

# ワイヤレス LAN コントローラでの NTP の設定例

## 目次

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[ワイヤレス LAN コントローラにおけるシステム日付と時刻の管理](#)

[設定](#)

[ネットワーク図](#)

[設定](#)

[L3 の設定保証された NTP サーバとして切り替えて下さい](#)

[NTP 認証の設定](#)

[NTP サーバのための WLC を設定して下さい](#)

[確認](#)

[NTP サーバ](#)

[WLC](#)

[GUI](#)

[WLC CLI](#)

[トラブルシューティング](#)

## 概要

このドキュメントでは、ネットワーク タイム プロトコル ( NTP ) サーバと日時を同期するためにワイヤレス LAN コントローラ ( WLC ) を設定する方法について説明します。

## 前提条件

### 要件

この設定を行う前に、次の要件が満たされていることを確認します。

- Cisco WLCs の設定の基本的な知識。
- NTP の基本的な知識。

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

- ソフトウェア バージョン 8.8.110.0 を実行する Cisco WLC 3504。
- Cisco Catalyst 3560-CX シリーズ L3 スイッチ Cisco IOS® ソフトウェア リリース 15.2(6)E2 を実行する。

本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。

このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。稼働中のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してください。

## ワイヤレス LAN コントローラにおけるシステム日付と時刻の管理

WLC では、システム日付と時刻を WLC から手動で設定するか、NTP サーバから日付と時刻を取得するように設定できます。

システム日付と時刻を手動で設定するには、CLI コンフィギュレーション ウィザードまたは WLC GUI/CLI を使用します。

このドキュメントでは、WLC のシステム日付と時刻を NTP サーバで同期する設定例を示します。

ネットワーク タイム プロトコル（NTP）は時間の参照にコンピュータのクロックを同期化する変数レイテンシー データネットワーク上の計算機 システム間のクロック 同期のためのネットワーク プロトコルです。[RFC 1305](#) および [RFC 5905](#) は NTPv3 および NTPv4 実装で詳細な情報を、それぞれ提供します。

NTP のネットワークでは通常、タイム サーバに接続された電波時計や原子時計など正規の時刻源から時刻を取得します。NTP は、この時間をネットワーク全体に配信します。

NTP クライアントは NTP サーバとクライアント間のネットワークの状態によって動的に一定時間にわたり変更するポーリング間隔上のサーバのトランザクションを作ります。

NTP では、信頼できるタイム ソースから各マシンが何 NTP ホップ隔たっているかを表すために、ストラタムという概念が使用されます。たとえば、ストラタム 1 のタイム サーバに電波時計または原子時計が直接接続されているとします。このタイム サーバはストラタム 2 のタイム サーバに NTP で時間を配信します。このサーバはさらに別のマシンへ時間を再配信します。

NTP 配備のための最良の方法に関する詳細については、[toNetwork タイムプロトコル](#)を参照して下さい:[ベスト プラクティス ホワイト ペーパー](#)」を参照してください。

この資料の例は NTP サーバとして Cisco Catalyst 3560-CX シリーズ L3 スイッチを使用します。この NTP サーバと日時を同期するように、WLC を設定します。

## 設定

### ネットワーク図

WLC ---- 3560-CX L3 スイッチ ---- NTP サーバ

### 設定

L3 の設定保証された NTP サーバとして切り替えて下さい

システムを正規の NTP サーバにする場合は、グローバル コンフィギュレーション モードで次の

コマンドを使用します。これは、システムが外部の時刻源と同期されていない場合でも同じです。

```
#ntp master
!--- Makes the system an authoritative NTP server
```

## NTP 認証の設定

セキュリティ上の目的でその他のシステムとの関連付けを認証する場合は、次のコマンドを使用します。最初のコマンドにより、NTP 認証機能が有効になります。2 番目のコマンドにより、それぞれの認証キーが定義されます。キーごとに、キー番号、タイプ、および値を 1 つずつ指定します。現在サポートされているキータイプは md5 だけです。3 番目のコマンドにより、「信頼される」認証キーのリストが定義されます。キーを信頼できる場合、このシステムは、このキーを NTP パケット内で使用する別のシステムに同期できます。NTP 認証を設定するには、次のコマンドをグローバル コンフィギュレーション モードで使用します。

```
#ntp authenticate
!--- Enables the NTP authentication feature #ntp authentication-key number md5 value !---
Defines the authentication keys #ntp trusted-key key-number !--- Defines trusted authentication keys
```

3560-CX L3 スイッチの NTP サーバコンフィギュレーション例ここにあります。スイッチは NTP マスターです、自体別の NTP サーバ「pool.ntp.org」としてから時間を取得していることをルータは保証された NTP サーバ機能するが、ことを意味する。

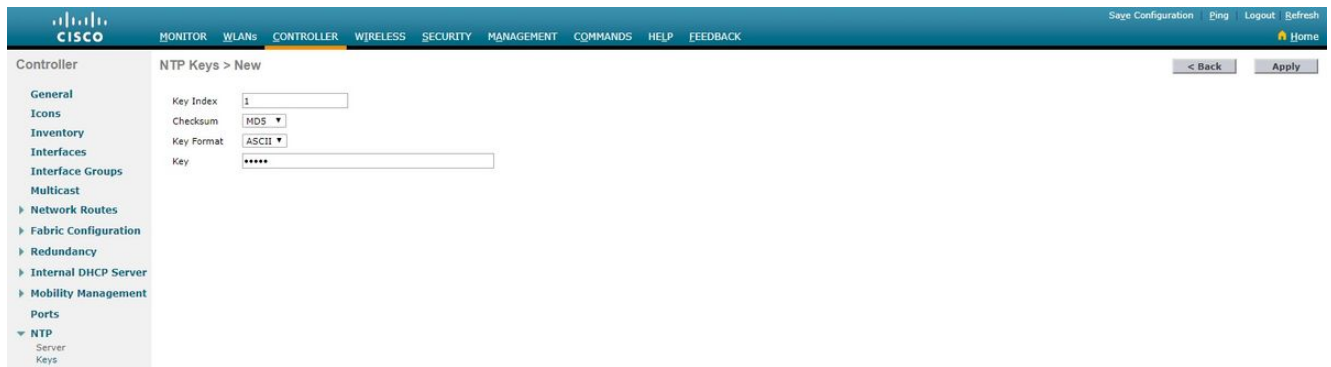
```
(config)#ntp authentication-key 1 md5 1511021F0725 7
(config)#ntp authenticate
(config)#ntp trusted-key 1
(config)#ntp master
(config)#ntp server pool.ntp.org
```

## NTP サーバのための WLC を設定して下さい

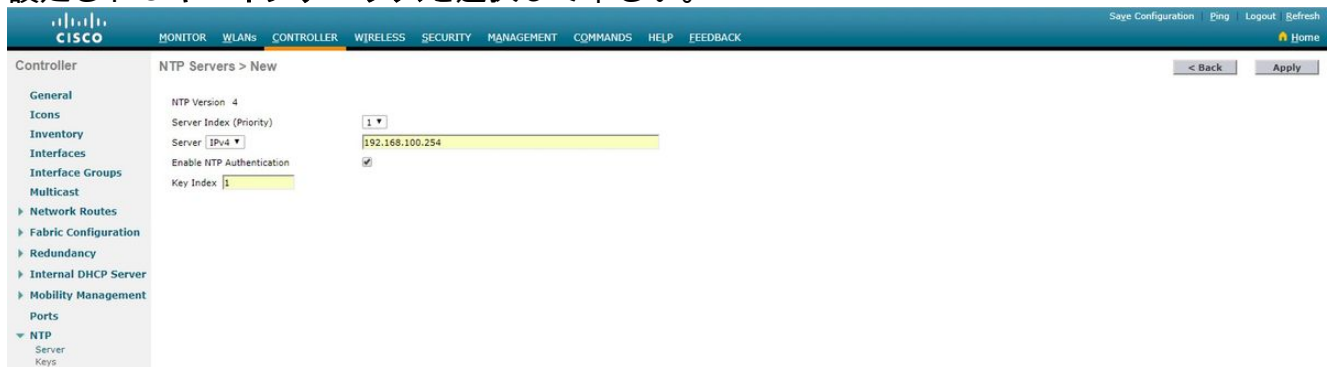
バージョン 8.6 から始まって NTPv4 を有効にすることができます。またコントローラと NTP サーバ間の認証チャンネルを設定できます。

コントローラの GUI を使用して NTP 認証を設定するには、次の手順を実行します。

1. ChooseController > NTP > キー。
  2. ClickNewto はキーを作成します。
  3. theKey Indextext ボックスでキーインデックスを入力して下さい。
  4. キー チェックサム ( MD5 か SHA1 ) および theKey Formatdrop-down リストを選択して下さい。
  5. theKeytext ボックスでキーを入力して下さい
- :

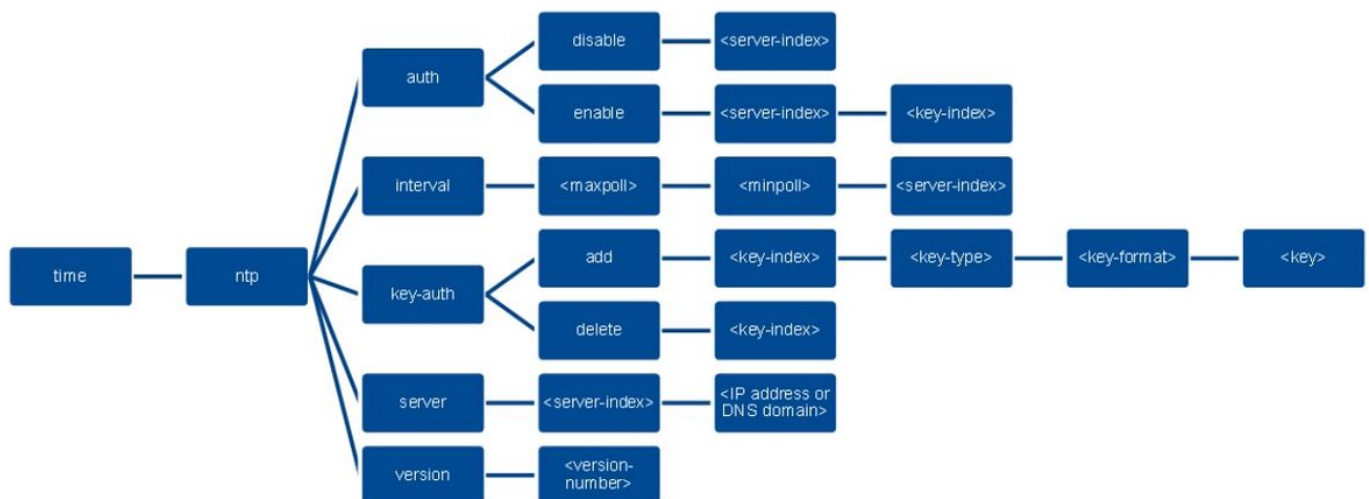


6. ChooseController > NTP > Serversto は NTP サーバ ページを開きます。 選定されたバージョン 3 または 4 はそれから clickNewto NTP サーバを追加し。 TheNTP サーバ > 新しい pageappears。
7. サーバ インデックス ( 優先順位 ) を選択して下さい。
8. theServer IP Adresstext ボックスで NTP サーバのIPアドレスを入力して下さい。
9. NTP サーバ認証を NTP サーバ認証 チェックボックスの選択によって有効にし、 前もって 設定されるキーインデックスを選択して下さい。



10. ClickApply.

NTP 認証をコントローラ CLI を使用して設定するため、followe このコマンド ツリー:



```
>config time ntp version 4
>config time ntp key-auth add 1 md5 ascii cisco
>config time ntp server 1 192.168.100.254
>config time ntp auth enable 1 1
```

## 確認

## NTP サーバ

```
#show ntp status
Clock is synchronized, stratum 3, reference is 193.136.152.72
nominal freq is 286.1023 Hz, actual freq is 286.0901 Hz, precision is 2**21
ntp uptime is 6591900 (1/100 of seconds), resolution is 3496
reference time is E007C909.80902653 (09:23:21.502 UTC Fri Feb 8 2019)
clock offset is 0.3406 msec, root delay is 59.97 msec
root dispersion is 25.98 msec, peer dispersion is 1.47 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000042509 s/s
system poll interval is 128, last update was 7 sec ago.
```

```
#show ntp associations
```

```
address ref clock st when poll reach delay offset disp
*~193.136.152.72 138.96.64.10 2 20 1024 17 13.634 0.024 1.626
~127.127.1.1 .LOCL. 7 9 16 377 0.000 0.000 0.232
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
#show ntp information
```

```
Ntp Software Name : Cisco-ntp4
Ntp Software Version : Cisco-ntp4-1.0
Ntp Software Vendor : CISCO
Ntp System Type : Cisco IOS / APM86XXX
```

## WLC

## GUI

通信の確立の間:

The screenshot shows the Cisco GUI for NTP Servers configuration. The 'NTP Servers' section is active, displaying a table with one server entry:

Server Index	Server Address(Ipv4/Ipv6)	Key Index	Key Type	Max Polling Interval	Min Polling Interval
1	192.168.100.254	1	MDS	10	6

Below the table, the 'NTP Query Status' section shows the following output:

```
ind assid status conf reach auth condition last_event cnt src_addr
-----
1 51059 c011 yes no bad reject mobilize 1 192.168.100.254
```

確立される接続の後:

The screenshot shows the Cisco GUI for NTP Servers configuration after successful connection establishment. The 'NTP Servers' section is active, displaying the same table as before:

Server Index	Server Address(Ipv4/Ipv6)	Key Index	Key Type	Max Polling Interval	Min Polling Interval
1	192.168.100.254	1	MDS	10	6

Below the table, the 'NTP Query Status' section shows the following output:

```
ind assid status conf reach auth condition last_event cnt src_addr
-----
1 51059 f63a yes yes ok sys.peer sys_peer 3 192.168.100.254
```

## WLC CLI

(Cisco Controller) >show time

Time..... Fri Feb 8 10:14:47 2019

Timezone delta..... 0:0

Timezone location.....

NTP Servers

NTP Version..... 4

Index NTP Key NTP Server NTP Key Polling Intervals

Index Type Max Min

-----  
1 1 192.168.100.254 MD5 10 6

NTPQ status list of NTP associations

assoc

ind assid status conf reach auth condition last\_event cnt src\_addr

=====  
1 1385 f63a yes yes ok sys.peer sys\_peer 3 192.168.100.254

(Cisco Controller) >

## トラブルシューティング

Cisco IOS を実行する NTP サーバ側ですべてのイネーブル デバッグ NTP」を使用「できます:

```
#debug ntp all
```

```
NTP events debugging is on
```

```
NTP core messages debugging is on
```

```
NTP clock adjustments debugging is on
```

```
NTP reference clocks debugging is on
```

```
NTP packets debugging is on
```

```
#
```

```
(communication between SW and NTP server pool.ntp.org)
```

```
Feb 8 09:52:30.563: NTP message sent to 195.22.17.7, from interface 'Vlan1' (192.168.1.81).
```

```
Feb 8 09:52:30.577: NTP message received from 195.22.17.7 on interface 'Vlan1' (192.168.1.81).
```

```
Feb 8 09:52:30.577: NTP Core(DEBUG): ntp_receive: message received
```

```
Feb 8 09:52:30.577: NTP Core(DEBUG): ntp_receive: peer is 0x0D284B34, next action is 1.
```

```
(communication between SW and WLC)
```

```
Feb 8 09:53:10.421: NTP message received from 192.168.100.253 on interface 'Vlan100'  
(192.168.100.254).
```

```
Feb 8 09:53:10.421: NTP Core(DEBUG): ntp_receive: message received
```

```
Feb 8 09:53:10.421: NTP Core(DEBUG): ntp_receive: peer is 0x00000000, next action is 3.
```

```
Feb 8 09:53:10.421: NTP message sent to 192.168.100.253, from interface 'Vlan100'  
(192.168.100.254).
```

```
(communication between SW and NTP server pool.ntp.org)
```

```
Feb 8 09:53:37.566: NTP message sent to 195.22.17.7, from interface 'Vlan1' (192.168.1.81).
```

```
Feb 8 09:53:37.580: NTP message received from 195.22.17.7 on interface 'Vlan1' (192.168.1.81).
```

```
Feb 8 09:53:37.580: NTP Core(DEBUG): ntp_receive: message received
```

```
Feb 8 09:53:37.580: NTP Core(DEBUG): ntp_receive: peer is 0x0D284B34, next action is 1.
```

```
(communication between SW and WLC)
```

```
Feb 8 09:54:17.421: NTP message received from 192.168.100.253 on interface 'Vlan100'  
(192.168.100.254).
```

```
Feb 8 09:54:17.421: NTP Core(DEBUG): ntp_receive: message received
```

Feb 8 09:54:17.421: NTP Core(DEBUG): ntp\_receive: peer is 0x00000000, next action is 3.  
Feb 8 09:54:17.421: NTP message sent to 192.168.100.253, from interface 'Vlan100'  
(192.168.100.254).

## WLC 側:

```
>debug ntp ?
```

detail Configures debug of detailed NTP messages.  
low Configures debug of NTP messages.  
packet Configures debug of NTP packets.

*(at the time of writting this doc there was a DDTS [CSCvo29660](#) on which the debugs of ntpv4 are not printed in the CLI. The below debugs are using NTPv3.)*

```
(Cisco Controller) >debug ntp detail enable
```

```
(Cisco Controller) >debug ntp packet enable
```

```
(Cisco Controller) >*emWeb: Feb 08 11:26:53.896: ntp Auth key Info = -1
```

```
*emWeb: Feb 08 11:26:58.143: ntp Auth key Info = -1
```

```
*emWeb: Feb 08 11:26:58.143: ntp Auth key Info = -1
```

```
*emWeb: Feb 08 11:26:58.143: Key Id = 1 found at Local Index = 0
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Initiating time sequence
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Fetching time from:192.168.100.254
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Started=3758614018.143350 2019 Feb 08 11:26:58.143
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: hostname=192.168.100.254 hostIdx=1 hostNum=0
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Looking for the socket addresses
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: NTP Polling cycle: accepts=0, count=5, attempts=1,  
retriesPerHost=6. Outgoing packet on NTP Server on socket 0:
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: sta=0 ver=3 mod=3 str=15 pol=8 dis=0.000000 ref=0.000000
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: ori=0.000000 rec=0.000000
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: tra=3758614018.143422 cur=3758614018.143422
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Host Supports NTP authentication with Key Id = 1
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: NTP Auth Key Id = 1 Key Length = 5
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: MD5 Hash and Key Id added in NTP Tx packet
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: 00000000: 1b 0f 08 00 00 00 00 00 00 00 00 00 00 00 00 00  
.....
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
.....
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: 00000020: 00 00 00 00 00 00 00 00 e0 07 e6 02 24 b7 50 00  
.....$.P.
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: 00000030: 00 00 00 01 e4 35 f3 1a 89 f0 93 c5 51 c7 c5 23  
.....5.....Q..#
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: 00000040: 01 dd 67 e0 ..g.
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Flushing outstanding packets
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Flushed 0 packets totalling 0 bytes
```

```
*sntpReceiveTask: Feb 08 11:26:58.143: Packet of length 68 sent to ::ffff:192.168.100.254  
UDPport=123
```

```
*emWeb: Feb 08 11:26:58.143: ntp Auth key Info = 0
```

```
*emWeb: Feb 08 11:26:58.143: idx != 0 : ntp key Id = 1 Msg auth Status = 66

*sntpReceiveTask: Feb 08 11:26:58.146: Packet of length 68 received from ::ffff:192.168.100.254
UDPport=123
*sntpReceiveTask: Feb 08 11:26:58.146: Incoming packet on socket 0: has Authentication Enabled
*sntpReceiveTask: Feb 08 11:26:58.146: 00000000: 1c 04 08 eb 00 00 0e a0 00 00 0b 2e c3 16 11 07
.....
*sntpReceiveTask: Feb 08 11:26:58.146: 00000010: e0 07 e5 f8 d3 21 bf 57 e0 07 e6 02 24 b7 50 00
.....!.W....$.P.
*sntpReceiveTask: Feb 08 11:26:58.146: 00000020: e0 07 e6 02 24 e5 e3 b4 e0 07 e6 02 24 f3 c7 5a
....$......$.Z
*sntpReceiveTask: Feb 08 11:26:58.146: 00000030: 00 00 00 01 32 e4 26 47 33 16 50 bd d1 37 63 b7
....2.&G3.P..7c.
*sntpReceiveTask: Feb 08 11:26:58.146: KeyId In Recieved NTP Packet 1

*sntpReceiveTask: Feb 08 11:26:58.146: KeyId 1 found in recieved NTP packet exists as part of
the trusted Key/s

*sntpReceiveTask: Feb 08 11:26:58.146: The NTP trusted Key Id 1 length = 5

*sntpReceiveTask: Feb 08 11:26:58.146: NTP Message Authentication - SUCCESS

*sntpReceiveTask: Feb 08 11:26:58.146: sta=0 ver=3 mod=4 str=4 pol=8 dis=0.043671
ref=3758614008.824734

*sntpReceiveTask: Feb 08 11:26:58.146: ori=3758614018.143422 rec=3758614018.144133
*sntpReceiveTask: Feb 08 11:26:58.146: Offset=-0.000683+/-0.002787 disp=1.937698

*sntpReceiveTask: Feb 08 11:26:58.146: best=-0.000683+/-0.002787

*sntpReceiveTask: Feb 08 11:26:58.146: accepts=1 rejects=0 flushes=0

*sntpReceiveTask: Feb 08 11:26:58.146: Correction: -0.000683 +/- 0.002787 disp=1.937698

*sntpReceiveTask: Feb 08 11:26:58.146: Setting clock to 2019 Feb 08 11:26:58.145 + 0.001 +/-
1.940 secs
*sntpReceiveTask: Feb 08 11:26:58.146: correction -0.001 +/- 1.938+0.003 secs - ignored
```

(Cisco Controller) >