



ZHC-GW8000 JSON Application Guidance

LoRa Concentrator

Version: ZHC-GW8000_JSON_Application Guidance_V1.0

Date: 2021-4-13

Content

1. Node management	4
1.1. Get the total number of nodes	4
1.2. Get the specified node package	5
1.3. Get the specified ID node information	8
1.4. Add nodes	11
1.5. Delete the specified node	13
1.6. Delete all nodes	14
1.7. Modify node parameters	15
1.8. Get the last 3 data of the node	17
2. Concentrator management	20
2.1. Query the networking information of the concentrator	20
2.2. Set concentrator networking information	21
2.3. Query LTE network information	23
2.4. Query LTE APN information	24
2.5. Set LTE APN information	25
2.6. Query SOCKET information	26
2.7. Set SOCKET information	28

2.8. Query Concentrator Basic Information	31
2.9. Set the basic information of the concentrator	32
2.10. Query concentrator timing action	33
2.11. Set the timing action of the concentrator	34
2.12. Query direct module communication	35
2.13. Setting up direct module communication	36
2.14. Query Concentrator Location Information	37
2.15. Set concentrator positioning information	38
2.16. Set Concentrator System Commands	39
3. Business data	41
3.1. Node access to the network	41
3.2. Node decision offline	41
3.3. Collected copy	42
3.4. Supplementary copy	43
3.5. Nodes report actively	44
3.6. Click to copy	46

1. Node management

1.1. Get the total number of nodes

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getNodeCnt
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqData frame format

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getNodeCntAck
gwID	Yes	character	Concentrator unique ID, 12 characters
ackData	Yes	character	ackData frame format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
count	Yes	character	Total number of nodes managed by the concentrator

Request frame example:

```
{
  "msgType": "getNodeCnt",
  "gwID": "GW80210401020260",
  "reqData": {}
}
```

Example response frame (empty table):

```
{
  "msgType": "getNodeCntAck",
  "gwID": "GW80210401020260",
  "ackData": {
    "status": "0",
    "count": "0"
  }
}
```

```
}

```

Example of response frame:

```
{
  "msgType": "getNodeCntAck",
  "gwID": "GW80210401020260",
  "ackData": {
    "status": "0",
    "count": "2"
  }
}
```

1.2. Get the specified node package

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getNodeIndex
gwID	Yes	character	Concentrator unique ID, 16bit character
reqData	Yes	character	reqData frame format

reqData frame format

field	Is it necessary	character	describe
index	Yes	character	nodeindex

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getNodeIndexAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:Correct reply 120: No node 121: Index does not exist
nodeID	Yes	character	Node unique identificationID, 12 characters
logicID	Yes	character	The order in which the nodes respond to the collection
version	Yes	character	Firmware version
online	Yes	character	Information in offline (0: offline 1: online)

rssl	Yes	character	Signal value
rsslAvg	Yes	character	Average signal value
snr	Yes	character	signal to noise ratio
timestamp	Yes	character	Last interaction timestamp
br	Yes	Integer	baud rate 0:600; 1:1200; 2:2400;3:4800; 4:9600; 5:14400;6:19200; 7:38400;8:46000; 9:57600;10:76800;11:115200;12:128000; 13:230400;14:256000; 15:460800;16:921600; 17:1382400
ws	Yes	Integer	data bits 0:8; 1:9
sb	Yes	Integer	stop bit 0:0.5; 1:1; 2:1.5; 3:2
pa	Yes	Integer	Check Digit 0: no verification; 1:Odd parity; 2: even parity
model	Yes	Integer	Device model 0:ZHCR;1:ZHC0921A;2:ZHC0921D 3:ZHC0921AD;4:ZHC0931O
cfgUpdate	Yes	Integer	0: Configuration has taken effect 1: Configuration has been modified but not taken effect
uartCfg	no	character	Serial port collection and configuration root node, model==0 is valid
cmd	no	character	Serial acquisition command, maximum 16 characters For example: hexadecimal 0x01 0x03 0x00 0x00 0x00 0x02 0xc4 0x0b Should be converted to: ASCII 010300000002c40b
std	no	character	Intercept the start bit of the reply to the acquisition command For example: std==1 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd then intercept the reply from 0x03start
len	no	character	Intercept the length of the reply to the acquisition command For example: std==1 len==6 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd The intercepted reply is 0x03 0x04 0x00 0xc5 0x02 0xba
aiCfg	no	character	The analog triggers the report to configure the root node, model==1 is valid
mode	no	Integer	0: Disable reporting 1:Entry interval report 2: Exit interval report 3: The above two
min	no	character	the lower limit of the interval
max	no	character	upper limit of the interval
diCfg	no	character	The dry/wet node triggers the report to configure the root node, model==2 is valid
mode	no	character	0:Disable Report 1: Close Report 2 Detected:Disconnect

			detected report 3:Both of the above
--	--	--	-------------------------------------

Request frame example:

```
{
  "msgType": "getNodeIndex",
  "gwID": "GW80210401020260",
  "reqData": {
    "index": "0"
  }
}
```

Response frame example (linked list is empty):

```
{
  "msgType": "getNodeIndexAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "120"
  }
}
```

Response frame example(index out of bounds):

```
{
  "msgType": "getNodeIndexAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "121"
  }
}
```

Example of response frame:

```
{
  "msgType": "getNodeIndexAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "nodeID": "210412020661",
    "logicID": "1",
    "version": "1004",
    "online": 1,
    "rssi": "-13",
    "rssiAvg": "-12",
    "snr": "11",
    "timestamp": "1618541644",
    "br": 4,
  }
}
```

```
"ws": 0,  
"sb": 1,  
"pa": 0,  
"model": 2,  
"cfgUpdate": 1,  
"diCfg": {  
    "mode": 0  
}  
}  
}
```

1.3. Get the specified ID node information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getNodeInfo
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqDataframe format

reqData frame format:

field	Is it necessary	type	describe
nodeID	Yes	character	Node unique ID, 12 characters

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getNodeInfoAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:Correct reply 120: No node 121: Index does not exist
nodeID	Yes	character	Node unique identificationID, 12 characters
logicID	Yes	character	The order in which the nodes respond to the collection
version	Yes	character	Firmware version
online	Yes	character	Information in offline (0: offline 1: online)

rssi	Yes	character	Signal value
rssiAvg	Yes	character	Average signal value
snr	Yes	character	signal to noise ratio
timestamp	Yes	character	Last interaction timestamp
br	Yes	Integer	baud rate 0:600; 1:1200; 2:2400;3:4800; 4:9600; 5:14400;6:19200; 7:38400;8:46000; 9:57600;10:76800;11:115200;12:128000; 13:230400;14:256000; 15:460800;16:921600; 17:1382400
ws	Yes	Integer	data bits 0:8; 1:9
sb	Yes	Integer	stop bit 0:0.5; 1:1; 2:1.5; 3:2
pa	Yes	Integer	Check Digit 0: no verification; 1:Odd parity; 2: even parity
model	Yes	Integer	Device model 0:ZHCR;1:ZHC0921A;2:ZHC0921D 3:ZHC0921AD;4:ZHC0931O
cfgUpdate	Yes	Integer	0: Configuration has taken effect 1: Configuration has been modified but not taken effect
uartCfg	no	character	Serial port collection and configuration root node, model==0 is valid
cmd	no	character	Serial acquisition command, maximum 16 characters For example: hexadecimal 0x01 0x03 0x00 0x00 0x00 0x02 0xc4 0x0b Should be converted to: ASCII 010300000002c40b
std	no	character	Intercept the start bit of the reply to the acquisition command For example: std==1 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd then intercept the reply from 0x03start
len	no	character	Intercept the length of the reply to the acquisition command For example: std==1 len==6 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd The intercepted reply is 0x03 0x04 0x00 0xc5 0x02 0xba
aiCfg	no	character	The analog triggers the report to configure the root node, model==1 is valid
mode	no	Integer	0: Disable reporting 1:Entry interval report 2: Exit interval report 3: The above two
min	no	character	the lower limit of the interval
max	no	character	upper limit of the interval
diCfg	no	character	The dry/wet node triggers the report to configure the root node, model==2 is valid
mode	no	character	0:Disable Report 1: Close Report 2 Detected:Disconnect

			detected report 3:Both of the above
--	--	--	-------------------------------------

Request frame example:

```
{
  "msgType": "getNodeInfo",
  "gwID": "GW80210401020272",
  "reqData": {
    "nodeID": "210412020661"
  }
}
```

Response frame example(node does not exist):

```
{
  "msgType": "getNodeInfoAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "130"
  }
}
```

Example of response frame:

```
{
  "msgType": "getNodeInfoAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "nodeID": "210412020661",
    "logicID": "1",
    "version": "1004",
    "online": "1",
    "rssi": "-37",
    "rssiAvg": "-36",
    "snr": "13",
    "timestamp": "1618542685",
    "br": "4",
    "ws": "0",
    "sb": "1",
    "pa": "0",
    "model": "2",
    "cfgUpdate": 1,
    "diCfg": {
      "mode": "0"
    }
  }
}
```

}

1.4. Add nodes

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	Node unique identificationID, 12 characters
gwID	Yes	character	The order in which the nodes respond to the collection
reqData	Yes	character	reqData frame format

reqData frame format

field	Is it necessary	type	describe
nodeID	Yes	character	Node unique identificationID, 12 characters
logicID	Yes	character	The order in which the nodes respond to the collection
br	no	Integer	baud rate 0:600; 1:1200; 2:2400;3:4800; 4:9600; 5:14400;6:19200; 7:38400;8:46000; 9:57600;10:76800;11:115200;12:128000; 13:230400;14:256000; 15:460800;16:921600; 17:1382400
ws	no	Integer	data bits 0:8; 1:9
sb	no	Integer	stop bit 0:0.5; 1:1; 2:1.5; 3:2
pa	no	Integer	Check Digit 0: no verification; 1:Odd parity; 2: even parity
model	Yes	Integer	Device model 0:ZHCR;1:ZHC0921A;2:ZHC0921D 3:ZHC0921AD;4:ZHC0931O
uartCfg	no	character	Serial port collection and configuration root node, model==0 is valid
cmd	no	character	Serial acquisition command, maximum 16 characters For example: hexadecimal 0x01 0x03 0x00 0x00 0x00 0x02 0xc4 0x0b Should be converted to: ASCII 010300000002c40b
std	no	character	Intercept the start bit of the reply to the acquisition command For example: std==1 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd then intercept the reply from 0x03start
len	no	character	Intercept the length of the reply to the acquisition command

			For example: std==1 len==6 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd The intercepted reply is 0x03 0x04 0x00 0xc5 0x02 0xba
aiCfg	no	character	The analog triggers the report to configure the root node, model==1 is valid
mode	no	Integer	0: Disable reporting 1:Entry interval report 2: Exit interval report 3: The above two
min	no	character	the lower limit of the interval
max	no	character	upper limit of the interval
diCfg	no	character	The dry/wet node triggers the report to configure the root node, model==2 is valid
mode	no	character	0:Disable Report 1: Close Report 2 Detected:Disconnect detected report 3:Both of the above

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	addNodeAck
gwID	Yes	character	Concentrator unique ID, 16bit character
ackData	Yes	character	ackDataframe format

ackDataframe format:

field	Is it necessary	type	describe
status	Yes	character	0:Correct reply 140: Node already exists 141: Node list full

Request frame example:

```
{
  "msgType": "addNode",
  "gwID": "GW80210401020272",
  "reqData": {
    "nodeID": "210315018659",
    "logicID": "",
    "br": 4,
    "ws": 0,
    "sb": 1,
    "pa": 0,
    "model": 0,
    "uartCfg": {
      "cmd": "",
      "std": "",
      "len": ""
    }
  }
}
```

```

    },
    "aiCfg": {
        "mode": 0,
        "min": "",
        "max": ""
    },
    "diCfg": {
        "mode": 0
    }
}
}
}

```

Example of response frame:

```

{
    "msgType": "addNodeAck",
    "gwID": "GW80210401020272",
    "ackData": {
        "status": "0"
    }
}
}

```

1.5. Delete the specified node

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	delNode
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqDataframe format

reqData frame format:

field	Is it necessary	type	describe
nodeID	Yes	character	Node unique ID, 12 characters

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	delNodeAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply 150: Node does not exist

Request frame example:

```
{
  "msgType": "delNode",
  "gwID": "GW80210401020272",
  "reqData": {
    "nodeID": "210315018659"
  }
}
```

Example of response frame:

```
{
  "msgType": "delNodeAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

Example of response frame (node does not exist):

```
{
  "msgType": "delNodeAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "150"
  }
}
```

1.6. Delete all nodes

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	deleteAllNode
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	deleteAllNode
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackdata	Yes	character	0:correct reply

Request frame example:

```
{
  "msgType": "deleteAllNode",
  "gwID": "GW80210401020272",
  "reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "deleteAllNodeAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

1.7. Modify node parameters

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	editNode
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqDataframe format

reqData frame format:

field	Is it necessary	type	describe
nodeID	Yes	character	Node unique ID, 12 characters
logicID	no	character	The order in which the nodes respond to the collection
br	no	Integer	baud rate 0:600; 1:1200; 2:2400;3:4800; 4:9600; 5:14400;6:19200; 7:38400;8:46000; 9:57600;10:76800;11:115200;12:128000; 13:230400;14:256000; 15:460800;16:921600; 17:1382400
ws	no	Integer	data bits

			0:8; 1:9
sb	no	Integer	stop bit 0:0.5; 1:1; 2:1.5; 3:2
pa	no	Integer	Check Digit 0: no verification; 1:Odd parity; 2: even parity
model	no	Integer	Device model 0:ZHCR;1:ZHC0921A;2:ZHC0921D 3:ZHC0921AD;4:ZHC0931O
uartCfg	no	character	Serial port collection and configuration root node, model==0 is valid
cmd	no	character	Serial acquisition command, maximum 16 characters For example: hexadecimal 0x01 0x03 0x00 0x00 0x00 0x02 0xc4 0x0b Should be converted to: ASCII 010300000002c40b
std	no	character	Intercept the start bit of the reply to the acquisition command For example: std==1 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd then intercept the reply from 0x03start
len	no	character	Intercept the length of the reply to the acquisition command For example: std==1 len==6 data reply 0x01 0x03 0x04 0x00 0xc5 0x02 0xba 0x6a 0xdd The intercepted reply is 0x03 0x04 0x00 0xc5 0x02 0xba
aiCfg	no	character	The analog triggers the report to configure the root node, model==1 is valid
mode	no	Integer	0: Disable reporting 1:Entry interval report 2: Exit interval report 3: The above two
min	no	character	the lower limit of the interval
max	no	character	upper limit of the interval
diCfg	no	character	The dry/wet node triggers the report to configure the root node, model==2 is valid
mode	no	character	0:Disable Report 1: Close Report 2 Detected:Disconnect detected report 3:Both of the above

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	editNodeAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

reqData frame format:

field	Is it necessary	type	describe
-------	-----------------	------	----------

status	Yes	character	0:Correct reply 170: Node does not exist
cfgUpdate	Yes	character	0: Reboot required and new configuration applied 1:No reboot required

Request frame example:

```
{
  "msgType": "editNode",
  "gwID": "GW80210401020272",
  "reqData": {
    "nodeID": "210412030661",
    "logicID": "",
    "br": 4,
    "ws": 0,
    "sb": 1,
    "pa": 0,
    "model": 0,
    "uartCfg": {
      "cmd": "010300000002c40b",
      "std": "0",
      "len": "9"
    },
    "aiCfg": {
      "mode": 0,
      "min": "",
      "max": ""
    },
    "diCfg": {
      "mode": 0
    }
  }
}
```

Example of response frame:

```
{
  "msgType": "editNodeAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "cfgUpdate": 0
  }
}
```

1.8. Get the last 3 data of the node

Request frame format:

IIoT empowers the edge

field	Is it necessary	type	describe
msgType	Yes	character	getNodeData
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqDataframe format

reqData frame format:

field	Is it necessary	type	describe
nodeID	Yes	character	Node unique ID, 12 characters

Response frame format:

field	Is it necessary	type	describe
nodeID	Yes	character	getNodeDataAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0: Reply 360 correctly:node does not exist
nodeID	Yes	character	Node unique ID, 12 characters
logicID	Yes	character	Node data transmission time coordinates
version	Yes	character	Firmware version
model	Yes	character	Device model 0:ZHCR;1:ZHC0921A;2:ZHC0921D 3:ZHC0921AD;4:ZHC0931O
localData	Yes	character	local data root node
payload	no	character	Serial port collection value. model==0 is valid
ai	no	character	Analog acquisition value. model==1 model==3 is valid
di	no	character	Switch value acquisition value. model==2 model==3 is valid
rssI	Yes	character	Report the signal strength of this piece of data
rssIAvg	Yes	character	Average signal value
snr	Yes	character	Report the signal-to-noise ratio of this data
timestamp	Yes	character	Timestamp when this piece of data was reported

Request frame example:

```
{
  "msgType": "getNodeData",
  "gwID": "GW80210401020272",
  "reqData": {
```

```
    "nodeID": "210412020661"  
  }  
}
```

Example of response frame:

```
{  
  "msgType": "getNodeDataAck",  
  "gwID": "GW80210401020272",  
  "ackData": {  
    "status": "0",  
    "nodeID": "210412020661",  
    "logicID": "1",  
    "version": "1004",  
    "model": "1",  
    "localData": [{  
      "ai": "0",  
      "rssi": "-13",  
      "rssiAvg": "-12",  
      "snr": "11",  
      "timestamp": "1618632589"  
    }, {  
      "ai": "0",  
      "rssi": "-13",  
      "rssiAvg": "-12",  
      "snr": "10",  
      "timestamp": "1618632590"  
    }, {  
      "ai": "0",  
      "rssi": "-13",  
      "rssiAvg": "-12",  
      "snr": "10",  
      "timestamp": "1618632591"  
    }  
  ]  
}
```

Response frame example(node does not exist):

```
{  
  "msgType": "getNodeDataAck",  
  "gwID": "GW80210401020272",  
  "ackData": {  
    "status": "360"  
  }  
}
```

2. Concentrator management

2.1. Query the networking information of the concentrator

askframeFormat:

field	Is it necessary	type	describe
msgType	Yes	character	getGroupInfoAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getGroupInfoAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
channel	Yes	character	channel. 0~31, single channel 1MHz, CHA start channel 471.2MHz, CHB start channel 471.5MHz, CHC start channel 471.8MHz.
sf	Yes	character	Spreading factor. SF7~SF12, in theory, SF12 has the farthest communication distance and the lowest communication rate.
appID	Yes	character	App ID. Differentiate between different concentrators.
slot	Yes	character	Transmission time coordinate The time difference between two adjacent nodes sending data.
collect	Yes	character	Collection cycle. The time interval between two collections. $\geq(\text{Number of nodes}+2)*\text{slot}$
cmdMode	Yes	character	Instruction execution mode (0: immediate delivery 1: idle delivery) Immediate delivery: No matter whether the concentrator is idle or busy, as long as it receives data or instructions from the server, it will be sent immediately, which may cause data collision; Idle delivery: After the concentrator receives the data from the server, it mounts it to the instruction list, waits for the arrival of an idle time slot, and then takes it out from the list and sends it

netState	Yes	character	Network status: 0 idle 1 busy Returns busy if the concentrator is processing internal logic; otherwise returns idle
----------	-----	-----------	--

Request frame example:

```
{
  "msgType": "getGroupInfo",
  "gwID": "GW80210401020272",
  "reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "getGroupInfoAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "channel": "20",
    "sf": "9",
    "appID": "2",
    "slot": "3",
    "collect": "360",
    "cmdMode": "0",
    "netState": "0"
  }
}
```

2.2. Set concentrator networking information

askframeFormat:

field	Is it necessary	type	describe
msgType	Yes	character	setGroupInfo
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqData frame format

reqData frame format

field	Is it necessary	type	describe
channel	Yes	character	channel. 0~31, single channel 1MHz, CHA start channel 471.2MHz, CHB start channel 471.5MHz, CHC start channel 471.8MHz.
sf	Yes	character	Spreading factor. SF7~SF12, in theory, SF12 has the farthest communication distance and the lowest communication rate.
appID	Yes	character	App ID. Differentiate between different concentrators.

slot	Yes	character	Transmission time coordinate The time difference between two adjacent nodes sending data.
collect	Yes	character	Collection cycle. The time interval between two collections. $\geq(\text{Number of nodes}+2)*\text{slot}$
cmdMode	Yes	character	Instruction execution mode (0: immediate delivery 1: idle delivery) Immediate delivery: No matter whether the concentrator is idle or busy, as long as it receives data or instructions from the server, it will be sent immediately, which may cause data collision; Idle delivery: After the concentrator receives the data from the server, it mounts it to the instruction list, waits for the arrival of an idle time slot, and then takes it out from the list and sends it

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setGroupInfo
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```

{
  "msgType": "setGroupInfo",
  "gwID": "GW80210401020272",
  "reqData": {
    "channel": "20",
    "sf": "9",
    "appID": "2",
    "slot": "3",
    "collect": "360",
    "cmdMode": "0"
  }
}
    
```

Example of response frame:

```

{
  "msgType": "setGroupInfoAck",
    
```

```

"gwID": "GW80210401020272",
"ackData": {
  "status": "0"
}
}

```

2.3. Query LTE network information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getLTEBasic
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getLTEBasicAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe																																																						
status	Yes	character	0:correct reply																																																						
ccid	Yes	character	SIM card number																																																						
qcsq	Yes	character	signal strength. <table border="1" data-bbox="751 1442 1310 1615"> <thead> <tr> <th><sysmdoe></th> <th><value1></th> <th><value2></th> <th><value3></th> <th><value4></th> <th><value5></th> </tr> </thead> <tbody> <tr> <td>"NOSERVICE"</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>"GSM"</td> <td>gsm_rssi</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>"WCDMA"</td> <td>wcdma_rssi</td> <td>wcdma_rscp</td> <td>wcdma_ecio</td> <td></td> <td></td> </tr> <tr> <td>"TDSCDMA"</td> <td>tdscdma_rssi</td> <td>tdscdma_rscp</td> <td>tdscdma_ecio</td> <td></td> <td></td> </tr> <tr> <td>"LTE"</td> <td>lte_rssi</td> <td>lte_rsrp</td> <td>lte_sinr</td> <td>lte_rsrq</td> <td></td> </tr> <tr> <td>"CDMA"</td> <td>cdma_rssi</td> <td>cdma_ecio</td> <td></td> <td></td> <td></td> </tr> <tr> <td>"EVDO"</td> <td>evdo_rssi</td> <td>evdo_ecio</td> <td>evdo_sinr</td> <td></td> <td></td> </tr> <tr> <td>"CDMA-EVDO"</td> <td>cdma_rssi</td> <td>cdma_ecio</td> <td>evdo_ecio</td> <td>evdo_ecio</td> <td>evdo_sinr</td> </tr> </tbody> </table> 1. You can refer to China Mobile's acceptance criteria, mainly rsrp: Excellent: rsrp>-85dbm; Good: rsrp=-85~-95dbm; Middle: rsrp=-95~-105dbm; Poor: rsrp=-105dbm~-115dbm; Very poor: rsrp<115dbm;	<sysmdoe>	<value1>	<value2>	<value3>	<value4>	<value5>	"NOSERVICE"						"GSM"	gsm_rssi					"WCDMA"	wcdma_rssi	wcdma_rscp	wcdma_ecio			"TDSCDMA"	tdscdma_rssi	tdscdma_rscp	tdscdma_ecio			"LTE"	lte_rssi	lte_rsrp	lte_sinr	lte_rsrq		"CDMA"	cdma_rssi	cdma_ecio				"EVDO"	evdo_rssi	evdo_ecio	evdo_sinr			"CDMA-EVDO"	cdma_rssi	cdma_ecio	evdo_ecio	evdo_ecio	evdo_sinr
<sysmdoe>	<value1>	<value2>	<value3>	<value4>	<value5>																																																				
"NOSERVICE"																																																									
"GSM"	gsm_rssi																																																								
"WCDMA"	wcdma_rssi	wcdma_rscp	wcdma_ecio																																																						
"TDSCDMA"	tdscdma_rssi	tdscdma_rscp	tdscdma_ecio																																																						
"LTE"	lte_rssi	lte_rsrp	lte_sinr	lte_rsrq																																																					
"CDMA"	cdma_rssi	cdma_ecio																																																							
"EVDO"	evdo_rssi	evdo_ecio	evdo_sinr																																																						
"CDMA-EVDO"	cdma_rssi	cdma_ecio	evdo_ecio	evdo_ecio	evdo_sinr																																																				

Request frame example:

```

{
  "msgType": "getLTEBasic",
  "gwID": "GW80210401020272",

```

```
"reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "getLTEBasicAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "ccid": "89860474282090004427",
    "qcsq": "\"LTE\",84,-105,165,-5"
  }
}
```

2.4. Query LTE APN information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getLTEAPN
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getLTEAPNAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackDataframe format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
addr	Yes	character	APN address
user	Yes	character	APN username
pass	Yes	character	APN password

Request frame example:

```
{
  "msgType": "getLTEAPN",
```



```

    "gwID": "GW80210401020272",
    "reqData": {}
}
    
```

Example of response frame:

```

{
  "msgType": "getLTEAPNAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "addr": "",
    "user": "",
    "pass": ""
  }
}
    
```

2.5. Set LTE APN information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setLTEAPN
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqDataframe format

reqDataFrame format:

field	Is it necessary	type	describe
addr	no	character	APN address
user	no	character	APN username
pass	no	character	APN password

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setLTEAPNAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackDataFrame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```
{
  "msgType": "setLTEAPN",
  "gwID": "GW80210401020272",
  "reqData": {
    "addr": "192.168.0.60",
    "user": "admin",
    "pass": "123456"
  }
}
```

Example of response frame:

```
{
  "msgType": "setLTEAPNAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

2.6. Query SOCKET information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getSOCKET1Info getSOCKET2Info getSOCKET3Info getSOCKET4Info
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getSOCKET1InfoAck getSOCKET2InfoAck getSOCKET3InfoAck getSOCKET4InfoAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackDataFrame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
enable	Yes	character	Whether to enable this connection (0:disable 1:enable)
mode	Yes	Integer	Working mode (1: TCP Client 4: MQTT)
desIp	Yes	character	Destination server (IP/domain name)
desPort	Yes	character	destination port
regMode	Yes	Integer	Registration package mode (<ul style="list-style-type: none"> 0: Disable 1:default 2:customize 3:device ID 4: SIM card number From the perspective of the server, it is used to distinguish different devices
regPos	Yes	Integer	Trigger mode (0: send when connected 1: data carry 2: the above two)
regPkg	Yes	character	Registration Package Contents. Hex data ASCII format. E.g: hello world=> 68656C6C6F20776F726C64
hbtMode	Yes	Integer	Heartbeat Packet Mode Implement device liveness detection on business processes 0: enable 1: disable
hbtCyc	Yes	character	Heartbeat packet cycle
hbtPkg	Yes	character	Heartbeat package content. Hex data ASCII format. E.g: hello world=> 68656C6C6F20776F726C64
clientID	Yes	character	MQTT client ID Maximum 60 characters Only valid for Socket 1
username	Yes	character	MQTT username Maximum 60 characters only valid for Socket 1
password	Yes	character	MQTT password Maximum 60 characters only valid for Socket 1
subTopic	Yes	character	MQTT subscription topic Up to 100 characters Only valid for Socket 1
pubTopic	Yes	character	MQTT publish topic Maximum 100 characters only valid for Socket 1
keepAlive	Yes	character	MQTT keep alive time 10-65535s Only valid for Socket 1

Request frame example:

```
{
  "msgType": "getSOCKET1Info",
```

```
"gwID": "GW80210401020272",
"reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "getSOCKET1InfoAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "enable": 1,
    "mode": 1,
    "desIp": "cloud.iotrouter.cn",
    "desPort": "56000",
    "regMode": 1,
    "regPos": 1,
    "regPkg":
    "415F315F312E475738303231303430313032303237322E4D54497A4E4455322E3737333336661
    63356165373764356563353138656462656437316565373764382E41424344453132333435",
    "hbtMode": 1,
    "hbtCyc": "30",
    "hbtPkg": "50494E472150494E472150494E4721",
    "clientID": "",
    "username": "",
    "password": "",
    "subTopic": "/public/zhcpub",
    "pubTopic": "/public/zhcsb",
    "keepAlive": "30"
  }
}
```

2.7. Set SOCKET information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setSOCKET1InfoAck setSOCKET2InfoAck setSOCKET3InfoAck setSOCKET4InfoAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqDataframe format

reqData frame format

field	Is it necessary	type	describe
enable	Yes	character	Whether to enable this connection (0:disable 1:enable)
mode	Yes	Integer	Working mode (1: TCP Client 4: MQTT)
desIp	Yes	character	Destination server (IP/domain name)
desPort	Yes	character	destination port
regMode	Yes	Integer	Registration package mode (0: default 1: Customize 2: Device number 3: SIM card number 4: Disable) From the perspective of the server, it is used to distinguish different devices
regPos	Yes	Integer	Trigger mode (0: send when connected 1: data carry 2: the above two)
regPkg	Yes	character	Registration Package Contents. Hex data ASCII format. E.g: hello world=> 68656C6C6F20776F726C64
hbtMode	Yes	Integer	Heartbeat Packet Mode Implement device liveness detection on business processes
hbtCyc	Yes	character	Heartbeat packet cycle
hbtPkg	Yes	character	Heartbeat package content. Hex data ASCII format. E.g: hello world=> 68656C6C6F20776F726C64
clientID	Yes	character	MQTT client ID Maximum 60 characters only valid for Socket 1
username	Yes	character	MQTT username Maximum 60 characters only valid for Socket 1
password	Yes	character	MQTT password Maximum 60 characters only valid for Socket 1
subTopic	Yes	character	MQTT subscription topic Up to 100 characters Only valid for Socket 1
pubTopic	Yes	character	MQTT publish topic Maximum 100 characters only valid for Socket 1
keepAlive	Yes	character	MQTT keep alive time 10-65535s Only valid for Socket 1

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setSOCKET1Info setSOCKET2Info setSOCKET3Info setSOCKET4Info
gwID	Yes	character	Concentrator unique identification ID, 16 characters

ackData	Yes	character	ackData frame format
---------	-----	-----------	----------------------

ackData frame:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```
{
  "msgType": "setSOCKET1Info",
  "gwID": "GW80210401020272",
  "reqData": {
    "enable": 1,
    "mode": 1,
    "desIp": "cloud.iotrouter.cn",
    "desPort": "56000",
    "regMode": 1,
    "regPos": 1,
    "regPkg":
"415F315F312E475738303231303430313032303237322E4D54497A4E4455322E3737333336661
63356165373764356563353138656462656437316565373764382E41424344453132333435",
    "hbtMode": 1,
    "hbtCyc": "30",
    "hbtPkg": "50494E472150494E472150494E4721",
    "clientID": "",
    "username": "",
    "password": "",
    "subTopic": "/public/zhcpub",
    "pubTopic": "/public/zhcsb",
    "keepAlive": "30"
  }
}
```

Example of response frame:

```
{
  "msgType": "setSOCKET1InfoAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

2.8. Query Concentrator Basic Information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getDEVBasic
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getDEVBasicAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0: correct reply
devID	Yes	character	Concentrator unique identification ID, 16 characters
devPW	Yes	character	Concentrator password (valid for logging in to Zonghengyun)
cpuID	Yes	character	reserved
rptDec	Yes	Integer	The direction of actively reporting data. 0: send 1 to LTE:Send 2 to MAIN:Both of the above
version	Yes	character	Concentrator firmware version

Request frame example:

```
{
  "msgType": "getDEVBasic",
  "gwID": "AAAAAAAAAAAAAA",
  "reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "getDEVBasicAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "devID": "GW80210401020272",
    "devPW": "123456",
    "cpuID": "HgA7ABFHOTQwNjcz",
  }
}
```

```

        "rptDec": 2,
        "version": "2002"
    }
}
    
```

2.9. Set the basic information of the concentrator

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setDEVBasic
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqData frame format

reqData frame format

field	Is it necessary	type	describe
devPW	Yes	character	Concentrator password (valid for logging in to Zonghengyun)
rptDec	Yes	Integer	The direction of actively reporting data. 0: send 1 to LTE:Send 2 to MAIN:Both of the above

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	Concentrator password (valid for logging in to Zonghengyun)
gwID	Yes	character	The direction of actively reporting data. 0: send 1 to LTE:Send 2 to MAIN:Both of the above
ackdata	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```

{
  "msgType": "setDEVBasic",
  "gwID": "GW80210401020272",
  "reqData": {
    "devPW": "123456",
    "rptDec": "0"
  }
}
    
```



```
}

```

Example of response frame:

```
{
  "msgType": "setDEVBasicAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

2.10. Query concentrator timing action

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getTimeTrigger
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getTimeTriggerAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0: correct reply
action	Yes	Integer	The action to be performed when the set time is reached. 0: Disable 1:reboot
hour	Yes	character	Set time: Hour
minute	Yes	character	Set time: minutes
second	Yes	character	Set time: seconds

Request frame example:

```
{
  "msgType": "getTimeTrigger",
  "gwID": "GW80210401020272",
```

```
"reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "getTimeTriggerAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "action": 1,
    "hour": "0",
    "minute": "0",
    "second": "0"
  }
}
```

2.11. Set the timing action of the concentrator

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setTimeTrigger
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqData frame format

reqData frame format:

field	Is it necessary	type	describe
action	Yes	Integer	The action to be performed when the set time is reached. 0: Disable 1:reboot
hour	Yes	character	Set time: Hour
minute	Yes	character	Set time: minutes
second	Yes	character	Set time: seconds

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setTimeTriggerAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackdata	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```
{
  "msgType": "setTimeTrigger",
  "gwID": "GW80210401020272",
  "reqData": {
    "action": 1,
    "hour": "12",
    "minute": "0",
    "second": "0"
  }
}
```

Example of response frame:

```
{
  "msgType": "setTimeTriggerAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

2.12. Query direct module communication

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getLTEInter
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getLTEInterAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0: correct reply
superCmd	Yes	Integer	Enables direct communication with the LTE module through MAIN. 0: Disable 1:enable

Request frame example:

```
{
  "msgType": "getLTEInter",
  "gwID": "GW80210401020272",
  "reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "getLTEInterAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "superCmd": 0
  }
}
```

2.13. Setting up direct module communication

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setLTEInter
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqData frame format

reqData frame format:

field	Is it necessary	type	describe
superCmd	Yes	Integer	enable throughMAIN establishes communication with the LTE module directly. 0: Disable 1: Enable

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setLTEInterAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackdata	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```
{
  "msgType": "setLTEInter",
  "gwID": "GW80210401020272",
  "reqData": {
    "superCmd": 1
  }
}
```

Example of response frame:

```
{
  "msgType": "setLTEInterAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

2.14. Query Concentrator Location Information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	getGNSSInfo
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
-------	-----------------	------	----------

msgType	Yes	character	getGNSSInfoAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0: correct reply
cyc	Yes	character	Active reporting period of positioning information. reserved
location	Yes	character	location information. reserved.

Request frame example:

```
{
  "msgType": "getGNSSInfo",
  "gwID": "GW80210401020272",
  "reqData": {}
}
```

Example of response frame:

```
{
  "msgType": "getGNSSInfoAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "cyc": "0",
    "location": ""
  }
}
```

2.15. Set concentrator positioning information

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setGNSSInfo
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqData frame format

reqData frame format:

field	Is it necessary	type	describe
cyc	Yes	character	Active reporting period of positioning information. reserved

location	Yes	character	location information. reserved.
----------	-----	-----------	---------------------------------

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setGNSSInfoAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackdata	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```
{
  "msgType": "setGNSSInfo",
  "gwID": "GW80210401020272",
  "reqData": {
    "cyc": "0",
    "location": "hello world"
  }
}
```

Example of response frame:

```
{
  "msgType": "setGNSSInfoAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```

2.16. Set Concentrator System Commands

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setSysCmd
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqData frame format

reqData frame format:

field	Is it necessary	type	describe
cmd	Yes	character	1: reboot 2: reset 3: local upgrade 4:Remote upgrade

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	setSysCmdAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackdata	Yes	character	ackData frame format

ackData frame format

field	Is it necessary	type	describe
status	Yes	character	0:correct reply

Request frame example:

```
{
  "msgType": "setSysCmd",
  "gwID": "GW80210401020272",
  "reqData": {
    "cmd": "1"
  }
}
```

Example of response frame:

```
{
  "msgType": "setSysCmdAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0"
  }
}
```


3. Business data

3.1. Node access to the network

Reporting frame format:

field	Is it necessary	type	describe
msgType	Yes	character	nodeLoginAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackData frame format:

field	Is it necessary	type	describe
nodeID	Yes	character	Node factory serial number
version	Yes	character	Firmware version
rssI	Yes	character	Signal value
rssIAvg	Yes	character	Average signal value
snr	Yes	character	signal to noise ratio
timestamp	Yes	character	timestamp

Example of response frame:

```
{
  "msgType": "nodeLoginAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "nodeID": "210412020661",
    "version": "1004",
    "rssI": "-19",
    "rssIAvg": "-18",
    "snr": "11",
    "timestamp": "1618641399"
  }
}
```

3.2. Node decision offline

Reporting frame format:

field	Is it necessary	type	describe
msgType	Yes	character	nodeLogoutAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackData frame format:

field	Is it necessary	type	describe
nodeID	Yes	character	Node factory serial number
timestamp	Yes	character	timestamp

```
{
  "msgType": "nodeLogoutAck",
  "gwID": "GW80210317018762",
  "ackData": {
    "nodeID": "210315018657",
    "timestamp": "1615797310"
  }
}
```

3.3. Collected copy

Request frame format:

field	Is it necessary	type	describe
msgType	Yes	character	collecNodeData
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	null

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	collecNodeDataAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
nodeID	Yes	character	nodeUnique ID, 12bit character
model	Yes	Integer	Node model
data	Yes	character	data root node
payload	no	character	Serial port collection value. model==0 is valid
ai	no	character	Analog acquisition value. model==1 model==3 is valid
di	no	character	Switch value acquisition value. model==2 model==3 is valid
rsi	Yes	character	signal strength
rsiAvg	Yes	character	Average signal value
snr	Yes	character	signal to noise ratio
timestamp	Yes	character	timestamp

Request frame example:

```
{
  "msgType": "collecNodeData",
  "gwID": "GW80210401020272",
  "reqData": {}
}
```

Response frame format:

```
{
  "msgType": "collecNodeDataAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "nodeID": "210412020661",
    "model": "1",
    "data": [{
      "ai": "0"
    }],
    "rsi": "-18",
    "rsiAvg": "-18",
    "snr": "10",
    "timestamp": "1618642087"
  }
}
```

3.4. Supplementary copy

Reporting frame format:

field	Is it	type	describe
-------	-------	------	----------

	necessary		
msgType	Yes	character	remedyNodeDataAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
nodeID	Yes	character	nodeUnique ID, 12bit character
model	Yes	Integer	Node model
data	Yes	character	data root node
payload	no	character	Serial port collection value. model==0 is valid
ai	no	character	Analog acquisition value. model==1 model==3 is valid
di	no	character	Switch value acquisition value. model==2 model==3 is valid
rssI	Yes	character	signal strength
rssIAvg	Yes	character	Average signal value
snr	Yes	character	signal to noise ratio
timestamp	Yes	character	timestamp

Response frame format:

```
{
  "msgType": "remedyNodeDataAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "nodeID": "210412020661",
    "model": "1",
    "data": [{
      "ai": "0"
    }],
    "rssI": "-16",
    "rssIAvg": "-16",
    "snr": "11",
    "timestamp": "1618642195"
  }
}
```

3.5. Nodes report actively

Reporting frame format:

field	Is it necessary	type	describe
msgType	Yes	character	triggerNodeDataAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
nodeID	Yes	character	nodeUnique ID, 12bit character
model	Yes	Integer	Node model
data	Yes	character	data root node
payload	no	character	Serial port collection value. model==0 is valid
ai	no	character	Analog acquisition value. model==1 model==3 is valid
di	no	character	Switch value acquisition value. model==2 model==3 is valid
rsi	Yes	character	signal strength
rsiAvg	Yes	character	Average signal value
snr	Yes	character	signal to noise ratio
timestamp	Yes	character	timestamp

Response frame format:

```

{
  "msgType": "triggerNodeDataAck",
  "gwID": "GW80210401020272",
  "ackData": {
    "status": "0",
    "nodeID": "210412020661",
    "model": "1",
    "data": [{
      "ai": "0"
    }],
    "rsi": "-16",
    "rsiAvg": "-16",
    "snr": "11",
    "timestamp": "1618642195"
  }
}
    
```

3.6. Click to copy

Request format:

field	Is it necessary	type	describe
msgType	Yes	character	pointNodeData
gwID	Yes	character	Concentrator unique identification ID, 16 characters
reqData	Yes	character	reqDataframe format

reqData frame format

field	Is it necessary	type	describe
nodeID	Yes	character	Node unique ID, 12 characters

Response frame format:

field	Is it necessary	type	describe
msgType	Yes	character	pointNodeDataAck
gwID	Yes	character	Concentrator unique identification ID, 16 characters
ackData	Yes	character	ackDataframe format

ackData frame format:

field	Is it necessary	type	describe
status	Yes	character	0:correct reply
nodeID	Yes	character	nodeUnique ID, 12bit character
model	Yes	Integer	Node model
data	Yes	character	data root node
payload	no	character	Serial port collection value. model==0 is valid
ai	no	character	Analog acquisition value. model==1 model==3 is valid
di	no	character	Switch value acquisition value. model==2 model==3 is valid
rssI	Yes	character	signal strength
rssIAvg	Yes	character	Average signal value
snr	Yes	character	signal to noise ratio
timestamp	Yes	character	timestamp

Request frame example:

```
{
  "msgType": "pointNodeData",
  "gwID": "GW80210401020272",
```

```
"reqData": {  
  "nodeID": "210412020661"  
}  
}
```

Example of response frame:

```
{  
  "msgType": "pointNodeDataAck",  
  "gwID": "GW80210401020272",  
  "ackData": {  
    "status": "0",  
    "nodeID": "210412020661",  
    "model": "1",  
    "data": [{  
      "ai": "0"  
    }],  
    "rssi": "-13",  
    "rssiAvg": "-12",  
    "snr": "10",  
    "timestamp": "1618642662"  
  }  
}
```