

# 一种提高 SPICE 虚拟视频性能的反馈丢帧策略

陈丹伟<sup>1,2</sup>, 张 增<sup>1</sup>

(1. 南京邮电大学 计算机学院, 江苏 南京 210003;  
2. 南京邮电大学 信息安全联合实验室, 江苏 南京 210003)

**摘 要:**针对 SPICE(Simple Protocol for Independent Computing Environment)远程桌面视频播放质量问题,文中在原有架构基础上提出一种反馈比较丢帧的策略。该方案首先根据实际网络带宽和 SPICE 客户端缓冲区占有率之间取舍得到一个最优速率,然后将其反馈至 SPICE 服务器端,SPICE 服务器端根据缓冲区中的数据占有率与反馈值做比值,并根据比较结果采用缓冲区管理方案适当地丢弃视频帧。实验通过在有无权衡因子和缓冲区管理两个方面对视频的传输过程进行验证,结果表明该方案有效地减少了视频延迟,提高了视频流畅度。

**关键词:**SPICE;反馈比较丢帧;缓冲区管理

**中图分类号:**TP31

**文献标识码:**A

**文章编号:**1673-629X(2014)01-0136-04

doi:10.3969/j.issn.1673-629X.2014.01.035

## A Feedback Frame-dropping Strategy of Improving SPICE Virtual Video Performance

CHEN Dan-wei<sup>1,2</sup>, ZHANG Zeng<sup>1</sup>

(1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;  
2. Association Laboratory of Information Security, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:**According to the present problems of the SPICE (Simple Protocol for Independent Computing Environment) remote desktop video quality, a feedback comparing frame-dropping strategy based on the original architecture is proposed. This scheme obtains an optimal frame rate according to the trade-off between the actual network bandwidth and the SPICE client buffer occupancy, then sends it to the SPICE server. The SPICE server calculates the ratio of the occupancy rate of data in buffer to the feedback parameter, selectively drops the video frame based on the ratio and buffer management scheme. By experiment verifying at both weighing factor and buffer management in the video transmission, the results show that it can not only reduce the video delay but also improve the video fluency.

**Key words:**SPICE; feedback comparing frame-dropping; buffer management

## 0 引 言

随着桌面虚拟化技术的发展,人们对于虚拟桌面的需求也越来越高。虚拟化技术已经趋于成熟,也为人们带来越来越多的便利。目前的远程桌面协议有 RDP, ICA 和 SPICE 等<sup>[1]</sup>。RDP 和 ICA 拥有较长的研发历史, SPICE 相对较新。SPICE 是一种专门设计应用于虚拟环境的自适应远程传送协议。利用现代的高性能通信技术, SPICE 旨在为今天的带宽密集型应用(如多媒体、VoIP)提供无缝的用户体验,使用户在使

用虚拟桌面系统时感受到与使用物理桌面系统同样的体验<sup>[2]</sup>。

从 Red Hat 公司开源 SPICE 至今, SPICE 已经拥有越来越多的用户,凭借着良好的虚拟桌面体验也吸引了越来越多的客户。然而,美中不足的是 SPICE 虚拟桌面的视频播放仍然是一个亟待解决的瓶颈问题。与之而来的视频播放的实时性和视频播放质量对用户体验有着很大的影响<sup>[3]</sup>。

因此,文中在对 SPICE 作深入研究的基础上提出

收稿日期:2013-03-13

修回日期:2013-06-17

网络出版时间:2013-11-12

基金项目:国家自然科学基金资助项目(61073188)

作者简介:陈丹伟(1970-),男,教授,博士,硕士生导师,研究方向为计算机通信网络与安全、计算机取证技术;张 增(1986-),男,河南焦作人,硕士研究生,研究方向为信息安全。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20131112.1653.055.html>

一种基于 SPICE 架构上的反馈比较丢帧策略<sup>[4]</sup>。该策略首先得到 SPICE 客户端视频帧的情况,并基于网络带宽和客户端缓冲区情况做出权衡,反馈给服务器端,服务器端采用积极的缓冲区管理方案来对缓冲区进行管理,从而使得视频流在服务端到客户端的各个阶段都有适当的传输速率,进而提高视频实时性能<sup>[5]</sup>。

- (1) 根据客户端缓冲区利用率和带宽利用率来动态调整服务端缓冲区发送数据速率。
- (2) 提出积极的服务端缓冲区管理方案。
- (3) 实验表明文中提出的方案能够增强视频服务的质量。

1 相关工作

1.1 SPICE

SPICE (Simple Protocol for Independent Computing Environment) 是 Red Hat 公司收购 Qumranet 后获得的虚拟技术,并将其作为开源代码,用于在服务器和远程计算机如桌面和瘦客户端设备上部署虚拟桌面。

SPICE 架构分为三部分,如图 1 所示。

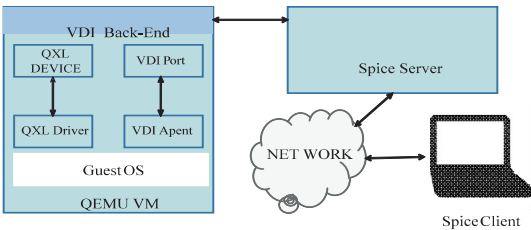


图 1 SPICE 架构

- SPICE 驱动器:存在于每个虚拟桌面内的组件;
  - SPICE 设备:存在于红帽企业虚拟化 Hypervisor 内的组件;
  - SPICE 客户端:存在于终端设备上的组件,可以是瘦客户机或专用的 PC,用于接入每个虚拟桌面。
- 前端和后端通过 VDI (Virtual Device Interfaces) 进行交互<sup>[3]</sup>。

1.2 网络架构

视频流从服务端到客户端传输编解码过程如图 2 所示。

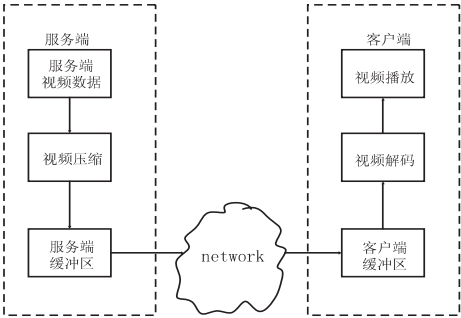


图 2 视频传输框架

视频数据在服务端进行 MJPEG 编码,生成一帧一

帧视频帧数据进入服务端缓冲区,然后分为若干的数据包经过网络栈各层数据处理后发送到网络中传输,客户端接收到数据包经过网络栈各层处理后先放入缓冲区,根据包头数据和时间戳连接为完整的视频帧数据,然后客户端从缓冲区中取得数据进行解码播放<sup>[6]</sup>。具体流程图见图 3。

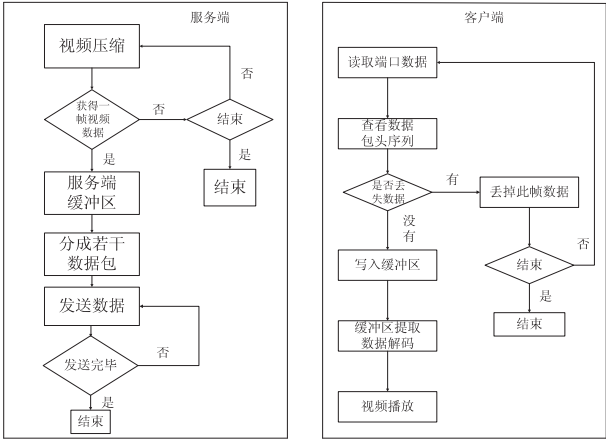


图 3 数据传输过程

视频数据在无线网络中是通过 IP 数据包来传输,由于一个视频帧大于一个数据包,通常在传输过程中一个视频帧被分为  $n$  个 IP 数据包来传输,文中采用基于视频帧的模型研究。

1.3 服务端缓冲区概述

服务器端缓冲技术是流媒体系统中常用的一种应用控制技术,能够平滑网络抖动,提高客户端视频播放质量<sup>[7]</sup>。通过服务端缓冲区的积极管理方案达到平滑网络抖动,提高客户端与服务端桌面视频实时性,这是基于视频帧的模型研究,由于一帧视频数据大于在无线网络中传输的数据包,一帧数据会分成  $n$  个 IP 数据包来传输<sup>[8]</sup>,该模型以帧为单位来研究。

该文的缓冲区管理增加了码率控制的功能,控制策略能进一步提高视频质量,特别适合于系统资源有限、带宽窄或强实时性的视频流式传输。

2 方案设计

文中在原有 SPICE 框架基础上<sup>[3]</sup>提出了一种反馈策略,通过服务端对比丢帧得到数据的最优发送速率。该方法权衡客户端缓冲区利用率和带宽因素动态调整服务端数据发送速率,以减少传输延迟,提高视频画面的流畅性。

2.1 一种反馈机制的提出

为了保证 SPICE 虚拟桌面视频服务质量 (Quality of Service, QoS),就要尽量保证视频流畅,低延迟,实时性能好<sup>[9]</sup>。对于 SPICE 来讲视频压缩算法已经确定,因此可以通过控制视频流从服务端传到客户端速率来保证整个流程的流畅,从而保证视频流畅。服务端视

视频帧发送速率控制着整个流程。该部分设计了一种反馈策略动态地求出服务端的发送速率,该方法基于提高客户端缓冲区的利用率和节省带宽的权衡,得到最优发送速率  $\text{Innet\_rate}(t)$  的方法。

如图 4,从开始算的话,  $t_1 = 0$ 。

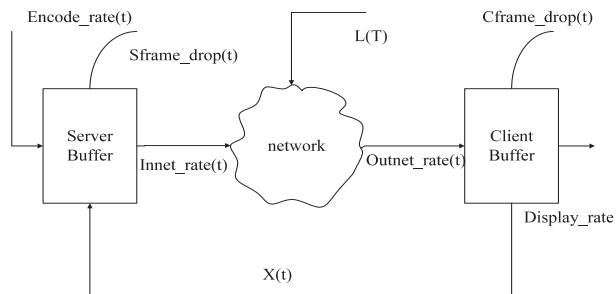


图 4 数据传输过程速率架构图

服务端:

$$\int_{t_1}^{t_2} \text{Encode\_rate}(t) dt = \int_{t_1}^{t_2} \text{Sframe\_drop}(t) dt + \int_{t_1}^{t_2} \text{Sbuffer\_filled}(t) dt + \int_{t_1}^{t_2} \text{Innet\_rate}(t) dt \quad (1)$$

式中,  $\text{Encode\_rate}(t)$  为视频流编码速率,即进入缓冲区的视频帧速率;  $\text{Sframe\_drop}(t)$  为服务端缓冲区视频帧丢帧速率;  $\text{Sbuffer\_filled}(t)$  为  $t$  时刻服务端缓冲区已经缓存视频流的总大小。

$$\int_{t_1}^{t_2} \text{Sbuffer\_filled}(t) dt + \int_{t_1}^{t_2} \text{Sbuffer\_left}(t) dt = \text{Sbuffer\_size} \quad (2)$$

式中,  $\text{Sbuffer\_left}(t)$  是服务端缓冲区剩余的情况,服务端缓冲区大小是一定的,  $\text{Sbuffer\_left}(t)$  取决于  $\text{Sbuffer\_filled}(t)$ ,  $\text{Sbuffer\_filled}(t)$  受到网络数据发送速率  $\text{Innet\_rate}(t)$  的影响。

网络前后:

$$\text{Outnet\_rate}(t + \tau) = \text{Innet\_rate}(t) + L(t) \quad (3)$$

式中,  $\text{Innet\_rate}(t)$  代表服务端发送数据速率;  $L(t)$  是在无线网络传输中数据包丢失造成的干扰;  $\tau$  是数据包在传输过程中的网络延迟;  $\text{Outnet\_rate}(t)$  代表客户端接收数据速率,客户端接收速率意味着数据进入缓冲区的速率,因此有式(4):

$$\int_{t_1}^{t_2} \text{Outnet\_rate}(t) dt = \int_{t_1}^{t_2} \text{Cbuffer\_filled}(t) dt + \int_{t_1}^{t_2} \text{Display\_rate}(t) dt + \int_{t_1}^{t_2} \text{Cframe\_drop}(t) dt \quad (4)$$

客户端:

根据式(4)可知,到达客户端缓冲区的视频帧分为三种:一部分是按照时间戳排好顺序的视频帧送到预取缓冲区中用于解码播放  $\text{Display\_rate}(t)$ ;一部分是存储在缓冲区中的数据  $\text{Cbuffer\_filled}(t)$ ,缓冲区满的时候再来的数据就丢弃了;  $\text{Cframe\_drop}(t)$  代表客户端缓冲区视频帧丢帧速率。用  $\text{Sbuffer\_size}$  来标识服

务端缓冲区大小,那么式(5)

$$\int_{t_1}^{t_2} x(t) dt = \text{Cbuffer\_size} - \int_{t_1}^{t_2} \text{Sbuffer\_filled}(t) dt \quad (5)$$

表示客户端缓冲区的空闲率。  $x(t)$  作为反馈信息从客户端反馈给服务端,因为客户端缓冲区是从内存中划分出来的,所以要尽量提高它的利用率,同时需要更多的传输带宽,为此引入了  $b$  和  $r$  作为缓冲区因子和信道带宽因子用来衡量缓冲区利用率和信道带宽利用情况,令  $a = \sqrt{b/r}$  ( $a$  为权重因子)得到:

$$\text{Innet\_rate}(t)^* = a \cdot x(t) + \text{Display\_rate}(t) \quad (6)$$

这样得到的就是最优输入速率,即服务端缓冲区最优发送视频帧的速率。

## 2.2 服务端缓冲区比较丢帧策略

由式(2)可知,  $\text{Sbuffer\_filled}(t)$  受到网络数据发送速率  $\text{Innet\_rate}(t)$  的影响,  $\text{Innet\_rate}(t)$  取决于客户端的反馈信息和  $a$  的值的大小,令  $\text{Ratio\_rate}(t) = \text{Innet\_rate}(t) / \text{Sbuffer\_filled}(t)$ ,即把  $\text{Ratio\_rate}(t)$  作为客户端缓冲区中最优发送速率和缓冲区视频帧占有量的比值,为了防止数据拥堵造成的延迟,提出了积极的缓冲区管理方案,根据缓冲区占有率而采取不同的丢帧政策,缓解了客户端数据压力,提高视频的实时性能。

```
While (视频流存在) {
  If (Ratio_rate(t) > 100%) {
    按照原始编码速率进行编码;
  } else {
    If (Ratio_rate(t) > 70%) { 隔 3 帧丢一帧 }
    else {
      If (Ratio_rate(t) > 40%) { 隔两帧丢一帧 }
      else { 隔一帧丢一帧 } }
    }
  If (缓冲区的空闲区 > 帧长) { 直接进入缓冲区 }
  else { while (丢弃该帧) }
}
```

该算法在服务端缓冲区得到最优发送速率的情况下,根据自己缓冲区视频帧的占有情况做出适当的丢帧策略,在不影响视频效果的前提下有效地保证了视频的流畅性,能够有效地防止缓冲区溢出。这种方案对于带宽窄实时性要求高的情况会有较好的应用,它采取合理的丢帧政策,有效缓解了网络传输和客户端数据拥堵造成的后果,提高了视频的实时性,从而提高了视频服务质量。

## 3 实验评估

### 3.1 实验环境搭建

如表 1,在 IEEE802.11b 无线局域网的环境中,在 4 M 带宽下测试,搭建的客户端服务端采用 SPICE 虚

拟机,在 SPICE 下对该方案进行测试。影片采用“泰坦尼克号”编码传输,视频帧大小为  $320 * 240$ ,平均压缩率为  $1/15$ ,帧传输速率为  $30\text{ fps}$ ,数据包出错率为  $0 \sim 20\%$ ,数据帧从服务器端向客户端发送数据帧。

表1 实验参数列表

影片名称	泰坦尼克号
视频编码方式	MJPEG
视频帧大小	$320 * 240$
平均压缩率	$1/15$
帧传输速率/fps	30
网络速率/Mbps	10
客户端最大缓冲区/Mbyte	1
服务端最大缓冲区/Mbyte	1
数据包出错率	$0 \sim 20\%$
权重因子 $\alpha$	0.2,0.5
平均播放速率/Mbps	3.5

3.2 实验结果评估

图5和图6为在有权重因子  $\alpha(\alpha=0.5)$  和没有的情况下客户端的视频播放速率的对比。

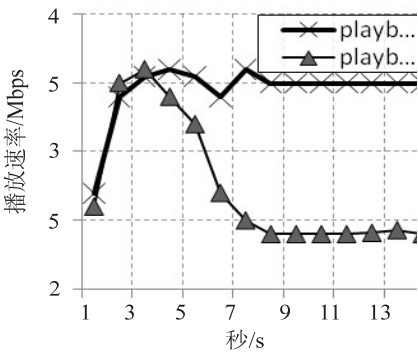


图5 有无因子情况对比

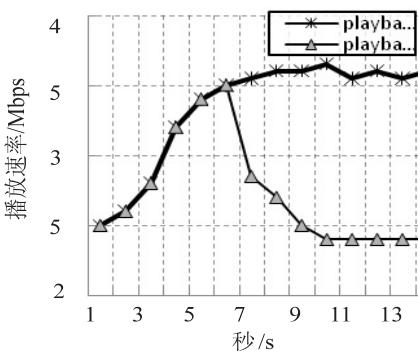


图6 有无服务端缓冲区丢帧对比

对于有  $\alpha$  的情况,根据客户端缓冲区反馈的情况,服务端缓冲区根据反馈的结果动态调整网络数据发送速率  $Innet\_rate(t)$ ,这样就保证了客户端不会因为拥堵而产生视频播放质量的下降。图5中,在  $\alpha = 0.2$  时,反馈信息  $x(t)$  较小,此时网络数据发送速率  $Innet\_rate(t)$  较小,这样  $Cbuffer\_filled$  增大,服务端缓冲区的占有率增大,客户端播放速率会因此而受到影响,而对于有服务端缓冲区管理简单算法的情况,会提高视频质量。

4 结束语

文中在原有架构基础上提出一种反馈比较丢帧的策略。该方案首先根据实际网络带宽和 SPICE 客户端缓冲区占有率之间取舍得到一个最优速率,然后将其反馈至 SPICE 服务器端,SPICE 服务器端根据缓冲区中的数据占有率与反馈值做比值,并通过比较结果采用缓冲区管理方案适当地丢弃视频帧。

对于权重因子  $\alpha$  是最大客户端缓冲区利用率和节省带宽的一个权衡,反馈给服务端,从而能够使服务端得到一个最优的发送数据速率<sup>[10]</sup>,并且在服务端速率较高的情况下提出服务端缓冲区管理方案,来缓解网络抖动拥塞,从而提高视频服务性能<sup>[11]</sup>。实验证明该方案能有效地缓解网络拥塞,减少数据丢失,提高视频服务实时性能。

参考文献:

[1] 张建勋,古志民,郑超. 云计算研究进展综述[J]. 计算机应用研究,2010,27(2):429-433.

[2] 梁飞蝶,李锦涛. 瘦客户计算应用协议中远程显示机制的比较[J]. 计算机工程与应用,2004(21):135-137.

[3] Glibos AAn open remote computing solution[EB/OL]. 2009-09-04. <http://SPICE-space.org/>.

[4] Deboosere L, Wachter J D, Simoens P, et al. Thin client computing solutions in low and high motion scenarios[C]//Proceedings of the third international conference on networking and services. Washington, DC, USA: IEEE Computer Society, 2007.

[5] 杨智楠. 虚拟桌面技术应用与研究[J]. 信息安全与技术, 2011(12):75-77.

[6] Liebl G, Jenkac H, Stockhammer T, et al. Radio link buffer management and scheduling for video streaming over wireless shared channels[C]//Proc of packet video workshop. Irvine, CA, USA; [s. n.], 2004.

[7] MacKenzie R, Hands D, Farrell T O. QoS of video delivered over 802.11e WLANs[C]//Proc of ICC 2009 QoS and modeling symposium. [s. l.]: [s. n.], 2009.

[8] Orlov Z, Necker M C. Enhancement of video streaming QoS with active buffer management in wireless environments[C]//Proc of EW 2007. [s. l.]: [s. n.], 2007.

[9] Takahata K, Shibata Y. QoS control of multimedia communication over wireless network[C]//Proc of on ICDSCS workshop (MNSA'2002). [s. l.]: [s. n.], 2002:336-340.

[10] 文远保,林建明. 嵌入式流媒体客户端缓冲控制策略的研究[J]. 华中科技大学学报(自然科学版),2005,33(10):83-85.

[11] 欧巧凤,邓谦. 无线流媒体系统的多模式自适应缓冲设计[J]. 南昌航空大学学报(自然科学版),2010,24(3):11-15.

一种提高SPICE虚拟视频性能的反馈丢帧策略

作者:

陈丹伟, 张增, [CHEN Dan-wei](#), [ZHANG Zeng](#)

作者单位:

[陈丹伟, CHEN Dan-wei \(南京邮电大学 计算机学院, 江苏 南京210003; 南京邮电大学 信息安全联合实验室, 江苏 南京210003\)](#), [张增, ZHANG Zeng \(南京邮电大学 计算机学院, 江苏 南京, 210003\)](#)

刊名:

[计算机技术与发展](#)

ISTIC

英文刊名:

[Computer Technology and Development](#)

年, 卷(期):

[2014\(1\)](#)

本文链接: [http://d.wanfangdata.com.cn/Periodical\\_wjfz201401035.aspx](http://d.wanfangdata.com.cn/Periodical_wjfz201401035.aspx)