

# 基于被动测量的 TCP 报文丢包定位方法研究

高中耀，程光

(1. 东南大学计算机科学与工程学院, 南京, 211189; 2. 计算机网络和信息集成教育部重点实验室, 南京, 211189)

**摘要:** TCP 报文丢失会影响网络及应用程序的性能, 检测出丢包发生的位置范围将有利于网络管理, 因此, 本文提出一种基于单点被动测量的报文丢包分段检测方法 (Half Path Detection of Packet Loss, HPDPL), 该方法通过监测 IP 的 ID 号、TCP 的序号以及报文间隔时间来定位报文丢包位置, 并推算出测量点前后网络的丢包率, 论文最后搭建实验环境对算法的准确性及性能进行评估, 并利用 CERNET 主干链路测量到的数据进行乱序率和丢包率分析。

**关键词:** 被动测量; TCP; 乱序; 丢包; 定位; 重传

**中图分类号:** TP393

## Locating TCP Packet Loss based on Passive Measurement

Zhongyao Gao, Guang Cheng

(1. School of Computer Science & Engineering, Southeast University, Nanjing, 211189;

2. the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, 211189)

**Abstract:** Packet loss may cause the degradation of network and application performance, detecting packet loss location will help network management. Firstly, we propose a method called HPDPL in this paper to analyse TCP packet loss based on single point's passive measurement, this method can locate loss position and infer loss rates based on IP header's ID field, TCP header's sequence and packets's interval, finally we build a testbed to evaluate the accuracy and performance of the algorithm, and analyze reordering rates and loss rates based on the trace collected from CERNET.

**Key words:** passive measurement; TCP; reordering; packet loss; locate; retransmit

### 1 引言<sup>1</sup>

TCP 是互联网中广泛使用的传输层协议, 它提供一种可靠的面向连接的字节流服务。很多网络行为会引起 TCP 报文丢失, 比如流量过大导致网络拥塞或中断、错误的网络接线或配置导致报文不可达、网络攻击引起网络质量下降或瘫痪。

TCP 报文传输的一个重要特征就是传输过程中的序号, 报文的 IP 首部和 TCP 首部分别有 ID 号和序号, 正常顺序到达的报文 ID 号和序号应该是非递减的, 当到达的报文序号没有按照非递减方式出现时, 那么网络中出现乱序, 一定程度上的乱序会导致接收端认为报文丢失, 引起拥塞窗口的调整和快速重传, 从而可能会导致网络或者应用程序的性能下降。当发送端接收到多个重复 ACK 或者出现定时器超时, 发送端会认为报文在传输中丢失, 重传该报文, 同样会引起拥塞窗口的调整而影响网络性能。

当网络中出现丢包时, 只有定位出丢包位置才能寻找丢包原因以解决丢包问题。所以, 定位出丢包发生的位置以及推算丢包率将有利于网络管理的进行。众多学者提出各种不同的研究方法, 最常见的是通过 ping 来分析, 此类基于主动测量的方法向网络中注入流量, 可能会增加网络负载甚至对测量结果产生影响, 不适合于经常性的网络管理和维护; Sommers 等人通过在发送端主动发送探测包, 在接收端查看探测包抵达的数目以推测端到端的丢包率<sup>[1]</sup>; Benko 和 Veres 提出基于 TCP 报文序号的被动测量丢包

<sup>1</sup>基金项目: 国家 973 研究计划 (2009CB320505), 国家自然科学基金项目 (60973123)

作者简介: 高中耀, (1982-), 男, 硕士研究生, E-mail: zhygao@njnet.edu.cn

程光, (1973-), 男, 教授, 博导, E-mail: gcheng@njnet.edu.cn

率方法，它将测量点布置在核心或边界路由器上，把端到端路径分成两部分，通过观测 TCP 序号来定位丢包及推算丢包率<sup>[2]</sup>，Ohta 和 Miyazaki 利用两个测量点通过被动测量对报文进行哈希辨别来推算丢包率<sup>[3]</sup>，Friedl 等人通过对比两个测量点的流信息来计算丢包率<sup>[4]</sup>，Jaiswal 等人对 Sprint 骨干链路上的 TCP 连接进行分析，对失序的报文进行分类，其中包括对丢失报文的定位和推算<sup>[5]</sup>，Allman 等人通过观测发送端的重传报文来推算丢包率<sup>[6]</sup>，Nguyen 等人构建了一种叫 HSMM 的模型来对报文丢包进行分析<sup>[7]</sup>，Zhang 等人从入口的随机性和报文丢包入手，对丢包、时延、带宽进行分析<sup>[8]</sup>。

上述基于被动测量研究丢包率及丢包定位的方法多数仅从 TCP 序号的角度考虑，没有结合报文超时来分析丢包，因为三个重复 ACK 或发送端超时均会导致报文丢包。也有学者通过分析往返报文来研究丢包，这样对丢包率的推算会更为精确，但由于网络多路径或负载平衡问题，同一个测量点不一定能完全监测到往返报文。因此本文从序号和时间角度综合考虑，研究单向 TCP 报文流，提出一种基于单点被动测量的报文丢包分段检测方法（Half Path Detection of Packet Loss, HPDPL），测量点可选择在核心或边界路由器上，它将端到端路径分成两部分，通过对网络中的 TCP 报文进行组流和过滤分析以定位丢包位置及推算测量点之前和之后链路的丢包率。利用本文提出的 HPDPL 方法，ISP 可通过二分法定位到出现异常的网络，将异常点缩小到小范围，并且此方法可在日常网络管理中实时观测链路丢包情况。

本文共分为五节，第二节主要介绍定义的测度，第三节介绍报文丢包分段检测算法（HPDPL），第四节介绍搭建的实验环境以及实验结果，第五节为总结。

## 2 测度定义

既然报文丢包会降低网络和应用程序的性能，那么定义相关的测度来描述和分析丢包是非常必要的。假定 ID[i]为接收端接收到报文的 ID 首部 ID 号，Seq[i]为接收端接收到报文的 TCP 首部序号，其中  $i, j=0,1,2,\dots,n$ ，且  $j<i$ 。T(i,j)为接收端接收到的第 i 个报文和第 j 个报文的时间间隔，RTO 为发送端计时器的超时重传时间。

我们定义下面四个条件式：

A、Seq[i]<Seq[j]

B、Seq[i]=Seq[j]

C、T(i,j)>RTO

D、ID[i]-ID[j]≥3

### 2.1 乱序率

随着互联网的飞速发展，各种新兴技术被应用到互联网中，如多路由技术、并行处理技术、链路层重传技术等。这些技术在提升互联网性能的同时，也导致了传输层报文乱序的出现。大量研究表明，报文乱序是互联网中普遍存在的现象<sup>[6]</sup>，随着端到端报文延迟的降低和网络传输速率的增快，报文乱序出现的比例同时也呈增长趋势<sup>[9]</sup>。

RFC4737<sup>[10]</sup>中定义了 11 个报文乱序测度，包括基于延迟的乱序比例，乱序范围等，RFC5236<sup>[11]</sup>深层次的对乱序测度进行定义，提出乱序密度 RD(Reorder Density)和乱序缓冲区使用密度 RBD(Reorder Buffer-occupancy Density)两个测度。

若报文按正常顺序到达，则报文的 TCP 首部序号应该呈非递减出现，即  $Seq[i] \geq Seq[j]$ ，当到达的报文没有按非递减方式出现，即  $Seq[i] < Seq[j]$ ，网络中出现乱序，定义序号为 Seq[i]的报文为乱序报文，乱序率定义为：

乱序率 = 乱序报文个数 / 报文总个数

乱序率在一定程度上影响着丢包率，特别在骨干链路，小部分乱序甚至会引起吞吐量的严重下降<sup>[7]</sup>。

### 2.2 丢包率

RFC2680<sup>[12]</sup>给出了 IP 报文单个丢包事件的测度，RFC3357<sup>[13]</sup>进一步给出了丢包模式的测度，包括基本定义以及基本定义之上的测度序列和测度统计值。HPDPL 方法提出测量点之前的丢包率和测量点之后的丢包率，以下简称“前丢包率”和“后丢包率”。测量点可选择在核心或边界路由器上，它将端到端路

径分成前后两部分，如图 1 所示。位于网络 B 中的客户端从位于网络 A 中的服务器下载文件，那么客户端和服务端建立了 TCP 连接，TCP 报文从发送端发送后到各自网络边界，再经过测量点抵达接收端。

由于多路由等原因，发送端发送报文至接收端和接收端回复 ACK 至发送端的路径可能是不同路径，所以本文选择基于单向报文的 IP 首部 ID 号、TCP 首部序号以及报文间的时间间隔来推算丢包率。

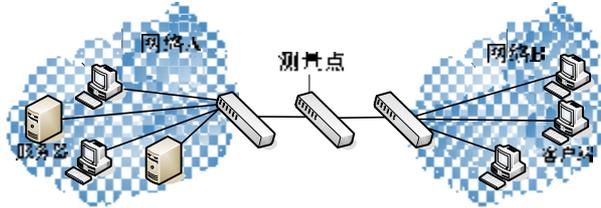


图 1 测量点的布置

### 2.2.1 前丢包率

若报文在测量点之前丢失，如图 2 所示，序号为 1 的报文在抵达测量点前丢失，测量点将看不到序号为 1 的报文经过，发送端接收到三个重复 ACK 后，发送端会重传序号为 1 的报文，此时测量点监测到该报文的重复，即序号为 1 的报文在测量点之前丢包表现为乱序报文的出现。

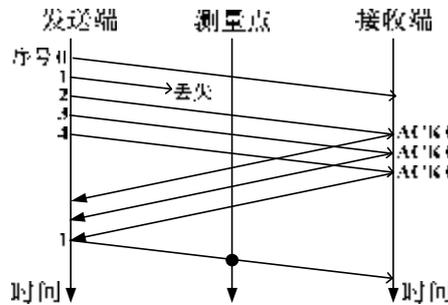


图 2 测量点之前丢包

当报文使得条件式  $A \text{and}(\text{CorD})$  为真时，可判断该报文在测量点之前发生丢失，条件式 A 代表报文出现乱序，条件式 C 代表报文出现超时，条件式 D 代表报文 i 与报文 j 间距大于 3。前丢包率定义为：

$$\text{前丢包率} = \frac{\text{测量点之前的丢包个数}}{\text{报文总个数}}$$

### 2.2.2 后丢包率

若报文在测量点之后丢失，如图 3 所示，序号为 1 的报文在抵达测量点后丢失，测量点监测到序号为 1 的报文经过，发送端接收到三个重复 ACK 后，发送端会重传序号为 1 的报文，此时测量点再次监测到该报文的重复，即序号为 1 的报文在测量点之后丢包表现为重复报文的出现。

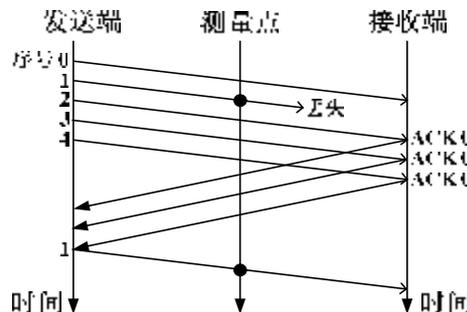


图 3 测量点之后丢包

当报文使得条件式  $B \text{and}(\text{CorD})$  为真时，可判断该报文在测量点之后发生丢失，条件式 B 代表报文为重复报文，条件式 C 代表报文出现超时，条件式 D 代表报文 i 与报文 j 间距大于 3。后丢包率定义为：

$$\text{后丢包率} = \frac{\text{测量点之后的丢包个数}}{\text{报文总个数}}$$

### 3 报文丢包分段检测（HPDPL）算法

#### 3.1 算法描述

HPDPL 算法主要分为五个模块，如图 4 所示，分别是：预处理、数据处理、状态维护、计算及输出模块。预处理模块负责报文到达后的预处理，对报文进行过滤并提取 TCP 报文交给数据处理模块；数据处理模块对报文按照五元组（源 IP、目的 IP、源端口、目的端口、协议号）进行组流，同时记录报文的 IP 首部 ID 号、TCP 首部序号、报文到达时间等相关参数信息；组流的同时，状态维护模块对流信息进行维护，比如维护丢包推算的窗口及超时流信息更新等事务；在前三个模块工作的同时，计算和输出模块利用维护的流信息进行乱序率和丢包率的推算及输出。

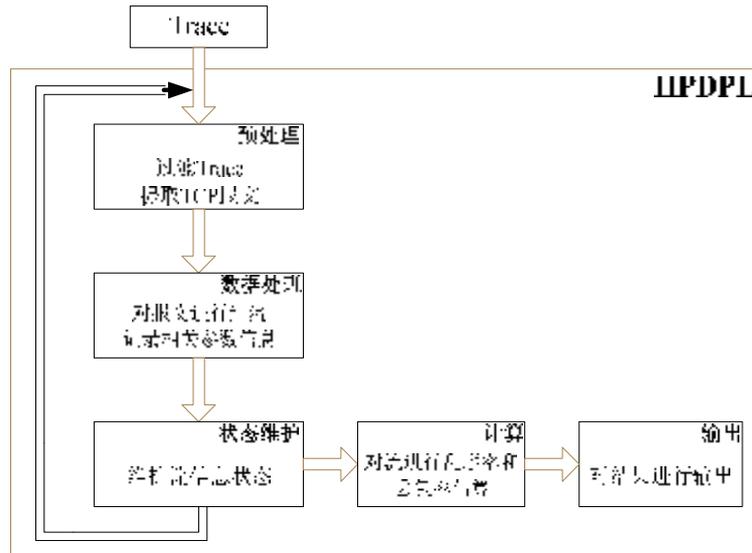


图 4 HPDPL 算法框图

HPDPL 算法的核心是计算模块，即乱序率和丢包率的推算部分，算法如图 5 所示。

```

1      While( packet[i] arrived )
2      {
3          num_all++;
4          Calc_RTT(packet[i] );
5          Calc_RTO( RTT );
6          if( Seq[i]<Seq[j] )
7          {
8              num_outofseq++;
9              if( T[i,j]>RTO or ID[i]-ID[j] ≥ 3 )
10                 num_loss_before++;
11          }
12          if( Seq[i]==Seq[j] and ( T[i,j]>RTO or ID[i]-ID[j] ≥ 3 ) )
13              num_loss_after++;
14          outofseq_rate = num_outofseq / num_all;
15          loss_rate_before = num_loss_before / ( num_all + num_loss_before );
16          loss_rate_after = num_loss_after / num_all;
17      }
    
```

图 5 HPDPL 算法伪代码

第一行是对新报文的获取，第三和四行是对 RTT、RTO 进行计算，根据报文的到达时间提取相邻两

个发送轮次之间的间隙，进而推算 RTT 以及计算 RTO，第六到第十一行是统计符合条件的乱序报文数和测量点之前的丢包个数，第十二和十三行是统计测量点之后的丢包个数，其中  $T[i,j]>RTO$  和  $ID[i]-ID[j] \geq 3$  起到互相补充的作用，主要是考虑到极少数报文的 ID 号可能不是递增出现，利用时间信息对它进行补充。最后的第十四、十五、十六行是对乱序率、前丢包率、后丢包率进行计算。

### 3.2 性能分析

对于本算法来说，算法的计算时间主要花费在 TCP 流信息的维护以及乱序和丢包个数的统计上，对于一个 TCP 流信息需要维护所属报文的 ID、Seq、时间间隔信息，我们对 4.2 节的 CERNET 实验数据进行前、后丢包率推算的同时，对所需维护报文的个数（窗口大小）进行统计，统计结果如图 6 所示，横轴是窗口大小的统计值，纵轴是出现窗口大小的累积比例。

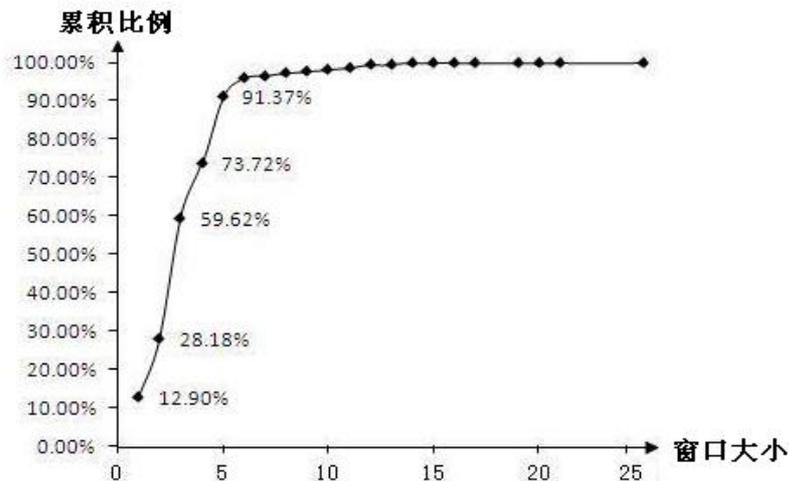


图 6 窗口大小累积比例曲线

观察图 6 可得知：

A、丢包报文与重传报文之间距离出现频率最高的是 3，所占比例达  $59.62\% - 28.18\% = 31.433\%$ ，也验证了收到三个重复 ack 触发快速重传的控制机制；

B、91.373% 的丢包报文与重传报文的距离小于等于 5；

C、丢包报文个数从距离 6 开始出现了骤减，骤减过程一直持续到 25；

D、仅需维护 25 个报文信息，就可观测到 100% 的丢包，即维护的窗口大小为 25。

出于全面性考虑，窗口大小选择为比 25 稍大的值，资源上也不会占用过多，算法的精确度更高。流信息维护过程的时间复杂度是一个常量，即  $O(1)$ ，对于计算模块来说，计算 RTT 和 RTO 的时间复杂度也是一个常量，即  $O(1)$ 。对于 n 个报文来说，总的时间复杂度为  $O(n)$ ，完全符合在线实时报文分析的实际情况。

### 3.3 算法中数据集的选取

网络中报文丢失情况错综复杂，由于我们需要做的是推算丢包率，而不是准确计算丢失报文的个数，所以对部分未知情况不应考虑在推算丢包率的报文数据集之内，只需选择能准确估计丢包率的数据进行推算，数据集通过测量点采集报文构成，对于那些不能确定丢包的报文排除在数据集之外，包括以下几种情况：

A、没有数据内容的 TCP 控制报文

没有数据内容的 TCP 控制报文仅由 TCP 首部组成，这些 TCP 控制报文首部的序号为固定值，在传输过程中不变，也就是说当丢包发生时，中间测量点将无法感知到丢包。所以对于没有数据内容的 TCP 控制报文排除在数据集之外。

B、特殊的超时

由于网络拥塞导致往返时间突然增大，即使没有报文丢失出现，发送端也可能出现超时，发送端会重传报文并等待接收端的 ACK，此时当第一次发送的报文抵达接收端，接收端发送 ACK，发送端可能

会重传丢失报文之后发送的所有报文，那么这些重传的报文不应该认为是丢失的报文。所以当连续丢包的报文个数超过 3 个时，这些报文排除在数据集之外。

#### C、环路

环路是由于网络配置失误或其他因素，导致报文找不到正确的路径，一直在网络中循环发送。为了推算的准确性，环路报文排除在数据集之外。

### 3.4 几种影响推算精度的情况

#### A、特殊的 IP 首部 ID 号

IP 首部的 ID 字段在绝大多数操作系统下随着报文的发送呈递增变化，这些操作系统包括 Windows 95/98/NT/2000/XP、多数的 linux、FreeBSD、Solaris。随着报文的不断发送，当 ID 号出现不规整的情况时，ID 号不能作为丢包的判断条件之一。

#### B、同一报文出现大于两次的丢包

当报文丢失时，发送端会对它进行重传，若重传的报文又丢失了，发送端会再重传，在中间测量点监测到重传报文时，无法辨别该报文是一次还是两次丢包进行的重传，本算法将这种情况认为是对一次丢包进行的重传。

#### C、不同报文的连续丢包

若报文 i 发生丢包，紧接的报文 i+1 也发生丢包，计时器超时后，发送端重传报文 i、i+1，此时，测量点通过超时可判断报文 i 发生丢失，但报文 i+1 是正常到达还是重传到达，中间测量点无法辨别。

#### D、序号回转

IP 首部的 ID 长度为 2 字节，ID 的范围是从 0 到 0xffff，当 ID 到达 0xffff 后从 0 开始回转。TCP 头的 Seq 长度为 4 字节，Seq 的范围从 0 到 0xffffffff，首部的 Seq 长度为 4 字节，范围是从 0 到 0xffffffff，当 Seq 到达 0xffffffff 后也从 0 开始回转。那么在算法计算过程中可能由于反转报文的出现，导致丢包判断出现错误。对于两个临近的没有出现反转的序号，其差值不可能超过序号范围的一半，所以对于 ID 来说，差值不会超过 0x7fff；对于 Seq 来说，差值不会超过 0x7fffffff。

## 4 实验分析

搭建的实验环境分为两种，一种是仿真实验环境，另一种是真实网络环境。由于需要对算法计算的前、后丢包率进行验证，仅通过单个测量点无法做到，所以我们通过搭建仿真实验环境解决此问题，在仿真环境中可采集到发送端、中间测量点、接收端共三个点的报文数据，通过对比三点采集到的报文可以计算出实际的前丢包率和后丢包率，再结合 HPDPL 算法推算出的前、后丢包率，可以对 HPDPL 算法的准确性进行验证。

### 4.1 仿真实验

为验证算法对前、后丢包率推算的准确性，我们搭建仿真环境具体布置如图 7 所示。

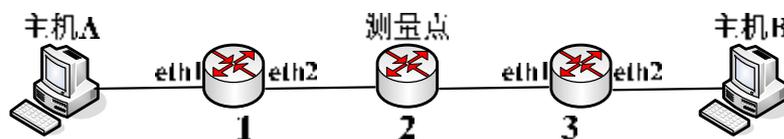


图 7 实验场景

在 fedora14 操作系统上安装五台 fedora8 的虚拟机，使用 Zebra 来实现路由器功能，用 Netem 对路由器端口进行丢包控制。在主机 A 给主机 B 传送文件时，对路由器 1、3 的 eth2 端口进行丢包控制，同时在路由器 1、2、3 上进行被动测量，可通过对比路由器 1、2、3 的测量数据计算出真实前、后丢包率，再利用测量点 M（路由器 2）上的测量数据，运用 HPDPL 方法对前、后丢包率进行推算，具体结果如表 1 所示，前三列是前丢包率的实际值、推算值、误差，后三列是后丢包率的实际值、推算值、误差，其中误差计算公式为： $|实际值-推算值|/实际值$ ，它反映了推算值的准确度，误差越小，准确度越高。

表 1 准确性实验结果

前丢包率			后丢包率		
实际值	推算值	误差	实际值	推算值	误差
0.437%	0.437%	0.000	0.521%	0.576%	0.106
0.648%	0.648%	0.000	5.321%	5.374%	0.010
0.450%	0.426%	0.053	8.547%	8.599%	0.006
0.954%	0.929%	0.026	5.389%	5.443%	0.010
1.064%	0.967%	0.091	8.982%	9.109%	0.014
3.421%	3.244%	0.052	1.035%	1.090%	0.053
3.069%	2.798%	0.088	4.919%	4.920%	0.0002
2.778%	2.425%	0.127	9.590%	9.443%	0.015
5.631%	5.218%	0.073	0.817%	0.954%	0.168
4.931%	4.336%	0.121	8.207%	8.210%	0.003
5.413%	4.900%	0.095	5.307%	5.438%	0.025
9.071%	8.311%	0.084	4.884%	5.042%	0.032
8.819%	8.049%	0.087	9.249%	9.351%	0.011
平均误差		0.069	平均误差		0.035

从表 1 的实验结果可得知，推算的前丢包率平均误差为 0.069，推算的后丢包率平均误差为 0.035，推算值与实际丢包率比较接近，具有不错的准确性。其中推算的后丢包率误差小于前丢包率误差，这是因为推算后丢包率的判断条件明确，仅需要出现重复报文且超时或间隔大于等于 3，而推算前丢包率的判断条件相对没那么明确，有可能存在丢包交错等特殊情况影响精度。另一方面，实际的丢包率越小，推算的丢包率越精确，反之当实际的丢包率越大时，丢包情况可能更复杂，推算的丢包率精度会存在一定程度的下降。

#### 4.2 CERNET 网络环境

实验数据是 CERNET 华东（北）地区网络中心在不同时间段采集的 trace，其采集点位于江苏省教育网边界路由到国家主干路由之间。本文对 2010 年 4 月 12 日 23: 55: 16、2010 年 4 月 20 日 13: 55: 16 和 2010 年 4 月 21 日 15: 55: 16 为开始时间的 5 分钟 trace 进行分析。把 trace 中的 TCP 报文按照五元组（源 IP、目的 IP、源端口、目的端口、协议号）进行组流，相同五元组的报文出现时间间隔超过 16 秒定义为超时，超时后重新组新流。一个流包含两个方向交互的流，从江苏省网到国家主干网的流量方向定义为正向，反之为反向。正向流的源 IP、目的 IP 等于反向流的目的 IP、源 IP，正向流的源端口、目的端口等于反向流的目的端口、源端口。三次测量的实验数据信息如表 2 所示，由于后两次测量数据的测量时间选在下午，所以流量明显高于第一次测量的流量。

表 2 trace 信息统计表

测量	总流数	总报文个数	总字节数
trace1	1059134	23598739	14.23 GB
trace2	2076326	45082259	28.08 GB
trace3	2074568	45078547	27.97 GB

由于报文个数太少会影响计算的准确性，所以采用 HPDPL 方法对报文个数大于 300 的流进行分析，统计结果如表 3 所示，大约 80% 数据量的流满足统计条件。对于正向流来说，三次测量的前丢包率都小于后丢包率，这也验证了正向流在省网内的丢包率小于国家主干网丢包率的事实。

表 3 流信息统计表

		trace1	trace2	trace3
满足条件的流数		5504	10274	9876
正向流	报文数	7872722	13441114	13437532
	字节数	4.72 GB	8.14 GB	8.11 GB
	平均乱序率	4.012%	3.498%	3.949%
	前平均丢包率	1.686%	1.533%	1.661%
	后平均丢包率	2.705%	2.443%	2.571%
反向流	报文数	9650460	16584584	16578534
	字节数	7.22 GB	12.79 GB	12.73 GB
	平均乱序率	1.494%	2.555%	3.117%
	前平均丢包率	0.836%	1.338%	2.001%
	后平均丢包率	1.220%	1.749%	1.981%

## 5 总结

论文从报文丢失的成因进行分析,提出一种基于单点被动测量的报文丢包分段检测方法(Half Path Detection of Packet Loss, HPDPL)来研究 TCP 报文丢包,并搭建实验环境对算法的准确性及性能进行评估分析。最后对 CERNET 华东东北地区网络中心采集的数据进行组流分析,并推算乱序率及前后丢包率。

## 参考文献

- [1] Sommers J, Barford P, Duffield N, et al. Improving accuracy in end-to-end packet loss measurement[J].Computer Communication Review, 2005,35(4).
- [2] Peter Benko and Andreas Veres. A Passive Method for Estimating End-to-End TCP Packet Loss[C].IEEE Globecom, pages 2609-2613, 2002.
- [3] S.Ohta and T.Miyazaki. Passive packet loss monitoring that employs the hash-based identification technique[C].Ninth IFIP/IEEE International symposium on Integrated Network Management, 2005.
- [4] Aleš Friedl, Sven Ubik, Alexandros Kapravelos, et al. Realistic Passive Packet Loss Measurement for High-Speed Networks[J].Computer Science, 2009, Volume 5537/2009, 1-7.
- [5] S. Jaiswal, G Iannaccone, C. Diot, etc al. Measurement and classification of out-of-sequence packets in Tier-1 IP backbone, IEEE/ACM Transactions on networking[C]. Vol.15, NO.1, February 2007.
- [6] Mark Allman, Wesley M.Eddy, Shawn Ostermann. Estimating Loss Rates With TCP[J].ACM Performance Evaluation Review, December 2003.
- [7] H. Nguyen, M. Roughan, Rigorous statistical analysis of internet loss measurements[J].ACM SIGMETRICS Performance Evaluation Review - Performance evaluation review, Volume 38 Issue 1, June 2010.
- [8] Dongli Zhang, Ionescu D, Online Packet Loss Measurement and Estimation for VPN-Based Services[J].Instrumentation and Measurement, Aug 2010.
- [9] L. Gharai, C. Perkins, T. Lehman, Packet reordering, high speed networks and transport protocol performance[C].IEEE 13th IntlConf. On Computer Comm, and Networks, ICCCN 2004, Oct. 2004,pp. 73-78.
- [10] A. Morton, L. Ciavattone, G Ramachandran, et al. RFC 4737, November 2006, Packet reordering metrics[S].
- [11] A Jayasumana, N Piratla, T Banka, et al. RFC5236, June 2008, Improved Packet Reordering Metrics[S].
- [12] G Almes, S Kalidindi, M Zekauskas, RFC2680, 1999, A One-way Packet Loss Metric for IPPM[S].
- [13] R Koodli,R Ravikanth, RFC3357, August 2002, One-way Loss Pattern Sample Metrics[S].