID)
              (FLOOR-REQUEST-ID)
              (REQUEST-STATUS)
              [STATUS-INFO]
              *[EXTENSION-ATTRIBUTE]
```



Message Format (3)

ChairActionAck: Floor control servers confirm that they have accepted a ChairAction message.

```
ChairActionAck = (COMMON-HEADER )
                 *[EXTENSION-ATTRIBUTE]
```

Hello: FloorParticipants and floor chairs checks the liveness of floor control servers.

```
Hello = (COMMON-HEADER )
         *[EXTENSION-ATTRIBUTE]
```

HelloAck: Floor Control servers confirm that they are alive on reception of a Hello message.

```
HelloAck= (COMMON-HEADER )
           (SUPPORTED-PRIMITIVES)
           (SUPPORTED-ATTRIBUTES)
           *[EXTENSION-ATTRIBUTE]
```

Error: Floor control servers inform floor participants and floor chairs about errors processing requests.

```
Error= (COMMON-HEADER )
        (ERROR-CODE)
        [ERROR-INFO]
        *[EXTENSION-ATTRIBUTE]
```

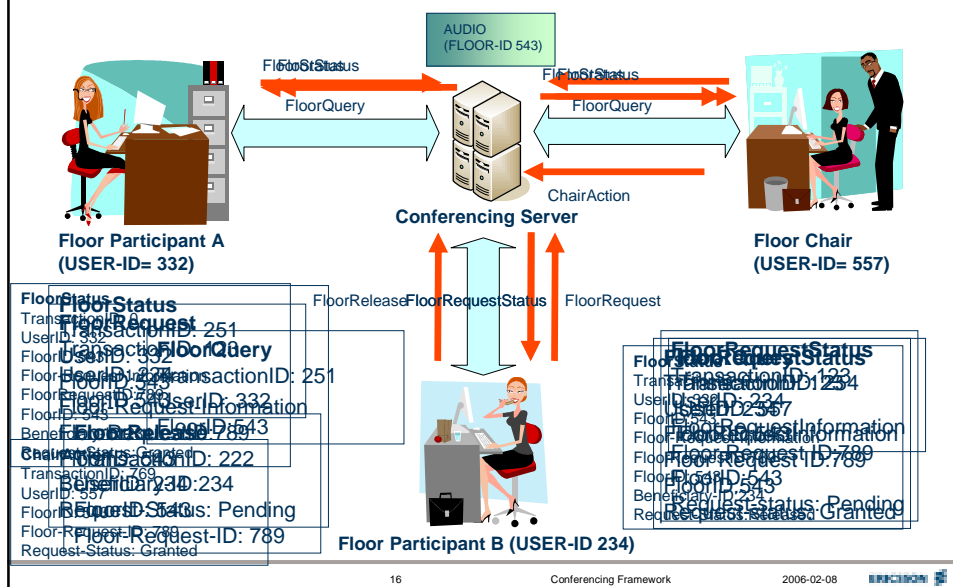


Transaction Format

- There are two types of transactions:
 - client-initiated transactions: they consist of a request from a client to a floor control server and a response from the floor control server to the client. The request carries a Transaction ID in its common header which the floor control server copies into the response. Clients use Transaction ID values to match responses with previously- issued requests.
 - server-initiated transactions (notifications): They consist of a single message from a floor control server to a client. Since they do not trigger any response, their Transaction ID is set to 0.



Operations: Example of an audio conference



Connection establishment in BFCP

- **Offer/answer exchange mechanism:** Floor Control clients establish BFCP connections with the Floor Control server within the context of an offer/answer exchange using SDP.
- **General mechanism:** Floor Control client establishes a connection to a BFCP floor control server outside the context of an offer/answer exchange. This mechanism also specifies a digest authentication mechanism for BFCP based on shared secrets.

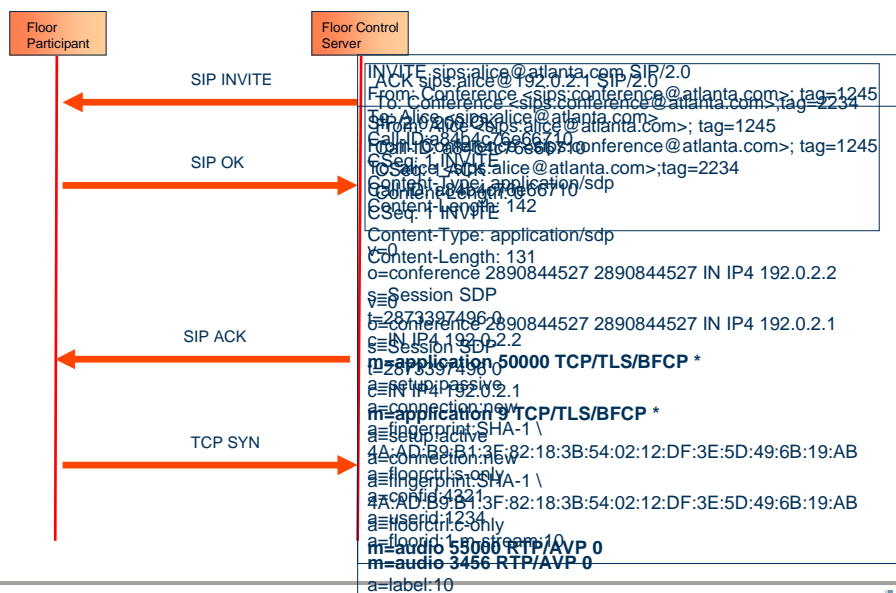
17

Conferencing Framework

2006-02-08



Offer/answer exchange mechanism



18

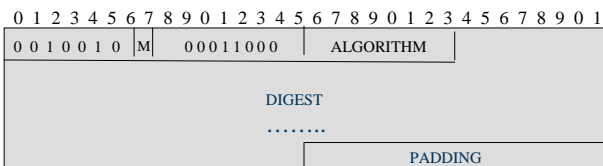
Conferencing Framework

2006-02-08

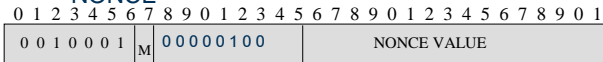


General mechanism:attributes

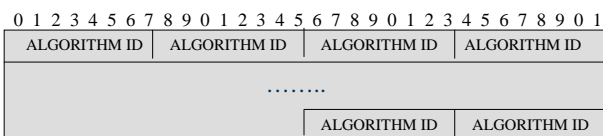
- DIGEST



- NONCE



- Definition of Error Specific Details for Error Code 10 (DIGEST Attribute Needed)



General mechanism:Example

