

# 基于 NS-2 仿真的接纳控制性能分析<sup>\*</sup>

王春峰 卢 睿 陈秀忠

(中国科学院计算技术研究所信息网络室 北京100080)

## Performance Analysis of Admission Control Mechanism Based on NS-2 Simulation

WANG Chun-Feng LU Rui CHEN Xiu-Zhong

(Institute of Computing Technology China Academy of Sciences, Beijing 100080)

**Abstract** Performance analysis is an important aspect in the designing of admission control mechanism. In order to evaluate the admission control mechanism, we should get the critical data in the experiments, which is difficult in real network. Network simulation is an effective method in performance analysis, network design etc. In this paper, we design and develop the traffic generating and admission control modules in NS-2 simulation platform. Experiment is carried out using those modules to evaluate the admission delay and admission probability. Last, we discuss their future trends and present several issues remained to be studied by combining the experiences of our research.

**Keywords** QoS, Admission control, Simulation

### 1. 引言

近年来,随着 Internet 应用范围的日益扩展和其用户数量的不断扩大,人们开始希望能够在 Internet 上进行实时通信。实时应用需要端到端的服务质量保证。

当前的 IP 网络只提供简单的“尽力而为”的服务。从连接的角度,由于“尽力而为”服务中业务流之间的相互依赖性,链路上的某些特性(如包丢失、延迟和延迟抖动等)随时间不断变化。实时业务流对这些网络特性事先并不了解,很难保证连接的性能(如最小带宽和最大延迟等)。因此,如何确保语音通信的服务质量就成了一个亟待解决的问题。

接纳控制是一种重要的 QoS 管理机制。接纳控制是采用特定控制算法,综合分析当前的网络状态(包括已接纳的业务流信息)、新进业务流的资源需求,判定当前网络的可用资源是否满足新进业务流的 QoS 需求,并做出接纳决定,从而保证允许接入的实时业务的端到端服务质量。为了适应大规模网络的需要,接纳控制机制应具有较小的接纳延迟和较高的接纳概率。较小的接纳延迟保证网络在支持大量实时业务时的可扩展性;较高的接纳概率可以提高网络整体的资源,避免不必要的资源浪费;接纳概率越高,资源的使用就越有效。

接纳控制机制的性能评价是接纳控制设计的重要方面。由于实时应用的特殊性,评价接纳控制时往往需要临界状态数据,这在实际网络环境中很难实现。网络仿真是网络性能理论分析、评估网络设计方案以及网络故障诊断的有力工具。在设计阶段,依靠仿真的方法对设计方案进行评估,这对于减少投资风险,降低网络实现费用等都有着巨大的好处<sup>[1]</sup>。NS-2<sup>[2]</sup>作为当今业界备受推崇的网络仿真平台,其模拟和仿真评价的结论是较为可信的。

本文将采用 NS-2 网络仿真平台从接纳概率和接纳延迟两个方面对接纳控制机制进行评价。首先介绍在 QNet<sup>[3]</sup>项目中采用的接纳控制机制;然后,结合 NS-2 网络仿真平台,详细讨论了仿真方案的设计与实现;接着搭建仿真实验,对我们采

用的接入控制算法进行评价。最后讨论了有待进一步研究或解决的一些问题。

### 2. 接纳控制

QNet 的目标是为实时应用提供面向连接的、绝对延迟保证的服务。接纳控制要综合考虑网络中已接受的和新进实时业务流的流量描述信息,以及网络资源状况,应用网络计算(Network Calculus)方法,计算最坏情况下的端到端延迟,并据此做出接纳决定。

实时应用的端到端延迟包括应用处理延迟、传播延迟、传输延迟、排队延迟,其中仅排队延迟是可变的。因此,我们主要考虑网络中包的排队延迟。在这里,我们将采用分解法,将对网络排队延迟的计算问题转化为对每个路由器的排队延迟的计算问题。

**定理1** 假定会话  $i$  的连接路径经历的路由器集合为  $N_i$ , 经历路由器  $j$  的排队延迟记为  $d_j$ , 会话  $i$  端到端的延迟记为  $d^{re}$ , 并假定固定延迟(含应用处理延迟、传播延迟、传输延迟)为  $d^{fix}$ , 则有

$$d^{re} = \sum_{j \in N_i} d_j + d^{fix} \quad (1)$$

对于排队延迟计算问题,Chang、Cruz<sup>[3-5]</sup>的计算理论由于缺乏可操作性,目前无实际的应用实例。而 Zhao 提出用最大速率函数描述每个实时语音流,基于对每个输入业务流的描述,以及被分析的网络设备的调度策略,应用排队分析方法可以得到业务流在最坏情况下的性能特征,包括最坏延迟和输出流的描述。有兴趣的读者可以参考文[6,7]。该延迟计算方法具有可操作的特点,Zhao 等基于 ATM 网络开发的 NetEx<sup>[6]</sup>就是应用实例。QNet 也沿用了该计算方法计算每个会话流经历每个路由器的排队延迟  $d_j$ ; 系统中所有路由器都采用 FCFS 作为其调度策略。

接纳控制就是对每个新进会话流查找连接路径,结合系统已有的、且与该连接路径存在带宽共享的活动连接,依据式(1)计算新进流可能经历的端到端延迟  $d^{re}$ 。当  $d^{re}$  小于新进

<sup>\*</sup> 本课题得到国家自然科学基金( #90104006)和国家“863”高技术研究发展规划项目( #2001AA112135)的资助。

流的端到端延迟要求,并且不破坏系统已存在会话的延迟保证时,系统允许接纳新会话;否则就拒绝新会话。

### 3. 仿真实验

#### 3.1 NS-2实验平台

该实验的仿真引擎采用 NS-2<sup>[2,9]</sup>,NS-2是由美国加利福尼亚大学 Berkeley 分校等四家教育和研究机构共同开发的网络仿真平台,它具有以下几个特点:一是它提供了丰富的模块库,如传输层的 UDP、TCP、调度算法,路由算法以及应用层的 FTP、CBR 等<sup>[9]</sup>;二是提供强大的跟踪和数据采集功能,为数据分析提供全面而详细的数据;三是易于扩充,NS-2为使用者提供了简单、有效的方法,使他们可以根据自己的需要方便地扩充 NS-2 的功能,实现自己的仿真,这也是本文使用 NS-2 的原因。

在 NS-2 中,构成模拟网络基本拓扑结构的是节点(node)和链路(link)。NS-2 没有区分主机和路由器,任何一个节点都可以完成存储-转发的工作;在每一个节点之上,Agent 负责实现某一具体的网络协议,这些协议既可以是传输层的,也可以是应用层的。NS-2 预先实现了 UDP Agent、TCP Agent 以及一些常用网络应用协议的 Agent,对于没有实现的协议,既可以从相似的、已实现的 Agent 派生,也可根据需要重新编写。实际上,在 NS-2 中,创建了一个 Agent 就相当于构建了基于某一协议的应用。调度机制通过相应的排队策略模块实现,结合到连路上实现数据包的调度。

NS-2 采用 tcl 和 c++ 两种语言编写而成,其中 tcl 语言是解释性的,主要负责编写进行网络仿真时的配置信息;而具体的协议模拟部分则交由 c++ 语言来完成。利用 NS-2 进行仿真,主要分三步:

- 首先,根据需要确定仿真需要实现的功能,采用 c++ 语言编写具体的实现部分;
- 其次,确定仿真的环境配置,编写 tcl 脚本。在编写 tcl 脚本时,除了可以使用 tcl 语言本身规定的命令之外,还可以使用 NS-2 本身的一些特殊命令,以及在为 c++ 代码中为设置 Agent 而编写的命令;
- 最后,执行 tcl 脚本,进行实际的仿真。NS-2 会根据 tcl 脚本的设置记录仿真的中间结果,用于数据的分析和处理。

#### 3.2 接纳控制的仿真设计

接纳控制的仿真要模拟真实网络环境中接纳控制算法的性能,它包括两个部分:业务流的生成和接纳控制过程。我们将通过这两个方面仿真接纳控制机制的接纳概率和接入延迟。下面,我们将详细介绍这两部分工作。

研究发现,到达某一网络的业务流是服从泊松分布的。遗憾的是,在 NS-2 中并没有这样的泊松流量产生器;因此,我们需要自己实现泊松业务流的产生。我们采用数值方法近似泊松过程,并用 C++ 语言编写了 QnetPosionAgent 类用于产生泊松分布业务流,业务流由 QnetHostAgent 实现, QnetHostAgent 主要负责两个方面的工作:向接纳控制服务器发送接纳请求和对允许接入的业务流模拟真实数据通信。

QnetPosionAgent 类从 RTPAgent 中派生而来,该类具体实现参照附录表 2。Start() 和 Stop() 负责泊松过程的开始和停止;Gen-posion() 用于产生泊松分布;Start-agent() 和 Stop-agent() 开始和停止相应的 QnetHostAgent 业务流。Rate-change() 和 Command() 则提供一组操作以供用户设置 QnetPosionAgent 的相关参数。

如前所述, QnetHostAgent 要模拟真实的通信过程,从某种意义上说, QnetHostAgent 是一个流量产生器。NS-2 中的 RTP Agent 本身实现了流量产生功能,并且当前大多数的实时业务采用 RTP 协议, QnetHostAgent 从 RTP Agent 中派生,能较好地模拟实时业务,减少了实现时的工作量。而对于接纳请求工作,在每次进行通信前, QnetHostAgent 向控制中心发送本次通信的相关信息,包括源、目标地址、业务流描述等等,在得到控制中心的许可之后再再进行数据传输,否则向用户返回失败信息,这些功能由 Send-To-Controller() 和 Recv() 完成,在信息交互过程中记录了控制信息的发送时间及接收时间,以便计算接入延迟。该类具体实现参照附录表 3。

接纳控制服务器的功能我们用 QnetCtrlAgent 实现,该模块除了负责接纳控制之外,还根据记录计算接纳概率。QnetCtrlAgent 在得到来自于 QnetHost 的请求之后,根据表 5 中(附录)的网络拓扑信息计算出实际传输所经过的路由器节点,计算该连接在网络中传输的延迟,如果延迟小于延迟上限,则允许该连接,否则拒绝该连接。端到端延迟的计算是接纳控制的关键,当加入或删除一条连接时,由于节点间的相互依赖关系,各个节点的延迟可能发生变化,从而影响网络中的所有连接的端到端延迟,这需要通过不反复的计算直到各个节点延迟不再变化为止,在 QnetCtrlAgent 中计算流程如图 1 所示。类具体实现参照附录表 4, Compute-Parameters()、Compute-Server()、Max-Traffic-Delay()、Get-Infect-Rtinfo()、Get-Infect-Traffic() 等过程实现延迟计算的功能, Insert-Traffic() 将允许接入的连接加入连接表, Delete-Traffic() 将结束通信的连接从连接表中删除。

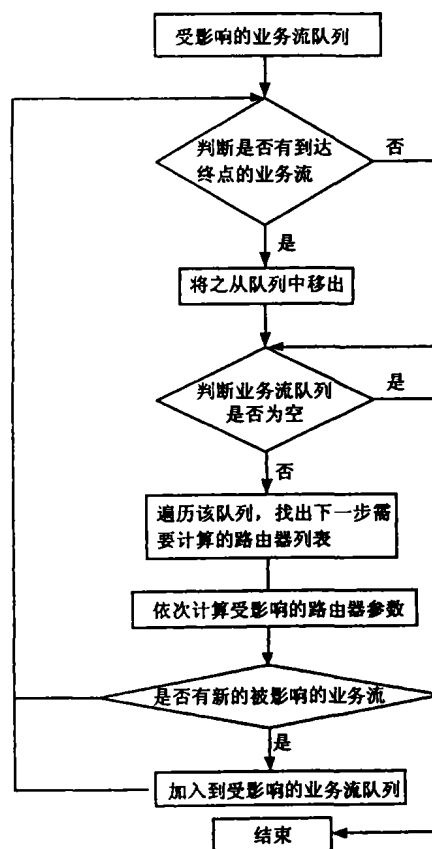


图1 网络延迟计算过程

#### 3.3 实验结果

图2是仿真实验的拓扑结构图。节点3、7、11为网络节点,

它们之间链路容量为1Mbps,链路延迟为10ms。节点0、1、2,节点4、5、6及节点8、9、10仿真三个子网中的主机节点,用于产生具体的流量;而节点13、14、15上分别有一个Poson 信息流发生器,用于相应子网上的泊松业务流产生;节点12上有一个QnetCtrlAgent,用作整个网络的接入控制服务器;这些主机节点的链路均为10Mb,链路延迟为5ms。

节点13、14、15将按照平均到达率为3次/秒,平均连接持续时间为180秒的泊松过程产生相应的实时业务流。整个仿真运行了近一个小时,所得结果如表1所示。

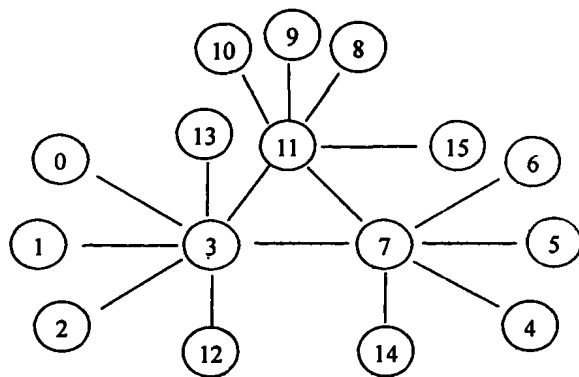


图2 仿真拓扑结构图

表1 仿真实验结果

总请求数	16023次
允许请求数	14260
接入率	89%
计算时延	稳定态:5-6ms;峰值:7ms。

**结束语** 接纳控制机制的性能分析是接纳控制设计的重要方面。网络仿真方法是网络性能分析的有效方法,本文基于NS-2网络仿真平台,具体设计并实现了相应的流量产生和接纳控制模块,并通过具体试验对接纳控制的接入延迟和接入率进行评价,为理论研究提供了有力的数据支持。但是,仿真实验中忽略了很多其他的影响因素,如路由器性能、链路状况、背景流量等等。下一步,我们将在节点及链路的设计中考虑这些因素,以进一步完善仿真实验,力求仿真能更为精确地描述实际网络中的行为。

## 参考文献

- 1 逯昭义,王思明. 计算机通信网信息量理论. 电子工业出版社,1997
- 2 Raha A, Kamat S, Zhao W. Guaranteeing End-to-End Deadlines in ATM Network. In: Proc. of the IEEE Intl. Conf. on Distributed Computing Systems, May 1995
- 3 Chang C S. Stability, Queue Length and Delay, Part I: Deterministic Queuing Networks. [Technical Report RC 17708]. IBM, 1992
- 4 Cruz R L. A Calculus for Network Delay, Part I: Network Elements in isolation. IEEE Trans. Inform. Theory, 1991, 37 (1):114~131
- 5 Cruz R L. A Calculus for Network Delay, Part II: Network Analysis. IEEE Trans. Inform. Theory, 1991, 37 (1):132~141
- 6 Sahoo A, Li C, Devalla B, Zhao W. Design and Implementation of NetEx: A Toolkit for Delay Guaranteed Communications. In: Proc. of Milcom, Volume:2, 1997. 733~739
- 7 Raha A, Kamat S, Zhao W. Guaranteeing End-to-End Deadlines in ATM Network. In: Proc. of the IEEE Intl. Conf. on Distributed Computing Systems, May 1995
- 8 陈秀忠,王春峰,黄晓璐. VoIP 服务质量管理机制及原型系统. 计算机科学,已录用
- 9 Breslau L, Estrin D, et al. Advances in Network Simulation. IEEE

Computer, 2000(5)

## 附录

表2 QnetPosionAgent 的主要成员

类别	名称	功能
成员函数	Start()	开始泊松业务流的产生。
	Stop()	中止泊松业务流的产生。
	Rate-Change()	改变泊松过程的平均到达率。
	Create-Agent()	创建 QnetHostAgent 业务流。
	Start-Agent()	开始 QnetHostAgent 业务流传送。
	Stop-Agent()	停止 QnetHostAgent 业务流的传送。
	Gen-Posion()	产生泊松分布。
	Timeout()	接收泊松分布或 QnetHostAgent 定时器的调用,在其超时产生相应的动作。
	Command()	提供一组操作以供用户设置泊松流量产生器的参数。

表3 QnetHostAgent 的主要成员

类别	名称	功能
成员函数	Start()	调用相应函数向接入控制服务器发送请求信息。
	Stop()	调用相应函数向接入控制服务器发送结束信息。
	Start-Transport()	开始数据传输。
	Stop-Transport()	结束数据传输。
	Send-To-Controller(int)	向接入控制服务器发送接纳请求。
	Recv()	接受接纳控制服务器的接纳响应。
	Command()	提供一组操作以供用户进行相应的设置或发出开始或结束的命令。

表4 QnetCtrlAgent 的主要成员

类别	名称	功能
成员变量	vector < RouteInfo > rtinfo_	网络中的路径信息。
	vector < Traffic > traffic_	网络中已有的业务流信息。
	int requestnum_	请求接入的业务流总数。
	int admitnum_	允许接入的业务流总数。
	string fname_	输出文件名。
成员函数	void start()	虚函数,仅为改写父类的实现,无内容。
	void stop()	虚函数,仅为改写父类的实现,无内容。
	Insert-Route()	为 QnetCtrlAgent 设置当前网络的路由信息。
	Insert-Traffic()	向当前网络中插入一条新的业务流。
	Delete-Traffic()	删除当前网络中的一条业务流。
	Compute-Parameters()	计算网络中受到影响的路由器和业务流的参数。
	Get-Infect-Rtinfo()	查询当前受影响的路由器信息。
	Compute-Server()	计算受影响的路由器参数。
	Get-Infect-Traffic()	查询当前受影响的业务流信息。
	Max-Traffic-Delay()	计算当前网络中受影响的业务流的最大延时。
	Command()	提供一组操作以供用户设置控制器的参数。
	Recv()	接收来自于主机发出的数据包。