CONTENTS

1. INTRODUCTION

This document provides information about network configuration requirements for the AudioFetch system. The intended audience includes:

- Network Administrators
- IT Personnel
- Independent Network Installation and Management professionals

There are two main subject areas covered:

1. General network/WiFi requirements and qualification guidelines to ensure the network is capable of supporting real-time audio streaming.

2. Information about packet traffic used by the AudioFetch system as related to discovery and audio streaming so that the network can be configured to allow the required traffic.

2. GENERAL NETWORK REQUIREMENTS

AudioFetch delivers real-time audio streams to WiFi connected client devices. Key elements of a good user experience are:

- Little delay in delivery of audio data to client devices
  - "low latency"
  - Audio remains in-sync with video displayed on TV screens (no "lip sync" issues)
  - Similar to the low-latency requirements of VOIP

- Consistent delivery of audio data to client devices
  - Temporary/intermittent delays on the network translate to noticeable audio drop-outs
  - Low latency requirement means the app running on client devices cannot pre-buffer the audio data to mask transient delays on the network
  - Different than pure internet streaming applications like Netflix, YouTube, etc. – they can pre-buffer for many seconds allowing operation on inconsistent networks

Therefore, a network supporting AudioFetch must perform well providing consistent low-delay packet delivery from the AudioFetch APB device (serving the audio streams to the wired portion of the network) out through the WiFi to the mobile clients.

The amount of data in each AudioFetch stream (one stream for each client) is not large, and even in when running with the maximum number of simultaneous users/streams (250) from a single APB device, the total streaming bandwidth consumed is substantially less than 100 Mbps:

Each user audio stream consumes about 90 Kbps (including all packet overhead)
250 streams x 90 Kbps/stream = 22.5 Mbps

This explains why a gigabit port on the APB device is not necessary, more than adequate bandwidth exists in a 100Base-T connection between APB and a network switch. If other wired portions of the network are gigabit, then this level of worst case AudioFetch traffic represents only about 3% of available bandwidth in the wired portions of the network.

A more typical use scenario for many facilities might be 50 simultaneous users, which would consume only 4.5 Mbps bandwidth, less than 1% of a gigabit connection's bandwidth.

Therefore, the amount of (audio) streaming data should not stress capacity of the wired portions of a network and consequently there should be no negative impact to consistent low-delay transport of the streaming audio packets.

The WiFi portion of the network is a different matter. There are many factors which can affect a WiFi network's ability to deliver data to client devices consistently with low-delay, a brief review of the most common issues to avoid will be presented as a guide to help the network administrator plan for successful deployment of an AudioFetch system.

3. WiFi REQUIREMENTS - General

The usual requirements for a well-deployed WiFi network should be met:

- Access Points should be of sufficient quality to support the expected user/traffic load

  - Consumer-grade equipment acceptable in limited circumstances: small facility with limited number of users. 25 max simultaneous users is simple rule of thumb for 2.4GHz WiFi, restrict WiFi to operate in 5GHz band only for more users.
  - If consumer equipment used, it should be high-end consumer-grade
- Enough Access Points to ensure not too many clients on any single one
- Access Points deployed to ensure adequate signal strength/quality throughout the facility
- Access Points should support modern N-mode speeds
  - Consider disabling legacy b-mode (and maybe even g-mode), which consumes lots of available bandwidth
- Access Points should be dual-band to reduce client load in crowded 2.4 GHz band
- If using just one Access Point (or wireless router) with more than 25 simultaneous users, it should be configured to operate only in the 5GHz WiFi band in order to avoid audio degradation
- Access Points should be configured to minimize interference from other WiFi networks
  - Careful management of channel use/configuration
  - Careful management of transmit power amongst all Access Points
- If Access Points are configured to use a "Guest Network" for end user access, devices connected to the wired portion of the network are generally not discoverable or accessible. In this situation:
  - Almost always, DNS discovery must be used (see information later in this document regarding "DNS LOOKUP" method.
  - Security settings must allow traffic between the guest network and the AudioFetch box residing on wired network.
  - Sometimes it can be helpful to configure the AudioFetch box to be part of the Guest Network using a Virtual LAN.
- Disable automatic channel hopping (some advanced access points support this)
  - Each channel hop will cause a long dropout in audio
  - Or set the channel hopping to only occur late at night when users not present
- Enable WMM in the Access Points
- Access Points do not interpret QoS settings in IP headers consistently. AudioFetch packets are marked as VOIP per Cisco recommendations, some other brand Access Points will interpret as VIDEO.
  - Try adjusting the "Force alternate WMM VOICE QoS" setting in the AudioFetch Doghouse configuration – this will cause these other Access Points to interpret/handle the packets as VOIP
- If possible, enable per-user bandwidth limiting so that one user can't hog the WiFi bandwidth (with a large file download for example) – limit to what's reasonable for expected/allowed use
- Ensure the WiFi network's DHCP server is set to provide lease times at least as long as the maximum expected time users will be connected
  - Re-acquiring a DHCP lease could disrupt the audio
- Other interference sources should be identified and minimized, here's a list of common things to look for:

- "Leaky" microwave ovens (microwave ovens operate at 2.4 GHz)
- 2.4/5 GHz cordless landline phone systems
- Leaky coax cable connectors used in Direct Satellite Service (DSS) signal distribution
- Wireless speaker systems or dedicated headphones operating in 2.4 GHz band
- RF Remote Controls operating in 2.4 GHz band
- Wireless video transmitters (WiDi technology, etc.)
- Wireless video monitoring systems
- Wireless automation (wireless thermostats, lighting controls, etc.)
- Radar Motion Sensors (sometimes used for automatic lighting control)
- Some computer monitors can leak 2.4 GHz signals
- Certain power lines, if in close proximity to the facility, can cause interference

All of these "usual" requirements are even more important for a WiFi network supporting real-time streaming. Don't be fooled by a network deployment which seems to function adequately for non-real-time applications such as email, surfing, internet audio/video streaming (where large pre-buffers are used), etc. – this is not necessarily a good indication the network will perform well for ***real-time*** streaming (consistent low-delay delivery of packets).

A few examples will be presented to help illustrate how a network deployment might seem to be adequate, but will in fact impact performance of a real-time streaming application such as AudioFetch. These examples are not meant to imply that it is difficult to deploy an adequate network for AudioFetch, the point is simply that attention to detail is required and a careful deployment will result in excellent AudioFetch performance.

Perhaps the most simple example is that of too many users/clients connected to a single Access Point causing intermittent delays at moments of peak traffic. Imagine there are occasional delays of 1 second or so. Connected users doing things like downloading email or web surfing won't notice these delays, but someone listening to a real-time audio stream will experience an audio drop-out at each and every delay. Clients receiving internet audio or video streams won't be affected because internet streaming applications typically pre-buffer more than 1 second of content, specifically to mitigate playback issues due to traffic delays like this.

Note that AudioFetch cannot employ a large pre-buffering strategy (such as used by internet streaming applications) to mask traffic delays on the network. Pre-buffering causes a delay in the playback of the stream, in the case of AudioFetch this would cause a loss of synchronization between the audio playback and on-screen video (lip-sync issues).

Another example is that of a client or small group of client devices operating in a weak signal area of the WiFi network. This situation causes the radios in the Access Point and client devices to throttle down their WiFi link speeds, in extreme situations the speeds could throttle down below 10 Mbps or less (we've seen 1 Mbps). Now imagine that 2 of those clients are streaming HD videos from the

internet (maybe 1.5 Mbps streams) and are doing so with a link speed of 5 Mbps. The simple math here is that the three clients are consuming 3 Mbps of the 5 Mbps link capability, in effect this means that these three devices are consuming 60% of the Access Point's available time to transmit packets. In other words, just a few devices operating at very low link speeds have consumed a majority of available bandwidth from the Access Point. Actually the real situation will probably be worse as the WiFi link to these clients will be experiencing a lot of retries, consuming even more of the Access Point's time. All other clients connected to that same Access Point, even those in close proximity operating at very high link speeds, will be impacted in terms of not being able to receive consistent low-delay delivery of their own packet streams as the Access Point struggles to (slowly) deliver data to the problem clients. This is an extreme example but it serves to illustrate how just a few devices operating in poor signal areas can impact overall performance of a WiFi network and why it is important to identify and mitigate areas of poor signal quality in a facility.

This kind of situation (previous example) may not cause a significant impact to users downloading email, surfing, etc., but likely will impact real-time audio streaming with occasional audio drop-outs due to inconsistent/delayed delivery of their streaming data. And of course any users trying to receive AudioFetch in these poor signal areas will likely experience poor audio quality.

The previous example also illustrates how per-user bandwidth limits can help reduce opportunities for individual client devices to impact general performance of the WiFi network, this subject will be discussed further in a subsequent section.

A last example is that of an external interference source. Imagine a neighboring business has a WiFi network operating on the same channel as the WiFi network supplying AudioFetch, and imagine it is set to high Tx power level. Then imagine that business can experience high levels of WiFi traffic from time to time such as large print jobs to a wirelessly connected printer. During these times, traffic on the WiFi network carrying AudioFetch data will be impacted due to the interference – delivery of audio stream packets will be delayed causing dropouts in the audio playback. Again this type of transient interference won't be noticed by when downloading email, surfing, or even streaming internet audio/video, but it impacts the real-time AudioFetch streams. Solution to this problem is easy: configure the AudioFetch WiFi to operate on a different channel.

Bottom line: paying careful attention to detail in a WiFi network deployment will help ensure good performance from an AudioFetch system.

4. WiFi REQUIREMENTS – QoS, WMM, and user bandwidth limiting

Beyond the usual requirements for a well-deployed WiFi network, there are a few specific configuration items which can greatly improve the network's ability to deliver consistent low-delay real-time audio traffic.

AudioFetch marks its streaming audio packets (TOS byte in IP headers) with the same Differentiated Services value as used by VOIP systems. Purpose is to leverage the QoS capabilities which already exist in most network equipment for consistent low-delay delivery of VOIP data, as needed by AudioFetch.

The details of packet marking for QoS and how networks should respond are beyond the scope of this document, but here are a few pieces of information which might be helpful when examining QoS configuration settings in network equipment:

- AudioFetch audio packets are marked with a value of 0xB8 in the TOS byte of the IP header
- 0xB8 is the recommended value for VOIP packets, lots of references to this value can be found in Cisco and other documentation
- For WiFi, the WMM standard defines a translation/mapping between the TOS byte and WMM "Access Categories"
    - The WMM defined mapping has TOS value of 0xB8 mapped to "AC_VI" (video), instead of the higher priority "AC_VO" (voice)
    - However Cisco, and possibly other Access Point vendors, in their WMM implementations, map 0xB8 as a special case to AC_VO to give VOIP-marked packets priority over video and audio
    - AudioFetch follows the Cisco recommendations for TOS value.
    - Consumer-grade equipment typically does not handle 0xB8 as a special case and map 0xB8 to AC_VI
    - Some interesting light reading on this subject can be found here

To help mitigate the issues and differences in translation/mapping between TOS byte and WMM, the AudioFetch Doghouse configuration offers a setting called "Force alternate WMM VOICE QoS" (in the Tools & Utilities page). When un-checked, AudioFetch produces packets marked with TOS value: 0xB8, Cisco and certain other brand Access Points will translate this to WMM AC_VO (voice) and manage the packets with good priority. When checked, AudioFetch produces packets marked with TOS value: 0xC0, certain non-Cisco brand Access Points will translate this to WMM AC_VO (voice) and manage the packets with good priority. An easy way to check what this setting should be is to monitor/sniff (with Wireshark) wireless packets transmitted to a connected mobile device receiving AudioFetch audio stream. Then use Wireshark to examine the "Qos Control" value in the IEEE 802.11 header, it should be equal to 0x0006 (which is the WMM AC_VO setting). If not equal to this value, try changing the setting in the AudioFetch Doghouse.

Bottom line is that equipment in the wired portion of the network should be configured to enable QoS support and adjusted if necessary to give VOIP highest priority, and WMM support should be enabled in all Access Points. In fact WMM is often enabled by default and not even configurable in commercial Access Points (such as Ubiquiti Unifi), however configuration settings should be reviewed

to be sure and especially if a consumer-grade device is being used in a smaller facility one should check its WiFi settings to ensure WMM is enabled – this can make a big improvement in real-time audio performance.

Another important configuration is per-user bandwidth limits in the WiFi portion of the network. Such limits prevent one or a few users from consuming large portions of the available WiFi bandwidth with large/fast downloads, high-bitrate internet video streaming, etc., where even a transient spike in consumed bandwidth can disturb real-time audio delivery to other WiFi clients.

Most commercial/enterprise-grade Access Points can be configured with per-user bandwidth limits, most consumer-grade equipment (Wireless Routers) cannot.

One notable exception for consumer-grade equipment is TP-Link, their Wireless Routers can be configured for bandwidth limits to specific devices on the network (not useful), but the same configuration feature also allows limits to be set for a range of IP addresses, which in effect becomes a per-user limit.

There is no hard and fast rule for what the per-user bandwidth limit should be, as it depends on what other kind of activity must be allowed on the network. For example internet video streaming in full HD resolution might require 1.5 Mbps, however it might be reasonable to assume full-HD is not required on the WiFi connected mobile devices (with small screens) and thus some lower limit would be appropriate. A starting point might be to try a per-user download limit of 500 Kbps and an upload limit of something less, then fine tune based on actual user experience on the network. In general a lower per-user bandwidth limit will enhance the network's ability to support real-time applications such as AudioFetch.

5. DISCOVERY - Introduction

"Discovery" refers to the mechanism, and in particular to the network traffic, used by the AudioFetch app on client devices to locate the AudioFetch APB devices on the network.

Automated discovery avoids any need for end users to manually enter information into their mobile devices. Instead, when a user opens the AudioFetch app, it executes the discovery process to determine the IP addresses of all APBs present on the facility's network. Once the IP addresses are known, the AudioFetch app is able to request and receive an audio stream.

Primary discovery method is SSDP protocol, a widely accepted standard. SSDP relies in part on multicast traffic (detailed in next section), however many network installations restrict multicast traffic on the public WiFi in which case SSDP will not work. The AudioFetch system employs several fallback discovery mechanisms, each having a specific set of network traffic requirements. The

fallback mechanisms operate automatically and the network administrator simply needs to ensure the traffic from at least one is not blocked (between APB and mobile clients). Details of the required traffic for each mechanism are presented in the next section.

A few examples will help illustrate why it is important to consider how the discovery will work when deploying an AudioFetch system.

A typical easiest plug-and-play scenario can be illustrated using a consumer-grade Wireless Router with integrated switch:

- Primary WiFi SSID is used
- Switch ports are part of the same subnet as WiFi
- APB is connected to one of the switch ports
- WAN port is connected to an ISP modem, or perhaps an "upstream" LAN to provide internet access
    - Firewall on the WAN will not impact AudioFetch – because all things AudioFetch are on the same LAN subnet
- Default settings are used – typically no restriction on traffic within the LAN
- In this case the primary SSDP protocol will work – multicast traffic can occur without restriction between clients and APB since all are on the same subnet with no traffic restrictions.

A simple but often-encountered problem scenario can be illustrated using the same equipment, with one simple configuration difference:

- Guest WiFi network is enabled and required for customers to use
    - This may be desirable to restrict customer access to business assets such as printers, NAS devices, etc.
- The Guest WiFi is a separate SSID but more importantly is a separate LAN/subnet
    - APB device connected to one of the switch ports will be on the primary LAN, which is a different subnet
- The Guest WiFi is also typically configured for restricted access – devices on the Guest LAN cannot access devices on the primary LAN
- This means that the multicast traffic required by SSDP discovery will not be passed between the mobile clients and APB, SSDP will fail.
- In fact all of the fallback methods described in the next section will fail as well, unless the Wireless Router can be configured with rules to allow specific traffic between Guest and primary LANs (this would be unusual in a consumer-grade router).

The Guest Network scenario is actually difficult to overcome within a single consumer-grade Wireless Router. The easiest solution (while retaining the security advantages) is to employ a second Wireless

Router for the customer-accessible WiFi, using the primary SSID on that router and also connecting the APB to one of its switch ports so that the APB operates on the same subnet as the mobile client devices. This secondary Wireless Router's WAN port can be connected into the main business LAN for internet access, and its firewall can be configured for the desired level of security/isolation. The only complication is that two independent Access Points now exist, so they should be carefully configured to operate on different channels with appropriate power levels, etc.

A simple commercial-grade network with a single Router and Access Point has very similar considerations – if a single LAN is created (everything on one subnet) then AudioFetch is usually plug-and-play, but if multiple/guest SSIDs are used or other advanced security features enabled, then some configuration settings might need to be adjusted, or some of the techniques in the examples below might need to be used.

Larger networks employing enterprise Access Points (Ubiquiti Unifi for example) are often configured to restrict traffic between the mobile client devices, as a security measure. This results in a restriction of multicast traffic originating from mobile clients which breaks SSDP discovery. However such Access Points can usually be configured to allow outbound multicast traffic to the mobile clients (not a security risk) in which case the SSDP Fallback discovery method described in the next section will work.

Larger networks often operate with more than one subnet. Most networking equipment by default will not route multicast traffic between subnets. If not possible to manually configure rules allowing the required traffic for SSDP between subnets, there are a few options to resolve:

If operating with Virtual LANS, it might be possible to configure the specific switch port being used by the APB to operate on the same VLAN as the WiFi LAN, this will usually allow SSDP to function correctly.

Finally, the DNS Discovery method can be used as detailed in the next section. This requires advanced router capability (local DNS server with capability to add local host entries) but is supported on a variety of commercial-grade networking equipment. For example, Cisco and Linksys small business routers support a "DNS Local Database" feature, Ubiquti routers support "DNS Forwarding", MikroTik routers support "DNS Cache", Linux-based routers can employ "DNSmasq," etc. This type of local DNS server feature is usually not found in consumer-grade equipment.

6. TRAFFIC DETAILS – DISCOVERY

The AudioFetch App automatically attempts each of the discovery methods listed here, in the order listed, until one succeeds. Therefore a network need only support one. All traffic listed for a

particular discovery method must be allowed on the network in order for that discovery method to work.

Discovery Method: SSDP (preferred)

SSDP is a standard discovery mechanism employed by many types of equipment. Network traffic requirements for SSDP are:

Traffic direction: Mobile Device (AudioFetch app) → AudioFetch APB device:

Protocol	source port	dest port
UDP multicast (239.255.255.250)	p_1 *	1900
TCP unicast	p_2 *	80

Traffic direction: AudioFetch APB device → Mobile Device (AudioFetch app):

Protocol	source port	dest port
UDP unicast	1900	p_1 *
TCP unicast	80	p_2 *

* p_1 and p_2 indicate port numbers which have been chosen by the mobile device's operating system and typically are different each time the app runs

Comments on SSDP: This mechanism requires the network to allow multicast traffic from WiFi connected devices onto the wired portion of the network (where the AudioFetch APB is connected). In a complex network this traffic may also have to span subnets. Both of these are often challenges for larger networks where multicast traffic can be restricted for security reasons, and/or it is difficult to configure routers to pass multicast traffic between subnets.

Note about SSDP discovery on SECONDARY (management) port on AudioFetch Signature units with dual Ethernet ports: On these units the upper Ethernet port is called SECONDARY and is used to access the Doghouse configuration pages, which are not available on the lower/PRIMARY Ethernet port (used only for audio streaming). Purpose is to isolate and prevent access to the Doghouse by users. Discovery on the PRIMARY port works as described above. Discovery on the SECONDARY port works as described above except the UDP port number is 1901 instead of 1900.

Discovery Method: SSDP FALLBACK

SSDP FALLBACK is a modified version of SSDP where multicast packets from the WiFi-connected mobile devices are not required. Instead, the AudioFetch APB device (connected to the wired portion of the network) transmits unsolicited packets to a multicast address once every second. The idea here is that network security often allows multicast traffic from the wired portion onto the WiFi portion, but not vice-versa. After the mobile device receives the initial multicast packet from APB device, the balance of the discovery follows SSDP:

Traffic direction: Mobile Device (AudioFetch app) → AudioFetch APB device:

Protocol	source port	dest port
TCP unicast	p_2	80

Traffic direction: **AudioFetch APB device → Mobile Device (AudioFetch app):**

Protocol	source port	dest port
UDP multicast (239.255.255.250)	1900	1900
TCP unicast	80	p_2

Note about SSDP FALLBACK discovery on SECONDARY (management) port on AudioFetch Signature units with dual Ethernet ports: On these units the upper Ethernet port is called SECONDARY and is used to access the Doghouse configuration pages, which are not available on the lower/PRIMARY Ethernet port (used only for audio streaming). Purpose is to isolate and prevent access to the Doghouse by users. Discovery on the PRIMARY port works as described above. Discovery on the SECONDARY port works as described above except the UDP port number is 1901 instead of 1900.

Discovery Method: BROADCAST FALLBACK

BROADCAST FALLBACK employs broadcast packets from the WiFi-connected mobile devices in an attempt to circumvent situations where multicast doesn't work, somewhat of a "last ditch" kind of effort by the app:

Traffic direction: **Mobile Device (AudioFetch app) → AudioFetch APB device:**

Protocol	source port	dest port
UDP broadcast (255.255.255.255)	30981	30981
TCP unicast	p_2	80

Traffic direction: **AudioFetch APB device → Mobile Device (AudioFetch app):**

Protocol	source port	dest port
UDP unicast	30981	30981
TCP unicast	80	p_2

Note about BROADCAST FALLBACK discovery on SECONDARY (management) port on AudioFetch Signature units with dual Ethernet ports: On these units the upper Ethernet port is called SECONDARY and is used to access the Doghouse configuration pages, which are not available on the lower/PRIMARY Ethernet port (used only for audio streaming). Purpose is to isolate and prevent access to the Doghouse by users. Discovery on the PRIMARY port works as described above. Discovery on the SECONDARY port works as described above except the UDP port number is 30982 instead of 30981.

Discovery Method: mDNS one-shot

mDNS is a standard discovery mechanism employed by many types of equipment. mDNS "one-shot" is a subset of the mDNS protocol described in RFC 6762 section 5.1. Network traffic requirements for mDNS one-shot are:

Traffic direction: **Mobile Device (AudioFetch app) → AudioFetch APB device:**

Protocol	source port	dest port	
UDP multicast (224.0.0.251)	p_1 *	5353	(source port must not be 5353)
TCP unicast	p_2 *	80	

Traffic direction: AudioFetch APB device → Mobile Device (AudioFetch app):

Protocol	source port	dest port
UDP unicast	5353	p_1 *
TCP unicast	80	p_2 *

* p_1 and p_2 indicate port numbers which have been chosen by the mobile device's operating system and typically are different each time the app runs

Discovery Method: DNS LOOKUP

Some networks simply do not allow any multicast or broadcast traffic in either direction between WiFi connected mobile device and the AudioFetch APB. This is especially true in situations where the two may be connected to different subnets, making it difficult to route multicast traffic between the subnets. Therefore, the DNS LOOKUP method may be employed which does not require any multicast or broadcast traffic.

It does however require a manual entry be added into the network router's local DNS server hosts list. Not all routers (or gateways that provide router functions) offer a local DNS server function – check your router documentation. For example on many Cisco small business routers, the local DNS server capability is called "DNS Local Database" in the user manuals and can be configured from the router's GUI configuration interface. Other routers may require manual configuration through a command line interface (Ubiquiti). Some Linux-based routers might contain a software package called "dnsmasq" which provides this capability.

How DNS LOOKUP works in an AudioFetch system:

1. AudioFetch hardware device is assigned a static IP address on the network.
2. AudioFetch app running in mobile device sends a DNS query for the local host:
   "audiofetch.localdomain.localextension"
   A DNS lookup does not require any multicast/broadcast traffic and is almost always supported even on highly secured networks.
3. "localdomain.localextension" would be the connection-specific DNS Search Suffix provided to the mobile device by the DHCP server for the WiFi network. Easiest way to find this is connect a PC to the WiFi network and then run the command "ipconfig" from a command prompt, it should provide the "Connection-specific DNS Suffix" for the Wireless LAN adapter.
4. A local DNS hosts lookup table must be supported within the network (usually in the Router) and configured with an entry for this hostname (audiofetch.localdomain.localextension), which specifies the exact IP address of the AudioFetch APB connected to the network, the IP address does not need to be within the same subnet as the WiFi connected mobile device, however the network needs to be configured to route/allow traffic between the mobile devices and this IP address.
5. Note that some DHCP servers are not configured to provide a DNS Search Suffix, in these cases the DNS query sent by the mobile device will simply be:
   "audiofetch"

in which case the local DNS hosts lookup entry would need to be: audiofetch Best case is to configure the local DNS hosts lookup to return responses for either type of query:
"audiofetch.localdomain.localextension"
"audiofetch"

6. Mobile device uses the IP address returned in the DNS response to complete the discovery process.
7. Where multiple AudioFetch boxes are deployed in an installation, a larger set of entries in the local DNS hosts lookup table must be used. For a 2-box installation, the following entries are required:
"audiofetch" (router should respond to either audiofetch or audiofetch.localdomain.localextension)
"audiofetch-2" "
For a 3-box installation:
"audiofetch" (router should respond to either audiofetch or audiofetch.localdomain.localextension)
"audiofetch-2" "
"audiofetch-3" "
For a 4-box installation:
"audiofetch" (router should respond to either audiofetch or audiofetch.localdomain.localextension)
"audiofetch-2" "
"audiofetch-3" "
"audiofetch-4" "
Note that there is no particular requirement to match a specific DNS name with a specific AudioFetch box/IP-address on the network, the only key requirement is that there be a unique entry for each and every AudioFetch box.

Here is what needs to be done to configure your network correctly:

1. Add a static IP address reservation in your DHCP server settings for each AudioFetch hardware device connected to the network.
    a. NOTE: when the AudioFetch hardware device(s) request IP addresses from the DHCP server, they will provide their own host names. THESE SELF-REPORTED HOST NAMES ARE NOT INVOLVED WITH THE DNS DISCOVERY PROCESS, PLEASE IGNORE THEM.
2. In your network router's Local DNS Lookup table, add the host names described in items 5-7 just above (add the appropriate number of host names depending on how many AudioFetch hardware devices connected to your network), and add the associated static IP addresses for each.

Traffic direction: Mobile Device (AudioFetch app) → AudioFetch APB device:

Protocol	source port	dest port
DNS query (won't describe here)	(as usual)	(as usual)
TCP unicast	p_2	80

Traffic direction: AudioFetch APB device → Mobile Device (AudioFetch app):

Protocol	source port	dest port
DNS response (unicast)	(as usual)	(as usual)
TCP unicast	80	p_2

7. TRAFFIC DETAILS – AUDIO STREAMING

Audio streaming occurs after successful completion of discovery and primarily requires one-way UDP-unicast network traffic between the AudioFetch APB device and the WiFi connected mobile devices. Occasional bi-directional traffic is required for channel selection and keep-alive. None of this traffic requires multicast or broadcast, all audio streaming traffic is unicast:

Traffic direction: **Mobile Device (AudioFetch app) → AudioFetch APB device:**

Protocol	source port	dest port	
TCP unicast	p_3	6971	Control & keep-alive

Traffic direction: **AudioFetch APB device → Mobile Device (AudioFetch app):**

Protocol	source port	dest port	
TCP unicast	6971	p_3	Control & keep-alive
UDP unicast	6970	6970	This is the audio stream

8. TRAFFIC DETAILS – DOGHOUSE CONFIGURATION

Access to the AudioFetch "Doghouse" web pages (to configure APB operation) requires the following traffic:

Traffic direction: **Mobile Device (AudioFetch app) → AudioFetch APB device:**

Protocol	source port	dest port
TCP unicast	p_2	80

Traffic direction: **AudioFetch APB device → Mobile Device (AudioFetch app):**

Protocol	source port	dest port
TCP unicast	80	p_2

Note that discovery is not required for access to the Doghouse configuration web pages, if the IP address of the APB device is known then it may be entered directly into a web browser for access. However the provided AudioFetch Doghouse Discovery software does rely on the standard discovery methods outlined in this document (currently it does not support the DNS method).

9. TRAFFIC DETAILS – ONLINE FIRMWARE UPDATES

Access to online firmware updates requires the following traffic:

Traffic direction: **AudioFetch APB device → Internet:**

Protocol	source port	dest port

DNS (UDP packets)	53	53
TCP unicast	63713	80
TCP unicast	63714	80

Traffic direction: **Internet → AudioFetch APB device:**

Protocol	**source port**	**dest port**
DNS (UDP packets)	53	53
TCP unicast	80	63713
TCP unicast	80	63714