

排查和调试网络时间协议(NTP)问题

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[NTP show命令](#)

[show ntp association](#)

[show ntp association detail](#)

[show ntp status](#)

[使用调试排除NTP故障](#)

[未收到NTP数据包](#)

[未处理NTP数据包](#)

[同步丢失](#)

[debug ntp validity](#)

[debug ntp packets](#)

[debug ntp sync and debug ntp events](#)

[手动设置NTP时钟周期](#)

[相关信息](#)

简介

本文档介绍如何使用 `debug` 命令和 `show ntp` 命令对网络时间协议(NTP)问题进行故障排除。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处

于活动状态，请确保您了解所有命令的潜在影响。

NTP show命令

在查看NTP问题的原因之前，您必须了解以下命令的用法和输出：

- show ntp association
- show ntp association detail
- show ntp status

注意：请使用命令查找工具获取有关本部分所用命令的详细信息。只有思科注册用户才能访问内部工具和信息。

注意：命令输出解释程序工具支持某些show命令。使用输出解释器工具来查看 show 命令输出的分析。只有思科注册用户才能访问内部工具和信息。

show ntp association

NTP关联可以是对等关联（一个系统愿意与另一个系统同步或允许另一个系统与其同步）或服务器关联（只有一个系统与另一个系统同步，而不是与另一个系统同步）。

以下是show ntp association命令的输出示例：

```
CLA_PASA#sh ntp association
  address      ref clock    st  when  poll reach  delay  offset  disp
~10.127.7.1    10.127.7.1   9   50    64  377    0.0    0.00    0.0
~10.50.44.69   10.50.36.106 5  21231 1024  0     3.8    -4.26  16000.
+~10.50.44.101 10.50.38.114 5   57    64   1     3.6    -4.30  15875.
+~10.50.44.37  10.50.36.50  5    1    256  377    0.8    1.24   0.2
```

~10.50.44.133	10.50.38.170	5	12142	1024	0	3.2	1.24	16000.
+~10.50.44.165	10.50.38.178	5	35	256	357	2.5	-4.09	0.2
+~10.50.38.42	10.79.127.250	4	7	256	377	0.8	-0.29	0.2
*~10.50.36.42	10.79.127.250	4	188	256	377	0.7	-0.17	0.3
+~10.50.38.50	10.79.127.250	4	42	256	377	0.9	1.02	0.4
+~10.50.36.50	10.79.127.250	4	20	256	377	0.7	0.87	0.5

* primary (synced), # primary (unsynced), + selected, - candidate, ~ configured

期限	说明
	<p>地址前面的字符有以下定义：</p> <ul style="list-style-type: none"> * 已同步到此对等体 #几乎同步到此对等体 +为可能的同步选择对等体 - 对等体是候选选择 ~对等体是静态配置的
地址	<p>这是对等设备的IP地址。在示例中，第一个条目显示127.127.7.1。这表示本地计算机已与其自身同步。通常，只有一个NTP主设备与其自身同步。</p>
参考时钟	<p>这是对等设备的参考时钟的地址。在本例中，前六个对等体/服务器使用私有IP作为参考时钟，因此它们的主节点可能是本地网络中的路由器、交换机或服务器。对于最后四个条目，参考时钟是公共IP，因此它们的主时钟可能是公共时间源。</p>
st	<p>NTP使用层的概念来描述机器与权威时间源的距离（在NTP跳数中）。例如，第1层时钟服务器有无线电或原子时钟直接与它连接。它通过NTP将其时间发送到第2层时间服务器，以此类推，直到第16层。运行NTP的计算机自动选择层数最低的计算机进行通信，并使用NTP作为其时间源。</p>
何时	<p>自从对等设备收到上次NTP数据包以来的时间以秒为单位报告。该值必须小于轮询间隔。</p>
轮询	<p>轮询间隔以秒为单位报告。该间隔通常以至少64秒的轮询间隔开始。RFC指定同步两台计算机时每分钟不需要超过一个NTP事务。当客户端和服务器之间的NTP变得稳定时，轮询间隔可以小步从64秒增加到1024秒，并且通常在中间稳定下来。但是，此值会根据客户端和服务器之间的网络条件以及NTP数据包丢失动态更改。如果服务器在一段时间内无法访问，轮询间隔会逐步增加到1024秒，以减少网络开销。</p>

	<p>无法调整路由器上的NTP轮询间隔，因为内部轮询由启发式算法确定。</p>
<p>覆盖范围</p>	<p>对等体可达性是以八进制值报告的位字符串。此字段显示最后八个数据包是否由Cisco IOS®软件上的NTP进程接收。数据包必须由NTP进程接收、处理和接受为有效，而不只是由接收NTP IP数据包的路由器或交换机接收。</p> <p>Reach使用超时轮询间隔来确定是否收到数据包。轮询间隔是NTP在断定数据包丢失之前等待的时间。不同对等体的轮询时间可能不同，因此到达之前确定数据包丢失的时间对于不同对等体也可能不同。</p> <p>在本例中，有四个不同的到达值：</p> <ul style="list-style-type: none"> • 377八进制= 11111111二进制，表示NTP进程收到了最后八个数据包。 • 0 octal = 00000000，表示NTP进程未收到任何数据包。 • 1 octal = 00000001，表示NTP进程仅收到最新的数据包。 • 357 octal = 11101111，表示在最近四个数据包丢失之前的数据包。 <p>Reach是一个很好的指示器，指示是否由于链路不良、CPU问题和其他间歇性问题导致NTP数据包被丢弃。</p> <p>Unit Converter是用于此转换和许多其他转换的联机单元转换器。</p>
<p>延迟</p>	<p>对等设备的往返延迟报告以毫秒为单位。为了更准确地设置时钟，在设置时钟时间时会考虑此延迟。</p>
<p>偏移</p>	<p>Offset是对等体之间或主要与客户端之间的时钟时间差。此值是应用于客户端时钟以便进行同步的修正。正值表示服务器时钟较高。负值表示客户端时钟较高。</p>
<p>disp</p>	<p>Dispersion (以秒为单位)是本地时钟和服务器时钟之间观察到的最大时钟时间差。在本例中，服务器10.50.36.42的色散为0.3，因此本地时钟和服务器时钟之间观察到的最大时间差为0.3秒。</p>

时钟初始同步时，您会看到较高的值。但是，如果其他时间的分散度太高，则客户端上的NTP进程不接受来自服务器的NTP消息。最大分散是16000；在本例中，即服务器10.50.44.69和10.50.44.133的分散，因此本地客户端不接受来自这些服务器的时间。

如果到达率为零且分散度非常高，则客户端可能不接受来自该服务器的消息。请参阅本示例的第二行：

address	ref clock	st	when	poll	reach	delay	offset	disp
~10.50.44.69	10.50.36.106	5	21231	1024	0	3.8	-4.26	16000.

尽管偏移量仅为-4.26，但频散非常高（可能由于过去的事件），并且到达范围为零，因此该客户端不接受来自此服务器的时间。

show ntp association detail

以下是show ntp association detail命令的输出示例：

```
Router#sho ntp assoc detail
10.4.2.254 configured, our_primary, sane, valid, stratum 1
ref ID .GPS., time D36968AA.CC528FE7 (02:10:50.798 UTC Fri May 25 2012)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.44, reach 377, sync dist 207.565
delay 2.99 msec, offset 268.3044 msec, dispersion 205.54
precision 2**19, version 3
org time D36968B7.E74172BF (02:11:03.903 UTC Fri May 25 2012)
rcv time D36968B7.A2F44E2C (02:11:03.636 UTC Fri May 25 2012)
xmt time D36968B7.A21D3780 (02:11:03.633 UTC Fri May 25 2012)
filtdelay =    2.99    2.88  976.61  574.65  984.71  220.26  168.12    2.72
filtoffset =  268.30  172.15 -452.49 -253.59 -462.03  -81.98  -58.04   22.38
filterror =    0.02    0.99    1.95    1.97    2.00    2.01    2.03    2.04
```

```
10.3.2.254 configured, selected, sane, valid, stratum 1
ref ID .GPS., time D36968BB.B16C4A21 (02:11:07.693 UTC Fri May 25 2012)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 3.34, reach 377, sync dist 192.169
delay 0.84 msec, offset 280.3251 msec, dispersion 188.42
precision 2**19, version 3
org time D36968BD.E69085E4 (02:11:09.900 UTC Fri May 25 2012)
rcv time D36968BD.9EE9048B (02:11:09.620 UTC Fri May 25 2012)
xmt time D36968BD.9EA943EF (02:11:09.619 UTC Fri May 25 2012)
filtdelay =    0.84    0.75  663.68    0.67    0.72  968.05  714.07    1.14
filtoffset =  280.33  178.13 -286.52  42.88  41.41 -444.37 -320.25   35.15
filterror =    0.02    0.99    1.97    1.98    1.98    2.00    2.03    2.03
```

```
10.1.2.254 configured, insane, invalid, stratum 1
ref ID .GPS., time D3696D3D.BBB4FF24 (02:30:21.733 UTC Fri May 25 2012)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 4.15, reach 1, sync dist 15879.654
```

```
delay 0.98 msec, offset 11.9876 msec, dispersion 15875.02
precision 2**19, version 3
org time D3696D3D.E4C253FE (02:30:21.893 UTC Fri May 25 2012)
rcv time D3696D3D.E1D0C1B9 (02:30:21.882 UTC Fri May 25 2012)
xmt time D3696D3D.E18A748D (02:30:21.881 UTC Fri May 25 2012)
filtdelay =    0.98    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtoffset =   11.99    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtererror =    0.02 16000.0 16000.0 16000.0 16000.0 16000.0 16000.0 16000.0
```

此处不重复显示关联部分中已定义的术语。

说明

期限

已配置	此NTP时钟源已配置为服务器。此值也可以是动态的，对等体/服务器是动态发现的。
our_primary	本地客户端将同步到此对等体。
已选择	当“our_primary”失败或客户端失去同步时，将对等/服务器进行选择以进行可能的同步。
理智	健全性测试用于测试从服务器接收的NTP数据包。这些测试在 RFC 1305，网络时间协议（第3版）规范，实施和分析 中指定。测试包括：

测试 掩码		说明
1	0x01	收到重复的数据包
2	0x02	收到的伪造数据包
3	0x04	协议不同步
4	0x08	对等体延迟/色散未通过边界检查
5	0x10	对等身份验证失败
6	0x20	对等体时钟不同步 (对于未同步的服务器通用)
7	0x40	对等层超出范围
8	0x80	根延迟/色散边界检查失败

如果通过了测试1到4，则数据包数据有效。然后使用数据计算偏移、延迟和色散。

如果通过测试5到8，则数据包报头有效。只有具有有效报头的数据包可用于确定是否可以选择对等设备进行同步。

疯了

健全性检查失败，因此不接受来自服务器的时间。服务器未同步。

有效

对等/服务器时间有效。如果此对等体成为主要对等体，则本地客户端接受此时间。

无效

对等体/服务器时间无效，不能接受时间。

参考ID	每个对等体/服务器都分配有一个参考ID (标签)。
时间	Time是从该对等体/服务器收到的最后一个时间戳。
我们的模式 /对等模式	这是本地客户端/对等体的状态。
我们的poll intvl/peer poll intvl	这是从轮询到此对等设备或从对等设备到本地计算机的轮询间隔。
根延迟	根延迟是到NTP设置根的延迟 (以毫秒为单位)。第1层时钟被认为是NTP设置/设计的根。在本例中，所有三台服务器都可以是根服务器，因为它们都位于第1层。
根色散	根色散是本地时钟和根时钟之间观察到的最大时钟时间差。有关详细信息，请参阅显示关联下对“disp”的说明。
同步分发	<p>这是对层0源上的时间与客户端测量的时间之间的最大差值的估计；它包括自上次实际读取层源以来往返时间、系统精度和时钟漂移的组成部分。</p> <p>在包含多个层级的服务器/客户端的大型NTP设置 (INTERNET中第1层的NTP服务器，不同层级的服务器提供时间) 中，必须组织NTP同步拓扑以产生最高精度，但绝不能允许形成时间同步循环。另一个因素是，层中的每一次增量都涉及一个可能不可靠的时间服务器，这带来了额外的测量误差。NTP中使用的选择算法使用Bellman-Ford分布式路由算法的变体，以便计算根植在主服务器上的最小权重生成树。算法使用的距离度量包括层数加上同步距离，而同步距离本身则包括色散加上绝对延迟的一半。因此，同步路径始终将最小数量的服务器置于根目录；根据最大错误解决关联。</p>
延迟	这是到对等设备的往返延迟。
精度	这是以Hz为单位的对等时钟的精度。
version	这是对等体使用的NTP版本号。

组织时间	这是NTP数据包发送方的时间戳；换句话说，它是创建NTP数据包时而不是将数据包发送到本地客户端之前的对等时间戳。
RCV时间	<p>这是本地客户端收到消息的时间戳。组织时间和rcv时间之间的差异是此对等体的偏移量。在本示例中，主10.4.2.254具有以下时间：</p> <pre>org time D36968B7.E74172BF (02:11:03.903 UTC Fri May 25 2012) rcv time D36968B7.A2F44E2C (02:11:03.636 UTC Fri May 25 2012)</pre> <p>差异为268.3044毫秒的偏移量。</p>
xmt time	这是本地客户端发送到此对等体/服务器的NTP数据包的传输时间戳。
filtdelay filtoffset filtererror	<p>这是每个样本的往返延迟（以毫秒为单位）。 这是每个样本的时钟偏差（以毫秒为单位）。 这是每个样本的近似误差。</p> <p>示例是最后收到的NTP数据包。在本例中，主10.4.2.254具有以下值：</p> <pre>filtdelay = 2.99 2.88 976.61 574.65 984.71 220.26 168.12 2.72 filtoffset = 268.30 172.15 -452.49 -253.59 -462.03 -81.98 -58.04 22.38 filtererror = 0.02 0.99 1.95 1.97 2.00 2.01 2.03 2.04</pre> <p>这8个样本对应于reach字段的值，该字段显示本地客户端是否收到最后八个NTP数据包。</p>

show ntp status

以下是show ntp status命令的输出示例：

```
USSP-B33S-SW01#sho ntp status
Clock is synchronized, stratum 2, reference is 10.4.2.254
nominal freq is 250.0000 Hz, actual freq is 250.5630 Hz, precision is 2**18
```

reference time is D36968F7.7E3019A9 (02:12:07.492 UTC Fri May 25 2012)
clock offset is 417.2868 msec, root delay is 2.85 msec
root dispersion is 673.42 msec, peer dispersion is 261.80 msec

不重复在show up association部分或show ntp association detail部分中定义的术语。

期限	说明
精度	<p>精度是自动确定的，并且以2的幂来测量。在本例中，2**18表示$2^{(-18)}$，或3.8微秒。</p> <p>NTP对等体之间或主要与客户端之间的同步丢失可能是由多种原因造成的。NTP避免与时间在以下方面可能不明的机器同步：</p> <ol style="list-style-type: none">1. NTP从不与自身未同步的计算机同步。1. NTP会比较多台计算机报告的时间，并且不会与时间明显不同于其他计算机的计算机同步，即使其层级较低。

使用调试排除NTP故障

NTP问题的一些最常见原因是：

- 未收到NTP数据包。
- 收到NTP数据包，但思科IOS上的NTP进程不对其进行处理。
- NTP数据包会得到处理，但错误因素或数据包数据会导致同步丢失。
- 手动设置NTP时钟周期。

有助于查明这些问题的原因的重要debug命令包括：

- `debug ip packets <acl>`
- `debug ntp packets`
- `debug ntp validity`
- `debug ntp sync`
- `debug ntp events`

接下来的部分将介绍如何使用debug命令来解决这些常见问题。

注意：请使用命令查找工具获取有关本部分所用命令的详细信息。只有思科注册用户才能访问内部工具和信息。

注意：使用debug命令之前，请参阅[有关debug命令的重要信息](#)。

未收到NTP数据包

使用debug ip packet命令检查NTP数据包是否接收和发送。由于调试输出可能是频繁的，因此您可以使用访问控制列表(ACL)来限制调试输出。NTP使用用户数据报协议(UDP)端口123。

1. 创建ACL 101：

```
access-list 101 permit udp any any eq 123
access-list 101 permit udp any eq 123 any
```

NTP数据包的源端口和目的端口通常为123，因此这有助于：

```
permit udp any eq 123 any eq 123
```

2. 使用以下ACL以限制debug ip packet命令的输出：

```
debug ip packet 101
```

3. 如果特定对等体存在问题，请将ACL 101的范围缩小到这些对等体。如果对等体是172.16.1.1，请将ACL 101更改为：

```
access-list 101 permit udp host 172.16.1.1 any eq 123
access-list 101 permit udp any eq 123 host 172.16.1.1
```

此示例输出指示未发送数据包：

```
241925: Apr 23 2012 15:46:26.101 ETE: IP: s=10.50.38.70 (Tunnel99), d=10.50.44.101, len 76, input featur
241926: Apr 23 2012 15:46:26.101 ETE:      UDP src=123, dst=123, Ingress-NetFlow(13), rtype 0, forus FAL
sendself FALSE, mtu 0
241927: Apr 23 2012 15:46:26.101 ETE: IP: s=10.50.38.70 (Tunnel99), d=10.50.44.101, len 76, input featur
241928: Apr 23 2012 15:46:26.101 ETE:      UDP src=123, dst=123, MCI Check(55), rtype 0, forus FALSE,
sendself FALSE, mtu 0
```

确认未收到NTP数据包后，您必须：

- 检查NTP是否配置正确。
- 检查ACL是否阻止了NTP数据包。
- 检查到源IP或目标IP的路由问题。

未处理NTP数据包

启用debug ip packet和debug ntp packets命令后，您可以看到接收和传输的数据包，并且可以看到NTP对这些数据包起作用。对于收到的每个NTP数据包（如debug ip packet所示），都有由debug ntp packets生成的对应条目。

以下是NTP进程对收到的数据包工作时的调试输出：

```
Apr 20 00:16:34.143 UTC: IP: tableid=0, s=10.3.2.31 (local), d=10.1.2.254 (Vlan2), routed via FIB
.Apr 20 00:16:34.143 UTC: IP: s=10.3.2.31 (local), d=10.1.2.254 (Vlan2), len 76, sending
.Apr 20 00:16:34.143 UTC: IP: s=10.3.2.31 (local), d=10.1.2.254 (Vlan2), len 76, sending full packet
.Apr 20 00:16:34.143 UTC: NTP: xmit packet to 10.1.2.254:
.Apr 20 00:16:34.143 UTC:  leap 3, mode 3, version 3, stratum 0, ppoll 64
.Apr 20 00:16:34.143 UTC:  rtde1 0021 (0.504), rtdsp 1105E7 (17023.056), refid 0A0102FE (10.1.2.254)
.Apr 20 00:16:34.143 UTC:  ref D33B2922.24FEBDC7 (00:15:30.144 UTC Fri Apr 20 2012)
.Apr 20 00:16:34.143 UTC:  org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
.Apr 20 00:16:34.143 UTC:  rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
.Apr 20 00:16:34.143 UTC:  xmt D33B2962.24CAFAD1 (00:16:34.143 UTC Fri Apr 20 2012)
.Apr 20 00:16:34.143 UTC: IP: s=10.1.2.254 (Vlan2), d=10.3.2.31, len 76, rcvd 2
.Apr 20 00:16:34.143 UTC: NTP: rcv packet from 10.1.2.254 to 10.3.2.31 on Vlan2:
.Apr 20 00:16:34.143 UTC:  leap 0, mode 4, version 3, stratum 1, ppoll 64
.Apr 20 00:16:34.143 UTC:  rtde1 0000 (0.000), rtdsp 009D (2.396), refid 47505300 (10.80.83.0)
.Apr 20 00:16:34.143 UTC:  ref D33B2952.4CC11CCF (00:16:18.299 UTC Fri Apr 20 2012)
.Apr 20 00:16:34.143 UTC:  org D33B2962.24CAFAD1 (00:16:34.143 UTC Fri Apr 20 2012)
.Apr 20 00:16:34.143 UTC:  rec D33B2962.49D3724D (00:16:34.288 UTC Fri Apr 20 2012)
```



```

.Apr 20 00:16:34.143 UTC: xmt D33B2962.49D997D0 (00:16:34.288 UTC Fri Apr 20 2012)
.Apr 20 00:16:34.143 UTC: inp D33B2962.25010310 (00:16:34.144 UTC Fri Apr 20 2012)
.Apr 20 00:16:36.283 UTC: IP: tableid=0, s=10.3.2.31 (local), d=10.8.2.254 (Vlan2), routed via FIB
.Apr 20 00:16:36.283 UTC: IP: s=10.3.2.31 (local), d=10.8.2.254 (Vlan2), len 76, sending
.Apr 20 00:16:36.283 UTC: IP: s=10.3.2.31 (local), d=10.8.2.254 (Vlan2), len 76, sending full packet
.Apr 20 00:16:36.283 UTC: NTP: xmit packet to 10.8.2.254:
.Apr 20 00:16:36.283 UTC: leap 3, mode 3, version 3, stratum 0, ppoll 64
.Apr 20 00:16:36.283 UTC: rtde1 002F (0.717), rtdsp 11058F (17021.713), refid 0A0102FE (10.1.2.254)
.Apr 20 00:16:36.283 UTC: ref D33B2962.25010310 (00:16:34.144 UTC Fri Apr 20 2012)
.Apr 20 00:16:36.283 UTC: org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
.Apr 20 00:16:36.283 UTC: rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
.Apr 20 00:16:36.283 UTC: xmt D33B2964.48947E87 (00:16:36.283 UTC Fri Apr 20 2012)
.Apr 20 00:16:36.283 UTC: IP: s=10.8.2.254 (Vlan2), d=10.3.2.31, len 76, rcvd 2
.Apr 20 00:16:36.283 UTC: NTP: rcv packet from 10.8.2.254 to 10.3.2.31 on Vlan2:
.Apr 20 00:16:36.283 UTC: leap 0, mode 4, version 3, stratum 1, ppoll 64
.Apr 20 00:16:36.283 UTC: rtde1 0000 (0.000), rtdsp 0017 (0.351), refid 47505300 (10.80.83.0)
.Apr 20 00:16:36.283 UTC: ref D33B295B.8AF7FE33 (00:16:27.542 UTC Fri Apr 20 2012)
.Apr 20 00:16:36.283 UTC: org D33B2964.48947E87 (00:16:36.283 UTC Fri Apr 20 2012)
.Apr 20 00:16:36.283 UTC: rec D33B2964.4A6AD269 (00:16:36.290 UTC Fri Apr 20 2012)
.Apr 20 00:16:36.283 UTC: xmt D33B2964.4A7C00D0 (00:16:36.290 UTC Fri Apr 20 2012)
.Apr 20 00:16:36.283 UTC: inp D33B2964.498A755D (00:16:36.287 UTC Fri Apr 20 2012)

```

这是NTP对收到的数据包不起作用的示例。虽然接收到NTP数据包（如debug ip packets所示），但NTP进程不对它们执行操作。对于发出的NTP数据包，存在相应的debug ntp packets输出，因为NTP进程必须生成数据包。此问题特定于收到的未处理的NTP数据包。

```

071564: Apr 23 2012 15:46:26.100 ETE: NTP: xmit packet to 10.50.44.101:
071565: Apr 23 2012 15:46:26.100 ETE: leap 0, mode 1, version 3, stratum 5, ppoll 1024
071566: Apr 23 2012 15:46:26.100 ETE: rtde1 07B5 (30.106), rtdsp 0855 (32.547), refid 0A32266A
(10.50.38.106)
071567: Apr 23 2012 15:46:26.100 ETE: ref D33FDB05.1A084831 (15:43:33.101 ETE Mon Apr 23 2012)
071568: Apr 23 2012 15:46:26.100 ETE: org 00000000.00000000 (01:00:00.000 HIVER Mon Jan 1 1900)
071569: Apr 23 2012 15:46:26.100 ETE: rec 00000000.00000000 (01:00:00.000 HIVER Mon Jan 1 1900)
071570: Apr 23 2012 15:46:26.100 ETE: xmt D33FDBB2.19D3457C (15:46:26.100 ETE Mon Apr 23 2012)
PCY_PAS1#
071571: Apr 23 2012 15:47:31.497 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, input featur
071572: Apr 23 2012 15:47:31.497 ETE: UDP src=123, dst=123, Ingress-NetFlow(13), rtype 0, forus FALSE,
sendself FALSE, mtu 0
071573: Apr 23 2012 15:47:31.497 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, input featur
071574: Apr 23 2012 15:47:31.497 ETE: UDP src=123, dst=123, MCI Check(55), rtype 0, forus FALSE,
sendself FALSE, mtu 0
071575: Apr 23 2012 15:47:31.497 ETE: FIBipv4-packet-proc: route packet from Tunnel99 src 10.50.38.78 d
10.50.44.69
071576: Apr 23 2012 15:47:31.497 ETE: FIBfwd-proc: base:10.50.44.69/32 receive entry
PCY_PAS1#
071577: Apr 23 2012 15:47:31.497 ETE: FIBipv4-packet-proc: packet routing failed
071578: Apr 23 2012 15:47:31.497 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, rcvd 2
071579: Apr 23 2012 15:47:31.497 ETE: UDP src=123, dst=123
071580: Apr 23 2012 15:47:31.497 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, stop process
for forus packet
071581: Apr 23 2012 15:47:31.497 ETE: UDP src=123, dst=123
PCY_PAS1#
071582: Apr 23 2012 16:03:30.105 ETE: NTP: xmit packet to 10.50.44.101:
071583: Apr 23 2012 16:03:30.105 ETE: leap 0, mode 1, version 3, stratum 5, ppoll 1024
071584: Apr 23 2012 16:03:30.105 ETE: rtde1 0759 (28.702), rtdsp 087D (33.157), refid 0A32266A
(10.50.38.106)
071585: Apr 23 2012 16:03:30.105 ETE: ref D33FDF05.1B2CC3D4 (16:00:37.106 ETE Mon Apr 23 2012)
071586: Apr 23 2012 16:03:30.105 ETE: org 00000000.00000000 (01:00:00.000 HIVER Mon Jan 1 1900)

```

```
071587: Apr 23 2012 16:03:30.105 ETE: rec 00000000.00000000 (01:00:00.000 HIVER Mon Jan 1 1900)
071588: Apr 23 2012 16:03:30.105 ETE: xmt D33FDFB2.1B1D5E7E (16:03:30.105 ETE Mon Apr 23 2012)
PCY_PAS1#
071589: Apr 23 2012 16:04:35.502 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, input featur
071590: Apr 23 2012 16:04:35.506 ETE: UDP src=123, dst=123, Ingress-NetFlow(13), rtype 0, forus FAL
sendself FALSE, mtu 0
071591: Apr 23 2012 16:04:35.506 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, input featur
071592: Apr 23 2012 16:04:35.506 ETE: UDP src=123, dst=123, MCI Check(55), rtype 0, forus FALSE,
sendself FALSE, mtu 0
071593: Apr 23 2012 16:04:35.506 ETE: FIBipv4-packet-proc: route packet from Tunnel99 src 10.50.38.78 d
10.50.44.69
071594: Apr 23 2012 16:04:35.506 ETE: FIBfwd-proc: base:10.50.44.69/32 receive entry
PCY_PAS1#
071595: Apr 23 2012 16:04:35.506 ETE: FIBipv4-packet-proc: packet routing failed
071596: Apr 23 2012 16:04:35.506 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, rcvd 2
071597: Apr 23 2012 16:04:35.506 ETE: UDP src=123, dst=123
071598: Apr 23 2012 16:04:35.506 ETE: IP: s=10.50.38.78 (Tunnel99), d=10.50.44.69, len 76, stop process
for forus packet
071599: Apr 23 2012 16:04:35.506 ETE: UDP src=123, dst=123
PCY_PAS1#
```

同步丢失

如果服务器的色散和/或延迟值非常高，则可能会发生同步丢失。高值表示参照时钟的根目录，数据包从服务器/对等设备到达客户端所用的时间太长。因此，本地计算机无法信任数据包中当前时间的准确性，因为它不知道数据包到达此处需要多长时间。

NTP对时间非常精细，无法与它无法信任的其他设备同步，或无法以可信的方式进行调整。

如果存在饱和链路，并且沿途发生缓冲，数据包在到达NTP客户端时会被延迟。因此，后续NTP数据包中包含的时间戳有时可能变化很大，而本地客户端无法真正针对此变化进行调整。

除非使用SNTP（简单网络时间协议），否则NTP不提供关闭这些数据包验证的方法。SNTP不是替代方案，因为软件中并不广泛支持它。

如果遇到同步丢失的情况，必须检查以下链接：

- 它们饱和了吗？
- 您的广域网(WAN)链路中是否存在任何类型的丢包

- 加密是否发生？

监控show ntp associations detail命令的到达值。最大值为377。如果该值为0或低，则间歇性接收NTP数据包，并且本地客户端与服务
器不同步。

debug ntp validity

debug ntp validity命令指示NTP数据包是否通过健全性检查或有效性检查，并揭示失败原因。将此输出与RFC1305中指定的健全性测试
进行比较，这些测试用于测试从服务器接收的NTP数据包。定义了八个测试：

测试 掩码		说明
1	0x01	收到重复的数据包
2	0x02	收到的伪造数据包
3	0x04	协议不同步
4	0x08	对等体延迟/色散未通过边界检查
5	0x10	对等身份验证失败
6	0x20	对等体时钟不同步 (对于未同步的服务器通用)
7	0x40	对等层超出范围

8	0x80	根延迟/色散边界检查失败
---	------	--------------

以下是debug ntp validity命令的示例输出：

```
PCY_PAS1#debug ntp validity
NTP peer validity debugging is on
```

```
009585: Mar 1 2012 09:14:32.670 HIVER: NTP: packet from 192.168.113.57 failed validity tests 52
009586: Mar 1 2012 09:14:32.670 HIVER: Authentication failed
009587: Mar 1 2012 09:14:32.670 HIVER: Peer/Server Stratum out of bound
PCY_PAS1#
009588: Mar 1 2012 09:14:38.210 HIVER: NTP: packet from 192.168.56.1 failed validity tests 14
009589: Mar 1 2012 09:14:38.210 HIVER: Authentication failed
PCY_PAS1#
009590: Mar 1 2012 09:14:43.606 HIVER: NTP: packet from 10.110.103.27 failed validity tests 14
009591: Mar 1 2012 09:14:43.606 HIVER: Authentication failed
PCY_PAS1#
009592: Mar 1 2012 09:14:48.686 HIVER: NTP: packet from 192.168.113.57failed validity tests 52
009593: Mar 1 2012 09:14:48.686 HIVER: Authentication failed
009594: Mar 1 2012 09:14:48.686 HIVER: Peer/Server Stratum out of bound
PCY_PAS1#
009596: Mar 1 2012 09:14:54.222 HIVER: NTP: packet from 10.110.103.35 failed validity tests 14
009597: Mar 1 2012 09:14:54.222 HIVER: Authentication failed
PCY_PAS1#
009598: Mar 1 2012 09:14:54.886 HIVER: NTP: synced to new peer 10.50.38.106
009599: Mar 1 2012 09:14:54.886 HIVER: NTP: 10.50.38.106 synced to new peer
PCY_PAS1#
009600: Mar 1 2012 09:14:59.606 HIVER: NTP: packet from 10.110.103.27 failed validity tests 14
009601: Mar 1 2012 09:14:59.606 HIVER: Authentication failed
PCY_PAS1#
009602: Mar 1 2012 09:15:04.622 HIVER: NTP: packet from 192.168.113.137 failed validity tests 52
009603: Mar 1 2012 09:15:04.622 HIVER: Authentication failed
009604: Mar 1 2012 09:15:04.622 HIVER: Peer/Server Stratum out of bound
PCY_PAS1#
009605: Mar 1 2012 09:15:10.238 HIVER: NTP: packet from 192.168.56.1 failed validity tests 14
009606: Mar 1 2012 09:15:10.238 HIVER: Authentication failed
PCY_PAS1#
009607: Mar 1 2012 09:15:15.338 HIVER: NTP: packet from 10.83.23.140 failed validity tests 52
009608: Mar 1 2012 09:15:15.338 HIVER: Authentication failed
009609: Mar 1 2012 09:15:15.338 HIVER: Peer/Server Stratum out of bound
PCY_PAS1#
009610: Mar 1 2012 09:15:20.402 HIVER: NTP: packet from 192.168.113.92 failed validity tests 74
009611: Mar 1 2012 09:15:20.402 HIVER: Authentication failed
009612: Mar 1 2012 09:15:20.402 HIVER: Peer/Server Clock unsynchronized
009613: Mar 1 2012 09:15:20.402 HIVER: Peer/Server Stratum out of bound
```

debug ntp packets

您可以使用debug ntp packets命令来查看对等体/服务器在收到的数据包中向您提供的时间。时间本地机器还告知它知道所传输数据包中对等体/服务器的时间。

字段	rcv数据包	xmit数据包
组织	发起方时间戳，是服务器时间。	发送数据包时的发送方（客户端）时间戳。（客户端向服务器发送数据包。）
rec	客户端收到数据包时的时间戳。	客户端当前时间。

在此示例输出中，从服务器接收的数据包和发送到另一服务器的数据包的时间戳相同，这表示客户端NTP处于同步状态。

```

USSP-B33S-SW01#debug ntp packets
NTP packets debugging is on
USSP-B33S-SW01#
May 25 02:21:48.182 UTC: NTP: rcv packet from 10.1.2.254 to 10.3.2.31 on Vlan2:
May 25 02:21:48.182 UTC: leap 0, mode 4, version 3, stratum 1, poll 64
May 25 02:21:48.182 UTC: rtde1 0000 (0.000), rtdsp 00F2 (3.693), refid 47505300 (10.80.83.0)
May 25 02:21:48.182 UTC: ref D3696B38.B722C417 (02:21:44.715 UTC Fri May 25 2012)
May 25 02:21:48.182 UTC: org D3696B3C.2EA179BA (02:21:48.182 UTC Fri May 25 2012)
May 25 02:21:48.182 UTC: rec D3696B3D.E58DE1BE (02:21:49.896 UTC Fri May 25 2012)
May 25 02:21:48.182 UTC: xmt D3696B3D.E594E7AF (02:21:49.896 UTC Fri May 25 2012)
May 25 02:21:48.182 UTC: inp D3696B3C.2EDFC333 (02:21:48.183 UTC Fri May 25 2012)
May 25 02:22:46.051 UTC: NTP: xmit packet to 10.4.2.254:
May 25 02:22:46.051 UTC: leap 0, mode 3, version 3, stratum 2, poll 64
May 25 02:22:46.051 UTC: rtde1 00C0 (2.930), rtdsp 1C6FA (1777.252), refid 0A0402FE (10.4.2.254)
May 25 02:22:46.051 UTC: ref D3696B36.33D43F44 (02:21:42.202 UTC Fri May 25 2012)
May 25 02:22:46.051 UTC: org D3696B37.E72C75AE (02:21:43.903 UTC Fri May 25 2012)
May 25 02:22:46.051 UTC: rec D3696B36.33D43F44 (02:21:42.202 UTC Fri May 25 2012)
May 25 02:22:46.051 UTC: xmt D3696B76.0D43AE7D (02:22:46.051 UTC Fri May 25 2012)

```

这是时钟不同步时的输出示例。注意xmit数据包和rcv数据包之间的时间差。对等体色散可以处于最大值16000，对等体的到达范围可以显示0。

```
USSP-B33S-SW01#
.May 25 02:05:59.011 UTC: NTP: xmit packet to 10.4.2.254:
.May 25 02:05:59.011 UTC: leap 3, mode 3, version 3, stratum 0, ppoll 64
.May 25 02:05:59.011 UTC: rtde1 00A3 (2.487), rtdsp 1104D0 (17018.799), refid 0A0402FE (10.4.2.254)
.May 25 02:05:59.011 UTC: ref D3696747.03D8661A (02:04:55.015 UTC Fri May 25 2012)
.May 25 02:05:59.011 UTC: org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
.May 25 02:05:59.011 UTC: rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
.May 25 02:05:59.011 UTC: xmt D3696787.03105783 (02:05:59.011 UTC Fri May 25 2012)
.May 25 02:05:59.011 UTC: NTP: rcv packet from 10.4.2.254 to 10.3.2.31 on Vlan2:
.May 25 02:05:59.011 UTC: leap 0, mode 4, version 3, stratum 1, ppoll 64
.May 25 02:05:59.011 UTC: rtde1 0000 (0.000), rtdsp 0014 (0.305), refid 47505300 (10.80.83.0)
.May 25 02:05:59.011 UTC: ref D3696782.C96FD778 (02:05:54.786 UTC Fri May 25 2012)
.May 25 02:05:59.011 UTC: org D3696787.03105783 (02:05:59.011 UTC Fri May 25 2012)
.May 25 02:05:59.011 UTC: rec D3696787.281A963F (02:05:59.156 UTC Fri May 25 2012)
.May 25 02:05:59.011 UTC: xmt D3696787.282832C4 (02:05:59.156 UTC Fri May 25 2012)
.May 25 02:05:59.011 UTC: inp D3696787.03C63542 (02:05:59.014 UTC Fri May 25 2012)
```

debug ntp sync and debug ntp events

debug ntp sync命令生成一行输出，显示时钟是否已同步或同步是否已更改。该命令通常通过debug ntp events启用。

debug ntp events命令可显示发生的所有NTP事件，这有助于您确定NTP中的更改是否触发了问题，例如时钟不同步。（换句话说，如果快乐同步的时钟突然变得疯狂，你就知道要寻找变化或触发因素！）

这是两个调试的示例。最初，客户端时钟已同步。debug ntp events命令显示NTP对等体层级发生更改，然后时钟不同步。

```
USSP-B33S-SW01#debug ntp sync
NTP clock synchronization debugging is on
USSP-B33S-SW01#
USSP-B33S-SW01#
USSP-B33S-SW01#debug ntp events
NTP events debugging is on
USSP-B33S-SW01#
USSP-B33S-SW01#
May 25 02:25:57.620 UTC: NTP: xmit packet to 10.4.2.254:
May 25 02:25:57.620 UTC: leap 0, mode 3, version 3, stratum 2, ppoll 64
May 25 02:25:57.620 UTC: rtde1 00D4 (3.235), rtdsp 26B26 (2418.549), refid 0A0402FE (10.4.2.254)
May 25 02:25:57.620 UTC: ref D3696BF5.C47EB880 (02:24:53.767 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: org D3696BF7.E5F91077 (02:24:55.898 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: rec D3696BF5.C47EB880 (02:24:53.767 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: xmt D3696C35.9ED1CE97 (02:25:57.620 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: NTP: rcv packet from 10.4.2.254 to 10.3.2.31 on Vlan2:
May 25 02:25:57.620 UTC: leap 0, mode 4, version 3, stratum 1, ppoll 64
```

```
May 25 02:25:57.620 UTC: rtde1 0000 (0.000), rtdsp 000E (0.214), refid 47505300 (10.80.83.0)
May 25 02:25:57.620 UTC: ref D3696C37.D528800E (02:25:59.832 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: org D3696C35.9ED1CE97 (02:25:57.620 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: rec D3696C37.E5C7AB3D (02:25:59.897 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: xmt D3696C37.E5D1F273 (02:25:59.897 UTC Fri May 25 2012)
May 25 02:25:57.620 UTC: inp D3696C35.9F9EA2C4 (02:25:57.623 UTC Fri May 25 2012)
May 25 02:25:59.830 UTC: NTP: peer stratum change
May 25 02:25:59.830 UTC: NTP: clock reset
May 25 02:25:59.830 UTC: NTP: sync change
May 25 02:25:59.830 UTC: NTP: peer stratum change
May 25 02:26:05.817 UTC: NTP: xmit packet to 10.1.2.254:
May 25 02:26:05.817 UTC: leap 3, mode 3, version 3, stratum 0, ppoll 64
May 25 02:26:05.817 UTC: rtde1 00C2 (2.960), rtdsp 38E9C (3557.068), refid 0A0402FE (10.4.2.254)
May 25 02:26:05.817 UTC: ref D3696C35.9F9EA2C4 (02:25:57.623 UTC Fri May 25 2012)
May 25 02:26:05.817 UTC: org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
May 25 02:26:05.817 UTC: rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
May 25 02:26:05.817 UTC: xmt D3696C3D.D12D0565 (02:26:05.817 UTC Fri May 25 2012)
```

手动设置NTP时钟周期

Cisco.com网站警告说：

“当输入copy running-configuration startup-configuration命令以将配置保存到NVRAM时，ntp clock-period命令会自动生成，以反映不断更改的更正系数。请勿尝试手动使用ntp clock-period命令。请确保在将配置文件复制到其他设备时删除此命令行。”

时钟周期值取决于硬件，因此它因设备而异。

启用NTP时，ntp clock-period命令会自动出现在配置中。命令用于调整软件时钟。“调整值”用于补偿4毫秒的勾选间隔，因此，在次要调整中，您在间隔结束时会有1秒。

如果设备计算其系统时钟丢失时间（可能需从路由器的基本级别进行频率补偿），它会将该值自动添加到系统时钟中，以保持其同步性。

注意：用户不能更改此命令。

路由器的默认NTP时钟周期为17179869，实际上用于启动NTP进程。

转换公式为 $17179869 * 2^{(-32)} = 0.00399999995715916156768798828125$ ，或大约4毫秒。

例如，发现Cisco 2611路由器（Cisco 2600系列路由器之一）的系统时钟稍有不同步，可以使用以下命令重新同步：

```
ntp clock-period 17208078
```

这等于 $17208078 * 2^{(-32)} = 0.0040065678767859935760498046875$ ，或略高于4毫秒。

Cisco建议您在正常网络条件下允许路由器运行一周左右，然后使用wr mem命令来保存值。这为您提供了下次重新启动的准确数字，并允许NTP更快速地同步。

在保存配置以供另一台设备使用时，请使用no ntp clock-period命令，因为此命令会将时钟周期恢复为该特定设备的默认值。您可以重新计算实际值（但在重新计算期间可以降低系统时钟的精度）。

请记住，此值取决于硬件，因此，如果您复制配置并将其用于不同的设备，则可能会引起问题。为解决此问题，思科计划将NTP第3版替换为第4版。

如果您不知道这些问题，则可以决定手动修改此值。为了从一台设备迁移到另一台设备，您可以决定复制旧配置并将其粘贴到新设备上。很遗憾，因为ntp clock-period命令出现在running-config和startup-config中，所以NTP clock-period粘贴在新设备上。发生这种情况时，新客户端上的NTP总是与对等体色散值较高的服务器不同步。

相反，应使用no ntp clock-period命令清除NTP clock-period，然后保存配置。路由器最终会计算出适合自身的时钟周期。

Cisco IOS软件版本15.0或更高版本中不再提供ntp clock-period命令；解析器现在拒绝该命令，并显示以下错误：

```
"%NTP: This configuration command is deprecated."
```

不允许手动配置时钟周期，并且运行配置中不允许使用时钟周期。由于如果命令在启动配置中（在较早的Cisco IOS版本，如12.4），解析器会拒绝该命令，因此在启动时将启动配置复制到运行配置时，解析器会拒绝该命令。

新的替换命令是ntp clear drift。

相关信息

- [支持论坛主题：未配置NTP时钟周期](#)
- [网络时间协议：最佳实践白皮书](#)
- [排除网络时间协议\(NTP\)故障](#)
- [思科技术支持和下载](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。