面向互动式网络场景再现的流速控制系统与方法

褚伟波[1)] 管晓宏[1),2)] 蔡忠闽[1)] 陶 敬[1)]
[1)](西安交通大学智能网络与网络安全教育部重点实验室 西安 710049)
[2)](清华大学自动化系 清华信息科学与技术国家实验室 北京 100084)

摘 要 互动式网络场景再现是一种重要的真实网络流量产生方法.然而,由于流量产生过程的复杂性,基于精确的数学模型对该过程产生流速进行控制是一个较为困难的新问题.文中设计实现了一个面向互动式网络场景再现的流速控制系统,并将网络场景再现过程中的流速控制问题转化为目标跟踪控制问题进行求解.该系统采用一种基于函数近似器的流速控制方法,利用函数近似器对系统的输入输出关系进行描述,通过动态调整系统输入流量来对回放过程输出流量进行跟踪.最后,利用真实网络流量实验考察了文中系统和方法在不同丢包、传输延迟以及会话阻断环境下的实际控制效果,并从收敛时间、产生输入输出、控制误差等角度对系统的控制性能进行了分析.

关键词 互动式网络场景再现;流量回放;流速控制;函数近似器
中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2012.01485

System and Method for Real-Time Volume Control in Reproducing Network Scenario

CHU Wei-Bo[1)] GUAN Xiao-Hong[1),2)] CAI Zhong-Min[1)] TAO Jing[1)]
[1)](*MOE Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an* 710049)
[2)](*Department of Automation, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Interactive network scenario reproduction is an important method for generating realistic and responsive traffic workloads. However, due to the complex traffic generation mechanisms, controlling traffic volume based on accurate mathematical models in interactive network scenario reproduction becomes a challenging problem. In this paper, we design and implement a volume control system for interactive network traffic replay, and formulate the volume control task as a target tracking problem where the output traffic volume is regulated through adjustment of input traffic volume. We then adopt a function approximator (FA) in our system to generate the desired input volumes. The FA characterizes the mapping that takes system measurements as input and directly produces controls as output. Meanwhile, simultaneous perturbation stochastic approximation (SPSA) algorithm is employed to adaptively estimate the parameters in control. To validate the system and the method we have implemented it and conducted experiments with actual traffic traces. Empirical studies show that our system can track target output volumes effectively under a wide range of network conditions. The performance of the control system is further investigated in terms of its convergence time, generated input/output traffic volume and target tracking error.

Keywords interactive network scenario reproduction; traffic replay; volume control; function approximator

收稿日期:2011-02-24;最终修改稿收到日期:2012-05-14.本课题得到国家自然科学基金(60921003,61175039,61103241)、中央高校基本科研业务费专项资金(xjj20100051)资助.**褚伟波**,男,1982年生,博士研究生,主要研究方向为网络流量分析与建模、网络流量还原回放和网络设备测试评估.E-mail: wbchu@sei.xjtu.edu.cn.**管晓宏**,男,1955年生,博士,教授,主要研究领域为计算机信息安全、系统优化与调度.**蔡忠闽**(通信作者),男,1975年生,博士,副教授,主要研究方向为计算机网络安全和入侵检测.E-mail: zmcai@sei.xjtu.edu.cn.**陶 敬**,男,1978年生,硕士,工程师,主要研究方向为网络场景仿真与再现.

1 引 言

网络场景再现[1-15]是近年来逐渐受到重视的一个新兴研究方向,其核心问题是再现符合真实网络场景的流量.网络场景再现研究对于网络安全、网络管理、网络设备测试评估等领域具有重要意义.当前网络流量再现方法主要有两种,即基于仿真的流量再现方法[1-6]和基于真实网络流量还原回放[7-15]的方法.基于仿真的流量再现方法能够通过调整模型参数来灵活产生需要的流量(如 Spirient's Smartbits①, IXIA's Traffic Generator②).然而,由于实际网络流量异常复杂,基于仿真方法产生的流量往往与实际流量相差较大(如数据包大小、IP 地址和端口分布、应用类型等),利用仿真流量对系统进行测评得到的结果并不足以代表系统在真实网络环境下的功能和性能.在实际情况中也经常发现系统在仿真流量环境下功能和性能均表现正常而在实际网络中却出现各种问题;相反地,网络流量回放方法则是将真实网络中的流量进行捕获记录[16]并还原回放到测试网络的一种流量产生方法.由于采用了实际网络中捕获得到的流量,流量回放方法可以在数据包层面(包大小、包内容、包时序等方面)还原实际网络场景、达到直接考察被测系统在真实网络中的功能和性能的目的.

根据流量回放机制的不同,流量回放方法又分为直接回放[7-11]和互动式回放[12-15]两种,分别对应直接的流量再现以及互动式流量再现.直接流量再现方法(如 TCPReplay③、TCPivo[7]、Monkey[8]、Tomahawk④)简单地将捕获到的数据包注入到测试网络来对网络场景进行还原,主要用于测试 IDS(入侵检测系统)、Sniffer(嗅探器)等各类被动型旁路设备;而互动式再现方法则在流量产生过程中仿真实现了 TCP/IP 协议栈并严格基于网络协议规范来产生数据报文,有效解决了直接流量再现方法在测试 Firewall(防火墙)、IPS(入侵阻断系统)等各类串接型安全设备过程中产生大量违反协议语义数据包的问题[12,14](如采用直接再现方法在测试 Firewall 的过程中,SYN_ACK 报文会在 SYN 报文被 Firewall 阻断的情况下仍然被回放到测试网络中,从而极大地影响到测试结果的准确性).由于在流量再现过程中考虑了数据包之间的协议语义并模拟会话两侧的行为来产生流量,因而互动式网络流量再现方法可以通过产生流量的变化反映被测设备的作用,达到考察被测设备串接在实际网络时的功能和性能的目的.当前,互动式网络流量再现方法已经成为了对 IPS、Firewall 等各类串接型网络设备进行测试评估的最新方法.

在网络场景再现过程中,对回放产生流量的流速进行控制具有重要意义.流速控制主要有两个方面,一方面是产生尽可能大的流速的流量,另外一方面是产生满足用户设定流速的流量.产生高速流量可用于考察被测系统在高速网络环境下的功能和性能,而产生用户指定流速的流量在测试过程中同样非常重要.在实际网络管理和设备测试过程中经常发现被测系统的一些深层次故障和问题往往是在流速突然发生变化的时候或在一些特定流量水平上才体现出来.所以,流速控制在流量再现过程中便成为了一个非常重要的环节.

在本文中,考虑在互动式网络场景再现过程中产生满足目标网络环境且具有指定流速的流量的问题.在直接网络场景再现过程中,由于数据包被直接注入到测试网络,在流量产生过程中对流速进行控制是一个相对较为简单的问题,而在互动式场景再现过程中对流速进行控制则较为复杂,其主要原因如下:

(1)作为系统输入的网络流量的不稳定性.大量研究表明真实网络流量是一个非常复杂的非平稳过程[17-19],各种特征诸如流量大小、数据包大小,IP 地址和端口分布等均是随时间动态变化的.特别是由于互动式流量回放过程是基于流来产生测试流量,流量中各种与流相关的特征诸如流的长短、流的持续时间和流的到达速率等均会影响到系统性能;

(2)测试环境的作用.流量产生过程中测试环境的丢包、会话阻断和数据包传输延迟等作用会影响到回放时流的状态,进而影响到回放系统的性能;

(3)回放系统自身的动态特性.互动式流量回放系统通常是由软件实现,因此操作系统中的进程调度、磁盘 IO、网络 IO 等随机因素也会影响到流量回放过程.

考虑到互动式场景再现过程的复杂性,对其建立精确的数学模型并基于模型进行流速控制便成为

① http://www.spirent.com/
② http://www.ixiacom.com/
③ http://tcpreplay.synfin.net/trac/
④ http://tomahawk.sourceforge.net/

了一个非常困难的过程.由于不知道系统的数学模型,一些传统基于模型的控制方法(如 H 无穷最优控制、模型参考控制等等)便无法适用.

在文中,我们设计实现了一个面向互动式网络场景再现的流速控制系统,并将网络场景再现过程中的流速控制问题转化为目标跟踪控制问题进行求解.所设计的系统采用一种基于函数近似器的流速控制方法,利用函数近似器对系统的输入输出关系进行描述,通过动态调整系统输入流量来对回放过程输出流量进行跟踪,并同时通过并行扰动随机估计理论[20]对近似器中的参数进行实时估计.该系统的最大优点在于可以在并不知道系统模型的情况下仍然对复杂过程进行有效控制.

我们采用真实网络流量对所提出的流量控制系统和方法在不同网络环境下的实际控制效果进行了分析与验证.实验结果表明,本文所设计实现的流速控制系统可以精确地对回放过程产生的流速进行控制,在测试网络丢包率达到 10%以及会话阻断率达到 30%等极端情况下,系统仍然能够达到很好的控制效果.此外,还从控制过程的收敛时间、产生输入输出以及控制误差等角度对该系统的控制性能进行了分析.

本文第 2 节介绍相关研究工作;第 3 节介绍互动式网络场景再现过程流速控制问题;第 4 节对影响互动式网络场景再现过程产生流速的因素进行分析;第 5 节介绍基于函数近似器和并行扰动随机估计理论的流速控制系统和方法;第 6 节给出用真实网络流量对所提系统和方法实际控制效果进行实验验证与分析的结果;第 7 节对全文工作进行小结并对未来进行展望.

2　相关研究工作

基于真实网络流量的场景再现研究工作始于 2001 年左右.最早研究人员在 sourceforge 上建立了 TCPDump① 和 Libpcap② 等开源项目,用于开发在网络中能够实时捕获数据报文的系统.在这之后很大一部分研究开始转向流量还原回放方面.文献[7]介绍了 TCPivo——一个 Linux 平台下的高性能数据包回放系统,并提出了一些用于提高数据报文回放效率的系统实现方法;Monkey[8] 是加州大学和 Google 共同开发出来用于测试服务器端性能的流量回放系统,分为 Monkey see 和 Monkey do 两个功能模块,Monkey see 用于在服务器端一侧捕获服务器与客户端交互的流量,并提取包括数据报文间隔、网络传输延迟、丢包率等参数信息;而 Monkey do 用于模拟客户端和传输网络行为,根据接收到的服务器端发送过来的数据包来产生(回放)客户端的流量,以此考察服务器端的参数变化(如不同的缓存大小)对于网络服务性能的影响.文献[9-11]考察了大规模网络流量并行回放中的流分割和回放质量评估问题.

上述研究工作主要定位于对捕获流量进行单向回放.在 TCPReplay 和 Tomahawk 中,研究人员开发出用于测试各类串接型设备的数据报文回放系统.该类系统采用了流量的双向收发机制,即利用两个测试接口来回放捕获的双向流量,每个测试接口用来回放一个方向的流量并接收来自另外一个方向的流量;被测设备串接在两个测试接口之间,回放系统通过对比接收的流量和回放的原始流量来考察串接设备对于流量施加的作用.

为了解决流量直接回放方法在测试 IPS、Firewall 等各类串接型安全设备过程中产生大量违反协议语义数据包的问题,在 2005 年加州大学的 Hong 和 Wu 首次提出了互动式的网络流量场景再现方法,并开发了原型系统 TCPOpera[12].在这种新型的流量回放方法中,除了采用流量的双向收发机制外,回放系统还模拟实现了 TCP/IP 协议栈,为每个被回放 TCP 流会话维护状态.在流量回放时,回放系统严格根据 TCP/IP 协议,采用基于状态的方法来回放 TCP 流量.由于在流量产生过程中考虑了数据包之间的协议语义并模拟会话两侧的行为来产生流量,互动式网络流量回放方法可以产生满足协议语义的流量,并通过产生流量的变化反映被测设备的作用,达到考察被测设备串接在实际网络时的功能和性能的目的.文献[13]对互动式流量回放测试系统的具体实现作了详细的介绍.

也有学者将仿真方法和流量回放技术相结合来产生目标流量,如文献[21-22].

本文作者所在研究小组自 2007 年开始对互动式网络场景再现方法进行了相关研究.在文献[14-15]中,作者基于 TCP 协议的确认机制和停等特性(在流量回放时体现为收发平衡现象),提出了一种基于收发平衡和状态判定相结合的新的 TCP 流量回放

① http://www.tcpdump.com/
② http://sourceforge.net/projects/libpcap/

方法.通过在回放 TCP 流量时将满足收发平衡条件的数据包优先发送出去,所提方法能够在保持 TCP 流量回放语义的前提下,有效减少数据包发送时的状态判定开销,从而提升回放性能.而本文的研究工作则是关注 TCP 流量回放过程在受到丢包、传输延迟和会话阻断等网络环境作用时的流速控制问题.因此,虽然两者的研究对象均为互动式流量回放过程,但是关注点不同.

与本文研究工作最为接近的是加州大学的 Vishwanash 和 Vahdat 的工作,他们开发的 Swing[23] 是一个将网络模型和真实流量数据相结合的流量产生系统.具体地,Vishwanash 和 Vahdat 认为产生真实网络流量需要一个层次化模型,从上到下分别对应用户层、会话层、链接层和网络层.他们将产生流量的模型参数分成两类,一类是流量特征参数,包括应用层协议分布、会话大小、会话时间间隔、链接数目、链接到达速率等;另外一类是网络特征参数,包括数据包丢包率、传输延迟和链路带宽.从捕获流量中提取出这些参数之后,Vishwanash 和 Vahdat 通过搭建模拟环境(包括两端的流量产生器和中间的网络仿真器)来产生流量.具体产生流量的方法是在流量产生器中设置链接产生时间、链接大小、链接持续时间等相关流量特征参数,而在网络仿真器中设置丢包率、传输延迟等网络参数.与 Swing 系统相比,本文工作虽然也是通过搭建模拟环境来产生流量,但存在如下不同之处:

(1)本文工作通过在模拟环境中回放捕获的真实流量来产生用户需要的流量,无需对真实流量中的特征参数进行提取.

(2)本文工作通过模拟 TCP/IP 协议栈来回放捕获的网络流量,产生的数据是真实的网络流量(数据包内容与原始网络流量相同).而 Swing 系统则是通过在流量产生器中建立真实 TCP 链接来传输一定大小人工仿真数据的方法来产生流量.因此,两者产生流量的机理不同.Swing 系统更多地关注产生流量的特征参数符合真实网络特性,对流量中传输内容并不关注.而本文工作则关注流量内容的真实性.

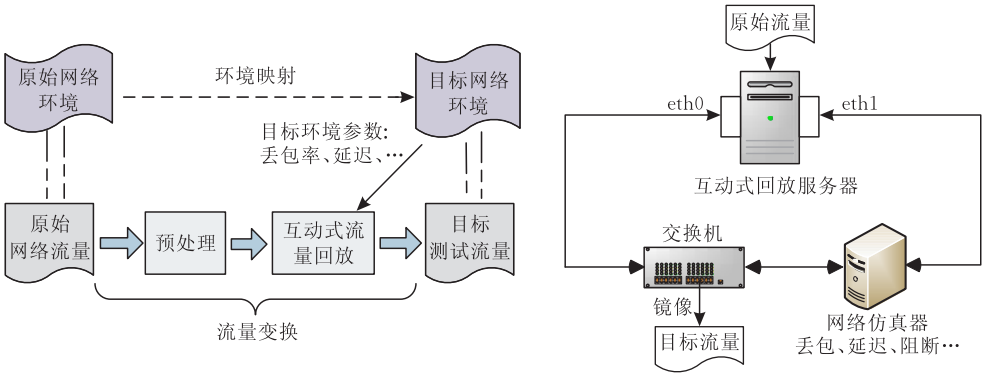
(3)本文主要研究回放过程在受到传输网络作用的条件下的流速控制问题,而 Swing 系统则主要解决产生满足网络环境特征的流量(产生流量的特征参数分布符合真实网络流量)的问题.

3 互动式网络场景再现流速控制

3.1 基于互动式场景再现的流量产生方法

除了用于设备测试,互动式网络场景再现还可用来当作真实网络流量的产生机制,用于产生目标网络环境下符合协议语义的真实流量.由于实际网络流量通常是在某个特定时间特定网络环境下捕获的,而对设备进行测试时往往又需要一些具有指定网络环境的流量(比如需要一定丢包率和网络延迟环境的真实流量,用来当作背景流量对 IDS 等设备进行功能和性能测试,而实际这样的网络环境却不容易获得).在这种情况下,就需要对从实际网络中捕获到的真实流量进行变换(从流量捕获时的原始网络环境映射到目标网络环境),使之产生目标网络环境下满足协议语义的流量.由于网络流量的复杂性,直接基于原始流量产生目标流量的仿真方法很容易破坏产生流量的协议语义,而互动式流量回放方法具有保持流量协议语义的特性,因此可用来产生指定网络环境下的真实流量.

图 1 是基于互动式流量回放过程产生目标流量的方法示意图.如图 1(a)所示,在具体应用该方法之前,一般需要先对原始流量进行处理(比如去除重



(a)过程示意图 (b)系统示意图

图 1 基于互动式流量回放过程产生目标流量的方法示意图

传的数据包、过滤掉不需要的流量等). 之后，将一台可配置丢包率、数据包传输延迟等环境参数的网络仿真设备当作中间设备串接在互动式流量回放系统中(该网络仿真设备用于模拟目标网络环境，见图 1(b))，通过控制互动式流量回放过程产生指定网络条件下的符合协议语义的目标流量.

3.2　互动式网络场景再现过程的流速控制

互动式网络场景再现的核心是流量的双向收发以及基于状态的流量产生机制. 在系统实现中涉及到磁盘 IO、网络 IO、会话状态维护和管理等多个方面. 基于流量在回放时的具体处理流程以及系统实现时的性能考虑(如磁盘 IO 和网络 IO 的并行化等)，互动式流量回放过程通常包含如下 5 个子过程(如图 2)：

① 数据包输入子过程. 该过程从磁盘文件中依序读入数据包，并至于系统的输入缓冲区中.

② 流量回放子过程. 该过程从系统输入缓冲区中获取数据包，并根据对应会话的状态，采用基于状态判定的方法尝试发送该数据包. 如果待发送数据包通过状态判定，则回放系统直接调用对应的网络测试接口将数据包发送出去并更新对应会话的状态；否则，该数据包将被缓存于系统中并等待会话状态更新后再次发送.

③ 数据包接收子过程. 该过程从回放系统的网络测试接口中提取到达的数据包，并至于系统的接收数据包缓冲区中.

④ 会话接收更新子过程. 该过程从回放系统的接收数据包缓冲区中取出到达系统的数据包，并根据数据包包含的协议状态信息更新对应会话的状态；在会话状态更新后，该过程还负责将缓存于系统中满足发送条件的数据包发送出去.

⑤ 超时重传子过程：该过程负责将当前系统已经发送出去但尚未被系统另外一个测试接口接收到的超时的数据报文依据网络协议规范进行重传.

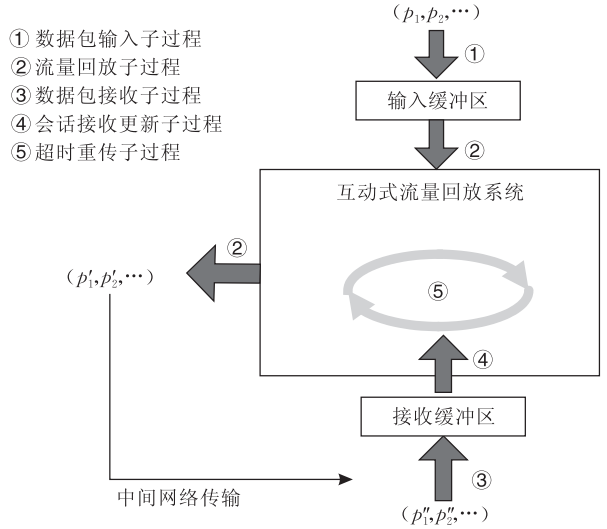


图 2　互动式网络流量回放系统的设计实现

本文考虑在互动式流量回放过程中，通过调整系统输入流量速率来控制回放过程输出流量. 由于不知道具体数学模型，因此考虑基于系统输入输出信息来对系统进行控制. 该问题建模成离散时间目标跟踪控制问题，具体描述如下：

将互动式流量回放过程视为一个动态的含有噪声的非线性过程，其中系统数学方程并不可知，但却可以通过测量得到一系列离散时间回放过程输出流量信息. 记回放控制的目标流量大小为 $t_1,t_2,\cdots$，测量得到的离散时间系统输出流量大小为 $y_1,y_2,\cdots$，相应的输入流量控制为 $u_0,u_1,\cdots$. 其中，当前时刻 k 的系统输入 u_k 影响 y_{k+1}，$y_{k+2},\cdots$. 基于到当前为止的历史流量输出信息 $y_1,y_2,\cdots,y_k$ 以及历史控制 $u_0,u_1,\cdots,u_{k-1}$，回放过程流速控制的目标是决定当前时刻 k 的输入 u_k，以便在该输入下回放过程产生的下一个时刻的输出 y_{k+1} 满足目标值 t_{k+1}.

4　影响网络场景再现过程产生流速的因素分析

本节将从网络环境行为以及网络流量特性两大方面对影响互动式网络场景再现过程产生流速的因素进行分析.

4.1　网络环境行为

互动式网络场景再现过程的本质是模拟 TCP/IP 协议栈来对 TCP 流量传输过程进行再现. 因此，所有影响 TCP 传输性能的网络行为都会影响到回放过程. 以下分别从网络丢包、传输延迟、会话阻断以及报文乱序等几方面进行分析，并对这些因素影响 TCP 流量回放过程的机理进行说明.

(1) 网络丢包. 网络传输过程中的数据包丢包行为会影响到 TCP 流量的传输性能. 根据 TCP/IP 协议规范，丢包是网络发生拥塞的表现. TCP 发送端在通过超时机制感知到丢包发生时，会减小数据发送窗口并进入拥塞避免阶段[24]，同时对丢失的数据报文进行重传. 由于发送窗口减小，TCP 数据传输速率也随之降低.

(2) 传输延迟. 数据包端到端传输延迟也会影响到 TCP 流量的传输性能. 根据文献[25]，TCP 会话的吞吐率与网络传输延迟近似成反比关系，即网络

传输延迟越大,则 TCP 流量的传输性能也就越低.

(3)会话阻断.在实际的流量传输过程中,若一个 TCP 会话被阻断(如接收到 Firewall 发送的 RST 报文),则该会话将会被终止,且其后续的流量将不再传输.

(4)报文乱序.报文乱序对 TCP 传输性能的影响体现在接收端的确认行为上.当发送方传输的数据报文在网络中发生乱序时,接收方将需要缓存乱序到达的数据报文[26].对于采用累计确认的 TCP 协议实现,报文乱序将明显降低 TCP 的数据传输效率.

4.2 输入流量特性

除了丢包、传输延迟等网络环境行为,输入流量特性同样会影响到回放过程.由于被回放流量通常是由多个 TCP 会话组成,因此,从单个 TCP 会话特性以及会话并发两个方面进行分析.

(1)单个 TCP 会话特性:从真实网络中采集的流量包含多种应用,产生的 TCP 会话特性也会根据应用的不同而不同.参考文献[1],我们将 TCP 会话大体分成两类,一类是交互式会话,对应的应用包括 Telnet、SSH、Rlogin、VoIP 等;另外一类是成块数据传输会话,如 FTP 文件传输、文件共享等.对于交互式会话,一般来说这类会话的流速较低,数据包之间的时间间隔相对较长,因此,在相同的输出流速下需要并发的会话个数较多;而成块数据传输的会话对应的流速较高,数据包之间的时间间隔较短,通常情况下同一时刻发送窗口中会有多个数据报文进行传输.因此,在相同的输出流速下需要的并发会话个数也较少.

(2)输入流量的会话并发:会话并发会显著改变流量回放过程输出速率.站在回放过程流速控制的角度来看,当网络丢包、传输延迟、会话阻断以及报文乱序等现象越严重时,单个 TCP 会话的传输性能也就越低.因此,为了产生相同速率的流量,回放系统需要在同一时刻回放尽可能多的并发会话,也即在系统控制时需要施加更大的流量输入速率.而当被回放流量中同一时刻并发会话数目达不到要求时(比如,输入流量中只包含少数几个会话),则无论施加多大输入流量,系统都不可能产生期望输出速率(在本文实验部分可以观察到).

5 基于 FA 和 SPSA 的互动式场景再现过程流速控制系统和方法

由于互动式流量产生机制异常复杂,很难通过内部机理分析建立精确的数学模型,因此考虑在系统的输入和输出层面来对系统进行建模并对产生流速进行控制.以下具体介绍基于函数近似器(Function Approximator,FA)和并行扰动随机估计理论(SPSA)的流速控制系统和方法.

5.1 互动式网络场景再现过程的 FA 控制方法

将流量回放系统当作一个黑盒(如图 3),采用函数近似器来产生控制输入并对系统进行控制.函数近似器用来描述目标流量和需要施加流量之间的映射,具体数学形式可以采用多层神经网络、代数多项式、三角型序列等.根据 Stone-Weierstrass 理论[27],这些近似器均可对任意一个数学函数进行精确逼近.

我们考虑采用结构固定的函数近似器(如多层神经网络中层数和节点数目一定、代数多项式的阶数一定),但却可以让近似器的参数随时间动态变化(比如权重可动态改变).函数近似器的输入是系统观测到的历史测量数据和历史控制信息(还可包含目标流量信息),而相应的输出则是系统当前时刻产生的控制.假设时刻 k 函数近似器的输入包含前 M 个历史测量数据和前 N 个历史控制信息,当函数近似器参数不变化时,近似器的输入可简单记为

$$I_k = \{y_k, y_{k-1}, \cdots, y_{k-M+1}; u_{k-1}, u_{k-2}, \cdots, u_{k-N}\} \tag{1}$$

当函数近似器的参数动态更新时(由扰动机制产生,在下面小节将会介绍),上述近似器的输入信息中还将包含一部分参数动态更新时得到的数据.

图 3 是基于函数近似器进行流速控制的互动式流量回放系统示意图.在该系统中,当前时刻 k 函数近似器的控制输入是目标流量信息 t_{k+1} 和前一个时刻的系统观测信息 $I_k=\{y_k\}$,而近似器的输出则直接对应当前时刻产生的控制 u_k.

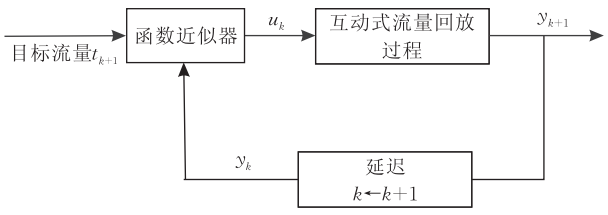


图 3 基于函数近似器的互动式流量回放过程流速控制系统框图(其中 $I_k=\{y_k\}$)

5.2 基于 SPSA 方法的参数估计

在基于函数近似器的控制系统和方法中,函数近似器被用来根据历史信息产生当前时刻需要的输入.然而,由于不知道近似器包含的参数,因此需要对近似器的参数 θ_k 进行实时估计.在假定函数近似

器结构固定的条件下，近似器的参数估计问题等同于求取使得代价函数 $L_k(\theta_k)$ 最小化的参数 $\theta_k \in R^p$（由于近似器结构固定，p 是与 k 无关的一个常量，代表参数个数）. 代价函数 $L_k(\theta_k)$ 可采用如下二次型：

$$L_k(\theta_k)=E[(\boldsymbol{y}_{k+1}-\boldsymbol{t}_{k+1})^{\mathrm{T}}\boldsymbol{A}_k(\boldsymbol{y}_{k+1}-\boldsymbol{t}_{k+1})+\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{u}_k] \tag{2}$$

式中 $\boldsymbol{A}_k$ 和 $\boldsymbol{B}_k$ 是两个半正定矩阵，反映控制器对于输出流量和施加控制的不同权重；$\boldsymbol{y}_{k+1}$，$\boldsymbol{t}_{k+1}$，$\boldsymbol{u}_k$ 分别是由标量 y_{k+1}，t_{k+1} 和 u_k 组成的向量.

上述函数近似器的参数估计问题等价于在时刻 k 求取 θ_k^*，使之满足如下方程：

$$g_k(\theta_k)=\frac{\partial L_k}{\partial \theta_k}=\frac{\partial \boldsymbol{u}_k^{\mathrm{T}}}{\partial \theta_k}\cdot\frac{\partial L_k}{\partial \boldsymbol{u}_k}=0 \tag{3}$$

其中，$g_k(\theta_k)$ 即为代价函数 $L_k(\theta_k)$ 对参数 $\theta_k \in R^p$ 的梯度. 然而，由于系统数学模型并不可知，式(3)中 $\partial L_k/\partial \boldsymbol{u}_k$ 项便无法求得. 因此，传统基于目标函数梯度信息进行求解的方法或其它一些利用 $g_k(\theta_k)$ 进行参数求解的方法便不再适用.

为了获取系统参数 $\{\theta_k\}$，我们考虑采用随机估计的方法：

$$\hat{\theta}_k=\hat{\theta}_{k-1}-a_k\times(gradientapprox)_k \tag{4}$$

其中，$\hat{\theta}_k$ 是 k 时刻近似器参数 θ_k 的估计值，$\{a_k\}$ 是一个满足一定条件的标量序列. 此处，采用并行扰动随机估计方法(SPSA)[20] 进行参数估计，该方法具有如下形式：

$$\hat{\theta}_k=\hat{\theta}_{k-1}-a_k\,\hat{g}_k(\hat{\theta}_{k-1}) \tag{5}$$

式(5)中 $\hat{g}_k(\hat{\theta}_{k-1})$ 是扰动后 $g_k(\hat{\theta}_{k-1})$ 的估计值. $\hat{g}_k(\hat{\theta}_{k-1})$ 的第 l 个成员变量($l=1,2,\cdots,p$)可以由如下式子求得：

$$\hat{g}_{kl}(\hat{\theta}_{k-1})=\frac{\hat{L}_k^{(+)}-\hat{L}_k^{(-)}}{2c_k\Delta_{kl}} \tag{6}$$

在式(6)中：

(1) $\hat{L}_k^{(\pm)}$ 是利用 $y_{k+1}^{(\pm)}$ 和 $u_k^{(\pm)}$ 对 $L_k(\hat{\theta}_{k-1}\pm c_k\Delta_k)$ 进行计算得到的值. 对于形如式(2)的代价函数，$\hat{L}_k^{(\pm)}=(\boldsymbol{y}_{k+1}^{(\pm)}-\boldsymbol{t}_{k+1})^{\mathrm{T}}\boldsymbol{A}_k(\boldsymbol{y}_{k+1}^{(\pm)}-\boldsymbol{t}_{k+1})+\boldsymbol{u}_k^{(\pm)\mathrm{T}}\boldsymbol{B}_k\boldsymbol{u}_k^{(\pm)}$.

(2) $u_k^{(\pm)}$ 是基于扰动后的参数 $\theta_k=\hat{\theta}_{k-1}\pm c_k\Delta_k$ 时得到的函数近似器的控制输出，其中 $\Delta_k=(\Delta_{k1},\Delta_{k2},\cdots,\Delta_{kp})^{\mathrm{T}}$ 是一个随机序列. 通常情况下，$\{\Delta_{ki}\}$ 是独立同分布、有界、相对于 0 对称分布的随机量，并且 $\forall k,i$ 满足 $E(\Delta_{ki}^{-2})$ 一致有界的条件.

(3) $y_{k+1}^{(\pm)}$ 是当控制为 $u_k^{(\pm)}$ 时回放系统产生的流量测量值.

(4) $\{c_k\}$ 是满足一定条件的正实数序列. 通常情况下，根据被控系统是否稳定取 $c_k\to 0$ 或者 $c_k=c$.

基于并行扰动随机估计理论进行参数估计的算法如图 4 所示.

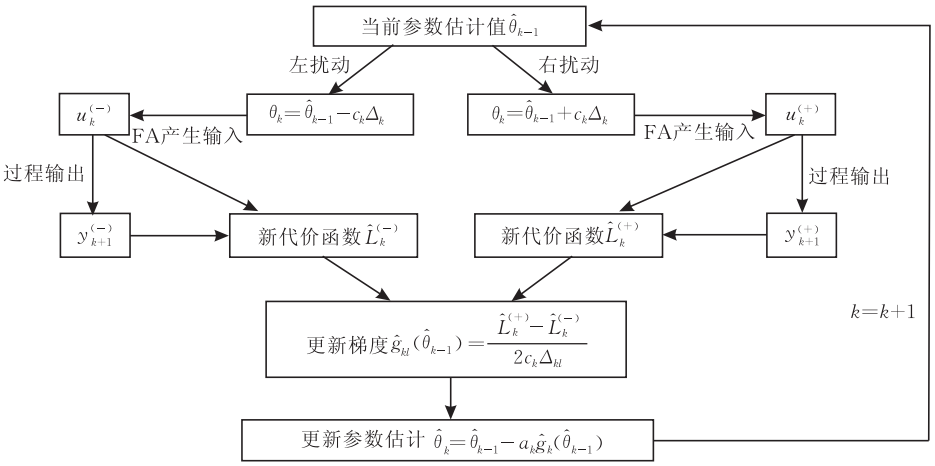


图 4　并行扰动随机估计方法的算法步骤

当扰动机制引入到控制过程时，函数近似器的输入将包含参数扰动后得到的系统测量值和控制输入. 此时，系统的测量值变为 $\{y_0,y_1^{(+)},y_1^{(-)},y_1,y_2^{(+)},y_2^{(-)},y_2,\cdots\}$，而对应的控制输入也将变为 $\{u_0,u_1^{(+)},u_1^{(-)},u_1,u_2^{(+)},u_2^{(-)},u_2,\cdots\}$. 由于每一个产生的控制均是基于前 M 个历史测量数据和前 N 个历史控制信息，对于 $u_k^{(+)}$、$u_k^{(-)}$ 和 u_k，相应的函数近似器的输入就变成了集合 $I_k^{(+)}$、$I_k^{(-)}$ 和 I_k. 比如，当 $M=N=2$ 且当前最新系统测量值为 $y_{k+1}^{(+)}$ 时，下一时刻系统的控制 $u_k^{(-)}$ 是基于参数 $\theta_k=\hat{\theta}_{k-1}-c_k\Delta_k$ 和集合 $I_k^{(-)}=\{y_{k+1}^{(+)},y_k,u_k^{(+)},u_{k-1}\}$ 来产生的.

5.3　流量回放过程流速控制流程

基于函数近似器的流速控制系统在采用 SPSA 算法进行参数估计后的具体流速控制过程如算法 1.

算法 1. 基于函数近似器和 SPSA 的流量回放过程流速控制流程.

1. 读入当前时刻 k 的参数估计值 $\hat{\theta}_{k-1}$ 和当前时刻近似器的输入 I_k;

2. 参数右扰动 $\theta_k=\hat{\theta}_{k-1}+c_k\Delta_k$,函数近似器根据扰动后的参数和当前输入信息产生需要施加的流量 $u_k^{(+)}$;

3. 测量流量回放过程的输出流量大小 $y_{k+1}^{(+)}$;

4. 计算参数右扰动后的代价函数 $\hat{L}_k^{(+)}$;

5. 更新当前函数近似器的输入信息;

6. 参数左扰动 $\theta_k=\hat{\theta}_{k-1}-c_k\Delta_k$,函数近似器根据扰动后的参数和当前输入信息产生需要施加的流量 $u_k^{(-)}$;

7. 测量流量回放过程的输出流量大小 $y_{k+1}^{(-)}$;

8. 计算参数左扰动后的代价函数 $\hat{L}_k^{(-)}$;

9. 更新当前函数近似器的输入信息;

10. 根据参数左右扰动后的代价函数变化信息更新当前参数 $\hat{\theta}_k=\hat{\theta}_{k-1}-a_k\hat{g}_k(\hat{\theta}_{k-1})$;

11. 当前时刻 $k=k+1$;

12. 返回步 1,直到流量回放过程结束.

采用函数近似器和 SPSA 方法进行系统控制的最大优点在于可以在并不知道系统模型的情况下仍然对动态系统进行精确控制. 然而,引入参数扰动机制也给该方法和对应的控制系统在具体应用时在输入输出和收敛时间等方面带来了一些副作用. 我们将在实验部分对此进行详细论述.

6 实验验证及结果分析

6.1 实验环境设置

为考察设计的系统和方法的实际控制效果,实现了一个高性能的互动式网络流量回放测试系统,如图 5 所示. 整个测试系统由组成测试回路的流量回放服务器、千兆交换机、FreeBSD 双穴主机(当作网络仿真器)和流量监测器组成. 其中,流量回放服务器由一台高性能 Linux 服务器充当(Redhat 9.0 系统,2.4.20 内核),配置 2×2.0 GHz Intel Xeon 处理器,4 GB RAM.

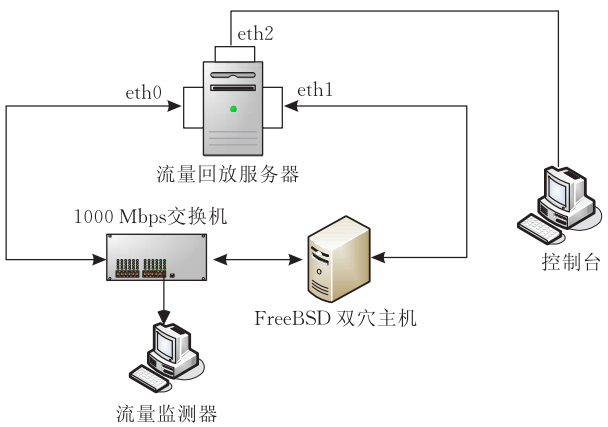


图 5 流量回放实验环境

由于 TCP 协议栈的实现在不同的操作系统平台以及不同系统的版本下存在差异,本文流量回放服务器上的 TCP 协议栈的模拟实现参考了 TCP-Opera,具体也是基于 BSD 4.4-Lite RELEASE 平台的 TCP 协议栈实现[24]. 目前实现的一些核心功能包括:6 个 TCP 计时器、传输过程中的数据包超时和重传、快速重传和快速恢复、拥塞控制以及 RTT 动态测量等.

网络仿真器的实现采用的是 FreeBSD 平台上的 Dummynet① 这一开源工具. Dummynet 可用于模拟多种复杂的网络环境,对包括传输延迟、网络带宽、丢包率以及排队策略等在内的多项参数进行选择设置,是当前网络研究领域最常用的网络仿真工具.

在我们的实验中,将网络仿真平台(安装 FreeBSD 8.1-RELEASE 系统)的两块网卡设置成网桥模式进行数据包转发,同时配合 Dummynet 对网桥传输流量的数据包延迟和丢包率进行控制;对回放过程会话阻断功能的实现则是通过预先在输入流量中对需要阻断会话的数据包的 MAC 地址进行标定,而后在网络仿真平台上通过 IP Firewall 设置相应的规则(根据 MAC 地址设置)进行阻断.

此外,流量回放服务器和 FreeBSD 双穴主机均配备 Intel e1000 千兆网卡,整个测试回路是一个纯千兆的环境. 在千兆交换机上将回放产生的流量镜像到流量监测器上进行观测.

实验中采用真实网络流量对本文方法进行考察. 所用流量在西安交通大学校园网出口处采集得到,共包含超过 20000000 个 TCP 数据包和 300000 个 TCP 会话. 实验中对串接在测试回路中的 FreeBSD 双穴主机设置不同的丢包率和传输延迟,同时在输入流量中以不同的概率对阻断会话进行标识,以此来模拟不同的目标网络环境.

6.2 实验结果及分析

考虑丢包、数据包传输延迟以及会话阻断 3 种不同的网络行为,实验过程中设置 FreeBSD 双穴主机分别以 0.1%,1%,5%和 10%的概率进行随机丢包,以 0.1 ms,1 ms,10 ms 进行数据包传输延迟. 另外,设置以 10%,20%和 30%的概率进行随机会话阻断. 这样设置 FreeBSD 双穴主机的丢包率、传输

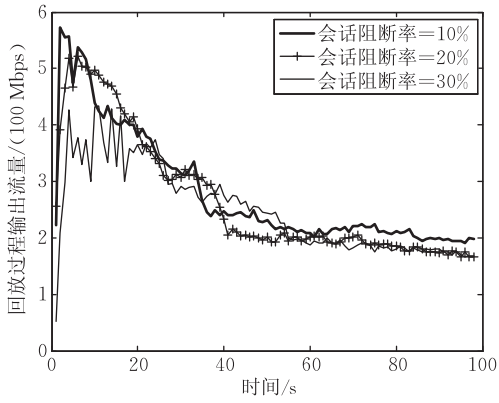
① http://info.iet.unipi.it/~luigi/ip_dummynet/

延迟以及回放过程会话阻断率的目的在于在一个较为宽广的目标网络环境下考察所提方法的实际控制性能.

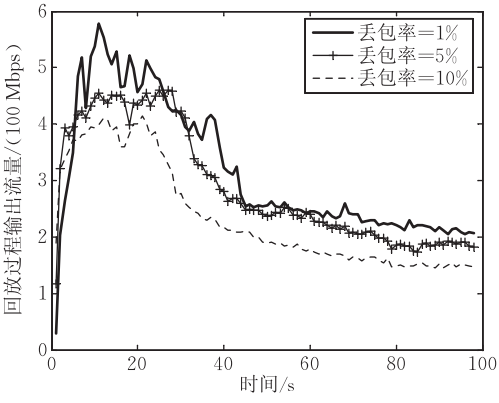
实验过程中将需要产生的目标流量大小设置为100 Mbps,离散采样时间取为0.5 s.对每个不同设置的实验均重复进行10次,最后取平均性能进行分析.

6.2.1　未施加控制时的回放性能

首先考察系统在不施加任何控制时的回放性能,实验结果如图6所示.在实验中观察到整个真实网络流量的互动式回放过程是一个非常不平稳的过程;受到丢包和会话阻断等网络因素的影响,系统性能发生显著变化,而且,两种因素对产生流量的影响较为复杂,难以用一个确切的数学方程进行描述.可以预见,在同时有丢包、会话阻断以及传输延迟等多种因素存在的网络环境下,系统性能的变化将变得更加复杂.



(a) 会话随机阻断情况



(b) 数据包随机丢包情况

图6　没有施加控制的情况下回放系统产生的流量大小

6.2.2　采用FA和SPSA方法的回放结果

(1) 方法参数选取

考察基于函数近似器和并行扰动随机估计理论的控制系统和方法的控制效果.实验中利用神经网络当作函数近似器,该神经网络的节点采用$1/(1+e^{-z})$的输入输出函数,每个节点的输入包括前一层节点的输出加权和再加上一个属于自身节点的偏移量.整个函数近似器的输入包括到当前时刻为止的前两个时刻回放系统产生流量值($M=2$)、上一个时刻的输入控制量($N=1$)和下一个时刻的目标流量大小,近似器的输出则是当前时刻需要施加的控制量.因此,作为函数近似器的神经网络具有4个输入节点,1个输出节点.假定包含两个隐层,每个隐层有5个节点,则该神经网络可记为$N_{4,5,5,1}$,共包含$(4\times5+5)+(5\times5+5)+(5\times1+1)=61$个参数需要同时估计.神经网络的具体结构如图7所示(实验中发现采用结构更为复杂的神经网络并没有显著提升控制性能,但却增加了计算量).

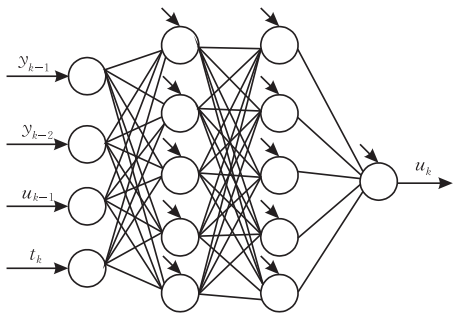


图7　作为函数近似器的神经网络

在SPSA算法中,参数$\{\Delta_{ki}\}$取为伯努利±1分布,$a_k=a=0.05$,$c_k=c=0.005$,作为函数近似器的神经网络的权值初始量取为标准正态分布$N(0,1)$;式(2)中代价函数的参数$\boldsymbol{A}_k=[1]$,$\boldsymbol{B}_k=[10]$.

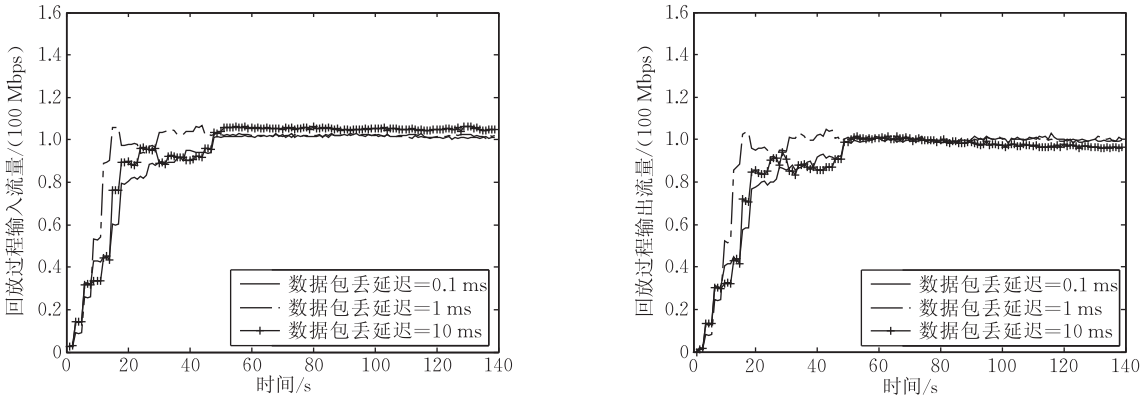
(2) 实际控制结果与分析

图8、图9以及图10分别是在不同数据包传输延迟、会话阻断和丢包情况下控制系统产生的输入流量大小和输出流量大小.

从实验结果中可以得到如下结论:

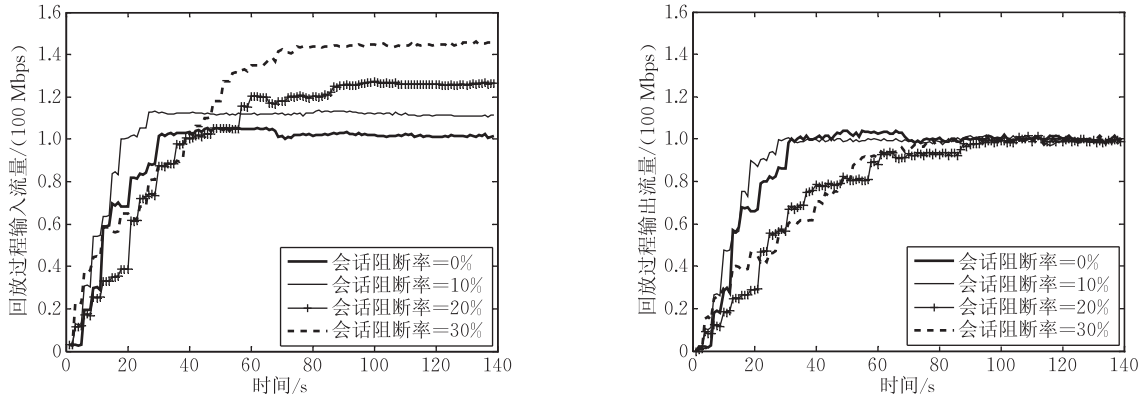
(1) 在互动式流量回放控制过程中,系统的输入流量和输出流量之间的关系确实是非线性的.比如从图9(a)中可以看出,为了追踪目标流量,系统在10%的会话阻断率下达到稳定时需要施加的输入流量约为110 Mbps,在20%的情况下为128 Mbps,而在30%的情况下需要施加的输入流量上升到145 Mbps.

(2) 在达到稳定状态时,基于FA和SPSA方法的控制系统在会话阻断、丢包和数据包传输延迟3种网络因素影响下均能够精确地对回放过程产生流速进行控制.在测试网络丢包率达到10%以及会话阻断率达到30%等极端情况下,系统仍然能够达到很好的控制效果.



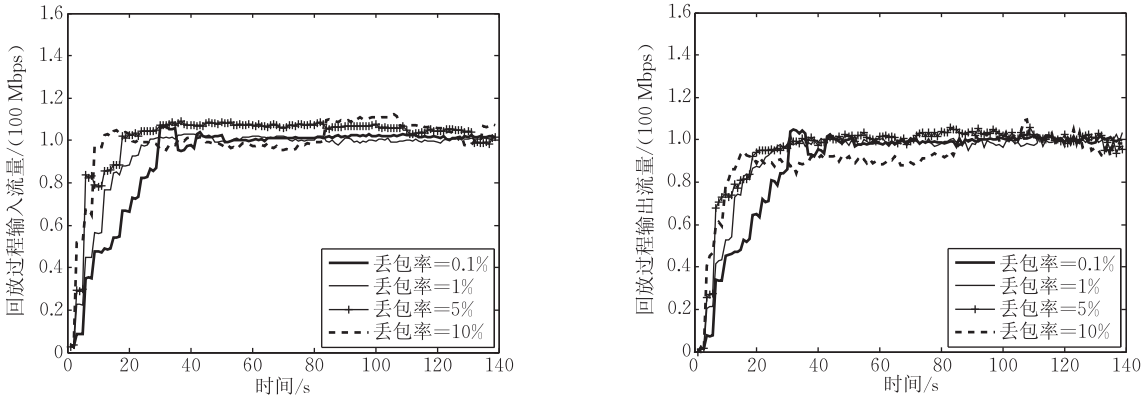
(a) 施加的输入流量大小 (b) 回放系统输出流量大小

图 8 基于函数近似器的控制系统在不同数据包传输延迟情况下产生输入和输出流量大小



(a) 施加的输入流量大小 (b) 回放系统输出流量大小

图 9 基于函数近似器的控制系统在不同会话阻断率的情况下产生输入和输出流量大小



(a) 施加的输入流量大小 (b) 回放系统输出流量大小

图 10 基于函数近似器的控制系统在不同数据包丢包率的情况下产生输入和输出流量大小

(3) 基于 FA 和 SPSA 方法的控制系统在不同网络环境因素影响下所需要的收敛时间不同. 总体上来说,在随机丢包情况下系统所需要的收敛时间最少,而在会话阻断情况下需要的收敛时间最长.

(4) 从稳定误差上面来看,在会话阻断、丢包和数据包传输延迟 3 种网络影响因素中,基于 FA 和 SPSA 方法的控制系统在会话随机阻断和数据包传输延迟情况下的控制效果要好于在随机丢包情况下(见图 8(b)、图 9(b)和图 10(b)). 我们认为引起该现象的主要原因在于不同网络事件引起的回放系统行为复杂性不同. 数据包传输延迟是一个相对较为简单的网络行为;当会话被阻断时(接收到 RST 报文),回放系统将简单地结束当前流的回放并且后续输入流量中所有属于该会话的数据包将不再被回放出去. 因此,会话阻断情况下系统的行为也相对简单. 而当数据包发生丢包时,回放系统将执行超时重

传行为.根据 TCP/IP 协议规范,TCP 数据报文的超时重传机制是一个非常复杂的过程[28-29].因此,随机丢包情况下的系统行为是最复杂的,相应地导致系统在随机丢包情况下的控制效果要低于在会话随机阻断以及数据包传输延迟时的情况.

(5)对于数据包传输延迟在 10 ms 的情况下,发现基于 FA 和 SPSA 方法的控制系统产生的输出流量在第 80 个采样周期后缓慢下降而跟踪不上目标流量(见图 8(b)).我们猜测引起该现象的原因在于输入流量的特性,即当数据包传输延迟在 10 ms 时,输入流量中的并发会话个数不足以使得回放过程跟踪上目标流量.通过监测流量回放过程发现此时并发会话个数约为 80.同时,在第 80 个时刻之后人为设置输入流量大小持续增长,同样发现系统的实际输出流量不再随着输入流量的增长而增长.因此,并发会话的个数会决定输出流量的大小,实验结果与我们的分析一致.

此外,我们将基于 FA 和 SPSA 的控制方法与单输入单输出系统的一些自适应控制方法进行了比较,发现有如下性质:

(1)稳定误差. 基于 FA 和 SPSA 的方法产生的误差要优于一般的自适应控制方法,尤其是在大丢包率的情况下.

(2)收敛时间. 基于 FA 和 SPSA 的方法需要的收敛时间要比一般的自适应控制方法长.而且,一般的自适应控制方法的收敛时间比较稳定,而基于 FA 和 SPSA 的方法在丢包、会话阻断以及不同数据包传输延迟情况下则相差较大.

(3)产生输入和输出.在系统稳定之前基于 FA 和 SPSA 的方法产生的输入和输出要比一般的自适应控制方法波动大.

对于基于 FA 和 SPSA 的方法与一般的自适应控制方法在收敛时间和产生输入输出方面的差异,我们认为主要原因是由引入的扰动机制造成的.为了对参数进行估计,基于 FA 和 SPSA 的方法通过在参数空间随机扰动来获得代价函数对参数的变化信息.而一般的自适应方法则是直接基于系统输出反馈信息产生需要的输入.因此,随机扰动增加了产生输入和输出的波动性,同时也使得方法和系统需要更多数据量和收敛时间来跟踪上目标流量大小.

下面,通过表 1 所示的数据对基于 FA 和 SPSA 方法的控制系统在稳定时的控制效果进行更为细致的分析和说明.从平均误差和误差方差上可以看出,总体上基于 FA 和 SPSA 方法的控制系统具有较好的稳定性.

表 1　基于函数近似器的控制系统在不同网络环境下稳定时的控制效果

性能指标		平均误差	误差方差
丢包率/%	0.1	0.0032	1.54E-4
	1.0	0.0068	2.19E-4
	10.0	0.0010	1.20E-3
会话阻断/%	10.0	0.0047	1.01E-4
	20.0	0.0071	8.33E-5
	30.0	0.0022	7.84E-5
数据包传输延迟/ms	0.1	0.0019	6.68E-5
	1.0	0.0029	1.24E-4
	10.0	0.0161	2.98E-4

利用其它网络流量对所提方法和系统进行了实验考察,得到的结果同上述结果基本一致,基于文章篇幅考虑此处便不再陈列.从实验结果中可以发现,本文所设计的系统与方法可以在一个较为宽广的网络环境下对回放系统产生的流量速率进行有效控制.

7　结论和未来展望

本文对互动式网络场景再现过程的流速控制问题进行了研究,设计实现了一个基于函数近似器的流速控制系统,并采用真实网络流量对系统的实际性能进行了实验验证与分析.

在未来,将重点进行如下工作:

(1)定量考察网络延迟、丢包和会话阻断等不同网络行为以及流量的各种特性如会话到达速率、会话大小分布等对回放过程性能的影响.

(2)在上述工作基础上,尝试对回放系统建立精确的数学模型.虽然采用无模型控制方法可以达到较为满意的控制效果,然而,精确的数学模型仍然是更好地对系统施加控制,并进行性能分析、最优化设计和稳定性分析等多项重要工作的前提.

参　考　文　献

[1] Danzig P B, Jamin S. Tcplib: A library of tcp internetwork traffic characteristics. Computer Science Department, University of Southern California, Technical Report USC-CS-91-495, 1991

[2] Terdik G, Gyires T. Lévy flights and fractal modeling of internet traffic. IEEE/ACM Transactions on Networking, 2009, 17(1): 120-129

[3] Dainotti A, Pescape A, Rossi P S, Palmieri F, Ventre G. Internet traffic modeling by means of hidden markov models. Computer networks, 2008, 52(14): 2645-2662

[4] Anand N C, Scoglio C, Natarajan B. GARCH-non-linear time series model for traffic modeling and prediction//

Proceedings of the 2008 IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubiquitous Networks and Services. Salvador, Bahia, Brazil, 2008: 694-697

[5] Chimeh J D, Hakkak M, Azmi P. Internet traffic modeling and capacity evaluation in UMTS. International Journal of Hybrid Information Technology, 2008, 1(2): 109-120

[6] Fras M, Mohorko J, Cucej Z. Packet size process modeling of measured self-similar network traffic with defragmentation method//Proceedings of the 15th International Conference on Systems, Signals and Image Processing. Bratislava, Slovakia, 2008: 253-256

[7] Wu C F, Goel A, Bezzaz A et al. TCPivo: A high performance packet replay engine. MoMeTools'03//Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research, 2003: 57-64

[8] Cheng Y, Hölzle U, Cardwell N, Savage S, Voelker G M. Monkey see, Monkey do: A tool for TCP tracing and replaying//USENIX Annual Technical Conference, General Track. Boston, MA, USA, 2004: 87-98

[9] Ye Tao, Veitch D, Iannaccone G, Bhattacharyya S. Divide and conquer: PC-based packet trace replay at OC-48 speeds//Proceedings of the 1st International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENTCOM 2005). Trento, Italy, 2005: 262-271

[10] Van P D, Zhanikeev M, Tanaka Y. Effective high speed traffic replay based on IP space//Proceedings of the 11th International Conference on Advanced Communication Technology. Gangwon-Do, 2009, 1: 151-156

[11] Chu Wei-Bo, Guan Xiao-Hong, Cai Zhong-Min, Chen Ming-Xu. A traffic splitting method of high-speed network using sub-network flow optimization. Journal of Xi'an Jiaotong University, 2011, 45(12): 22-27(in Chinese)
(褚伟波，管晓宏，蔡忠闽，陈明旭. 采用子网流量组合优化的网络流量分割方法. 西安交通大学学报，2011，45(12)：22-27)

[12] Hong S, Wu S F. On interactive internet traffic replay//Recent Advances in Intrusion Detection. Seattle, Washington, USA, 2005: 247-264

[13] Chen Zhong-Qiang, Delis Alex, Wei Peter. A pragmatic methodology for testing intrusion prevention system. Computer Journal, 2009, 52(4): 429-460

[14] Chu Wei-Bo, Guan Xiao-Hong, Cai Zhong-Min, Chen Ming-Xu. Balance based performance enhancement for interactive TCP traffic replay//Proceedings of the 2010 IEEE International Conference on Communications (ICC 2010). Cape Town, South Africa, 2010: 1-5

[15] Chu Wei-Bo, Guan Xiao-Hong, Cai Zhong-Min, Chen Ming-Xu. A new method for interactive TCP traffic replay based on balance-checking between transmitted and received packets. Chinese Journal of Computers, 2009, 32(4): 835-846(in Chinese)
(褚伟波，管晓宏，蔡忠闽，陈明旭. 基于收发平衡判定的TCP流量回放方法. 计算机学报，2009，32(4)：835-846)

[16] Kornexl S, Paxson V, Dreger H, Feldmann A, Sommer R. Building a time machine for efficient recording and retrieval of high-volume network traffic//Proceedings of the 5th ACM Internet Measurement Conference (IMC 2005). Berkeley, California, 2005: 267-272

[17] Qian F, Gerber A, Mao Z M et al. TCP revisited: A fresh look at TCP in the wild//Proceedings of the 9th ACM internet measurement conference (IMC 2009). Chicago, Illinois, 2009: 76-89

[18] Labovitz C, Iekel-Johnson S, McPherson D et al. Internet inter-domain traffic//Proceedings of the ACM SIGCOMM 2010 Conference. New York, USA, 2010: 75-86

[19] Zhang Yin, Breslau Lee, Paxson Vern, Shenker Scott. On the characteristics and origins of internet flow rates//Proceedings of the ACM SIGCOMM Computer Communication Review. Pittsburgh, PA, USA, 2002

[20] Spall J C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions Automatic Control, 1992, 37(3): 332-341

[21] Sommers J, Barford P. Self-configuring network traffic generation//Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (IMC 2004). Taormina, Sicily, Italy, 2004: 68-81

[22] Antonatos S, Anagnostakis K G, Markatos E P. Generating realistic workloads for network intrusion detection systems//Proceedings of the 4th Workshop on Software and Performance (WOSP'04). Redwood Shores, CA, 2004: 207-215

[23] Vishwanash K V, Vahdat A. Swing: Realistic and responsive network traffic generation. IEEE/ACM Transactions on Networking (TON), 2009, 17(3): 712-725

[24] Stevens W R, Write G R. TCP/IP illustrated (vol. 2): The implementation. Addison-Wesley Longman Publishing Co. Inc., 1995

[25] Klein Thierry E, Leung Kin K, Parkinson R, Samuel Louis G. Avoiding spurious TCP timeouts in wireless networks by delay injection//Proceedings of IEEE Globecom 2004. Dallas, TX, 2004: 2754-2759

[26] Piratla N M, Jayasumana A P. Metrics for packet reordering—A comparative analysis. International Journal of Communication Systems, 2008, 21(1): 99-113

[27] Rudin W. Principles of Mathematical Analysis. New York: McGrawHill, 1964

[28] Padhye Jitendra, Firoiu Victor, Towsley Donald F, Kurose James F. Modeling TCP Reno performance: A simple model and its empirical validation. IEEE/ACM Transactions on Networking(TON), 2000, 8(2): 133-145

[29] Paxson V. Automated packet trace analysis of TCP implementations//Preceedings of the SIGCOMM'97. Cannes, France, 1997: 167-179

CHU Wei-Bo, born in 1982, Ph. D. candidate. His research interests include internet measurement and modeling, traffic analysis and performance evaluation.

GUAN Xiao-Hong, born in 1955, Ph. D., professor, Ph. D. supervisor. His research interests include network security, system optimization and scheduling.

CAI Zhong-Min, born in 1975, Ph. D., associate professor. His research interests include networked systems, data mining and computer security.

TAO Jing, born in 1978, M.S.. His research interests include network simulation and network scenario reproduction.

Background

Generating realistic and responsive traffic workloads is of particular importance in many areas such as network security, network management, network testing and evaluation, etc. Recently, a new traffic generation method called interactive network traffic replay is proposed. Besides as a completely new method for testing in-line networking devices, this method is also capable of transforming traffic workloads under a wide range of "what-if" scenario due to its novel ability to maintain consistency of protocol semantics in the generated traffic.

This paper considers the problem of controlling output traffic volume in the nonlinear interactive traffic replay process, which is a rather challenging new problem due to the complex traffic generation mechnisms (i. e., the instability of input traffic, the changing dynamics of the replay system, the impact of testing environment such as transmission delay, packet loss, connection blocking, etc). In this paper, we design and implement a volume control system for interactive network traffic replay, and formulate the volume control task as a target tracking problem where the output traffic volume is regulated through adjustment of input traffic volume. We then adopt a function approximator (FA) in our system as a controller to generate the desired input volumes. The FA characterizes the mapping that takes system measurements as input and directly produces controls as output. Meanwhile, simultaneous perturbation stochastic approximation (SPSA) algorithm is employed to adaptively estimate the parameters of the FA in control. To validate the system and the method we have implemented it and conducted experiments with actual traffic traces. Empirical studies show that our system can track target output volumes effectively under a wide range of network conditions. The performance of the control system is further investigated in terms of its convergence time, generated input/output traffic volume and target tracking error.

The work presented in this paper is supported in part by National Natural Science Foundation of China (grant Nos. 60921003, 61175039, 61103241) and Fundamental Research Funds for Central Universities (xjj20100051).