

USB Power Delivery ENGINEERING CHANGE NOTICE

Title: Removing the usage of Ping Message

Applied to: USB Power Delivery Specification Revision 3.2

Version 1.0

Brief description of the functional changes proposed:
--

Ping messages were used in PD1 to verify that the Sink is still connected. It has no significance in PD2 and newer versions.

Benefits as a result of the proposed changes:
--

Removing a number of requirements that complicate both HW and SW implementations.

An assessment of the impact to the existing revision and systems that currently conform to the USB specification:
--

None, Ping is not really used any more.

An analysis of the hardware implications:
--

None needed. Less complex newer designs. Fewer details to maintain.

An analysis of the software implications:
--

None expected. Less complex newer designs. Fewer details to maintain.

An analysis of the compliance testing implications:
--

Very minor if any. Less need to make more tests.
--

USB Power Delivery ENGINEERING CHANGE NOTICE

Actual Change Requested

(a). Section 6.2.1.1.4 Port Power Role

From Text:

Messages, such as *Ping*, and *GotoMin*, that are only ever sent by a Source, *Shall* always have the *Port Power Role* field set to Source. Similarly, Messages such as the *Request* Message that are only ever sent by a Sink *Shall* always have the *Port Power Role* field set to Sink.

To Text:

Messages, such as *Ping*, and *GotoMin*, that are only ever sent by a Source, *Shall* always have the *Port Power Role* field set to Source. Similarly, Messages such as the *Request* Message that are only ever sent by a Sink *Shall* always have the *Port Power Role* field set to Sink.

(b). Table 6.5 “Control Message Types”

From Text:

0_0101	Ping	Source only	Section 6.3.5.	SOP only
--------	------	-------------	----------------	----------

To Text:

0_0101	Ping (Deprecated)	-	Deprecated	-
--------	----------------------	---	------------	---

(c). Section 6.3.5

From Text:

6.3.5 Ping Message

The *Ping* Message was previously used on USB Type-A and USB Type-B connectors to determine the continued presence of the Sink when no other messaging was taking place. USB Type-C® connectors have a mechanism to determine Sink presence so when the Port Partners are both connected using USB Type-C® connectors the *Ping* Message is not necessary but *May* be sent by a Source if desired. A Sink using a USB Type-C® connector cannot expect to receive *Ping* Messages but *Shall Not* treat *Ping* Messages as an error if they are received.

To Text:

6.3.5 Ping Message

USB Power Delivery ENGINEERING CHANGE NOTICE

The **Ping** Message has been **Deprecated**. The 0_0101 Message Type is no longer **Valid**.

A Port that receives a **Ping** Message **May** respond with a **Not Supported** Message or **Ignore** the Ping Message. A Cable Plug that receives a Ping Message **Shall Ignore** the **Ping** Message.

(d). Section 6.11

From Text:

6.11 Message Discarding

On receiving a received Message on SOP (except for a **Ping** Message), the Protocol Layer **Shall Discard** any pending SOP* Messages. A received Message on SOP'/SOP'' **Shall Not** cause any pending SOP* Messages to be **Discarded**.

It is assumed that Messages using SOP'/SOP'' constitute a simple request/response AMS, with the Cable Plug providing the response so there is no reason for a pending SOP* Message to be **Discarded**. There can only be one AMS between the Port Partners, and these also take priority over Cable Plug communications so a Message received on SOP will always cause a Message pending on SOP* to be **Discarded**.

To Text:

6.11 Message Discarding

On receiving a received Message on SOP (except for a **Ping** Message), the Protocol Layer **Shall Discard** any pending SOP* Messages. A received Message on SOP'/SOP'' **Shall Not** cause any pending SOP* Messages to be **Discarded**.

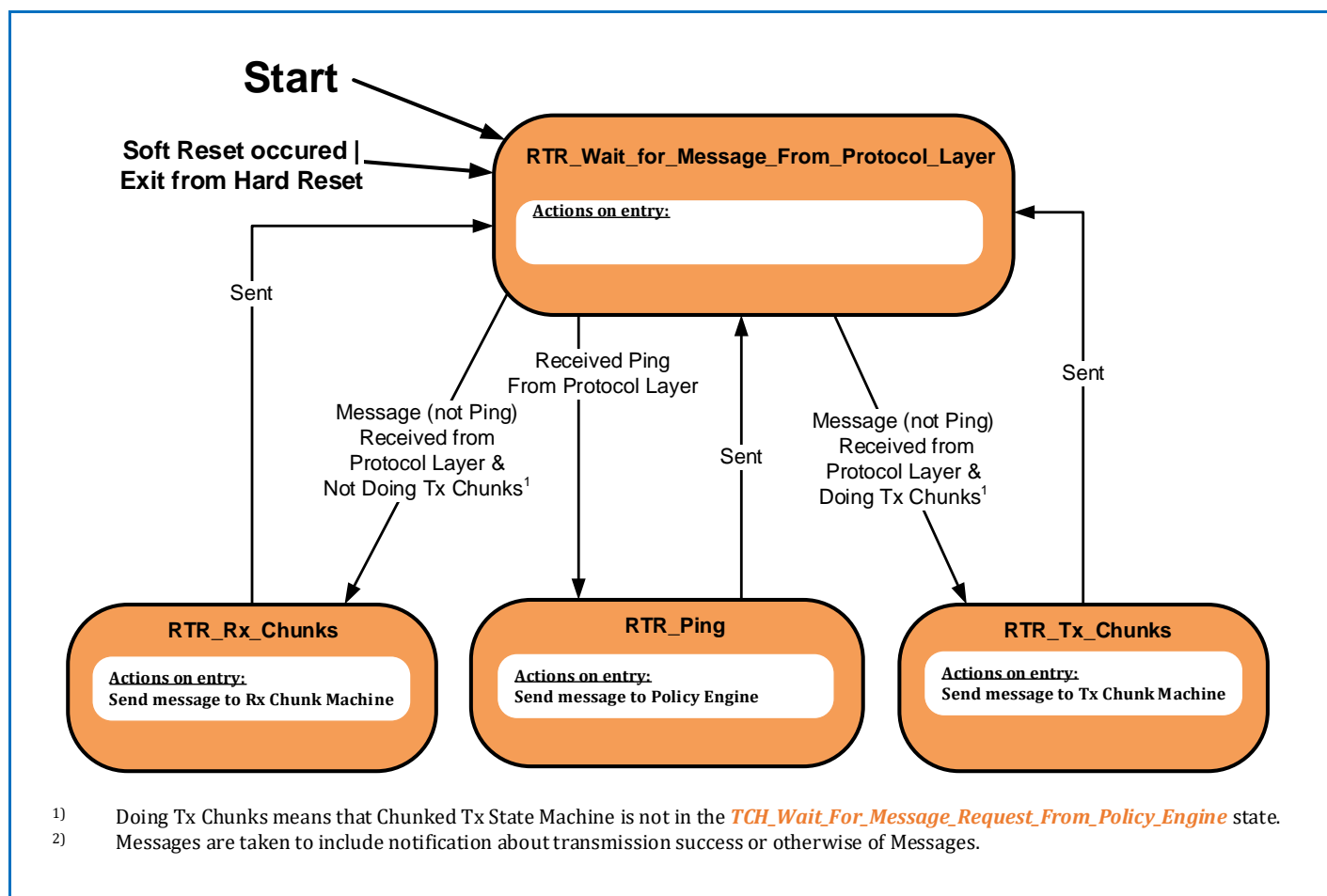
It is assumed that Messages using SOP'/SOP'' constitute a simple request/response AMS, with the Cable Plug providing the response so there is no reason for a pending SOP* Message to be **Discarded**. There can only be one AMS between the Port Partners, and these also take priority over Cable Plug communications so a Message received on SOP will always cause a Message pending on SOP* to be **Discarded**.

(e). Figure 6-60

From Text:

Figure 6-60 "Chunked Message Router State Diagram"

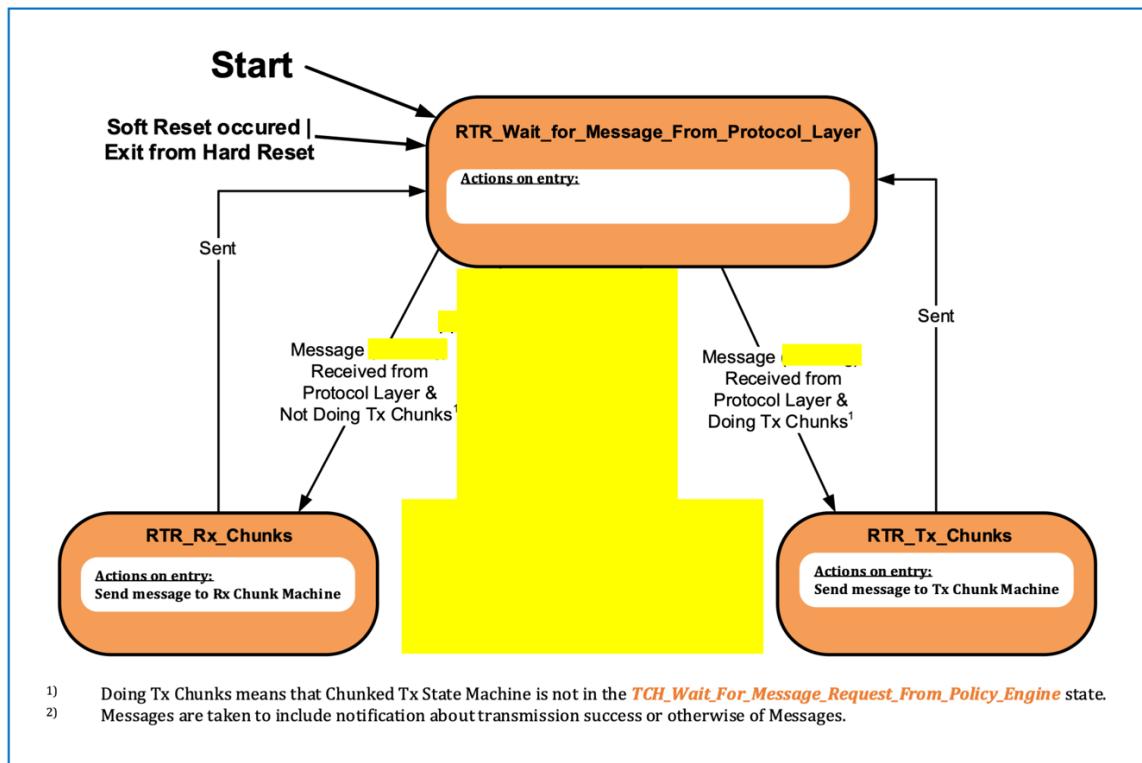
USB Power Delivery ENGINEERING CHANGE NOTICE



To Text:

Figure 6-60 "Chunked Message Router State Diagram"

USB Power Delivery ENGINEERING CHANGE NOTICE



(f). Section 6.12.2.1.4.1

From Text:

6.12.2.1.4.1

RTR_Wait_for_Message_From_Protocol_Layer State

In the *RTR_Wait_for_Message_From_Protocol_Layer* state the Chunked Message Router waits until the Protocol Layer sends it a received Message.

The Chunked Message Router **Shall** transition to the *RTR_Rx_Chunks* state when:

- A Message other than a *Ping* Message is received from the Protocol Layer, and the combined Chunking is not doing Tx Chunks.

The Chunked Message Router **Shall** transition to the *RTR_Tx_Chunks* state when:

- A Message other than a *Ping* Message is received from the Protocol Layer, and the combined Chunking is doing Tx Chunks.

The Chunked Message Router **Shall** transition to the *RTR_Ping* state when:

- A *Ping* Message is received from the Protocol Layer.

6.12.2.1.4.2

RTR_Rx_Chunks State

On entry to the *RTR_Rx_Chunks* state the Chunked Message Router **Shall**:

USB Power Delivery ENGINEERING CHANGE NOTICE

- Send the message to the Chunked Rx State Machine.
- Transition to the *RTR_Wait_for_Message_From_Protocol_Layer* state.

6.12.2.1.4.3

RTR_Ping State

On entry to the *RTR_Ping* state the Chunked Message Router *Shall*:

- Send the message to the Policy Engine.
- Transition to the *RTR_Wait_for_Message_From_Protocol_Layer* state.

6.12.2.1.4.4

RTR_Tx_Chunks State

On entry to the *RTR_Tx_Chunks* state the Chunked Message Router *Shall*:

- Send the message to the Chunked Tx State Machine.
- Transition to the *RTR_Wait_for_Message_From_Protocol_Layer* state.

To Text:

6.12.2.1.4.1

RTR_Wait_for_Message_From_Protocol_Layer State

In the *RTR_Wait_for_Message_From_Protocol_Layer* state the Chunked Message Router waits until the Protocol Layer sends it a received Message.

The Chunked Message Router *Shall* transition to the *RTR_Rx_Chunks* state when:

- A Message ~~other than a *Ping* Message~~ is received from the Protocol Layer, and the combined Chunking is not doing Tx Chunks.

The Chunked Message Router *Shall* transition to the *RTR_Tx_Chunks* state when:

- A Message ~~other than a *Ping* Message~~ is received from the Protocol Layer, and the combined Chunking is doing Tx Chunks.

~~The Chunked Message Router *Shall* transition to the *RTR_Ping* state when:~~

- ~~• A *Ping* Message is received from the Protocol Layer.~~

6.12.2.1.4.2

RTR_Rx_Chunks State

On entry to the *RTR_Rx_Chunks* state the Chunked Message Router *Shall*:

- Send the message to the Chunked Rx State Machine.
- Transition to the *RTR_Wait_for_Message_From_Protocol_Layer* state.

6.12.2.1.4.3

~~RTR_Ping State~~

~~On entry to the *RTR_Ping* state the Chunked Message Router *Shall*:~~

- ~~• Send the message to the Policy Engine.~~
- ~~• Transition to the *RTR_Wait_for_Message_From_Protocol_Layer* state.~~

6.12.2.1.4.4

RTR_Tx_Chunks State

USB Power Delivery ENGINEERING CHANGE NOTICE

On entry to the *RTR_Tx_Chunks* state the Chunked Message Router **Shall**:

- Send the message to the Chunked Tx State Machine.
- Transition to the *RTR_Wait_for_Message_From_Protocol_Layer* state.

(g). Section 6.12.2.3.5

From Text:

6.12.2.3.5 PRL_Rx_Store_MessageID state

On entry to the *PRL_Rx_Store_MessageID* state the Protocol Layer **Shall** transition Protocol Layer Message transmission to the *PRL_Tx_Discard_Message* state (except when a *Ping* Message has been received in which case the *PRL_Tx_Discard_Message* state **Should Not** be entered), replace the stored value of *MessageID* with the value of *MessageID* in the received Message and pass the Message up to the Policy Engine.

To Text:

6.12.2.3.5 PRL_Rx_Store_MessageID state

On entry to the *PRL_Rx_Store_MessageID* state the Protocol Layer **Shall** transition Protocol Layer Message transmission to the *PRL_Tx_Discard_Message* state (except when a *Ping* Message has been received in which case the *PRL_Tx_Discard_Message* state **Should Not** be entered), replace the stored value of *MessageID* with the value of *MessageID* in the received Message and pass the Message up to the Policy Engine.

(h). Table 6.76 “Protocol Layer States”

Remove Text:

RTR_Ping

Section 6.11.2.1.4.3

(i). Table 6.78 “Applicability of Control Messages”

From Text:

<i>Not_Supported</i>	N	N			NA	NA
<i>Ping</i>	O	NA			NA	NA

USB Power Delivery ENGINEERING CHANGE NOTICE

<i>PR_Swap</i>	NA	NA	N		NA	NA
----------------	----	----	---	--	----	----

To Text:

<i>Not_Supported</i>	N	N			NA	NA
<i>Ping (Deprecated)</i>	NA	NA			NA	NA
<i>PR_Swap</i>	NA	NA	N		NA	NA

(j). Table 6.78 “Applicability of Control Messages”

From Text:

<i>Not_Supported</i>	<i>N</i>	<i>N</i>			<i>CN11/I</i>	<i>I</i>
<i>Ping</i>	<i>NS</i>	<i>I</i>			<i>I</i>	<i>I</i>
<i>PR_Swap</i>	<i>NS</i>	<i>NS</i>	<i>N</i>		<i>I</i>	<i>I</i>

To Text:

<i>Not_Supported</i>	<i>N</i>	<i>N</i>			<i>CN11/I</i>	<i>I</i>
<i>Ping (Deprecated)</i>	<i>NS</i>	<i>NS/I</i>			<i>I</i>	<i>I</i>
<i>PR_Swap</i>	<i>NS</i>	<i>NS</i>	<i>N</i>		<i>I</i>	<i>I</i>

(k). Table 8.4 “Atomic Message Sequences”

Remove:

Ping	Table 8.8 “AMS: Ping”	Section 8.3.2.40
------	-----------------------	------------------

USB Power Delivery ENGINEERING CHANGE NOTICE

(l). Section 8.3.2.1.3.4 AMS: Ping

Remove:

(m). Section 8.3.2.4 Ping

Remove:

(n). Section 8.3.3.3.7 PE_SNK_Ready State

From Text:

In the **PE_SNK_Ready** state the PD Sink **shall** be operating at a stable power level with no ongoing negotiation. It **shall** respond to requests from the Source, events from the Device Policy Manager and **May** monitor for **Ping** Messages to maintain the PD link.

To Text:

In the **PE_SNK_Ready** state the PD Sink **shall** be operating at a stable power level with no ongoing negotiation. It **shall** respond to requests from the Source, events from the Device Policy Manager ~~and **May** monitor for **Ping** Messages to maintain the PD link.~~

(o). Section 8.3.3.7 Source Port Ping State Diagram

Remove:

(p). Section 8.3.3.20.6.6 PE_FRS_SNK_SRC_Source_on State

From Text:

On exit from the **PE_FRS_SNK_SRC_Source_on** state (except if the exit is to send a **Ping** Message) the Policy Engine **shall** send a **PS_RDY** Message.

USB Power Delivery ENGINEERING CHANGE NOTICE

To Text:

On exit from the *PE_FRS_SNK_SRC_Source_on* state (except if the exit is to send a *Ping* Message) the Policy Engine *Shall* send a *PS_RDY* Message.

(q). Table 8.157 Policy Engine States

Remove:

Source Port Ping	
<i>PE_SRC_Ping</i>	<i>Section 8.3.3.7.1</i>