<http://www.baike.com/wiki/TTL>

TTL是IP协议包中的一个值，它告诉网络[路由器](http://www.baike.com/sowiki/%E8%B7%AF%E7%94%B1%E5%99%A8?prd=content_doc_search)包在网络中的时间是否太长而应被丢弃。有很多原因使包在一定时间内不能被传递到目的地。例如，不正确的路由表可能导致包的无限循环。一个解决方法就是在一段时间后丢弃这个包，然后给发送者一个报文，由发送者决定是否要重发。TTL的初值通常是系统缺省值，是包头中的8位的域。TTL的最初设想是确定一个时间范围，超过此时间就把包丢弃。由于每个路由器都至少要把TTL域减一，TTL通常表示包在被丢弃前最多能经过的路由器个数。当记数到0时，路由器决定丢弃该包，并发送一个ICMP报文给最初的发送者。  
指定数据报被路由器丢弃之前允许通过的网段数量。   
TTL 是由发送主机设置的，以防止数据包不断在 IP 互联网络上永不终止地循环。转发 IP 数据包时，要求路由器至少将 TTL 减小 1。   
使用PING时涉及到的 ICMP 报文类型   
一个为ICMP请求回显（ICMP Echo Request）   
一个为ICMP回显应答（ICMP Echo Reply）   
TTL 字段值可以帮助我们识别操作系统类型。   
UNIX 及类 UNIX 操作系统 ICMP 回显应答的 TTL 字段值为 255   
Compaq Tru64 5.0 ICMP 回显应答的 TTL 字段值为 64   
微软 Windows NT/2K操作系统 ICMP 回显应答的 TTL 字段值为 128   
微软 Windows 95 操作系统 ICMP 回显应答的 TTL 字段值为 32   
当然，返回的TTL值是相同的   
但有些情况下有所特殊   
LINUX Kernel 2.2.x & 2.4.x ICMP 回显应答的 TTL 字段值为 64   
FreeBSD 4.1, 4.0, 3.4;   
Sun Solaris 2.5.1, 2.6, 2.7, 2.8;   
OpenBSD 2.6, 2.7,   
NetBSD   
HP UX 10.20   
ICMP 回显应答的 TTL 字段值为 255   
Windows 95/98/98SE   
Windows ME   
ICMP 回显应答的 TTL 字段值为 32   
Windows NT4 WRKS   
Windows NT4 Server   
Windows 2000   
ICMP 回显应答的 TTL 字段值为 128   
这样，我们就可以通过这种方法来辨别[操作系统](http://www.baike.com/sowiki/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F?prd=content_doc_search)  
[TTL(逻辑门电路)](http://www.baike.com/sowiki/TTL(%E9%80%BB%E8%BE%91%E9%97%A8%E7%94%B5%E8%B7%AF)?prd=content_doc_search)  
全称Transistor-Transistor Logic,即BJT-BJT[逻辑门](http://www.baike.com/sowiki/%E9%80%BB%E8%BE%91%E9%97%A8?prd=content_doc_search)电路，是数字电子技术中常用的一种逻辑门电路，应用较早，技术已比较成熟。TTL主要有BJT（Bipolar Junction Transistor 即[双极结型晶体管](http://www.baike.com/sowiki/%E5%8F%8C%E6%9E%81%E7%BB%93%E5%9E%8B%E6%99%B6%E4%BD%93%E7%AE%A1?prd=content_doc_search" \o "双极结型晶体管)，晶体三极管）和电阻构成，具有速度快的特点。最早的TTL门电路是74系列，后来出现了74H系列，74L系列，74LS,74AS,74ALS等系列。但是由于TTL功耗大等缺点，正逐渐被CMOS电路取代。

## Ping中的TTL/TTL [编辑](http://www.baike.com/wiki/TTL)

举例来说：  
以下是ping曙光博客的返回值：  
C:\Documents and Settings\user>ping www.hinn.cn

Pinging www.hinn.cn [66.235.202.42] with 32 bytes of data:

Reply from 66.235.202.42: bytes=32 time=254ms TTL=51  
Reply from 66.235.202.42: bytes=32 time=256ms TTL=51  
[Request timed out](http://www.baike.com/sowiki/Request%20timed%20out?prd=content_doc_search).  
Reply from 66.235.202.42: bytes=32 time=260ms TTL=51

Ping statistics for 66.235.202.42:  
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 254ms, Maximum = 260ms, Average = 256ms   
从结果中可以看出曙光博客服务器的IP地址是：66.235.202.42，所用的时间是256ms等，那TTL等与51是什么意思呢？  
TTL是生存时间的意思，就是说这个ping的[数据包](http://www.baike.com/sowiki/%E6%95%B0%E6%8D%AE%E5%8C%85?prd=content_doc_search" \o "数据包)能在网络上存在多少时间。当对[网络](http://www.baike.com/sowiki/%E7%BD%91%E7%BB%9C?prd=content_doc_search)上的主机进行ping操作的时候，本地机器会发出一个数据包，数据包经过一定数量的[路由器](http://www.baike.com/sowiki/%E8%B7%AF%E7%94%B1%E5%99%A8?prd=content_doc_search" \o "路由器)传送到目的主机，但是由于很多的原因，一些数据包不能正常传送到目的主机，那如果不给这些数据包一个生存时间的话，这些数据包会一直在网络上传送，导致网络开销的增大。当数据包传送到一个路由器之后，TTL就自动减1，如果减到0了还是没有传送到目的主机，那么就自动丢失。就像上面ping曙光博客的时候第三次那样，出现Request timed out的情况，增加TTL来减少网络资源的消耗。默认情况下，[Linux](http://www.baike.com/sowiki/Linux?prd=content_doc_search" \o "Linux)系统的TTL值为64或255，Windows NT/2000/XP系统的TTL值为128，Windows 98系统的TTL值为32，UNIX主机的TTL值为255。（这个是从网络上找到的），曙光博客的目的主机是采用FreeBSD系统的（可能已经更换），在这里可能TTL值是64，而不是UNIX主机的255，所以在从这里到目的主机经过了64-51=13个路由。当不知道目的主机的操作系统的时候我们可以根据TTL来猜测，但是不一定100%准确，如果目的主机是windows，但是经过了比如75个路由器，那么TTL的返回值是128-75=53，那么你可能认为这个目的主机是Linux系统，但是一般不会经过那么多的路由器，所以通过TTL来判断目的主机的操作系统还是有一定的依据的。

## 示例/TTL [编辑](http://www.baike.com/wiki/TTL)

生存时间，简单的说它表示DNS记录在DNS服务器上缓存时间。要理解它的值，请先看下面的一个例子：   
假设，有这样一个域名myhost.baiwan-han.c o m（其实，这就是一条DNS记录，通常表示在baiwan-han.c o m域中有一台名为myhost的主机）对应IP地址为1.1.1.1，它的TTL为10分钟。这个域名或称这条记录存储在一台名为-ns.baiwan-han.c o m的DNS服务器上。  
此时如果有一个用户在浏览器中键入一下地址（又称URL）：myhost.baiwan-han.c o m 那么会发生些什么呢？  
该访问者指定的DNS服务器（或是他的ISP，互联网服务商，动态分配给他的)8.8.8.8就会试图为他解释myhost.baiwan-han.c o m，当然8.8.8.8这台DNS服务器由于没有包含myhost.baiwan-han.c o m这条信息，因此无法立即解析，但是通过全球DNS的递归查询后，最终定位到-ns.baiwan-han.c o m这台DNS服务器，-ns.baiwan-han.c o m这台DNS服务器将myhost.baiwan-han.c o m对应的IP地址1.1.1.1告诉8.8.8.8这台DNS服务器，然有再由8.8.8.8告诉用户结果。8.8.8.8为了以后加快对myhost.baiwan-han.c o m这条记录的解析，就将刚才的1.1.1.1结果保留一段时间，这就是TTL时间，在这段时间内如果用户又有对myhost.baiwan-han.c o m这条记录的解析请求，它就直接告诉用户1.1.1.1，当TTL到期则又会重复上面的过程。