ANT File Share (ANT-FS)

Technical Specification

Copyright Information and Usage Notice

Revision History

Revision	Effective Date	Description
2.2	January 2012	Updated document format. Added Revision Table. Added disconnect "undiscoverable mode" command Added to Command Pipe: set passkey, set friendly name, factory reset

Table of Contents

1 Introduction

ANT-FS is a session-based transport mechanism that has been designed to securely and automatically transfer data files between a host device and a client device. The ANT-FS system provides a robust framework for transferring files between two ANT enabled devices, and defines link, authentication, and transport layers between client and host devices.

This document describes an application level extension of the ANT protocol. The ANT-FS protocol defines the layers between the ANT protocol stack and the application layer. The ANT-FS protocol outlines the details of establishing a Transport Layer connection between an ANT-FS client device and a host.

2 Relevant Documents

It is assumed that the reader of this document is familiar with basic ANT concepts, and strongly recommended that the reader be familiar with the following documents:

- ANT Message Protocol and Usage
- ANT-FS Reference Design User Manual
- Interfacing with ANT General Purpose Chipsets and Modules
- AN04 – Burst Transfers

3 ANT-FS Overview

A typical ANT-FS use case is shown below in Figure 3-1. A user with an ANT-FS enabled watch collects data during an activity, which is then downloaded to a PC application at a later time.

On commencing the sports activity, the user enables data logging on the watch which will then collect and store sensor data during the activity, such as heart rate, speed, distance, etc. On finishing the activity, the watch (ANT-FS client device) shall transmit a beacon containing information about the device, such as its data availability and capabilities.

Following the workout, the user approaches an ANT-FS enabled PC (host device), which detects the watch's ANT-FS beacon. The host shall use the information in this beacon to determine how to management its connection to the client device. In this case, the PC application recognizes from the beacon that new data is available for download and initiates an ANT-FS session.

When the host chooses to initiate communication with the client, it sends a Link command to the client device; which reconfigures the connection parameters and enables further communication. Once a link has been established, the ANT-FS application specifies what level of authentication is required. The Authenticate command signals to the client device that a connection is desired, and initiates the authentication process. Once the link has been successfully authenticated, data transport may begin. If at any time the client device fails to receive new commands from the host device for a defined timeout period, the client device will fall back to its default, unconnected link state.

The necessary pairing and/or authentication shall be performed, and data downloaded from the watch to the PC fitness application. Data on the watch may or may not be erased, depending on the implementation or user input. In this use case, the watch is considered as the *client* device, and the PC acts as the *host*.

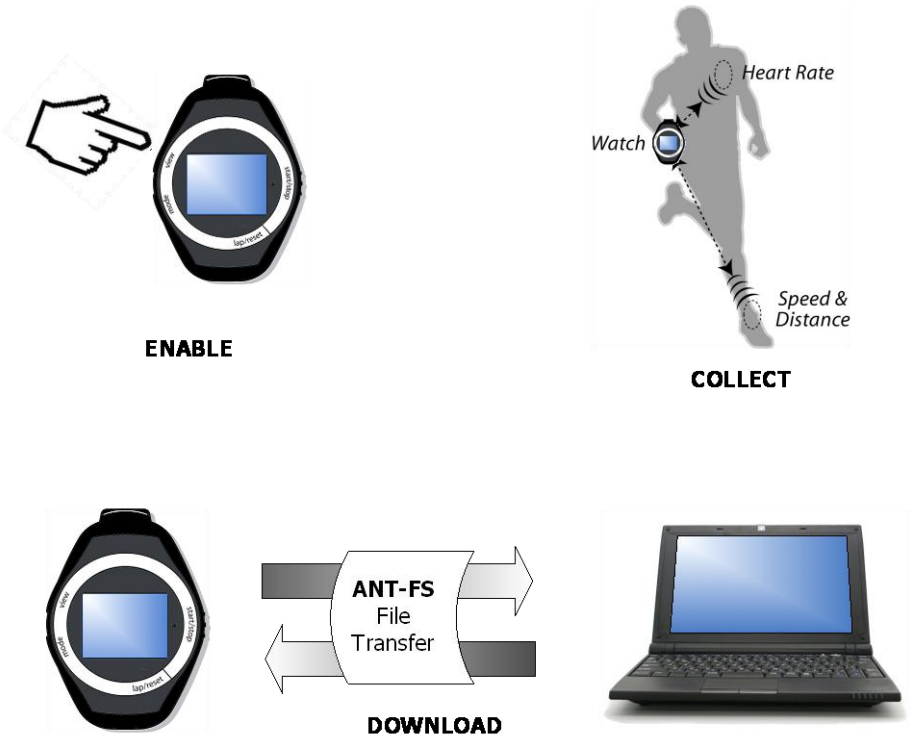

Figure 3-1. Watch Use Case

3.1 ANT-FS Client

The ANT-FS client is typically a mobile device which collects and stores sensor data which may be transferred to other devices. This data may be obtained from sensors broadcasting real time data (e.g. watch use case in Figure 3-1), or it may stored by the sensor itself (e.g. ANT+ blood pressure monitor) for transfer to a collecting device (i.e. the ANT-FS Host). The ANT-FS client device is configured as an ANT master.

3.2 ANT-FS Host

The ANT-FS host is typically a device looking to download sensor data from an ANT-FS client. The host searches for the client beacon and may decide to initiate an ANT-FS session if it is interested in receiving the data from the client. The host is typically the hub to which data is transferred (i.e. ANT enabled PC). The ANT-FS host device is configured as an ANT slave.

3.3 ANT-FS Beacon

When active, an ANT-FS client device always transmits an ANT-FS "beacon" signal containing current information about the device, such as its data availability, capabilities and state. The beacon is discoverable by a host device, and its information used to determine if a connection is desired, and how that connection should be managed.

The beacon is transmitted every channel period, and any ANT-FS operation (i.e. Command/Response) shall occur only on the host's request (Figure 3-2).

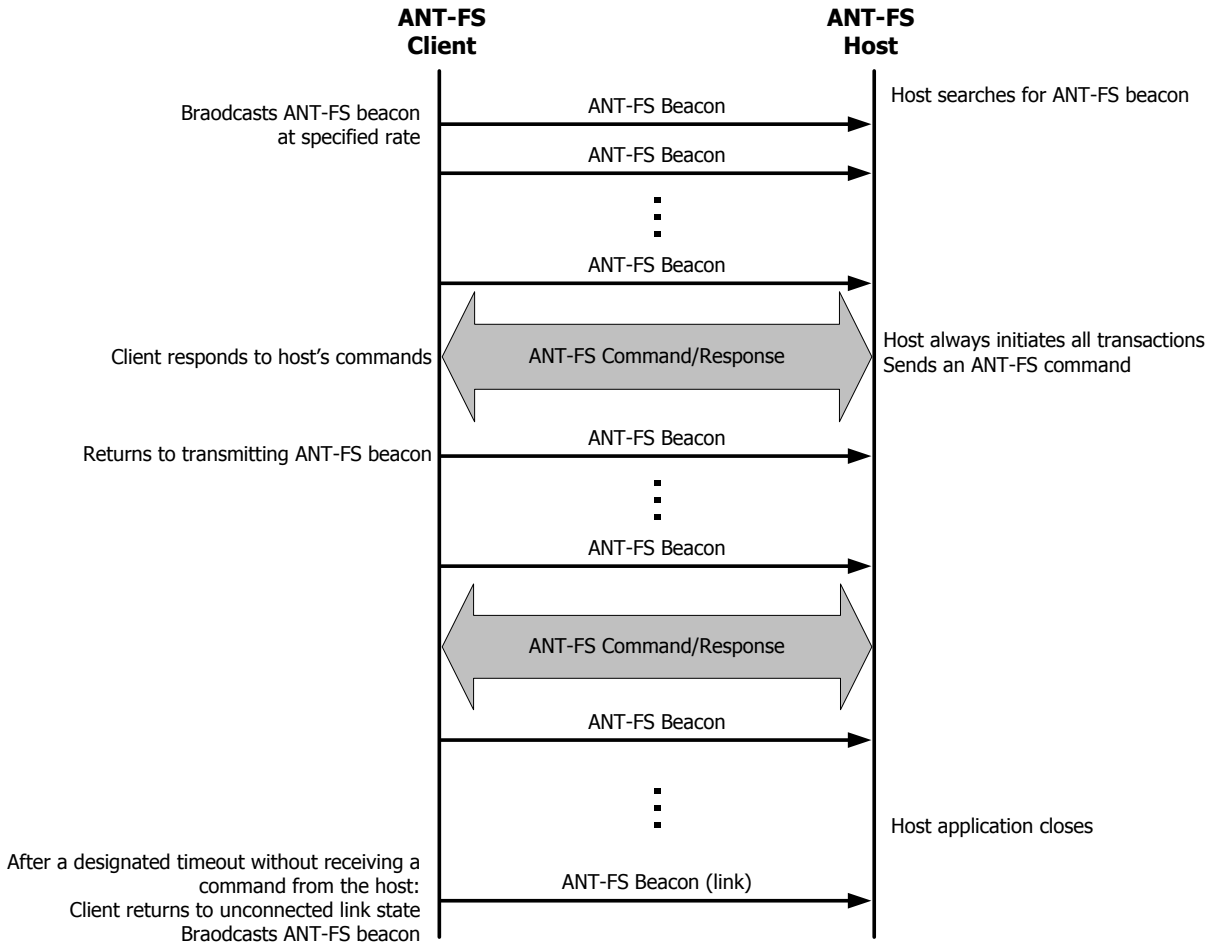

Figure 3-2. Client Transmits ANT-FS Beacon by Default

The host initiates an ANT-FS transaction by transmitting an ANT-FS command. The ANT-FS command is sent as an acknowledged message, or as a burst transfer if the command consists of multiple packets. The client device will respond by either broadcasting the ANT-FS beacon (indicating any changes) or with a burst transfer if the response requires multiple packets. If the client does respond with a burst transfer, the first packet of the transfer contains the ANT-FS beacon (Figure 3-3).

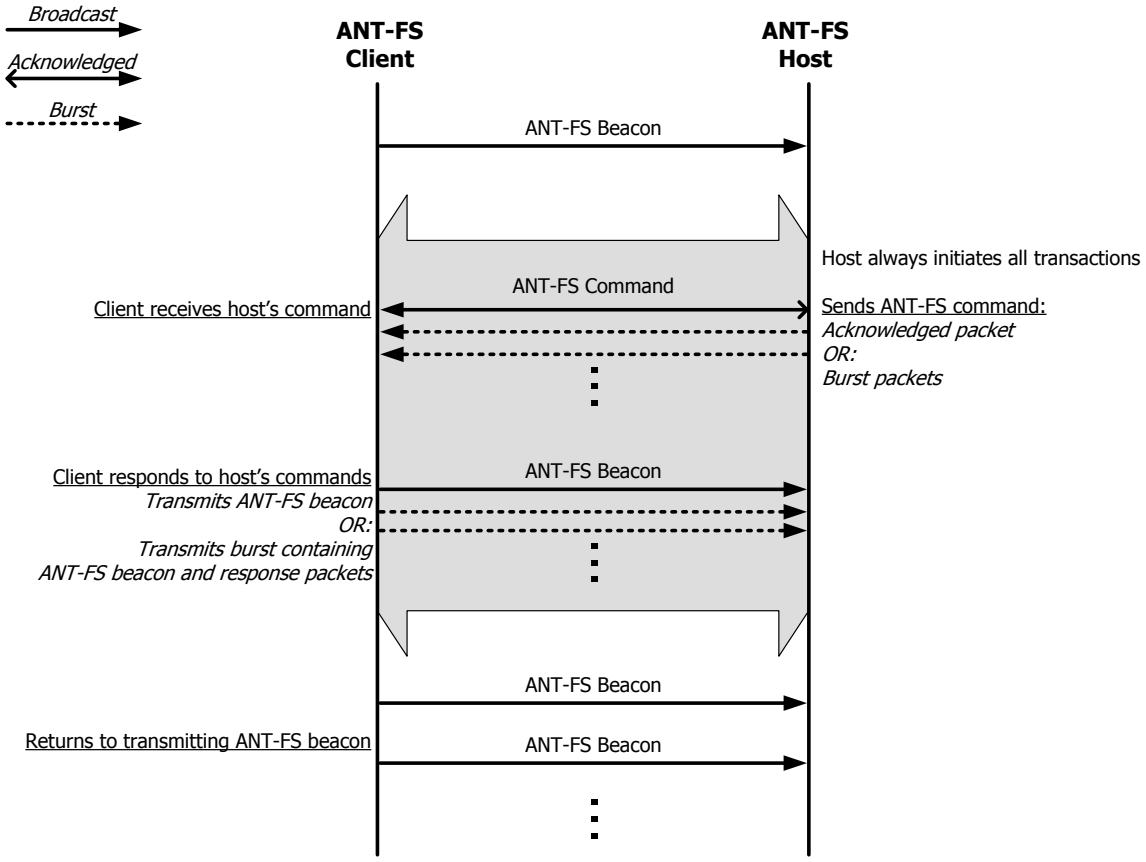

Figure 3-3. ANT-FS Command/Response

The beacon channel period is application dependent, and has a number of allowable values from 0.5 Hz to 8 Hz, in order to manage the client device's power consumption. Using a low message rate to conserve power on the client device comes at the expense of increased latency for the host to discover the client's beacon.

The ANT-FS beacon contains information about the capabilities and state of the client device. The ANT-FS layer that the client device is currently executing is provided in a status field of the Beacon message. The Beacon message also contains an ANT-FS Device Descriptor which uniquely identifies the client device.

Allowable beacon message rates and the beacon message structure are detailed in section 10.

3.4 ANT-FS Stack

ANT-FS is a specific application of ANT designed to allow seamless, automatic file transfer. The protocol is implemented on top of the ANT MCU, serial drivers and ANT interface; or embedded in the ANTMCU for parts supporting integrated ANT-FS (refer to data sheets for device capabilities). File protocols such as the Flexible and Interoperable Data Transfer (FIT) protocol may be used with ANT-FS to allow a greater level of interoperability amongst devices (Figure 3-4).

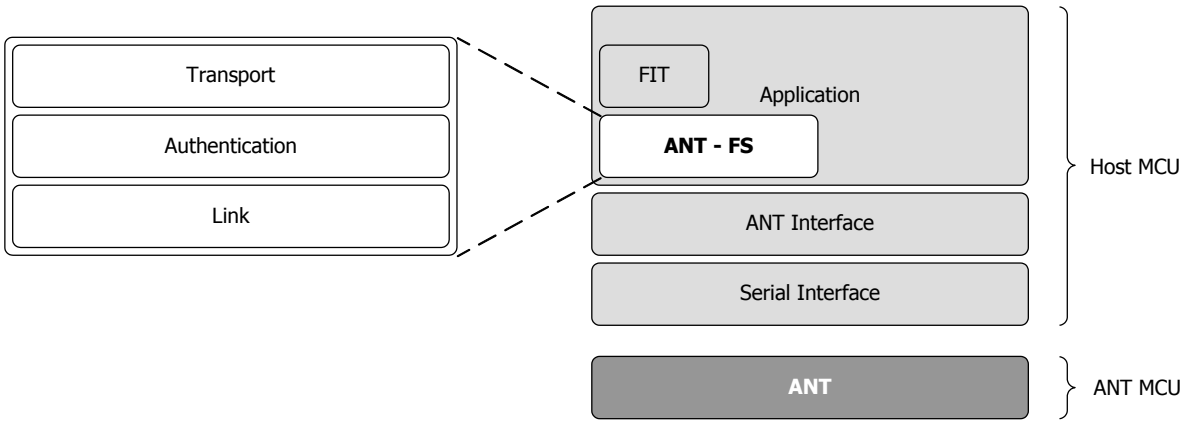

Figure 3-4. ANT-FS Architecture

ANT-FS specifies three basic layers of operation:

- Link Layer
- Authentication Layer
- Transport layer

A device may only be operating within a single layer at a time. The client and host may not necessarily be in the same layer at the same time; however, the idea is that the client and host progress through the layers in parallel, to accomplish a seamless and secure file transfer.

The unconnected Link layer is the default state of both client and host devices. In this state, the client device transmits its presence and capabilities via the beacon. The host device, on the other hand, searches for a device it may wish to connect to. The link layer is described in detail in section 4.2.

The Authentication layer is the state during which the host and client negotiate a trusted (i.e. authenticated) relationship. There are three types of authentication defined in ANT-FS:

- Pass-through: the client authenticates on host's request. The host does not require prior knowledge of the client device. Not secure. Automatic.
- Pairing Request: host requests pairing which the client may accept/reject. The host does not require prior knowledge of the client device. Secure. Not Automatic (generally requires a user interface).
- Pass Key: Host sends passkey for authentication. Host must have prior knowledge of the client's passkey. Secure and automatic.

Once authentication has been established, the client and host may progress to the Transport layer. If authentication fails, both will return to the Link layer. The authentication layer is described in detail in section 5.

Finally, the Transport layer defines the protocol to transfer user data from the client to the host device, or vice versa. Data and/or commands may be downloaded, uploaded, and/or erased in this state. Once data has been transferred,

both devices may return to the Link state until further communication is required. The transport layer is described in detail in section 6.

While in any state, both client and host will automatically return to the unconnected Link state after a specific time out, or after a specified number of attempts to perform a function.

4 Link Layer

The Link Layer is the default layer of the ANT-FS protocol and is responsible for establishing the initial ANT communication between a client and a host device such that authentication and data transport may occur.

The client device will initiate its Link layer by establishing a master ANT channel and transmitting the beacon signal. The beacon signal will contain information about the current state, capabilities and identification of the client device. The purpose of this beacon is to provide a potential host with enough information to decide whether or not to establish a connection. The beacon shall be sent once every channel period.

The initial message period, RF channel frequency and channel ID of the client device in the link layer are application dependent.

The host device will initiate its Link layer by initiating a search for an ANT-FS beacon. The host channel shall be configured as a slave channel, with parameters set according to the application (i.e. to match the client). Once the host device finds a matching client, it may check the beacon's content (such as device type, data availability, etc) and decide whether to establish a link.

After the host successfully links to the client device, the protocol then proceeds at the Authentication Layer.

For details on client and host devices' ANT channel parameters in the Link State, refer to section 10.

4.1 Link Layer State Machines

4.1.1 Client Device

The ANT-FS client link state machine is shown in Figure 4-1. The ANT-FS client device enters the Link state after the ANT master channel is initialized and opened (1). On entering the Link state (2), the client device will transmit the beacon every channel period (3), and wait for a host device to request a connection.

Once a host device has discovered the client's beacon and decided to establish a connection, the host will send a Link command to the client device. The host may use the Link command to specify a different channel period or RF frequency for subsequent interactions. The client device will reconfigure the beacon channel parameters as described in the Link command (4).

Once the client device has received a Link command and reconfigured its channel parameters, it will proceed to the Authentication (AUTH) layer. This state progression will be reflected in the beacon's status byte (5).

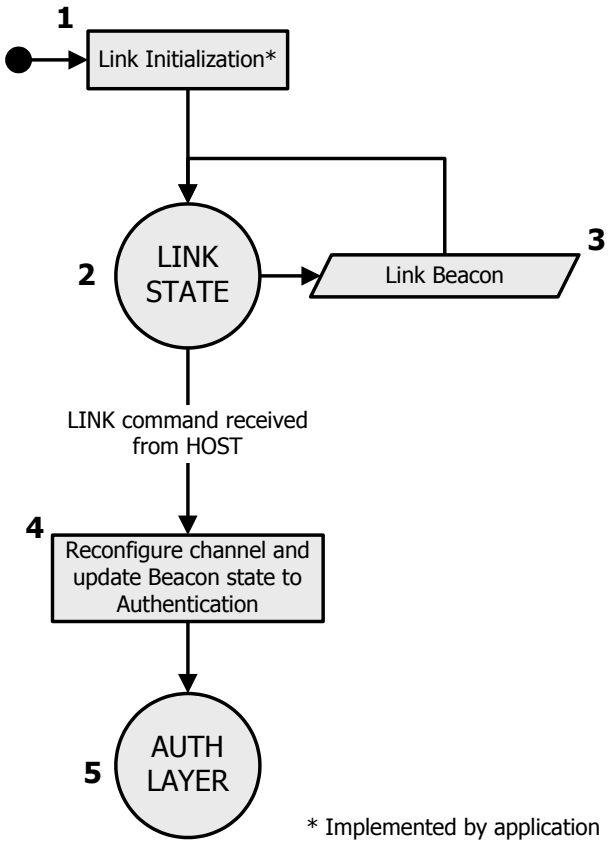

Figure 4-1. Client Device Link Layer

4.1.2 Host Device

The ANT-FS host link state machine is shown in Figure 4-2. The host enters the Link state after the ANT slave channel is initialized and opened (1). On entering the Link state, the host device will search for a client device configured as an ANT master and transmitting an ANT-FS beacon (2).

Upon acquiring a client device, the host will decide if the received message is a valid beacon (3). In checking the validity, the host shall also process the message to determine if a connection is desired. A host may choose to connect to a client for one of two reasons: either the client has data available for download, or the host wishes the potential to connect with any valid client.

If a connection is desired, the host device will make 1 or more attempts to send the Link command. The command is successful if the beacon state and channel configuration reflects the desired change (4-8); if successful, the host progresses to the authentication (AUTH) layer (9). If unsuccessful, the host will reset, remain in the unconnected link state and continue searching for valid client devices (1, 2).

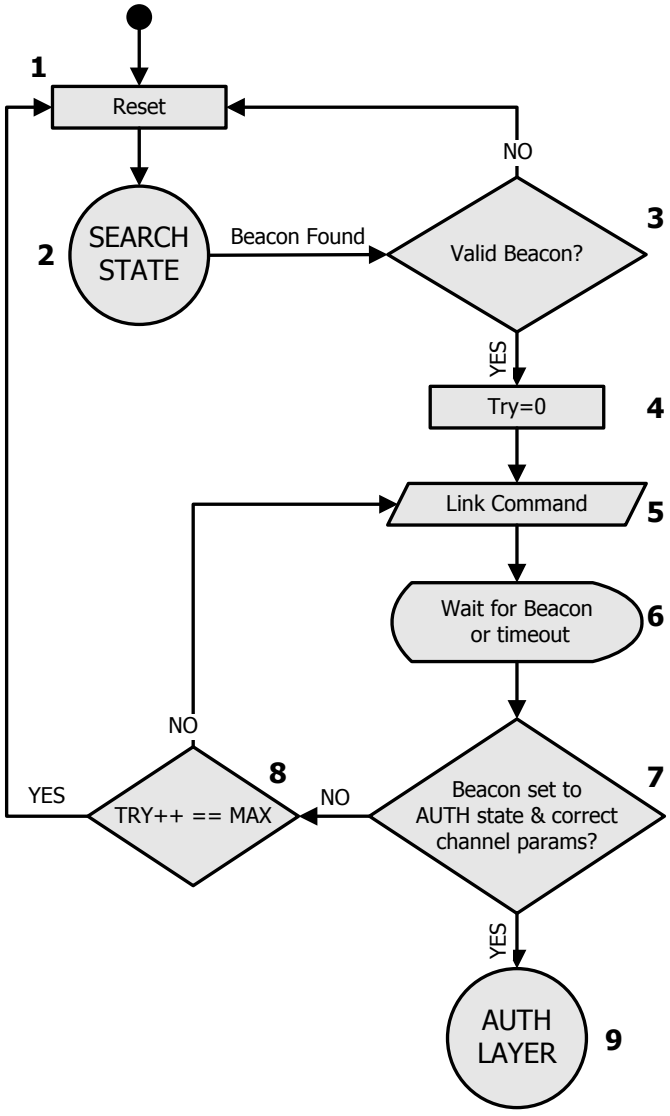

Figure 4-2. Host Device Link Layer

4.2 Link Layer Sequence Diagrams

The sequence of messages between a client and host device in the link layer is shown in Figure 4-3. An ANT-FS client device in the link state will transmit its beacon every channel period.

A host device in the link layer will search for a client device to connect to. On detecting a client beacon, the host must first decide whether or not it wishes to initiate communication. The received beacon must indicate that the client device is in a Link state before the host may attempt a connection. The host may also choose to connect based on other parameters available in the beacon message, such as the availability of data. Please refer to sections 11 and 12 for detailed descriptions of the ANT-FS beacon and Link command.

Once the host detects an appropriate beacon from a client device it wishes to communicate with, it will send the Link command. The Link command is send as an acknowledged message.

On receiving a Link command, the client device will update its channel parameters as specified in the link command fields, and progress to the Authentication layer. The client's beacon will change to reflect this change of state. Furthermore, to prevent race conditions of multiple hosts trying to connect to a single client, the client will echo the host's serial number (sent within the Link command) in the beacon.

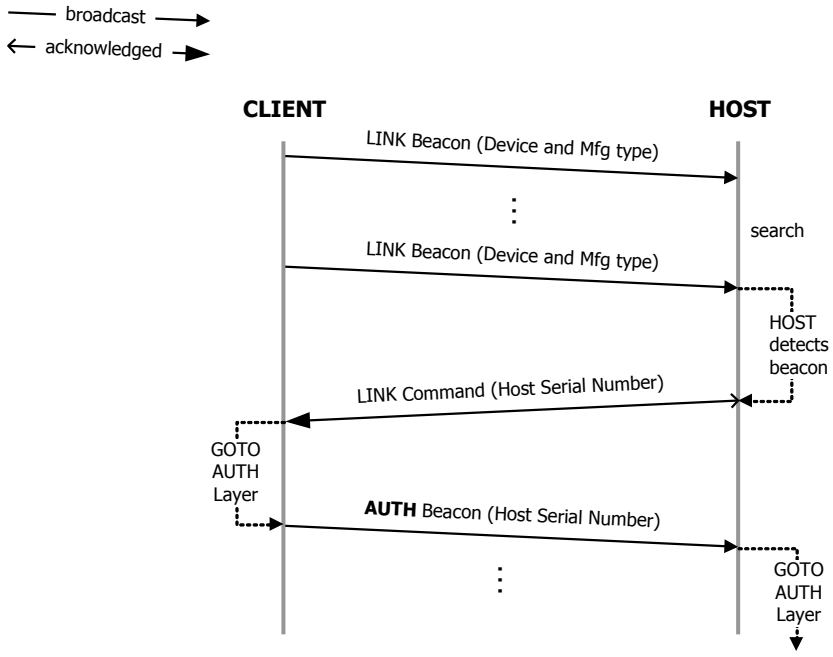

Figure 4-3. Link Layer Sequence Diagram

The host will only progress to the Authentication layer once the client's beacon has indicated that it is in the Authentication layer AND that the hosts own serial number is reflected in the beacon message. The host should not progress to the Authentication layer if it does not see its own serial number in the beacon, as this would mean that the client has actually connected to a different host device.

5 Authentication Layer

The Authentication Layer protocol defines a second level of device verification after the first level of beacon matching provided in the Link Layer. The Authentication Layer includes pass-through, pairing and passkey exchange authentication mechanisms.

The purpose of the Authentication layer is to establish a trusted relationship between the host and client devices. The particular authentication routine is specified by the application. If authentication is successful, the client and host will proceed to the Transport layer. If authentication is not successful, a failure will occur and the client and host will both return to the Link layer.

The beacon will indicate the client's supported authentication methods as shown in Table 5-1. The host shall only request a specific authentication method if it is supported by the client. For more details on the client beacon refer to section11.

Table 5-1. Client Beacon Authentication Type

Client Beacon Authentication Type	Pass-through	Passkey	Pairing
Pass-through	Yes	Application Specific	Application Specific
Passkey	No	Yes	Yes
Pairing	No	No	Yes

When pairing authentication is requested by the host, the client may accept or reject the request. If pairing is accepted, both the client and host devices proceed to the Transport layer. Note also that if pairing is accepted, the client may send a passkey to the host for future authentication purposes. If pairing is rejected, both devices move back to the unconnected Link layer.

If the client device has already been paired with the host, a simple passkey exchange authentication method is used. The host simply sends the passkey to the client device, and if the passkey is correct, authentication is successful. If the passkey is incorrect and authentication fails, both devices return to the unconnected Link layer.

It is also possible to implement a multi-stage authentication routine; for example, using a serial number as an interim step to decide whether passkey or pairing should authenticate the connection (Figure 5-1). This requires the host device to first request a serial number from the client device, using the Authenticate (AUTH) command. The client device responds by sending its own serial number in the AUTH response message. The client may also include a friendly name with the serial number response.

Upon receiving the serial number, the host may decide to use pairing or a passkey exchange for authenticating with the client. Generally, a client should only need to pair once with a host, in their first communication session. The host should then remember the client's serial number and passkey for future connections.

The following sections describe the Authentication layer and command sequences. For details of the ANT-FS command and responses, refer to sections 11 and 12.

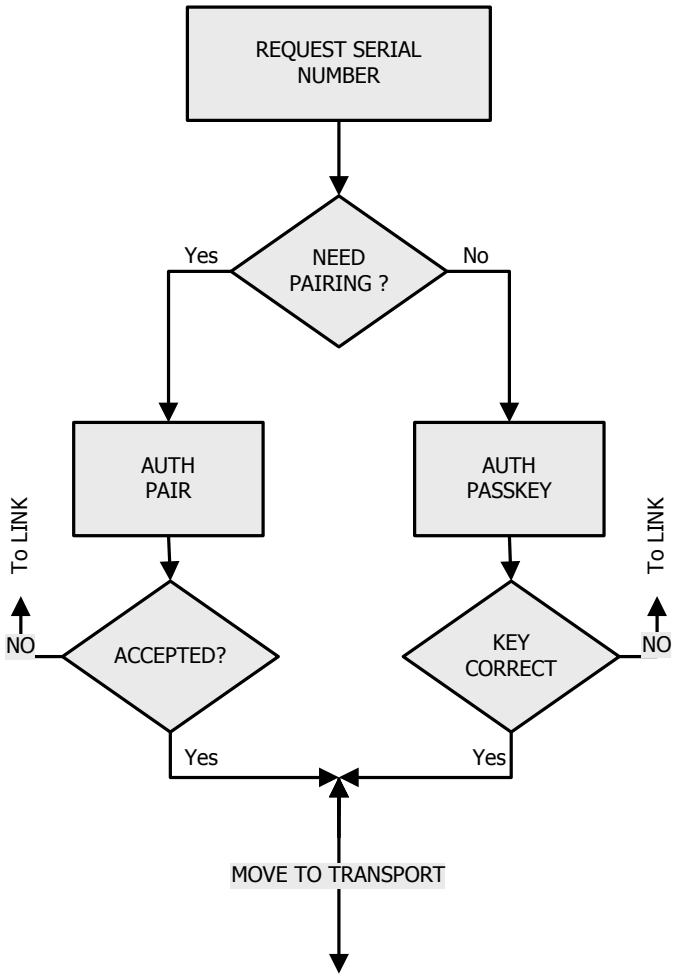

Figure 5-1. Multi-Stage Authentication

5.1 Authentication Layer State Machines

5.1.1 Client Device

An ANT-FS client device's authentication layer state machine is shown in Figure 5-2. On entering the Authentication Layer, the client device goes to the default authentication state (1). In this state, the client broadcasts a beacon indicating that the client device is in the Authentication State (2) and waits for an authentication (AUTH) command form the host. Once an Authentication command is received (3), the client will proceed to handling the authentication process (4). This process is specific to the application, and is described further in section 5.1.1.1 and 5.2.

If the authentication process is successful, the client will update its beacon (5) and proceed to the Transport layer (6). If the authentication process fails, the client will revert back to the Link Layer (7). If the authentication process has multiple stages, the state machine may process the command and return to the default authentication state (1) to wait for further authentication commands from the host.

If no commands are received from the host within a set period of time (defined by the application), the client device will timeout and automatically return to the Link Layer (7).

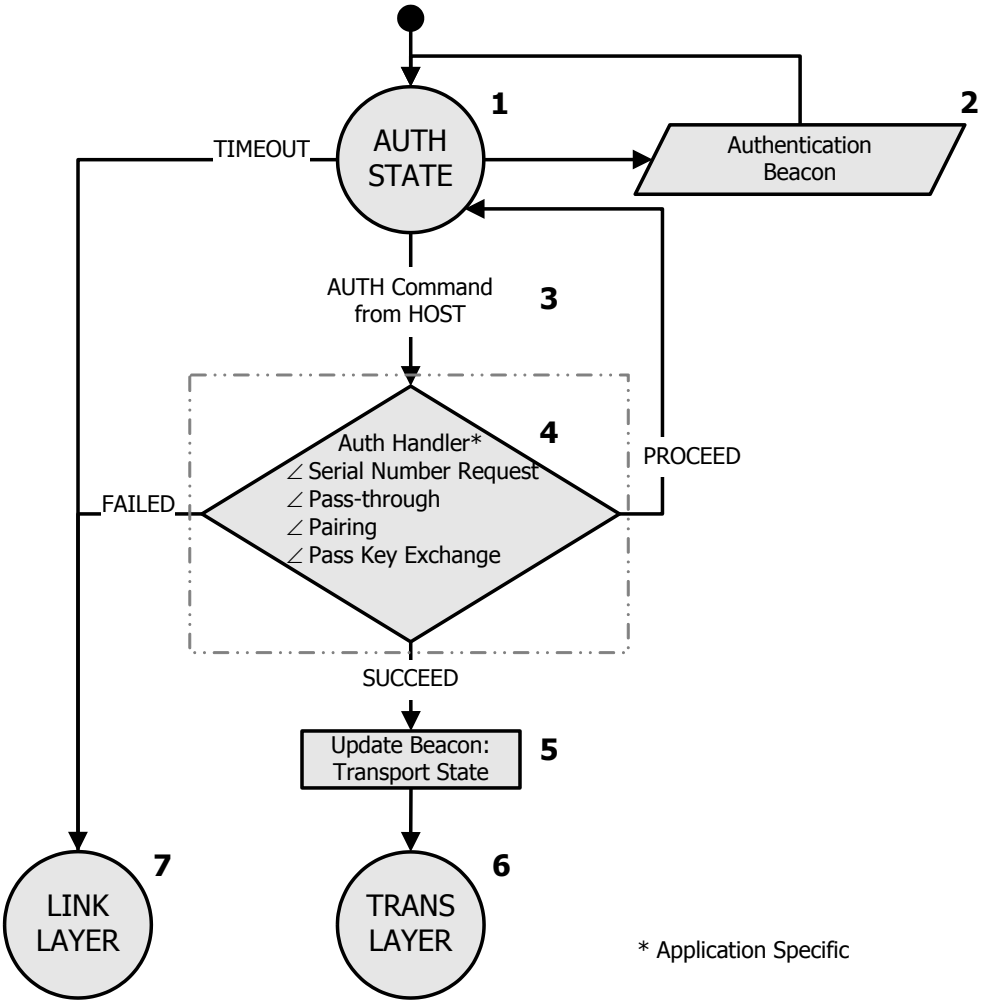

Figure 5-2. Client Device Authentication Layer

5.1.1.1 Authentication Handler

The authentication handler for the ANT-FS client device may implement all, one, or some of the authentication methods described in the following sections.

In all cases, the client's authentication handler should perform a simple check of the serial number (and optional friendly name) received in the host's authentication command. This is to ensure that the command received was actually sent by the host that requested the link. If the serial number matches, the authentication handler shall proceed with the authentication; otherwise, if the host serial number does not match, the authentication handler will return a failure to the state machine and the client device will return to the Link Layer

5.1.1.1.1 Pass-through

The simplest authentication method is the pass-through approach. Once the authentication handler has determined that an Authentication command has been received, and that the requested authentication method is pass-through, the handler will process the request as illustrated in Figure 5-3.

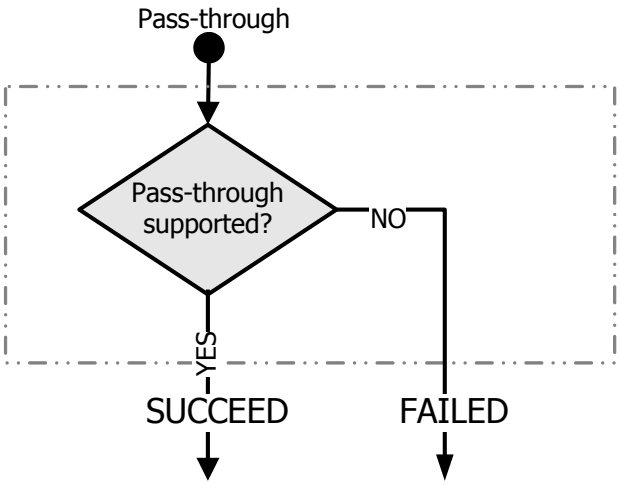

Figure 5-3. Client Device Authentication Layer – Pass-through

Pass-through offers the least security as, provided the client supports pass-through, the state machine will immediately proceed to the Transport Layer on receiving the request from the host.

5.1.1.1.2 Serial Number Request

While the serial number request command is not an actual authentication method in itself, it can be used in a multi-stage authentication process and, as such, **shall be supported by all ANT-FS client devices**. When the authentication handler has determined that a serial number request has been received, it shall process the request as illustrated in Figure 5-4.

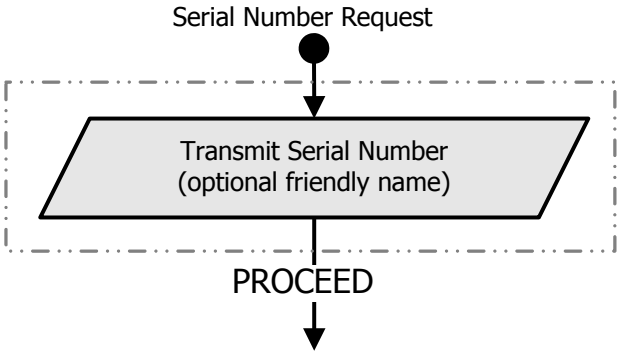

Figure 5-4. Client Device Authentication Layer – Serial Number Request

The handler simply transmits the client's serial number along with the next beacon, and remains in the authentication state until a further host command is received.

Note, if desired, an optional friendly name may be sent along with the client's serial number.

5.1.1.1.3 Pairing

A more sophisticated authentication mechanism involves pairing. This requires that a client/user either accept or refuse a pairing request from the host. Once a valid pairing request has been received, the handler will check if pairing is actually enabled on the client device. If the client has not enabled pairing, authentication will fail and the state machine will return to the default authentication state.

The state machine will update the beacon to "busy" mode and go to the UI pairing state. During this state the handler is waiting for a response from the user or the client application layer. The user/application may reject, accept or ignore the pairing request. If the user accepts the pairing request, the handler will return a success to the authentication state machine and the client will progress to the Transport Layer. If the user/application rejects the pairing request, then authentication has failed and the client device will return to the unconnected Link Layer. If the user ignores the pairing request, then the handler will timeout, authentication will fail and the client device will go back to the unconnected Link Layer.

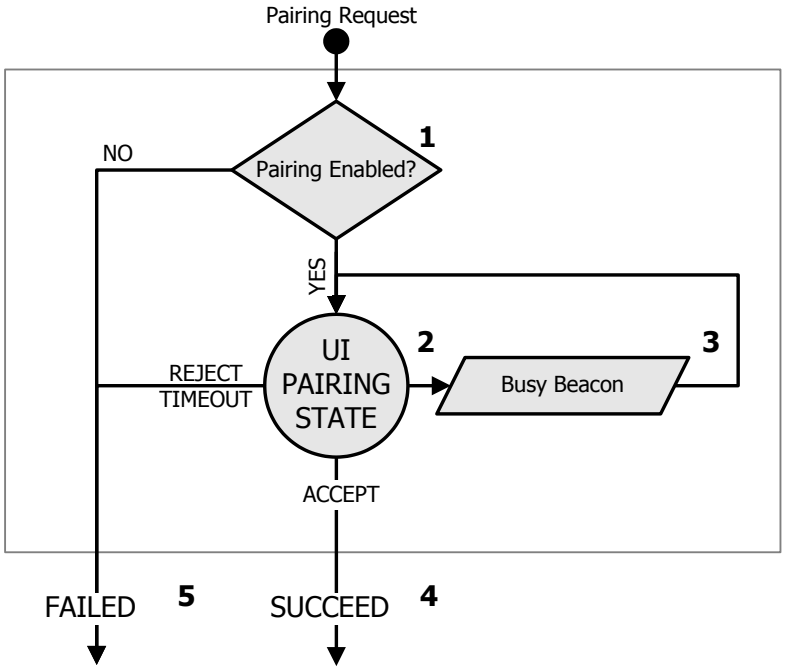

Figure 5-5. Client Device Authentication Layer – Pairing

5.1.1.1.4 Passkey

Passkey requires that a host device have knowledge of the client device (i.e. it must know the passkey for that specific client device). Once a valid passkey request has been received, the handler will check if passkey is actually enabled on the client device. If the client has not enabled passkey, authentication will fail and the state machine will return to the default unconnected link state. If passkey is enabled on the client, the state machine will provide the Application Layer with the passkey it received from the host (the passkey is included along with the Authentication command in a burst transfer).

If the passkey received from the host, matches the passkey of the client, the handler will return a success to the authentication state machine and the client will progress to the Transport Layer. If the passkeys do not match, then authentication has failed and the client device will return to the unconnected Link Layer.

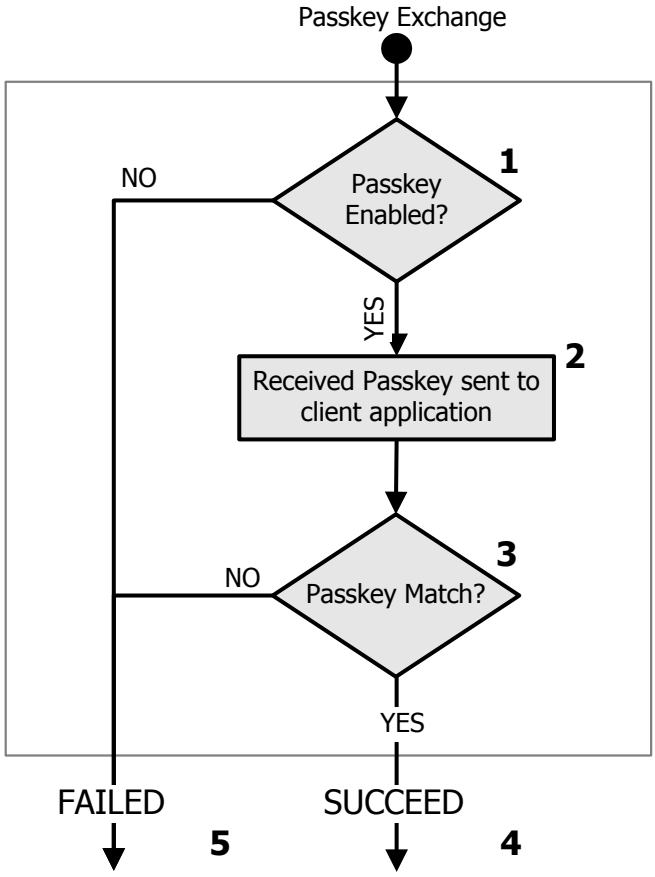

Figure 5-6. Client Device Authentication Layer – Passkey

5.1.2 Host Device

An ANT-FS host device's authentication layer state machine is shown in Figure 5-7. Upon entering the Authentication Layer, the host will attempt to execute an application specific authentication handler (2). The authentication handler will be executed until:

- The received beacon indicates that the client device has progressed to the transport state (3), or
- the maximum number of re-tries has been exhausted (5), or
- authentication fails.

If authentication is successful and the beacon changes to Transport State (3), the state machine will continue to the Transport (4) layer; otherwise, authentication fails and the host will return to the Link Layer (6).

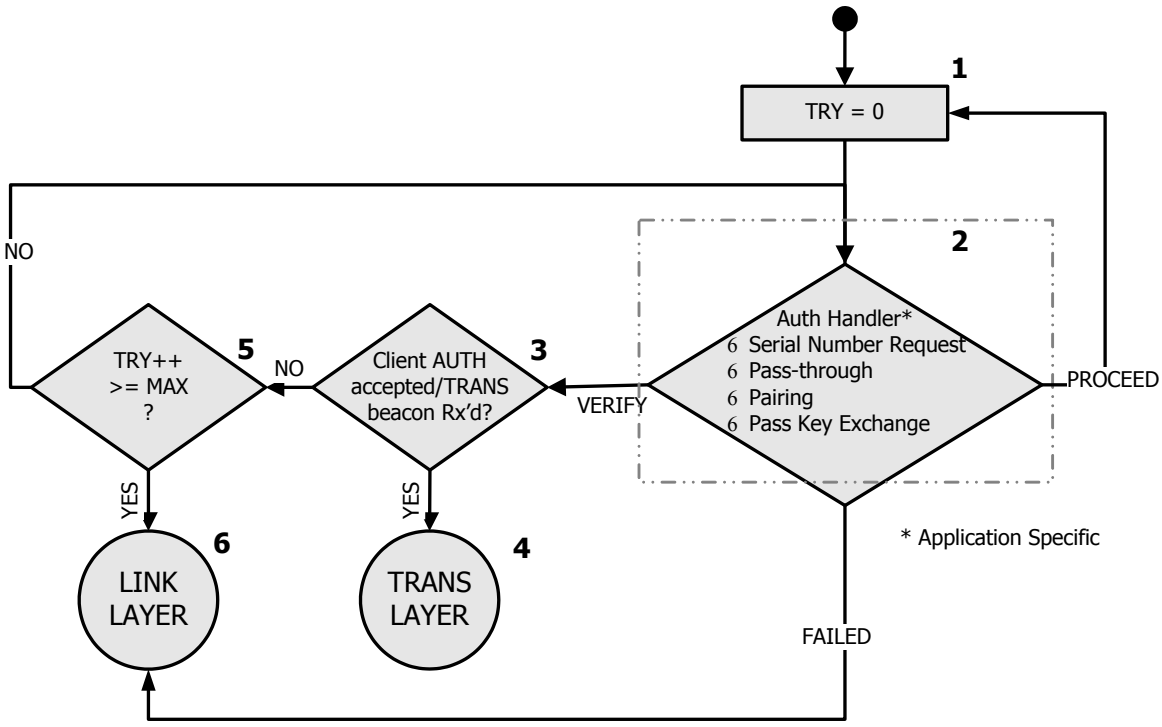

Figure 5-7. Host Device Authentication Layer

5.1.2.1 Authentication Handler

The authentication handler for a host device is shown in Figure 5-8. The handler will first check and verify that the beacon is in the Authentication state and what authentication types are supported by the client. If the client is not in the authentication state, authentication will fail and the state machine will return the host back to the Link Layer.

The host may first request a serial number from the client device, receive the proper serial number and subsequently proceed with further authentication. If a serial number is not require (or has already been received) the handler may initiate pass-through, pairing or passkey authentication with the client device and wait for a response indicating if the authentication request was accepted or rejected.

If the client accepted the authentication request, the handler will return a success and the host may then check the beacon has changed to the Transport state, and may itself progress into Transport Layer. If authentication was rejected, or if the beacon does not change to the Transport state, the host will retry authentication until the maximum number of tries has been exhausted. If the client beacon changes to the Link state, authentication will fail and the host will revert back to the Link Layer.

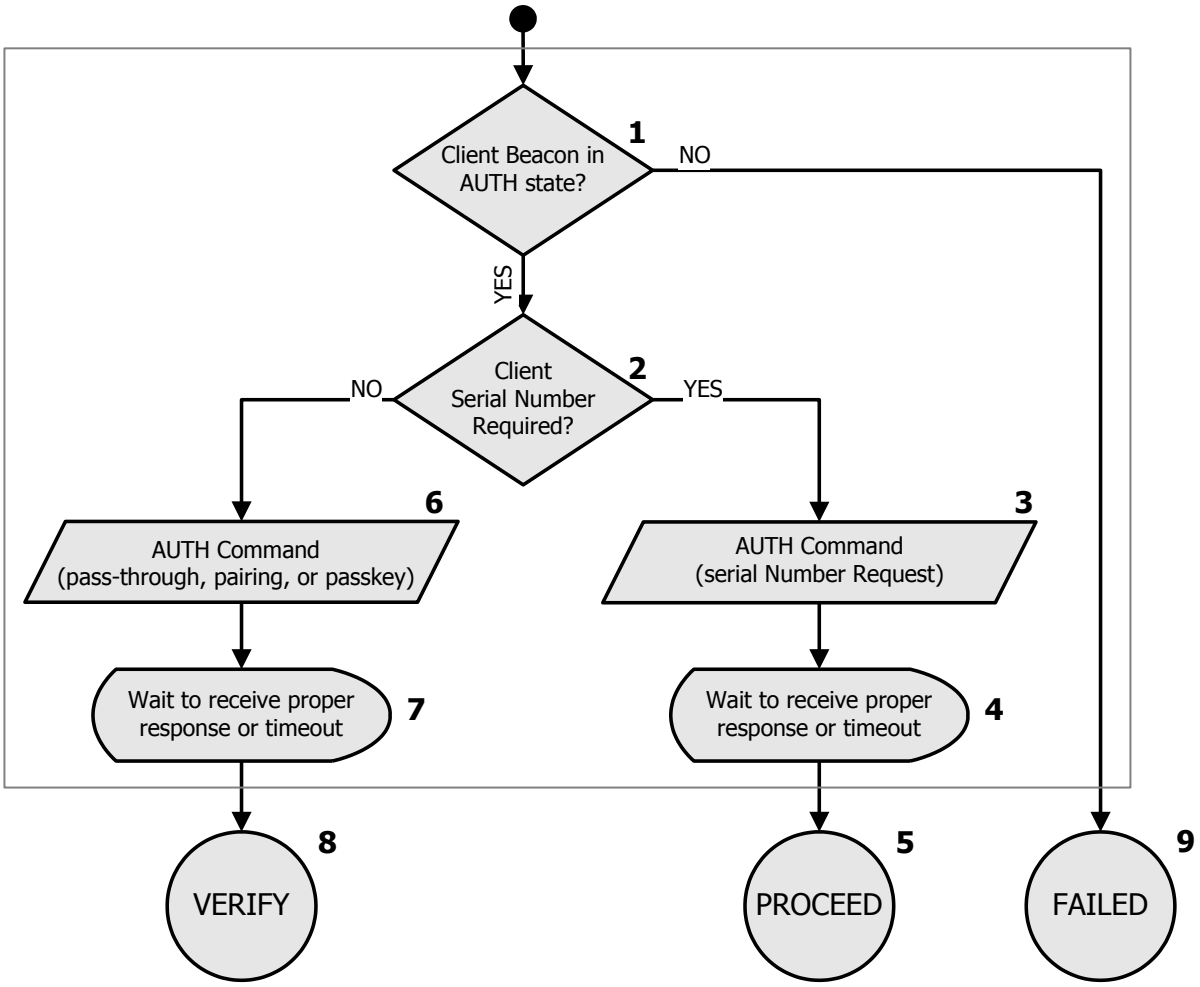

Figure 5-8. Host Device Authentication Layer Handler

5.2 Authentication Layer Sequence Diagrams

Sequence diagrams of the messages between a client and host device in the authentication layer are provided in the following sections.

5.2.1 Pass-through

Pass-through authentication effectively offers a method of by-passing authentication without breaking the overall architecture of ANT-FS. When using this method, the host device requests to be authenticated by the client. It does not provide a passkey, or require any user interaction. If the client accepts this request, authentication passes and both devices move to the Transport layer.

This is the simplest, and least secure, method of authenticating a client device. It may be appropriate for some use cases where authentication is not strictly necessary, or can be achieved by other means (such as proximity), and a truly seamless method of moving to the transport layer is required.

As with other methods of authentication, pass-through is initiated by the host sending an AUTH command. If the client supports this method of authentication, the client device will respond by bursting the beacon and AUTH "accept" response. The client will then progress to the Transport layer, and this state change will be reflected in the beacon. This sequence of commands is illustrated in Figure 5-9.

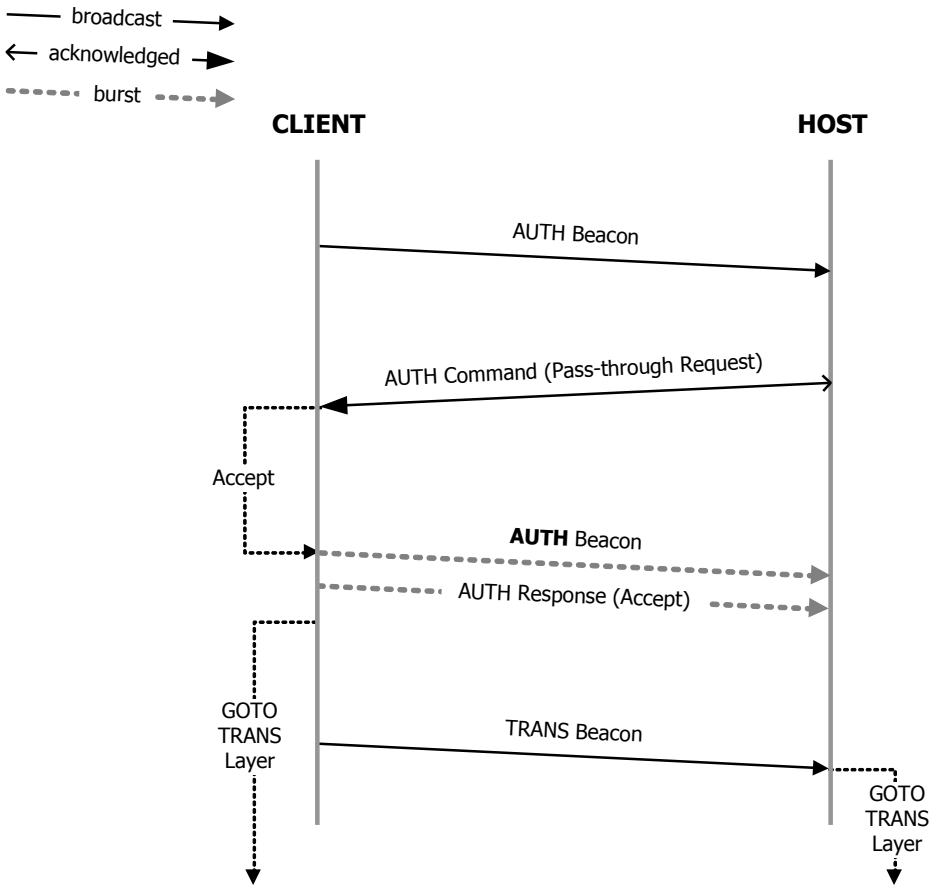

Figure 5-9. Pass-through Authentication (accept) Message Flow

If pass-through is not supported, an AUTH "reject" response is sent analogous to the "accept" case; however, the client device will immediately return to the Link layer rather than progressing to the Transport layer.

5.2.2 Serial Number Request

The host may request a serial number from the client device. Based on the serial number, the host can determine whether or not the client has already been paired. **An ANT-FS client device shall always support a serial number request from a host.** The client shall respond with a burst transfer of the beacon and AUTH response containing the serial number. A serial number request should never allow the client or host to progress to the Transport layer.

The sequence of messages for a serial number request is illustrated in Figure 5-10.

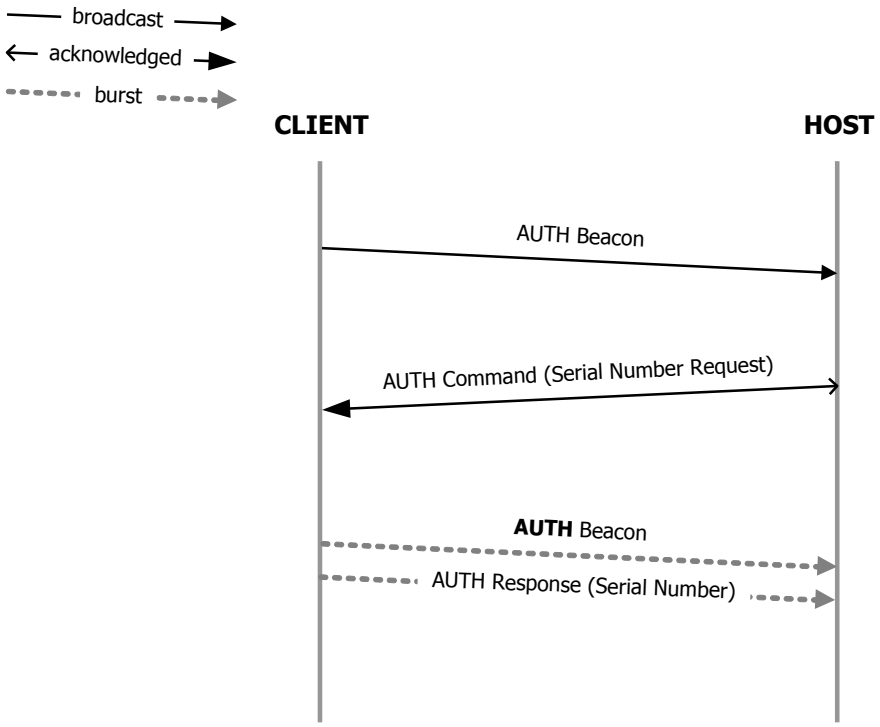

Figure 5-10. Serial Number Request Message Flow

The host may compare the received serial number to a stored database of clients to which it has already been paired. If the received serial number is not in the database, meaning the host and client have not been previously paired, the host may request pairing. If the client's serial number is already in the database, meaning the host and client have previously paired, the host may proceed to authenticate using a passkey exchange.

5.2.3 Pairing

Pairing requires that the client device accept a request from the host to connect and authenticate. Typically, this would involve user input. In the case of a watch, for example, a message might be displayed indicating a pairing attempt by the host. The user may then accept or reject this request. If the pairing request is accepted, the client device will send an AUTH response (along with the beacon) to the host indicating its acceptance. The AUTH response will also include a passkey that may be stored by the host for future authentication purposes. If the client device rejects the pairing request, it will send an AUTH response indicating the request's rejection and no passkey will be exchanged. Due to the nature of the pairing process, it is important that timeout functionality be built into the client device to account for inaction by the host or user. If a timeout occurs, the equivalent of a rejection response is sent to the host.

A host may also optionally provide a friendly 'name' to the client device. This would be burst along with the AUTH command during the pairing request process, and would be useful in helping the user identify the device. A sequence of commands and responses for the pairing process is illustrated in Figure 5-11.

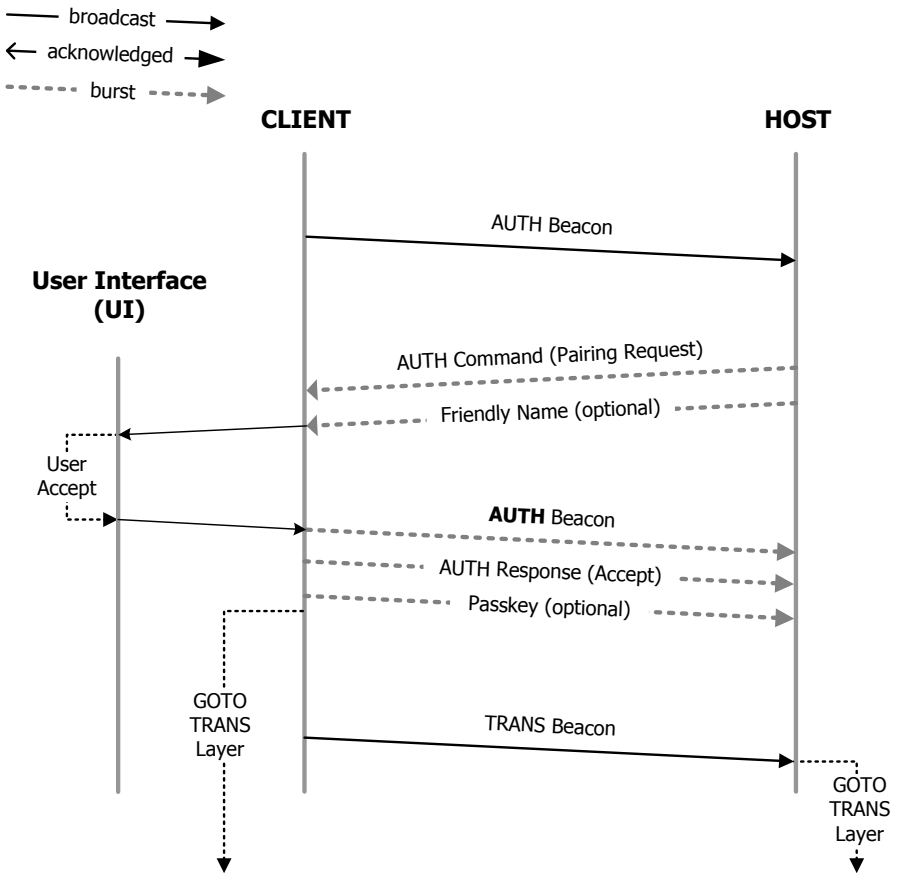

Figure 5-11. Pairing Request (accept) Message Flow

Once the client has accepted the pairing request, its beacon will change to the Transport state and both devices may proceed to the Transport layer. If pairing is rejected, or fails, both devices should return to the unconnected Link layer.

5.2.4 Passkey

Passkey authentication requires the host to send an appropriate passkey to the client device. The client compares the received passkey with its own and, if the passkeys match, authentication is accepted, and both devices progress to the Transport layer. If the passkeys do not match, authentication is rejected and the client device shall return to the Link layer.

The passkey may, or may not, be passed to the host device during the pairing process (in the AUTH response) depending on the specific application. Passkey need not necessarily be used in conjunction with the pairing method.

The host initiates Passkey authentication by bursting the AUTH command, which shall include the passkey, to the client device. On receiving the command and passkey, the client shall determine whether to accept or reject the request. User action is not generally required. If the passkey is accepted, the client device will respond by bursting the AUTH "accept" response along with the beacon. The sequence of commands for Passkey authentication is illustrated in Figure 5-12.

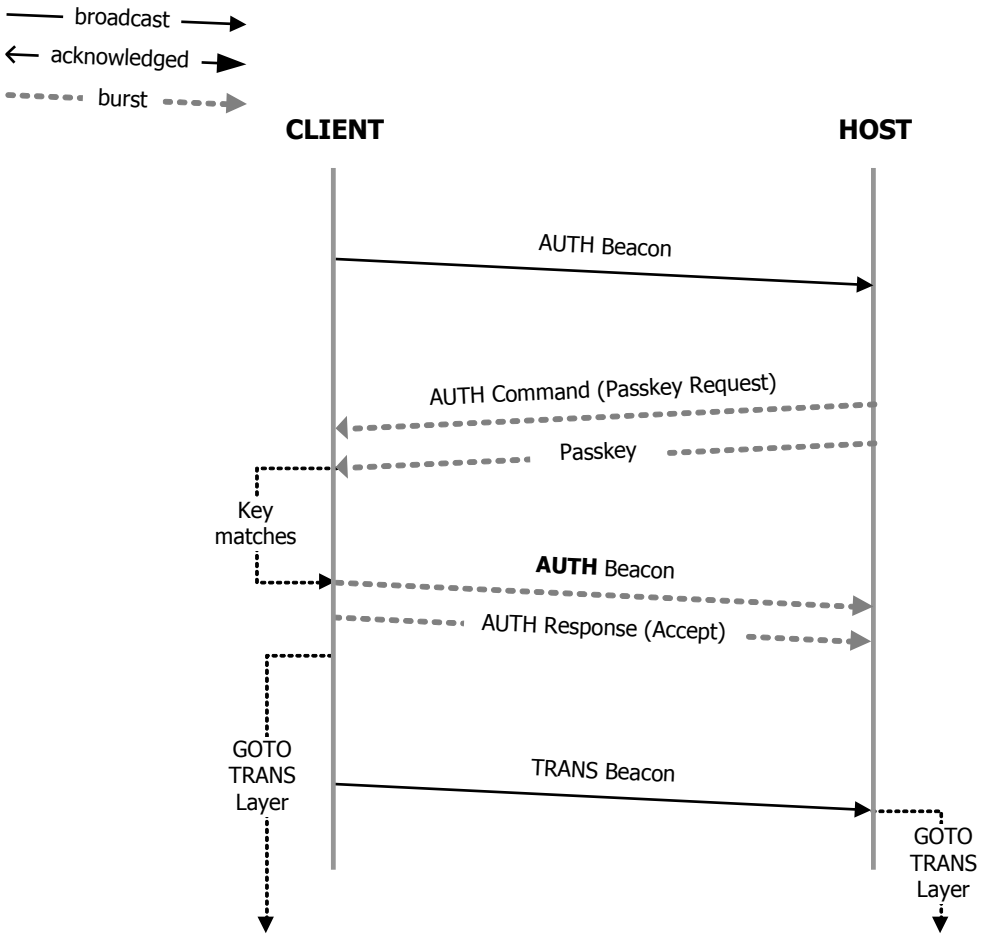

Figure 5-12. Passkey Authentication (accept) Message Flow

If the passkey does not match, an AUTH "reject" response is sent; however, the client device will immediately return to the Link layer instead of proceeding to the Transport layer.

6 Transport Layer

The Transport layer supports a number of operations all related to the data content of the client device. In the Transport layer, the host may initiate download or upload of data, erase data, and query the client for data content. The host may also use the command pipe upload commands to the client, and download command responses while in the transport layer. Refer to section 13.2 for details on the command pipe.

The ANT-FS client must maintain a directory specifying the data content currently stored on the device. This directory is treated as a special file located at file index 0, and can be requested using the download command as for any other file. The directory contains important information regarding the files stored on the client device. Refer to section 13.1 for details on the format of the directory table.

6.1 Transport Layer State Machine

6.1.1 Client Device

An ANT-FS client device's transport layer state machine is shown in Figure 6-1. By default, the client will continuously transmit its beacon message indicating that it is in the transport state (1&2). Any command received from the host (3) is processed by the transport layer handler (4), which will return the client to the default transport state (1), unless the command failed and the client will return to the unconnected link state (5).

The Transport Layer timeout is reset after every received command from the host (3), but if a message is not received within the timeout period the client device will return to the Link state (5). The Transport layer timeout value is set by the application.

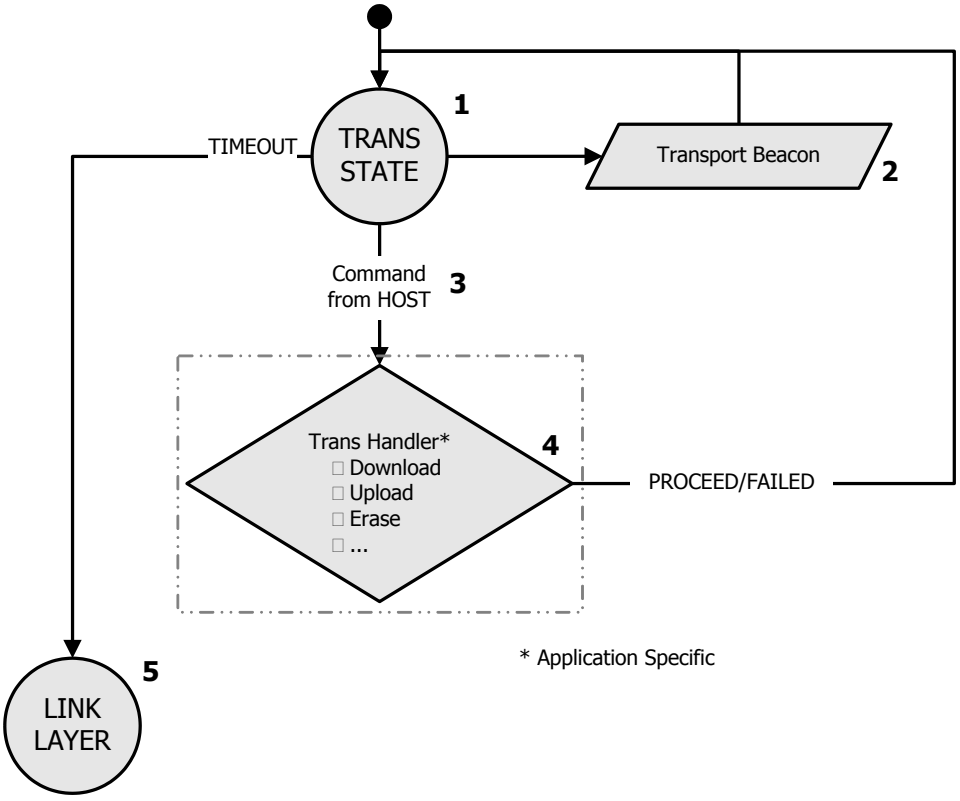

Figure 6-1. Client Device Transport Layer

The client device's command handler manages the commands that may be sent by the host and performs the required response. The state machine of the handler is not provided in this document; however, details of the command/response messages are provided in section 12 .

6.1.2 Host Device

An ANT-FS host device's transport layer state machine is shown in Figure 6-2. The host application controls which commands shall be sent to the client device. ANT-FS defines a small number of commands designed to create a flexible communication environment. Communicating with the client is customizable and extendable by using the Data Format Type field to create custom commands for any given device.

All host commands are sent to the client device as acknowledged or burst messages. If the host does not receive acknowledgement, it must repeat the command. If no beacon message or other response is received from the client within a specified timeout period then the host device will return to the Link state. Any received message from the client device will reset the timeout.

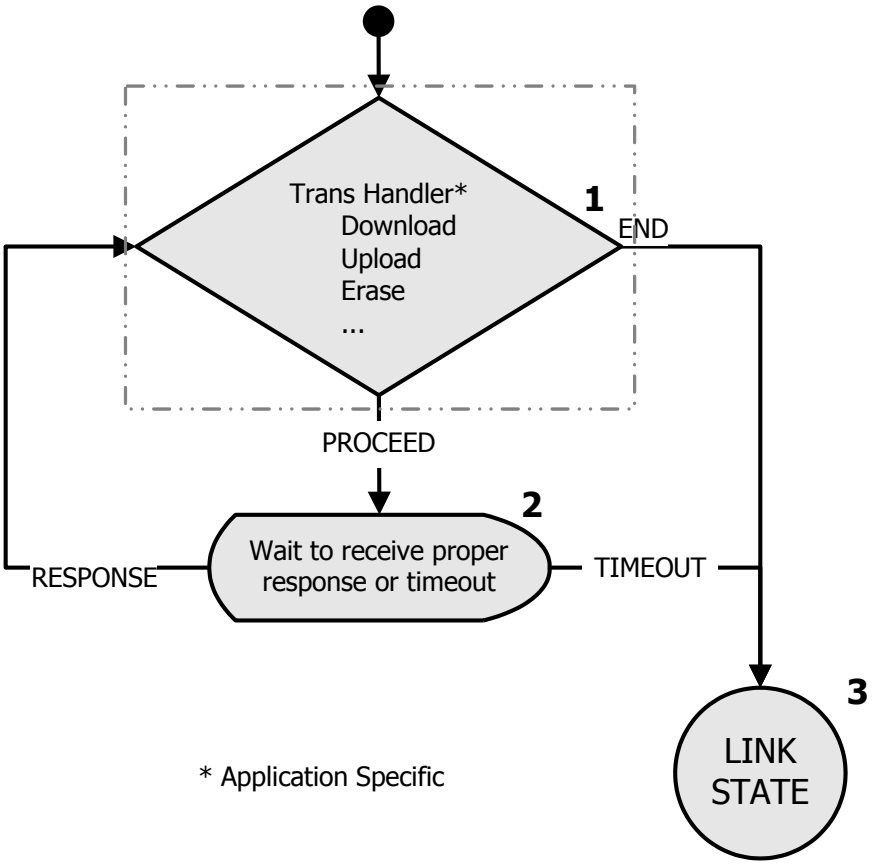

Figure 6-2. Host Device Transport Layer

6.2 Transport Layer Sequence Diagrams

The sequence diagrams of the message flow between a host and client device when uploading or downloading data is provided in the following sections.

6.2.1 Downloading

A host device shall first query the client directory structure in order to determine what data is available for download. If the host determines that data is available while in the Transport layer, it may attempt to download the data.

Whether downloading the client's directory structure, or downloading a data file, the download process is the same. Figure 6-3 shows the series of messages between the host and client for a download to occur. The host will burst the two packet Download command. The client device will reply with a burst Download response which includes the beacon and requested data.

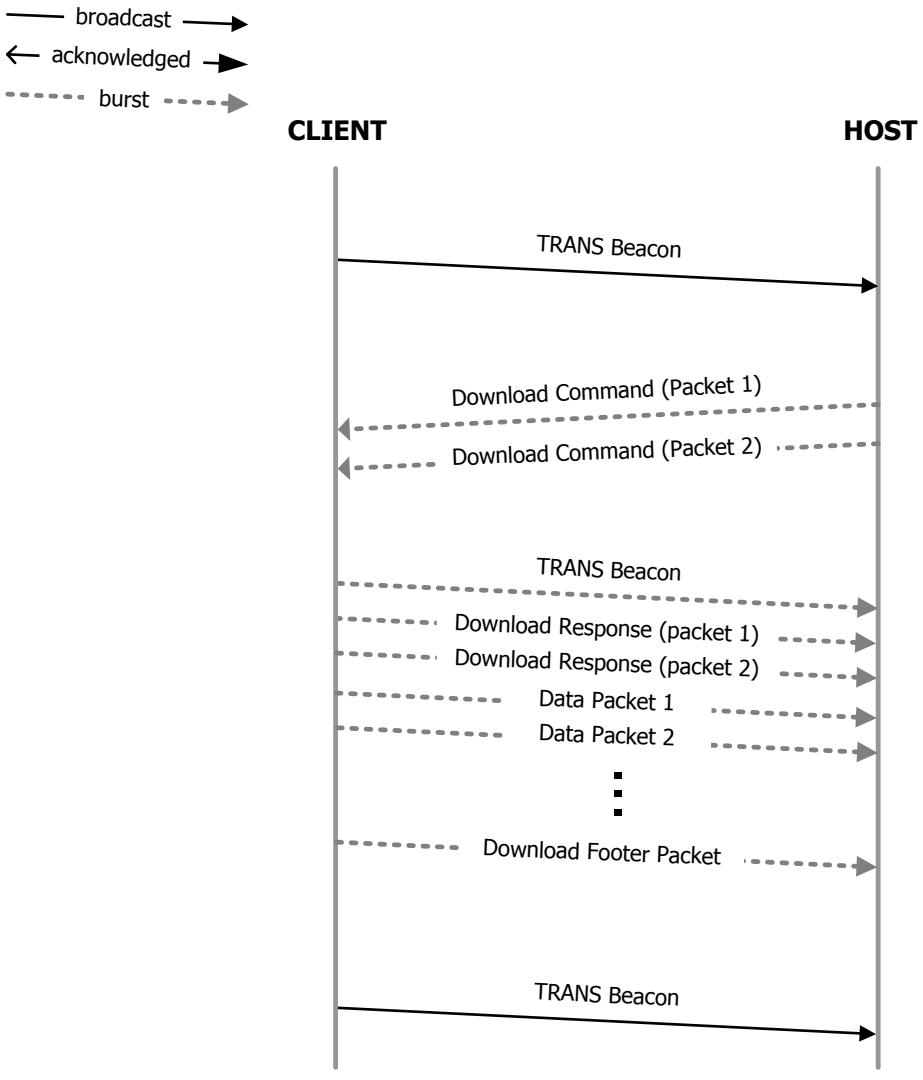

Figure 6-3. Download Message Flow

The Download command has several fields which can be set to control the downloading process. Specifically, the download command has been designed to allow downloads to be re-started from any section of a file. This is important

in case a download fails for any reason and needs to be restarted. Instead of having to re-download an entire file, the host can choose to download from where it last received data.

To retry a failed download, the host shall request a Download in the same way it did before ; however, it will set the Data Offset field according to the last successfully received packet and the value of the CRC at that point. The client will then burst data to the host from that point in the file, starting with that CRC value.

For details on the download command and responses refer to section 12.

6.2.2 Uploading

The Upload process involves sending bulk file data from the host to client device. Before uploading data, the host should first download the directory from the client to determine the supported file types.

Figure 6-4 shows the message flow between a client and host during an upload. Prior to starting the upload, the host will first request the upload to ensure the client is capable of receiving data. The client may be limited by the data type it can receive, storage space or the amount of data it can receive in a single burst. Once the client accepts the upload request from the host, it will reply with an Upload response; the host will then burst data packets to the client, starting with a single packet header and ending with a single packet footer. Once the upload is complete, the client will send a response back to the host indicating the success of the upload. For details on the upload command and responses refer to section 12.

The ANT-FS Upload mechanism is designed to allow easy recovery from a failed upload attempt. This means the host may resume the file transfer from the point at which the file transfer failed, without having to re-transmit the entire file. This requires that the client monitors the last successful data offset as well as the value of the CRC at that point.

To retry a failed upload, the host shall request an Upload in the same way it did before ; however, it will set the Data Offset field of the UPLOAD REQUEST command to MAX_ULONG (i.e. 0xFFFFFFFF). The client will respond with the value of the last good data offset and the value of the CRC at that point. The host will then burst data to the client from that point in the file, starting with that CRC value.

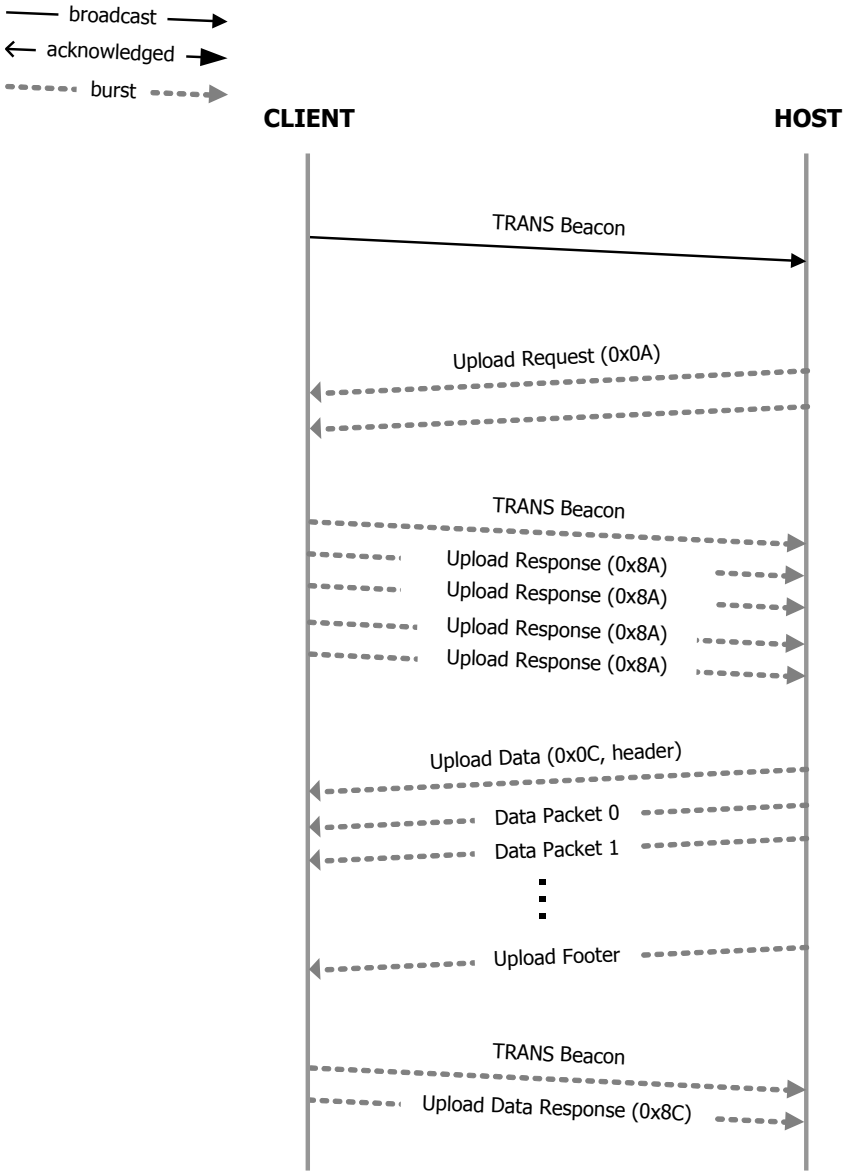

Figure 6-4. Upload Message Flow

6.2.3 Erasing

Erasing data is accomplished by sending the ERASE command to the client device, specifying the Data File Index of the file to be erased. Upon receiving the ERASE command, the client should proceed to erase the data and respond with an ERASE response. It is recommended that the host first download the directory structure of the client prior to erasing a file.

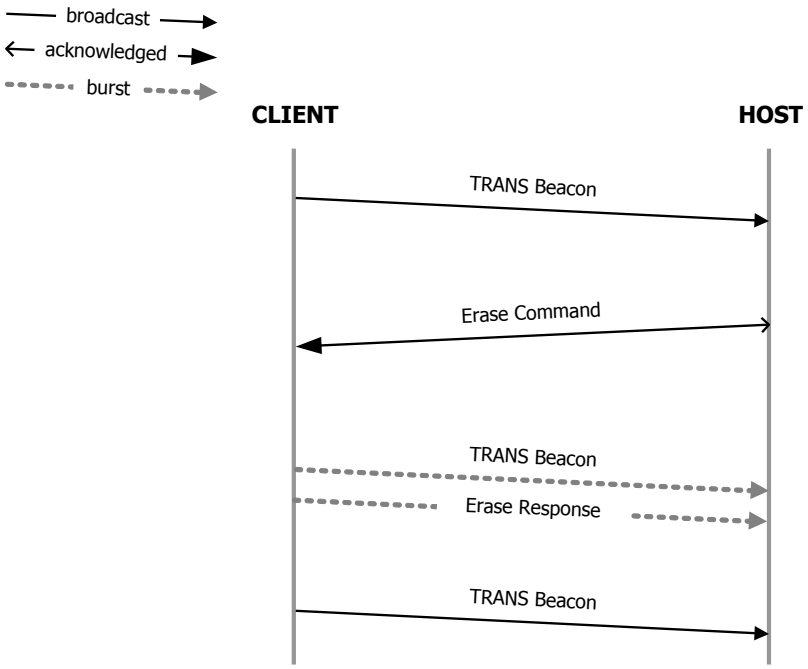

Figure 6-5. Upload Message Flow

7 Cross Layer Commands

The following ANT-FS commands may be issued within multiple ANT-FS layers.

7.1 Ping

To prevent the client from timing out, the host may send a Ping command. The Ping command simply resets the connection timer on the client. There is no response to the PING command.

The ping command may be sent in either authentication or transport layers.

7.2 Link

The Link command may also be sent to a client to change the operational RF channel frequency and/or channel period. The response shall be reflected in the ANT-FS beacon.

The link command may be sent in any ANT-FS layer.

7.3 Disconnect

The Disconnect command may be sent by a host device at any time, within any layer. When a client device receives the disconnect command it shall return immediately to its default behaviour in the link layer. This means that the client shall return to it's link beacon state, RF channel frequency and channel period.

The only exception is in the case of Broadcast ANT-FS. When a broadcast ANT-FS session is in effect, the client shall return to its default broadcast state. Refer to section 8 for details on the broadcast ANT-FS use case.

Additionally, the host may request the client return to the requested link/broadcast state after a specified duration of being undiscoverable (refer to section 12.4.2 for details). There are two ways a client can become undiscoverable:

- Set the pairing bit in the device's channel ID (i.e. most significant bit of the device type)
- Close the channel

This undiscoverable mode is useful in use cases involving multiple client devices connecting to a single host (i.e. ANT-FS access point). This allows the host to connect to a client device, down load the data and place the client into undiscoverable mode, preventing an immediate connection to the same client when searching for other client devices.

Note, there is no response to the DISCONNECT command, the client simply returns to its default link or broadcast state.

8 Broadcast ANT-FS

Broadcast ANT-FS provides a mechanism for ANT-FS to occur during a standard ANT broadcast session, as opposed to initiating a continuously beaconing link state. This will allow an ANT-FS session to occur on a device that is already broadcasting on an established channel, without having to open a second channel. In this case, the device that is acting as the channel master will become the client, and the slave device will be the host.

As with all ANT-FS sessions, the host device initiates all ANT-FS commands, and as such, the slave device in the established broadcast channel shall request the ANT-FS session. The broadcast ANT-FS session will operate as shown below in Figure 8-1.

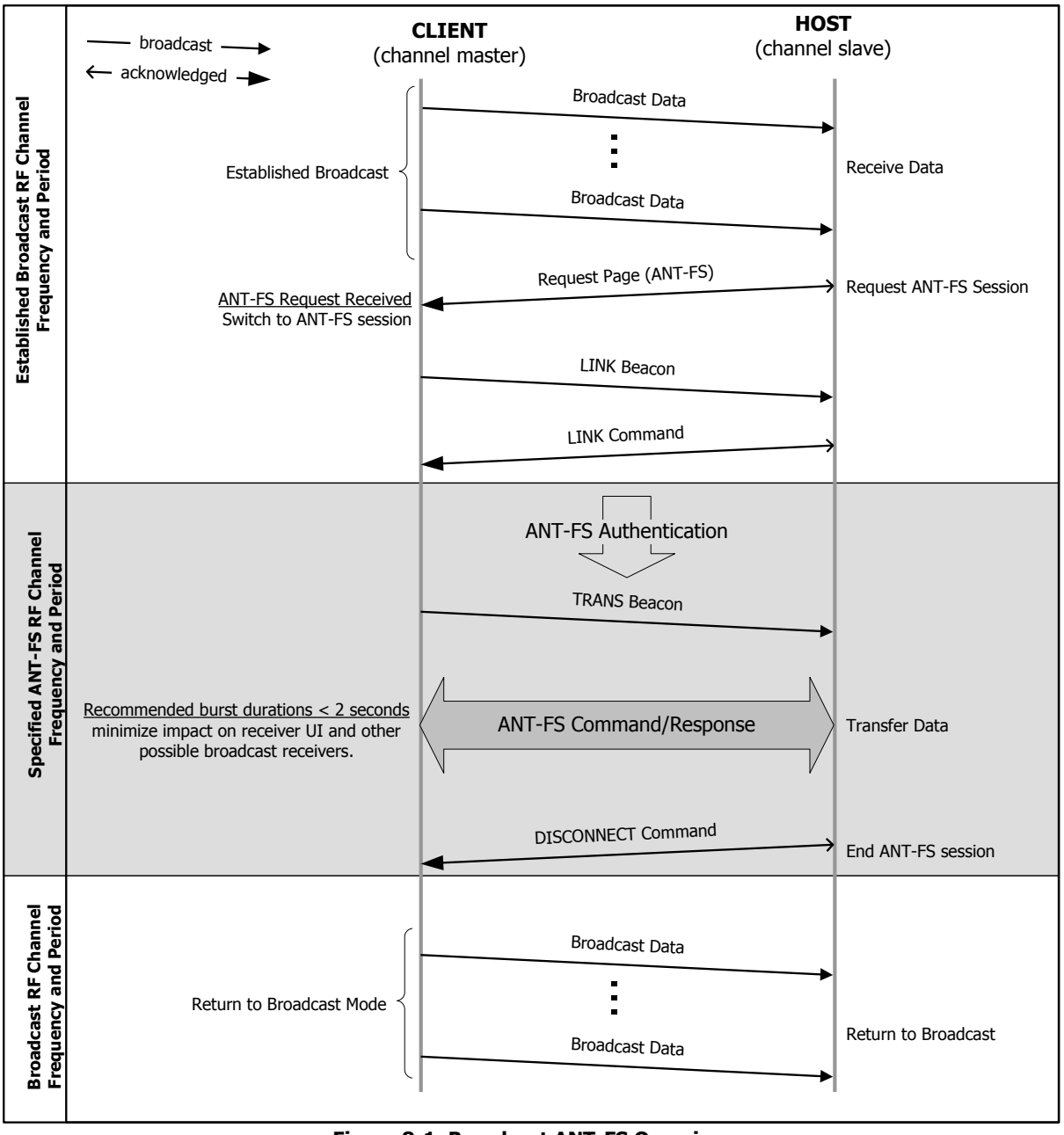

Figure 8-1. Broadcast ANT-FS Overview

A standard broadcast channel is established, and available data transmitted. The slave device may request an ANT-FS session at any time during the broadcast, using common data page 70 (Request Data Page). On receiving the ANT-FS request, the master device shall stop broadcasting data and transmit the ANT-FS Link beacon instead, effectively starting the ANT-FS session.

The host shall request a change of RF Channel Frequency and Channel Period in order to avoid any neighbouring broadcasting devices.

On the completion of the ANT-FS session, the host shall send a disconnect command, indicating that the device shall return to broadcast mode; at which point the client shall return to broadcasting as a master device on the original established channel's RF channel frequency and channel period.

8.1 Request ANT-FS session

The slave device in the broadcast session requests an ANT-FS session using the common pages Request Data Page. This data page shall be used to request the beacon (page 0x43), and explicitly request an ANT-FS session (command type = 2). Refer to the ANT+ Common Pages document for more details on the Request Data Page.

The request data page should be sent as an acknowledged message. It may not be a requirement that the master device supports ANT-FS or the request page. If the master does not support one, either or both, it shall ignore the slave's request. The master device shall handle this "no response" case elegantly.

8.2 Broadcast ANT-FS Best Practices

Once an ANT-FS session has been requested, the master and slave devices shall behave as ANT-FS client and host devices respectively. The following points should be considered when implementing broadcast ANT-FS:

- Initial Link State Channel Period: The initial beacon channel period used by the client device shall match that of the master's established broadcast channel period.
- RF Channel Frequency: Both devices shall initially operate in the link layer at the established RF Channel Frequency.
- The host shall request a change in RF channel frequency in the link command. Changing the RF channel frequency will allow the devices to transfer files without creating interference to other, neighboring broadcasting devices.
- Authentication: defined by the application
- Once in the Link state, ANT-FS operates as per standard ANT-FS.
- Once the data has been transferred, the host shall send the Disconnect (return to broadcast) command, and both devices shall return to the standard Broadcast session.

9 Best Practices

Various applications of the ANT-FS protocol may not require that all ANT-FS features are supported. For example, some client devices may not support uploads, or stipulate that only pairing authentication is allowed. Regardless of the particular implementation, if a specific feature is not supported both host and client devices must handle each situation gracefully.

In general, the client defines its supported features in its beacon, and the host shall note these capabilities and only request supported functions. However, if the host were to request unsupported features, the client shall be able to respond appropriately. This means that the client shall reject (or ignore) the unsupported command. The host shall be able to handle these rejected/ignored requests in such a way as to not cause error. In this way, both devices shall always return to a recoverable state.

The following sections provide some basic best practices that should be considered when implementing ANT-FS applications. Note that file based ANT+ device profiles may provide additional stipulations regarding ANT-FS implementations for specific devices.

9.1 Beacon

- Prompt updating of beacon status is important to reduce race condition possibilities between host and client. Ideally, the beacon shall be updated immediately following a command such that status details are correct by the next channel period.

- Client devices should keep information such as data availability, up to date to allow host devices to decide whether a session needs to be established.

- Client beacon rates will be application dependent. A trade off is made between power and latency. For example, an application may set initial beacon period to a low rate for power savings while in the unconnected link state, and a higher rate after establishing a connection for lower latency data transfers.

9.2 Frequency

- In general, ANT-FS connections should be established on a different RF channel frequency than the initial Link State frequency, or pre-defined RF channel frequencies such as the ANT+ frequency (2457 MH). This will prevent interference from burst transfers affecting other neighbouring broadcast devices.

- To simplify RF certification, it may be necessary to specify a range of allowable RF Channel Frequencies. For example, ANT+ devices operate within 2403 MHz to 2480 MHz.

9.3 Timeouts

Selection of timeouts is application dependent. On the client side there are two timeouts: session timeout and pairing timeout. The session timeout is the duration a client shall wait for an ANT-FS command from a host while in the authentication of transport layers. If a command has not been received within this time, the client shall return to the unconnected link state. The pairing timeout is the duration a client will wait for the application/user to respond to a pairing request. If a response has not been received within this time, the client shall return a pairing rejected response.

The host may have multiple timeout, all of which are application dependent.

- Selection of timeouts is application dependent and care must be taken to ensure that they are appropriate for the desired use case.

- Recommended client session timeout, and the default used in reference designs, is 10 seconds.

- Host design needs to maintain connections with a ping command when there is no activity. It is recommended the host sends a command at least every 5 seconds.

- Pairing Timeout depends on device use case and user interface. Several minutes is typical.

- Host operation timeouts shall be chosen based on worst case client operation delays. For example, a client device may have a worst case ~30 seconds response time from the download request to download response. It is important the beacon is updated appropriately to indicate the state of the client (refer to Busy State section).

- Note: session timeouts are suspended while the client is processing an ANT-FS request, and a ping is not required. Using the prior example, once the client has received the download request, the session timeout is suspended until the client has completed the request. The session timeout is reset immediately after the client sends the download response.

9.4 Busy State

It is important to use the Busy State correctly such that the host can function correctly with respect to timeouts and retries.

- The client shall set the beacon to indicate busy state immediately after a command has been received from the host.

- The busy state is not cleared from the client beacon until *after* the appropriate response has been sent. The beacon included with the response shall indicate busy state.

- The client application should include a failsafe such that the client cannot stay busy indefinitely (i.e. hung in the busy state).

- The host shall not send a command to the client while the beacon indicates it is in the busy state. This includes the ping command, as client session timeout is suspended while the client is busy processing a request.

9.5 File/Burst Management

- Efficient file access methods are important to allow for maximum transfer speed and to minimize the number of retries due to burst starvation. ANT burst transfers can be sustained at ~10 kbps or less; however, while this is functional, at only half the speed (and twice the power) it is far from optimal.

- Latency in retries can be reduced using techniques like progressive CRC saving, such that the CRC does not need to be recalculated from the start of a file (saves up to 10 packets may be required due to the uncertainty on exactly where the transfer failed).

- Client devices should maintain accurate information about their data content in the directory to allow host devices to initiate data transfers as needed for the application.

- After establishing a connection, host devices should always request a client's directory and should never assume that data will be in a given index. This will ensure interoperability across implementations from other manufacturers and account for any external changes to the file system.

- Host devices must be able to adapt to files being larger than listed in the directory. This may occur if a file is periodically updated, and the update occurs in the time between directory download and the file's download request.

- The client shall give accurate error messages for failed or rejected requests. In particular, reasons for failed/rejected upload and download requests should be clearly represented. This will allow the host to take appropriate action, such as retrying the operation if the device is temporarily busy with an operation.

- Setting Archive bit correctly in directory is important. Anytime a new or updated file has been stored in the directory, ensure the archive bit is not set. Anytime a new/updated file has been downloaded, the archive bit shall be set to indicate that file has been downloaded at least once by the same, or a different, device.

Downloading an 'old' file (i.e. one that has already been downloaded) will not change the status of the archive bit.

9.6 Command Pipe

- Command pipe usage depends on proper use of sequence numbers.
- Wait time to check for a command pipe response will depend on the client's maximum operation time. A polling mechanism should be used.

10 ANT Channel Configuration

For ANT-FS, the client device must be configured as a master ANT channel and the host as a slave. Most channel parameters are application defined. The following sub-sections detail the client device and host channel parameter configuration.

Note that these channel parameters are only restricted to standard ANT-FS sessions. Broadcast ANT-FS sessions are initialized off previously configured channels and will initially maintain those parameters.

10.1 Client Device ANT Configuration

The client device's channel parameters are configured as shown below:

Table 10-1. Client Device ANT Configuration

Parameter	Value	Comment
Channel Type	Master (0x10)	The client device transmits a beacon at a fixed interval
Network Key	Application Dependent	The Network Key is dependent on the application. ANT+ and ANT-FS may be used. Private keys may also be obtained. Please contact Dynastream for details on obtaining a key. Refer to the ANT Message Protocol and Usage document
RF Channel	Application Dependent	Application Dependent Refer to the ANT Message Protocol and Usage document
Transmission Type	1-255	Application Dependent Refer to the ANT Message Protocol and Usage document
Device Type	1-127	Application Dependent Refer to the ANT Message Protocol and Usage document
Device Number	1-65535	Application Dependent Refer to the ANT Message Protocol and Usage document
Beacon Channel Period (s/32768)	65535, 32768, 16384, 8192, 4096	The beacon transmitter may select any of the periods defined in the "Value" column. The higher the period (i.e. faster rate), the greater the power consumption; however search time is reduced.

10.1.1 Beacon Channel Period

While the beacon channel period is somewhat application dependent, ANT-FS defines a number of allowable values. These values have been defined in order to manage power consumption of the client device. The trade-off to improving power consumption on the client, is the time it takes the host to discover the client's beacon. The following beacon periods are usable in the ANT-FS protocol: 0.5 Hz, 1 Hz, 2 Hz, 4 Hz, 8 Hz or broadcast ANT-FS established channel period. Refer to section 9, for best practices on setting the link beacon channel period.

10.2 Host Device ANT Configuration

The following table lists the Host device's configuration details:

Table 10-2. Host ANT Configuration

Parameter	Value	Comment
Channel Type	Slave (0x00)	The host will be configured as a bidirectional slave device
Network Key	Application Dependent	The Network Key is dependent on the application. ANT+ and ANT-FS network keys are available for use. Private keys may also be obtained. Please contact Dynastream for details on obtaining a key. Refer to the ANT Message Protocol and Usage document
RF Channel	Application Dependent	Application Dependent Refer to the ANT Message Protocol and Usage document
Transmission Type	1-255 0 for searching	Application Dependent Refer to the ANT Message Protocol and Usage document
Device Type	1-127 0 for searching	Application Dependent Refer to the ANT Message Protocol and Usage document
Device Number	1-65535 0 for searching	Application Dependent Refer to the ANT Message Protocol and Usage document
Channel Period (s/32768)	4096 (8Hz)	The host channel period is initially configured to the fastest ANT-FS message rate. The channel period can be changed to match the beacon's channel period once a beacon has been found.
Timeout	255	The timeout is set to an infinite time out. However, note that for nRF24AP1, this value represents a 10.5min timeout and the channel will need to be reopened after a time out.

The host shall be configured as a slave channel. The device number of the channel ID may be wild carded (i.e. set to 0) or set appropriately to connect to the desired client device. In the event of a channel timeout, the slave channel should be re-opened immediately. Alternatively the channel search timeout may be set to infinite in order to ensure that searching never stops. Scanning mode may also be used, if appropriate. Note that infinite timeouts and scanning modes are only available on select hardware; refer to datasheets for device capabilities.

11 ANT-FS Client Device Beacon

The eight-byte payload for the ANT-FS beacon is formatter as shown in Table 11-1.

Table 11-1. ANT-FS Beacon Format

Byte #	Field	Length	Description
0	ANT-FS Beacon ID (0x43)	1 Byte	This ID is used to identify how to interpret the command below
1	Status Byte 1	1 Byte	This is a bit field indicating client device state information
2	Status Byte 2	1 Byte	This is a bit field indicating client device state information
3	Authentication Type	1 Byte	The authentication type supported by this device
4	Descriptor/Host SN LSB	4 Bytes	ANT-FS Device Descriptor This field is used to identify the manufacturer and device for the purposes of message decoding/encoding while in the Link Layer. Host Serial Number In Authentication and Transport Layers, this field is filled in with the host's serial number
5	Descriptor/Host SN		
6	Descriptor/Host SN		
7	Descriptor/Host SN MSB		

The ANT-FS client device transmits a beacon message every channel period. The beacon informs the host of the client's current state and supported modes. Any required response from the client to a host command is appended to the beacon as a burst transfer. In this case, the client sets a flag such that the response is burst along with the beacon on the next channel period (Figure 11-1).

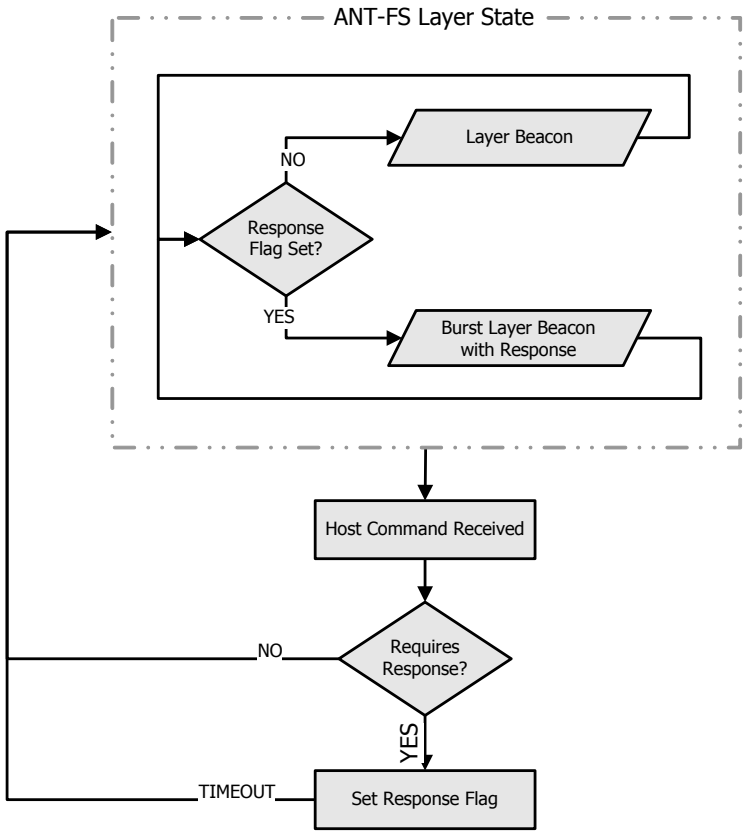

Figure 11-1. Beacon/Burst Response Mechanism

Only single responses may be sent in any burst transfer. The first packet following the beacon shall include a formatted response as detailed in section 12.

11.1 Status Byte 1

Status Byte 1 is described in Table 11-2. Bits are numbered from most to least significant.

Table 11-2. Beacon Status Byte 1

Bit(s)	Field	Value	Description
6,7	Reserved	N/A	N/A
5	Data Available	0 1	No data available Data is available for download
4	Upload Enabled	0 1	Upload is disabled Upload is enabled
3	Pairing Enabled	0 1	Pairing is disabled Pairing is enabled
2..0	Beacon Channel Period	000 001 010 011 100 101-110 111	0.5 Hz (65535) 1Hz (32768) 2Hz (16384) 4 Hz (8192) 8 Hz (4096) Reserved Match Established Channel Period **(broadcast ANT-FS only)**

Data Available bit indicates that the client device currently has data available for download to a host device.

Upload Enable bit indicates that the client device currently supports uploads.

Pairing Enable bit indicates that the client device will currently support a pairing operation, which will typically involve user involvement with the client UI.

Beacon Channel Period contains the current beacon rate of the client device. The host device will use this field to verify that the message rates of the host and client match.

Note that a special value exists for broadcast ANT-FS. This will allow both devices to switch into an ANT-FS session without changing their established message rate.

11.2 Status Byte 2

Status Byte 2 is described in Table 11-3 (bits numbered most to least significant).

Table 11-3. Beacon Status Byte 2

Bit(s)	Field	Value	Description
7..4	Reserved	N/A	N/A
3..0	Client Device State	0000 0001 0010 0011 0100-1111	Link State Authentication State Transport State Busy State Reserved

Client Device State shows the current client ANT-FS layer state. This field is used by the host device to verify that requested state transitions have occurred. The Busy State can be used at any time, and in any state, to indicate that the client is processing a command from the host.

11.3 Authentication Type

Currently the ANT-FS protocol supports three authentication types: pass-through, pairing and passkey. The authentication type field provides an indication to the types of authentication a client device supports and is described in Table 11-4.

Table 11-4. Beacon Authentication Types

Authentication Type	Description
0	Pass-through supported (pairing & passkey optional)
1 (N/A)	**Not applicable for client devices** (type defined for host devices only)
2	Pairing Only
3	PassKey and Pairing Only
4..255	Reserved

If the Authentication type is set to a non-zero number, the Authentication routine must be executed before the host may proceed to the transport layer. An authentication type of 1 is not applicable to a client device.

Table 5-1 and the following sections describe the relationship between the authentication type field and supported authentication methods.

11.3.1 Pass-through Supported

A client will only support pass-through if the authentication type is set to 0. Some client devices with this setting may also support pairing and passkey, depending on the application. The pairing enabled bit in status byte 1 will indicate if the client supports pairing. If pairing is supported and requested, the client will include a passkey in the pairing response. Refer to section 12 for details of ANT-FS commands and responses.

11.3.2 Pairing Only

An ANT-FS client that supports pairing only, shall set this field to 2. An ANT-FS host shall not request passkey or pass-through authentication when this value is set.

11.3.3 Passkey& Pairing Only

When the authentication type is set to passkey and pairing, pass-through authentication shall not be requested by the host.

11.4 ANT-FS Device Descriptor / Host Serial Number

While in the Link layer, the client device will broadcast a device descriptor within its beacon data payload. The ANT-FS Device Descriptor is an identifier which may be used to indicate the types of communication, and data, that a specific device supports. The device descriptor is defined in Table 11-5.

Table 11-5. Beacon ANT-FS Device Descriptor Format

Byte #	Type	Description
4	Device Type LSB	Device Type
5	Device Type MSB	
6	Manufacturer ID LSB	Manufacturer ID. Most significant bit indicates if Device Type is ANT+ (msb=1) or Dynastream managed (msb=0)
7	Manufacturer ID MSB	

The manufacturer ID is maintained by Dynastream Innovations. Please contact Dynastream to inquire about specific manufacturer IDs, or to request new IDs. The most significant bit of the manufacturer ID is reserved. If set, this bit indicates the Device Type is managed by the ANT+ Alliance; however, if this bit is 0, the device type is managed by the manufacturer.

While in the Authentication or Transport layer, the client device will broadcast the host's serial number in lieu of the device type and manufacturer ID. This allows a host device to determine that the client device is communicating with a host, and provides the identity of that host. The format of the host serial number is as follows:

Table 11-6. Beacon Host Serial Number Format

Byte #	Type	Description
4	LSB	Host Device Serial Number
5	…	
6	…	
7	MSB	

12 ANT-FS Host Command/Response

ANT-FS interactions are initiated by a host using an ANT-FS command, and the client will respond with an updated beacon or a beacon appended with a response message.

The host shall send ANT-FS commands as acknowledged messages or burst transfers, depending on the size of the commands. The client device shall respond with an updated beacon sent as a broadcast message, or with a burst transfer that includes the beacon in the first packet and the ANT-FS response in the remaining packets. The ANT-FS command/response is formatted according to Table 12-1.

Table 12-1. ANT-FS Command/Response Format

Byte #	Field	Length	Description
0	ANT-FS Command/Response ID (0x44)	1 Byte	This ID identifies the message as an ANT-FS Command/Response Message
1	Command	1 Byte	The ID of the Command/Response byte
2..(N+1)	Parameters	N Bytes	The number of parameters is dependent on the ANT-FS Command/Response sent. Refer to Table 12-2 and Table 12-3.

Notes:

- All ANT-FS parameters are formatted little endian (i.e. least significant byte first).
- Unless explicitly stated otherwise, all reserved and unused bytes are set to 0

12.1 ANT-FS Command

Table 12-2 summarizes the list of ANT-FS commands that may be sent from the host. Each command's ID, parameters, and availability (i.e. layer within which the command may be used) are also shown.

Table 12-2. ANT-FS Command Messages

Command (Section)	Layer	ID	Param0 (Byte 2)	Param1 (Byte 3)	Param2 (Byte 4)	Param3 (Byte 5)	Param4 (Byte 6)	Param5 (Byte 7)	Extra Packets
Link Command (12.3)	Link/ Trans	0x02	Channel Frequency	Channel Period	Host Device Serial Number				N
Disconnect Command (0)	Auth/ Trans	0x03	Command Type	Time Duration	App' Specific Duration				
Authenticate Command (0)	Auth	0x04	Command Type	Auth String Length	Host Device Serial Number				App' Dep'
Ping Command (0)	Auth/ Trans	0x05							
Download Request (0)	Trans	0x09	Data File Index		Download Parameters…				Y
Upload Request (12.8)	Trans	0x0A	Data File Index		Upload Request Parameters…				Y
Erase Request (12.10)	Trans	0x0B	Data File Index						
Upload Data (12.8)	Trans	0x0C	CRC Seed		Upload Data Parameters…				Y

12.2 ANT-FS Response

Table 12-3 summarizes the available ANT-FS responses and their respective IDs, parameters and layer availability. Note that if a client responds with one of the ANT-FS response messages listed below, this response will be appended to the beacon and sent as a burst transfer.

Table 12-3. ANT-FS Response Messages

Response	Layer	ID	Param0 (Byte 2)	Param1 (Byte 3)	Param2 (Byte 4)	Param3 (Byte 5)	Param4 (Byte 6)	Param5 (Byte 7)	Extra Packets
Authenticate Response (0)	Auth	0x84	Response Type	Auth String Length	Client Device Serial Number				App' Dep'
Download Request Response (0)	Trans	0x89	Response	0	Download Data Response Parameters…				Y
Upload Request Response (12.8)	Trans	0x8A	Response	0	Upload Data Response Parameters…				Y
Erase Response (12.10)	Trans	0x8B	Response						
Upload Data Response (12.8)	Trans	0x8C	Response						

12.3 Link Command (0x02)

While in the link state, the host will send this command when it wishes to connect to a client device. The expected radio frequency and channel period for further communication shall be specified in this command. The client must change its parameters to the requested values. The client beacon shall update its beacon accordingly (i.e. state and period change).

Table 12-4. ANT-FS Authenticate Command

Parameter	Type	Description	Range
Channel Frequency	U*1	The RF Channel Frequency that the client device shall switch to.	Varies
Channel Period	U*1	The Beacon Channel Period that the client device shall switch to. Refer to Table 11-2 for available beacon channel period values.	Varies
Host Serial Number	U*4	Serial number of host device	Varies

While in the transport state, this command may be sent to request a change of RF frequency and/or channel period for subsequent interactions. The client shall remain in the transport state, but it must change to the requested values, and update its beacon accordingly.

This command is sent as a single acknowledged message by the host device. The host shall resend the command in the event of failure.

12.4 Disconnect Command (0x03)

At any given time, the host may request that the client device disconnects and returns to either the unconnected link state (standard ANT-FS), broadcast mode (broadcast ANT-FS) or undiscoverable mode as indicated by the Command Type and duration fields (Table 12-5).

Table 12-5. ANT-FS Disconnect Command

Parameter	Type	Description	Range
Command Type	U*1	Indicates if the disconnect is request to link layer, or return to broadcast: 0: Return to Link Layer(default) 1: Return to Broadcast mode (Broadcast ANT-FS) 2-127: Reserved 128-255: Device specific disconnect	0-255
Time Duration	U*1	Indicates the amount of time (in 30 second increments) the client device shall remain in undiscoverable mode after the disconnect has been requested: Special Values: 0x00 – Disabled/Invalid	0-255
Application Specific Duration	U*1	Indicates an application specific duration the client device shall remain in undiscoverable mode after the disconnect has been requested: Special Values: 0x00 – Disabled/Invalid	0-255

This command is sent as an acknowledged message by the host device. The host shall resend the command in the event of failure. The client shall not respond directly to this message; rather, the client will return to a state as indicated in the command type field

12.4.1 Command Type

This field indicates the requested client behaviour on disconnecting from the host:

- If command type = 0, the client device shall return to the link state, or the client's default behaviour
- If command type = 1, the client shall return to broadcast mode
- If command type = 128-255, the client shall return the manufacturer specific behaviour

12.4.2 Duration Fields

The host may request the client return to the requested state after a specified duration of being undiscoverable (refer to section 7.3). There are two ways a client can become undiscoverable:

- Set the pairing bit in the device's channel ID (i.e. most significant bit of the device type)
- Close the channel

The duration time field shall indicate the amount of time a client device shall remain undiscoverable. On receiving a disconnect command, the client device shall become undiscoverable for the shortest of the two durations specified in the time duration field and application specific duration field. This will temporarily prevent the client from being found by the host.

For example, if the client were a pedometer, the host may issue the disconnect command to the pedometer requesting it be undiscoverable for a time duration of 10 minutes and/or an application specific duration of 240 steps. On receiving this command, the client device shall either close its channel, or set its pairing bit, until the user has taken 240 steps, or until 10 minutes has passed, whichever happens first.

Note, not all client devices will interpret these fields.

12.4.2.1 Time Duration

The duration time field indicates the amount of time that the client device shall remain in undiscoverable mode. The client device shall close its channel until this amount of time has passed, or until the application specific duration limit has been exceeded.

If undiscoverable mode is not desired, or based on the application specific duration only (i.e. not time based), this field shall be set to invalid/disable (0x00).

12.4.2.2 Application Specific Duration

This field is an application specific duration, and is left to the developer, or defined in some ANT+ device profiles. The client device shall close its channel until this duration has been exceeded, or until the time duration has been met.

If undiscoverable mode is not desired, or purely time based, this field shall be set to invalid/disable (0x00).

12.5 Authentication

12.5.1 Authenticate Command (0x04)

The authenticate command is sent by a host in the Authentication layer to establish a trusted relationship with a client device. Unless using pass-through, this process usually involves requesting the full serial number of the client device to determine if a passkey, or other information, is already known about that client. Other possible actions are pairing, passkey exchange, or custom authentication processes implemented at the application level.

The parameters for the authenticate command (0x04) are given in Table 12-6.

Table 12-6. ANT-FS Authenticate Command

Parameter	Type	Description	Range
Command Type	U*1	Authentication command types correspond to the Authentication types found in the beacon message (i.e. Table 11-4): 0: Proceed to Transport (pass-through) 1: Request client device serial number 2: Request Pairing 3: Request Passkey Exchange	0..3
Authentication String Length	U*1	Length of Authentication string. The string is burst to the client immediately following this command. Set to 0 if no authentication is to be supplied	Varies
Host Serial Number	U*4	Serial number of host device	Varies

If the authentication string length is not 0, the authentication string is appended to the authenticate command and burst to the client.

12.5.1.1 Command Type

12.5.1.1.1 Proceed to Transport

If the client device does not require additional authentication, this authentication type may be used to proceed to the transport layer.

12.5.1.1.2 Request Client Serial Number

This command will serve to receive the client device's full serial number. The client device may optionally append a friendly name following the response.

12.5.1.1.3 Pairing

The pairing process is a method to enable device verification on both the host and client devices. As part of the pairing process the host can optionally send a friendly name when using the pairing command by specifying a length of a friendly name inside the authentication command. The host then bursts this friendly name in the packets immediately following the command. User intervention on the interface of the client device may be required before the client will respond with acceptance of the pairi4ng request. The response may optionally include a passkey for future use by the host device.

12.5.1.1.4 Passkey Authentication

The host may establish a connection to a client directly if a shared password is known to both devices. This key should be generated by the remote device and acquired by the host using the pairing process. Then all future communication may be done without user intervention.

In some client devices, the passkey may be set using command pipe (refer to section 13.2).

12.5.1.2 Authentication String Length

An authentication string may be appended to this message to fulfill any requirement of a particular authentication type. If the Auth String Length parameter is set to 0, this message may be sent as an acknowledged message.

If the Auth String Length parameter is non-zero, the command must be sent as a burst with the authentication string contained in packets following the initial command packet.

The authentication string must correspond to the authentication type used. For example, when pairing, the host may wish to supply an optional friendly name. Another example is if the host is using passkey authentication, the passkey must be supplied in the authentication string.

12.5.1.3 Host Serial Number

The host device serial number must be provided in the Authenticate command. The serial number is transmitted LSB first.

12.5.2 Authenticate Response (0x84)

The parameters for the Authenticate Response (0x84) are provided in Table 12-7.

Table 12-7. ANT-FS Authenticate Response

Parameter	Type	Description	Range
Response Type	U*1	Authentication type defines as follows: 0: N/A (Response for client serial number request) 1: Accept 2: Reject	0..2
Authentication String Length	U*1	Length of Authentication string. The string is burst to the host immediately following this response. Zero indicates there is no authentication string supplied	Varies
Client Serial Number	U*4	Serial number of client device (little endian)	Varies

The authenticate response is appended to the beaconIf the authentication string length is not 0, the authentication string is appended to the authenticate response and burst to the host.

12.5.2.1 Response Type

The response type is used to indicate if the authenticate command was successful (i.e. accept) or not (reject).

12.5.2.2 Authentication String Length

The response may also append an authentication string. For example, if the host requests a pairing operation with a client that supports passkey authentication; the passkey may be appended to the successful authentication response such that the host can request passkey, rather than pairing, authentication in their next ANT-FS session (saving user action).

If the Auth String Length is zero, the authenticate response is appended to the beacon and burst to the host. If the Auth String Length parameter is non-zero, the beacon, response, and authentication string packets are burst to the host.

12.5.2.3 Client Serial Number

The client device's serial number shall also be provided in the Authenticate response. The serial number is transmitted LSB first.

12.6 Ping Command (0x05)

An ANT-FS client will typically time out and return to the link layer if a command is not received from a host within a specified period of time. If a host wishes to maintain a connection with a client, it may send a Ping command. A client device will reset its connection timer with every message received from a host.

12.7 Downloading

12.7.1 Download Request Command (0x09)

A host sends a Download Request command to request data from a client. The Download Request command is formatted according to Table 12-8.

Table 12-8. ANT-FS Download Request

Parameter	Type	Description	Range
Data Index	U*2	Specifies the file's index location within a file directory	Any
Data Offset (Bytes)	U*4	Specifies download from a known offset *within* a file. This allows the host to download specific blocks of data within a fail, or to resume cancelled/failed downloads without having to re-download the entire file	Any
Initial Request	U*1	Identifies weather the request is for a new transfer, or continuation of a partially completed download. **0 – Request is a continuation of a partially completed transfer**: CRC Seed shall be used to verify the accuracy of the data previously received and shall be used as the seed value for CRC checking of the remaining data transfer. **1 – Request is a new transfer:** The CRC Seed does not require checking, but should be set to zero and used as the seed value for CRC checking of the data.	0 or 1
CRC Seed	U*2	Used to verify data. If requesting a new transfer, the seed value should be set to zero. For continuation of a transfer, the seed value should equal the CRC value of the data received prior to the requested data offset.	
Maximum Block Size (Bytes)	U*4	Indicates the maximum number of bytes a client may transfer to the host in a single burst of data. This field may be set to zero if the host does not wish to limit the block size.	Any

12.7.1.1 Burst Format

The Download Request (0x09) command requires two pages of information. As such, this command must be sent as a burst message with the following format (Table 12-9).

Table 12-9. ANT-FS Download Request Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Packet 1	0x44 (ANT-FS CMD)	0x09 (CMD ID)	Data File Index		Data Offset (Bytes)			
Packet 2	0	Initial Request	CRC Seed		Maximum Block Size (Bytes)			

12.7.2 Download Request Response (0x89)

The client responds to a Download Request command with a Download Response message as shown in Table 12-10.

Table 12-10. ANT-FS Download Response

Header Parameters	Type	Description	Range
Response	U*1	0: Download Request OK 1: Data does not exist 2: Data exists but is not downloadable 3: Not ready to download 4: Request invalid 5: CRC incorrect	0-5
Total Remaining Data Length	U*4	Total number of bytes remaining in the data block. If no data is available for the specified data block (i.e. response is NOT = 0), the total remaining data length shall be set to 0, and the burst transfer is completed following the response (i.e. the footer packet is not included).	Any
Data Offset	U*4	The offset the data will start from in this block.	Any
File size	U*4	The size of the file on the client device.	Any

Footer Parameters	Type	Description	Range
Reserved	U*6	0 Pad Bytes	0
CRC	U*2	16 bit CRC for all data packets in this block (the beacon, header and footer is not included). The seed value for the CRC calculation should be the CRC seed provided in the download request (0x09).	

Note the data is transferred within header and footer packets.

12.7.2.1 Download Response Header

The download response header contains the information the host will need to successfully receive the data. The response field indicates if the data request was valid or not. If valid, it will provide transfer information such as the size of the data in the transfer, the offset at which the data transfer starts, and the size of the entire file on the client device.

For partial file transfers, the host requests a data offset. If the client is not able to provide the data from that exact offset value, it must provide data from an earlier offset only. The host should take note of the data offset value provided in the download response, in case it does not match that of the requested offset.

12.7.2.2 Download Response Footer

The response footer includes 6 zero bytes and a 16 bit CRC value calculated for all data packets in this block (beacon, header and footer packets are not included). The seed value for the CRC calculation should be the CRC seed provided in the download request (0x09).

ANT-FS reference designs, and libraries, use the CRC-16 polynomial $x^{16} + x^{15} + x^2 + 1$.

12.7.2.3 Burst Format

The client's Download Response (0x89) contains the beacon, two header packets, the download data packets followed by a footer packet containing the CRC (Table 12-11). This information is transferred in a single burst transfer.

Table 12-11. ANT-FS Download Response Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Packet 1	Beacon							
Packet 2	0x44 (ANT-FS CMD)	0x89 (CMD ID)	Response	0	Total Remaining Data Length (Bytes)			
Packet 3	Data Offset (Bytes)				File Size (Bytes)			
Packet 4:N	DATA PACKETS							
Packet N+1	0	0	0	0	0	0	CRC	

12.9 Uploading

12.9.1 Upload Request Command (0x0A)

The host sends an upload request command to ready the client device to receive a data upload. The command parameters are shown in Table 12-12.

Table 12-12. ANT-FS Upload Request Command

Parameter	Type	Description	Range
Data File Index	U*2	Specifies the index of the client's file directory to which the data will be uploaded	Any
Max Size	U*4	Indicates maximum file size and is equivalent to (offset + total remaining bytes). The last offset that will be written to is Max Size – 1.	Any
Data Offset	U*4	The data offset the requested upload will start at. If the request is to continue a transfer the value of MAX_ULONG (0xFFFFFFFF) can be used. This will indicate that the host will continue the upload at the Last Data Offset specified by the client in the Upload Response.	Any

This command requires two pages of information. As such, it must be sent as a burst message and will have the following format (Table 12-13).

Table 12-13. ANT-FS Upload Request Command Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Packet 1	0x44 (ANT-FS CMD)	0x0A (CMD ID)	Data File Index		Max Size			
Packet 2	0				Data Offset			

12.9.2 Upload Request Response (0x8A)

The client device replies to the Upload request with an Upload Request Response (0x8A) message which is formatted as shown in Table 12-14.

Table 12-14. ANT-FS Upload Request Response

Parameter	Type	Description	Range
Response	U*1	0: Upload Request OK 1: Data File Index does not exist 2: Data File Index exists but is not writeable 3: Not enough space to complete write 4: Request invalid 5: Not ready to upload	0-3
Last Data Offset	U*4	Last valid data offset written to the file. This value is only used if the host Upload Request Data specified MAX_ULONG (0xFFFFFFFF) in the Data Offset field. If the Upload Request specified any other value in the Data Offset value, this should be set to match, and the CRC value below should be reset to zero.	Any
Maximum File Size	U*4	Indicates the maximum number of bytes that can be written to the file. The last writeable offset is Maximum File Size – 1.	Any
Maximum Block size	U*4	Indicates the max number of bytes that the client can receive in a single burst.	Any
CRC	U*2	CRC value at last data offset. This value can be checked to ensure the integrity of the data up to the last data offset. If a Data Offset value other than MAX_ULONG (0xFFFFFFFF) was specified in the Upload Request, the CRC should be reset to zero.	Any

12.9.2.1 Response Type

The response field indicates if the request upload was successful or not. If successful, the client shall populate the remaining fields accordingly. If not successful, this field shall be set to indicate the cause of failure, and all remaining fields shall be set set to zero.

12.9.2.2 Burst Format

The upload request response is appended to the ANT-FS beacon and sent as a burst transfer (Table 12-15).

Table 12-15. ANT-FS Upload Request Response Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Packet 1	Beacon							
Packet 2	0x44 (ANT-FS CMD)	0x8A (CMD ID)	Response	0	Last Data Offset			
Packet 3	Maximum File Size				Maximum Block Size			
Packet 4	0	0	0	0	0	0	CRC	

12.9.3 Upload Data Command (0x0C)

After receiving a successful upload request response message, the host may transfer the data data to the client using the Upload Data command (0x0C). Similar to the Download Request Response, data is transferred within header and footer packets. The upload header and footer fields are outlined in Table 12-16

Table 12-16. ANT-FS Upload Data

Header Parameters	Type	Description	Range
CRC Seed	U*2	The seed value that should be used to start the CRC calculation for the data in this block of the transfer.	Any
Data Offset	U*4	Byte offset to which the data must be stored within the client's file. If this is a new upload request, this data offset will be the host's desired offset location. If the host wishes to resume a failed transfer, it may continue uploading from the Last Data Offset field found in the client's upload request response.	Any

Footer Parameters	Type	Description	Range
Reserved	U*6	0 Pad Bytes	0
CRC	U*2	The CRC value for the data packets contained in this block of the transfer. The CRC calculation is seeded with the CRC Seed value.	Any

12.9.3.1 Burst Format

The upload data command is sent as a burst transfer with the header parameters contained in the first burst packet, followed by the data packets and concluding with the footer packet containing the CRC.

Table 12-17. ANT-FS Upload Data Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Packet 1	0x44 (ANT-FS CMD)	0x0C (CMD ID)	CRC Seed		Data Offset			
Packet 2:N	DATA PACKETS							
Packet N+1	0						CRC	

12.9.4 Upload Data Response (0x8C)

The client responds to an Upload Command with the Upload Data Response (0x8C). This message will indicate if the upload transfer has been terminated successfully or unsuccessfully or not. The client will return failure if the transfer is interrupted, or if an error occurs, while handling the data on the client device. The Upload Data Response is shown in Table 12-18.

Table 12-18. ANT-FS Upload Data Response

Parameter	Type	Description	Range
Response	U*1	0: Data Upload Successful OK 1: Data Upload Failed	0-1

12.9.4.1 Burst Format

The upload request response is appended to the ANT-FS beacon and sent as a burst transfer (Table 12-19).

Table 12-19. ANT-FS Upload Data Response Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Packet 1	Beacon							
Packet 2	0x44 (ANT-FS CMD)	0x8C (CMD ID)	Response	0				

12.10 Erase

12.10.1 Erase Request Command (0x0B)

The Erase command is used by the host to erase a data file from the client deviceand is formatted according to Table 12-20.

Table 12-20. ANT-FS Erase Request

Parameter	Type	Description	Range
Data File Index	U*2	The index specifies the index of the data file in the data file table. The data file table can be logged using the Dir command.	Any

The Erase Request should be sent as an Acknowledged message. The host may wish to retry the command should it fail.

12.10.1.1 Erase Response (0x8B)

The client device will respond to the erase command with the erase response (0x8B). The response is sent after the erase cycle has completed, or if the erase request is refused. The client must also change the Data Available state of its beacon if applicable. The erase response is formatter as shown below (Table 12-21).

Table 12-21. ANT-FS Erase Response

Parameter	Type	Description	Range
Response	U*1	0: Erase Successful 1: Erase Failed 2: Not Ready	0-1

12.10.1.2 Burst Format

The erase response is appended to the ANT-FS beacon and sent as a burst transfer (Table 12-22).

Table 12-22. ANT-FS Erase Response Format

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Packet 1	Beacon							
Packet 2	0x44 (ANT-FS CMD)	0x8B (CMD ID)	Response	0				

13 Reserved File Indexes

The ANT-FS protocol has reserved a small number of file indexes for specific purposes. These reserved file indexes are outline in Table 13-1.

Table 13-1. ANT-FS Reserved File Indexes

File Index	Description
0	Directory Structure
0xFC00 to 0xFFFD	Reserved
0xFFFE	Command Pipe
0xFFFF	Reserved

13.1 Directory File

The Directory file is found at reserved file index zero (0). The directory may be read by a host device, but it may not be written to or erased by a host.

The Directory file begins with a 16 byte header described below (Table 13-2).

Table 13-2. ANT-FS Directory File Header

Parameter	Type	Description	Range
Version	U*1	Directory file structure version number. The most significant 4 bits indicate major revision (format of Index and File Data Type) while the least significant 4 bits indicate a minor revision.	1
Structure Length	U*1	The length of each structure in bytes.	16
Time Format	U*1	Defines the system's method of stamping Date/Time 0 – Device will use time as described in the Date parameter below, if the correct time is not known (prior to initial time sync) the device will use system time instead. 1 - Device will only use system time (seconds since power up). 2 – Device will only use the Date parameter as a counter.	0,1,2
Reserved	5 bytes	0 Pad Bytes	0
Current System Time	U*4	The number of seconds elapsed since system power up. If Time Format is set to 0, the system time should roll over at 268435455 (0x0FFFFFFF). If system time is not used, it shall be set to 0xFFFFFFFF.	0-268435455 (0x0FFFFFFF)
Directory Last Modified Date/Time	U*4	Number of seconds elapsed since 00:00 (i.e. morning of) December 31, 1989. 0xFFFFFFFF specifies an unknown date. Values of less than 268435455 (0x0FFFFFFF) will be interpreted as being system time or some other custom time format. This field is updated each time the directory entries are modified. In systems where time is unknown, this variable may be used as a counter that is incremented each time the directory entries are modified.	Varies

Table 13-3 describes how each file's information is stored in the directory structure. For every file, there will be a corresponding structure.

Table 13-3. ANT-FS Directory File Content Structure

Parameter	Type	Description	Range
Index	U*2	Specifies the data file index. This value is used when referencing files for download, upload, and erase.	1-65535
File Data Type	U*1	Specifies the data type of the file. The File Data Type informs the host how the file may be interpreted.	0-255
Identifier	U*3	Used in conjunction with data type, to uniquely identify a file. The structure of this ID is defined by the Data Type. Structures may include data sub-types. Refer to Appendix A for an example.	0-16777215
File Data Type Specific Flags	U*1	Bit mapped flags of file data type specific permissions Bits are defined by the File Data Type. Bits should be allocated starting from Bit 0 to Bit 7.	Bit map
General File Flags	U*1	Bit mapped flags of file permissions Mapping of Bits (0-7) shown below: 7 6 5 4 3 2 1 0 | | | | | | X X -----> (bit 0-1) Reserved | | | | | X ----------> (bit 2) Crypto (file is encrypted) | | | | X ----------> (bit 3) Append (can append to file only) | | | X ----------> (bit 4) Archive (been downloaded) | | X ----------> (bit 5) Erase (can erase) | X ----------> (bit 6) Write (can upload) X ----------> (bit 7) Read (can download)	Bit map
File Size	U*4	Size of file in bytes. If file size is 0, then no file is available for download, but may be available for upload.	Varies
Date	U*4	The number of seconds elapsed since 00:00 in the morning of December 31, 1989. 0xFFFFFFFF specifies an unknown date. Values of less than 268435455 (0x0FFFFFFF) will be interpreted as being system time or some other custom time format.	Varies

13.2 Command Pipe

The command pipe is found at the reserved file location 0xFFFE and, like any typical file, may be written or read using the upload/download functions. The command pipe can be used by the host application to pass commands to, and receive responses from, the client application. It is not a requirement to list the command pipe in the directory.

13.2.1 Using the Command Pipe

A file containing a command for the client application can be uploaded to the command pipe. The host may read the client's response by downloading from the command pipe. These command/response "files" shall all follow the format outlined in Table 13-4.

Table 13-4. ANT-FS Command Pipe "Files"

Parameter	Type	Description	Range
Command Pipe Message ID	U*1	ID of the command or response.	0-255
Reserved	2 Bytes	These bytes are reserved and should be set to zero.	0
Sequence Number	U*1	The sequence number is used to correlate commands and responses. The host should increment the sequence number every time it uploads a new command. The response for a particular command will have the same sequence number as the command.	0-255
Data	-	Data attached to the command/response	0

13.2.2 Reserved Command Pipe Message IDs

The following command pipe message ID's have been reserved (Table 13-5).

Table 13-5. ANT-FS Command Pipe Message IDs

Command Pipe Message ID	Description
0x00 to 0x7F	ANT-FS defined message IDs. These IDs are reserved for ANT-FS related commands. It is not mandatory that all devices support these commands. They are defined here to provide a consistent interface for all devices that support them.
0x80 to 0xFF	Manufacturer/device-defined command IDs.

13.2.3 Request (0x01)

The request message is used to request certain pieces of information from the client application.

Table 13-6. ANT-FS Command Pipe Request Message

Data Parameter	Type	Description	Range
Requested ID	U*1	ID of the message being requested	0-255

The format of the request message file is given in Table 13-7.

Table 13-7. ANT-FS Command Pipe Request Message Format

PIPE CMD ID			Sequence Number	Requested ID			
0x01	0	0	Sn	Required ID	0	0	0

13.2.4 Response (0x02)

The response message is used by the client application to respond to commands from the host.

Table 13-8. ANT-FS Command Pipe Response Message

Data Parameter	Type	Description	Range
Command ID	U*1	Command ID of the message being responded to	0-255
Reserved	1 Byte		0
Response	U*1	Response code: 0 – No error 1 – Command failed 2 – Command rejected 3 – Command not supported	

The format of the response message file is provided in Table 13-9.

Table 13-9. ANT-FS Command Pipe Response Message Format

PIPE CMD ID			Sequence Number	Command ID		Response Code	
0x02	0	0	Sn	CMDID	0	Response	0

13.2.5 Time (0x03)

The time message is used by the host application to send the current time to the client application. It may also be requested by the host application, in which case, the time message will be sent back to the host application as a response.

Table 13-10. ANT-FS Command Pipe Time Message

Data Parameter	Type	Description	Range
Current Time	U*4	The number of seconds elapsed since 00:00 in the morning of December 31, 1989. (1989-12-31 00:00:00 TAI). 0xFFFFFFFF specifies an unknown time.	Varies
System Time	U*4	The number of seconds elapsed since the system was powered up. During a time sync this parameter may be ignored by the client, the host may also choose to omit this parameter from the message file. This parameter only needs to be included by the client if the time message is requested by the host. 0xFFFFFFFF specifies an unknown time.	
Time Format	U*1	Defines how the system will keep track of Date/Time Stamps. 0 – Device will use time as described in the Date parameter of the directory, if the correct time is not know (prior to initial time sync) the device will use system time instead. 1 - Device will only use system time (seconds since power up). 2 – Device will only use the Date parameter as a counter. During a time sync this parameter may be ignored by the client, the host may also choose to omit this parameter from the message file. This parameter only needs to be included by the client if the time message is requested by the host.	

The format of the time message file:

Table 13-11. ANT-FS Command Pipe Time Message Format

PIPE CMD ID			Sequence Number	Parameters			
0x03	0	0	Sn	Current Time			
System Time				Time Format	0	0	0

13.2.6 Create File (0x04)

The Create File message is used by the host application to request that the client create a new file.

Table 13-12. ANT-FS Command Pipe Create File Message

Data Parameter	Type	Description	Range
File Size	U*4	The expected size of the file to be created	Any
File Data Type	U*1	This field specifies the data type of the file.	0-255
File Identifier	U*3	Specifies the file identifier of the file.	0-16777215
Reserved	1 Byte		0
File Identifier Mask	U*3	Specifies the portion of the file identifier that is wild carded. Any bit that is set to 1 will be wild carded. Eg: A value of (0x00FFFF) would specify that the last 2 bytes of the file identifier is wild carded and will be determined by the client device. The value of the final file identifier will be returned in the response message.	0 – 0xFFFFFF

The format of the create file message is given in Table 13-13.

Table 13-13. ANT-FS Command Pipe Create File Message Format

PIPE CMD ID			Sequence Number	Parameters	
0x04	0	0	Sn	File Size	
File Data Type	File Identifier		0	File Identifier Mask	

The client will respond with a response (0x02) message, with the parameters described in Table 13-14.

Table 13-14. ANT-FS Command Pipe Create File Response Message

Data Parameter	Type	Description	Range
Command ID	U*1	Command ID of the message being responded to	0x04
Reserved	1 Byte		0
Response	U*1	Response code: 0 – No error 1 – Command failed 2 – Command rejected 3 – Command not supported	0,1,2 or 3
Reserved	1 Byte		0
File Data Type	U*1	This field specifies the data type of the new file.	0-255
File Identifier	U*3	Specifies the file identifier of the new file.	0-16777215
File Index	U*2	Specifies the file index of the new file.	0-65535

Table 13-15 shows the format of the response message file for responding to a create file command.

Table 13-15. ANT-FS Command Pipe Create File Response Message Format

PIPE CMD ID			Sequence Number	Command ID		Response Code	
0x02	0	0	Sn	0x04	0	Response	0
File Data Type	File Identifier			File Identifier Mask			

13.2.7 Directory Filter (0x05)

The Directory Filter message is used by the host application to request that the client apply the requested filter to the directory listing. Once the filter is set it should remain in effect until the current transport session is terminated or until replaced by another filter.

Table 13-16. ANT-FS Command Pipe Directory Filter Message

Data Parameter	Type	Description	Range
Filter Type	U*1	Filter Type: 0 - No Filter (show all files) 1 – Device specific default filter	0-255

The format of the Directory Filter message file:

Table 13-17. ANT-FS Command Pipe Directory Filter Message Format

PIPE CMD ID			Sequence Number	Parameters			
0x05	0	0	Sn	Filter Type	0	0	0

13.2.8 Set Authentication Passkey (0x06)

The passkey message is issued by the host application to set the client's authentication passkey.

Table 13-18. ANT-FS Command Pipe Set Passkey Message

Parameter	Type	Description	Range
Version	U*1	Passkey file structure version number. The most significant 4 bits indicate major revision, while the least significant 4 bits indicate a minor revision.	1
Reserved	1 Byte		0
Authentication String Length	U*1	Length of Authentication string (i.e. Passkey).	Varies
Reserved	1 Byte		0
Authentication String	-	Contains the binary authentication string (i.e. passkey)	Varies

The format of the set passkey message is given in Table 13-19.

Table 13-19. ANT-FS Command Pipe Set Passkey Message Format

PIPE CMD ID			Sequence Number	Command ID		Response Code	
0x06	0	0	Sn	Version	0	Auth String Length	0
Authentication String							

13.2.9 Set Client Friendly Name (0x07)

The set friendly name message is issued by the host application to set the client's friendly name.

Table 13-20. ANT-FS Command Pipe Set Friendly Name Message

Parameter	Type	Description	Range
Version	U*1	Friendly name file structure version number. The most significant 4 bits indicate major revision, while the least significant 4 bits indicate a minor revision.	1
Reserved	1 Byte		0
Authentication String Length	U*1	Length of Authentication string (i.e. Friendly Name).	Varies
Reserved	1 Byte		0
Authentication String	-	Contains the binary authentication string (i.e. ASCII representation of the friendly name)	Varies

The format of the set friendly name message is given in Table 13-21.

Table 13-21. ANT-FS Command Pipe Set Friendly Name Message Format

PIPE CMD ID			Sequence Number	Command ID		Response Code	
0x07	0	0	Sn	Version	0	Auth String Length	0
Authentication String							

13.2.10 Factory Reset Command (0x08)

The factory reset message is issued by the host application to indicate to the client to perform a factory reset.

Table 13-22. ANT-FS Command Pipe Factory Reset Message

Parameter	Type	Description	Range
Reset Type	U*1	Reset Type: 0: Factory Reset 1-127: Reserved 128 –255: Device specific reset modes	Varies

The format of the factory reset command is given in Table 13-23.

Table 13-23. ANT-FS Command Pipe Factory Reset Message Format

PIPE CMD ID			Sequence Number	Parameters			
0x08	0	0	Sn	Reset Type	0	0	0

14 Reserved File Data Types

File Data Types are used to indicate how a particular file is encoded. A small set of file types are reserved for manufacturer/device specific use. The rest are set aside to be defined for universal encoding information.

Table 14-1. ANT-FS Reserved File Data Types

File Type	Description
0x00 to 0x0F	Manufacturer/device-defined file data types.
0x10 to 0xFF	Reserved file data types for global encoding. Refer to Table 14-2.

14.1 Defined File Data Types

Table 14-2 provides a summary of the currently defined global file data types.

Table 14-2. ANT-FS Defined Global File Data Types

File Type	Description
0x10-7F	Reserved
0x80	FIT files (.fit)
0x81-0xFF	Reserved

Appendix A – Example of FIT Directory Definition

The following table describes the ANT-FS directory entries for files formatted according to the Flexible and Interoperable Data Transfer (FIT) protocol.

Table A-1. ANT-FS Directory Entries for FIT Files

Parameter	Type	Description	Range
Index	U*2	Specifies the data file index. This value is used when referencing files for download, upload, and erase.	1-65535
File Data Type	U*1	Data type of the file. The File Data Type informs the host how the file may be interpreted. For example, a FIT file will have a File Data Type = 128	128
File Sub Type	U*1	Specifies the data sub type of the FIT file. For example, a settings file (refer to FIT File Types document)	0-255
File Number	U*2	Number used to identify a specific instance of a file sub-type. Can refer to user, session number, etc? For example, in a FIT file, this may correspond to file_id.number.	0-65535
File Data Type Specific Flags	U*1	Bit mapped flags of file data type specific permissions Mapping of Bits (0-7) are shown below 7 6 5 4 3 2 1 0 | | | | | | | X -----> (bit 0) Selected (file is user selected) X X X X X X X --------> (bit 1-7) Reserved* * Bits 1-7 are reserved in this version of the FIT protocol.	Bit map
General File Flags	U*1	Bit mapped flags of file permissions Mapping of Bits (0-7) are shown below 7 6 5 4 3 2 1 0 | | | | | X X X -----> (bit 0-2) Reserved | | | | X ----------> (bit 3) Append (can append to file only) | | | X ----------> (bit 4) Archive (been downloaded) | | X ----------> (bit 5) Erase (can erase) | X ----------> (bit 6) Write (can upload) X ----------> (bit 7) Read (can download)	Bit map
File Size	U*4	Size of file in bytes. If file size is 0, then file has not been uploaded but may be available for upload.	Varies
Date	U*4	The number of seconds elapsed since 00:00 in the morning of December 31, 1989. 0XFFFFFFFF specifies an unknown date. Values of less than 268435455 (0x0FFFFFFF) will be interpreted as being system time.	Varies

For details on FIT files, refer to the Flexible and Interoperable Data Transfer (FIT) Protocol and FIT File Types documents.

Appendix B – Example of Passkey Directory Definition

The following table describes the ANT-FS directory entries for passkey files.

Table A-1. ANT-FS Directory Entries for FIT Files

Parameter	Type	Description	Value
Index	U*2	Specifies the data file index. This value is used when referencing files for download, upload, and erase.	0xFFFD
File Data Type	U*1	Data type of the file. The File Data Type informs the host how the file may be interpreted. For example, a Passkey file will have a File Data Type = 129	129
Identifier	U*3	Reserved. 0xFFFFFF	0xFFFFFF
File Data Type Specific Flags	U*1	Bit mapped flags of file data type specific permissions Mapping of Bits (0-7) are shown below 7 6 5 4 3 2 1 0 X X X X X X X X--------> (bit 1-7) Reserved* * All bits are reserved in this version of the ANT-FS protocol.	0x00
General File Flags	U*1	Bit mapped flags of file permissions Mapping of Bits (0-7) are shown below 7 6 5 4 3 2 1 0 | | | | | X X X -----> (bit 0-2) Reserved | | | | X ----------> (bit 3) Append (can append to file only) | | | X ----------> (bit 4) Archive (been downloaded) | | X ----------> (bit 5) Erase (can erase) | X ----------> (bit 6) Write (can upload) X ----------> (bit 7) Read (can download)	Varies
File Size	U*4	Size of file in bytes. If file size is 0, then file has not been uploaded but may be available for upload.	Varies
Date	U*4	The number of seconds elapsed since 00:00 in the morning of December 31, 1989. 0XFFFFFFFF specifies an unknown date. Values of less than 268435455 (0x0FFFFFFF) will be interpreted as being system time.	Varies