

采用子网流量组合优化的网络流量分割方法

褚伟波^{1,2}, 蔡忠闽^{1,2}, 管晓宏^{1,2,3,4}, 陈明旭^{1,2}

(1. 西安交通大学智能网络与网络安全教育部重点实验室, 710049, 西安; 2. 西安交通大学机械制造系统工程国家重点实验室, 710049, 西安; 3. 清华大学智能与网络化系统研究中心, 100084, 北京; 4. 清华大学清华信息科学与技术国家实验室, 100084, 北京)

摘要: 针对当前基于仿真的流量分割方法存在处理开销过大且仅能保证流层面流量完整性的问题, 提出了一种组合优化的网络流量分割方法. 将流量经分割后得到的数据包总延迟次数作为优化目标, 由此网络流量分割问题转化为二次分配优化问题; 用子网流量作为基本单元进行分配, 可在减小问题求解规模的同时保证子网流量的完整性. 利用真实网络流量对所提方法的分割效果进行了验证, 并与基于数据包和流的轮询流量分割方法进行了比较, 结果表明, 所提方法能够达到基于流的轮询流量分割方法的性能, 数据包延迟较之不采用优化的基于子网流量轮询方法显著减小, 性能提高近 10%.

关键词: 网络流量分割; 数据包延迟; 二次分配; 子网流量

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2011)12-0022-06

A Traffic Splitting Method of High-Speed Network Using Sub-Network Flow Optimization

CHU Weibo^{1,2}, CAI Zhongmin^{1,2}, GUAN Xiaohong^{1,2,3,4}, CHEN Mingxu^{1,2}

(1. MOE Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China; 2. State Key Laboratory for Manufacturing Systems, Xi'an Jiaotong University, Xi'an 710049, China; 3. Center for Intelligent and Networked Systems, Tsinghua University, Beijing 100084, China; 4. Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: The aim of traffic splitting in parallel traffic replay is to distribute high-speed network traffic to multiple replay nodes such that the discrepancies between the generated traffic and the original traffic can be minimized. Currently adopted simulation based methods have a high computational cost and can only guarantee flow level traffic integrity. In this paper, a new method for traffic splitting of high-speed network is proposed based on packet delay time minimization. The problem is converted into a quadratic assignment problem. The computational cost can be dramatically reduced and sub-network level traffic integrity can be achieved by aggregating flows from the same sub-network and assigning traffic in unit of sub-network flows. The method is validated by using actual traffic traces, and a comprehensive comparison between the proposed method and other available traffic splitting methods such as packet based and flow based round-robin approaches is carried out. Empirical studies show that the proposed method can achieve the same replay performance as the flow-based round-robin approach, and achieves 10% performance enhancement over the other approaches with the same split unit.

Keywords: traffic splitting; packet delay; quadratic assignment; sub-network flow

收稿日期: 2011-04-28. 作者简介: 褚伟波(1982—),男,博士生;蔡忠闽(通信作者),男,博士,教授. 基金项目: 国家自然科学基金资助项目(60574087);国家高技术研究发展计划资助项目(2007AA01Z464,2007AA01Z475,2007AA01Z480,2008AA01Z415);国家“242”信息安全计划资助项目(2009A08).

网络出版时间: 2011-09-05

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20110905.1657.002.html>

网络流量还原回放方法^[1-8]是近年来提出的一种新型网络设备测试方法, 其将真实网络中的流量通过捕获、记录后还原回放测试网络中。流量回放可以在数据包层面(包内容、包大小、包间隔等)还原、再现实际网络的场景, 相比人工仿真流量测试^[9-13]和接入真实网络试运行的常规测试方法, 具有测试数据真实、测试过程可控、测试场景可再现以及便于发现、分析和定位问题等优点, 因此被广泛应用于入侵检测系统(IDS)、防火墙等各类网络安全设备的测试评估。

目前的流量还原回放主要采用单点回放的方法, 即采用一台服务器来回放捕获的流量。单点回放简单且在其处理能力之内能够保证流量回放的质量, 但是当网络流量规模超过回放服务器的处理能力时, 它的磁盘存储空间、流量带宽均无法满足实际需求。多年来的研究表明, 网络流量的增长速度远远超过计算机的发展速度, 因此一味地提高单个回放节点的处理能力并不能应对高速网络流量带来的问题。

大规模高速网络流量的回放通常采用并行回放的方法, 如图 1 所示。在该方法中, 高速网络流量先被分配到不同的处理节点上, 而多个回放节点产生的流量经过路由器或交换机等汇流设备汇聚后形成高速流量, 这样通过简单地增加回放服务器的数量便可以应对大规模流量回放的需求。

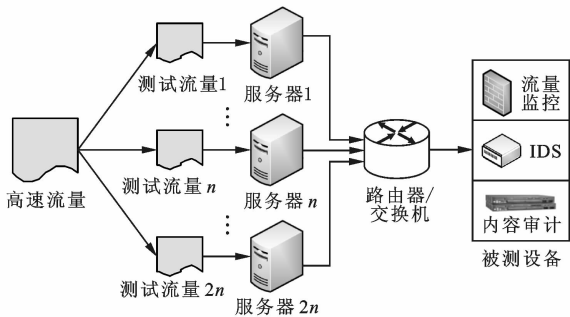


图 1 高速网络流量的并行回放示意图

在并行回放中, 流量先分配给多个节点, 再由多个子流量汇聚而成, 其与原始流量相比在数据包时序特性上将发生变化。引起这种变化的主要原因有: ①节点之间存在着时间同步误差, 即节点报文的发送时间取决于节点本地的时间, 而原始流量中数据包序列具有唯一的时间顺序, 所以节点间的这种时间同步误差会影响回放流量中的数据包时序特性; ②节点的报文发送存有延迟, 即网络流量在回放时会出现数据包延迟的现象(见图 2); ③网络流量汇

聚设备(路由器、交换机等汇流设备)采用数据包调度算法会影响流量的时序特性。

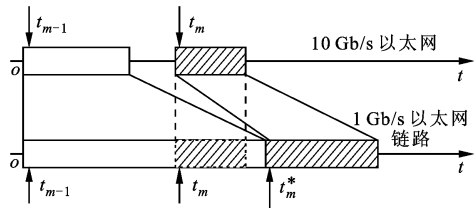


图 2 高速链路数据包在低速链路回放时的延迟现象

网络流量分割的核心是流量分配时使回放产生的流量与原始网络流量在时间特性上尽可能保持一致。本文将流量分割后的数据包总延迟次数作为流量分割质量的度量指标, 基于此又提出了一种组合优化的网络流量分割方法, 同时用子网流量作为基本单元进行分配, 这样网络流量分割问题将转化为二次分配优化问题。

1 并行回放的流量分割问题

1.1 流量分割问题的描述

首先考察单个节点在发送数据包时所产生的延迟。高速网络链路流量由低速链路回放时所引发的数据包发送延迟如图 2 所示。记 t_m 为高速链路中第 m 号数据包的到达时间, t_m^* 为该数据包在低速链路中被回放的时间。在理想情况下, $t_m = t_m^*$ 。数据包的前一个报文(第 $m-1$ 号数据包)在低速链路中的发送时间比高速链路中的发送时间长, 因此第 m 号数据包的发送时间 $t_m^* > t_m$, $|t_m^* - t_m|$ 则为该数据包的时间延迟。

为了使节点在流量回放过程中的时间延迟尽可能小, 就需要采用一种合理的方式将高速网络流量分配到各个节点上。直接以数据包为单位的分割方法(如将数据包以 round-robin 的方式分配给各个节点)虽然可以确保回放时的时间延迟很小, 但它会误将同一流的数据包分配到不同的节点上, 从而造成流量汇聚时流内数据包乱序。流内数据包的乱序现象会影响 IDS 等设备的测试, 因此本文将通过增大粒度单元来进行流量分割。

记 $F = \{f_1, f_2, \dots, f_n\}$ 为高速网络流量, 其中 f_i 为第 i 个流量单元。流量分割的目标是将高速网络流量 F 分割成若干个子流量 $\{R_1, R_2, \dots, R_m\}$, 每一个子流量由一个节点来回放, 则有 $\bigcup_{i=1}^m R_i = \{f_1, f_2, \dots, f_n\}$, $R_i \cap R_j = \emptyset, 1 \leq i \neq j \leq m$ 。同时, 所有子流量在节点回放时的时间延迟应满足目标要求。

1.2 基于仿真的流量分割方法

文献[14]的面向并行回放流量分割方法是一种基于仿真的方法,如图3所示,它由流量分割与仿真评估两部分组成.流量分割主要根据当前确定的回放节点数 N ,以流为基本分割单元,用轮询或者随机分配的方法将高速流量均匀地分配给各个节点.当一次流量分割结束时,进行回放质量评估.每个回放节点用一个单队列的排队系统进行仿真,此时的回放节点将作为队列的处理节点,该节点分配到的数据包作为顾客,数据包的时间戳为顾客到达时间,数据包回放时间则为顾客服务的所需时间.这样,仿真评估便可以考察数据包在队列(回放节点)中的平均等待时间和平均队列长度等性能指标.仿真评估完成后,系统将考察当前的性能指标是否满足回放的要求,若有一个节点不满足要求,则递增回放节点数 N ,重新进行流量分割和仿真评估,直到当前的分割方案满足回放需求为止.

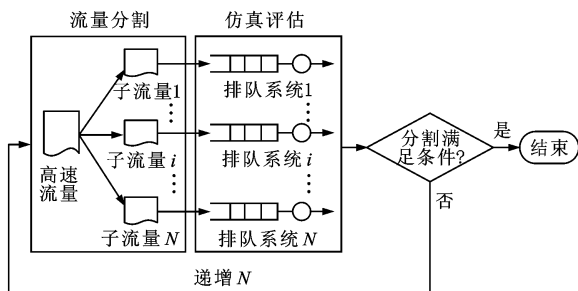


图3 基于仿真的流量分割方法

仿真的方法虽然可以确定并行回放时的回放节点数以及每个节点分配到的流量,但是在对每一轮的流量进行分割时需要为单个回放节点用一个排队系统进行仿真,这种流量分割的开销较大.仿真的方法是一种循环试错(trial-and-error)的方法,理论上无法保证回放的质量,仅能保证流层面上流量的完整性,而流之间容易出现乱序的现象,如将同一个主机顺序产生的流分配到了不同的节点上.

2 组合优化的网络流量分割方法

2.1 系统建模和求解

在回放节点资源一定的前提下,将网络流量分配给各个节点,使得回放的流量与原始的网络流量在时间特性上的差异尽可能小.为了做到这一点,本文将数据包的总延迟次数作为流量分割后的回放质量度量指标.

定义1 单个数据包的延迟次数为子流量中数据包

发送时间跨越该数据包起始时刻的其他数据报文的个数.

数据包的发送时间是指,以数据包的时间戳(高速流量中每个数据包所拥有的时间戳)为起始时间,长度为低速链路中该数据包发送时间的区间.如图4所示,若流量单元 f_1, f_2, f_3 同时分配到一个节点上,则会有一个数据包的发送时间跨越了数据包 P_4, P_5, P_6 的发送时间,如 P_1 的发送时间跨越了 P_4 的发送时间, P_4 的发送时间跨越了 P_6 的发送时间, P_2 的发送时间跨越了 P_5 的发送时间,因此 P_4, P_5, P_6 的延迟次数均为1,而 P_2 和 P_5 的发送时间跨越了 P_7 的发送时间,因此 P_7 的延迟次数为2.

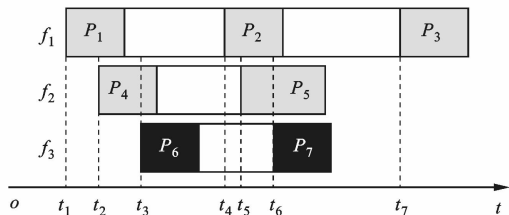


图4 数据包延迟次数示意图

定义2 流量分割后的数据包总延迟次数为分割后子流量中所有数据包的延迟次数之和.

在大规模高速网络中,单个流量单元(如一台主机)的流速较低,可认为其被节点回放时不会产生延迟,即单个流量单元的数据包延迟次数为0,可见数据包的延迟是流量单元聚集在一起时引起的.

记 d_{ij} 表示流量单元 f_i 和 f_j 聚集在一起时产生的数据包延迟次数,则 d_{ij} 具有如下性质.

性质1 数据包延迟次数具有对称性,其与流量单元聚集的次序无关,即 $d_{ij} = d_{ji}$.

性质2 数据包的延迟次数具有累加性,即以 $Z(f_i, f_j) = d_{ij} = d_{ji}$ 表示流量单元 f_i 和 f_j 聚集在一起时的数据包延迟次数,则 $Z(f_1, f_2, \dots, f_n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij}$ 成立.

证明 利用数学归纳法证明如下.

(1) 当 $n=2$ 时,由定义可知, $Z(f_1, f_2) = d_{12}$ 成立.

(2) 当 $n=k$ 时, $Z(f_1, f_2, \dots, f_k) = \sum_{i=1}^{k-1} \sum_{j=i+1}^k d_{ij}$

成立. 当 $n=k+1$ 时,数据包延迟情况如下:记 $D((f_1, f_2, \dots, f_k), f_{k+1})$ 为 f_{k+1} 与集合 $(f_1, f_2, \dots, f_k)$ 聚集在一起时新增加的数据包延迟次数,由数据包延迟次数的定义可知, $f_i (1 \leq i \leq k)$ 中任意一个数据包新增加的延迟次数是 f_{k+1} 中发送时间跨越该数据包起始时间的数据报文个数.对于 f_{k+1} 中任意

一个数据包,由于 f_{k+1} 和 $f_i (1 \leq i \leq k)$ 聚集在一起而引起的数据包延迟次数是 f_i 中发送时间跨越该数据包起始时间的数据报文个数,因此可以将 f_{k+1} 与 f_i 聚集在一起时新增加的数据包延迟次数(由 $f_i (1 \leq i \leq k)$ 中增加的数据包延迟次数和 f_{k+1} 中增加的数据包延迟次数 2 部分组成)表示为 $Z(f_i, f_{k+1}) = d_{i,k+1}$,相应地,可以将 f_{k+1} 与集合 $(f_1, f_2, \dots, f_k)$ 聚集在一起时新增加的数据包延迟次数 $D((f_1, f_2, \dots, f_k), f_{k+1})$ 表示为

$$D((f_1, f_2, \dots, f_k), f_{k+1}) = \sum_{i=1}^k Z(f_i, f_{k+1}) = \sum_{i=1}^k d_{i,k+1}$$

所以 f_{k+1} 与集合 $(f_1, f_2, \dots, f_k)$ 聚集在一起时的数据包延迟次数

$$Z((f_1, f_2, \dots, f_k), f_{k+1}) = Z(f_1, f_2, \dots, f_k) +$$

$$D((f_1, f_2, \dots, f_k), f_{k+1}) =$$

$$\sum_{i=1}^{k-1} \sum_{j=i+1}^k d_{ij} + \sum_{i=1}^k d_{i,k+1} = \sum_{i=1}^k \sum_{j=i+1}^{k+1} d_{ij}$$

综合上述(1)和(2),由 n 的任意性可知, $Z(f_1,$

$$f_2, \dots, f_n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} \text{ 成立. 证毕.}$$

性质 2 的物理意义是:当一个新的流量单元添加到一个子流量时,形成的新的子流量中数据包总延迟次数是原来子流量中数据包总延迟次数加上新流量单元与原来子流量中每个单元两两聚集在一起时的数据包延迟次数。

设原始高速网络中包含了 M 个流量单元,通过定义数据包延迟次数,可将网络流量抽象为 $M \times M$ 的矩阵 $\mathbf{D}_{M \times M}$,矩阵中每个元素表示对应位置的流量单元聚集在一起被回放时的数据包延迟次数(由于每个单元只能唯一地分配给一个节点进行回放,因此 $\mathbf{D}_{M \times M}$ 中的对角线元素全为 0),即

$$\mathbf{D}_{M \times M} = \begin{Bmatrix} d_{11} & d_{12} & \cdots & d_{1M} \\ d_{21} & d_{22} & \cdots & d_{2M} \\ \vdots & \vdots & & \vdots \\ d_{M1} & d_{M2} & \cdots & d_{MM} \end{Bmatrix} \quad (1)$$

式中: $d_{ij} = d_{ji}$, 且 $d_{ii} = 0, \forall i$.

又设并行回放的节点数为 N ,定义流量分配矩阵

$$\mathbf{X}_{M \times N} = \begin{Bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{Bmatrix} \quad (2)$$

$$x_{ij} = \begin{cases} 1, & f_i \text{ 分配给第 } j \text{ 个节点} \\ 0, & f_i \text{ 未分配给第 } j \text{ 个节点} \end{cases}$$

式中: x_{ij} 表示 f_i 与第 j 个节点之间的分配关系. 于是,网络流量的最优分割问题可转化成如下优化目标为最小化数据包总延迟次数的组合优化问题,即

$$\begin{aligned} & \min \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^M x_{is} x_{js} d_{ij} \\ \text{s. t. } & \begin{cases} x_{ij} = \{0, 1\}, & 1 \leq i \leq M, \quad 1 \leq j \leq N \\ \sum_{j=1}^N x_{ij} = 1, & \forall i \end{cases} \end{aligned} \quad (3)$$

上述基于最小化数据包总延迟次数的流量分割方法在宏观上是高速网络流量在低速链路回放时相互交错的数据包尽可能地分散处理. 由于回放过程中数据包的延迟主要是报文之间相互交错引起的,因此可以通过对数据包总延迟次数进行优化来保证分割后的流量在被多个节点回放时所产生的数据包的延迟尽可能的小. 也就是说,可以用数据包总延迟次数来反映流量回放的质量。

更具体地说,式(3)所示的组合优化问题是一个二次分配优化问题. 针对该问题,当前已提出了不少的求解方法,如各类搜索算法^[15-16]等. 本文采用文献^[16]中的方法进行求解。

2.2 流量分割单元的选取

流量分割单元的选取需要同时满足:①问题规模显著下降;②最大程度地保证单个流量单元被节点回放时不产生数据包延迟。

为此,应先将采集到的流量按照一侧(如内网)的 IP 地址空间划分为若干子网网段,再以子网流量作为基本单元来降低问题的规模,最后通过合理设置子网的大小来保证单个流量单元被节点回放时不出现延迟。

2.3 优化性能分析

理论上,数据包总延迟次数应当是单个流量单元被节点回放时的数据包延迟次数与流量单元相互聚集时的数据包延迟次数之和. 由于本文假设单个流量单元的流速较低,节点回放时的数据包延迟次数为 0,因此优化结果实际反映的是流量单元相互聚集时的数据包延迟次数. 所以,流量单元的粒度会影响到优化的效果. 粒度越小,模型忽略的单个流量单元在节点回放时的数据包延迟次数就减少,求解结果就越接近于真实情况,优化效果也就越好。

另外,流量单元的粒度还会影响到优化的性能. 粒度越大,优化时的组合方案就减少,优化后性能提高也就减少;粒度越小,组合方案就越多,优化效果就越好. 如果将数据包、流和子网流量分别作为流量

单元进行流量分割,其优化效果应当依次下降,但流量完整性依次递增.

性能的提高与回放节点数有关.回放节点数越多,单个回放节点分配到的流量就越少,回放性能随之提高.在极端情况下,若子网大小固定不变,每一个子网流量均由一个独立节点进行回放,则分割方法的数据包延迟达到最小.

通常情况下,回放节点数是固定的,比如为了回放 1 Gb/s 的链路流量,会使用 10 个 100 Mb/s 的节点.因此,分割方法的性能主要依赖于子网流量的定义.

在实际的流量回放中,子网流量的大小是通过考察目标节点的处理能力来确定的.比如,目标节点的处理能力为 100 Mb/s,则以子网流量的大小最好不超过节点处理能力的 1/5 来划分子网的大小.

3 实验结果与分析

由西安交通大学校园网的真实流量作为测试用例,来考察本文方法的分割与回放效果.流量中共包含有 3 403 664 个数据包,126 894 条流,持续时间为 58 s,平均流量为 280 Mb/s.采集流量的内网 IP 区间为 115.154.16.0~115.154.127.255,产生流量的内网主机数为 15 094.实验中将该流量分配给 3 台网络带宽为 100 Mb/s 的普通 PC 机进行回放.

采用子网掩码 ff. ff. ff. c0(即以 IP 地址的前 26 位作为子网标识)对采集流量的内网 IP 区间进行划分,共得到 320 个子网流量.其中,最大的子网流量包含了 151 250 个数据包,流速约为 18 Mb/s.可见,采用子网流量作为基本流量单元能够满足流量分割在单元选取上的 2 个条件.

实验还考察了数据包轮询、流轮询、子网流量组合优化以及子网流量轮询这 4 种不同的流量分割方法.以子网流量作为基本分割单元,由本文的基于优化的方法计算得到的数据包延迟次数为 1 852 500,基于子网流量轮询的数据包延迟次数为 2 031 396,

显然本文方法的数据包延迟次数显著减少,性能提高近 10%.

图 5 是不同流量分割方法的数据包延迟对比.由图 5a 看出,子网数为 320 时的数据包延迟情形有:①在数据包轮询、流轮询以及子网流量轮询这 3 种流量分割方法中,回放产生的数据包延迟 Δt 随着分割单元粒度的增大而增大,分割单元粒度越大,数据包延迟越大;②本文方法的流量回放质量与流轮询方法基本相同,但却保证了同一个子网内的流量分配给同一个节点,从而保证了子网流量的完整性;③与子网流量轮询分割方法相比,本文优化方法的流量回放质量显著提高.

由图 5b、5c 看出,子网数分别为 80、160 时的数据包延迟情形有:①数据包轮询、流轮询方法的数据包延迟不会随着子网大小的变化而变;②当子网流量作为基本分割单元时,子网数为 80 时的本文方法和子网流量轮询方法的数据包延迟次数分别是 1 695 260 和 1 866 308,子网数为 160 时分别为 1 794 820 和 1 943 125,可见若不考虑子网内的数据包延迟,以子网流量为基本分割单元得到的数据包延迟次数会随着子网数的增加(子网规模变小)而增加;③本文方法在回放时的数据包延迟与流轮询方法基本相同;④以相同的子网流量作为基本分割单元,本文方法的数据包延迟次数比子网流量轮询方法更少,回放后的数据包延迟更小;⑤子网流量轮询方法的回放质量会随着子网规模的增大而显著下降,而本文方法则较为稳定.

图 5 表明,本文方法的数据包延迟与流轮询方法的类似,这种相似性是由如下 2 种原因综合引起的:①流轮询方法的流量分割单元是单条流,子网流量优化方法是将子网流量作为基本分割单元,而流的粒度要比子网流量的小,因此若同样采用轮询的方法,流轮询要优于子网流量轮询;②本文的优化方法比基于子网流量轮询方法在性能上大幅提高.

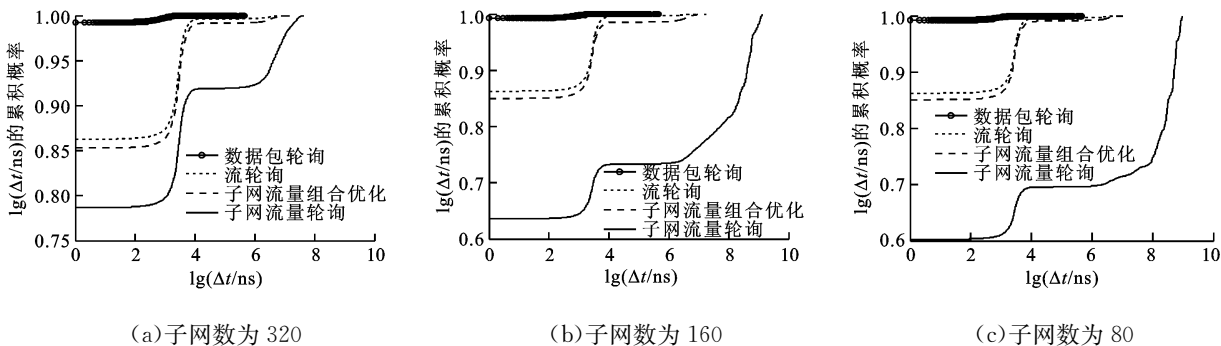


图 5 不同流量分割方法的数据包延迟对比

此外,我们还给出了基于子网流量组合优化分割方法的模型求解时间(流量分割时增加的时间). 实验中,当子网数为 80 时,模型求解时间为 2 s;当子网数为 160 时,模型求解时间为 4 s;当子网数为 320 时,模型求解时间为 126 s. 可见,本文方法的计算时间增加得并不多,考虑到流量分割问题是一个静态问题(并不需要在线处理),因此分割方法所需时间是可以接受的. 在其他时段对网络流量进行了实验,结论也是一致的.

最后需要指出的是,本文定义的子流量中单个数据包的平均延迟次数作为当前节点数下的流量分割的质量度量指标,也可以用来确定回放过程中的节点数.

4 结束语

本文提出了一种基于组合优化的网络流量分割方法,可将网络流量分割问题转化为二次分配优化问题. 采用子网流量作为基本单元进行分配,可在减小问题求解规模的同时保证子网网络流量的完整性. 利用真实流量对本文方法的分割效果进行了验证,结果表明,本文方法能够达到基于流轮询分割方法的性能,较之不采用优化的基于子网流量轮询方法等所产生的数据包延迟显著减小,性能提升约 10%.

参考文献:

- [1] Syn Fin dot Net. The tcpreplay & flowreplay homepage [EB/OL]. [2011-01-21]. <http://tcpreplay.synfin.net/trac/>.
- [2] Sourceforge 开源社区. The Tomahawk test tool homepage [EB/OL]. [2011-01-21]. <http://tomahawk.sourceforge.net/>.
- [3] CHEN Z, DELIS A, WEI P. A pragmatic methodology for testing intrusion prevention systems[J]. Computer Journal, 2009, 52(4): 429-460.
- [4] CHENG Y C, HOLZLE U, CARDWELL N, et al. Monkey see, monkey do: a tool for TCP tracing and replaying[C]//USENIX Annual Technical Conference on General Track, Berkeley, CA, USA: USENIX Assoc., 2004: 87-98.
- [5] CHU W, GUAN X, CAI Z, et al. Balance based performance enhancement for interactive TCP traffic replay[C]//IEEE International Conference on Communications, Piscataway, NJ, USA: IEEE, 2010:1-5.
- [6] FENG W C, GOEL A, BEZZAZ A, et al. TCPivo: a high-performance packet replay engine[C]//Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research, New York, USA: ACM, 2003: 57-64.
- [7] HONG S S, WU S F. On interactive internet traffic replay[C]//8th International Symposium on Recent Advances in Intrusion Detection, Berlin, Germany: Springer-Verlag, 2005:247-264.
- [8] 褚伟波,蔡忠闽,管晓宏,等. 基于收发平衡判定的 TCP 流量回放方法[J]. 计算机学报, 2009, 32 (4): 835-846.
- [9] ANTONATOS S, ANAGNOSTAKIS K G, MARKATOS E P. Generating realistic workloads for network intrusion detection systems[C]//Proceedings of the 4th Workshop on Software and Performance, New York, USA: ACM, 2004:207-215.
- [10] JAMIN S, DANZIG P. TcpIib: a library of TCP internetwork traffic characteristics, tech report USC-CS-91-495[R]. Los Angeles, CA, USA: University of Southern California, 1991.
- [11] NICOL D M, YAN G. Simulation of network traffic at coarse time-scales[C]//Proceedings of the 19th Workshop on Parallel and Distributed Simulation, Los Alamitos, CA, USA: IEEE Comput. Soc., 2005:141-150.
- [12] NICOL D M, YAN G. High-performance simulation of low-resolution network flows [J]. Simulation, 2006, 82(1): 21-42.
- [13] SARVOTHAM S, RIEDI R, BARANIUK R. Network and user driven alpha-beta on-off source model for network traffic[J]. Computer Networks, 2005, 48 (3): 335-350.
- [14] YE T, VEITCH D, IANNACONE G, et al. Divide and conquer: PC-based packet trace replay at OC-48 speeds [C]//Proceedings of 1st International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, Los Alamitos, CA, USA: IEEE Comput. Soc., 2005: 262-271.
- [15] JAMES T, REGO C, GLOVER F. Multistart tabu search and diversification strategies for the quadratic assignment problem[J]. IEEE Transactions on Systems, Man, and Cybernetics; Part A Systems and Humans, 2009, 39(3): 579-596.
- [16] STUTZLE T. Iterated local search for the quadratic assignment problem[J]. European Journal of Operational Research, 2006, 174(3): 1519-1539.

(编辑 苗凌 武红江)