IEEE P802.15
Wireless Personal Area Networks

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)	
Title	**TG3 LB12 St. Louis comment resolution**	
Date Submitted	[7 May, 2002]	
Source	[James P. K. Gilb] [Appairent Technologies] [9921 Carmel Mountain Rd. #247, San Diego, CA 92129]	Voice: [858-538-3903] Fax: [858-538-3903] E-mail: [gilb@ieee.org]
Re:	[]	
Abstract	[This document is an additional record of comment resolution of LB12 completed in St. Louis.]	
Purpose	[To provide a record of comment resolution, particularly for comments that are resolved based on the resolution of prior comments.]	
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. Comment resolution order

1.1 Results of email comment resolution

Forgot suggested resolution of 1294:

1294 (Roberts, TR): Mention is made of a trusted relationship. The standard must explain how that trusted relationship is established. Suggest accept in principle, “The 802.15.3 committee is going to issue a CFP, evaluate and choose a mandatory cipher suite for DEVs that implement security.”

Comment numbers 35, 34, 400, 401, 403, 404, 405, 406, 407, 409, 410, 411, 412, 415, 416, 438, 440, 441, 760, 1184, 1187, 1189, 1194, 1201, 1207, 1208, 1211, 1215, 1217, 1218, 1219, 1279, 1280, 1281, 1283, 1285, 1286, 1287, 1288, 1289, 1294, 1594, 1592, 1595, 1598, 1599, 1625, 1626, 1627, 1630, 1654, 1726, 1727. See 02/075r15 for resolutions.

Accept.

1.2 Resolution in St. Louis

1.2.1 Miscellaneous (1 unresolved)

301 (Gilb, T): Need a figure to show how the bit ordering is used in the figures that follow. Add the figure once it has been generated and reviewed. Figure should have multiple fields with LSb and MSb indicated for each of the fields, an indication of the order in which they are sent over the air and an example of a simple command or information element with specific values. Suggest accept in principle, “A figure will be created that indicates the bit ordering in the fields and in the overall frame. Also include some sample frames with the associated bits to the informative annex B with a cross reference to them in clause 7.”

Accept.

1628 (Shvodian, TR): Need to add a text on how asynchronus data will be efficiently handled. Need to specify how asynchronous data will be handled in a scheme that is both power and bandwidth efficient.

Suggest accept, “The text on how to handle asynchronous data will be added based on the text in 01/469r4.”

56 (Bain, T): Is there a case where an open MTS is less than one per superframe? If so, is there appropriate wording to change the responsivness of the PNC to requests for change. I believe that up to 4 superframes may pass from CTR till CTA reflecting the change. The 4 superframe lag is long already. It should not go beyond that. put the appropriate SHALL to keep the lag from CTR (or Stream) till CTA from getting higher than 4 superframes when MTS is used. Suggest that Jay make a better suggestion.

Text from Jay Bain, “pg 151 line 6, ‘For PNCs allocating open MTSs at less than the superframe rate, the PNC shall remain obligated to respond to a channel time request within four superframes from when the requesting DEV would have made the request had the open MTS been present.’

Suggest reject, “The PNC is not obligated to respond to a CTR within 4 superframes. If the piconet is already busy, the PNC will hold the CTR until channel time is available. Also, since the PNC is in charge of allocation uplink and downlink MTSs, it is able to allocate the necessary MTSs for either stream management or channel time requests.”

Accept in principle:

Add the following text either to 8.4.3.2 or a new section 8.4.3.3 Responsiveness of PNC

'The PNC should respond to a channel time request, {xref 7.5.x}, within CTRRespTime number of superframes. The CTRRespTime parameter is communicated to the DEVs in the piconet via the association command response command, {xref 7.5.x}, and the new PNC announcement command, {xref 7.5.x}. The PNC is only required to give a response to the requesting DEV. The response may be a rejection of the request due to capacity or other issues with the piconet.

The PNC should respond to a request to change to a predefined SPS CTA, {xref 7.5.x}, within SPS-RespTime number of superframes. The SPSRespTime parameter is communicated to the DEVs in the piconet via the association command response command, {xref 7.5.x}, and the new PNC announcement command, {xref 7.5.x}. The PNC is only required to give a response to the requesting DEV. The response may be a rejection of the request due to capacity or other issues with the piconet.

In the case of either a CTR or a change to an SPS CTA, the requesting DEV would not expect the PNC to respond unless the DEV has received an ACK to the request.

A DEV may also request that the PNC adjust the repetition rate of open MTSs for the piconet using the channel time request command, {xref 7.5.x}. The PNC may either grant or reject the request.'

7.ww add to the Channel Time Request command

Add the field to the channel time request frame format, 1 bit, Open MTS request

'The open MTS request is a single bit that designates that this channel time request is requesting a specific characteristic of the open MTS repetition rate. If the value is 0, then this CTR is normal. If the value is 1, then this CTR is to request open MTS specification.'

Change CTR Control field such that CTR type is 2 bits.

'The CTR Type field indicates the type of request:

— 0 The request is for an ACTIVE time allocation.
— 1 The request is for an SPS time allocation.
— 2 The request is to change the open MTS repetition rate.
— 3 Reserved.'

7.5.x add to New PNC announcement message

Add PNC response field to the frame format, 1 octet in length. Add the following to the text.

'The PNC response field shall be formatted as illustrated in Figure 1.

bits: 0-3	4-7
CTRRespTime	SPSRespTime

Figure 1—PNC response field format

The CTRRespTime field indicates the maximum number of superframes that the PNC will use to provide a respone to a channel time request by a DEV, {xref 8.x.x}.

The CTRRespTime field indicates the maximum number of superframes that the PNC will use to provide a respone to a PS information command by a DEV, {xref 8.x.x}.'

7.xx add to association command response

Add 1 octet to the frame format, PNC response field and add the following text:

'The PNC response field is defined in {xref 7.5.x.}'

Make mods to clause 6 to match new frame formats for New PNC announcement and association response.

1.2.2 Gap with no CAP

1346 (Schrader, T): The text should elaborate on the Non-CAP case, especially with regards to processing time for the beacon. If the CAP is not present there must be a gap or unallocated time slot allocated to allow all deviecs to process the information in the beacon. If the amout time is not specified, a PNC may assign slots before a device can interpret its CTA. Indicate in the text that a minimum size CAP will be assigned even for the MTS only case, where the CAP will serve only as a gap between the Beacon and the GTS slots. Suggest accept in principle, "Indicate in the text that the PNC shall not allocate any time slots within aNew-Parameter time of the end of the beacon."

Accept in principle, "Add text to 8.4.3.2 that says 'The PNC shall not allocate any MTS or dynamic GTS within xref aFirstGTSGap following the beacon except with the PNC DEV as the source.' Add to table 73 a parameter aFirstGTSGap with a value of 100 us."

57 (Bain, T): In the absence of CAP, the first GTS is bumped against the end of the beacon. The development of real implementations of this standard may be hindered if the parsing of the beacon body must occur in the very few microseconds available. Provide guidance to implementers but also place a minimum time till the beginning of the first GTS in the absense of CAP. If a PNC to DEVs MTS is always present, then this would not be a problem. Suggest accept in principle, "Resolve as indicated in 1346."

Accept in principle, "Resolve as indicated in the resolution comment 1346."

1.2.3 Del-ACK and fragmentation (3 unresolved)

1564 (Shvodian, TR): Fragmentation and defragmentation should not be in the MAC. It will overly complicate the designs and require large per-stream buffers. Fragmentation should be done at the convergence layer. Move Fragmenation to the convergence layer. Suggest reject, "Fragmentation is required for commands as well as data. The standard specifies the logical requirements of the protocol, not the archictecture. Regardless of where the functionality is described in the draft, the implementer is free to put that functionality anywhere in the final architecture. For example, if fragmentation/defragmentation is the first/last things done to data in the MAC (which is up to the implementer to design), then it could easily reside with either the MAC or the convergence layer in the implementation. However, while it could be put into either, there are three reasons why fragmentation belongs in the MAC: 1) fragmentation is required for commands, which generally do not travel to the convergence layer. 2) Fragmentation can be used to overcome difficulties in the channel, e.g. interferers. The channel information is monitored by the DME, not the convergence layer. 3) The fragmentation specification needs to be the same for all of the potential convergence layers or else it would not be possible to send from one DEV that is using one CL while the receiver is using another CL. Placing identical definitions for the same functionality in more than one location in the draft will lead to incompatibilities, make it difficult to maintain the standard and make it harder to implement more CL's."

Table until Monday afternoon.

1464 (Shvodian, T): Get rid of Delayed ACK. This will unnecessarily complicate the MAC to implement. We should keep a WPAN as simple as possible. Eliminate Delayed ACK. Suggest reject “The use of delayed ACK greatly increase the throughput, particularly at higher data rates. Because of this, the task group feels that the added complexity is justified by the increased throughput.”

Accept, resolution is to reject.

1151 (Roberts, T): In general, this paragraph deals with sending the delayed ack. The sentence at line 18 that beguns "However the recipient ..." since it implies a DEV should send a delayed ACK at the expiration of the retransmission window. My question is this ... do you sent this at the expiration only if not previously send or do you send it again regardless. MAC committee to comment and clarify text if necessary.

Table, waiting on the 1564.

1570 (Shvodian, TR): DEVS cannot reject a delayed ACK. This must be from when we were calling them "retransmission requests" Change the paragraph as follows: "When Del-ACK is used for a stream, the DEV transmitting the data frame may abort retransmssion attempts once the negotiated retransmission window for the stream has been reached. When retransmissions are aborted, the DEV transmitting the stream shall send transmission sequence sync command, as defined in 7.5.9.2, to the recipient of the stream in order to synchronize the delayed-ACKs."

Table, waiting on the 1564.

1.2.4 Tripartite negotiation

1434 (Shvodian, TR): Eliminate tripartate negotiation. bipartate negotiaon between the PNC and DEV is all that is needed.

Accept in principle, “The new stream negotiation will be added as indicated in document 01/469r4.”

1120 (Schrader, T): This comment supplements as suggestion for adding stream management command in section 7.5.10.3, page 133 for final confirmation or acceptance of stream connection. One of the primary purposes of the stream connection process is to determine if the originator and the target agree on a single set of QoS parameters. As currently proposed the communication flow is: Originator->PNC->Target->PNC->Originator. The originator will then reply to only to the PNC only if it rejects the Targets modified QoS values. The trigger that starts PNC generation of time slots should be a response from the Originator to the PNC conThe stream connection process involves the PNC to determine if it can provide the GTS slot allocation requested, and the two peers must agree on a set of QoS parameters. As currently proposed the communication flow is Originator->PNC->Target->PNC->Originator. The originator will then reply to only to the PNC if it rejects the Targets modified QoS values. The trigger for PNC generation of time slots should be a response from the Target to the PNC confirming acceptance of the final QoS parameters relayed from the Target, not the absence of any negative response. The following is a rewrite of lines 12-27 on page 153. Either the sending DEV or the intended recipient DEV for the new stream may send a stream management command with the request for stream connection. The process of stream connection is illustrated in Figure 88. In this figure, DEV A is the originator of stream connection request and DEV B is the target, consistent with the stream management command section 7.5.10.3. In all stream management communications from the PNC to the other involved DEV, the PNC appropriately changes the value of the direction field to imply the same direction of the stream as originally requested. The values for direction, security, stream type and priority shall be non negotiable and are decided by the DEV A that is sending the stream connection request. These values shall not be changed anytime after the first transmission of the command frame containing the request for that stream. The target DEV B responding to the forwarded stream connection request may modify the remaining QoS parameters including bandwidth and latency requirements. All the bandwidth and

latency related requirements of the stream shall be confirmed or rejected by the originator of the stream connection request in response to the final PNC acceptance message. The PNC decision on the values of the stream QoS parameters that are supported in the piconet shall be final. If the originating DEV A does not accept the PNCs final stream parameters, then DEV A shall send a stream management command to the PNC with action type set to disconnection/rejection as specified in 7.5.10.3. Then the PNC shall then send a stream management command to the target DEV B with action type set to disconnection/rejection. Otherwise, DEV A shall send a stream management command with a final confirm/accept action type, and the PNC shall then begin generating ACTIVE type CTA elements and GTS timeslots as specified upon receipt of this command. Suggest that this comment wins the prize for longest comment.

Suggest accept in principle, “Change the paragraphs ‘Either the sending ... set to disconnection/rejection’ in lines 12-27 on page 153 to read

> ‘The source DEV for the new stream may send a stream management command with the request for stream connection. The process of stream connection is illustrated in Figure 88. In this figure, DEV A is the originator of stream connection request and DEV B is the target of the stream management command. In addtion, the originator the command is also the souce of the stream while the target of the command is the destination of the stream.
>
> The values for direction, security, stream type and priority shall be non negotiable and are decided by the originating DEV. These values shall not be changed anytime after the first transmission of the command frame containing the request for that stream.
>
> The target DEV responding to the forwarded stream connection request may modify only the receive window size. Any change the receive window size of the stream shall be confirmed or rejected by the originator of the stream connection request in response to the final PNC acceptance message. The PNC’s decision on the values of the stream parameters that are supported in the piconet shall be final.
>
> If the originating DEV does not accept the PNC’s final stream parameters, then the originating DEV shall send a stream management command to the PNC with action type set to disconnection/rejection as specified in xref 7.5.10.3. Then the PNC shall then send a stream management command to the target DEV with action type set to disconnection/rejection. Otherwise, the originating DEV shall send a stream management command with a final confirm/accept action type. The PNC shall then allocate the GTSs via CTAs in the beacon.’”
>
> Accept in principle, “The stream negotiation will be rewritten as indicated in document 01/469r4.”

378 (Gilb, T): Since the direction field is constant during the negotiation (since the target and originator addresses are now included), this sentence is incorrect. Delete the sentence "In all stream management ... as originally requested." Suggest accept in principle, “Resolve as indicated in the resolution of comment 1120.”

> Accept in principle, “Delete the sentence ‘In all stream management ... as originally requested.’ and all other references to the direction field in the draft.”

1.2.5 Stream managment cleanup

1563 (Shvodian, TR): "All the bandwidth and latency related requirements of the stream shall be negotiated between the sender of the stream and the PNC. The PNC decision on the values of the stream QoS parameters that are supported in the piconet shall be final." The PNC knows nothing about bandwidth. It only knows about channel time. I don't think these QoS parameters should be negotiated at the MAC, but if they are the other DEV should have the oportunity to negotiate the values down.

Accept in principle, “The stream negotiation will be simplified as indicated in document 01/469r4.”

602 (Heberling, TR): The MLME-CHANNEL-TIME.request, indication,response and confirm are missing. Please insert clauses 6.xxxx from 01/410r1 into the space just before clause 6.3.13 Stream creation.

Accept in principle, “The functionality of MLME-CHANNEL-TIME.xxx has been replaced by MLME-CREATE/MODIFY/TERMINATE-STREAM.xxx as indicated in document 01/469r4.”

725 (Heberling, TR): The Channel Time request command is inadequately defined for the functions required of it in this protocol. The Channel Time request command clause in doc 02/037r0 provides detailed resolution to this issue. Suggest accept when 02/037r0 is ready for review.

Accept in principle, “The channel time request command will be defined as indicated in document 01/469r4.”

726 (Heberling, TR): The Stream managment command is an inordinately complicated frame command for the functions it is needed in this draft. Replace the Stream managment command with the upgraded Channel Time request command described in doc 02/037r0. Suggest accept when 02/037r0 is ready for review.

Accept in principle, “The stream managment command functionality will be replaced by the channel time request and channel time status commands as indicated in document 01/469r4.”

1716 (Song-Lin, TR): It is confusing that this command seems suggesting a DEV seeking to communicate with target DEV needs to use this command, even if after a stream connection has been established. While CTA for one stream is assigned at the end of stream conection (Fig.3). Clarify if this command is used in conjunction with stream management command for establishment of communication and required for allocating time slots for the stream. Suggest accept when 02/037r0 is ready for review, “Documents 02/100r3, 02/115r1 and 02/037r0 will be used to modify the stream management and channel time request commands. The commands will be merged into one set of commands to allocate time and to manage streams.”

Accept in principle, “The functionality channel time request command and the stream management commands have been merged into a single command, called the channel time request command. This command is the only one used to request or modify asynchronous or isochronous streams. This change is documented in 01/469r4.”

1.2.6 PNC selection process.

670, 704, 723, 724 (Heberling, TR): PNC selection, request to change the previously accepted process, see document 02/037. (tagged PNC selection).

Accept in principle, “Change the PNC selection process to the one described in document 02/037r2. Delete other references to the PNC selection process. Move Table 68 and associated text to the coordination handover subclause.”

174 (DuVal, T): Diagram hard to read. How does this diagram relate to the previous paragraph? Where are the terms aCSFrameRepeat and aCSFrameBroadcast in this diagram? I would like to see their timing relationships.

Accept in principle, “Change the PNC selection process to the one described in 02/037r2. Delete other references to the PNC selection process. Move Table 68 and associated text to the coordination handover. Because of this, figure 78 will be deleted.”

1526 (Shvodian, T): Why bother with PNC selection at all? Now that we can do handover if a better PNC shows up, Just wait a random time and start sending out beacons. This would be a much simpler process.

Also, the odds of turning on a bunch of machines all at the exact time is small. Eliminate PNC selection and simplify by just waiting a random amount of time then start sending out beacons. Then, handover if more qualified PNC.

> Accept in principle, “Change the PNC selection process to the one described in document 02/037r2. Delete other references to the PNC selection process. Move Table 68 and associated text to the coordination handover subclause.”

1.2.7 Piconet shutdown

1529 (Shvodian, TR), 597 (Heberling, T): Piconet shutdown element. (tagged PiconetShutdown)

> Suggest accept in principle, Add a new sub-clause to 8.2 (probably 8.2.8) named “Shutting down a piconet” with the following text. “A piconet ends when the PNC shuts down, either abruptly or in a controlled manner. If the PNC wishes to shut down and there are no AC capable DEVs in the piconet, the PNC should disassociate all of the DEVs in the piconet with the reason code that indicates that the piconet is shutting down, xref 7.5.2.3.”

> Accept in principle, “Adopt the PNC shutdown element as defined in 02/037r1, add this to the beacon information element list and add text to clause 8 that describes that this should be sent at least aMinBeaconInfoRepeat times in the beacon before shutting down the piconet.”

1.2.8 Power management

1185 (Roberts, T): We talk about EPS sets ... but this is still vague. Perhaps a figure should be added to illustrate the concept. This figure should go into clause 8.13.3.1. The figure can be generated with help from the Power Management subcommittee. Suggest accept in principle, “A figure that illustrates the relationship between the WAKE beacon and WAKE superframe will be added based on document 02/115r1.”

> Accept

41 (Bain, T): Should provide clarification on recovery from incorrect beacon. Add: This may as simple as dealing with a PNC that is checking another channel for better rf conditions. It may also be that the PNC has changed the superframe duration while a DEV was not awake. Procedures outside of EPS power management process are used to recover. Suggest accept.

> Accept in principle, “Resolve as indicated in the resolution of comment 1210”

1210 (Roberts, TR): Line 10 indicates that the EPS DEVs recognize an error condition. How is this done? Via Table 2 ReasonCode? The ReasonCodes in Table 2 are not defined. Refer to Power Management folks.

> “Add text 8.13.3 that says ‘An SPS DEV recognizes that there is an error condition when it does not receive the beacon that was expected. The DEV will take different actions depending on the setting of the MACPIBPowerManagementRecovery. If the PIB value is set to 0, then the DEV shall simply wait until its next scheduled AWAKE beacon time before it listens for the beacon again. If the PIB value is set to 1, then the DEV shall attempt to re-synchronize with the piconet.’ Also add a new MAC PIB element, Managed element = MACPIBPowerManagementRecovery, number of octets = 1, Definition = ‘Determines what action to take when a beacon is missed by an SPS DEV, as defined in xref 8.13.3, type = dynamic.”
> Accept in principle with the above resolution.

1213 (Roberts, T): Line 12 refers to a "recovery operation" Where is this recovery operation described. Supply reference clause. Suggest accept in principle, “Resolve as indicated in the resolution of comment 41.”

Accept in principle, “Resolve as indicated in the resolution of comment 1210.”

1633 (Shvodian, TR): The beacon count of the next awake beacon for every member of the EPS set will not be synchronized. By the time the requestor gets the information, the Beacon pointed to may already have passed. The requesting DEV will think that it has to wait for 2^16 cycles of the beacon counter. Find a way to eliminate the need to use a frame request that ties to real time data like a beacon number.

Accept in principle, “The next AWAKE beacon parameter has been extended to 4 octets in all cases and the DEV is able to determine if the beacon has passed and using SPS interval determine the beacon when the DEVs in the SPS set will next be AWAKE. This is documented in 02/118r0 and will be added to the draft.”

Accept

1.2.8.1 Broadcast traffic in SPS and APS modes.

1584 (Shvodian, TR): How is broadcast traffic handled when a device is in EPS mode? Is the PNC forced to perform repeater service to every DEV that is in EPS. Need to decide how to handle broadcast traffic when devices are in EPS mode since TCP/IP uses broadcast.

1209 (Roberts, T): In line 170, reference is made that the "PNC shall save these for the next superframe". Does this mean these have to be buffered by the PNC? Refer to the power management folks. How is the size limit of this buffer specified?

Options

a) PNC buffers all broadcast frames and forwards them when the SPS sets are awake
b) The source of a broadcast frame repeats it in all of the superframes necessary to communicate with all of the SPS sets.
c) The PNC has a broadcast SPS set for which all SPS DEVs are also a member. Broadcast is done when the broadcast SPS set is AWAKE.
   Add text “SPS set 0xF is the broadcast SPS set. The PNC sets the parameters of this set. SPS DEVs may wake up to listen to broadcast frame during the AWAKE beacons of this set. Likewise, DEVs that wish to send broadcast messages to SPS DEVs should send those frames during the AWAKE beacon for the broadcast SPS set.”
d) Broadcast is not supported when there is one SPS DEV in sleep mode.
e) No provision made to ensure that SPS DEVs receive broadcast messages.

Add text “Note that DEVs in SPS SLEEP state may not receive broadcast messages. If a source DEV wants the SPS DEVs in SLEEP state to receive the broadcast frame, then the source DEV should send the frame in all of the superframes required to reach the SPS DEVs in their AWAKE beacons.

For comments 1584 and 1209, Accept in principle, “Add text to an appropriate place in subclause 8.13 ‘Note that it is possible that DEVs in SPS mode will not receive broadcast messages. If a DEV wishes to have the opportunity to receive all of the broadcast streams, it should not use SPS mode. If a source DEV wants the SPS DEVs in SLEEP state to receive the broadcast frame, then the source DEV should send the frame in all of the superframes required to reach the SPS DEVs in their AWAKE beacons.’”

1.2.8.2 General comments on PM

1724 (Rofheart, TR): The power management method is overly complex and vague. Refer to the remedy indicated by Bill Shvodian. “The power management section is going to be rewritten based on proposals 01/384r2, 02/067r1, 02/118r1, 02/100r4, 01/469r4 and the minutes.”

Accept, Rofheart confirmed via email.

1629 (Shvodian, TR): A complex power management solution has been specified, but we need a very simple approach that will be useful for very low power devices that are not. If the current apprach stays it sould be optional, and we should add a low-complexity solution as an alternative. I will be proposing a low complexity power management solution. Suggest accept in principle, “The power management section is going to be rewritten based on proposals 01/384r2, 02/067r1, 02/118r1, 02/100r4, 01/469r4 and the minutes.”

Accept.

1.2.9 SIFS, range or value?

1567 (Shvodian, TR): "The start of transmission of the response data frame shall start at the end of a SIFS, like an Imm-ACK frame transmission." This is inconsistent with the previous section which says that Immediate ACK is sent within a SIFS. Need to clarify the use of SIFS. Suggest accept in principle, “The previous section that said that the Imm-ACK ‘shall start within a SIFS duration’ has been changed to be ‘shall start a SIFS duration’ as part of the resolution of comment 1543, which is now consistent with the current section.”

Accept in principle, “Change all references to timing with a SIFS to be that it happens on a SIFS, not before a SIFS or after a SIFS. Retain the definition in clause 11 that the SIFS is a range, currently 10-11 us. Technical Editor will post all other comments changed by this for review.”

1.2.10 Modified association protocol

1601 (Shvodian, TR): A AssocRespConfirmTime of 5 ms is too short. There may not be any CAP or GTS slot time to respond and the PNC may be busy. Change AssocRespConfirmTime to 2 superframe durations. Suggest accept, “Change the value of aAssocRespConfirmTime to be 2*aMaxSuperframeDuration.”

Accept in principle, “The association protocol will be modified as indicated in 02/037r0. The AssocRespConfirmTime has been replaced by aAssocConfirmTime and will be set to 4*aMaxSuperframeDuration’.”

576, 662, 717, 718 (Heberling, TR), 661 (Heberling, T)

Accept in principle, “The association protocol will be modified as indicated in 02/037r0.”

719 (Heberling, TR): Suggest change to current association process. (tagged AssociationInfo)

Accept in principle, “The association protocol will be modified as indicated in 02/037r0.”

721 (Heberling, TR): Change broadcasting DEV (now CTR) information description. (tagged AssociationInfo)

Accept in principle, “The association protocol will be modified as indicated in 02/037r0. Remove the requirement to broadcast the DEVInfo table when a DEV disassociates from the piconet.”

1.2.11 PNID related issues

(tagged PNID)

1524 (Shvodian, T): Piconet randomization does not address if the PNID is the same each time the piconet starts, or if it chooses a different random PNID each time. Clarify if each PNC calculates the same random number each time they generate a PNID, or if it is different each time. Resolve with 1467.

> Accept in principle, “Instead of sending the PNCs MAC address, we are now sending a BSID, as described in 02/129r2.”

1467 (Shvodian, TR): "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." "May be persistent"? How is it determined if it is persistent? Up to the implenter? Do PNCs always use the same PNID? Need to describe the details of persistence of the PNID. Suggest ?

> Accept in principle, “Instead of sending the PNCs MAC address, we are now sending a BSID, as described in 02/129r2.”

800 (Kinney, T): There is a possibility of duplicate network id's. A device will check to see if there are any similar ids but this search cannot be 100% sure, additionally, a PAN may walk into another's coverage area. I did not see any detection nor resolution of this event. Describe the techniques to detect network id duplication and the procedures to resolve it. Suggest accept in principle, “The beacons in any piconet are unique since they contain the PNCs address. However it is possible for a DEV to hear packets from adjacent piconets that are using the same PNID. Add to the end of the paragraph in subclause 8.2.2 the following sentence, ‘However, when a DEV starts a piconet, it shall not use a PNID that was found in the scan that was used to start the piconet.’”

> Accept in principle, “Instead of sending the PNCs MAC address, we are now sending a BSID, as described in 02/129r2. In the resolution of other comments, we added that the PNC should periodically not send the beacon in order to listen on the current channel for other piconets that have moved into its area of operation.”

2. Resolution text

2.1 Text for resolution of PNID comments

In 7.3.1, table 60, change ‘device identifier’ to be ‘piconet BSID’ (add BSID to be beacon source identifier in acronyms) in two locations. Add information element to 7.4, as follows

‘7.4.2 Piconet BSID

The piconet BSID is used to provide a text string to identify the piconet. The piconet BSID shall be formatted as illustrated in {xref Figure xx}..

octets: 1	**1**	**6-32**
Element ID	Length (=6)	BSID

Figure 2—BSID element

The BSID is an set of ISO - (ASCII) encoded characters that is used to identify the piconet. The setting of the BSID is described in {xref 8.2.x}.'

7.4.20 Overlapping PNID

The overlapping PNID element is used to communicate the PNIDs that a DEV has detected either in its channel or in other channels. The overlapping PNID element shall be formatted as illustrated in {xref Figure xx}..

octets: 1	**1**	**2**	**1**	**...**	**2**	**1**
Element ID	Length (=variable)	PNID 1	Channel index 1	...	PNID n	Channel index n

Figure 3—Overlapping PNID element

The PNID field contains the PNID in a frame that a DEV has received since the last time this command was sent.

If the DEV has received a beacon from a different piconet on the current channel with the same PNID, e.g. the BSID is different, it will add that PNID and channel index to this element. Otherwise, the element shall not contain the same PNID/channel index pair as the current piconet. Thus a DEV will report piconets with the same PNID in other channels but will not erroneously report frames from the current piconet as being an overlapping piconet.

The channel index field contains the channel on which the PNID was found.'

Add to clause 8 a new clause

'8.2.9 Setting the BSID

The BSID is used to provide a way identify the piconet. The BSID is set via the MLME-SET command and is stored in the MAC PIB item MACPIBBSID, {xref 6.x.x.}. The BSID shall be persistent for the duration of the piconet, except that it may be changed if the PNC detect another piconet is using the same BSID in same channel. This means that the BSID is preserved in the PNC handover process. However, the PNC is able to change the BSID via the piconet parameter change information element in the beacon, {xref 7.4.x} using the process described in {xref 8.x.x}. The BSID may be persistent when the PNC restarts a piconet that ended, {xref 8.x.x}without handing over control to an AC.

The PNID is chosen by the PNC when it starts the piconet and shall only be changed if the PNC detects another piconet with the same PNID in the same channel. The same PNID may be persistent when the PNC restarts a piconet that ended, {xref 8.x.x}without handing over control to an AC.

If the PNC detects that another piconet is using the same BSID in its operational area, it may change the BSID using the piconet parameter change information element in the beacon. If the PNC detects that another piconet is using the same PNID in the same channel, it shall choose another PNID and change it via the piconet parameter change information element in the beacon. The PNC shall not simulaneously change both the PNID and BSID unless it detects that an adjacent piconet that is using the same PNID and BSID.

If a DEV detects a piconet on the same channel with the same PNID, it shall send a probe command to the PNC with the overlapping piconets element that contains the current PNID and channel index. Once this

command has been successfully sent, the DEV shall not send this information again until after the current PNID has been changed by the PNC.

Change ‘Channel change information element’ to be ‘piconet parameter change information element’ throughout.

Note also that in the resolution of other comments, we added that the PNC should periodically not send the beacon in order to listen on the current channel for other piconets that have moved into its area of operation.

Add MAC PIB element,(note addition of underscore, change throughout).

Table 1—MAC PIB PNC group parameters

Managed Object	Number of octets	Definition	Type
MACPIB_BSID	6-32	Identifies the piconet	Dynamic

Change figure 24 to be:.

octets: 1	1	1	1	2	6-32
Element ID	Length (=10-36)	New channel index	Change time-out	PNID	BSID

Figure 4—Piconet parameter change element

Add the following text to the description:

‘The new channel index indicates the channel to which the PNC is intending to move the piconet.If the PNC is only changing either the PNID or the BSID, then the new channel index is the same as the current channel index. The values of this field are PHY dependent. For the 2.4 GHz PHY, the valid channels are defined in 11.2.3.

The change time-out is the time within which the DEVs shall expect beacon from PNC in the channel indicated by the new channel index with the PNID and BSID specified in this element. This time duration is indicated in Kμs.

The PNID field is the PNID that will take effect when the change timeout has expired. Thus if the PNID is not changing, this field is set to the current PNID.

The BSID field is the BSID that will take effect when the change timeout has expired. Thus if the BSID is not changing, this field is set to the current BSID’

Also, change the ‘MLME-CHANNEL-CHANGE.xxx’ to be ‘MLME-PICONET-PARM-CHANGE.xxx’

2.2 Capabilites discovery, comment 30

Add information element to clause 7

7.4.x Piconet services

The piconet services element is used to provide information about the application layer capabilities of either the DEVs in a piconet or a individual DEV. The format of the piconet services element shall be formatted as illustrated in Figure 2.

octets: 1	**1**	**2**	**(0-128)**
Element ID	Length (=2 – 130)	Type	Piconet services

Figure 5—Piconet services element

The type field indicates the format of the piconet services field. There are currently no defined types for the services field.

The piconet services field is used to indicate the application layer capabilities of the DEV that sends this element. In the case of the PNC sending the element, it is the aggregate application layer capabilities of the DEVs in the piconet. The value of this field is set with the MLME association commands. The contents of the field are application dependent and so is outside of the scope of this standard.

If a DEV or the PNC does not support the piconet services element or if its policy, as set in the MACPIB_DEVServicesBroadcast or MACPIB_PNCServicesBroadcast, respectively, xref 6.x.x, it shall set the type to 0x00 and the length of the element set to 2, i.e. a zero length piconet services field

Add to clause 8

8.x.x The piconet services field

The piconet services information element may be provided in the association command and association response command, xref 7.5.x. An associating DEV may inspect the piconet services information element returned by the PNC to determine information about other DEVs in the piconet prior to completion of the association and, if requried, authentication process. Associating DEVs may place information in the piconet services information element of the association command to the PNC. If a DEV has a need for privacy, it may not desire to provide information that would be available outside of the security operations of the piconet. The MAC PIB element MACPIB_DEVServicesBroadcast indicates if the DEV will sned the piconet services field. Likewise the PNC is not required furnish this information if it violates the security policy as set in the MAC PIB element MACPIB_PNCServicesBroadcast.

The capabilities field is up to 128 octets in length in the direction PNC to associating DEVs. If the PNC supports this capability, it manages the information and merges information sent by associating DEVs into the piconet's aggregate capabilities field. DEVs with identical information in their piconet services element are presented as a single field in the PNC to DEV field. Not all DEVs in the piconet may be provided with space in the PNC to DEV field. The field is filled on a first come basis by DEVs in the piconet with content in the DEV to PNC field.

It is outside of the scope of this standard to define the content or use of the capabilities field.

2.3 Text for resolution of Del-ACK and fragmentation

The motivation is that if a DEV misses the end of one fragment and the start of another it could overflow its buffer before the end of the packets arrive. This is a problem with non-stream data as it is presumed that for stream data a new fragment would not start until another one has arrived.

Suggestion is to add a fragment number

Change the ‘sequence number’ field to be ‘sequence control’ field with two sub fields, a 12 bit sequence number first followed by a 4 bit fragment number. Change the sequence number field, 7.2.5 to be ‘Sequence control field’ with two subclauses, 7.2.5.1 ‘Sequence number’ and 7.2.5.2 ‘Fragment number’ Use the test for sequence number except change 16 bit to 12 bit and change 65536 to be 4096. Change ‘at the end of frame’ to be ‘and the end of the MPDU’ so that the sequence number is for SDUs but not for fragments.

One way to solve is to require that ‘For frames with Imm-ACK or Implied-ACK policy, a DEV shall not send another fragment or frame with the same stream number to the same DEV until you have received an Imm-ACK or Implied-ACK on that frame or it has timed out.’

Now the case of no-ACK. If I RX a frame with missing fragments, I drop the entire frame. ‘If a DEV receives a no-ACK with a sequence number that is non-sequential, then it should drop all frames from that source DEV with that stream-ID until it receives a frame with the start fragment bit set.’

Add text that says that ‘The MAC may pass packets which fail FCS through the MAC SAP, but that the SSCS shall filter these as appropriate. The implementer is cautioned that some higher layers do not tolerate bad packets.’

Add text to the 802.2 LLC SSCS that ‘The 802.2 LLC SSCS shall not transfer any frames that the MAC SAP has indicated have failed FCS through the 802.2 LLC SSCS SAP.’

The case of Del-ACK. ‘The RX window is a sliding window of sequence numbers of size aRXWindow where the end of the window is sequence number of the last successfully received frame. When a DEV sends a frame with Del-ACK policy set, it shall not send another frame that exceeds the RX window unless it intends to drop the frame. The receiving DEV may drop any frame for which the RX window has bypassed a missing fragment, i.e. the last successfully received sequence number minus aRXWindow.

Add to table 73, parameter aRXWindow, value 32.

Or, add a negotiation sequence that is between DEVs

Delayed-ACK command

The delayed-ACK configuration command structure shall be formatted as illustrated in Figure 6.

octets: 2	**2**	**1**	**1**	**1**	**2**
Command type	Length (=5)	Action type	Stream index	Retry TX duration	RX window size

Figure 6—Delayed-ACK configuration command format

The action type field indicates what the frame is used for and is set as follows:

— 0 - > The originating DEV requesting the values of

2.4 Efficient use of asynchronous data

Ed note: This section is contains compromise text being worked on by individuals in the group, in particular Knut Odman (KO), Bill Shvodian (WMS), Mark Schrader (MS), Jay Bain (JB) and James Gilb (JPKG).

5.5.3 Stream establishment

A DEV that wants to establish either an isochronous stream or channel time for asynchronous data, requests channel time from the PNC during the MTS or CAP with the channel time request command, {xref 7.5.x}. In the case of an isochronous stream, the PNC assigns a stream index to the requesting DEV. Asynchronous data, on the other hand, all use the same stream index. Once the channel time has been allocated for either the isochronous or asynchronous data, the communication is between the source and destination DEVs in a peer-to-peer manner.

Ed. note: Add the more data bit as indicated in 02/100r8 to allow DEVs to turn off in the remainder of a GTSs that will not be used.

7.4.10 Channel time allocation element

Ed. note: We decided on no changes here from what was adopted for the other comments. The PNC is allowed to allocate CTAs in the beacon that overlap in time if:

1) They are for for asynchronous allocations

2) They are from the same source DEV that has requested it.

7.5.10.1 Channel time request command

Ed. note: The CTR below needs to be merged with the CTR construct that allows multiple CTRs in a single frame.

octets: 1	**1**	**2-128**	**1**	**1**	**8**
Command type	Length (=12-138)	Destination list	Stream index	Stream Request ID	Channel time allocation

Figure 7—Channel time request command frame format

The channel time request block shall be formatted as illustrated in Figure 8.

1	**1**	**2**	**2**	**1**	**1**
SPS set Index	CTR Control	CTR Interval	CTR time unit	Minimum number of TUs	Desired number of TUs

Figure 8—Channel time allocation block

The destination list lists the possible destination DEVs of an asynchronous time slot (ATS) or the target DEV of an isochronous stream. It shall be formatted as in Figure 8.

The stream index field shall be set to to the stream request index, {xref 7.x.x}, for the request of a new GTS.

Ed. note: Keep the channel time requests the same as the resolved comments. Add text to 8.6.2.

"A DEV may requests channel time for multiple destinations by using multiple destination addresses in a destination list field of a channel time request. In response to such a request, the PNC shall create multiple CTAs in the beacon, one for each of the destination addresses. The CTAs shall overlap in time. Each of the destination DEVs shall listen to the slot and only receive frames addressed to it."

Ed. note: Change the desired CTR time units field definition to be as follows (I am suggesting renaming these to be minimum number of TUs and desired number of TUs, but other suggestions are welcome.):

"For an isochronous request, the desired number of TUs field indicates the number of CTR TUs per slot that is desired by the requesting DEV. The desired number of TUs shall be greater than or equal to minimum number of TUs.

For an asynchronous request, the concatination of the minimum number of TUs field and the desired number of TUs field indicates the total number of TUs that are requested for this allocation, i.e. it is interpreted as a single, 2-octet field. Note that this is a request for a total amount of time rather than a recurring use of time in the superframe. The use of this field is defined in {xref 8.x.x}."

Ed. note: The rest of the fields are OK as in 01/469r4.

Ed. note: For the channel time status command, change the available CTR TUs field to be 2 octets to handle up to 65535 allocated TUs for asynchronous traffic. Change the description to be the following:

"The available TUs is used by the PNC to indicate to the requesting DEV the TUs it has assigned to the requested stream. For isochronous stream requests, if the available CTR TUs is greater than or equal to the minimum number of TUs requested or less than or equal to the desired number of TUs requested, then the requesting DEV is informed that there is channel time available. If, however, the available CTR TUs field is less than the minimum number of TUs requested, then the requesting DEV is informed that the PNC is unable to fulfill the DEV's request for channel time.

For asynchronous stream requests, the available TUs is the total number of TUs yet to be allocated by the PNC for this stream. If the available TUs is 0, then the requesting DEV is informed that the PNC is unable to fulfill the DEVs request for channel time."

Ed. note: The following is an addition to clause 8 and supersedes the text in 01/469r4.

8.6.2 Asynchronous channel time management

This subclause describes the process used to allocate asynchonous channel time with other DEVs in the piconet.

8.6.2.1 Asynchronous channel time creation and modification

There are two methods for requesting asyncrhonous channel time:

a) Request a single CTA for multiple target DEVs.
b) Request individual CTAs for each of the target DEV.

The DEV requesting asyncrhonous channel time shall only use one of the two methods at a time. The DEV switches between the two methods by sending an channel time request command that utilizes the new method. If the DEV changes methods, the PNC shall drop previously received asynchronous CTRs from that DEV. A DEV shall not send a channel time request that requests both types of asynchronous allocations and the PNC shall reject any request received from a DEV that requests both types of asyncrhonous allocations.

When a DEV desires to reserve channel time for asynchronous data transmissions with a target DEV or DEVs, the originating DEV shall send a channel time request command, {xref 7.5.x, channel time request command}, to the PNC with these parameter values:

a) The destination list shall contain either:
   1) A list of all of the target DEVs. Only one CTR-Block is used for all destinations with the same TU for all of the target DEVs.
   2) Only one DEV in the destination list. In this case the originating DEV may send multiple CTRs in the command and the TU may be different in each of the CTRs.
b) Stream index field shall be set to zero.
c) Priority field shall be set to a value of either 0b000 or 0b001 as defined in {xref A.3}
d) Stream type field shall be set to the bit value indicating an asynchronous stream, {xref 7.2.x stream index definition}.
e) The CTR interval, SPS set index, CTR-SPS type, GTS type and CTR interval type shall be set to 0 and may be ignored upon reception.
f) All the other channel time request command parameters are set to appropriate values as defined in {xref 7.5.x, channel time request command}.

The PNC upon receiving the channel time request command from the originating DEV shall respond with an Imm-ACK to the requesting DEV. If the requested channel time is available the PNC shall respond to the requesting DEV with a beacon containing CTAs with the source and target DEVID fields appropriately set.

In the case when the PNC allocates a single CTA with multiple destinations, the PNC shall place multiple CTAs in the beacon, one for each of the destinations. Each CTA shall have the asynchronous stream index and the same destination, start time and duration.

However, there is no guarantee of what the length of the delay will be between the time of the request and the reception of a beacon containing the requested channel time allocation. If a frame's timeout interval expires while waiting for its requested CTA in the beacon, a MAC-UNTIDATA.confirm shall be sent with the ReasonCode set to TX_TIMEOUT.

When the PNC allocates a asynchronous GTS, it decrements its count of TUs pending allocation by the amount of TUs allocated in the GTS. When the count of TUs pending reaches 0, the PNC no longer allocates the GTS and drops the channel time request information.

If the request is rejected, the PNC shall send a channel time status command indicating the rejection. Note that if an asynchronous request is queued by the PNC but that channel time is not immediately available due to resource constraints, that does not constitute a rejection of the request.

A new asynchronous CTR to a target DEV or grourp of target DEVs replaces the previous one and unallocated TUs from a previous request shall be replaced by the current request. The originating DEV may change the request method, TU size, destinations and desired TUs between subsequent asyncrhonous GTS requests.

Every separate CTR block in the same channel time request command shall be allocated, if the request is accpeted and the time is available, a separate CTA. Upon request of a single CTA to a destination list, the PNC will allocate, when available, a CTA to the same destination list. The PNC may also split an allocation into several GTSs in a single superframe. Such splits shall only be done on the TU boundries.

The PNC may time out the request for an ansynchronous GTS and purge them after {xref aAsyncRequest-Lifetime}. The requesting DEV should resend a new request after {xref aAsyncRequestLifetime} if it desires more channel time.

Ed. note: What is the definition of aAsyncRequestLifetime? 1 second?

8.6.2.2 Asynchronous GTS termination

Only the PNC, the originating DEV, or the target DEV shall be able to terminate an asyncrhonous GTS. In the case where either the originating DEV or the target DEV desires to terminate a specific asynchronous GTS, it shall send to the PNC the channel time request command with these parameter values:

— TargetID field shall be set to the DEV ID of the DEV to which the originating DEV is sending asynchronous data via an asyncrhonous GTS.
— Stream index field shall be set to the asynchronous stream index, {xref 7.4.x},
— The CTR-TU, minimum number of TUs and desired number of TUs fields shall be set to zero.

The PNC, upon receiving the channel time request command from a DEV requesting termination of the GTS, shall respond with an Imm-ACK followed by a beacon in which the asynchronous CTAs assigned to the indicated SrcID and DestID are set to a null CTA value for {xref aBeaconInfoElementRepeat} duration.

3. Summaries

3.1 Monday, 11 March, 2002

Comments resolved: 41, 57, 301, 378, 1185, 1210, 1213, 1346, 1567

3.2 Tuesday, 12 March, 2002

Comments resolved: 602, 725, 726, 1120, 1209, 1431, 1563, 1564, 1716

3.3 Wednesday, 13 March, 2002

Comments resolved: 800, 1467, 1524, 1629, 1633, 1724

3.4 Thursday, 14 March, 2002

Comments resolved: 576, 597, 661, 662, 717, 718, 719, 721, 1464, 1529, 1601.

4. Status at closing

a) All of the comments have been addressed by the committee
b) There are 2 outstanding issues with 3 TR comments from one voter
   1) More efficient handling of fragmentation for Delayed-ACK
   2) More efficient handling of asynchronous traffic.
c) The committee has agreed in principle to the general approach to resolve them.
   1) Use some type of fragment number and SDU number to assist in placing received delayed-ACK packets in the appropriate queue.
   2) Change asynchronous (i.e. non-stream) CTRs to specify a total number of time units allocated, decremented as allocated, as opposed to a time slot that never expires.
d) We are waiting for more precise text from the commenter that would provide a solution to his comments.
e) All security comments have been resolved and accepted.

Table 2—Ballot resolution as of close of St. Louis meeting

Type	LB12	Unresolved as of 15 March, 2002
T (technical)	613	2
TR (Technical required)	553	3
T and TR	1166	5
E (editorial)	685	347
Total	1851	352