



# [12] 发明专利申请公开说明书

[21]申请号 95116981.5

[51]Int.Cl<sup>6</sup>

H04L 12/56

[43]公开日 1996年5月22日

[22]申请日 95.8.28

[30]优先权

[32]94.8.29 [33]US[31]297,209

[71]申请人 美国电报电话公司

地址 美国纽约

[72]发明人 斯里尼瓦森·凯斯哈夫

罗森·沙玛

[74]专利代理机构 中国国际贸易促进委员会专利商  
标事务所

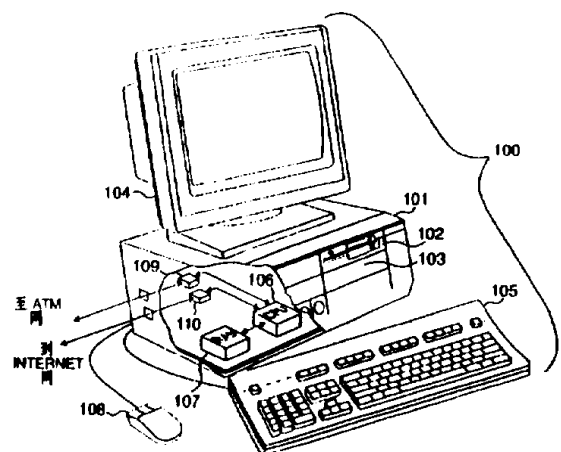
代理人 郭晓梅

权利要求书 10 页 说明书 22 页 附图页数 6 页

[54]发明名称 过程间通信和网络间数据传送的方法和系统

[57]摘要

本发明的方法和系统通过在处理系统上运行的面向无连接的服务器和用户程序以及在不同的处理系统上运行的面向连接的例行程序之间建立连接或通信虚拟线路使前一种程序能够与后一种程序进行通信。在启动与面向连接的例行程序的通信之前，可以修改面向无连接的服务器和用户程序来与连接管理器进行通信，以便建立虚拟线路。本发明还利用封装器和解封装器来使通信程序能够在按照第二格式传送数据的网络内按照第一格式传送数据包。



# 权 利 要 求 书

---

1. 在第一网络上的源设备中运行的源例行程序和为目标设备中运行的目标例行程序之间传送数据的方法, 该源和目标例行程序产生和接收异步传送模式("ATM")格式帧, 该第一网络传送"Internet 网协议"("IP")数据包, 该方法包括以下步骤:

- a) 产生包含源和目标信息的中间数据格式首部;
- b) 将产生的中间数据格式首部附加到一 ATM 格式帧来形成中间格式数据包, ATM 格式帧由源例行程序产生;
- c) 在 IP 数据包的数据部分中封装该中间格式数据包;
- d) 在第一网络上传送该 IP 数据包;
- e) 解封装该 IP 数据包来检索中间格式数据包;
- f) 从被解封装中间格式数据包检索 ATM 格式帧和中间数据格式首部; 以及
- g) 根据在被检索中间数据格式首部中的信息确定被检索 ATM 格式帧的路由。

2. 权利要求 1 的方法, 其中在步骤 a) 产生的中间数据格式首部包含表示一 ATM 格式帧相对于按照相应的顺序传送的其它 ATM 格式帧的位置的数据包顺序位置号。

3. 权利要求 1 的方法, 其中的目标设备是将第二网络与第一网络连接的中间连接器(gateway)处理系统, 第二网络传送 ATM 格式帧, 其中的步骤 e)、f) 和 g) 由该中间连接器处理系统来执行。

4. 权利要求 1 的方法,其中在步骤 a)产生的中间数据格式首部中的信息包含源设备网络地址,源和目标设备之间已建立的虚拟线路的虚拟线路标识符值,以及表示该 ATM 格式帧相对于按顺序传送的其它数据包顺序的位置的数据包顺序位置号。

5. 权利要求 1 的方法,还包括在封装步骤 c)之前产生包含目标 Internet 网设备的网络地址的 Internet 网协议数据包首部和表示数据包是被封装 ATM 帧的 Internet 网协议类型字段的步骤;其中的步骤 c)还包括将产生的 Internet 网协议数据包首部附加到中间格式数据包。

6. 权利要求 1 的方法,其中的 ATM 格式帧是 AAL 帧。

7. 权利要求 6 的方法,其中的 AAL 帧是 AAL5 帧。

8. 在 IP 数据包中封装 ATM 格式帧以便通过能够传送 IP 数据包的第一网络进行传送的方法,该方法包括以下步骤:

a)产生包含源和目标信息的中间数据格式首部;

b)将产生的中间数据格式首部附加到 ATM 格式帧来形成中间格式数据包;

c)产生 IP 数据包首部;以及

d)将 IP 数据包首部附加到中间格式数据包来形成 IP 数据包。

9. 权利要求 8 的方法,其中在步骤 a)产生的中间数据格式首部中的信息包含源设备网络地址,源和目标设备之间已建立的虚拟线路的虚拟线路标识符值,以及表示该 ATM 格式帧相对于按照顺序传送的其它数据包的位置的数据包顺序位置号。

10. 权利要求 8 的方法,还包括在封装步骤 c)之前产生包含

目标 *Internet* 网设备的网络地址的 *Internet* 网协议数据包头部和表示该数据包是被封装 *ATM* 帧的 *Internet* 网协议类型字段的步骤; 其中的步骤 c) 还包括将产生的 *Internet* 网协议数据包头部附加到中间格式数据包。

11. 权利要求 8 的方法, 其中的 *ATM* 格式帧是 *AAL* 帧。

12. 权利要求 11 的方法, 其中的 *AAL* 帧是 *AAL5* 帧。

13. 通过能够传送 *IP* 数据包的第一网络传送 *ATM* 格式帧的处理系统, 该处理系统包括:

与第一网络连接的源设备, 该源设备具有至少一个能够存储多条处理系统指令的存储器存储设备、以及至少一个控制数据传送和执行所述存储器存储设备的所述处理系统指令中的至少一条的处理单元, 所述处理单元能够产生包含源和目标信息的中间数据格式首部、将产生的中间数据格式首部附加到 *ATM* 格式帧来形成中间格式数据包、以及封装该中间格式数据包来形成 *IP* 数据的数据部分以便在第一网络上进行传送; 以及

与第一网络连接的 *第一网络* 目标设备, 该目标设备具有至少一个能够存储多条处理系统指令的存储器存储设备、以及至少一个控制数据接收和执行所述存储器存储设备的所述处理系统指令中的至少一条的处理单元, 所述处理单元能够解封装 *IP* 数据包来检索中间格式数据包、从被解封装中间格式数据包获得 *ATM* 格式帧和中间数据格式首部、以及根据在被检索中间数据格式首部中的信息确定被检索 *ATM* 格式帧的路由。

14. 权利要求 13 的处理系统, 还包括:

与 *第一网络* 目标设备连接的 *第二网络*, 该 *第二网络* 能够传送

*ATM* 格式帧；以及

第二网络目标设备，其中的第一网络目标设备是能够根据在被检索中间数据格式首部中的信息把在 *IP* 数据包中被封装的 *ATM* 格式帧送到第二网络目标设备的中间连接器设备。

15. 权利要求 13 的处理系统，其中的中间数据格式首部中的信息包含源设备网络地址，源和目标设备之间已建立的虚拟线路的虚拟线路标识符值，以及表示该 *ATM* 格式帧相对于按照顺序传送的其它 *ATM* 格式帧的位置的数据包顺序位置号。

16. 权利要求 13 的处理系统，其中的源设备还能够：

产生包含第一网络目标设备的 *Internet* 网地址的 *Internet* 网协议数据包首部和表示该数据包是被封装 *ATM* 格式帧的 *Internet* 网协议类型字段；以及

将产生的 *Internet* 网协议数据包首部附加到中间格式数据包来产生 *IP* 数据包。

17. 权利要求 13 的处理系统，其中的 *ATM* 格式帧是 *AAL* 帧。

18. 权利要求 17 的处理系统，其中的 *AAL* 帧是 *AAL5* 帧。

19. 通过能够传送 *IP* 数据包的第一网络传送 *ATM* 格式帧的处理系统，该处理系统包括：

与第一网络连接的源设备，该源设备具有与封装器通信的 *IP* 堆栈接口和 *ATM* 协议堆栈接口，该 *IP* 堆栈接口和与第一网络连接的 *IP* 网络接口连接；以及

与第一网络连接的第一网络目标设备，该第一网络目标设备具有与解封装器通信的 *IP* 堆栈接口和 *ATM* 协议堆栈接口，该

IP 堆栈接口和与第一网络连接的 IP 网络接口连接。

20. 权利要求 19 的处理系统,其中的第一网络目标设备还包括:

与 ATM 协议堆栈接口连接的 ATM 网络接口;

与该 ATM 网络接口连接的第二网络,该第二网络能够传送 ATM 格式帧;以及

与该第二网络连接的第二网络目标设备,如果源设备传送的 ATM 格式帧的目的地是第二网络目标设备,则第一网络目标设备的 ATM 协议堆栈接口能够通过第二网络把被解封装 ATM 格式帧送到第二网络目标设备。

21. 权利要求 19 的处理系统,其中的封装器是源设备的 IP 堆栈接口的一部分。

22. 权利要求 19 的处理系统,其中的解封装器是第一网络目标设备的 ATM 协议堆栈接口的一部分。

23. 服务器应用程序在该服务器应用程序和用户例行程序之间建立被连接管理器维护的通信虚拟线路的方法,该服务器应用程序和该用户例行程序在与无连接网络连接的各自处理系统上运行,该方法包括以下步骤:

将可用服务信息传送给连接管理器;

等待来自连接管理器的可用服务登记的确认;

等待来自连接管理器的连接请求,连接管理器一旦从用户例行程序接收到与服务器应用程序进行连接的就传送该连接请求;

将接受连接信息传送给连接管理器;

等待被连接管理器传送的虚拟线路标识符值的接收；以及命令在服务器应用程序处理系统中的协议堆栈接口例行程序传送从具有该虚拟线路标识符值的网络接收的全部数据包。

24. 权利要求 23 的方法,还包括如果在接收了连接请求后,如该连接不是可接受的,就传送拒绝连接信息的步骤。

25. 权利要求 23 的方法,其中的连接管理器利用连接请求将连接关键字值传送服务器应用程序,当传送接受连接信息来保证将与合适的服务器应用程序建立连接时,该值被传回给连接管理器。

26. 权利要求 23 的方法,其中的向连接管理器发送信息和从连接管理器接收信息由与服务器应用程序通信的连接服务例行程序库来完成。

27. 权利要求 23 的方法,其中的用户例行程序驻留在利用到达面向连接的网络的链路与无连接网络连接的系统中,该面向连接的网络和与该无连接网络连接的中间连接器处理系统连接。

28. 权利要求 27 的方法,其中的连接管理器是在中间连接器处理系统上运行的例行程序。

29. 权利要求 27 的方法,其中的面向连接的网络是 ATM 网络。

30. 用户应用程序在该用户应用程序和服务器例行程序之间建立被连接管理器维护的通信虚拟线路的方法,该用户应用程序和该服务器例行程序在与无连接网络连接的各自处理系统上运行,该方法包括以下步骤:

向连接管理器发送请求与特定服务器例行程序连接的信息；  
等待来自连接管理器的该请求的确认；

等待来自连接管理器的连接建立通知，当该特定服务器例行程序接受连接管理器请求的连接时，连接管理器就传送该连接建立通知，连接管理器还传送相应于具有该连接建立通知的连接的虚拟线路标识符值；以及

命令在用户应用程序处理系统中的协议堆栈接口例行程序发送从具有该虚拟线路标识符值的网络接收的全部数据包。

31. 权利要求 30 的方法，其中的连接管理器利用连接请求将连接关键字值传送给用户应用程序，该连接关键字值相应于虚拟线路标识符值并被传送给具有虚拟线路标识符值的协议堆栈接口。

32. 权利要求 30 的方法，其中的给连接管理器传送信息和从连接管理器接收信息由与用户机应用程序通信的连接服务例行程序库来完成。

33. 权利要求 30 的方法，其中的服务器例行程序驻留在利用到达面向连接的网络的链路与无连接网络连接的系统中，该面向连接的网络和与该无连接网络连接的中间连接器处理系统连接。

34. 权利要求 33 的方法，其中的连接管理程序是在中间连接器处理系统上运行的例行程序。

35. 权利要求 33 的方法，其中的面向连接的网络是 ATM 网络。

36. 与无连接网络连接的系统中，该处理系统在在该处理系



统上运行的服务器应用程序和用户例行程序之间建立被连接管理器维护的通信虚拟线路，该用户例行程序在与该无连接网络连接各个处理系统上运行，该处理系统包括：

至少一个能够存储多条处理系统指令的存储器存储设备；以及

至少一个控制在服务器应用程序和无连接网络之间的数据传送和接收并执行所述存储器存储设备的所述处理系统指令中的至少一条的处理单元，所述处理单元能够：

向驻留在与无连接网络连接的设备上的连接管理器发送可用服务信息；

等待来自连接管理器的可用服务登记的确认；

等待来自连接管理器的连接请求，连接管理器一旦从用户例行程序接收到与服务器应用程序进行连接请求就传送该连接请求；

将接受的连接信息发送给连接管理器，等待被连接管理器发送的虚拟线路标识符值的接收；以及

命令在该处理系统中的协议堆栈接口例行程序传送从具有该虚拟线路标识符值的网络接收的全部数据包。

37. 权利要求 36 的处理系统，在该处理系统中，如果在接收了连接请求后该连接是不可接受的，则处理单元还能够传送拒绝连接信息。

38. 权利要求 36 的处理系统，其中的连接管理器利用连接请求将连接关键字值但送给该处理系统，当传送接受连接信息来保证将与合适的服务器应用程序建立连接时，该处理系统能够将连

接关键字值传回给连接管理器。

39. 权利要求 36 的处理系统, 其中的用户机例行程序驻留在利用到达面向连接的网络的链路与无连接网络连接的设备中, 该面向连接的网络和与该无连接网络连接的中间连接器处理系统连接。

40. 权利要求 39 的处理系统, 其中的连接管理器是在中间连接器设备上运行的例行程序。

41. 权利要求 39 的处理系统, 其中的面向连接的网络是 ATM 网络。

42. 与无连接网络连接的處理系統, 該處理系統在在該系統上運行的用戶應用程序和服務器例行程序之間建立被連接管理器維護的通信虛拟線路, 該服務器例行程序在與該無連接網絡連接的各個處理系統上運行, 該處理系統包括:

至少一個能夠存儲多條處理系統指令的存儲器存儲設備;

至少一個控制在用戶機應用程序和無連接網絡之間的数据傳送和接收并執行所述存儲器存儲設備的所述處理系統指令中的至少一條的處理單元, 所述處理單元能夠:

向連接管理器發送請求與某一服務器例行程序連接的信息;

等待來自連接管理器的對於服務器例行程序的請求的確認;

等待來自連接管理器的連接建立通知, 當該服務器例行程序接受連接管理器請求的連接時, 連接管理器就發送連接建立通知, 連接管理器還發送相對於具有該連接建立通知的連接的虛拟線路標識符值; 以及

命令在該處理系統中的協議堆棧接口例行程序發送從具有該

虚拟线路标识符值的网络接收的全部数据包。

43. 权利要求 42 的处理系统, 其中的连接管理器利用连接请求将连接关键字值发送给该处理系统, 该连接关键字值相应于虚拟线路标识符值并被传送给具有该虚拟线路标识符值的协议堆栈接口。

44. 权利要求 42 的处理系统, 其中的服务器例行程序驻留在利用到达面向连接的网络的链路与无连接网络连接的处理设备中, 该面向连接的网络和与该无连接网络连接的中间连接器处理设备连接。

45. 权利要求 44 的处理系统, 其中的连接管理器是在中间连接器处理设备上运行的例行程序。

46. 权利要求 44 的处理系统, 其中的面向连接的网络是 ATM 网络。

# 说 明 书

---

## 过程间通信和

## 网络间数据传送的方法和系统

一般来说,本发明涉及处理系统网络,更具体地说,本发明涉及在网络间处理系统的应用程序之间建立通信和网络间数据传送的方法和系统。

由于异步传送模式(“ATM”)网络的高数据速率和灵活性,利用这种网络来传送包括视频、声音和数据的多媒体信息已变得越来越普遍。ATM网络使用CCITT规定的分组交换技术标准,M. dePrycker在“异步传送模式:来自宽带SIDN的解决办法”(Ellis Horwood, 1993)中描述了这一标准。ATM网络已被计算机和无线电通信工业用来连网已有和未来的多媒体应用,例如电视会议、付费电视(video-on-demand)和电话应用。ATM网络是面向连接的网络,在这种网络中,网络设备之间每次数据传送之前是调用网络连接管理器来建立设备之间的虚拟线路或连接。虚拟线路相当于被网络连接管理器识别的、用于在设备之间传送数据的特定发送通路。但是,几乎没有市场上买得到的计算机是ATM兼容的。此外,由于目前在ATM网络的构成方面所用的数据路由交换机的高成本,当前可用的与少数已有ATM网络的连接是有限的。

相反地,通过无连接网络、例如Internet网相互进行通信的计算机或其它处理系统正被广泛使用,并用于办公室、学校和许多家

庭。这些网络被称为无连接网，是由于数据通过网络从源设备传送到目标设备，不必象在 ATM 网络中所需要的那样首先建立“连接”或虚拟线路。

计算机利用若干协议中的任一协议、例如“Internet 网协议” (“IP”) 通过无连接网络进行通信。协议规定了文件传送、远端登记、电子邮件和其它服务，包括分布处理。IP 的功能之一是使数据分组能够通过一个或多个网络从源设备到达最终的目标设备。在无连接网络中，源设备利用目标设备的地址传送数据，无连接网络将利用任意多条网络路径把该数据送到所需的目标设备。

这些数据传送技术的差别一直是链接大量的计算机和在无连接网络、例如“交互 (internet) 网”上可用的应用软件与预期在 ATM 网络上获得的设备和宽范围的服务的主要障碍。在各种网络上传送信息所用的数据分组格式的不同进一步阻碍了网络系统的这种连接。

因此，已认识到需要在无连接网和 ATM 网络之间建立通信和数据传送。

根据本发明的原理，克服了现有技术的过程间通信和网络间数据传送的许多问题。

已有的无连接服务器和客户程序可通过在这些程序间建立连接或虚拟电路修改成和在与无连接或面向连接的网络连接的不同设备上运行的面向连接的例程进行通信。服务器程序为其它程序和规程提供服务，而客户程序利用服务器程序提供的服务。

这种无连接服务器和客户程序在传送数据之前与连接管理器进行通信。连接管理器是在程序之间建立和维持连接的例行程序或

线路。通过传送可用服务信息给连接管理器就可以修改服务器程序来建立与远端客户机例行程序的通信。连接管理器然后根据可用服务消息，登记可用服务并将登记消息的确认传回给服务器程序。用户程序然后可以向连接管理器传送连接请求消息来请求由服务器程序执行的到一个特定可用服务的连接。一旦接收到这种消息，连接管理器就传送相应的连接请求给服务器程序。服务器程序然后可以发送接受连接消息给连接管理器，后者作出响应、发回相应于服务器程序和远端客户例行程序之间的已建立连接的虚拟线路标识符(“VCI”)的值。该VCI值然后被正在运行服务器程序的设备的网络接口用来传送从具有该VCI值的网络接收的数据分组。

在相应的方式中，通过向连接管理器传送请求与一特定服务器例行程序连接的信息就可以修改用户程序来建立与远端服务器例行程序的通信。连接管理器然后通过传送返回确认消息对该请求作出确认。连接管理器确定被请求的服务是否被可用的服务器例行程序执行，如果是，就建立与该服务器例行程序的连接。一旦建立了与该服务器例行程序的连接，连接管理器就传送连接建立通知给用户程序。该连接建立通知包含相应于被建立连接的VCI值。该VCI值然后被正在运行用户程序的设备的网络接口用来传送从具有该VCI值的网络接收的数据分组。

本发明的一个方面涉及在连接已被建立之后数据分组的传输。本发明允许信息在源设备中运行的源例行程序和为目标设备中运行的目标例行程序之间的传送，源和目标例行程序按照ATM格式帧的方式产生和接收数据，而网络以Internet协议(“IP”)数据

分组的方式传送数据。利用封装器和解封装器来实现这种数据传送在 IP 分组的数据部分中封装 ATM 格式帧以便在网络上传送。

根据本发明的一种封装方法,产生中间数据格式首部,该首部包含要被传送的 ATM 格式帧的源和目标信息。被产生的中间格式首部然后被附加到 ATM 格式帧以便形成中间格式数据分组。该中间格式数据分组然后被封装到 IP 数据分组的数据部分中并通过网络被传送。目标设备接收的封装 IP 数据分组然后可被解封装为原来的 ATM 格式数据分组。

本发明的一种解封装方法首先从接收的 IP 数据分组中检索中间格式数据分组。然后从中间格式数据分组检索 ATM 格式帧和中间数据格式首部。目标设备然后可把检索的 ATM 格式帧送到相应的目标行例程序,或者,如果该目标设备是中间网间连接器(*gateway*)目标设备,就可在第二网络上将第一格式的数据分组传送给最终的目标设备。

本发明使在一个或多个链接网络上的通信例行程序之间的数据传送更为方便,在本发明中,被例行程序传送和读出的信息的数据格式不同于一个或多个网络能够传送的数据格式。

通过参看以下的详细描述和附图将更加容易理解本发明的上述特征以及其它特征和优点。

图 1 表示可以按照本发明运行的个人计算机的等角图;

图 2 表示微处理系统的方框图,该系统可以与图 1 的个人计算机一起被使用;

图 3 表示图 1 的个人计算机与之连接的无连接网络和异步传送模式("ATM")网络的方框图;

图 4 表示在与图 3 的网络连接的处理系统中执行的通信例行程序和示范性应用程序的方框图，这些程序可用来在网络之间通信和传送数据；

图 5 表示可以在图 4 的处理系统中使用的服务器连接例行程序的流程图；

图 6 表示可以在图 4 的处理系统中使用的用户连接例行程序的流程图；

图 7 表示可被图 4 所示的封装程序—解封装程序使用的封装例行程序的流程图；

图 8 表示可被图 4 所示的封装程序—解封装程序使用的解封装例行程序的流程图。

图 1 表示可在图 3 和 4 所示的处理系统网络内起异步传送模式(“ATM”)启动主机作用的个人计算机(“PC”)100 的等角图。该 PC 100 由硬件机箱 101 (表示为具有剖开的视图)、监视器 104、键盘 105 和任选的鼠标器 108 组成。应当指出，监视器 104 以及键盘 105 和鼠标 108 都可以分别被任何恰当地设置的输出和输入设备来代替。硬件机箱 101 包括软盘驱动器 102 和硬盘驱动器 103。软盘驱动器 102 能够接收、读出和写入外部磁盘，而硬盘驱动器 103 能够提供快速存取的数据存储和检索。还可以给 PC100 安装用于接收和传送数据的任何恰当地设置的结构，例如包括磁带和光盘驱动器，以及串行和并行数据端口。

在硬件机箱 101 的剖开部分内的是例如与存储器存储设备连接的中央处理单元(“CPU”)106 这样的处理单元，在所示的实施例中，存储器存储设备是随机存取存储器(“RAM”)107。CPU 106



还与 ATM 网络接口 109 以及例如“Internet”接口 110 这样的无连接网络接口连接。合适的 ATM 网络接口 109 是可以位于宾夕法尼亚州的匹兹堡的 Fore 系统公司(Fore System, Inc.)大量获得的 ATM 主机适配器卡。合适的无连接网络接口 110 包括在商业上可用于与 Internet 网连接的网络接口。

虽然 PC 100 被表示为具有单个 CPU 106, 但 PC 100 也可设有多个 CPU 106 共同实现本发明的原理。同样地, 虽然 PC 100 被表示为具有单个本地存储器存储设备 107, 但 PC 100 也可装多个本地存储器存储设备。另外, 虽然 PC 100 被用来说明在处理系统网络内的 ATM 启动主机的一种实现方式, 但本发明也可以在任何具有至少一个处理单元的处理系统内被实现, 这些系统例如包括复杂计算器以及手持、微型计算机, 大型机和超级计算机, 还包括 RISC 和并行处理体系结构, 本发明还可以在前述处理系统网络的组合中被实现。

图 2 表示可以与图 1 的个人计算机 100 一起被使用的微处理系统的方框图。该微处理系统包括通过数据总线 203 与例如 RAM 107 这样的存储器存储设备连接的例如 CPU 106 这样的单个处理单元。存储器存储设备 107 能够存储可被处理单元 106 检索、解释和执行的一条或多条指令。

处理单元 106 包括控制单元 200、算术逻辑单元(“ALU”)201 和本地存储器存储设备 202, 该本地存储器存储设备 202 例如可以是可堆栈的高速缓冲存储器或多个寄存器。控制单元 200 能够从存储器存储设备 107 检索指令。ALU 201 能够执行多种操作, 包括执行指令所需的加法运算和布尔“与”运算。本地存储器存储设备

202 能够提供用于存储暂时结果和控制信息的高速存储功能。

本发明的过程间通信和网间数据传送方法和系统的预期使用就是在位于 *internet* 和 ATM 网络上的设备之间提供通信和数据传送。因此，以下参照这些无连接和面向连接的网络分别描述本发明，但这并不意味着对可以恰当地利用本发明的网络种类的限制。

图 3 表示能够使 ATM 为主机，诸如图 100 的处理系统，链接到 ATM 网 300 和无连接网，诸如网间接口 310 的框图。处理系统 100 可作为在网络 300 和 310 之间进行数据传送的网间连接器。ATM 网络 300 将处理系统 100 与处理系统 320 和 321、付费电视 (*video on demand*) 服务设备 322、电话系统 323 以及可视电话系统 324 进行连接。ATM 网络是通过将语音、数据或视频信息组织成为小的单元或数据分组就能够在与 ATM 网络 300 连接的设备之间传送这种信息的信元交换网络。如在“ATM 讨论会用户网络接口 3.0 (*ATM Forum User - Network Interface 3.0*)” (*Prentice-Hall, 1993*) (“*ATM Forum UNI reference*”) 中所描述的那样，CCITT 和 ATM 讨论会已经标准化 ATM 网络的许多特性和特征，该文件的全文援引在此作为参考文献。ATM 信元通过 ATM 网络 300 被高速地传送，目前以每秒 50 兆位到每秒 2.4 千兆位。

*internet* 网 310 将处理系统 100 与处理系统 330、331 和 332 以及局域网 (“LAN”) 333 进行链接。LAN 333 是利用连续电缆或室内语音数据电话系统在建筑物或综合结构内将各种硬件设备连接在一起的通信网络。*internet* 通常被定义为作为一个整体进行操作的独立或特殊的无连接网络的任何组合，可以包括利用若干种

协议中的任一协议、例如“internet 网协议”(“IP”)相互进行连接的各种网络的世界范围的网络。协议规定了文件传送、远端登记、电子邮件和包括分布处理的其它服务以及其它资源。IP 的功能之一是使源节点、例如处理系统 100 的 IP 数据分组能够在其传送过程中通过多个网络到达最终的目标节点,不必首先建立虚拟线路或“连接”。

由于 ATM 网络需要建立虚拟线路以及数据分组格式的失配,在无连接网络、例如 internet 310 上已有的应用程序不经修改就不能利用在 ATM 网络、例如网络 300 上可获得的服务。

图 4 表示根据本发明的使示范性应用程序 A 400 和 B 400 能够与在 ATM 网络 300 上的设备进行通信的一实施例的通信过程和例行程序的方框图布置。在以下描述中,当另一例行程序请求时进行服务的例行程序被称为服务器例行程序。另外,请求该服务的例行程序被称为用户例行程序。

在图 4 中,被图 3 的处理系统 100 和 330 执行的例行程序被分别包括在标为 100 和 330 的虚线框内。在处理系统 100 内,示范性应用程序 A 400,它可以是无连接的服务器或用户程序,与连接服务例行程序库 405 通信。可以利用在图 2 所示的处理单元 106 内的过程间通信或利用在处理系统 100 内的两个合适编程的电路或设备之间的连接来进行通信。

连接服务例行程序库 405 与连接管理器 410 通信。连接管理器 410 还与 internet 协议(“IP”)堆栈接口 415 和 ATM 协议堆栈接口 420 通信。堆栈接口 415 和 420 还与封装器—解封装器 425 和应用程序 A 400 通信。连接管理器 410 可以是在图 1 所示的处

理单元 106 上操作的程序或例行程序，或是包含在处理系统 100 内的合适编程的设备。在作为参考文献的“*ATM Forum UNI reference*”的第 149—292 页的 § 5 中描述了合适的连接管理器 410 的例行程序。

IP 堆栈接口 415 与网络接口 110 通信并使其产生处理系统 100 和 *internet* 310 之间的数据传送所必须的信号。IP 堆栈接口 415 是 IP 语义的实现。换句话说，IP 堆栈接口 415 执行 *internet* 网协议(包含信令)规定的每一层所需的抽象功能。IP 堆栈接口 415 可以是在处理单元 106 中运行的例行程序或是包含在处理系统 100 内的合适编程的电路或插件板。在 S. J. Leffler、M. K. McKusick、M. J. Karels 和 J. S. Quarterman 的“4.3 BSD UNIX 操作系统的设计和实现”(Addison—Wesley, 1989) (“*BSD reference*”)中描述了合适的 IP 堆栈接口 415，该著作的全文在此作为参考文献。IP 堆栈接口 415 与应用程序 A 400 通信以便以普通方式实现与在 *internet* 网 310 上的其它设备的数据传送。

同样地，ATM 协议堆栈接口 420 与 ATM 网络接口 109 通信并使其产生启动处理系统 100 和在 ATM 网络 300 上的设备、例如处理系统 320 之间的数据传送的正确信号。ATM 协议堆栈接口 420 实现这些语义或本地模式 ATM 堆栈的抽象的功能。在“*BSD reference*”中描述了合适的 ATM 协议堆栈接口 420。

封装器—解封装器执行 IP 数据分组中 ATM 格式帧的封装和解封装，以便通过 *internet* 网 310 传送和接收 ATM 格式帧。可以在封装器—解封装器 425 中用来对 ATM 格式帧进行封装和解封装的方法在以下参看图 7 和 8 分别进行详细的描述。

因此, *ATM* 协议堆栈接口 420 可以通过 *ATM* 网络接口 109 在应用程序 A 400 和 *ATM* 网络 300 上的设备之间、或通过封装器—解封装器 425 和 *IP* 协议堆栈接口 415 在应用程序 A 400 和 *internet* 网 310 上的设备之间进行 *ATM* 格式帧的数据传送。此外, 堆栈接口 415 和 420 以及封装器—解封装器 425 还可以用作中间连接器处理系统或能在 *internet* 网 310 上执行 *ATM* 的主机, 以便在位于网络 300 和 310 上的设备之间传送数据。

以类似的方式, 在处理系统 330 内, 应用程序 B 430, 也可以是无连接服务器或用户例行程序, 与相应的连接服务例行程序库 435、*IP* 堆栈接口 440 和 *ATM* 协议堆栈接口 455 通信。连接服务例行程序库 435 还与 *IP* 堆栈接口 440 通信。*IP* 堆栈接口 440 还从与 *internet* 网 310 连接的 *internet* 网网络接口 445 传送和接收 *IP* 数据分组。

*IP* 堆栈接口 440 和 *internet* 网网络接口 445 可分别按照基本上与处理系统 100 中的 *IP* 堆栈接口 415 和网络接口 110 相同的方式运行。同样地, *ATM* 协议堆栈接口 455 和封装器—解封装器 460 按照与在处理系统 100 中基本相同的方式运行。但是, *ATM* 协议堆栈接口 455 只发送和接收应用程序 B 430 和封装器—解封装器 460 之间 *ATM* 格式帧, 不需任何 *ATM* 网络。

图 4 所示的应用程序 A 400 和 B 430 可以是已有的应用例行程序, 它们已被修改, 用来传送 *ATM* 格式帧并且包括用于建立与远端用户或服务器程序的通信连接的连接服务例行程序库。远端程序可以在与 *ATM* 网络 300 连接的设备、例如图 3 和 4 的处理系统 320 上运行。连接管理器 410 建立并维护实现例行程序之间的通

信的连接或虚拟线路。因此，处理系统 100 内的连接服务例行程序库 405 可以直接与连接管理器 410 通信。类似地，连接服务例行程序库 435 可以按照通常的方法经由 IP 堆栈接口 415 和 440 通过 internet 网 310 与连接管理器 410 通信。

连接服务例行程序库 435 包含与连接管理器 410 通信的合适例行程序。以便建立和维护虚拟线路或连接。以下参看图 5 和 6 更详细地描述可包括在连接例行程序库中的若干例行程序的操作。另外，应用程序 A 400 和 B 430 可以是面向连接的程序，例如在 ATM 网络 300 上运行的程序。在这种情况下，程序 A 400 和 B 430 不必修改就能够使用相应的连接服务例行程序库 405 和 435 来建立连接。

在已建立了两例行程序、例如应用程序 B 430 和在处理系统 330 上的远端程序之间的连接之后，就可以按照以下方式在这两例行程序之间传送数据。参看图 4，假定连接的例行程序能够利用 ATM 协议传送和接收数据，程序 430 将数据传送给在处理系统 330 中的 ATM 协议堆栈接口 455。ATM 协议堆栈接口 455 将数据整理成 ATM 格式帧并将其传送给封装 IP 数据分组中数据分组的封装器—解封装器。以下参看图 7 描述一种合适的封装例行程序。

被封装 IP 数据分组然后被传送给 IP 堆栈接口 440，IP 堆栈接口 440 经由 IP 网络接口通过 Internet 网再将其传送给处理系统 100。在处理系统 100 中，IP 堆栈接口 415 从 IP 网络接口 110 接收 IP 数据分组。IP 堆栈接口 415 将 IP 数据分组识别为被封装 IP 数据包并将其传送给封装程序—解封装程序 425 进行解封装。以下参看图 8 描述一种合适的解封装例行程序。

解封装 ATM 格式帧然后被封装器—解封装器 425 传送给 ATM 协议堆栈接口 420, ATM 协议堆栈接口 420 根据控制管理器 410 的信息识别 ATM 格式帧的目的地。ATM 协议堆栈接口 420 然后控制 ATM 网络接口 109, 以便通过 ATM 网络 300 将 ATM 格式帧传送给处理系统 320。

图 5 表示 可被用来建立连接以便与远端面向连接的例行程序交换数据的服务器连接例行程序 500 的流程图。远端面向连接的例行程序可以在与 ATM 网络 300 或 Internet 网 310 连接的设备上运行。相对于在图 4 的处理系统 330 上实现的应用程序 B 430 将服务器连接例行程序 500 描述为服务器程序。但是, 该方法可被在与 Internet 网 310 连接的设备上运行的任何无连接服务器例行程序、例如在处理系统 100 上运行的应用程序 A 400 使用。

参看图 5, 在步骤 510 中, 当服务器应用程序 B 430 希望宣传其服务时, 它命令在连接服务例行程序库 435 中的服务例行程序向连接管理器 410 输出可用的服务器程序信息。该服务器程序信息包含可用服务的服务器名。该信息可以按照通常的方式通过 internet 网 310 经由 IP 堆栈接口 440 和 415 以及它们相应的网络接口 445 和 110 被传送给连接管理器 410。该输出信息还可以包含网络端口号, 服务器应用程序 B 430 将一直在该网络端口上监听来自连接管理器 410 的信息。一旦从服务器应用程序接收到可用服务信息, 连接管理器 410 就在可用服务表中登记该可用服务器名及其端口号。连接管理器 410 然后传回确认这种登记的信息。

在步骤 510 中输出了可用服务信息后, 服务器应用程序 B 430 就在步骤 520 中等待连接管理器 410 将确认登记信息传送给指定

的网络端口。在步骤 520 中接收了确认登记信息后，服务器应用程序 B 430 再在步骤 530 中等待连接管理器 410 传送的即将连接信息。这种信息表示远端用户程序已经向连接管理程序 410 请求与服务器应用程序 B 430 进行连接。在例如与 ATM 网络 300 连接的处理系统 320 上运行的远程客户程序可以产生这样的服务连接请求。一旦接收到远程客户程序对于服务器应用程序 B 430 提供的可用服务的请求，连接管理程序 410 就会将即将连接信息传送给服务器应用程序 B 430。

为了提供安全措施，传送给服务器应用程序 B 430 的即将连接信息可以包含连接关键字。该连接关键字可以是 16 位信息字，该 16 位信息字向保持 IP 堆栈接口、例如在这一例子中的 IP 堆栈接口 440 给出了按照下述方式建立连接所需的信息。

在步骤 530 中接收了即将连接信息后，服务器应用程序 B 430 就在步骤 540 中决定它是否将接受连接。如果程序 430 在步骤 540 中决定不接受连接，则在步骤 550 中将拒绝连接信息传送给连接管理器 410 并终止服务器连接例程序。拒绝连接信息被连接管理器 410 的接收可以使连接管理器 410 从可用服务表中消除相应的可用服务名。在另一种方案中，一旦在步骤 550 中发送了拒绝连接信息，服务器连接例程序 500 可以返回到步骤 530，在此等待连接管理器 410 的另一连接请求。在这样一种可供选择的方案中，连接管理器 410 在接收到拒绝连接信息之后将不从可用服务表中消除相应的服务名。

但是，如果服务器连接例程序 500 在步骤 540 中决定接受连接，则例程序 500 就到达步骤 560。在步骤 560，服务器应用程



序 B 430 使连接服务例程序库 435 向连接管理器 410 传送接受连接信息。该接受连接信息可以包含连接管理器 410 先前传送的连接关键字,以便保证将在合适的用户和服务器程序之间建立连接。

然后服务器应用程序 B 430 在步骤 570 中等待接收被连接管理器 410 传送的、相应于被建立的连接的虚拟线路标识符(“VCI”)值。VCI 值是在 ATM 网络中使用的唯一标识符,它与数据将在相应的处理系统 330 和 320 中的服务器和远端用户程序之间通过的虚拟线路或数据通路相对应。VCI 值识别被连接管理器 410 维护的服务器和远端用户程序之间的连接,并确保连接管理器 410 将把利用该 VCI 值传送的数据送到合适的处理系统。在以上引用的“ATM forum UNI reference”中提供了对 VCI 更详细的说明。

在步骤 570 中接收了 VCI 值后,服务器应用程序 B 430 就在步骤 580 中将该 VCI 值传送给 ATM 协议堆栈接口 455。该 ATM 协议堆栈接口 455 利用该 VCI 值来将从 Internet 网 310 接收的、具有该特征 VCI 值的任何被接收的 ATM 格式帧传送给服务器应用程序 B430。换句话说,步骤 580 的功能可被看作是将 VCI 值配联给相应的 ATM 协议堆栈接口 455 和服务器应用程序 430。处理系统 100 的连接管理器 410 保存了 VCI 映射表,该 VCI 映射表包含被建立连接的 VCI 值和执行被连接程序的相应处理系统的 internet 网网络端口号。然后在连接管理器 410 的信息的控制下,利用处理系统 100,通过网络 300 和 310,将被连接服务器和远端用户程序产生的数据传送到它们各自的目的地。

连接管理器 410 可以通过不断地检查和更新 5 个表来执行其

操作。这 5 个表包括上述的可用服务表和 VCI 映射表、等待可用服务器程序的用户程序的用户请求表、存储连接等待接受的标识的连接请求表、以及配联连接表。配联连接表包含那些已被连接管理器 410 建立但尚未被配联给相应的 ATM 协议堆栈接口的连接。

通过把相对于图 5 的服务器连接例行程序 500 描述的连接服务例行程序库调用包括进来就可以容易地修改任何已有的无连接服务器应用程序来建立连接。例如，可将服务器应用程序修改为包括以下的连接服务例行程序库调用：

- (1) `export_service(service Name, TCP_PORT);`
- (2) `conkey = await_service_request();`
- (3) `VCI = accept_connection(conkey);`
- (4) `bind(VCI);`

库调用 `export-service` 相当于例行程序 500 的步骤 510，并发送服务名 `service Name` 和本地网络端口号 `TCP-PORT`。该本地网络端口号相应于服务器程序将在其上一一直监听来自连接管理器 410 的响应的网络端口。库调用 `await-service-request` 相当于步骤 530，传回任选连接关键字 `conkey`。库调用 `accept-connection` 相当于步骤 560 和 570，发送任选关键字以便提供连接管理器 410 正在与合适的服务器程序连接的安全措施。库调用 `accept-connection` 传回接收的 VCI 值。库调用 `bind` 相当于配联相应的协议堆栈接口，以便利用在 internet 网 310 上传送的合适的 VCI 值将任何数据传送给服务器应用程序。

图 6 表示 可被处理系统用来获得与所需远端面向连接的服务器程序连接的虚拟线路的用户连接例行程序 600。例如，该用户连

接例行程序 600 将相对于在图 4 的处理系统 100 上运行的应用程序 A 400 被描述为用户程序。相应的远端服务器程序可位于与 ATM 网络 300 或 Internet 网 310 连接的设备上。

参看图 6, 在步骤 610 中, 用户应用程序 A 400 使连接服务例行程序库 405 向连接管理器 410 发送请求信息。该请求信息可以包含例如某一所需服务器名和本地网络端口地址这样的信息, 用户程序将在该本地网络端口地址处监听响应。用户应用程序 400 然后在步骤 620 中等待对被连接管理器 410 传送到指定的本地网络端口地址的请求的确认。连接管理器 410 传送的后续确认可以包含相应的连接关键字, 可以按照与以上相对于图 5 的服务器连接例行程序 500 所描述的关键字相同的方式来使用该关键字。

在步骤 620 从连接管理器 410 接收了确认后, 例行程序 600 然后在步骤 630 中等待来自连接管理器的、已建立了与被请求远端服务器程序的连接的后续信息。连接管理器 410 可以按照以上相对于图 5 的例行程序 500 所描述的方式建立与远端服务器程序的连接。表示连接已被建立的接收信息可以包含该连接的相应 VCI 值。用户应用程序 A 400 按照与以上也是相对于图 5 的例行程序 500 所描述的方式基本相同的方式使用该 VCI 值。

在步骤 630 中接收了表示连接已被建立的信息后, 例行程序到达步骤 640。在步骤 640, 用户机连接例行程序将 VCI 值传送给 ATM 协议堆栈接口 420, 以便将该 VCI 值配联给用户应用程序 A 400。配联步骤 640 使 ATM 协议堆栈接口 420 将具有合适的 VCI 值的任何数据分组传送给用户应用程序 A 400。应当指出, ATM 协议堆栈接口 420 能够经由 ATM 网络接口 109 通过 ATM 网络

300 以及经由封装器—解封装器 425 和 IP 堆栈接口 415 通过 internet 网 310 接收和传送 ATM 格式帧。因此, 利用 ATM 格式帧进行通信的用户和服务器程序可以在与 internet 网 310 连接的并具有 ATM 协议堆栈接口和封装器—解封装器的两台设备上进行。这种例行程序将正常地运行, 不理睬 ATM 网络的不存在。

通过把可被包括在连接服务例行程序库中的以下两条示范性库调用包括进来就可以修改已有的无连接用户应用程序来与面向连接的服务器程序通信:

- (1) `VCI = open_connect(ServerName, TCP_PORT);`
- (2) `connect(VCI).`

库调用 `open-connect` 发送所需的服务器名 `Server Name` 和本地网络端口地址 `TCP-PORT`, 用户程序将在该地址处监听来自连接管理器 410 的响应。一旦发现可用服务器程序, 连接管理器 410 就传回相应连接的 `VCI` 值。库调用 `open-connect` 将接收的 `VCI` 值传给用户应用程序 A 400。相应地, 库调用 `open-connect` 执行图 6 中的用户连接例行程序 600 的步骤 610、620 和 630。库调用 `connect(VCI)` 将 `VCI` 值传送给相应的 ATM 协议堆栈接口, 以便命令其在网络上传送具有合适的 `VCI` 值的任何保留数据分组。服务例行程序 `connect(VCI)` 相应于图 6 的步骤 640。

图 5 和 6 的服务器和用户连接例行程序 500 和 600 分别使采用无连接网络技术的已有用户和服务器程序向面向连接的网络、例如 ATM 网络 300 提供服务或向其请求服务。本发明对无连接用户或服务器程序进行极小的修改就获得了附加的灵活性。因此, 根据本发明的一个方面, 可以容易地修改大量在 Internet 网上可获

得的程序来利用在 ATM 网络上提供的或者为供面向连接的网络使用而设置的各种服务。

将 IP 格式帧内的 ATM 格式帧进行封装的图 4 的封装器—解封装器 425 和 460 克服了在 internet 网 310 上传送 ATM 格式帧时数据格式失配的问题。图 7 表示封装例行程序 700 的流程图，该封装例行程序可以在封装器—解封装器中被执行以便在 IP 数据分组中对 ATM 格式帧进行封装，internet 网 310 进行传送。此外，图 8 表示解封装例行程序 800 的流程图，该解封装例行程序也可以在封装器—解封装器 425 和 460 中被执行、将接收的 IP 封装数据分组解封装成为 ATM 格式帧。

以下将相对于上述关于在应用程序 B 430 和在处理系统 320 上实现的远端例行程序之间的数据传送的例子来讨论例行程序 700 和 800。应用程序 B 430 这样传送 ATM 格式帧：首先将其送到 ATM 协议堆栈接口 455，ATM 协议堆栈接口 455 产生相应的 ATM 格式数据分组、例如 AAL 帧。在 IP 数据分组内进行封装的一种合适的 AAL 帧格式是 AAL 5 格式。AAL 帧和 IP 数据分组的详细说明可以分别在上述的“ATM from UNI reference”和“BSD reference”中找到。ATM 格式帧然后被传送给对每一帧执行图 7 的封装例行程序 700 的封装器—解封装器 460。

参看图 7，在步骤 710 中，封装器—解封装器 460 获得要被封装和传送的 ATM 格式帧的 VCI 值和中间目标地址。中间目标地址相应于具有与 internet 网 310 和 ATM 网络 300 连接的链路的最近的具有 ATM 的主机或中间连接器、例如图 4 中的处理系统 100 的 Internet 网网络地址。

例行程序 700 然后在步骤 720 中编制中间数据分组首部。该中间数据分组首部可以包含例如发送数据的源设备的 ATM 地址、连接的相应 VCI 值和数据分组顺序位置号这样的信息。数据分组顺序位置号是相应于当被传送信息包括若干帧时，帧的顺序位置的值。由于 Internet 网 310 是可以根据在每一数据分组首部中的目标地址通过不同的路径将每一 IP 数据分组送到目标设备的无连接网络，所以数据可以不按顺序接收。中间数据分组首部中的数据分组顺序位置号使中间连接器处理系统能够检测被传送的封装 IP 数据分组是否不按顺序被接收，如果是，就检测接收数据分组的固有顺序。

然后在步骤 730 中通过将在步骤 720 中编制的中间数据分组首部附加到要被传送的 ATM 格式帧来产生中间格式数据分组。例行程序 700 接着在步骤 740 中产生 IP 数据分组首部。该 IP 数据分组首部是 20 字节的首部，包含在 internet 网 310 上传送相应 IP 数据分组所需的信息和格式。这种首部包含最接近的 Internet 网处理系统的 Internet 网络地址和规定数据分组的 IP 格式类型的数据字段。包含在 IP 格式类型字段中的值应当指出数据分组是被封装的 ATM 格式帧。

然后在步骤 750 中将产生的 IP 数据分组首部附加到产生的中间数据分组以便形成 IP 数据分组。IP 数据分组由 20 字节的 IP 数据分组首部和变长数据部分或有效负载组成。例行程序 700 产生的 IP 数据分组的有效负载包含中间数据分组首部和 ATM 格式帧。产生的 IP 数据分组具有通过 Internet 网 310 进行传送的合适数据分组格式。

产生的 IP 数据分组然后在 Internet 网 310 上被传送并被处理系统 100 接收。相应的 IP 堆栈接口 415 读 IP 数据分组首部来获得 IP 数据分组格式类型信息。如果该 IP 数据分组格式类型如在这一例子中那样指出该数据分组包含被封装的 ATM 格式帧, 则该 IP 数据分组被传送给封装器—解封装器进行解封装。可在封装器—解封装器中被用来对 ATM 格式帧进行解封装的解封装例行程序 800 如图 8 所示。

参看图 8, 在步骤 810 中, 从接收的 IP 数据分组中去除 IP 数据分组首部来获得中间数据分组。该中间数据分组然后在步骤 820 中被分离为中间数据分组首部和 ATM 格式帧。然后在步骤 830 中从分离的中间数据分组首部中获得源地址、数据分组顺序位置号和 VCI 值。

然后在步骤 840 中, 根据数据分组顺序位置号, 将该 ATM 格式帧连同以前接收的其它 ATM 格式帧排列到其固有顺序位置。按照固有顺序排列 ATM 格式帧的一种合适的技术是将接收帧存储到与封装器—解封装器 425 连接的存储器缓冲器中。然后根据如在步骤 850 中指出的数据分组顺序位置号将存储帧按照它们的固有顺序输出给 ATM 协议堆栈接口 420。此外, 相应的中间数据分组首部信息可以借助每一帧传送给 ATM 协议堆栈接口 420。

在这一例子中, ATM 协议堆栈接口 420 将使 ATM 网络接口 109 产生所需的尾部, 例如 AAL5, 如果 ATM 格式帧是 AAL5 帧的话, 并通过 ATM 网络 300 将具有附加尾部的帧传送给目标处理系统 320。ATM 协议堆栈接口 420 将根据来自连接管理器 410 的信息将该帧传送给特定的目标。连接管理器 410 根据其 VCI 值

确定单元的正确目标。在第一可供选择的实施例中,可以在相应的帧被封装器—解封装器 460 封装之前由以上例子中的 ATM 协议堆栈接口 455 来产生尾部。

当数据沿相反方向传送时,封装器—解封装器 425 和 460 将按照基本相同的方式运行。例如,当将 ATM 格式帧从图 4 中的处理系统 320 传送到应用程序 B430 时,处理系统 320 将根据来自连接管理器 410 的信息通过 ATM 网络 300 将 ATM 格式帧传送给处理系统 100 的 ATM 网络接口 109 和 ATM 协议堆栈接口 420。ATM 协议堆栈接口 420 也将根据来自连接管理器 410 的信息将 ATM 格式帧传送给封装器—解封装器 425 进行封装。相应的被产生 IP 数据分组然后将通过 Internet310 被传送给处理系统 330。

在处理系统 330 内,被接收 IP 数据分组将被 IP 堆栈接口 440 传送给封装器—解封装器 460。封装器—解封装器 460 然后解封装该数据分组并将该被解封装 ATM 格式帧传送给 ATM 协议堆栈接口 455, ATM 协议堆栈接口 455 把该帧提供给应用程序 B430。

在另一可供选择实施例中,可以在 ATM 协议堆栈接口中实现封装例行程序 700,在相应的 IP 堆栈接口中实现解封装例行程序 800,这些堆栈接口在处理系统中相互直接进行通信。这一实施例消除了每一处理系统中对于单独的封装器—解封装器例行程序或电路的需要。

因此,本发明通过采用例如图 3 和 4 的系统 100 这样的中间连接器处理系统以及按照上述方式在远端例行程序之间建立连接和网络间数据传送的方法和系统扩展了与 internet 网和 ATM 网络连接的设备可得到的服务。



虽然已在上面对描述了提供过程间通信和网络间数据传送的几个实施例，但本领域普通技术人员很容易理解不违背本发明的教导可以对所述实施例进行许多修改。要求保护的本发明将覆盖全部这样的修改。例如，可以在无连接网络和面向连接的网络之间使用多个网关处理系统，每一个这种系统可以具有控制在无连接网络上最接近的设备的连接的管理器，例如连接管理器 410。

还有，上述每一个例行程序、包括连接管理器，连接服务例行程序库和封装器—解封装器都可以作为在相应的处理系统上运行的软件例行程序或作为软件例行程序和专用电路的组合来实现，以便提供它们指定的功能。

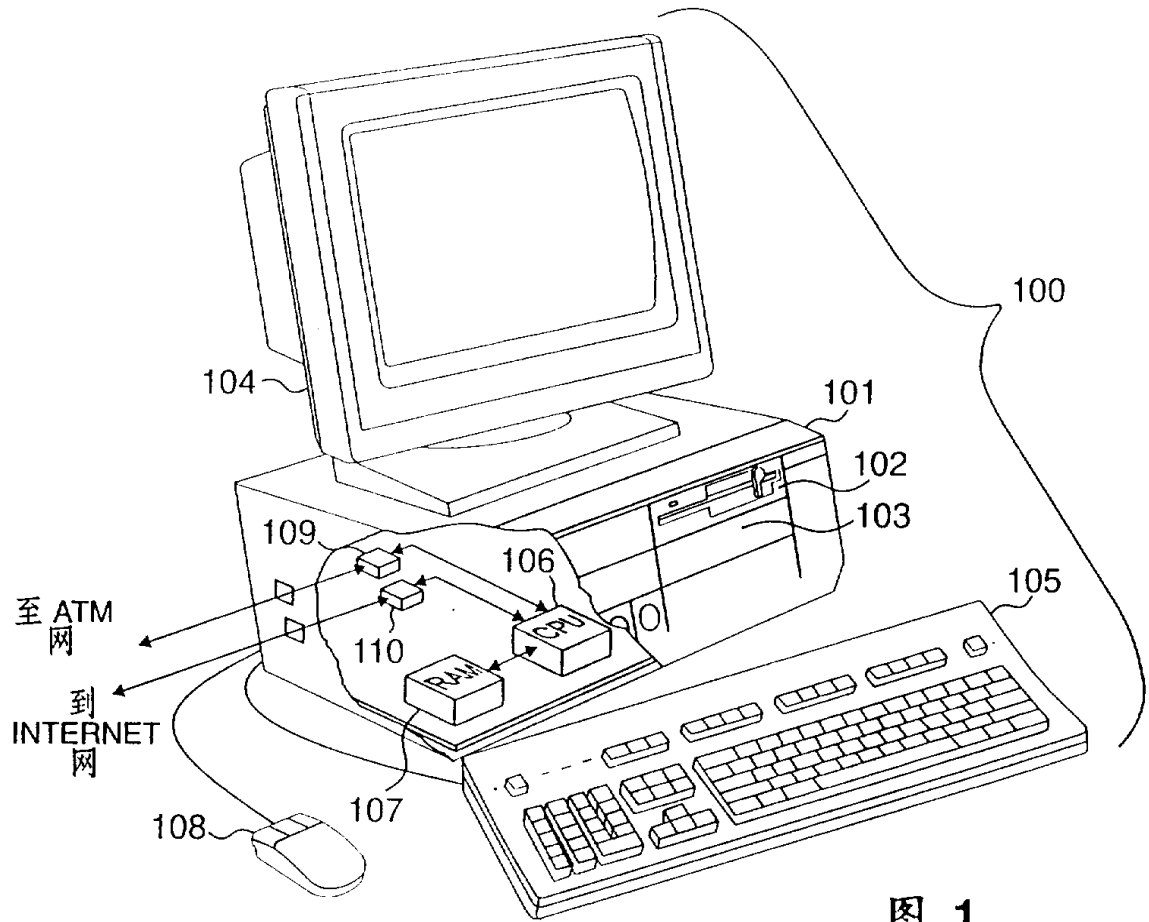


图. 1

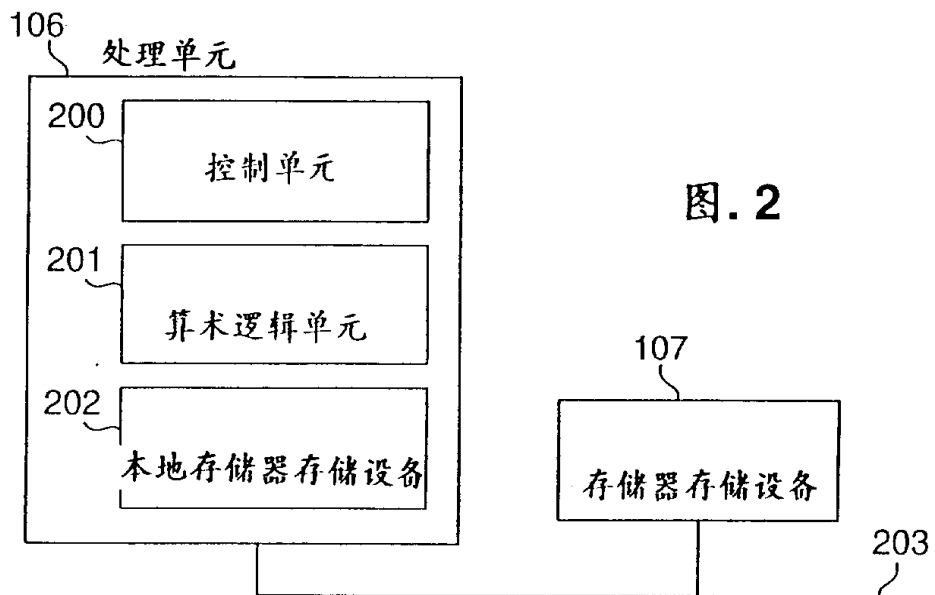


图. 2

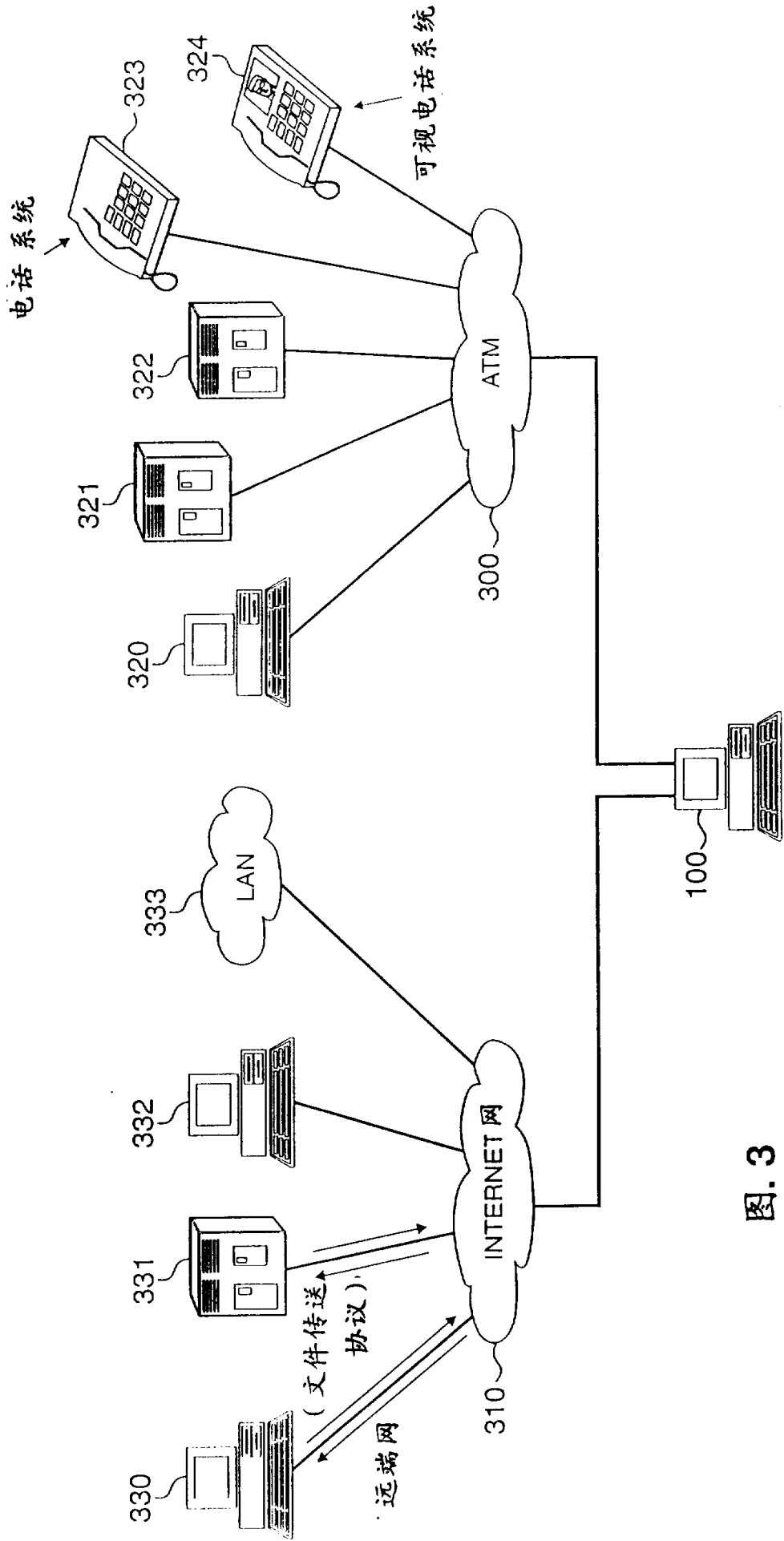


图. 3

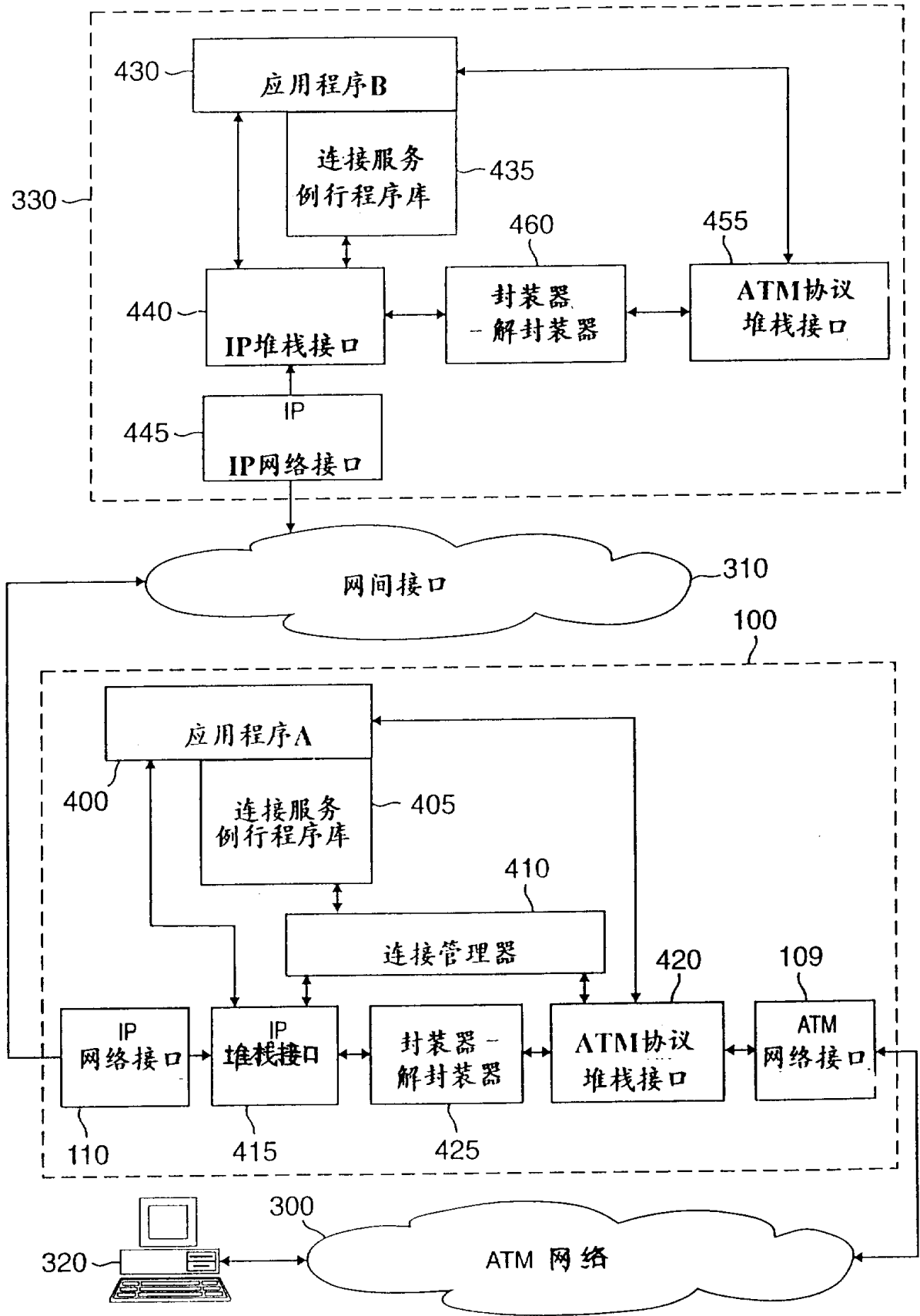


图. 4

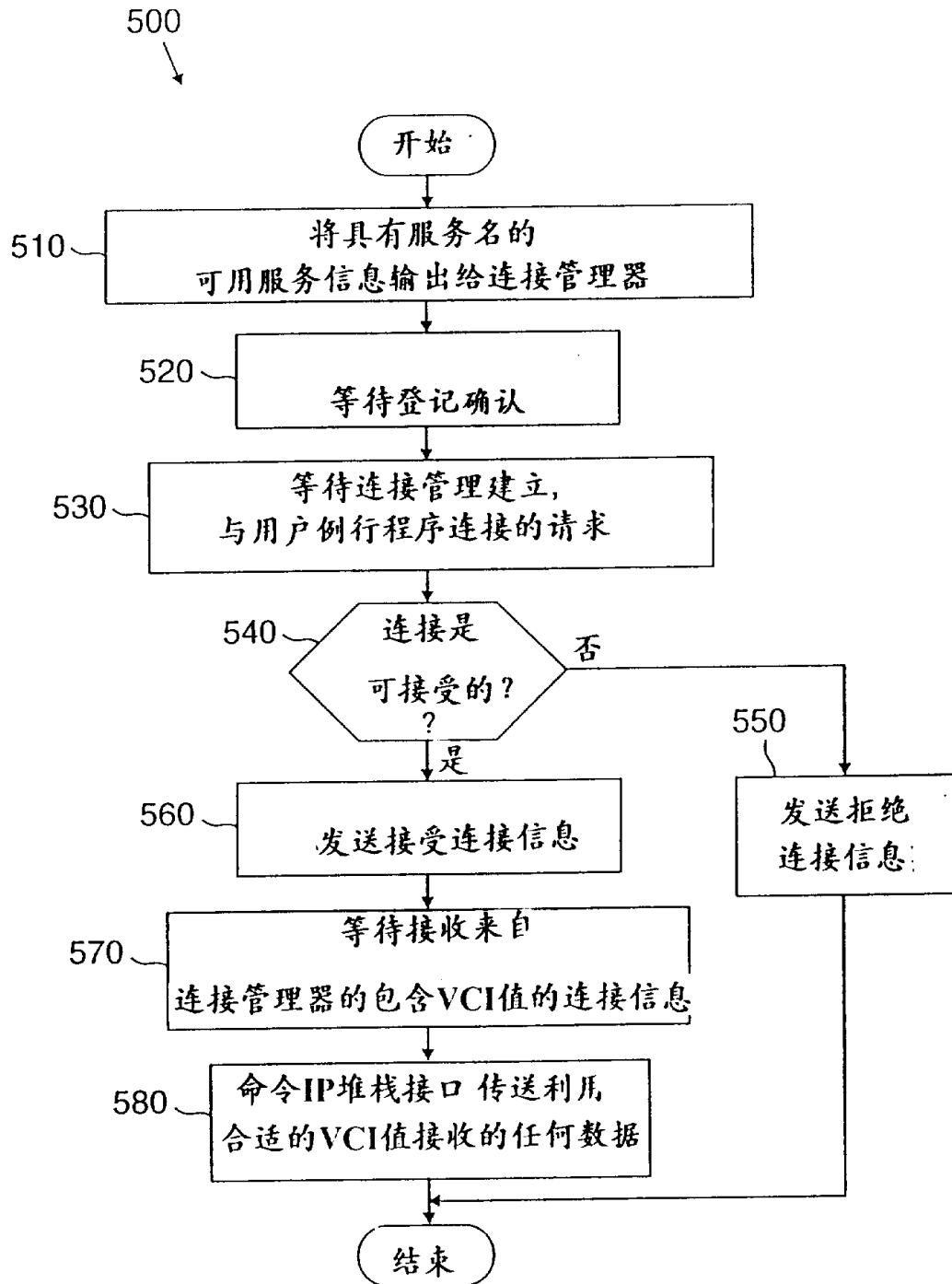


图. 5

600

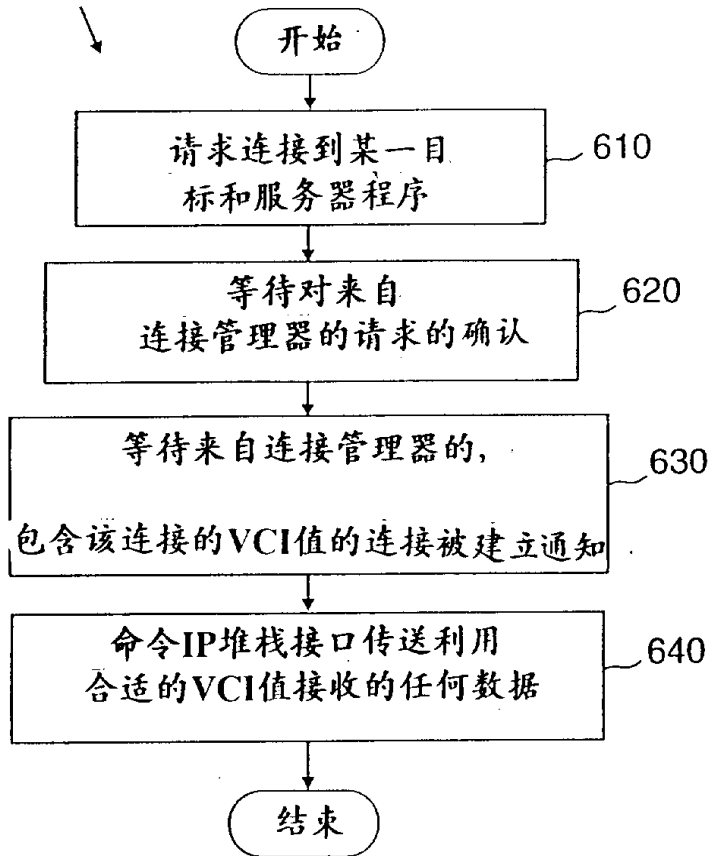


图. 6

700

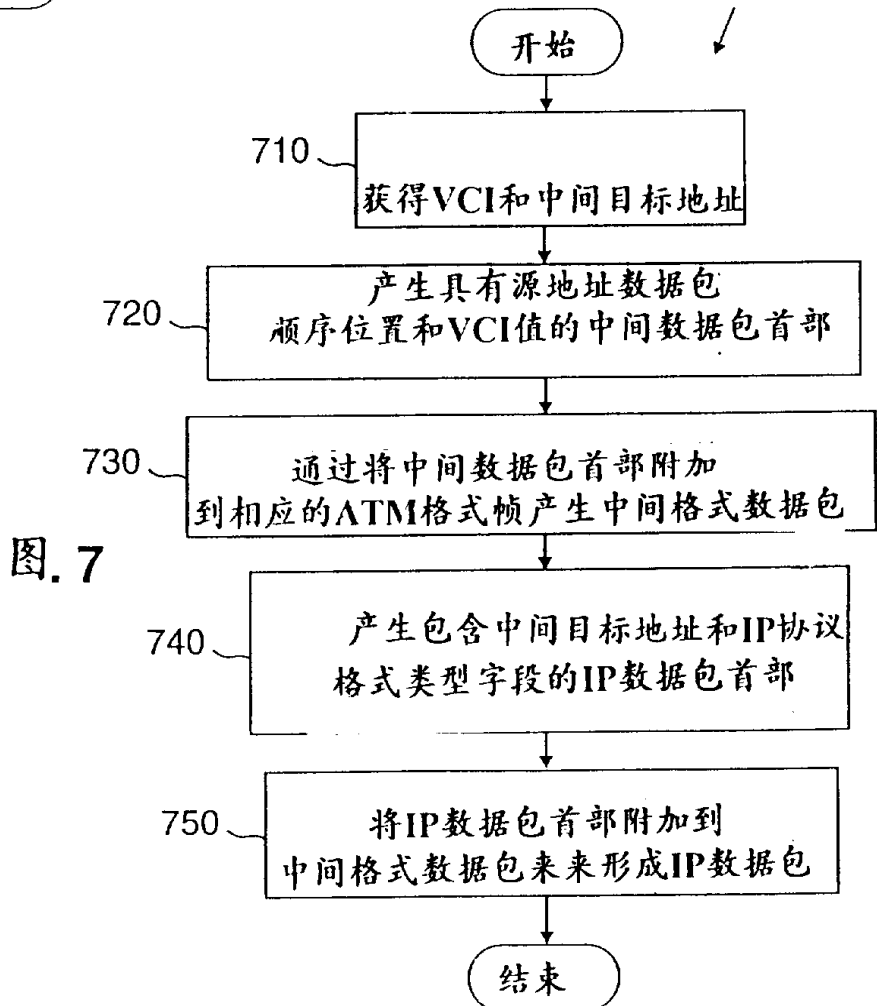


图. 7

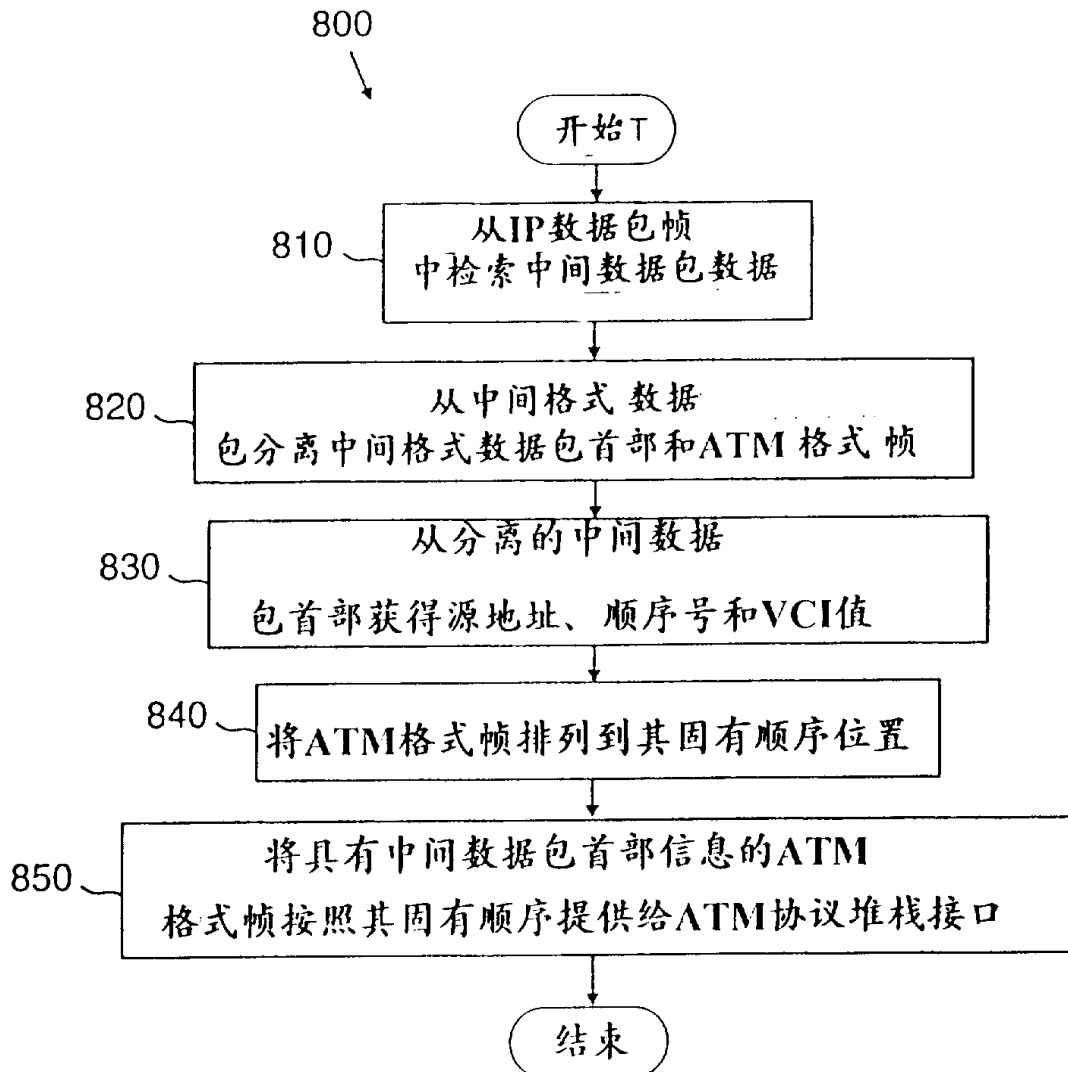


图. 8