IEEE P802.15
Wireless Personal Area Networks

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)
Title	**TG3 Rolling Meadows Ad-Hoc Working Document**
Date Submitted	[1 December, 2001]
Source	[James P. K. Gilb] [Mobilian] [12707 High Bluff Dr., Suite 335, San Diego, CA 92130] Voice: [858-436-2201] Fax: [858-436-2301] E-mail: [gilb@ieee.org]
Re:	[]
Abstract	[This document is contains the issues and resolutions developed by the TG3 subcommittee at the Rolling Meadows ad-hoc meeting. It also serves to document the addtions made to the draft standard D07 to produce D08.]
Purpose	[To provide a record of the resolutions of TG3 and the changes to the draft standard D07 to make D08.]
Notice	This document has been prepared to assist the IEEE P802.15. 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 acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15.

1. Goals

1) Power Management Proposal and suggested wording
   Proposal by Jay Bain, 01/430r0, 01/429r1 - **Unused CTA bits - accept, need text. Null CTAs, accept in principle, but waiting for the details.**
2) Geographic coordinator selection and daughter network
   proposal by B. Huang, 01/304r3, 01/305r3 - **Cleanup and modify GIP to do reconfiguration, due at Austin. Child network only requires 1 extra information element, agreed that this is a good idea.**
3) Resolve QoS Policy and QoS MLME-primitives
   proposal by A. Heberling, 01/469r0, 01/470r0 - **Most of the text is complete, needs some work. Agree in principle to multiple possible convergence layers, the concept of a service flow and the commands at the MLME-SAP level.**
4) Security Proposal recommendation and wording
   Propsal by G. Rasor (SEC, Auth, Assoc), 01/423r0 - **Still waiting for proposal.**
5) Management control outside of the CAP
   Proposal by W. Shvodian, 01/xxxr0- **Direct approved, waiting text**
6) 11 Mb/s QPSK-TCM mode
   Proposal by J. Karaoguz, 01/448r0 - **DONE**
7) The rest of the prioritized issues list. (perform triage prioritization on issues and resolve the most important ones first.)

2. Agenda

Tuesday October 9th:

8:00 am	Call meeting to order
8:01 am	Approve/modify agenda
8:05 am	Begin work on goals* in the order listed
10:00 am	Recess for break
10:20 am	Meeting called to order
10:21 am	Continue work
12:00 pm	Recess for lunch
1:00 pm	Meeting called to order
1:01 pm	Continue work
3:00 pm	Recess for break
3:20 pm	Meeting called to order
3:21 pm	Continue work
5:30 pm	Recess for dinner

Wednesday, October 10th:

8:00 am	Meeting called to order
8:01 am	Continue work on goals
10:00 am	Recess for break
10:20 am	Meeting called to order
10:21 am	Continue work
12:00 pm	Recess for lunch
1:00 pm	Meeting called to order
1:01 pm	Continue work
3:00 pm	Recess for break
3:20 pm	Meeting called to order
3:21 pm	Continue work
6:00 pm	Recess for dinner

Thursday, October 11th:

8:00 am	Meeting called to order
8:01 am	Continue work on goals
10:00 am	Recess for break
10:20 am	Meeting called to order
10:21 am	Continue work
12:00 pm	Working lunch
1:00 pm	Adjourn

Items to communicate:

1) Describe Protocol Implementation Conformance Spec. (PICS) to authors
2) SDL Scope and Effort

3. Items to resolve

3.1 Power managment/Wake on WPAN

Consensus is that we already have the mechanisms that allow the higher layers to perform Wake on WPAN. Suggestion is to provide text in an informative annex that describes how this feature can be used.

3.2 Addtions to 7.5.11 EPS Configuration request

Suggest following definitions

EPSTime has a range of 0 to 65,535 ms. A value of zero has the interpretation of DEVs waking for each superframe. Depending on the value of superframe duration parameter, values of EPSTime that are a value that is less than the current value of superframe duration are interpreted as wake for each superframe. Since wake time is bounded by superframe beacon location, the beacon start point preceding the completion of EPSTime shall be the wake point.

EPSPhase is a value of beacon number as defined in Piconet synchronization parameters element. The value is determined by the EPSSync parameter in the MLME-POWERMGT.request primitive. The current beacon number when that primitive is received by the SME is used for EPSPhase in building the EPS configuration request command.

EPSTime is OK, EPSPhase definition needs some work, perhaps a new name, the EPSBarr. EPSSync needs a definitions and rename it to EPSBain.

EPSBarr is a value of beacon number as defined in xref. The value may be determined by higher layers, but this is not specified in the standard.

3.3 MAX CTAs elements

The text says “maximum number of GTS slots that may be assigned to a DEV, where the DEV is either the source or destination, including group or multicast destinations.” However the elements does not have an AD-AD or device ID, so where is the DEV defined? Is it implicit, i.e. it is the DEV that is sending the element? If so, then the DEV cannot be a group or multicast destination.

3.4 Coordinator selection, 8.2.3

We need to add text that all ACs that hear the PNC selection frame and are not currently associated with piconet shall participate in the coordinator selection process. Also need to add that ACs that are currently associated with a piconet shall not participate.

3.5 11 Mb/s QPSK-TCM mode

Agree to change BPSK mode to QPSK-TCM. Agree to modify the the current TCM encoder for the extra feedback path.

3.6 Max Burst Size definition

Sub-clause 7.5.20, figure 51. Proposed definition: Max Burst Size is the maximum number of octets that can be sent by a stream in any one superframe.

The max burst size is the maximum number of octets sent at the maximum data rate that can be received by the DEV. (This definition leaves out the duty cycle length which is required).

Upon further review, perhaps this should be deleted. Keep this as an issue. Review all of the parameters, which are really necessary without overly complicating the MAC.

4. Editorial changes

- 7.4.12 MAX CTAs element, reformat to match the previous sub-clauses in 7.4 and move the functional description to clause 8.
- Check the PNC handover process to make sure that we state that the current PNC checks the DEV-info table to find the most qualified AC to become the new PNC.
- 7.3.1 Add xref to the channel change element and the CTA element in table 58 verify the use of the correct terminology. Also, change the Device ID entry to be device identifier element with description “IEEE 802 address of the PNC, xref 7.4.1” Change to piconet synchronization element and xref in notes.
- 7.3.1 Add text that says that the beacon elements can occur in any order and that DEVs shall ingnore information elements other than the ones defined.
- 7.4.6 Add pad byte for word alignment and text for how to handle the pad byte.
- 7.4.3 Fix the TBD in the PHY table mapping by adding the appropriate section to the PHY and the xref to it in 7.4.3.
- 7.4.2 Fix length parameter for piconet sychronization parameter.
- 7.3.3 and 7.3.4, define the SA and DA for the command and data frames.
- 7.5.1, 7.5.2, 7.5.3, Make notes that the PNC selection commands, and the association commands are never sent with other commands. Or perhaps better is to say that a command can only be piggybacked with commands that are sent with the same SA and DA pair.
- 7.2.1.9 Note that the beacon shall not have the SEC bit set (perhaps this is true of the command frames?).
- 7.5.2 ATP definitions disagrees with 8.2.5 definition of ATP. Did we fix this since Schaumburg.
- 7.5.17 (D06) add xref to stream identifier element to the appropriate information element (stream ID)
- Page 97, change aa to a
- 7.5.21 Fix QoS paramters length in table 49, fix table sizing for table 51
- 7.5.21 Need to get a good definition for Max Burst Size
- Figure 27, change length of the individual elements to be 7 instead of 6.

— 8.8 The acronyms TPC and DCS are mentioned, but are not defined. It should be DFS rather than DCS. Perhaps delete this sentence.

5. Changes from conference calls

5.1 September 20, 2001

5.1.1 Usage of RTC, issue 397

Shvodian suggested that RTC in Section 7.4.3, and the acronyms section be deleted because it is not currently used. Gilb agreed that we could delete it. There was consensus agreement to remove it and its references. There was consensus agreement to remove it and its references. We discussed what to do with the bit. Gilb suggested we make it the PS (power save) bit in this capability field. This makes PNC DES mode, and whether you are plugged into the wall, periodically posted to the network. This change showed that PS got dropped my mistake, and confused with a similar term

Action: Gilb will put PS (power saving) into 7.4.3 We also need definition. Shvodian opened issue 397 to "get rid of RTC bit and clean this up " and rename PS to PSAVE for power save.

Gilb suggested we re-look at positioning the bits in the order in which they are evaluated. Gilb will put them in order if it makes a difference. The assumption is that MSB is high bit and sent last. Rank high to low in order of comparison with AC in position 15.

Capabilities Field by bit number:.

15 AC, 14 PNC DES, 13 SEC, 12 PSRC, 11 PSAVE, 10-6 Max PHY rate, 5-0 is reserved.

Ed. Actions:

RTC references deleted in D08. PSAVE and PSRC bits added, the definition of the bits in D08 is:

> The PSAVE bit is set to 1 if if the DEV is planning to use sleep state as a part of power management. Otherwise the PS bit is set to 0. The PNC shall always set this bit to 0 in its capability field.
>
> The PSRC bit shall be set to 1 if the DEV is receiving power from the AC (alternating current) mains and shall be set to 0 otherwise.

The power source field was deleted from table 64, 7.5.1.3 and from figure 21 (now a reserved byte) since it is now PSRC bit in the capabilities field.

The bits in figure 12 have been re-ordered as indicated above. I added a sentence to the beginning of 7.4.3 to explain the order of the bits in the capability field. The sentence that was added was:

> Bits b11-b15 are arranged in order of preference for PNC selection, with the highest preference (AC) corresponding to the MSb

5.2 September 25, 2001

5.2.1 PNC selection

Issue: Table 64 in D07 belongs in clause 8 rather than clause 7

Resolution: Move Table 64 to 8.2.3

Ed. action: Changed as indicated.

5.3 September 27, 2001

5.3.1 PNC-Des mode

Issue 229: Need to clarify the use of the Des-mode bit.

Resolution: (none suggested)

Issue 262: Definition of PNC-DES mode is not clear,

Resolution: Should say: The PNC-Des-Mode is the designated mode of the DEV as currently set. The PNC-Des-Mode bit is set to one if the DEV is designated by a user to be PNC for the piconet. Otherwise the bit shall be zero. The PNC-Des-Mode bit shall always be set to zero if the DEV is not PNC capable.

Issue 331: If the PNC-DesMode bit is set…

Resolution: Please clarify the use of the PNC-DesMode bit.

Ed. action: Based on email discussion, we adopted the following text:

If more than one AC participates in the PNC selection process, then the AC receiving an alternate PNC selection command from another AC shall compare the received PNC selection parameters with its own using the evaluation criteria defined in Table 64. If the AC finds that its parameters score higher than the received ones, the AC shall continue to broadcast the PNC-selection frame. If the AC finds that its parameters score lower than the ones received, the AC shall no longer send the alternate PNC selection command and wait for the piconet to start within the last received indicated timeout. Thus the DEV with the highest capabilities will be chosen as the PNC. The MAC address acts as a tie breaker for DEVs with identical capabilities, i.e. the highest MAC address will be chosen as the PNC.

As Table 64 shows, PNC Des-Mode is the top priority field in the PNC selection criteria. Since the PNC Des-Mode is the highest priority, a DEV with this bit set is more likely to become the PNC of the piconet. Thus, this bit should be set if it is desirable for the DEV to be the PNC of the piconet. If only one DEV has PNC Des-Mode set, that DEV should become the PNC. If two or more DEVs that are competing to be PNC all have the PNC Des-Mode bit set, the rest of the capabilities are used to determine which of the DEVs will be the PNC.

5.4 October 2, 2001

No items

5.5 October 4, 2001

Issue 304: The ACK policy field definition is TBD

Resolution: Defined by cross reference to 7.2.1.2 in 7.5.21 in D08

Ed. action: Changed as indicated in D08.

Issue: This paragraph referenced here is dependent upon there being a channel time request rejection indication which does not exist. For instance, there is nothing in the CTA info elements to indicate a failure. And there is nothing in the Channel Time Grant to indicate a failure. The only command that provides the type of indication required here is the stream command when its Action Type field is set to a value of 3. Consequently, it is my opinion that this paragraph is broken and should be deleted from the document.

Resolution: Resolved in conference calls, details in the minutes.

Ed. action: The paragraph in question has been modified according to the minutes. Actual text follows:

The PNC may compute more than one superframe slot allocation at a time and keep them repeating over time until the situation changes. The allocation by the PNC may not exactly match the duration of time requested by a DEV. However, the allocation shall be at least the minimum requested by the DEV in the channel time request command except when the request is rejected. If the PNC allocates less time than requested but more than minimum, it shall allocate more time if it becomes available in the CFP. If the time slot is not allocated via the beacon, the DEV is free to request the time slot again.

Issue 259: The guaranteed start time parameter is redundant since the CFP duration and the superframe duration uniequely define the start of the CFP. One of these needs to be deleted.

Resolution: Guaranteed Start Time Removed

Ed. action: Changed as indicated in D08.

Issue 292: Not clear whether Tx and Rx frame count includes immediate ACKs.

Resolution: Add text “not including Imm-ACK frames” to each of the RX frame count items.

Ed. action: Changed as indicated in D08.

Issue 293: Does the Rx Frame Count include errored frames?

Resolution: Add the word “correctly”

Ed. action: changed as indicated. Paragraph now says:

The RX frames count is the total number of frames, not including Imm-ACK frames, that were correctly received by the sender of this command. Only the directed frames intended for this DEV are included.

Issue 283: Kus is not enough resolution for the Duration between time slots.

Resolution: Change Duration between time slots to be 2 bytes and change figure 25 number 6-8, also delete the pad byte and change length to n*8. In section 7.5.5.1, change the first sentence to "first block " and delete this size. Change Latency to "Stream" twice in the paragraph. On page 85, line 5 Change "kmicroseconds" to 8 us and do the range of requested time is 0 to 524280 us ((2^16-1)*8us).

Ed. action: changed as indicated. New text for duration between time slots is given below:

The duration between transmissions indicates the requested frequency of time slots that needs to be allocated for the DEV. The resolution of this field is 8 ms and so the range of this field is [0-524280] ms.

5.6 October 16, 2001

Issue 102: Power Save Parms are TBD.

Resolution: Not needed, deletd from D08.

Ed. action: Changed as indicated in D08.

Issue 260: When a DEV sends this element in one of its frames, for all the fields, it sends the same...

Resolution: The frame formats only describe the organization of the frame, not their usage. The usage should be described in the MAC functional description. The sentence needs to be deleted anyway since this information element is only sent in the beacon by the PNC.

Ed. action: Deleted in D08.

Issue 313: The priority field in stream ID field contained in channel time requet block…

Resolution: StreamID changed to stream control. Stream Control will be referenced.

Ed. action: Lots of changes in D08, rename Stream ID to stream control which is 2 bytes. Use stream index as the handle for the specific stream, it is 1 byte.

In 7.5.21.1, retransmission request, the Stream ID of 2 bytes was changed to stream index, 1 byte and the change in the length of the individual blocks was updated. Same change for 7.5.21.2 Retransmission sequence resync.

5.7 October 18, 2001

Issue 102: 8 bit pkt window size only allows 256K Octets, which limits rate to 20 Mbps with a 100 ms super-frame.

Resolution: Receive Window Size needs to be increased to 2 Octets.

Ed. action: Changed as indicated in D08.

Change ..

5.8 October 23, 2001

Issue 102: Why is minimum channel time in 32 us increments? Should be the same as requested channel time.

Resolution: Change to 8 us, change to 2 bytes. Remove the pad byte that was added.

Ed. note: Changed as indicated.

5.9 October 25, 2001

Issue 394: Need to clarify how SME knows that the station is PNC when a piconet is started or when PNC handover occurs.

Resolution: Add the following Sentence: ResultCode of SUCCESS indicates that the DEV is PNC. If another DEV is chosen as PNC or a piconet is already established, the result code shall be ALREADY_STARTED_OR_SYNCHed

Ed. action: The following sentence was added: A ResultCode of SUCCESS indicates that the DEV is the PNC. If another DEV is chosen as the PNC or a piconet is already established, the result code shall be ALREADY_STARTED_OR_SYNCHed

Issue 241 and 250: FCS should not be sent if frame body is zero bytes.

Resolution: Add the following sentence: If the frame bodiy has zero length (as in an immediate ACK frame) the FCS shal not be sent. In this case, the length of the FCS field is zero bytes. Change 4 in the FCS octets to 0 or 4.

Ed. action: Added sentence “If the frame body has zero length (as in an immediate ACK frame) the FCS shall not be sent.”

5.10 October 30, 2001

Issue 369: 11 bits gives 0-2047 bytes, not 0-2048

Resolution: Change 11.2.7 to match definition of max frame body size in 7.2, change example in Table 76 to have a proper example.

Ed. action: change 11.2.7 to be:

The maximum frame length allowed, aMaxFrameSize, shall be 2048 octets. This total includes the frame body and FCS but not the PHY preamble, PHY header or MAC header.

The example in the table is:

An 11 bit field that contains the length of the frame body, in octets, MSB is b5, LSB is b15, e.g. 4 octets of data, is encoded as 0b00000000100. A zero length frame body is encoded as 0b00000000000 and there is no FCS for this packet.

Issue 336: … Beacon frame transmission shall use RIFS for their transmission. Huh? Why?

Resolution: Please strike this sentence, it is technically incorrect.

Ed. action: changed the sentence to:

The beacon frame, which is neither in the CAP nor the CFP, does not use interframe spacing.

5.11 November 1, 2001

Issue 386: Incorrect or missing parameters in MLME-START.req table.

Resolution: Please add this error message to the indicated table: No_Channels_Available

Ed action: The error message NO_CHANNELS_AVAILABLE. Description of the error message was added to D09, text follows:

If all of the channels for the PHY are either occupied by other 802.15.3 piconets or have unacceptable interference, then the result code shall be NO_CHANNELS_AVAILABLE.

Issue 351: … at the end of the beacon header… relative to the end of the beacon header. … where it expects the end of the beacon.

Resolution: Please change the indicated fragments to: … at the beginning of the beacon preamble… relative to the beginning of the beacon preamble. … where it expects the beginning of the beacon preamble.

Ed. action: Change end of beacon header to beginning of beacon preamble. Text now reads:

Each DEV in the piconet, including the PNC, shall reset its clock to zero at the beginning of the beacon preamble as shown in Figure 60. All times in the superframe shall be measured relative to the beginning of the beacon preamble. If a

DEV does not hear a beacon, it should reset its clock to zero at the instant where it expects the beginning of the beacon preamble.

Issue 352: Figure 50 shows t=0 at the end of the header. Shouldn't it be at the beginning of the preamble?

Resolution: Please change the diagram to illustrate the correct reference Time-zero.

Ed. action: Moved t=0 time to the beginning of the beacon preamble in the figure.

Issue 353: … the accuracy of the timer to be +/-0.01% of its resolution. Comment: This doesn't seem sufficiently accurate. Particularly, since we are going to be specifying guardTime intervals. Guardtimes are going to be dependent upon the clock drift. Consequently, I see a need for a more precise specification here.

Resolution: Please provide the specification in PPM.

Ed. action: Changed to +/- 25ppm, text now reads:

A compliant implementation shall maintain the accuracy of the timer to be ±25 ppm.

Item 295, Time Unit. This will be replaced with microseconds or milliseconds as Gilb sees fit. He will note all the changes so it is easier to review.

Ed. action: Changed TU for beacon period to ms in 6.3.3.1, 6.3.9.1. Changed TU in AuthenticationFailureTimeout to ms in 6.3.4.1

Issue 246: … The PNID remains constant during the life of the piconet. The question regarding this sentence is: "Does this imply that the PNID is a transient value? Does this mean that when the PNC or the piconet AC go away, the PNID is not persistent?

Resolution: Change the text indicated to: "… The PNID remains constant during the current instantiation of the piconet and may be persistent for multiple sequential instantiations of the same PNC initiated piconet.

Ed. action: Accept text except use “to instantiations of the piconet by the same PNC” in the place of “instantiation” Text now reads:

The PNID is a unique identificaion for the piconet. The PNID remains constant during the current instantiation of the piconet and may be persistent for multiple sequential instantiations of the piconet by the same PNC.

No issue number: Draft uses both octets and bytes to refer to groups of 8 bits

Change byte(s) to octet(s) throughout the draft

Ed. action: All occurances (except one Mbyte reference) have been changes.

6. MAC changes, 01/259r2

Proposal by Bill Shvodian

One item that was suggested in this document didn’t make D07.

6.1 Item 2, Coordinator challenge frame

Draft04, Table 30

Add the following PIB values:

Managed Object	Data Type
MACPIBBattery/AC Power	Binary, 0=Battery, 1=AC
MACPIBMaxAllocatableGTSSlots	Integer

Ed. action: The PIB values have been added to D08, the current table (table 36) now reads:

Table 1—MAC PIB characteristic group parameters

Managed Object	Number of octets	Definition	Type
MACPIBDeviceID	6	Any valid MAC address	Static
MACPIBPowerManagement-Mode			
MACPIBNumMaxStreams			
MACPIBMaxAssignedCTAs	1		
MACPIBMaxProcessedCTAs	1		
MACPIBMaxAllocatableGTS			
MACPIBPowerSource	1	0x00 for mains power 0x01 for battery power	

7. Static GTS, MTS, 01/476r0

Proposal by Bill Shvodian

7.1 Management Time Slots

7.1.1 Clause 7.4.x Where to put CAP restriction bits? Piconet Synchronization IE?

Table or fig:

B0 - Data in the CAP
B1 - Commands (not including Association) in the CAP
B2 - Association Command in the CAP
B3 - EPS in the CAP

Frames of each type are only allowed in the CAP if the corresponding CAP restriction bit is set to one. If the CAP restriction bit is set to zero, those type of frames are only allowed in the CFP. The PNC may choose to limit the types of frames in the CAP so that it can minimize the size of or eliminate the CAP altogether.

Should no CAP be default or even the only mode?

Ed. action: this requires 2 new bytes (1 plus reserved). New figures and supporting text shown below

Octet: 1	**1**	**4**	**2**	**2**	**2**	**1**	**1**
Element ID	Length (=12)	Beacon number	Superframe duration	CFP duration	CAP MaxBurstDuration	CAP mode	Reserved

Figure 1—Piconet synchronization parameters element.

... (descriptions of the other elements)

The CAP mode indicates what type of information is allowed to be sent in the CAP of the current superframe. The encoding of this octet is illustrated in Figure 2. If a bit is set in the CAP mode field, i.e. its value is 1, then that type of data or command is allowed to be sent in the CAP of the current superframe. Otherwise, that type of frame is not allowed to be sent in the CAP. The use of this command is described in 8.3.2

b0	**b1**	**b2**	**b3**	**b4-b7**
Data	Commands (except association)	Association commands	EPS	Reserved

Figure 2—CAP mode field

The paragraph describing the restrictions belongs in the functional description and has been added to 8.3.2 in D08. The added text is shown below:

The PNC controls what type of data or commands may be sent in the CAP via the CAP mode field of the piconet synchronization information element, 7.4.2, in the beacon. A DEV shall only send frames of the type indicated by the CAP mode field in beacon for the current superframe. The CAP mode field may be changed by the PNC from superframe to superframe.

7.1.2 New Clause 8.3.3 Slotted Aloha Access in MTS slots

Management Time Slots (MTSs) are identical to GTSs except that the PNC address (zero) is the source or the destination address in the CTA. An open MTS is an MTS where the source address in the CTA for the MTS is the broadcast address. Any DEV associated to the piconet can attempt to send a command frame to the PNC in an open MTS. An MTS with the association address as the SA in the CTA for the MTS is called an association MTS. Any station not currently associated to the piconet can attempt to send an association command to the PNC in an association MTS. Association commands are not permitted in open MTSs. Likewise, only association commands are allowed in association MTSs.

Slotted Aloha is the access mechanism in an open MTS or an association MTS. The access to an open or association MTS shall be controlled by a contention window CW_a maintained by each DEV. The contention window shall be derived from the number a, where a is the number of retransmission attempts made by the

DEV. For the first access attempt, *a* shall be set to 0. The size of the contention window, CW_a,is defined as follows:

$$\mathrm{CW_a} = \begin{cases} 256 & 2^{a+1} \geq 256 \\ 2^{a+1} & 2^{a+1} \leq 256 \end{cases}$$

The open or association MTS used for the ath retransmission attempt shall be chosen by a uniformly distributed random integer value r_a within the interval [1, CW_a]. The random number generator is not specified. The DEV shall start counting r_a from the open or association MTS slot in the next superframe. The lack of an ACK indicates the failure of the previous access attempt, but the presence of an ACK does not necessarily equal success for an association frame since all stations attempting to associate use the same unassigned SA. See clause 8.2.5 for a description of the Association process.

This first broadcast or unassigned MTS is specified by number 'r_a=1'. The open or association MTS with number equal to r_a is the slot that the DEV shall access. The DEV shall not access the MTS before its counter has reached the open or association MTS with the number equal to r_a. After receiving an ACK, *a* shall be reset to 0.

Ed. action: Text that was added is given below.

8.3.3.1 Mangement Time Slots

Management Time Slots (MTSs) are identical to GTSs except that the PNC address (zero) is the source or the destination address in the CTA. An open MTS is an MTS where the source address in the CTA for the MTS is the broadcast address. Any DEV associated to the piconet can attempt to send a command frame to the PNC in an open MTS. An MTS with the association address as the SA in the CTA for the MTS is called an association MTS. Any station not currently associated to the piconet may attempt to send an association command to the PNC in an association MTS. Association commands shall not be sent in open MTSs. Likewise, only association commands shall be sent in association MTSs.

The access mechanism for regular MTSs, i.e. neither open nor association MTSs, is TDMA, as described in 8.3.3.1

Slotted Aloha is the access mechanism in an open MTS or an association MTS. The access to an open or association MTS shall be controlled by a contention window CW_a maintained by each DEV. The contention window shall be derived from the number *a*, where *a* is the number of retransmission attempts made by the DEV. For the first access attempt, *a* shall be set to 0. The size of the contention window, CW_a, is defined as follows:

$$CW_a = \begin{cases} 256 & 2^{a+1} \geq 256 \\ 2^{a+1} & 2^{a+1} \leq 256 \end{cases} \qquad (1)$$

The open or association MTS used for the a^{th} retransmission attempt shall be chosen by a uniformly distributed random integer value, r_a, within the interval [1, CW_a]. While the random number generator is not specified, it is important that designers recognize the need for statistical independence among the random number streams among DEVs. The DEV shall start counting r_a from the open or association MTS slot in the current superframe and continue across superframes. The lack of an ACK indicates the failure of the previous access attempt, but the presence of an ACK does not necessarily equal success for an association frame since all stations attempting to associate use the same unassigned SA. The association process is described in more detail in 8.2.5.

This first broadcast or unassigned MTS after the DEV begins the access process is specified by number 'r_a=1'. The open or association MTS with number equal to r_a is the slot that the DEV shall access. The DEV shall not access the MTS before its counter has reached the open or association MTS with the number equal to r_a. After receiving an ACK, *a* shall be reset to 0.

7.1.3 Clause 8.12.3.5 Additional Traffic to EPS DEVs

Add the following paragraph:

If management time slots are used, the PNC shall only assign management time slots for a device in EPS mode during superframes when the EPS device is scheduled to be listening to the beacon.

Ed. action: Above text added to the end of 8.12.3.5

7.2 Static GTS

7.2.1 Clause 7.5.21 Figure 49

Change Reason code to 4 bits. Use 2 bits for GTS type. 2 bits reserved.

Note: Modified stream management command appears below.

Ed. actions: Reason code modified, see 7.3.2 in this document.

7.2.2 Clause 8.3.3.1

Add the following sentence:

There are three types of GTS: dynamic GTS, Static GTS and pseudo-static GTS. The type of GTS slots are indicated in the stream management command as specified in 7.5.21.

7.2.3 New Clause 8.3.3..1.1 Dynamic Guaranteed Time Slots

The PNC is free to move dynamic GTSs within the superframe on a superframe by superframe basis. This allows the PNC the flexibility to rearrange GTS assignments to optimize the utilization of the slot assignments. The PNC can move a dynamic GTS by simply changing the CTA parameters in the Beacon.

7.2.4 New Clause 8.3.3.1.2 Static Guaranteed Time Slots

The PNC is not allowed to move the position of a static GTS slot in a superframe after the GTS has been assigned. Static GTSs require a stream connection - non stream GTSs cannot be static. The indication of the type of GTS is contained in the Stream Management command. DEVs can request a change in a static Slot channel time using the Channel Time Request command.

7.2.5 New Clause 8.3.3.1.3 Pseudo Static Guaranteed Time Slots

Pseudo static GTSs can be moved by the PNC, but the PNC must notify the affected DEVs by sending them acknowledged Channel Time Grant frames with the new CTA. As with dynamic GTSs, the PNC can rearrange pseudo static GTSs so that the GTS assignments can be optimized, but it must use the Channel Time Grant and coordinate the channel time grants with the CTAs in the beacon.

Before a pseudo static GTS is moved, the PNC shall ensure that the new position is unoccupied by another GTS. Then, the PNC sends a directed channel time grant to the receiving DEV so that the receiving DEV is listening to both the old GTS position and the new position. This channel time grant contains both the old and the new CTA. If the old and the new position overlap, the CTA can be one larger CTA.

Next the PNC sends a channel time grant to the transmitting DEV that contains only the new CTA. By moving the receiver first, the PNC ensures that no frames are lost if Channel time requests are corrupted.

Lastly, the PNC sends a channel time grant to the receiving DEV which only contains the new CTA.

DEVs can request a change in a pseudo-static Slot channel time using the Channel Time Request command.

(Figure here, not reproduced) Figure 1 CTA Assignment Sequence for pseudo-static GTSs

Ed action: The text added is given below:

There are three types of GTSs: dynamic GTS, static GTS and pseudo-static GTS. The type of a GTS is indicated in the stream management command as specified in 7.5.22.

The PNC may move dynamic GTSs within the superframe on a superframe by superframe basis. This allows the PNC the flexibility to rearrange GTS assignments to optimize the utilization of the slot assignments. The PNC moves a dynamic GTS by simply changing the CTA parameters in the beacon.

The PNC shall not move the position of a static GTS slot in a superframe after the GTS has been assigned. Static GTSs require a stream connection - non stream GTSs shall not be static. The indication of the type of GTS is contained in the Stream Management command. DEVs may request a change in a static slot channel time using the channel time request command, 7.5.6.1.

Pseudo-static GTSs may be moved by the PNC, but the PNC needs to notify the affected DEVs by sending channel time grant frames, 7.5.6.2, with the new CTA. As with dynamic GTSs, the PNC may rearrange pseudo static GTSs so that the GTS assignments will be optimized, but it must use the channel time grant command and coordinate the channel time grants with the CTAs in the beacon.

Before a pseudo static GTS is moved, the PNC shall ensure that the new position is unoccupied by another GTS. Then, the PNC shall send a directed channel time grant to the receiving DEV so that the receiving DEV is listening to both the old GTS position and the new position. The channel time grant shall be acknowledged with an Imm-ACK by the DEV if it is received correctly. If the PNC does not receive an Imm-ACK, it may re-send the channel time grant command, but it shall not continue with the process until the acknowledgement is received. The channel time grant sent by the PNC to the receiving DEV contains both the old and the new CTA. If the old and the new position overlap, the CTA may be one larger CTA.

The PNC shall then send a channel time grant to the transmitting DEV that contains only the new CTA. The transmitting DEV shall acknowledge with an Imm-ACK if it is received correctly. If the PNC does not receive the Imm-ACK, it may re-send the channel time grant command, but it shall not move the pseudo-static GTS until the acknowledgement is received. By moving the receiver first, the PNC ensures that no frames are lost if channel time requests are corrupted.

Finally, the PNC shall send a channel time grant to the receiving DEV which only contains the new CTA. The receiving DEV shall acknowledge this command with an Imm-ACK if it is received correctly. If the PNC does not correctly receive the acknowledgement, it may re-send the channel time grant command. However, the PNC may re-allocate the original pseudo-static GTS even if it does not receive the acknowledgement to the command.

Throughout the process, the PNC will ensure that the beacon contains the current CTAs for the pseudo-static GTS that it is moving.

DEVs may request a change in a pseudo-static GTS channel time using the channel time request command.

A graphical illustration of the procedure for moving a pseudo-static GTS is illustrated in Figure 3

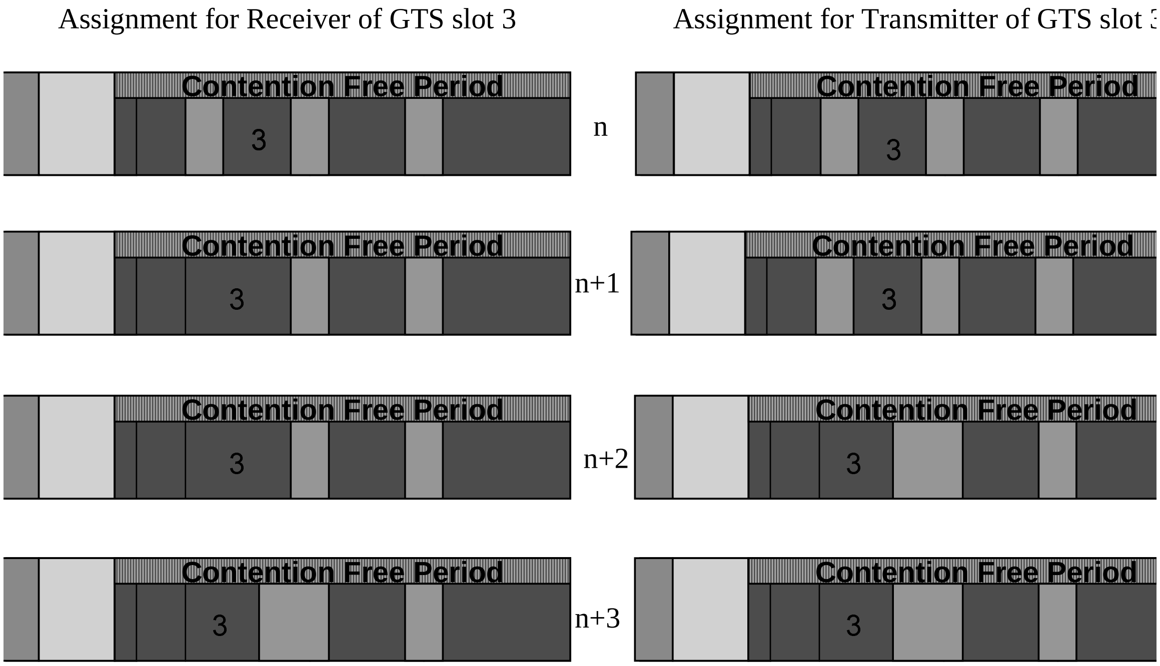


Figure 3—CTA assignment sequence for pseudo-static GTSs

7.2.6 Modified Clause 8.3.3.2 paragraph 1

The DEVs associated with a PNC shall send their changes in channel time requirement whenever they wish to make a change. Once a request for channel time is received from a station, the PNC shall remember that as the outstanding request for every superframe until, a change in request is received from the DEV. In addition to this the PNC shall make use of the properties of the stream provided during the stream connection process. The slot assignments within the CFP are based on the current pending requests from all the DEVs and the currently available channel time within the CFP. The slot assignments for dynamic GTS may change from superframe to superframe as required by the PNC. A lot assignments for pseudo-static GTS slots require directed channel time grant commands. All of the slot assignments are broadcast in the beacon. The PNC may announce the slot assignments in directed or broadcast channel time grant command in addition to the announcement in the beacon. However, additional announcements by the PNC shall not change from what was broadcast in the beacon. The start time of all the GTSs are with reference to the start of beacon frame, whether they were announced in beacon or channel time grant command. The algorithm used to allocate the channel time and assign slots is beyond the scope of this standard. Channel time requests that are ACKed are valid until the next channel time request is made.

Ed. actions: Text added to D08, changed sentence “A lot ... commands.” to “Changes to the assignments of pseudo-static GTSs require directed channel time grant commands, xref”.

7.2.7 Modified Clause 8.3.3.2 paragraph 4

In any superframe there may be one or more DEVs in the piconet that receives the Beacon in error. This may not happen to the same DEV all the time but may happen to different DEVs at different times depending upon their location and type of interference they are subjected to. If a DEV did not receive the ~~CTA information~~ beacon ~~correctly~~, it shall not access the ~~channel~~ dynamic GTS during CFP. Stations with static or pseudo static GTS(s) are allowed to transmit during these GTS(s) as long as the number of consecutive lost beacons is less than or equal to MaxLostBeacons. A DEV shall stop transmitting in its static and pseudo-static GTS when the number of consecutive lost beacons exceeds MaxLostBeacons.

The channel time grant command gives the flexibility to the PNC to broadcast the CTA information during the superframe in addition to the beacon. This increases the chances of all DEVs obtaining the allocation information. In addition, this also provides the flexibility to the PNC to help preserve the QoS by sending directed channel time grant command to a DEV that may be experiencing more than usual channel errors during certain time segments. The PNC may use the channel statistics to decide whether to send such a directed channel time grant command. Note that when channel becomes too severe for the DEV to receive the beacon, the directed channel time grant command or the data frame itself there is little help that can be provided to that DEV through these channel time grant commands.

Ed action: Text added, minor editorial changes. aMaxLostBeacons is an xref to a TBD at the end of the MAC functional description. New paragraph is below:

In any superframe there may be one or more DEVs in the piconet that receives the beacon in error. This may not happen to the same DEV all the time but may happen to different DEVs at different times depending upon their location and type of interference to which they are subjected. If a DEV did not receive the beacon, it shall not access any dynamic GTSs during CFP. Stations with static or pseudo static GTS(s) are allowed to transmit during these GTS(s) as long as the number of consecutive lost beacons is less than or equal to aMaxLostBeacons.. A DEV shall stop transmitting in its static and pseudo-static GTS when the number of consecutive lost beacons exceeds aMaxLostBeacons.

7.3 Modified Stream Management Command

Reserved Byte for max Delayed ACK frames in stream Mgt Command (Issue 310) and source/target address field in stream management command (Issue 400)

7.3.1 Figure 49 Modified Stream Management Command

Octets: 2	2	2	1	1	1	1	1	1	20
Command type	Length (=30)	Stream request identifier	Originator AD-AD	Target AD-AD	DSAA	Max Frames (delayed ACK)	Reason code/GTS type	reserved	Stream QoS parameter

Note: Reserved field will not be needed if the stream QoS parameters become odd after pending changes.

Ed actions: New table shown below: I have not added a reserved field pending the outcome of the Stream Oos paramters evaluation.

Octets: 2	**2**	**2**	**1**	**1**	**1**	**2**	**20**
Command type	Length (=26)	Stream request ID	Originator AD-AD	Target AD-AD	Max Frames (del-ACK	Control Information	Stream QoS parameters

Figure 4—Stream management command format

7.3.2 New Figure Reason code / GTS Type field (after figure 50)

4 bits	2 bits	2 bits
Reason Code	GTS Type	reserved

Ed actions: this is merged with DSAA which was temporarily frame policy, now management infromation field, shown below:.

Bits b0:b1	b2:b3	b4:b6	b7	b8:b11	b12:b13	b14-b15
Action Type	ACK Policy	Security	Direction	Reason code	GTS type	Reserved

Figure 5—Control information field in the stream management command

7.3.3 Replace line 50 and 51 of page 95 with the following:

The Originator AD-AD is the 8 bit address of the originator of the stream management command. The Target AD-AD is the 8 bit address of the target of the stream management command.

Ed action: text added, exact text is listed below.

The originator AD-AD the allocated address of the DEV that is the originator of the stream manament command. Depending on the value of the direction field, this DEV may be either the source or the destination of the stream.

The target AD-AD is the allocated address of the DEV that is the target of the stream managment command. Depending on the value of the direction field, this DEV may be either the source or the destination of the stream.

7.3.4 Add the following text after line 29, page 96:

Max Frames specifies the maximum number of frames that can be outstanding when the ACK policy for the stream is Delayed ACK.

Ed action: text added, exact text is listed below:

The max frames field specifies the maximum number of frames that can be outstanding when the ACK policy for the stream is Del-ACK.

8. Power management additional proposals, 01/430r1

Proposal by Jay Bain and Mark Schrader

Global change: Through the document, change occurrences of EPSPhase to EPSNext.

Ed action: Global find/replace accomplished in D08.

8.1 Add new section following 8.12.3.3 and before existing 8.12.3.4

8.12.3.4 Control of remote equipment power

Operations of EPS and AWAKE modes are extended into the equipment a DEV is incorporated into (DEV-host). MLME-POWERMGT.request and indicate primitives provide parameters to facilitate power saving in the equipment.

The MLME-POWERMGT.request Wakeup parameter causes a Switch to AWAKE CTA mode or Switch to EPS CTA mode to be sent to an EPS DEV. The EPS DEV receiving this command shall provide the MLME-POWERMGT.indicate primitive parameter PeerWakeup to its DEV-host.

The PeerWakeup parameter shall also provide indication that a packet has been received as a result of operation in EPS mode either with low duty cycle periodic information or momentary information. For each received packet while in EPS mode, the MLME-POWERMGT.indicate primitive shall be generated.

Annex xx provides information to achieve the appropriate power management benefits both in the EPS DEV and the DEV-host.

PS editor note: the following annex information is a placeholder for text to be added based on doc 01/430r0 and decisions reached in Rolling Meadows

Annex xx Considerations for power management in DEV-hosts

Informative

Power management techniques described in this standard are specific to the equipment that represent the PHY and MAC. This informative annex considers the relationship of power management in the DEV-host to the power management in the EPS DEV.

Remaining text TBD

Ed. action: text added to D08 and new annex (A). Also added a definition for device-host and acronym definitions for DEV and DEV-host. The definition is:

3.8 device: A device that implements an IEEE P802.15.3 conformant media access control and physical layer interface to the wireless medium.

3.9 device-host: The equipment that incoprorates and IEEE P802.15.3 device. The device-host may have more than one device incorporated in it as well as other networking connections, both wired and wireless.

8.2 Add the following in the 7.5.11 EPS configuration request

New descriptions -

EPSTime has a range of 0 to 65,535 milliseconds. A value of zero has the interpretation of DEVs waking for each superframe. Depending on the value of superframe duration parameter, values of EPSTime that are a value that is less than the current value of superframe duration are interpreted as wake for each superframe. Since wake time is bounded by superframe beacon location, the beacon start point preceding the completion of EPSTime shall be the wake point.

EPSNext is a value of beacon number as defined in Piconet synchronization parameters element. EPSNext informs PNC or DEVs where the next EPSTime will occur. For this command the value of EPSNext is taken from the EPSSync parameter in the MLME-POWERMGT.request primitive. The current beacon number when that primitive is received by the SME is used to calculate the beacon number for the next EPSTime event and inserts that beacon number as EPSNext in building the EPS configuration request command.

Ed. action: Added the following text (minor editorial changes).

EPSTime has a range of 0 to 65,535 ms. A value of zero indicates that the DEV is waking for each superframe. Depending on the value of superframe duration parameter, values of EPSTime that are less than the current value of superframe duration indicate that the DEV wakes for each superframe. Since the wake time is bounded by superframe beacon location, the beacon start point immediately preceding the completion of EPSTime shall be the wake point.

EPSNext is a beacon number as defined in piconet synchronization parameters element, 7.4.2. EPSNext informs the PNC or DEVs when the next EPSTime will occur. For this command, the value of EPSNext is taken from the EPSSync parameter in the MLME-POWERMGT.request primitive. The current beacon number when that primitive is received by the SME is used to calculate the beacon number for the next EPSTime event and inserts that beacon number as EPSNext when building the EPS configuration request command.

8.3 Replace the existing 6.3.1.3 MLME-POWERMGT.indication with the following

6.3.1.3 MLME-POWERMGT.indication

This primitive reports power management changes from a specific peer (peer, master, and slave) MAC entity. The primitive parameters are as follows

```
MLME-POWERMGT.indication        (
                                PeerPowerManagementMode,
                                PeerPowerManagementRole,
                                PeerWakeup,
                                PeerEPSTime,
                                DeviceID,
```

Table 2—MLME-POWERMGT.indication primitive parameters

Name	Type	Valid Range	Description
PeerPowerManagementMode	Enumeration	POWERED, RPS, EPS	Indicates the Power management mode of the peer
PeerPowerManagementRole	Boolean	True, False	Indicates the Power management role of the peer
PeerWakeup	Enumeration	EPS, ACTIVE, SHORTTERM	An indication that a mode change has occurred based on network information as described in clause 8
PeerEPSTime	Integer	65,535 ms	EPSTime from a specific peer. This should match sent EPSTime. If not, this is a negotiation request
DeviceID	MAC Address	Any valid individual MAC address.	Identifies the source of the indication primitive

6.3.1.3.1 When generated

This primitive is generated by the DEV as a result of a command or activity by another DEV in the piconet.

6.3.1.3.2 Effect of receipt

The SME is notified of changes in power management configuration or to wake up for information reception.

Ed. action: Table in D08 updated to include new parameter definitions and additional text listed above.

9. Child and Neighbor Piconet Functionality, 01/481r3

Proposal by Robert Huang, Shige Sugaya, K. Takamura, M. Suzuki, Masa Akahane and Mark Schrader.

9.1 Clause 3 additions

3. Definitions

Editorial instruction: Insert new definitions

9.1.1 child piconet: A piconet that exists entirely within a private guaranteed time slot of another piconet (parent piconet) and is controlled by an alternate coordinator that is a member of the parent piconet. The area of overlapping coverage between the two piconets may vary between congruent with the parent coverage area to mostly non-overlapping.

9.1.2 parent piconet: A piconet which allocates a private guaranteed time slot for another piconet of the same system (child piconet) or of another system (neighbor piconet).

9.1.3 neighbor piconet: A piconet that exist entirely within a private guaranteed time slot of another piconet (parent piconet) and is controlled by an alternate coordinator that is not a member of the parent piconet. The area of overlapping coverage between the two piconets may vary between congruent with the parent coverage area to mostly non-overlapping. The PNC of the neighbor piconet will request a private GTS from the parent piconet using the 802.15.3 protocol and will be able to accept 802.15.3 protocol requests for a private GTS (or equivalent) from a (potential) neighbor 802.15.3 piconet.

9.1.4 alternate coordinator: A member of the piconet that is capable of being the coordinator but is not functioning as the coordinator.

Ed action: Text added as indicated with the word “private” deleted since there are no private GTSs in 802.15.3. The neighbor piconet definition had the additional text describing the functional aspects deleted. The actual text is:

9.1.5 neighbor piconet: A piconet that exist entirely within a guaranteed time slot of another piconet (parent piconet) and is controlled by an alternate coordinator that is not a member of the parent piconet. The area of overlapping coverage between the two piconets may vary between congruent with the parent coverage area to mostly non-overlapping.

9.2 Clause 4 additions

4. Acronyms and abbreviations

GTS guaranteed time slot

Ed. action: Excellent catch, I can’t believe we wen this far without anyone noticing this one missing. Text added as suggested in D08.

9.3 Clause 5 additions

5.4 Overview of MAC functionality

IEEE 803.15.3 provides for the following functionality:

- Allow stations to form and terminate PANs
- Transport data between stations
- Authenticate stations with each other
- Allow stations (including the coordinator) to minimize power requirements and still maintain the network.
- A procedure for the coordinator to pass coordination to another station in the WPAN.
- A procedure to establish a child piconet (subnetwork with same ownership).
- A procedure to establish a neighbor piconet (subnetwork with different ownership).

Ed. action: The above two items in the list added, wording changed slightly, actual text from D09 is below:

- A procedure to establish a child piconet. A child piconet is a sub-piconet that is linked to the parent piconet through its PNC which is a member of the parent piconet.
- A procedure to establish a neighbor piconet which is a sub-piconet that is disjoint from the parent piconet. The PNC of the neighbor piconet is not a member of the parent piconet.

5.4.7 Child Piconet

A child piconet may be formed under an established piconet, wherein the established piconet becomes the parent piconet. The child piconet uses its own piconet ID (PNID) but acts as an autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments are handled within in the child piconet and do not involve the parent PNC.

The child PNC device is a member of the parent piconet and thus can exchange data with any device in the parent piconet. The child PNC device is also member of the child piconet and thus can exchange data with any device in the child piconet.

5.4.8 Neighbor Piconet

A neighbor piconet may be formed under an established piconet, wherein the established piconet becomes the parent piconet. The neighbor piconet has a different owner than the parent piconet and is autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments are handled within in the neighbor piconet and do not involve the parent PNC.

The neighbor PNC device is not a member of the parent piconet and thus can not exchange any data with any device in the parent piconet.

The neighbor piconet mechanism may be used by other 802-compliant wireless devices as a means of sharing the frequency spectrum. A device capable of initiating (requesting status as) a 802.15.3 neighbor piconet should also be capable for supporting a 802.15.3 piconet as a neighbor.

Ed. action: Small changes, removed a couple of may's, added some commas, etc. Actual text is below:

5.4.7 Child Piconet

A child piconet is onet that is formed under an established piconet, wherein the established piconet becomes the parent piconet. The child piconet uses a distinct piconet ID (PIND) and acts as an autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments are handled within in the child piconet and do not involve the parent PNC.

The child PNC device is a member of the parent piconet and thus is able to exchange data with any device in the parent piconet. The child PNC device is also member of the child piconet and thus is able exchange data with any device in the child piconet.

5.4.8 Neighbor Piconet

A neighbor piconet is formed under an established piconet, wherein the established piconet becomes the parent piconet. The neighbor piconet uses a distinct PNID and is an autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments are handled within in the neighbor piconet and do not involve the parent PNC.

The neighbor PNC device is not a member of the parent piconet and thus does not exchange any data with any device in the parent piconet.

The neighbor piconet mechanism is available to other 802-compliant wireless devices as a means of sharing the frequency spectrum. A device capable of initiating (i.e. requesting status as) an 802.15.3 neighbor piconet would also be capable for supporting a 802.15.3 piconet as a neighbor.

5.6.6 Network termination

5.6.6.1 Involving Child Piconet

Upon termination of a parent piconet, the child piconet may continue to operate without disruption. The child piconet will remove the Parent Device ID Element from its beacon frame, signifying that it is a freestanding piconet. In the case where the parent piconet was temporarily disrupted, the parent PNC may join the child piconet, challenge for control and receive a transfer of control (promotion to PNC or handover). In this case, the child piconet should continue without disruption.

Upon termination of a parent piconet, the child piconet may also terminate. The PNC of the (former) child piconet may immediately attempt to establish a new piconet. The termination of a child piconet shall have no effect on the parent piconet except to release resources.

5.6.6.1 Involving Neighbor Piconet

Upon termination of a parent piconet, the neighbor piconet may continue to operate without disruption. The neighbor piconet will remove the Parent Device ID Element from its beacon frame, signifying that it is a freestanding piconet. In the case of a temporary termination of the parent piconet, the (former) parent PNC can restart as a neighbor (i.e., role reversal).

The neighbor piconet shall terminate its relationship with the parent PNC using the disassociation request command. The termination of a neighbor piconet shall have no effect on the parent piconet except to release resources.

Ed. action: Text still has may's and shall's in clause 5 text, which I have declared to be an editorial no-no. So the text is cleaned up a little bit. Actual text follows:

5.6.6.1 Ending a piconet with a child piconet involved

If the parent piconet ends operation, the child piconet is able to continue to operate without disruption. The child piconet removes the parent device ID element from its beacon frame, signifying that it is a free-standing piconet. In the case where the parent piconet was temporarily disrupted, the parent PNC is able to join the child piconet, challenge for control and receive a transfer of control (i.e. PNC handover). In this case, the child piconet would continue without disruption.

However, the child piconet might instead choose to also end operations if the parent piconet ends operations. In this case, the PNC of the (former) child piconet could immediately attempt to establish a new piconet.

When a child piconet ends operation it will have no effect on the parent piconet except to release resources.

5.6.6.2 Ending a piconet with a neighbor piconet involved

If a parent piconet ends operation, the neighbor piconet is able to continue to operate without disruption. The neighbor piconet removes the parent device ID element from its beacon frame, signifying that it is a freestanding piconet. In the case of a temporary termination of the parent piconet, the (former) parent PNC is able to restart as a neighbor (i.e., role reversal).

The neighbor piconet uses the disassociation requrest command to end its relationship with the parent PNC. When a neighbor piconet ends operation it has no effect on the parent piconet except to release resources.

9.4 Clause 7 additions

7.2.3 Address fields

There are two address fields in the MAC frame format and each of these fields is 8 bits in length. These fields are used to indicate the destination address (DA) and the source address (SA). An address for a DEV is assigned by the PNC during the association of the DEV. The address of a DEV is unique to an associated DEV within a piconet. The following addresses are reserved.

— The address value 0 is reserved for the PNC, for coordinator related transmissions and receptions
— The address value of all-ones (0xFF) is reserved for broadcast frames
— The address value of 0xFE is reserved for use by all new clients during their association until a unique address is allocated to each one of those new clients by the PNC.
— The address value of 0xFD is reserved for multicast frames
— The address values of 0xFA, 0xFB or 0xFC are reserved for neighbor piconets

Ed. action: Added above text for last item, changing network to piconet.

7.3.1 Beacon frame format

Editorial instruction: Add new element:

Table 58-Beacon frame body

Information element	**Note**
Device ID	Element ID + Length + 48 bit IEEE 802 address of the PNC
Piconet synchronization parameters	TSF element and other time duration elements
TPC element	Sets the max TX power level in the piconet
Channel change	During change to new channel
Channel time allocation (CTA)	All the channel time allocation in the current superframe
Parent Device ID (if child or neighbor piconet)	48 bit IEEE 802 address of the Parent PNC

Ed. action: new element added.

7.4.11 Channel time allocation (CTA) element

:Editorial note: insert in the position one paragraph after Figure 19.

The source DEV address indicates the DEV to whom the channel time is being allocated.

For a child PNC, the source DEV and destination DEV addresses shall both be the AD-AD of the child PNC (assigned by the parent PNC).

For a neighbor piconet, the source and destination addresses shall be the same and shall be in the range 0xFA to 0xFC.

Ed. action: Small changes to avoid defining the same thing twice in different places. Actual text is below:

The source DEV address indicates the DEV to whom the channel time is being allocated.

The destination DEV address indicates the DEV to whom the source DEV may send the frames. If this is a broadcast address, then the source DEV shall send broadcast frames only during that time slot.

For a child PNC, the source DEV and destination DEV addresses shall both be the AD-AD of the DEV that is the child piconet's PNC.

For a neighbor piconet, the source and destination addresses shall both be the AD-AD assigned by the parent PNC for the neighbor piconet and shall be one of the reserved neighbor piconet addresses, 7.2.3.

7.5 Command types

Command type hex value	Command name
0x0000	Retransmission request
…	…..
0x001B	Momentary EPS CTA
0x001C	Neighbor association request
0x001D	Neighbor channel time request
0x001E	Child or neighbor information response
0x001C-0xFFFF	Reserved

Table 63-Command types

Ed. action: New command types added as indicated.

7.5.2 Association request command format

Editors note: This insertion will cause a renumbering of following items.

7.5.3 Neighbor association request command format

This command is used by a neighbor PNC to associate (without gaining network membership) with a piconet in order to establish a neighbor piconet. The format of the neighbor association request command is shown in figure LL.

2	2	6	2
Command type	Length	Device ID	ATP

Figure LL-Association request command format

The device ID field is the 48-bit IEEE 802 address of the neighbor PNC involved in the association.

The ATP (association timeout, period), which is given by aAssocTimeoutPeriod, is the timeout during which if the frames from parent PNC meant for the neighbor PNC (current DEV) are not received at the neighbor PNC, the neighbor PNC disassociates and tries to associate again. Similarly, if PNC did not receive any frame originating from the current neighbor PNC within this timeout duration, the PNC may disassociate the neighbor PNC and expect the neighbor PNC to associate again. The PNC shall not disassociate the current neighbor PNC for the reason of absence of frames from the neighbor PNC within that timeout period.

7.5.4 Association response command format

Ed. action: Inserted text, small edits to the last paragraph, actual text below:

The ATP (association timeout period) is the timeout, in ms, during which if the frames from parent PNC meant for the neighbor PNC (current DEV) are not received at the neighbor PNC, the neighbor PNC disassociates and tries to associate again. Similarly, if PNC did not receive any frame originating from the current neighbor PNC within this timeout duration, the PNC may disassociate the neighbor PNC, although the neighbor PNC may attempt to associate again.

Editors note: insert text after the sixth paragraph. The sixth paragraph begins " The AD-AD field…"

For the association of a neighbor PNC, the AD-AD shall be 0xFA, 0xFB or 0xFC.

Ed. action: This is already defined in 7.2.3 and so will not be added here since it would define the same thing in two different places.

7.5.5.2 Neighbor channel time request command

The neighbor PNC shall send a channel time request using the format in 7.5.5.1

Ed. action: Instead of adding the addititional section, I modified the previous sub-clause to be valid for the neighbor piconet. Actual text is below:

The channel time request command is used by associated members of the piconet to allocate channel time. In the case of neighbor piconets, the neighbor PNC uses this command to request channel time to set up the neighbor piconet. The structure of this commandis illustrated in Figure 29.

7.5 Command Types

7.5.19.3 Child or neighbor information response command

This command, following the same response rules a the device information response command, defined in 7.5.19.2, uses the structure illustrated in Figures ZCZ and YZY

ctets: 2	2		12	12	….	
ommand Type	Length		Record for child or neighbor PNC	Record for child or neighbor PNC	….	

Figure ZCZ - Child or neighbor information response command

1	Octets: 6	1	1	1	2
Type ID	Neighbor PNC Device ID	Stream Index	Duration between time slots	Minimum requested channel time in a time slot	Reques channel per time

Figure YZY - Child or neighbor PNC information response record

The type ID shall be 0x01 for a child piconet and 0x02 for a neighbor piconet.

The duration between transmissions indicates the requested frequency of time slots that needs to be allocated for the DEV. The resolution of this field is Kms.

The minimum requested channel time is the minimum duration of the time that is acceptable at the requesting DEV in any time slot. If the allocated time slot is smaller than the time indicated in this field, the PNC may not allocate that time-slot at all for the DEV. The resolution of this field is 32 ms and so the range of this field is less than [0-8160] ms.

The requested channel time is the time required at the DEV in every time slot for transmission of the indicated stream. Hence this field represents the time required for the given latency and not necessarily for the entire superframe. The resolution of the channel time field is 8 µs and so the range of requested time is less than [0-524280] ms.

Ed. action: Sub-clause and figures added, some of the descriptive text modified. I deleted the allocation requirements on the PNC since this is defined in clause 8 rather than here. Deleted the same words from 7.5.6.1 Channel Time Allocation Request. Actual text follows:

The minimum requested channel time is the minimum duration of the time that is acceptable to the requesting DEV in any time slot. The resolution of this field is 32 µs and so the range of this field is less than [0-8160] µs.

The requested channel time is the time required at the DEV in every time slot for transmission of the indicated stream. Hence this field represents the time required for the given latency and not necessarily for the entire superframe. The resolution of the channel time field is 8 µs and so the range of requested time is in the range of [0-524280] µs.

9.5 Clause 8 additions

Editorial instruction: New text

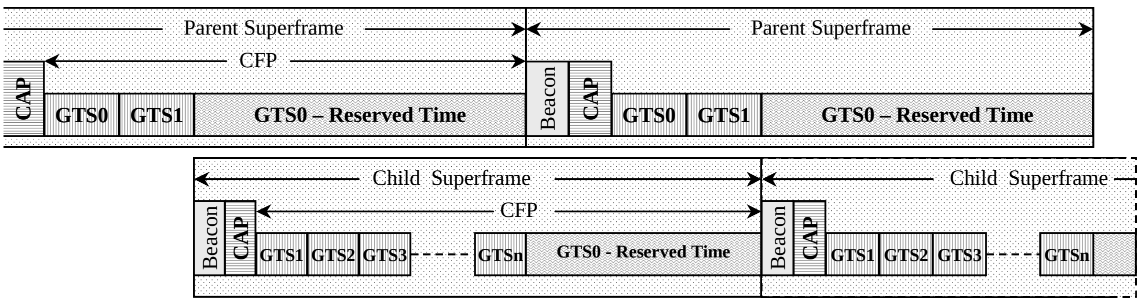
8.2.9 Child piconet

When an alternate coordinator of an existing piconet wants to form a child piconet, the alternate coordinator shall request a GTS in which to operate a child piconet. By setting the source and destination addresses as the same.

The channel time request command, defined in 7.5.5.1, is used, setting the source and destination addresses to the child PNC AD-AD. The parent PNC will recognize this as a request for a child piconet.

After receiving theprivate GTS, the child PNC will start sending its beacon. Included in the child piconet beacon is a private GTS for the parent piconet, using the PNC's address (0x00) for the source device address. This is provided to reserve the slot, not to convey any information to the Parent PNC.

Figure RR illustrates the relationship between the parent piconet superframe and the child piconet superframe. It can be seen that the superframe duration is the same for both the child and the parent piconets.



Note: The slot positions GTS0, GTS1, et al, are illustrative.

Figure RR - Parent piconet and child piconet superframe relationship.

The message sequence chart for the initiation of the child piconet is shown in figure abc.

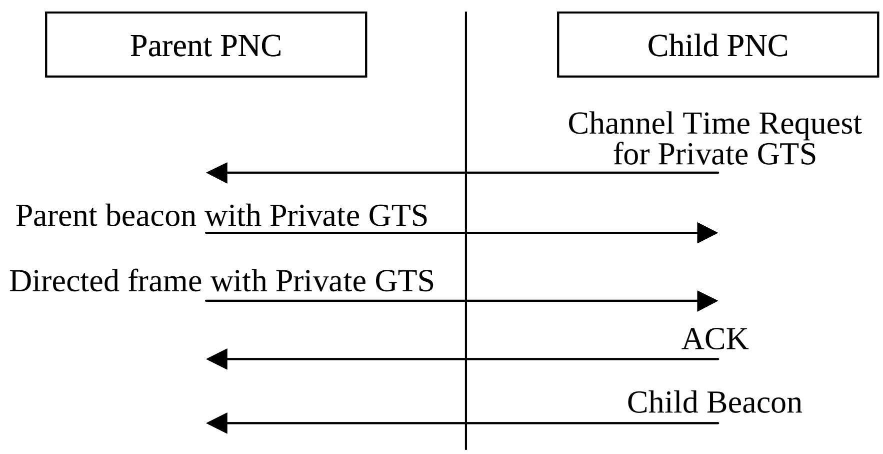


Figure abc – Message Sequence Chart: initiation of a child piconet

The child piconet is an autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments are handled within the child piconet and do not involve the parent PNC.

The child PNC device is a member of the parent piconet and thus can exchange data with any device in the parent piconet. The child PNC device is also a member of the child piconet and thus can exchange data with any device in the child piconet.

Ed. action: Text added mostly as is. Some cleanup of the wording. Actual text is below:

When an alternate coordinator of an existing piconet wants to form a child piconet, the alternate coordinator shall use the channel time request command, defined in 7.5.8.1 to request a private GTS. The AC shall set the source and destination addresses in the channel time request command to be the AD-AD of the AC. The parent PNC will recognize this as a request for a child piconet. The PNC may allocate a private GTS for the child piconet depending on the availability of network resources and security policy.

If the AC (now the child PNC) receives a private GTS, the child PNC shall start sending its beacon in its allocated private GTS. The child PNC shall use a PNID that is distinct from the parent PNID. The child piconet beacon contains the parent device ID, a shown in Table 71.

Included in the child piconet beacon is a private GTS for the parent piconet, using the PNC address (0x00) for both the source and destination DEV address. This is provided to reserve the slot, not to convey any information to the parent PNC.

Figure 66 illustrates the relationship between the parent piconet superframe and the child piconet superframe. It can be seen that the superframe duration is the same for both the child and the parent piconets.

(figure goes here)

Note that the slot positions GTS0, GTS1, et al, are not to scale in Figure 60 are illustrative only.

The message sequence chart for the initiation of the child piconet is shown in Figure 67.

(another figure goes here).

The child piconet is an autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments shall be handled within the child piconet and do not involve the parent PNC.

The child PNC device is a member of the parent piconet and thus may exchange data with any device in the parent piconet. The child PNC device is also a member of the child piconet and thus may exchange data with any device in the child piconet.

8.2.10 Neighbor piconet

A neighbor alternate PNC (alternate PNC of a different system), wanting to start a piconet, shall first scan for a piconet to join, then finding none, shall scan for an open channel in which to initiate a piconet. If no channels are available, then the neighbor alternate PNC may attempt to start a neighbor piconet within a private GTS of an existing piconet. To start a neighbor piconet, the potential neighbor PNC shall send a neighbor association request, defined in 7.5.3, using the AD?AD of 0xFE (refer 7.2.3).

The parent PNC shall respond with a association response command (refer 7.5.4). If the neighbor association request is accepted, then the AD-AD shall be 0xFA, 0xFB or 0xFC (refer 7.2.3). If not accepted, the AD-AD shall be set to 0xFE.

To obtain a private GTS, the neighbor PNC sends a channel time request, defined in 7.5.5.1.

If the parent PNC permits the formation of a neighbor piconet, a private GTS is sent using 0xFA, 0xFC or 0xFC as both the source and destination. These addresses are reserved for the neighbor piconet (refer 7.2.3). After receiving this channel time allocation, the neighbor PNC will start sending its beacon. The neighbor beacon contains the parent piconet ID (refer Table 58).

Included in the neighbor superframe is a private GTS for the parent piconet, using the parent PNC's address (0x00) for the source device address. This is provided to reserve the slot, not to convey any information to the Parent PNC.

Figure RR illustrates the relationship between the parent piconet superframe and the neighbor piconet superframe. It can be seen that the superframe duration is the same for both the neighbor and the parent piconets.

The message sequence chart for the initiation of the neighbor piconet is shown in figure def.

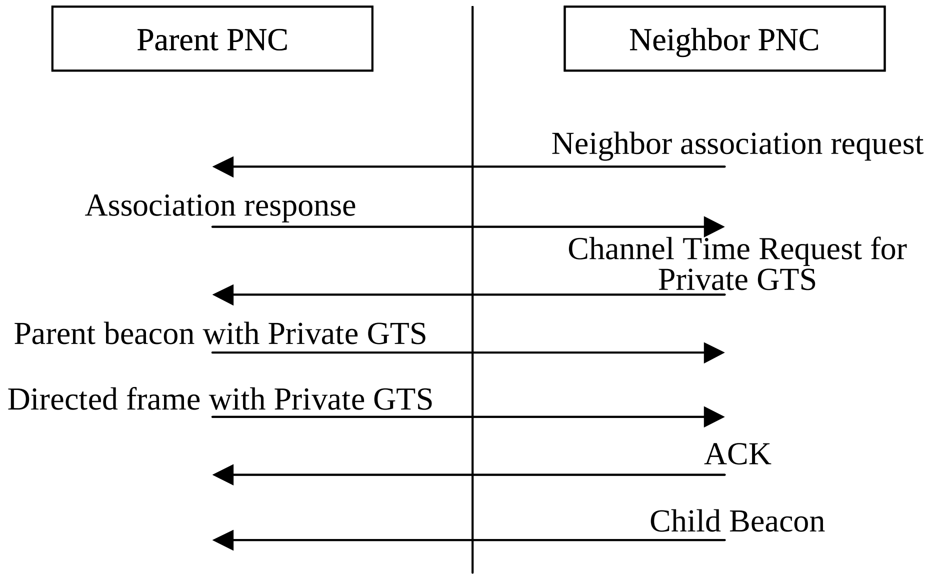


Figure def – Message Sequence Chart: initiation of a neighbor piconet

The neighbor piconet is an autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments are handled within the neighbor piconet and do not involve the parent PNC.

The neighbor PNC is not a member of the parent piconet (not authenticated and not associated) and thus is restricted to only sending a channel time request and reading the parent piconet beacon. The parent PNC is not a member of the neighbor piconet.

Ed. action: Text added mostly as is. Some cleanup of the wording. Actual text is below:

Neighbor piconet

If after following the scan procedure in 8.2.1, no channels are available, then a neighbor alternate coordinator (i.e. an AC from a different system), may attempt to start a neighbor piconet within an existing piconet. To start a neighbor piconet, the neighbor AC shall send an association request, defined in 7.5.2.2, using the association address, 7.2.3, as the source address of the frame.

If the parent PNC receives the neighbor association request correctly, it shall respond to the request with an association response command, 7.5.2.3. If the neighbor association request is accepted, then the PNC shall set the AD-AD in the command to be one of the unused neighbor piconet addresses, 7.2.3. If the request is not accepted, then the PNC shall set the AD-AD in the command to the association address, 7.2.3.

The neighbor PNC then sends a channel time request, 7.5.8.1, to obtain a private GTS for the neighbor piconet. The channel time request shall have both the source and destination addresses set to the neighbor piconet address that was assigned to the neighbor PNC by the parent PNC.

If the parent PNC permits the formation of a neighbor piconet, it shall allocate a private GTS using the neighbor piconet address as both the source and destination addresses. After receiving this channel time allocation in the parent PNC's beacon, the neighbor PNC shall start sending its beacon in the neighbor piconet's private GTS. The neighbor PNC shall use a PNID that is distinct from the parent PNID. The neighbor beacon contains the parent device ID, a shown in Table 71.

Included in the neighbor superframe is a private GTS for the parent piconet, using the PNC address for both the source and destination device addresses. This is provided to reserve the slot, not to convey any information to the parent PNC.

Figure 66 illustrates the relationship between the parent piconet superframe and the neighbor piconet superframe. It can be seen that the superframe duration is the same for both the neighbor and the parent piconets.

The message sequence chart for the initiation of the neighbor piconet is shown in figure Figure 68.

(figure goes here)

The neighbor piconet is an autonomous piconet except that it is dependent on a private GTS from the parent piconet. Association, authentication, security and acknowledgments shall be handled within the neighbor piconet and do not involve the parent PNC.

The neighbor PNC is not a member of the parent piconet (i.e. it is neither authenticated nor associated) and thus is restricted to only sending channel time requests and reading the parent piconet beacon. The parent PNC is not a member of the neighbor piconet.

10. QoS Policy Proposal for TG3-MAC, 01/469r2

Proposal by Allen Heberling.

10.1 Clause 3: Definitions

3.1 Quality of Service

A collective measure of the level of service delivered between DEVs. QoS can be characterized by several basic performance criteria, including availability (low downtime), error performance, response time and throughput, lost calls or transmissions, connection set-up time, and speed of fault detection and correction.

3.2 QoS Strictness

Three broad categories of QoS, in a WPAN, are: best effort, differentiated, and guaranteed. The "strictness" of the QoS service describes how tightly the service can be bound by specific bandwidth, delay, jitter, and loss characteristics.

3.3 Service Flow:

A Service Flow is a unidirectional flow of MAC SDUs via a stream associated with a defined Quality of Service.

Ed. action: Above text added with minor edits to match IEEE style, text given below:.

10.2 quality of service: A collective measure of the level of service delivered between devices. Quality of service can be characterized by several basic performance criteria, including availability (low downtime), error performance, response time and throughput, lost calls or transmissions, connection set-up time, and speed of fault detection and correction.

10.3 quality of service strictness: Three broad categories of quality of service, in a wireless personal area network, are: best effort, differentiated, and guaranteed. The "strictness" of the quality of service describes how tightly the service can be bound by specific bandwidth, delay, jitter, and loss characteristics.

10.4 service flow: A service flow is a unidirectional flow of media access controller service data units via a stream associated with a defined quality of service.

10.5 Clause 4: Abbreviations and acronyms

CPS	Common Part Sub layer
DME	Device Management Entity
PCS	Packet Convergence Sub layer
SFID	Service Flow ID
SSCS	Service Specific Convergence Sub layer
TOS	Type of Service

Ed. action: Added the above acronyms and deleted the following acronyms since they are not used in the draft.

AN-ID	allocated network identifier
CTP	contention time out
DBPSK	differential binary phase shift keying
DCE	data communication equipment
DOQPSK	differential offset quadrature phase shift keying
IDU	interface data unit
LMP	link manager protocol
MDF	management defined field
RTC	real time traffic capable
RTS	request to send
RTX	response timeout expired
SDP	service discovery protocol
SEQN	sequential numbering scheme
SLRC	station long retry count
SQ	signal quality
SRC	short retry count
SRES	signed response
SS	station services
SSRC	station short retry count
TA	transmitter address
TDD	time division duplex
TSF	timing synchronization function
TXE	transmit enable

10.6 Clause 6. Service Specific Convergence Sub layer

The Service Specific Convergence Sub layer (SSCS), as illustrated in Figure 1, exists conceptually above the MAC Common Part Sub layer (MAC-CPS). The SSCS, using the services of the MAC-CPS, supports these functions:

a) receiving PDUs from upper protocol layers
b) classifying received PDUs and processing the same if required
c) delivering each SSCS PDU to the MAC CPS-SAP
d) receiving SSCS PDUs from a peer SSCS entity

Currently the Packet CS is the only SSCS specified for 802.15.3.

(figure 1 goes here, file 15-3Protocol Layer Model.eps)

6.1 Packet Convergence Sub layer

The Packet Convergence Sub layer (PCS) exists above the 802.15.3 MAC-CPS. The PCS, using the services of the MAC-CPS, supports these functions:

a) classifying received, upper layer PDUs and mapping each PDU to a specific Stream-Index.

b) Delivering each valid Packet PDU to the MAC CPS-SAP

c) Receiving Packet PDUs from peer PCSs

The PCS supports transport of packets received from these upper layer protocols: IP and IEEE802.3/.2.

6.1.1 Classification

The PCS PDU classification process maps each PCS PDU to a specific Stream-ID and its associated Service Flow. Each Service Flow has associated with it a set of QoS characteristics. Consequently, after classification, each PCS PDU will be delivered using the QoS parameters specified for the Stream-ID/Service Flow pair.

The classification process uses one or more Classification parameters sets to analyze each packet entering the 802.15.3 PCS. Each set includes a classification priority, Stream-Index, and protocol specific parameters such as destination MAC address, source MAC address, and optionally a priority parameter. If a PCS PDU, received from an upper layer protocol, matches the specified protocol specific parameters, it is then sent to the MAC CPS-SAP for delivery via the stream indicated by the Stream-ID. If the PCS PDU does not match the specified protocol parameters, the packet may be delivered using either a default Stream-ID or the packet may be discarded. The policy for deciding how to handle a packet in this instance is outside the scope of this standard. Figure 2 provides a graphical representation of the entities involved.

In the case where more than one classification parameters set is available, the classification process first shall use the classification parameters set containing the highest valued classification priority parameter. If no match is found with the first classification parameters set, the next highest priority classification parameters set will be applied. This process will repeat itself until either the incoming packet is properly matched and assigned to a specific Stream-ID/Service Flow pair for subsequent delivery, or there are no more classification parameters sets available and the incoming packet is either discarded or delivered with a default delivery QoS.

(Figure 2 goes here, file name ClassificationAndStrm-Indx-Mapping.eps)

6.1.2 802.2 CS-SAP

See text in D0.7 Page 51, Clause 6.7.2

~~7~~6.1.1.1 MA-UNITDATA.request

See text in D0.7 Page 51, Clause 6.7.2.1

~~7~~6.1.1.2 MA-UNITDATA.indication

See text in D0.7 Page 52, Clause 6.7.2.2

~~7~~6.1.1.3 MA-UNITDATA-STATUS.indication

See text in D0.7 Page 53, Clause 6.7.2.3

6.1.3 IEEE 802.2 Logical Link Control specific part

The 802.2 SSCS supports two QoS types, which are negotiated during the association process:

a) Best Effort
This is the default QoS type that be shall be supported. All xx PDUs are handled the same (i.e. no QoS guarantees are provided regarding delivery of the received PDU).

b) IEEE 802.1p priority based QoS scheme
The 802.1p priority scheme provides basic support for 1 to eight (0-7) traffic types. The 8 different traffic types are illustrated in Table 1 IEEE 802.1p Traffic Types.

Table A.1—IEEE 802.1p Traffic Types

User Priority	Traffic Type	Comments
0 (Default)	Best Effort (BE)	Default WPAN traffic
1	Background (BK)	
2	-	Spare
3	Excellent Effort (EE)	For valued customers
4	Controlled Load (CL)	Traffic will have to conform to some higher protocol layer admission control.
5	Video (VI)	< 100 ms delay and jitter
6	Voice (VO)	< 10 ms delay and jitter
7	Network Control (NC)	

6.1.3.1 IEEE 802.2 CS classification parameters set

The 802.2 CS classification parameters set contains zero or more of these 802.2 parameters:

a) Destination MAC address

This parameter specifies a 48-bit IEEE MAC Destination Address.

b) Source MAC address
This parameter specifies a 48-bit IEEE MAC Source Address.

c) Priority
This parameter specifies the priority assigned to each 802.2 PCS-PDU according to IEEE 802.1D.

In addition, these classification parameters are used:

a) classification priority
a) Stream-Index

6.1.3.2 Data entity inter-relationships

Figure 3 illustrates the relationship among the SDU, Classification Parameters Set, Service Flow, Connection and PDU entities. These entities and the underlying protocol mechanisms that establish these entities and their relationship with each other are key to enabling support for QoS delivery of PDUs from one MAC entity to another.

(figure goes here,DataEntityInter-relationshipModel.eps)

Figure 3 Data Entity Inter-relationship Model

6.1.4 Internet Protocol specific part

When the Internet Protocol, specified in RFC-791 and RFC-2460, is the upper protocol layer, these IP CS classification parameters are used:

a) P TOS Range/Mask
b) IP Protocol
c) IP Source Address/Mask
d) IP Destination Address/Mask
e) Protocol Source Port Start
f) Protocol Source Port End
g) Protocol Destination Port Start
h) Protocol Destination Port End

Ed. action: Text added as indicated to create new annex A, changed section depth in a few places. Deleted 6.7.2 from Layer Managment to insert here.

10.7 Clause 7

Ed note: this is actually supposed to go into Layer Management, clause 6.

7. MAC Common Part Sublayer (MAC CPS)

The MAC CPS provides stream-oriented service to the SSCS. This service enables the SSCS to map Service Flows and their associated QoS parameters to specific streams. This mapping of a Service Flow to a specific stream is a primary attribute of this MAC protocol. Service Flows, in this context, provide a mechanism for managing the QoS characteristics of uplink (DEV->PNC), downlink (PNC->DEV), and peer link (DEV->DEV) streams.

Streams are dynamic in nature, in that they may be created, modified, and deleted. An established stream may need to be modified due to the type of service assigned to it. For instance, IP services may require the QoS characteristics of the stream to be modified due to the burst nature of the service.

7.1 MAC CPS SAP

The MAC CPS SAP defines the logical interface between the MAC CPS and the SSCS above it. This logical interface description includes a list of primitives and their definitions. Although these primitives and their definitions are informative, they provide a context in which to understand the parameters, which need to be passed between the MAC CPS and the SSCS so that each sub layer may fulfill its specified functions.

7.1.1 Primitives

The IEEE 802.15.3 MAC CPS supports these primitives at the MAC CPS SAP:

— MAC_DATA.request
— MAC_DATA.indication

7.1.1.1 MAC_DATA.request

This primitive is used to initiate the transfer of a MAC-CPS PDU from one MAC-CPS entity to another MAC-CPS entity or entities.

7.1.1.1.1 Semantics of the service primitive

```
MAC_DATA.request
                        (
                        destination DEV address,
                        source DEV address,
                        stream_Index,
                        length,
                        data,
                        )
```

The destination DEV address is the DEV address to which the MAC-CPS PDU is to be sent.

The source DEV address is the DEV address from which the MAC-CPS PDU is sent.

The stream_Index parameter specifies the stream over which the data is to be sent.

The length parameter specifies the length of the MAC-CPS SDU in bytes.

The data parameter specifies the data portion of the MAC-CPS SDU.

7.1.1.1.2 When generated

This primitive is generated by the PCS upon receiving a MA-UNITDATA.request from the LLC sub layer.

7.1.1.1.3 Effect upon receipt

The MAC-CPS upon receiving this primitive uses the received parameters to format an appropriate MAC-CPS PDU which is then sent to the PHY-SAP for transfer over the wireless media to a peer MAC-CPS entity or entities.

7.1.1.2 MAC_DATA.indication

This primitive is used to indicate the reception of a MAC-CPS SDU

7.1.1.2.1 Semantics of the service primitive

MAC_DATA.indication
(
destination DEV address,
source DEV address,
stream_Index,
length,
data,
reception status,
)

The destination DEV address, source DEV address, stream_Index, length, and data parameter definitions are the same as in clause 7.1.1.1.1

7.1.1.2.2 When generated

This primitive is generated by the MAC-CPS upon successfully processing a received MAC-CPS PDU.

7.1.1.2.3 Effect upon receipt

When the PCS receives this primitive from the MAC-CPS it will generate a MA-UNITDATA.indication.

Ed. note: Text added as indicated. First section, 7, was moved to the introduction, which still needs to be rewritten. Description of parameters put into a table as done with other primitives in this clause with what should be appropriate values for type and range.

7.2 MLME SAP

7.2.1 Primitives

The IEEE 802.15.3 MLME supports these primitives at the MLME SAP:

— MLME_CREATE_STREAM.request
— MLME_CREATE_STREAM.indication
— MLME_CREATE_STREAM.response
— MLME_CREATE_STREAM.confirm

— MLME_CHANGE_STREAM.request
— MLME_CHANGE_STREAM.indication
— MLME_CHANGE_STREAM.response
— MLME_CHANGE_STREAM.confirm

— MLME_TERMINATE_STREAM.request
— MLME_TERMINATE_STREAM.indication
— MLME_TERMINATE_STREAM.response
— MLME_TERMINATE_STREAM.confirm

7.2.1.1 MLME_CREATE_STREAM.request

This primitive is used to request the creation of a stream.

7.2.1.1.1 Semantics of the service primitive

The parameters for this primitive are:

```
MLME_CREATE_STREAM.request    (
                              DEV destination address,
                              DEV source address,
                              Convergence sub layer id,
                              service flow parameters,
                              ARQ parameters,
                              length indicator,
                              sequence number
                              )
```

The DEV destination address is the 8-bit address with which the initiating DEV wants to establish a stream.

The DEV source address is the 8-bit address of the initiating DEV.

The convergence sub layer parameter indicates which convergence sub layer handles the data to be sent or received on this requested stream.

The service flow parameters list includes such as items as Traffic priority, Peak Rate, Minimum Rate, and Maximum Burst size. Also included are these parameters: inter_timeslot_duration, minimum_requested_channel_time, requested_channel_time, and maximum Tx delay variation.

The ARQ parameters are: ACK Policy (No ACK, Immediate ACK, Delayed ACK, Implied ACK), and Maximum ReTX duration.

The length indicator …

The sequence number is a unique 2-octet value created by the initiating DME to correlate this primitive with the response primitive it receives from the PNC MLME.

7.2.1.1.2 When generated

This primitive is generated by an initiating DME to initiate either a tri-partite negotiation (among two DEVs and their PNC) to establish a new stream between two DEVs, or a bi-partite negotiation (between a DEV and its PNC) to establish a new stream between a DEV and its PNC.

7.2.1.1.3 Effect of receipt

When a DEV MLME receives this primitive from its DME via the MLME-SAP, it generates a STREAM_CMD.request (actionType=request), which is sent to its corresponding PNC MLME. The PNC MLME upon receiving the STREAM_CMD.request (actionType=request) will send an MLME_CREATE_STREAM.indication to the PNC DME via the MLME-SAP. Similarly, when a DEV MLME receives a STREAM_CMD.REQUEST (actionType=request), it will send an MLME_CREATE_STREAM.indication to its DME via the MLME-SAP.

7.2.1.2 MLME_CREATE_STREAM.indication

This primitive is used to indicate a received "request to create a stream".

7.2.1.2.1 Semantics of the service primitive

The parameters of this primitive are:

```
MLME_CREATE_STREAM.indication (
                               DEV destination address,
                               DEV source address,
                               convergence sublayer id,
                               service flow parameters,
                               sequence number
                               )
```

See the parameters defined for MLME_CREATE_STREAM.request. The ARQ parameters list does not get passed up to the DME.

7.2.1.2.2 When generated

This primitive is sent by the non-initiating MLME to its associated DME upon receiving a STREAM_CMD.request (actionType=request) from the initiating DEV MLME.

7.2.1.2.3 Effect of receipt

When the PNC DME receives this primitive it will either send an MLME_CREATE_STREAM.request, or an MLME_CREATE_STREAM.response to its associated PNC MLME via the MLME-SAP. The MLME_CREATE_STREAM.request is sent to the MLME if the QoS parameters included in the MLME_CREATE_STREAM.indication can be satisfied. If the QoS parameters cannot be satisfied, then the MLME_CREATE_STREAM.response is sent to the MLME with an appropriate reasonCode.

When the destination DME receives the MLME_CREATE_STREAM.indication it will send an MLME_CREATE_STREAM.response with reasonCode to its associated MLME.

7.2.1.3 MLME_CREATE_STREAM.response

This primitive is used to initiate a response to an MLME_CREATE_STREAM.indication

7.2.1.3.1 Semantics of the service primitive

The parameters of this primitive are:

```
MLME_CREATE_STREAM.response (
                                stream_Index,
                               service flow parameters,
                               reason_Code,
                               sequence number
                               )
```

The stream_Index is a unique value assigned by the PNC DME to the requested stream.

The service flow parameters…

The reason_Code indicates whether the MLME_MODIFY_STREAM.request was successful or unsuccessful. If successful, the reason_Code is a zero hexadecimal value. If unsuccessful, the reason_Code is set to a non-zero hexadecimal code indicating the reason for the failure.

The sequence number is returned to the initiating DEV so that the initiating DEV may correlate this response with the appropriate request.

7.2.1.3.2 When generated

This primitive is generated by a non-initiating DME upon receiving an MLME_CREATE_STREAM.indication.

7.2.1.3.3 Effect of receipt

When the non-initiating DEV MLME receives this primitive from its associated DME, it generates a STREAM_CMD.response (actionType=Acceptance or Rejection), which is then sent to the PNC MLME.

When the PNC MLME receives an MLME_CREATE_STREAM.response from its associated DME, it generates a STREAM_CMD.response (actionType=Acceptance or Rejection), which is then sent to the initiating DEV MLME.

7.2.1.4 MLME_CREATE_STREAM.confirm

This primitive is used to confirm the acceptance or rejection of a request to create a stream.

7.2.1.4.1 Semantics of the service primitive

The parameters of this primitive are:

```
MLME_CREATE_STREAM.confirm    (
                              stream_Index,
                              service_flow_parameters,
                              reason_Code,
                              sequence number,
                              )
```

See the parameters defined for MLME_CREATE_STREAM.response.

7.2.1.4.2 When generated

This primitive is sent by the initiating MLME to its DME upon receiving a STREAM_CMD.response (actionType=acceptance or rejection).

7.2.1.4.3 Effect of receipt

In the case where a DEV to DEV stream is requested:

The PNC DME, when it receives this primitive with reason_Code indicating "acceptance or rejection" from its associated MLME, will send an MLME_CREATE_STREAM.response with appropriate reason_Code to its MLME. The PNC MLME will then send a STREAM_CMD.response (actionType=acceptance or rejection) to the initiating DEV MLME.

The source DME, when it receives the MLME_CREATE_STREAM.response primitive with reason_Code indicating "acceptance or rejection" from its associated MLME, is informed the stream it requested is either established or not established.

In the case where a DEV to PNC, or PNC to DEV stream is requested:

The PNC DME, when it receives this primitive with reasonCode indicating "acceptance or rejection" from its associated MLME, is informed the stream it requested is either established or not established.

The source DME, when it receives the MLME_CREATE_STREAM.response primitive with reasonCode indicating "acceptance or rejection" from its associated MLME, is informed the stream it requested is either established or not established.

7.2.1.5 Create Stream Message Sequence Charts

Figure 4 illustrates the sequence of messages involved in successfully establishing a DEV to DEV stream in a piconet.

(figure goes here, 15-3Tri-PartiteMSC.eps)

Figure 4 Tri-partite Message Sequence for Creating a DEV to DEV Stream

Figure 5 illustrates the sequence of messages involved in successfully establishing a DEV to PNC stream in a piconet.

(figure goes here,15-3Bi-PartiteMSC.eps)

Figure 5 Bi-partite Message Sequence for Creating a DEV to PNC Stream

7.2.1.6 MLME_MODIFY_STREAM.request

This primitive is used to request a modification to an existing stream.

7.2.1.6.1 Semantics of the service primitive

The parameters of this primitive are:

```
MLME_MODIFY_STREAM.request    (
                              stream_Index,
                              channel_time_parm_set
                              )
```

The stream_Index parameter indicates the stream to be modified.

The channel_time_parm_set includes these parameters:

- inter_timeslot_duration
- minimum_requested_channel_time
- requested_channel_time.

7.2.1.6.2 When generated

This primitive is generated by the DME to request a modification to an existing stream.

7.2.1.6.3 Effect of receipt

When a DEV MLME receives this primitive from its associated DME, it will generate a CHANNEL_TIME_CMD.request, which it will send to the PNC MLME. The PNC MLME upon receiving the CHANNEL_TIME_CMD.request will generate an MLME_MODIFY_STREAM.indication.

When a PNC MLME receives this primitive from its associated DME, it will generate a CHANNEL_TIME_CMD.request, which it will send to the DEV MLME. The DEV MLME upon receiving the CHANNEL_TIME_CMD.request will generate an MLME_MODIFY_STREAM.indication.

7.2.1.7 MLME_MODIFY_STREAM.indication

This primitive is used to indicate a received "request to modify a stream".

7.2.1.7.1 Semantics of the service primitive

The parameters of this primitive are:

```
MLME_MODIFY_STREAM.indication (
                                stream_Index,
                                channel_time_parm_set
                                )
```

See parameter definitions in Clause 7.2.1.6.1

7.2.1.7.2 When generated

This primitive is sent by the non-initiating MLME to its associated DME upon receiving a CHANNEL_TIME_CMD.request.

7.2.1.7.3 Effect upon receipt

Ed. note: need to define the effect upon receipt.

7.2.1.8 MLME_MODIFY_STREAM.response

This primitive is used to initiate a response to an MLME_MODIFY_STREAM.indication.

7.2.1.8.1 Semantics of the service primitive

The parameters of this primitive are:

```
MLME_MODIFY_STREAM.response (
                                stream_Index,
                                slot_start_time_set,
                                reason_Code
                                )
```

The stream_Index parameter indicates the stream requested to be modified.

Slot_start_time_set is the set of slot_start_times associated with the stream_Index. The slot_start_time is derived from the CTA info elements assigned to the stream with the indicated stream_Index.

The reason_Code indicates whether the MLME_MODIFY_STREAM.request was successful or unsuccessful. If successful, the reason_Code is a zero hexadecimal value. If unsuccessful, the reason_Code is set to a non-zero hexadecimal code indicating the reason for the failure.

7.2.1.8.2 When generated

This primitive is generated by the PNC DME upon receiving an MLME_MODIFY_STREAM.indication.

7.2.1.8.3 Effect upon receipt

When the PNC MLME receives this primitive from its associated DME, it generates a CHANNEL_TIME_CMD.response

7.2.1.9 MLME_MODIFY_STREAM.confirm

This primitive is used to inform the initiating DME whether the requested stream modification was successful or unsuccessful.

7.2.1.9.1 Semantics of the service primitive

The parameters of this primitive are:

```
MLME_MODIFY_STREAM.confirm    (
                              stream_Index,
                              slot_start_time_set,
                              reason_Code
                              )
```

See parameters defined for MLME_MODIFY_STREAM.response in clause 7.2.1.8 .

7.2.1.9.2 When generated

The initiating MLME sends this primitive to its associated DME after the MLME receives either a BEACON or an ACK_TIMEOUT.

7.2.1.9.3 Effect upon receipt

When the initiating MLME receives an ACK_TIMEOUT it will send an MLME_MODIFY_STREAM.confirm with a reasonCode = ACK_TIMEOUT to its associated DME.

When the initiating MLME receives a BEACON.indication with the requested change in CTA, it will send an MLME_MODIFY_STREAM.confirm to its associated DME.

7.2.1.10 Modify Stream Message Sequence

Figure 6 illustrates the message sequence involved when requesting a modification to an existing stream.

(figure goes here, 15-3ModifyStreamMSC.eps)

Figure 6 Message Sequence for Modifying a Stream

Figure 7 illustrates the message sequence involved when a requested stream modification for an existing stream is denied.

(figure goes here,15-3ModifyStreamDenied.eps)

Figure 7 Message Sequence for a Denied Stream Modification

7.2.1.11 MLME_TERMINATE_STREAM.request

This primitive is used to request the termination of a specific stream.

7.2.1.11.1Semantics of the service primitive

The parameters of this primitive are:

```
MLME_TERMINATE_STREAM.request (
                                        stream_Index
                                        )
```

The stream_Index parameter specifies the stream to be terminated.

7.2.1.11.2When generated

This primitive is generated by the DME to request the termination of an existing stream.

7.2.1.11.3Effect of receipt

When a DEV MLME receives this primitive from its associated DME, it will generate a STREAM_CMD.request (actionType=disconnection), which it will send to the PNC MLME. The PNC MLME upon receiving the STREAM_CMD.request (actionType=disconnection) will send an ACK to the initiating DEV and send an MLME_TERMINATE_STREAM.indication to its associated DME. Similarly, when a DEV MLME receives a STREAM_CMD.request (actionType=disconnection) it will send an ACK back to the PNC, and send an MLME_TERMINATE_STREAM.indication to its associated DME.

7.2.1.12 MLME_TERMINATE_STREAM.indication

This primitive is used to indicate a received "request to terminate a stream".

7.2.1.12.1Semantics of the service primitive

The parameters of this primitive are:

```
MLME_TERMINATE_STREAM.indication (
                                        stream_Index
                                        )
```

The stream_Index parameter specifies the stream to be terminated.

7.2.1.12.2When generated

This primitive is sent by a non-initiating MLME to its associated DME upon receiving a STREAM_CMD.request (actionType = disconnect).

7.2.1.12.3Effect of receipt

In the case where a DEV to DEV stream exists:

The PNC DME, upon receiving this primitive, will generate an MLME_TERMINATE_STREAM.request , which it will send to its associated MLME. The PNC MLME will then send a STREAM_CMD.request (actionType=disconnect) to the destination DEV MLME.

When the destination DME receives this primitive, it will terminate the indicated stream.

In the case where a stream exists between a DEV and a PNC:

The PNC DME, upon receiving this primitive, will terminate the indicated stream.

The destination DME, upon receiving this, will terminate the indicated stream.

7.2.1.13 MLME_TERMINATE_STREAM.confirm

The primitive informs the initiating DME whether the requested stream termination was successful or not.

7.2.1.13.1Semantics of the service primitive

The parameters of this primitive are:

```
MLME_TERMINATE_STREAM.confirm (
                                    stream_Index
                                    reason_Code
                                    )
```

The stream_Index parameter indicates the stream to be terminated.

The reason_Code indicates whether the MLME_TERMINATE_STREAM.request was successful or unsuccessful. If successful, the reason_Code is set to a zero hexadecimal value. If unsuccessful, the reason_Code provides a non-zero hexadecimal value indicating the reason for the failure.

7.2.1.13.2When generated

The initiating sends this primitive DEV MLME to its associated DME after the DEV MLME receives either an ACK to its STREAM_CMD.request (actionType=disconnect) or its ACK_TIMEOUT has expired.

7.2.1.13.3Effect of receipt

If the initiating DME or the PNC DME receives this primitive with the reasonCode indicating "DISCONNECT" from its associated MLME, it will terminate the indicated stream. If the initiating DME or the PNC DME receives this primitive with the reasonCode indicating "ACK_TIMEOUT" from its associated MLME, it will resend an MLME_TERMINATE_STREAM.request to its associated MLME. The MLME will then resend a STREAM_CMD.request (actionType=disconnect).

7.2.1.14 Terminate Stream Message Sequence Chart

Figure 8 Illustrates the message sequence involved when terminating a DEV to DEV Stream in a piconet.

(figure goes here,15-3TerminatingStreamMSC.eps)

Figure 8 Message Sequence for Terminating a DEV to DEV Stream

Ed. action: These sub-clauses replace 6.3.10, 6.3.11 and add 6.3.12. The primitive parameter names were changed to be consistent with the rest of the Layer Management clause. Still missing many definitions for the parameters.

10.8 QoS Policies, Clause 9

9. Quality of Service (QoS) Policies

The requirements for QoS include:

a) An association procedure during which pre-configured default Service Flows and their associated QoS parameters are applied.
b) A link management protocol for dynamically establishing custom Service Flows with appropriate QoS parameter values.
c) A scheduling algorithm, which uses Service Flow QoS parameter values to allocate piconet resources when establishing uplink, downlink, or peer link connections.

The principal mechanism for addressing the QoS requirements in a WPAN environment is the assignment of received upper protocol layer packets to an appropriate Stream-ID/Service Flow pair possessing the required QoS characteristics. These QoS characteristics of the Stream-ID/Service Flow pair are specified in the QoS Parameter Set associated with the Service Flow.

The elements of the QoS Parameter Set are: Traffic Priority, Peak Rate, Minimum Rate, and Maximum burst size. These elements are used by the PNC and DEVs of a piconet to schedule the transmission order of packets accessing the wireless medium.

9.1 Service Flow

A Service Flow is an SSCS service that is used to define the transport characteristics of an uplink (DEV->PNC), downlink (PNC->DEV), or peer link (DEV->DEV) connection.

Some attributes of a Service Flow are:

a) Service Flow ID (SFID) : a unique binary value, TBD bits in length, assigned by the PNC to each Service Flow.
b) Stream-Index: is a unique value, TBD bits in length, used to identify a connection and to associate it with an active Service Flow.
c) ProvQoSParmSet: defines the set of QoS parameters associated with a Provisioned Service Flow.
d) ActiveQoSParamSet: defines the set of QoS parameters that are now available for use by a specific Stream-Index.

9.1.1 Types of service flows

This section defines two basic types of Service Flow: Provisioned, and Active.

9.1.1.1 Provisioned service flow

A provisioned Service Flow is one, which has been assigned an SFID and has not yet been assigned a Stream-Index by the PNC. In addition, the PNC has yet to reserve the QoS resources specified in the ProvQoSParmSet. The purpose of a provisioned Service Flow is to enable WPAN DEVs in a piconet to have access to predefined QoS parameters. The benefit of which is a more efficient initialization process for a pre-defined Service Flow.

For instance, a DEV could request activation of a provisioned Service Flow by sending the PNC the SFID for a known provisioned Service Flow. The PNC would then, if piconet resources were available, respond with both the requested SFID and newly assigned Stream-ID. This mapping of the Provisioned Service Flow's SFID to a Stream-ID will now define an Active Service Flow.

Similarly, a PNC could activate a provisioned Service Flow by sending a DEV the SFID of a provisioned Service Flow, its associated QoS Parameter Set, and a Stream-ID to which the SFID has been associated. The DEV would then, if it has the internal resources available, respond in the affirmative.

9.1.1.2 Active service flow

An Active Service Flow is a one, which has been assigned both an SFID and a Stream-ID. In addition, an Active Service Flow is one, which has a non-NULL set of ActiveQoSParameters, which indicates that the PNC has reserved the piconet resources requested.

An Active Service Flow is the result of either a Provisioned Service Flow being assigned an Stream-ID or a Service Flow being created from scratch (i.e. QoS parameters being negotiated between the PNC and DEV or among a DEV, PNC, and destination DEV) before becoming active.

10.9 Bibliography

[B1] G. Xylomenos and G.C. Polyzos, "Link Layer Support for Quality of Service on Wireless Internet Links," IEEE Personal Communications, vol. 6, no. 5, Oct. 1999, pp. 52-60.

[B2] IEEE Draft 802.16/D4-2001. IEEE Draft Standard for Local and Metropolitan Area Networks - Part 16: Standard Air Interface for Fixed Broadband Wireless Access Systems.

[B3] QoS Forum, "White Paper - QoS protocols and architectures," Stardust.com, Inc., Jul. 1999

[B4] QoS Forum, "White Paper - Introduction to QoS Policies," Startdust.com, Inc., Jul. 1999

[B5] QoS Forum, "Quality of Service - Glossary of Terms," Stardust.com, Inc., May 1999

Ed. action: text added as indicated.

11. Corrections sugggested by Wim VanHouton

(all references to D07)

o) On page 133 line 7 you wrote: ".....the first symbol of the PHY header....." but it should be at my opinion ".....the first symbol of the PHY preamble....."

Changed as indicated in D08.

oa) On Page 133 line 24 you use the word MPDU, this is not by my opinion, the commonly used terminology within the 15.3 standard.

Changed to “The maximum frame length allowed” in D08.

ob) On page 133 line 25 you wrote MAC Header or HCS, but Figure 3 on page 66 shows that HCS is part of the MAC header.

“or HCS” deleted in D08, text now says “... not the PHY preamble, PHY header or MAC header.”

i) On page135 line 49 .....above is given... should be at my opinion......above are given......

Changed to “... above are given ...” in D08.

ii) On page 137 line 10 till 12: The higher.........to right. This sentence is in my opinion not adding a lot, I have to be honest. I actually do not understand the meaning of this statement. Should it be possible to leave this out and only writing: The decimal representations of the bitmapping.......... (line 13 and further)

This is an informative comment to explain why we chose this mapping. However, we will try to improve the explanation in D09.

iii) On page 140 line 45 Section 11.3.4 is called QPSK modulation but if I read further it is actually Differential QPSK (DQPSK), is it not sensible to talk about DQPSK instead of QPSK also for BPSK in the whole document? I can not clearly ditinguish what is differntial modulated and what is not...........

Change QPSK to DQPSK in PHY, BPSK will changed to QPSK-TCM in either D08 or D09.

iv) On page 140 line 48 regarding differential encoding What is the starting dibit for the DQPSK modulation (the reference dibit for the first phase change....)

Added sentence in D08 to clarify the issue: "The phase change of the first symbol is determined relative to the phase of the last symbol in the CAZAC sequence, 11.4.6."

iva) On page 140 line 53 you stated that differential encoding gives you the possibility for non coherent reception, but the CAZAC sequence (PHY preamble) is not differential encoded so you need a coherent "like" reception anyway (carrier offset compensation, equalization), don't you agree.....?

To get the most information out of the CAZAC sequence, you would do soft decisions with a coherent receiver. However, you can still get useful information (particularly AGC and packet timing) with differential detection. We discussed this particular issue before adopting the differential encoding. No change made, request the commenter withdrawn his objection.

v) In Table 74 on page 141 there is written pi in stead of \pi.

Changed to \pi in D08, thanks.

va) On page 141 line 23 you wrote PHY header , MAC header and HCS but in Figure 3 on page 66, the HCS is shown as part of the MAC header. Do you mean FCS or uses this field the same modulation as the Frame body?

Changed to "PHY header and MAC header" in D08.

vb) On page 141 line36 you also talk about MAC header and HCS.......

The text ", HCS" deleted in D08.

vi) The header of Table 75-Scramber seed selection should be Table 75-Scrambler seed selection.

"Scrambler" spelled correctly in D08.

vii) On page 143 line 3 you use octets in stead of bytes, why?

Octets is a carry over from 802.11 where they may have picked it up from someone else. The rest of the draft uses octets instead of bytes, so I have changed all occurances of bytes to octets in the PHY clause as well. Change completed in D08.

viii) In Table 76 you miss the reference Table 75 for the dibits b0-b1.

Cross reference to the scrambler seed selection table added to D08 (xref is to table 75)

ix) The content field of Table 76 for b5-b15 there is written Freme body length and this should be Frame body length

"Frame" correctly spelled in D08

x) In Table 77 is not clear if b2 is the most right bit or the one just to the right of the x.

The table now has 5 columns, the last three are for b2, b3 and b4, to clarify the mapping.

xi) On page 143 line 44 you talk about the end of the message payload, but I did not saw the definition in the frame format. In my opinion you should use something like the end of a packet after the FCS.

Text is changed to “... added to the end of frame body, i.e. after the FCS, ...” in D08.

xii) On page 144 line 48 and further, you talk about tail symbols for QPSK and BPSK, but in this mode no-trellis encoder is used, why do you need tail symbols then?

The reason is that it allows soft symbol detection of DQPSK modulation in a clean manner with a very tiny impact on the overhead. While this is not required, an implementer could use this technique to improve performance of the modem. Do you think a description of why it was chosen will be useful?

xiii) On page 145 line 22 you use: "....added before the message payload" but, by using the same terminology as in Chapter 7 and section 11.4.2 of the standard, it should be in my opinion something like: "A PHY preample shall be added before the PHY Header, MAC header......."

Good catch, the preamble goes in front of the PHY header, not the message payload. The text reads “... shall be added prior to the PHY header ...” in D08 now.

xiv) On page 148 Table 80 there is no EVM for BPSK

With the change of BPSK to QPSK-TCM, the EVM for DQPSK will apply to QPSK-TCM and BPSK EVM will not be necessary.

xv) On page 148 line 17 you wrote 6 MHz of the center frequency, is this -3 MHz and +3 MHz or -6 MHz and +6 MHz or only + 6 MHz?

Changed text to read “... measured within +/- 6 MHz of ...” in D08

xvi) On page 148 line 12 the transmit mask is specified, why is the transmit mask specified and not the the type of filtering at the transmitter and receiver i.e. raised cosine with a certain roll-off.....?

We discussed this at various meetings. The reason is that the filter is an implementation detail whereas the spectral mask directly affects the adjacent channel performance (regardless of the type of the filtering). There was a proposal to provide a recommended filter, but that text has not yet been approved. Note that it is not sufficient to specify a raised cosine filter with excess bandwidth parameter since an implementation must use an approximation to the filter. The recommended filter should specify what approximation is sufficient.

xvii) On page 149 line17 you wrote: "....the emissions conform with...." it should be at my opinion "....the emissions are conform with...."

Changed “... the emissions should also conform with the ...” to “... the emissions should also conform to the ...” in D08.

xix) On page 150 line 5 till 9 you use MPDU and MAC Header and the HCS, same remarks as above...

Changed “... only the MPDU of 1024 bytes, but also the PHY preamble, PHY header, MAC header and the HCS.” to “... only the frame body of 1024 octets, but also the PHY preamble, PHY header, MAC header and FCS.” in D08.

Also changed “... with an MPDU length of 1024 bytes ...” to “... with an frame body length of 1024 octets ...” in the prior paragraph in D08.

xx) On page 150 line 16 you wrote: "sensitivity in listed in Table 83" it should be at my opinion "sensitivity listed in Table 83"

Extra “in” removed, now reads “... reference sensitivity listed in Table 83 ...” in D08.

xxi) On page 150 line 47 you wrote: "......is the alternate channels Channel......." it should be at my opinion "......is the alternate channel. Channel......."

Changed as indicated in D08.

xxii) On page 150 line 53 you wrote "......one of the four modulation......." it should be at my opinion "......one of the five modulation......."

Changed as indicated in D08.

xxiii) On page 151 the field in the right lower corner of Table 84 should be 16 dB i.s.o. 16 db.

Capitalization of dB fixed in D08.

xxiv) On page 151 line 29 and line 38 interferring should be at my opinion interfering.

Spelling on line 29 corrected (it was spelled correctly on line 38) in D08.