Zbtool User Guide

Version	Changed Item	Author	Date
1.0	First version	Feng.He	2020.06.15
1.1	Add group、scene	Feng.He	2020.06.18
2.1	Add zdo request	Feng.He	2020.11.25

1.1 Description

Zbtool is a cmd line tool for debug gl-zigbee module which bases on gl-zb-api. You can use it for quick contoning and managing zigbee network and devices.

1.2 Command Table

Command	Need Parameters	Simple Description
help	NO	Get command list
get_module_message	NO	Get zigbee module message
get_nwk_status	NO	Get current network status
get_dev_list	NO	Get current child/neighbor device table
create_nwk	YES	Create a new zigbee network(as coordinator)
leave_nwk	NO	Destroy current zigbee network
allow_join	YES	Allow device join in current zigbee network
delete_dev	YES	Remove device from zigbee network
listen	NO	Listen the module_message callback
global-zcl	YES	Create and send a global zcl cmd
zcl-on/off	YES	Create and send a on-off cluster cmd
zcl-window_covering	YES	Create and send a window_covering cluster cmd
zcl-level_control	YES	Create and send a level_control cluster cmd
zcl-color_control	YES	Create and send a color_control cluster cmd
zcl-group	YES	Create and send a group cluster cmd
zcl-scene	YES	Create and send a scene cluster cmd
zdo-request	YES	Create and send a zdo request

1.3 Command Instruction

1.3.1 help

zbtool help

- Get command list

```
root@GL-S1300:~# zbtool help
help                        Get command list
get_module_message          Get zigbee module message
get_nwk_status              Get current network status
get_dev_list                Get current child/neighbor device table
create_nwk                  Create a new zigbee network(as coordinator)
leave_nwk                   Destroy current zigbee network
allow_join                  Allow device join in current zigbee network
delete_dev                  Remove device from zigbee network
listen                      Listen the module_message callback
zdo-request                 Create and send a zdo request
global-zcl                  Create and send a global zcl cmd
zcl-on/off                  Create and send a on-off cluster cmd
zcl-window_covering         Create and send a window_covering cluster cmd
zcl-level_control           Create and send a level_control cluster cmd
zcl-color_control           Create and send a color_control cluster cmd
zcl-group                   Create and send a group cluster cmd
zcl-scene                   Create and send a scene cluster cmd
root@GL-S1300:~#
```

1.3.2 get_module_message

zbtool get_module_message

- Get zigbee module message

```
root@GL-S1300:~# zbtool get_module_message
  mac: 14:b4:57:ff:fe:f1:5a:bf
  stack ver. [6.7.7 GA build 347]
root@GL-S1300:~#
```

1.3.3 get_nwk_status

zbtool get_nwk_status

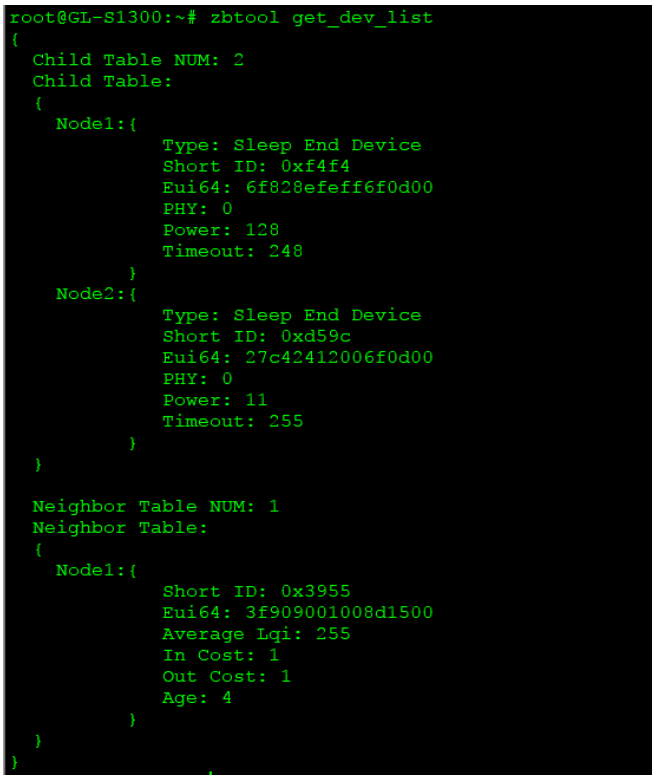
- Get current network status

```
root@GL-S1300:~# zbtool get_nwk_status
{
  NWK Status: joined network!
  Node Type: Coordinator
  Extended PAN ID: 016bb468d0c28f7c
  PAN ID: 0x1111
  Tx Power: 11
  Radio Channel: 11
  Join Method: Use NWK rejoin
  NWK Manager ID: 0x0000
  NWK Update ID: 0x00
}
```

1.3.4 get_dev_list

zbtool get_dev_list

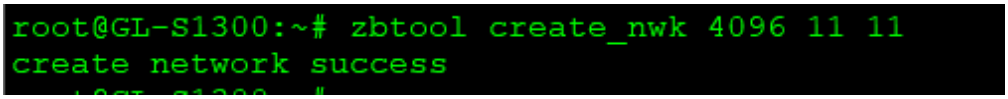
- Get current child/neighbor device table.
    - A child device is usually a zigbee-end_device or zigbee-sleepy_end_device mounted on the current device. A neighbor device is usually a zigbee-router_device.



1.3.5 create_nwk

zbtool create_nwk [panID:2] [channel:1] [txPower:1]

- Create a new zigbee network(as coordinator).
    - panID: Two byte pan ID
    - channel: One byte channel
    - txPower: One byte tx power –dbm–
        - The tx power can be set to 0-255, but the actual tx power is limited by the chip, so setting the tx power beyond the upper limit will not take effect.



1.3.6 leave_nwk

zbtool leave_nwk

- Destroy current zigbee network

```
root@GL-S1300:~# zbtool leave_nwk
leave network success
```

1.3.7 allow_join

zbtool allow_join [limitTime:1]

- Allow device join in current zigbee network
  - limitTime: One byte time allowed to access the network

```
root@GL-S1300:~# zbtool allow_join 180
allow device join success
```

1.3.8 delete_dev

zbtool delete_dev [target:2] [targetEUI64:8]

- Remove device from zigbee network
  - Target: Two byte target device short ID
  - targetEUI64: -IEEE_ADDRESS- The EUI64 of the target device

```
root@GL-S1300:~# zbtool delete_dev 30246 6f828efeff6f0d00
delete_dev success
```

1.3.9 listen

zbtool listen

- Listen the module_message callback.

There are three types of module message, zcl report message, zdo report message and zigbee device manage message.

- **Zcl report message:**

```
zcl_report!
{
  short ID: 7626
  profile ID: 0104
  cluster ID: 0019
  src_endpoint: 1
  dst_endpoint: 1
  cmd type: 00
  cmd ID: 01
  message length: 22
  message: 017c118711317600220100
}
```

- **Zdo report message**

```
zdo_report!
{
  short ID: eb24
  profile ID: 0000
  cluster ID: 0013
  message length: 22
  message: 24eb5dea9414006f0d0080
}
```

- **Zigbee device manage message:**

```
new device manage message!
{
  new device short ID: 7626
  new device eui64: 6f828efeff6f0d00
  parent of new device: 0000
  new device status: unsecured join
  coordinator decision: Allow the node to join. Send the key to the node.
}
```

```
new device manage message!
{
  new device short ID: 6461
  new device eui64: 6f828efeff6f0d00
  parent of new device: ffff
  new device status: left network
  coordinator decision: Deny join.
}
```

1.3.10 global-zcl

zbtool global-zcl [target:2] [clusterID:2] [commandID:1] [frameType:1] [data:x]

- Create and send a global zcl cmd.
  - target: Two byte target device short ID
  - clusterID: Two byte global cluster ID
  - commandID: One byte command ID
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast
  - data: Array of data

Note: API support all global zcl cmd, but zbtool only implemented “read attribute(0x00)” cmd now.

```
root@GL-S1300:~# zbtool global-zcl 30246 0 0 0 0000
tmp_data[4]: 0000
send zcl cmd!
```

Get cmd resp (listen):

```
zcl_report!
{
  short ID: 7626
  profile ID: 0104
  cluster ID: 0000
  src_endpoint: 1
  dst_endpoint: 1
  cmd type: 00
  cmd ID: 01
  message length: 10
  message: 0000002003
}
```

1.3.11 zcl-on/off

zbtool zcl-on/off [target:2] [commandID:1] [frameType:1]

- Create and send a on-off cluster cmd
  - target: Two byte target device short ID
  - commandID: One byte command ID
    - 0-off；1-on；2-toggle
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast

```
root@GL-S1300:~# zbtool zcl-on/off 59944 0 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-on/off 59944 1 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-on/off 59944 2 0
send zcl cmd!
```

1.3.12 zcl-window_covering

zbtool zcl-window_covering [target:2] [commandID:1] [frameType:1]

- Create and send a window_covering cluster cmd
  - target: Two byte target device short ID
  - commandID: One byte command ID
    - 0-down；1-up；2-stop
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast

```
root@GL-S1300:~# zbtool zcl-window_covering 30246 0 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-window_covering 30246 1 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-window_covering 30246 2 0
send zcl cmd!
```

1.3.13 zcl-level_control

zbtool zcl-level_control [target:2] [commandID:1] [level:1] [frameType:1]

- Create and send a level_control cluster cmd

Note: API support all zcl cmd, but zbtool only implemented “move to level(0x00)” cmd now.

  - target: Two byte target device short ID
  - commandID: One byte command ID
    - 0- move to level
  - level:
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast

```
root@GL-S1300:~# zbtool zcl-level_control 59944 0 100 0
send zcl level control cmd!
root@GL-S1300:~# zbtool zcl-level_control 59944 0 200 0
send zcl level control cmd!
root@GL-S1300:~# zbtool zcl-level_control 59944 0 10 0
send zcl level control cmd!
```

1.3.14 zcl-color_control

zbtool zcl-color_control [target:2] [commandID:1] [data:x] [frameType:1]

- Create and send a color_control cluster cmd

Note: API support all zcl cmd, but zbtool only implemented “Move to Color(0x07)” and “Move to Color Temperature(0x0a)” cmd now.

- Move to Color:
  - target: Two byte target device short ID
  - commandID: One byte command ID
  - x value: Two byte value of x
  - y value: Two byte value of y
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast

- Move to Color Temperature:
  - target: Two byte target device short ID
  - commandID: One byte command ID
  - color_temperature: Two byte value of color temperature
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast

```
root@GL-S1300:~# zbtool zcl-color_control 59944 7 100 100 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-color_control 59944 7 1000 1000 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-color_control 59944 7 1000 10000 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-color_control 59944 10 1000 0
send zcl cmd!
```

1.3.15 zcl-group

zbtool zcl-group [target:2] [commandID:1] [data:x] [frameType:1]

- Create and send a group cluster cmd
  - target: Two byte target device short ID
  - commandID: One byte command ID
  - data: Array of data (little endian)
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast

- Example: “add group” cmd

```
root@GL-S1300:~# zbtool zcl-group 34180 0 010000 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-group 59944 0 010000 0
send zcl cmd!
```

1.3.15 zcl-scene

zbtool zcl-scene [target:2] [commandID:1] [data:x] [frameType:1]

- Create and send a scene cluster cmd
  - target: Two byte target device short ID
  - commandID: One byte command ID
  - data: Array of data (little endian)
  - frameType: One byte frame type
    - 0: unicast; 1: multicast; 2: broadcast

- Example: “add scene” cmd

```
root@GL-S1300:~# zbtool zcl-scene 59944 0 01000101000006000100 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-scene 59944 0 01000201000006000101 0
send zcl cmd!
```

- Example: “recall scene” cmd

```
root@GL-S1300:~# zbtool zcl-scene 59944 5 010001 0
send zcl cmd!
root@GL-S1300:~# zbtool zcl-scene 59944 5 010002 0
send zcl cmd!
```

1.3.16 zdo-request

ZDO request cmd table

Command	Need Parameters	Simple Description
help	NO	Get zdo request list
match_descriptor	YES	Send match descriptor request
node_descriptor	YES	Send node descriptor request
power_descriptor	YES	Send power descriptor request
active_endpoints	YES	Send active endpoints request
simple_descriptor	YES	Send simple descriptor request
lqi_table_request	YES	Send lqi table request
routing_table_request	YES	Send routing table request
bind_request	YES	Send bind request
binding_table_request	YES	Send binding table request

1.3.16.1 zdo-request help

Get zdo request list

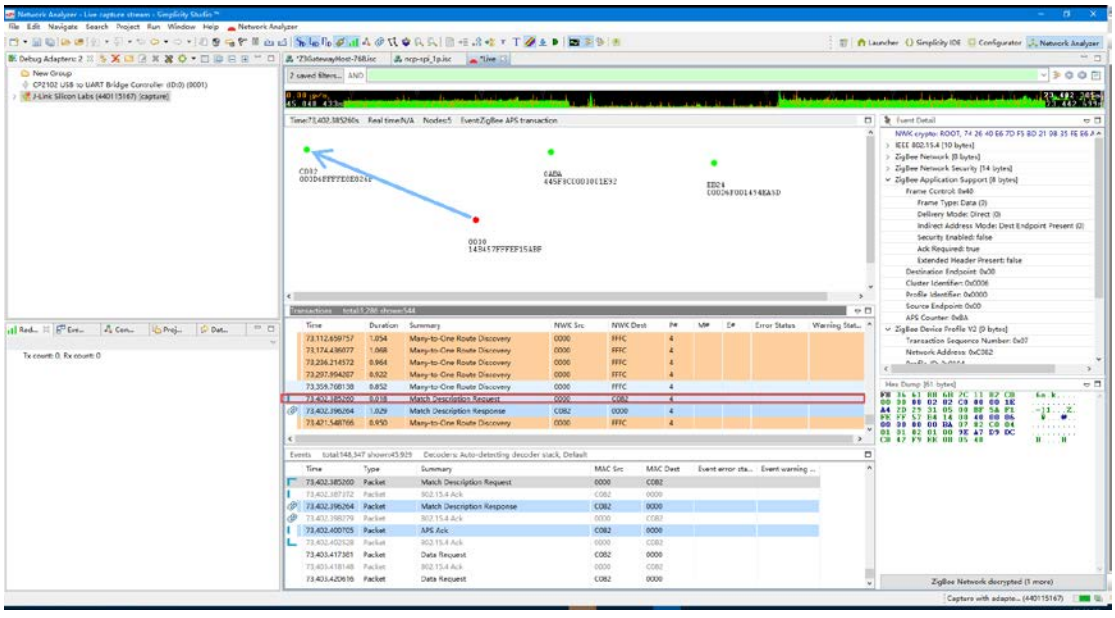
```
root@GL-S1300:~# zbtool zdo-request help
help                        Get zdo request list
match_descriptor            Send match descriptor request
node_descriptor             Send node descriptor request
power_descriptor            Send power descriptor request
active_endpoints            Send active endpoints request
simple_descriptor           Send simple descriptor request
lqi_table_request           Send lqi table request
routing_table_request       Send routing table request
bind_request                Send bind request
binding_table_request       Send binding table request
root@GL-S1300:~#
```

1.3.16.2 zdo-request match_descriptor

zbtool zdo-request match_descriptor [target:2] [profile:2] [inCount:1] [inClusters:2x] [outCount:2] [outClusters:2y]

- Send match descriptor request
  - target: Two byte target device short ID
  - profile: Two byte profile id for the match descriptor request
  - inCount: One byte num of in clusters
  - inClusters: Array of in clusters
  - outCount: One byte num of out clusters
  - outClusters: Array of out clusters

```
root@GL-S1300:~# zbtool zdo-request match_descriptor 49282 260 1 0102 0 0000
Send match descriptor request success
root@GL-S1300:~#
```

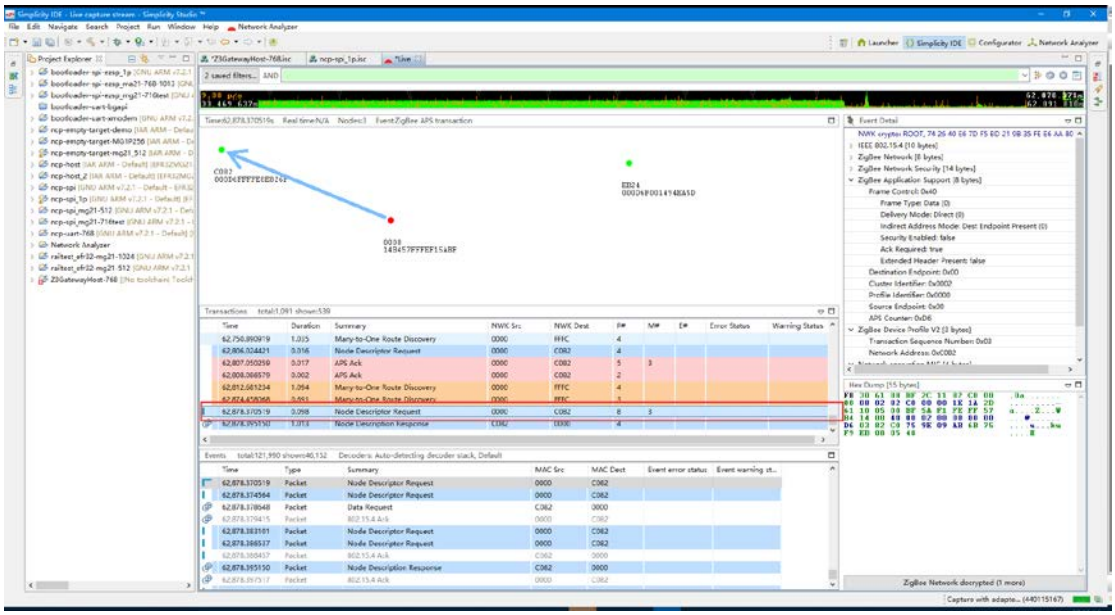


1.3.16.3 zdo-request node_descriptor

zbtool zdo-request node_descriptor [target:2]

- Send node descriptor request
  - target: Two byte target device short ID

```
root@GL-S1300:~# zbtool zdo-request node_descriptor 49282
Send node descriptor request success
root@GL-S1300:~#
```

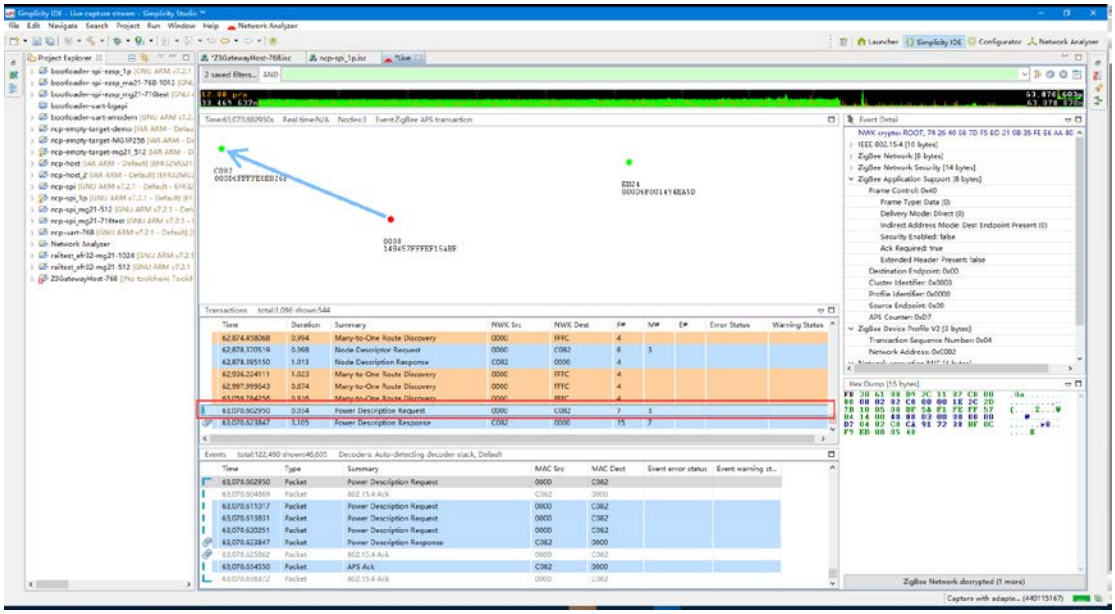


1.3.16.4 zdo-request power_descriptor

zbtool zdo-request power_descriptor [target:2]

- Send power descriptor request
  - target: Two byte target device short ID

```
root@GL-S1300:~# zbtool zdo-request power_descriptor 49282
Send power descriptor request success
root@GL-S1300:~#
```

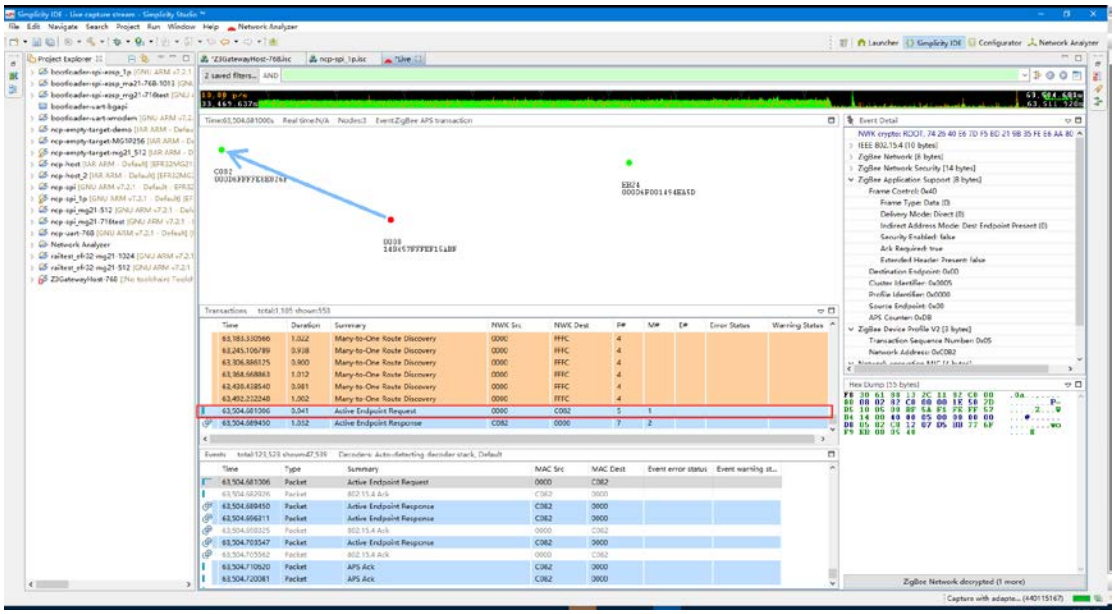


1.3.16.5 zdo-request active_endpoints

`zbtool zdo-request active_endpoints [target:2]`

- Send active_endpoints request
  - target: Two byte target device short ID

```
root@GL-S1300:~# zbtool zdo-request active_endpoints 49282
Send active endpoints request success
root@GL-S1300:~#
```

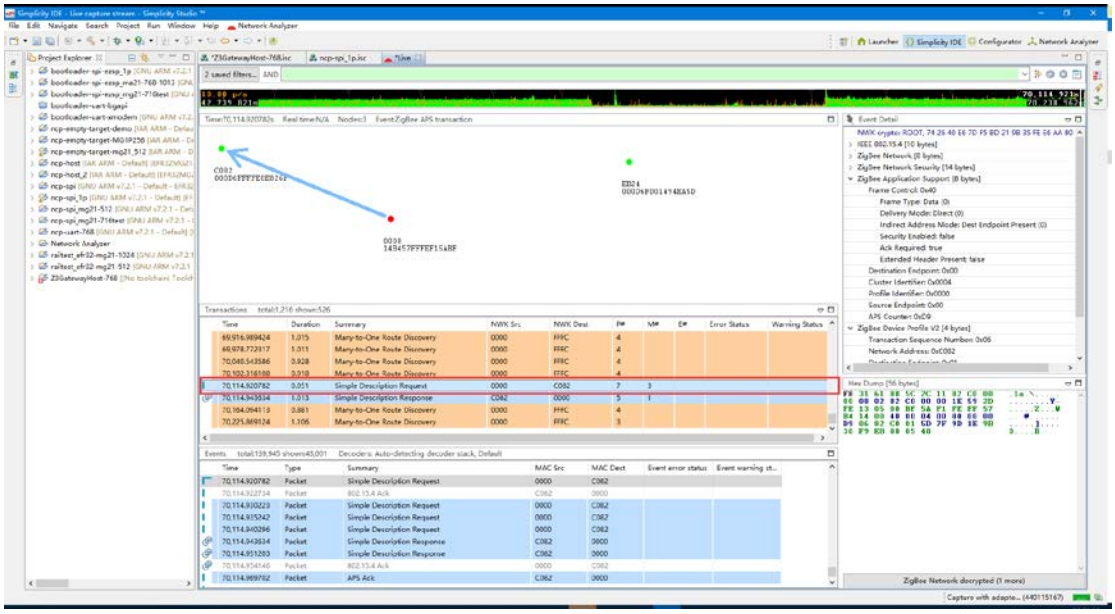


1.3.16.6 zdo-request simple_descriptor

`zbtool zdo-request simple_descriptor [target:2] [endpoint:1]`

- Send simple_descriptor request
  - target: Two byte target device short ID
  - endpoint: One byte the endpoint on the target device where the simple descriptor request will be sent

```
root@GL-S1300:~# zbtool zdo-request simple_descriptor 49282 1
Send simple descriptor request success
root@GL-S1300:~#
```

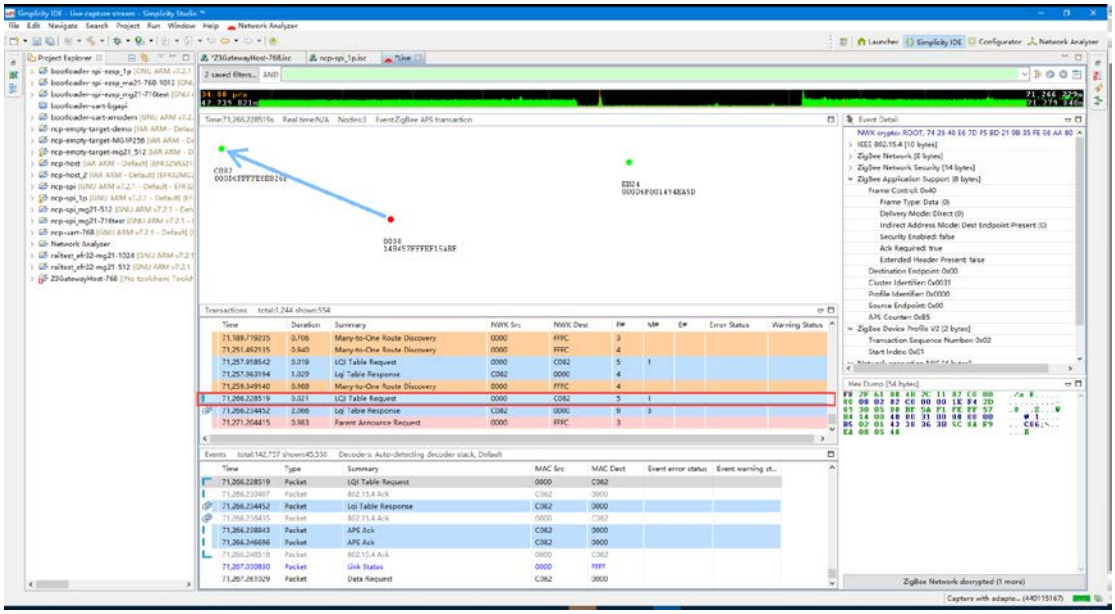


1.3.16.7 zdo-request lqi_table_request

zbtool zdo-request lqi_table_request [target:2] [startIndex:1]

- Send lqi_table_request request
  - target: Two byte target device short ID
  - startIndex: One byte starting index into table query

```
root@GL-S1300:~# zbtool zdo-request lqi_table_request 49282 1
Send lqi table request success
root@GL-S1300:~#
```

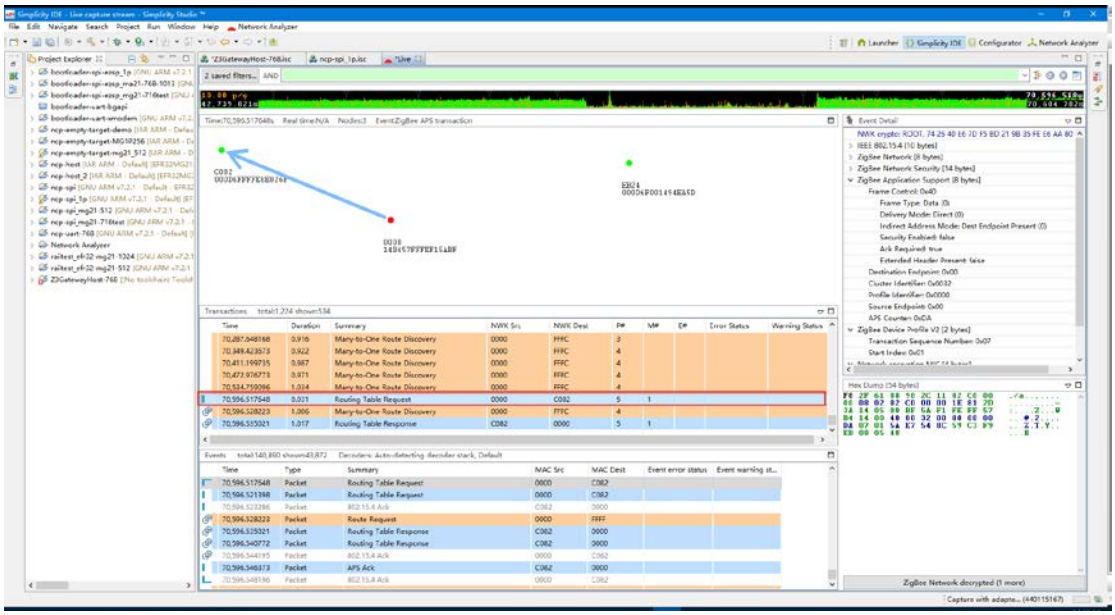


1.3.16.8 zdo-request routing_table_request

zbtool zdo-request routing_table_request [target:2] [startIndex:1]

- Send routing_table_request request
  - target: Two byte target device short ID
  - startIndex: One byte starting index into table query

```
root@GL-S1300:~# zbtool zdo-request routing_table_reques 49282 1
Send routing table request success
root@GL-S1300:~#
```

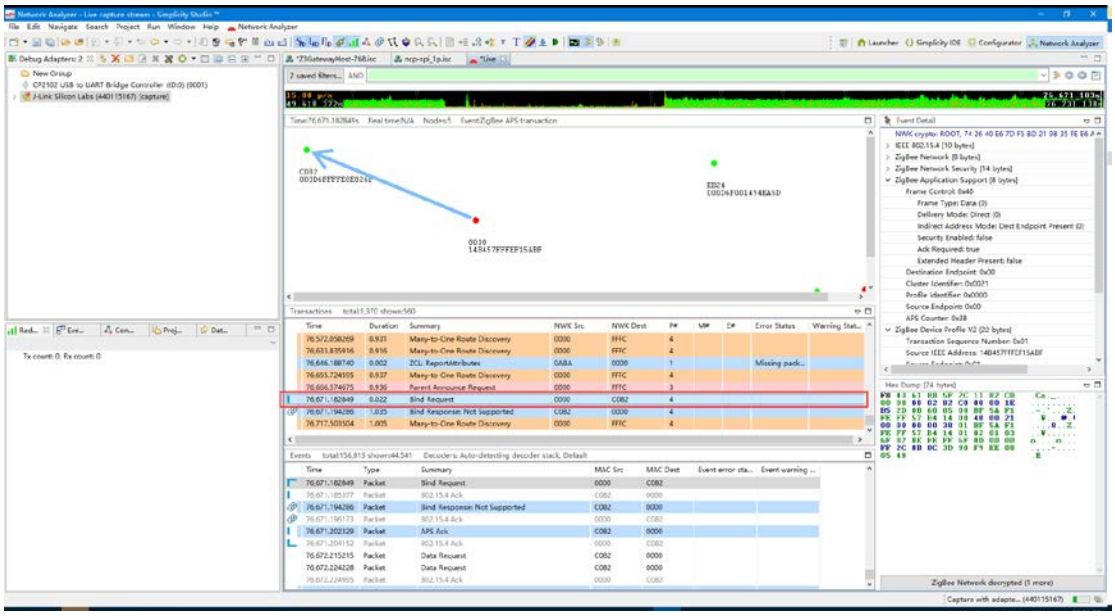


1.3.16.9 zdo-request bind_request

zbtool zdo-request bind_request [target:2] [bindClusterID:2] [sourceEUI64:8] [sourceEp:1] [clusterID:2] [type:1] [destEUI64:8] [groupAddr:2] [destEp:1]

- Send simple_descriptor request
  - target: Two byte target device short ID
  - bindClusterID: Two byte bind cluster ID
    - #define BIND_REQUEST 0x0021
    - #define UNBIND_REQUEST 0x0022
  - sourceEUI64: -IEEE_ADDRESS- The source EUI64 of the binding
  - sourceEp: One byte the source endpoint of the binding
  - clusterID: Two byte the cluster ID to bind
  - type: One byte the type of bind request
    - #define UNICAST_BINDING 0x03
    - #define UNICAST_MANY_TO_ONE_BINDING 0x83
    - #define MULTICAST_BINDING 0x01
  - destEUI64: -IEEE_ADDRESS- The destination EUI64 of the binding
  - groupAddr: Two byte the group address in the binding if use group bind
  - destEp: One byte the destination endpoint of the binding

```
root@GL-S1300:~# zbtool zdo-request bind_request 49282 33 bf5af1feff57b414 1 258 3 6f828efeff6f0d00 0 0
Send bind request success
```



1.3.16.10 zdo-request binding_table_request

zbtool zdo-request binding_table_request [target:2] [startIndex:1]

- Send binding_table_request request
  - target: Two byte target device short ID
  - startIndex: One byte starting index into table query

```
root@GL-S1300:~# zbtool zdo-request binding_table_request 49282 0
Send binding table request success
root@GL-S1300:~#
```