Cluster-Hub firmware
Release Notes

20 March 2023

The Cluster-Hub firmware was first released to run on the Afar “Classic” radio model AR-24027, which is no longer in production. That firmware version is identified as version 1.xx.

In 2010 we ported this firmware to the newer family of Afar radios, and tested it in two of those models (AR-09010E and AR-24027E). This firmware is identified as version 2.xx or higher.

Version 2.xx was then back-ported to the “Classic” radio model to provide a compatibility path between the Classic radio and the new model AR-24027E.

With release 2.20 and later the new radio model AR-24027E can be mixed with the Classic radio AR-24027 as long as both run versions 2.20 or later. Note however that even though they have the same version the firmware files for the “Classic” radio and the new model radios are different as they run on different processors. For version 2.20 the file names are:

CLH02_20.BZ	Runs on the Classic radios
CLH02_20.BZE	Runs on the newer models

Version: 1.29
Date: 15 Feb 2010

Last version released for the “Classic” radio models.

Version: 2.12
Date: 22 Dec 2010

Validated in the 900 MHz radio model AR-9010E. There are some deficiencies uncovered later which are described in the version 2.20 below

---

Version: 2.20
Date: 1 Mar 2011

Compatibility:

1. Validated in following models: AR-9010E, AR-24027, AR-24027E,
2. In the model AR-9010E this version is compatible with version 2.12.
3. The two 2.4 GHz models can inter-operate when running this version.

New Features:

1. At the master hub you can now use the “monitor-link” command to monitor all links at the same time by specifying the link number as zero.

Problems Fixed:

1. In a two hub system, if the hub with the higher serial number is rebooted, the resulting two-hub cluster does not synchronize properly and oscillates losing and regaining sync. This does not occur when there are three or more hubs.
2. With very high traffic on the hub ethernet, some heartbeat packets that were delayed on the LAN were not being rejected. This can result in a slave-hub adjusting its cycle improperly and possibly causing an RF collision.
3. With a TDD cycle set to 20 ms the slave hubs reject a high number of heartbeat packets unnecessarily, filling the event log quickly.
4. If the master-hub in a cluster was the only one needing to transmit over RF for that cycle, it did not send out the outbound schedule on the Ethernet. This resulted in the slave hubs logging a warning that could quickly fill their event logs.

Known problems:

1. The “monitor-link” issued at a remote radio is not working properly. It constantly clears the counts.
2. The master hub may not wait for missing inbound packets resulting in discarding packets that arrive later out of order.

Version: 3.00
Date: 4 May 2011

Compatibility:

1. Validated in following models: AR-9010E, AR-24027, AR-24027E,
2. All radios must have version 3.xx. Can not be mixed with version 2.xx

New Features:

1. Supports networks with up to 16 different clusters automatically selecting one of the cluster masters to act as the clock source for all radios in the multiple clusters. All radios in the multiple clusters then synchronize their cycles to the single clock.
2. Each hub dynamically measures the Ethernet transit time from the clock master. It then offsets its cycle to cancel the effects of this transit time.
3. Improved filtering of the cycle period errors, both for hub synchronization and the remote radio synchronization over RF, resulting in much tighter synchronization with less probability of RF collisions.

Problems Fixed:

1. Fixed the known problems listed for version 2.20

Version: 3.01
Date: 20 May 2011

New Features:

The highest value for the "max-response-bytes" in the UDP interface was limited to the maximum size of an Ethernet packet resulting in the radio responses to certain commands to get truncated. Now the highest value was increased to 65521 bytes. These "jumbo" UDP packets are fragmented into multiple Ethernet packets by the IP stack and reassembled on the host side.

Problems Fixed:

1. Hub radios check for consistency in their configuration (cycle period, split, priority) against other hub radios in the same LAN and report any conflicts in the event log and "show"command. The message text did not match the conflict type.
2. Issuing the "load-configuration"command on a hub radio with multiple remotes, could, on occasion, cause the radio to reboot.

Version: 3.02
Date: 10 Oct 2011

Problems Fixed:

1. The cluster synchronization over Ethernet did not work properly when the nominal Ethernet delay between the synchronization master and a slave hub exceeded ~55 us. Now it supports delays on the Ethernet of up to ~250 us.
2. When a radio restarts its resynchronization over the Ethernet it could lose a buffer. After many of those events the radio would reboot.
3. For the 900 MHz models the maximum allowed transmit power is now 26 dBm. With previous max of 27 dBm, the transmitter power amplifier could get damaged when transmitting into a reactive load.

Known Problems:

The network-id in the node command is erroneously being truncated to 8 bit wide (fixed in 3.04).

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.03
Date: 2 Aug 2012

New Features:

Increased the maximum transmit power in the 900 MHz model back to 27 dBm (it had been limited to 26 dBm in October 2011 to prevent damage to the Power Amplifier). All new shipments of the 900 MHz radios have a different last stage Power Amplifier (PA) which is not susceptible to break under reactive loads and provides a cleaner signal. You can issue the "version" command to find which model you have:

```
  Hardware Type:  256x-00A4   (Original 900 MHz with weaker PA - power limited to 26 dBm)
  Hardware Type:  256x-00A6   (Improved 900 MHz model with stronger PA)
```

Compatibility Issues: Version 3.03 runs on all radio models and identifies the different 900 MHz hardware types. It allows you to set a transmit power in the range of 0 to 27 dBm. However, if the hardware is the original 900 MHz model (with a weaker PA) it clips the maximum power to 26 dBm.

Earlier software versions do not recognize the new hardware type and do not run properly if back loaded into the new 900 MHz radio models.

Problems Fixed:

The switching power supply in all model radios (except the Classic) could create noise in VHF channels around 120 to 180 MHz. This could interfere with a "walkie-talkie" operating in close proximity to the Afar radio. With this version the power supply switching frequency is dithered randomly which eliminates the energy in the harmonics that were causing this problem.

Known Problems:

In a cluster with more than 2 hubs, the third and subsequent hubs has difficulty synchronizing to the existing heartbeat. Fixed in version 3.05

The network-id in the node command is erroneously being truncated to 8 bit wide (fixed in 3.04).

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.04
Date: 25 Oct 2012

Problems Fixed:

The network-id in the node command was erroneously being truncated to 8 bit wide. Now it accepts and stores full 32-bits.

Opening and closing Telnet or Econ sessions with the radio caused a slow memory leak. After many of those events, the radio might stop responding to pings and eventually reboot.

Known Problems:

In a cluster with more than 2 hubs, the third and subsequent hubs has difficulty synchronizing to the existing heartbeat. Fixed in version 3.05

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.05
Date: 3 Nov 2012

Problems Fixed:

In a cluster with more than 2 hubs, the third and subsequent hubs had difficulty synchronizing to the existing heartbeat. This was introduced in version 3.03, worked better in version 3.02. It is now fixed.

Known Problems:

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.10
Date: 13 Aug 2013

New Features:

1. The DHCP parameter in the IP command now accepts specifying the port (local-only or radio-only) where to look exclusively for a DHCP server. Previously, you could only turn DHCP "on", and the radio would look for a server on both ports. The value "on" is still allowed.
2. You can now specify the "min-cluster-size" in the node command. This allows to configure hubs such that, if they become disconnected from the LAN, they shut down the RF transmissions instead of creating a new cluster and collide with the other hubs. The value of 0 disables this feature and keeps compatibility with previous versions.

Problems Fixed:

1. The "load factory" command could induce an immediate reboot due to the single node timeout feature being turned on. Now it resets the reboot timeout first and no reboot happens.
2. In the command ">display-config factory", the dhcp-client field was incorrectly displaying the current value instead of the factory default (off).

Known Problems:

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.11
Date: 21 Aug 2013

Problems Fixed:

1. When a remote is getting out of range of the cluster, the master hub could queue a "link management packet" to that remote and later find that no hubs are covering it. That packet was not transmitted and was not being freed resulting in a slow buffer leak which, over time, would eventually cause the master hub to run out of memory and reboot.
2. On power up a remote radio seemed vulnerable to some Ethernet packets that were failing a consistency check and causing the radio reboot. This condition now logs an event but does not reboot the radio.
3. The ">download" command, if given a file that did not exist, could cause the radio to reboot.
4. If a hub with an incompatible software is present in the same cluster, the radios issue a warning (in the "show" command) and do not attempt to synchronize. If you then turned off the hub with the incorrect software the other hubs failed to recognize that the problem had been removed and would not synchronize until rebooted.

Known Problems:

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.12
Date: 27 Oct 2013

New Features:

1. Added a new parameter to "rf-receive-setup" command: "disconnect=boolean". When set to 1 a remote radio will disconnect from the current cluster and start looking for a new parent right away. When set to 0 (or not specified), a remote radio changes the channels or antennas but stays attached to the current cluster. This is useful if that one cluster has hubs set to transmit on two channels, in which case the radio stays attached. If there are no hubs in the same cluster transmitting on the new frequency, the remote radio times out after 1 second, only then tries to attach to a new cluster.

Problems Fixed:

1. The algorithm to compute the delays over the Ethernet (used in the synchronization of all the hubs), has a window of acceptable jitter for a new heartbeat packet to be accepted for averaging. That window was too narrow and after a few packets missing it, it could cause the clock to drift and then force the slave-hub to restart the synchronization. This would be seen as a "Dropped hub" event logged by the hub master followed a few seconds later by adding that hub back again.
2. There was a warning event being logged erroneously in the IP stack ("Bad free of MESS..."). This was being seen when the radio received and processed a PING or an ARP for example.

Known Problems:

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.14
Date: 29 Oct 2013

Changed the default of the "disconnect" parameter for "rf-receive-setup" (added in version 2.12). Now, if not specified and the channel or antenna changed, the remote radio automatically disconnects from the current cluster and starts looking for a new one right away. You can override this default with the parameter "disconnect=0", which prevents the automatic disconnect.

Problems Fixed:

When a new IP address was obtained though DHCP, the IP stack was not being initialized correctly and the radio might not respond to IP packets sent to that address right away.

Known Problems:

There is a potential for a memory leak if you use the UDP command interface and issue commands that result in replies that use jumbo UDP packets. See fix in version 3.16

Version: 3.15
Date: 10 Dec 2013

Known problems:

Memory leak (applies to previous version): When the radio is functioning as the cluster master, under some conditions, it may experience a memory leak (see fix in version 3.16). This can be tracked with the "memory" command which displays various parameters regarding the free memory pool status.

```
Memory information:
    1059 allocated buffers,      5 fragments
 1211552 total free bytes, largest fragment = 1202720
```

Over time the number of allocated buffers and fragments increases, while the size of the largest fragment decreases. Once the size of the largest free fragment is too low the software can not allocate memory and it reboots.

In this version we added the "monitor-memory"command which continuously displays the above values and adds tracking the number of the IP stack memory allocations and releases from/to the memory pool, as well as events with troubleshooting information to help track this incident.

Problems Fixed:

1. If a remote radio misses receiving an outbound data packet sent to the Ethernet broadcast address, a subsequent packet (received correctly) could stay stuck in the radio (waiting for the missing packet), either indefinitely or until a new broadcast packet arrived. Now the timeout of about 300 ms works and releases those packets.
2. When a remote radio is in range of more than 4 hubs, the "monitor-coverage" command at the master hub (which displays the signal strengths of up to 4 hubs) was displaying its information incorrectly.
3. If you turn on DHCP mode and the radio already had an IP address, the radio did not immediately request a new IP address from a server. It would instead wait until the address had to be renewed, only then would change the IP address. Now it requests a new IP address right away.
4. For the 900 MHz models using an older RF board version, the maximum transmit power is now capped at 24 dBm (even though the command accepts values up to 27). With higher power levels the transmitter power amplifier could get damaged when transmitting into a reactive load. This does not apply to the newer RF board revision which can transmit the full 27 dBm. The software recognizes which board is installed and only caps the power for the older version. You can identify the board in your radio in the power up banner (or with the "version" command) as follows:

    Hardware type: xxxx-00A4 (older RF Board version)

Hardware type: xxxx-00A6 (newer RF Board version)

Version: 3.16
Date: 20 Feb 2014

Problems Fixed:

1. Memory leak: when using the UDP command interface, if you set the max-response-bytes to a large number and issue a command that results in a reply that makes use of jumbo UDP packets (larger than 1500 bytes), the integrity of the free memory pool becomes compromised. This was reflected as a slow decrease of free available memory, or a sudden drop of available free bytes down to 0 (both shown with the "memory" command). This has been fixed.
2. With the Ethernet running at 10 Mbps long packets could be received with a frame error and discarded (with an entry in the event log). This was due to a clock tolerance setting which was now increased to correct the issue.
3. With the Ethernet running at half duplex a packet collision would result in an incorrect event being logged indicating an error in the packet Time of Arrival. This event is no longer logged.

Version: 3.17
Date: 24 Oct 2014

Problems Fixed:

When the number of hubs in the cluster matched its maximum allowed value (which is shown in the "show" command), and there were no remotes, then broadcast packets were not being transmitted over RF. This could result in the broadcast queue to fill up, and the master radio to log a warning event. Now, while there are no remotes, the master hub discards all broadcast packets so the broadcast queue remains empty until a remote is accepted into the cluster.

The cycle split computation (in auto mode) was not taking into account bytes that had been pulled out of the transmit queues but not yet transmitted over RF. When there is inbound traffic this could cause an unnecessary extra delay in the transmission of outbound packets.

The computation of slots to assign to a remote was slightly underestimated. Under some conditions it could result in assigning a single slot when the remote minimum number was 2. This would result in the remote logging an error, and not transmitting in that cycle.

The engineering command to show the transmit queues (dtq) now only runs in the master hub, as it was intended. Also, it now shows the number of bytes that have moved out of the transmit queues but are waiting to be transmitted.

Version: 3.18
Date: 17 Sep 2015

New Features:

Added a parameter “block-remote-broadcast” to the ethernet command. If "on" then the master-hub will NOT re-transmit (to other remotes) broadcast packets received from a remote. If your system never requires remote to remote communications, this prevents the unnecessary broadcast of packets over RF to reach every remote.

Problems Fixed:

While a hub has no remotes, when it receives a broadcast packet it was incorrectly logging an error event (“215 Plan Ahead Error 13”). This causes no harm. It was a side effect from a fix in version 3.17 where we flush those broadcast packets while the unit has no remotes.

Improved the backoff timing when collisions occur due to many new nodes joining the network simultaneously.

In the process of joining a cluster a remote was not checking if the cycle schedule packet was coming from the selected cluster. If there were two clusters within range, on the same channel (which the design should avoid), this could slow down the attachment process.

Known Problems:

When performing a code download over a telnet connection in a fast LAN, you may see "S-rec length error" leading to an unusable file. The workaround is to configure your terminal emulator to insert a delay after each line of text pushed down to the radio. A delay of 1 ms per line makes the problem go away. This problem also appears in older versions.

Version: 3.19
Date: 6 Nov 2015

New Features:

When a remote attaches to a new cluster, it now immediately transmits over RF, to the new cluster, ethernet packet(s) with the source address set as each of the stations in its LAN. The cluster hub-master broadcasts these packets on its LAN which triggers the network fabric to re-route packets to the remote stations in their new place in the network.

Below is a lengthier explanation for this change:

The feature addresses the issue created by mobility where your equipment on the train changes its place in the network from one cluster to another.

Between the trackside hub radios (made up of various clusters) and your central computer there will be a "network fabric" made of various Ethernet switches. When your central computer sends a packet to your equipment on the train, that packet makes its way, across the network fabric, to the cluster-hub that the train is associated with. The network fabric has learned which cluster that is from previous inbound traffic and routes the packet appropriately.

Now the train moves to a new cluster. The Ethernet switches in the fabric do not know this. The central computer sends a packet to the equipment on that train and the network fabric routes it to the old cluster. That outbound packet never reaches the train. Eventually your equipment on the train sends an inbound packet to the central computer. As it makes its way through the network fabric, the Ethernet switches along the way learn its new location in the network. Now, future outbound packets will be routed to the new cluster.

This new feature speeds up this re-learning process. As soon as a train attaches to a new cluster, the radio train sends a "proxy" Ethernet packet for each equipment in the train. These packets have the Ethernet source address of your equipment (not the radio). The hub master of the new cluster broadcasts them on the LAN. The network fabric now learns the new position of that train equipment right away. These packets have a zero length payload, make their way through the various switches which use them to update their switching tables, but are discarded by any other equipment.

You may actually address this mobility issue in a different way, for example having separate computers managing each cluster and actively participating in the hand-off before changing channels on the train. Even if you do handle this in a different way, these proxy packets cause no harm.

Problems Fixed:

Open and closing an Econ session on a radio, might, on occasion, result in a buffer leak. After many of those occurrences (more than 150) the radio could then run out of buffers at which time it would reboot.

The master hub would drop a slave-hub from its list if two consecutive status packets from the slave hub were dropped on the Ethernet. This should be very unusual but the timeout was increased and now the slave hub is only dropped after four consecutive packets are dropped.

An error in processing the response to an ARP request could cause an unexpected reboot of the radio. The error is now detected in time to allow the radio to place an entry in the event log and continue running.

Version: 3.20
Date: 9 Jun 2016

Problems Fixed:

On some radios, the commands “spectrum-analysis” and “time-analysis” could erroneously display a noise pulse every 500 ms. This has been corrected.

The event “212 Other hub on same channel (Mastr …)” could be logged incorrectly when transitioning from one cluster to the next

Improved transmit power calibration on certain radios with serial numbers between AF040001 and AF040400.

Version: 3.30
Date: 28 April 2017

New Features:

At a deployment in Singapore a faulty media converter (fiber to Ethernet converter in the wired network) caused a large and sudden drop of Ethernet packets going into one hub radio. That hub perceived this drop as the master having been disconnected and, since it was the alternate master, attempt to take over the mastership. This resulted in having two masters in the cluster, causing a conflict that resulted in communications with the trains to be impaired.

This version introduces several features to make the system tolerant to failures in the wired Ethernet. The hubs now must be "master-capable" in order to attempt to become the master of the cluster (prior to this version every hub was considered master-capable). In order to become "master-capable" the following conditions must be true:

1. The number of hubs in the cluster must equal or exceed the "min-cluster-size" specified in the ">node" command.
2. The Ethernet receive packet drops into that hub must not exceed a certain threshold.

Condition 2 requires the hubs to measure the packet drop percentage on their Ethernet receive ports. They compute this from the number of sync-status packets received over each second (sync-status packets are multicast, once per second, by every hub in the cluster). This computation is a running average since packet drops can be random and, in a small cluster, there is not enough sync-status packets to get an accurate number within one second.

If or when a media converter fails suddenly, the running average of the packet drops will take some time to rise and make the hub not "master-capable". Under those conditions, having two masters on a cluster is still possible for a short period. All hubs now detect when a second hub attempts to become a master and, if there are two masters in the cluster, they ignore the new one. This allows the system to continue to work until the hub experiencing Ethernet packet drops becomes not "master-capable".

There is also a "slave-capable" feature introduced in this version. If the Ethernet packet drop percentage exceeds a higher threshold, the hub declares itself as not "slave-capable". When not "slave-capable" the master will not assign that hub to transmit any packets to the train radios. Otherwise, a train that is being covered by an affected hub might lose communications with the wayside.

The current percentage of Ethernet packet drops is now displayed in the "show" command.

The command ">show radios" now reports whether the hub is capable to perform as a master and/or slave. At the master this information is displayed for every hub in the cluster.

If a hub is not capable of being either a master or slave, it displays a warning in the ">show" command.

There is a new engineering command ">monitor-master-capable" which shows the running average for the Ethernet packet drop percentage and several other parameters.

To test these features there is also a new engineering parameter to the ">ethernet" command to artificially introduce random packet drops on packets received over the Ethernet. You can issue the command:

```
>ethernet drop-packet-rate=90
```

to simulate a random packet drop of 90% in the Ethernet receive packets into this radio.

Problems Fixed:

In a slave hub the event "`187 Bridge enet inq overflow`" could be erroneously logged. This resulted from the process of opening a Telnet session on a slave hub which was queueing an extraneous broadcast packet into a queue that is not used. After several such sessions the queue was full. Other than the erroneous log this has no impact in the operation.

In a slave hub the event "200 Cycle evt 1 overrun: 65 us (0, 75 us)" was being logged occasionally by a slave hub that had been schedule to transmit in the second slot of the cycle. This is not a problem, the overrun is usually less than 100 us. We changed the code such that this condition does not result in a log.

Problems Remaining:

After being up for over 8 or 16 months the date/time tracking has a rollover problem and displays the date incorrectly. Rebooting the radio solves the problem. Solved in version 3.32.

Version: 3.31
Date: 13 July 2017

New Features:

In a cluster hub deployment any portion of the track is typically covered by multiple hubs. This provides redundancy and allows the hub-master to smoothly perform the hand-off between hubs as the train moves along the track.

When a slave-hub receives an inbound transmission from a remote (train), it forwards that packet to the hub-master for processing. Since any portion of the track is covered by multiple hubs this results in several repeated packets being sent to the master.

These multiple packets are usually not an issue. But if there are many remotes present (say, 10 or more) and there is a "dense" coverage (each remote covered by, say, 12 or more hubs) this may generate an excessive number of packets for the hub-master to process and it may fall behind. This can result in delays and packets being dropped.

With this release the slave-hubs monitor the traffic on the Ethernet and track which slave hubs are covering each remote and their signal strengths. Then only the four strongest slave-hubs forward their packets to the hub-master. Therefore, even with a "dense" coverage, there will not be an excessive number of packets for the hub-master to process.

This feature can be turned on or off with the command:

```
>rf-receive forward-all-packets=on or off
```

The default is off, meaning that the slave-hubs will limit the number of packets flowing to the hub-master. Turning the forward-all-packets to **on** can be useful during commissioning of a new deployment to verify the actual coverage of the track. The number of hubs that cover a specific remote is displayed in the “monitor-link” report, under the column “Rd”(redundancy) as shown below:

```
REMOTES:                                     % received by         Recent
                                     Total   N hubs (ever) (now)Hubs  RSSI
  #   Name     S/N        Since     Replies  3+  2   1   0   0 Rd Hi-Max Min
 -- --------- ----- --------------- ------- --- --- --- --- -- -- -- --- ---
  3 rmt-17010 17010        00:20:40   45588 99. .01 .00 .00  0  3  2 -43 -65
```

With the forward-all-packets=on the redundancy number is the true receive redundancy. With the forward-all-packets=off that number will be clipped at 4 or 5. You only need to issue this command at the hub-master, all slave hubs will change their setting to match.

This release also has a new command “>monitor-ethernet” which reports the number of packets received and transmitted on the Ethernet, by that radio, over the past one second. It displays the total number of packets and breaks them down by their destination address: Unicast, Broadcast, Multicast1 and Multicast2.

Version: 3.32
Date: 19 May 2018

Problems Remaining:

Do not use this build – the command “>show radios” issued at a remote radio can induce a reboot of a hub (fixed in version 3.33)

Version: 3.33
Date: 25 Oct 2018

New Features:

Problems Fixed:

Fixed the problem introduced in version 3.32 where the command ">show radios" issued on a remote could induce a reboot of a hub.

After being up for over 8 or 16 months the date/time tracking had a rollover problem and displayed the date incorrectly. This is now fixed.

Dropped the priority of ROAMPROXY event log message to avoid filling the event log from normal movement of the trains between clusters.

Short packets were being padded to 64 bytes, now to the minimum of 60 (had no bad effects).

Version: 3.34
Date: 06 May 2019

New Features:

Using the ">time" or ">date" commands to set a new time/date in one radio in the network (hub or remote) now triggers a packet transmission to propagate the new time/date to all the radios in the network.

Introduced a new option that limits the maximum RF transmit power to 20 dBm for compatibility with EU requirements ("ETSI bit")

The ">memory" command now shows a low water mark of free buffers (IRP buffers).

Problems Fixed:

Certain NTP packets ("Mode 7 extensions") could cause the radio to set its time/date incorrectly. It is now fixed.

An assert could cause a reboot in slave-hubs when the number of hubs in the cluster changed suddenly:

```
Assertion cluster_n_dhubs <= (hi_cluster_idx + 1) failed in hub_mstr.c
```

This no longer causes a reboot, only an event being logged.

The event "Pkt from non-existing rmt x" could be logged repeatedly and incorrectly at a slave hub that had not seen that remote directly. Introduced in version 3.31, has no bad effects, it is now fixed.

Version: 3.40
Date: 26 Feb 2020

Compatibility:

When upgrading to this version all hubs in a cluster should be upgraded at the same time. However, at the remote radios you can run an older version (3.3x) with version 3.40 on the trackside.

Known Problems:

This version does not run properly in the Classic radio model (AR24027). Use at least version 3.41 or later.

New Features:

1. Media Converter failure recovery enhancements: The Ethernet network connecting the various hubs in a cluster is often implemented with fiber-optic links. In these deployments each Ethernet link includes media converters at both ends of the link. In a deployment in Singapore it was found that these media converters could go into a failure mode dropping a large number of Ethernet packets, which in turn would cause the cluster network to become unstable with multiple hubs assuming the master role.

Version 3.30 introduced a mechanism to detect when Ethernet packets are being dropped, and for the affected hub to become "non-master-capable". However, in spite of this mechanism, there are more complex media converter failure modes that were still causing the cluster to become unstable for periods exceeding 40 seconds. This version introduces several enhancements to deal with those failure modes:

1.1 Alternate master "sticky" selection: While the network is running, the various hubs pre-select a hub that will assume the master role if the master were to drop out (this selection can now be monitored with the ">show radios" command where both the **master** and **alternate** are identified – see example below). In version 3.30 any hub behind a failed media converter might momentarily assume the master role and then drop off when its measurement of packet drops reached the threshold to make it non-master-capable. Now, if the hub is not pre-selected as the alternate, it will not attempt to become the master for a short period, allowing the packet drop measurement algorithm to catch up and make it non-master-capable before it tries to become the master.

1.2 Illegitimate master: If the media converter having problems is the one connected to the **alternate** master, that hub will still attempt to become the master. As part of being master it may also become the heartbeat source. In version 3.30 hubs accepted all heartbeat packets, from any source. This was (usually) not a problem because the cycles of both masters remained synchronized. But the rare cases when the cycles of the two masters were not in sync triggered the unstable events that persisted in version 3.30. In this version the hubs identify the new master as illegitimate and discard all packets received from it, including heartbeat packets.

1.3 Packet drop measurement enhancements: In version 3.30, when the cycles of the two masters were not in sync, all other hubs would restart their cycles, causing the true master to make an error in its measurement of packet drops and possibly become non-master capable. The packet drop measurement was enhanced such that the master will not be fooled by the other hubs resetting their cycles.

1.4 Multiple Master-hubs: This is explained as a separate heading below.

1.5 Transmit Ethernet packets drops detection and recovery: All the mechanisms described above detect and recover from packet drops arriving at one specific hub. Packet drops in the opposite direction (transmitted by a hub into the network) had not been addressed in version 3.30. This version introduces a mechanism to detect and recover from those Ethernet failures.

Packet drops on traffic transmitted by a non-master hub have little impact. As those drops increase, the inbound traffic from remotes will not reach the master hub. As a result the master will use that hub less and less to transmit outbound traffic to those remotes. That hub may be dropped altogether and, if the system is designed with redundancy, it will keep running.

On the other hand Ethernet packet drops of packets transmitted by the master hub have a very severe impact. In every cycle the master multicasts two key packets (the inbound and outbound schedules) to all the hubs. If either of those are dropped no hubs will transmit in that cycle. As the drop rate increases the network will stop functioning but the hub-master remains unaware of the problem and it will persist. This situation is very apparent when examining the event logs of any non-master hub which will be full of the event indicating that schedule packets were missing.

This version introduces features that detect when packet drops are occurring, and, in the case of a master, recover from the situation. These are described below:

<u>Packet drop measurement:</u> All hubs multicast a sync_status packets once per second. These packets now include a packet sequence number. Each hub uses this sequence number to compute the average packet drops on the packet flow from each of the other hubs. These measurements are displayed in the “>show radios” output which now includes a new column, Ether Drop, shown below:

```
mst-30005 #> show radios

CLUSTER HUBS (redundancy 1):
                                      Capable   Ethr      Qd Tx
  #   Name     S/N        Since       Mas/Sla   Drop      bytes
 -- --------- ----- ---------------   --- ---   ----   ---------
  0 mst-30033 30033        00:00:03    Master     0%
  1 hub-30041 30041        00:00:03   yes yes     0%
* 2 hub-30005 30005        00:00:03   yes yes     0%          0
  3 mst-30038 30038        00:00:04    Altrnt     0%
  4 hub-30307 30307        00:00:04   yes yes     0%
  5 hub-30297 30297        00:00:05   yes yes    35%
  6 hub-41877 41877        00:00:06   yes yes     0%
  7 hub-44016 44016        00:00:06   yes yes     0%
```

To compute the drop rate from the master the hubs use the two schedule packets that the master hub transmits in every cycle. These come at a much higher rate than the sync_status packets so this drop rate is more accurate and recent. When a hub measures a packet drop rate from the master higher than 10% it indicates this fact in the sync_status packet. The master hub collects this information, and, if it finds that **all** hubs are reporting this issue it resigns the master role allowing the alternate master to take over. It is important for the alternate master to be configured with the same node type as the master (both as "hub", or both "master-hub"), otherwise the hub with the faulty Ethernet will take over the master role again after a timeout of 30 seconds.

Since packet drops from a non-master hub have a minor impact in performance, a faulty media converter associated with those hubs might go undetected for a while. You should collect the ">show radios" output from all hubs on a regular basis. The faulty media converter can be identified when all hubs report a sustained non-zero Ether Drop originating at the same hub. In the example above, if the reports from all hubs show an equally non-zero packet drop rate from hub-30297, it is very clear that a media converter associated with that hub is faulty. But note that with a very high packet drop rate (~50% and above) the hub may not show on the list. So you must also check that all hubs in the network are accounted for in the ">show radios" list.

2. Multiple Master-Hubs: You can now configure more than one hub as the **master-hub** in a cluster. When two hubs are configured as master-hub they will select among them the one that will assume the master role with the other becoming the **alternate** master as described above. If the unit that assumed the master role subsequently fails or reboots the alternate assumes the master role. This is no different than if you had only one master, and the **alternate** was a unit selected among the generic hubs. The difference is that when the first unit comes back online it will join the cluster and become the **alternate** master without disrupting the current master.

Previously you could configure only one unit as the master-hub and you may still choose to do that. The pros and cons between having one or two units configured as master-hub are the following:

a) With a single master hub, if that unit reboots there are two master transitions, first when the unit drops, and then when the unit joins the network again. Even though a master transition is brief it still causes a drop in the communications with the remotes for a couple of seconds. With two master hubs there is only one transition (when the first master drops out).

b) If a master hub happens to have a problem in its media converter resulting in a moderate drop of Ethernet packets, that unit will relinquish the master role. If the packet drop rate subsequently recovers, if that hub is the single master-hub, it will take over the master role again. This can go on for a while as the media converter behavior fluctuates. But if there are two master-hub units, the second one will not relinquish the master role avoiding the unit with a faulty media converter from ever becoming master.

c) With a single master hub there is only one unit that you need collect logs from on a regular basis to get the logs from the master. When two units are configured as master you need to collect the logs from both (or find first which one is the master at that time).

We recommend configuring two radios as master-hub, mainly for reason b) above.

3. RF overlap between clusters: When two clusters operate on the same channel and there is some (unexpected) RF overlap, a remote radio can hear from hubs in both clusters. Previously the remote might attach to the distant cluster for a short time until collisions from the closer hubs would force a disconnect. Now, during the attachment process, if the remote hears from the closer cluster (with a much stronger signal) it aborts attaching to the far one to attach to the closer one.

4. Broadcast traffic: The "Ethernet" command includes an option, "block-remote-broadcast", that prevents broadcast packets originating at a remote from being transmitted back over RF to reach all other remotes. In Singapore there are some areas deployed with two clusters with inter-cluster synchronization that share broadcast traffic. In that situation broadcast packets generated by a remote in one cluster reach the other cluster master over the Ethernet and are then transmitted over RF to all its remotes regardless of the block-remote-broadcast setting. With this version, if block-remote-broadcast is set, the two masters now flag the addresses of stations that are associated with all remote radios and will not transmit, over RF, broadcast packets originated by those stations. The addresses flagged this way will remain active for the time specified in the **ethernet multicast-timeout** parameter. This timeout must be set to at least 60 seconds for this feature to work (the new factory default for this parameter is now 600 seconds).

5. Media converter failure simulation enhancements: Modified the simulation of the Ethernet packet drops (available in engineering mode with the **ethernet** command) such that Econsole packets are never dropped, allowing to control the radio over the Ethernet, even when packet drops are being simulated. Added simulation of dropping packets in the opposite direction (transmitted from the radio to the network).

Also added a new engineering command, **delay xxx**, that introduces a delay of xxx ms to the command parser. This allows to build a script file with radio commands and introduce a delay between them. You can use this to build more elaborate failure modes of the media converter, varying the Ethernet packet drops dynamically.

6. Miscellaneous:

Added an event to track when the free memory buffer pool is getting below some low threshold and keep tracking it down. However, if the radio runs out of buffers, the unit will reboot and those events will be lost. This requires to have the logs collected regularly, to show the rate at which buffers are leaking.

Modified several of the events being logged for better visibility into the processes underneath. Several events now report the RSSI associated with RF transactions, others report the number of hubs in the cluster which is used to compute Ethernet packet drops.

Problems Fixed:

1. In a cluster some slave-hubs might repeatedly log the event:

```
133 Ether sync pkt delayed (3). tt: 28/51 (19)
```

which was being logged incorrectly, induced by a larger than nominal difference in the oscillator frequencies between the synchronization master and the slave-hub. The consequence was a less precise synchronization of the cycles between hubs. In this version we modified the FPGA to route a more precise clock to the cycle timer. As a result the cycle synchronization is more precise and this event is no longer logged (unless the heartbeat packet is actually delayed on the Ethernet).

2. When changing channels at a remote radio (from chan A to chan B) there was a possible race condition that could result in a delay in attaching to the new cluster by as much as 10 seconds. This happens only if the command to change channels is received during the one cycle while the radio is in the middle of attaching to the cluster in the current channel. These are the details:

   1. The remote radio was trying to attach to the current cluster (on chan A), and had received an **invitation-to-join** and just transmitted an **associate rqst** packet
   2. Immediately after this transmission and before it receives an acknowledgment on chan A, the user changes the radio receive channel to chan B.
   3. The remote radio now receives packets from the new cluster in chan B. Instead of associating to this new cluster it reports that there is a conflict of two clusters in the same channel. This could last as long as 10 seconds before giving up, and attaching to the new cluster on chan B

This only happened if the command to change channels is received in the cycle between transmitting an associate request (on chan A) and receiving its acknowledge.

This has now been fixed.

3. The master hub keeps a count of the remotes currently attached to its cluster. With a weak RF link, a remote can lose the connection and reattach frequently. Under those conditions there was a race condition where the count of remotes, kept by the master, might become below the actual number of remotes. Once this happens, when all remotes move away from the area the count of remotes would become negative, inducing a reboot of the master. This has been fixed.

4. The “display factory” command was not displaying the correct values of the Ethernet station timeouts. This has been fixed.

5. In a small number of units the factory calibration stored a parameter in error that resulted in high transmit power levels when the unit was set to power levels below 5 dBm. This and all future versions detect and correct this parameter. Once this firmware (or later) runs once, the problem will be fixed, even when running older versions of the firmware.

Version: 3.41
Date: 27 October 2020

Problems Fixed:

1. This version now runs in the Classic model (AR24027)
2. If the remote radio was commanded to go to one channel, started the association with a hub, then immediately commanded to another channel on which it received a packet from the different hub, two incorrect things could happen:

   a) the radio logged the event: "Two hubs on same channel"

   b) the radio might reboot

New Features:

In the >monitor-ethernet command added a "peak" value showing the maximum number of packets received or transmitted on the Ethernet over any previous one second.

Version: 3.42
Date: 22 December 2020

Modified the "show radios" output by adding a new "Role"column (Master or Alternate). In version 3.40 and 3.41 that information was in the "Capable Mas/Sla" column. New format is shown below:

```
mst-30005 #> show radios

CLUSTER HUBS (redundancy 1):
                                      Capable  Ethr             Qd Tx
  #   Name     S/N        Since       Mas/Sla  Drop   Role      bytes
 -- --------- ----- ---------------   --- ---  ----  ------  ---------
  0 mst-30033 30033        00:00:03   yes yes    0%  Master
  1 hub-30041 30041        00:00:03   yes yes    0%
* 2 hub-30005 30005        00:00:03   yes yes    0%                  0
  3 mst-30038 30038        00:00:04   yes yes    0%  Altrnt
```

Version: 3.43
Date: 26 January 2021

Modified the “show radios” output by adding an extra space before the “Capable Mas/Sla” column. Requested by Siemens for compatibility with their monitoring software. New format is shown below:

```
mst-30005 #> show radios

CLUSTER HUBS (redundancy 1):
                                      Capable  Ethr             Qd Tx
  #   Name     S/N        Since        Mas/Sla  Drop   Role      bytes
 -- --------- ----- ---------------    --- ---  ----  ------  ---------
  0 mst-30033 30033        00:00:03    yes yes    0%  Master
```

Version: 3.44
Date: 17 May 2021

Problems Fixed:

If the Ethernet connecting the hubs in a cluster consists of two or more switches, each of them connecting several hubs together, then, if you disconnect the Ethernet line between two switches and reconnect it a few seconds later, the following could occur:

1. As expected, when the switches get disconnected the hubs form two clusters, each with its own master.
2. When the switches get reconnected the two masters, if they were both configured as “masters”, will not give up the master role. They both declare the other as illegitimate and continue acting as a master indefinitely. The single cluster, operating with two masters, will not operate properly.

This scenario is now identified correctly by the hubs, and the master with the lower serial number gives up master role within a few seconds after the two sub-clusters were merged.

Version: 3.45
Date: 27 July 2021

Problems Fixed:

At a master radio with multiple remotes attaching and dropping off, on rare occasions, the event log was reporting “PMP Error” events. These could be a set of three, as follows:

```
16:49:12.470 24-Jun  236 PMP Error #24 (0, 116, 12)
16:49:12.470 24-Jun  236 PMP Error #32 (2, 0, 12)
16:49:12.470 24-Jun  236 PMP Error #35 (1, 124, 8)
```

or a string of many repeated events, over a longer period of time:

```
16:46:25.230 24-Jun  236 PMP Error #24 (0, 108, 24)
16:46:25.230 24-Jun  236 PMP Error #35 (0, 108, 24)
16:46:25.270 24-Jun  236 PMP Error #24 (0, 108, 24)
16:46:25.270 24-Jun  236 PMP Error #35 (0, 108, 24)
16:46:25.310 24-Jun  236 PMP Error #24 (0, 108, 24)
16:46:25.310 24-Jun  236 PMP Error #35 (0, 108, 24)
16:46:25.350 24-Jun  236 PMP Error #24 (0, 108, 24)
...
```

These was due to a dropped remote radio still having data to go out and competing for the outbound bandwidth with other (attached) remotes. In the first instance the master recovered with no data loss. In the second instance the recovery was not working and data to the attached remotes was being lost.

This is now fixed.

Version: 3.46
Date: 20 March 2023

Problems Fixed:

In the Singapore DTL installation, once in a while, a remote radio could reboot with the following event logged in the reboot reasons:

```
Assertion quicc->timer_tcn34.cascaded < tdm_evt_table[evt_tbl_idx].time failed in pmp_mac.c
at line 5044
```

This is related with the STEA changing the radio channel as the radio is in the middle of attaching to a hub on a different channel. Implemented a local recovery that will log an event and avoid the radio reboot which was causing the remote to stay offline for more than 10 seconds.

A hub with the 'min-cluster-size' set to 0, and without an Ethernet connection, should be master capable and assume the master role as a single hub. This got broken in version 3.40. It is now corrected.