**Bridge APP command format and an example**

**NXP Semiconductors.**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Authors | Revision | Description |
| 06/21/2022 | Qiankun Li | 1.0 | Initial draft. |

Table of Contents

[1. The format of command and response 4](#_Toc106712602)

[1.1 Command Header Format 4](#_Toc106712603)

[1.2 Response Header Format 4](#_Toc106712604)

[2. An example 5](#_Toc106712605)

[2.1 The structure of command/response and tlv header 5](#_Toc106712606)

[2.2 wlan-scan 5](#_Toc106712607)

[2.2.1 Usage 5](#_Toc106712608)

[2.2.2 Command Format 6](#_Toc106712609)

[2.3 get-scan-result 6](#_Toc106712610)

[2.3.1 Usage 6](#_Toc106712611)

[2.3.2 Command Format 6](#_Toc106712612)

[2.3.3 Response Format 6](#_Toc106712613)

[2.4 wlan-connect 8](#_Toc106712614)

[2.4.1 Usage 8](#_Toc106712615)

[2.4.2 Command Format 8](#_Toc106712616)

[2.5 get-connect-result 9](#_Toc106712617)

[2.5.1 Usage 9](#_Toc106712618)

[2.5.2 Command Format 9](#_Toc106712619)

[2.5.3 Response Format 10](#_Toc106712620)

[2.6 wlan-disconnect 11](#_Toc106712621)

[2.6.1 Usage 11](#_Toc106712622)

[2.6.2 Command Format 11](#_Toc106712623)

[2.6.3 Response Format 12](#_Toc106712624)

[2.7 ping 12](#_Toc106712625)

[2.7.1 Usage 12](#_Toc106712626)

[2.7.2 Command Format 12](#_Toc106712627)

[2.8 get-ping-result 13](#_Toc106712628)

[2.8.1 Usage 13](#_Toc106712629)

[2.8.2 Command Format 14](#_Toc106712630)

[2.8.3 Response Format 14](#_Toc106712631)

[2.9 iperf 15](#_Toc106712632)

[2.9.1 Usage 15](#_Toc106712633)

[2.9.2 Command Format 15](#_Toc106712634)

[2.10 get-iperf-result 17](#_Toc106712635)

[2.10.1 Usage 17](#_Toc106712636)

[2.10.2 Command Format 17](#_Toc106712637)

[2.10.3 Response Format 17](#_Toc106712638)

# The format of command and response

## 1.1 Command Header Format

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Size | Description |
| Command Code | 2 bytes | command code |
| Size | 2 bytes | Total bytes of command header and tlv |
| SeqNum | 1 byte | Command Sequence Number |
| BSS | 1 byte | Client/Server |
| Result | 2 bytes | Success/Fail/… |
| Action | 1 byte | Set/Get/… |

## 1.2 Response Header Format

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Size | Description |
| Command Code | 2 bytes | command code |
| Size | 2 bytes | Total bytes of response header and tlv |
| SeqNum | 1 byte | Copy from Command |
| BSS | 1 byte | Client/Server |
| Result | 2 bytes | 0:Success !0:Failed |
| Action | 1 byte | Set/Get/… |

# 2. An example

## 2.1 The structure of command/response and tlv header

typedef struct bridge\_commands

{

uint16\_t cmd;

uint16\_t size;

uint8\_t Seqnum;

uint8\_t bss;

uint16\_t result;

uint8\_t action;

}W\_CMD;

typdef struct TLVTypeHeader\_t

{

uint8\_t type;

uint16\_t size ;

} TypeHeader\_t;

typedef struct bridge\_reponses

{

uint16\_t cmd;

uint16\_t size;

uint8\_t Seqnum;

uint8\_t bss;

uint16\_t result;

uint8\_t action;

}W\_RES;

#define BRIDGE\_COMMAND\_LEN 9

#define TLV\_HEADER\_LEN 3

## 2.2 wlan-scan

### 2.2.1 Usage

wlan-scan

wlan\_scan\_command()(wifi\_bridge\_command.c) processes ‘wlan-scan’ command and converts it the specified command format.

### 2.2.2 Command Format

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x01 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT8 | 0x01 |

Function wlan\_bridge\_scan() (MCU bridge\_app) processes the scan command sent by LINUX APP.

## 2.3 get-scan-result

### 2.3.1 Usage

get-scan-result

Use this command to get the network info scanned by using ‘wlan-scan’. Because it takes some time to scan networks, it’s best to wait at least 20s to get network info.

### 2.3.2 Command Format

wlan\_get\_scan\_res\_command()(wifi\_bridge\_command.c) processes this command and converts it the specified command format.

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x01 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT8 | 0x00 |

wlan\_bridge\_get\_scan()(MCU bridge\_app) processes the command and returns the scan result to the LINUX APP in the specified command format.

### 2.3.3 Response Format

typedef struct SCAN\_NETWORK\_INFO

{

    uint8\_t mac[6];

    uint8\_t ssid[32];

    uint8\_t channel;

    uint8\_t rssi;

    uint8\_t security;

}scan\_network\_info;

typedef struct SCAN\_RES

{

    TypeHeader\_t header;

    uint8\_t network\_count;

    scan\_network\_info net\_info[20];

}SCAN\_res\_tlv;

#define SCAN\_NETWORK\_INFO\_LEN 41

The format of scan result response:

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action | type | size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| net\_count | scan\_network\_info |

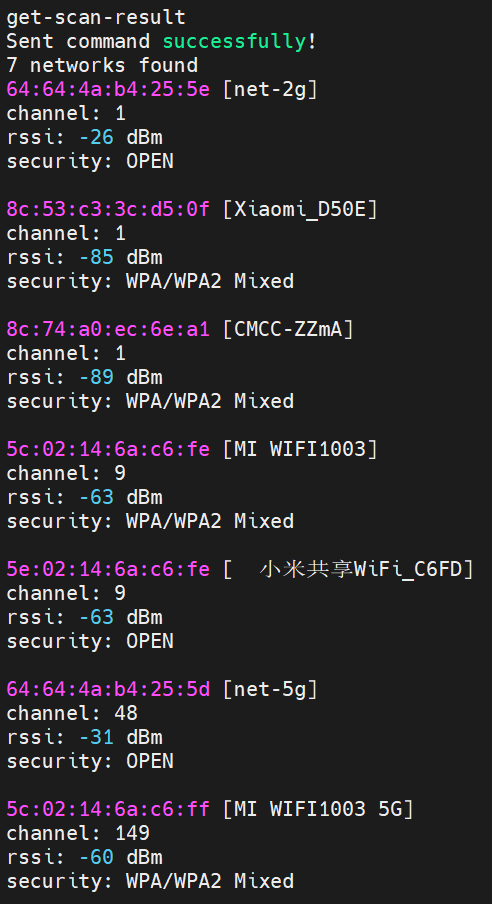
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| scan\_network\_info

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x01 |
| Size | UINT16 | BRIDGE\_COMMAND\_LEN + TLV\_HEADER\_LEN + network\_count \* SCAN\_NETWORK\_INFO\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0:Success !0:Failed |
| Action | UINT8 | 0x00 |
| SCAN\_res\_tlv | UINT8 | Scan result tlv |

Function wlan\_process\_scan\_response()(wifi\_bridge\_command.c) processes the response sent by bridge\_app. If the result is !0, there is on scan info appended to the response header.



## 2.4 wlan-connect

### 2.4.1 Usage

wlan-connect <ssid>

(Only supports connecting to unencrypted APs.)

### 2.4.2 Command Format

wlan\_connect\_command(char \*arg)(wifi\_bridge\_command.c) processes ‘wlan-connect <ssid>’ command and converts it the specified command format.

Para arg points to the ssid to be connected. The characters in ssid are converted to corresponding ASCII.

typedef struct SSID\_ParaSet

{

TypeHeader\_t header;

uint8\_t ssid[1];

}SSID\_tlv;

SSID\_tlv ssid\_tlv;

ssid\_tlv.header.type = 0x02;

ssid\_tlv.header.size = strlen(ssid);

ssid\_tlv.ssid = the ASCII of ssid(char\*);

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action | type | size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x02 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN + TLV\_HEADER\_LEN + ssid\_tlv.header.size |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT8 | 0x01 |
| SSID\_Tlv | UINT8 | ssid tlv |

Function wlan\_bridge\_connect(t\_u8 \*cmd)(MCU bridge\_app) processes the connect command sent by LINUX APP. Para cmd points to the received command.

## 2.5 get-connect-result

### 2.5.1 Usage

get-connect-result

Use this command to get the connected network info. Because it takes some time to connect AP, it’s best to wait at least 10s to get network info after entering ‘wlan-connect <ssid>’.

### 2.5.2 Command Format

wlan\_get\_connect\_res\_command()(wifi\_bridge\_command.c) processes this command and converts it the specified command format.

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x02 |
| Size | UINT16 | BRIDGE\_COMMAND\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT8 | 0x00 |

### 2.5.3 Response Format

wlan\_bridge\_get\_connect()(MCU bridge\_app) processes the command and returns the connected network info to the LINUX APP in the specified command format.

typedef struct CONNECT\_RES

{

    TypeHeader\_t header;

    uint8\_t ip[16];

    uint8\_t ssid[32];

}CONNECT\_res\_tlv;

#define BRIDGE\_CONNECT\_RES\_TLV\_LEN 48

The format of connected result response:

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action | type | size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ip address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ip address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ip address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ip address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

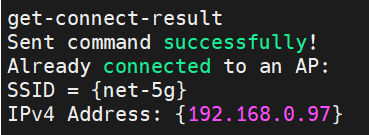
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ssid |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x02 |
| Size | UINT16 | BRIDGE\_COMMAND\_LEN + TLV\_HEADER\_LEN + BRIDGE\_CONNECT\_RES\_TLV\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0:Success !0:Failed |
| Action | UINT8 | 0x00 |
| SSID\_res\_tlv | UINT8 | IPv4 address ssid |

Function wlan\_process\_con\_response()(wifi\_bridge\_command.c) processes the response sent by bridge\_app.



## 2.6 wlan-disconnect

### 2.6.1 Usage

wlan-disconnect

Disconnect from the connected network.

### 2.6.2 Command Format

wlan\_disconnect\_command()(wifi\_bridge\_command.c) processes ‘wlan-disconnect’ command and converts it the specified command format.

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x03 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT8 | 0x01 |

### 2.6.3 Response Format

Function wlan\_bridge\_disconnect()(MCU bridge\_app) processes the connect command sent by LINUX APP. After disconnecting from the network, sends a response to the LINUX APP.

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x03 |
| Size | UINT16 | BRIDGE\_COMMAND\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0:Success !0:Failed |
| Action | UINT8 | 0x00 |

Function wlan\_process\_discon\_response()(wifi\_bridge\_command.c) processes the response sent by bridge\_app.

## 2.7 ping

### 2.7.1 Usage

ping [c <packet\_count>] <ip address>

### 2.7.2 Command Format

wlan\_ping\_command()(wifi\_bridge\_command.c) processes this command and converts it the specified command format.

typedef struct PING\_ParaSet

{

TypeHeader\_t header;

uint16\_t packet\_count;

uint8\_t ping\_ip[1];

}PING\_tlv;

PING\_tlv ping\_tlv;

ping\_tlv.header.type = 0x04;

ping\_tlv.header.size = strlen(ping\_ip) + sizeof(packet\_count);

ping\_tlv.packet\_count = the count of ping packets;

ping\_tlv.ping\_ip = <IP Address>

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action | type | size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| packet\_count | ping\_ip

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x04 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN + TLV\_HEADER\_LEN + ping\_tlv.header.size |
| SeqNum | UINT8 | 0X00 |
| BSS | UINT8 | 0X00 |
| Result | UINT16 | 0X00 |
| Action | UINT8 | 0X01 |
| PING\_Tlv | UINT8 | The tlv of ping parameters |

Function wlan\_bridge\_ping(t\_u8 \*cmd)(MCU bridge\_app) processes ‘ping’ command sent by LINUX APP. Para cmd points to the received command.

## 2.8 get-ping-result

### 2.8.1 Usage

get-ping-result

Use this command to get the ping statistics info. Because it takes some time to ping, it’s best to wait at least 10s(If the packet count is 10.) to get ping info after entering ‘ping <ip address>’.

### 2.8.2 Command Format

wlan\_get\_ping\_res\_command()(wifi\_bridge\_command.c) processes ‘get-ping-result’ command and converts it the specified command format.

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x04 |
| Size | UINT16 | BRIDGE\_COMMAND\_LEN |
| SeqNum | UINT8 | 0X00 |
| BSS | UINT8 | 0X00 |
| Result | UINT16 | 0X00 |
| Action | UINT16 | 0X00 |

### 2.8.3 Response Format

wlan\_bridge\_get\_ping()(MCU bridge\_app) processes the command and returns the ping statistics info to the LINUX APP in the specified command format.

typedef struct PING\_RES

{

    TypeHeader\_t header;

    uint8\_t status;

    uint32\_t packet\_transmit;

    uint32\_t packet\_received;

}PING\_res\_tlv;

#define PING\_RES\_INFO 9

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action | type | size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| status | packet\_trans |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|packet\_trans | packet\_recv |

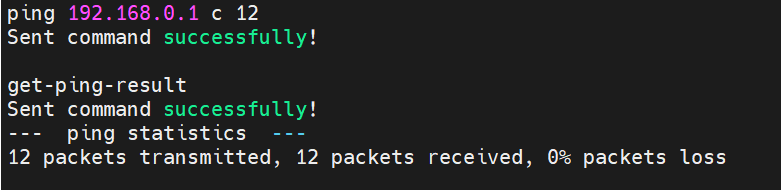
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| packet\_recv |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x04 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN + TLV\_HEADER\_LEN + PING\_RES\_INFO |
| SeqNum | UINT8 | 0X00 |
| BSS | UINT8 | 0X00 |
| Result | UINT16 | 0:Success !0:Failed |
| Action | UINT8 | 0X00 |
| PING\_Tlv | UINT8 | The tlv of ping parameters |

Function wlan\_process\_ping\_response()(wifi\_bridge\_command.c) processes the response sent by bridge\_app.



## 2.9 iperf

### 2.9.1 Usage

iperf s TCP Server

iperf su UDP Server

iperf c <IP address> t <time> TCP Client <time is optional, the default time is 10s>

iperf cu <IP address> t <time> UDP Client <time is optional, the default time is 10s>

iperf a Abort current iperf mode

### 2.9.2 Command Format

wlan\_iperf\_command()(wifi\_bridge\_command.c) processes ‘iperf’ command and converts it the specified command format.

typedef struct IPERF\_ParaSet

{

TypeHeader\_t header;

uint16\_t time;

uint8\_t iperf\_ip [1];

}IPERF\_tlv;

IPERF\_tlv iperf\_tlv;

TCP Server:

iperf\_tlv.header.type = 0x51;

iperf\_tlv.header.size = 0;

UDP Server:

iperf\_tlv.header.type = 0x52;

iperf\_tlv.header.size = 0;

TCP Client:

iperf\_tlv.header.type = 0x53;

iperf\_tlv.header.size = strlen(<IP Address>) + sizeof(iperf\_tlv.time);

iperf\_tlv.time = the time of throughput test(default time is 10s);

iperf\_tlv.iperf\_ip = <IP Address>

UDP Client:

Iperf\_tlv.header.type = 0x54;

iperf\_tlv.header.size = strlen(<IP Address>) + sizeoof(iperf\_tlv.time);

iperf\_tlv.time = the time of throughput test(default time is 10s);

iperf\_tlv.iperf\_ip = <IP Address>

Iperf Abort

perf\_tlv.header.type = 0x55;

iperf\_tlv.header.size = 0;

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action | type | size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| time | iperf\_ip

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x05 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN + TLV\_HEADER\_LEN + iperf\_tlv.header.size |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT8 | 0x01 |
| IPERF\_Tlv | UINT8 | iperf parameter |

Function wlan\_bridge\_iperf(t\_u8 \*cmd)(MCU bridge\_app) processes ‘iperf’ command sent by LINUX APP. Para cmd points to the ‘iperf’ command.

## 2.10 get-iperf-result

### 2.10.1 Usage

get-iperf-result

Use this command to get the iperf statistics info. Because it takes some time to test throughput, it’s best to wait until the throughput test is over.

### 2.10.2 Command Format

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x05 |
| Size | UINT16 | BRIDGE\_COMMAND\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT16 | 0x00 |

wlan\_bridge\_get\_iperf()(MCU bridge\_app) processes the command and returns ‘iperf’ statistics info to the LINUX APP in the specified command format.

### 2.10.3 Response Format

typedef struct IPERF\_INFO

{

    uint8\_t   role;

    uint8\_t   local\_addr[4];

    uint16\_t local\_port;

    uint8\_t   remote\_addr[4];

    uint16\_t remote\_port;

    uint64\_t bytes\_transferred;

    uint32\_t ms\_duration;

    uint32\_t bandwidth\_Mbitpsec;

}IPERF\_info;

typedef struct IPERF\_RES

{

    TypeHeader\_t header;

    IPERF\_info iperf\_info;

}IPERF\_res\_tlv;

#define IPERF\_INFO\_LEN 29

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Command code | Size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| SeqNum | BSS | Result |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Action | type | size |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| role | local\_addr |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| local\_addr | local \_port | remote\_addr |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| remote\_addr | remote\_port |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| remote\_port | bytes\_transfer |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| bytes\_transfer |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|bytes\_transfer | ms\_duration |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ms\_duration | bandwidth\_Mp |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|bandwidth\_Mp|

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|  |  |  |
| --- | --- | --- |
| Filed Name | Type | Description |
| Command Code | UINT16 | 0x05 |
| Size | UNIT16 | BRIDGE\_COMMAND\_LEN + TLV\_HEADER\_LEN + IPERF\_INFO\_LEN |
| SeqNum | UINT8 | 0x00 |
| BSS | UINT8 | 0x00 |
| Result | UINT16 | 0x00 |
| Action | UINT8 | 0x00 |
| IPERF\_res\_tlv | UINT8 | iperf throughput test info |

Function wlan\_process\_iperf\_response()(wifi\_bridge\_command.c) processes ‘iperf’ response sent by bridge\_app.

