基于被动测量的 TCP 报文乱序研究

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：网络报文乱序是互联网中不可避免的现象，导致网络报文乱序的主要原因包括报文的重传、多路径上的负载平衡、并行处理、路由抖动等等。网络报文乱序会导致网络性能的下降，同时会影响了应用程序的性能。本文首先分析了网络报文乱序的原因以及它对网络性能的影响，然后总结了国内外学者所提出的乱序的测度以及各种针对乱序的解决方案，最后对 CERNET 测量的数据进行乱序率和丢包率计算，发现乱序率在一定程度上影响着丢包率。

关键词：报文乱序；测度；重传；丢包

TCP Packet Reordering based on Passive Measurement

Gao Zhongyao, Cheng Guang

(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 reordering is an inevitable phenomenon on the internet. It can result from many events including packet retransmission, load balancing on the multipaths, parallelism, route fluttering and so on. Packet reordering may cause the degradation of network performance, and it may affect the performance of application simultaneously. In this paper, firstly we analyze the cause of packet reordering and its effect of network performance, Secondly we dicuss the related work about the reordering metrics and solutions. Finally we will calculate the reordering rate and packet loss rate based on the trace data measuring from the CERNET, and find that reordering rate have affects on the loss rate to some extent.

Key words: packet reordering; metric; retransmission;packet loss

1 引言[1]

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，也导致了传输层乱序报文的出现。大量研究表明，网络报文乱序是互联网中普遍存在的现象[1,2]，随着端到端报文延迟的降低和网络传输速率的增快，乱序报文出现的比例同时也呈增长趋势[3]。

在网络报文传输中， 报文的 IP 头和 TCP 头分别有 ID 号和序号，正常顺序到达的网络报文序号 TCP 头的序号应该是逐渐递增的，当到达的序号没有按照递增增长时，那么网络中出现乱序。乱序报文会使位于传输层的协议性能大幅下降，并且对端到端应用程序性能影响重大，有可能会导致网络或者应用端的性能下降。对于 TCP 协议来说，报文乱序可能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[1,4,5]，从而进一步降低了网络性能。对于基于 UDP 的、对延迟敏感的应用程序，例如 VoIP，一个乱序报文在一定时间内没到达也相当于丢包，因此会导致语音质量的下降，同时乱序也会使接收端的缓冲需求增大。在骨干链路上，一小部分乱序甚至会引起吞吐量的严重下降[6]。各种减轻乱序对网络性能影响的方法逐渐被关注，并提出不同的解决方案[7,8]。

本文共分为六节，第二节主要分析乱序产生的原因，第三节分析乱序对网络性能及应用程序带来的影响和危害，第四节对已有乱序测度进行描述，第五节对处理乱序的解决方案进行分析，第六节采用 CERNET 数据对乱序进行实验分析，第七节为总结和展望。

[1] **基金项目：**国家 973 研究计划（2009CB320505），国家自然基金项目（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

2 导致乱序的原因

可能导致乱序的原因很多，依据其发生位置不同可分为发送端产生乱序和网络中产生乱序[9]。

2.1 发送端产生乱序

在发送端进行报文重传会引起乱序的发生[2]。对 TCP 协议来说，当发送端接收到多个重复 ACK 或者出现定时器超时时，发送端会认为报文网络传输中丢失，马上重传该报文。当超时重传发生时，重传的报文到达接收端时比超时期间发送的报文更晚到达，而超时期间发送的报文的序号更大；当发生多个重复 ACK 引起的重传时，重传之前就会有更大序号的报文到达，所以这两种情况必然会导致乱序的出现。

2.2 网络中产生乱序

网络层提供的是无连接、不可靠的报文传输服务，它尽力将报文传送给接收端网络层，再经由接收端网络层向上传送给传输层，整个处理过程中，多路径传输、路由抖动以及路由器内部的并行处理都可能会引起报文乱序。

传输数据时，为了负载平衡，不同报文会在不同路径上进行传输，而不同路径的传输速率及延迟不相等，所以采用多路径传输可能会引起乱序[10]。如图 1 所示，路径 R0-R4、R0-R1-R4、R0-R2-R3-R4 的传输延迟依次递增，当报文 1、2、3 按顺序从 R0 发送时，由于三个报文走的路径上的延迟不同，接收端接收到的数据顺序变为报文 2、3、1，发生了乱序。

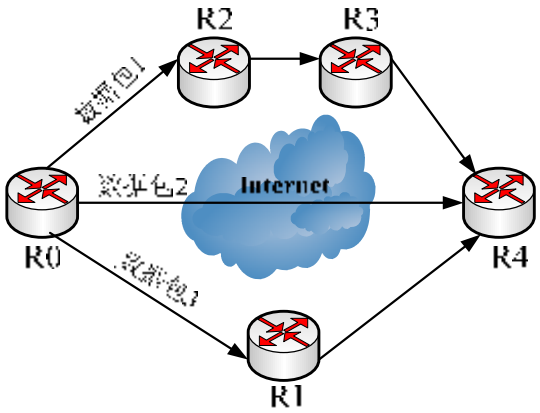

图 1 多路径导致的乱序

为了提高路由器的处理能力和速度，路由器通常采用多处理器的体系架构，即多个处理单元，每个处理单元都是一个独立的微处理器。在路由器内部实现并行处理时，每个单元处理不同的报文。并行处理能提高路由器的性能，但也会由于报文的各异以及资源分配的不同等原因导致转发报文的时间不一致，从而引起报文乱序[10]。J.Bennett 等人[1]对交换中心路由策略进行了升级，并对报文乱序进行测量，测量结果表明报文乱序的概率在升级前后差别很大，这充分说明路由器内部的并行处理会引起报文乱序。

在 QoS 中采用等调度方案也会引起报文乱序[12,13],当对流量超过预定带宽限制时，对于流量超过的报文通常是丢弃或者将它放入低优先级的队列，因此降低优先级的报文可能会比高优先级的报文更晚处理，所以会致使乱序的发生。

在无线网络中，由于网络节点会频繁移动及切换，所以其路由会经常性的变化，当两个节点间的路径传输延迟发生变化时，路由变化前后发送的报文到达目的节点时也可能会产生乱序。

3 乱序带来的影响

面向连接的 TCP 协议，接收端需要对已接收到的报文发送应答，并且对报文在缓冲区中进行排序，排好序后才能传递给应用程序，因此会受到乱序的影响；而对于 UDP，它提供的是无连接、不可靠传输层服务，拥塞和乱序完全由应用层进行控制和处理。

3.1 乱序对 TCP 性能的影响

基于 TCP 的应用程序在进行数据交换前必须要通过三次握手建立连接，数据传输完成后要通过四次握手来断开连接。TCP 在提供可靠传输的同时，其机制也使得它在遇到报文乱序时更加脆弱，其受乱序的四点影响分析如下：

(1) 应用层传来的大数据块按照协议规则被分割成若干个小的数据块，加上 TCP 头部后再传给网络层进行传输。由于应用层数据的分块传输，各条路径的延迟不尽相同，所以数据块到达接收端可能是乱序的。

(2) 当接收端收到一个报文时，会发送一个确认 ACK，告诉发送端接收端期望收到的下一个序

号，所以发送端知道哪些数据已经成功抵达接收端。当报文正常顺序时，报文的序号是递增的，而当序号大的报文到达而序号小的报文没有抵达时，接收端会发送重复 ACK，告诉对方接收端期望接收的报文序号，当收到多个重复 ACK 时，发送端会认为该报文已丢失，并再次发送该 ACK 指定的报文，也就是快速重传，同时也可能会因乱序而致使错误重传。

(3) 当 TCP 报文发生乱序时，接收端对接收到的数据在缓存中重新排序，并按照正确报文顺序交付给应用程序，这可能会导致所有已到达的报文在缓存中等待序号小的报文到达，而当缓存满时，后面的报文就不能再接收了，这样会导致应用程序性能的下降，也可能会影响到资源的利用率。

(4) TCP 协议采用了慢启动和拥塞避免机制来提高传输速率和控制拥塞，慢启动和拥塞避免有两个变量：拥塞窗口(congestion window,cwnd)和慢启动门限(slow start threshold,ssthresh)。当 cwnd 小于等于 ssthresh 时，采用的是慢启动，cwnd 呈指数增长，即每收到一个 ACK 就把 cwnd 加一个报文的大小；当 cwnd 大于 ssthresh 时，进入拥塞避免阶段，cwnd 呈线性增长，即每个 RTT 之内最多增长一个报文的大小，当收到 3 个重复的 ACK 后，ssthresh 设置为 cwnd 的一半，等重传数据成功后，cwnd 设置为 ssthresh 的值，即 cwnd 设置为拥塞之前的一半。也就是说进入拥塞避免阶段，传输速率大大降低，因此如果发送严重的乱序时，TCP 传输性能将大大降低。

Ye Xia 等人[14]分析了 TCP 为缓存乱序报文分配的缓冲区大小以及导致的报文延时，结果表明乱序报文的重新排序对缓存容量有较高的要求，并且由排序引起的报文延时也很显著，当分组较小和吞吐率较高时，乱序对应用层的性能产生较大的影响。

文献[10]研究了 HSTCP，BIC 等几种高速网络中报文乱序特性以及其对高速网中 TCP 性能的影响，在仿真实验中通过设置不同的乱序间隔、乱序延时和乱序块大小三个参数来观察 TCP 性能的变化，研究表明：乱序间隔越小，吞吐量这越小；当乱序间隔小于 20ms 时，BIC 的吞吐量急剧下降；HSTCP 受乱序的影响大于 BIC 的影响，得出报文乱序使得各种高速网络的 TCP 传输性能变得很差。

3.2 乱序对 UDP 性能的影响

UDP 是一种面向数据报的传输层协议，它不同于 TCP，它没有重传和拥塞控制的功能，是一种不可靠无连接的协议，其可靠性由应用程序来保证。

因为 UDP 不提供重传和拥塞控制，也不会在接收端对乱序报文进行重新排序，所以报文在一定时间内还没到达就等同于丢包。文献[15]的分析表明象 VoIP 这样的 UDP 应用，在播放时间点之后抵达的乱序报文将作为丢包处理，这样导致了声音质量低下的现象，并且浪费了网络资源。当前常用的另外一种方法是对延迟不是太大的乱序报文进行缓存，等待乱序报文排好序后再交由应用程序处理。也就是说为了缓存乱序的报文，接收端的缓存使用会随着乱序程度的增大而增加，这将对接收端的资源提出更高的要求，由于资源的限制，过多的乱序报文等同于丢包处理，从而使应用程序性能大大降低。

Colin M 等人[16]研究表明 MPEG-4 采用的预测编码技术对报文的有序性要求很高，乱序对应用程序的性能带来负面影响。他们使用 Windows 媒体播放器作为研究对象，分析乱序对视频质量的影响，发现大量乱序将导致缓冲区溢出，最终导致出现视频帧冻结现象。

4 乱序定义的测度

既然报文乱序会一定程度的降低网络和应用程序的性能，那么定义相关的乱序测度来描述和分析乱序情况是非常必要的。

报文乱序比例是一个经常用来描述乱序的测度[1,2]，然而它的定义略带含糊，例如两个报文序列(1,2,4,5,3)和(1,2,5,4,3)，可以不同的理解的以下情况：

(1) 两种情况的报文 3,4,5 都乱序，
(2) 第一种情况仅报文 3 乱序，第二种情况报文 4,5 乱序，
(3) 两种情况的报文 3,4 出现乱序，

从该例子可以看出，需要一个精确的定义来描述乱序。有一种认为仅晚到的报文是乱序的[11]，一个全面的乱序测度必须要考虑到早到和晚到的报文，并且能够区分两者。RFC4737[11]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[12]中则深层次的对乱序测度进

行了定义，提出了两个测度，没有把乱序仅仅局限于晚到的报文，而是把早到和晚到的报文都归于乱序报文。

4.1 乱序测度的属性

对于乱序测度，关键是能从序号捕获到乱序的数量及范围。根据需求可将乱序测度属性分为两类：一类是必须满足的属性，另一类是期望满足的属性。

其中必须满足的属性包括：

● 捕获乱序：根据报文序号来捕获乱序是最首要和最基本的属性，如果这个属性不满足，则其他属性无从谈起。一个能捕获早到和晚到报文的测度全方位的描述了乱序。

● 低敏感性：一个乱序测度对报文丢失和重复应该有低敏感性，必须只把真正的乱序归类为乱序，不会由于丢包和重复包出现错误的判断为乱序。例如报文序号(1,3,4,5,6)和(1,2,3,2,4,5)都不归类为乱序。

● 有用性：测度不仅仅是一个随着数据流而改变的值，它应该对数据流控制、资源分配、网络性能诊断有用。

期望满足的属性包括：

● 简单易懂：一个理想的测度应该简单易懂，易于计算和使用，对乱序报文进行全面的描述。

● 低复杂度：在实际网络测量中，计算测度的时间和空间复杂度扮演着重要的角色。

● 稳健性：对于不同的网络现象，乱序测度应该有不错的稳健性。例如报文序号(1,5430,2,3,4,5)中 5430 看似是早到的报文，但也有可能是由于报文轮回或者其他原因导致的。

● 可扩展性：对于两个单独的网络，用乱序测度来测量它们后，该测度是否具有预测这两个层叠网络乱序能力是至关重要的。

4.2 乱序测度的定义

RFC4737 和 RFC5236[11, 12]中定义了许多个乱序测度，下面主要描述其中四个比较有代表性的测度。

● 乱序密度 RD(Reorder Density)

RD 根据到达报文的序号，捕获了乱序报文的数量和范围。若发送方在网络上传输报文的序号为(1,2,3…N)，定义 RI(Receive Index)是接收方收到的报文序号，这些序号不指派给丢失的或者重复的报文。当 RI 不等于接收到报文的序号时，称为乱序，它们的差值称为位移 D(Displacement)，当位移为负时，意味着报文提前到达，当为正时，意味着报文延时到达。位移门限 DT(Displacement Threshold)是用来对报文归类为丢包还是重复包的门限，也就是说，一个报文在 DT 内没有到达，将它归类为丢包。位移次数(Displacement Frequency)FD[k]是位移为 k 的报文个数，k 的范围是[-DT , DT]。乱序密度 RD(Reorder Density) 是位移次数 FD[k] 对于 sum(FD[k])规格化后的分布。

由于 RI 跳过了丢失和重复的报文，那么在知道报文丢失之前如何指派 RI 呢，这个问题对于实时数据存在，对于离线数据则不存在，有两种处理方法：

(1)后退法：在报文到达时就计算 RD，当认定一个报文丢失时，纠正 RI 并重计算 RD，此方法只有在报文丢失情况下才会被使用。

(2)延缓法：在报文到达后计算 RD，所以 RI 和 E 都会被正确的指派，最差情况下，RD 的计算滞后于到达报文 DT 个单位。此方法在报文丢失情况下才会被使用。

表 1 是计算存在重复包序列的 RD 例子(DT=2)：

表 1 计算 RD

到达序号	1	3	2	3	4	5
RI	1	2	3	-	4	5
D	0	-1	1	-	0	0
FD[D]	1	1	1	-	2	3

计算 RD：			
D	-1	0	1
FD[D]	1	3	1
RD[D]	0.2	0.6	0.2

● 乱序缓冲区使用密度 RBD(Reorder Buffer-occupancy Density(RBD)

RBD 是从乱序恢复的角度出发的，由于缓冲区的存在，它可以缓存乱序报文，RBD 描述了缓冲区的占用情况。期望的报文 E(Expected Packet)是指，所有小于 E 的报文都已经到达或者丢失，序号为 E 的报文被期望到达。缓冲区使用大小 B(Buffer Occupancy)是指缓冲区中存储报文的个数。缓冲区使用门限 BT(Buffer-Occupancy Threshold)是用来从乱序中恢复的最大缓冲区大小。就像 DT 对于 RD 一样，BT 是用来从 RBD 计算中分类出丢包和重复包情况。缓冲区使用次数 FB[k](Buffer-Occupancy

Frequency)是指使用了大小为 k 的缓冲区出现的次数。k 的范围是[0,BT]。RBD 是缓冲区使用次数 FB[k]对于 sum(FB[k])的规格化，即 RBD[k]=FB[k]/sum(FB[k])，k 的范围是[0,BT]。

当由于乱序致使缓冲区到达 BT，期望的报文还没到达则认为它丢失了。如果到达报文的序号小于期望的序号或者在缓冲区中有同样序号的报文，则认为到达的数据是重复包。

表 2 是计算存在丢包序列的 RBD 例子(BT=3)：

表 2 计算 RBD

到达序号	1	2	4	5	6	7
E	1	2	3	3	3	7
B	0	0	1	2	3	0
FB[B]	1	2	1	1	1	3

计算 RBD：

B	0	1	2	3
FB[B]	3	1	1	1
RBD[B]	0.5	0.167	0.167	0.167

● 乱序范围(Reordering Extent)

乱序范围和 n-Reordering 都是基于延迟的测度，也就是只有晚到的报文才认为是乱序的。乱序范围是晚到的报文和早到的且有更大序号的报文之间的最大距离。假定发送方报文为(1, 2 … N)，N’为接收方接收到的报文数目，因为乱序范围是假定重复的报文已经被剔除，所以必然有 N’<N。假定 s[1],s[2],…s[N’]代表接收方按顺序接收到的报文序号，假定一个乱序报文的序号是 s[i]，若存在一组 j(1<=j<i)，且 s[j]>s[i]，那么报文 s[i]的乱序范围 e 等于 i-j，其中 j 是满足 s[j]>s[i]的最小值。表 3 是计算到达序号为(1, 2, 3, 4, 7, 5, 6, 8)乱序范围的例子：

表 3 计算乱序范围

s[i]	1	2	3	4	7	5	6	8
i	1	2	3	4	5	6	7	8
e	-	-	-	-	-	1	2	-

其中，对于接收到的序号 5，s[6]=5，满足 s[j]>s[i]的是 s[5]=7；那么对于序号 5，e=6-5=1；对于接收到的序号 6，s[7]=6，满足 s[j]>s[i]最小的是 s[5]=7，那么对于序号 6，e=7-5=2。

● n-Reordering

若接收到的报文序号 s[i](n<i<=N)是 n-reordered，当且仅当对于任意 j, i-n<=j<i，s[j]>s[i]都成立。也就是说，对于到达的序号 s={1,2,3,7,8,9,4,5,6}，报文 4,5,6 是乱序的，但只有报文 4 是 n-reordered，其中 n=3。

假定 m 是样本中 n-reordered 报文数目，那么这个样本的 n-reordering 度是 m/N’，表 4 是计算到达序号为(1,2,3,4,7,5,6,8)n-Reordering 的例子：

表 4 计算 n-Reordering

s[i]	1	2	3	4	7	5	6	8
i	1	2	3	4	5	6	7	8
n-Reordering	0	0	0	0	0	1	0	0

其中，对于接收到的序号 6，s[7]=6，不满足对于任意 j, i-n<=j<i，s[j]>s[i]都成立。所以 1-reordering 的度为 1/8。

4.3 乱序测度的属性比较

分析比较上述四个乱序测度发现，对于乱序范围和 n-Reordering 不能发现重复包和丢包，也就是对丢包和重复包很敏感，且它们是基于延迟报文的测度，发现不了早到的报文。我们还发现，RD 满足所有必须和期望的属性要求，其他测度只满足部分属性要求[13]。具体请参见表 5，其中√代表满足属性，×代表不满足属性，**£**代表满足部分属性：

表 5 乱序测度属性比较

属性/测度	RD	RBD	Extent	N-Reordering
捕获乱序	√	**£**	**£**	×
对丢包和重复包的低敏感性	√	**£**	**£**	**£**
有用性 TCP 控制	√	×	×	√
有用性-资源分配	√	√	×	×
简单易懂	√	√	**£**	**£**
空间复杂度	常数	常数	$O(N)$	$O(N)$
测度计算复杂度	$O(N)$	$O(N)$	$O(N^2)$	$O(N^2)$
稳健性	√	×	×	√
对于层叠网络的课扩展性	√	×	×	×

5 处理报文乱序的解决方案

针对网络报文对于网络传输性能以及应用程序的影响，众多学者提出了不同的解决方案，可以概括为下列几种：

5.1 增大快速重传的门限值

大多数针对乱序的解决方案都通过增大快速重传的门限值(dupthresh)来实现[10,17]，把 TCP 快速重传的默认门限值 3 调整为更大的值，以阻止不必要的快速重传和快速恢复。因此可以改变 dupthresh 的取值来处理乱序，主要有三种 dupthresh 设定算法：

（1）使用固定参数 K 来增大 dupthresh 的取值。

该算法的主要思想是：当乱序报文出现时，将 dupthresh 增大为 dupthresh+K。所以接收端要能检测乱序事件。检测方法如下：如果报文在 dupthresh 之后到达，认为是丢失的报文到达，即为一个乱序事件。同时将 dupthresh 增大为 dupthresh+K。之后将按照新的 dupthresh 进行丢包判断。这种方法的优点是在实现上比较简单，缺点也很明显，就是接收端可能需要多次才能将 dupthresh 设定为合理值。而且整个算法的性能和收敛过程依赖于 K 的选取。

（2）根据乱序报文数量动态设定 dupthresh。

该算法的主要思想是：当接收方检测到丢包后开始记录乱序报文的数量，称为乱序报文数量 M，直到丢失的报文抵达。此时将 dupthresh 修改为(dupthresh+M)/2。

此方法的优点是收敛性优于第一种算法，缺点是可能由于某个过大的乱序报文数量造成 dupthresh 过大，从而影响网络性能。

（3）根据乱序报文数量利用加权平均算法设定 dupthresh。

针对以上两种算法的缺陷，Leung 等人提出使用指数加权移动平均算法(EWMA:Exponentially Weighted Moving Average)动态设定 dupthresh[17]。类似于第二种算法，当接收方检测到报文丢失时，记录乱序报文的数量 M，直到丢失的报文到达。此时可根据以下公式计算平均乱序报文数量：

if L > avg

avg = (1-α)*avg +α*L

else

avg = (1-α*x)*avg + (α*x)*L

其中，α 为 EWMA 因子，通常取 1/3，x 为乘性因子，通常取 4。然后将 dupthresh 设定为 avg。这种算法的优点是 dupthresh 可以根据网络状态动态改变，并且避免了第二种算法中某个过大的乱序报文数量对 dupthresh 造成过大影响。缺点是接收方需要增加统计变量，且经常更新乱序报文数量，也就是需要使用接收方的资源，这样可能会对接收方的性能造成一定影响。

5.2 Eifel 算法

R.Ludwig 和 R.Katz 提出 Eifel 算法来减少因为乱序导致的超时和重传对网络的影响[17]。Eifel 的原理如图 2 所示，发送方在 T1 时刻发生报文 S 的同时，将时间戳 T1 插入 TCP 头部。在 T2 时刻发送方由于检测到 S 丢失，所以重传 S，并执行拥塞控制算法。两次重传的 S 包含不同的时间戳，当接收方收到原始 S 后，发送带时间戳 T1 的 ACK。当 ACK 抵达发送方时，发送方发现此 ACK 的时间戳 T1 小于存储于重传 TS 变量中的 T2，由此判断此重传为伪重传，并将拥塞窗口和慢启动门限值恢复到伪重传前的值，如同没有错误重传一样。

该算法的优点是能够在短时间内检测出伪重传，从而避免了后续不必要的重传和拥塞。缺陷是该算法仅仅是在检测到伪重传时避免了拥塞窗口的减少，并没有减少伪重传报文数量，所以不能减少由于伪重传带来的资源消耗。

5.3 DSACK 机制

Floyd 等人提出使用重复选择确认机制(DSACK)来检测乱序，此方法是通过扩展 TCP 协议的 SACK 选项来克服乱序报文造成的影响[17]。该算法的原理如图 3 所示，假定发送方发送顺序是 S1-S4，由于网络原因造成 S1 乱序，它在 S4 到达之后抵达接收方。所以接收方收到 S2，S3，S4 后都会产生对 S1 的重复应答，当收到 3 个重复应答后，发送方马上重传 S1。当重传的 S1 到达接收方后会产生一个对于 S5 的重复应答，同时 SACK 信息会指明此重复应答是由 S1 引起的。当此应答抵达发送方后，发送方就知道刚才对于 S1 的重传是伪重传。

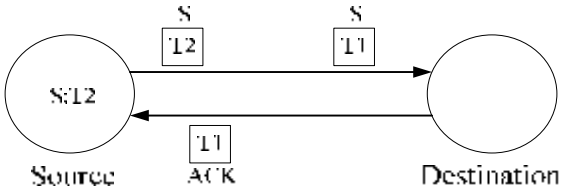

图 2 Eifel 算法示意图

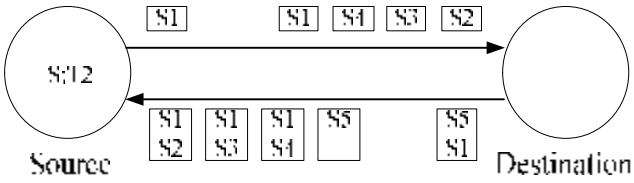

图 3 DSACK 算法示意图

DSACK 算法要求发送方检测到丢包时的拥塞窗口和慢启动门限值等信息，发送方根据 DSACK 信息检测到伪重传时，将拥塞窗口和慢启动门限值恢复到伪重传前的状态。此方法没有增加过多的消耗，但它没有解决 ACK 丢包和重复包问题。若含有 DSACK 信息的 ACK 丢失，则接收方无法恢复。而且因 DSACK 信息在丢失恢复结束后才到达发送方，所以发送方在丢失恢复阶段无法检测到伪重传。

5.4 缓冲区机制

由于 UDP 协议没有拥塞控制，所以基于 UDP 的应用程序要减少乱序报文造成的影响，需要从应用程序出发，比如利用缓冲区机制，此方法在网络多媒体应用程序中使用广泛。使用缓冲区机制可以让应用程序从乱序中尽快得到恢复。但是资源毕竟是有限的，所以如何有效的对缓冲区进行使用是一个关键的问题。一般有两种使用缓冲区的方法：

(1) 从时间上限制。

在应用程序端可以开辟一个缓冲区，用于存储由于乱序而早到的报文。定义一个时间门限，比如 1 分钟，一旦使用缓冲区的累积时间达到该门限，则乱序的报文认为丢包。

(2) 从空间上限制。

可以定义一个大小固定的缓冲区，专门用来存储由于乱序而早到的报文。当缓冲区存满一定大小后，还没有抵达的乱序报文则认为丢失。

6 实验分析

实验数据是 CERNET 华东北地区网络中心在不同时段获得的 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。实验对 2010 年 4 月 12 日 23：55：16 和 2010 年 4 月 20 日 13：55：16 为开始时间的 5 分钟 trace 进行研究。

首先，对 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，将相同五元组的报文出现间隔时间超过 16 秒的定义为超时，超时后重新组新流。一个流包含两个方向交互的流，即正向流和反向流，正向流的源 IP、目的 IP、源端口、目的端口等于反向流的目的 IP、源 IP、目的端口、源端口。对两次测量的流信息进行统计，结果如表 6 所示：

表 6 trace 信息统计表

测量	总流数	总报文个数	总字节数
1	1062040	23598739	14.23 GB
2	2075883	45082259	28.08 GB

其次，统计每个流的相关信息，包括：所含报文个数、所含报文总字节数、乱序率、测量点前丢包率、测量点后丢包率。其中乱序率和丢包率的计算仅针对带有数据传输的 TCP 包，因为非数据传输的 TCP 包仅有 IP 头和 TCP 头，这种情况下 TCP 头里面的 Sequence 字段不会发生改变，所以不适合进行乱序率和丢包率计算。一个流的乱序率 = 乱序报文个数 / 总报文个数，丢包率 = 丢失报文个数 / 总报文个数。

最后对结果进行分析统计，若某正向流或反向流包含的报文个数小于 300，则该流不在统计范围之内，因为报文个数太少影响计算的准确率。统计结果如表 7 所示：

表 7 流信息统计表

		trace1	trace2
满足条件的流数		5529 /0.52%	10298 /0.50%
正向流	报文数	9124028	15595113
	字节数	5.30GB/37.3%	10.97GB/39.1%
	平均乱序率	5.800%	5.773%
	测量点前平均丢包率	0.419%	0.442%
	测量点后平均丢包率	0.161%	0.166%
反	报文数	8395393	14425056
	字节数	6.64GB/46.7%	9.97GB/35.5%
	平均乱序率	4.846%	6.203%

向流	测量点前平均丢包率	0.331%	0.482%
	测量点后平均丢包率	0.171%	0.245%

通过上表可以看出，大约 80%数据量的流满足统计条件，每条正向流或反向流的乱序率大概在 5%左右，在测量点前和后的丢包率在 0.4%和 0.2%左右。也就是说乱序的发生在一定程度上造成了网络报文丢失，会影响传输层和应用层的性能。

7 总结

论文从乱序报文的成因进行分析，简述了网络中报文乱序可能造成的影响，分析一个全面的乱序测度应该具有的属性，评价现有四种典型乱序测度的优缺点，分析比较了报文乱序的解决算法，并对这些算法进行分析和比较。最后对 CERNET 华东北地区网络中心采集的数据进行组流分析，每条流的乱序率大约在 5%左右，测量点前和后的丢包率大约在 0.4%和 0.2%左右，乱序率在一定程度上影响着丢包率。综上所述，该领域仍存在不少有待解决的问题，如通过乱序来分析网络性能和对网络性能进行评估是今后研究的方向。

参考文献

[1]J.C.R. Bennett, C. Partridge, N. Shectman, Packet reordering is not pathological network behavior, IEEE/ACM Trans. Networking(1999) 789–798.

[2] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, D.Towsley, Measurement and classification of out-of-sequence packets in Tier-1 IP backbone, Proc.IEEE INFOCOM, Mar. 2003, pp. 1199–1209

[3] L. Gharai, C. Perkins, T. Lehman, Packet reordering, high speed networks and transport protocol performance, Proc. IEEE 13th IntlConf. On Computer Comm, and Networks, ICCCN 2004, Oct. 2004,pp. 73–78.

[4] E. Blanton, M. Allman, On making TCP more robust to packet reordering, ACM Comput. Comm. Rev. 32 (1) (2002) 20–30.

[5] X. Luo, R. Chang, Novel Approaches to End-to-end Packet Reordering Measurement, Proc. ACM/USENIX Conf. Internet Measurement, October 2005, 2005.

[6] M. Laor, L. Gendel, The effect of packet reordering in a backbone link on application throughput, IEEE Network, September/October 2002 (2002) 28–36.

[7] PIRATLA N., JAYASUMANA A., BARE A., BANKA T.: 'Reorder bufferoccupancy density and its application for measurement and evaluation of packet reordering', Comput. Commun., 2007, 30, (9), pp. 1980–1993.

[8] 赵丽莉，孙伟，TCP 协议乱序数据包处理算法综述，软件工程师，2010(7)

[9] 陈均华，多路径传输中乱序与负载均衡研究

[10] Jie Feng，Zhipeng Ouyang，Lisong Xu，Byrav Ramamurthy. Packet reordering in high-speed networks and its impact on high-speed TCP variants．Computer Communications．2009，32(1)：62-68．

[11] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, J. Perser, Packet reordering metrics, RFC 4737.

[12] A．Jayasumana，N．Piratla，T．Banka，et a1．Improved Packet Reordering Metrics．RFC5236．June 2008．

[13] N.M.Piratla，A.P.Jayasumana，A.Bare，A Comparative Analysis of Packet Reordering Metrics

[14] Ye Xia，David Tse．Analysis on Packet Resequencing for Reliable Network Protocols．Performance Evaluation 61 (2005) 299－328

[15] Bare A.A，Jayasumana A.P.and Piratla N.M.. On Growth of Parallelism within Routers and Its Impact on Packet Reordering．15th IEEE Workshop on Local& Metropolitan Area Networks．Princeton：IEEE Press，2007：145-150．

[16] Colin M．Arthur，Demessie Girma，et a1．The Effects of Packet Reordering in a Wireless Multimedia Environment．1st International Symposium on Wireless Communication Systems．UK：IEEE Press，2004：453-457．

[17] K.-C. Leung, V.O.K. Li, D. Yang, An overview of packet reordering in transmission control protocol (tcp): problems, solutions, and challenges, in:IEEE Transactions on Parallel and Distributed Systems (IEEE TPDS),2006.