Binary Sensor Service

***Bluetooth*® Service Specification**

- **Revision:** v1.0
- **Revision Date:** 2019-07-02
- **Group Prepared By:** PUID Working Group
- **Feedback Email:** rd-main@bluetooth.org

Abstract:

The Binary Sensor Service (BSS) enables devices with one or more binary (for example, open/closed, on/off) sensors to report the state of the binary sensors to a client device.

Revision History

Revision Number	Date	Comments
v1.0	2019-07-02	Adopted by the Bluetooth SIG Board of Directors.

Contributors

Name	Company
Hironari Ushikubo	NTT DOCOMO
Frank Berntsen	Nordic Semiconductor

Contents

1 Introduction

The Binary Sensor Service exposes two characteristics that are used for a protocol between the Server and a client to transfer the status of binary sensors on the server side.

1.1 Conformance

If conformance to this specification is claimed, all capabilities indicated as mandatory for this specification shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated.

1.2 Service dependencies

This service is not dependent upon any other services.

1.3 Bluetooth Core Specification release compatibility

This specification is compatible with any Bluetooth Core Specification that includes the Generic Attribute Profile (GATT) portion of the Bluetooth Core Specification [1].

1.4 GATT sub-procedure requirements

Requirements in this section represent a minimum set of requirements for a Server implementing the Binary Sensor Service. Other GATT sub-procedures may be used if supported by both client and Server.

Table 1.1 summarizes additional GATT sub-procedure requirements beyond those required by all GATT servers.

GATT sub-procedure	Requirements
Write Characteristic Value	M
Indications	M

Table 1.1: Additional GATT sub-procedure requirements

M: Mandatory

1.5 Transport dependencies

This service is specified for operation over the Bluetooth Low Energy (LE) transport.

1.6 Application error codes

This service does not define any application APA error codes. This service does not define any application Attribute Protocol Application error codes (as defined in [1] Volume 3, Part F).

1.7 Byte transmission order

All characteristics used with this service shall be transmitted with the least significant octet first (i.e., little endian). The least significant octet is identified in the characteristic definitions in [2].

1.8 Language

1.8.1 Language conventions

The Bluetooth SIG has established the following conventions for use of the words ***shall***, ***must***, ***will***, ***should***, ***may***, ***can***, ***is***, and ***note*** in the development of specifications:

shall	is required to – used to define requirements.
must	is used to express: a natural consequence of a previously stated mandatory requirement. OR an indisputable statement of fact (one that is always true regardless of the circumstances).
will	it is true that – only used in statements of fact.
should	is recommended that – used to indicate that among several possibilities one is recommended as particularly suitable, but not required.
may	is permitted to – used to allow options.
can	is able to – used to relate statements in a causal manner.
is	is defined as – used to further explain elements that are previously required or allowed.
note	Used to indicate text that is included for informational purposes only and is not required in order to implement the specification. Each note is clearly designated as a “Note” and set off in a separate paragraph.

For clarity of the definition of those terms, see Core Specification Volume 1, Part E, Section 1.

1.8.2 Reserved for Future Use

Where a field in a packet, Protocol Data Unit (PDU), or other data structure is described as "Reserved for Future Use" (irrespective of whether in uppercase or lowercase), the device creating the structure shall set its value to zero unless otherwise specified. Any device receiving or interpreting the structure shall ignore that field; in particular, it shall not reject the structure because of the value of the field.

Where a field, parameter, or other variable object can take a range of values, and some values are described as "Reserved for Future Use," a device sending the object shall not set the object to those values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous; however, this does not apply in a context where the object is described as being ignored or it is specified to ignore unrecognized values.

When a field value is a bit field, unassigned bits can be marked as Reserved for Future Use and shall be set to 0. Implementations that receive a message that contains a Reserved for Future Use bit that is set to 1 shall process the message as if that bit was set to 0, except where specified otherwise.

The acronym RFU is equivalent to Reserved for Future Use.

1.8.3 Prohibited

When a field value is an enumeration, unassigned values can be marked as “Prohibited.” These values shall never be used by an implementation, and any message received that includes a Prohibited value shall be ignored and shall not be processed and shall not be responded to.

Where a field, parameter, or other variable object can take a range of values, and some values are described as “Prohibited,” devices shall not set the object to any of those Prohibited values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous.

“Prohibited” is never abbreviated.

1.9 Terminology

Initially capitalized terms used in this specification have the meanings ascribed to them in Table 1.2.

Term	Definition
Multiple Sensor	Exists when there is more than one Sensor Element of the same type (open/closed [O/C], human detection [HD], or vibration [VIB]) in a device; together, all of the Sensor Elements of the same sensor type are a Multiple Sensor.
Sensor Element	A “physical” sensor with binary output connected to the device.
Sensor State	The Boolean state of the Sensor Element, where 0 and 1 represents closed and open or not-detected and detected.
Sensor Status	The combination of Sensor State and Count (the number of times Sensor State has changed from 0 to 1) in one 2-octet entity.
Server	A device implementing the Binary Sensor Service. This device can support more than one sensor.
Single Sensor	Exists when there is only one Sensor Element of a particular type in a device, even if there are other Sensor Elements of other sensor types.

Table 1.2: Terminology

2 Service

2.1 Declaration

The service UUID shall be set to «Binary Sensor» as defined in [2].

2.2 Behavior

The Binary Sensor Service (BSS) monitors one or more binary sensors connected to the device. The status of the sensors (for example, on/off or open/closed) shall be reported when requested by a client device, when a sensor changes state, or at a regular interval controlled by a timer.

The Binary Sensor Service implements a protocol using two characteristics of the Binary Sensor Service. Commands are received on the BSS Control Point characteristic. Responses and unsolicited events are indicated to the BSS Response characteristic. The details of the protocol can be found in Section 4.

2.2.1 Timer

The Server may have a timer to schedule regular reports of the Sensor Status.

If a timer is implemented, the timer shall create an event at a given fixed rate. The interval shall be longer than or equal to the connection interval.

2.2.2 Sequence

This section contains more details about the behavior of the Binary Sensor Service.

When the Server is unknown to the client (for example, after the first connection is established), the client will normally start with sending Get Sensor Status Command messages or Setting Sensor Command messages (as defined in Table 4.3) for all possible sensor types to discover which sensors are available in the sensor device. Figure 2.1 illustrates the discovery of a Server with a single open/closed (O/C) sensor.

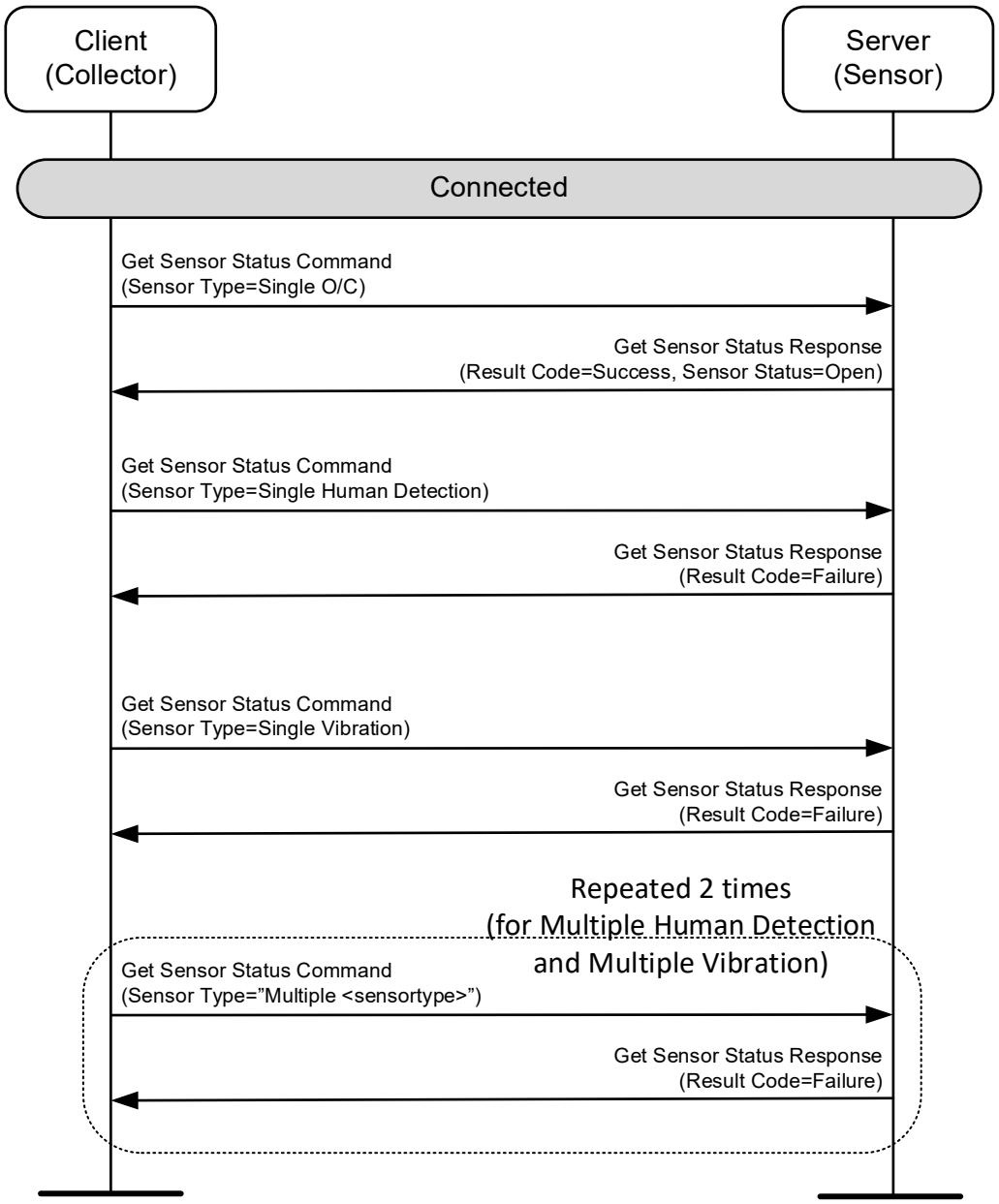

Figure 2.1: Discovery of a simple O/C sensor

After this sequence, the client will know that this device has 1 O/C sensor.

Figure 2.2 illustrates the discovery of a device with 7 named vibration sensors.

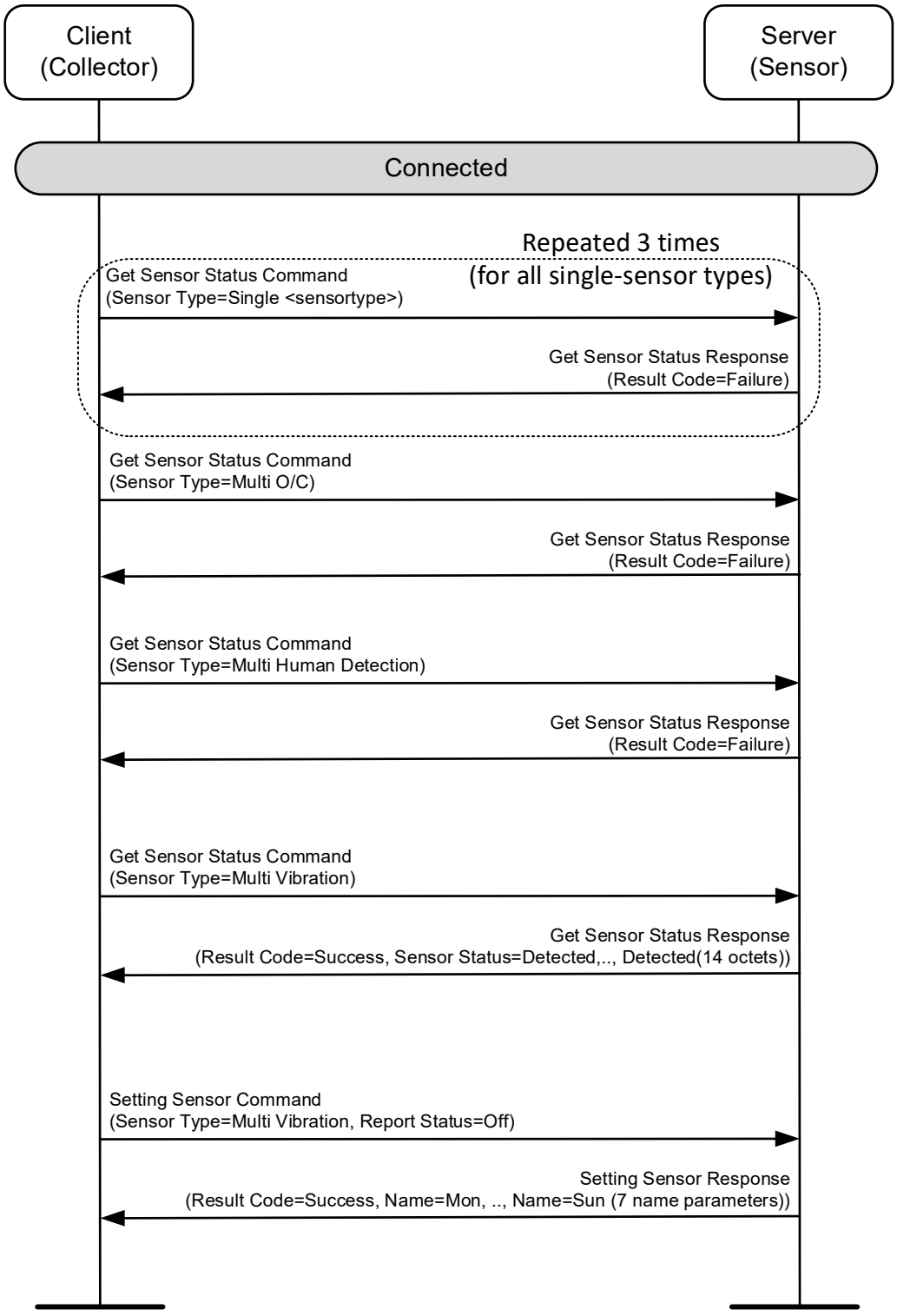

Figure 2.2: Discovery of a complex vibration sensor

After this sequence, the client will know that the device has 7 vibration sensors named “Mon,” “Tue,” "Wed," "Thu," "Fri," "Sat," and ”Sun.”

Triggered by the client request to start/stop notification, the Server responds with the sequence shown in Figure 2.3.

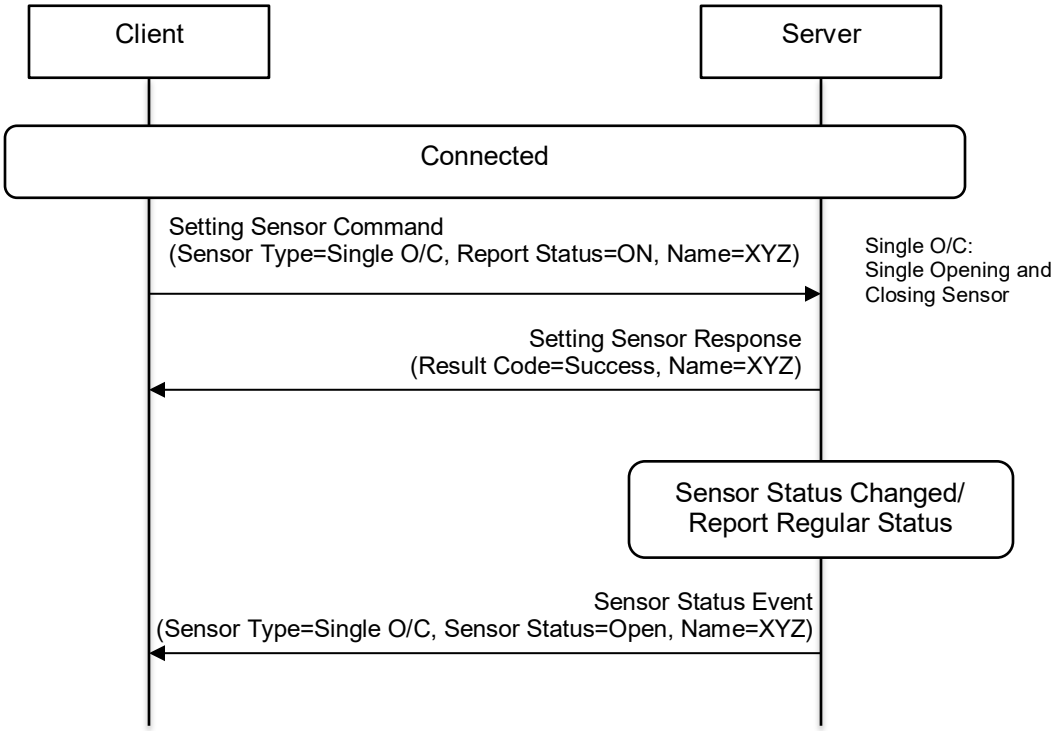

Figure 2.3: Sequence for starting/stopping Sensor Status notifications

Triggered by the client request to obtain the current sensor value, the Server responds as shown in Figure 2.4.

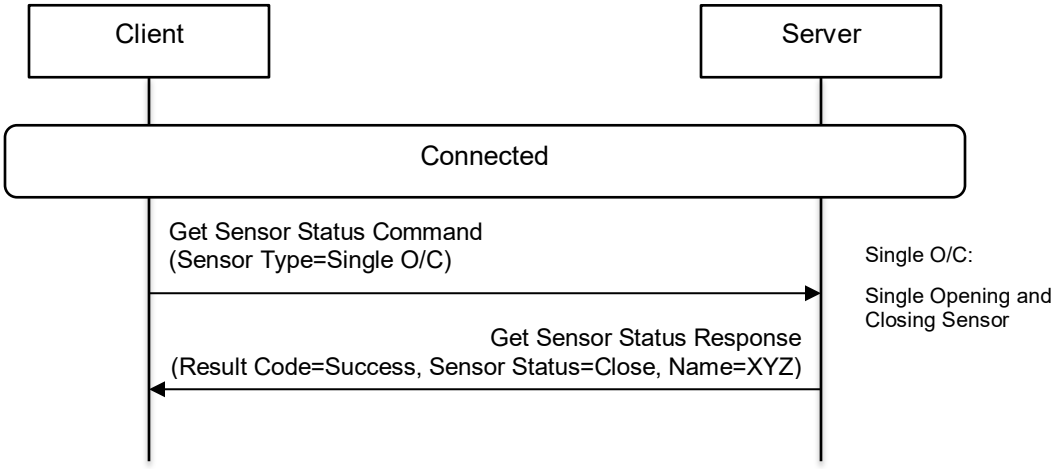

Figure 2.4: : Sequence for obtaining Sensor Status

3 Service characteristics

This section defines the characteristic and descriptor requirements for the Binary Sensor Service. Table 3.1 defines the requirements for the two service characteristics.

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
BSS Control Point	M	Write	N/A	Defined by higher-level specification.
BSS Response	M	Indicate	N/A	Defined by higher-level specification.

Table 3.1: Service characteristics requirements

M: Mandatory

The structure of the service characteristics is defined in Table 3.2.

	Split Header	Payload
Byte Order	N/A	LSO…MSO
Data type	uint8	Variable
Size	1 octet	1 to 19 octets
Units	None	None

Table 3.2: Structure of the BSS Control Point and BSS Response characteristics

3.1 BSS Control Point

The BSS Control Point characteristics shall be used to receive the commands for the Server.

The structure of the characteristic is defined in Table 3.2.

3.1.1 Characteristic behavior

The Server shall implement the GATT Write Characteristic Value sub-procedure (as defined in [1] Volume 3, Part G). When the BSS Control Point is written using this procedure, the server shall interpret the Split Header field to assemble a complete command payload as described in Section 5. When a complete command is assembled, the Server shall execute the appropriate procedure as described in Section 4.

3.1.1.1 Split Header field

The Split Header field is used for reassembling commands.

Table 3.3 and Figure 3.1 describe the Split Header field.

Field Name	Size(bits)	Description	Requirement
Execute Flag	1	Flag to indicate packet end	M
Sequence Number	5	Sequence Number of split packets	M
RFU	1	Reserved for Future Use	-
Source Flag	1	Flag to indicate packet transmitted direction.	M

Table 3.3: Split Header format

M: Mandatory

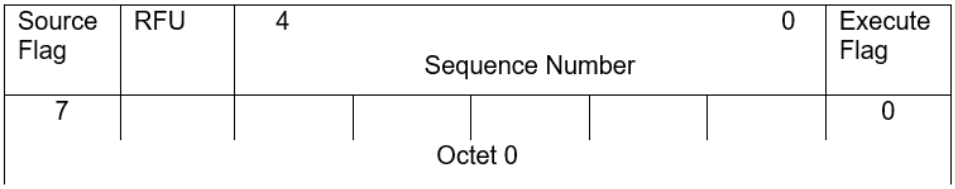

Figure 3.1: Split Header bit field ordering

See Section B.2 (Appendix B) for an example of the Split Header field usage.

3.1.1.1.1 Source Flag

The Source Flag is a simple mark to comprehend the direction of transmitted packets.

The Source Flag enables a service application to easily control and maintain data exchanges in packet level.

Table 3.4 describes values for the Source Flag.

Value	Description
0	Indicate direction from client to Server
1	Indicate direction from Server to client

Table 3.4: Source Flag values

3.1.1.1.2 Sequence Number

Sequence Number is an unsigned 5-bit integer (range 0 to 31) that indicates a sequence order of split packets.

When there is no split packet, the value 0x00 shall be set for a non-split packet.

3.1.1.1.3 Execute Flag

Execute Flag indicates that a split packet is terminated.

The service application can distinguish whether the packet is split or final by this flag.

Table 3.5 describes values for Execute Flag.

Value	Description
0	Split packet, not final packet
1	Not a split packet or, indicates final packet

Table 3.5: Execute Flag values

For a packet that is not a split packet, Execute Flag shall be set to 1.

3.1.1.2 Payload field

The Payload field holds a fragment of command data.

3.2 BSS Response

The BSS Response characteristics shall be used to send responses and unsolicited events from the server.

The structure of the characteristic is defined in Table 3.2.

3.2.1 Characteristic behavior

When the Server executes any of the procedures described in Section 4, the Server shall set the Split Header field and the Payload field as described in Section 5 and indicate this characteristic for each fragment, by executing the GATT Indications sub-procedure, until the entire Payload is sent.

3.2.1.1 Split Header

The Split Header field is used in for reassembling commands. See Section 3.1.1.1 for the format of the Split Header field.

3.2.1.2 Payload

The Payload field is used to hold a fragment of the response or event data.

4 Protocol

This section defines the protocol used for the Server.

Procedures involve sending one or more messages between the service and a client device. For command procedures, the procedure is initiated by the Server receiving a command message and responding with a response message. For event procedures, the Server will send an event message without any request from a client.

Each message consists of a message identifier followed by one or more parameters.

The protocol defines two different sub-types of sensors, Single Sensor and Multiple Sensor.

- If the Server has only one Sensor Element of the same type, that Sensor Element is a Single Sensor.
- If the Server has more than one Sensor Element of the same type, those Sensor Elements are a Multiple Sensor.

See Appendix A for some examples of a Single Sensor and a Multiple Sensor in a device.

The procedure requirements are the same for both a Single Sensor and a Multiple Sensor, but some message formats are different.

The Server shall support at least one Single Sensor or one Multiple Sensor.

The Server may support a combination of several sensors, each of which can be either a Single Sensor or a Multiple Sensor.

The Server may support Named Sensors to allow a client to read and/or change the name for each individual sensor.

4.1 BSS Control Point and BSS Response procedures

Table 4.1 lists the procedures and options for the protocol.

Procedure	Requirement	Description
Get Sensor Status Command	M	See Section 4.1.1
Setting Sensor Command	M	See Section 4.1.2
Sensor Status Event	M	See Section 4.1.3
Named Sensors	O	See Sections 4, 4.2.3, and 4.3.3.7
Single Sensor	C.1	See Sections 4, 4.2.3, and 4.3.3.5
Multiple Sensor	C.1	See Sections 4, 4.2.3, and 4.3.3.6

Table 4.1: Procedures and options for the Server

M: Mandatory

O: Optional

C.1: The Server shall support at least one Single Sensor or one Multiple Sensor.

These procedures are executed for one sensor type (any one value of the Sensor Type parameter) at a time.

4.1.1 Get Sensor Status Command procedure

When the Server receives a Get Sensor Status Command Message on the BSS Control Point characteristic, the Server shall respond with the Get Sensor Status Response message on the BSS Response characteristic.

If the Server has the requested sensor data, the Result Code parameter shall be set to Success and the Sensor Status parameter or the Multi Sensor Status parameter, depending on the value of the Sensor Type parameter, shall be set to Sensor State and Count (the number of times the Sensor State has changed from 0 to 1, as defined in Table 4.13) for each Sensor Element in the sensor.

If an internal error in the Server is preventing it from performing the action, the Result Code parameter shall be set to Failure, and the response shall not contain other parameters.

4.1.2 Setting Sensor Command procedure

When the Server receives a Setting Sensor Command message on the BSS Control Point characteristic, the Server shall perform the requested setting and shall respond with the Setting Sensor Response message on the BSS Response characteristic.

The Setting Sensor Command shall apply to the sensor that matches the sensor type provided in the Sensor Type parameter.

If the Server does not have any sensor with the specified sensor type, the Result Code parameter shall be set to Failure, and the response shall not contain other parameters.

If the command includes one or more Name parameters and the sensor does not support Named Sensors, the Result Code parameter shall be set to Failure, and the response shall not contain other parameters.

If an internal error in the Server is preventing it from performing the action, the Result Code parameter shall be set to Failure, and the response shall not contain other parameters.

The Server shall set the sensor notification status (as defined in Table 4.12) for the designated sensor to the state indicated by the Report Status parameter. The initial value of the notification status after connection establishment is Off.

If the command includes one or more Name parameters, and the Server supports Named Sensors, the Server shall set the sensor names of the individual sensors as specified by the Name parameters.

If the change of notification status and setting of sensor names is successful, the Server shall set the Result Code parameter of the response command to Success.

If the Server supports Named Sensors, the Server shall include one Name parameter for each individual sensor in the response command. If the command also included Name parameter(s), the Server shall report the names after the application of the command.

4.1.3 Sensor Status Event procedure

If the Server detects an event trigger, the Server shall send a Sensor Status Event on the BSS Response characteristic.

The event trigger can be one or more binary sensors changing the state, the timer indicating the time for a regular status report, or the sensor application deciding to halt reporting (for example, to conserve energy).

The following parameters shall be set in the message:

- When the event trigger is a change in the sensor state, the Server shall set the Sensor Type parameter to the sensor type that detected the state change, and the Sensor Status parameter or Multiple Sensor Status parameter, depending on the value of the Sensor Type parameter, to the Sensor State after the state change and the Count for each Sensor Element in the sensor.
- When the event trigger is a timer (scheduled status report), the Server shall set the Sensor Type parameter to the type of sensor being reported and the Sensor Status parameter or Multiple Sensor Status parameter, depending on the value of the Sensor Type parameter, to the Sensor State and Count at the time of the event trigger for each Sensor Element in the sensor.
- When the event trigger is the sensor halting its reporting, the Server shall set the Cancel parameter to Stop Report.

4.2 Messages

Each message is composed of a message header followed by a message payload. The message header is sub-divided into four fields: Message ID, Number of Parameters, and two fields reserved for future use. The structure of a message is defined in Table 4.2.

LSO **MSO**

	Message Header				**Message Payload**
	RFU	**Message ID**	**RFU**	**Number of Parameters**	
Byte Order	-	-	-	-	LSO…MSO
Data type	-	uint8	-	uint8	-
Size	1 octet	1 octet	1 octet	1 octet	Variable
Field Status	M	M	M	M	M

Table 4.2: Structure of a message

M: Mandatory

4.2.1 Message ID field

The Message ID identifies the type of the message.

The Message IDs are defined in Table 4.3.

Message ID	Message	Characteristic	Requirement
0x00	Get Sensor Status Command	BSS Control Point	M
0x01	Get Sensor Status Response	BSS Response	M
0x02	Setting Sensor Command	BSS Control Point	M
0x03	Setting Sensor Response	BSS Response	M
0x04	Sensor Status Event	BSS Response	M
0x05 – 0xFF	RFU	-	-

Table 4.3: Message IDs

M: Mandatory

4.2.2 Number of Parameters field

Number of Parameters indicates the number of parameters contained in the Payload.

	Data Type	Range
Number of Parameters	uint8	1 – 255

Table 4.4: Number of Parameters

Note: In practice, the number of parameters in a message might be limited to less than 255 because of the limitation of the Message Payload length (limited by the maximum payload length and the maximum number of segments in a message).

4.2.3 Message Payload field

Message Payload field contains one or more parameters.

The parameters included in a message are different for each Message ID. For the response messages and the event message, there are two different sets of parameters depending on whether the sensor type is a Single Sensor or Multiple Sensor. The requirements for each message type and sensor class are defined in Table 4.5.

Message	Parameter 0	Parameter 1	Parameter 2	…	Parameter N	Parameter N+1
Get Sensor Status Command	Sensor Type (M)	(X)	(X)	(X)	(X)	(X)
Get Sensor Status Response (Single)	Result Code (M)	Sensor Status (C.1)	(X)	(X)	(X)	(X)

Message	Parameter 0	Parameter 1	Parameter 2	…	Parameter N	Parameter N+1
Get Sensor Status Response (Multiple)	Result Code (M)	Multiple Sensor Status (C.1)	(X)	(X)	(X)	(X)
Setting Sensor Command	Sensor Type (M)	Report Status (M)	Name (O)	…	Name (O)	Name (O)
Setting Sensor Response (Single)	Result Code (M)	Name (O)	(X)	(X)	(X)	(X)
Setting Sensor Response (Multiple)	Result Code (M)	Name (O)	Name (O)	…	Name (O)	(X)
Sensor Status Event (Single)	Sensor Type (M)	Sensor Status (M)	(X)	(X)	(X)	(X)
Sensor Status Event (Multiple)	Sensor Type (M)	Multiple Sensor Status (M)	(X)	(X)	(X)	(X)
Sensor Status Event (Stopping)	Cancel (M)	(X)	(X)	(X)	(X)	(X)

Table 4.5: Required and optional parameters per Message ID

M: Mandatory

O: Optional

X: Excluded

C.1: Mandatory when the Server successfully obtains sensor status; otherwise excluded

4.3 Parameters

Parameters include a field to identify a data type and a value. Each parameter in a message is composed of four fields; Parameter ID, Parameter Length, Parameter Value, and one field reserved for future use.

Table 4.6 shows the structure of a parameter.

	LSO			MSO
	Parameter ID	**Parameter Length**	**RFU**	**Parameter Value**
Byte Order	-	-	-	LSO…MSO
Data type	uint8	uint8	-	-

	LSO			MSO
	Parameter ID	**Parameter Length**	**RFU**	**Parameter Value**
Size	1 octet	1 octet	2 octets	Variable The Parameter Length field defines the size, in octets, of the Parameter Value field.
Requirement	M	M	M	M

Table 4.6: Structure of a parameter

M: Mandatory

4.3.1 Parameter ID field

Parameter ID identifies the type of parameter. Parameter values are different depending on the parameter type.

Table 4.7 defines the parameter types.

Parameter ID	Parameter	Parameter Value Field Length	Description	Section
0x00	Result Code	1 octet	Operation result in Server for request (command) from Client	4.3.3.1
0x01	Cancel	1 octet	Stop notification of sensor status by Server event	4.3.3.2
0x02	Sensor Type	1 octet	Sensor type	4.3.3.3
0x03	Report Status	1 octet	Notification On/Off	4.3.3.4
0x04 – 0x09	RFU	-	Reserved for Future Use	-
0x0A	Sensor Status	2 octets	Sensor status	4.3.3.5
0x0B	Multiple Sensor Status	variable	Sensor status aggregated for the whole sensor	4.3.3.6
0x0C	Name	variable	Sensor name	4.3.3.7
0x0D – 0xFF	RFU	-	Reserved for Future Use	-

Table 4.7: Parameter format – Parameter IDs

4.3.2 Parameter Length field

Parameter Length defines the data size of the Parameter Value field.

	Data Type	Range	Units
Parameter Length	uint8	1 – 255	octet

Table 4.8: Parameter Format – Parameter Length

Note: In practice the length of each parameter in a message might be limited to less than 255 because of the limitation of the Message Payload length (limited by the maximum payload length and the maximum number of segments in a message).

4.3.3 Parameter Value field

Parameter Value has different formats depending on the Parameter ID.

Available values are described in Sections 4.3.3.1 to 4.3.3.7.

4.3.3.1 Result Code

Result Code indicates operation results in the Server for a request (command) from the client.

Table 4.9 describes the values of the Result Code.

Parameter Value	Parameter	Description
0x00	Success	Normal completion
0x01	Failure	Requested operation not performed
0x02 – 0xFF	RFU	Reserved for Future Use

Table 4.9: Result Code parameter values

4.3.3.2 Cancel

Cancel indicates that the sensor has stopped notification of sensor status.

Table 4.10 describes the value of the Cancel parameter.

Parameter Value	Parameter	Description
0x00	Stop Report	Notification of sensor status from this Server was stopped.
0x01 – 0xFF	RFU	Reserved for Future Use

Table 4.10: Cancel parameter values

4.3.3.3 Sensor Type

Sensor Type indicates type of sensor.

Table 4.11 describes the values of the Sensor Type.

Parameter Value	Parameter	Description
0x00	Opening and closing sensor	Message applies to an opening and closing sensor.
0x01	Human detection sensor	Message applies to a human detection sensor.
0x02	Vibration sensor	Message applies to a vibration sensor.
0x03 – 0x7F	RFU	Reserved for Future Use
0x80	Multiple opening and closing sensor	Message applies to a multiple opening and closing sensor.
0x81	Multiple human detection sensor	Message applies to a multiple human detection sensor.
0x82	Multiple vibration sensor	Message applies to a multiple vibration sensor.
0x83 – 0xFF	RFU	Reserved for Future Use

Table 4.11: Sensor Type parameter values

4.3.3.4 Report Status

The Report Status indicates whether event status notifications are turned on or off.

Table 4.12 describes the values of the Report Status.

Parameter Value	Parameter	Description
0x00	Off	Stop event notification of Sensor Status
0x01	On	Start event notification of Sensor Status
0x02 – 0xFF	RFU	Reserved for Future Use

Table 4.12: Report Status parameter

4.3.3.5 Sensor Status

Sensor Status indicates sensor status.

Sensor Status shall have the format shown in Table 4.13 and Figure 4.1.

Field Name	Size (bits)	Description
Count	11	A value from 0 to 2047 indicating the number of times the Sensor State has changed from 0 to 1. When the Count reaches 2047, the count restarts at 0.

Field Name	Size (bits)	Description
Sensor State	1	Indicates the sensor state. ▪ Opening and closing sensor: - 0: closed - 1: open ▪ Human detection sensor or Vibration sensor: - 0: not detected - 1: detected
RFU	4	Reserved for Future Use

Table 4.13: Sensor Status parameter

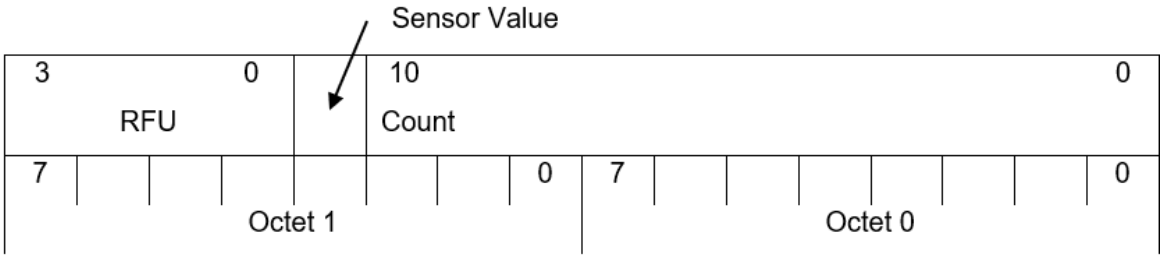

Figure 4.1: Sensor Status bit field ordering

4.3.3.6 Multiple Sensor Status

Multiple Sensor Status indicates sensor status of a Multiple Sensor. Multiple Sensor Status shall contain one Sensor Status field for each Sensor Element in the sensor.

Multiple Sensor Status shall have the format shown in Table 4.14.

	LSO			MSO
	Sensor Status 0	**Sensor Status 1**	**…**	**Sensor Status N-1**
Byte Order	LSO…MSO	LSO…MSO	…	LSO…MSO
Data type	struct*2	struct*2	…	struct*2
Size	2 octets	2 octets	…	2 octets

Table 4.14: Structure of Multiple Sensor Status

Note 1: N is the number of Sensor Elements in the Multiple Sensor.
Note 2: The *struct* for each Sensor Element complies with the format described in Section 4.3.3.5.

The Parameter Length for Multiple Sensor Status is 2 times the number of sensor elements. (Parameter Length = N * 2)

Multiple Sensor Status shall be treated as a sensor group status for the same type of sensors.

The Sensor Status for the Sensor Elements shall be collated in the same order in all reports. That is, Sensor Status 0 will always represent the same Sensor Element, as will Sensor Status 1, Sensor Status 2, and so forth through Sensor Status N-1.

4.3.3.7 Name

Name indicates sensor name.

Name shall have the format shown in Table 4.15.

LSO **MSO**

	Name
Byte Order	LSO…MSO
Data type	UTF8
Size	Variable (Parameter Length) octets

Table 4.15: Structure of Name

The length of a Name parameter value shall be from 1 to 32 octets.

5 Split transmission

The number of sensors and the sensor status data are different for each Server. To support various kinds of Servers for different usages, the Server might need to support messages longer than the size of the payload for the service characteristic (either the BSS Control Point characteristic or the BSS Response characteristic), as defined in Table 3.2. When a message exceeds the size of the payload that can be exchanged in a single message, the message will be transmitted using the procedures in this section.

The maximum payload for the BSS Control Point characteristic is referred to as Max Payload in Section 5.1.

5.1 Split transmission procedures

If the total size of the message is greater than Max Payload, multiple segments shall be sent over the BSS Control Point or BSS Response characteristic to send all of the data.

See Section 0.2 (Appendix B) for an example of how segmentation and reassembly works (split data transmission).

Table 5.1 lists the requirements for segmentation and reassembly procedures for the Server.

Procedure	Requirement	Description
Segmentation	C.1	See Section 5.1.1
Reassembly	C.2	See Section 5.1.2

Table 5.1: Segmentation and reassembly requirements for the Server

C.1. Mandatory if the Server has more than three Sensor Elements of the same type; otherwise excluded.

C.2: Mandatory if the Server supports the Named Sensors option; otherwise excluded.

5.1.1 Segmentation procedure

The number of segments (N) needed for transfer of the total message shall be calculated as:

N = Message size (octets) / Max Payload (octets), rounded up to the nearest whole number

The value of the Execute Flag bit of the Split Header field shall be set to 1 for the last segment transmitted; otherwise, it shall be set to 0.

The value of the Sequence Number field of the Split Header field shall be set to 0 for the first segment transmitted and incremented by one for each subsequent segment.

For each segment sent, the value of the Payload field shall be set to the next default Max Payload octets of the message or the remaining octets of the message, whichever is the smaller.

5.1.2 Re-assembly procedure

The receiving device shall check the value of the Execute Flag bit and the value of the Sequence Number field of the Split Header field for each received segment.

If the value of Sequence Number of the Split Header field is 0, the receiver shall start assembly of a new message. If the previous message is not complete, any data received shall be discarded.

If the value of the Sequence Number field of the Split Header field is not 0 and is not equal to the value of the Sequence Number field of the Split Header field for the previous segment incremented by 1, the receiver shall wait for the next Split Header with a value of 0 for the Sequence Number field.

If the Execute Flag bit of the Split Header field is 1, the receiver shall consider the message complete.

6 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
BS	Binary Sensor
BSS	Binary Sensor Service
GATT	Generic Attribute Profile
HD	human detection sensor
LSO	Least Significant Octet
MSO	Most Significant Octet
O/C	open/closed sensor
RFU	Reserved for Future Use
UUID	Universally Unique Identifier
VIB	vibration sensor

Table 6.1: Acronyms and abbreviations

7 References

[1] Bluetooth Core Specification, Version 5.0 or later

[2] Characteristic and Descriptor descriptions are accessible via the Bluetooth SIG Assigned Numbers

Appendix A: Examples of Single Sensor and Multiple Sensor

Example of a device equipped with one sensor (i.e., a Single Sensor).

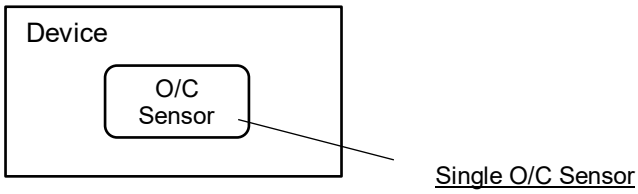

Figure A.1: A sensor device that has one Single Sensor

Example of a device equipped with two Sensor Elements that are not the same type, making each a Single Sensor.

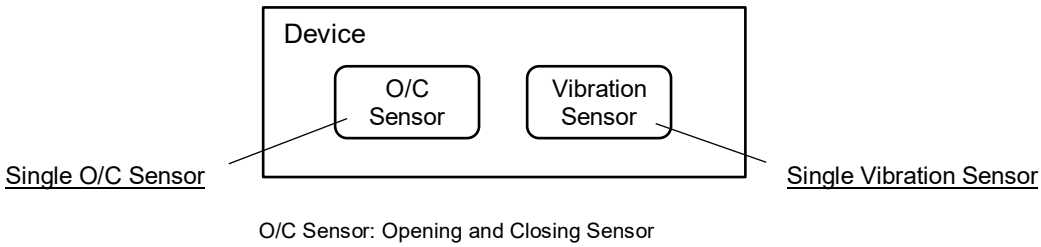

Figure A.2: Two Single Sensors

Example of a device equipped with two Sensor Elements of the same type (both of the Sensor Elements together are referred to as a Multiple Sensor).

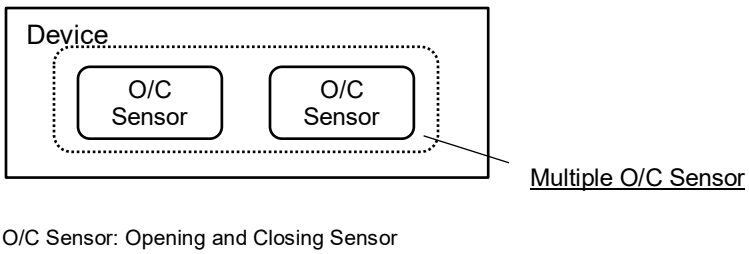

Figure A.3: One Multiple Sensor

Example of a device with one Single Sensor and one Multiple Sensor.

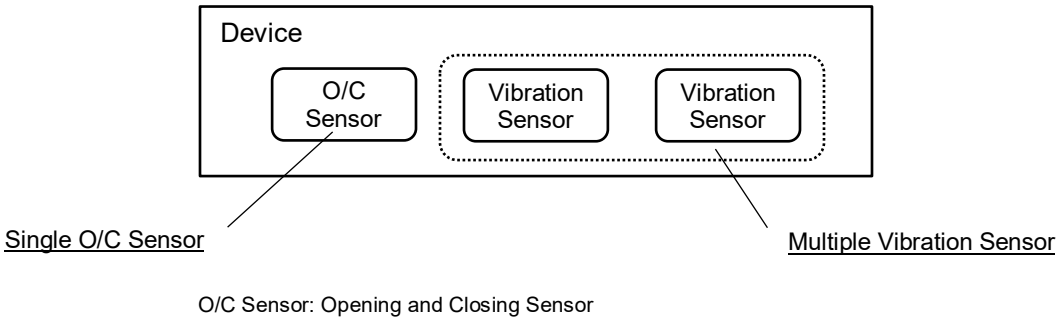

Figure A.4: One Single Sensor and one Multiple Sensor

Appendix B: Examples of Split Header with Segmentation Sequence

B.1 Non-split data transmission

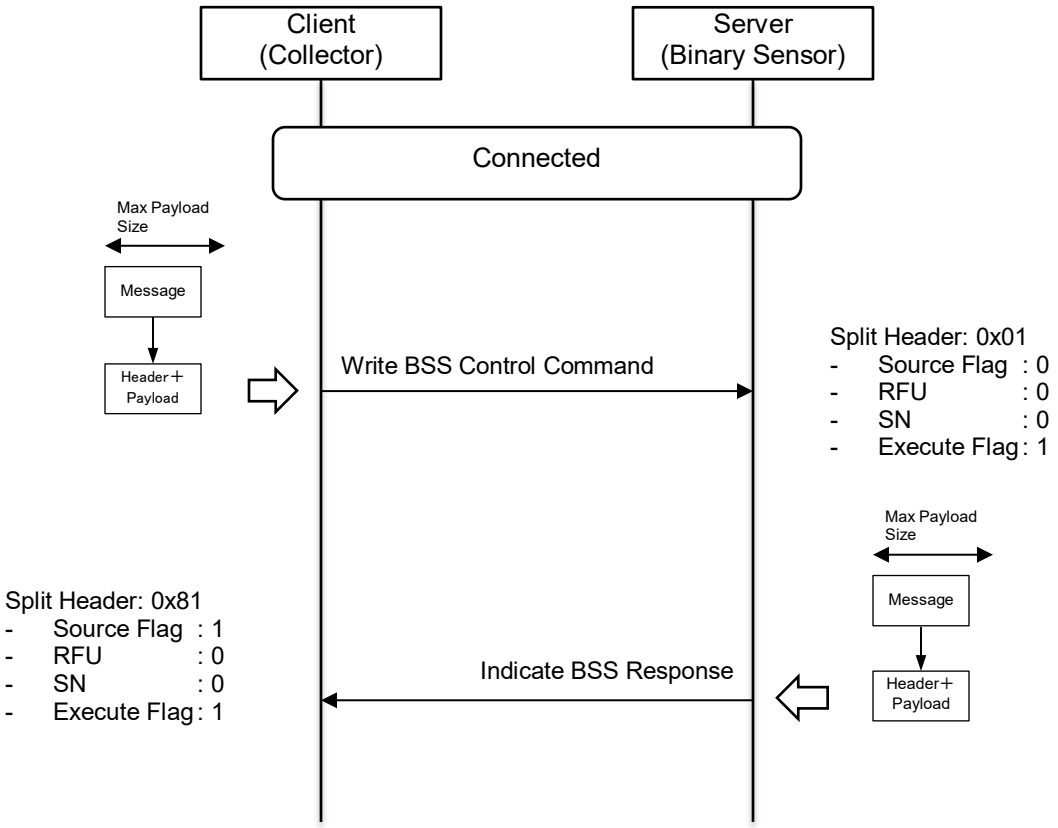

Figure B.1: Non-split data transmission

B.2 Split data transmission

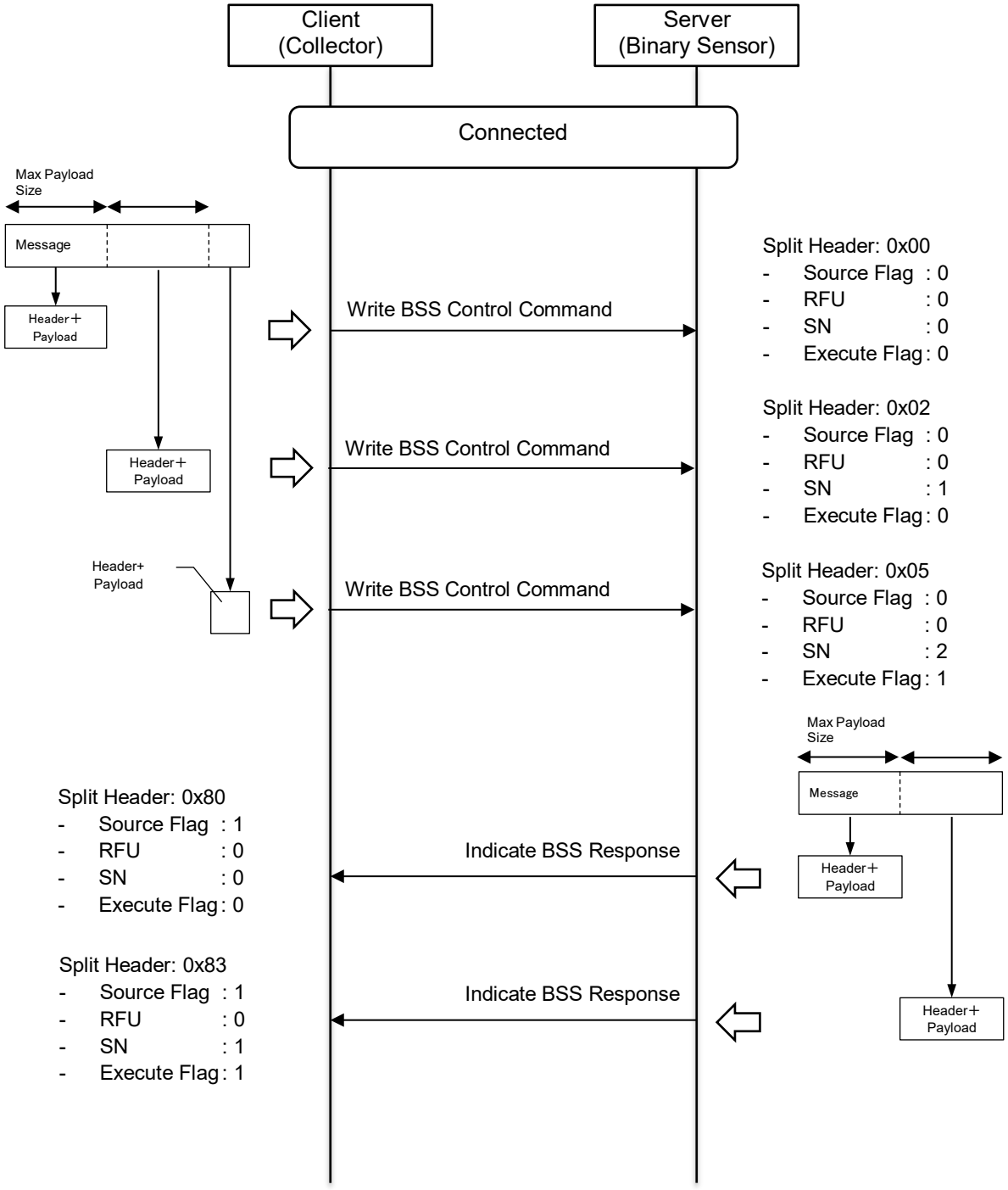

Figure B.2: Split data transmission