基于经验值的分层 PMTU 探测算法

黄永峰, 张 珂
(清华大学电子工程系网络研究所, 北京 100084)

摘 要: IPv6 协议规定路由器不再对 IP 分组进行分段, 因此端到端的 PMTU 探测成为 IPv6 应用中的关键环节, 研究高性能的 PMTU 探测算法是当前下一代互联网领域中的研究热点. 论文针对目前分层探测算法存在等概率探测点带来的发送探测包较多的问题, 提出了一种基于经验值的分层探测算法, 并从探测精度、总花费时间和总发包数等方面分析了该算法的测试性能, 在 CERNET2 环境下, 对比测试了一些常见 PMTU 算法的性能. 实验结果表明, 该算法的测试精度、测试耗时和发送探测分组数等性能方面都优于其他探测算法.

关键词: PMTU; IPv6; 互联网
中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2007) 10-1865-05

A Hierarchical Path MTU Discovery Method Based on Empiristic Data

HUANG Yong-feng, ZHANG Ke
(*Department of Electronic Engineering, Tsinghua University, Beijing* 100084, *China*)

Abstract: In this paper, we suggest some parameters to evaluate the performance of various PMTU discovery methods, such as the precision, the consuming time, and the number of packages to be sent, etc. And we also suggest a new efficient and agile method called Hierarchical-Test based Empiristic data to detect the PMTU. Then we elaborately discuss the quantitative analyses method of the algorithm. Meanwhile, we have realized the algorithm under Linux operating system environment, and have done some real-world tests based on 58 websites given stochastically within the Education and Research Network of China. Finally, we further analyze the performance of the method through these data that we have attained by experiment.

Key words: PMTU; IPv6; Internet

1 引言

MTU (Maximum Transfer Unit) 指的是网络某链路中的最大传输单元, 是网络应用程序中封装 IP 分组的重要参数, 也是网络优化的重要指标. 不同网络传输介质限定的 MTU 是不一样的. PMTU (Path MTU) 是指网络端点之间的传输路径容许通过的最大传输单元值, PMTU 具有"木桶效应", 一条网络路径的 PMTU 是组成该传输路径的所有链路中 MTU 最小值. 由于 IPv6 路由器已经不再具有 IPv4 路由器对 IP 分组进行分段和重组功能, 因此, 在 IPv6 网络中, PMTU 探测显得尤其重要, 它是优化网络传输性能的最重要方法. 目前, PMTU 探测方法成为下一代网络研究的兴奋点.

IETF 制定了一系列有关 PMTU 探测标准和建议, 其中 RFC1981[1]规定了 IPv6 网络各个协议层间测量 PMTU 工作职能和范畴, 但对具体探测算法没有做规定; RFC 2923[2]对 IPv6 网络实施 PMTU 测量过程中 TCP 层可能出现的问题制定了相应解决方案, 这些问题包括路由黑洞效应、应答延迟效应以及最大传输段确定等. 另外, 国内外很多学者也对 PMTU 的探测算法进行了研究, 例如 R. M. Bournas[3] 提出了英式拍卖算法, 该算法采用递减方式, 从最大包长开始对目标站点进行测试, 直到有一个包被成功发送为止. 那么这个数据包的大小即为这条链路上的 PMTU 值. 该算法总的探测包次数 N 随着 PMTU 的实际测量值不同而不同, 也与测试精度相关. J. Mogul, Decwel[4] 提出了荷兰拍卖算法. 该算法是采用递增方式, 从小到大对目标站点进行测试. 如果有一个数据包发送不成功, 意味着该数据包长超过了链路 PMTU 值, 那么可以确定链路 PMTU 值为最后一个发送成功数

收稿日期: 2006-04-16; 修回日期: 2007-03-12
基金项目: 国家 863 高技术研究发展计划(No. 2006AA01Z444); 国家自然科学基金(No. 60773140)

据包的大小. 这两种算法在等待时间和总发包数等性能方面不相上下. 但在测量精度方面, 递增法不如递减法. 因为网络丢包会造成探测误判, 误判带来的误差是当前发包大小和实际 PMTU 之差, 是一个很不确定的值. 因此在很大程度上影响着递增法的精度. 为了进一步提高探测效率, Nick Christenson[6] 提出了一种分层探测算法, 基本思想是将 PMTU 存在空间$[M_{\min}, M_{\max}]$分为 K 段, 然后通过发少量大小不一的数据包来确定 PMTU 的取值空间, 并通过迭代来一步一步确定这个空间. 然而, 该算法对初始探测空间$[M_{\min}, M_{\max}]$中所有试探点采取相等的概率进行测试, 结果发送了许多无用探测包, 在测试性能方面需要进一步改进. 因此, 本论文提出了一种基于经验值的 K 段分层探测算法, 该算法是根据"PMTU 分布"知识库中的数据, 在初始探测空间$[M_{\min}, M_{\max}]$中有针对性地确定一些试探点进行定点探测, 从而来提高探测性能.

本文的结构如下: 第 2 节分析分层探测算法基本原理以及探测性能. 第3 节提出了一种基于经验值的分层探测算法, 分析了该算法的探测精度、总花费时间和总发包数等方面性能, 第 4 部分介绍该算法在 CERNET2 网络环境下的实际测试结果.

2 分层探测算法

为了提高 PMTU 探测算法的性能, Nick Christenson 等人提出了分层探测算法, 其中比较典型是 3 段分层探测算法. 该方法是根据已知 PMTU 探测初始空间$[M_{\min}, M_{\max}]$, 将空间$[M_{\min}, M_{\max}]$分为 $\left|\frac{M_{\min}}{3}+\frac{2M_{\max}}{3}\right|$, $\left|\frac{2M_{\min}}{3}+\frac{M_{\max}}{3}\right|$2 个子空间进行测试, 得到 PMTU 更明确的存在子空间$[M'_{\min}, M'_{\max}]$. 如图 1 所示.

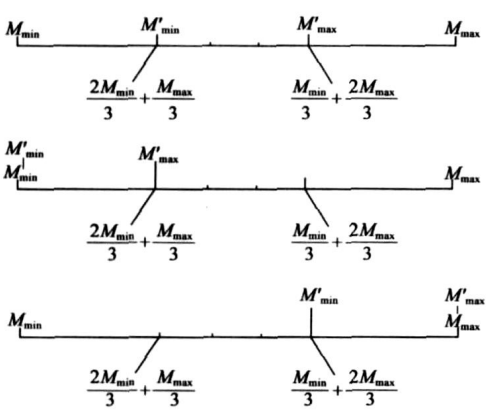

图 1 3段分层探测算法

如果通过 N 层探测才能查找到 MTU 值, 则我们称该探测过程为 N 层探测. 对于分层探测算法来说, 影响探测性能的主要因素是探测层数 N, 它决定着探测过程需要发送的总分组数. 下面分析 3 段分层探测算法需要完成的探测层数 N.

如果不考虑第 0 层的 $M_{\min}$和 $M_{\max}$, 第 1 层需要发送 2 个探测包, 它们的大小分别是$\frac{M_{\min}}{3}+\frac{2M_{\max}}{3}$和$\frac{2M_{\min}}{3}+\frac{M_{\max}}{3}$, 第 2 层发送 6 个探测包, 它们的大小分别是 $\frac{2M_{\min}+M_1^{(1)}}{3}$, $\frac{M_{\min}+2M_1^{(1)}}{3}$, $\frac{2M_1^{(1)}+M_2^{(1)}}{3}$, $\frac{M_1^{(1)}+2M_2^{(1)}}{3}$, $\frac{2M_2^{(1)}+M_{\max}}{3}$和$\frac{M_2^{(1)}+2M_{\max}}{3}$一共 6 个值. 其中 $M_1^{(1)}$, $M_2^{(1)}$ 分别为第 1 层分段标志点, 同时也是第一次可以查找出 PMTU 值. 基于上述规律, 可计算出第 N 层发送探测包的数目为 $2\times 3^{n-1}$, 其中 N 为层数. 层数 N 的计算方法如式(1)所示.

$$N=\log_3(M_{\max}-M_{\min})-1 \tag{1}$$

如果考虑到 PMTU 值允许一定的粗糙度, 可以模糊识别的最小步长 $step$, 那么最后探测出的 PMTU 值将会只是整数集$[M_{\min}, M_{\max}]$的一个子集, 即最后一层探测要求是 $M_{i+1}^{(N)}-M_i^{(N)}\geqslant step$. 那么总层数的计算方法如式(2)所示.

$$N=\log_3\left|\frac{M_{\max}-M_{\min}}{step}\right| \tag{2}$$

因此, 在 3 段分层探测算法过程中, 需要发送的总探测包数目为 $M(M=2^*N)$.

式(2)意味着一般需要通过 N 层探测才能确定出最接近真实值的 PMTU. 由式(2)可以看出, 探测总层数 N 不仅与初始探测空间$[M_{\min}, M_{\max}]$, 而且也与最小步长 $step$ 以及探测点的确定有关. 因此, 确定一个合理的初始探测空间$[M_{\min}, M_{\max}]$和最佳探测点, 是提高 MTU 探测算法性能的有效途径.

3 基于经验值的分层 PMTU 探测算法

3.1 算法过程

通过分析大量的实验数据可以发现, IPv6 网络的 PMTU 主要聚类部分值上, 例如, 1500、576 等. 因此, 我们在设计分层探测算法时, 对初始探测空间$[M_{\min}, M_{\max}]$中的所有试探点采用等概率测试, 显然是不合理的. 因此, 基于当前网络环境中 PMTU 的非等概率分布规律, 我们针对 Nick Christenson 提出的均匀分布测试点的等概率探测算法进行改进, 提出了一种基于经验值的分层探测算法. 算法过程如下.

根据经验数据, 假设 PMTU 可能聚类在初始探测空间$[M_{\min}, M_{\max}]$中的 $M_1, M_2, \cdots, M_n$. 因此, 我们可以按照如下过程进行探测.

第 1 步: 根据经验值个数确定 n 个测试点, 发送 n 个分组来测试 PMTU 的第 2 次探测子空间.

第 2 步: 假设探测到 PMTU 是属于$[M_i, M_j)$范围, 分别发送 $M_i+ step$ 和$M_j- step$ 探测分组, 并按表 1 进行判决.

第 3 步: 按照新的探测空间$[M_i, M_j]$, 重复第 1 步和第 2 步的过程继续探测, 直至找到 PMTU.

整个算法过程如图 2 所示.

表 1　探测判决算法

	If$(M_i+ step)$成功	If$(M_i+ step)$失败
If$(M_j- step)$成功	$M_i= M_j- step$ $M_j= M_j$	
If$(M_j- step)$失败	$M_i= M_i+ step$ $M_j= M_j- step$	$M_j= M_i$ $M_i= M_i$

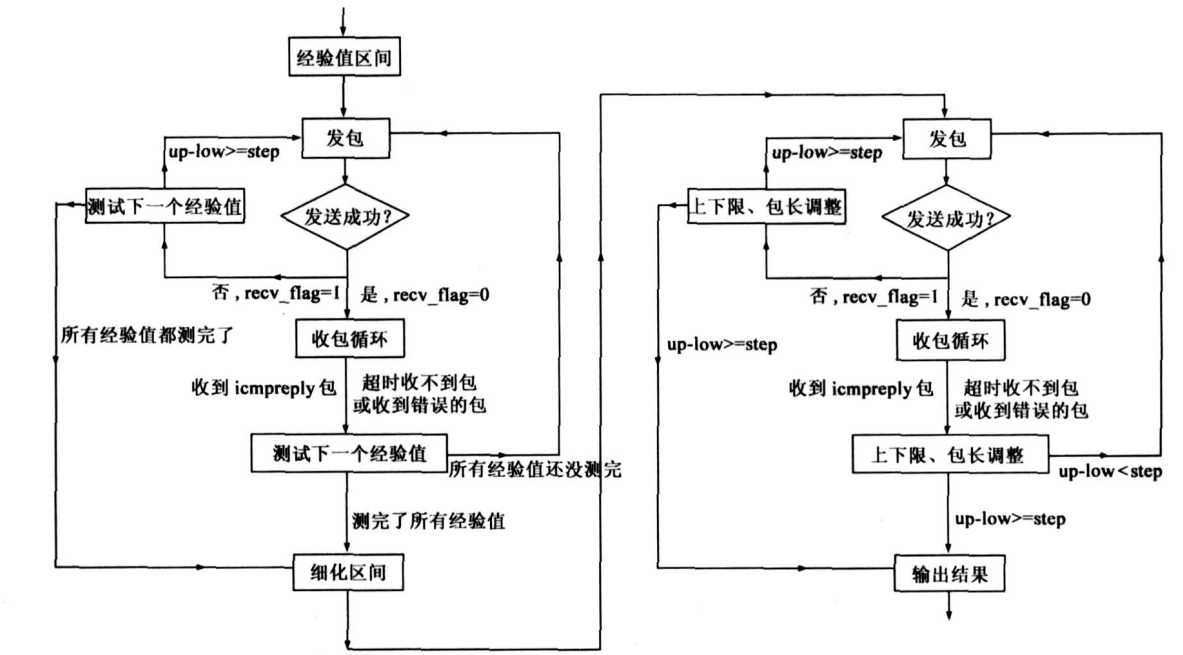

图 2　基于经验值的分层探测算法

3.3　算法性能分析

基于经验值的分层探测算法是在多段分层算法的基础上, 根据经验值来实现定点探测. 该算法最显著的优点是一次定位比较准确. 其次, 由于和某个经验值 M_k 接近, 作为 PMTU 的可能性就越大, 二次和多次定位的准确度相对于 Nick 分层法来说也很高. 如果离最初的 PMTU 聚集值越近, 探测 PMTU 的可能性也越大. 对于经验值 M_i 来说, PMTU 的概率分布情况是一个叠加过的高斯分布, 如式(3)所示.

$$f_1(x)= \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x- M_i)^2}{2\sigma_i^2}} \tag{3}$$

其中$f_1(x)$表示 PMTU 为 x 的概率, σ 表示路径丢包率系数.

同样, 对于经验值 M_j, PMTU 的概率分布情况也是一个叠加过的高斯分布, 如式(4)所示.

$$f_2(x)= \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x- M_j)^2}{2\sigma_j^2}} \tag{4}$$

PMTU 的平均概率平均分布$f(x)$如式(5)所示.

$$f(x)= \frac{1}{2}[f_1(x)+ f_2(x)], \quad x \in [M_i, M_j] \tag{5}$$

在 M_i 和 M_j 相差较大的情况下, 对于 $x \in \left[M_i, \frac{M_i+ M_j}{2}\right]$, $f_2(x) \to 0$, 则

$$f(x) \approx \frac{1}{2} f_1(x) \tag{6}$$

同理, $x \in \left[\frac{M_i+ M_j}{2}, M_j\right]$, $f_1(x) \to 0$, $f(x) \approx \frac{1}{2} f_2(x)$ (7)

根据 PMTU 的概率分布函数, 可以初步估计出 PMTU 的可以得到期望的探测步数.

如果将它离散化, 则它的概率密度分布分别如式(8)和式(9)所示.

$$f(X_K^{(1)})= \int_{x= X_K^{(1)}}^{X_{K+1}^{(1)}} f_1(x)\,\mathrm{d}x \tag{8}$$

$$f(X_K^{(2)})= \int_{x= X_K^{(2)}}^{X_{K+1}^{(2)}} f_2(x)\,\mathrm{d}x \tag{9}$$

根据式(8)和式(9), 可以得到期望的探测步数, 如式(10)所示.

$$E(K)= \max\left(\sum_{i=1}^{n} i f(X_i^{(1)}), \sum_{i=1}^{n} i f(X_i^{(2)})\right) \tag{10}$$

假设 $X_K^{(1)}= 20K, K \leqslant 10$, $X_K^{(2)}= 200- 20K, K \leqslant 10$, σ= 2, 则可以得到 $E(K)= 1.37$, 也就是说, 平均需要走两步才能探测出 PMTU. 同时, 我们可以得到总发包数 M, 如

式(11)所示.

$$M = n + 2E(K) \quad (11)$$

其中, n 表示在第 1 次发送的包数, 即经验值的个数.

同样, 根据 $E(K)$ 可以计算出总耗时量 T, 如式(12)所示.

$$T = (n + 2E(K))t + 2E(K)\tau_0 \quad (12)$$

其中, τ_0 表示路径等待极限, 发出一个数据包后放弃等待默认传送失败的最短时间. t 为发送一个包所需的时间.

4 实验仿真

为了验证经验探测算法的性能, 我们在 CERNET2 环境下, 采用该算法对于给定的 57 个教育网站点进行测试. 步长分别采用 $step = 2, 10, 20, 100$; 测试因子变化顺序先后为: 站点、测试组数、步长; 结果包括: PMTU 值, 总等待耗时, 总发包数, 总收包数. 设置测试上限为 up= 9000; 测试下限为 low= 10. 同时, 采用其他探测方法在相同条件下, 对上述站点进行了测试, 并对比分析了它们在性能上的差异.

4.1 精度测试

图 3 为不同步长条件下的精度测试. 从图 3 中可以看出, 整体来说由于经验值选择得比较到位, 每一个步长的测量结果都很好, 其中以小步长 2 的效果为最佳.

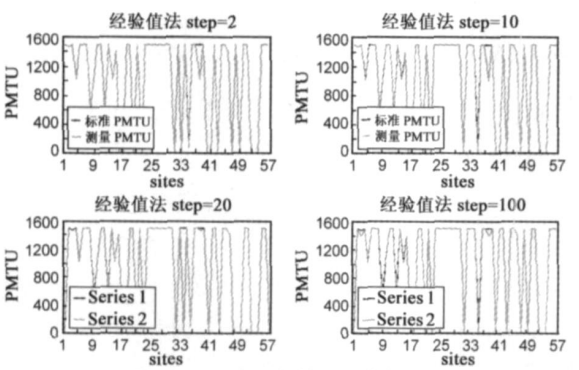

图 3 不同步长条件下的精度测试

同时, 我们对比了不同探测方法在实际测试情况下的精度值. 在表 2 中, 我们之所以选择不同的步长, 是因为这些步长在各种探测方法中, 测试精度是最好的.

表 2 不同探测方法的实际测试精度值比较

method	递增	递减	二分	三层	K 层	经验值
$\alpha_{site[]}$	0.308112	2.62852E-03	6.62927E-03	0.00782	0.006011	0.005223
Δ	62	2	0	2	3	0-2

其中, 精度评价标准分别采用 $\alpha_{site[]}$ 和 Δ 表示.

$$\text{指标 } \alpha_{site[]} = \frac{1}{N}\sum_{n}\left(\frac{\text{标准 } PMTU - \text{测量 } PMTU}{\text{标准 } PMTU}\right)^2$$

来考察测量的准确度. $\alpha_{site[]}$ 是一个 $[0, 1]$ 之间的实数, 越接近 1 反映测量的误差越大, 越接近 0 表示测量的误差越小.

用指标 Δ 来测量平稳误差, 也就是(标准 $PMTU$ - 测量 $PMTU$)的大众值, 反映测量的理论期望误差.

4.2 发包总数和总耗时量

图 4 和图 5 分别为不同探测方法的探测消耗时间和发包总数的测试结果. 结果表明, 经验探测方法无论是发包总数还是总耗时量都比其他探测方法要小.

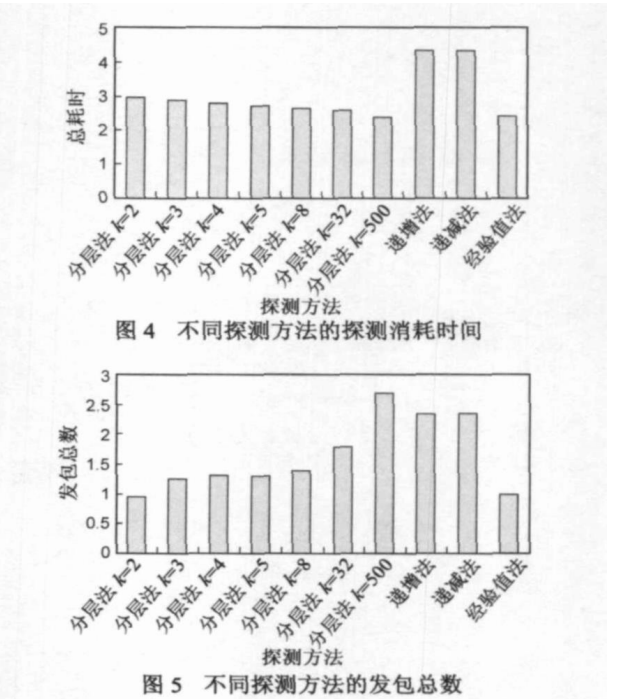

图 4 不同探测方法的探测消耗时间

图 5 不同探测方法的发包总数

从图 4 和图 5 可以得到如下结论: 无论是总耗时, 还是总发包数, 基于经验的分层探测方法, 相比其他探测方法都是最佳.

5 小结

对于经验值的分层探测方法, 和其他探测方法一样, 在实际测试时, 需要考虑实际的网络环境的影响, 其中影响最大的是丢包率. 对于没有响应探测包, 需要重新发送. 实验分析可知, 当 K 很大时, 分层法的误差是很小的. 比如, 如果选定 $\sigma = 5\%$, 那么二分法的误差为 225, 而 $K = 50$ 时, 误差只有 9.97. 因此, 从精度上来说, 分层多的分层法在抗丢包干扰的情况下表现比较好. 另外, 基于经验值的分层探测方法, 很关键一个环节是"PMTU 分布"知识库上建立问题. 从上面分析可知, 该算法性能的好坏除了与分层多少有关外, 还与经验值的选择有关. 如果我们能将采用多种方法的测试结果加到"经验值"库中, 必然能够再度提高该算法的性能和精准率. 例如, 我们的测试结果表明, 1500 的最大传输单元限制仍然是主流. 少量最大传输单元值集中在 1024、576 和 100 附近, 这样的站点大约占测试站点的 1.7%. 因此, 采用基于经验值的分层探测算法, 首

要工作是建立完备的"PMTU 分布"知识库.

致谢:本论文的数据测量得到课题组黄潇同学的支持

参考文献:

[1] J McCann, S Deering. RFC: 1981-Path MTU discovery for IP version 6[DB/OL]. www. ietf. org, 2003.

[2] K Lahey, dotRocket. RFC: 2923-TCP problems with path MTU discovery[DB/OL]. www. ietf. org, 2003.

[3] R M Bournas. IBM software solutions division, optimization of TCP segment size for file transfer[DB/OL]. www. ibm. com, 2004.

[4] Kjersti Modeklev and Per Gunningberg, How a large AMT MTU causes deadlocks in TCP data transfers[R]. Norwegian Telecom Research, Swedish Institution of Computer Science, 2004. 23-31.

[5] J Mogul, DECWEL, S. Deering. RFC: 1191-Path MTU Discovery, Stanford University[DB]. www. stanford. edu/, 2003.

[6] Nick Christenson. Different MTUs in the network, sendmail performance tuning[DB/OL]. www. sendmail. org, 2004.

[7] Metz C. Moving towards an IPv6 future[J]. IEEE Internet Computing, 7(3): 25-26.

[8] Adams A, Bu T, C′aceres R, et al. The use of end-to-end multicast measurements for characterizing internal network behavior [J]. IEEE Communication Magazine, 2004, 38(2): 341-349.

作者简介:

黄永峰　男, 1967 年生, 计算机系统结构专业博士, 副教授. 主要从事计算机网络方向的研究. E-mail: yfhuang@tsinghua. edu. cn

张　珂　男, 1978 年生, 信号与信息处理专业博士生. 主要从事计算机网络方向的研究.