

USB Power Delivery ENGINEERING CHANGE NOTICE

Title: Deprecate the GotoMin/GiveBack Features and Update Power Reserve

**Applied to: USB Power Delivery Specification Revision 3.2
Version 1.0**

Brief description of the functional changes proposed:
--

This ECN deprecates the GotoMin/GiveBack features of USB PD . The intended use of these features has not materialized in the market and its implementation in the specification unnecessarily complicates the Sink Request message usage. With these features being eliminated, some additional adjustments in the areas of Capabilities Mismatch are also in the proposed changes and includes deprecating the Maximum Operating Current field in the Sink Request message. Finally, the Power Reserved sections in Chapter 6 and 8 are updated to align with these changes.
--

Benefits as a result of the proposed changes:
--

Removes unused features from the USB PD spec.
--

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

No intended impact.

An analysis of the hardware implications:
--

No intended implications.

An analysis of the software implications:
--

No intended implications.

An analysis of the compliance testing implications:
--

Will impact the CTS, removing some tests and potentially resulting in adjustments to tests that remain.

USB Power Delivery ENGINEERING CHANGE NOTICE

Actual Change Requested

(a'). Section 1.6

To:

Table 1.3 Terms and Abbreviations

Term	Description
Power Reserve	Power which is kept back by a Source to ensure that it can meet total power requirements of Attached Sinks on at least one Port.

(a). Section 2.5.1

From:

2.5.1 Source Operation

The Source operates differently depending on its Attachment status:

⋮

- When the Source Port is part of a multi-port system:
 - Will issue GotoMin requests when the Power Reserve is needed.

⋮

To:

2.5.1 Source Operation

The Source operates differently depending on its Attachment status:

⋮

- ~~• When the Source Port is part of a multi-port system:~~
 - ~~○ Will issue GotoMin requests when the Power Reserve is needed.~~

⋮

(b). Section 6.2.1.1.4

From:

6.2.1.1.4 Port Power Role

The 1-bit **Port Power Role** field **shall** indicate the Port's present power role:

USB Power Delivery ENGINEERING CHANGE NOTICE

- 0b Sink
- 1b Source

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:

6.2.1.1.4 Port Power Role

The 1-bit *Port Power Role* field **Shall** indicate the Port's present power role:

- 0b Sink
- 1b Source

Messages, such as ~~*Get Sink Cap Extended*~~*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.

⋮

(c). Section 6.3, Table 6.5

From:

Table 6.5 "Control Message Types"

Bits 4...0	Message Type	Sent by	Description	Valid Start of Packet
0_0010	<i>GotoMin</i>	Source only	<i>Section 6.3.2.</i>	SOP only

To:

Table 6.5 "Control Message Types"

Bits 4...0	Message Type	Sent by	Description	Valid Start of Packet
0_0010	<i>GotoMin</i> (Deprecated)	Source only N/A	<i>Section 6.3.2.</i>	SOP only N/A

USB Power Delivery ENGINEERING CHANGE NOTICE

(d). Section 6.3.2

From:

6.3.2 GotoMin Message

The **GotoMin** Message applies only to those Sinks that have requested power with the GiveBack capable flag set in the Sink Request Data Object.

It is a directive to the Sink Port to reduce its operating power level to the amount specified in the Minimum Operating Current field of its latest Sink Request Data Object.

The GotoMin process is designed to allow the Source to temporarily reallocate power to meet a short-term requirement. For example, a Source can reduce a Sink's power consumption for 10-20 seconds to allow another Sink (e.g., an HDD to spin up).

The Source sends this Message to harvest power to meet a request for power that it cannot otherwise meet. The Device Policy Manager determines which Port or ports will receive the Message.

The Sink **shall** respond to a **GotoMin** Message by reducing its power consumption to less than or equal to the pre-negotiated value (Minimum Operating Current) within **tSnkNewPower** time.

The Source sends a **GotoMin** Message as a shortcut in the power negotiation process since the Source and Sink have already made a Contract with respect to the power to be returned. The Source does not have to Advertise its Capabilities and the Sink does not have to make a Request based on them. The Source simply sends the **GotoMin** Message in place of the **Accept** Message normally sent during the power negotiation process (see step 19 in [Figure 8-5 "Successful Fixed, Variable or Battery SPR Power Negotiation"](#)). The power negotiation process then completes from this point in the normal manner with the Source sending a **PS_RDY** Message once the power supply transition is complete. The steps of the GotoMin process are fully described in [Figure 8-8 "Successful GotoMin operation"](#).

The Source **shall** return power to the Sink(s) it has 'borrowed' from using the GotoMin mechanism before it can allocate any 'new' power to other devices.

To:

6.3.2 GotoMin Message **(Deprecated)**

The **GotoMin** Message **has been Deprecated**. The 0_0010 **Message Type** is no longer valid and **shall** be responded to by a **Not Supported** message.

applies only to those Sinks that have requested power with the GiveBack capable flag set in the Sink Request Data Object

It is a directive to the Sink Port to reduce its operating power level to the amount specified in the Minimum Operating Current field of its latest Sink Request Data Object.

The GotoMin process is designed to allow the Source to temporarily reallocate power to meet a short-term requirement. For example, a Source can reduce a Sink's power consumption for 10-20 seconds to allow another Sink (e.g., an HDD to spin up).

The Source sends this Message to harvest power to meet a request for power that it cannot otherwise meet. The Device Policy Manager determines which Port or ports will receive the Message.

The Sink **shall** respond to a **GotoMin** Message by reducing its power consumption to less than or equal to the pre-negotiated value (Minimum Operating Current) within **tSnkNewPower** time.

USB Power Delivery ENGINEERING CHANGE NOTICE

The Source sends a **GotoMin** Message as a shortcut in the power negotiation process since the Source and Sink have already made a Contract with respect to the power to be returned. The Source does not have to Advertise its Capabilities and the Sink does not have to make a Request based on them. The Source simply sends the **GotoMin** Message in place of the **Accept** Message normally sent during the power negotiation process (see step 19 in **Figure 8-5 “Successful Fixed, Variable or Battery SPR Power Negotiation”**). The power negotiation process then completes from this point in the normal manner with the Source sending a **PS_RDY** Message once the power supply transition is complete. The steps of the GotoMin process are fully described in **Figure 8-8 “Successful GotoMin operation”**.

The Source **Shall** return power to the Sink(s) it has ‘borrowed’ from using the GotoMin mechanism before it can allocate any ‘new’ power to other devices.

(e). Section 6.3.12 and 6.3.12.1

From:

6.3.12 Wait Message

The **Wait** Message is a **Valid** response to one of the following Messages:

- It **Shall** be sent to signal the Sink, in response to a **Request** Message in SPR Mode during Power Negotiation, to indicate that the Source is currently unable to meet the request.
- It **Shall** be sent to signal the Sink, in response to a **EPR_Request** Message in EPR Mode during Power Negotiation, to indicate that the Source is currently unable to meet the request.
- It **Shall** be sent by the recipient of a **PR_Swap** Message to indicate it is currently unable to do a Power Role Swap.
- It **Shall** be sent by the recipient of a **DR_Swap** Message to indicate it is currently unable to do a Data Role Swap.
- It **Shall** be sent by the recipient of a **VCONN_Swap** Message that is not presently the VCONN Source to indicate it is currently unable to do a VCONN Swap.
- It **Shall** be sent by the recipient of an **Enter_USB** Message to indicate it is currently unable to enter the requested USB Mode.

The **Wait** Message **Shall** be sent within **tReceiverResponse** of the receipt of the last bit of the Message (see **Section 6.9 “Accept, Reject and Wait”**).

6.3.12.1 Wait in response to a Request Message

The **Wait** Message is used by the Source when a Sink that has **Reserved** power, requests it. The **Wait** Message allows the Source time to recover the power it requires to meet the request through the GotoMin process. A Source **Should** only send a **Wait** Message in response to a **Request** Message when an Explicit Contract exists between the Port Partners.

The Sink is allowed to repeat the **Request** Message using the **SinkRequestTimer** and **Shall** ensure that there is **tSinkRequest** after receiving the **Wait** Message before sending another **Request** Message.

To:

6.3.12 Wait Message

The **Wait** Message is a **Valid** response to one of the following Messages:

USB Power Delivery ENGINEERING CHANGE NOTICE

- It **Shall** be sent to signal the Sink, in response to a **Request** Message in SPR Mode during Power Negotiation, to indicate that the Source is currently unable to meet the request.
- It **Shall** be sent to signal the Sink, in response to a **EPR_Request** Message in EPR Mode during Power Negotiation, to indicate that the Source is currently unable to meet the request.
- It **Shall** be sent by the recipient of a **PR_Swap** Message to indicate it is currently unable to do a Power Role Swap.
- It **Shall** be sent by the recipient of a **DR_Swap** Message to indicate it is currently unable to do a Data Role Swap.
- It **Shall** be sent by the recipient of a **VCONN_Swap** Message that is not presently the VCONN Source to indicate it is currently unable to do a VCONN Swap.
- It **Shall** be sent by the recipient of an **Enter_USB** Message to indicate it is currently unable to enter the requested USB Mode.

The **Wait** Message **Shall** be sent within **tReceiverResponse** of the receipt of the last bit of the Message (see [Section 6.9 “Accept, Reject and Wait”](#)).

6.3.12.1 Wait in response to a Request Message

The **Wait** Message is used by the Source when a Sink that has **Reserved** power, requests it. The **Wait** Message allows the Source time to recover the power it requires to meet the request, e.g., through **Re-negotiating with other Sinks or an upstream source** the **GotoMin** process. A Source **Should** only send a **Wait** Message in response to a **Request** Message when an Explicit Contract exists between the Port Partners.

The Sink is allowed to repeat the **Request** Message using the **SinkRequestTimer** and **Shall** ensure that there is **tSinkRequest** after receiving the **Wait** Message before sending another **Request** Message.

(e). Section 6.4.1.2.1

From:

6.4.1.2.1 Management of the Power Reserve

A Power Reserve **May** be allocated to a Sink when it makes a request from Source Capabilities which includes a Maximum Operating Current/Power. The size of the Power Reserve for a particular Sink is calculated as the difference between its Maximum Operating Current/Power field and its Operating Current/Power field. For a Hub with multiple ports this same Power Reserve **May** be shared between several Sinks. The Power Reserve **May** also be temporarily used by a Sink which has indicated it can give back power by setting the GiveBack flag.

Where a Power Reserve has been allocated to a Sink the Source **Shall** indicate the Power Reserve as part of every **Source_Capabilities** Message it sends. When the same Power Reserve is shared between several Sinks the Source **Shall** indicate the Power Reserve as part of every **Source_Capabilities** Message it sends to every Sink. Every time a Source sends capabilities including the Power Reserve capability and then accepts a request from a Sink including the Power Reserve indicated by its Maximum Operating Current/Power it is confirming that the Power Reserve is part of the Explicit Contract with the Sink.

When the Reserve is being temporarily used by a giveback capable Sink the Source **Shall** indicate the Power Reserve as available in every **Source_Capabilities** Message it sends. However, in this situation, when the Power Reserve is requested by a Sink, the Source **Shall** return a **Wait** Message while it retrieves this power using a **GotoMin** Message. Once the additional power has been retrieved the Source **Shall** send a new **Source_Capabilities** Message in order to trigger a new request from the Sink requesting the Power Reserve.

The Power Reserve **May** be de-allocated by the Source at any time, but the de-allocation **Shall** be indicated to the Sink or Sinks using the Power Reserve by sending a new **Source_Capabilities** Message.

USB Power Delivery ENGINEERING CHANGE NOTICE

To:

6.4.1.2.1 Management of the Power Reserve

This section has been removed. Refer to **Section 8.2.5 “Managing Power Requirements”**.

A Power Reserve **May** be allocated to a Sink when it makes a request from Source Capabilities which includes a Maximum Operating Current/Power. The size of the Power Reserve for a particular Sink is calculated as the difference between its Maximum Operating Current/Power field and its Operating Current/Power field. For a Hub with multiple ports this same Power Reserve **May** be shared between several Sinks. The Power Reserve **May** also be temporarily used by a Sink which has indicated it can give back power by setting the GiveBack flag.

Where a Power Reserve has been allocated to a Sink the Source **Shall** indicate the Power Reserve as part of every **Source_Capabilities** Message it sends. When the same Power Reserve is shared between several Sinks the Source **Shall** indicate the Power Reserve as part of every **Source_Capabilities** Message it sends to every Sink. Every time a Source sends capabilities including the Power Reserve capability and then accepts a request from a Sink including the Power Reserve indicated by its Maximum Operating Current/Power it is confirming that the Power Reserve is part of the Explicit Contract with the Sink.

When the Reserve is being temporarily used by a giveback capable Sink the Source **Shall** indicate the Power Reserve as available in every **Source_Capabilities** Message it sends. However, in this situation, when the Power Reserve is requested by a Sink, the Source **Shall** return a **Wait** Message while it retrieves this power using a **GotoMin** Message. Once the additional power has been retrieved the Source **Shall** send a new **Source_Capabilities** Message in order to trigger a new request from the Sink requesting the Power Reserve.

The Power Reserve **May** be de-allocated by the Source at any time, but the de-allocation **Shall** be indicated to the Sink or Sinks using the Power Reserve by sending a new **Source_Capabilities** Message.

(f). Section 6.4.2

From:

6.4.2 Request Message

A **Request** Message **Shall** be sent by a Sink to request power, typically during the request phase of an SPR power negotiation. The Request Data Object **Shall** be returned by the Sink making a request for power. It **Shall** be sent in response to the most recent **Source_Capabilities** Message (see **Section 8.3.2.2 “Power Negotiation”**) when in SPR Mode. A **Request** Message **Shall** return one and only one Sink Request Data Object that **Shall** identify the Power Data Object being requested.

The Source **Shall** respond to a **Request** Message with an **Accept** Message, a **Wait** Message or a **Reject** Message (see **Section 6.9 “Accept, Reject and Wait”**).

The **Request** Message includes the requested power level. For example, if the **Source_Capabilities** Message includes a Fixed Supply PDO that offers 9V @ 1.5A and if the Sink only wants 9V @ 0.5A, it will set the Operating Current field to 50 (i.e., 10mA * 50 = 0.5A). The **Request** Message requests the highest current the Sink will ever require in the Maximum Operating Current Field (in this example it would be 100 (100 * 10mA = 1.0A)).

The request uses a different format depending on the kind of power requested.

The Fixed Power Data Object and Variable Power Data Object share a common format shown in:

- **Table 6.22 “Fixed and Variable Request Data Object”**.
- **Table 6.23 “Fixed and Variable Request Data Object with GiveBack Support”**.

USB Power Delivery ENGINEERING CHANGE NOTICE

The Battery Power Data Object uses the format shown in:

- [Table 6.24 “Battery Request Data Object”](#).
- [Table 6.25 “Battery Request Data Object with GiveBack Support”](#).

The PPS Request Data Object’s format is shown in [Table 6.26 “PPS Request Data Object”](#).

The AVS Request Data Object’s format is shown in [Table 6.27 “AVS Request Data Object”](#).

The Request Data Objects are also used by the [EPR_Request](#) Message when operating in EPR Mode. See [Section 6.4.9 “EPR_Request Message”](#) for information about the use of the [EPR_Request](#) Message.

A Source operating in EPR Mode that receives a [Request](#) Message **Shall** initiate a Hard Reset.

Table 6.22 “Fixed and Variable Request Data Object”

Bits	Description
B31...28	Object position (0000b and 1110b...1111b are Reserved and Shall Not be used)
B27	GiveBack flag = 0
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	Reserved - Shall be set to zero.
B19...10	Operating current in 10mA units
B9...0	Maximum Operating Current 10mA units

Table 6.23 “Fixed and Variable Request Data Object with GiveBack Support”

Bits	Description
B31...28	Object position (0000b and 1110b...1111b are Reserved and Shall Not be used)
B27	GiveBack flag =1
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	Reserved - Shall be set to zero.
B19...10	Operating Current in 10mA units
B9...0	Minimum Operating Current 10mA units

Table 6.24 “Battery Request Data Object”

Bits	Description
------	-------------

USB Power Delivery ENGINEERING CHANGE NOTICE

B31...28	Object position (0000b and 1110b...1111b are Reserved and Shall Not be used)
B27	GiveBackFlag = 0
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	Reserved - Shall be set to zero.
B19...10	Operating Power in 250mW units
B9...0	Maximum Operating Power in 250mW units

Table 6.25 “Battery Request Data Object with GiveBack Support”

Bits	Description
B31...28	Object position (0000b and 1110b...1111b are Reserved and Shall Not be used)
B27	GiveBackFlag = 1
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	Reserved - Shall be set to zero.
B19...10	Operating Power in 250mW units
B9...0	Minimum Operating Power in 250mW units

To:

6.4.2 Request Message

A **Request** Message **Shall** be sent by a Sink to request power, typically during the request phase of an SPR power negotiation. The Request Data Object **Shall** be returned by the Sink making a request for power. It **Shall** be sent in response to the most recent **Source_Capabilities** Message (see [Section 8.3.2.2 “Power Negotiation”](#)) when in SPR Mode. A **Request** Message **Shall** return one and only one Sink Request Data Object that **Shall** identify the Power Data Object being requested.

The Source **Shall** respond to a **Request** Message with an **Accept** Message, a **Wait** Message or a **Reject** Message (see [Section 6.9 “Accept, Reject and Wait”](#)).

The **Request** Message includes the requested power level. For example, if the **Source_Capabilities** Message includes a Fixed Supply PDO that offers 9V @ 1.5A and if the Sink only wants 9V @ 0.5A, it will set the Operating Current field to 50 (i.e., 10mA * 50 = 0.5A). **The Request Message requests the highest current the Sink will ever require in the Maximum Operating Current Field (in this example it would be 100 (100 * 10mA = 1.0A)).**

The request uses a different format depending on the kind of power requested.

The Fixed Power Data Object and Variable Power Data Object share a common format shown in:

USB Power Delivery ENGINEERING CHANGE NOTICE

- [Table 6.22 “Fixed and Variable Request Data Object”](#).

~~• [Table 6.23 “Fixed and Variable Request Data Object with GiveBack Support”](#)~~

The Battery Power Data Object uses the format shown in:

- [Table 6.24 “Battery Request Data Object”](#).

~~• [Table 6.25 “Battery Request Data Object with GiveBack Support”](#)~~

The PPS Request Data Object’s format is shown in [Table 6.26 “PPS Request Data Object”](#).

The AVS Request Data Object’s format is shown in [Table 6.27 “AVS Request Data Object”](#).

The Request Data Objects are also used by the [EPR_Request](#) Message when operating in EPR Mode. See [Section 6.4.9 “EPR_Request Message”](#) for information about the use of the [EPR_Request](#) Message.

A Source operating in EPR Mode that receives a [Request](#) Message **Shall** initiate a Hard Reset.

Table 6.22 “Fixed and Variable Request Data Object”

Bits	Description
B31...28	Object position (0000b and 1110b...1111b are Reserved and Shall Not be used)
B27	GiveBack flag = 0 Deprecated and Shall be set to zero
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	Reserved - Shall be set to zero.
B19...10	Operating current in 10mA units
B9...0	Maximum Operating Current 10mA units

~~Table 6.23 “Fixed and Variable Request Data Object with GiveBack Support”~~

Bits	Description
B31...28	Object position (0000b and 1110b...1111b are Reserved and Shall Not be used)
B27	GiveBack flag = 1
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	Reserved—Shall be set to zero.
B19...10	Operating Current in 10mA units
B9...0	Minimum Operating Current 10mA units

Table 6.24 “Battery Request Data Object”

Bits	Description
------	-------------

USB Power Delivery ENGINEERING CHANGE NOTICE

B31...28	Object position (0000b and 1110b...1111b are <i>Reserved</i> and <i>Shall Not</i> be used)
B27	GiveBack flag =0= <i>Deprecated and Shall be set to zero.</i>
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	<i>Reserved - Shall be set to zero.</i>
B19...10	Operating Power in 250mW units
B9...0	Maximum Operating Power in 250mW units

Table 6.25 “Battery Request Data Object with GiveBack Support”

Bits	Description
B31...28	Object position (0000b and 1110b...1111b are <i>Reserved</i> and <i>Shall Not</i> be used)
B27	GiveBackFlag = 1
B26	Capability Mismatch
B25	USB Communications Capable
B24	No USB Suspend
B23	Unchunked Extended Messages Supported
B22	EPR Mode Capable
B21...20	<i>Reserved – Shall be set to zero.</i>
B19...10	Operating Power in 250mW units
B9...0	Minimum Operating Power in 250mW units

(g). Section 6.4.2.2

From:

6.4.2.2 GiveBack Flag

The GiveBack flag *Shall* be set to indicate that the Sink will respond to a *GotoMin* Message by reducing its load to the Minimum Operating Current. It will typically be used by a USB Device while charging its Battery because a short interruption of the charge will have minimal impact on the user and will allow the Source to manage its load better.

To:

6.4.2.2 GiveBack Flag **(Deprecated)**

The GiveBack flag ~~has been deprecated and~~ *Shall* be set to ~~zero~~ *indicate that the Sink will respond to a GotoMin Message by reducing its load to the Minimum Operating Current. It will typically be used by a USB Device while charging its Battery because a short interruption of the charge will have minimal impact on the user and will allow the Source to manage its load better.*

USB Power Delivery ENGINEERING CHANGE NOTICE

(h). Section 6.4.3

From:

6.4.2.3 Capability Mismatch

A Capability Mismatch occurs when the Source cannot satisfy the Sink's power requirements based on the Source Capabilities it has offered. In this case the Sink **Shall** make a **Valid** request from the offered Source Capabilities and **Shall** set the Capability Mismatch bit (see [Section 8.2.5.2 "Power Capability Mismatch"](#)). When a Capabilities Mismatch condition does not exist, the Sink **Shall Not** set the Capabilities Mismatch bit.

When a Sink returns a Request Data Object with the Capabilities Mismatch bit set in response to a [Source_Capabilities](#) Message, it indicates that it wants more power than the Source is currently offering. This can be due to either a specific Voltage that is not being offered or there is not sufficient current for the Voltages that are being offered.

Sources whose [Port Reported PDP](#) is less than their [Port Present PDP](#) (see [Section 6.4.11 "Source_Info Message"](#)) **Shall** respond to the Requests with the Capabilities Mismatch bit set as follows. The Source within 2 seconds of the [PS_RDY](#) Message **Shall** send a new Source Capabilities Message (a [Source_Capabilities](#) Message or an [EPR_Source_Capabilities](#) Message depending on operating mode) that offers either:

- The maximum power the Source can supply at this time as reported by the [Port Present PDP](#) or
- Enough power to satisfy the Sink's requirements based on the power actually required by the Sink for full operation from either the:
 - [Sink_Capabilities_Extended](#) Message (Sink Operational PDP in SPR Mode or EPR Sink Operational PDP in EPR Mode) or
 - [Sink_Capabilities](#) or [EPR_Sink_Capabilities](#) Message if the [Sink_Capabilities_Extended](#) Message is not supported by the Sink.

To prevent looping, Sources **Should Not** send a new [Source_Capabilities](#) or [EPR_Source_Capabilities](#) Message in response to subsequent [Request](#) Message or [EPR_Request](#) Message with the Capabilities Mismatch flag set until its Port Present PDP changes.

Once a Guaranteed Capability Source that has responded to a Capabilities Mismatch, it **Shall Not** subsequently send out another [Source_Capabilities](#) Message or [EPR_Source_Capabilities](#) Message at a lower PDP unless the power required by the Sink (as indicated in its [Sink_Capabilities](#) Message or [EPR_Sink_Capabilities](#) Message or [Sink_Capabilities_Extended](#) Message) has also been reduced. Sources wishing to manage their power **May** periodically check [Sink_Capabilities](#) or [EPR_Sink_Capabilities](#) Message or [Sink_Capabilities_Extended](#) to determine whether these have changed.

Note: a Source Capabilities Message refers to a [Source_Capabilities](#) Message or an [EPR_Source_Capabilities](#) Message, and a Sink Capabilities Message refers to a [Sink_Capabilities](#) Message or [EPR_Sink_Capabilities](#) Message depending on operating mode.

In this context a **Valid Request** Message means the following:

- The Object position field **Shall** contain a reference to an object in the last received Source Capabilities Message.
- The Operating Current/Power field **Shall** contain a value which is less than or equal to the maximum current/power offered in the Source Capabilities Message.

USB Power Delivery ENGINEERING CHANGE NOTICE

- If the GiveBack flag is set to zero i.e., there is a Maximum Operating Current/Power field:
 - If the Capability Mismatch bit is set to one:
 - The Maximum Operating Current/Power field **May** contain a value larger than the maximum current/power offered in the Source Capabilities Message's PDO as referenced by the Object position field. This enables the Sink to indicate that it requires more current/power than is being offered. If the Sink requires a different Voltage this will be indicated by its Sink Capabilities Message.
 - Else if the Capability Mismatch bit is set to zero:
 - The Maximum Operating Current/Power field **Shall** contain a value less than or equal to the maximum current/power offered in the Sink Capabilities Message's PDO as referenced by the Object position field.
- Else if the GiveBack flag is set to one i.e., there is a Minimum Operating Current/Power field:
 - The Minimum Operating Current/Power field **Shall** contain a value less than the Operating Current/Power field.

To:

6.4.2.3 Capability Mismatch

A Capability Mismatch occurs when the Source cannot satisfy the Sink's power requirements based on the Source Capabilities it has offered. In this case the Sink **Shall** make a **Valid** request from the offered Source Capabilities and **Shall** set the Capability Mismatch bit (see [Section 8.2.5.2 "Power Capability Mismatch"](#)). When a Capabilities Mismatch condition does not exist, the Sink **Shall Not** set the Capabilities Mismatch bit.

When a Sink returns a Request Data Object with the Capabilities Mismatch bit set in response to a [Source_Capabilities](#) Message, it indicates that it wants more power than the Source is currently offering. This can be due to either a specific Voltage that is not being offered or there is not sufficient current for the Voltages that are being offered.

Sources whose [Port Reported PDP](#) is less than their [Port Present PDP](#) (see [Section 6.4.11 "Source_Info Message"](#)) **Shall** respond to the Requests with the Capabilities Mismatch bit set as follows. The Source within 2 seconds of the [PS_RDY](#) Message **Shall** send a new Source Capabilities Message (a [Source_Capabilities](#) Message or an [EPR_Source_Capabilities](#) Message depending on operating mode) that offers either:

- The maximum power the Source can supply at this time as reported by the [Port Present PDP](#) or
- Enough power to satisfy the Sink's requirements based on the power actually required by the Sink for full operation from either the:
 - [Sink_Capabilities_Extended](#) Message (Sink Operational PDP in SPR Mode or EPR Sink Operational PDP in EPR Mode) or
 - [Sink_Capabilities](#) or [EPR_Sink_Capabilities](#) Message if the [Sink_Capabilities_Extended](#) Message is not supported by the Sink.

To prevent looping, Sources **Should Not** send a new [Source_Capabilities](#) or [EPR_Source_Capabilities](#) Message in response to subsequent [Request](#) Message or [EPR_Request](#) Message with the Capabilities Mismatch flag set until its Port Present PDP changes.

Once a Guaranteed Capability Source that has responded to a Capabilities Mismatch, it **Shall Not** subsequently send out another [Source_Capabilities](#) Message or [EPR_Source_Capabilities](#) Message at a lower PDP unless the power required by the Sink (as indicated in its [Sink_Capabilities](#) Message or [EPR_Sink_Capabilities](#) Message or

USB Power Delivery ENGINEERING CHANGE NOTICE

Sink_Capabilities_Extended Message) has also been reduced. Sources wishing to manage their power *May* periodically check *Sink_Capabilities* or *EPR_Sink_Capabilities* Message or *Sink_Capabilities_Extended* to determine whether these have changed.

Note: a Source Capabilities Message refers to a *Source_Capabilities* Message or an *EPR_Source_Capabilities* Message, and a Sink Capabilities Message refers to a *Sink_Capabilities* Message or *EPR_Sink_Capabilities* Message depending on operating mode.

In this context a *Valid Request* Message means the following:

- The Object position field **Shall** contain a reference to an object in the last received Source Capabilities Message.
- The Operating Current/Power field **Shall** contain a value which is less than or equal to the maximum current/power offered in the Source Capabilities Message.

~~• If the GiveBack flag is set to zero i.e., there is a Maximum Operating Current/Power field:~~

~~○ If the Capability Mismatch bit is set to one:~~

- ~~• The Maximum Operating Current/Power field **May** contain a value larger than the maximum current/power offered in the Source Capabilities Message's PDO as referenced by the Object position field. This enables the Sink to indicate that it requires more current/power than is being offered. If the Sink requires a different Voltage this will be indicated by its Sink Capabilities Message.~~

~~○ Else if the Capability Mismatch bit is set to zero:~~

- ~~• The Maximum Operating Current/Power field **Shall** contain a value less than or equal to the maximum current/power offered in the Sink Capabilities Message's PDO as referenced by the Object position field.~~

~~• Else if the GiveBack flag is set to one i.e., there is a Minimum Operating Current/Power field:~~

~~○ The Minimum Operating Current/Power field **Shall** contain a value less than the Operating Current/Power field.~~

(i). Section 6.4.2.8, 6.4.2.9 and 6.4.2.10

From:

6.4.2.8 Operating Current

The Operating Current field in the Request Data Object **Shall** be set to the actual amount of current the Sink needs to operate at a given time. A new *Request* Message or *EPR_Request* Message, with an updated Operating Current value, **Shall** be issued whenever the Sink's power needs change e.g., from Maximum Operating Current down to a lower current level. In conjunction with the Maximum Operating Current field or Minimum Operating Current field, it provides the Source with additional information that allows it to better manage the distribution of its power.

The Operating Current field in the SPR Programmable Request Data Object is used in addition by the Sink to request the Source for the Current Limit level it needs. When the request is accepted the Source's output current supplied into any load **Shall** be less than or equal to the Operating Current. When the Sink attempts to consume more current, the Source **Shall** reduce the output Voltage so as not to exceed the Operating Current value.

The Operating Current field in the EPR AVS Request Data Object **Shall** be set to the actual amount of current the Sink needs to operate at a given time. Note a Source in AVS mode, unlike the SPR Source in PPS mode, does not support current limit; the Sink is responsible not to take more current than it requested. A new *Request* / *EPR_Request*

USB Power Delivery ENGINEERING CHANGE NOTICE

Message, with an updated Operating Current value, **Shall** be issued whenever the Sink's power needs change e.g., from Maximum Operating Current down to a lower current level.

The value in the Operating Current field **Shall Not** exceed the value in the Maximum Current field. For EPR AVS, the Operating Current field **Shall Not** exceed the Source PDP / Output Voltage rounded down to the nearest 50 mA.

This field **Shall** apply to the Fixed, Variable, Programmable and AVS RDOs.

6.4.2.9 Maximum Operating Current

The Maximum Operating Current field in the **Request** Message or **EPR_Request** Message **Shall** be set to the highest current the Sink will ever require. The difference between the Operating Current and Maximum Operating Current fields (when the GiveBack Flag is cleared) is used by the Device Policy Manager in the Source to calculate the size of the Power Reserve to be maintained (see [Section 8.2.5.1 "Managing the Power Reserve"](#)). The Operating Current value **Shall** be less than or equal to the Maximum Operating Current value.

When the Capabilities Mismatch bit is set to zero the requested Maximum Operating Current **Shall** be less than or equal to the current in the offered Source Capabilities since the Source will need to reserve this power for future use. The Maximum Operating Current field **Shall** continue to be set to the highest current needed in order to maintain the allocation of the Power Reserve. If Maximum Operating Current is requested when the Power Reserve is being used by a GotoMin capable device then the resulting Message will be a **Wait** Message to enable the Source to reclaim the additional current (see [Section 6.3.12.1 "Wait in response to a Request Message"](#) and [Section 8.2.5.1 "Managing the Power Reserve"](#)).

When the Capabilities Mismatch bit is set to one the requested Maximum Operating Current **May** be greater than the current in the offered Source Capabilities since the Source will need this information to ascertain the Sink's actual needs.

See [Section 6.4.2.3 "Capability Mismatch"](#) for more details of the usage of the Capabilities Mismatch bit.

This field **Shall** apply to the Fixed and Variable RDO in SPR mode and the Fixed RDO in EPR mode.

6.4.2.10 Minimum Operating Current

The Minimum Operating Current field in the **Request** Message or **EPR_Request** Message **Shall** be set to the lowest current the Sink requires to maintain operation. The difference between the Operating Current and Minimum Operating Current fields (when the GiveBack Flag is set) is used by the Device Policy Manager to calculate the amount of power which can be reclaimed using a **GotoMin** Message. The Operating Current value **Shall** be greater than the Minimum Operating Current value.

This field **Shall** apply to the Fixed and Variable RDO in SPR mode and the Fixed RDO in EPR mode.

To:

6.4.2.8 Operating Current

The Operating Current field in the Request Data Object **Shall** be set to the **highest actual amount of current the Sink will draw during the Explicit Contract needs to operate at a given time**. A new **Request** Message or **EPR_Request** Message, with an updated Operating Current value, **Shall** be issued whenever the Sink's power needs change **e.g., from Maximum Operating Current down to a lower current level**. **In conjunction with the Maximum Operating Current field or Minimum Operating Current field, it provides the Source with additional information that allows it to better manage the distribution of its power.**

The Operating Current field in the SPR Programmable Request Data Object is used in addition by the Sink to request the Source for the Current Limit level it needs. When the request is accepted the Source's output current supplied into any load **Shall** be less than or equal to the Operating Current. When the Sink attempts to consume more current, the Source **Shall** reduce the output Voltage so as not to exceed the Operating Current value.

USB Power Delivery ENGINEERING CHANGE NOTICE

The Operating Current field in the EPR AVS Request Data Object **Shall** be set to the **highest actual amount of** current the Sink **will draw during the Explicit Contract needs to operate at a given time**. Note a Source in AVS mode, unlike the SPR Source in PPS mode, does not support current limit; the Sink is responsible not to take more current than it requested. A new **Request / EPR_Request** Message, with an updated Operating Current value, **Shall** be issued whenever the Sink's power needs change **e.g., from Maximum Operating Current down to a lower current level**.

The value in the Operating Current field **Shall Not** exceed the value in the Maximum Current field **of the Source Capabilities Message**. For EPR AVS, the Operating Current field **Shall Not** exceed the Source PDP / Output Voltage rounded down to the nearest 50 mA.

This field **Shall** apply to the Fixed, Variable, Programmable and AVS RDOs.

6.4.2.9 Maximum Operating Current

The Maximum Operating Current field has been functionally deprecated. In order to maintain backward compatibility with Sources that may try to interpret the Maximum Operating Current field in the **Request** Message or **EPR_Request** Message, the field **Shall** be set equal to the value of the Operating Current field. To ensure backward compatibility, the Source **Should** ignore this field.

The Maximum Operating Current field in the **Request** Message or **EPR_Request** Message **Shall** be set to the highest current the Sink will ever require. The difference between the Operating Current and Maximum Operating Current fields (when the GiveBack Flag is cleared) is used by the Device Policy Manager in the Source to calculate the size of the Power Reserve to be maintained (see **Section 8.2.5.1 "Managing the Power Reserve"**). The Operating Current value **Shall** be less than or equal to the Maximum Operating Current value.

When the Capabilities Mismatch bit is set to zero the requested Maximum Operating Current **Shall** be less than or equal to the current in the offered Source Capabilities since the Source will need to reserve this power for future use. The Maximum Operating Current field **Shall** continue to be set to the highest current needed in order to maintain the allocation of the Power Reserve. If Maximum Operating Current is requested when the Power Reserve is being used by a GotoMin capable device then the resulting Message will be a **Wait** Message to enable the Source to reclaim the additional current (see **Section 6.3.12.1 "Wait in response to a Request Message"** and **Section 8.2.5.1 "Managing the Power Reserve"**).

When the Capabilities Mismatch bit is set to one the requested Maximum Operating Current **May** be greater than the current in the offered Source Capabilities since the Source will need this information to ascertain the Sink's actual needs.

See **Section 6.4.2.3 "Capability Mismatch"** for more details of the usage of the Capabilities Mismatch bit.

This field **Shall** apply to the Fixed and Variable RDO in SPR mode and the Fixed RDO in EPR mode.

6.4.2.10 Minimum Operating Current

The Minimum Operating Current field in the **Request** Message or **EPR_Request** Message **Shall** be set to the lowest current the Sink requires to maintain operation. The difference between the Operating Current and Minimum Operating Current fields (when the GiveBack Flag is set) is used by the Device Policy Manager to calculate the amount of power which can be reclaimed using a **GotoMin** Message. The Operating Current value **Shall** be greater than the Minimum Operating Current value.

This field **Shall** apply to the Fixed and Variable RDO in SPR mode and the Fixed RDO in EPR mode.

(j). Section 6.4.2.11, 6.4.2.12 and 6.4.2.13

From:

USB Power Delivery ENGINEERING CHANGE NOTICE

6.4.2.11 Operating Power

The Operating Power field in the Request Data Object **Shall** be set to the actual amount of power the Sink wants at this time. In conjunction with the Maximum Operating Power field, it provides the Source with additional information that allows it to better manage the distribution of its power.

This field **Shall** apply to the Battery RDO.

6.4.2.12 Maximum Operating Power

The Maximum Operating Power field in the **Request** Message **Shall** be set to the highest power the Sink will ever require. This allows a Source with a power supply shared amongst multiple ports to intelligently distribute power.

When the Capabilities Mismatch bit is set to zero the requested Maximum Operating Power **Shall** be less than or equal to the power in the offered Source Capabilities since the Source will need to reserve this power for future use. The Maximum Operating Power field **Shall** continue to be set to the highest power needed in order to maintain the allocation of the Power Reserve. If Maximum Operating Power is requested when the Power Reserve is being used by a GotoMin capable device then the resulting Message will be a **Wait** Message to enable the Source to reclaim the additional power (see [Section 6.3.12.1 “Wait in response to a Request Message”](#) and [Section 8.2.5.1 “Managing the Power Reserve”](#)).

When the Capabilities Mismatch bit is set to one the requested Maximum Operating Power **May** be greater than the current in the offered Source Capabilities since the Source will need this information to ascertain the Sink’s actual needs

See [Section 6.4.2.3 “Capability Mismatch”](#) for more details of the usage of the Capabilities Mismatch bit.

This field **Shall** apply to the Battery RDO.

6.4.2.13 Minimum Operating Power

The Minimum Operating Power field in the **Request** Message **Shall** be set to the lowest current the Sink requires to maintain operation. When combined with the Operating Power, it gives a Source with a power supply shared amongst multiple ports information about how much power it can temporarily get back so it can intelligently distribute power.

This field **Shall** apply to the Battery RDO.

To:

6.4.2.11 Operating Power

The Operating Power field in the Request Data Object **Shall** be set to the **highest actual amount of** power the Sink **will draw throughout the Explicit Contract** wants at this time. In conjunction with the Maximum Operating Power field, it provides the Source with additional information that allows it to better manage the distribution of its power.

This field **Shall** apply to the Battery RDO.

6.4.2.12 Maximum Operating Power

The Maximum Operating Power field has been functionally deprecated. In order to maintain backward compatibility with Sources that may try to interpret the Maximum Operating Power field in the **Request** Message, the field **Shall** be set equal to the value of the Operating Power field. To ensure backward compatibility, the Source **Should** ignore this field.

The Maximum Operating Power field in the **Request** Message **Shall** be set to the highest power the Sink will ever require. This allows a Source with a power supply shared amongst multiple ports to intelligently distribute power.

USB Power Delivery ENGINEERING CHANGE NOTICE

When the Capabilities Mismatch bit is set to zero the requested Maximum Operating Power **Shall** be less than or equal to the power in the offered Source Capabilities since the Source will need to reserve this power for future use. The Maximum Operating Power field **Shall** continue to be set to the highest power needed in order to maintain the allocation of the Power Reserve. If Maximum Operating Power is requested when the Power Reserve is being used by a GotoMin capable device then the resulting Message will be a **Wait** Message to enable the Source to reclaim the additional power (see **Section 6.3.12.1 “Wait in response to a Request Message”** and **Section 8.2.5.1 “Managing the Power Reserve”**).

When the Capabilities Mismatch bit is set to one the requested Maximum Operating Power **May** be greater than the current in the offered Source Capabilities since the Source will need this information to ascertain the Sink's actual needs.

See **Section 6.4.2.3 “Capability Mismatch”** for more details of the usage of the Capabilities Mismatch bit.

This field **Shall** apply to the Battery RDO.

6.4.2.13 — Minimum Operating Power

The Minimum Operating Power field in the **Request** Message **Shall** be set to the lowest current the Sink requires to maintain operation. When combined with the Operating Power, it gives a Source with a power supply shared amongst multiple ports information about how much power it can temporarily get back so it can intelligently distribute power.

This field **Shall** apply to the Battery RDO.

(k). Section 6.6.5.1

From:

6.6.5.1 PStTransitionTimer

The **PStTransitionTimer** is used by the Policy Engine to timeout on a **PS_RDY** Message. It is started when a request for a new Capability has been accepted and will timeout after **tpStTransition** if a **PS_RDY** Message has not been received. This condition leads to a Hard Reset and a return to USB Default Operation. The **PStTransitionTimer** relates to the time taken for the Source to transition from one Voltage, or current level, to another (see **Section 7.1 “Source Requirements”**).

The **PStTransitionTimer** **Shall** be started when the last bit of the **GoodCRC** Message **EOP**, corresponding to an **Accept** or **GotoMin** Message, has been transmitted by the Physical Layer. The **PStTransitionTimer** **Shall** be stopped when the last bit of the **GoodCRC** Message **EOP**, corresponding to the **PS_RDY** Message, has been transmitted by the Physical Layer.

To:

6.6.5.1 PStTransitionTimer

The **PStTransitionTimer** is used by the Policy Engine to timeout on a **PS_RDY** Message. It is started when a request for a new Capability has been accepted and will timeout after **tpStTransition** if a **PS_RDY** Message has not been received. This condition leads to a Hard Reset and a return to USB Default Operation. The **PStTransitionTimer** relates to the time taken for the Source to transition from one Voltage, or current level, to another (see **Section 7.1 “Source Requirements”**).

The **PStTransitionTimer** **Shall** be started when the last bit of the **GoodCRC** Message **EOP**, corresponding to an **Accept** or **GotoMin** Message, has been transmitted by the Physical Layer. The **PStTransitionTimer** **Shall** be stopped when

USB Power Delivery ENGINEERING CHANGE NOTICE

the last bit of the *GoodCRC* Message *EOP*, corresponding to the *PS_RDY* Message, has been transmitted by the Physical Layer.

(I). Section 6.13 and 6.13.1

From:

6.13 Message Applicability

The following tables outline the Messages supported by a given port, depending on its capability.

When a Message is supported the feature and Message sequence implied by the Message **Shall** also be supported. For example, Sinks using power for charging that support the *GotoMin* Message **Shall** be able to reduce their current draw when requested via a *GotoMin* Message.

The abbreviations in *Table 6.77 “Message Applicability Abbreviations”* are used in this section to denote the level of support required.

Table 6.77 “Message Applicability Abbreviations”

Abbreviation	Meaning	Description
N	Normative	Shall be supported by this Port/Cable Plug.
CN	Conditional Normative	Shall supported by a given Port/Cable Plug based on features.
R	Recommended	Should be supported by this Port/Cable Plug.
O	Optional	May be supported by this Port/Cable Plug.
NS	Not Supported	Shall result in a <i>Not_Supported</i> Message response by this Port/Cable Plug when received.
I	Ignore	Shall be <i>Ignored</i> by this Port/Cable Plug when received.
NK	NAK	This Port/Cable Plug Shall return Responder NAK to this Command when received.
NA	Not allowed	Shall Not be transmitted by this Port/Cable Plug.
DR	Don't Recognize	There Shall no response at all (i.e., not even a <i>GoodCRC</i> Message) from this Port/Cable Plug when received.

For the case of **Conditional Normative** a note has been added to indicate the condition. “CN/” notation is used to indicate the level of support when the condition is not present.

“R/” and “O/” notation is used to indicate the response when the Recommended or **Optional** Message is not supported.

Note: that where NS/R/NK is indicated for Received Messages this **Shall** apply to the *PE_CBL_Ready*, *PE_SNK_Ready* or *PE_SRC_Ready* states only since unexpected Messages received during a Message sequence are Protocol Errors (see *Section 6.8.1 “Soft Reset and Protocol Error”*).

This section covers Control and Data Message support for Sources, Sink and Cable Plugs. It also covers VDM Command support for DFPs, UFPs and Cable Plugs.

USB Power Delivery ENGINEERING CHANGE NOTICE

6.13.1 Applicability of Control Messages

Table 6.78 “Applicability of Control Messages” details Control Messages that **Shall/Should/Should Not** be transmitted and received by a Source, Sink, Cable Plug or VPD. Requirements for Dual-Role Power Ports and Dual-Role Data Ports **Shall** override any requirements for Source-only or Sink-Only Ports.

Table 6.78 “Applicability of Control Messages”

Message Type	Source	Sink	Dual-Role Power	Dual-Role Data	Cable Plug	VPD ¹²
Transmitted Message						
<i>Accept</i>	N	N			N	N
<i>Data_Reset</i>	CN ¹³ /R	CN ¹³ /R			NA	NA
<i>DR_Swap</i>	O	O		N	NA	NA
<i>FR_Swap</i>	NA	NA	R		NA	NA
<i>Get_Country_Codes</i>	CN ¹⁰ /NA	CN ¹⁰ /NA			NA	NA
<i>Get_PPS_Status</i>	NA	CN ⁹			NA	NA
<i>Get_Sink_Cap</i>	R	NA	N		NA	NA
<i>Get_Sink_Cap_Extended</i>	R	NA	R		NA	NA
<i>Get_Source_Cap</i>	NA	R	N		NA	NA
<i>Get_Source_Cap_Extended</i>	NA	R	R		NA	NA
<i>Get_Source_Info</i>	NA	R	R		NA	NA
<i>Get_Revision</i>	R	R			NA	NA
<i>Get_Status</i>	R	R			NA	NA
<i>GoodCRC</i>	N	N			N	N
<i>GotoMin</i>	CN ¹ /O	NA			NA	NA
<i>Not_Supported</i>	N	N			NA	NA
<i>Ping</i>	O	NA			NA	NA
<i>PR_Swap</i>	NA	NA	N		NA	NA
<i>PS_RDY</i>	N	CN ⁴ /NA	N		NA	NA
<i>Reject</i>	N	O	O	O	CN ¹³ /NA	NA
<i>Soft_Reset</i>	N	N			NA	NA
<i>VCONN_Swap</i>	R	R			NA	NA
<i>Wait</i>	CN ² /O	NA	O	O	NA	NA
Received Message						
<i>Accept</i>	N	N	N	N	I	I
<i>Data_Reset</i>	CN ¹³ /R	CN ¹³ /R			I	I
<i>DR_Swap</i>	O/NS	O/NS		N	I	I
<i>FR_Swap</i>	NS	NS	CN ⁷ /NS		I	I
<i>Get_Country_Codes</i>	CN ¹⁰ /NS	CN ¹⁰ /NS			I	I
<i>Get_PPS_Status</i>	CN ⁹ /NS	NS			I	I
<i>Get_Sink_Cap</i>	NS	N	N		I	I
<i>Get_Sink_Cap_Extended</i>	NS	N	N		I	I

USB Power Delivery ENGINEERING CHANGE NOTICE

<i>Get_Source_Cap</i>	N	NS	N		I	I
<i>Get_Source_Cap_Extended</i>	CN ⁵ /NS	NS	CN ⁵ /NS		I	I
<i>Get_Source_Info</i>	CN ¹⁴	NS	N		I	I
<i>Get_Revision</i>	N	N			O/I	O/I
<i>Get_Status</i>	CN ⁶ /NS	CN ⁶ /NS	CN ⁶ /NS		CN ¹¹ /I	I
<i>GoodCRC</i>	N	N			N	N
<i>GotoMin</i>	NS	R ³			I	I
<i>Not_Supported</i>	N	N			CN ¹¹ /I	I
<i>Ping</i>	NS	I			I	I
<i>PR_Swap</i>	NS	NS	N		I	I
<i>PS_RDY</i>	CN ⁴ /NS	N	N		I	I
<i>Reject</i>	CN ⁸ /NS	N	N	N	I	I
<i>Soft_Reset</i>	N	N			N	N
<i>VCONN_Swap</i>	CN ⁴ / NS	CN ⁴ / NS			I	I
<i>Wait</i>	CN ⁸ /NS	N	N	N	I	I

- 1) **Should** be supported by a PDUSB Hub with multiple Downstream Ports. **Should** be supported by a Host with multiple Downstream Ports.
- 2) **Shall** be supported when transmission of **GotoMin** Messages is supported.
- 3) **Should** be supported by Sinks which use PD power for charging.
- 4) **Shall** be supported by any Port that can supply VCONN.
- 5) **Shall** be supported products that support the **Source_Capabilities_Extended** Message.
- 6) **Shall** be supported by Sources that support the **Alert** Message.
- 7) **Shall** be supported when the Fast Role Swap signal is supported.
- 8) **Shall** be supported when **VCONN_Swap** is supported.
- 9) **Shall** be supported when SPR PPS is supported.
- 10) **Shall** be supported when required by a country authority.
- 11) **Shall** be supported by *Active Cables*.
- 12) VPD includes CT-VPDs when not Connected to a Charger. PD communication with a CT-VPD *Shall* only take place when not Connected to a Charger.
- 13) **Shall** be supported by products that support **[USB4]**.
- 14) **Shall** be supported by all Sources except single port chargers with invariant PDOs.

To:

6.13 Message Applicability

The following tables outline the Messages supported by a given port, depending on its capability.

USB Power Delivery ENGINEERING CHANGE NOTICE

When a Message is supported, the feature and Message sequence implied by the Message **Shall** also be supported. For example, Sinks using power for charging that support the ~~GotoMin~~ Message **Shall** be able to reduce their current draw when requested via a ~~GotoMin~~ Message.

The abbreviations in [Table 6.77 “Message Applicability Abbreviations”](#) are used in this section to denote the level of support required.

Table 6.77 “Message Applicability Abbreviations”

Abbreviation	Meaning	Description
N	Normative	Shall be supported by this Port/Cable Plug.
CN	Conditional Normative	Shall supported by a given Port/Cable Plug based on features.
R	Recommended	Should be supported by this Port/Cable Plug.
O	Optional	May be supported by this Port/Cable Plug.
NS	Not Supported	Shall result in a Not Supported Message response by this Port/Cable Plug when received.
I	Ignore	Shall be Ignored by this Port/Cable Plug when received.
NK	NAK	This Port/Cable Plug Shall return Responder NAK to this Command when received.
NA	Not allowed	Shall Not be transmitted by this Port/Cable Plug.
DR	Don't Recognize	There Shall no response at all (i.e., not even a GoodCRC Message) from this Port/Cable Plug when received.

For the case of **Conditional Normative** a note has been added to indicate the condition. “CN/” notation is used to indicate the level of support when the condition is not present.

“R/” and “O/” notation is used to indicate the response when the Recommended or **Optional** Message is not supported.

Note: that where NS/R/NK is indicated for Received Messages this **Shall** apply to the **PE_CBL_Ready**, **PE_SNK_Ready** or **PE_SRC_Ready** states only since unexpected Messages received during a Message sequence are Protocol Errors (see [Section 6.8.1 “Soft Reset and Protocol Error”](#)).

This section covers Control and Data Message support for Sources, Sink and Cable Plugs. It also covers VDM Command support for DFPs, UFPs and Cable Plugs.

USB Power Delivery ENGINEERING CHANGE NOTICE

6.13.1 Applicability of Control Messages

Table 6.78 “Applicability of Control Messages” details Control Messages that **Shall/Should/Should Not** be transmitted and received by a Source, Sink, Cable Plug or VPD. Requirements for Dual-Role Power Ports and Dual-Role Data Ports **Shall** override any requirements for Source-only or Sink-Only Ports.

Table 6.78 “Applicability of Control Messages”

Message Type	Source	Sink	Dual-Role Power	Dual-Role Data	Cable Plug	VPD ¹²
Transmitted Message						
<i>Accept</i>	N	N			N	N
<i>Data_Reset</i>	CN ¹³ /R	CN ¹³ /R			NA	NA
<i>DR_Swap</i>	O	O		N	NA	NA
<i>FR_Swap</i>	NA	NA	R		NA	NA
<i>Get_Country_Codes</i>	CN ¹⁰ /NA	CN ¹⁰ /NA			NA	NA
<i>Get_PPS_Status</i>	NA	CN ⁹			NA	NA
<i>Get_Sink_Cap</i>	R	NA	N		NA	NA
<i>Get_Sink_Cap_Extended</i>	R	NA	R		NA	NA
<i>Get_Source_Cap</i>	NA	R	N		NA	NA
<i>Get_Source_Cap_Extended</i>	NA	R	R		NA	NA
<i>Get_Source_Info</i>	NA	R	R		NA	NA
<i>Get_Revision</i>	R	R			NA	NA
<i>Get_Status</i>	R	R			NA	NA
<i>GoodCRC</i>	N	N			N	N
<i>GotoMin</i> (Deprecated)	NA/CN ⁴ /O	NA			NA	NA
<i>Not_Supported</i>	N	N			NA	NA
<i>Ping</i>	O	NA			NA	NA
<i>PR_Swap</i>	NA	NA	N		NA	NA
<i>PS_RDY</i>	N	CN ⁴ /NA	N		NA	NA
<i>Reject</i>	N	O	O	O	CN ¹³ /NA	NA
<i>Soft_Reset</i>	N	N			NA	NA
<i>VCONN_Swap</i>	R	R			NA	NA
<i>Wait</i>	CN ² /O	NA	O	O	NA	NA
Received Message						
<i>Accept</i>	N	N	N	N	I	I
<i>Data_Reset</i>	CN ¹³ /R	CN ¹³ /R			I	I
<i>DR_Swap</i>	O/NS	O/NS		N	I	I
<i>FR_Swap</i>	NS	NS	CN ⁷ /NS		I	I
<i>Get_Country_Codes</i>	CN ¹⁰ /NS	CN ¹⁰ /NS			I	I
<i>Get_PPS_Status</i>	CN ⁹ /NS	NS			I	I
<i>Get_Sink_Cap</i>	NS	N	N		I	I
<i>Get_Sink_Cap_Extended</i>	NS	N	N		I	I

USB Power Delivery ENGINEERING CHANGE NOTICE

<i>Get_Source_Cap</i>	N	NS	N		I	I
<i>Get_Source_Cap_Extended</i>	CN ⁵ /NS	NS	CN ⁵ /NS		I	I
<i>Get_Source_Info</i>	CN ¹⁴	NS	N		I	I
<i>Get_Revision</i>	N	N			O/I	O/I
<i>Get_Status</i>	CN ⁶ /NS	CN ⁶ /NS	CN ⁶ /NS		CN ¹¹ /I	I
<i>GoodCRC</i>	N	N			N	N
<i>GotoMin</i> (Deprecated)	NS	NSP ³			I	I
<i>Not_Supported</i>	N	N			CN ¹¹ /I	I
<i>Ping</i>	NS	I			I	I
<i>PR_Swap</i>	NS	NS	N		I	I
<i>PS_RDY</i>	CN ⁴ /NS	N	N		I	I
<i>Reject</i>	CN ⁸ /NS	N	N	N	I	I
<i>Soft_Reset</i>	N	N			N	N
<i>VCONN_Swap</i>	CN ⁴ / NS	CN ⁴ / NS			I	I
<i>Wait</i>	CN ⁸ /NS	N	N	N	I	I

¹⁾ ~~Should be supported by a PDUSB Hub with multiple Downstream Ports. Should be supported by a Host with multiple Downstream Ports.~~

²⁾ ~~Shall be supported when transmission of *GotoMin* Messages is supported.~~

³⁾ ~~Should be supported by Sinks which use PD power for charging.~~

4) **Shall** be supported by any Port that can supply VCONN.

5) **Shall** be supported products that support the *Source_Capabilities_Extended* Message.

6) **Shall** be supported by Sources that support the *Alert* Message.

7) **Shall** be supported when the Fast Role Swap signal is supported.

8) **Shall** be supported when *VCONN_Swap* is supported.

9) **Shall** be supported when SPR PPS is supported.

10) **Shall** be supported when required by a country authority.

11) **Shall** be supported by *Active Cables*.

12) VPD includes CT-VPDs when not Connected to a Charger. PD communication with a CT-VPD *Shall* only take place when not Connected to a Charger.

13) **Shall** be supported by products that support *[USB4]*.

14) **Shall** be supported by all Sources except single port chargers with invariant PDOs.

(m). Section 7.3

From:

USB Power Delivery ENGINEERING CHANGE NOTICE

7.3 Transitions

The following sections illustrate the power supply's response to various types of negotiations. The negotiations are triggered by certain Messages or Signaling. It provides examples of the transitions and is organized around each of the Messages and Signals that result in a response from the power supply. The response to a Message or Signal can result in different transitions depending upon the power supply's starting conditions and the requested change.

⋮

- Transitions caused by the **GotoMin** Message:
 - Sink decreases its current draw to pre-negotiated minimum.

⋮

To:

7.3 Transitions

The following sections illustrate the power supply's response to various types of negotiations. The negotiations are triggered by certain Messages or Signaling. It provides examples of the transitions and is organized around each of the Messages and Signals that result in a response from the power supply. The response to a Message or Signal can result in different transitions depending upon the power supply's starting conditions and the requested change.

⋮

- ~~Transitions caused by the **GotoMin** Message:~~
 - ~~Sink decreases its current draw to pre-negotiated minimum.~~

⋮

(n). Section 7.3.3

From:

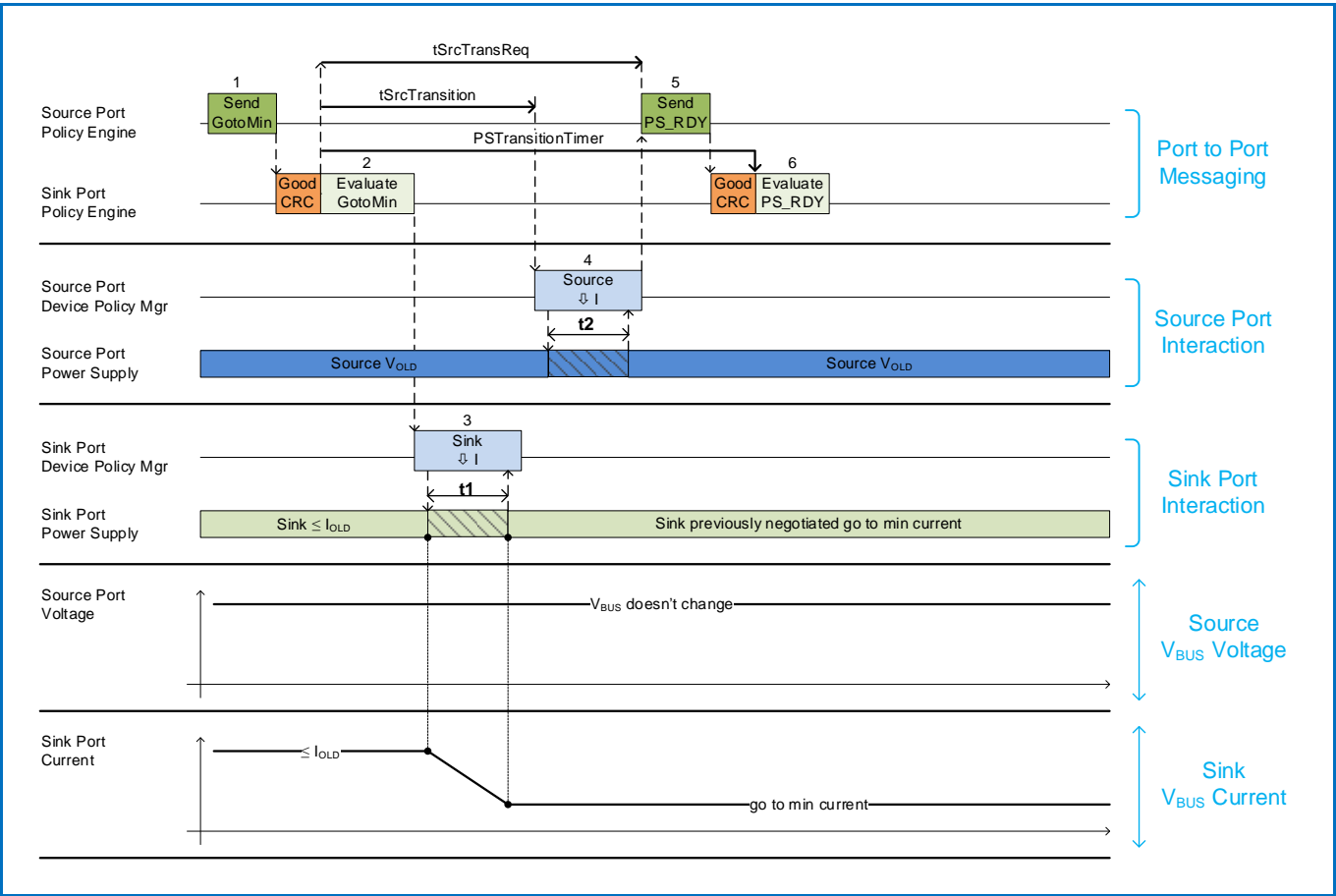
7.3.3 Transitions Caused by GotoMin

7.3.3.1 GotoMin Current Decrease

The interaction of the System Policy, Device Policy, and power supply that **shall** be followed during a GotoMin current decrease is shown in **Figure 7-43 “Transition Diagram for a GotoMin Current Decrease”**. The sequence that **shall** be followed is described in **Table 7.21 “Sequence Description for a GotoMin Current Decrease”**. The timing parameters that **shall** be followed are listed in **Table 7.25 “Source Electrical Parameters”**, **Table 7.26 “Sink Electrical Parameters”**, and **Table 7.27 “Common Source/Sink Electrical Parameters”**.

USB Power Delivery ENGINEERING CHANGE NOTICE

Figure 7-43 “Transition Diagram for a GotoMin Current Decrease”



USB Power Delivery ENGINEERING CHANGE NOTICE

Table 7.21 “Sequence Description for a GotoMin Current Decrease”

Step	Source Port	Sink Port
1	Policy Engine sends the GotoMin Message to the Sink.	Policy Engine receives the GotoMin Message.
2	Protocol Layer receives the GoodCRC Message from the Sink. The Policy Engine tells the Device Policy Manager to instruct the power supply to modify its output power.	Protocol Layer sends the GoodCRC Message to the Source. Policy Engine then starts the PSTransitionTimer and evaluates the GotoMin Message.
3		Policy Engine tells the Device Policy Manager to instruct the power supply to reduce power consumption, within tSnkNewPower (t1), to the pre-negotiated go to reduced power level); t1 Shall complete before tSrcTransition min. The Sink Shall Not violate the transient load behavior defined in Section 7.2.6 “Transient Load Behavior” while transitioning to and operating at the new power level.
4	tSrcTransition after the GoodCRC Message was received the power supply starts to change its output power capability. The power supply Shall be ready to operate at the new power level within tSrcReady (t2). The power supply informs the Device Policy Manager that it is ready to operate at the new power level. The power supply status is passed to the Policy Engine.	
5	The Policy Engine sends the PS_RDY Message to the Sink starting within tSrcTransReq of the end of the GoodCRC Message following the Accept message.	The Policy Engine receives the PS_RDY Message.
6	Protocol Layer receives the GoodCRC Message from the Sink.	Protocol Layer sends the GoodCRC Message to the Source. Policy Engine then stops the PSTransitionTimer and evaluates the PS_RDY Message from the Source and no further action is required. If the PS_RDY Message is not received before PSTransitionTimer times out the Sink sends Hard Reset signaling.

To:

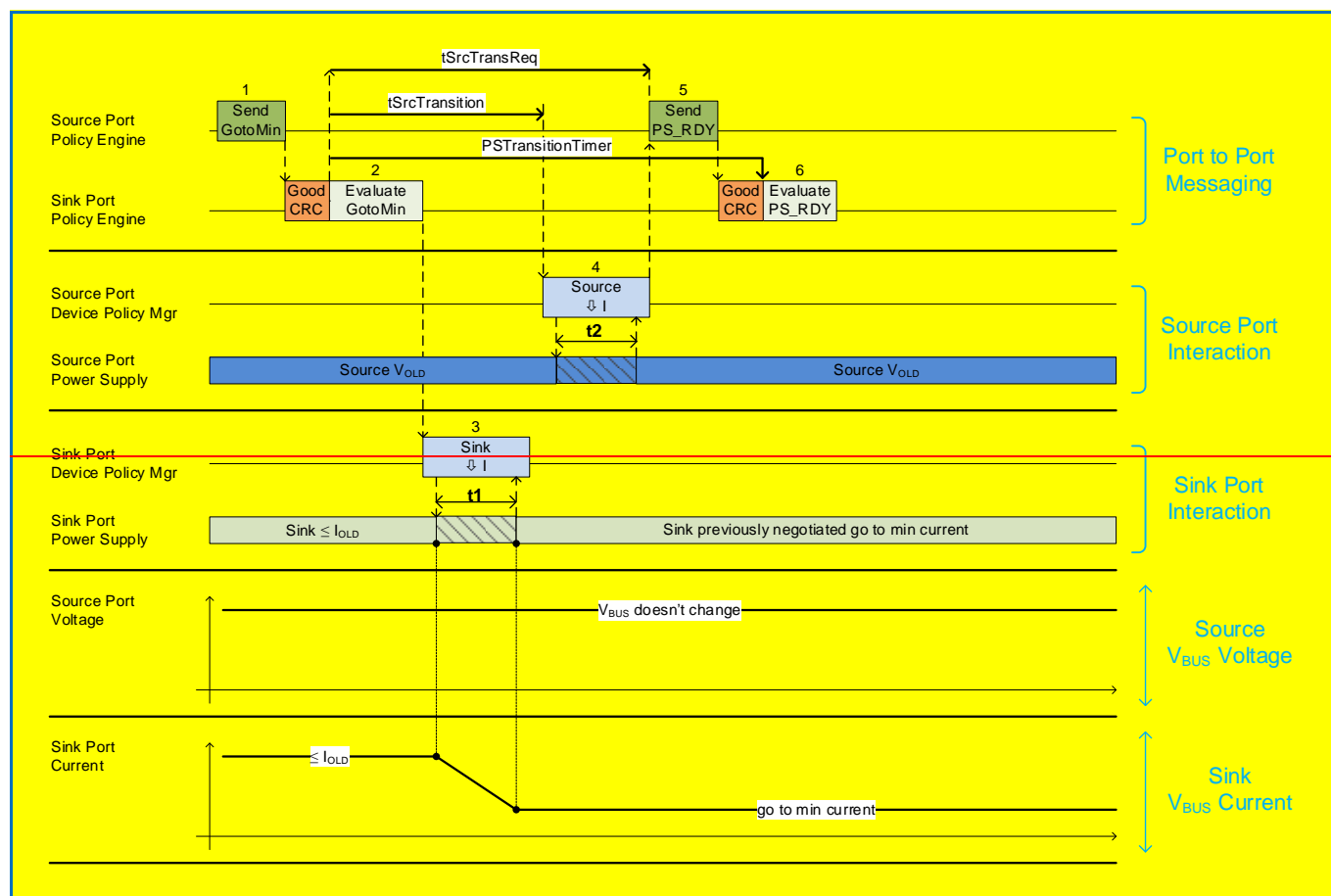
7.3.3 Transitions Caused by GotoMin

7.3.3.1 GotoMin Current Decrease

The interaction of the System Policy, Device Policy, and power supply that **Shall** be followed during a GotoMin current decrease is shown in **Figure 7-43 “Transition Diagram for a GotoMin Current Decrease”**. The sequence that **Shall** be followed is described in **Table 7.21 “Sequence Description for a GotoMin Current Decrease”**. The timing parameters that **Shall** be followed are listed in **Table 7.25 “Source Electrical Parameters”**, **Table 7.26 “Sink Electrical Parameters”**, and **Table 7.27 “Common Source/Sink Electrical Parameters”**.

USB Power Delivery ENGINEERING CHANGE NOTICE

Figure 7-43 "Transition Diagram for a GotoMin Current Decrease"



USB Power Delivery ENGINEERING CHANGE NOTICE

Table 7.21 “Sequence Description for a GotoMin Current Decrease”

Step	Source Port	Sink Port
1	Policy Engine sends the GotoMin Message to the Sink.	Policy Engine receives the GotoMin Message.
2	Protocol Layer receives the GoodCRC Message from the Sink. The Policy Engine tells the Device Policy Manager to instruct the power supply to modify its output power.	Protocol Layer sends the GoodCRC Message to the Source. Policy Engine then starts the PSTransitionTimer and evaluates the GotoMin Message.
3		Policy Engine tells the Device Policy Manager to instruct the power supply to reduce power consumption, within tSnkNewPower (t1), to the pre-negotiated go to reduced power level); t1 Shall complete before tSrcTransition min. The Sink Shall Not violate the transient load behavior defined in Section 7.2.6 “Transient Load Behavior” while transitioning to and operating at the new power level.
4	tSrcTransition after the GoodCRC Message was received the power supply starts to change its output power capability. The power supply Shall be ready to operate at the new power level within tSrcReady (t2). The power supply informs the Device Policy Manager that it is ready to operate at the new power level. The power supply status is passed to the Policy Engine.	
5	The Policy Engine sends the PS_RDY Message to the Sink starting within tSrcTransReq of the end of the GoodCRC Message following the Accept message.	The Policy Engine receives the PS_RDY Message.
6	Protocol Layer receives the GoodCRC Message from the Sink.	Protocol Layer sends the GoodCRC Message to the Source. Policy Engine then stops the PSTransitionTimer and evaluates the PS_RDY Message from the Source and no further action is required. If the PS_RDY Message is not received before PSTransitionTimer times out the Sink sends Hard Reset signaling.

(o). Sections 8.2.5 and 8.2.5.1

From:

8.2.5 Managing Power Requirements

The Device Policy Manager in a Provider **Shall** be aware of the power requirements of all devices connected to its Source Ports. This includes being aware of any reserve power that might be required by devices in the future and ensuring that power is shared optimally amongst Attached PD Capable devices. This is a key function of the Device Policy Manager; whose implementation is critical to ensuring that all PD Capable devices get the power they require in a timely fashion in order to facilitate smooth operation. This is balanced by the fact that the Device Policy Manager is responsible for managing the sources of power that are, by definition, finite.

The Consumer’s Device Policy Manager **Shall** ensure that it takes no more power than is required to perform its functions and gives back unneeded power whenever possible (in such cases the Provider **Shall** maintain a Power Reserve to ensure future operation is possible).

USB Power Delivery ENGINEERING CHANGE NOTICE

8.2.5.1 Managing the Power Reserve

There might be some products where a Device has certain functionality at one power level and a greater functionality at another, for example a Printer/Scanner that operates only as a printer with one power level and as a scanner if it can get more power. Visibility of the linkage between power and functionality will only be apparent at the USB Host; however, the Device Policy Manager provides the mechanisms to manage the power requirements of such Devices.

Devices with the GiveBack flag cleared report Operating Current and Maximum Operating Current (see [Section 6.4.2.2 “GiveBack Flag”](#)). For many Devices the Operating Current and the Maximum Operating Current will be the same. Devices with highly variable loads, such as Hard Disk Drives, might use Maximum Operating Current.

Devices with the GiveBack flag set report Operating Current and Minimum Operating Current (see [Section 6.4.2.2 “GiveBack Flag”](#)). For many Devices the Operating Current and the Minimum Operating Current will be the same. Devices that charge their own batteries might use the Minimum Operating Current and GiveBack flag.

For example, in the first case, a mobile device might require 500mA to operate, but would like an additional 1000mA to charge its Battery. The mobile device would set the GiveBack flag (see [Section 6.4.2.2 “GiveBack Flag”](#)) and request 500mA in the Minimum Operating Current field and 1500mA in the Operating Current field (provided that 1500mA was offered by the Source) indicating to the Provider that it could temporarily recover the 1000mA to meet a transitory request.

In the second case, a Hard Disk Drive (HDD) might require 2A to spin-up, but only 1A to operate. At startup the HDD would request Maximum Operating Current of 2A and an Operating Current of 2A. After the drive is spun-up and ready to operate it would make another request of 1A for its Operating Current and 2A for its Maximum Operating Current. Over time, its inactivity timers might expire, and the HDD will go to a lower power state. When the HDD is next accessed, it has to spin-up again. So, it will request an Operating Current of 2A and a Maximum Operating Current of 2A. The Provider might have the extra power available immediately and can immediately honor the request. If the power is not available, the Provider might have to harvest power, for example use the [GotoMin](#) Message to get back some power before honoring the HDD’s request. In such a case, the HDD would be told to wait via a [Wait](#) Message. The HDD continues to Request additional power until the request is finally granted.

It **Shall** be the Device Policy Manager’s responsibility to allocate power and maintain a Power Reserve so as not to over-subscribe its available power resource. A Device with multiple ports such as a Hub **Shall** always be able to meet the incremental demands of the Port requiring the highest incremental power from its Power Reserve.

The [GotoMin](#) Message is designed to allow the Provider to reclaim power from one Port to support a Consumer on another Port that temporarily requires additional power to perform some short-term operation. In the example above, the mobile device that is being charged reduces its charge rate to allow a Device Policy Manager to meet a request from an HDD for start-up current required to spin-up its platters. Any power which is available to be reclaimed using a [GotoMin](#) Message **May** be counted as part of the Power Reserve.

A Consumer requesting power **Shall** take into account its operational requirements when advertising its ability to temporarily return power. For example, a mobile device with a Dead Battery that is being used to make a call **Should** make a request that retains sufficient power to continue the call. When the Consumer’s requirements change, it **Shall** re-negotiate its power to reflect the changed requirements.

USB Power Delivery ENGINEERING CHANGE NOTICE

To:

8.2.5 Managing Power Requirements

~~The~~It is the responsibility of the Device Policy Manager in a Provider ~~shall~~to be aware of the power requirements of all devices connected to its Source Ports. This includes being aware of any reserve power that might be required by devices in the future and ensuring that power is shared optimally amongst Attached PD Capable devices. This is a key function of the Device Policy Manager, whose implementation is critical to ensuring that all PD Capable devices get the power they require in a timely fashion in order to facilitate smooth operation. This is balanced by the fact that the Device Policy Manager is responsible for managing the sources of power that are, by definition, finite.

The Consumer's Device Policy Manager **shall** ensure that it takes no more power than is required to perform its functions and ~~when its requirements change, it **Should** make a new Request. gives back unneeded power whenever possible (in such cases ~~the~~ The Provider, after satisfying the Request, **Should** reclaim any unused power to ensure that it can meet total power requirements of Attached Sinks on at least one Portshall maintain a Power Reserve to ensure future operation is possible).~~

[Note: It is expected that a future design guide will provide additional guidance.]

8.2.5.1 Managing the Power Reserve

There might be some products where a Device has certain functionality at one power level and a greater functionality at another, for example a Printer/Scanner that operates only as a printer with one power level and as a scanner if it can get more power. ~~While the Vi~~isibility of the linkage between power and functionality ~~will~~might only be apparent ~~at~~to the USB Host; ~~however,~~ the Device Policy Manager **Should** provide ~~s the~~ mechanisms to manage the power requirements of such Devices.

~~Devices with the GiveBack flag cleared report Operating Current and Maximum Operating Current (see Section 6.4.2.2 "GiveBack Flag"). For many Devices the Operating Current and the Maximum Operating Current will be the same. Devices with highly variable loads, such as Hard Disk Drives, might use Maximum Operating Current.~~

~~Devices with the GiveBack flag set report Operating Current and Minimum Operating Current (see Section 6.4.2.2 "GiveBack Flag"). For many Devices the Operating Current and the Minimum Operating Current will be the same. Devices that charge their own batteries might use the Minimum Operating Current and GiveBack flag.~~

~~For example, in the first case, a mobile device might require 500mA to operate, but would like an additional 1000mA to charge its Battery. The mobile device would set the GiveBack flag (see Section 6.4.2.2 "GiveBack Flag") and request 500mA in the Minimum Operating Current field and 1500mA in the Operating Current field (provided that 1500mA was offered by the Source) indicating to the Provider that it could temporarily recover the 1000mA to meet a transitory request.~~

~~In the second case, a Hard Disk Drive (HDD) might require 2A to spin-up, but only 1A to operate. At startup the HDD would request Maximum Operating Current of 2A and an Operating Current of 2A. After the drive is spun-up and ready to operate it would make another request of 1A for its Operating Current and 2A for its Maximum Operating Current. Over time, its inactivity timers might expire, and the HDD will go to a lower power state. When the HDD is next accessed, it has to spin-up again. So, it will request an Operating Current of 2A and a Maximum Operating Current of 2A. The Provider might have the extra power available immediately and can immediately honor the request. If the power is not available, the Provider might have to harvest power, for example use the **GotoMin** Message to get back some power before honoring the HDD's request. In such a case, the HDD would be told to wait via a **Wait** Message. The HDD continues to Request additional power until the request is finally granted.~~

USB Power Delivery ENGINEERING CHANGE NOTICE

It ~~shall~~ be the Device Policy Manager's responsibility to allocate power and maintain a P_{power} R_{reserve} so as not to over-subscribe its available power resource. A Device with multiple ports such as a Hub **shall** always be able to attempt to meet the incremental demands of the Port requiring the highest incremental power from its P_{power} R_{reserve}.

The **GotoMin** Message is designed to allow the Provider to reclaim power from one Port to support a Consumer on another Port that temporarily requires additional power to perform some short-term operation. In the example above, the mobile device that is being charged reduces its charge rate to allow a Device Policy Manager to meet a request from an HDD for start-up current required to spin-up its platters. Any power which is available to be reclaimed using a **GotoMin** Message **May** be counted as part of the Power Reserve.

A Consumer requesting power **shall** take into account its operational requirements when advertising its ability to temporarily return power. For example, a mobile device with a Dead Battery that is being used to make a call **should** make a request that retains sufficient power to continue the call. When the Consumer's requirements change, it **shall** re-negotiate its power to reflect the changed requirements.

(p). Section 8.3.2.1.3.1

From:

USB Power Delivery ENGINEERING CHANGE NOTICE

Table 8.5 “AMS: Power Negotiation (SPR)”

AMS	Interruptible	Message Sequence	Conditions	AMS Ref
SPR Explicit Contract Negotiation (Accept)	1) <i>Source_Capabilities</i> Message 2) <i>Request</i> Message 3) <i>Accept</i> Message 4) <i>PS_RDY</i> Message	Started by Source, SPR Mode	<i>Section 8.3.2.2.1.1.1</i>	<i>Section 8.3.3.2, Section 8.3.3.3</i>
SPR Explicit Contract Negotiation (Reject)	1) <i>Source_Capabilities</i> Message 2) <i>Request</i> Message 3) <i>Reject</i> Message		<i>Section 8.3.2.2.1.1.2</i>	
SPR Explicit Contract Negotiation (Wait)	1) <i>Source_Capabilities</i> Message 2) <i>Request</i> Message 3) <i>Wait</i> Message		<i>Section 8.3.2.2.1.1.3</i>	
Reclaiming Power with GotoMin Message	4) <i>GotoMin</i> Message 5) <i>PS_RDY</i> Message		<i>Section 8.3.2.2.1.2</i>	
SPR PPS Keep Alive	1) <i>Request</i> Message 2) <i>Accept</i> Message 3) <i>PS_RDY</i> Message	Started by Sink, SPR Mode	<i>Section 8.3.2.2.1.3</i>	<i>Section 8.3.3.3</i>
SPR Sink Makes Request (Accept)	1) <i>Request</i> Message 2) <i>Accept</i> Message 3) <i>PS_RDY</i> Message		<i>Section 8.3.2.2.1.4.1</i>	
SPR Sink Makes Request (Reject)	1) <i>Request</i> Message 2) <i>Reject</i> Message		<i>Section 8.3.2.2.1.4.2</i>	
SPR Sink Makes Request (Wait)	1) <i>Request</i> Message 2) <i>Wait</i> Message		<i>Section 8.3.2.2.1.4.30</i>	

USB Power Delivery ENGINEERING CHANGE NOTICE

To:

Table 8.5 “AMS: Power Negotiation (SPR)”

AMS	Interruptible	Message Sequence	Conditions	AMS Ref
SPR Explicit Contract Negotiation (Accept)	3) <i>Source_Capabilities</i> Message 4) <i>Request</i> Message 5) <i>Accept</i> Message 6) <i>PS_RDY</i> Message	Started by Source, SPR Mode	<i>Section 8.3.2.2.1.1.1</i>	<i>Section 8.3.3.2, Section 8.3.3.3</i>
SPR Explicit Contract Negotiation (Reject)	7) <i>Source_Capabilities</i> Message 8) <i>Request</i> Message 9) <i>Reject</i> Message		<i>Section 8.3.2.2.1.1.2</i>	
SPR Explicit Contract Negotiation (Wait)	10) <i>Source_Capabilities</i> Message 11) <i>Request</i> Message 12) <i>Wait</i> Message		<i>Section 8.3.2.2.1.1.3</i>	
Reclaiming Power with GotoMin Message	13) <i>GotoMin</i> Message 14) <i>PS_RDY</i> Message		<i>Section 8.3.2.2.1.2</i>	
SPR PPS Keep Alive	15) <i>Request</i> Message 16) <i>Accept</i> Message 17) <i>PS_RDY</i> Message	Started by Sink, SPR Mode	<i>Section 8.3.2.2.1.3</i>	<i>Section 8.3.3.3</i>
SPR Sink Makes Request (Accept)	18) <i>Request</i> Message 19) <i>Accept</i> Message 20) <i>PS_RDY</i> Message		<i>Section 8.3.2.2.1.4.1</i>	
SPR Sink Makes Request (Reject)	21) <i>Request</i> Message 22) <i>Reject</i> Message		<i>Section 8.3.2.2.1.4.2</i>	
SPR Sink Makes Request (Wait)	23) <i>Request</i> Message 24) <i>Wait</i> Message		<i>Section 8.3.2.2.1.4.30</i>	

(q). Section 8.3.2.2.1.2

From:

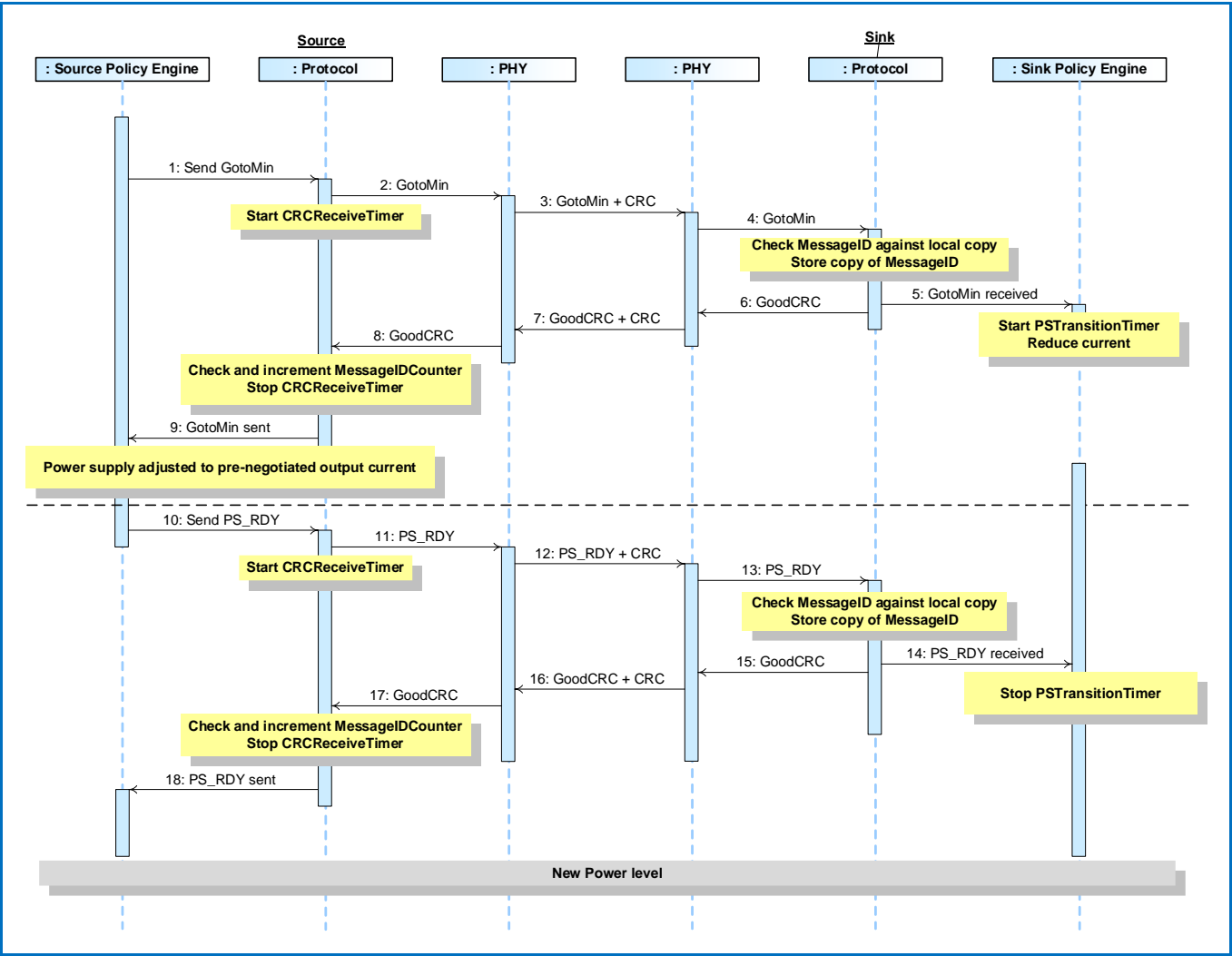
8.3.2.2.1.2

Reclaiming Power with GotoMin Message

This is an example of a GotoMin operation. *Figure 8-8 “Successful GotoMin operation”* shows the Messages as they flow across the bus and within the devices to accomplish the GotoMin.

USB Power Delivery ENGINEERING CHANGE NOTICE

Figure 8-8 “Successful GotoMin operation”



USB Power Delivery ENGINEERING CHANGE NOTICE

Table 8.36 “Steps for a GotoMin Negotiation” provides a detailed explanation of what happens at each labeled step in Figure 8-8 “Successful GotoMin operation” above.

Table 8.36 “Steps for a GotoMin Negotiation”

Step	Source	Sink
1	Policy Engine tells the Protocol Layer to form a <i>GotoMin</i> Message.	
2	The Protocol Layer forms the <i>GotoMin</i> Message that is passed to the Physical Layer.	
3	Physical Layer appends CRC and sends the <i>GotoMin</i> Message. Starts <i>CRCReceiveTimer</i> .	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.
4		Physical Layer forwards the <i>GotoMin</i> Message to the Protocol Layer.
5		Protocol Layer checks the <i>MessageID</i> in the incoming Message is different from the previously stored value and then stores a copy of the new value. Protocol Layer informs the Policy Engine that a <i>GotoMin</i> Message has been received. The Policy starts the <i>PSTransitionTimer</i> and reduces its current draw. The Policy Engine prepares the Power supply for transition to the new power level.
6		The Protocol Layer generates a <i>GoodCRC</i> Message and passes it to its Physical Layer.
7	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.	Physical Layer appends CRC and sends the Message.
8	Physical Layer forwards the <i>GoodCRC</i> Message to the Protocol Layer. The Protocol Layer verifies and increments the <i>MessageIDCounter</i> and stops the <i>CRCReceiveTimer</i> .	
9	The Protocol Layer informs the Policy Engine that a <i>GotoMin</i> Message was successfully sent.	
Power supply Adjusts its Output to the Negotiated Value		
10	Policy Engine sees the power supply has settled at the new operating condition and tells the Protocol Layer to send a <i>PS_RDY</i> Message.	
11	The Protocol Layer forms the <i>PS_RDY</i> Message.	
12	Physical Layer appends CRC and sends the <i>PS_RDY</i> Message. Starts <i>CRCReceiveTimer</i> .	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.
13		Physical Layer forwards the <i>PS_RDY</i> Message to the Protocol Layer.

USB Power Delivery ENGINEERING CHANGE NOTICE

14		Protocol Layer checks the <i>MessageID</i> in the incoming Message is different from the previously stored value and then stores a copy of the new value. Protocol Layer informs the Policy Engine that a <i>PS_RDY</i> Message has been received. The Policy Engine stops the <i>PSTransitionTimer</i> .
15		The Protocol Layer generates a <i>GoodCRC</i> Message and passes it to its Physical Layer.
16	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.	Physical Layer appends CRC and sends the Message.
17	Physical Layer forwards the <i>GoodCRC</i> Message to the Protocol Layer. The Protocol Layer verifies and increments the <i>MessageIDCounter</i> and stops the <i>CRCReceiveTimer</i> .	
18	The Protocol Layer informs the Policy Engine that the <i>PS_RDY</i> Message was successfully sent.	
New Power Level Negotiated		

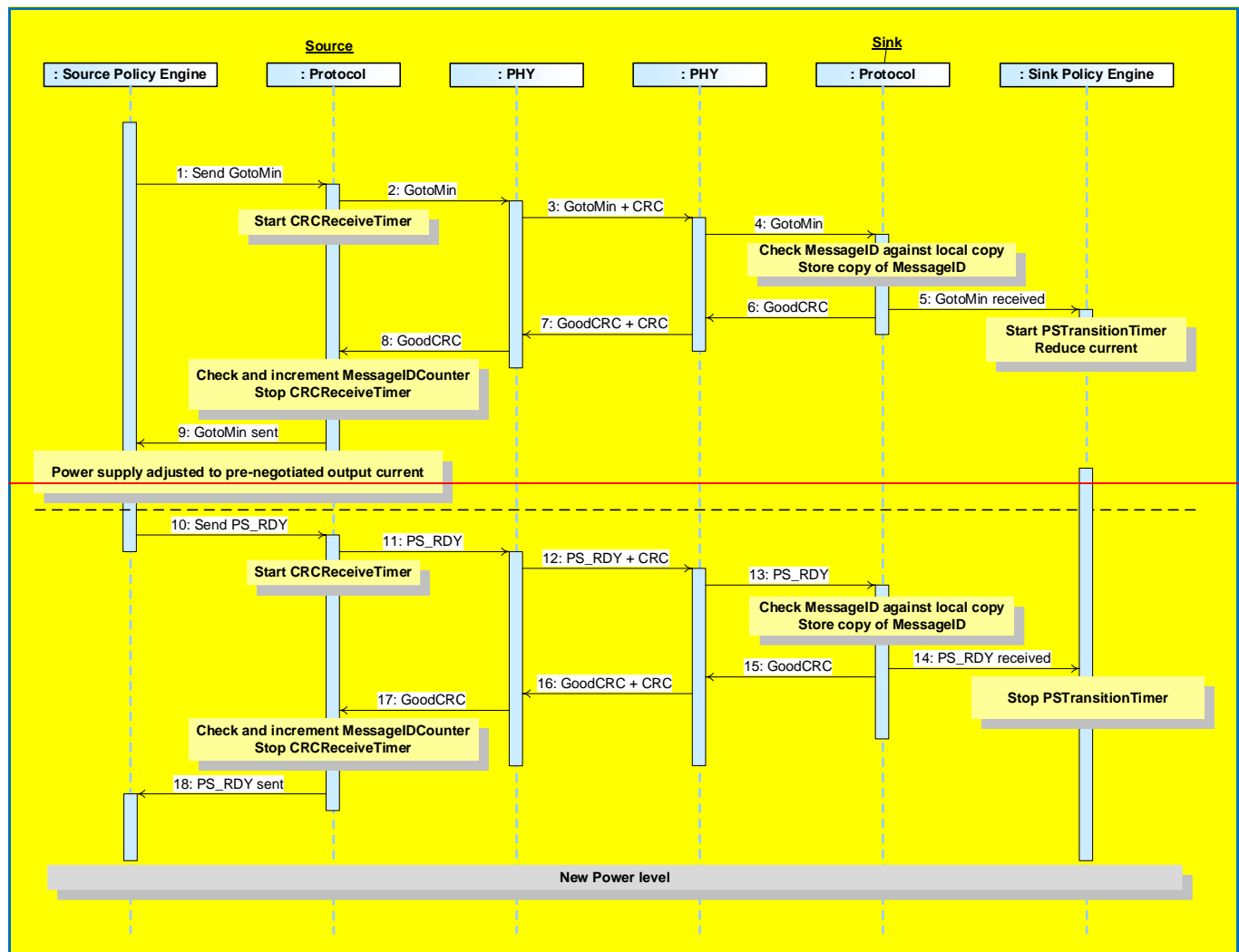
To:

8.3.2.2.1.2 Reclaiming Power with GotoMin Message

This is an example of a GotoMin operation. **Figure 8-8 “Successful GotoMin operation”** shows the Messages as they flow across the bus and within the devices to accomplish the GotoMin.

USB Power Delivery ENGINEERING CHANGE NOTICE

Figure 8-8 “Successful GotoMin operation”



USB Power Delivery ENGINEERING CHANGE NOTICE

Table 8.36 “Steps for a GotoMin Negotiation” provides a detailed explanation of what happens at each labeled step in Figure 8-8 “Successful GotoMin operation” above.

Table 8.36 “Steps for a GotoMin Negotiation”

Step	Source	Sink
1	Policy Engine tells the Protocol Layer to form a GotoMin Message.	
2	The Protocol Layer forms the GotoMin Message that is passed to the Physical Layer.	
3	Physical Layer appends CRC and sends the GotoMin Message. Starts CRCReceiveTimer .	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.
4		Physical Layer forwards the GotoMin Message to the Protocol Layer.
5		Protocol Layer checks the MessageID in the incoming Message is different from the previously stored value and then stores a copy of the new value. Protocol Layer informs the Policy Engine that a GotoMin Message has been received. The Policy starts the PSTransitionTimer and reduces its current draw. The Policy Engine prepares the Power supply for transition to the new power level.
6		The Protocol Layer generates a GoodCRC Message and passes it to its Physical Layer.
7	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.	Physical Layer appends CRC and sends the Message.
8	Physical Layer forwards the GoodCRC Message to the Protocol Layer. The Protocol Layer verifies and increments the MessageIDCounter and stops the CRCReceiveTimer .	
9	The Protocol Layer informs the Policy Engine that a GotoMin Message was successfully sent.	
Power supply Adjusts its Output to the Negotiated Value		
10	Policy Engine sees the power supply has settled at the new operating condition and tells the Protocol Layer to send a PS_RDY Message.	
11	The Protocol Layer forms the PS_RDY Message.	
12	Physical Layer appends CRC and sends the PS_RDY Message. Starts CRCReceiveTimer .	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.
13		Physical Layer forwards the PS_RDY Message to the Protocol Layer.

USB Power Delivery ENGINEERING CHANGE NOTICE

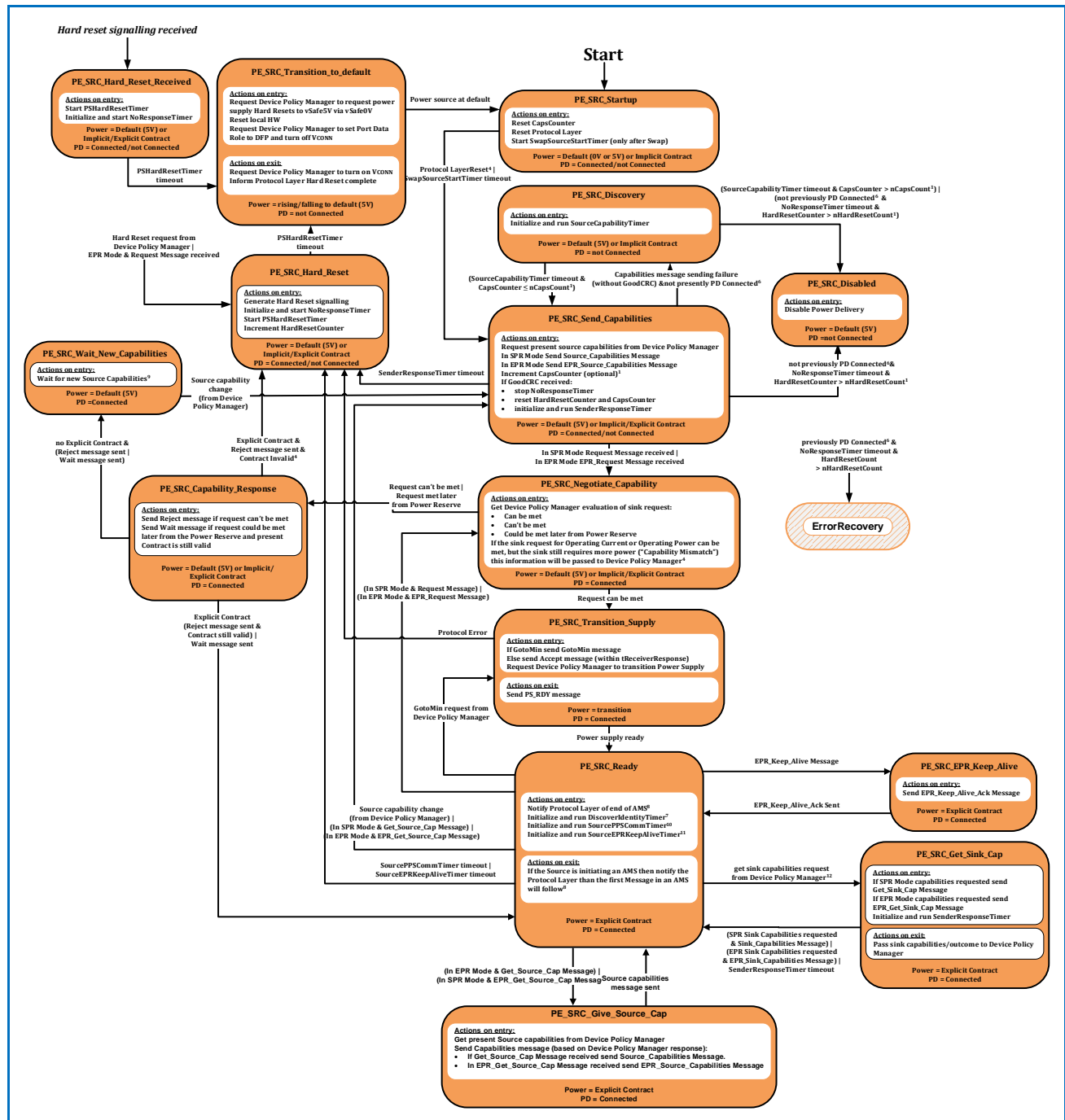
14		Protocol Layer checks the MessageID in the incoming Message is different from the previously stored value and then stores a copy of the new value. Protocol Layer informs the Policy Engine that a PS_RDY Message has been received. The Policy Engine stops the PSTransitionTimer .
15		The Protocol Layer generates a GoodCRC Message and passes it to its Physical Layer.
16	Physical Layer receives the Message and compares the CRC it calculated with the one sent to verify the Message.	Physical Layer appends CRC and sends the Message.
17	Physical Layer forwards the GoodCRC Message to the Protocol Layer. The Protocol Layer verifies and increments the MessageIDCounter and stops the CRCReceiveTimer .	
18	The Protocol Layer informs the Policy Engine that the PS_RDY Message was successfully sent.	
New Power Level Negotiated		

(r). Section 8.3.3.2, Figure 8-134

From:

USB Power Delivery ENGINEERING CHANGE NOTICE

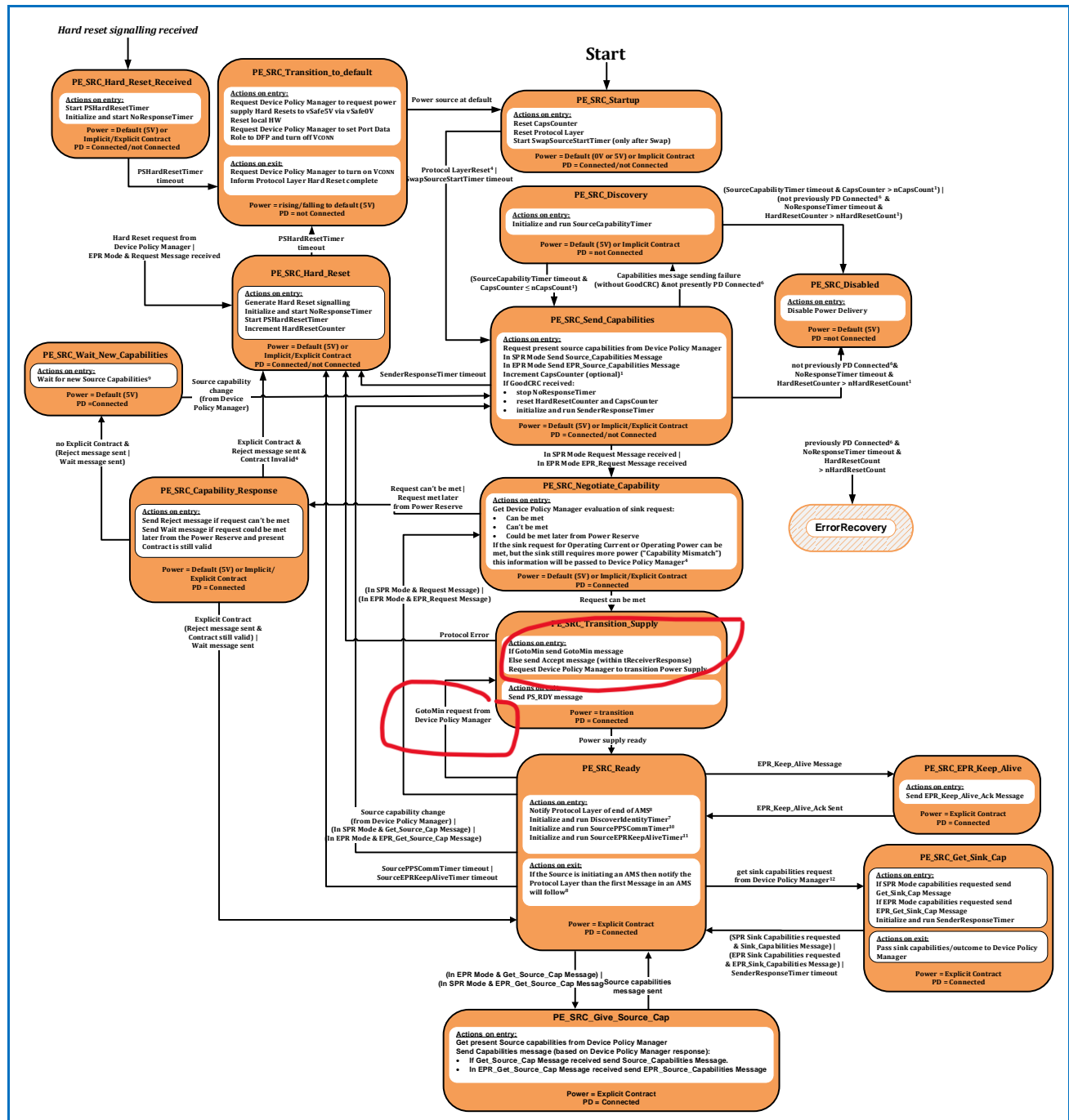
Figure 8-134 "Source Port State Diagram"



USB Power Delivery ENGINEERING CHANGE NOTICE

To: need to remove GotoMin from figure

Figure 8-134 "Source Port State Diagram"



(s). Section 8.3.3.2.5 and 8.3.3.2.6

USB Power Delivery ENGINEERING CHANGE NOTICE

From:

8.3.3.2.5 PE_SRC_Transition_Supply State

⋮

On entry to the **PE_SRC_Transition_Supply** state, the Policy Engine **Shall** request the Protocol Layer to either send a **GotoMin** Message (if this was requested by the Device Policy Manager) or otherwise an **Accept** Message and inform the Device Policy Manager that it **Shall** transition the power supply to the Requested power level. Note: that if the power supply is currently operating at the requested power no change will be necessary.

⋮

8.3.3.2.6 PE_SRC_Ready State

⋮

The Policy Engine **Shall** transition to the **PE_SRC_Transition_Supply** state when:

- A GotoMin request is received from the Device Policy Manager for the Attached Device to go to minimum power.

⋮

To:

8.3.3.2.5 PE_SRC_Transition_Supply State

⋮

On entry to the **PE_SRC_Transition_Supply** state, the Policy Engine **Shall** request the Protocol Layer to ~~either~~ send a **GotoMin** Message (if this was requested by the Device Policy Manager) or otherwise an **Accept** Message and inform the Device Policy Manager that it **Shall** transition the power supply to the Requested power level. Note: that if the power supply is currently operating at the requested power no change will be necessary.

⋮

8.3.3.2.6 PE_SRC_Ready State

⋮

~~The Policy Engine **Shall** transition to the **PE_SRC_Transition_Supply** state when:~~

- ~~• A GotoMin request is received from the Device Policy Manager for the Attached Device to go to minimum power.~~

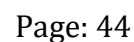
⋮

(t). Section 8.3.3.3, Figure 8-135

From:

USB Power Delivery

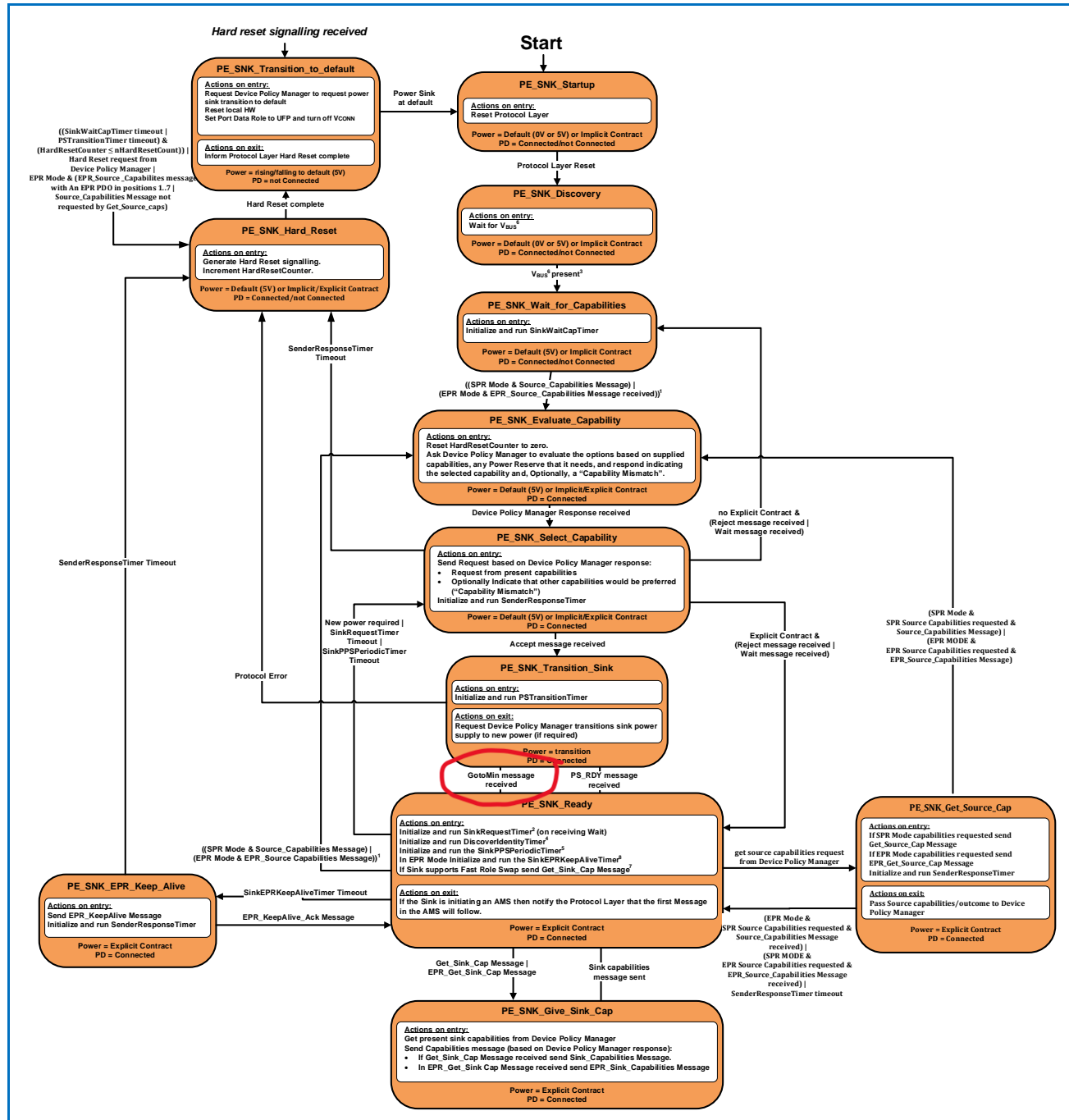
ECN Form 20231101



USB Power Delivery ENGINEERING CHANGE NOTICE

To: **need to remove GotoMin from figure**

Figure 8-135 "Sink Port State Diagram"



(u). Section 8.3.3.3.7

USB Power Delivery ENGINEERING CHANGE NOTICE

From:

8.3.3.3.7 PE_SNK_Ready State

:

The Policy Engine **shall** transition to the *PE_SNK_Transition_Sink* state when:

- A *GotoMin* Message is received.

:

To:

:

The Policy Engine **shall** transition to the *PE_SNK_Transition_Sink* state when:

- A *GotoMin* Message is received.

:

(v). Appendix B

From:

USB Power Delivery ENGINEERING CHANGE NOTICE

B. PD Message Sequence Examples

The following examples are intended to show how the Device Policy Manager might operate and the sequence of Power Delivery messaging which will result. The aim of this section is to inform implementer's how some of the mechanisms detailed in this specification might be applied; it does not contain any **Normative** requirements.

All ports are assumed to be Enhanced SuperSpeed capable, with a default operating Voltage of 5V and a unit load of 150mA. This 0.75W is assumed to be enough power to enable an externally powered device to maintain communication over USB and is enough to allow such a device to enumerate but not operate until more power is negotiated.

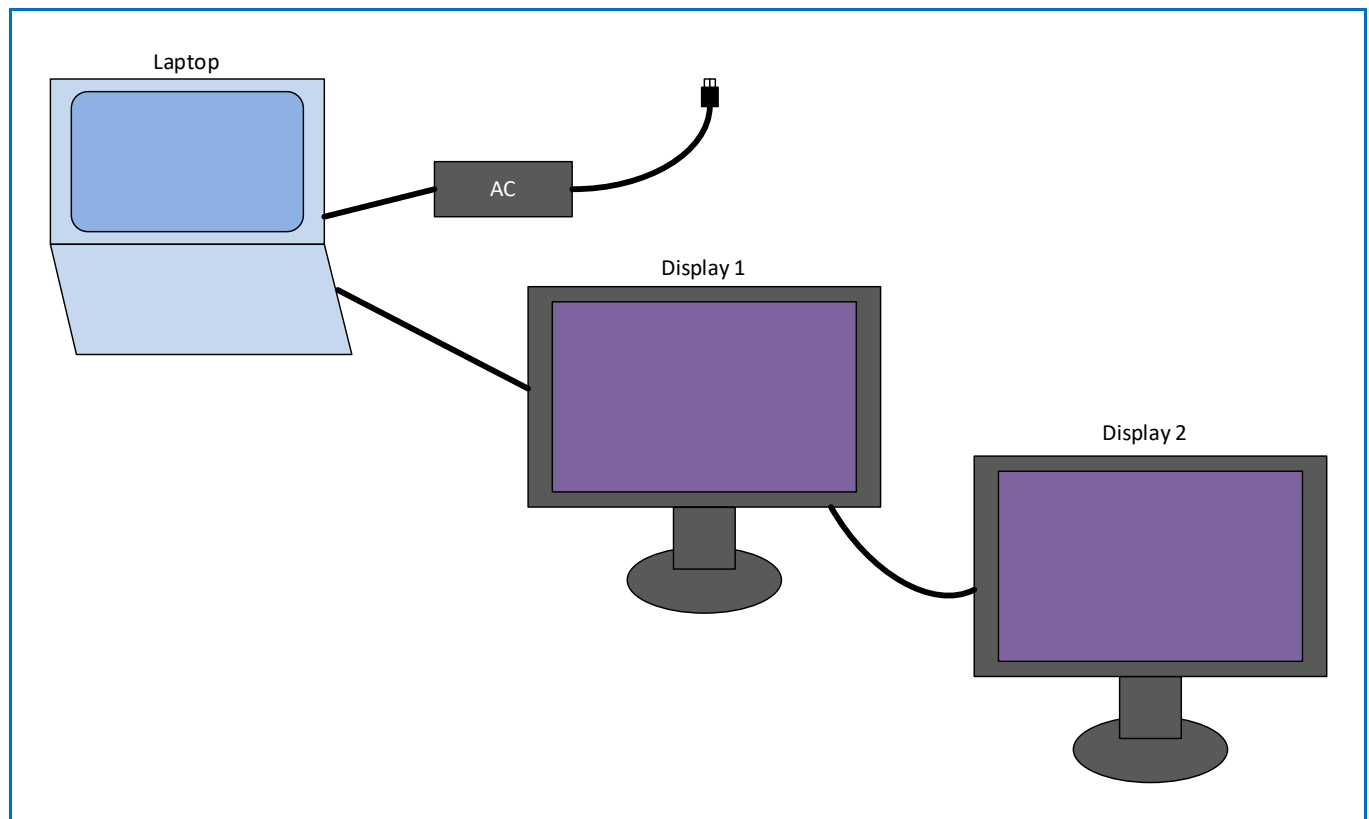
Although the Hubs in these illustrations support Power Delivery on both their UFPs and DFPs this is only one possible Hub implementation.

HDDs are assumed to spin up immediately after they are Attached. This follows the typical operation of current systems.

Ideal power transmission is assumed so that there are no power losses through a device; in practice these would need to be taken into account when requesting power.

B.1 External power is supplied downstream

Figure B-1 "External Power supplied downstream"



USB Power Delivery ENGINEERING CHANGE NOTICE

Configuration:

1. Laptop with an AC supply. AC supply provides sufficient power to charge the laptop and, in addition, to provide up to 60W downstream via its Enhanced SuperSpeed Port. According to the Source Power Rules described in [Section 10.2 "Source Power Rules"](#) this means that the Port has a PD Power of 60W and so can supply: 5V@3A, 9V@3A, 15V@3A and 20V@3A.
2. Display 1 requires 30W to display and therefore a PD Power of 60W to operate itself plus Display 2 connected downstream. Display 1 initially uses 15V@2A to operate itself, since this also allows operation with a Source of 30W PD Power. On connection of Display 2, Display 1 will move to operation at 20V@3A to allow operation of the additional 30W ganged display. According to the Sink Power Rules described in [Section 10.3 "Sink Power Rules"](#) this means that Display 1 requires a Source with a PD Power of 60W to fully operate. Display 1 contains a Hub allowing Display 2 to be connected to Display 1.
3. Display 2 requires 30W operate itself and does not support an additional display connected downstream. Display 1 uses 15V@2A to operate itself from a Source of 30W PD Power.
4. In USB suspend Display 1 and Display 2 will power down but can maintain USB connection using the PD power provided.

Table B-1 External power is supplied downstream

Step	Laptop	Display 1	Display 2	Device Policy Manager	Power (W)
Display 1					
1	Connected to wall supply	Detached	Detached		0
2	Display 1 Attached, V _{BUS} powered.	Attached, drawing 5V@150mA.	Detached		0.75
3	Set of Source Capabilities sent including: 5V@3A (15W), 9V@3A (27W), 15V@3A(45W) and 20V@3A (60W). The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Detached	Laptop determines its Source Capabilities based on its needs and the presence of a wall supply.	0.75
4	Request received	Requests 15V@2A (30W) from laptop	Detached	Display 1 knows it needs 20v@1.5A (30W) for its own operation, evaluates the supplied capabilities and determines that this is available.	0.75
5	Sends Accept	Accept received	Detached	Waiting for PS_RDY before drawing additional power.	0.75
6	Sends PS_RDY	PS_RDY received. Starts drawing 15V@2A. Display 1 turns on and starts operating.	Detached	Laptop evaluates the request, finds that it can meet this and so sends an accept.	30

USB Power Delivery ENGINEERING CHANGE NOTICE

Display 2					
7	Powering Display 1	Detects Attach	Attached, no V _{BUS}		30
8	Request received	Display 1 requests 20V@1.73A (34.6W) from Laptop.	Attached, no V _{BUS}	Display 1 detects Attach and requests additional 4.5W of power for [USB 3.2] Port.	30
9	Sends Accept	Accept received.	Attached, no V _{BUS}		34.6
10	Sends PS_RDY	PS_RDY received	Attached, no V _{BUS}		
11		Powers V _{BUS}	Attached, drawing 5V@150mA.		34.6
12		Sends out Source Capabilities including: 5V@0.9A to Display 2. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Display 1 has 4.5W to allocate to Display 1. This is offered as a standard [USB 3.2] Port.	34.6
13		Request received	Display 2 requests 5V@0.15A but indicates a Capability Mismatch. Display 2 remains off.	Display 2 decides it can manage to run its USB/PD function with 1-unit load but needs more power to function as a display.	34.6
14		Sends Accept	Accept received		34.6
15		Sends PS_RDY	PS_RDY received	Display 2 indicates a capability mismatch to the user.	34.6
16		Get Sink Capabilities sent	Get Sink Capabilities received	Display 1 needs to assess the capability mismatch by first determining what Display 2 actually needs.	34.6
17		Sink Capabilities received	Display 2 returns Sink Capabilities indicating operation at 15V@2A.		34.6
18	Request received	Display 1 requests 20V@3A (60W) from Laptop.		Display1 now knows what Display 2 needs and requests the additional power from the laptop.	34.6
19	Sends Accept	Accept received.			34.6
20	Sends PS_RDY	PS_RDY received		An additional 30W is now available to Display 1 to offer to Display 2.	60

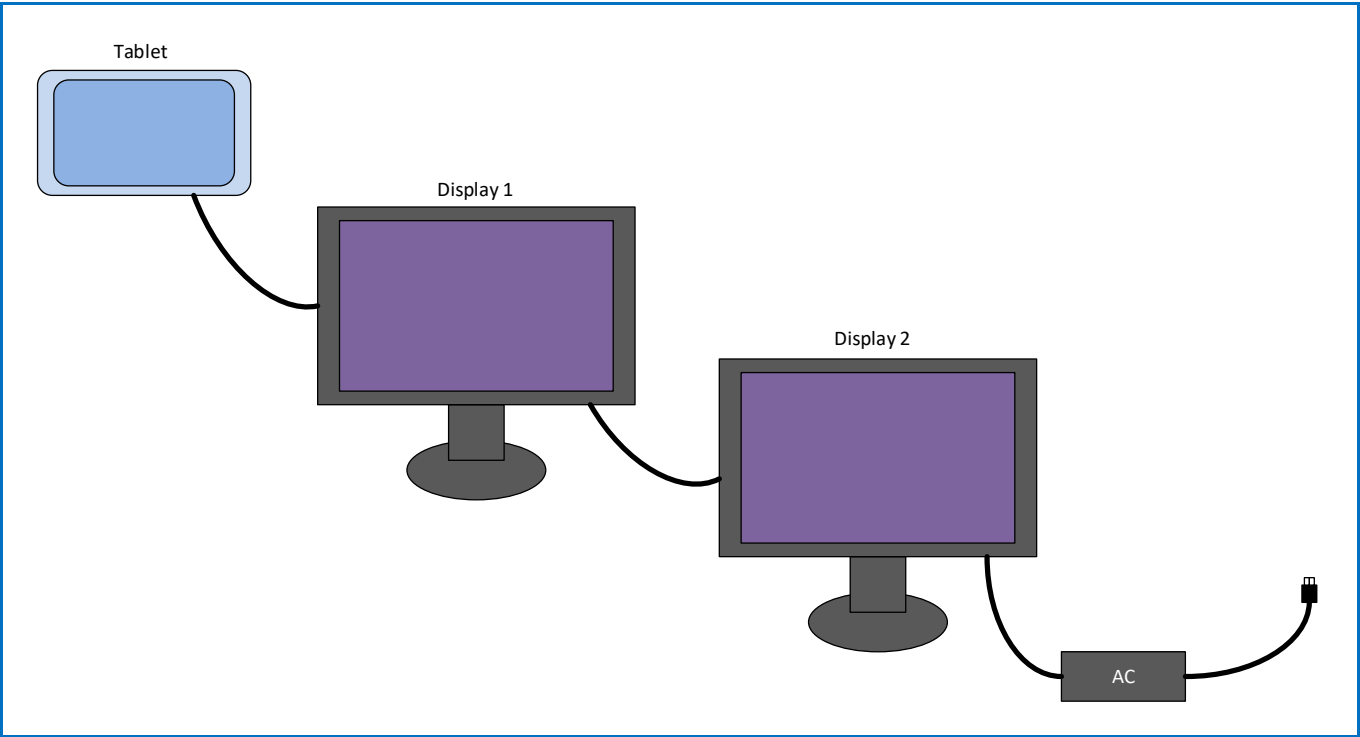
USB Power Delivery ENGINEERING CHANGE NOTICE

21		Sends out Source Capabilities including: 5V@0.9A and 20V@1.5A to Display 2. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Now that Display 1 can power Display 2 correctly this power is offered by Display 1 via a new capabilities Message.	60
22		Request received	Display 2 requests 15V@2A.		60
23		Sends Accept	Accept received	Display 1 determines that the request by Display 2 is within the offered capabilities, so the request is accepted.	60
24		Sends PS_RDY. Drawing 20V@3A from laptop.	PS_RDY received. Starts drawing 15V@2A, turns on and starts operating.	Display 2 now has the power it needs and can start working.	60
USB Suspend					
25	Laptop OS goes into suspend (S3), V_{BUS} remains on, but USB bus is also suspended.	Display 1 turns off but draws 50mW, 25mW to maintain PDUSB Hub functions. The additional 25mW is used to supply the Port used by Display 2.	Display 2 turns off but draws 25mW to maintain USB/PD functions.	No changes in Contract. This is a power reduction purely based on the USB state.	60
26	Laptop OS wakes up. USB is woken up.	Display 1 turns on and returns to drawing 20V@3A.	Display 2 turns on and returns to drawing 15V@2A.	No changes in PD Contract. This purely relates to USB bus state.	60

USB Power Delivery ENGINEERING CHANGE NOTICE

B.2 External power is supplied upstream

Figure B-2 External Power supplied upstream



Configuration:

1. Tablet with no AC supply. Tablet is a USB host and can use 5V@0.2A (1W) during normal operation and up to 5V@2.4A (12W) in order to charge.
2. Display 1 requires 30W to operate and therefore a PD Power of 42W to operate itself and charge the tablet. Display 1 uses 15V@2A to operate itself, since this allows operation with a Source of 30W PD Power and then moves to operation at 20V@2.1A to allow charging of the laptop. According to the Sink Power Rules described in [Section 10.3 "Sink Power Rules"](#) this means that the Display 1 requires a Source with a PD Power of 42W to fully operate.
3. Display 2 has an AC supply connected. AC supply provides sufficient power to power Display 2 and, in addition, to provide up to 60W PD Power upstream.

Table 10.2 External power is supplied upstream

Step	Tablet	Display 1	Display 2	Device Policy Manager	Power (W)
Display 1 – Dead Battery					
1	Detached	Detached	Connected to the wall supply.		0
2		Attached to Display 2	Display 1 Attached		0
3		USB Type-C® Power drawn 5V@1.5A	USB Type-C® Power Advertised 5V@1.5A		0

USB Power Delivery ENGINEERING CHANGE NOTICE

4		Attached to Display 2, drawing 5V@1.5A (7.5W)	Providing 1-unit load to Display 1.		7.5
5		Source Capabilities received	Display2 sends out a set of capabilities including: 5V@3A (15W), 9V@3A (27W), 15V@3A (45W) and 20V@3A (60W). The Unconstrained Power and USB suspend bits are set.	Based on the capabilities of the wall supply and its own needs Display 2 calculates what it can offer upstream.	7.5
6		Display 1 requests 15V@2A (30W) from Display 2.	Request received	Display 1 knows it needs 30W to operate so it requests this amount.	7.5
7		Accept received	Sends Accept	Display 2 accepts the offer since it is within its capabilities.	7.5
8		PS_RDY received. Display 1 starts drawing power and turns on.	Sends PS_RDY	Display 2 indicates its power supply is ready to offer the power.	30
Tablet – Power Role Swap					
9	Tablet is Attached to Display 1.	Attached, V_{BUS} powered.			30
10	Tablet sends out a set of capabilities including: 5V@0.5A (2.5W). The Unconstrained Power bit cleared, and USB suspend bit set.	Capabilities received			30
11	Request received	Display 1 requests 5V@0A from the Tablet. The Unconstrained Power and Dual-Role Power bits are set.		Display 1 has external power providing everything it needs so it does not request any more.	30
12	Sends Accept	Accept received.		No power has been requested from the Tablet, so the tablet has no reason to Reject this.	30
13	Sends PS_RDY	PS_RDY received.		Table completes the Explicit Contract by sending PS_RDY.	30

USB Power Delivery ENGINEERING CHANGE NOTICE

14	Get Sink Capabilities received.	Sends Get Sink Capabilities		Display 1 has access to an external supply so it needs to check whether the Tablet upstream, which has no external supply, could use some power. Display 1 also knows that there is excess capacity, based on the last capabilities it received, which it is not currently using from Display 2.	30
15	The Tablet returns Sink Capabilities indicating that it is a Dual-Role and that it can use 5V@0.2A (1W) as a Sink.	Sink Capabilities received			30
16		Display 1 requests 15V@2.1A (31.5W) from Display 2.	Request received		30
17		Accept received	Sends Accept	Request is within the available power so Display 2 sends an accept.	30
18		PS_RDY received	Sends PS_RDY	Display 2 indicates that the power supply is ready to supply the power.	31.5
19	PR_Swap received	Requests PR_Swap from Tablet.		Display 1 now offers to provide power to the Tablet by initiating a Power Role Swap.	31.5
20	Accept sent. Tablet turns off its V _{BUS} supply.	Accept received.		Tablet is happy to accept a Power Role Swap from any device offering it power.	31.5
21	Send PS_RDY	PS_RDY received. Display 1 turns on its V _{BUS} supply		Tablet indicates that its supply has been turned off.	31.5
22	PS_RDY received.	PS_RDY sent.		Display 1 indicates that its power supply is ready, so the Tablet starts drawing power.	31.5
23	Source Capabilities received	Display 1 sends out a set of capabilities to the Tablet including: 5V@0.48A (2.4W), 12V@0.2A (2.4W) and 20V@0.12 (2.4W). The Unconstrained Power and USB suspend bits are set.			31.5

USB Power Delivery ENGINEERING CHANGE NOTICE

24	The Tablet requests 12V@0.2A.	Request received.		Tablet can now request the power it needs.	31.5
25	Accept received	Accept sent		Power is within the capabilities of Display 1, so it accepts the request.	31.5
26	PS_RDY received. The Tablet starts drawing 12V@0.2A.	PS_RDY sent		Display 1 indicates that its power supply is ready, so the tablet starts drawing the power.	31.5
Tablet – Charge					
27	Tablet requests 12V@0.2A (2.4W) from Display 1. The Tablet needs to charge and so sets the Capability Mismatch bit and the No USB Suspend bit.	Request received.		Tablet needs to charge but the power offered is not sufficient. Since Display 1 claims to have an external supply, the Tablet will try to get more power using the Capability Mismatch Flag.	31.5
28	Accept received	Accept sent		A Valid request has been made so Display 1 accepts the request.	31.5
29	PS_RDY received	PS_RDY received		Tablet indicates a capability mismatch to the user.	31.5
30	Get Sink Capabilities received.	Get Sink Capabilities sent		Due to the Capability Mismatch Flag Display 1 requests Sink Capabilities from the Tablet?	31.5
31	The Tablet returns Sink capabilities containing: 5V@2.4A (12W). The Unconstrained Power bit is cleared.	Sink Capabilities received			31.5
32		Display 1 requests 15V@2.8A (42W) from Display 2. The No Suspend Bit is set to reflect the request from the Tablet.	Request received	Since the Tablet requires an additional 12W of power and Display 1 knows that this is available from Display 2 based on the last Capabilities received so it requests it. In addition, the Request from the Tablet indicated that it wanted No Suspend so this is reflected upwards.	31.5

USB Power Delivery ENGINEERING CHANGE NOTICE

33		Accept received	Sends Accept	Display 2 has 42W available and so accepts the request.	42
34		PS_RDY received	Sends PS_RDY	Display 2 completes the Explicit Contract but at this point has not accepted that power can be drawn during suspend.	42
35		Source Capabilities received	Display2 sends out a new set of capabilities including: 5V@3A (15W), 9V@3A (27W), 15V@3A (45W) and 20V@3A (60W). The Unconstrained Power and USB suspend bits is now set to zero.	Based on the capabilities of the wall supply and its own needs Display 2 calculates what it can offer upstream. It decides that it can continue to supply the power even during USB suspend and so resets the USB suspend bit.	42
36		Display 1 requests 15V@2.8A (42W) from Display 2. The No Suspend Bit is set to reflect the request from the Tablet.	Request received	Display 1 repeats its request since a new set of Capabilities have been sent out.	42
37		Accept received	Sends Accept	Display 2 has 42W available, even during suspend, and so accepts the request.	42
38		PS_RDY received	Sends PS_RDY	Display 2 completes the Explicit Contract.	42
39	Capabilities received	Display 1 sends out a set of capabilities to the Tablet including: 5V@2.4A (12W). The Unconstrained Power bit is set, and USB suspend bit is cleared.		Display 1 now has the additional power available and so offers this to the Tablet.	42

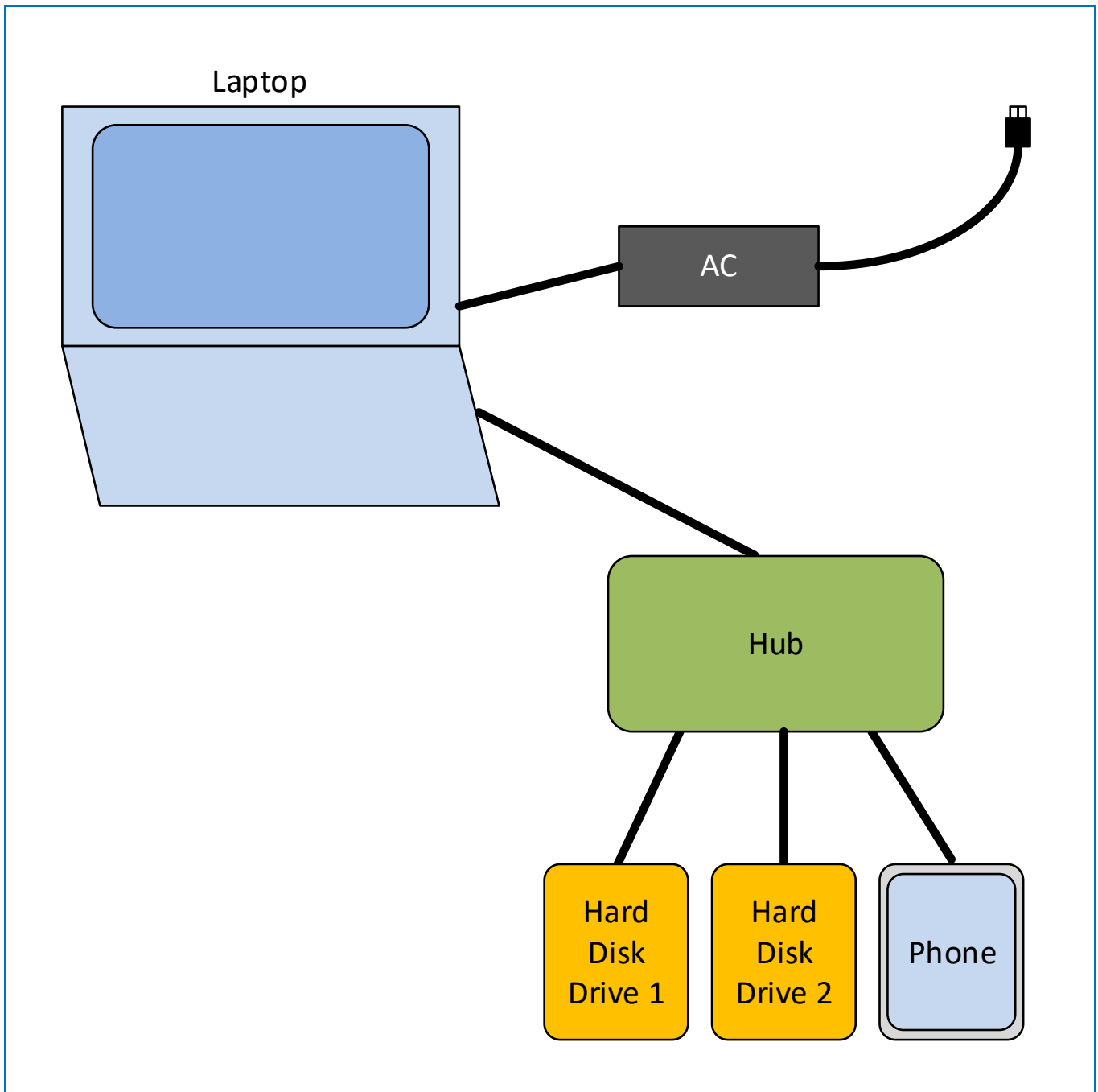
USB Power Delivery ENGINEERING CHANGE NOTICE

40	Tablet requests 5V@2.4A (12W) from Display 1.	Request received.		Tablet is being offered the power it needs to charge and so the Tablet requests this from Display 1.	42
41	Accept received	Sends Accept		Request is within the available Display 1's available power and so it accepts the request.	42
42	PS_RDY received. Tablet starts drawing 5V@2.4A (12W) Display 1 and starts to charge.	Sends PS_RDY		Display 1 indicates its supply is ready to supply power.	42

USB Power Delivery ENGINEERING CHANGE NOTICE

B.3 Giving back power

Figure B-3 “Giving Back Power”



USB Power Delivery ENGINEERING CHANGE NOTICE

Configuration:

1. Laptop with an AC supply. AC supply provides sufficient power to charge the laptop and, in addition, to provide up to 60W PD Power downstream.
2. A Hub with 4 downstream ports which initially provides 1-unit load (150mA) per Port plus 1-unit load for its internal functions.
3. Two Hard Disk Drives both of which require 5V@2A (10W) to spin up and 5V@1A (5W) while being accessed.
4. A phone which uses 5V@2A (10W) to charge and can give back all of this power when requested.

Table 10.3 Giving back power.

Step	Laptop	Hub	Peripherals	Device Policy Manager	Hub Power (W)
Connect Hub					
1	Connected to wall supply	Detached	Detached		Default
2	Hub is Attached	Attached, V_{BUS} powered			Default
3	Laptop sends out a set of capabilities including: 5V@3A (15W), 12V@3A (36W), and 20V@3A (60W). The Unconstrained Power and USB suspend bits are set.	Source Capabilities received		Laptop sends out details of all available power via external supply	Default
4	The Hub requests 5V@0.15A. This is the power for the Hubs internal operation.	Request received		Hub needs 1-unit load for its own operation and so requests this amount.	Default
5	Send Accept	Accept received		Laptop evaluates request and it is within its available power.	0.75
6	Send PS_RDY	PS_RDY received. Starts to draw 5V@0.15A		Laptop indicates that its power supply is ready.	0.75
Connect Hard Disk Drive 1					
7		Attached detected.	Hard Disk Drive 1 is Attached to one of the downstream ports of the Hub.		0.75
8	Request received	The Hub requests 5V@0.3A (1.5W) from the Laptop.		Hub needs 0.75W for its own operation plus 0.75W for USB communication on one Port.	0.75

USB Power Delivery ENGINEERING CHANGE NOTICE

9	Accept sent	Accept received		Request is within available power, so the laptop accepts.	1.5
10	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready	1.5
11		Hub turns on V_{BUS} and sends out a set of capabilities to Hard Disk Drive 1 including: 5V@0.15A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received		1.5
12		Request received	Hard Disk Drive 1 requests 5V@0.15A from the Hub.	Hard Disk Drive 1 only needs 1-unit load when not operating so requests this.	1.5
13		Accept sent	Accept received	Request is within available power, so the Hub accepts.	1.5
14		PS_RDY sent	PS_RDY received. The Hard Disk Drive starts drawing 1-unit load 5V@0.15A.	Laptop indicates its power supply is ready and the Hard Disk Drive starts drawing power.	1.5
Hard Disk Drive 1 spin up					
15		Request received	Hard Disk Drive 1 requests 5V@0.15A from the Hub but sets the Capability Mismatch bit.	Hard Disk Drive 1 needs 20V@0.5A to spin up but this is not available so it re-requests the available power flagging a capability mismatch.	1.5
16		Accept sent	Accept received	Request is within available power, so the Hub accepts.	1.5
17		PS_RDY sent	PS_RDY received	Hard Disk Drive 1 indicates a capability mismatch to the user.	1.5
18		The Hub requests the Sink Capabilities from Hard Disk Drive 1.	Get Sink Capabilities received	Due to the Capability Mismatch the Hub needs to determine what Hard Disk Drive 1 actually needs	1.5
19		Sink Capabilities received	Hard Disk Drive 1 returns capabilities indicating that it requires 5V@2A.		1.5

USB Power Delivery ENGINEERING CHANGE NOTICE

20	Request received	The Hub requests 5V@2.2A (11W) from the Laptop.		The Hub evaluates that it now needs 0.75W for the Hub and 10W for Hard Disk Drive 1.	1.5
21	Accept sent	Accept received		Power request from the Hub is within the Laptop's capabilities so the Laptop accepts the request.	11
22	PS_RDY sent	PS_RDY received		Laptop completes the Explicit Contract.	11
23		Hub sends out a set of capabilities to Hard Disk Drive 1 including: 5V@2A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Hub now offers Hard Disk Drive 1 what it needs.	11
24		Request received	Hard Disk Drive 1 requests 5V@2A operating current and indicates 5V@2A maximum current.	Hard Disk Drive 1 is operating at its maximum current to spin up so sets operating current = maximum current.	11
25		Accept sent	Accept received	Request is within the Hubs capabilities, so it accepts.	11
26		PS_RDY sent	PS_RDY received. Hard Disk Drive 1 starts to draw 5V@2A and spins up.	Hub indicates its power supply is ready, so Hard Disk Drive 1 starts to draw power.	11
27		Request received	Once spun up Hard Disk Drive 1 requests 5V@1A operating current and 5V@2A maximum current.	Hard Disk Drive 1 is operating at a lower current so sets operating current < maximum current.	11
28		Accept sent	Accept received	The Hub will maintain a Power Reserve of 5V@1A (5W) for Hard Disk Drive 1 in addition to the 5V@1A (5W) it is currently using.	11
29		PS_RDY sent	PS_RDY received	Hub completes the Explicit Contract.	11
Hard Disk Drive 2 spin up					

USB Power Delivery ENGINEERING CHANGE NOTICE

30		Attach detected	Hard Disk Drive 2 is Attached to one of the downstream ports of the Hub.		11
31	Request received	The Hub requests 5V@2.3A (11.5W) from the Laptop.		The Hub needs 0.75W for itself, 0.75W for USB communication on one Port, 5W for Hard Disk Drive 1 operation and 5W for the Power Reserve.	11
32	Accept sent	Accept received		Power request from the Hub is within the Laptop's capabilities so it accepts the request.	11
33	PS_RDY sent	PS_RDY received		Laptop indicates its power supply is ready.	11.5
34		Hub sends out a set of capabilities to Hard Disk Drive 2 including: 5V@0.15A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by Hard Disk Drive 2	Hub offers Hard Disk Drive 2 enough power to enumerate.	11.5
35		Request received	Hard Disk Drive 2 requests 5V@0.15A from the Hub.		11.5
36		Accept sent to Hard Disk Drive 2	Accept received by Hard Disk Drive 2	Request is within available capabilities, so the Hub accepts	11.5
37		PS_RDY sent to Hard Disk Drive 2.	PS_RDY received. Hard Disk Drive 2 starts drawing 5V@0.15A.	Hard Disk Drive 2 takes the power that it needs	11.5
Phone charge					
38		Attach detected	The phone is Attached to one of the downstream ports of the Hub.		11.5

USB Power Delivery ENGINEERING CHANGE NOTICE

39	Request received	The Hub Requests 5V@2.5A (12.5W) from the Laptop.		The Hub needs 0.75W for itself, 1.5W for USB communications on two ports (Hard Disk Drive 1 and the Phone), 5W for Hard Disk Drive 1 operation and 5W for the Power Reserve.	11.5
40	Accept sent	Accept received		Request is within available capabilities, so the Laptop accepts	12.5
41	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	12.5
42		The Hub powers V_{BUS} and sends out a set of capabilities to the Phone including: 5V@0.15A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by the Phone	The Hub offers the Phone 1-unit load to enumerate.	12.5
43		Request received from the Phone	The Phone requests 5V@0.15A from the Hub but sets the Capability Mismatch bit.	The Phone would like to charge and so indicates this fact through the Capability Mismatch bit.	12.5
44		Accept sent	Accept received	Request is within available capabilities, so the Hub accepts	12.5
45		PS_RDY sent	PS_RDY received	Hub indicates that its power supply is ready	12.5
46		The Hub requests the Sink Capabilities from the phone.	Get Sink Capabilities received by the Phone	Due to the Capability Mismatch the Hub needs to determine what the Phone actually needs	12.5
47		Sink Capabilities received from the Phone	The Phone returns capabilities indicating that it requires 5V@2A.	Phone returns the Capabilities it needs to charge	12.5

USB Power Delivery ENGINEERING CHANGE NOTICE

48	Request received	The Hub Requests 9V@2.4A (21.6W) from the Laptop.		The Hub needs 0.75W for itself, 0.75W for Hard Disk Drive 2, 10W for the phone, 5W for Hard Disk Drive 1 operation and 5W for the Power Reserve.	12.5
49	Accept sent	Accept received		Request is within available capabilities, so the Laptop accepts	12.5
50	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	21.6
51		The Hub sends out a set of capabilities to the Phone including: 5V@2A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by the Phone	The Hub now has the power that the Phone needs and so sends out a new set of Capabilities.	21.6
52		Request received from the Phone	The Phone requests 5V@2A from the Hub and sets the No USB Suspend bit since it needs to charge constantly. It sets the GiveBack flag and sets the Minimum Operating Current to 5V@0A.	The Phone requests the power it needs to charge. It asks for the USB Suspend requirement to be removed.	21.6
53		Accept sent to the Phone	Accept received by the Phone		21.6
54		PS_RDY sent to the phone.	PS_RDY received by the phone. Phone starts to charge 5V@2A but has to follow USB Suspend rules		21.6
55	Request received	The Hub Requests 9V@1.9A (17.1W) from the Laptop but sets the No USB Suspend bit.		The Hub needs 0.75W for itself, 0.75W for Hard Disk Drive 2, 10W for the phone (includes the Power Reserve of 5W), and 5W for Hard Disk Drive 1 operation. It requests for USB Suspend rule to be removed.	21.6

USB Power Delivery ENGINEERING CHANGE NOTICE

56	Accept sent	Accept received		Request is within available capabilities, so the Laptop accepts. Note that the request for No Suspend has not been acted on by the Laptop. USB Suspend rules apply until the Laptop sends out new Source Capabilities with the USB Suspend bit cleared.	21.6
57	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	17.1
Hard Disk Drive 2 spin up					
58		Request received from Hard Disk Drive 2	Hard Disk Drive 2 requests 5V@0.15A from the Hub but sets the Capability Mismatch bit.	Hard Disk Drive 2 needs more power to spin up and so indicates a Capability Mismatch	17.1
59		Accept sent	Accept received	The request is within its capabilities, so the Hub accepts.	17.1
60		PS_RDY sent	PS_RDY received	The Hub indicates that its power supply is ready.	17.1
61		The Hub requests the Sink Capabilities from Hard Disk Drive 2.	Get Sink Capabilities received by Hard Disk Drive 2	Due to the Capability Mismatch the Hub has to determine what Hard Disk Drive 2 needs	17.1
62		Sink Capabilities received	Hard Disk Drive 2 returns capabilities indicating that it requires 20V@0.5A maximum current.		17.1
63		The Hub instructs the Phone to Goto Minimum operation.	Goto Min received by the Phone	Hub assesses that there is additional power available from the Phone and so tells it to Goto Min. In this case it is reallocating the Phone's Charging power as the Power Reserve for the Hard Disk Drives.	17.1

USB Power Delivery ENGINEERING CHANGE NOTICE

64			The Phone drops to zero current draw.		17.1
65		PD_RDY sent	PS_RDY received.	Hub indicates that its power supply has changed to the new level.	17.1
66	Request received	The Hub Requests 9V@2.4A (21.6W) from the Laptop		The Hub has an additional 10W from the Phone but needs 5W more to maintain its Power Reserve. The Hub needs 0.75W for itself, 10W for Hard Disk Drive 2, 5W for the Power Reserve, 5W for Hard Disk Drive 1 operation.	17.1
67	Accept sent	Accept received		Request is within available capabilities, so the Laptop accepts.	17.1
68	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	21.6
69		Hub sends out a set of capabilities to Hard Disk Drive 2 including: 5V@0.5A and 20V@0.5A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by Hard Disk Drive 2	The Hub now has the power that Hard Disk Drive 2 needs, so it sends out new Capabilities.	21.6
70		Request received from Hard Disk Drive 2	Hard Disk Drive 2 requests 20V@0.5A operating current and 20V@0.5A.	Hard Disk Drive 2 requests what it needs to spin up.	21.6
71		Accept sent to Hard Disk Drive 2	Accept received by Hard Disk Drive 2	The Hub assesses that the request is within its Capabilities, so it accepts.	21.6
72		PS_RDY sent.	PS_RDY sent. Hard Disk Drive 2 starts to draw 20V@0.5A and spins up.		21.6
73		Request received from Hard Disk Drive 2	Once spun up Hard Disk Drive 2 requests 20V@0.25A operating current and 20V@0.5A maximum current.	Hard Disk Drive 2 no longer needs the additional power, so it gives back what it does not need.	21.6

USB Power Delivery ENGINEERING CHANGE NOTICE

74		Accept sent to Hard Disk Drive 2	Accept received by Hard Disk Drive 2	The Hub assesses that the request is within its Capabilities, so it accepts.	21.6
75		PS_RDY sent to Hard Disk Drive 2.	PS_RDY received by Hard Disk Drive 2.	The Hub indicates that its power supply is ready.	21.6
76		The Hub sends out a set of capabilities to the Phone including: 5V@2A. The Unconstrained Power bit is set, and the USB suspend bit is set.	Source Capabilities received by the Phone	The Hub now has the power available to charge the phone, so it sends out new Capabilities	21.6
77		Request received from the Phone	The Phone requests 5V@2A operating current from the Hub and sets the No USB Suspend bit since it needs to charge constantly. It sets the GiveBack flag and sets the Minimum Operating Current to 5V@0A.	The Phone requests the power it needs to charge. It asks for the USB Suspend requirement to be removed.	21.6
78		Accept sent to the Phone	Accept received by the Phone	The Hub assesses that the request is within its Capabilities, so it accepts but maintains USB Suspend rules.	21.6
79		PS_RDY sent to the Phone.	PS_RDY received by the Phone. The phone starts to draw 5V@2A but has to follow USB Suspend.	The Hub has allocated 0.75W for itself, 5W for Hard Disk Drive 2, 10W for the Phone (including 5W for the Power Reserve), and 5W for Hard Disk Drive 1 operation.	21.6

To: [Deprecate Appendix B]

USB Power Delivery ENGINEERING CHANGE NOTICE

B. PD Message Sequence Examples

This appendix has been **Deprecated**.

The following examples are intended to show how the Device Policy Manager might operate and the sequence of Power Delivery messaging which will result. The aim of this section is to inform implementer's how some of the mechanisms detailed in this specification might be applied; it does not contain any **Normative** requirements.

All ports are assumed to be Enhanced SuperSpeed capable, with a default operating Voltage of 5V and a unit load of 150mA. This 0.75W is assumed to be enough power to enable an externally powered device to maintain communication over USB and is enough to allow such a device to enumerate but not operate until more power is negotiated.

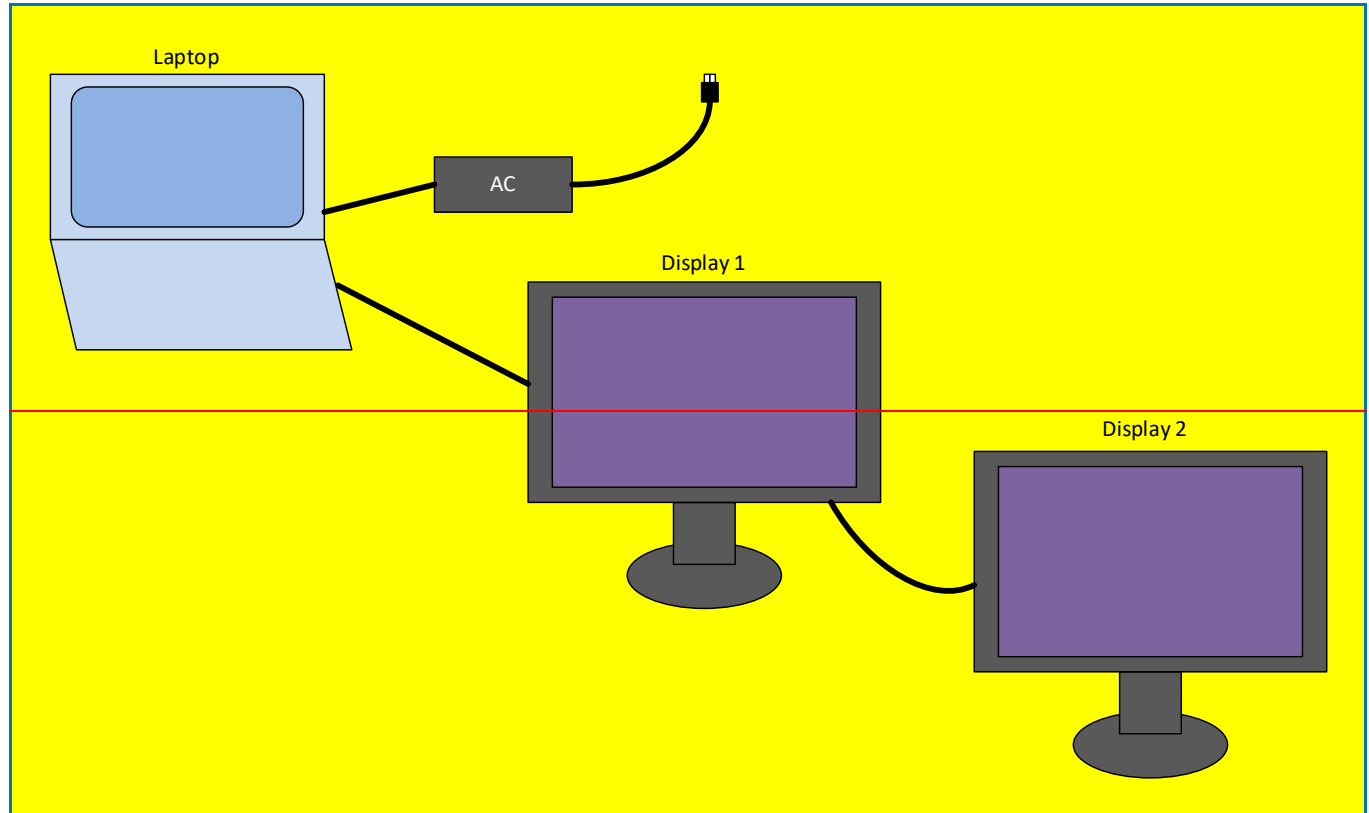
Although the Hubs in these illustrations support Power Delivery on both their UFPs and DFPs this is only one possible Hub implementation.

HDDs are assumed to spin-up immediately after they are Attached. This follows the typical operation of current systems.

Ideal power transmission is assumed so that there are no power losses through a device; in practice these would need to be taken into account when requesting power.

B.1 External power is supplied downstream

Figure B-1 "External Power supplied downstream"



USB Power Delivery ENGINEERING CHANGE NOTICE

Configuration:

5. Laptop with an AC supply. AC supply provides sufficient power to charge the laptop and, in addition, to provide up to 60W downstream via its Enhanced SuperSpeed Port. According to the Source Power Rules described in **Section 10.2 "Source Power Rules"** this means that the Port has a PD Power of 60W and so can supply: 5V@3A, 9V@3A, 15V@3A and 20V@3A.
6. Display 1 requires 30W to display and therefore a PD Power of 60W to operate itself plus Display 2 connected downstream. Display 1 initially uses 15V@2A to operate itself, since this also allows operation with a Source of 30W PD Power. On connection of Display 2, Display 1 will move to operation at 20V@3A to allow operation of the additional 30W ganged display. According to the Sink Power Rules described in **Section 10.3 "Sink Power Rules"** this means that Display 1 requires a Source with a PD Power of 60W to fully operate. Display 1 contains a Hub allowing Display 2 to be connected to Display 1.
7. Display 2 requires 30W operate itself and does not support an additional display connected downstream. Display 1 uses 15V@2A to operate itself from a Source of 30W PD Power.
8. In USB suspend Display 1 and Display 2 will power down but can maintain USB connection using the PD power provided.

Table B-1 External power is supplied downstream

Step	Laptop	Display 1	Display 2	Device Policy Manager	Power (W)
Display 1					
1	Connected to wall supply	Detached	Detached	!	0
2	Display 1 Attached, V _{BUS} powered.	Attached, drawing 5V@150mA.	Detached	!	0.75
3	Set of Source Capabilities sent including: 5V@3A (15W), 9V@3A (27W), 15V@3A(45W) and 20V@3A (60W). The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Detached	Laptop determines its Source Capabilities based on its needs and the presence of a wall supply.	0.75
4	Request received	Requests 15V@2A (30W) from laptop	Detached	Display 1 knows it needs 20v@1.5A (30W) for its own operation, evaluates the supplied capabilities and determines that this is available.	0.75
5	Sends Accept	Accept received	Detached	Waiting for PS_RDY before drawing additional power.	0.75
6	Sends PS_RDY	PS_RDY received. Starts drawing 15V@2A. Display 1 turns on and starts operating.	Detached	Laptop evaluates the request, finds that it can meet this and so sends an accept.	30

USB Power Delivery ENGINEERING CHANGE NOTICE

Display 2					
7	Powering Display 1	Detects Attach	Attached, no V_{BUS}		30
8	Request received	Display 1 requests 20V@1.73A (34.6W) from Laptop.	Attached, no V_{BUS}	Display 1 detects Attach and requests additional 4.5W of power for [USB 3.2] Port.	30
9	Sends Accept	Accept received.	Attached, no V_{BUS}		34.6
10	Sends PS_RDY	PS_RDY received	Attached, no V_{BUS}		
11		Powers V_{BUS}	Attached, drawing 5V@150mA.		34.6
12		Sends out Source Capabilities including: 5V@0.9A to Display 2. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Display 1 has 4.5W to allocate to Display 1. This is offered as a standard [USB 3.2] Port.	34.6
13		Request received	Display 2 requests 5V@0.15A but indicates a Capability Mismatch. Display 2 remains off.	Display 2 decides it can manage to run its USB/PD function with 1 unit load but needs more power to function as a display.	34.6
14		Sends Accept	Accept received		34.6
15		Sends PS_RDY	PS_RDY received	Display 2 indicates a capability mismatch to the user.	34.6
16		Get Sink Capabilities sent	Get Sink Capabilities received	Display 1 needs to assess the capability mismatch by first determining what Display 2 actually needs.	34.6
17		Sink Capabilities received	Display 2 returns Sink Capabilities indicating operation at 15V@2A.		34.6
18	Request received	Display 1 requests 20V@3A (60W) from Laptop.		Display1 now knows what Display 2 needs and requests the additional power from the laptop.	34.6
19	Sends Accept	Accept received.			34.6
20	Sends PS_RDY	PS_RDY received		An additional 30W is now available to Display 1 to offer to Display 2.	60

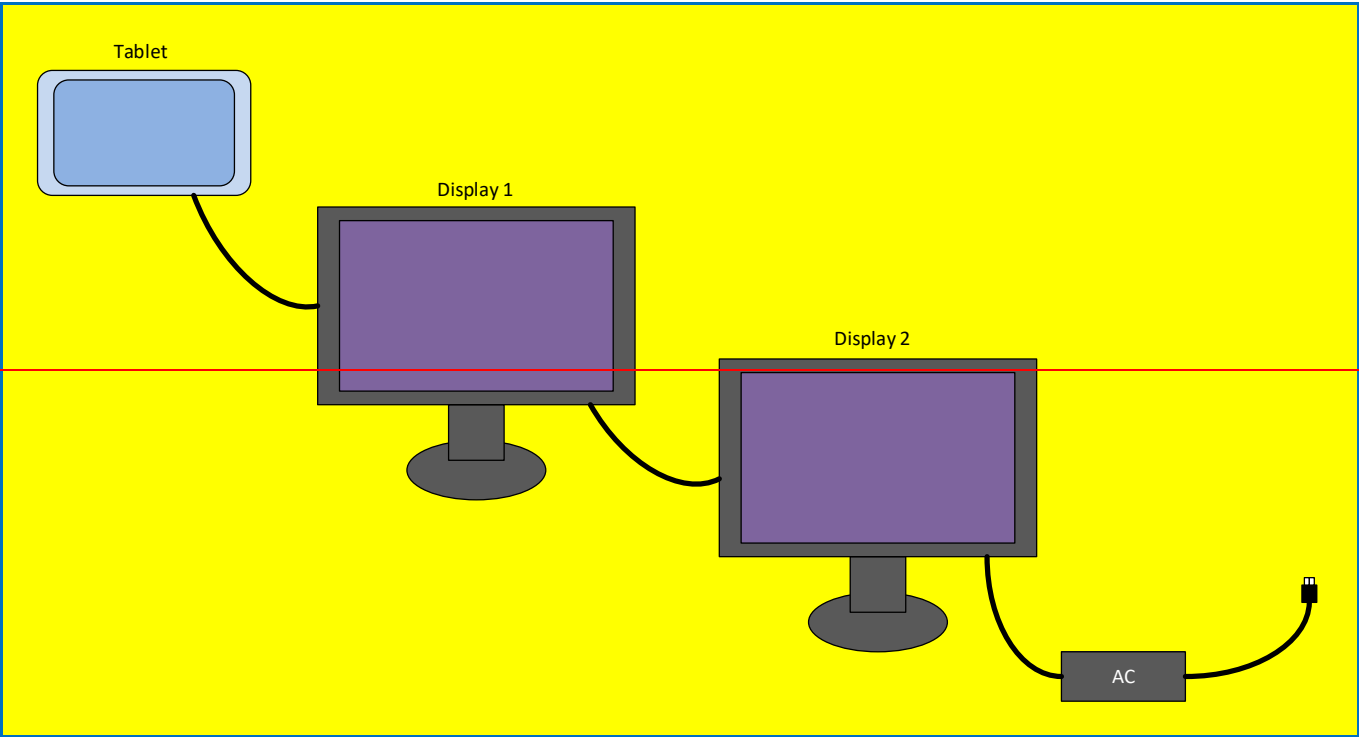
USB Power Delivery ENGINEERING CHANGE NOTICE

21	;	Sends out Source Capabilities including: 5V@0.9A and 20V@1.5A to Display 2. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Now that Display 1 can power Display 2 correctly this power is offered by Display 1 via a new capabilities Message.	60
22	;	Request received	Display 2 requests 15V@2A.	;	60
23	;	Sends Accept	Accept received	Display 1 determines that the request by Display 2 is within the offered capabilities, so the request is accepted.	60
24	;	Sends PS_RDY. Drawing 20V@3A from laptop.	PS_RDY received. Starts drawing 15V@2A, turns on and starts operating.	Display 2 now has the power it needs and can start working.	60
USB Suspend					
25	Laptop OS goes into suspend (S3), V_{BUS} remains on, but USB bus is also suspended.	Display 1 turns off but draws 50mW, 25mW to maintain PDUSB Hub functions. The additional 25mW is used to supply the Port used by Display 2.	Display 2 turns off but draws 25mW to maintain USB/PD functions.	No changes in Contract. This is a power reduction purely based on the USB state.	60
26	Laptop OS wakes up. USB is woken up.	Display 1 turns on and returns to drawing 20V@3A.	Display 2 turns on and returns to drawing 15V@2A.	No changes in PD Contract. This purely relates to USB bus state.	60

USB Power Delivery ENGINEERING CHANGE NOTICE

B.2 External power is supplied upstream

Figure B-2 External Power supplied upstream



Configuration:

- 4. Tablet with no AC supply. Tablet is a USB host and can use 5V@0.2A (1W) during normal operation and up to 5V@2.4A (12W) in order to charge.
- 5. Display 1 requires 30W to operate and therefore a PD Power of 42W to operate itself and charge the tablet. Display 1 uses 15V@2A to operate itself, since this allows operation with a Source of 30W PD Power and then moves to operation at 20V@2.1A to allow charging of the laptop. According to the Sink Power Rules described in **Section 10.3 "Sink Power Rules"** this means that the Display 1 requires a Source with a PD Power of 42W to fully operate.
- 6. Display 2 has an AC supply connected. AC supply provides sufficient power to power Display 2 and, in addition, to provide up to 60W PD Power upstream.

Table 10.2 External power is supplied upstream

Step	Tablet	Display 1	Display 2	Device Policy Manager	Power (W)
Display 1— Dead Battery					
1	Detached	Detached	Connected to the wall supply		0
2		Attached to Display 2	Display 1 Attached		0
3		USB Type-C® Power drawn 5V@1.5A	USB Type-C® Power Advertised 5V@1.5A		0

USB Power Delivery ENGINEERING CHANGE NOTICE

4		Attached to Display 2, drawing 5V@1.5A (7.5W)	Providing 1-unit load to Display 1.		7.5
5		Source Capabilities received	Display 2 sends out a set of capabilities including: 5V@3A (15W), 9V@3A (27W), 15V@3A (45W) and 20V@3A (60W). The Unconstrained Power and USB suspend bits are set.	Based on the capabilities of the wall supply and its own needs Display 2 calculates what it can offer upstream.	7.5
6		Display 1 requests 15V@2A (30W) from Display 2.	Request received	Display 1 knows it needs 30W to operate so it requests this amount.	7.5
7		Accept received	Sends Accept	Display 2 accepts the offer since it is within its capabilities.	7.5
8		PS_RDY received. Display 1 starts drawing power and turns on.	Sends PS_RDY	Display 2 indicates its power supply is ready to offer the power.	30
Tablet – Power Role Swap					
9	Tablet is Attached to Display 1.	Attached, V _{bus} powered.			30
10	Tablet sends out a set of capabilities including: 5V@0.5A (2.5W). The Unconstrained Power bit cleared, and USB suspend bit set.	Capabilities received			30
11	Request received	Display 1 requests 5V@0A from the Tablet. The Unconstrained Power and Dual Role Power bits are set.		Display 1 has external power providing everything it needs so it does not request any more.	30
12	Sends Accept	Accept received.		No power has been requested from the Tablet, so the tablet has no reason to Reject this.	30
13	Sends PS_RDY	PS_RDY received.		Table completes the Explicit Contract by sending PS_RDY.	30

USB Power Delivery ENGINEERING CHANGE NOTICE

14	Get Sink Capabilities received.	Sends Get Sink Capabilities		Display 1 has access to an external supply so it needs to check whether the Tablet upstream, which has no external supply, could use some power. Display 1 also knows that there is excess capacity, based on the last capabilities it received, which it is not currently using from Display 2.	30
15	The Tablet returns Sink Capabilities indicating that it is a Dual-Role and that it can use 5V@0.2A (1W) as a Sink	Sink Capabilities received			30
16		Display 1 requests 15V@2.1A (31.5W) from Display 2.	Request received		30
17		Accept received	Sends Accept	Request is within the available power so Display 2 sends an accept.	30
18		PS_RDY received	Sends PS_RDY	Display 2 indicates that the power supply is ready to supply the power.	31.5
19	PR_Swap received	Requests PR_Swap from Tablet.		Display 1 now offers to provide power to the Tablet by initiating a Power Role Swap.	31.5
20	Accept sent. Tablet turns off its V _{BUS} supply.	Accept received.		Tablet is happy to accept a Power Role Swap from any device offering it power.	31.5
21	Send PS_RDY	PS_RDY received. Display 1 turns on its V _{BUS} supply.		Tablet indicates that its supply has been turned off.	31.5
22	PS_RDY received.	PS_RDY sent.		Display 1 indicates that its power supply is ready, so the Tablet starts drawing power.	31.5
23	Source Capabilities received	Display 1 sends out a set of capabilities to the Tablet including: 5V@0.48A (2.4W), 12V@0.2A (2.4W) and 20V@0.12 (2.4W). The Unconstrained Power and USB suspend bits are set.			31.5

USB Power Delivery ENGINEERING CHANGE NOTICE

24	The Tablet requests 12V@0.2A.	Request received.		Tablet can now request the power it needs.	31.5
25	Accept received	Accept sent		Power is within the capabilities of Display 1, so it accepts the request.	31.5
26	PS_RDY received. The Tablet starts drawing 12V@0.2A.	PS_RDY sent		Display 1 indicates that its power supply is ready, so the tablet starts drawing the power.	31.5
Tablet – Charge					
27	Tablet requests 12V@0.2A (2.4W) from Display 1. The Tablet needs to charge and so sets the Capability Mismatch bit and the No USB Suspend bit.	Request received.		Tablet needs to charge but the power offered is not sufficient. Since Display 1 claims to have an external supply, the Tablet will try to get more power using the Capability Mismatch Flag.	31.5
28	Accept received	Accept sent		A <i>Valid</i> request has been made so Display 1 accepts the request.	31.5
29	PS_RDY received	PS_RDY received		Tablet indicates a capability mismatch to the user.	31.5
30	Get Sink Capabilities received.	Get Sink Capabilities sent		Due to the Capability Mismatch Flag Display 1 requests Sink Capabilities from the Tablet?	31.5
31	The Tablet returns Sink capabilities containing: 5V@2.4A (12W). The Unconstrained Power bit is cleared.	Sink Capabilities received			31.5
32		Display 1 requests 15V@2.8A (42W) from Display 2. The No Suspend Bit is set to reflect the request from the Tablet.	Request received	Since the Tablet requires an additional 12W of power and Display 1 knows that this is available from Display 2 based on the last Capabilities received so it requests it. In addition, the Request from the Tablet indicated that it wanted No Suspend so this is reflected upwards.	31.5

USB Power Delivery ENGINEERING CHANGE NOTICE

33		Accept received	Sends Accept	Display 2 has 42W available and so accepts the request.	42
34		PS_RDY received	Sends PS_RDY	Display 2 completes the Explicit Contract but at this point has not accepted that power can be drawn during suspend.	42
35		Source Capabilities received	Display2 sends out a new set of capabilities including: 5V@3A (15W), 9V@3A (27W), 15V@3A (45W) and 20V@3A (60W). The Unconstrained Power and USB suspend bits is now set to zero.	Based on the capabilities of the wall supply and its own needs Display 2 calculates what it can offer upstream. It decides that it can continue to supply the power even during USB suspend and so resets the USB suspend bit.	42
36		Display 1 requests 15V@2.8A (42W) from Display 2. The No Suspend Bit is set to reflect the request from the Tablet.	Request received	Display 1 repeats its request since a new set of Capabilities have been sent out.	42
37		Accept received	Sends Accept	Display 2 has 42W available, even during suspend, and so accepts the request.	42
38		PS_RDY received	Sends PS_RDY	Display 2 completes the Explicit Contract.	42
39	Capabilities received	Display 1 sends out a set of capabilities to the Tablet including: 5V@2.4A (12W). The Unconstrained Power bit is set, and USB suspend bit is cleared.		Display 1 now has the additional power available and so offers this to the Tablet.	42

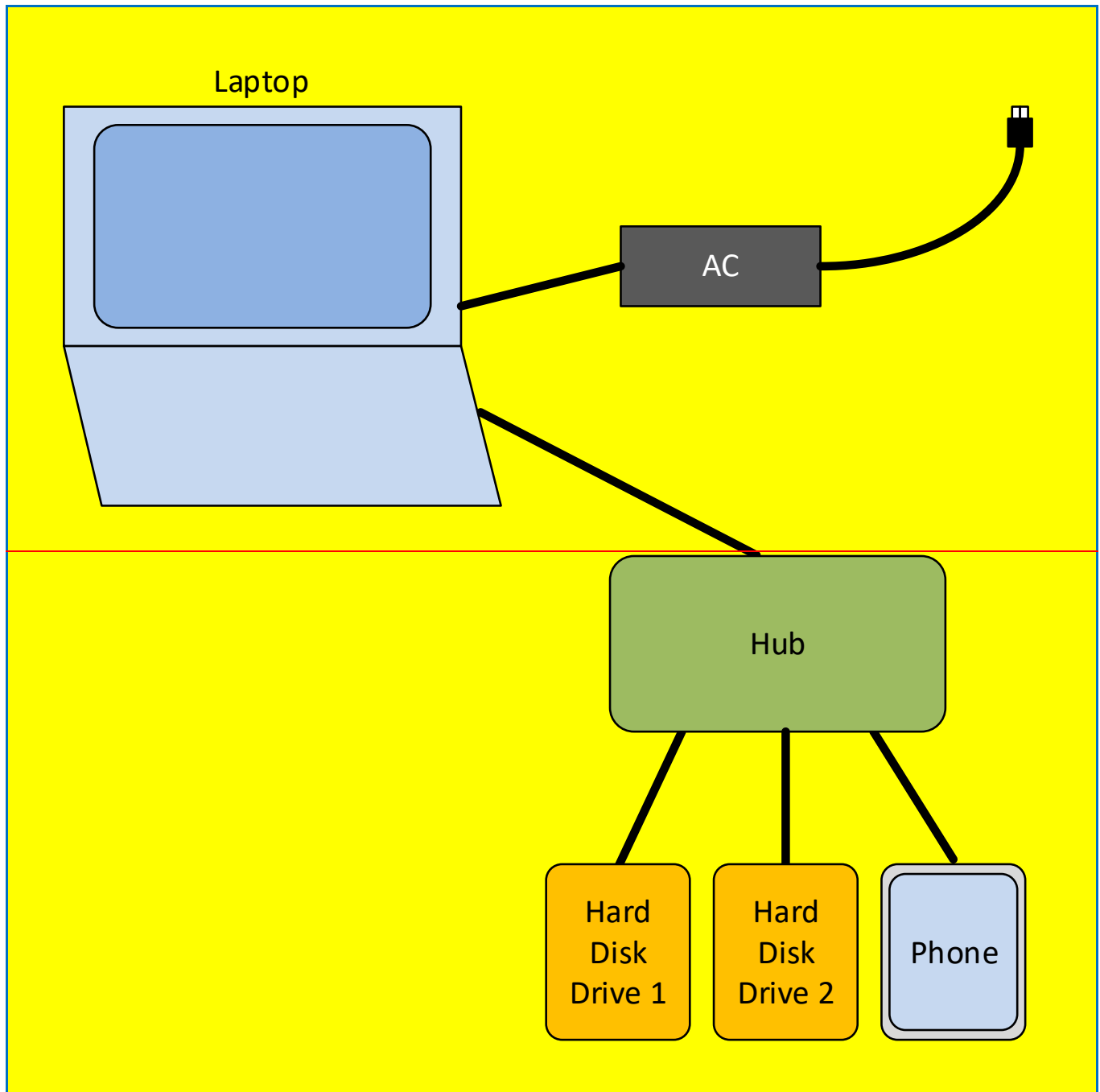
USB Power Delivery ENGINEERING CHANGE NOTICE

40	Tablet requests 5V@2.4A (12W) from Display 1.	Request received.		Tablet is being offered the power it needs to charge and so the Tablet requests this from Display 1.	42
41	Accept received	Sends Accept		Request is within the available Display 1's available power and so it accepts the request.	42
42	PS_RDY received. Tablet starts drawing 5V@2.4A (12W) Display 1 and starts to charge.	Sends PS_RDY		Display 1 indicates its supply is ready to supply power.	42

USB Power Delivery ENGINEERING CHANGE NOTICE

B.3 ~~Giving back power~~

Figure B-3 "Giving Back Power"



USB Power Delivery ENGINEERING CHANGE NOTICE

Configuration:

5. Laptop with an AC supply. AC supply provides sufficient power to charge the laptop and, in addition, to provide up to 60W PD Power downstream.
6. A Hub with 4 downstream ports which initially provides 1-unit load (150mA) per Port plus 1-unit load for its internal functions.
7. Two Hard Disk Drives both of which require 5V@2A (10W) to spin up and 5V@1A (5W) while being accessed.
8. A phone which uses 5V@2A (10W) to charge and can give back all of this power when requested.

Table 10.3 Giving back power.

Step	Laptop	Hub	Peripherals	Device Policy Manager	Hub Power (W)
Connect Hub					
1	Connected to wall supply	Detached	Detached		Default
2	Hub is Attached	Attached, V _{BUS} powered			Default
3	Laptop sends out a set of capabilities including: 5V@3A (15W), 12V@3A (36W), and 20V@3A (60W). The Unconstrained Power and USB suspend bits are set.	Source Capabilities received		Laptop sends out details of all available power via external supply	Default
4	The Hub requests 5V@0.15A. This is the power for the Hubs internal operation.	Request received		Hub needs 1-unit load for its own operation and so requests this amount.	Default
5	Send Accept	Accept received		Laptop evaluates request and it is within its available power.	0.75
6	Send PS_RDY	PS_RDY received. Starts to draw 5V@0.15A		Laptop indicates that its power supply is ready.	0.75
Connect Hard Disk Drive 1					
7		Attached detected.	Hard Disk Drive 1 is Attached to one of the downstream ports of the Hub.		0.75
8	Request received	The Hub requests 5V@0.3A (1.5W) from the Laptop.		Hub needs 0.75W for its own operation plus 0.75W for USB communication on one Port.	0.75

USB Power Delivery ENGINEERING CHANGE NOTICE

9	Accept sent	Accept received		Request is within available power, so the laptop accepts.	1.5
10	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready	1.5
11		Hub turns on V_{BUS} and sends out a set of capabilities to Hard Disk Drive 1 including: 5V@0.15A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received		1.5
12		Request received	Hard Disk Drive 1 requests 5V@0.15A from the Hub.	Hard Disk Drive 1 only needs 1-unit load when not operating so requests this.	1.5
13		Accept sent	Accept received	Request is within available power, so the Hub accepts.	1.5
14		PS_RDY sent	PS_RDY received. The Hard Disk Drive starts drawing 1-unit load 5V@0.15A.	Laptop indicates its power supply is ready and the Hard Disk Drive starts drawing power.	1.5
Hard Disk Drive 1 spin-up					
15		Request received	Hard Disk Drive 1 requests 5V@0.15A from the Hub but sets the Capability Mismatch bit.	Hard Disk Drive 1 needs 20V@0.5A to spin-up but this is not available so it re-requests the available power flagging a capability mismatch.	1.5
16		Accept sent	Accept received	Request is within available power, so the Hub accepts.	1.5
17		PS_RDY sent	PS_RDY received	Hard Disk Drive 1 indicates a capability mismatch to the user.	1.5
18		The Hub requests the Sink Capabilities from Hard Disk Drive 1.	Get Sink Capabilities received	Due to the Capability Mismatch the Hub needs to determine what Hard Disk Drive 1 actually needs	1.5
19		Sink Capabilities received	Hard Disk Drive 1 returns capabilities indicating that it requires 5V@2A.	!	1.5

USB Power Delivery ENGINEERING CHANGE NOTICE

20	Request received	The Hub requests 5V@2.2A (11W) from the Laptop.	-	The Hub evaluates that it now needs 0.75W for the Hub and 10W for Hard Disk Drive 1.	1.5
21	Accept sent	Accept received	-	Power request from the Hub is within the Laptop's capabilities so the Laptop accepts the request.	11
22	PS_RDY sent	PS_RDY received		Laptop completes the Explicit Contract.	11
23		Hub sends out a set of capabilities to Hard Disk Drive 1 including: 5V@2A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received	Hub now offers Hard Disk Drive 1 what it needs.	11
24		Request received	Hard Disk Drive 1 requests 5V@2A operating current and indicates 5V@2A maximum current.	Hard Disk Drive 1 is operating at its maximum current to spin up so sets operating current = maximum current.	11
25		Accept sent	Accept received	Request is within the Hubs capabilities, so it accepts.	11
26		PS_RDY sent	PS_RDY received. Hard Disk Drive 1 starts to draw 5V@2A and spins up.	Hub indicates its power supply is ready, so Hard Disk Drive 1 starts to draw power.	11
27	-	Request received	Once spun up Hard Disk Drive 1 requests 5V@1A operating current and 5V@2A maximum current.	Hard Disk Drive 1 is operating at a lower current so sets operating current < maximum current.	11
28	-	Accept sent	Accept received	The Hub will maintain a Power Reserve of 5V@1A (5W) for Hard Disk Drive 1 in addition to the 5V@1A (5W) it is currently using.	11
29		PS_RDY sent	PS_RDY received	Hub completes the Explicit Contract.	11
Hard Disk Drive 2 spin up					

USB Power Delivery ENGINEERING CHANGE NOTICE

30		Attach detected	Hard Disk Drive 2 is Attached to one of the downstream ports of the Hub.		11
31	Request received	The Hub requests 5V@2.3A (11.5W) from the Laptop.		The Hub needs 0.75W for itself, 0.75W for USB communication on one Port, 5W for Hard Disk Drive 1 operation and 5W for the Power Reserve.	11
32	Accept sent	Accept received		Power request from the Hub is within the Laptop's capabilities so it accepts the request.	11
33	PS_RDY sent	PS_RDY received		Laptop indicates its power supply is ready.	11.5
34	-	Hub sends out a set of capabilities to Hard Disk Drive 2 including: 5V@0.15A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by Hard Disk Drive 2	Hub offers Hard Disk Drive 2 enough power to enumerate.	11.5
35	-	Request received	Hard Disk Drive 2 requests 5V@0.15A from the Hub.		11.5
36	-	Accept sent to Hard Disk Drive 2	Accept received by Hard Disk Drive 2	Request is within available capabilities, so the Hub accepts	11.5
37	-	PS_RDY sent to Hard Disk Drive 2.	PS_RDY received. Hard Disk Drive 2 starts drawing 5V@0.15A.	Hard Disk Drive 2 takes the power that it needs	11.5
Phone charge					
38	-	Attach detected	The phone is Attached to one of the downstream ports of the Hub.	-	11.5

USB Power Delivery ENGINEERING CHANGE NOTICE

39	Request received	The Hub Requests 5V@2.5A (12.5W) from the Laptop.	-	The Hub needs 0.75W for itself, 1.5W for USB communications on two ports (Hard Disk Drive 1 and the Phone), 5W for Hard Disk Drive 1 operation and 5W for the Power Reserve.	11.5
40	Accept sent	Accept received	-	Request is within available capabilities, so the Laptop accepts	12.5
41	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	12.5
42	-	The Hub powers V_{bus} and sends out a set of capabilities to the Phone including 5V@0.15A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by the Phone	The Hub offers the Phone 1-unit load to enumerate.	12.5
43	-	Request received from the Phone	The Phone requests 5V@0.15A from the Hub but sets the Capability Mismatch bit.	The Phone would like to charge and so indicates this fact through the Capability Mismatch bit.	12.5
44		Accept sent	Accept received	Request is within available capabilities, so the Hub accepts	12.5
45		PS_RDY sent	PS_RDY received	Hub indicates that its power supply is ready	12.5
46		The Hub requests the Sink Capabilities from the phone.	Get Sink Capabilities received by the Phone	Due to the Capability Mismatch the Hub needs to determine what the Phone actually needs	12.5
47		Sink Capabilities received from the Phone	The Phone returns capabilities indicating that it requires 5V@2A.	Phone returns the Capabilities it needs to charge	12.5

USB Power Delivery ENGINEERING CHANGE NOTICE

48	Request received	The Hub Requests 9V@2.4A (21.6W) from the Laptop.		The Hub needs 0.75W for itself, 0.75W for Hard Disk Drive 2, 10W for the phone, 5W for Hard Disk Drive 1 operation and 5W for the Power Reserve.	12.5
49	Accept sent	Accept received		Request is within available capabilities, so the Laptop accepts	12.5
50	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	21.6
51		The Hub sends out a set of capabilities to the Phone including: 5V@2A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by the Phone	The Hub now has the power that the Phone needs and so sends out a new set of Capabilities.	21.6
52		Request received from the Phone	The Phone requests 5V@2A from the Hub and sets the No USB Suspend bit since it needs to charge constantly. It sets the GiveBack flag and sets the Minimum Operating Current to 5V@0A.	The Phone requests the power it needs to charge. It asks for the USB Suspend requirement to be removed.	21.6
53		Accept sent to the Phone	Accept received by the Phone		21.6
54		PS_RDY sent to the phone.	PS_RDY received by the phone. Phone starts to charge 5V@2A but has to follow USB Suspend rules		21.6
55	Request received	The Hub Requests 9V@1.9A (17.1W) from the Laptop but sets the No USB Suspend bit.	!	The Hub needs 0.75W for itself, 0.75W for Hard Disk Drive 2, 10W for the phone (includes the Power Reserve of 5W), and 5W for Hard Disk Drive 1 operation. It requests for USB Suspend rule to be removed.	21.6

USB Power Delivery ENGINEERING CHANGE NOTICE

56	Accept sent	Accept received	-	Request is within available capabilities, so the Laptop accepts. Note that the request for No Suspend has not been acted on by the Laptop. USB Suspend rules apply until the Laptop sends out new Source Capabilities with the USB Suspend bit cleared.	21.6
57	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	17.1
Hard-Disk Drive 2 spin up					
58	-	Request received from Hard Disk Drive 2	Hard Disk Drive 2 requests 5V@0.15A from the Hub but sets the Capability Mismatch bit.	Hard Disk Drive 2 needs more power to spin up and so indicates a Capability Mismatch	17.1
59		Accept sent	Accept received	The request is within its capabilities, so the Hub accepts.	17.1
60		PS_RDY sent	PS_RDY received	The Hub indicates that its power supply is ready.	17.1
61		The Hub requests the Sink Capabilities from Hard Disk Drive 2.	Get Sink Capabilities received by Hard Disk Drive 2	Due to the Capability Mismatch the Hub has to determine what Hard Disk Drive 2 needs	17.1
62		Sink Capabilities received	Hard Disk Drive 2 returns capabilities indicating that it requires 20V@0.5A maximum current.		17.1
63		The Hub instructs the Phone to Goto Minimum operation.	Goto Min received by the Phone	Hub assesses that there is additional power available from the Phone and so tells it to Goto Min. In this case it is reallocating the Phone's Charging power as the Power Reserve for the Hard Disk Drives.	17.1

USB Power Delivery ENGINEERING CHANGE NOTICE

64			The Phone drops to zero current draw.		17.1
65		PD_RDY sent	PS_RDY received.	Hub indicates that its power supply has changed to the new level.	17.1
66	Request received	The Hub Requests 9V@2.4A (21.6W) from the Laptop		The Hub has an additional 10W from the Phone but needs 5W more to maintain its Power Reserve. The Hub needs 0.75W for itself, 10W for Hard Disk Drive 2, 5W for the Power Reserve, 5W for Hard Disk Drive 1 operation.	17.1
67	Accept sent	Accept received		Request is within available capabilities, so the Laptop accepts.	17.1
68	PS_RDY sent	PS_RDY received		Laptop indicates that its power supply is ready.	21.6
69		Hub sends out a set of capabilities to Hard Disk Drive 2 including: 5V@0.5A and 20V@0.5A. The Unconstrained Power and USB suspend bits are set.	Source Capabilities received by Hard Disk Drive 2	The Hub now has the power that Hard Disk Drive 2 needs, so it sends out new Capabilities.	21.6
70		Request received from Hard Disk Drive 2	Hard Disk Drive 2 requests 20V@0.5A operating current and 20V@0.5A.	Hard Disk Drive 2 requests what it needs to spin-up.	21.6
71		Accept sent to Hard Disk Drive 2	Accept received by Hard Disk Drive 2	The Hub assesses that the request is within its Capabilities, so it accepts.	21.6
72		PS_RDY sent	PS_RDY sent. Hard Disk Drive 2 starts to draw 20V@0.5A and spins up.		21.6
73		Request received from Hard Disk Drive 2	Once spun up Hard Disk Drive 2 requests 20V@0.25A operating current and 20V@0.5A maximum current.	Hard Disk Drive 2 no longer needs the additional power, so it gives back what it does not need.	21.6

USB Power Delivery ENGINEERING CHANGE NOTICE

74		Accept sent to Hard Disk Drive 2	Accept received by Hard Disk Drive 2	The Hub assesses that the request is within its Capabilities, so it accepts.	21.6
75		PS_RDY sent to Hard Disk Drive 2.	PS_RDY received by Hard Disk Drive 2.	The Hub indicates that its power supply is ready.	21.6
76		The Hub sends out a set of capabilities to the Phone including: 5V@2A. The Unconstrained Power bit is set, and the USB suspend bit is set.	Source Capabilities received by the Phone	The Hub now has the power available to charge the phone, so it sends out new Capabilities	21.6
77		Request received from the Phone	The Phone requests 5V@2A operating current from the Hub and sets the No USB Suspend bit since it needs to charge constantly. It sets the GiveBack flag and sets the Minimum Operating Current to 5V@0A.	The Phone requests the power it needs to charge. It asks for the USB Suspend requirement to be removed.	21.6
78		Accept sent to the Phone	Accept received by the Phone	The Hub assesses that the request is within its Capabilities, so it accepts but maintains USB Suspend rules.	21.6
79		PS_RDY sent to the Phone.	PS_RDY received by the Phone. The phone starts to draw 5V@2A but has to follow USB Suspend.	The Hub has allocated 0.75W for itself, 5W for Hard Disk Drive 2, 10W for the Phone (including 5W for the Power Reserve), and 5W for Hard Disk Drive 1 operation.	21.6

USB Power Delivery ENGINEERING CHANGE REQUEST FORM

: