

粒度自适应的多径流量分割算法

王鹏, 兰巨龙, 陈庶樵

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

摘要: 针对多径传输存在的负载均衡和分组乱序之间的矛盾, 提出了一种粒度自适应的多径流量分割算法。该算法在“流分片”的基础上, 通过对流量特性的分析, 将网络流量区分为正常流和强势流, 基于“区分服务”的思想, 不同流量分别采用适应其流量特性的流片分割粒度。理论分析和仿真结果表明, 在考虑路径时延估计误差的情况下, 该算法在降低分组乱序的同时具有良好的负载均衡性能。

关键词: 多径; 流量分割; 负载均衡; 分组乱序; 分割粒度

中图分类号: TP393.01

文献标识码: A

Multipath traffic splitting algorithm based on adaptive granularity

WANG Peng, LAN Ju-long, CHEN Shu-qiao

(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: Aiming at the dilemma between load balancing and packet reordering in multipath routing, an adaptive dynamic multipath traffic splitting (ADTS) algorithm was proposed. The algorithm classified Internet flows into two categories: the aggressive and the normal through analyzing the characters of the traffic, and applies different splitting granularities to the two classes of flows based on the idea of differentiated service. The theoretical analysis and simulation results show that ADTS algorithm reduces the number of reordering packets while has better performance of load balancing.

Key words: multipath; traffic splitting; load balancing; packet reordering; splitting granularity

1 引言

网络用户的不断增加和网络业务的快速发展对网络传输性能的要求越来越高, 千变万化的网络业务和多种多样的用户需求使网络流量的动态和随机突发特性更加明显^[1], 传统的单径传输机制已不能满足网络的发展需求。多径传输机制能够充分利用网络路径的多样性, 在提高传输可靠性的同时还能有效地进行负载均衡, 降低拥塞, 提高资源利用效率^[2]。

多径传输在带来好处的同时也引入了一些问题。首先, 多径传输需要在中间节点进行动态的流量分割。千变万化的网络业务使网络流量和环境的动态特

性更加突出, 如网络流量的随机性和突发性造成路径流量不断变化, 这就要求多径流量分割算法能够适应路径流量变化动态进行流量分配; 其次, 多径流量分割存在分割粒度和分组保序的两难困境^[3]。以数据分组的粒度进行多径流量分割能够快速准确地按比例分配流量, 达到理想的负载均衡效果, 但由于各路径传输时延的不同可能会引起 TCP 流中的数据分组乱序, 破坏 TCP 的拥塞控制机制, 进而降低业务性能^[4], 而对于大量的 UDP 多媒体应用如 VOIP 等, 分组乱序将影响服务质量^[5]; 以数据分组头部中“五元组”定义的流粒度进行多径流量分割能够避免分组乱序, 但在动态的网络环境下流量分配准确性较差, 降低了端到端的吞吐率; 以 TCP 流中数据分组的突

收稿日期: 2013-09-15; 修回日期: 2013-12-20

基金项目: 国家重点基础研究发展计划(“973”计划)基金资助项目(2012CB315901, 2012CB315906); 国家高技术研究发展计划(“863”计划)基金资助项目(2011AA01A103)

Foundation Items: The National Basic Research Development Program of China (973 Program) (2012CB315901, 2012CB315906); The National High Technology Research and Development Program of China (863 Program) (2011AA01A103)

发作为分割粒度进行多径流量分配, 试图在分组保序和流量分配准确性之间到达平衡, 理想情况下可以在保证数据分组不乱序的同时提高负载均衡的精度, 但受路径传输时延估计误差的影响, 负载均衡性能和数据分组乱序同样存在矛盾。因此, 需要研究粒度自适应的多径流量分割机制, 以满足各种网络流量情况下的多径流量分配。

2 相关工作

当前, 多径流量分割算法按照分配粒度的不同可以分为3类: 以数据分组粒度分割的算法^[6]、以流粒度分割的算法^[7]和以TCP流中数据分组的突发作为分割粒度的算法^[8-10]。以数据分组作为分割粒度的算法没有考虑数据分组之间的联系, 容易引起数据分组的乱序, 进而导致业务性能下降; 以流作为分割粒度的算法流量分配的准确性较差, 导致网络端到端的吞吐率较低; 以TCP流中数据分组的突发作为分割粒度的算法把路径估计时延和同一流中数据分组的到达时间间隔结合起来, 在保证数据分组按序到达的同时提高了负载均衡效果。

文献[8]提出了一种流分片的多径流量分割 (FLARE, flowlet aware routing engine) 算法。FLARE 算法的主要思想是利用同一流中相邻数据分组到达时间间隔、多径间的最大传输时延之差与数据分组乱序之间的关系, 当流量在传输时延不同的两条路径上传输时, 如果同一流相邻数据分组到达时间间隔大于并行路径的传输时延之差, 那么无论当前分组选择哪一条路径, 都不会早于前一个分组到达目的节点, 即不会引起乱序。FLARE 将多径最大最小时延之差 (MMDD, max min delay difference) 作为流分片的依据, 同一流中的相邻分组到达间隔大于 MMDD 可以在不同路径上传输, 小于 MMDD 只能在同一路径上传输, 这样流就被分割成流片, 流片中的数据分组在相同路径上传输可以保证分组按序达到, 不同的流片可以在不同的路径上传输, 因此 FLARE 算法在保证数据分组按序

到达的同时比以流作为粒度进行流量分配的算法更细粒度地实现路径间的负载均衡。

相比于以数据分组和流作为分割粒度的算法, FLARE 既能保证分组的有序性又能具有较好的负载均衡效果。但当 MMDD 较大而同一流相邻分组到达间隔较小时, FLARE 的分片粒度接近于流粒度, 负载均衡效果较差, 而一般流量较大的流相邻分组到达间隔较小^[9]。针对这一问题, 文献[9]提出了一种基于游标的流量分割 (TSBN, traffic splitting based on nonius) 算法。TSBN 充分利用同一流中相邻分组的到达间隔压缩流量分割粒度, 把当前路径时延与相邻分组到达时间间隔之差作为游标, 游标作为选择路径的时延基线, 时延大于游标的路径传输数据分组不会引起乱序, 游标会随着路径时延或相邻分组到达间隔的不同而动态地滑动, 通过游标的动态滑动可以很好地实现负载均衡。理想情况下, TSBN 算法在保证分组有序到达的同时进一步降低了流量分割的粒度, 负载均衡效果更好, 但现实中受路径时延估计误差的影响, TSBN 算法具有较高的分组乱序度, 实际应用中会对业务性能造成影响。

3 问题描述

如图1所示, 假设流量以速率 R 到达位于路由器中的流量分割模块, 流量分割模块根据分割算法的分割粒度把流量分割成流量单元, 路径选择模块根据选路策略把流量单元均衡地分配到 N 条不相关的并行路径上。根据多径流量转发的过程来看, 本文要研究的多径流量分割问题可表述为: 给定 N 条不相关的并行路径、流量分割向量 $\vec{F} = (F_1, F_2, \dots, F_N)$ 以及由一系列数据分组 P_i 组成的流量 P , 其中, $F_i \in [0, 1]$ 且 $\sum_{i=1}^N F_i = 1$, 每一个数据分组 P_i 都属于某一个流 f_i 。在保证流内数据分组不乱序的情况下, 寻找一种数据分组在多径之间的分配方案, 使第 i 条路径的流量速率等于 $F_i R$, 达到流量在多条路径之间的负载均衡。本文研究的多径流量分割问题只关注具

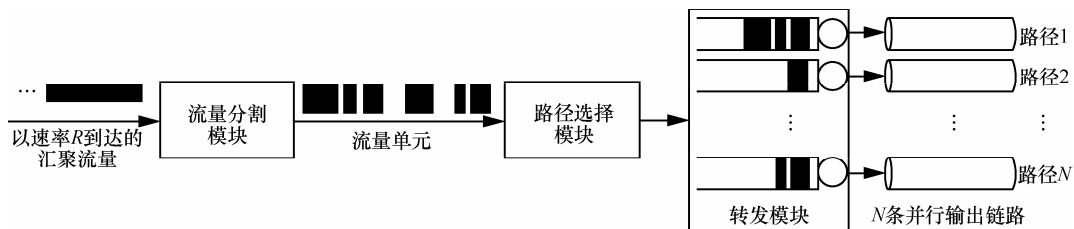


图1 多径流量转发机制组成

体的流量分割算法，至于流量分割向量 \vec{F} ，可以根据路径带宽情况进行静态配置也可以根据网络中路径拥塞情况由相应的路由协议进行动态提供。为便于描述算法，首先定义几个下文将涉及到的基本概念。

定义 1 强势流 (aggressive flow)。强势流是指数据分组到达间隔较小的高速流。强势流包括流中数据分组数量多而且速率高的大流和流中数据分组数量少但是速率高的小流，这里所说的大流是引起长期负载失衡的主要原因，而小流是引起短期负载失衡的主要原因。

定义 2 正常流 (normal flow)。正常流是指除强势流以外的网络流量，这部分流是网络流量的主力。

定义 3 流分片 (flow slice)。流分片是指根据流内相邻数据分组之间的到达时间间隔与流分片门限 δ 的大小关系把流分为不同的流片，流片内部的相邻数据分组之间的时间间隔小于等于分片门限。流片主要是由于 TCP 协议传输数据分组时在 RTT 或低于 RTT 时间尺度上的突发性造成的^[8,11]。

定义 4 负载失衡度 (degree of load imbalance)。负载失衡度是指各条路径负载分配偏差度之和的均值，如式(1)所示。其中， N 表示路径的个数， $|F'_i - F_i|/F_i$ 表示各条路径负载分配偏差度， F'_i 表示路径 i 实际分配的负载量， F_i 表示路径 i 期望分配的负载量， F_i 越大表明路径 i 可用带宽越大，优先选择 $|F'_i - F_i|$ 大的路径能够保证负载尽可能均衡。负载失衡度是衡量负载均衡性的指标，值越大说明负载均衡性越差。

$$DLI = \frac{1}{N} \sum_{i=1}^N \frac{|F'_i - F_i|}{F_i} \quad (1)$$

定义 5 分组乱序度 (degree of packet reordering)。分组乱序度是指发生乱序的分组数与到达目的端总的分组数之比。正常情况下，数据分组的到达时间戳是以单调递增的顺序排列，如果出现较小时时间戳的数据分组晚于较大时间戳的数据分组到达，则表明出现了乱序。分组乱序度是衡量流量分割算法保序性能的指标，值越大说明保序性能越差。

4 粒度自适应的多径流量分割算法

负载失衡度和分组乱序度是衡量流量分割算法性能的 2 个重要指标。按流片进行负载均衡在理论上能够保证数据分组按序到达，但受路径时延估

计误差的影响，负载失衡度和分组乱序度又是存在矛盾的 2 个指标，降低负载失衡度的方法是尽可能细粒度地切分流，但过细的流片容易受路径时延估计误差的影响不可避免地具有较高的分组乱序度。因此，本文提出了一种粒度自适应的动态流量分割算法 (ADTS, adaptive dynamic traffic splitting algorithm)。ADTS 对网络流量进行“区别对待”，不同的流量采用不同的粒度分割，在将路径时延估计误差引起的数据分组乱序控制在较低水平的同时尽可能地提高了负载均衡性能。

4.1 算法思想

理想情况下，现有的基于流片的多径流量分割算法在保证数据分组按序到达的同时又能提供很好的负载均衡性能。但在真实网络环境中，受路径时延估计误差的影响，特别是在大量突发且数据分组到达间隔较小的网络流量情况下，基于流片的多径流量分割算法要么存在分割粒度较大时负载均衡效果不理想的问题，要么存在分割粒度过小时出现大量数据分组乱序的问题。针对这一两难困境，引入了“区分服务”的思想，针对不同的网络流量采取不同粒度的流量分割方法，从而达到了降低数据分组乱序的同时尽可能地提高负载均衡性能的效果。

假设多路径的最大最小时延为 D_{\max} 和 D_{\min} ，当前传输路径的时延为 D_r ，多径中任意路径的时延为 D_x ，当前数据分组和上一个数据分组的到达时间分别为 t_r 和 t_p 。ADTS 首先将到达的网络流量分为强势流和正常流，强势流和正常流的定义已在上节中给出。对于正常流，当 $t_r - t_p > D_{\max} - D_{\min}$ 时，对流进行分片，选择多径中剩余额度最大的路径进行流量均衡，正常流采用这种粒度的流量分割方法已能保证流片在多径之间的负载均衡性能，而且这时正常流受路径时延估计误差影响引起的分组乱序度较小；对于强势流，当存在路径时延 D_x 满足 $t_r - t_p > D_i - D_x$ 时，对强势流进行分片，并选择满足要求的 D_x 集合中剩余额度最大的路径进行流量均衡，这样就能保证以较细的粒度对强势流进行分片，提高了强势流的负载均衡性能。

4.2 算法描述

ADTS 算法要维护流分类表、流分片表和路径信息表，流分类表记录当前检测到的强势流标识，流分片表记录流标识、该流前一个分组的达到时间和传输路径，路径信息表记录路径的时延、期望额

度和实际分配额度。ADTS 算法采用窗口检测法检测到达的网络流量，并将检测到的强势流标识记录到流分类表中，后续到达的数据分组首先查找流分类表，如果属于强势流，查询流分片表中的流标识并选择满足 $t_r - t_p > D_i - D_x$ 的路径形成的可用路径集合中剩余额度最大的路径传输该数据分组；如果属于正常流，查询流分片表中流标识并判断当前数据分组的传输时延与前一个数据分组的传输时延之差是否大于 MMDD，如果大于则进行流分片并选择所有路径中剩余额度最大的路径进行传输。算法 1 给出了 ADTS 的伪码描述，其具体步骤如下。

步骤 1 利用窗口检测法检测到达的网络流量，构建流分类表 FCT ， FCT 记录强势流的流标识，流标识通过对数据分组五元组进行散列计算得到。

步骤 2 到达的数据分组查询流分类表 FCT ，如果所对应的流标识在 FCT 中，则该数据分组属于强势流分组；否则该数据分组属于正常流分组。

步骤 3 查询流分片表 FST ，如果数据分组对应的流标识在 FST 中不存在，则创建一个新的 FST 表项，并选择所有路径中剩余带宽最大的路径作为传输路径。

步骤 4 如果数据分组对应的流标识在 FST 中存在，判断该数据分组为强势流分组还是正常流分组，对于强势流分组，根据 $t_r - t_p > D_i - D_x$ 计算可用路径集合，并从可用路径集合中选出剩余带宽最大的路径作为传输路径；对于正常流分组，根据 $t_r - t_p > D_{\max} - D_{\min}$ 判断是沿着上一个分组的传输路径传输还是选择所有路径中剩余带宽最大的路径作为传输路径。

步骤 5 更新路径信息表 PIT 和流分类表 FST 。

算法 1 ADTS 算法

Input: PIT : the path information table $PIT = \{P_1,$

$P_2, \dots, P_i\}$, thereinto, $P_i = (D_i, F_i, F_i')$

Output: P_o : the path that is selected as output

- 1) $FCT \leftarrow 0$; // FCT : 流分类表
- 2) $FST \leftarrow 0$; // FST : 流分片表
- 3) $f_i = \text{hash}(\text{five tuples})$; // f_i 表示流标识
- 4) $\text{detect}(\text{flow})$; // 检测强势流
- 5) $\text{construct}(FCT)$; // 构建 FCT 表
- 6) If $f_i \in FCT$
- 7) $\text{query_fst}(f_i)$;
- 8) if f_i exists in FST

9) $P_x = \{P_j | t_r - t_p > D_i - D_j\}$ // t_r 表示当前分组到达时间， t_p 表示前一个分组到达时间， P_x 表示时延满足要求的可用路径集合

10) $P_o = \text{select}(P_x)$; // 选择可用路径集合中剩余额度最大的路径作为输出路径

11) $\text{update}(FST)$; // 更新 FST 表项

12) else

13) $P_o = \text{select}(P_i)$ // 选择所有路径中剩余额度最大的路径作为输出路径

14) $\text{creat_entry}(f_i)$; // 在 FST 中创建新的表项

15) else

16) $\text{query_fst}(f_i)$;

17) if f_i exists in FST

18) if $t_r - t_p > D_{\max} - D_{\min}$ // D_{\max} 、 D_{\min} 分别表示所有路径时延的最大和最小值

19) $P_o = \text{select}(P_i)$ // 选择所有路径中剩余额度最大的路径作为输出路径

20) $\text{update}(FST)$;

21) else

22) $P_o = \text{select}(P_r)$ // 选择上一个分组的传输路径作为输出路径

23) $\text{update}(FST)$;

24) else

25) $P_o = \text{select}(P_i)$; // 选择所有路径中剩余额度最大的路径作为输出路径

26) $\text{creat_entry}(f_i)$; // 在 FST 中创建新的表项

27) $\text{update}(PIT)$; // 更新路径信息表信息

28) end

5 性能分析

本节对 ADTS 算法的负载均衡性能和保序性能进行理论分析。通过分析文献[8]中提出的路径负载失衡度边界条件，负载失衡度大于某一个值的概率定义如下：假设路径 i 的期望额度为 f_i ，任意时间间隔 $[0, t]$ 内到达的流数目为 $N(t)$ ，路径 i 的负载失衡度为 $DLI_i(t)$ ，负载失衡度大于任意值 ξ 的概率为

$$P(|DLI_i(t)| > \xi) < (\gamma^2 + 1) / (4\xi^2 E[N(t)]) \quad (2)$$

其中， $\gamma = \sigma / \mu$ 表示变异系数，为标准差与均值之比。从式(2)可以看出，失衡概率受 γ 和 $E[N(t)]$ 的影响。

定理 1 ADTS 算法的负载均衡性能具有确定的上下限，即负载失衡度 DLI_{ADTS} 满足

$$DLI_{TSBN} < DLI_{ADTS} < DLI_{FLARE} \quad (3)$$

证明 由式(2)可知，负载失衡度大于任意值的概率受 γ 和 $N(t)$ 的共同影响。对 γ 值来说，流分片的粒度越细流片的数量就越多，相应的 γ 值就越小，ADTS 算法的切分粒度介于 TSBN 和 FLARE 算法之间，因此，ADTS 的 γ 值也介于 FLARE 和 TSBN 算法之间，即 $\gamma_{TSBN} < \gamma_{ADTS} < \gamma_{FLARE}$ 。对于 $N(t)$ 来说， $N(t)$ 表示 $[0, t]$ 时间间隔内到达的流或流片数量，ADTS 算法正常流的切分粒度与 FLARE 相同，强势流的切分粒度与 TSBN 相同，整体的流片数量介于 FLARE 与 TSBN 算法之间，即 $N_{FLARE}(t) < N_{ADTS}(t) < N_{TSBN}(t)$ 。结合式 (2) 可知，ADTS 算法的负载均衡性能介于 FLARE 和 TSBN 算法之间。

理论上，同一流中 2 个相邻的数据分组从传输时延小的路径切换到传输时延大的路径不会引起数据分组乱序，相反的情况下，只要保证相邻数据分组的到达时间间隔大于两路径时延之差也不会出现乱序。ADTS、FLARE 和 TSBN 在路径时延估计准确的情况下都不会出现数据分组乱序，但在现实的网络中受测量手段的限制存在路径时延估计误差，路径时延估计误差会对算法的保序性能产生影响。

定理 2 ADTS 算法的分组乱序度 DPR_{ADTS} 具有确定的上下限，即 DPR_{ADTS} 满足

$$DPR_{FLARE} < DPR_{ADTS} < DPR_{TSBN} \quad (4)$$

证明 假设路径 i 和 j 的传输时延之差为 D_{ij} ，流片的切分门限为 δ_{th} ，路径时延估计误差为 ε 。由

文献[3]中的分析可知，基于流分片的负载均衡算法从路径 i 切换到路径 j 时数据分组乱序的概率 Ω 可以表示为 δ_{th} 和 D_{ij} 的函数，即

$$\Omega(\delta_{th}, D_{ij}) = P[\delta_{th} < X < D_{ij}] = \begin{cases} H(D_{ij}) - H(\delta_{th}), & \delta_{th} < D_{ij} \\ 0, & \delta_{th} \geq D_{ij} \end{cases} \quad (5)$$

其中， X 表示相邻数据分组的到达时间间隔。从式 (5) 可以看出切分粒度的选取直接决定了数据分组乱序的概率大小，由于路径估计时延误差 ε 的存在，切分粒度 δ_{th} 即使选取多径中的最大最小路径时延之差，也会存在一定的乱序概率。如果选择更细的切分粒度如 TSBN 算法的切分粒度，数据分组乱序的概率更大。ADTS 算法的切分粒度下限为 TSBN 算法的切分粒度，上限为 FLARE 算法的切分粒度。因此，ADTS 算法的数据分组乱序概率 $\Omega_{ADTS}(\delta_{th}, D_{ij})$ 的上限为 $\Omega_{TSBN}(\delta_{th}, D_{ij})$ ，下限为 $\Omega_{FLARE}(\delta_{th}, D_{ij})$ ，在相同的流量输入下，ADTS 算法的分组乱序度上下限满足式(4)。

6 性能评估

为了验证 ADTS 算法的负载失衡度、分组乱序度以及估计误差对性能的影响，本节采用真实的网络流量数据进行仿真评估。数据来源于现实的骨干链路^[12]，数据集描述如表 1 所示。

实验假设源和目的 2 个节点之间存在 3 条并行的传输路径，拓扑结构如图 2 所示。数据流量在源节点进行切分均衡地分配到 3 条传输路径上，3 条路径的传输时延依次为 20 ms、30 ms 和 45 ms，最

表 1 数据集描述

数据集	流数目	流大小 (以数据分组个数计)		流到达速率 (flow/s)		
		平均值	变异系数	平均值	最大值	最小值
DS1	48 021	24.54	17.13	139.14	214	81
DS2	6 456	424.64	8.13	129.78	202	74
DS3	13 120	312.29	7.09	192.49	273	92

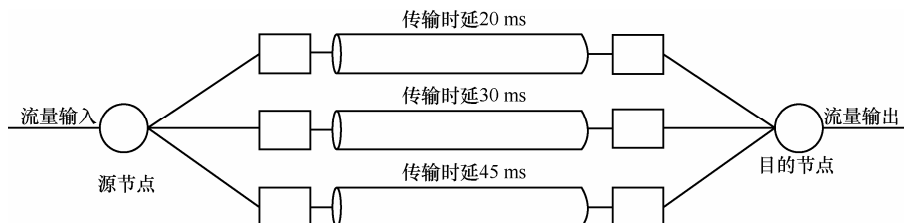


图 2 实验拓扑结构

大最小路径时延之差为 25 ms，数据流量在 3 条路径上的分配比例分别为 0.3、0.3 和 0.4。实际应用中，需要对路径时延进行测量，受测量手段的限制，会存在一定的误差，为了模拟真实应用中路径时延估计误差，实验中 2 条路径时延之差的估计误差设定为 20 ms。之所以这样设定，是因为根据文献[8]对时延估计误差的定义以及估计误差对算法性能影响的分析可知，当估计误差大于等于 20 ms 时，对算法性能影响较为显著。为了验证评估 ADTS 算法的性能，实验同时评估已提出同类算法 FLARE 和 TSNB 算法，并进行对比分析。

6.1 流片数量

分割粒度越细，每秒钟到达的流片数量越多，负载在路径间均衡次数也就越多，路径间的负载也就越均衡；相反，分割粒度越粗，每秒钟到达的流片数量越少，负载在路径间均衡次数也就越少，路径间的负载均衡也就较差。ADTS 算法将网络流量分为强势流和正常流，强势流采用较细粒度的分割方法，正常流采用较粗粒度的分割方法，强势流通常是流量较大的流，采用较细粒度的分割方法能够在损失较小分组乱序度的情况下获得较好的负载均衡效果；TSNB 算法对所有流都采用较细粒度的分割方法，流片粒度较小，每秒钟到达的流片数量较多，负载均衡效果较好；FLARE 算法对所有流都采用较粗的流量分割方法，流片粒度较大，每秒钟到达的流片数量较少，负载均衡效果较差。所以，ADTS 是一种流分片数量介于 TSNB 和 FLARE 之间的算法。从图 3 可以看出，对于实验所用的 3 个数据，ADTS 算法的流片平均到达速率都介于 TSNB 和 FLARE 算法之间，即 ADTS 的负载均衡效果介于 TSNB 和 FLARE 算法之间。

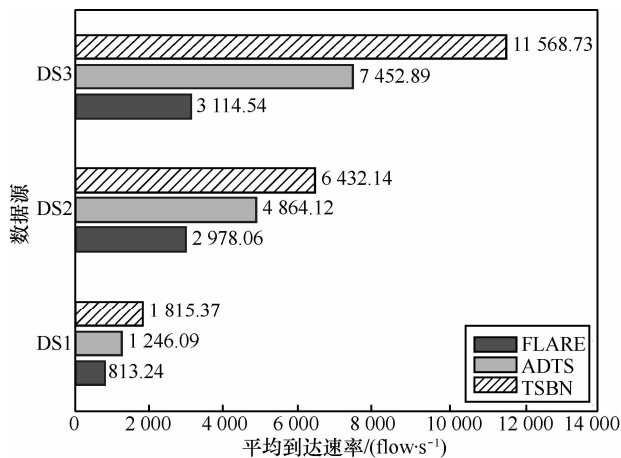


图3 不同数据源下3种算法的流片平均到达速率对比

6.2 负载失衡度

负载失衡度是衡量流量分割算法负载均衡效果的直接指标，负载失衡度越小，负载均衡效果就越好。TSNB 算法在每个数据分组到达时都会在游标右边的路径中进行负载均衡，由于分割粒度较小，流片数量较多，负载失衡度较低；FLARE 算法只有在相邻分组到达时间间隔大于 MMDD 时才会多径中进行负载均衡，由于分割粒度较粗，流片数量较少，负载失衡度较高；而 ADTS 算法对强势流进行细粒度流量分割，正常流进行粗粒度分割，负载失衡度相比于 FLARE 算法大大降低。从图 4 中可以看出，每个采样时刻 ADTS 算法的负载失衡度都大大低于 FLARE 算法，略高于 TSNB 算法，整体负载失衡度相比于 FLARE 算法平均降低 50% 以上。

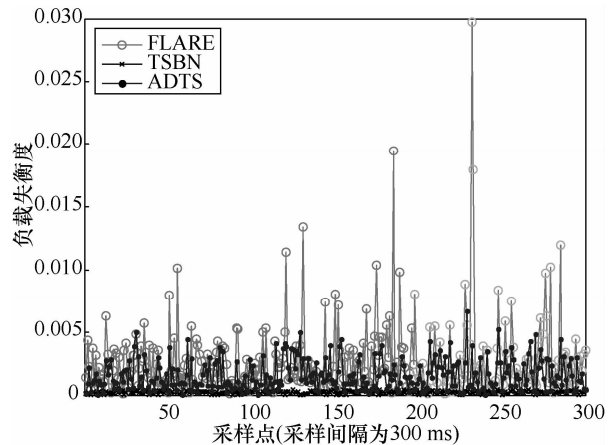


图4 3种算法的负载失衡度对比

6.3 分组乱序度

分组乱序度是衡量流量分割算法是否可用的重要指标。分组乱序将会破坏 TCP 的拥塞控制机制，进而导致网络整体性能的下降，特别是当乱序的分组较多时，严重影响网络性能。TSNB 算法采用细粒度流量分割方法取得了较好的负载均衡效果，但受路径时延估计误差的影响，分组乱序度也较高；FLARE 算法采用粗粒度流量分割方法分组乱序度较低；而 ADTS 算法的分割粒度根据不同的流量而变化，分组乱序度介于 TSNB 和 FLARE 算法之间。从图 5 中可以看出，TSNB 的分组乱序度最高达到了 23%，这种情况将会严重影响网络性能；而 ADTS 算法的分组乱序度最差也不会高于 10%，属于可接受的范围，是一种比较实用的流量分割算法。

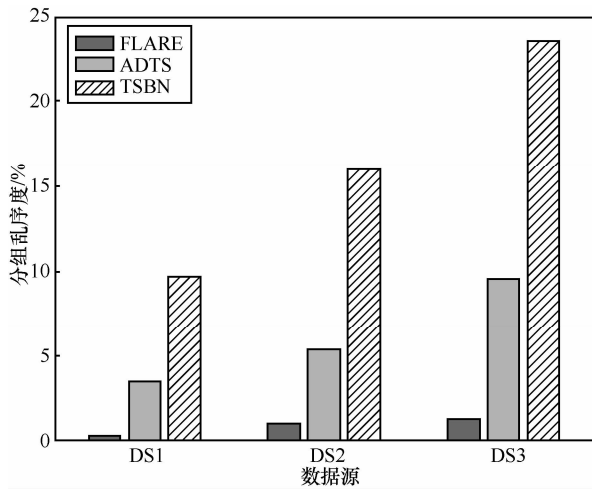


图5 不同数据源下3种算法的分组乱序度对比

7 结束语

本文在对当前提出的多径流量分割算法进行分析总结的基础上,提出了一种粒度自适应的多径流量分割算法——ADTS。考虑到现实网络中路径时延估计误差的存在和网络流量特性的不同,ADTS算法将网络流量分为强势流和正常流,根据2种流量的不同特性分别采用不同的分割粒度,在保证数据分组乱序度较低的情况下获得了较好的负载均衡效果。真实数据实验仿真表明,ADTS算法的综合性能优于已提出的其他同类算法。

虽然ADTS与其他算法相比是一种实际可用的算法,但强势流和正常流能否准确实时检测仍然会对算法的性能产生影响。所以本文还需进一步研究如何精确实时检测强势流,以更好地将ADTS算法应用到实际网络中进行多径流量的分配。

参考文献:

- [1] REXFORD J. Rethinking Internet routing[A]. Proceedings of Symposium on Theory of Computing Conference(STOC'08)[C]. Victoria, Canada, 2008.
- [2] PRABHAVAT S, NISHIYAMA H, ANSARI N, *et al.* On load distribution over multipath networks[J]. IEEE Communications Surveys & Tutorials, 2012, 14(3): 662-680.
- [3] PRABHAVAT S, NISHIYAMA H, ANSARI N, *et al.* On the performance analysis of traffic splitting on load imbalance and packet reordering of bursty traffic[A]. Proc IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009)[C]. Bei-

jing, China, 2009. 236-240.

- [4] LEUNG K C, LIV O K, YANG D. An overview of packet reordering in transmission control protocol(TCP): problems, solutions, and challenges[J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(4): 522-535.
- [5] FENG J, OUYANGZ P, XUL S, *et al.* Packet reordering in high-speed networks and its impact on high-speed TCP variants[J]. Computer Communications, 2009, 32(1):62-68.
- [6] SHREEDHAR M, VARGHESE G. Efficient fair queuing using deficit round robin[J]. IEEE/ACM Trans Networking, 1996, 4(3):375-385.
- [7] SHI W, MACGREGOR M H, GBURZYNSKI P. Load balancing for parallel forwarding[J]. IEEE/ACM Trans Networking, 2005, 13(4):790-801.
- [8] KANDULA S, KATABI D, SINHA S, *et al.* Dynamic load balancing without packet reordering[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(2):53-62.
- [9] 吴春明, 王保进, 陈均华等. 一种基于游标的多径流量分割算法[J]. 电子学报, 2010, 38(11): 2550-2554.
- [10] WU C M, WANG B J, CHEN J H, *et al.* A traffic splitting algorithm based on nonion in multi-path[J]. Chinese Journal of Electronics, 2010, 38(11): 2550-2554.
- [11] PRABHAVAT S, NISHIYAMA H, ANSARI N, *et al.* Effective delay-controlled load distribution over multipath networks[J]. IEEE Trans Parallel and Distributed Systems, 2011, 22(10):1730-1741.
- [12] SHI L, LIU B, SUNC H, *et al.* Load-balancing multipath switching system with flow slice[J]. IEEE Transactions on Computers, 2012, 61(3):350-365.
- [13] DANZIG P, MOGUL J, PAXSON V, *et al.* The internet traffic archive[EB/OL]. <http://ita.ee.lbl.gov/index.html>.

作者简介:



王鹏(1985-),男,河南周口人,国家数字交换系统工程技术研究中心博士生,主要研究方向为新型网络体系结构、路由与交换技术。



兰巨龙(1962-),男,河北张北人,国家数字交换系统工程技术研究中心总工程师、教授、博士生导师,主要研究方向为高性能路由与交换技术、新型网络体系结构。

陈庶樵(1973-),男,陕西西安人,国家数字交换系统工程技术研究中心教授、硕士生导师,主要研究方向为宽带信息网路、新型网络体系结构。