一种采用多 Agent 的视频监控系统的设计与研究

Design and Research On A Model of Video Monitor and Control System Based on Multi – agent

赵纪胜　高胜法　张新民　文 坤　(济南 山东大学计算机科学与技术学院 250061)

摘要:在视频监控系统中,当监控点和用户增加时视频数据在网络上的流量会大大增加,网络冲突严重,无法保证用户的 QoS。该文针对此问题提出了一种采用多 Agent 的视频监控系统模型,讨论了采用多 Agent 的视频监控模型的原理和内部结构,并使用 C + + 中的 Socket 套接字对多 Agent 的通信机制进行了实现。

关键词:多 Agent　视频控制　Socket

1　引言

随着分布式视频监控系统的应用领域日益广泛,功能也日趋复杂和多样化,需要处理的信息量急剧膨胀,传统的视频监控系统面对如此巨量的信息,不断暴露出固有的局限性,在实际应用中会出现以下两个问题:(1)网络的带宽是有限的,并且监控点和用户往往是多个,一个用户可以向多个监控点申请视频监控图像,一个监控点也可以向多个用户发送视频图像,这样在网络中有多路视频数据流在传送,这势必会占用大量的网络带宽;(2)用户是动态的,可以动态地加入,当用户加入并向监控点申请视频监控图像时,也会占用一定的网络带宽,这样,当监控点上申请的用户个数达到一定数量时,视频数据的网络传输量会急剧增加,局域网中可用的带宽就会大大减少,从而导致视频图像质量严重下降,出现帧丢失、图像抖动,视、音频不同步等一系列现象以至无法保证用户 QoS。为了解决此问题,我们研究设计了一种采用多 Agent 视频监控系统模型,该模型可以较好的保证用户的 QoS,充分利用系统的资源,合理利用网络带宽,减少网络冲突。

2　基于多 Agent 的视频监控系统的模型及其描述

2.1　模型设计

设计的网络模型是基于 C/S 结构,在客户端和服务器端分别包含两种类型的 Agent,分别是 Control Agent 和 Data Agent,如图 1 所示。

Control Agent 是用来处理用户的要求(如用户 QoS 请求、用户所发出对监控设备的控制命令等)以及管理各个用户的信息(如用户的登录、离开和身份验证等)。Data Agent 用来处理视频监控数据流的。

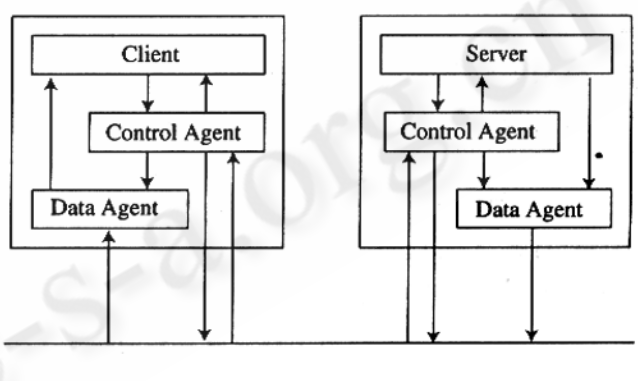

图 1

2.2　系统工作过程描述

该模型的工作过程如图 2 所示。

整个工作过程可分为三个阶段:Client 呼叫视频监控服务,网络传输,关闭传输。具体过程描述如下:

(1) Client 向 Control Agent 发出请求监控服务消息。

(2) Control Agent 收到服务请求,分解成多个子服务,并行向提供子服务的 Server 发请求。

(3) Server 收到子服务请求,将其监控服务要求

的 QoS 发给 Control Agent。

(4) Control Agent 收集所有 Server 返回的 QoS，并存储，最后合并成一个 QoS 发给 Client。

(5) Client 收到请求监控服务的 QoS 以后，检查是否满足要求，并且把结果用确认消息发给 Control Agent。

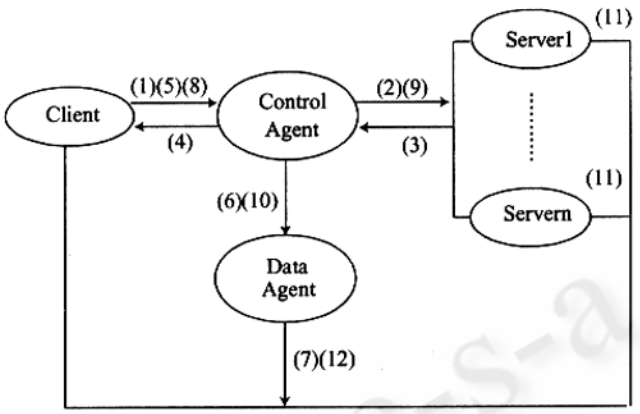

图 2

(6) Control Agent 收到确认消息后，若 QoS 不能满足服务，则删去 Server 发来的 QoS；若 QoS 满足服务，则向 Data Agent 发建立网络路由的消息。

(7) Data Agent 收到建立网络路由的消息，动态地建立 Data Agent 到 Client 和 Server 的网络路由。Server 上的监控服务按照 QoS 要求将数据发给 Data Agent，Data Agent 同步地合并多个多媒体流以后，按照 QoS 将数据发给 Client。Data Agent 根据网络负载和性能情况，动态调整媒体流传输速率和网络带宽以保证媒体流的 QoS。

(8) 传输结束 Client 向 Control Agent 发出关闭服务的请求。

(9) Control Agent 收到关闭请求，向提供子服务的 Server 发出关闭请求。

(10) Control Agent 收到关闭请求，向 Data Agent 发出关闭请求。

(11) Server 收到关闭请求，关闭服务，释放资源。

(12) Data Agent 收到关闭请求，释放监控服务网络路由。

3 实现多 Agent 通信的关键技术

在 Visual C++ 6.0 中用 Socket 套接字实现多 Agent 通信的关键技术如下：服务器端设置一个 Socket 变量 m_hControlSocket 和 Socket 数组 m_aClientSocket[MAXClient]，其中 MAXClient 代表能接受客户器端连接的最大数目，m_hControlSocket 用来在指定的端口进行侦听，如果有客户器端请求连接，则在 m_aClientSocket 数组中查找一个空 socket，将客户器端的地址赋予此 Socket。核心代码如下：

```
char m_chMsgBuffer[100];
SOCKADDR_IN m_sockClientAddr; //Socket 地址结构
BOOL m_bControlIsOpen; //控制端 socket 是否打开
SOCKET m_hControlSocket;
SOCKET m_aClientSocket[MAXClient];
void CcontrolView::OnControlOpen()
{
WSADATA wsaData;
int iErrorCode;
char chLocalInfo[64];
m_bControlIsOpen = FALSE;
//初始化 Socket 数组
for(int i = 0;(i < MAXClient);i++) m_aClientSocket[i] = INVALID - SOCKET;
M_pDoc -> m_hControlSocket = socket(PF_INET, SOCKSTREAM,
DEFAUL T_PROTOCOL);
M_pDoc -> m_sockControlAddr. sin_family = AF_INET;
M_pDoc -> m_sockControlAddr. sin_addr. s_addr = INADDR_ANY;
M_pDoc -> m_sockControlAddr. sin_port = htons(m_pDoc ->
M_nControlPort);
//绑定
bind(m_pDoc -> m_hControlSocket,(LPSOCKADDR) &m_pDoc
-> m_sockControlAddr, sizeof(m_pDoc -> m_sockControlAddr));
WSAAsyncSelect(m_pDoc -> m_hControlSocket, m_hWnd,
```

```
WMCONTROL_ACCEPT,FD_ACCEPT);
//开始监听
listen(m_pDoc->m_hControlSocket,QUEUE_SIZE);
m_bControlIsOpen=TRUE;
}
```

客户端只需设置一个 socket 变量 m_hSocket，与客户端进行连接。连接建立好后，通过此 Socket 进行通信。

其核心代码如下：

```
void CClientDlg::OnSocketConnect()
{
WSADATA wsaData;
DWORD dwIPAddr;
SOCKADDR_IN sockAddr;
WSAStartup(WINSOCK_VERSION,&wsaData);
M_hSocket=socket(PF_INET,SOCK_STREAM,0);
sockAddr.sin_family=AF_INET;
sockAddr.sin_port=m_iPort;
sockAddr.sin_addr.S_un.S_addr=dwIPAddr;
int Connect=connect(m_hSocket,(LPSOCKADDR)&sockAddr,
sizeof(sockAddr));
}
```

当有客户端 Socket 对服务器端 Socket 发出请求时，服务器端 Socket 就调用 OnControlAccept 方法，该方法的功能是建立服务器端和客户端的连接，并在服务器端创建一个客户端 Socket 负责与请求的 Socket 进行通信。OnControlAccept 方法的核心代码如下：

```
LRESULT CControlView::OnControlAccept(WPARAM wParam,LPARAM lParam)
{
int iErrorCode;
int nLength=sizeof(SOCKADDR);
int i;
if(WSAGETSELECTEVENT(lParam)==FD_ACCEPT)
{
for(i=0;(i<MAXClient)&&(m_aClientSocket[i]!=INVALID_SOCKET);i++);
//如果没有空 Socket 可供分配，则返回，否则，连接成功
if(i==MAXClient) return 0L;
m_aClientSocket[i]=accept(m_pDoc->m_hControlSocket,
(LPSOCKADDR)&m_sockClientAddr,(LPINT)&nLength);
WSAAsyncSelect(m_aClientSocket[i],m_hWnd,WMCLIENT_READCLOSE,
FD_READ|FD_CLOSE);
}
}
```

4 结束语

本文探讨了一种采用多 Agent 的视频监控系统模型，与传统模式的视频监控技术相比，该系统具有以下优点：(1) 系统采用多线程的组织结构，不同的代理使用不同线程，提高了系统资源的利用率；(2) 由于采用了 Control Agent 机制，可以引入服务质量保证机制，能更好地满足用户的 QoS 需求；(3) Control Agent 和 Data Agent 可以使用不同的缓存，并通过不同的代理管理，有效地减少了网络冲突。文中提出的采用多 Agent 机制为视频监控系统提供了更好的灵活性、可扩充性和适应性，具有良好的控制效果和稳定性，在远程视频监控中具有广泛的应用前景。

参考文献

1 杨学良、张占军，分布式多媒体计算机系统教程[M]，电子工业出版社，2002。

2 周洞汝，视频数据库管理系统导论[M]，科学出版社，2000。

3 徐华，基于 IP 的远程视频监控系统[J]，山东科学，2003，16(3)。

4 徐建军、李强，一种基于多 Agent 的视频监控系统模型，计算机工程[J]，2003，29(9)。

5 张友生，基于移动 Agent 的远程控制技术，计算机应用[J]，2004，24(1)。

6 任晓明、杨大鉴，移动代理系统的体系结构分析[J]，计算机工程与应用，2001，27(1)。