

引用格式：缪新育, 胡昌军, 李曙方, 等. NTP 服务器压力测试方法的设计与实现[J]. 时间频率学报, 2021, 44(3): 197-203.

# NTP 服务器压力测试方法的设计与实现

缪新育<sup>1,2</sup>, 胡昌军<sup>2</sup>, 李曙方<sup>2</sup>, 吕博<sup>2</sup>

(1. 北京邮电大学, 北京 100876 ;  
2. 中国信息通信研究院, 北京 100191)

**摘要:** 针对网络时间协议 (network time protocol, NTP) 服务器的压力测试, 传统的基于软件模拟客户端报文的方法存在不能达到最大理论报文发送速率的不足, 本文采用硬件处理方式, 设计了一个可模拟多客户端多端口的高速率 NTP 压力测试方法。该方法不受软件处理延时的影响, 可实现最大的理论报文发送速率, 满足高速 NTP 报文请求的需求。最后通过实际测试对该方法进行验证, 实验结果表明该方法可满足多客户端多端口任意速率场景的 NTP 压力测试需求。

**关键词:** 网络时间协议; 压力测试; 报文构建; 报文延时

DOI: 10.13875/j.issn.1674-0637.2021-03-0197-07

## Design and implementation of NTP server stress test method

MIAO Xin-yu<sup>1,2</sup>, HU Chang-jun<sup>2</sup>, LI Shu-fang<sup>2</sup>, LYU Bo<sup>2</sup>

(1. Beijing University of Post and Telecommunications, Beijing 100876, China;  
2. China Academy of Information and Communications Technology, Beijing 100191, China)

**Abstract:** Aiming for the stress test of the network time protocol (NTP) server, the method of software simulation is used to generate client packet traditionally, but this method cannot reach the theoretical maximum rate of packet transmission. A high-rate NTP stress test method is demonstrated in this paper, which can simulate multi-client and multi-port scenarios. This method uses the way of hardware processing. It is not affected by the software processing delay, and can achieve the theoretical maximum packet transmission rate to meet the requirement of the high-speed NTP request packet. This method is verified and analyzed through actual tests at last. The experimental results show that this method can simulate multi-client, multi-port, and arbitrary packet rate scenario and it can meet the requirements of NTP pressure test properly.

**Key words:** network time protocol (NTP); pressure test; packet construction; packet delay

## 0 引言

NTP 协议是目前 Internet 上最常用的时间同步方式<sup>[1]</sup>, 它可以将计算机的时间同步到某些时间标准, 适应各种规模、速度和连接通路状况的互联网环境, 在互联网中得到了广泛的应用。然而, 随着 NTP 服

务需求的增加<sup>[2]</sup>,对 NTP 服务器的性能提出了更高的要求,也需要对 NTP 服务器响应能力进行合理的评价。本文主要针对 NTP 服务器响应能力设计了一个测试方法,以满足各种场景的 NTP 服务器响应能力测试需求。

NTP 服务器的响应能力是指一台服务器在单位时间里对客户端时间请求的处理能力,处理的请求越多,服务器的并发响应能力越强。响应能力一般取决于服务器处理器对缓存数据的处理能力<sup>[3]</sup>,如果请求量超过处理器的处理能力,则会出现请求拒绝现象,或宕机现象,将大大降低服务器的服务功能。因此有必要对服务器的处理能力进行评估。

目前已有的测试方法主要有基于 CPU 处理架构的单端口软件模拟方法<sup>[4-6]</sup>,该类方法通常采用计算机语言编写,运行于 Linux 等平台上。软件通过多线程方式在某一时间段内对目标 NTP 服务器发出若干个 NTP 请求,并接收其返回的数据报文,通过返回数据报文形式判断通信是否成功,以此方式探测目标服务器的抗压能力,结束后返回在设定时间内产生的链接数与响应数,实现对目标服务器响应能力的评估。由于软件处理过程相比硬件处理方式多了几道工序,同时软件是基于串行执行指令,而硬件可实现并行操作,因此软件模拟方式不能达到最大的理论报文发送速率。

本文提出一种基于 FPGA 硬件处理的报文发送方法,直接通过硬件方式进行报文编辑来模拟客户端报文发送,由于硬件自身可编程,数据处理不需要经过 CPU 指令调度等步骤,直接在硬件电路上就实现了,因此可以达到最大理论报文发送速率。主要流程为在短时间内创建并向服务器发送大量 NTP 请求报文,同时检测服务器的响应报文,根据发送报文数量与响应报文数量定量地确定服务器的性能指标,实现对 NTP 服务器单位时间内响应阈值的测试与评估。

## 1 NTP 协议介绍

NTP 协议是美国德拉瓦大学的 D.L.Mills 教授于 1985 年提出的<sup>[7]</sup>。其基本工作原理如图 1 所示,Slave Clock 和 Master Clock 通过网络相连,它们都有自己独立的系统时钟,通过 NTP 实现主从时间同步<sup>[8-9]</sup>。具体流程如下所示:

Slave Clock 发送一个 NTP 请求报文给 Master Clock,该报文带有它离开 Slave Clock 时的时间  $T_1$ ;

当请求报文到达 Master Clock 时,Master Clock 记录报文到达的时间,该时间为  $T_2$ ;

Master Clock 收到 NTP 请求报文后,返回 NTP 响应报文,并记录 NTP 响应报文离开 Master Clock 的时间  $T_3$ ,报文里携带了  $T_1, T_2, T_3$  时间戳;

当 Slave Clock 接收到该响应报文时,记录报文到达的时间  $T_4$ ;

至此,Slave Clock 已经拥有足够的信息来计算主从时钟的时间偏差和链路时延。

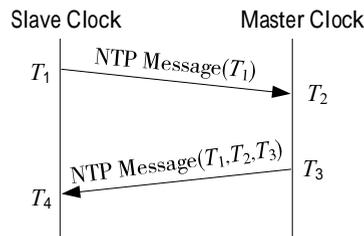


图 1 NTP 传输模型

假设 Slave Clock 与 Master Clock 之间时差值为  $\Delta T$ ,Slave Clock 与 Master Clock 之间的网络传输延时为  $L$  (假设网络来回时延一致),则有:

$$\begin{cases} T_2 = T_1 + \Delta T + L \\ T_4 = T_3 - \Delta T + L \end{cases} \quad (1)$$

由式(1)可得：

$$\begin{cases} \Delta T = [(T_2 - T_1) - (T_4 - T_3)]/2 \\ L = [(T_2 - T_1) + (T_4 - T_3)]/2 \end{cases} \quad (2)$$

通过式(2), Slave Clock 即可计算得主从时钟的时间偏差和链路时延。

NTP 的报文格式如图 2 所示, 主要字段解释参见参考文献[8]和[9]。

0	1	4	7	15	23	31
LI	VN	Mode	Stratum	Poll	Precision	
Root delay (32 bits)						
Root dispersion (32 bits)						
Reference identifier (32 bits)						
Reference timestamp (64 bits)						
Originate timestamp (64 bits)						
Receiver timestamp (64 bits)						
Transmit timestamp (64 bits)						
Authenticator (option 96 bits)						

图 2 NTP 报文格式

## 2 NTP 压力测试设计方案

NTP 协议采用请求/响应模型定义了客户端如何从服务器请求时间信息, 以及服务器如何把时间信息传送给客户端。客户端向服务器发送的请求报文包含了报文封装方式、以太帧、IP 协议、UDP 协议、NTP 协议版本、请求数据等信息。服务器以同样格式报文作为响应, 响应的内容包括除了通用的格式外, 还包含服务器的时间信息。

以下是 NTP 请求/响应报文的处理步骤：

客户端连接到 NTP 服务器

NTP 客户端通常需要与服务器的 IP 端口建立地址解析协议 ARP( address resolution protocol )连接<sup>[10]</sup>。

客户端发送 NTP 请求报文

通过报文构建和处理器处理, 客户端向服务器发送一个请求报文, 请求报文由通用帧格式、客户端标识、客户端时间信息等部分组成。

服务器接受请求并返回 NTP 响应报文

服务器解析请求, 定位请求资源, 并将资源复本写到报文响应字段。响应报文由通用帧格式、服务器标识、服务器响应时间等部分组成。

客户端接收 NTP 响应报文并解析报文内容

客户端处理器首先解析响应报文, 查看报文中的时间信息。然后结合服务器响应报文的接收时间, 根据 NTP 主从时间差原理, 计算出主从时间差。

### 3 压力测试详细设计

#### 3.1 设计方案

为了构造 NTP 压力测试控制器，我们采用以太网分析仪 TestCenter 来产生 NTP 请求报文<sup>[11]</sup>。TestCenter 是一种可以在各种现实情况下测量路由器或交换机等设备性能的仪表，具有状态层 2-7 流量生成和分析，及以太网协议和网络的验证等功能。TestCenter 采用 FPGA 处理架构，设计了通用报文处理器，可直接设置报文 payload 为需要的内容，具有较快的报文发送速度。

设计步骤如下：

使用抓包工具 wireshark 抓取已有的 NTP 报文，分析报文的字段结构，去除以太报文头，提取 NTP 协议内容。

使用 TestCenter 的报文构造功能进行 NTP 模拟报文格式构建，将 NTP 协议内容复制进报文内容中，形成完整 NTP 报文。

利用 TestCenter 的端口设置和报文编辑功能，配置报文的速率、IP 地址等参数，及多端口输出。

将构造好的报文发送给指定的被测 NTP 服务器，同时接收服务器返回的 NTP 响应报文，识别并统计收到的响应报文数量，计算服务器的响应能力。

最终可模拟实现多客户端多端口任意速率场景的 NTP 服务器压力测试。

#### 3.2 设计步骤

通过 wireshark 抓取 NTP 报文，如图 3 所示。

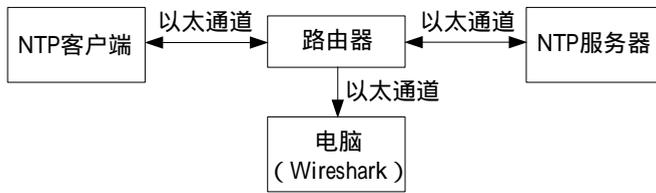


图 3 NTP 连接拓扑

获得的报文如图 4 所示。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.173.1.131	192.173.1.10	NTP	90	NTP Version 3, client

```

Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
Ethernet II, Src: HonHaiPr_65:0c:1c (30:0e:d5:65:0c:1c), Dst: Zhejiang_0e:77 (78:ca:83:b0:0e:77)
Internet Protocol Version 4, Src: 192.173.1.131, Dst: 192.173.1.10
User Datagram Protocol, Src Port: 61427, Dst Port: 123
Network Time Protocol (NTP Version 3, client)
  
```

图 4 NTP 报文解析

打开 TestCenter 软件的报文构造功能，设置物理层参数，包括传输媒介类型：铜缆/光缆，传输速率、全/半双工。创建新的原始数据流，根据报文帧结构，设置相应的协议字段，把抓取的 NTP 报文的相应字段填入以上创建的数据流字段内，将 NTP 协议内容复制进报文内容字段中，如图 5 所示，即可生成完整的 NTP 报文。

Name	Value
Frame	
Ethernet II	
IPv4 Header	
UDP Header	
Custom Header	

图 5 NTP 报文结构构造

假设报文长度为  $A$ ，前导码长度为  $B$ ，帧间隔为  $C$ ，报文速率为  $n$ ，端口带宽为  $D$ ，则这几个变量满足以下关系：

$$(A+B+C) \times n = D, \quad (3)$$

通过式 (3)，可计算得以太报文的理论速率  $n$  为

$$n = \frac{D}{A+B+C}。 \quad (4)$$

最后设置报文的不同 IP 地址循环，可模拟不同 IP 的客户端，通过在其他端口设置同样的报文流，可以对 NTP 服务器的不同端口同时进行压力测试。

## 4 实验验证

### 4.1 自环验证

为了验证模拟 NTP 请求报文的发送性能，我们对 TestCenter 报文进行了自环验证，即 TestCenter 的 Tx/Rx1 口发送 NTP 请求报文，由 Tx/Rx2 口进行接收测量，测试拓扑如图 6 所示。

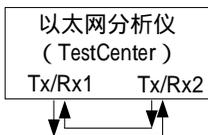


图 6 TestCenter 自环测试拓扑图

测试过程如下：收发端口均选择 100 Mb/s 速率以太模块，发送端口负载为 10%，则发送端口的最大传输速率为 10 Mb/s。发送口发出的报文由接收口进行测量，最后读取并比较发送端显示的帧发送率和接收端的帧接收率。结果表明两者数值相同，且均为 10 965 帧/s，同时通过接收报文解析，确认收到的报文均为 NTP 请求报文，如图 7 所示。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.169	192.168.1.10	NTP	94	NTP Version 3, client
2	0.000091200	192.168.1.170	192.168.1.10	NTP	94	NTP Version 3, client
3	0.000182400	192.168.1.171	192.168.1.10	NTP	94	NTP Version 3, client
4	0.000273600	192.168.1.172	192.168.1.10	NTP	94	NTP Version 3, client
5	0.000364800	192.168.1.173	192.168.1.10	NTP	94	NTP Version 3, client
6	0.000456000	192.168.1.174	192.168.1.10	NTP	94	NTP Version 3, client
7	0.000547200	192.168.1.175	192.168.1.10	NTP	94	NTP Version 3, client

```

Frame 15: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
Ethernet II, Src: ArcComCon_08:ac:10 (00:80:66:08:ac:10), Dst: Advantec_b7:0b:c0 (00:0b:ab:b7:0b:c0)
Internet Protocol Version 4, Src: 192.168.1.173, Dst: 192.168.1.10
User Datagram Protocol, Src Port: 123, Dst Port: 123
Network Time Protocol (NTP Version 3, client)

```

图 7 TestCenter 自环接收到的 NTP 报文

由 NTP 报文结构可知，报文长度为 94 Byte，加上以太帧的前导码为 8 Byte，最小帧间隔为 12 Byte，通过计算，得到端口速率为 10 000 080 bit/s，与设置的标称速率 10 Mb/s 一致，说明该设计方法模拟的 NTP 请求报文达到了理论的最大速率。

### 4.2 单端口实测验证

为了进一步验证该方法的实际测试效果，我们用 TestCenter 对某 NTP 服务器的响应能力进行了实际测试，该服务器是基于单 CPU 处理架构设计的，通过软件模拟方式进行报文响应，测试方案如下。TestCenter 通过 Tx/Rx 口发送 NTP 请求报文至被测 NTP 服务器的 Rx/Tx 口，NTP 服务器接收到 NTP 请求报文后，由原端口将 NTP 响应报文发送回 TestCenter，TestCenter 进行报文检测和统计，测试拓扑如图 8 所示。

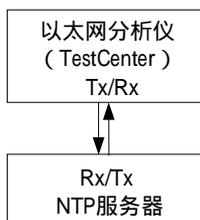


图 8 NTP 服务器响应能力测试拓扑（单端口）

测试过程如下：选择 100 Mb/s 速率以太模块，端口负载为 10%，也就是最大传输速率为 10 Mb/s。TestCenter 先与 NTP 服务器进行 ARP 握手，然后发送 NTP 请求报文。测试结果显示以太网分析仪的帧发送速率为 10 965 帧/s，接收速率为 1 701 帧/s。继续调节发送的报文速率，只要大于 1 701 帧/s，TestCenter 接收到的响应报文速率均为 1 701 帧/s，且通过报文解析，确认收到的报文均为 NTP 响应报文，如图 9 所示。

No.	Time	Source	Destination	Protocol	Length	Info
4	0.001135270	192.168.1.10	192.168.1.168	NTP	94	NTP Version 3, server
5	0.001468230	192.168.1.10	192.168.1.169	NTP	94	NTP Version 3, server
6	0.001798660	192.168.1.10	192.168.1.170	NTP	94	NTP Version 3, server
7	0.002127550	192.168.1.10	192.168.1.171	NTP	94	NTP Version 3, server
8	0.002556850	192.168.1.10	192.168.1.172	NTP	94	NTP Version 3, server
9	0.002887830	192.168.1.10	192.168.1.173	NTP	94	NTP Version 3, server
10	0.003266390	192.168.1.10	192.168.1.174	NTP	94	NTP Version 3, server

> Frame 12: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)  
 > Ethernet II, Src: Zhejiang\_32:da (78:ca:83:b0:32:da), Dst: ArcComCon\_08:ac:17 (00:80:66:08:ac:17)  
 > Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.175  
 > User Datagram Protocol, Src Port: 123, Dst Port: 123  
 > Network Time Protocol (NTP Version 3, server)

图 9 TestCenter 压力测试接收到的 NTP 报文

通过以上过程，可得到被测 NTP 服务器的最大响应能力为 1 701 帧/s，说明该方法能有效测量出 NTP 服务器的响应能力。

为了验证该方法在不同端口速率下的报文发送能力，我们选了不同速率端口模块对另一款 NTP 服务器进行了压力测试，该服务器是基于 FPGA 处理架构设计的，采用硬件处理方式，具有较高的处理速度。选择 1 000 Mb/s 速率以太网模块，端口负载为 70%，也就是最大传输速率为 700 Mb/s。测试结果显示以太网分析仪帧发送速率为 767 544 帧/s，接收速率为 531 916 帧/s，容易验证分析仪达到了理论的最大帧发送速率。继续调节发送的报文速率，只要大于 53 196 帧/s，TestCenter 接收到的报文速率均为 53 196 帧/s，且通过报文解析，确认收到的报文均为 NTP 响应报文。

通过以上过程，说明该方法的报文发送能力和端口速率无关，不管什么速率，均能达到理论的最大速率，能满足 NTP 服务器的任意速率响应能力测试需求。

### 4.3 多端口实测验证

为了模拟多客户端进行 NTP 请求，我们进一步在 TestCenter 的其他端口设置同样的报文发送功能，对 4.2 节第一款 NTP 服务器的两个端口同时进行压力测试，测试拓扑如图 10 所示。

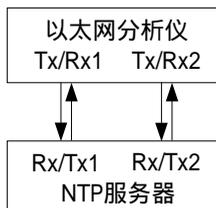


图 10 NTP 服务器响应能力测试拓扑（多端口）

由于 NTP 服务器的处理能力是由 CPU 决定，所以所有端口的处理能力之和等于 CPU 处理能力。测试结果显示 TestCenter 的两端口收到的 NTP 响应报文数量分别为 1 378 帧/s 和 323 帧/s，两端口的报文数量随时间会发生变化，但总和都是 1 701 帧/s，和单端口的总数是一样的。通过以上过程，可以验证该方法对于多端口场景测试也是有效的，能满足多客户端多端口情况下 NTP 服务器的响应能力测试。

对于具备多个 CPU 的服务器，可以用该方法进行类似累加处理。

## 5 结语

本文针对 NTP 服务器的压力测试，设计了一种基于硬件构造报文方式的压力测试方法。该方法对 NTP 报文进行拆解分析，其次用以太网分析仪进行 NTP 报文的构建，并用以太网分析仪的报文编辑功能，对报文的速率、IP 地址，及端口进行设置来模拟多客户端多端口的压力测试场景。文章通过自环测试、单端口测试和多端口测试对该方法进行了验证，自环测试结果表明该方法能够达到最大的理论发送速率，单端口和多端口测试结果表明该方法可以适用不同响应速率和多端口的测试需求。综上所述，该方法能满足任意报文速率的多客户端多端口场景的 NTP 服务器压力测试需求，将在时间同步领域具有广泛的应用前景。

### 参考文献：

- [1] 屠乐奇. 高精度计算机网络时钟同步技术的研究[D]. 北京: 北京化工大学, 2010.
- [2] 康飞. 基于北斗 GPS 的 NTP 时间服务器的研究与应用[D]. 西安: 西安电子科技大学, 2019.
- [3] 易建勋. 微处理器(CPU)的结构与性能[M]. 北京: 清华大学出版社, 2003: 40-60.
- [4] 吴鹏, 华宇, 张旭海. NTP 服务器的响应阈值测试软件设计[J]. 时间频率学报, 2017, 40(1): 36-42.
- [5] 张尤长. 支持 HTTP 协议压力测试框架中环境部署的设计与实现[D]. 北京: 北京交通大学, 2012.
- [6] Titan, Inc. TimeAcc-007——全球第一款精密的时间测量工具[EB/OL]. (2005-12-19)[2020-12-27]. [http://www.sh-titan.com/website/Timing\\_TimeAcc-007.htm?id=spansjzhcly](http://www.sh-titan.com/website/Timing_TimeAcc-007.htm?id=spansjzhcly).
- [7] MILLS D L. Internet time synchronization-the network time protocol[J]. IEEE Transactions on Communications(S0090-6778), 1993, 39(10): 1482-1492.
- [8] IETF. Network Time Protocol(Version 3) Specification, Implementation and Analysis[S].
- [9] IETF. Network Time Protocol Version 4: Protocol and Algorithms Specification[S].
- [10] STEVENS W R. TCP-IP 详解卷 1-协议[M]. 北京: 机械工业出版社, 2016: 38-46.
- [11] Spirent, Inc. Hypermetrics CV 2/8-Port 10 Gigabit Ethernet Test Modual[EB/OL]. (2019-12-10)[2020-12-27]. [https://assets.ctfassets.net/wcxs9ap8i19s/1tZwDk5nv4fqAt38tFvcIH/936be23f7b3d3ed1129ef95ed54bce4e/STC\\_HyperMetrics\\_CV-Module\\_datasheet.pdf](https://assets.ctfassets.net/wcxs9ap8i19s/1tZwDk5nv4fqAt38tFvcIH/936be23f7b3d3ed1129ef95ed54bce4e/STC_HyperMetrics_CV-Module_datasheet.pdf).