Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	**802.16g_D6 GPCS_PROTOCOL_TYPE encoding Issues**	
Date Submitted	**2008-01-16**	
Source(s)	David Johnston NextWave	Voice: 503 629 0238 Fax: david.johnston@ieee.org
Re:	802.16g_D6	
Abstract	The GPCS_PROTOCOL_TYPE encoding in 802.16g_D6 is flawed since it inhibits upper layer protocol multiplexing and is incompatible with conventional 802 architectural models. This document explores the protocol layering cases that expose the problem and proposes a new definition that can satisfy both the purpose of the current encoding and the more flexible multiplexing model that IEEE Std. 802-2001 demands.	
Purpose	Consider and adopt the proposed changes to 802.16g_D6 as the resolution of David Johnston's sponsor ballot comment against 11.13.19.5.1	
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.	

802.16g_D6 GPCS_PROTOCOL_TYPE encoding Issues

David Johnston
NextWave Broadband

Problems with the use of Ethertype

11.13.19.5.1 Defines the GPCS_PROTOCOL_TYPE TLV. It is included in the DSx-REQ, DSx-RSP messages to indicate the protocol that is carried over the GPCS terminated connection that is being set up. The value is in the form of a 2 byte Ethertype.

The text specified that for a GPCS connection setup “this TLV shall be used to indicate the protocol carried over the connection” and also specifies that other CS types do not use it.

This means that when a GPCS connection is being set up, it is mandatory that one and only one Ethertype be specified as the protocol type carried over GPCS. This is contrary to the intent described in 5.3 that GPCS can multiplex multiple protocols over one connection.

Since the Ethertype value encodes for a protocol that the LLC itself demultiplexes, the Ethertype value does not have an encoding for a protocol multiplexor. So it cannot support the ‘multi-protocol’ encoding that the specification calls for.

The Ethertype also cannot encode for PPP, which is desirable to some service providers.

In the 802 architecture, the Ethertype is conventionally encoded in the SNAP field that follows an LLC PDU header or directly in the Ethertype field of a medium that has one. For example 802.5 and 11 use the LLC encoding, 802.3 directly encodes the Ethertype at the expense of a minor layer violation and the additional flexibility afforded by the 802.2 LLC to support proprietary protocol code spaces.

The LLC PDU header/SNAP encoding is described in Std 802-2001, clauses 10.3 (SNAP) and 9.3 (Protocol Identifier).

The population of fields within the LLC PDU header is described in ANSI/IEEE Std 802.2, 1998 edition, clause 3.

The 802.3 encoding of the Ethertype is described in Std 802.3-2005, clause 3.2.6.

When Ethernet frames are sent, existing practice defined by 802 and the IETF determines whether the Ethernet ethertype field is used directly or the 802.2 LLC PDU format.

Problems with the GPCS architectural model

The GPCS specification does not clarify the relationship between GPCS service points and SSs. It may be that there is one GPCS for each SS. It could be that the BS has only one GPCS instance serving all SSs or it could be something in between. The specification does not explain which it is.

Problems with the GPCS service

The GPCS service primitive includes the packet (length & data) and MS MAC Address and an SFID:

```
GPCS_DATA.indication
    (
        SFID,
        MS MAC Address,
        length,
        data
    )
```

This allows an implementation to identify the destination MS of outgoing packets and the source of incoming packets, however it does not match the service that currently defined upper layers are expecting to see. For instance and IP stack is unaware of SFIDs. An 802.1D bridge port expects to see the standard 802 ISS primitives:

```
M_UNITDATA.indication
    (
        frame_type,
        destination_address,
        source_address,
        mac_service_data_unit,
        user_ priority,
        frame_check_sequence
    )
```

The mapping is not described anywhere. Bolting an ISS onto a GPCS SAP cannot be done without proprietary mapping functionality that cannot be interoperable.

The mapping between the ISS and the GPCS service must be defined.

Similarly, the mapping between the IP packets from an IP stack and the SFID and MS_MAC Address parameters must be defined.

Proposal

It is proposed to change the definition of the GPCS_PROTOCOL_TYPE to encode for the upper layer type instead of an Ethertype.

GPCS_PROTOCOL_TYPE remains as a 16 bit value but the value is defined from the following list:

GPCS_PROTOCOL_TYPE	Layer Above GPCS	Notes
0x0000	Ethernet MAC Service	An upper layer that sinks and sources Ethernet formatted frames consistent with those used in the Ethernet CS.
0x0001	MPLS	Raw MPLS packets with MPLS label and payload
0x0002	PPP	The Point to Point Protocol
0x0003	Raw IP	Raw IP packets. This is necessarily a point to point IP link since ARP cannot be supported. Note that the first byte of every IP packet allows the distinction between IPv4, IPv6 and ROHC IP packets so these protocols may be multiplexed over the same GPCS connection.
0x0004-0x7fff	*Reserved*	Reserved for future additional encodings. These encodings shall not be used.
0x8000-0xffff	*Reserved Playpen*	These encodings shall not be used in deployed equipment and are reserved for experimental use.

This list has been defined here since there appeared to be no suitable encoding defined elsewhere.

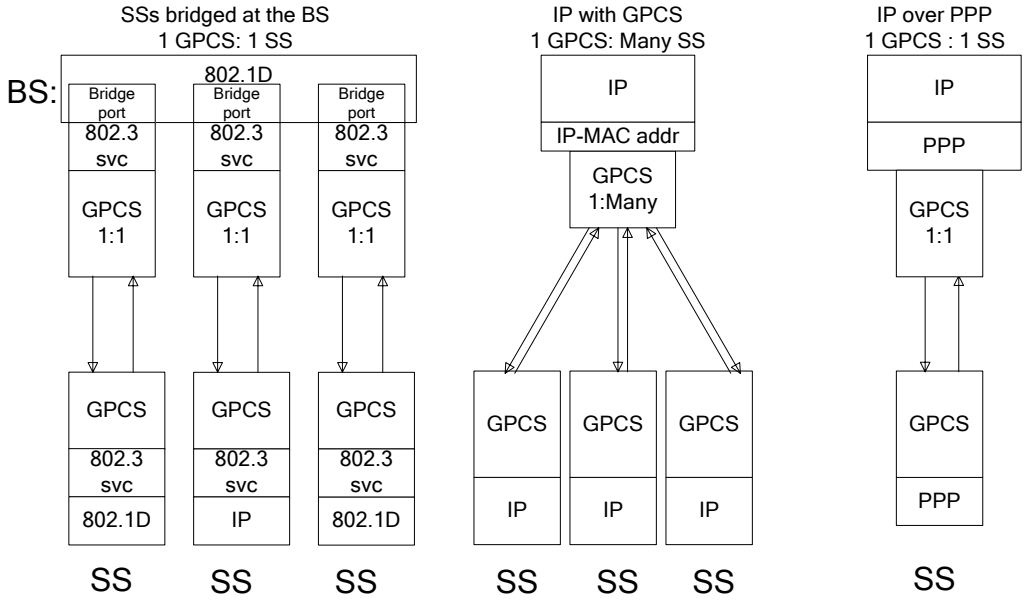
Support for 802.2 style Ethertype based protocol multiplexting would be achived by using the 0x0001 encoding for Ethernet frames with LCC. Payloads would be formatted as an LLC PDU, consistent with 802.5 and 802.11.

Ethernet style protocol multiplexing (DIX) would be achieved using the 0x000 encoding. This would be consistent with what it used in the 802.3 specific part of the Packet CS.

An 802.1D bridge directly above the GPCS, expecting an ISS compliant service, must use

The relationship between GPCS instances and SSs needs to be clarified. To support GPCS-as-protocol-multiplexer style operation, 1 GPCS at the BS : many SSs should be allowed. To allow multiple point to point links with a bridge making the routing decisions, a 1:1 mapping should be allowed. It would be permitted for a BS to simultaneously provide a mix of such models, with SSs mapped to different GPCS instances either individually or in groups.

E.G.:



In the case of a GPCS instance at the BS peered with a single SS, the MS MAC Address primitive parameter on the GPCS_DATA primitive would be constant, set to the address of the peer MS.

The proposed text changes come in two parts, 1 and 2. Proposed text changes 2 addresses a proposal by Max Riegel to permit GPCS at the BS to interoperate with SSs that do not support GPCS.

Proposed Text Changes 1

Change bullet item 6 in 5.3 as follows

- GPCS allows multiplexing of multiple layer protocol types (e.g., IPv4, IPv6, Ethernet) over the same 802.16 connection. An appropriate ~~protocol type value~~ upper protocol layer that supports protocol multiplexing is used to ~~represent multiprotocol~~ do this , and it is ~~used~~ signaled in the GPCS_PROTOCOL_TYPE ~~protocol type~~ TLV in DSx messages to indicate that multiple protocols are supported for a connection/service flow. It is ~~beyond~~ outside the scope of the GPCS to specify how ~~to~~ the upper layer multiplexes and demultiplexes multiple protocol data packets over an 802.16 connection/service flow.

Renumber existing section 5.3.1 to be 5.3.2 and insert new section 5.3.1 after figure 17c in 5.3

5.3.1 Mapping of the GPCS service to upper layers

Multiple instances of the GPCS service may exist in the BS, each servicing 1 or more SSs.

In the case where the upper layer demands a point to point GPCS link (E.G. where 802.3 packets are carried over GPCS and are bridged using 802.1D), a GPCS instance at the BS must peer with only a single GPCS instance on a single SS. In this case the *MS MAC Address* field of the GPCS_DATA primitive shall be constant and set to the MAC address of the SS. The *SFID* field of the GPCS_DATA primitive shall be constant and set to the SFID of the service flow being carried.

In the case where a single GPCS instance services more than one SS, the *MS MAC Address* field of the GPCS_DATA primitive will indicate the SS that is the source or destination for the PDU. The *SFID* field of the GPCS_DATA primitive shall be set to the SFID of the service flow being carried.

Replace 11.13.19.5 and 11.13.19.5.1 with the following text

11.13.19.5 GPCS CS encodings for configuration and MAC messaging

11.13.19.5.1 GPCS PROTOCOL_TYPE encoding

The GPCS_PROTOCOL_TYPE TLV indicates the type of protocol layer that sits above the GPCS service. This allows the remote end to be able to correctly parse the payload contents of GPCS PDUs.

Type	Length	Value	Scope

[145/146].cst.7.1	2	Two byte upper layer protocol number	DSx-REQ, DSx-RSP

The value field is 16 bits and its encoding is defined in table [xxx].

GPCS_PROTOCOL_TYPE	Layer Above GPCS	Notes
0x0000	Ethernet MAC Service	An upper layer that sinks and sources Ethernet formatted frames consistent with those used in the Ethernet CS.
0x0001	MPLS	Raw MPLS packets with MPLS label and payload
0x0002	PPP	The Point to Point Protocol
0x0003	Raw IP	Raw IP packets. This is necessarily a point to point IP link since ARP cannot be supported. Note that the first byte of every IP packet allows the distinction between IPv4, IPv6 and ROHC IP packets so these protocols may be multiplexed over the same GPCS connection.
0x0004-0xEFFF	*Reserved*	Reserved for future additional encodings. These encoding shall not be used.
0xF000-0xFFFF	*Reserved Playpen*	These encodings shall not be used in deployed equipment and are reserved for experimental use.

Table [XXX] GPCS_PROTOCOL_TYPE Encoding

For a connection using Generic Packet CS, this TLV shall be used to indicate the protocol carried over the connection. For other packet CS types, GPCS_PROTOCOL_TYPE is not used.

Proposed Text Changes 2

Renumber 5.3.2 to be 5.3.3 and insert new section 5.3.2 with the following text

5.3.1 Operation of GPCS with SSs that do not support the GPCS

A BS that supports GPCS may interoperate with an SS that does not support GPCS. It can be observed that a GPCS service with GPCS_PROTOCOL_TYPE 0x0000 (Ethernet MAC Service) carries packets formatted identically to the packets used in the 802.3 specific part of the Packet CS. Also, a GPCS service with GPCS_PROTOCOL_TYPE 0x0003 (Raw IP) carries packets formatted identically to the packets used in the IP specific part of the Packet CS.

A BS may operate using GPCS at the base station, locally using GPCS_PROTOCOL_TYPE=0x0000, while signaling to the SS during connection setup that the 802.3 specific part of the packet CS is being used.

A BS may operate using GPCS at the base station, locally using GPCS_PROTOCOL_TYPE=0x0003, while signaling to the SS during connection setup that the IP specific part of the packet CS is being used.