Wi-Fi Serial Bus Specification

Version 1.0

Document revision history

Version	Date YYYY-MM-DD	Remarks
1.0	2018-11-29	Initial release.

Table of contents

List of tables

1 Introduction

This document is the specification for Wi-Fi Serial Bus, a solution for enabling existing USB applications to connect with Media Agnostic USB (MA USB) [3] hubs and devices over Wi-Fi. This specification describes the required operations, attributes and architecture for Wi-Fi Serial Bus 1.0.

Wi-Fi Serial Bus references the existing USB specification suite, specifically, it requires MA USB protocol implementation by devices implementing Wi-Fi Serial Bus, and specifies functionalities required for operation of MA USB over Wi-Fi connections. Wi-Fi Serial Bus defines the following architectural elements:

- Wi-Fi Serial Bus Host: a Wi-Fi Serial Bus Host device implementing MA USB host functionality.
- Wi-Fi Serial Bus Hub: a Wi-Fi Serial Bus Hub device implementing MA USB hub functionality.
- Wi-Fi Serial Bus Peripheral: a Wi-Fi Serial Bus Peripheral device implementing MA USB device functionality

In this specification, the term Device is used to denote a device that is either a Wi-Fi Serial Bus Hub or a Wi-Fi Serial Bus Peripheral. The term Host is used to denote a Wi-Fi Serial Bus Host, the term Hub is used to denote a Wi-Fi Serial Bus Hub, and the term Peripheral is used to denote a Wi-Fi Serial Bus Peripheral.

This specification defines the architecture and a set of protocols to perform Wi-Fi Serial Bus device/service discovery and connection setup between a Host and a Device, Wi-Fi specific configurations on a Host/Device, and the method of utilizing the MA USB protocol [3] for USB data transfers.

The set of protocols defined in the specification is referred to as the Wi-Fi Serial Bus Service, which is deployed as a service over the Wi-Fi Direct® Services (WFDS) Application Service Platform (ASP) [2]. ASP is a layer on top of Wi-Fi Direct for coordinating the discovery of services and managing the connections and sessions between two Peer-to-Peer (P2P) enabled devices. It also provides an abstract API defined in terms of primitives, which is used in this specification to denote the Wi-Fi Serial Bus Service specific requirements for discovery, connection setup and ASP session initiation.

This specification also contains an informative annex with several end-to-end operation scenarios that serve as an example of the correct usage of the combined operation of Wi-Fi Serial Bus and the MA USB protocol.

1.1 References

Knowledge of the documents listed in this section is required for understanding this technical specification. If a reference includes a date or a version identifier, only that specific version of the document is required. If the listing includes neither a date nor a version identifier, then the latest version of the document is required. In the event of a conflict between this specification and the following referenced documents, the contents of this specification take precedence.

[1] Wi-Fi Peer-to-Peer Technical Specification v1.7, https://www.wi-fi.org/members/certification-programs

[2] Wi-Fi Peer-to-Peer Services Technical Specification v1.2, https://www.wi-fi.org/members/certification-programs

[3] USB-IF Media Agnostic Universal Serial Bus Specification Release 1.0a, http://www.usb.org/developers/docs/devclass_docs/Media_Agnostic_USB_v1.0a.zip

[4] Wi-Fi Simple Configuration Technical Specification v2.0.6, https://www.wi-fi.org/file/wi-fi-simple-configuration-technical-specification-v206

[5] USB-IF Defined 1.0 Class Codes, http://www.usb.org/developers/defined_class

[6] IEEE Standard for Information Technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2016.

[7] Universal Serial Bus Specification, Rev. 2.0, April 2000, http://www.usb.org/developers/docs/usb20_docs/usb_20_042814.zip

[8] Universal Serial Bus 3.1 Specification, Rev.1.0, July 26, 2013, http://www.usb.org/developers/docs/usb_31_031114.zip

1.2 Definitions and acronyms

1.2.1 Shall/should/may/might word usage

The words shall, should, and may are used intentionally throughout this document to identify the requirements for the Wi-Fi Serial Bus program. The words can and might shall not be used to define requirements.

The word *shall* indicates a mandatory requirement. All mandatory requirements must be implemented to assure interoperability with other Wi-Fi Serial Bus products.

The word *should* denotes a recommended approach or action.

The word *may* indicates a permitted approach or action with no implied preference.

The words *might* and *can* indicate a possibility or suggestion and should be used sparingly.

1.2.2 Conventions

The ordering of bits and bytes in the fields within information elements, attributes and action frames shall follow the conventions in section 8.2.2 of IEEE Standard 802.11-2016 [6] unless otherwise stated.

The word *ignored* shall be used to describe bits, bytes, fields or parameters whose values are not verified by the recipient.

The word *reserved* shall be used to describe objects (bits, bytes, or fields or their assigned values) whose usage and interpretation will be defined in the future by this specification or by other technical specifications/bulletins. A reserved object shall be set to zero unless otherwise stated. The recipient of a reserved object shall ignore its value unless that object becomes defined at a later date. The sender of an object defined by this technical specification shall not use a reserved code value.

1.2.3 Abbreviations and acronyms

Table 1 defines the acronyms used throughout this document. Some acronyms are commonly used in publications and standards defining the operation of wireless local area networks, while others have been generated by Wi-Fi Alliance.

Table 1. Abbreviations and acronyms

Acronyms	Definition
AP	Access Point
ASP	Application Service Platform
GAS	Generic Advertisement Service
GO	Group Owner
IAD	Interface Association Descriptor
MA	Media Agnostic
PAL	Protocol Adaptation Layer
SNAP	Subnetwork Access Protocol
STA	Station
WFDS	Wi-Fi Direct Services
WMM®-PS	Wi-Fi Multimedia Power Save
WSC	Wi-Fi Simple Configuration

1.2.4 Definitions

Table 2 lists the definitions that are used in this specification.

Table 2. Definitions

Term	Definition
Application	A software process with a user interface that a user may employ to accomplish a task.
MA USB protocol	A protocol specified by [3] that encapsulates USB transfers over an underlying transport.
Wi-Fi Serial Bus Device or Device	A device that has the functionality of a Wi-Fi Serial Bus Hub or a Wi-Fi Serial Bus Peripheral.
Wi-Fi Serial Bus Host or Host	Wi-Fi Serial Bus Host device which performs MA USB host functions and enables a Wi-Fi connection to Wi-Fi Serial Bus Peripherals/Wi-Fi Serial Bus Hubs.
Wi-Fi Serial Bus Hub or Hub	Wi-Fi Serial Bus Hub device which performs MA USB hub functions and enables a Wi-Fi connection to a Wi-Fi Serial Bus Host. Wireless downstream connections of Wi-Fi Serial Bus Hubs are beyond scope of Wi-Fi Serial Bus.
Wi-Fi Serial Bus Peripheral or Peripheral	Wi-Fi Serial Bus Peripheral device which performs MA USB device functions and enables a Wi-Fi connection to a Wi-Fi Serial Bus Host.

2 Architectural overview

The architecture of a device that implements Wi-Fi Serial Bus is illustrated in Figure 1.

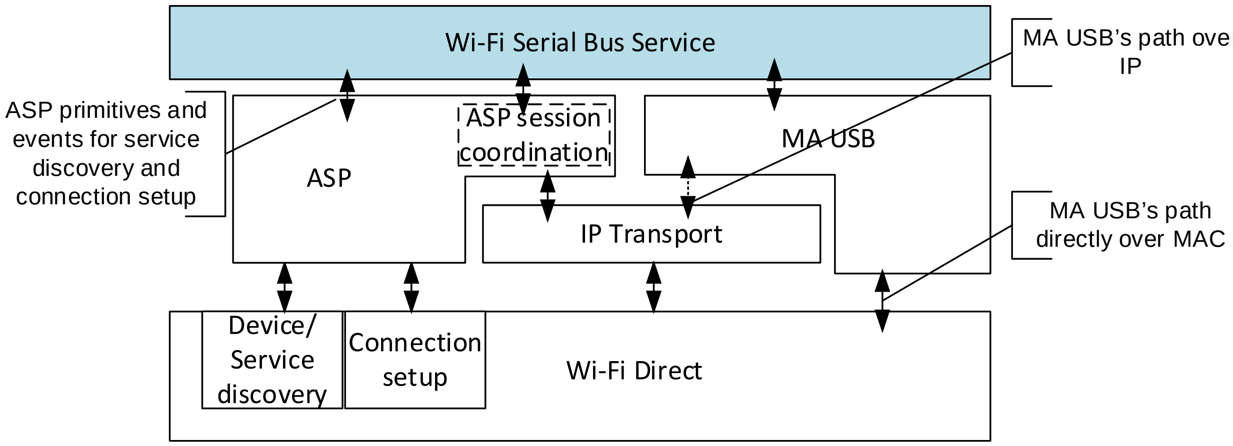

Figure 1. Wi-Fi Serial Bus architecture

The Wi-Fi Serial Bus Service performs service discovery and connection setup by utilizing primitives and events provided by the underlying WFDS ASP. After the connection setup, the ASP Coordination protocol as specified in [2] is used for session coordination to exchange information about:

1. The underlying transport to be used for the MA USB protocol: directly over MAC or over IP
2. Information about port numbers.

The MA USB component performs USB device enumeration and USB data transfers.

2.1 General requirements

As a minimum, a Wi-Fi Serial Bus Host/Device shall support:

- Wi-Fi Direct, including mandatory support for the following optional features:
  - P2P Service Discovery
  - Persistent groups using the P2P invitation procedure

Furthermore, a Host/Device is recommended to have the Wi-Fi Direct Services Toolkit/Enable certification.

Hosts shall support the Wi-Fi Serial Bus service seeker role as defined in this specification. Devices shall support the Wi-Fi Serial Bus service advertiser role as defined in this specification.

Hosts shall support transport of MA USB protocol directly over MAC and over the IP transport. Devices shall support transport of MA USB protocol over MAC, over IP, or both.

Hosts and Hubs shall support all MA USB transfer types such as Control, Interrupt, Bulk and Isochronous. Peripherals shall support the Control transfer type, and may support Interrupt, Bulk and/or Isochronous transfer types.

For a Host/Device that supports either USB 2.0 or USB 2.0 and 3.1, the minimum IEEE 802.11 technologies that need to be supported are listed in Table 3 for all valid combinations of Wi-Fi certified radios supporting Wi-Fi Serial Bus. For example,

- A Peripheral with support for USB 2.0 only, equipped with 2.4 GHz and 5 GHz radios supporting Wi-Fi Serial Bus, shall support the Wi-Fi Serial Bus protocol over either 802.11g or 802.11n.
- A Peripheral with support for USB 2.0 and 3.1, equipped with 2.4 GHz and 5 GHz radios supporting Wi-Fi Serial Bus, shall support the Wi-Fi Serial Bus protocol over either 802.11g or 802.11n and over 802.11ac.

Table 3. Minimum Wi-Fi CERTIFIED radio requirements for Host/Device

Wi-Fi CERTIFIED radios supporting Wi-Fi Serial Bus	USB support[1]	Minimum requirements		
		Host	Hub	Peripheral
2.4 GHz	USB 2.0	802.11g and n	802.11g and n	802.11g or n
	USB 2.0 and 3.1	NA[2]	NA[2]	NA[2]
2.4 GHz / 5 GHz	USB 2.0	802.11a, g, and n	802.11a, g, and n	802.11g or n [5]
	USB 2.0 and 3.1	802.11a, g, n, and ac	802.11a, g, n, and ac	802.11g or n [3] and 802.11ac [5]
2.4 GHz / 5 GHz / 60 GHz	USB 2.0	802.11a, g, and n	802.11a, g, and n	802.11g or n [5]
	USB 2.0 and 3.1	802.11a, g, and n and 802.11ad [6]	802.11a, g, and n and 802.11ad [6]	802.11g or n [3], [5] and 802.11ad [5]
2.4 GHz / 60G Hz	USB 2.0	802.11g and n	802.11g and n	802.11g or n
	USB 2.0 and 3.1	802.11g, n and ad	802.11g, n and ad	802.11g or n [3] and 802.11ad
60 GHz	USB 2.0	NA[3]	NA[3]	802.11ad
	USB 2.0 and 3.1	802.11ad	802.11ad	802.11ad

Notes:

1. USB support does not refer to the physical USB connector. It refers to the USB protocol as defined in [7], [8].
2. 802.11ac or 802.11ad is required for USB 3.1 support.
3. Peripherals with USB 3.1 support shall support enumeration at 802.11a, g, or n, even if the peripheral function requires a higher speed.
4. 60 GHz-only Wi-Fi Serial Bus Hosts and Hubs shall provide USB 3.1 support.
5. Support of Wi-Fi Serial Bus over 802.11a is optional.
6. Support of Wi-Fi Serial Bus over 802.11ac is optional.

2.2 Wi-Fi Serial Bus Service Setup procedure

The Wi-Fi Serial Bus Service Setup procedure is illustrated in Figure 2.

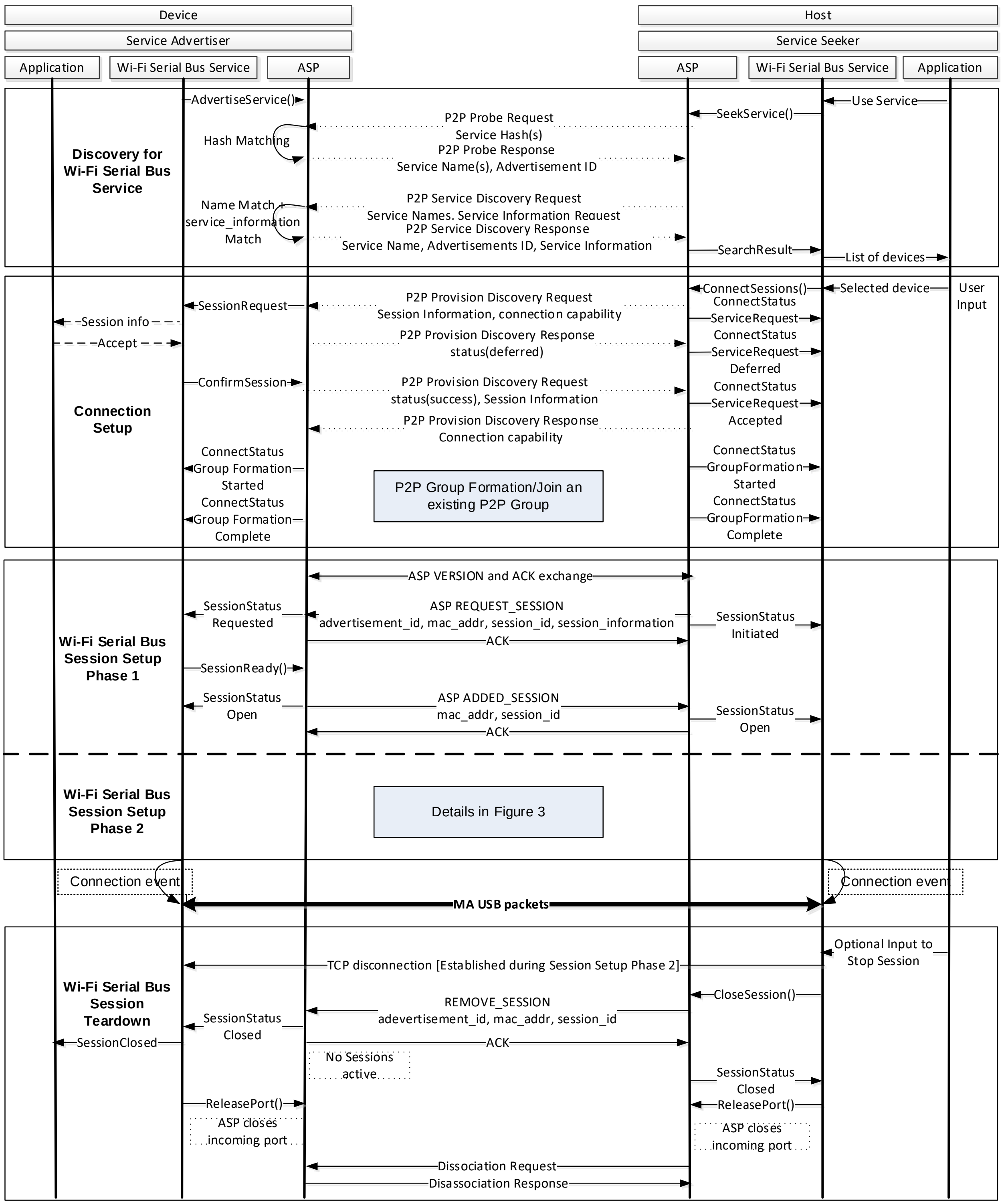

Figure 2. Wi-Fi Serial Bus Service setup

The Wi-Fi Serial Bus Service setup procedure includes three phases: Wi-Fi Serial Bus Discovery, Wi-Fi Serial Bus Connection Setup, and Wi-Fi Serial Bus Session Setup. Over-the-air messaging of each phase is executed by the ASP as described in [2] with changes specified in this specification.

In Wi-Fi Serial Bus Discovery, a Host/Device acting as a service advertiser advertises its service over WFDS ASP. The service is advertised with a service name (see section 3.1) and additional service information (see section 12). A counterpart Host/Device acting as a service seeker discovers such a service by performing hash matching of the service name, after which it can perform more detailed service discovery using a P2P Service Discovery Request/Response (Generic Advertisement Service) GAS frame exchange as defined in Annex G of [1].

In Wi-Fi Serial Bus Connection Setup, a Host/Device connects to a counterpart Host/Device at the link layer in a Wi-Fi P2P Group by using Provision Discovery Request/Response frames and P2P Group Owner (GO) negotiation/P2P association frames as specified in [2].

In Wi-Fi Serial Bus Session Setup, a Host/Device sets up an ASP session with the connected counterpart Host/Device by using the ASP coordination protocol as specified in [2]. The Host/Device that initiates the Wi-Fi Serial Bus Session Setup identifies the transport mode with which it intends to operate for its MA USB protocol. The counterpart Host/Device uses the received session information to decide whether to accept the Wi-Fi Serial Bus Session Setup attempt.

If the selected transport mode is IP, then the Wi-Fi Serial Bus Session Setup continues with Phase 2 as depicted in Figure 3. This initial TCP connection is used by the Host for transport of MA USB management traffic. It may also be used for transport of non-isochronous MA USB data.

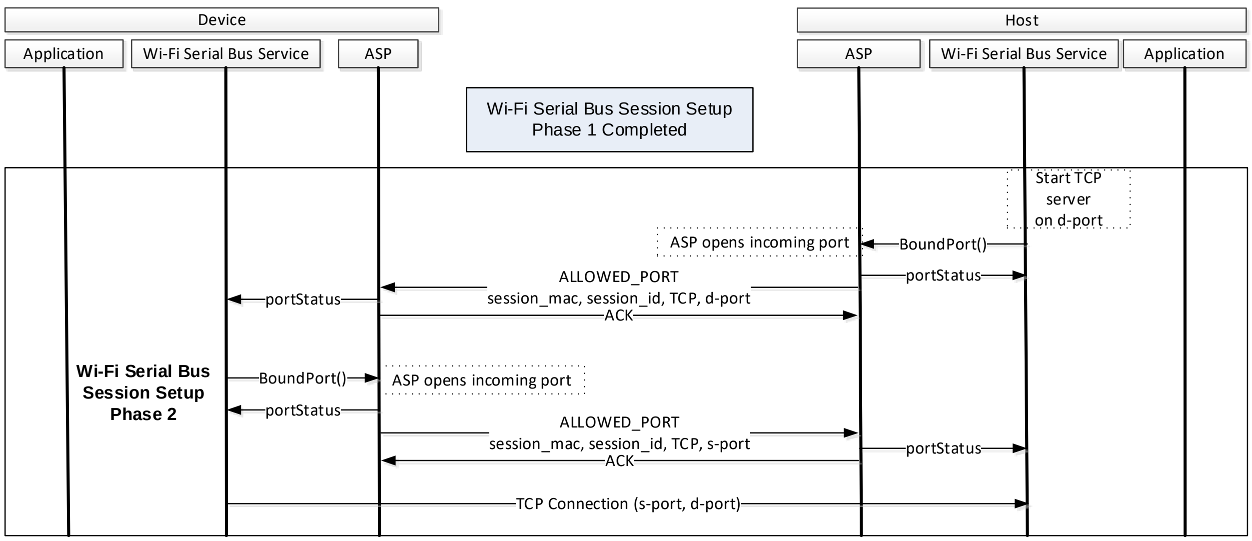

Figure 3. Wi-Fi Serial Bus Session Setup Phase 2

Once the Wi-Fi Serial Bus Session Setup is completed, the Host activates the MA USB module, which performs USB device enumeration and data transfers over the Wi-Fi connection set up by the Wi-Fi Serial Bus Service.

The Wi-Fi Serial Bus Service Teardown procedure terminates the Wi-Fi Serial Bus service. The procedure can be triggered by the Wi-Fi Serial Bus service at either the Host or the Device. Over-the-air messaging of this procedure is executed over WFDS ASP as described in [2] with changes specified in this specification.

Operations of these procedures are defined in the following sections.

3 Discovery

The Wi-Fi Serial Bus Discovery procedure is executed using the WFDS ASP. The Wi-Fi Serial Bus Discovery procedure enables a service advertiser to advertise its Wi-Fi Serial Bus services and enables a service seeker to search for devices that provide services of its interests.

A Host shall be able to operate as a service advertiser. A Host shall be able to operate as a service seeker for Devices. A Device shall be able to operate as a service advertiser. A Device may be able to operate as a service seeker for Hosts.

The primitives in sections 3.4.1 and 3.4.2 of [2] are used to denote the Wi-Fi Serial Bus service specific requirements for discovery using the WFDS ASP. For primitives that are not listed, no Wi-Fi Serial Bus specific behavior is defined. Section 3.4.3 of [2] specifies how these primitives translate into using P2P Probe Request/Response frames, Service Discovery Request/Response GAS frames and hash values for discovery.

3.1 Service name structure convention

A Wi-Fi Serial Bus Service is identified during the Wi-Fi Serial Bus Discovery process by its service name.

The service name advertised by a Host shall have the structure:

```
_wsbhost
```

The service name advertised by a Device shall have the structure:

```
_wsbdev-<baseclass>-<subclass>-<protocol>.<transport protocol>
```

Note that the rectangular brackets and the text included within those brackets (i.e. <baseclass>, <subclass>, <protocol>, and <transport protocol>) indicate placeholders that need to be replaced with values based on the rules defined below:

1. The value <baseclass> shall be set to the two hexadecimal digits that represent a USB base class code [5]. The value <subclass> shall be set to the two hexadecimal digits that represent a byte-long USB subclass code [5]. The value <protocol> shall be set to the two hexadecimal digits that represent a byte-long USB protocol code [5]. The codes in the "<baseclass>-<subclass>-<protocol>" substring indicate the primary function the Device intends to advertise as a Wi-Fi Serial Bus service. This information can be derived from a USB descriptor of the USB device such as the device descriptor, interface descriptor, IAD descriptor, etc. The information helps the Host identify the device's functionality of interest and its ability to load a device driver based on that functionality [5]. If the USB device associated with a Device has multiple interfaces or multiple configurations that have multiple USB interfaces, the Device shall advertise one or multiple Wi-Fi Serial Bus Services, each identified by a service name that contains the <baseclass>-<subclass>-<protocol> tuple of the corresponding interface or IAD descriptor.
2. A Hub shall advertise at least one Wi-Fi Serial Bus Service which includes the "<baseclass>-<subclass>-<protocol>" substring of the embedded USB hub device in the advertised service name. A Hub may optionally advertise Wi-Fi Serial Bus services for its downstream USB devices by advertising service names that contain the "<baseclass>-<subclass>-<protocol>" substring representing the USB descriptors of its downstream USB devices.
3. A Peripheral shall advertise at least one Wi-Fi Serial Bus Service with a service name that contains the "<baseclass>-<subclass>-<protocol>" substring representing the USB descriptor of its embedded USB device.
4. The value <transport protocol> identifies the transport protocol of the service data, which in this case is the MA USB protocol data. When set to "_tcp", the transport protocol of the MA USB data is over IP. When set to "_udp", the transport protocol of the MA USB data is directly over MAC.

For example, a Hub with a Full speed USB hub that has a USB mouse device attached downstream shall advertise a Wi-Fi Serial Bus Service with a service name _wsbdev-09-00-00._tcp, where "09-00-00" identifies the base class, subclass and protocol tuple of a Full speed USB hub. It may also advertise a Wi-Fi Serial Bus service with a service name _wsbdev-03-00-02._udp, where "03-00-02" identifies the base class, subclass and protocol tuple of a USB mouse device.

3.2 Wi-Fi Serial Bus SeekService primitive

```
SeekService(service_name, exact_search, mac_address, service_information_request)
```

A Host/Device that operates as a service seeker shall deploy the WFDS SeekService primitive [2] for use with Wi-Fi Serial Bus as described in this section.

A Device seeking a Host shall use the value "_wsbhost" for parameter service_name and value TRUE for parameter exact_search.

A Host that seeks for a Device, with the need of identifying a specific type of USB peripheral/hub and with a specific transport protocol supported, shall use the value TRUE for parameter exact_search, and set the parameter service_name to a string formatted as "_wsbdev-<baseclass>-<subclass>-<protocol>.<transport protocol>". For example: if the Host seeks a Wi-Fi Serial Bus mouse peripheral using MAC transport for MA USB protocol data, the portion "<baseclass>-<subclass>-<protocol>.<transport protocol>" shall be set to identify a USB mouse and direct over MAC transport as in service name _wsbdev-03-00-02._udp.

If the Host intends to discover a Hub without the need to discover of any specific USB devices being connected to the USB hub, the portion "<baseclass>-<subclass>-<protocol>" shall be set to identify a USB hub as in service name _wsbdev-09-00-00._tcp.

If the Host intends to discover a specific USB device connected to a Hub, and the corresponding Hub supports advertising services for its downstream USB devices, the portion "<baseclass>-<subclass>-<protocol>" may be set to identify the USB device of interest. Alternatively, or before doing so, the Host may inspect the service_information of a discovered Hub to find information about downstream devices from the values under the keys: "downstreamInfoAvailable" and "downstreamDevice".

Retrieving the service_information is done as follows. If the Host receives a response from a Device that offers a Wi-Fi Serial Bus service with a service name that exactly matches the service name that the Host is searching for, the Host may send a Service Discovery Query using a GAS frame for discovering detailed information of the Wi-Fi Serial Bus service provided by the Device.

When sending a Service Discovery Query using a GAS frame, the service seeker shall include the value of the service_information_request parameter as part of the Service Information Request field in the GAS frame. This is used for substring matching by the service advertiser. If the value of the Service Information Request field in the GAS frame is a valid substring, or has the value NULL (for example, the Service Information Request field is omitted by setting the Service Information Request Length field to value 0), then the service advertiser returns a GAS frame that includes the complete service information data as specified in the service_information parameter of the AdvertiseService primitive.

When the Host intends to find any Device, without the need of identifying any specific type of USB devices or transport protocol type, the service_name field shall be set to "_wsbdev", the parameter exact_search shall be set to FALSE, and the substrings "<baseclass>-<subclass>-<protocol>" and "<transport protocol>" shall be omitted in the service_name field. If the Host receives a response from a Device that offers a Wi-Fi Serial Bus service with a service name that includes "_wsbdev" as a prefix and advertisement ID 0, the Host shall send the Service Discovery Query using the GAS frame with the Service Name field having value "_wsbdev" appended with "*" for discovering detailed information of the Wi-Fi Serial Bus services provided by the Device.

NOTE: this is the Wi-Fi Serial Bus specific way of doing prefix search, which does not rely on the use of the "org.wi-fi.wfds" prefix.

When a Host discovers multiple Wi-Fi Serial Bus service names from the same Device and the Host needs to send the Service Discovery Query using the GAS frame for discovering detailed information of the Wi-Fi Serial Bus services provided by the Device, the Host shall only include one service name in one Service Discovery Query.

A Host may aggregate multiple SeekService requests by including multiple service hashes in the Service Hash attribute of the Probe Request frame.

3.3 Wi-Fi Serial Bus AdvertiseService primitive

```
AdvertiseService(service_name, auto_accept, service_information, service_status, network_role,
network_config, deferred_session_response)
```

A Host/Device that operates as a service advertiser shall deploy the WFDS AdvertiseService primitive. The parameters of this primitive are defined in section 3.4 of [2], however the restriction to not allow underscore '_' characters in service names does not hold for Wi-Fi Serial Bus. The parameters that require Wi-Fi Serial Bus specific context for their use are described in this section.

A Host that wishes to advertise its Wi-Fi Serial Bus service shall use the value "_wsbhost" for parameter service_name.

A Device that wishes to advertise its Wi-Fi Serial Bus service shall specify the service_name to be "_wsbdev-<baseclass>-<subclass>-<protocol>.<transport protocol>" as specified in the service name structure convention in section 3.1.

A Hub is only required to advertise services that identify the USB hub functionality itself. The Hub may advertise Wi-Fi Serial Bus services that identify USB functionalities offered by its downstream USB devices. If the Hub supports advertising services for its downstream USB devices, it shall include the key "downstreamInfoAvailable" with value TRUE in the service_information field, and include a listing of all its downstream USB devices for which it advertises a separate service using the key "downstreamDevice" with the service names of the Wi-Fi Serial Bus services representing downstream USB devices of the Hub.

The service_information field for a Device is a UTF-8 string without NULL termination, which contains Wi-Fi Serial Bus operation specific information. The format of the service_information field is defined in section 12.

The Device shall create a hash for each full Wi-Fi Serial Bus service name it advertises. In addition, the Device shall create an additional hash for "_wsbdev" to enable prefix search. If the Device receives a Probe Request frame in which the service_name's hash matches that of a full Wi-Fi Serial Bus service name, the Device shall respond with a Probe Response frame that includes the full Wi-Fi Serial Bus service name. If the Device receives a Probe Request frame in which the service name's hash matches that of "_wsbdev", the Device shall respond with a Probe Response frame that includes "_wsbdev" with advertisement ID 0 as its Wi-Fi Serial Bus service.

3.4 Discovery of the Host/Device user friendly name

A Host/Device shall identify its device name by specifying a user friendly description in the Device Name attribute in the WSC IE [4] in its Probe Request and Probe Response frames, in addition to putting it in the service discovery information using the friendlyName element.

3.5 Discovery of the Host/Device device Type

A Host/Device shall identify its function as a P2P device by including a category ID and a subcategory ID in the primary device type attribute in the WSC IE [4] in its Beacon frame and Probe Response frames.

3.6 Discovery of Wi-Fi Serial Bus service availability

If a Host/Device is the P2P GO of its P2P Group, it shall respond to any Probe Request frame it receives that contains a Service Hash value matching the hash value of its advertised Wi-Fi Serial Bus service and respond to any Service Discovery Requests that queries the service information of its advertised Wi-Fi Serial Bus service.

Following the Wi-Fi Direct Services specification, if a Host/Device is a P2P Client of a P2P Group, it shall respond to any Probe Request frame it receives on its P2P Group operating channel that contains a Service Hash value matching the hash value of its advertised Wi-Fi Serial Bus service and respond to any Service Discovery Request frame it receives on its P2P Group operating channel that queries the service information of its advertised Wi-Fi Serial Bus service.

A Host/Device shall identify its Wi-Fi Serial Bus service availability by setting a proper value for the service status field [5] in the Service Info Descriptor of its advertised Wi-Fi Serial Bus service.

4 Connection and session management

The Wi-Fi Serial Bus Connection Setup procedure and Wi-Fi Serial Bus Session Setup procedure are executed using the WFDS ASP. The Wi-Fi Serial Bus Connection Setup procedure establishes a P2P connection between the Host and the Device. After the P2P connection is established, Wi-Fi Serial Bus Session Setup procedure establishes an ASP session over the P2P connection for the Wi-Fi Serial Bus service using the ASP coordination protocol.

The service seeker shall launch the Wi-Fi Serial Bus Connection Setup procedure and the Wi-Fi Serial Bus Session Setup procedure. The service seeker may be either a Host or a Device.

Once a Wi-Fi Serial Bus session is established, the MA USB Host at the Wi-Fi Serial Bus Host drives the MA USB operation.

The Wi-Fi Serial Bus Service provides specific information to the ASP so that the ASP can use the underlining mechanisms, including P2P Provision Discovery, P2P Group Formation, and P2P Invitation, to set up the connection.

The Wi-Fi Direct Services primitives defined in section 3.6 of [2] are used to denote the Wi-Fi Serial Bus service specific requirements for Wi-Fi Serial Bus Connection Setup and Wi-Fi Serial Bus Session Setup. For primitives that are not listed, no Wi-Fi Serial Bus-specific behavior is defined. Sections 3.5 and 3.10 of [2] specify how these primitives translate into using P2P Provision Discovery Request/Response frames, P2P Group Formation procedures for P2P connection setup, and the use of the ASP coordination protocol for session setup.

4.1 Wi-Fi Serial Bus Connection Setup and Session Setup

4.1.1 Wi-Fi Serial Bus ConnectSessions primitive

```
ConnectSessions(List of(service_mac, advertisement_id), session_information, network_role,
network_config)
```

A Host/Device that operates as a service seeker shall deploy the WFDS ConnectSessions primitive [2] for use with Wi-Fi Serial Bus as described in this section.

The session_information is a single character UTF-8 string without NULL termination which has either value “0” or “1”, indicating the proposed transport for the MA USB protocol. The value “0” indicates that the proposed transport for the MA USB protocol is directly over the MAC. The value “1” indicates that the proposed transport for the MA USB protocol is the IP transport. A Host should take into account the transport modes that the Device supports as advertised as part of <transport protocol> suffix of the service name, either “._tcp” or “._udp” as specified in section 3.1. The session_information parameter is used to populate the Session Information Data Info field as part of the Provision Discovery Request frame. If this field is omitted when setting up the session, the default value “0” of using MA USB protocol directly over the MAC shall be assumed.

The network_role parameter is used for the Wi-Fi Serial Bus Service to identify whether the P2P device prefers to be a P2P GO for this Wi-Fi Serial Bus session or it has no preference. The connection can be established if the Host and the Device do not simultaneously request to be the P2P GO for the same Wi-Fi Serial Bus session. Either the Host or the Device can assume the P2P GO role of the P2P Group that includes the Host and the Device. The Device shall set network_role to 0 to indicate that it has no preference which P2P Group member role it assumes for the Wi-Fi Serial Bus session.

4.1.2 Wi-Fi Serial Bus SessionRequest event

```
SessionRequest(advertisement_id, session_mac, session_device_name, session_id, session_information)
```

A Host/Device that operates as a service advertiser shall support the WFDS SessionRequest event primitive in the manner described in this section.

When the Host/Device receives an incoming session request, the Wi-Fi Serial Bus Service should determine whether to accept the connection initiation based on the received information. The ASP then generates a SessionRequest event with the parameters as denoted above.

A Device shall only accept a session setup request from a Host if the Device supports the transport mode that the Host intends to operate on as specified in the session_information field. If the Device cannot accept the transport mode request of the Host, in case the Host in error indicates a transport mode not supported by the Device, the Device shall return an error "Fail; incompatible parameters" (i.e. value 2) through the Provision Discovery Response Status attribute. A Host supports both MAC and IP transport, hence this requirement does not apply to a Host.

The Host/Device may deliver the incoming session request to the Wi-Fi Serial Bus application layer if the session setup attempt is accepted by either user confirmation or if auto_accept is set to true.

4.1.3 Wi-Fi Serial Bus ConnectStatus event

```
ConnectStatus(session_mac, session_id, status, deferred_session_response)
```

After the Host/Device acting as advertiser has accepted the incoming connection/session request and sent a valid Provision Discovery Response frame, the connection status of the Host/Device acting as a service seeker changes to SessionRequestAccepted. This is communicated by the ASP through the ConnectStatus event primitive.

After the status has changed to SessionRequestAccepted, the Host and Device start P2P Group formation and/or P2P connection setup to an existing P2P Group as specified in section 3.10.3 of [2].

After the P2P connection is set up, both the Host and Device shall initiate the ASP Coordination protocol to finalize the setup of the ASP session as specified in sections 3.7 and 3.8 of [2], with the exception that the requirement that if an ASP Coordination protocol command is not acknowledged after 5 resends that the P2P connection be torn down does not hold for the initial ASP VERSION commands. The VERSION messages can be sent more than 5 times to account for any delay in readiness of ASP to respond.

4.1.4 Wi-Fi Serial Bus BoundPort primitive

```
BoundPort(session_mac, session_id, ip_address, port, proto)
```

The BoundPort primitive indicates that the service has bound the indicated port on the indicated ip_address. If port access is not already allowed, the ASP shall attempt to open the indicated port. An ALLOWED_PORT ASP Coordination protocol message shall be sent to the remote ASP of the ASP session to indicate that the port has become available for incoming traffic. Note that the ALLOWED_PORT messages are expected to be sent after successful completion of the ASP session setup using REQUEST_SESSION/ADDED_SESSION messages. The parameters of this primitive are defined in [2], section 3.6.

The use of the BoundPort primitive and the related ALLOWED_PORT messages is only relevant if the Wi-Fi Serial Bus service is deployed over IP, in which case the Wi-Fi Serial Bus service on the Host shall open a TCP port, and the Host shall send an ASP ALLOWED_PORT message to the connected Device to allow incoming connections from the Device on that TCP port. After receiving the ASP ALLOWED_PORT message from the Host, the Device shall send a corresponding ASP_ALLOWED_PORT message to the Host with its opened incoming TCP port and then initiate a TCP connection to the allowed TCP port on the Host. This first established TCP connection after establishing the ASP session serves as the default communication link for the MA USB protocol between the Host and the Device; i.e., it serves as the MA USB management channel, and it may also be used for non-isochronous MA USB traffic. Figure 3 illustrates this initial TCP connection setup.

If the Wi-Fi Serial Bus Service is deployed over the IP transport, and there are isochronous endpoints enumerated, the Host shall use the BoundPort primitive to signal to the Device that a UDP port is opened for isochronous traffic and send a corresponding ASP ALLOWED_PORT message to the connected Device. After receiving the ASP ALLOWED_PORT message from the Host, the Device shall send a corresponding ASP_ALLOWED_PORT message to the Host with its opened incoming UDP port. This UDP port-pair can be used for isochronous traffic. Figure 4 illustrates UDP session setup and release.

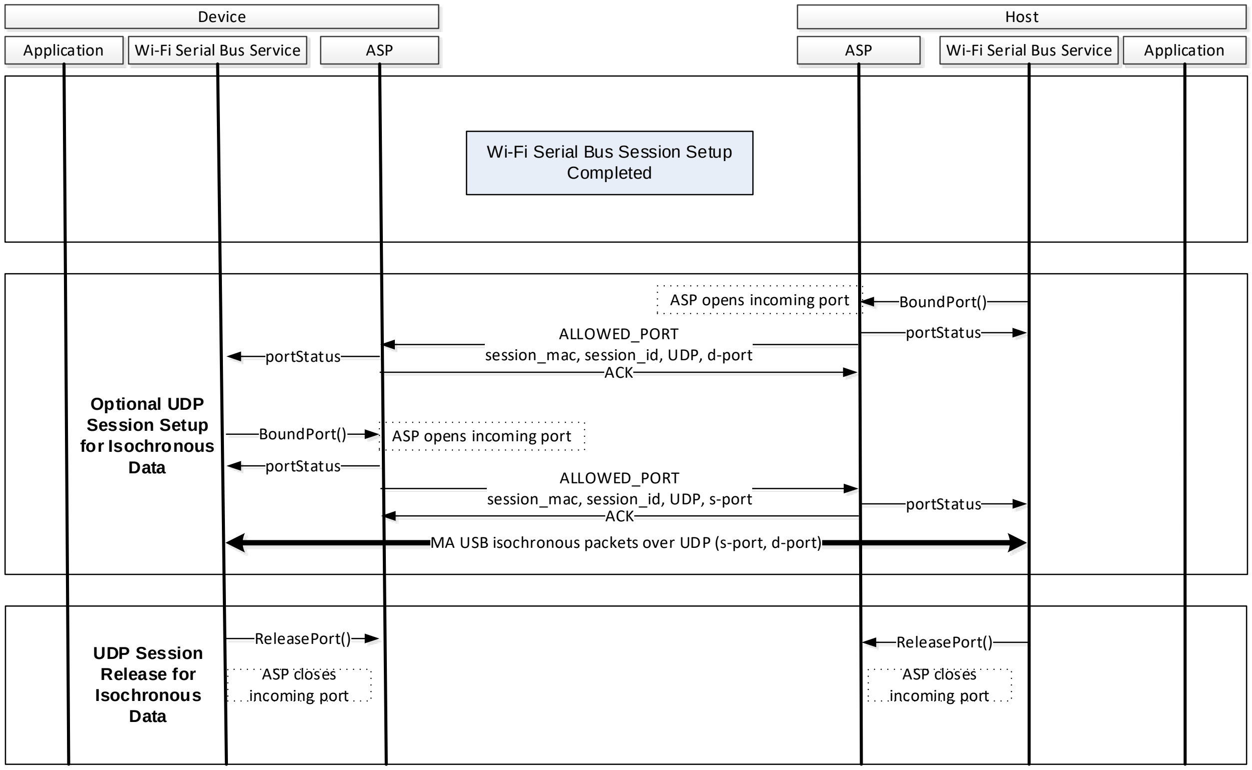

Figure 4. UDP session setup and release

At any time during MA USB protocol operation, the Host may choose to establish additional TCP and UDP sessions for different traffic streams within the MA USB protocol. For each allowed port on the Host, including the initial TCP port, the Device learns the port number through the ASP ALLOWED_PORT message before establishing a connection to that port. UDP ports are strictly for isochronous traffic. No isochronous traffic shall be transmitted over a TCP connection. A Device shall respond to the MA USB protocol messages received from the Host on the same connection on which they have been received. Figure 5 illustrates optional additional TCP session setup and release. A TCP session release can be initiated by either the Host or the Device.

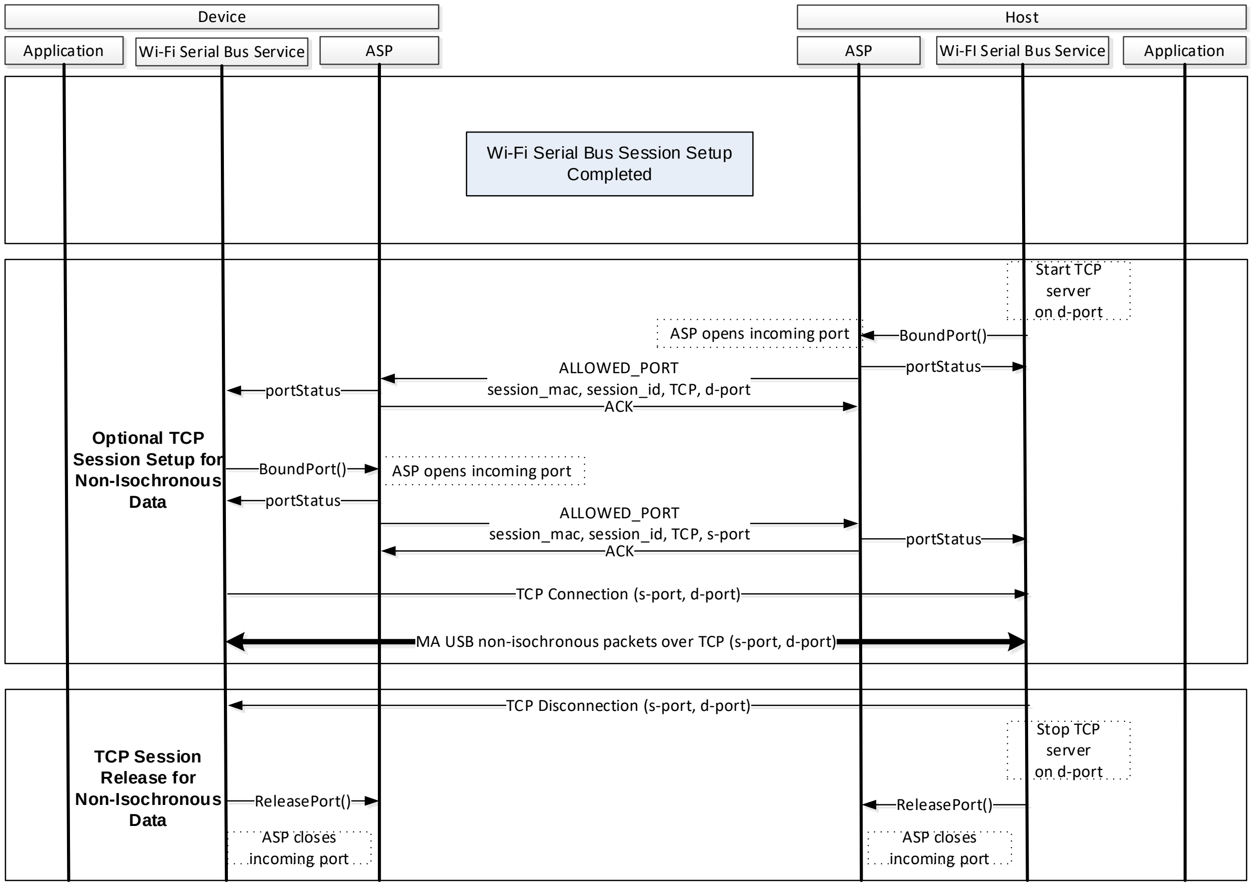

Figure 5. Optional additional TCP session setup and release

4.2 Wi-Fi Serial Bus Connection recovery and Session Teardown

4.2.1 Wi-Fi Serial Bus Connection recovery

Upon the occurrence of connection failure events, the P2P Client of the P2P Group of the temporarily failed P2P connection should try re-joining the P2P Group of the temporarily failed P2P connection before the P2P Client closes the Wi-Fi Serial Bus session. The P2P GO of the P2P Group of the temporarily failed P2P connection may choose to close its Wi-Fi Serial Bus session if the connection is not restored by any connection re-establishment attempts within 20 seconds.

4.2.2 Wi-Fi Serial Bus Session Teardown

A Wi-Fi Serial Bus Session Teardown shall be triggered if the Wi-Fi Serial Bus Service receives an explicit instruction from MA USB or the application to tear down the Wi-Fi Serial Bus session.

The MA USB may send an indication to the Wi-Fi Serial Bus service to terminate the Wi-Fi Serial Bus session, for example due to internal timeouts. In addition, the Wi-Fi Serial Bus service may autonomously close the Wi-Fi Serial Bus session when connection failure events happen.

When the Wi-Fi Serial Bus service autonomously closes the Wi-Fi Serial Bus session, the Wi-Fi Serial Bus service shall indicate the termination of the Wi-Fi Serial Bus session to MA USB.

The Wi-Fi Serial Bus Service performs the Wi-Fi Serial Bus Session Teardown using the Wi-Fi Serial Bus CloseSession primitive.

4.2.3 Wi-Fi Serial Bus CloseSession primitive

```
CloseSession(session_mac, session_id)
```

A Host and a Device shall implement the CloseSession primitive. The parameters of this primitive are defined in [2], section 3.6.

If the Wi-Fi Serial Bus Service is deployed over the IP stack, once CloseSession is executed, the port that is not associated with any other open ASP Sessions shall be blocked.

If the Wi-Fi Serial Bus Service is deployed directly over MAC, once CloseSession is executed, the Wi-Fi Serial Bus Service shall stop processing Wi-Fi Serial Bus packets.

5 MA USB transfers

Once the Wi-Fi Serial Bus session has successfully been established, the MA USB protocol is activated to perform USB device enumeration and USB data transfers. The operations of USB device enumeration and USB data transfers are specified by the MA USB protocol [3].

The MA USB protocol may operate over the IP transport or directly over MAC.

The Host makes a decision on which transport to use for the MA USB protocol based on the supported transport modes advertised by the Device as part of the <transport protocol> suffix of the service name and the Host's own preferred mode. The details of how to set up a session requesting a particular transport protocol are described in section 4.1.1. The details on how to open the necessary TCP and UDP ports in case the MA USB protocol operates over IP transport are described in section 4.1.4.

When the MA USB protocol is deployed directly over the MAC transport without IP, the MA USB protocol messages are encapsulated using a Subnetwork Access Protocol (SNAP) header with the first three bytes set to the Wi-Fi Alliance specific OUI value: 0x50 6F 9A, the fourth byte set to the Wi-Fi Serial Bus specific OUI Type value: 0x15, and the fifth byte set to 0x00. The packet structure is illustrated in Figure 6. The MAUSB Protocol Data field of the Wi-Fi Serial Bus packet carries MA USB protocol data.

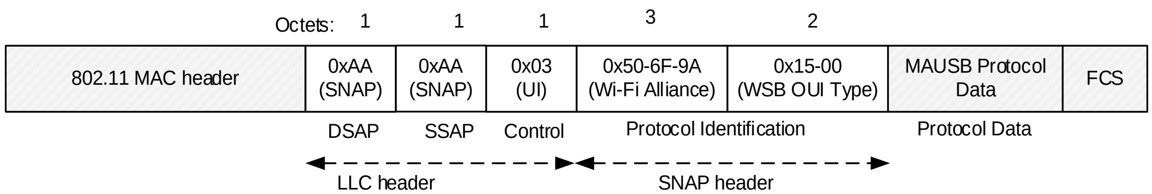

Figure 6. MA USB protocol data encapsulation directly over MAC

6 Reliability

6.1 Reliable transmissions

When the MA USB transport is directly over MAC, the Host and the Device should enable the MAC level retransmissions to recover transmission failures of MA USB messages at least for non-isochronous transfers. Alternatively, a Host and Device may deploy MA USB over IP using TCP/IP for non-isochronous transfers and UDP/IP for isochronous transfer.

For more information about how to handle temporary disconnections, see section 4.2.1.

6.2 Error indications for Wi-Fi Serial Bus link exceptions

When the data rate of the underlying radio is not capable of supporting the throughput and the latency requirement of an isochronous transfer endpoint enumerated at the Host, the Host should indicate this to the application that utilizes the Host function.

When the USB operation at the Host encounters errors identified by the MA USB host, the Host should indicate this to the application that utilizes the Host function.

7 MA USB Global Time synchronization for Wi-Fi Serial Bus

The MA USB protocol defines the mechanisms for synchronization of the MA USB Global Time (MGT) between an MA USB host and an MA USB hub or device. There are two MGT synchronization methods defined in the MA USB specification [3]: Transmission Delay based MGT synchronization and Media Time based MGT synchronization.

In the Transmission Delay MGT synchronization, the Transmission Delay field of the MA USB packets carries an estimate of the transmission delay the MA USB packet experiences in nanoseconds. The Media Time based MGT synchronization, however, relies on time synchronization at the underlying media between the MA USB host and the MA USB hub/device. In the Media Time based MGT synchronization, the MA USB host or device has access to a synchronized media time, which is carried in the Media Time field in the MA USB packets. The format of Media Time field is media-dependent. The Host and Device shall include the lowest 32 bits of the Timing Synchronization Function (TSF) counter in the Media Time field.

A Host, Hub and a Peripheral operating in 2.4/5GHz that supports isochronous endpoints shall support Media Time based MGT synchronization. A Host operating in 2.4/5GHz shall include the Synchronization Capabilities descriptor in the MA USB CapReq packet with the Media Time Available field set to one to indicate its access to Media Time. A Device operating in 2.4/5GHz shall set the Media Time Available field to one in Synchronization Capabilities descriptor if it includes the Synchronization Capabilities descriptor in the MA USB CapResp packet.

A Host, Hub and a Peripheral operating in 60 GHz that supports isochronous endpoints shall support the Transmission Delay based MGT synchronization and may support the Media Time based MGT synchronization. A Host and Device operating in 60 GHz may set the value of the Media Time Available field in the Synchronization Capabilities descriptor in the MA USB CapReq/Resp packet to one to indicate its access to Media Time. The negotiation procedure for determining which MGT synchronization method to use is defined in the MA USB Specification [3]. Wi-Fi Serial Bus devices operating in 60 GHz are recommended to also support the Media Time based MGT synchronization to enable uniform operation across all bands.

8 MA USB media dependent protocol constants for Wi-Fi Serial Bus

The MA USB protocol defines the protocol constants listed in Table 4 as media dependent.

Table 4. MA USB protocol constants and their definitions

Protocol Constant	Definition
aDataChannelDelay	The maximum time required for delivery of an MA USB control or data packet to the target MA USB Protocol Adaptation Layer (PAL) since its release to the data channel by the originating MA USB PAL.
aManagementChannelDelay	The maximum time required for delivery of an MA USB management packet to the target MA USB PAL since its release to the management channel by the originating MA USB PAL.
aMaxIsochLinkDelay	The maximum time required for delivery of an IsochTransferReq packet (if not dropped) to the target MA USB device network interface since the IsochTransferReq packet is released to the network by the originating MA USB PAL. Note: The moment of release and delivery are defined by the moment the first bit of the packet is released or delivered.

Table 5 specifies the values for the media dependent protocol constants when used in Wi-Fi Serial Bus for different connection types.

Table 5. The values of the MA USB protocol constants using Wi-Fi Serial Bus connection

Protocol Constant	2.4/5 GHz connection value	60 GHz connection value	IP connection value
aDataChannelDelay	50 msec	25 msec	100 msec
aManagementChannelDelay	50 msec	25 msec	100 msec
aMaxIsochLinkDelay	25 msec	25 msec	100 msec[1]
Notes: 1. USB isochronous traffic requires latencies of 25 msec or less for acceptable operation.			

9 Support for persistent P2P Group features

A Host and Device shall support P2P persistence using the P2P invitation as specified in the Wi-Fi P2P specification [1], and should support the persistent P2P Group features required by the ASP. The persistent P2P Group features required by the ASP are identified in [2].

A Host and Device shall support maintaining at least one persistent P2P Group record.

10 Power management

When a Host/Device operates in 2.4 or 5 GHz, the Host/Device shall support the following Wi-Fi power management mechanisms:

- Capabilities to support a P2P client operating with WMM™-Power Save when acting as Wi-Fi P2P GO, as specified in [1].
- Capabilities to follow Opportunistic Power Save and Notice of Absence from a Wi-Fi P2P GO when acting as a Wi-Fi P2P client, as specified in [1].

When a Host/Device operates in 60 GHz, the Host/Device shall support the following Wi-Fi power management mechanisms:

- Capabilities to support a P2P client operating with the non-PCP STA power save mechanism when acting as Wi-Fi P2P GO, as specified in §10.2.6 of [6] and if mandated by the Wi-Fi P2P certification.
- Capabilities to follow the PCP power save mechanism when acting as a Wi-Fi P2P client, as specified in §10.2.6 of [6] and if mandated by the Wi-Fi P2P certification.

In the MA USB session inactive state, the Host/Device shall configure its Wi-Fi Power Save mode so that it can meet the timing requirements specified by the Management Request Timeout or Data Request Timeout in the sleep request.

In the MA USB session active state, the Host/Device shall configure its Wi-Fi Power Save mode so that it can meet the timing requirements specified by the aManagementRequestTimeout or aTransferTimeout fields.

11 Security

A Host shall support both WFDS default configuration and WSC PIN [4] as its WSC configuration methods for connection.

A headless Device shall support WFDS default configuration as its WSC configuration method for connection, preferably with some audio/visual feedback or confirmation button to prevent setting up a connection with the wrong device.

A Device equipped with a display or keypad shall support WFDS default configuration and WSC PIN as its WSC configuration methods for connection.

The WFDS default configuration provides limited security and does not protect against man-in-the-middle attacks. Hence, Devices are expected to adopt additional security configuration method(s) for headless devices when they become available.

12 Format for the service_information field advertised by a Host/Device

The service_information field is a UTF-8 string without NULL termination in plain text format that contains Wi-Fi Serial Bus operation specific information. The text shall be formatted using ASCII codes in key-value pairs as follows:

```
keyX=value
keyY=[valueA valueB valueC]
```

The key-value pairs may be arranged in any order. A key may have one or more values based on its use.

There shall be no space in the key text string. There can be spaces in the value text string, but no preceding space or trailing space is allowed. When there are multiple values for a key, all value text strings shall be enclosed in square brackets ([]) and be separated by one or more space characters. No quotation marks are required around the value, even if it contains spaces, '=' characters, or other punctuation marks. The comma character ‘,’ and the semicolon character ‘;’ are not allowed neither as part of the key nor as part of the value(s).

A semicolon shall separate key-value pairs.

The key-value pairs of the service_information field used by the Host and Device are listed in Table 6.

Table 6. Key-Value pairs of the service_information field

Key	Value(s)	Presence
friendlyName	One text string that gives a user-friendly description of the Host/Device (e.g., “John’s Mass Storage USB Device at Home” or “Amanda’s Wi-Fi Serial Bus Capable Mobile Phone”).	Mandatory for Host/Device
wsbVersion	One or more text strings that identify the versions of the Wi-Fi Serial Bus specification that the Host/Device supports. The value 1.0 indicates the Host/Device supports Wi-Fi Serial Bus specification v1.0.	Mandatory for Host/Device
mausbVersion	One or more text strings that identify the versions of the MA USB specification that the Host/Device supports. The value 1.0a indicates the Host/Device supports USB-IF MA USB specification v1.0a.	Mandatory for Host/Device
usbVersion	One or more text strings that identify the versions of the USB specification that the Host/Device supports. The value 2.0 indicates the Host/Device supports USB 2.0. The value 3.1 indicates the Host/Device supports USB 3.1.	Mandatory for Host/Device
downstreamInfoAvailable	One text string that indicates if the Hub supports advertising services for its downstream USB devices that lists information about its downstream USB devices. • The value TRUE indicates the Hub supports advertising services for its downstream USB devices. When this value is used, there shall be the key-value pair of “downstreamDevice”. • The value FALSE indicates the Hub does not support advertising services for its downstream USB devices. This key-value pair shall be present for a Hub and shall not be present otherwise.	Mandatory for Hub
downstreamDevice	One or more text strings that identify the service name of a Wi-Fi Serial Bus service representing a downstream USB device of the Hub. The service name of a Wi-Fi Serial Bus service that represents a downstream USB device of the Wi-Fi Serial Bus shall have the same service name format as defined in section 3.1. This key-value pair shall be present if “downstreamInfoAvailable” is set to true.	Mandatory for Hub

Appendix A End-to-end operation scenarios (informative)

This Appendix provides several end-to-end operation scenarios targeting combined operation of Wi-Fi Serial Bus and MA USB.

Given that Wi-Fi Serial Bus mandates Hosts to support both IP and MAC transport of MA USB and support isochronous transfers, all three scenarios described in this Appendix apply to Hosts. Given that a Device may support either IP transport or MAC transport of MA USB and may or may not support isochronous transfer type (except for Hubs for which isochronous transfer type support is mandatory), one or more of the scenarios in this Appendix apply to Devices depending on the support of these features.

A.1 Scenario 1: Wi-Fi Serial Bus connection with TCP transport

The following scenario shows the Host and Device connection procedure if the Device supports IP transport. Note that TCP transport shall be used for all traffic types except for isochronous traffic.

Step 1:

If the Device supports MA USB IP transport, the Host and Device connection begins by the Host enabling Wi-Fi Serial Bus service as a Host and enabling the service seeker role for the underlying WFDS device discovery and service discovery. The Device is enabled with the Wi-Fi Serial Bus service as a Device and with the service advertiser role (using a service name ending with "._tcp") for the underlying WFDS device discovery and service discovery.

Step 2:

The Host discovers the Device's Wi-Fi Serial Bus service and displays or logs the Device information on its UI.

Step 3:

When the Host selects the Device's Wi-Fi Serial Bus service, it initiates the P2P connection and ASP session setup, after which the Device should provide some audio/visual feedback for the incoming connection request if possible.

Step 4:

Once the P2P connection is established and the ASP Session for the Wi-Fi Serial Bus service is initiated, the Host opens an ASP coordination protocol port to exchange ASP messages to the Device and sends ASP VERSION message(s). The Device also opens an ASP coordination protocol port, receives the ASP VERSION message and sends an acknowledgement, and sends ASP VERSION message(s) to the Host.

Step 5:

Once the Host has received the ASP VERSION message from the Device, and has acknowledged it, the Host sends a REQUEST_SESSION message with session_information field set to “1” (single octet with decimal value 49) to the Device. The Device sends an acknowledgement and sends an ADDED_SESSION message to Host.

Step 6:

The Host opens an initial TCP service port for MA USB, and exchanges with the Device an ALLOWED_PORT message with proto value set to 6 (TCP) to share the port information for the initial service port. The Host and Device then use this port information to open an initial TCP connection for MA USB management traffic.

Step 7:

The Host sends an MA-USB DevResetReq TCP packet to the Device. In response, the Device sends an MA-USB DevResetResp TCP packet.

A.2 Scenario 2: Wi-Fi Serial Bus connection with UDP transport

The following scenario shows the Host/Device connection procedure in case the Device supports IP transport with isochronous endpoint(s).

Steps 1 through 7:

The same as steps 1 through 7 of Scenario 1 in Appendix A.1.

Step 8:

The Host opens a UDP port for isochronous MA-USB traffic, and sends an ALLOWED_PORT message (with proto value 17 ‘UDP’) to the Device to share the port information for the isochronous communication. The Device opens a UDP port for isochronous MA-USB traffic, and sends an ALLOWED_PORT message (with proto value 17 ‘UDP’) to the Host to share the port information for the isochronous communication.

Step 9:

The Host sends an MA-USB IsochTransferReq packet over the UDP port. In response, the Device sends an MA-USB IsochTransferResp packet over the UDP port.

A.3 Scenario 3: Wi-Fi Serial Bus connection with MAC transport

The following scenario shows the Host/Device connection procedure in case the Device supports MAC transport.

Step 1:

If the Device supports MA USB MAC transport, the Host and Device connection begins by the Host enabling Wi-Fi Serial Bus service as a Host and enabling the service seeker role for the underlying WFDS device discovery and service discovery. The Device is enabled with the Wi-Fi Serial Bus service as a Device and with the service advertiser role (using a service name ending with "._udp") for the underlying WFDS device discovery and service discovery.

Step 2:

The Host discovers the Device's Wi-Fi Serial Bus service and displays or logs the Device information on its UI.

Step 3:

When the Host selects the Device's Wi-Fi Serial Bus service, it initiates the P2P connection and ASP session setup, after which the Device should provide some audio/visual feedback for the incoming connection request if possible.

Step 4:

Once the P2P connection is established and the ASP Session for the Wi-Fi Serial Bus service is initiated, the Host opens an ASP coordination protocol port to exchange ASP messages to the Device and sends the ASP VERSION message(s). The Device also opens an ASP coordination protocol port, receives the ASP VERSION message, sends an acknowledgement, and sends the ASP VERSION message(s) to the Host.

Step 5:

Once the Host has received and acknowledged the ASP VERSION message from the Device, the Host sends a REQUEST_SESSION message with session_information field set to “0” (single octet with decimal value 48) to the Device. The Device sends an acknowledgement and sends an ADDED_SESSION message to the Host.

Step 6:

The Host sends an MA-USB DevResetReq message as a MAC frame with the frame structure as specified in section 5 to the Device. In response, the Device sends an MA-USB DevResetResp MAC frame to the Host with the frame structure as specified in section 5.