RW BLE Internet Protocol Support Profile Interface Specification

Interface Specification

RW-BLE-PRF-ISPP-IS

Version 2.00

2016-05-10

RivieraWaves
http://www.rivierawaves.com/

Revision History

Version	Date	Revision Description	Author
1.0	Mar 16th 2016	Initial Released version	GF
2.0	May 6th 2016	Update to align with new L2CAP CO CB Interface	GF

Table of Contents

1 Overview

1.1 Document Overview

This document describes the non-standard interface of the RW BLE Internet Protocol Support Profile (IPSP) implementation. Throughout this document, the interface messages will be referred to as API messages for the profile block(s).

Their description will include their usage and reason for implementation for a better understanding of the user and the developer that may one day need to interface them from a higher application.

1.2 Protocol Overview

The Bluetooth Low Energy Internet Protocol Support profile allows devices to discover and communicate to other devices that support IPSP. The communication between the devices that support IPSP is done using IPv6 packets over the Bluetooth Low Energy transport. However, the transmission of IPv6 packets over Bluetooth Low Energy is not part of this specification/implementation.

Within the profile, two roles can be supported: **Router** and **Node**. The Router role is used for devices that can route IPv6 packets. While the Node role is used for devices that can only originate or consume IPv6 application packets. A Router device must support the GAP Central Role but may also support the GAP Peripheral role. Similarly a Node device shall support the GAP Peripheral role, and may additionally support the GAP Central role. The profile requires a connection to be established between the two devices for its functionality.

The functionality of the profile requires only the presence of single IPSP service on one of the two devices, which the other device can discover. In this case, the Node device must have one instance of the Internet Protocol Service (IPS) in its attribute database. The Internet Protocol Router (IPR) will discover this service and initiates an underlying L2CAP connection oriented channel in LE Credit Based Flow Control Mode for the transport of IPv6 data.

The IPSP profile has been adopted by the Bluetooth SIG on December 16th 2014 ([1]). Their related Test Specifications have been released on December 23rd 2014 and is referenced in [2].

The profile is implemented in the RW-BLE software stack as two tasks, one for the server (Node- IPS Server – IPSS) role and one for the Client (Router- IPS Client - IPSC) role. Each task has an API decided after the study of the profile specifications and test specifications, and it is considered to be minimalistic and designed for a future application which would combine the profile functionality with the device connectivity and security procedures.

1.3 Firmware Implementation Overview

Basically, if a device needs only be IPS Node, the firmware should be compiled with this role only, and inversely for the Router role.

The Applications which will control the roles on end-products are responsible for the connection between the devices, using suggested advertising intervals and data, connection intervals, security levels, etc. The Profile implementation allows modification of the the behavior depending on the final needs. The Profile role should be enabled immediately after connection creation in order to allow correct profile behavior with the peer device.

1.4 Profile / Application Partitioning.

The IPS profile does not explicitly consider and the LE ACL link establishment. These can be done using generic procedures of the GAP and GATT [4][5]. Nor is the transfer of User Data (IPv6 packets) explicitly described in this document, as this will be performed using the interface to the L2CAP[6].

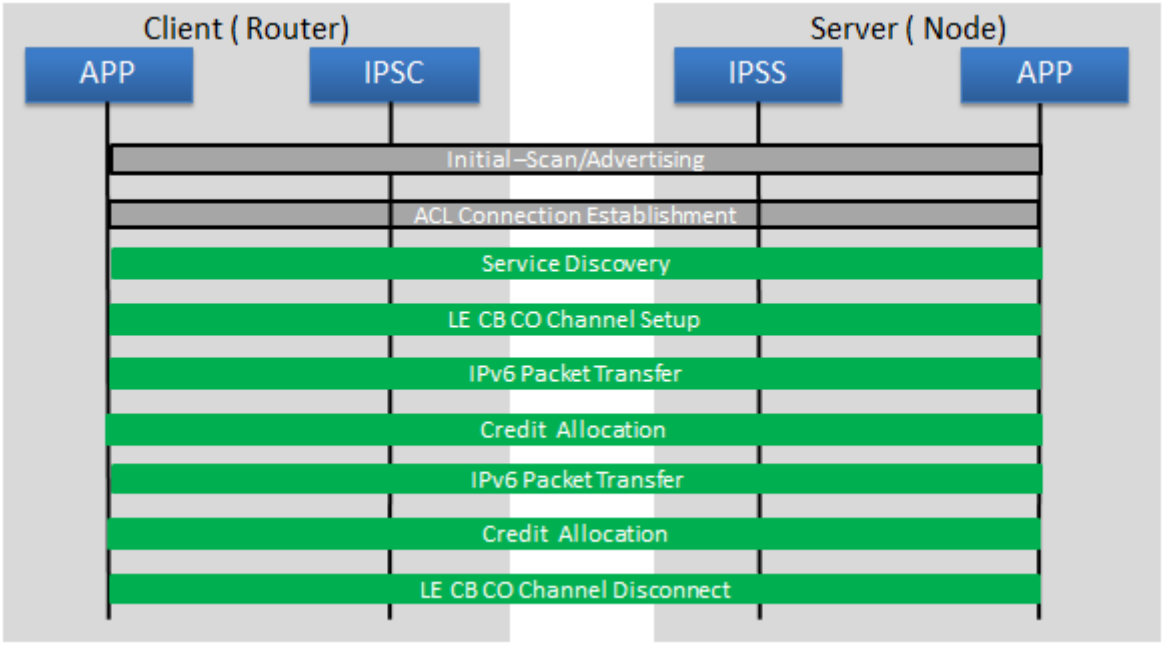

Figure 1 Division of responsibility between Application and Profile

The scope of the IPS is limited to the Establishment and Release of LE Credit Based Connection Oriented channels, and the Data Transfer and flow control on this channel. The IPS also provides a very simple discovery on the client (IPSC) to determine if the peer supports the IPS service. The initial ACL link establishment is common to all profile implementations and thus not considered explicitly to be part of the ISP.

Action	Application	IPS profile
Scanning on the Client	uses GAP API	N/A – Outside Scope of profile
Advertising on the Server	uses GAP API	N/A – Outside Scope of profile
ACL Connection Establishment	uses GAP API	N/A – Outside Scope of profile
Service Discovery	uses IPSC - API	Interacts with GATT to discover IPS service on peer device.
LE CB Connection Oriented Channel Establishment	uses IPS – API	Interacts with L2CAP to establish the L2CAP Channel
IPv6 Packet Transfer	uses IPS - API	Interacts with L2CAP for the Tx/Rx of IPv6 packets
Credit Allocation	uses IPS – API	Interacts with L2CAP to perform Credit allocation.
LE CB Connection Oriented Channel Release	uses IPS – API	Interacts with L2CAP to perform Channel disconnection.

Table 1 Division of Functionality between Application and IPS Profile Layer

2 Internet Protocol Support Server (Node)

This role is meant to be activated on the device that sources or sinks IPv6 packets and sends/receives IPv6 packets to/from a Router. The 'Node' shall not perform any routing or forwarding of packets. It implies it is a GAP Peripheral. The FW task for this role will accept an incoming L2CAP Credit Based Connection Oriented Channel which was initiated by the router. LE Credit based flow control is negotiated on this channel – but it is the responsibility of the application to perform the flow control. It is the responsibility of the LE stack to ensure that the flow control is not violated – it does not perform the flow control itself. Please refer to "ipss_task.h" for implementation of this API.

This task has multiple states, with the two main ones being IPSS_IDLE and IPSS_ACTIVE which reflect the state of an LE Credit Based Connection Oriented channel.

2.1 Internet Protocol Support Node Requirements

The Internet Protocol Support Node is quite unique as it does not have any characteristics or descriptors. Only a single service definition is provided. The IPS Service Definition has a UUID = 0x1820.

2.2 Initialization / Database Creation

During the initialization phase of the device, to use the IPS node task, the IPSS task has to be allocated and corresponding attribute database initialized, using GAPM API. The application has to send GAPM_PROFILE_TASK_ADD_CMD [4] with specific device required security level. The Database for the IPS consists of only a single Service Definition, no additional attributes are part of the IPS service.

2.3 API Messages

2.3.1 IPSS_CONNECT_REQ_IND

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint16_t	peer_max_sdu	Peer maximum SDU size

Response: IPSS_CONNECT_CFM :- To accept or reject the connection

Description: This message is triggered when we have received an incoming LE Credit Based Channel Request from the peer. The **max_sdu** is the maximum SDU size the peer can handle on its receive.

.

2.3.2 IPSS_CONNECT_CFM

Destination: TASK_IPSS

Parameters:

Type	Parameters	Description
Uint8_t	accept	Accept = 0x01 Indicates the connection should be accepted. Any other value will result in the connection being rejected.
Uint16_t	local_max_sdu	The Maximum size of SDU which the local device can receive.
Uint16_t	local_init_credits	The number of Max_SDU buffers which the local device can receive.

Response: None

Description: This message is used by the Application to accept or reject an incoming connection.

2.3.3 IPSS_CONNECT_IND

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint16_t	peer_max_sdu	The maximum SDU size which the peer on the link can receive.

Response: None

Description: This message is delivered to the application when an incoming LE Credit Based connection has completed establishment.

The **peer_max_sdu** is the maximum SDU size the peer can handle on its receive.

2.3.4 IPSS_SEND_DATA_CMD

Destination: TASK_IPSS

Parameters:

Type	Parameters	Description
uint8_t	operation	Should not be used – this will be assigned by the IPS profile
uint16_t	offset	Should not be used – this will be assigned by the IPS profile
struct ips_sdu	sdu	This structure describes the SDU to be transferred to the peer device

Format of the struct ips_sdu

Type	Parameters	Description
uint16_t	cid	Should not be used – this will be assigned by the IPS profile
uint16_t	credit	Should not be used – this will be assigned by the IPS profile
uint16_t	length	The length of the data field
uint8_t	data[..]	The data field of the SDU.

Response: IPSS_SEND_DATA_CMP_EVT is used indicate the data has been sent to the peer.

Description: This message is used by the Application to transfer user data to the peer device. While sufficient credits are available this command will transfer user data to the peer. When credits are not available, upto one user data pdu will be buffered in the L2CAP layer, and will be transferred to the peer when credits become available, transparent to the user. However, it the user tries to transfer more than one user data pdu when there are insufficient credits – the second transfer will be rejected with “L2C_ERR_INSUFF_CREDITS” in the IPSS_CMP_EVT.

Both the message and struct ips_sdu contains fields which are not used on the API. The values of these fields/parameters are used by the underlying IPS and L2CAP layers.

2.3.5 IPSS_DATA_IND

Destination: TASK_APP

Parameters:

Type	Parameters	Description
struct ips_sdu	sdu	This structure contains the SDU received from the peer device. Only the length and data fields are relevant to the application – and values of other fields should be ignored.

Response: none

Description: Event triggered when the local device receives data from L2CAP.

2.3.6 IPSS_RELEASE_BUFF_CMD

Destination: TASK_IPSS

Parameters:

Type	Parameters	Description
uint16_t	**buffer_size**	Total buffer size to release.

Response: IPSS_RELEASE_BUFF_CMP_EVT is used indicate the credits associated have been released

Description: This message is used by the Application to release the buffers which have been consumed by preceding data reception (previous IPSS_DATA_IND messages). The **buffer_size** of the size total size (in bytes) which has been consumed by pre-ceeding received SDUs.

NOTE :- The IPSS_RELEASE_BUFF_CMD command is only responsible for releaseing the Credits associated with a number of SDUs. It does not free the memory allocated for the SDU. This should be done seperatly by the application.

2.3.7 IPSS_DISCONNECT_CMD

Destination: TASK_IPSC

Parameters: None

Response:

IPSS_DISCONNECT_IND :- Triggered when connection is disconnected

IPSS_CMP_EVT :- When operation is complete.

Description: Used by the Application to Disconnect an L2CAP Credit Based Connection Oriented link.

2.3.8 IPSS_DISCONNECT_IND

Destination: TASK_APP

Parameters:

Type	Parameters	Description
Uint16_t	Reason	The disconnect Reason

Response: none

Description: Indicates that an LE Credit Based Connection Orientated link has been disconnected. This can be due to either a local or peer initiated disconnect of the LE Credit Based Connection Oriented link

2.3.9 IPSS_CMP_EVT

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	operation	Indicates the command for which this is the complete event
uint8_t	Status	Status code (see [3])

Response: none

Description: This indicates the success or failure of a preceding Command. Any non-zero value in the status field indicates failure.

The operation is one of the following :-

IPSS_DISCONNECT_CMD_OP_CODE (0x01)
This event indicates to the Application the completion of the locally initiated Disconnection

IPSS_SEND_DATA_CMD_OP_CODE (0x02)
Indicates that success or failure of a preceding IPSS_SEND_DATA_CMD

IPSS_RELEASE_BUFF_CMD_OP_CODE (0x03)
This indicates the success or failure of a preceding IPSS_RELEASE_BUFF_CMD

3 Internet Protocol Support – Client

The IPS Client normally acts as a Router communicating IPv6 packets to/from a Node.

This task has number of states, the main states are IPSC_ACTIVE and IPSC_IDLE – which indicate the current state of an LE Credit Based Connection Oriented channel. In addition there are a number of intermediate states which are entered when waiting for a message from the Application or the underlying GAP layer.

3.1 Initialization

During the initialization phase of the device, to use the IPSC task, the IPSC task has to be allocated using GAPM API. Application has to send GAPM_PROFILE_TASK_ADD_CMD [4].

3.1.1 IPSC_ENABLE_REQ

Destination: TASK_IPSC

Parameters: none

Response: ISPC_ENABLE_RSP

Description: Request if the peer on the link supports the IPS service. This triggers the GATT Service Discovery procedure.

3.1.2 IPSC_ENABLE_RSP

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Indicates success/failure of the search for the service on the peer device

Response: none

Description: Initiates if the peer device supports the IPS service. Any none zero value of status indicates failure.

3.1.3 IPSC_CONNECT_CMD

Destination: TASK_IPSC

Parameters:

Type	Parameters	Description
uint16_t	local_max_sdu	The Maximum size of SDU which the local device can receive.
uint16_t	local_init_credit	The number of max_local_sdu buffers that local device can receive.

Response: ISPC_CONNECT_IND and/or ISPC_CONNECT_CMP_EVT

Description: Initiates an LE Credit Based Connection Oriented Connection on an already established ACL.

3.1.4 IPSC_CONNECT_IND

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint16_t	peer_max_sdu	The maximum SDU size which the peer on the link can receive.

Response: None

Description: This message is delivered to the application when an outgoing LE Credit Based connection has completed establishment.

3.1.5 IPSC_SEND_DATA_CMD

Destination: TASK_IPSC

Parameters:

Type	Parameters	Description
uint8_t	operation	Should not be used – this will be assigned by the IPS profile
uint16_t	offset	Should not be used – this will be assigned by the IPS profile
struct ips_sdu	sdu	This structure describes the SDU to be transferred to the peer device

Format of the struct ips_sdu

Type	Parameters	Description
uint16_t	cid	Should not be used – this will be assigned by the IPS profile
uint16_t	credit	Should not be used – this will be assigned by the IPS profile
uint16_t	length	The length of the data field
uint8_t	data[..]	The data field of the SDU.

Response: IPSC_CMP_EVT (operation = IPSC_SEND_DATA_CMD_OP_CODE) is used indicate the data has been sent to the peer.

Description: This message is used by the Application transfer user data to the peer device. While sufficient credits are available this command will transfer user data to the peer. When credits are not available upto one user data pdu will be buffered in the IPS layer, and will be transferred to the peer when credits become available, transparent to the user. However, it the user tries to transfer more than one user data pdu when there are insufficient credits – the second transfer will be rejected with "L2C_ERR_INSUFF_CREDITS" in the IPSC_CMP_EVT.

3.1.6 IPSC_DATA_IND

Destination: TASK_APP

Parameters:

Type	Parameters	Description
struct ips_sdu	Sdu	This structure contains the SDU received from the peer device. Only the length and data fields are relevant to the application – and values of other fields should be ignored.

Response: none

Description: Event triggered when the local device receives data from L2CAP. Only the length and data fields are relevant to the application – the other fields in the SDU structure should not be interpreted by the application.

3.1.7 IPSC_RELEASE_BUFF_CMD

Destination: TASK_IPSC

Parameters:

Type	Parameters	Description
uint16_t	**buffer_size**	Total buffer size to release.

Response: IPSC_CMP_EVT (operation = IPSC_RELEASE_BUFF_CMD_OP_CODE) is used indicate the credits associated with the buffer have been released.

Description: This message is used by the Application to release the buffers which have been consumed by preceding data reception (previous IPSC_DATA_IND messages). The **buffer_size** of the size total size (in bytes) which has been consumed by pre-ceeding received SDUs.

NOTE :- The IPSC_RELEASE_BUF command is only responsible for releaseing the Credits associated with a number of SDUs. It does not free the memory allocated for the SDU. This should be done seperatly by the application.

3.1.8 IPSC_DISCONNECT

Destination: TASK_IPSC

Parameters: none

Response:

IPSC_DISCONNECT_IND :- Triggered when connection is disconnected

IPSC_CMP_EVT (operation = IPSC_DISCONNECT_CMD_OP_CODE) :- When operation is complete.

Description: Used by the Application to Disconnect an L2CAP Credit Based Connection Oriented link.

3.1.9 IPSC_DISCONNECT_IND

Destination: TASK_APP

Parameters:

Type	Parameters	Description
Uint16_t	reason	The disconnect Reason

Response: none

Description: Indicates that an LE Credit Based Connection Orientated link has been disconnected. This can be due to either a local or peer initiated disconnect of the LE Credit Based Connection Oriented link

3.1.10 IPSC_CMP_EVT

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	operation	Indicates the command for which this is the complete event
uint8_t	status	Status code (see [3])

Response: none

Description: This indicates the success or failure of a preceding Command. Any non-zero value in the status field indicates failure.

The operation is one of the following :-

IPSC_DISCONNECT_CMD_OP_CODE (0x01)
This event indicates to the Application the completion of the locally initiated Disconnection

IPSC_SEND_DATA_CMD_OP_CODE (0x02)
Indicates that success or failure of a preceding IPSS_SEND_DATA_CMD

IPSC_RELEASE_BUFF_CMD_OP_CODE (0x03)
This indicates the success or failure of a preceding IPSS_RELEASE_BUFF_CMD

IPSC_CONNECT_CMD_OP_CODE (0x04)
This message is delivered to the application when the outgoing LE Credit Based connection has completed establishment.

4 IPS connection establishment and data Transfer

During connection establishment each side allocates a number of credits which allow the underlying layers to perform flow control on the connection. From the IPS Profile perspective, credits are initially allocated as an integral number of the Maximum SDU which can be received on a link. The IPS profile layer transparently maps these to an integral number of L2CAP Frames, which is used for L2CAP Flow Control by the L2CAP layer.

This is shown below in figure 2. If the Router Application has a max receive SDU size of 0x500 bytes, and has buffering to support 6 of these SDUs – then it sends an IPSC_CONNECT_CMD (local_max_sdu = 0x500, local_init_credit = 6). Similarly if the Application in the Node can support a max receive SDU size of 0x800 bytes and has buffering to support 4 of these SDUs – this it send an IPSS_CONNECT_CFM (local_max_sdu = 0x800, local_init_credit = 4).

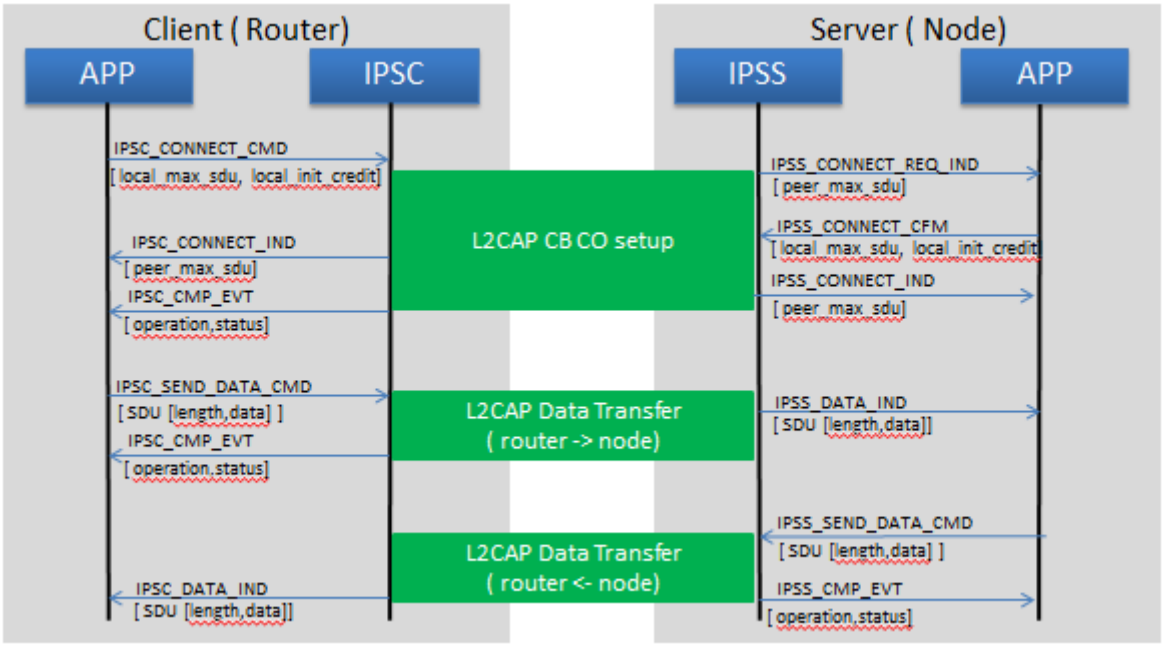

Figure 2 – IPS connection establishment and data exchange.

5 Flow Control and Credit Allocation

As SDUs are received from the peer device and consumed, the user of the API should release these buffers when finished using them. The user of the API uses the IPSx_RELEASE_BUF_CMD to release the buffers linked to previous received SDUs. The command indicates the size of the data contained in the previously received buffer which should be released . This results in a number of L2CAP Frame credits being released by the L2CAP layer**.**

The initial IPSC_CONNECT_CMD and IPSS_CONNECT_CFM contain the initial local credits (in terms of N * Max_SDU) for the local receive. Subsequently the IPSC_RELEASE_BUF & IPSS_RELEASE_BUF free up local receive buffers and allow the underlying layers allocate more L2CAP credits for the link. The 'buffer size' carried in the two RELEASE_BUF messages is the total size (in bytes) of the preceding received SDUs.

Figure 3 shows the message exchange for data transfer and buffer release. The following steps occure

1. The router transfers SDU1 to the node
2. The node transfers SDU2 to the router
3. The router transfers SDU3 to the node
4. The node releases the credits for SDU1 and SDU3 (buffer_size = length of SDU1 + length of SDU3)
5. The router releases the credits for SDU2 (buffer_size = length of SDU2)

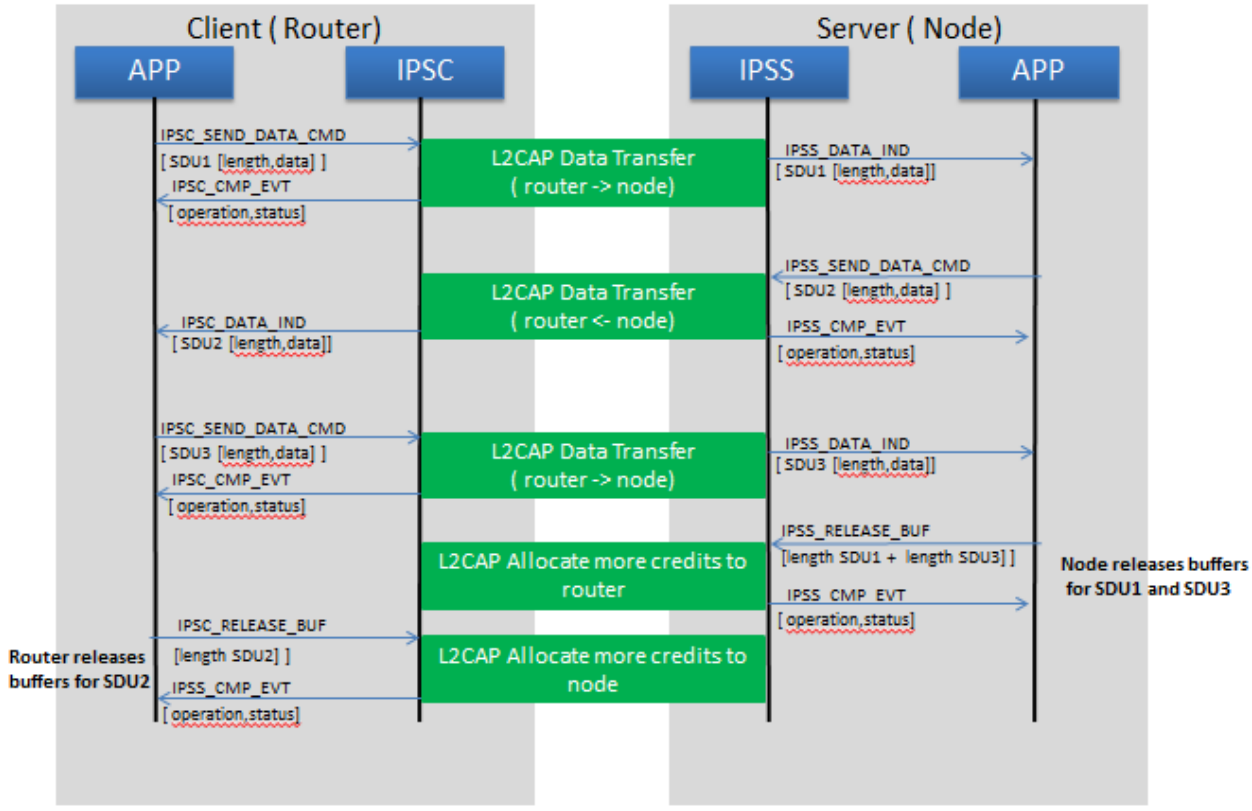

Figure 3:- IPS data transfer and credit management.

NOTE :- The IPSx_RELEASE_BUF command is only responsible for releaseing the Credits associated with a number of SDUs. It does not free the memory allocated for the SDU. This should be done seperatly by the application.

6 Abbreviations

Abbreviation	Original Terminology
API	Application Programming Interface
BLE	Bluetooth Low Energy
IPS	Internet Protocol Support
IPSC	Internet Protocol Support Client
IPSS	Internet Protocol Support Server
GAP	Generic Access Profile
GATT	Generic Attribute Profile
RW	RivieraWaves

References

[1]	**Title**	Internet Protocol Support Profile		
	Reference	IPSP_SPEC_V10		
	Version	V10r00	**Date**	December 16th 2014
	Source	Bluetooth SIG – Internet Working Group		

[2]	**Title**	Internet Profile Support Profile		
	Reference	IPSP.TS.1.0.0		
	Version	1.0.0	**Date**	December 23rd 2014
	Source	Bluetooth SIG – Internet Working Group		

[3]	**Title**	RW BLE Host Error Code Interface Specification		
	Reference	RW-BLE-HOST-ERR-CODE-IS		
	Version	7.00	**Date**	2014-06-30
	Source	RivieraWaves SAS		

[4]	**Title**	GAP Interface Specification		
	Reference	RW-BLE-GAP-IS		
	Version	7.00	**Date**	2014-06-30
	Source	RivieraWaves SAS		