Frequently Asked Questions

1 Revision History

Revision	Date	Description
1	03/18/2013	Initial Release
2	10/21/2013	Additional questions added

2 General SmartMesh FAQs

2.1 Basics

Q: How often can my sensors send data?

A: When the mote joins, it will be given enough network resources to stay connected to the network. The sensor can request extra network resources in the form of a service request. That tells the network how much traffic this sensor is expected to offer to the network. If the sensor is configured to report a measurement every 60s, then the sensor should submit a 60s service request. The network will inform the sensor whether the service has been granted. If EVERY sensor in the network is configured to send data at the same rate, it may be more convenient to set the network-wide base bandwidth at the manager and not do any service requests at the edge.

Since all traffic flows through the AP, the maximum theoretical capacity of the network is 1 packet/slot, with ~90 bytes of payload space per packet. In practice, several factors serve to derate this value:

- Slot size differs between SmartMesh IP (7.25 ms) and SmartMesh WirelessHART (10 ms), so the total packets/s available differs.
- The manager sets aside ~15% of the total bandwidth for downstream communications, advertising, and neighbor discovery
- We choose a default provisioning factor of 3x (3 links assigned for every one used) to allow for the large expected variation of path stability

Consult the table below for the total system egress bandwidth. This total bandwidth needs to be shared across all the motes in the network. So if there are 24 motes in an LTP5901 network, then you can send 1 packet/s, 2 packets/s for 12 motes, etc.

Manager	#AP RX Links	#AP RX per sec	Pkt/s @ 3x prov	Pkt/s @ 2x prov
LTC5800/LTP5901/LTP5902-IPRA	150	72.9	24.3	36.4
LTC5800/LTP5901/LTP5902-IPRB or IPRC + External RAM	223	108.3	36.1	54.2
LTP5903-WHR	737	72.0	24.0	36.0

Q: Is it better to send small payloads or big payloads?

A: It is always better to send only the data that is needed. Transmitting one small payload consumes less power than transmitting one large payload. But sending 10 small payloads is more costly than sending one large payload. Concatenating measurements and sending them less frequently is more efficient. The cost is that the older measurements are more 'stale'. For trending data, this might be useful. For real-time monitoring, the most valuable data is typically the freshest data. If you have an event that requires you to send 8000 bytes of data, it is far better to send 100 80-byte payloads than it is to send 1000 8-byte payloads.

Q: My data messages are larger than a single payload. Does the network automatically fragment and defragment my payloads?

A: No. Fragmentation and reassembly is the responsibility of the application on each end.

Q: Can a mote's firmware be updated wirelessly? Mote firmware and sensor firmware?

A: The mote firmware can be programmed over the air (OTAP). There is built in manager support for this in SmartMesh WirelessHART. In SmartMesh IP, an external application is needed to control OTAP - we provide a reference implementation called OTAPCommunicator. There is no native OTAP support for sensor firmware, but many customers have implemented sensor OTAP simply using the network to transport those payloads.

Q: My sensor requested a 30s service, and it was granted. Will something bad happen if I send data once every 60s instead? What if I send data once every 10s instead?

A: There is no validation at the mote holding you to your service request. Nothing bad will happen if you send less often than the service level you requested. In fact, some sensor applications will ask for extra service deliberately in order to achieve lower latency. If a sensor asks for a 10s service and sends data once every 60s, that data will be delivered more quickly than if the sensor had only asked for a 60s service. Sending more data than your service allows may cause congestion in the network. That congestion may happen locally at the mote, or it may happen upstream at a router that is routing more traffic than the services indicate should be expected. In general it is not a good practice to send more data than you have been granted.

Q: Do all my sensors have to publish data at the same rate (i.e. get the same services)?

A: No. Each sensor can send unique service requests. One sensor may ask for a 60 s service, and another may ask for a 10s service and another can ask for a 1 s service. The network manager will lay in links to accommodate all granted services. Heterogeneous services are totally supported.

Q: Is there a limit to the services my sensors can ask for?

A: Yes. Services are not an infinite resource. With default manager settings, sensors will start getting denied services when the total of all services exceeds the maximum throughput of the manager (refer to individual product documentation). For example, for the manager that supports 25 packets/s, you should not exceed more than 25 motes publishing at 1 second each. So, if sensors send data once per second, you should not plan on using more than 25 motes per manager. Similarly, if you want to build a 250 mote network, then 10 s is the fastest service you could expect to get for all motes.

There is also a limit on the fastest service an individual mote can expect to receive. With default settings this is ~100 ms. A shared bandwidth backbone feature allows packet latency to fall well below this number for shallow networks, but it is designed for infrequent data, not routine publishing.

Q: Can I guarantee my power consumption by setting my data reporting rate?

A: No. A sensor that sends data less often will consume less power, but the amount of data forwarded from children in the mesh also has a large impact on total power consumption. If you have a power supply that has a hard limit (like a scavenger circuit), you can use the mote API to set power source information of the device, but that will limit the manager's ability to build the best possible mesh, and those limits should be used with care.

Q: My sensors send data once an hour. Can I turn off the network and turn it back on when my sensors want to send data and save power?

A: This is generally not recommended. The motes often consume less power in the network than they do when searching for the network. Furthermore, the motes being on and maintaining connection to the network means that you can easily implement alarm type reporting over and above your very infrequent regular updates. Also you have the ability to send downstream commands for actuation, configuration, etc., at minimal extra power. An IP mote can live in a network AND report every 30 seconds for an average current under 10 µA.

On the other hand, if you have an application such as container tracking where the whole network (on a shipping container) may not need to be monitored for days at a time, there may be other overriding considerations.

Q. Why does my network have a single parent mote?

A: All networks will have one mote with the AP as its only parent. This is required to prevent timing and data loops in the network. A network with more than one single parent mote indicates that you need to either add a mote or move a mote to improve the connectivity.

Q: Are out of order packets received in a wireless mesh network?

A: Yes, a packet flowing upstream in a mesh network can follow any one of many sets of redundant paths to the manager. The application layer is responsible for reordering received packets if necessary, which can be done using the accurate timestamps in each packet.

Q: How can I verify the integrity of a wireless mesh network?

A: In order to ensure the integrity of a wireless mesh network, every mote must have at least 3 *good* neighbors with path stability of 50% or better. Path stability data is only available on paths with active upstream links. If there aren't 3 paths being actively used by the mote and shown with a path stability in the path statistics for that mote, then the measured RSSI on that path can be used as an approximation. In that case, an RSSI higher than -75 dBm is considered a good neighbor path.

Path statistics can be returned through the API of the Manager for every mote. This allows the client to write software to verify network integrity. It's also a good idea to give the network at least an hour to discover all the usable paths before making a decision to add repeaters.

Q: Why are the motes slow to join the manager? Why does my network take so long to form?

A: The most common reasons for slow join time are:

1. Motes that have joined the manager are suddenly moved to a new location. When this happens, the mote must detect that all its paths have failed, which can take several minutes. The mote will then reset itself and then attempt to join the network again at its new location. This join time takes more time. To fix this problem, simply reset the mote any time it is moved to a new location. The mote will immediately try and rejoin the manager.
2. Poor path stability - the presence of interference, or motes that are too far apart can result in the message sequence to join the mote timing out, and the mote will need to reset and attempt to synchronize and join again.
3. Mote search duty cycle is low - See “How can I make my motes join faster?” below.
4. The size of the network is large (more than 100 motes) and is deep (covers a large area). Typical join rate for larger networks is 1 min/mote.

Q: How can I make my motes join faster?

A: Increase the mote's join duty cycle using the mote’s API command *setParameter*<joinDutyCycle> to be up to 255 (100%). The default setting is 5% for WirelessHART motes, and 25% for IP motes, which conserves power when trying to join. High duty cycle search will consume mA of current, which may be undesirable, particularly if motes are deployed in advance of a manager to join. A developer working at their desk may have to watch the mote on their desk join 20 times in a work day. That person needs the mote to join fast. A salesperson doing a demonstration in front of a board room of people probably also wants joining to be as fast as possible. For most real deployments, though, it may be wise to save a lot of power in search and accept that joining might not be instant.

Q: Why would a mote suddenly reset?

A: A mote will reset if its /RST hardware pin is asserted or it receives a reset command from the sensor processor. A mote will also reset in the rare event that it loses communication with both its parents. If you are seeing a mote reset many times, pay close attention to how many neighbors have RSSI > -75 dBm and could potentially become good parents.

If you are seeing that a mote appears to have good neighbors and parents but still resets, you may have an RF interference problem. This can be verified by calculating a path stability vs. RSSI curve for any troubled motes. The ratio should be relatively consistent for many deployments. If you see a sudden drop in path stability relative to RSSI, you should suspect RF interference is causing the problem. Wireless sniffers can be used to measure in-band interference.

Q: What happens when a mote has the wrong credentials to join a network?

A: In the case of a wrong Network ID, the mote will continue in the “search” state forever. It is looking for an advertising packet with a Network ID matching its configuration, which may never come. After a period of time, the microprocessor may want to reduce the join duty cycle time to conserve power consumption. This can be done by using the mote API command *setParameter*<joinDutyCycle>.

If the mote has the correct Network ID but the wrong join key or is not on the ACL, it will attempt to join but the manager will not respond. The mote will reset itself after several minutes and the microprocessor will get a boot-event.

If the mote's flash has been erased, its join counter will be reset to 1. Since the manager keeps a history of each mote's joining, it will interpret the new joins as a replay attack and not accept the mote. To fix this problem, you should delete the mote from manager's ACL and re-add it.

The mote serial API command *getParameter*<moteStatus> can be used to verify the state of a mote. A mote with the wrong Network ID will always be in the **Searching** state. A mote with the wrong join key or join counter will continue past the searching state but will never reach the **Operational** state.

Q: Why does a mote have two parents that are far away, when there are other motes that are closer?

A: The network manager's algorithms are designed to build the best network it can, using paths that are 'good enough'. It does not make decisions that are locally greedy. It is far better to send data two hops (even if the packet success rate is only 80% at each hop) than it is to send data 5 hops (even if every hop is 100% packet success rate). The manager will tend to make the 'flattest' network possible (fewest average hops).

Q: Can a SmartMesh manager be accessed via an IP address?

A: SmartMesh WirelessHART managers support connection by Ethernet, allowing access to the manager API via the manager's IP address (IPv4), but only the packaged managers have an RJ45 Ethernet jack. It is up to the system integrator to provide network access on an embedded WirelessHART manager. SmartMesh IP managers are serial devices - they get internet connectivity (IPv6) through a border router - Dust has provided a reference implementation of a border router to demonstrate this capability.

Q: Can a mote be accessed via its IP address ?

A: When a low power border router is present, SmartMesh IP motes can be directly addressed by their IP address. SmartMesh WirelessHART motes do not have an IP address, so while they can be addressed through manager APIs, they cannot be addresses directly via an IP address.

2.2 Advanced Topics

Q: What resources are used to maintain the network, collect info on link health, assign and re-assign links to motes, etc?

A: Each mote generates three health report packets every 15 minutes. These packets summarize the information the manager uses to maintain and optimize the network. Downstream, we typically only see a few packets per hour required to repair and optimize the network.

Q: Is a single superframe/slotframe configuration used across an entire network?

A: In general use, all motes are assigned all superframes/slotframes. For SmartMesh WirelessHART pipes, the endpoint is specified by the user, and the manager handles setup/teardown of the superframes and links. In SmartMesh IP, the manager does not give low-power motes (info given by the mote at join) the advertisement slotframe, but it is identical for all other motes.

Q: If there are multiple networks with the same networkID in the same radio space, which network will a mote join and how can this be managed?

A: A mote will synchronize to the first advertisement it hears, and it will attempt to join the network that sent this advertisement. Mote allocation can be controlled by using ACLs on each manager, however, motes will still try to join whichever network they hear first. If a mote is not on that network's ACL, it will reset and start listening again.

Q: If line power is available, how does this affect the opportunity to create low-latency communications?

A: The low-latency backbone can be activated to reduce the latency throughout the network without increasing the energy draw at battery-powered devices. The more line-powered devices are available, the better the performance of the backbone. However, the backbone is not appropriate for every use case, see "Application Note: Using the Powered Backbone to Improve Latency" for more details.

Q: If a customer has physically lost a mote, can our system help find it?

A: While we do not provide a mechanism for querying the location of a mote, triangulation based on the strength of used paths (available through mote statistics) may help locate a device.

Q: What are the performance issues for building deep networks, e.g., 25-100 motes in a straight line down a pipe. Applications like steam distribution and corrosion monitoring for short runs of oil line come to mind.

A: The fundamental issues are being able to form a mesh and the total hop depth of the network. To build a good mesh, we want every mote to be within range of as many other devices as possible. The more the better, but the absolute bare minimum to have a good chance for the network to form at all is 3. Every mote should have a good quality connection to at least three other motes. If we don't know what the range of the motes will be, we recommend planning on that range being 50 m, meaning the spacing along the pipeline MUST be at a spacing of 16.7 m or less. Only then will a mote on the end of the chain have 3 devices within range. If the range is better than that, then the spacing can be greater. If the range is worse, the spacing must be closer. The pipeline monitoring case and the perimeter monitoring case are particularly challenging for this reason. You have to know from the outset that you are willing to place devices at a spacing that is much smaller than the range of the devices. If a sensor is needed every 10 ft, then range is not a constraint. But if a sensor is only needed every 200 m, then the user must be able to guarantee 600 m range (i.e. motes have unobstructed line of sight communications well above the ground).

We encourage using SmartMesh IP for deeper networks, see "Application Note: Building Deep IP Networks" for more details. We do not recommend using SmartMesh WirelessHART systems when the network will be deeper than 8 hops.

Q: How long does it take for a network to form based on x number of motes/managers? How about the time to add a single mote to an existing network?

A: This depends to some degree on the join duty cycle – what fraction of time an unsychronized mote spends listening. One minute per mote is our rule of thumb. For small networks it can be longer, since individual mote timeouts dominate. If advertising is on, a single mote should join within 2 minutes. If advertising is off in a SmartMesh IP network, no motes will ever join or rejoin. In SmartMesh WirelessHART, advertising cannot be fully deactivated, but when running in slow mode to conserve power, it could take a single mote ten minutes to join an existing network.

In larger systems running on multiple managers, each network will form in parallel. As an example, a deployment consisting of two 100-mote networks should fully form in the same time as a single 100-mote network in the same conditions.

Q: How is message routing handled? If there are multiple paths, can a message reach an intermediate mote or the destination mote through multiple paths, and if so does the system handle duplicate messages automatically or does the user application have to handle this?

A: There are two types of routing in a Dust mesh network – graph routing, where a graph ID is used to track where a packet should go on a per-hop basis, and source routing, where an explicit list is used on a per-hop basis. In all of our products, we use graph routing for all upstream packets and downstream broadcast (to all motes) packets, while source routing is used exclusively for downstream unicast packets (i.e. those packets destined for a specific mote).

Messages are duplicated when a forwarding mote receives a message, but the sender does not hear the acknowledgement and retries through another parent. These duplicates are filtered by the manager. The manager keeps a count of the number of duplicates received from each mote. If an intermediate mote receives a duplicate upstream or downstream packet due to it being part of multiple routes to the destination, the mote will delete the duplicate packet if the original copy was heard recently enough.

Q: Does the network assign preferred paths so that duplicate paths are not used for message delivery, and if so is this done dynamically by the motes based on what they know about their neighbors, or handled by the network manager as a configuration process?

A: The manager will assign more links to the better quality, closest to manager ("preferred") path of each mote, so that on average the bulk of transmission attempts will happen on that path. The mote will simply use whichever link comes up next, so the non-preferred path will eventually be used, and could be used frequently if message generation happens to occur right before that link. The manager is constantly evaluating the health of the network and will make link changes as path qualities change.

Q: How is the data from each manager controlled centrally in a multiple-manager deployment?

A: The integrator is responsible for merging data from different managers. A single "gateway" can subscribe to data notifications from multiple managers on a LAN and present that data to a customer application as a single dataset.

Q: If a truck with a mote on it comes near a network that it has the credentials to join, how long must it stay near the network for that mote to join and send/receive data ? What are the considerations when selected nodes are moving in the networks? How quickly can a mote be moving?

A: In this scenario, we recommend keeping advertising active in the network, then the mobile mote should join in less than two minutes. At this time, SmartMesh networks do not support mobility where motes are moving > ~1 m/s: once a mote has moved beyond the range of its neighbors it has to leave and rejoin the network. In order to speed up the process as much as possible, the customer should reset the mobile mote as soon as it has reached each new destination.

2.3 Security

Q: If motes are located on two different end customer's premises, and those customers are concerned about the privacy of their specific data in the network, what can we tell them about SmartMesh security that will help them feel that their information is secure?

A: By using ACLs and per-mote unique join keys, the plaintext traffic in one network cannot be intercepted by a neighboring network (whether it is the customer's or a neighboring network), nor can that manager "steal" motes to try to get them to report their data to it instead of the intended manager.

2.4 Product Compatibility

Q: Are Dust motes ISA 100 compliant?

A: No, SmartMesh WirelessHART is compliant to the WirelessHART (IEC62591) standard, which is different from and incompatible with the ISA100 standard.

Q. Is there a Dust product for the 900 MHz ISM band?

A. There are no Dust products operating in the 900 Mhz band. Most customers require worldwide operation and the 900 MHz band is only defined in North America.

Q. Product X is compliant to IEEE 802.15.4 - does that mean Dust products interoperate with it?

A. For two devices to interoperate, they must understand each other on all levels of the protocol stack. 802.15.4 covers the two lowest levels of the stack - the physical layer (PHY) which describes how the radio operates (frequency, modulation, etc.) and the medium access control (MAC) layer, which describes how one devices transfers data to another device. It does not cover the content of messages, routing, or many of the possible settings that the MAC and PHY can use - unless they are driven the same way, even two 802.15.4 devices may not be able to communicate. This is analogous to a French and English speaker both using the same set of consonants and vowels, and using sentences made of words, but still not being able to understand each other.

In contrast, WirelessHART is an example of a complete stack specification - it describes operation of the PHY and MAC, routing and security, and the format and content of application layer messages. Any two devices compliant to WirelessHART should be able to interoperate.

3 SmartMesh IP

Q: How is advertising handled on a SmartMesh IP network?

A: The LTC5800-IPR-based managers, unlike SmartMesh WirelessHART managers, do not contain logic to activate/deactivate advertising after a mote is lost or a timeout occurs. There is a *setAdvertising* API that allows the application to implement logic to control advertising in response to network conditions.

Q: On my SmartMesh IP network manager, the mote id to MAC address mapping changes every time I reboot the manager. Why do the mote IDs of my motes change?

A: Every time you reboot the SmartMesh IP manager, the mote list is erased. The first mote to join will be the AP mote, and will be mote ID 1. The next mote will be mote ID 2, and so on. Mote ID is just a convenient nickname that is used to conserve bytes in messages, but should not be considered persistent across network resets. The MAC address is the persistent identifying name of a particular mote.

Q: What ports do I use?

A: The IP manager supports any UDP port number, but some port numbers are special in that they can be compressed such that they only take up 1 byte in the header, rather than the normal 4 bytes (2B source, 2B destination). These ports are in the range 0xF0Bx, where x goes from 0-F. When using an arbitrary port for either source or destination (carried as 2B), then the range of compressible ports for the other expands to 0xF0xx, at the cost of 1B. Refer to the The SmartMesh IP Mote for more information .

Q: For the SmartMesh IP network there is a claim that the network is configurable to support multiple report rates and message latencies within a single network. How does this work and how large of a range of report rates and message latencies are possible?

In addition to the homogeneous bandwidth available through the base bandwidth setting, motes can request *services* - individual heterogeneous requests for BW (with optional latency target). The manager assigns links as needed to grant the service, or deny it if it cannot. Range of report rates is a function of the depth of the network (which determines how often a message is forwarded) and number of motes. Small numbers of motes can report at < 500 ms intervals while the rest of the network can publish at 10 seconds or slower. Obviously a service to reach a latency target at a mote three hops deep results in more links at the ancestors of the target mote so their latency will decrease too.

Q: What is the difference between IPv4 and IPv6?

A: The number refers to the version of the Internet Protocol standard - there are a number of differences but the important ones for our products are:

- Address size - v4 addresses are 4 bytes long, typically displayed as 4 numbers separated by dots, e.g. 74.125.224.225. Unfortunately, the world ran out of v4 addresses recently. v6 addresses are 16 bytes long, allowing for every conceivable device on the planet to have its own unique address. Unfortunately, many homes and businesses have not switched over to using the newer protocol yet, but will likely do so in the years to come.

- Multicast groups - v6 allows for defining groups of recipients - something not found in v4.
- Packet size - v4 supported 576-byte packets. v6 supports 1280-byte packets. In either case, we need to fragment IP packets to fit inside our wireless packet, which is limited to ~90 bytes (protocol dependent) of user payload per packet. Current products don't support fragmentation within the network, so for now users need to keep packets small.

Q. With an Evaluation kit, how can I get per packet latency and the number of hops traveled by that packet?

A: Using the manager CLI, you can use the "trace stats on" command to get a print of per packet statistics. An example print shows

```
STAT: #5: ASN Rx/Tx = 59692/59686, latency = 43, hops = 2
```

This packet traveled 2 hops. It was generated by the mote at ASN 59686 and received by the manager at ASN 59692. This difference of six slots corresponds to a latency of 43 ms.

Q. In Stargazer, how is mote latency calculated?

A: Stargazer monitors the packet notifications published by the manager. Each notification contains the originating mote and the latency for the packet. Stargazer averages this value on a mote-by-mote basis to obtain an estimate of mote latency.

Q: Is it possible to simply connect an external A/D (that would measure something like a thermocouple) and not have a third party micro-processor involved?

A: On SmartMesh IP motes running in Master mode, the resident application can be configured to periodically poll the internal multi-channel A/D using the On-chip application protocol. See the SmartMesh IP Tools Guide for details.

4 SmartMesh WirelessHART

4.1 Network Size

Q: Do we have any examples of large scale WirelessHART mesh networks (250 to 500 motes)?

A: Yes, there are number of customers doing deployment with hundreds of devices.

4.2 Network Console - nwconsole

Q: How many nwconsole sessions can be going on at the same time? If there are too many, how do I connect?

A: Three concurrent sessions are possible. There is no mechanism to notify one of these sessions that another user wants a session. If multiple people need to review the status of the network, it is perhaps better to write an application that queries the Manager API and publishes it to a web host designed to support numerous concurrent connections.

Q: What does the PkLost field mean in the manager's network statistics?

A: The mote increments a security counter on each packet it generates. If the manager gets packets with counters k and k+2, it knows that it must have missed a packet with k+1. This does not result in the session breaking, it just results in the manager incrementing the PkLost (= packets lost) field for that mote. The reliability target of >99.99% means that seeing any packets lost is uncommon. One way for a packet to get lost is when a routing mote accepts a packet from its child and then resets; in this case, the manager will not be able to receive that packet and will increment the PkLost counter for the child.

Q: What does "-" mean when looking at the manager network statistics?

A: It means that no data has been received for the current health report interval for that statistic. The interval in question could either be the lifetime, a day, or a 15 minute interval. Since health reports on the motes and statistics intervals on the manager are not perfectly aligned, this typically occurs when the interval finishes without having received a mote health report. All health reports are still accumulated as they arrive though, so no information is lost in these cases.

A second possibility is that the path in question was not used during that interval. This happens most frequently when the path is used only for downstream communication.

A third possibility in a congested network is that a mote had a full packet queue when it came time to generate the health report. In that case the mote will not generate the report. The next report it generates, 15 minutes later, will then contain two intervals worth of statistics.

Q: What is the difference between ABPower & BAPower in RSSI path statistics?

A: ABPower is the average (in dBm) Received Signal Strength Indicator (RSSI) - also called Received Signal Level (RSL) of packets received by mote B from mote A. BAPower is the RSL of packets received by mote A from mote B. These values are reported by the motes in their neighbor health reports. The power values AB/BA should be within 6 dBm of each other - if not, it can indicate a problem with hardware, especially if we see asymmetries on all paths involving the "bad" mote.

The access point does not generate health reports, but it does report RSL to the manager on a per packet basis. The manager averages these per-packet values and fills in the appropriate entries in the statistics.

Q: What's the difference between an Idle mote and a Lost mote? I have a 10 mote network, but there are 15 mote ids on the list.

A: The SmartMesh WirelessHART manager remembers all motes that have ever joined to it. When the manager boots up, all of these are listed as **Idle**. Once a mote has transitioned through the various joining states (**Negotiation**->**Connected**-> **Operational**), if that mote leaves the network, it will go to state **Lost**. Anytime another brand new mote joins, it will be given the next available Mote ID. You can delete any mote that is in state **Idle** or **Lost**. That will 'free up' a Mote ID, and the next new mote will be assigned the next available Mote ID. Mote ids in general persist in SmartMesh WirelessHART (unless you delete the mote from the manager). In the manager CLI (nwconsole), the command `sm` will show motes in the negotiation stages, the connected, operational, or lost stages. The `sm -s` command will show **Idle** motes as well. The XML and serial APIs list idle motes as well.

Q: Why am I unable to start the Manager CLI (nwconsole)?

A: There could be several reasons for this. One reason could be that the underlying process (nwconsole) is unable to start because the main software process (dcc) is not running. Try to restart the dcc process and then try to start the CLI. If the dcc process does not start, a Manager reboot might help to flush things out in case some files have inadvertently gotten corrupted. Another thing is to make sure that the `/etc/network/interfaces` file is not corrupted or empty. This is possible if you try to edit that file manually.

Q: What are the units for the discharge time in the *get mote* command?

A:The units are in terms of clock ticks based on a 32 kHz crystal. Specifically 32,768 ticks equals 1 second.

4.3 LTP5903-WHR/LTP5903EN-WHR Managers

Q: What TCP ports does the manager listen on? Does the manager need to be firewalled?

A: LTP5903-WHR/LTP5903EN-WHR managers have configurable firewalls - see "Installing the Manager" in the SmartMesh WirelessHART User's Guide for details on configuration. The linked page also lists the TCP/UDP ports that are open by default.

Q: How do I properly firewall a manager connected to a LAN?

A: Reconfigure the firewall to expose only the SSL encrypted API and notification ports. Optionally, you can open the SSH port for linux command line access, and HTTPS port for the admin toolset configuration application. See "Installing the Manager" for details.

Q: What are the units of dustTime in the XML-RPC documentation?

A: The dustTime element in config/System/time is the number of milliseconds since the start of the Unix epoch, or 00:00:00.000 1/1/1970 GMT.

Q: Are the ports for the control channel and notification channel for the Manager XML API fixed?

A: No, these ports can be changed by the user via the CLI. See "Installing the Manager" for details on configuration.

Q: How do I restore my manager to factory default settings?

A: Packaged managers have a factory restore button, labeled "mode" (don't ask) on the case. With an embedded LTP5903-WHR manager, a script (/usr/bin/restore-factory-conf) needs to be executed from the Linux command line (CLI). See "Restoring Manager Factory Default Settings" in the SmartMesh WirelessHART User's Guide for additional details on the procedure and what is reset by the restore process. See the "Introduction" section of the SmartMesh WirelessHART Manager CLI Guide for details on making a Linux CLI connection.

Q: How do I restore/change the IP address of the manager?

A: After connecting to the manager via a serial port and logging into the Linux command line:

- Use command `ifconfig` to see your current IP address on eth0. If the manager was booted without an Ethernet cable connected, you may not see an entry for eth0.
- Use the `sudo ifswitch-to-static` script to give the manager a static IP address, or
- Use the `sudo ifswitch-to-dhcp` script to have the manager get its IP via a DHCP server.

Q: Why does a mote take longer to join a network after it has previously joined a Manager?

A: The LTP5903-WHR/LTP5903EN-WHR managers fall back to a “slow” advertising mode one hour after the last mote joins - the advertising rate drops by a factor of 16 to save power, so the time for a lost device to synchronize the network goes up by a factor of 16. The manager will reactivate "fast" advertisement when it realizes a mote has become lost, but the process of verifying that a mote is lost may take much longer than the expected synchronization time in fast advertising mode.

The AP advertises more than any other mote in both slow and fast modes. Expect motes that are in range of the AP to join faster than those further away.

Q: How long should it take the manager to figure out a mote is no longer in a network?

A: When a mote powers down, its neighbors must first detect that it is gone, then the manager will attempt to contact the mote to determine if it is still available:

- Mote powers down
- Neighbor reaches a path down time out in 4 minutes and sends a path alarm

- Manager sends a reliable command to the missing mote and sends it again after the reliable transport timeout, which is configured dynamically based on frame length and current network conditions, but is ~2 minutes . It gives up after 5 retries and changes the mote to the **Unknown** state.

Thus the time to figure out a mote is no longer in the network could be up to 14 minutes. In the meantime, if the mote resynchronizes to the network and sends in a new join request, the manager will immediately begin rejoining the mote to the network.

Q: Under what conditions does the manager change its fast advertising state (ON/OFF)?

A: Conditions to turn ON fast advertising:

- The manager and AP are alone in the network
- A new mote (since network was rebooted) attempts to join
- Any mote that had been operational changes to **Lost**
- The API to turn on advertising in used

Conditions to turn it OFF:

- 2nd and 3rd bullets above haven't happened in the last hour
- The mote that triggered advertising by going lost rejoins
- The timeout supplied in the advertising API expires
- The advertising API is called with a timeout of 0

Q: What are the units for the discharge time in the getMote API?

A: The units are in terms of clock ticks based on a 32 kHz crystal. To get the value of the displayed discharge time in seconds, divide it by 32,768.

Q: According to the Manager Serial API Guide, under the section "Sequence Numbers", "Only one unacknowledged packet can be outstanding at a time and the sender should not send the next packet (whether best-effort or reliable) until the previous packet is acknowledged." Does that mean that I cannot send two messages to different motes at a time?

A: That section refers to the Serial API protocol used between the manager and the client application. This is different from the protocol used between a manager and the motes. You are allowed to send packets to different motes at the same time.

Q: When I send a packet to the Manager using the Serial API, the connection closes. Why does this happen?

A: Check to make sure the packet is properly formatted. Also, ensure that the correct sequence number is being sent. If not, the Manager will close the serial connection.

Q: How do I set the wall clock time on the manager?

A: The manager uses an NTP client to synchronize its wall clock time with the network if connected to a network where an NTP server is available. Changing the manager's time should not be done while a network is operational, as time discontinuities can affect network behavior. The time can be set using the Linux `date` command in YYYY-MM-DD hh:mm[:ss] or hh:mm[:ss] format. The network should be restarted after changing the time.

If communication to the Dust Manager is by Serial API only, the *adjustTime* API can be used to change the manager's clock (after setting it) as it will drift slowly. The *adjustTime* API does not result in an immediate change - the time delta will be slowly absorbed by the network.

Q: How is mote bandwidth calculated from the Manager's configuration page?

A: All motes start with a default base bandwidth (period) of 100 seconds. Any services requested by the mote will be added in parallel, and the reciprocal of the periods are shown. For example if a mote has a 100 s base bandwidth, and is granted a 15 s publish service, the total bandwidth will be 1/(1/15 + 1/100) = 13.043 seconds

Q: How many notification channels can I concurrently subscribe to?

A: The LTP5903-WHR/LTP5903EN-WHR managers can support up to 3 live simultaneous connections to client applications. Each active application has its own login token that uniquely identifies itself to the manager.

4.4 LTC5800-WHM Motes

Q: What should I set the "discharge current" parameter to in the mote? How does this parameter influence network operation in terms of performance, reliability etc?

A: Setting it to the max value guarantees that the manager will never be prevented from doing the right thing for reliability. Setting discharge current to a lower value might hurt reliability. It should only be used to mark a current value above which this device will reset.

Q: Is the MAC address of the motes the same as what is specified in the HART7 spec: hcf_spec075.r1.0j, figure 9 EUI-64 address?

A: Dust motes come with an internal EUI-64 address consisting of a 3-byte Organizationally Unique Identifier (OUI) assigned to Dust (00-17-0D), followed by a 5-byte serial number. HART addresses use the HART OUI (00-1B-1E), followed by a 2-byte Expanded device type, and a 3-byte serial number. Thus there can only be ~16 million of any type of HART device. To be HART compliant, OEMs should use the mote serial API *setNVParameter <macAddress>* to set a HART compliant EUI, otherwise the mote will use its internal EUI.

Q: Are Dust-based products already certified to be WirelessHART complaint? Must I get a WirelessHART certification if I want one?

A: The modules have been pre-screened to ensure that you will meet WirelessHART requirements, but the HART foundation only certifies completely integrated devices. For field devices, this requires compliance on two levels:

1. Application interoperability – enables interoperability between HART compliant controllers and gateways and HART registered field devices. Compliance with general HART requirements is left to the OEM developer.
2. Wireless mesh network interoperability – ensures field devices from different vendors will be able to communicate and interoperate via the wireless mesh. Dust-based products have been certified to version 7.1 of the WirelessHART specification. Most aspects of operation have been tested by Dust, but some command behavior involving transport termination relies on proper integration with the OEM device.

Q: How do I get a WirelessHART certification?

A: Once the OEM has the final integrated wireless field device, the OEM must submit the field device to the HART Communications Foundation (HCF) for verification and registration of both the application and wireless mesh network layers. (HCF does not support a 'modular' certification for only the wireless portion.) See the Application Note "Testing for WirelessHART Certification" for a description of the process.Please contact HCF for further information on the HART specification, details on the HART registration process, and HCF membership. For those not already familiar with HART as an application standard, HCF offers several seminars and training sessions, including a "HART Fundamentals" multi-day technical workshop.HCF links: http://www.hartcomm.org – HART Communications Foundation webpage.
http://www.hartcomm.org/protocol/training/training_training.html - HCF training resources
http://www.hartcomm.org/protocol/training/training_training_fundamentals.html - "HART Fundamentals" workshop

Q: The Mote HART Response List seems very old and lists a large number of commands that are shown as not implemented. Is this still the current state? Do you have a revised document? How are we supposed to deal with these commands with respect to HART compliance?

A: In Wireless HART, the mote has a range of commands that it must respond to with a valid HART Response Code. These command IDs are clearly specified in the document. It is optional as to whether a command responds with data or not, but it is not optional whether to respond or not. For many of these commands, the motes responds with the valid HART response: "command not implemented". We may implement some of these commands in the future. We may never implement some of them but we do respond to all the commands in our range, just as your device must provide a valid HART response to all the commands that are in your range, even those that you choose to respond with "command not implemented". Our choice to implement some commands and not others will neither impact your ability to get your device certified by HART not put any additional software burden on your development.

Q: Why do the WirelessHART commands 780 and 787 sometimes return neighbor RSL values of -100 dBM or 0 dBm?

A: A value of -100 dBm can be returned if the 780/787 command is received just after the mote has cleared the internal health report (every 15 minutes) statistics and before the mote has heard any traffic on the established link.A value of 0 dBm means no RSSI is able to be read from the radio at that time. This happens because sometimes the radio is busy doing other things and can't get the RSSI. A value of 0 dBm will continually be returned in this case until either new traffic is heard or the end of the health report time frame (15 minutes) is reached.The client needs to query again later in order to get a true RSSI value.

Q: Why doesn't the mote respond when my sensor processor sends it a command?

A: The mote should always send a response packet back to the sensor processor after having received a command. The mote will do so even if the mote is sending an error response. If, however, the sensor processor has the "Do not ignore" bit enabled in the Flags field of all command packets, and then *does not* toggle the "packet ID" bit in the flag field, the mote will not send a response packet. In this case, the interface appears to "hang" waiting for a command. To fix this, either set the "Ignore" bit or toggle the "packet ID" bit for every command sent. If the sensor processor is doing either of the two suggestions above and the mote is still not sending back a response command, you should then suspect that the mote isn't receiving the command to begin with. This would happen if due to UART errors. In this case, verify that the sensor processor is capable of robustly sending UART bits to the mote by using an oscilloscope.

Q: After issuing a "disconnect" command, how long will it take for the mote to reset?

A: A LTC5800-WHM mote will reset about ~50 seconds after having received the "disconnect" command. The sensor processor should take steps in preparing for the mote reset.

Q: Is there a way to change the Network ID without resetting the mote?

A: Once the mote is joined to a network, it is not possible to change the Network ID without resetting. If the mote is searching but not yet joined, it is possible to change the Network ID without resetting, but this should only be used in cases where the charge to reset the device is unavailable - see the SmartMesh IP Mote API Guide.

Q: Is the Cold Start Bit distinguished between the Gateway Master and Network Manager Master or is there only one combined cold start bit handling?

A: There is only one cold start bit - the HartDeviceStatus byte does not distinguish between the GW and the Manager.

Q: Will the Cold Start Bit be always reset by the Mote after each cold start and thus the target doesn't need to care about Cold Start Bit handling on the wireless side?

A: No. The mote does not reset anything on its own. After reset, the OEM microprocessor needs to write the HartDeviceStatus byte, which will then be used by the mote to terminate commands. HartDeviceStatus is not preserved through reset (i.e. it is NOT a setNV) and hence it needs to be written each time, BEFORE the mote attempts to join and after each status change.

Q: Is the Configuration Changed Flag (CCF) distinguished between the Gateway Master and Network Manager Master or is there only one combined CCF handling ? If say, the Gateway Master resets its CCF by using CMDxx. The CMDxx-response from the target (OEM micro) has the CCF for the r espective Master reset. Does the mote use this response to reset a combined CCF in the mote or does it reset also only the CCF for the Gateway Master in this case ?

A: There is only one CCF bit - the HartDeviceStatus byte does not distinguish between the GW and the Manager. Separate CCF or any other status for different masters is NOT maintained. The network manager does NOT use the Hart Status Information. The only commands in the wireless space that affect the CCF are the 771 and 773. Spec 155 indicates that the GW is a slave for 773 so only the network manager can be the master. Cmd 771 is intended to be used over the wired interface.The mote does NOT reset anything on its own. The OEM microprocessor(target) needs to write the updated HartDeviceStatus byte in case it needs to change due to reception of any target terminated command.

Q: Some of the Wireless HART commands in the Mote HART command list respond with "not implemented" or "access restricted". HART Spec 155 specifies some to these to be mandatory. Is this going to be a problem to get HART certified?

A: See " **Q: The Mote HART Response List..."** above

Q: What does "access restricted" mean?

A: Many commands are restricted - they can only come from the manager - this is to avoid contention between different components trying to control the network. Alarm commands are also restricted to being published by the mote, not read from the gateway or manager.

4.5 Integration

Q: What is the difference between the “disconnect” command issued to the mote by the microprocessor and the “decommission device” command issued by the manager?

A: A mote receiving a "disconnect" command will notify its neighbors that it is going off line. The neighbors then report a path alarm immediately to the manager, thereby saving the four minutes that it usually takes to incite a path alarm. The manager will still attempt to communicate with the mote because a path alarm only means that a path no longer works. Only after the manager figures out that it can’t communicate with the mote will it be declared lost.A manager "decommission device" command is more like telling the manager that you are going to remove a mote from the network, and would like that removal to cause as little disruption as possible. In this case, the manager will re-route links to other motes before notifying the application processor that the decommission processes has been completed. This can take several minutes but it is the optimal method to remove a device from a network. This is the “graceful” method for removing a mote from a network. After decommission is complete, the mote will still be in the network, and will remain in the network until you power it down or reset it or remove it. Decommission just made it safer for the rest of the network.

Q: How long should a sensor processor wait before it should quit trying to send data over the network?

A: The sensor processor can always send data when the mote is in the **Operational** state and acknowledging its serial API. If the mote is unable to send data to the point that it loses synchronization, it will reset itself and attempt to rejoin. The mote will send a *disconnect* event to the sensor processor before it resets. If the mote does not respond on the serial API for 5 s, then it should reset the mote.

Q: How long should it take the sensor processor to figure out the mote is no longer in a network?

A: The sensor processor can depend on the mote sending a *disconnect* event when leaving the network. After resetting, the mote attempts to reconnect to the network. The sensor processor can use the *getParameter<moteStatus>* API to determine the status of the mote (searching, joined, etc.)

Q: Can downstream packets be sent reliably?

A: Yes

Q: Is it OK to delete a mote's Maintenance service?

A: No. Every mote gets maintenance service at join time and it cannot be deleted. This service is necessary for the mote to maintain itself in the network. The sensor processor can use this service, on a limited basis, to send/receive data to/from the manager. This service should only be used for "control" type of data and not "monitoring" type of data. Temperature, humidity, etc. is considered "monitoring" type of data. Other services are available for the sensor processor to use.

Q: Can the sensor processor's bandwidth service ever be taken away once it has been given?

A: Yes, the manager can remove a service that has been given to a mote. When this happens, the sensor processor must request the service again. The sensor processor should be prepared to continually ask for its service until it gets it. The manager will not assume a mote wants a deleted service back.

Q: Can the sensor processor's bandwidth amount change for a given service?

A: No. The manager will not simply reduce the amount of bandwidth that is allocated for a particular service. The manager will delete it and then inform the sensor processor. If the sensor processor is willing to use less bandwidth than was previously allocated, it can request a new service with less bandwidth.

Q: Can the sensor processor be denied a service request?

A: Yes. Only the Maintenance service is guaranteed to be allocated to the sensor processor.

5 Hardware Integration

Q: What is the reason for a limit of +2 dBi antenna gain when Europe allows transmits with 100 mW and US/Canada 1 W?

A: The limit on the antenna gain is not 2 dBi, and some customers have certified with antennas with higher gain. In supporting a single product that can be certified world wide, the maximum radiated output power in band is +10 dBm/MHz. For mesh networks, where one does not plan the direction for communication, omnidirectionality in an antenna is desirable. +2 dBi antenna's are common in many different form factors, so +8 dBm output power is a natural output power to target given the +10 dBm/MHz limit. While Europe allows 100 mW for *frequency hopping* module and only 10 mW for a *Direct Sequence Spread Spectrum (DSSS)* module, test data for a DSSS module can be taken as part of a data set used for certification in other geographies; whereas, data taken for a *frequency hopping* module is not consistent with other geographies. In addition, since the intention is to spread the energy across the band, channel hopping schemes are required to use at least 15 channels, which makes blacklisting of select channels impossible according to the IEEE 802.15.4 channel assignments. We chose to certify under spread spectrum rules to enable blacklisting, a required feature of the WirelessHART protocol. The draft version of ETSI EN300 328 V1.8.1 places further restrictions on for *frequency hopping* module that will further reduce the attractiveness of certifying as a frequency hopper, while at the same time providing an opportunity for a a *Direct Sequence Spread Spectrum (DSSS)* module to operate above 10 mW. We are monitoring the progress of ETSI EN300 328 V1.8.1 with a mindset to try to allow for additional customer flexibility within the limits of the new regulations. Practically, there may be a timing limitation for providing this based upon the relative approval of the specification relative the product's RPL date.

For North America, the rules for certification have changed, allowing a procedure that is not antenna specific and will allow for higher gain antennas. The goal for the newer module products, LTP5900 and LTP5902 is to provide a certification that is both independent of antenna type and higher gain.

Also note, Wireless Hart does specify that the nominal output power of a device must be 10 dBm, with our radios requiring a +2 dBi antenna.

Q. If I'm using a modularly certified mote, do I have to get the whole product certified?

A. Modular certification allows a complete product to be certified as an unintentional radiator, as is required for any electronic device that does not contain a radio. Radio specific testing is not required, and in fact disabling the radio for unintentional radiator testings is preferred. If a product complies with Part 15, has the appropriate labeling, a compliant antenna, and has performed the unintentional radiator scans, the product is certified.

Q. Are there other certifications (radio or otherwise) needed?

A. Per geographies that support modular certification, no additional radio certifications required. Depending on the complete target's application other certifications may be required for such requirements as safety, intrinsic safety, application specific certifications that can include operation under humidity, vibration, temperature cycling, salt fog, shock, etc.; however, those tests are not specific to a product including a radio or not.

Q. Do most customers work with 3rd party labs to get the certification completed or do they typically do it themselves? How much does certification cost and how long does the certification process typically take?

A. We have yet to see a customer that is both a certified body and a manufacturer of equipment, so 100% of customers to date have used an external certification house. The certification process for FCC (US), IC (Canada) and CE (EU), can be done in four weeks for a company that has done it before and is well organized. Certifications for other countries typically take longer due to logistics, both on the customer side and the certification body side, ranging from six to twelve weeks. Modular certification costs for FCC/IC/CE have typically cost ~$20k; however, ETSI EN300 328 V1.8.1, the new CE standard, will require more test time and cost more as a result.

Q. What other country certifications have we obtained?

A. See the product datasheets (available here) for certification details.

Q: Are/Will the mote modules be certified as intrinsically safe or do we only provide the data they require to get their own certification?

A: We do not certify the devices as intrinsically safe. We will provide complete documentation to support intrinsic safety certification.

Q: What is the effect on performance of the mote module under the following conditions: Vibration, Fast circulatory motion (Doppler), Exchange of position (picture two motes on a large moving carousel)?

A: Mote performance is not expected to change under any of these conditions.

Trademarks

Eterna, Mote-on-Chip, and SmartMesh IP, are trademarks of Dust Networks, Inc. The Dust Networks logo, Dust, Dust Networks, and SmartMesh are registered trademarks of Dust Networks, Inc. LT, LTC, LTM and are registered trademarks of Linear Technology Corp. All third-party brand and product names are the trademarks of their respective owners and are used solely for informational purposes.

Copyright

This documentation is protected by United States and international copyright and other intellectual and industrial property laws. It is solely owned by Linear Technology and its licensors and is distributed under a restrictive license. This product, or any portion thereof, may not be used, copied, modified, reverse assembled, reverse compiled, reverse engineered, distributed, or redistributed in any form by any means without the prior written authorization of Linear Technology.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015 (b)(6/95) and DFAR 227.7202-3(a), and any and all similar and successor legislation and regulation.

Disclaimer

This documentation is provided “as is” without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

This documentation might include technical inaccuracies or other errors. Corrections and improvements might be incorporated in new versions of the documentation.

Linear Technology does not assume any liability arising out of the application or use of any products or services and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

Linear Technology products are not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Linear Technology customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify and hold Linear Technology and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Linear Technology was negligent regarding the design or manufacture of its products.

Linear Technology reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products or services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to Dust Network's terms and conditions of sale supplied at the time of order acknowledgment or sale.

Linear Technology does not warrant or represent that any license, either express or implied, is granted under any Linear Technology patent right, copyright, mask work right, or other Linear Technology intellectual property right relating to any combination, machine, or process in which Linear Technology products or services are used. Information published by Linear Technology regarding third-party products or services does not constitute a license from Linear Technology to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Linear Technology under the patents or other intellectual property of Linear Technology.

Dust Networks, Inc is a wholly owned subsidiary of Linear Technology Corporation.