This section describes the minimal implementation of SNTPv4 as specified in RFC 4330. It is simple "SNTP" client for the lwIP raw API.

# Features and Limitations

Following are the salient features of the SNTP implementation:

1. SNTP is a subset of the Network Time Protocol (NTP), with the latest version being SNTP v4.
2. It can synchronize seamlessly to full-blown NTP servers, though it was originally developed for small computers and micro-controllers.
3. Requires less memory and processing power than NTP.
4. It is used in applications where precise clock synchronization is not critical.
5. Uses the TCP/IP protocol suite, UDP port 123.

Following are the limitations:

1. One of the fundamental disadvantages of SNTP is that it can be configured to a function from a solitary time source only.
2. SNTP delivers a significantly lower-quality time sync solution than its NTP counterpart, lacking certain algorithms that ensure total time accuracy.
3. SNTP applications do not assess the stability or quality of time references.

# Header file/s

*Components/sntp/src/sntp.h*.

# Data Structure Definitions

## sntp\_msg

Defines SNTP, which is a protocol for synchronizing clocks across a WAN or LAN through a specific formatted message.

The SNTP data structure is as follows:

|  |
| --- |
| struct sntp\_msg {  PACK\_STRUCT\_FLD\_8(u8\_t li\_vn\_mode);  PACK\_STRUCT\_FLD\_8(u8\_t stratum);  PACK\_STRUCT\_FLD\_8(u8\_t poll);  PACK\_STRUCT\_FLD\_8(u8\_t precision);  PACK\_STRUCT\_FIELD(u32\_t root\_delay);  PACK\_STRUCT\_FIELD(u32\_t root\_dispersion);  PACK\_STRUCT\_FIELD(u32\_t reference\_identifier);  PACK\_STRUCT\_FIELD(u32\_t reference\_timestamp[2]);  PACK\_STRUCT\_FIELD(u32\_t originate\_timestamp[2]);  PACK\_STRUCT\_FIELD(u32\_t receive\_timestamp[2]);  PACK\_STRUCT\_FIELD(u32\_t transmit\_timestamp[2]);  } PACK\_STRUCT\_STRUCT; |

where,

|  |  |
| --- | --- |
| ***sntp\_msg*** | Simple network time message protocol. |
| ***li\_vn\_mode*** | * li. Two bits. Leap indicator. * vn. Three bits. Version number of the protocol. * mode. Three bits. Indicates mode. User will need to use 3 for CLI. |
| ***stratum*** | Stratum level of the local clock. |
| ***poll*** | Maximum interval between successive messages. |
| ***precision*** | Precision of the local clock. |
| ***root\_delay*** | Total round trip delay time. |
| ***root\_dispersion*** | Maximum error aloud from primary clock source. |
| ***reference\_identifier*** | Reference clock identifier. |
| ***reference\_timestamp*** | Reference time-stamp seconds. |
| ***originate\_timestamp*** | Originate time-stamp seconds. |
| ***receive\_timestamp*** | Received time-stamp seconds. |
| ***transmit\_timestamp*** | Transmit time-stamp seconds. |

Table 1: sntp\_msg - Data structure definitions

# API Reference

## sntp\_init()

### Overview

This API initializes the SNTP of the Talaria TWO module. Sends out a request instantly or after sntp\_startup\_delay(func).

### Definition

|  |
| --- |
| void sntp\_init (void) |

### Parameters

None.

### Return

None.

## sntp\_stop()

### Overview

This function stops the Talaria TWO module.

### Definition

|  |
| --- |
| void sntp\_stop (void) |

### Parameters

None.

### Return

None.

## sntp\_setoperatingmode()

### Overview

Sets the operating mode of the Talaria TWO module.

### Definition

|  |
| --- |
| void sntp\_setoperatingmode ( u8\_t operating\_mode) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *u8\_t operating\_mode* | One of the available operating modes. |

Table 2: sntp\_setoperatingmode - parameter descriptions

### Return

None.

## sntp\_servermode\_dhcp()

### Overview

This function configures the SNTP with IP address, name of Talaria TWO Module or DHCP.

### Definition

|  |
| --- |
| void sntp\_servermode\_dhcp(int set\_servers\_from\_dhcp) |

### Parameters

|  |  |
| --- | --- |
| **Parameters** | **Description** |
| *set\_servers\_from\_dhcp* | Enable or disable procuring server addresses from DHCP. |

Table 3: sntp\_servermode\_dhcp - parameter description

### Return

None.

## sntp\_setservername

### Overview

This function initializes one of the NTP servers via the IP address of the Talaria TWO module.

### Definition

|  |
| --- |
| void sntp\_setservername(u8\_t idx, char \*server) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *idx* | Index of the NTP server. Its value must be less than SNTP\_MAX\_SERVERS. |
| *server* | DNS name of the NTP server to set, to be resolved at contact time |

Table 4: sntp\_setserver - parameter description

### Return

None.

## sntp\_getservername

### Overview

This function obtains one of the currently configured NTP servers by IP address.

### Definition

|  |
| --- |
| char \* sntp\_getservername(u8\_t idx) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *idx* | Index of the NTP server. |

Table 5: sntp\_getservername - parameter description

### Return

Success: IP address of the indexed NTP server.

Error: NULL. NTP server has not been configured by name (or at all).

## sntp\_retry

### Overview

This function sends a new request with increased retry timeout.

### Definition

|  |
| --- |
| static void sntp\_retry(void\* arg) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *arg* | Unused (only necessary to conform to sys\_timeout). |

Table 6: sntp\_retry - parameter description

### Return

None.

## sntp\_try\_next\_server

### Overview

This function tries the next server or retries the current server with increased retry timeout.

### Definition

|  |
| --- |
| static void sntp\_try\_next\_server(void\* arg) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *arg* | Unused (only necessary to conform to sys\_timeout). |

Table 7: sntp\_try\_next\_server - parameter description

### Return

None.

## sntp\_request

### Overview

This function sends out an SNTP request to the server.

### Definition

|  |
| --- |
| static void sntp\_request(void \*arg) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *arg* | Unused (only necessary to conform to sys\_timeout). |

Table 8: sntp\_request - parameter description

### Return

None.

# Example Application

For the example codes, refer: *component\sntp\_src.c* application.