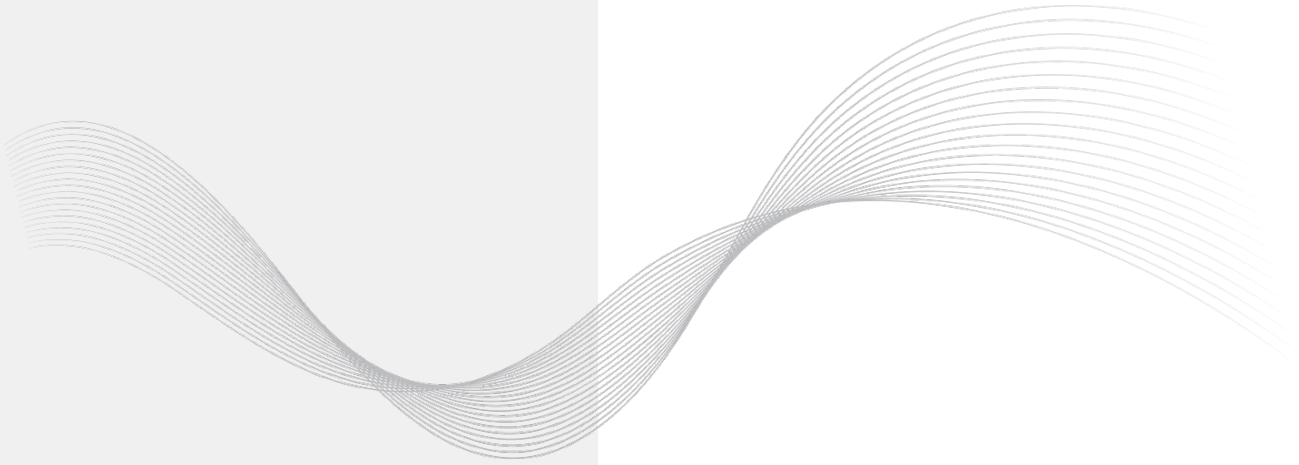

mint command

Table of Contents

1 General description

CAUTION

MINT firmware is not compatible with earlier RMA version. Don't try to update your current network to MINT version from RMA without prior documentation studying and laboratory testing.

MINT architecture gives a functionality to present a radio interface of a unit (as well as a network connected to it) as a traditional Ethernet in a bus topology. Therefore the unit can have several Ethernet interfaces and several pseudo-interfaces (tun, ppp, null etc). Any of Ethernet interfaces can be united in bridging groups which consist of two or more interfaces. Moreover, routing mode can also be used.

Full syntax:

```
mint IFNAME -type {mesh | master | slave}
mint IFNAME -mode {mobile | nomadic | fixed}
mint IFNAME -nodeid NUMBERID
mint IFNAME -name NAME
mint IFNAME -netid NUMBER
mint IFNAME -key SECRETKEY
mint IFNAME -authmode {public | static | remote}
mint IFNAME -[no]scrambling
mint IFNAME -[no]authrelay -[no]snmprelay -[no]extgw
mint IFNAME -[no]replicate [$ACL]
mint IFNAME -tpcmin {dBm|default} -tpcmax {dBm|default} -tpcadj
{+/-dBm|default}
mint IFNAME –autofactor 1..5 [3]
mint IFNAME -ratefall 0..8 [0]
mint IFNAME -[no]idfs
mint IFNAME -[no]autobitrate [+/-DB] | -fixedbitrate
mint IFNAME -minbitrate XX
mint IFNAME [-meshextracost N] [-extracost N] [-fixedcost N]
mint IFNAME -maxlinks N
mint IFNAME -mulcast [0..5]
mint IFNAME [-loamp N] [-hiamp N]

mint IFNAME -roaming {leader | enable [multiBS] | disable}
mint IFNAME profile N [-freq X[,Y,N-M,...] | auto] [-sid X[,Y,..]]
            [-band NN] [-bitr NN] [-miso | -mimo [greenfield |
legacy]]
            [-type {master|mesh|slave}] [-key XXX] [-nodeid N]
            [{-minbitr XXX [-autobitr [+/-dB]] | -
fixedbitr}]
```

```
            [enable | disable | delete]

mint IFNAME addnode [-defgw X.X.X.X] [-defmask X.X.X.X]
mint IFNAME addnode -mac X:X:X:X:X:X [-key STRING] [-note STRING]
[-maxrate N]
                    [-lip X.X.X.X] [-tip X.X.X.X] [-mask X.X.X.X]
                    [-lgw X.X.X.X] [-tgw {X.X.X.X | none}]
                    [-lcost XX] [-tcost XX] [{-setpri | -addpri}
NN | -1]
                    [-disable | -enable | -delete]

mint IFNAME delnode -mac X:X:X:X:X:X

mint IFNAME rcmd {-n ADDR|all} [-peer] [-self[2]] [-key KEY] [-t]
[-quiet]
            {"Command" | -file URL}
mint IFNAME -rcmdserver {disable | enable} [-guestKey STRING] [-
fullKey STRING]

mint IFNAME -odr hub
mint IFNAME -odr spoke [[-]connected [$ACL]] [[-]kernel [$ACL]]
mint IFNAME -odr disable | show

mint IFNAME poll {start [[-]qos] [[-]log] | stop | stat [clear]}
mint IFNAME monitor [-s] [-i SEC] [MAC [MAC ...]] | -[no]audio
[full] [-mac MAC]
mint IFNAME -airupdate {disable | {[active|passive]|force}}
[fast|normal|slow]

mint [IFNAME] map [routes | full | swg] [detail] [-a] [-m]
mint [IFNAME] info MAC
mint IFNAME ping [-n MAC -s LEN -swg N -p PRIO -i]
mint IFNAME -[no]long [detail]
mint -[no]colormap
mint rcmdserver -guestKey STRING -fullKey STRING

mint IFNAME start | stop | restart | clear
```

2 General command description

2.1 Setting the node type

Syntax:

```
mint IFNAME –type {mesh | master | slave}
```

The command sets the type of node.

Three node types are available:

- **MASTER:**

Master can establish connections with all other types of nodes. He is able to form a network of any topology with other masters or with nodes of mesh type.

On master node a marker access (polling) can be enabled. Only one master in a network segment can have this option enabled by means of which forming a star-topology segment (point-to-multipoint). With this, all other nodes break their connections with their respective neighbors (with exception of connections formed by join)

This type of nodes is usually used for static networks with no or small number of nomadic or mobile clients.

- **MESH:**

This type of node can be a part of a network with any topology. Establishes connections with master and mesh nodes. The difference between master and mesh is that master nodes will try avoid sending traffic of core transport network (master-master) through mesh nodes by setting the cost of master-mesh connection (from master side) higher (meshextracost parameter). Thus, mesh type of nodes can be used on mobile units where connection's parameters might change quickly in time.

Mesh nodes can work in a marker access node with master node being a polling master. With this, if master turns polling on, mesh node breaks all its connections with other nodes but the master (except join connections). If master node drops off or turns marker access off, mesh node restores its connections with its neighbors (if these connections existed).

- **SLAVE:**

Can only connect to the node with master type. When connection is lost, the device attempts to restore the connection to the master node. Slave node can work with master node using marker access in a classical point-to-multipoint topology.

Example,

```
mint rf5.0 –type master
```

2.2 Setting the node mode

Syntax:

```
mint IFNAME –mode {mobile | nomadic | fixed}
```

The command sets the mode of the node. The mode is defined by the application of the node for the network. Modes description:

- "*Fixed*" - the network node has a fixed allocation and never moves and never is switched off. This is a infrastructure node of the network
- "*Nomadic*" - node may change its physical allocation but all the data transmitting is made when the node is not moving (or moving very slowly)
- "*Mobile*" - the node may move and exchange data while moving.

Example,

```
mint rf5.0 –mode nomadic
```

2.3 Setting node sequential number

Syntax:

```
mint IFNAME -nodeid NUMBERID
```

The command sets the sequential number for the node. By default, it is set equal to the device' s serial number.

The number may be specified in the “XXX.YYY” format reflecting a part of the IP address (both “XXX” and “YYY” numbers can range from 1 to 255).

The parameter is optional.

Example,

```
mint rf5.0 –nodeid 5
mint rf5.0 –nodeid 123.112
```

2.4 Setting node name

Syntax:

```
mint IFNAME -name NAME
```

The command sets the name for the node. Node name will be displayed in "*mint map*" set of commands. Node name should not exceed 16 characters. Spaces in the node name are accepted if put between quotation marks.

Example,

```
mint rf5.0 –name My_node
mint rf5.0 -name "Master Unit"
```

2.5 Net ID

Syntax:

```
 mint IFNAME -netid NUMBER
```

The command sets the network system identifier (up to 8-digit HEX figure). It must be the same at both ends of the link.

2.6 Nodes authentication

Setting the secret key:

```
mint IFNAME –key SECRETKEY
```

This command sets the secret key for the current node. See different authentication modes descriptions below to learn how it is being used. The key can be up to 64 characters long and should not contain spaces (or should be put in quotes).

```
mint IFNAME -authmode {public | static | remote}
```

The command sets the type of nodes authentication.

There are three types of nodes authentication available:

- "*public*" – all nodes have the same key (password) for access. The simplest case of authentication. It can be used for small workgroups, point-to-point connections, mass public access networks and for MINT architecture testing purposes. Any two nodes of the network can establish a connection (given other settings are suitable) if their keys are equal. In public mode, having found a potential neighbor a node check for its information in the local database (defined by "*mint IFNAME addnode*" commands). If requested information is found, a key from a local database will be used. Otherwise, it is assumed that neighbor's key corresponds with node's own key ("mint IFNAME –key" parameter)
- "*static*" – every node has a full list of nodes (including their parameters and access keys) with which a connection can be established. This mode is suitable for an autonomous area of service with no need of centralized management and monitoring. Obviously, nodes that are included in each others access lists (local databases) should have a physical ability to connect to each other in order to establish a connection. In static mode each node must have a list of all permitted neighbors in a local database formed by a set of "*mint IFNAME addnode*" commands. If no information on the neighbor is found in the database the connection is being rejected.
- "*remote*" – centralized authentication mode with remote server (e.g. RADIUS or relay). In this mode any node can request the information from a remote authentication server (remote authentication server parameters are set using "*AAA*" command). This means that the node must have an access to this server (e.g. using IP).

A node having a local database of its neighbors or having an access to a remote authentication server can be configured as an authentication relay. For this purpose the following command is used:

```
mint IFNAME  –[no]authrelay
```

The information about authentication relay will be automatically distributed throughout the MINT network. Nodes which use remote mode of authentication but both do not have access to the remote server and do not have the information in their local database will use authentication relay in order to obtain the keys of potential neighbors.

```
mint IFNAME  –[no]authrelay
```

The information about SNMP relay will be automatically distributed throughout the MINT network. Nodes will use remote SNMP services.

Example 1:

Nodes A and B use the same key and can establish a connection with each other in public authentication mode.

- Node A

```
mint rf5.0 –key SECRETKEY
mint rf5.0 –authmode public
```

- Node B

```
mint rf5.0 –key SECRETKEY
mint rf5.0 –authmode public
```

Example 2:

Nodes A and B have different keys but they can establish a connection with each other using their local databases.

- Node A

```
mint rf5.0 –key SECRETKEY
mint rf5.0 –authmode public
mint rf5.0 addnode -mac B:B:B:B:B:B -key KEY2
```

- Node B

```
mint rf5.0 –key KEY2
mint rf5.0 –authmode public
mint rf5.0 addnode -mac A:A:A:A:A:A -key SECRETKEY
```

Moreover, each of these two nodes can set up connections with other nodes working in public mode if their keys correspond with each other.

Example 3:

Node A has a local database and acts as an authentication relay. Node B does not have a database and uses a relay in remote mode

- Node A

```
mint rf5.0 –key KEY1
mint rf5.0 –authmode static
mint rf5.0 –authrelay
mint rf5.0 addnode -mac B:B:B:B:B:B -key KEY2
mint rf5.0 addnode -mac A:A:A:A:A:A -key KEY3
```

- Node B

```
mint rf5.0 –key KEY2
mint rf5.0 –authmode remote
```

Node B will be getting neighbors' information via relay (Node A).

If Node A is switched to remote mode and there is no information in the local database, the authentication request will be forwarded to the remote server (if specified and accessible) or to another authentication relay.

2.7 Scrambling

Syntax:

```
mint IFNAME -[no]scrambling
```

Enables/disables the data scrambling to improve the connection stability.

2.8 Wireless subscriber stations “isolation” mode

Syntax:

```
mint IFNAME -[no]replicate [$ACL]
```

This feature allows you to do "isolation" of wireless subscriber stations from direct exchange of information with each other in switching mode.

If "*mint –noreplicate*" option is enabled on the base station then the traffic entering into a wireless network from wired segment of a subscriber station and coming to the base station from this subscriber station won’t be transmitted back to the wireless segment by the base station. He may return to the wireless segment only through an external wired switchboard connected to the base station.

The direct exchange is allowed by default (*mint -replicate*).

In addition $ACL list of “*num*” type may be specified (acl add $ ISOLATE num N1 N2 ...) with a list of switching group’s numbers for which you should enable or disable the listed feature (for all by default).

NOTE

This feature applies only to traffic entering a wireless network from the wired segment of a subscriber station. Inside a wireless network nodes are all accessible to each other at all times.

2.9 Setting ATPC thresholds

Syntax:

```
 mint IFNAME -tpcmin {dBm|default} -tpcmax {dBm|default} -tpcadj
{+/-dBm|default}
```

This command allows controlling ATPC (automatic transmit power control) function behavior. ATPC function is enabled/disabled by “rf <interface> pwrctl” command (see “*rfconfig*” command description).

- "*tpcmin dBm"*. This option sets the minimal transmit power level in dB ATPC function is allowed to set on the radio interface.
- "*tpcmax dBm"*. This option sets the maximal transmit power level in dB ATPC function is allowed to set on the radio interface.
- "*tpcadj +/-dBm"*. This option influences the optimal power level to be set on the radio interface by the ATPC function. The ATPC can be forced to set higher (tpcadj + <number in dBm>) or lower (tpcadj - <number in dBm>) power levels compared to the values it estimates itself.

2.10 Setting the connection factor

Syntax:

```
 mint IFNAME  –autofactor 1..5 [3]
```

This command sets the sensitivity of the device to establish a connection with a candidate via its radio interface. The more the “*autofactor*” value is the better should be the characteristics of the radio channel between the device and the candidate for establishing a connection.

Default value is 3.

2.11 Switching to automatic bitrate control mode

Syntax:

```
mint IFNAME –[no]autobitrate [+/-DB] -fixedbitrate
```

Enables/disables an automatic speed management mode.

In autobitrate mode every device controls the connection parameters independently (amplitude of the received signal, number of ARQs on transmitting, errors, SNR on the opposite side etc) and chooses such transmitting speed which provides necessary conditions for a reliable work with minimum number of ARQs and losses. Speed values can be different for each direction but it will be optimal.

When no autobitrate is used transmitting speed will be set according to the setting of “bitr” parameter of “*rfconfig*” command. When autobitrate is used, transmitting speed will be automatically adjusted according to current link conditions. The ranges of speed will be in between the setting of “bitr” parameter in “*rfconfig*” command (maximal speed) and “minbitrate” parameter (see below). If no “minbitrate” is specified the minimal RF interface speed will be taken as a lowest possible transmitting speed.

Minimal transmitting speed for autobitrate mode can be set with a help of the following command:

```
mint IFNAME -minbitrate BITRATE
```

Example,

```
mint rf5.0 -autobitrate
mint rf5.0 –minbitrate 9000
```

- "*+/-DB"* option influences the autobitrate function sensitivity. Autobitrate can be forced to set a higher bitrate (*mint IFNAME –autobitrate - <number in dB>*) even if the signal level is lower than expected on the specified number of dB. Or not to set a higher bitrate (*mint IFNAME –autobitrate + <number in dB>*) till the signal level won’t become higher than expected on the specified number of dB.

To disable the autobitrate mode the following command is used:

```
mint rf5.0 -fixedbitrate
```

In the fixedbitrate mode the actual bitrate is set with the “bitr” parameter of the “*rfconfig*” command.

"*mint IFNAME -ratefall 0..8"* command allows influencing autobitrate mechanism in the following way: it sets upper bitrate index threshold below which errors and retries checks are not performed, just energetic ability to upper bitrate is taken into consideration. Bitrate indexes are from 1 to 8 and correspond with bitrates available on the device's radio interface (to see bitrate list use «*rf rfX cap*» command). "0" ratefall's value cancels the command.

Example,

```
mint rf5.0 –ratefall 4
```

2.12 Cost setting

Syntax:

```
mint IFNAME [-meshextracost N] [-extracost N] [-fixedcost N]
```

- "*meshextracost"* – is configured for the interface. Sets an extra cost for all connections of master node with its mesh nodes. 500 – by default.
- "*fixedcost*" - this command will force all the costs for all the units connected to this unit to be fixed at the value specified in the command.
- "*extracost*" - is configured for the interface. Sets an extra cost for all connections on this interface. The value of the parameter is added to the cost automatically calculated by MINT protocol. Value of this parameter can only be positive. Zero value disables the parameter.

2.13 Max links

Syntax:

```
mint IFNAME -maxlinks N
```

This command sets the maximum allowed number of connected CPEs (in the case of radio connection) . When this value is reached, other attempts to connect to the base station will be rejected.

2.14 Transformation of "Multicast" to "Unicast"

Syntax:

```
mint IFNAME -mulcast [0..5]
```

[0..5] - the number of subscribers:

- Zero value disables the parameter;
- If value is not specified the transformation is always executed.

Default value is 3.

2.15 Setting signal levels thresholds

Syntax:

```
mint IFNAME [-loamp N] [-hiamp N]
```

- "*loamp"* - this option sets the minimal signal level for the neighbor. Signal level is measured in dB above the noise threshold for the current bitrate. If the level gets lower than specified value the connection with a neighbor will be lost. Default value - 2
- "*hiamp"* - this option sets the minimal SNR for a new neighbor. Signal level is measured in dB above the noise threshold for the current bitrate. If neighbor's signal level is equal or higher than a specified value the node will consider this neighbor to be a candidate. Default value – 6

Example,

```
mint rf5.0 –loamp 2
```

3 Frequency roaming

Syntax:

```
mint IFNAME -roaming {leader | enable [multiBS] | disable}mint
IFNAME profile N [-freq X[,Y,N-M,...] | auto] [-sid X[,Y,..]]
            [-band NN] [-bitr NN] [-miso | -mimo [greenfield |
legacy]]
            [-type {master|mesh|slave}] [-key XXX] [-nodeid N]
            [{-minbitr XXX [-autobitr [+/-dB]] | -fixedbitr}]
            [enable | disable | delete]
```

For a flexible management of frequency resource, higher noise immunity and throughput optimization InfiNet Wireless equipment supports frequency roaming capability based on MINT protocol.

Roaming is turned off by default – that means that the unit works using fixed radio interface configuration.

Any node of the network can be set up as a roaming leader. Roaming leader will define required radio frequency parameters of the wireless network. Roaming leader also works with a fixed radio interface parameters, however its radio parameters configuration is transmitted over the network in special packets so every node of the network knows whether it is connected to the roaming leader or to the network that has a roaming leader. If the network has several roaming leaders, their parameters should be identical. Roaming leader also supports DFS and Radar Detection features (if a special license is installed for selected countries).

Other network nodes can use roaming in order to search for the roaming leader or the network having a roaming leader (roaming enable). The search is implemented by switching between different sets of radio parameters that are defined in profiles. Each profile contains a fixed set of radio interface parameters which are set on each iteration of the search. Heuristic search algorithm can quickly evaluate general air media parameters and chooses the profile which defines the most suitable network.

The “*multiBS*” option enables the slave node to constantly check the link quality and try to find another BS if the quality become worse. When the option is disabled then if the link breaks the node will firstly try to reconnect to the same BS regardless of the link quality.

Profile parameters:

- "*freq X[,Y,N-M,...] | auto*" – radio interface frequency or list of frequencies. Auto keyword can be used - in this case all frequencies that the unit supports will be used
- "*sid X[,Y,..]*" – SID of the radio interface (or list of SIDs)
- "*bitr X*" – bitrate of the radio interface. Acts as a top limit for the bitrate if autobitrate mechanism is turned on
- "*band {double|full|half|quarter}*" – defines the channel width for the profile. If profiles use different channel widths, auto mode for frequency cannot be used
- "*type {master|mesh|slave}*" – node type
- "*key XXX*" – secret key
- "*nodeid N*" – node ID
- "*fixedbitr*" – sets fixed bitrate for the node
- "*minbitr XXX*" – minimum bitrate for operation in “autobitrate” mode
- "*autobitr [+/-dB]*" – operation mode with automatic bitrate control. [+/-dB] parameter allows to manage bitrate control sensitivity
- "*long {on|off}*" – enables/disables compulsory “long” mode
- "*enable | disable | delete*" – enables, disables or deletes the profile.

Examples,

```
mint rf5.0 profile 1 -freq 5920 -sid ABCDE
mint rf5.0 profile 2 -freq 5960 -sid ABCDE disable
mint rf5.0 profile 3 -freq auto -sid DEAD
mint rf5.0 roaming enable
```

4 Creating local nodes database

Syntax:

```
mint IFNAME addnode [-defgw X.X.X.X] [-defmask X.X.X.X]
mint IFNAME addnode -mac X:X:X:X:X:X [-key STRING] [-note STRING]
[-maxrate XX]
                    [-lip X.X.X.X] [-tip X.X.X.X] [-mask X.X.X.X]
                    [-lgw X.X.X.X] [-tgw {X.X.X.X | none}]
                    [-lcost XX] [-tcost XX] [{-setpri | -addpri}
NN | -1]
                    [-disable | -enable | -delete]
```

This set of parameters defines the nodes with which a node can work with. The following parameters can be specified:

- "*mac*". This parameter is mandatory. X:X:X:X:X:X is a MAC-address of the node with which a connection can be established.
- "*key*". Unique unit's key (key word or phrase up to 64 characters long; if contains spaces should be put into quotes). Used in authentication procedures. The same key should be specified in the settings of the connecting unit ("mint IFNAME –key").
- "*lip*". Local IP-address. This address will be assigned to this unit when the connection with a remote is established
- "*tip*" and "*mask*". Target IP-address and mask. This address will be assigned to the remote side when a connection is established. The mask is applied to both Local IP and Target IP. If mask is not specified these addresses will not be used
- "*lgw*". Local gateway IP-address (will be assigned to the local node once connection is established)
- "*tgw*". Target gateway IP-address (will be assigned to the remote node once connection is established). None option forbids providing information about default gateway (that is set by "*addnode –defgw*" command) to the remote node.
- "*lcost*". Local cost of the connection to this neighbor from current node. If not specified, MINT will automatically calculate the cost
- "*tcost*". Target cost of the connection from this neighbor to the current node. If not specified, MINT will automatically calculate the cost. If lcost and tcost parameters are set on a pair of neighbors, lcost has a higher priority.
- "*enable/disable/delete*". Self-explanatory – enables, disables or deletes a record in a local database.
- "*maxrate*". Target node maximum bitrate in kilobit per second.
- "*setpri | addpri*". This options allows setting/increasing the priority of packets passing through to the specified node. "*Setpri*" parameter is used to change a priority to the value specified in the command. When using "-1" value a package priority is dropped to the lowest priority. "*Addpri*" is used to change a priority only in case it is higher than the previous one (Note: the smaller is the value the higher is the priority). So you can only increase priority using "addpri" parameter.
- "*note*". This option allows making some word note (description) for the specified node.

Example,

```
mint rf5.0 addnode –mac 000028BAF234 –lip 1.1.1.1 –tip 1.1.1.2 –
mask 255.255.255.252 –lcost 120
```

For easy subscriber nodes definition in the local database of the given node (Base Station) "mint addnode" command is updated with two options: "*-defgw X.X.X.X*“ and “*-defmask X.X.X.X* ":

- "*-defgw X.X.X.X"*. Sets default gateway
- "*-defmask X.X.X.X"*. Set default mask.

When mask or gateway values are not defined for the subscriber node then default gateway or default mask will be used for this node. Thus, to add a subscriber node to the local database it is enough to define MAC-address (mac), target IP-address (tip) and a key:

```
mint rf5.0 addnode -mac 000435567322 -tip 10.1.1.1 -key SecretKey1
```

If a key is not specified for a subscriber node then Base Station’s key is assigned to this node.

CAUTION

Information about default gateway (that is set by «*addnode -tgw / addnode –defgw*» commands) is not provided to a subscriber node in case IP-address and network mask is not specified.

5 Deleting a node from local database

Syntax:

```
mint IFNAME delnode -mac X:X:X:X:X:X
```

The command deletes a record created using “*addnode*” command with a corresponding MAC-address.

Example,

```
mint rf5.0 delnode –mac 000028BAF234
```

6 Remote command management

Syntax:

```
mint IFNAME rcmd {-n ADDR|all} [-peer] [-self[2]] [-key KEY] [-t]
[-quiet]
            {"Command" | -file URL}
mint IFNAME -rcmdserver {disable | enable} [-guestKey STRING] [-
fullKey STRING]
```

Remote command management allows one MINT node to perform commands on one other or all MINT nodes in the network.

Options:

- "*-n ADDR|all*" – MAC-address of the destination node or access to all MINT nodes
- "*-peer*" – performs commands only on the nodes that is connected to the given device directly
- "*-self[2]*" – performs commands also on the device itself
- "*-key KEY*" – access key
- "*-quiet*" – disables writing replies from remote devices to a system log
- "*Command" | -file URL*" – command to be performed on the remote unit or root to a command txt file by ftp
- "*-rcmdserver {disable | enable}*" – disables/enables remote control management mode (enable by default)
- "*-guestKey STRING*" – guest key. Guest key allows to perform read only commands on the node
- "*-fullKey STRING*" – full key. Full key grants full access to the node (all commands can be performed)

Examples:

```
mint rf5.0 rcmd -node all -cmd "co sh"
mint rf5.0 rcmd -node all -file ftp_name:ftp_pswd@192.168.100.21/1
.txt
```

7 On-Demand Routing

On-Demand Routing (ODR) protocol is a structure of the MINT protocol. It provides dynamic routing capabilities in stub networks without the use of any routing protocol.

The main advantage of using ODR is the increase in the available bandwidth of your network by eliminating the service traffic of a separate routing protocol whilst still maintaining dynamic routing functionality. The ODR protocol propagates IP prefixes on the Layer 2 using the MINT protocol.

ODR is applicable only for stub networks of the hub-and-spoke topology, when all nodes (spokes) are connected only to a hub node. An example of the hub-and-spoke network is a simple "Point-to-Multipoint" network where each subscriber station has the only wireless connection to the Base Station.

Syntax:

```
mint IFNAME -odr hub
mint IFNAME -odr spoke [[-]connected [$ACL]] [[-]kernel [$ACL]]
mint IFNAME -odr disable | show
```

- "*-odr hub*" - sets a unit as a hub
- "*-odr spoke*" - sets a unit as a spoke
  - “*connected*” option allows announcing IP addresses/networks set on the spoke’s own interfaces
  - “*kernel*” option allows announcing static routes (set with the “*route add*” command).

Also one can specify a list of IP addresses and networks in the command using Access Control List (“$ACL”). Please see the description in the corresponding chapter "Access Control List" of this manual.

To show the current state of the protocol and established connections the following command is used:

```
mint IFNAME -odr show
```

To disable the "*on-demand routing"* on the unit the following command is used:

```
mint IFNAME -odr disable
```

8 Switching to marker access mode (polling)

Syntax:

```
mint IFNAME poll {start [[-]qos] [[-]log] | stop | stat [clear]}
```

The command turns on/off polling mode for the master station.

Polling mode is a method of accessing common radio channel under master station control, which consists in centralized distribution of transmission authorization markers by a master station to slaves. This mode greatly improves operational stability and throughput of master stations under conditions of heavy load and signal level misbalance between different slave units. It is particularly useful when slaves units are at long range from a base station and not in the direct visibility of each other, so that they cannot avoid mutual collisions in the radio channel by listening each other's transmission. The polling regime makes it possible to establish reliable communication between subscribers when the ordinary CSMA/CA wireless access method does not work at all.

Despite a slight decrease in the maximum transmission speed, the polling mode substantially increases the total throughput of a base station and provides for its fair distribution between client units. The polling algorithm is so designed as to minimize the protocol overhead while maintaining high efficiency and robustness.

The polling mode is enabled on the master station only. Configuration of client units needs not to be modified.

For fine tuning of polling regime there are three optional parameters which can be set using the following syntax:

```
mint IFname poll start [mi=XX] [ub=XX] [mt=XX]
```

- "*mi*" - marker interval. The primary time unit used in calculating the marker sending frequency. The approximate value is a half of the round-trip delay for the interface. Values are given in milliseconds, from 4 to 20
- "*ub*" - upper bound. Marker sending interval upper bound. This value provides a minimal guaranteed marker sending frequency. This parameter is chosen based on the compromise between the total number of markers loading up the channel for no purpose and the response time for the first key pressed in telnet program. The value is within 3 and 1000ms
- '*mt*' - marker timeout. The maximum waiting time for a client unit response to a marker or data packet. Expressed in milliseconds; default value is 120ms.

Example,

```
mint rf5.0 poll start ub=250
```

9 Continuous signal levels monitoring

Syntax:

```
mint IFNAME monitor [-s] [-i SEC] [MAC [MAC ...]] | -[no]audio
[full] [-mac MAC]
```

If no MAC-addresses are specified, the output of command will contain the information about all neighbors and candidates of the current node.

Instead of MAC-addresses “*nodeid*” and/or “*name*” of the node can be specified.

The sample output of the command is presented below.

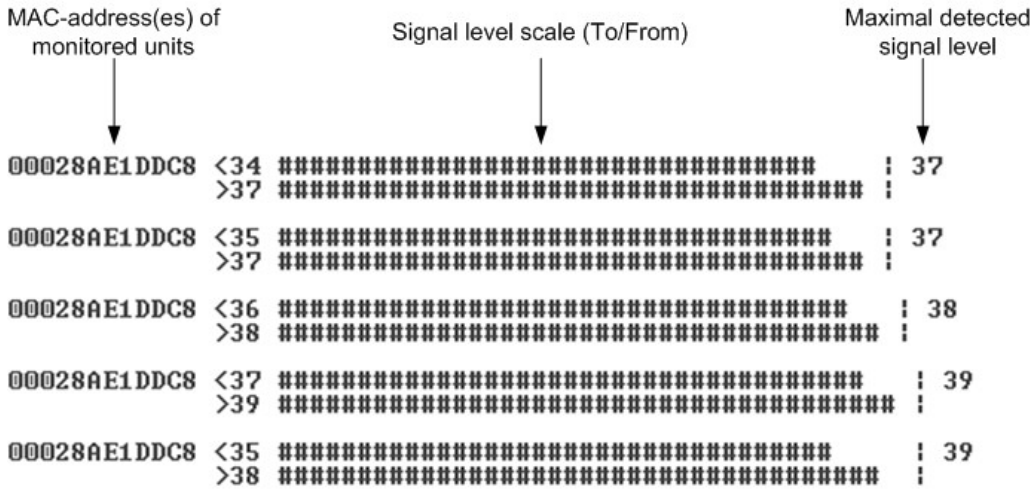

Figure - Sample output

- "*-s*" option keeps the output within one screen without line-by-line output
- "*-i SECONDS*" set the interval for information output in seconds.
- *"-[no]audio*" option enables/disables sound indication (AudioMonitor mode)
  - "*full*" - in this mode audio monitoring is performed according to a maximal receiving level among neighbors and candidates. If "*full*" parameter is not specified an arithmetical mean of receive and transmit signal is calculated (only for neighbors). The neighbor for which this value is maximal will be indicated in audiomonitor.
  - "*-mac MAC*" option allows to perform audio monitoring only for one neighbor with the specified MAC-address.

"*mint IFNAME monitor -audio*" command starts working immediately and can be saved in the configuration which allows using it even after unit's reboot.

10 Automatic over-the-air firmware upgrade

Syntax:

```
mint IFNAME -airupdate {disable | {[active|passive]|force}}
[fast|normal|slow]
```

This set of commands manages the Automatic Over-the-air Firmware Update system.

10.1 What is it?

The AirUpdate system provides with an easier ways of massive firmware upgrade in the MINT network for a big number of the nodes (same type). In order to do that only one unit of each type should be manually (or through the scheduler) upgraded – other units will get new firmware automatically.

10.2 How does it work?

Every unit can be configured for AirUpdate in passive or active mode. Active units periodically (every 30 minutes) announce the information about their firmware to the MINT network. The information includes the version of the firmware and the time of uninterrupted (without reboots) work with this version. All units of MINT network (both active and passive) receive and accumulate the information from active units choosing the source with the latest version of firmware with which the source has worked for the longest period of time.

After the period of information accumulation passive units send their requests for the new firmware to the chosen source. Active units form a list of requests and perform a group distribution of the firmware using special MINT multi-address distribution protocol.

The period of information accumulation can be changed using fast, normal and slow parameter of the command.

In fast mode the unit will wait for the potential source of the firmware to work with new version within two hours with no reboots. Only after two hours the request will be sent.

In normal mode the waiting period is 7 hours; in slow – 24 hours

By default, passive normal mode is turned on.

For immediate firmware upgrade there is a special option “*force*”. The command is not saved in the configuration and acts as a signal for all units to send their requests for upgrade regardless the work mode and information accumulation time period.

If during the process of new firmware distribution an error occurs (or link loss) the passive unit will stop the upload process and will resend its request after getting an announcement.

Examples:

The unit is in the active mode sending announcements about new firmware version. If units with newer firmware version are found on the network, the upload request will be sent in no less than 7 hours after uninterrupted work of the announcement source:

```
mint rf5.0 –airupdate active normal
```

The unit is the passive mode waiting for the source of the latest firmware version to work with it during no less than 24 hours:

```
mint rf5.0 –airupdate passive slow
```

The operator decides to immediately upgrade all the units with new firmware:

```
mint rf5.0 –airupdate force
```

The unit is not participating in AirUpdate process. It does not send announcements and does not generate requests for upgrades:

```
mint rf5.0 –airupdate disable
```

11 Managing MINT protocol

Syntax:

```
mint IFNAME start | stop | restart | clear
```

Example,

```
mint rf5.0 start
```

12 Join cost setting

Syntax:

```
mint IFNAME –joincost XX
```

- "*joincost"* – is configured for the interface. Sets the cost of all connections on the interface which were established by means of join (3 – by default). Zero value disables the parameter

Example,

```
mint rf5.0 –joincost 60
```

13 MINT log settings

The following command is used to control log settings for MINT protocol:

```
mint IFNAME -[no]log [detail]
```

Three different modes are available:

- No logging. "*-nolog*" option is used
- Limited logging. "*-log*" option is used. The messages on connecting/disconnecting neighbors will be put to the system log
- Detailed logging. "*-log detail*" option is used. Along with the messages from limited logging mode, messages on changing costs of the routes and changing bitrates (in autobitrate) mode will be put to the system log.

Example,

```
mint rf5.0 –log detail
```

This command will turn full logging on.

14 MINT protocol version

Syntax:

```
mint vers
```

The command shows current version of MINT protocol.

15 Joining interfaces

Syntax:

```
mint join IFACE1 IFACE2 …
```

Among all of exiting features of MINT architecture one can mention join capability which allows joining several interfaces of one unit into one mesh network.

Some units can have two or more radio interfaces of different types. Each of these interfaces may act as an independent MINT network node. However, the nodes from different networks will not be able to connect to each other as radio interface parameters will be different (frequencies, modulations or other parameters) and other limitations (authentication parameters, secret keys etc) will take place. JOIN function allows for two or more radio-interfaces of one unit to interconnect with each other as if they are two nodes of one network.

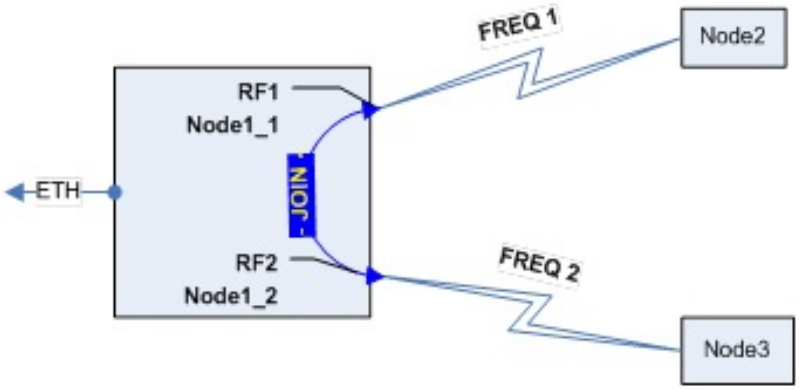

Figure - Join interfaces

```
mint join rf4.0 rf4.1
mint map
================================================
Interface rf4.0,
node 000000000011 "Node1_1" id:11 (mesh)

2 Neighbors:
00020 Node2            000000000002, Cost=40 , I/O=24/27 <36/36>
/mesh/
00012 Node1_2          000000000012, Cost=3  , I/O=0/0   <0/0>
/join/

Interface rf4.1,
node 000000000012 "Node1_2" id:12 (mesh)

2 Neighbors:
------------
00020 Node3            000000000003, Cost=40 , I/O=24/27 <36/36>
/mesh/
00030 Node1_1          000000000011, Cost=3  , I/O=0/0   <0/0>
/join/
```

As you can see from the example, each interface shows two neighbors. In reality the exchange of data between interfaces does not involve their physical activity, energetic parameters of the connection (signal levels and data rates) are not shown (zero). And this kind of connection has a constant and a very low cost.

In order to disjoin interfaces to make them independent, the following command is used:

```
mint disjoin
```

16 Pseudo radio interface (prf)

MINT protocol can work not only via radio but via wired Ethernet interface. “*prf*” pseudo radio-interface is used for this purpose. This interface can be “attached” to the physical interface like it’s done with vlanX interfaces.

```
prf 0 parent eth0
ifconfig prf0 up
```

Please find more information on prf interface in “*prf*” command description.

Such pseudo radio-interface can be configured as a MINT network node and even can be joined with other interfaces. From MINT protocol point of view, this pseudo interface will look like a usual radio interface through which a node can find neighbor and establish a connection with it.

```
mint prf0 start
mint join rf4.0 rf4.1 prf0
```

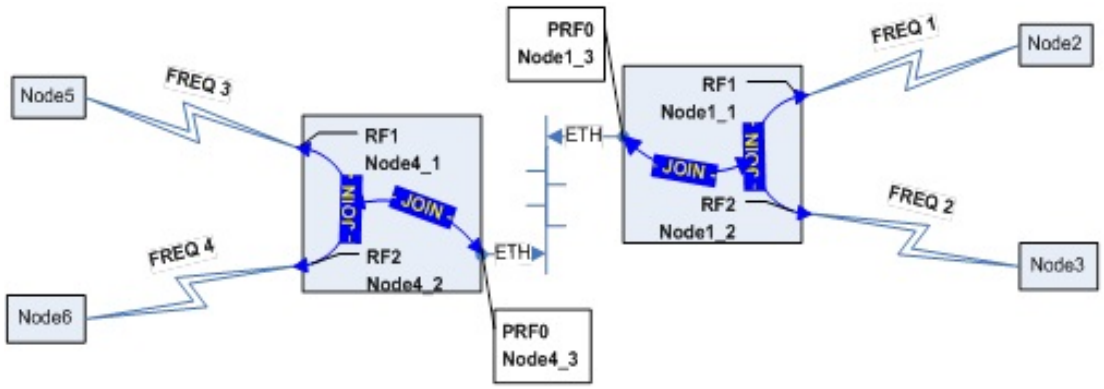

Figure - Several distributed network segmenets join

In this example we have joined several distributed (possibly even geographically distributed) network segments. Combining “joins” for different radio interfaces and pseudo radio interfaces the network will obtain a good number of alternative paths in any direction thus providing with the best delivery quality and less bottlenecks.

NOTE

If several interfaces are used with “join” function, when configuring switch groups only one of those joined interfaces should be mentioned in switch group configuration.

```
mint join rf4.0 rf4.1
switch group 1 add eth0 rf4.0
```

17 Learning the network neighbors

The following command is used to learn the state of the connections with neighbors:

```
mint IFNAME map [routes | full | swg] [detail] [-m]
mint -[no]colormap
```

Three options are used:

- **The default output is displayed**:

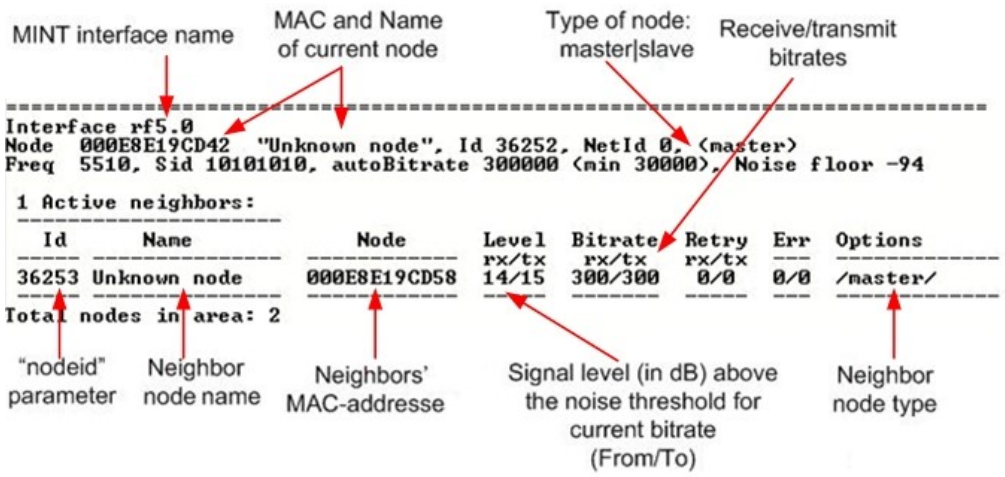

Figure - Default output

- **Routes**. The following output is displayed:

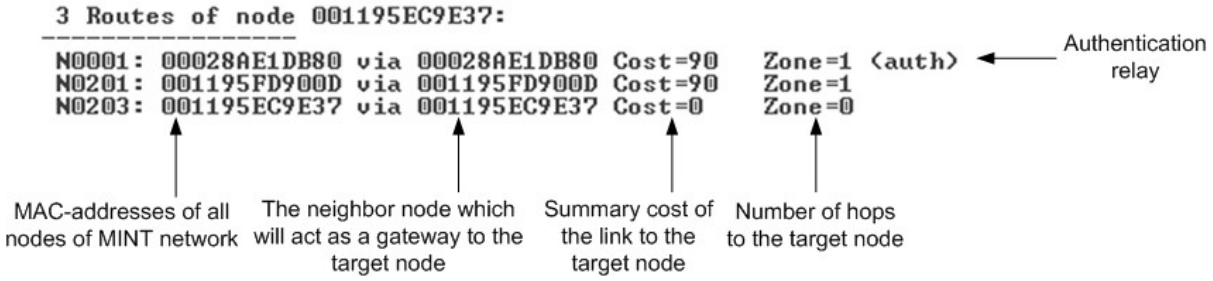

Figure - Routes output

- **Full**. A combination of “neighbors” and “routes” modes
- **Swg**. This option is used when switching groups are created in MINT network. It shows in which switching groups neighbor nodes are included:

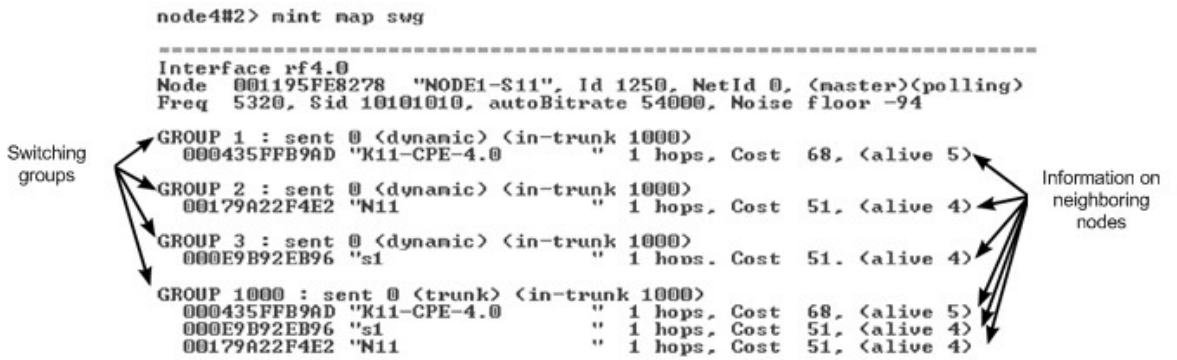

Figure - Swg output

Parameters:

- "*-detail*" – shows the following information for each link with a neighboring node: distance do the neighboring node in kilometers, load on receive/transmit in Mbps, on receive /transmit in packets per second, link «Cost», main IP-address of the neighboring node
- "*-m*" – shows I/O signal levels relative to minimal receive/transmit bitrate. (Without "*–m*" relative to current bitrates).

"Mint –[no]colormap" option enables/disables color indication of the command.

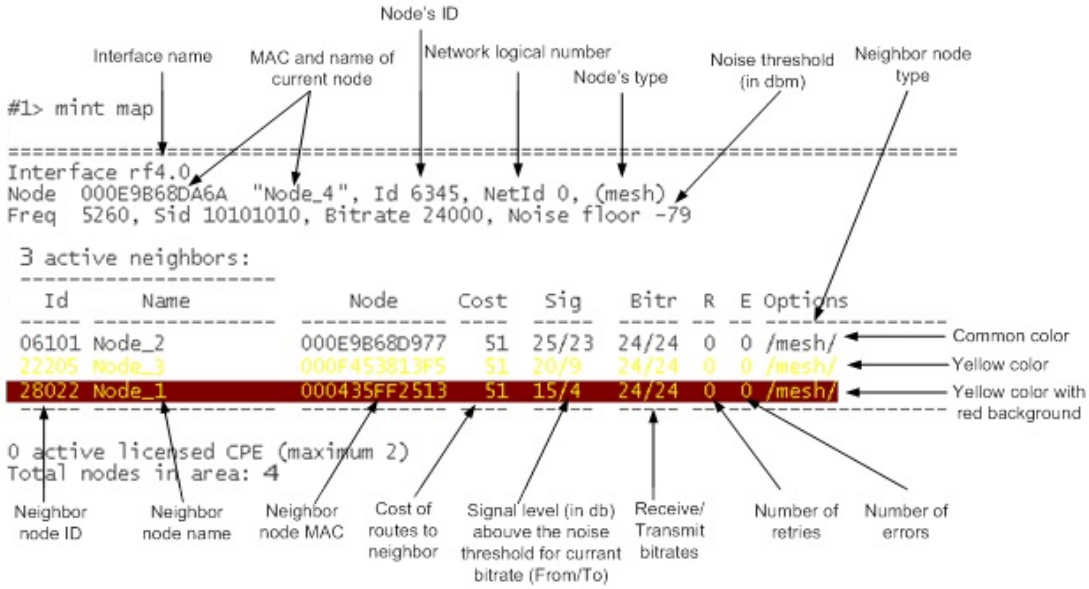

Figure - "Mint –[no]colormap" option

Color indication of «*mint map*» command output:

- **Common color** identifies neighbor nodes that have acceptable characteristics of a link to the current node.
- **Yellow color** identifies neighbor nodes that potentially may have problems with sustainability and quality of a link to the current node. In this case link quality can be improved through the change of certain parameters (for example, lowering bitrates).
- **Yellow color with red background** identifies neighbor nodes that have unsatisfactory characteristics of a link to the current node. For example, neighbor nodes that have low characteristics of a link on the lowest possible bitrate or have errors are marked this way. In this case link quality can be improved by such actions as antenna alignment, cable connectivity testing and so on.

When neighbor nodes are marking with certain color style it is not only signal level but also number of retries and errors are taken into consideration.

18 Trace command

Syntax:

```
mint IFNAME trace MAC
```

Trace command allows viewing information about node by its MAC address: its status, ID, name, cost, number of hops. This command also shows information about cost optimal pass to the node at the current time.

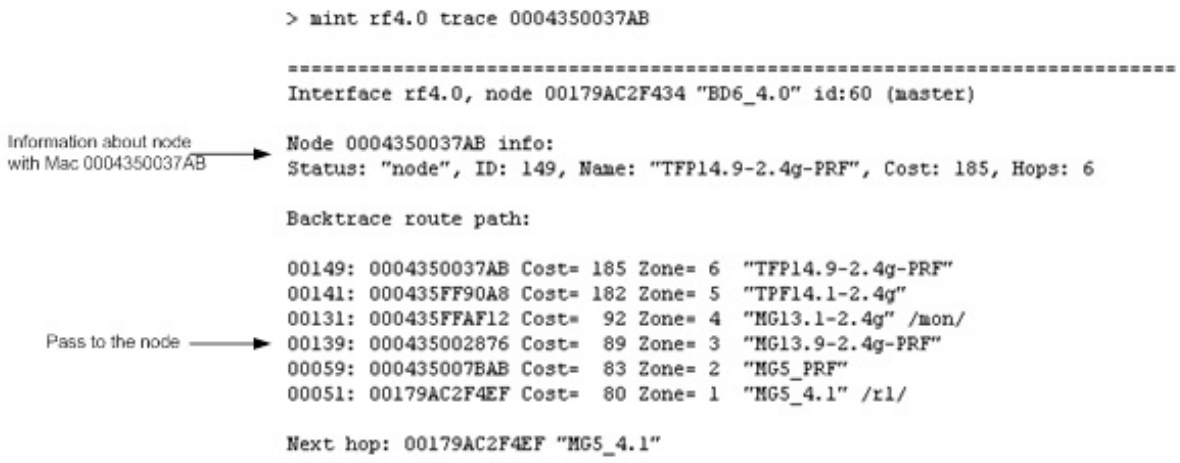

Figure - Trace command output

19 Upgrading current RMA network to MINT

CAUTION

It is recommended to study MINT technology features and test your basic configuration skills on the test devices before complying this instruction.

It is also recommended to consider principles of new network building in advance: will it be Mesh or "Point-to-Multipoint" network, how and what routing will be used, will it be a switched network?

1. Perform the following actions on each client device:
    a. Note down or remember MAC address of radio interface which is used to connect to base station. You can view it using command:

```
Ifc rf5.0
```

    b. Save current configuration (it is supposed that client device is connected to base station in the moment) using command:

```
config save
```

    c. Upload MINT firmware using command:

```
fl get user:password@server/file
```

2. Perform restart of all the client devices (one after another) using command:

```
restart y
```

3. Upload MINT firmware on the base station and restart base station

   After the restart all devices will start with MINT firmware. Upon discovering the old configuration (RMA) new MINT firmware will start neighbor search protocol on all radio interfaces of the device with default parameters (master, autobitrate, hiamp=4).

   Radio interface parameters will be delivered from the old configuration (at the time of its last saving).

   This is enougth for device to connect to MINT Network. The devices will lose IP address assigned by RMA.

4. Ensure all devices have connected to base station using “mint map” command.
5. Assign appropriate IP address on base station radio interface, for example,

```
ifc rf5.0 10.0.0.254/24 up
```

6. Using "*mint rcmd*" command assign IP addresses on radio interfaces of all client devices addressing them using MAC address (you can miss colon in MAC addresses), for example,

```
mint rf5.0 rcmd -node 17:9a:c2:f4:34 -cmd "ifc rf5.0 10.0.0.1
/24 up; co save;"
mint rf5.0 rcmd -node 17:9a:c3:ad:46 -cmd "ifc rf5.0 10.0.0.2
/24 up; co save;"
mint rf5.0 rcmd -node 179ab1f391     -cmd "ifc rf5.0 10.0.0.3
/24 up; co save;"
```

   ...

   and so on.

   After that all client devices will be accessible from base station by IP (if it wasn’t rejected by configuration options).

7. Using telnet perform the rest necessary configuration on each device for smooth network functioning (routing and so on)

If IP access is not obtained, it is always possible to verify or correct configuration of client devices using “*mint rcmd*” command:

```
mint rf5.0 rcmd -node 0023113231 -cmd "co show" -reply
```

with "*–reply*" parameter command result will appear in a local system log (sys log show).