[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06F 17/30
H04L 29/06

[12]发明专利申请公开说明书

[21]申请号 98806358.1

[43]公开日 2000年7月19日

[11]公开号 CN 1260890A

[22]申请日 1998.5.15 [21]申请号 98806358.1
[30]优先权
[32]1997.5.22 [33]US [31]08/861,934
[86]国际申请 PCT/US98/09943 1998.5.15
[87]国际公布 WO98/53410 英 1998.11.26
[85]进入国家阶段日期 1999.12.17
[71]申请人 波士顿大学理事会
地址 美国马萨诸塞州
[72]发明人 苏雷曼·A·米尔戴得
阿黛尔莎拉姆·A·希达亚
大卫·J·耶特斯

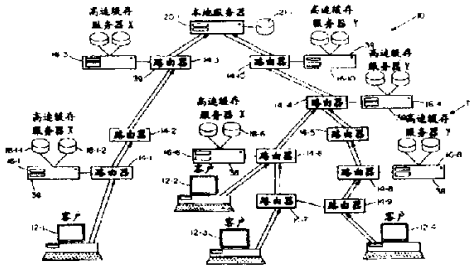
[74]专利代理机构 永新专利商标代理有限公司
代理人 程 伟

权利要求书 18 页 说明书 34 页 附图页数 10 页

[54]发明名称 分布式高速缓存、预取和复制的方法和系统

[57]摘要

一种在计算机网络中用于自动的、透明的、分布式的、可 定标的和稳定的高速缓存、预取和复制技术，其中请求具体文 档的报文遵照形成路由图的路径从客户到本地服务器。没有高 速缓存，客户请求将正常发生，此时它的路线是沿该路由图朝 本地服务器上行。但是，诸高速缓存服务器沿着该路径定位，并且如果它们能得到服务，高速缓存服务器可以监听请求。为 了在不脱离标准网络协议的情况下能够以这种方式为请求提供 服务，高速缓存服务器需要能够将滤包器插入与它相关的路由，还需要依据客户的意见代理该本地服务器。诸高速缓存服务器 可以协同工作基于其本地装载、在其毗邻高速缓存器上装载、附加通信路径装载和文档大众性通过高速缓存和废除文档来为 客户请求提供服务。高速缓存服务器还可以实现安全模式和其 它文档传输特征。



ISSN1008-4274

权 利 要 求 书

1. 在包含众多利用通信协议进行网上通信的计算机的系统中，一些在网络节点上的计算机作为本地服务器（或简称服务器）用于以文档的形式存储资料，而另一些计算机作为客户在应用层将文档请求报文发送给服务器，其中文档请求报文是请求储存在服务器的文档，在该系统中一种实现文档请求报文的方法包括下述步骤：

（a）将文档的本地高速缓存副本储存在网络中的众多中间节点位置；以及

（b）在响应其中一个具体客户时，产生倾向于发送给某个特定服务器的特定应用层文档请求报文，在低于该应用层选定的通信层监听文档请求报文并将一份本地高速缓存副本返回客户所在的应用层，借此来实现特定应用层的文档请求报文，以致该应用层请求报文在中间节点被较低层监听而且服务器上的应用层不接收应用层文档请求报文。

2. 根据权利要求 1 的方法，其中在选定的中间节点进一步包括如下步骤：

（c） 如果特定的文档请求报文不能通过提供一份本地高速缓存副本得以实现，将该文档请求报文发送到另一个节点。

3. 根据权利要求 1 的方法，其中在至少一个既非客户又非服务器的中间节点中实现特定文档请求报文的步骤进一步包括如下步骤：

（c） 储存特定文档的本地高速缓存副本，这些文档副本也被储存在其它中间节点；

（d） 确定储存本地高速缓存副本的毗邻中间节点的合法身份；以及

（e） 基于文档请求报文装载资源在中间节点的可用性在中间节点和毗邻中间节点之间分配执行文档请求报文。

4. 根据权利要求 3 的方法，其中报文装载执行资源包括高速缓存器。

5. 根据权利要求 3 的方法，其中报文装载执行资源包括通信路径装载。

6. 根据权利要求 3 的方法，其中分配执行文档请求报文的步骤进一步包括如下步骤：

(f)在中间节点和毗邻节点之间交换状态报文，该状态报文包括选自处理装载、通信路径装载、文档大小、文档请求报文响应时间、文档请求报文的请求率、文档请求报文的变化率或本地服务器可操作性中至少一项的资料。

7. 根据权利要求 1 的方法，其中实现特定文档请求的步骤通过进一步包括如下步骤进一步完成在诸节点相互连接的路径之间分配通信路径装载：

（c） 储存特定文档的本地高速缓存副本，这些文档副本也被储存在其它中间节点；

（d） 确定储存本地高速缓存副本的毗邻节点的合法身份；以及

（e） 基于通信路径装载在中间节点和毗邻中间节点之间分配执行文档请求报文。

8. 根据权利要求 2 的方法，该方法在选定的中间节点进一步包括如下步骤：

（d） 确定第一毗邻节点的合法身份，其中特定文档请求报文是从该节点收到的；以及

（e） 如果该特定的文档请求报文不能通过将本地高速缓存副本返回给客户，在通往本地服务器的路径上为特定文档请求报文确定通向第二毗邻节点的路径，并确定该第二毗邻节点的合法身份。

9. 根据权利要求 8 的方法，其中储存本地高速缓存副本的步骤进一步包括如下步骤：

（f）确定第一毗邻节点的节点状态参数和本地节点的节点状态参数，并且在第一毗邻节点和本地节点的状态参数彼此相差预定量时将至少一份本地高速缓存副本传送给第一毗邻节点以便储存在第一毗邻节点位置。

10. 根据权利要求 9 的方法，其中节点状态参数选自：请求的实现率、请求实现率的变化、文档大小、文档提请响应时间、通信路径装载、高速缓存服务器装载、或本地服务器的工作状态。

11. 根据权利要求 8 的方法，其中储存本地高速缓存副本的步骤进一步包括如下步骤：

（f） 确定第二毗邻节点的节点状态参数和本地节点的节点状态参数，并且在第二毗邻节点和本地节点的状态参数相差预定量时至少删除一份本地高速缓存副本。

12. 根据权利要求 11 的方法，其中节点状态参数选自请求的实现率、请求实现率的变化、文档大小、文档提请响应时间、通信路径装载、高速缓存服务器装载、或本地服务器的工作状态。

13. 根据权利要求 8 的方法，其中储存本地高速缓存副本的步骤进一步包括如下步骤：

（f） 确定第二毗邻节点的节点状态参数和本地节点的节点状态参数，并且在第二毗邻节点和本地节点的状态参数相差预定量时，在滤波步骤中通过向客户提供本地高速缓存副本实现降低文档请求的比例。

14. 根据权利要求 1 的方法，其中滤波步骤进一步包括如下步骤：

（c） 如果本地节点的节点状态参数不同于预定量，那么将本地高速缓存副本返回给客户。

15. 根据权利要求 1 的方法在选定的中间节点进一步包括如下步骤:

(c) 纪录与通过向客户提供一份本地高速缓存副本实现的请求报文数有关的装载统计结果。

16. 根据权利要求 1 的方法在选定的中间节点进一步包括如下步骤:

(c) 纪录与收到并实现对特定文档的请求报文数有关的请求报文统计结果。

17. 根据权利要求 1 的方法在选定的中间节点进一步包括如下步骤:

(c) 纪录与收到的对储存在特定服务器中的文档的请求报文数有关的响应时间统计结果。

18. 根据权利要求 1 的方法，其中文档请求报文是在众多通信路径上的诸中间节点接收的，并且该方法在选定的中间节点进一步包括如下步骤:

(c) 纪录请求报文统计结果以便跟踪从特定路径接收的文档请求报文。

19. 根据权利要求 1 的方法，其中文档请求报文是在众多通信路径上的诸中间节点接收的，并且该方法借助选自全面网络请求和区域网络请求的判据或借助客户的特殊请求进一步预测通信路径的用途。

20. 根据权利要求 1 的方法，其中滤波文档请求报文的步骤进一步包括如下步骤：

（c） 在发送文档请求报文之前鉴别来自客户的文档请求报文所抵达的节点。

21. 根据权利要求 1 的方法，其中滤波文档请求报文的步骤进一步包括如下步骤：

（c） 在将本地高速缓存副本返回给客户之前鉴别高速缓存文档的源节点。

22. 根据权利要求 1 的方法，其中服务器也储存运行文档的程序，并且在中间节点步骤(b)还包括如下步骤：

（c） 在从服务器获得与被请求的文档副本有关的某些选定程序时储存那些选定程序的本地高速缓存副本。

23. 根据权利要求 22 的方法，该方法还另外包括如下步骤：

（d） 在中间节点按来自客户的要求执行选定程序的本地高速缓存副本。

24. 根据权利要求 22 的方法，该方法还另外包括如下步骤：

（d） 保留在中间节点与客户之间的交互式通话，以致万一服务器已实现文档请求报文，中间节点将作为服务器发挥作用。

25. 根据权利要求 1 的方法，其中文档请求报文的滤波步骤还另外包括如下步骤：

（c） 将选定的程序施加给本地高速缓存副本，以便在将施加程序的结果返回客户之前获得该结果。

26. 根据权利要求 1 的方法，其中所述文档选自多媒体文档、程序、数据库、压缩数据或加密数据。

27. 根据权利要求 8 的方法，其中所述多个中间节点是定位在服务器和客户之间的，并且具体文档是通过将它发送给取决于预期文档需求率的众多节点被放进网络的。

28. 根据权利要求 9 的方法，其中所述预定的请求实现率取决于选自预期的请求实现率、请求实现率的预期变化、文档大小、预期的文档提取响应时间、预期的通信路径装载或预期的高速缓存服务器装载的文档属性。

29. 根据权利要求 1 的方法，该方法在选定的中间节点进一步包括如下步骤：

（c） 将指出本地高速缓存副本释放条件的数据与本地高速缓存副本一起储存起来；以及

（d） 只有在该条件得到满足时才将本地高速缓存副本返回给客户。

30. 根据权利要求 29 的方法，其中所述条件是释放该文档的时间。

31. 根据权利要求 1 的方法，其中储存高速缓存副本的步骤是基于预定的条件判据有选择地执行的。

32. 根据权利要求 31 的方法，其中预定的条件判据包括时钟指示的时间。

33. 根据权利要求 31 的方法，其中预定的条件判据是选自文档大小、所需的文档请求报文响应时间、文档请求报文率、文档请求报文率的变化率、服务器装载、通信路径装载或本地服务器的工作状态的至少一个判据。

34. 根据权利要求 1 的方法，其中储存高速缓存副本的步骤进一步包括储存比预定尺寸大的部分文档副本的步骤。

35. 根据权利要求 1 的方法，该方法进一步包括如下步骤：

（c） 基于预定的条件判据有选择地执行删除文档的高速缓存副本。

36. 根据权利要求 35 的方法，其中预定的条件判据包括时钟指示的时间。

37. 根据权利要求 35 的方法，其中预定的条件判据是选自文档大小、所需的文档请求报文响应时间、文档请求报文率、文档请求报文率的变化率、服务器装载或通信路径装载的至少一个判据。

38. 根据权利要求 1 的方法，其中储存高速缓存副本的步骤进一步包括储存比预定尺寸大的部分文档副本的步骤。

39. 一种提供适合在计算机网络中操作的逻辑通信网络的方法，在该网络中计算机通过众多路径在众多节点相互连接，该逻辑通信网络允许在第一网络节点的计算机安全地与在第二网络节点的第二计算机通信，该方法包括如下步骤：

（a） 沿着在第一节点和第一中间节点之间的路径提供安全的通信路径作为在第一节点和第二节点之间通信路径中的第一跃距；

（b） 在第二节点和最后一个中间节点之间提供安全的通信路径作为在第一节点和第二节点之间通信路径中的最后一个跃距；以及

（c） 将密钥提供给第一和最后一个节点，并且第一中间节点在将资料发送给路径中另一个中间节点之前用该密钥给该资料加密，而最后一个节点在将该资料发送给第二中间节点之前用该密钥给该资料解密，从而在第一和最后一个节点之间提供安全的通信路径。

40. 一种提供适合在计算机网络中操作的通信网络的方法，其中计算机通过众多路径在众多节点相互连接，该通信网络允许在第一网络节点的计算机与位于第二网络节点的第二计算机通信，该方法包括如下步骤：

（a） 沿着在第一节点和第一中间节点之间的路径提供通信路径作为在第一节点和第二节点之间通信路径中的第一跃距，所述通信路径包括众多路径；

（b） 沿着在第二节点和最后一个中间节点之间的路径提供通信路径作为在第一和第二节点之间的路径中的最后一个跃距；以及

（c） 将压缩和解压缩功能提供给第一中间节点和最后一个中间节点，并且第一中间节点在将资料发送给路径中另一个中间节点之前利用压缩功能压缩该资料，而最后一个中间节点在将该资料发送给第二中间节点之前利用解压缩功能给该资料解压缩，从而在第一和第二中间节点之间提供压缩带宽的通信路径。

41. 根据权利要求 1 的方法，该方法进一步包括如下步骤：

（c） 在服务器处，将一个文档截止时间标记附着到该文档上；

（d） 在中间节点处，检验附着在高速缓存文档上的时间标记，以确定该文档是否已经过期；

（e） 如果该高速缓存文档已经过期，则删除或刷新它。

42. 根据权利要求 1 的方法，该方法进一步包括如下步骤：

（c） 在服务器处，将一个文档修改时间标记附着到该文档上；

（d） 在中间节点处，评估该文档的修改率，并且如果文档修改率大于请求率某个预定量，则删除该高速缓存副本；如果修改率小于请求率某个预定量，则请求更新后的高速缓存副本。

43. 在包含众多利用分层通信协议进行网上通信的计算机的系统中，一些在网络节点上的计算机作为服务器以文档形式储存资料，而另一些计算机作为客户在应用层上将文档请求报文发送给诸服务器，其中文档请求报文是对储存在服务器处的文档的请求，通过透明的代理实现文档请求报文的方法包括下述步骤：

（a） 将文档的本地高速缓存副本储存在该网络中的众多中间节点位置；以及

（b） 在响应其中一个具体客户时，产生倾向于发送给某个特定服务器的特定应用层文档请求报文，在某个中间节点位置通过监听文档请求报文并将一份本地高速缓存副本返回客户所在的应用层，借此来实现特定应用层的文档请求报文，以致应用层请求报文在中间节点被监听而且不与服务器上的应用层建立网络连接。

44. 根据权利要求 43 的方法，其中中间节点包括高速缓存服务器和路由器，并且该方法在路由器处进一步包括如下步骤：

（c） 确认待监听的文档请求报文，并且提取这样的报文供高速缓存服务器处理。

45. 根据权利要求 1 的方法，该方法进一步包括如下步骤：

（c） 同时一起预取有关的文档，其中有关文档是那些近乎连续地最频繁地被请求的文档。

46. 根据权利要求 1 的方法，该方法进一步包括如下步骤：

在客户处，

（c） 产生一个连接请求报文，该请求报文请求在客户与服务器之间建立一条通信路径；以及

在中间节点处，

（d） 在收到来自客户的连接请求报文时，等待接收文档请求报文，然后将连接请求报文和文档请求报文一起发送给路径中的下一个中间节点。

47. 根据权利要求 46 的方法，该方法进一步包括如下步骤：

在高速缓存该请求报文所指定文档的中间节点，

（e） 对客户肯定应答连接请求，并将被请求的文档返回给客户。

48. 根据权利要求 1 的方法，该方法进一步包括如下步骤：

（c） 操作某个高速缓存服务器，该服务器这样选择准备存入本地高速缓存器的文档和准备从该本地高速缓存器中删除的文档，以致该高速缓存服务器选出受到最频繁请求的文档复制在位于该路径中某个节点的毗邻高速缓存服务器中，并且选出最不经常受到请求的文档从其本身的存储器中删掉。

49. 根据权利要求 1 的方法，其中每个中间节点包括与其相关的资源管理器，并且该资源管理器完成如下步骤：

（c） 分配用于缓冲进出报文的通信缓冲器的使用以有利于被指定为比较重要的报文交流。

50. 根据权利要求 49 的方法，该方法包括如下步骤：

（d） 基于选自文档大小、文档类型、传输方向或优先权的内容属性分配通信缓冲器和通信路径带宽。

51. 根据权利要求 1 的方法，该方法进一步包括如下步骤：

在中间节点，

（c） 将该中间节点处的节点状态参数与毗邻服务器处的节点状态参数进行比较；以及

（d） 如果该毗邻服务器和该中间节点的节点状态参数相差的预定量比预定的时间周期多，则意味着该毗邻节点没有高速缓存该中间节点正在请求的一份或多份文档。

52. 根据权利要求 51 的方法，其中节点状态参数选自请求实现率、请求实现率的变化、文档大小、文档提取响应时间、通信路径装载、本地服务器的可操作性或高速缓存服务器装载。

53. 在包含众多进行网上通信的计算机的系统中，一些在网络节点上的计算机作为服务器以文档形式储存资料，而另一些计算机作为发送文档请求报文的客户，其中文档请求报文是针对储存在那些服务器处的文档的请求，为了装载而确定实现文档请求报文的坐标位置的方法包括下述步骤：

（a） 将文档的本地高速缓存副本储存在该网络中的众多中间节点位置，并且在响应其中一个具体客户时，产生一个倾向于发送给某个特定服务器的特定的文档请求报文；

（b） 在某个中间节点位置通过返回一份对应于在特定文档请求报文中所规定的文档的本地高速缓存副本来实现该特定的文档请求报文，以致在中间节点位置的应用层监听请求报文，而服务器上的应用层不接收该文档请求报文。

54. 根据权利要求 53 的方法，该方法进一步包括如下步骤：

（c） 在这些中间节点位置，协调在这些中间节点位置当中储存高速缓存副本和分离文档复制的步骤，以便在中间节点高速缓存服务器或本地服务器上分配文档请求报文装载。

55. 根据权利要求 53 的方法，该方法进一步包括如下步骤：

（c） 在这些中间节点位置，协调在这些中间节点位置当中储存高速缓存副本和分离文档复制的步骤，以便分配通信路径装载。

56. 根据权利要求 53 的方法，其中为了减少文档装载而协调在这些中间节点位置当中储存高速缓存副本的步骤包括如下步骤：

（c） 在毗邻节点位置，复制一些文档。

57. 根据权利要求 53 的方法，其中在诸中间节点位置储存文档副本的步骤进一步包括如下步骤：

（c） 基于预定的判据具体地使这些副本可用于或不可用于实现文档请求。

58. 根据权利要求 57 的方法，其中预定的判据是时钟指示的时间。

59. 根据权利要求 53 的方法，其中在诸中间服务器之一实现特定文档请求报文的步骤进一步包括如下步骤：

（c） 开放叶片节点和服务器之间的通信连接，该叶片节点是最初接收来自客户的文档请求报文的中间节点位置之一；以及

（d） 将文档请求和通信连接传送给另一个中间节点，该中间节点在网络中的位置比叶片节点更接近该服务器。

60. 根据权利要求 53 的方法，其中在诸中间服务器之一实现特定文档请求报文的步骤进一步包括如下步骤：

（c） 开放叶片节点和服务器之间的通信连接，该叶片节点是最初接收来自客户的文档请求报文的中间节点位置之一；以及

（d） 将该文档请求转移到另一个中间节点，该中间节点在网络中的位置比该叶片节点更接近该服务器。

61. 根据权利要求 53 的方法，其中高速缓存服务器进一步完成如下步骤：

（c） 依据客户的意见充当本地服务器的通信代理。

62. 根据权利要求 53 的方法，其中实现文档请求报文的步骤还控制对文档的访问，并且该方法包括如下步骤：

（c） 基于文档请求报文中的统一资源定位符（URL）字段完成文档请求报文的滤波。

63. 根据权利要求 53 的方法，其中实现文档请求报文的步骤还控制对文档的访问，并且该方法包括如下步骤：

（c） 基于文档请求报文中的鉴别字段完成文档请求报文的滤波。

64. 根据权利要求 53 的方法，其中将文档的高速缓存副本返回给客户的步骤进一步包括如下步骤：

（c） 对该文档的病毒扫描。

65. 根据权利要求 53 的方法，其中将文档的高速缓存副本返回给客户的步骤进一步包括如下步骤：

（c） 对该文档的代码认证。

66. 根据权利要求 53 的方法，其中将文档的高速缓存副本返回给客户的步骤进一步包括如下步骤：

（c） 该文档的解码。

67. 根据权利要求 53 的方法，其中将文档的高速缓存副本返回给客户的步骤进一步包括如下步骤：

（c） 基于客户鉴别和本地服务器请求，控制对该文档的高速缓存副本的访问。

68. 一种在计算机网络中实现对资料的请求的方法，其中至少有一些计算机作为本地服务器用于储存资料，并且至少有一些计算机作为客户将请求报文发送给这些本地服务器，该方法包括如下步骤：

（a） 借助将所述资料的副本储存在众多在网络中充当高速缓存服务器的计算机中，在网络上分布该资料的高速缓存副本；

（b） 确定请求报文经过网络中众多中间计算机从客户至服务器的路由，其中至少有一些中间计算机充当高速缓存服务器，请求报文始于其中一个具体客户，以便从其中一个特定服务器获得资料；以及

（c） 在这些请求报文途经这些服务器时透明地处理它们，以致那些能由高速缓存服务器代替本地服务器提供服务的请求报文就由高速缓存服务器以对客户和本地服务器透明的方式提供服务。

69. 根据权利要求 59 的方法，该方法包括如下步骤：

（d） 对涉及高速缓存服务器为请求报文服务的预定判据的响应，使高速缓存副本自动地在网络中的诸高速缓存服务器当中移动。

70. 根据权利要求 1 的方法，其中多媒体文档包含选自文本、图片、声音、电视节目或其它数据的数字化资料。

71. 根据权利要求 1 的方法，其中网络包括无线网络。

72. 根据权利要求 1 的方法，其中诸中间节点还完成如下步骤：

确定特定的通信实体是否已经失效，如果失效，通知另一个网络实体。

73. 根据权利要求 72 的方法，其中所述通信实体是路由器、通信路径或通信路径之一。

74. 根据权利要求 72 的方法，其中其它网络实体是另一个中间节点、网络管理器或其它联网的计算机系统之一。

75. 根据权利要求 57 的方法，其中预定判据是客户发出的通知。

76. 根据权利要求 57 的方法，其中预定判据是本地服务器发出的通知。

77. 根据权利要求 2 的方法，该方法包括如下步骤：

在选定的中间节点，

（d） 确定接收特定的文档请求报文的第一毗邻节点的合法身份；

（e） 确定在通向本地服务器的路径上特定的文档请求报文将途经的第二毗邻节点的合法身份；

（f）如果由于在该中间节点出现失败的或缓慢的处理不能由该中间节点令人满意地实现特定的文档请求报文，则变更路径以便绕过该中间节点。

说 明 书

分布式高速缓存、预取和复制的方法和系统

本发明的现有技术

计算机网络（例如因特网、专用的内联网、外联网和虚拟专用网）正在越来越多地用于各种各样的企图，包括储存和查找资料、通信、电子商务、文化娱乐和其它应用。在这些网络中一些被称为服务器的计算机被用于储存和提供资料。一种类型服务器被称为主服务器或本地服务器（home server），它们负责访问本文中通常称之为“文档”的资料，例如以各种计算机文件格式储存的数据和程序。尽管通常在因特网中文档主要包括文本和图片，但是，每份这样的文档实际上可以是包含数据结构的高度格式化计算机文件，它们是包括文本、表格、图象、声音、电影、动画片、计算机程序代码和/或许多其它类型的数字化资料的各种资料的资料库。

在网络中另一些计算机被称为客户，它们允许用户通过请求本地服务器在网上将文档副本发送给客户来访问该文档。为了使客户从本地服务器获得资料，每份文档通常有一个能够找到它的地址。例如，在因特网的组织中和在被称为超级文本传输协议（Hyper text Transfer Protocol（HTTP））之内，地址通常是一个被称为统一资源定位符（Uniform Resource Locators (URLs)）的字母数字串，它规定（A）本地服务器的地址，从该本地服务器以文档名或数字地址的形式获取资料；和（B）本地资料正文串，它识别客户请求的资料，它可以是文件名、搜索请求或其它标识符。

在用户为客户计算机规定 URL 之后，该 URL 的地址部分在网上被发送给指定的服务机构（例如域名服务机构（DNS）），以便获得如何与正确的本地服务器建立连接的指令。一旦与该服务器建立起连接，客户就能通过将本地资料正文串在网上直接发送给本地服务器查找所需的文档。然后，该服务器从其本地磁盘或存储器检索该文档并且将该文档在网上发送给客户。然后，终止该本地服务器与客户之间的网络连接。

计算机和网络工业的系统分析员和专家目前十分关心网上通信业务量变得如此繁重，以致有可能使用因特网的道路的真性质可能发生变化。具体地说，现在，许多人认为因特网过于缓慢并且不再是及时交换资料的可靠入口。

现在的瓶颈无疑是用户数量和复杂文档（例如被发送的多媒体文件）的数量呈指数增长的结果。将更多的带宽简单地添加到服务器与客户之间的物理连接上。但是，这只有依靠在遍布全世界的家庭和街道安装高带宽的互连硬件，如同轴电缆或光缆以及相应的调制解调器才能实现。

此外，单独增加带宽或许不能保证性能的改进。具体地说，大的多媒体文件（例如电视娱乐节目）仍然可能代替优先权较高的数据（例如公司的电子邮件）。遗憾的是只要不修改现有的网络通信协议带宽分配计划就难以实现。在因特网上使用的通信技术（被称为 TCP/IP）是简单而精巧的协议，它允许使用许多不同类型的计算机（例如苹果机、IBM 兼容机和 UNIX 工作站）的人们分享数据。尽管有一些有野心的协议试图拓宽 TCP/IP 协议使地址可以包括有关数据包内容的报文，但是这些协议在技术上是复杂的并且需要成千上万个计算机网络操作员之间的配合。因此，指望修改现有的 TCP/IP 协议或许是不现实的。

一些人采用的途经是承认因特网的用户迅速增长将继续超出服务器的容量和通信媒体的带宽容量。这些计划以下述假设为起点，即假设客户–服务器基本模型（其中客户直接与本地服务器相连）是浪费资源的，特别是需要将资料从一个本地服务器广泛分配给许多客户。确实有许多实例说明因特网服务器由于它们缺乏适应意外要求的能力而简单地遭受失败。

为了减轻对本地服务器的要求，可以使用大的中心文档高速缓存器。高速缓存器是一种试图减少许多客户为同一文档向某个本地服务器提出重复请求的浪费的尝试。通过监听平行的请求，高速缓存器可以用于向多个客户位置提供同一文档的副本。

依据客户的观点，与高速缓存器的相互作用是这样发生的，即对于用户它是透明的但又略微不同于网络报文的立场。这种差别是当请求的地址部分提交给域名服务（Domain Name Service, DNS）查找连接本地服务器所需的资料时 DNS 的程序是为返回高速缓存器的地址而设计的，而不是为了返回原本有效的本地服务器。

另外，作为客户代理的服务器节点可以发出探测报文以便寻找高速缓存副本。一旦在特定的网络节点找到高速缓存副本，就将该报文发送给那个节点。例如，在国家科学基金的赞助下，已经将文档高速缓存器配置在美国的不同位置，以便消除跨洋网络连接处的瓶颈。具体地说，位于西海岸的那些高速缓存器处理亚洲、大洋洲和南美国家对文档的请求，而位于东海岸的那些高速缓存器处理欧洲对文档的请求。另有一些高速缓存器处理对遍布美国的大众文档的请求。

但是，这种高速缓存技术不是必需的或通常达不到文档请求装载的最佳分布。具体地说，为了最有效地使用高速缓存器，域名服务机构或其它报文路由机构必须得到适当的改造，以便监听对大众性期望值高的文档的请求。因此，引入高速缓存副本将增加名称分辨的通信附加费，因为需要确定过往副本的位置。在名称服务机构成立时，它们必须寄存这些副本、传播分配文档需求的资料以及保证及时消除对已取消的高速缓存副本的纪录。通常，高速缓存搜索命令是固定的，和/或文档分配中的变化必须通过人员干预才能完成。

遗憾的是，请求方式中频繁而明显的变化可能迫使高速缓存副本的合法身份、位置、甚至数量都是高度暂时性的。最终需要更新高速缓存器目录意味着通常不能有效地大规模地复制它们，因此这可能使名称服务机构本身变成瓶颈。

另一种可能实现高速缓存的途经是改变客户/服务器相互作用协议，使客户能利用全面分配协议主动识别适当的高速缓存副本，例如通过按随机化的方向发出探测信号。且不说修改现有协议的复杂性和这种途经带来的报文费用，这种计划也将一种或多种路由往返延迟加到客户能感觉到的总的文档服务执行时间。

本发明的概述

本发明是一个透明的分布式自动高速缓存方案，该方案利用文档请求通过计算机网络从客户到具体本地服务器上的具体文档所遵从的路径自然形成路由图（即路由树）这一事实。

按照本发明，高速缓存服务器遍布该网络，以致文档请求如果能沿着该路由图在某个中间节点实现，它将由该中间节点

提供服务，将高速缓存文档返回到客户。因此，文档请求报文在它们到达本地服务器之前就得到响应。由于允许文档请求报文以与不存在高速缓存时相同的方式在从客户到本地服务器的上行方向上选择路由，所以，命名服务机构不需要改造。

为了能够以不脱离标准网络协议的方式为请求提供服务，高速缓存服务器在它相关的路由器中包括一个滤包器。该滤包器提取文档请求包，它很可能命中相关的高速缓存器。

高速缓存服务器还优选充当本地服务器的通信协议代理。这就是说，作为在中间节点位置实现文档请求报文的一部分，依据在使用的通信协议,适当的报文发送给该客户，以便欺骗该客户使之相信实际上已收到来自本地服务器的文档。

本发明还提供一种高速缓存服务器相互合作为客户请求报文提供服务的方式。具体地说，每个服务器都有基于其局部装载、在其毗邻的高速缓存器上装载、相邻通信路径装载和文档的大众性的高速缓存和废弃文档的能力。例如，每个服务器保留对其邻居装载的判断，并且将它自己的装载估计通知毗邻的高速缓存服务器。如果某个高速缓存服务器注意到它相对其任何邻居过载，它将停止装载，即将其一部分工作转移给它正在装载的邻居。为了这样做，高速缓存服务器还优选学习在扎根于给定本地服务器的路由图上其毗邻的上游节点（或母节点）和下游节点（或子节点）的标识符。

依据本发明的文档高速缓存系统的基本概念有若干优点。

第一，该途经不需要请求查找高速缓存器索引的地址，不需要更改文档请求地址、或不需要以别的方式探测网络中其它要素以确定高速缓存副本的位置。因此，高速缓存副本的位置

是沿着请求报文所遵从的自然路径偶然出现的。因此，客户不经历与网络中其它客户寻找适当的高速缓存副本相关的延迟或瓶颈。

此外，该系统作为一个整体允许文档的高速缓存副本在需要时通过网络传播，这依次化解了在高速缓存器处和沿着通信路径的瓶颈。

因为高速缓存副本总是放在比客户近的原始服务器处，所以还相应地减少了网络带宽消耗和响应时间。所以文档请求报文和文档本身每次受到请求时通常不需要在该服务器与客户之间全程传送。因此，总网络带宽保持不变、响应时间减少、而负载更趋于全球平衡。

因此，本发明不仅有助于缩小主服务器之间装载要求的差别而且减少了在网络通信资源上的装载，同时不需要对现有的网络协议进行任何修改。

此外，因为高速缓存副本是通过网络分配的，所以在高速缓存系统中没有一个预期的故障点，因而该系统是稳定而可靠的。

这项技术还是可伸缩的，从感觉上讲添加了更多的高速缓存服务器，客户和服务器两者都将体验到同样的好处。

本发明还可以用于在网络中实现额外的功能。这些功能来源于下述事实，即配置在与高速缓存服务器相关的路由器的滤包器不是仅仅简单地将对文件副本的请求转移到本地高速缓存器。

例如，通过高速缓存服务器收集大众性的统计结果是必要的，以便控制高速缓存哪些文档。因此，每个高速缓存服务器保留接受过多少次文档查询以及这些查询来自何处的痕迹，并且能够确定请求的总量和请求的频率。这种数据可以在中心位置收集（例如通过联网区域），因此文档出版商不仅有可能获得有关其出版物的命中率的统计结果还有可能了解网络中的这种选择来自何处。这不仅对于文档的大众性分析是重要的而且对于电子商务和跟踪知识产权也是重要的。

依据服务供应商的观点，高速缓存服务器通过作为装载的分销商还可以用于主宰大众性文档（如数据库、搜索引擎索引文件等）的复制品。换言之，数据库供应商可以将他们的文档安排在网络中，向希望访问它的客户推出数据闭塞器，无论在哪里这或许都是最佳安排。

一组安全特征也可以轻易地安装到高速缓存服务器上。

这类特征之一是鉴别请求报文和其它资料的来源。这是可能的，因为高速缓存服务器保留关于文档请求报文和文档本身的实际来源的资料。该机理也从所有节点都有滤包器和数据包路由器这一事实产生。滤包器和数据包路由器不仅保留如何将请求转移到高速缓存副本的痕迹，而且保留限制访问高速缓存副本的痕迹，例如通过鉴别对该资料的请求。本发明还使各种类型的文档能够在网络中分配。例如，本发明允许文档压缩，这是另一种节省带宽的形式，或加密，只要特定的服务器和客户节点是通过利用高速缓存服务器提交给通信的，例如在包含若干高速缓存服务器的网络中沿着客户与服务器之间的路径的第一节点和最后一个节点。

本发明还允许有效地高速缓存动态内容的文档。这种动态内容文档是下述类型的文档，即准备返回到客户的东西飞快地变化，通常是按程序指令变化。当本发明确认在它的高速缓存器中存在动态内容文档时，它不仅高速缓存用于该文档的数据而且允许提取并执行规定如何显示该文档或何时取数据的程序。

由于减少了服务器与客户之间的网络节点数量，本发明还改进了储存的连续媒体（如声音和视频数据文件）的传输。

附图的简要说明

为了更全面地理解本发明提供的优点，必须参照附图阅读下面的详细说明，其中：

图 1 描绘典型的计算机网络，说明针对单一文档的请求路径和依据本发明沿该路径分布的高速缓存服务器的位置；

图 2 是通信层示意图，说明资源管理器、协议代理和监听器怎样用于实现本发明；

图 3 说明在网络上使用文档请求的几个典型阶段；

图 4 是依据本发明定位在路由选择路径上的叶片服务器所完成的操作的流程图；

图 5 是非叶片型的中间高速缓存服务器所完成的操作的流程图；

图 6 是在路由选择路径上最后一个高速缓存服务器所完成的操作的流程图；

图 7 说明中间服务器监听文档请求报文。

图 8 更详细地说明监听文档请求报文。

图 9A 和图 9B 说明在最坏情况的客户请求方案中扩散可能怎样继续进行；以及

图 10 说明高速缓存服务器如何实现文档转换功能。

本发明优选实施方案的详细叙述

1. 绪论

现在参照图 1，计算机网络 10（例如因特网、内联网、专用内联网、虚拟专用网、局域网或其它类型的计算机网络）由许多包括客户计算机 12-1、12-2、12-3、…、12-4（统称客户 12）、路由器 14-1、14-2、…、14-10、高速缓存服务器 16-1、16-3、16-4、16-6、16-8 和 16-10 以及本地服务器 20 的网络实体组成。该网络可以使用任何不同类型的物理层信号传输媒体，例如公众电话网和专用电话网、微波链路、蜂窝电话和无线电话、卫星链路和其它类型的数据传输。

在图示的网络中，某些路由器 14 已经与它们的高速缓存服务器 16 相关联，而另一些路由器不与高速缓存服务器相关联。高速缓存服务器 16 包括各种类型呈高速缓冲存储器 18-1 形式的文档存储空间，其中高速缓冲存储器可以包括盘式存储器 18-1-1 和/或存储装置 18-1-2。

客户 12 和本地服务器 20 象在现有技术中那样为了分配通常呈“文档”形式的各种各样的资料而工作。这种文档实际上可以包含文本、图片、画面、声音、电视节目、计算机程序和

任何能按计算机文件或部分的计算机文件形式储存的各种类型的资料。此外，某些文档可以通过执行程序在请求访问它们时产生。

在下面的讨论中将假设网络 10 是因特网、资料是按超级文本传输协议（Hyper Text Transfer Protocol, HTTP）的格式编码的文档、以及文档请求报文是利用 TCP/IP 分层协议以统一资源定位符（URLs）的形式发送的。这对于理解其它类型的有线网络、开关网络和无线网络以及其它类型的协议（如 FTP、Gopher、SMTP、NNTP 等）可以使用本发明是有利的。此外，虽然本发明是按客户–服务器型通信模型的来龙去脉进行讨论的，但是应当理解本发明的原理可以等同地应用于同位体网络（peer-to-peer networks）。

例如，对具体文档的请求报文起源于诸客户计算机之一，如客户 12–1。该报文是客户 12 要本地服务器 20 发送当前储存在本地服务器 20 位置（比如在磁盘上）的文档副本的请求。该文档请求报文在其路径上按图示箭头的方向通过一个或多个路由器 14（如路由器 14–1、14–2、14–3）被传送到本地服务器 20。

在诸如因特网之类的网络中，文档请求报文在到达其目的地之前通过路由器 14 可以途经多达 15 个或更多个节点或“跃距”。来自其它客户（如 12–2、12–3 或 12–4）对同一文档的请求也通过其路径上不同的路由器 14 同时传送给本地服务器。

还应当理解，虽然在图 1 中路由器 14 和高速缓存服务器 16 被表示成分开的单元，但是它们的功能可以合并成一个单元。

模型对于理解多个客户对一份具体文档的请求如何在计算机网络 10 的路径上传送是有用的。模型是在文档请求报文通过

网络传送到本地服务器 20 时由这些报文的路由选择算法效应导致的结构 T。如图 1 所示，本地服务器 20 可以被看作是在结构 T 的根节点，文档请求以最远离根的叶节点级为起点，即在客户 12–1、12–2、…12–4。结构 T 还包括许多位于路由器 14 处的中间节点。

尽管将朝给定的本地服务器 20 传送客户请求的这组路径的结构 T 描绘成无环的数据定向图，但是，现在的解释并非得益于追加的复杂性。具体地说，在一份具体文档被看作是定位在唯一的一个本地服务器处时，该结构可以被称为有单一根的树。如果可以使用图解模型，根据那个条件我们在此使用术语“树”来描绘结构 T。考虑到这个模型，整个因特网可以被想象成树或图形的森林型结构，每棵树植根于不同的本地服务器 20，这些本地服务器有责任提供某组文档的特许的永久性副本。

按照本发明，文档副本在网络中定位在高速缓存服务器 16 处。按照本发明，将高速缓存副本的布局（以及由此引起的装载扩散）约束在树结构 T 中的节点上。这避免需要客户通过直接接触本地服务器 20 或命名服务机构（如域名服务机构（DNS））查找高速缓存副本的位置，或通过探查网络搜索适当的高速缓存副本。

本发明还假设高速缓存服务器 16 就在文档请求报文沿着树从客户 12 到本地服务器 20 自然途经的路径上，高速缓存服务器 16 协助卸除在本地服务器 20 处的过剩装载，或者协助扩散其它潜在的性能瓶颈，如通信链路本身。实际上，已经与高速缓存服务器 16 缔合的路由器 14 在文档请求报文包通过时就对它们进行检查并监听任何它有可能通过提供高速缓存文档予以实现的请求。

在最一般地介绍本发明的操作时，文档请求报文沿着树从其起源客户（如客户 12–3）朝本地服务器 20 上行。该文档请求报文沿着该路径所遇到的某些路由器（如路由器 14–7）没有本地高速缓存服务器 16，因此简单地将该文档请求报文向上发送给该树中的下一个 Findeathe 路由器（如路由器 14–6）。

但是，另一些路由器（如路由器 14–6）有本地高速缓存服务器 16–6，在这种情况下该文档请求报文将受到考核以确定它是否正在寻找定位在本地高速缓存存储器 18 中的文档。如果在高速缓存服务器 16–6 处遇到高速缓存副本，那么将那份副本返回给客户 12 并且不允许该请求报文继续在其路径上朝本地服务器 20 传送。但是，如果在特定的高速缓存服务器 16–6 处没有找到高速缓存副本，该请求报文将继续在通向本地服务器 20 的路径上朝下一个路由器 14–4 传送。

当请求报文包进入路由器 14 时，该路由器首先将该请求报文送到在本文中称之为滤波器代码（the filter code）的路由器软件部分。路由器 14 的滤波器代码在必要时通过本地高速缓存服务器 16 予以更新。该滤波器代码取决于包的类型、高速缓存器容量、在本地高速缓存服务器 16 处的装载、或在增援通信链路上的装载。该滤波器导致监听该数据包（以便试图由本地高速缓存服务器 16 提供服务）或将该数据包送回路由器 14 以便确定该数据包在其通往本地服务器 20 的路径上应当采取的下一个跃距(hop)。

理想的是高速缓存服务器 16 的配置是这样的，它致使不需要改变客户 12 或服务器 20 的操作模式。另一个目标是有一种设计，它能被逐渐地部署到网络 10 的现有底层结构中。这也要求任何新机理是与现有的通信协议兼容的。

为了实现这个目的，高速缓存服务器 16 和与之缔合的路由器 14 优选由 4 个功能构件块组成，如图 2 的层次图所示。在较高层的协议级（例如应用层），高速缓存服务器 16 包括 HTTP 代理 22 和资源管理器 24。在较低层（例如物理层），路由器通常提供数据包滤波器 26 和 IP 代理或监测器(snooper)28。

HTTP 代理 22 有责任履行标准的 HTTP 协议，包括存储器管理和保存访问高速缓存文档必不可少的索引结构。如果 HTTP 代理 22 收到请求，而文档又不在本地高速缓存器 18 中，那么，它将请求从本地服务器 20 获取该文档，并且在该文档到达时对该请求作出反应。配置 HTTP 代理 22 是为了在收到资源管理器 24 的指令时高速缓存一些文档。

尽管图 2 展示两种类型的代理，即在 HTTP 级和 IP 级的代理，但是应当理解该配置还可以包括在其它级的代理，包括应用层、IP 层、或在诸如传送层、话路层、表示层或其它层之间的某些其它层。

资源管理器履行协议使文档副本通过网络 10 扩散，下面将更详细地介绍。资源管理器 24 负责维护文档装载扩散机构使用的状态资料。资源管理器不仅可以通过编程管理在高速缓存服务器 16 本身上的装载，而且可以通过编程管理在诸路由器 14 相互连接所用的通信路径上的通信业务量。

为了实现这种装载管理或装载均衡技术，资源管理器 24 保存有关其毗邻高速缓存服务器 30 的合法身份和装载的资料。如何保存毗邻高速缓存服务器资料的细节下面将在第 3 节中讨论。

此外，资源管理器 24 辨别通过其毗邻的每个高速缓存服务器 30 收到的对高速缓存器 18 中每份文档的请求。这是通过保

存对从毗邻的每个高速缓存服务器 30 收到的请求独立命中次数来完成的。资源管理器 24 利用这种资料计算待扩散的过剩装载所占的分数。一旦确定这些分数，资源管理器通知其欠装载的邻居们 30 它们应当承担哪些待高速缓存的文档和何种请求率。这些分数还被用于产生待注入缔合路由器 14 的新滤波器代码。类似的过程由欠装载的邻居们 30 完成。如果有必要，在欠装载的邻居 20 处的资源管理器 24 通知与其缔合的 HTTP 代理 22 将新文档添加到它的高速缓存器 18 中。

资源管理器 24 的其它责任包括邻域发现、将装载资料传播到毗邻服务器 30 以及发现潜在的壁垒并从潜在的壁垒恢复到装载平衡。这些机构将在下面更详细地讨论。

路由器 14 在协助高速缓存服务器 16 实现高速缓存服务器和/或通信路径平衡目标方面起积极作用。这是通过允许资源管理器 24 将功能性以代码形式注入路由器 14，其中所述代码将提供滤波器 26 和监测器 28。具体地说，所有通过未编址的路由器 14 直接向本地服务器 20 传送的数据包都首先被送到监测器 28。监测器 28 检查数据包并确定其类型、目的地和被请求的文档。依据高速缓存服务器 16 的状态和数据包的类型，监测器 28 可能监听该数据包或者简单地沿着通向目的地的路径将该数据包朝通向本地服务器 20 的下一个跃距或路由器 14 发送。

为了确定被请求的文档是否在本地高速缓存服务器 16 处，监测器 28 对滤波器 26 提出质疑。如果滤波器 26 表明被请求的文档被高速缓存在本地并且可以提供服务，那么该数据包被监听并发送给资源管理器 24。否则，将该数据包继续传送到通向目的地本地服务器 20 的下一个跃距。

监测器 28 通常了解 TCP 协议和 TCP 数据包和 HTTP 数据包两者的结构。监测器 28 的另一种功能性是提取 HTTP 请求数据包的副本并且将它们上行传送到资源管理器 24。这个特征被用于协助资源管理器 24 发现其邻域和退出潜在的壁垒。

2. 处理 TCP / IP 网络中的 HTTP 文档请求

使用 HTTP 的网络 10 当前的配置取决于在客户 12 与服务器 20 之间提供可靠的端–端通信的分层 TCP/IP 协议。这种分层将正常处理请求报文分成 3 个步骤：建立连接（即呈{SYN}报文形式的 TCP 级三路信号交换）、呈{GET}形式的 HTTP 文档请求/重播、以及呈{FIN}形式的终止连接。

这种过程用图 3 描绘，其中客户 12 首先将带顺序号的{SYN}报文发送给本地服务器 20，而本地服务器 20 返回一带肯定应答{ACK}的{SYN}报文。然后，客户 12 对这个报文作出反应以{GET}报文形式发送包括所需文档的 URL 的文档请求报文。然后，由本地服务器 20 将文档传送给客户 12。在客户 12 返回肯定应答之后，服务器 20 和客户 12 通过交换{FIN}和{ACK}报文终止连接。

在这种环境中象前面解释的那样实际配置高速缓存服务器 16 的主要障碍是要求这些服务器识别客户 12 请求的文档。但是，如图 3 所示，URL 资料通常是在与本地服务器 20 已经建立 TCP/IP 连接之后才由 HTTP 客户 12 播出。因此，一种可能的解决办法是与本地服务器建立所有的这类连接并且在中间路由器 14 处具有监听全部{GET}数据包的监测器 28。即使这种方法或许能使本地服务器免除相当大量的装载，但是它仍然需要与这些文档缔合的 TCP 连接抵达本地服务器 20，这达不到试图为本地服务器卸载的目的反而导致失败。在高需求期间，这类请求

将相当于一大批对本地服务器 20 的{SYN}请求。此外，如果最初的{SYN}被监听，建立连接和拆除连接两者变得相当复杂。

为了克服这个障碍，在优选实施方案中，中间路由器 14 对 TCP 协议有某些了解。了解 TCP 的路由器 14 能够检测对所有 HTTP 服务器的 TCP 连接请求（即指向 HTTP 口的{SYN}数据包）并且有能力作为本地服务器 20 的代理或“欺骗”本地服务器 20。

这种功能性是通过监测器 28 实现的。具体地说，监测器 28 定位在通向本地服务器 20 的路径上的路由器 14 中，它检查由此通过的数据包，识别这些数据包，并且监听指向 HTTP 本地服务器 20 的任何{SYN}数据包。当{SYN}数据包不包含识别客户 12 企图请求哪些文档的资料时，监测器 28 作为该本地服务器 20 的代理，或欺骗该本地服务器 20，其方法是在客户 12 与高速缓存服务器 16 中的本地传输层之间建立连接并且注意客户 12 和该本地传输层两者最初使用的顺序号。

在建立连接之后，监测器 28 检查所有由此经过的数据包并且等待相应的{GET}请求。一旦收到{GET}请求，该监测器 28 将查询本地滤波器 26 和资源管理器 24 以确定被请求的文档是否被高速缓存。如果该文档被高速缓存，那么监测器 28 将 HTTP{GET}报文朝本地资源管理器 24 发送，等待资源管理器 24 为该请求提供服务，然后终止连接。在相反的情况下，被请求的文档未被高速缓存（即滤波器 28 或资源管理器 24 失败）。此时可以采取几种不同的方法为该文档请求提供服务。

在第一种方法中，移交 TCP 连接，其中监测器 28 关闭半个服务器（即那半个与客户 12 建立 TCP 连接的被欺骗的服务器）并且在本地服务器 20 的方向上以复合捎带（composite piggy

back){SYN+GET}报文的形式发送文档请求。此外，该{SYN+GET}报文包含将 TCP 连接的半个服务器移交给通向偶尔高速缓存被请求文档的本地服务器 20 的路径上的任何其它中间高速缓存服务器所需要的全部状态资料。

在第二种方法中，监测器可以作为 TCP 中继站，保持与客户 12 的连接并且将有关独立连接的{SYN+GET}报文在通向本地服务器 20 的路径上转发给下一个中间高速缓存服务器。

上述的移交方法可以用图 4 所示的流程图予以说明。这种方法是由一类特殊的高速缓存服务器 16 完成的，这类高速缓存服务器被称为叶片节点服务器 38，这些服务器在树 T 中位于级别极低的节点上，即首先监听来自客户 12 的{SYN}数据包的服务器。在图 1 所描绘的树 T 中叶片节点服务器 28 是高速缓存服务器 16–1、16–6 和 16–8。

如图所示，在图 4 的步骤 41 中，当叶节点服务器 38 接收{SYN}数据包时，本地服务器 20 通过直接在叶节点服务器 38 与客户 12 之间建立 TCP 连接被代理，或“被欺骗”。而后，该叶节点服务器 38 等待监听来自客户 12 的应答{GET}请求。

注意，这种欺骗是在下述意义上发生的，即在客户 12 与高速缓存服务器 16 之间交换的数据包是由监测器 28 在上述情景中修改过的。具体地说，正在为请求提供服务的高速缓存服务器 16 的网址被本地服务器 20 的网址取代并且在连接移交时高速缓存服务器 16 发出的字节顺序号不得不跟在叶片服务器 38 确定的顺序号后面。

返回到步骤 41，如果在步骤 42 中被请求的文档通过滤波器 28 传递高速缓存器查询测试，并且如果该资源管理器 22 检测该

文档是否在本地高速缓存器中和是否允许访问它，那么在步骤 45 中该文档请求将得到本地服务。在步骤 45 中，{GET}命令被送到资源管理器，然后该资源管理器转发被请求的文档。最后，通过再一次欺骗本地服务器和将{FIN}和{ACK}关闭报文发给客户将叶片服务器 38 与客户 12 之间的 TCP 连接关闭。

另一种情况是，如果在步骤 42 和 43 中出现失败，监测器 28 在本地服务器 20 的方向上发送{SYB+GET}数据包，然后关闭被欺骗的 TCP 连接的半个服务器，以使树上的另一个高速缓存服务器可以提供服务（如果可能的话）。图 4 中的步骤(d)和(e)可以是不同步的事件并且通常可以并行地发生。然后，在叶片服务器 38 处的监测器 28 不得不肯定应答接受{GET}请求。

在图 1 描绘的情景中，上游的非叶片中间节点 39 包括带高速缓存服务器 16–3、16–4 和 16–10 的那些节点。定位在非叶片节点 39 处的高速缓存服务器 16 需要以略微不同的方式处理{SYN+GET}数据包。具体地说，非叶片节点 39 中的监测器 28 仅仅在被请求的文档被高速缓存并且本地高速缓存服务器 16 有足够的能力为它提供服务的情况下才监听{SYN+GET}数据包。

图 5 是在非叶片中间节点 39 处完成这一过程的详细流程图。如图所示在步骤 51 中，为了为这种请求提供服务，监测器 28 在从自叶片节点 38 接收{SYN}时首先欺骗并且监听随后的{GET}请求。在随后的步骤 52 和 53 中，如前所述对滤波器 26 和资源管理器 24 进行查询，以确定{GET}请求是否能在本地处理。

如果该请求能在本地处理，步骤 55 通过建立 TCP 连接客户 12 的半个服务器、将{GET}发送给资源管理器 24、将文档返回给客户 12 以及关闭 TCP 连接来完成对本地服务器 20 的代理。

如果{GET}报文不能在本地处理，那么执行步骤 56，将{SYN+GET}传送给树 T 中的下一个节点。

在非叶片中间节点 39 中以不同方式处理{SYN+GET}数据包的主要优点是 TCP 连接仅仅向实际上具有被请求文档的特定中间节点 39 移交一次。另一个优点是{SYN+GET}包含移交连接所需要的全部状态资料（即没有补充状态资料要在叶片节点服务器 38 的监测器 28 与实际上高速缓存有被请求文档的中间节点 39 的监测器之间交换。）

以这种方式捎带{SYN+GET}数据包的缺点之一是本地服务器 20 将在没有适合它们处理这种数据包的传输协议的情况下不能适当地解释这种数据包。为了避免这个问题和保证在当前的网络协议下的可通用性，可以在本地服务器 20 解释所有的{SYN+GET}数据包之前通过要求将监测器 28 定位在最后的中间节点 39 来设置补充警戒。因此，在没有一个叶片节点服务器 38 或中间节点服务器 39 高速缓存被请求文档时，最后一个中间节点服务器 39 解释{SYN+GET}并将显式 HTTP{GET}请求转发给本地服务器 20。

为了适应这种情况，图 5 所示步骤 54 可以用图 6 所示种种处理代替。在这种情况下，在步骤 61 中，如果沿着路径 T 的下一个下游节点（即母节点）不是本地服务器 20，那么进入步骤 62，在这种场合将{SYN+GET}传送给 T 上的下一个中间节点。

但是，如果下一个节点是本地服务器 20，那么完成步骤 63。具体地说，监测器 28 建立 TCP 与客户 12 连接的半个服务器并且用对本地资源管理器 24 的{PROXY_GET}请求代替{SYN+GET}。资源管理器 24 将{PROXY_GET}请求转译成发送给本地服务器 20 的显式{GET}。然后，以同样的方式将本地服

务器 20 应答的应答转发给客户 12，仿佛该高速缓存服务器储存着被请求的文档。

到此为止已介绍的高速缓存技术的另一个缺点是沿着树 T 在具体客户 12 与本地服务器 20 之间的路径在叶片节点服务器 38 或中间节点服务器 39 决定为请求提供服务之后还可能变化。例如，在两个服务器节点之间失去网络连接或链路时可能发生这种情况。图 7 说明这种比较少见的情况，其中当高速缓存中间服务器 16b 处理来自客户 12 的文档请求时，在客户 12 与本地服务器 20 之间的路径将发生变化。所有由客户 12 发出的{ACK}现在将按照新路径通过新高速缓存服务器 16X 到达本地服务器 20。这引起高速缓存服务器 16b 超时和重新转发其数据包。

为了解决这个问题，在服务器 16b 处的监测器 28 可以始终监视数据包被重新转发的的次数。如果数据包被重新转发的次数超过预定的次数（比如 3 次），于是监测器 28 则假设客户 12 与本地服务器 20 之间的路径已经变化。具体地说，监测器 28 废弃与客户 12 的连接并且废弃与高速缓存服务器 16b 的连接，同时欺骗本地服务器 20 并将重组数据包（即{RST}数据包发送给客户 12。

在另一种方法中，分别向最接近客户 12 的叶片节点服务器 28 和最接近服务器 20 的最后数个跃距节点提供一条唯一可能通向客户 12 和服务器 20 的路由。这是通过在高速缓存服务器至高速缓存服务器的永久性 TCP 连接上有高速缓存服务器传送客户请求报文而不是简单地让这些请求报文遵从它们的自然路由来实现的。因此，这组作为一组适当地连接起来的 TCP 连接配置的连接自动地适应网络构型变化时在 IP 路由选择中发生的任何变化。

3. 邻域的发现

但是，为了资源装载平衡和本发明可能带来的其它优点，在相邻的高速缓存服务器的布局中所发生的任何变化还必须通过与邻近的高速缓存服务器通信进行检测。具体地说，每个参与上述计划的高速缓存服务器 16 都不得不确定哪些其它的服务器在其邻域。此外，在每个路由选择树 T 上，高速缓存服务器 16 不得不区分上游服务器（位于母节点上）和下游服务器（位于子节点上）。在树 T 中某个具体的节点 i 是节点 j 的母节点，如果 i 是从 j 到本地服务器 20 的路由上第一个高速缓存服务器，在这种情况下节点 j 也被称为节点 i 的子节点。

高速缓存服务器发现其邻域的方法之一需要来自其下级路由器 14 和监测器 28 的某种帮助。在选定的时刻，资源管理器 24 要求本地路由器 14 将邻域发现报文发送给在该路由器 14 保存的路由选择表中所有的目的地。

然后，这些邻域发现数据包被在该树中另一个有高速缓存服务器 16 的节点处给定的监测器监听。于是，进行拦截的高速缓存服务器 16 负责将应答信号发送给在那个发出邻域发现数据包的高速缓存服务器 16 处的资源管理器 24，以便告知它是母节点（即它比发出数据包的高速缓存服务器更接近本地服务器 20）和它所在树的合法身份。用于邻域发现数据包的目的地端口可以被赋予不同的端口编号，以便保证该目的地的本地服务器 20 不试图处理非拦截的邻域数据包。跃距计数字段也可以用于限制邻域发现数据包过分传送。

这种方法的主要缺点是它将以邻域发现数据包淹没网络。一种替代方法是使用文档请求报文数据包（即{SYN+GET}数据包），该数据包无论如何都经过每个高速缓存服务器 16 中的滤波器。

在这种方法中，每个文档请求报文包含识别前一个跃距的字段，该字段在上述情景中变成某个具体请求数据包所经过的最后一个高速缓存服务器 16 的标识符。

在请求经过路由器 12 时（即它未被监听），本地监测器 28 标明附加高速缓存服务器 16 的 IP 地址。当某个高速缓存服务器想要发现其邻域时，于是它指令其附加监测器 28 从请求数据包中提取最后观测的目的地和最后一个跃距地址，并且将这些资料向上传送给本地资源管理器 24。

如图 8 所示，典型的 HTTP{GET}报文遵从从客户 12 经过 A 到本地服务器 20 的路径并且被中间高速缓存器 16c 监听。在高速缓存服务器 16c 正在处理该请求时，在本地服务器 20 与客户 12 之间的路径发生变化，引起所有的肯定应答使用不同的路径。

在高速缓存服务器 16c 处的资源管理器 24 利用这个资料确定它在哪些路由选择树上以及在每个树上有哪些下游高速缓存服务器 16。一旦高速缓存服务器 16c 确定服务器 16b 是其在该树 T 上的下游子节点，高速缓存服务器 16c 不得不明确地通知高速缓存服务器 16b 它是其在 T 上的母节点。为了减少在不同单元（即监测器 28 和资源管理器 24）之间交换报文的次数，监测器 28 可以高速缓存许多数据包并且立刻将它们全部传送给资源管理器 24。

邻域资料被保持一段时间，即邻域发现信号出现(neighborhood discovery epochs)的预定次数（例如两次）。如果没有通过子节点高速缓存服务器 16b 收到请求，那么该子节点高速缓存服务器 16b 将从高速缓存服务器 16c 的邻域模型中除去。然后，母节点高速缓存服务器 16c 还通知子节点高速缓存服务器 16b 它这样做的意图。

高速缓存服务器 16 也有可能在通向本地服务器 20 的路由选择树上没有母节点监测器 28。在这种情况下，高速缓存服务器 16b 处的监测器 28 在本地服务器 20 的方向上发送邻域发现数据包。上游监测器（例如在服务器 16c 处）接收该数据包并通知 16b 在通向本地服务器 20 的路由选择树上它是其母节点。但是，如果在 16b 处的监测器 28 在通向本地服务器 20 的路由选择树上没有母节点（例如 16c），它更换在邻域发现数据包上的 16b 地址并且在本地服务器 20 的方向上发送该数据包。

这种邻域发现方案有许多优点。首先，为了开始运行不必建造完整的适合高速缓存协议的路由选择树 T。另一个优点是相互协作的高速缓存服务器 16 可以动态地发现和适应路由变化。最后，协议是按总体分布的，所以也是稳定的，不怕服务器出故障。

4. 装载平衡

不同于大多数其它的高速缓存方案，按照本发明的高速缓存方案要求将高速缓存副本在客户实际请求它们之前分配给高速缓存服务器 16。换言之，是在预先考虑未来的文档请求时而不是在对客户 12 的某个具体文档请求报文作出响应时在诸高速缓存服务器 16 当中移动文档。

就高速缓存服务器 16 以及使它们相互连接的通信路径而言，上述的文档高速缓存和邻域发现方案适用于许多不同类型的高速缓存器装载分配和/或装载平衡目的。在优选的实施方案中，这种装载分配规划试图通过利用扩散严格依赖本地资料的高速缓存基础算法避免引入随着高速缓存系统的规模迅速增长的内务操作。

具体地说，资源管理器 24 仅仅在路由选择树 T 中的上游节点（“母节点”）检测到某个装载较少的下游节点（“子节点”）或链路时才建立高速缓存副本，它可以通过给出一份其高速缓存文档的副本将一些它的文档服务装载移交给所述下游节点或链路。在此“装载”可以是许多不同的性能判据，例如请求实现率、客户响应时间或该服务器的繁忙时间百分比。

在相反方向上的不平衡使子节点删除一些它的高速缓存文档或用别的方法减少对希望为其服务的这些文档的请求率。

通常，按照某种尺度最经常被子节点请求的母节点文档是指向装载较少的子节点的文档。

类似地，当高速缓存服务器 16 必须选择从其高速缓存文档库 18 中删掉某些文档时，这些被删掉的文档将是那些通常最少被请求的文档。

更具体地说，在将装载移交给邻居时，高速缓存服务器 16 可以将文档分别压入或释放给子节点或母节点。通过在传统的装载扩散方法中不存在的扩散机理进行文档高速缓存有两个目的。这在需要高速缓存服务器确定哪些待复制文档以及什么样的文档请求率（即装载率）应当移交给欠装载的邻居 30 时是显而易见的。

第一个目的是确定高速缓存服务器如何选择准备传送给欠装载邻居 30 的文档。此刻的目的是提取网络中所有的高速缓存服务器 16 的最大容量。第二个目的是减少响应时间，其方法是全部通过创造最少数量的复制品使大众性文档比较接近客户而使大众性较差的文档远离客户。这些目的在同时考虑到高速缓存服务器 16 之间的通信路径装载时得以实现。

为了达到这些目的，位于节点 i 的过载的高速缓存服务器确定定位在子节点 j 处的最欠装载的高速缓存服务器，以致

$$L_j \le L_k, \forall k \in C_i$$

其中 L_i 是在特定的高速缓存服务器 i 处的装载；C_i 是这组所有的高速缓存服务器。过载的高速缓存服务器 i 将一小部分对最大众化的文档的请求压入子节点 j。特别是，如果文档 d 具有下式所表达的性质，它将从节点 i 压入节点 j:

$$P_{ji}(d) = \max_{g \in D_i} P_{ji}(g)$$

其中 D_i 是这组被高速缓存在节点 i 的文档。在其它方向上，如果满足下式，文档 d 将被节点 i 释放给节点 j:

$$P_{ji}(d) = \min_{g \in D_i} P_{ji}(g)$$

这个对策被称为最大–最小文档扩散，并且它将满足上述的头两个目的。

给定的高速缓存器可以被看成是一个准备以有助于最大限度地减轻装载的方式用文档填充的匣子。因此，与给定文档相关的装载越大，需要放在特定的高速缓存器文档的数量就越小。最大–最小文档扩散为了满足这第三个条件在高速缓存器中提供最少数量的文档。

现在已经确定了哪些文档待扩散，高速缓存服务器还必须计算出在它们将其装载移交给邻居时它们监听的请求率将发生怎样的变化。这个分数可以通过考虑高速缓存服务器如何压入一小部分文档强加的装载来决定。具体地说，假设节点 i 是节点 j 的母节点以及 i 相对 j 而言是过载的。此外，假设节点 i 希望

将由它提供服务的对文档 d 的大量请求（等于总请求量的 $R_{ij}(d)$）压入节点 j。如果总请求量由下式定义：

$$\beta_{ji}(d)\Delta_{ji}(d)$$

那么由节点 i 提供服务的对文档 d 的请求量将由下式给出：

$$P_{ii}(d)=\sum_{m\in C_i}\beta_{mi}(d)\Delta_{mi}(d)$$

并且在将 $R_{ij}(d)$份请求压入节点 j 之后节点 i 提供服务的请求量由下式给出：

$$P'_{ii}(d)=\sum_{m\in C_i}\beta_{mi}(d)\Delta_{mi}(d)-\beta_{ji}(d)\Delta_{ji}(d)+\beta'_{mi}(d)\Delta'_{mi}(d)$$

其中：

$$\Delta'_{ji}(d)=\Delta_{ji}(d)-R_{ij}(d)$$

而且$\Delta_{ji}(d)$代表在压入节点 j 生效之后未被节点 j 监听的请求量。被节点 i 监听的$\Delta'_{ji}(d)$的新分数用$\beta'_{ji}(d)$表示。还要注意：

$$R_{ij}(d)=\beta_{ji}(d)\Delta_{ji}(d)-\beta'_{ji}(d)\Delta'_{ji}(d)$$

利用$\Delta'_{ji}(d)$和$\beta'_{ji}(d)$的上述数值可以直接得出：

$$R_{ij}(d)=\beta_{ji}(d)\Delta_{ji}(d)-\beta'_{ji}(d)(\Delta_{ji}(d)-R_{ij}(d))$$

并且通过代数运算得出新的值：

$$\beta'_{ji}(d)=\frac{\beta_{ji}(d)\Delta_{ji}(d)-R_{ji}(d)}{\Delta_{ji}(d)-R_{ji}(d)}$$

在计算新分数$\beta'_{ji}(d)$之后，节点 i 更新与其缔合的滤波器并且将文档 d 的副本传输给节点 j。指示节点 j 将其监听的请求量增加 $R_{ij}(d)$。

一旦节点 j 收到这个资料，它计算：

$$\beta'_{mj}(d)=\frac{R_{ij}}{\Delta_{ji}}(1-\beta_{mj}(d))+\beta_{mj}(d),\forall m\in C_j$$

并且将这些最新资料反映在它的本地滤波代码中。为了完成分析，请注意经过节点 j 滤波的请求量增加 $R_{ij}(d)$。还已知：

$$P'_{jj}(d)=\sum_{m\in C_j}\beta'_{mj}(d)\Delta_{mj}(d)$$

并且通过将$\beta'_{mj}(d)$的新数值代入，得到：

$$P'_{jj}(d)=\sum_{m\in C_j}[\frac{R_{ij}(d)}{\Delta_{ji}(d)}(1-\beta_{mj}(d)+\beta_{mj}(d)]\Delta_{mj}(d)$$

并且通过代数运算得到：

$$\begin{aligned}P'_{jj}(d)&=\frac{R_{ij}(d)}{\Delta_{ji}(d)}\Delta_{ji}(d)+P_{jj}(d)\\&=Pj_j(d)+R_{jj}(d)\end{aligned}$$

从而完成分析。

对于大文档，本地高速缓存器 18 只包括一部分被请求的文档可能是有利的。在这个事件中，高速缓存服务器可以开始将本地高速缓存的文档部分提供给客户 12，同时请求本地服务器 20 发送该文档的其余部分。

其它判据可以包括文档大小、文档类型、传输方向、与具体文档相关的优先权。

为了有利于为具有较高的优先权而确定的报文通信业务量，高速缓存服务器 16 和路由器 12 还可以使用通信缓冲器。

相同类型的相关文档或在某些方面被高速缓存服务器 16 认为是近乎连续地一起被请求的相关文档可以在同一操作中在高速缓存服务器之间移动。

因此，每个高速缓存服务器 16 都基于其本地装载、其邻居的装载、通信路径装载和文档的属性（如大众性、大小、移交成本等）被赋予高速缓存和废弃一些文档的能力。具体地说，每个高速缓存服务器保留对其邻居处装载的估计和/或在不同的时刻将这种估计或有关其实际装载的“随笔（gossips）”传送给相邻的高速缓存服务器 16。如果高速缓存服务器 16 注意到它与其邻居相比在某些方面过载，它会如上所述将其一小部分预测未来的工作移交给其装载较少的子节点或母节点邻居。

因此，本发明还适用于装载分离，在这种场合就给定的本地服务器而言在路由选择树 T 中的同一路径上相邻的（“同胞”）节点可以分享对一份具体文档的请求装载。

上述的文档扩散算法可能在优化的装载分布中经历某些困难。具体地说，给定的高速缓存服务器 j 在它至少有两个子节点服务器 k、k’和一个母节点高速缓存服务器 i（例如，装载 L）时可能变成潜在的壁垒，以致在每个服务器上的装载满足下式：

$$L'_k \leq L_j \leq L_i \leq L_k$$

并且高速缓存服务器 j 不高速缓存任何其欠装载的子节点服务器 k 所需要的文件。

图 9(a)说明这种状况的一个实例。高速缓存系统由一个本地服务器 20（节点号 1）和三个中间服务器 39（节点号 2、3、4）组成。请求刚刚由叶片节点产生，具体地说，文档 d_1 和 d_2 正在被节点 4 请求，而 d_3 正在被节点 3 请求。该图表示高速缓存副本在诸具体节点处的方位和通过每个高速缓存副本得到服务的请求。

在这个实例中，节点 2 处的高速缓存服务器 16 是潜在的壁垒。它不能将任何装载扩散到节点 3 处的高速缓存服务器，因为它没有高速缓存 d_3。此外，节点 2 处的高速缓存服务器 16 由于承认存在问题将节点 1 处的高速缓存服务器隔离。

一种可能的优化装载分配将载荷均匀地分布在所有的四个节点当中，其中每个节点为若干请求提供服务。图 9(b)说明满足这个条件的文件高速缓存器和载荷的分布。

但是，基于扩散的方案可以更改，以致高速缓存服务器 16 能够检测到图 9 所示的这种令人不快的状态并退出这种状态。具体地说，高速缓存服务器 k 可以假设：如果在一段时间内（即定时信号出现(time epochs)次数多于预定次数（例如两次））k 相对 j 仍然保持欠装载状态且 j 未采取行动纠正这种欠装载状况，那么其母节点服务器 j 就是潜在的壁垒。在图 9(b)所示实例中，高速缓存服务器 k 对应于节点 3，而高速缓存服务器 j 与节点 2 相对应。在检测缺乏来自其母节点 k 的扩散装载时，子节点可以推断母节点没有高速缓存任何被植根于 k 的子树所请求的文档。一旦最终要它为这些文档的副本服务，于是服务器 k 可以正常地高速缓存它们。这项技术被称为隧道效应，因为在这种情况下即使有代表高装载壁垒的母节点 j 存在 k 也能够获得并高速缓存那些副本。

5. 其它特征

本发明在网络中可以用于实现附加的功能性。这些不同的功能是下述事实的直接结果，即定位在路由器 14 处的滤波器 26 可以做更多的工作，不仅仅是简单地将对文档副本的请求转交给本地高速缓存服务器 16。

例如，必不可少的是由高速缓存服务器 16 收集文档的大众性统计结果，以便控制哪些文档需要高速缓存以及将储存在哪个高速缓存服务器中。每个节点处的高速缓存服务器 16 保留对正在入网的具体文档的访问次数、纪录这些文档是从何处入网的、并且了解累积请求量。

这个数据可以在中心位置收集，并且由数组高速缓存服务器 16 包办，例如由定位在某个特定网络区域中的数组高速缓存服务器 16 包办。通过以这种方式收集数据，文档出版商不仅有可能获得有关特定客户对其材料的“命中率”的统计结果，而且有可能得知命中者来自网络 10 的哪个子区域。

本发明还在预测需求时使加速分布型文档能够进入网络 10。

高速缓存服务器 16 还可以依据服务供应商的观点通过充当装载分离器被用于数据库、搜索索引文件和其它大众性文档的主复制品。换言之，数据库供应商可以安排将他们的文档放入网络 10，向希望访问它的客户推出数据闭塞器，无论在哪里或许都是这些安排。

高速缓存服务器还可以实现请求报文和其它资料的来源鉴别。能够这样做是因为高速缓存服务器 16 自动地保留关于客户 12 的资料，它是具体文档请求报文的来源。如果客户 12 不在该

文档的授权请求者之列，该请求报文甚至在它达到本地服务器 20 之前就可能被高速缓存服务器终止。

选择性保安措施也可以提供。该机构起源于下述事实，即每个节点都有滤波器 26、资源管理器 24、高速缓存器库 18 和 HTTP 代理 22。滤波器 26 不仅可以跟踪对高速缓存副本的请求是如何转交给 HTTP 代理 22 的，而且可以限制对该高速缓存副本的访问，例如通过鉴别对该资料的请求。只要沿着从客户 12 至服务器 20 的路径第一和最后一个跃距是可以信任的（因为在高速缓存服务器 16 之间的链路可以轻易地被安排成可信任的链路），就可以在客户 12 和本地服务器 20 之间提供经由高速缓存服务器 16 的安全链路。

更一般地说，高速缓存服务器 16 可以在本地服务器 20 分配文档期间在若干个点变换这些文档并且随后移交给客户 12。这些变换可以允许和/或实现若干使高速缓存文档或程序增值的特征，包括文档和/或程序的压缩或加密特征。此外，任何已知的保安技术都可以作为变换在源头、终点或两者（在交易的情况下）予以应用。除了加密和解密之外，这些保安技术包括鉴别、授权（访问控制）、非拒付（non-repudiation）和完整性控制。这些保安技术可以使用通常需要钥匙的密码技术，并且非必选地使用通向高速缓存服务器的物理路径以强化鉴别、授权和非拒付。

在典型的应用中，这些变换将是匹配的（例如，对于每种加密都应当有匹配的解密）和单独使用的。但是，在不同的点按文档的有效期它们也可以是复合的。例如，文档在不同的点可以是压缩的、加密的、解密的和解压缩的。

图 10 说明一种有可能将变换应用于文档的方法。在这个实例中，在本地服务器 20 分配文档并将它移交给客户 12 时，迥然不同的变换可以发生在四个不同的时间：

在时间 T1，客户 12 发出请求；

在时间 T2，该请求被传送到服务器 20；

在时间 T3，服务器 20 发送应答信号；

在时间 T4，该应答信号被传送到客户 12。

注意，即使图 10 中的变换与通信的四个阶段相关联，这些变换也是由这些节点本身完成的。因此，T1 可以是高速缓存服务器 16–6 收到请求后对该请求所完成的变换，或者是客户 12 在发送请求之前对它完成的变换。在前一种情况中，高速缓存服务器 16–6 正在完成变换。在后一种情况中，客户 12 正在完成变换并且仅仅在挖掘穿越高速缓存服务器 16–6 的隧道。类似地，当该请求向本地服务器 20 传送时，变换 T2 由高速缓存服务器 16–7 或本地服务器 20 完成。在应答信号从本地服务器 20 发出并最终传送到客户 12 时，同样的替代方案可以应用于 T3 和 T4。

在按图 10 完成的变换具有重要作用的场合，这些变换（或者任何要求不同于该文档的输入的其它变换）处于保安措施的关键部位。如果高速缓存服务器 16 本身配备有保安措施，在需要时必须将密匙分配给高速缓存服务器 16。在这种情况下，可以通过在客户 12 与高速缓存服务器 16–6 之间以及在高速缓存服务器 16–7 与本地服务器 20 之间添加保密信道来提供端–端保安措施。如果客户 12 和本地服务器 20 配置了它们自己的端–端保安措施，那么高速缓存服务器 16 不防碍，并且有可能协助，将密匙分配给客户 12 和本地服务器 20。

图 10 表明高速缓存服务器 16–6 和 16–7 在该网络的边缘，借此在客户 12 与高速缓存服务器 16–6 之间以及在高速缓存服务器 16–7 与本地服务器 20 之间有直接的路径。但是，在此介绍的方案不防碍在变换由中间高速缓存服务器 16–9 和 16–10 完成时所用的更一般的布局。换言之，在请求、应答或两者期间变换可以在图 10 中的中间高速缓存服务器 16–9、16–10 处进行。

只要是通过使用高速缓存服务器 16 来责成具体的本地服务器 20 和客户 12 进行通信，该方案还允许用于透明的文档压缩形式，该形式是另一种节省带宽的形式。具体地说，沿着客户 12 与服务器 20 之间的路径在第一跃距之后，第一叶片节点服务器 16–6 可以以对客户 12 和本地服务器 20 两者都透明的的方式完成压缩或加密。

文档在传送前还可以进行病毒扫描或代码检查。

本发明还允许有效地高速缓存动态内容的文档，在这种场合哪些数据被实际返回给客户的最终结果在传输时发生变化。这种文档可以包括已经嵌入 Java 代码的文档。

高速缓存服务器 16 可以通过高速缓存该文档的数据以及在取回该数据时如何显式该文档的程序来完成这种动态内容文档的有效分配。如果该程序属于在客户 12 处能在显式该文档时正常执行的那种类型，那么程序和数据可以简单地转移给客户 12。

但是，如果该程序属于客户 12 期待它将在本地服务器 20 上运行的类型，那么高速缓存服务器 16 通过运行该程序完成追加水平的本地服务器欺骗。在这个实例中，必不可少的可能是高速缓存服务器 16 保持与客户 12 相互交流的状态以便完成欺骗本地服务器 20。

本发明还由于减少了本地服务器 20 与客户 12 之间的跃距数量对储存媒体（如声频数据文件和视频数据文件）的移交做了特有的改进。

尽管我们已经展示并介绍了几个按照本发明的实施方案，但是，应当理解，本发明并非仅限于此，而对于熟悉这项技术的人所知道的各种各样的变化和改进是敏感的，所以我们不希望局限于在此展示和介绍的细节，而倾向于覆盖所有原本熟悉这项技术者显而易见的变化和改进。

说 明 书 附 图

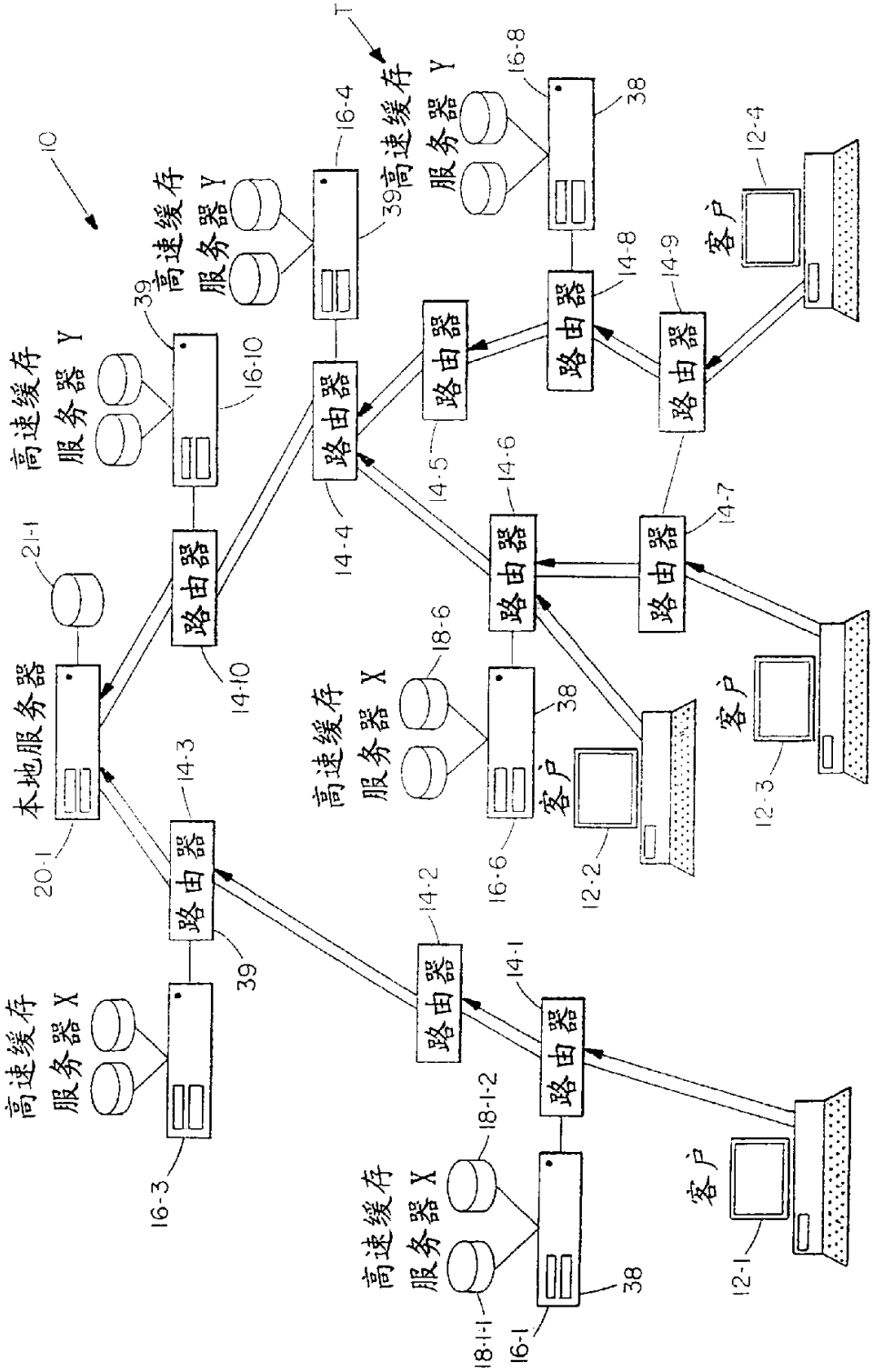

10
高速缓存
服务器 Y
39
16-10
高速缓存
服务器 Y
39
16-4
T
高速缓存
服务器 Y
16-8
38
客户
12-4
21-1
本地服务器
20-1
路由器
14-10
路由器
14-4
路由器
14-5
路由器
14-8
路由器
14-9
14-6
路由器
14-7
路由器
18-6
高速缓存
服务器 X
16-6
38
客户
12-2
客户
12-3
14-3
路由器
39
高速缓存
服务器 X
16-3
14-2
路由器
14-1
路由器
18-1-2
高速缓存
服务器 X
18-1-1
16-1
38
客户
12-1

图 1

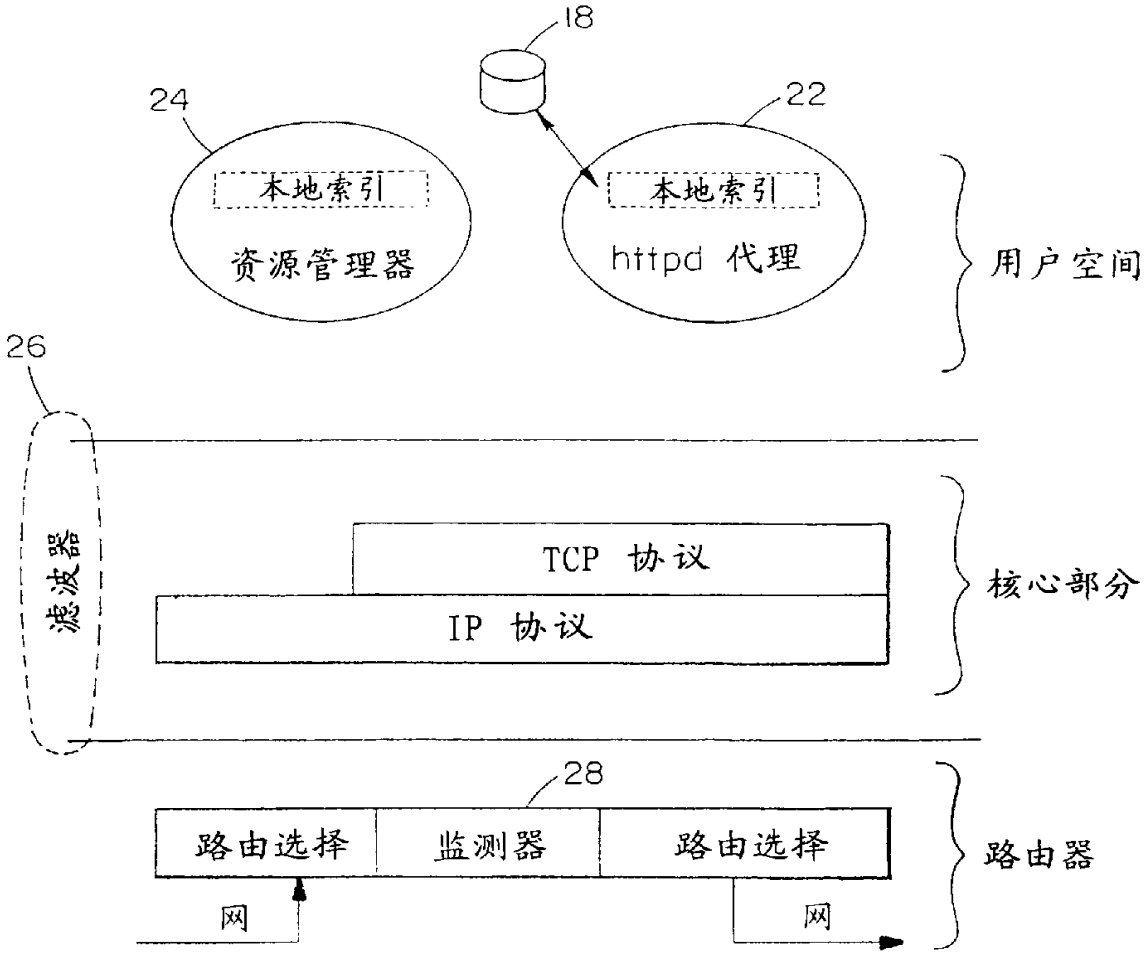

18
24
22
本地索引
本地索引
资源管理器
httpd 代理
用户空间
26
滤波器
TCP 协议
IP 协议
核心部分
28
路由选择
监测器
路由选择
网
网
路由器

图 2

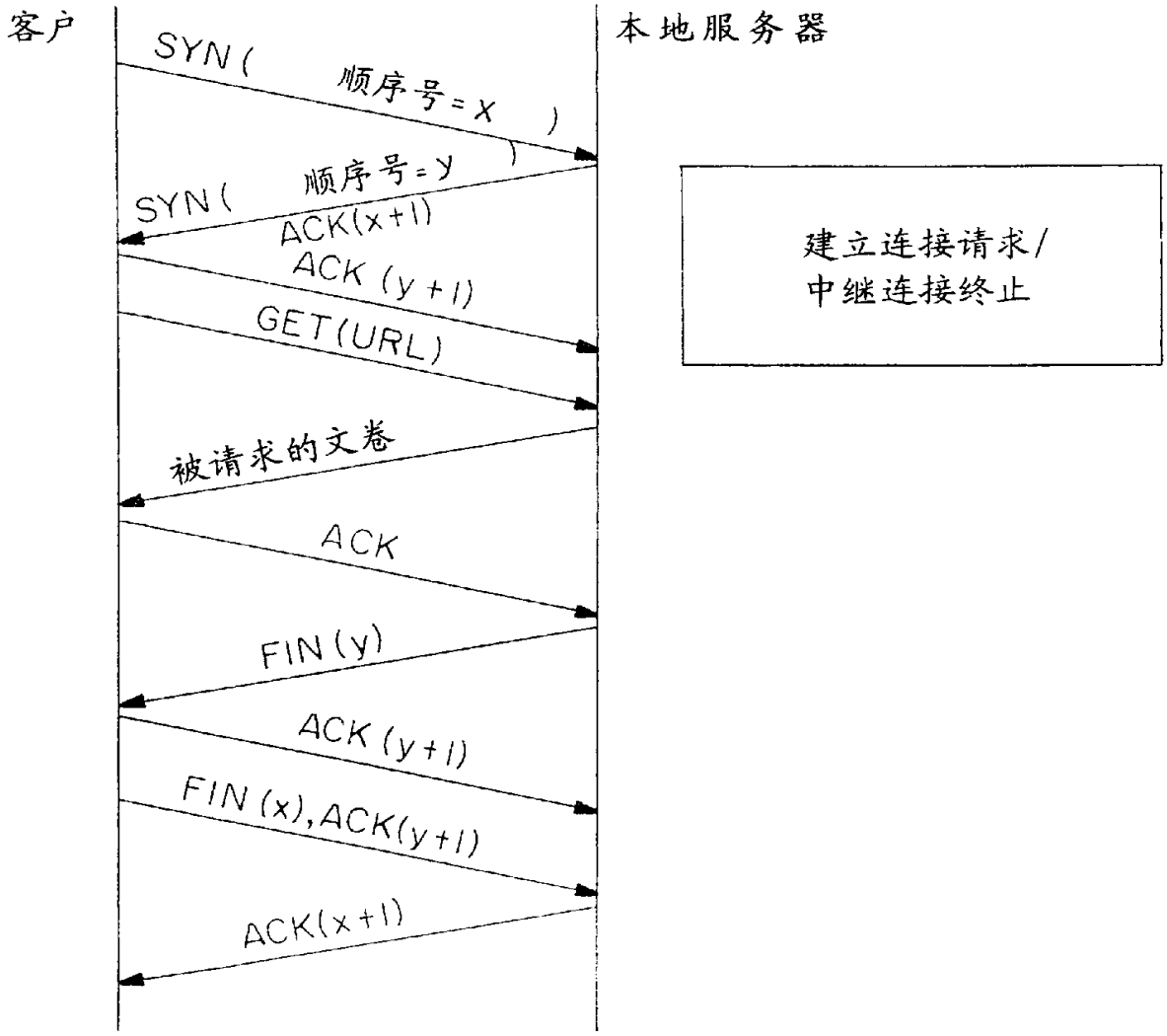

客户
本地服务器
SYN(顺序号=x)
SYN(顺序号=y)
ACK(x+1)
ACK (y+1)
GET(URL)
被请求的文卷
ACK
FIN(y)
ACK (y+1)
FIN (x), ACK(y+1)
ACK(x+1)
建立连接请求/
中继连接终止

图 3（现有技术）

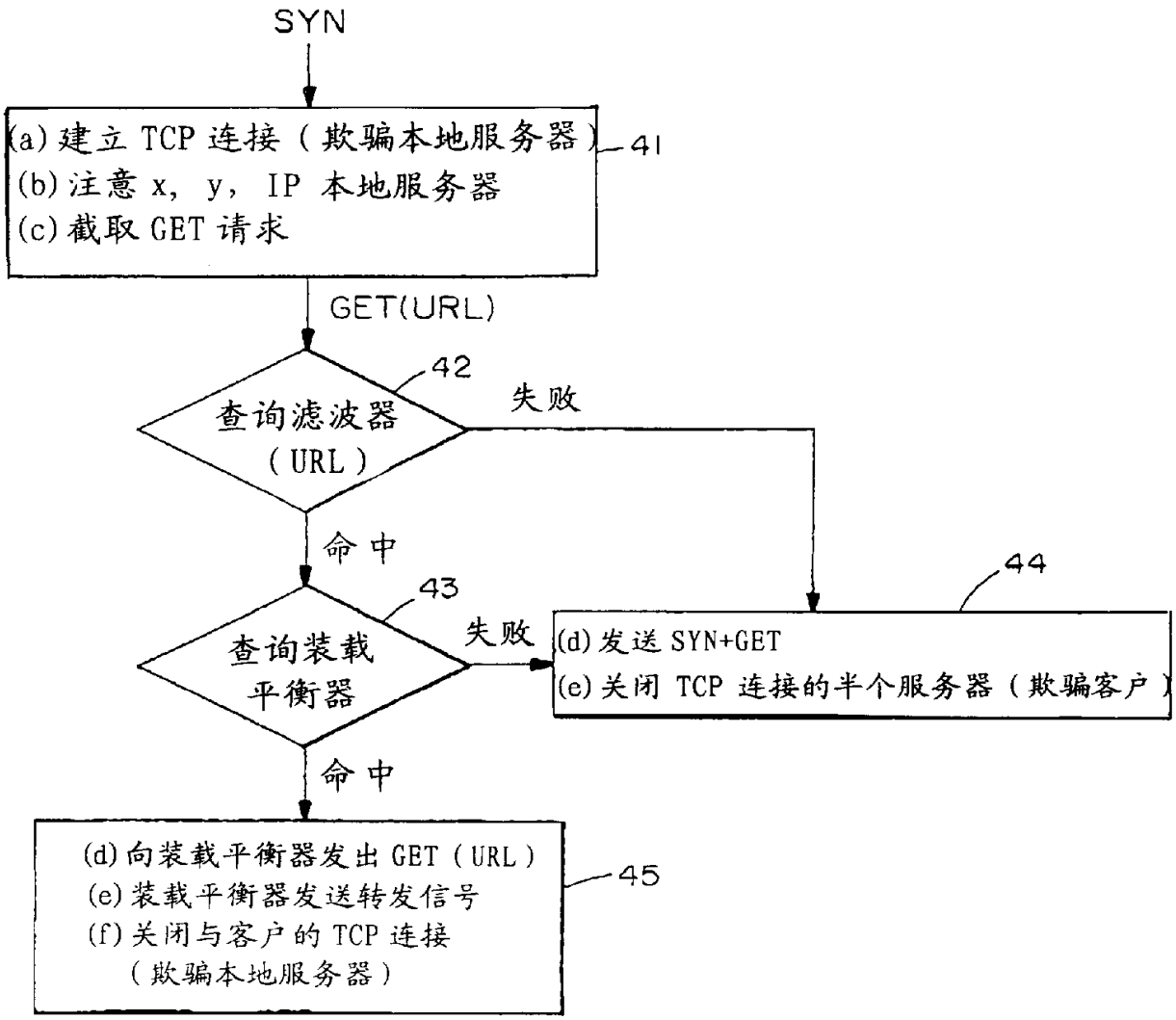

SYN
(a) 建立 TCP 连接（欺骗本地服务器）
(b) 注意 x, y, IP 本地服务器
(c) 截取 GET 请求
41
GET(URL)
42
查询滤波器
(URL)
失败
命中
43
查询装载
平衡器
失败
44
(d) 发送 SYN+GET
(e) 关闭 TCP 连接的半个服务器（欺骗客户）
命中
(d) 向装载平衡器发出 GET（URL）
(e) 装载平衡器发送转发信号
(f) 关闭与客户的 TCP 连接
（欺骗本地服务器）
45

图 4

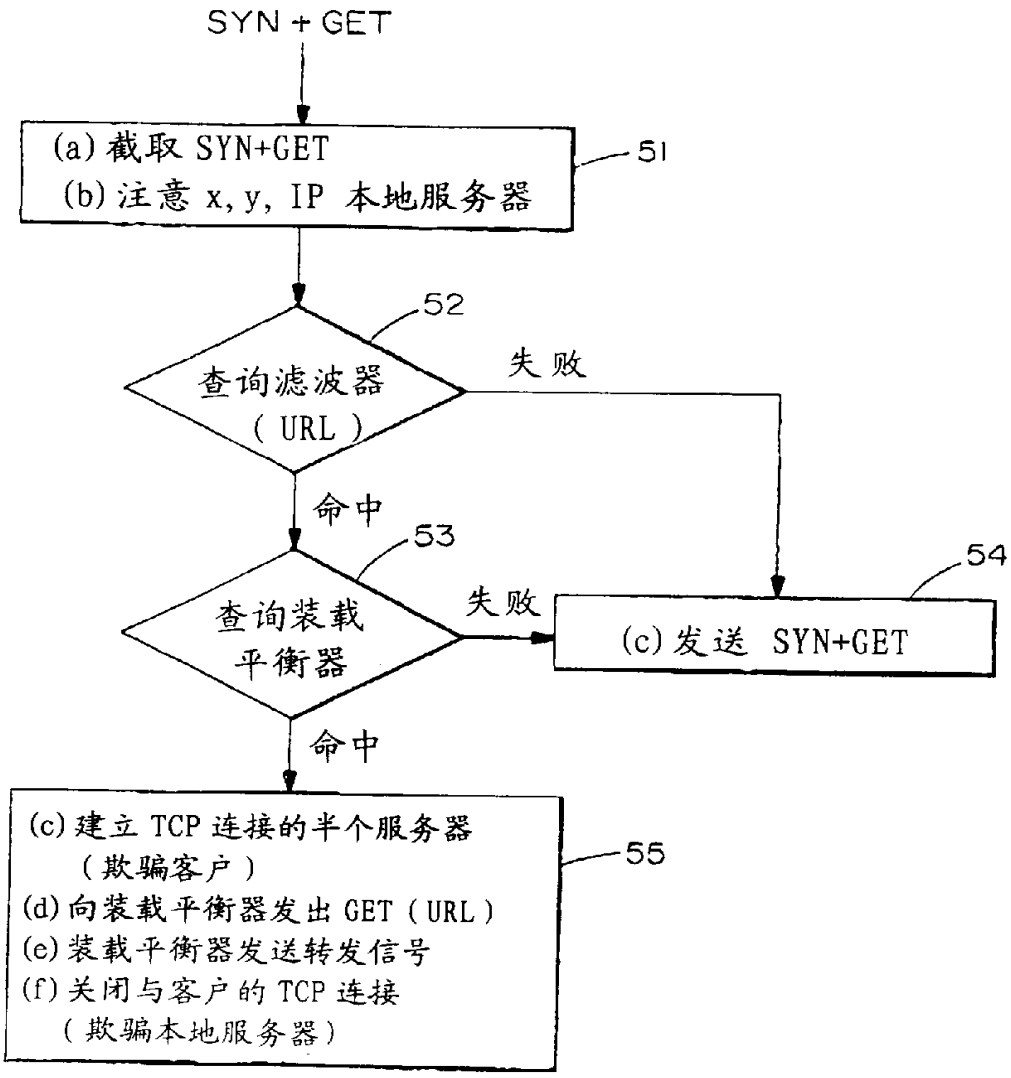

SYN + GET
(a) 截取 SYN+GET
(b) 注意 x, y, IP 本地服务器
51
52
查询滤波器
(URL)
失败
命中
53
查询装载
平衡器
失败
54
(c) 发送 SYN+GET
命中
(c) 建立 TCP 连接的半个服务器
(欺骗客户)
(d) 向装载平衡器发出 GET (URL)
(e) 装载平衡器发送转发信号
(f) 关闭与客户的 TCP 连接
(欺骗本地服务器)
55

图 5

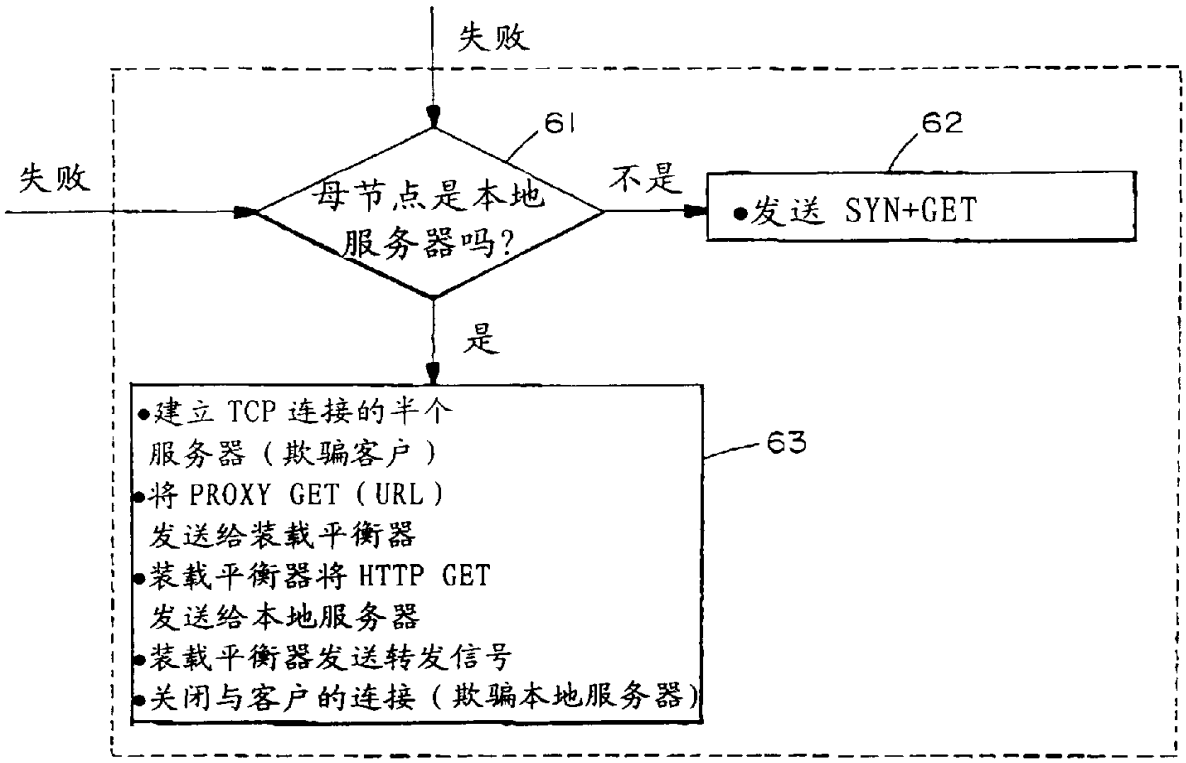

失败
61
母节点是本地服务器吗?
失败
不是
62
•发送 SYN+GET
是
•建立 TCP 连接的半个服务器（欺骗客户）
•将 PROXY GET（URL）发送给装载平衡器
•装载平衡器将 HTTP GET 发送给本地服务器
•装载平衡器发送转发信号
•关闭与客户的连接（欺骗本地服务器）
63

图 6

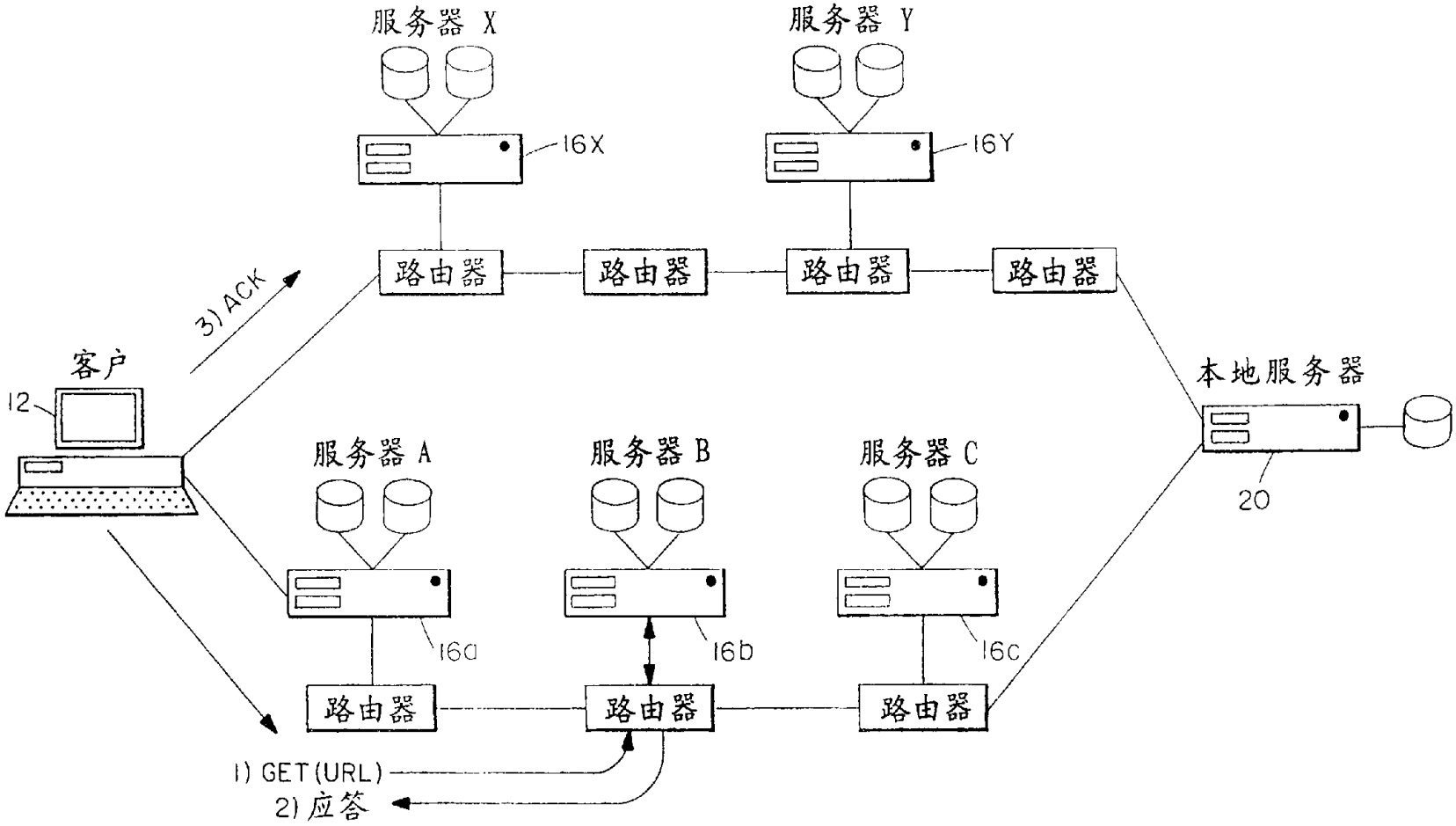

服务器 X
服务器 Y
16X
16Y
路由器
路由器
路由器
路由器
3) ACK
客户
12
本地服务器
20
服务器 A
服务器 B
服务器 C
16a
16b
16c
路由器
路由器
路由器
1) GET (URL)
2) 应答

图 7

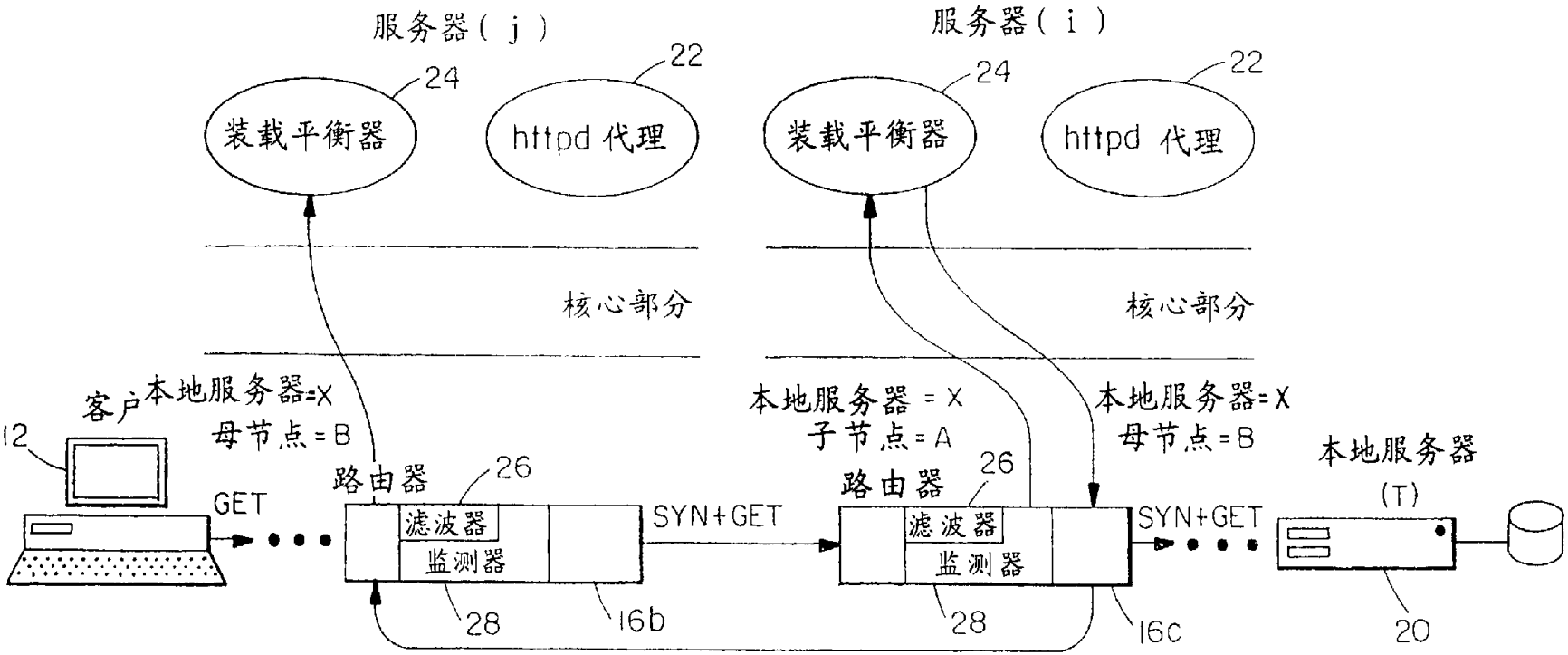

服务器(j)
服务器(i)
24
22
装载平衡器
httpd 代理
核心部分
客户
12
本地服务器=X
母节点=B
本地服务器 = X
子节点=A
本地服务器=X
母节点=B
本地服务器
(T)
GET
路由器
26
滤波器
监测器
SYN+GET
28
16b
16c
20

图 8

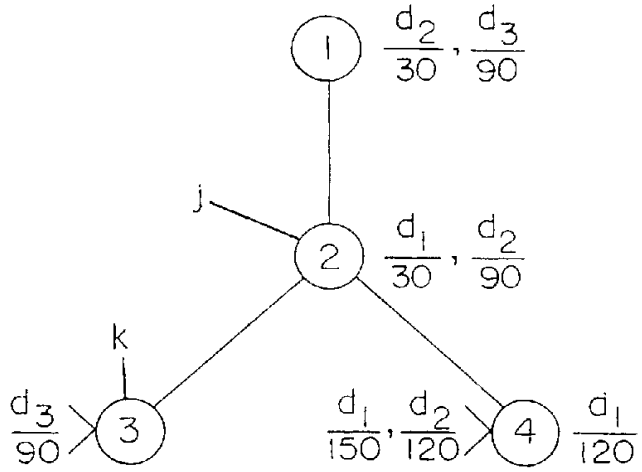

1
$\frac{d_2}{30}, \frac{d_3}{90}$
j
2
$\frac{d_1}{30}, \frac{d_2}{90}$
k
$\frac{d_3}{90}$
3
$\frac{d_1}{150}, \frac{d_2}{120}$
4
$\frac{d_1}{120}$

图 9A

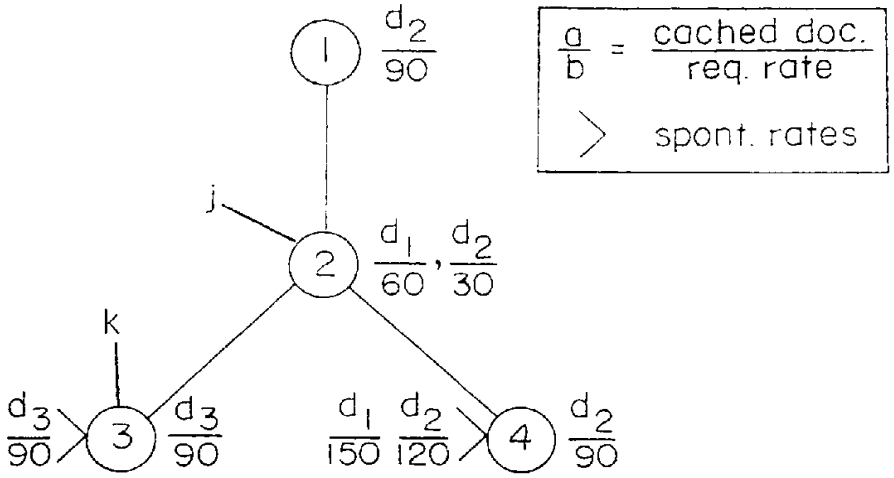

1
$\frac{d_2}{90}$
$\frac{a}{b}$ = cached doc. / req. rate
spont. rates
j
2
$\frac{d_1}{60}, \frac{d_2}{30}$
k
$\frac{d_3}{90}$
3
$\frac{d_3}{90}$
$\frac{d_1}{150}$ $\frac{d_2}{120}$
4
$\frac{d_2}{90}$

图 9B

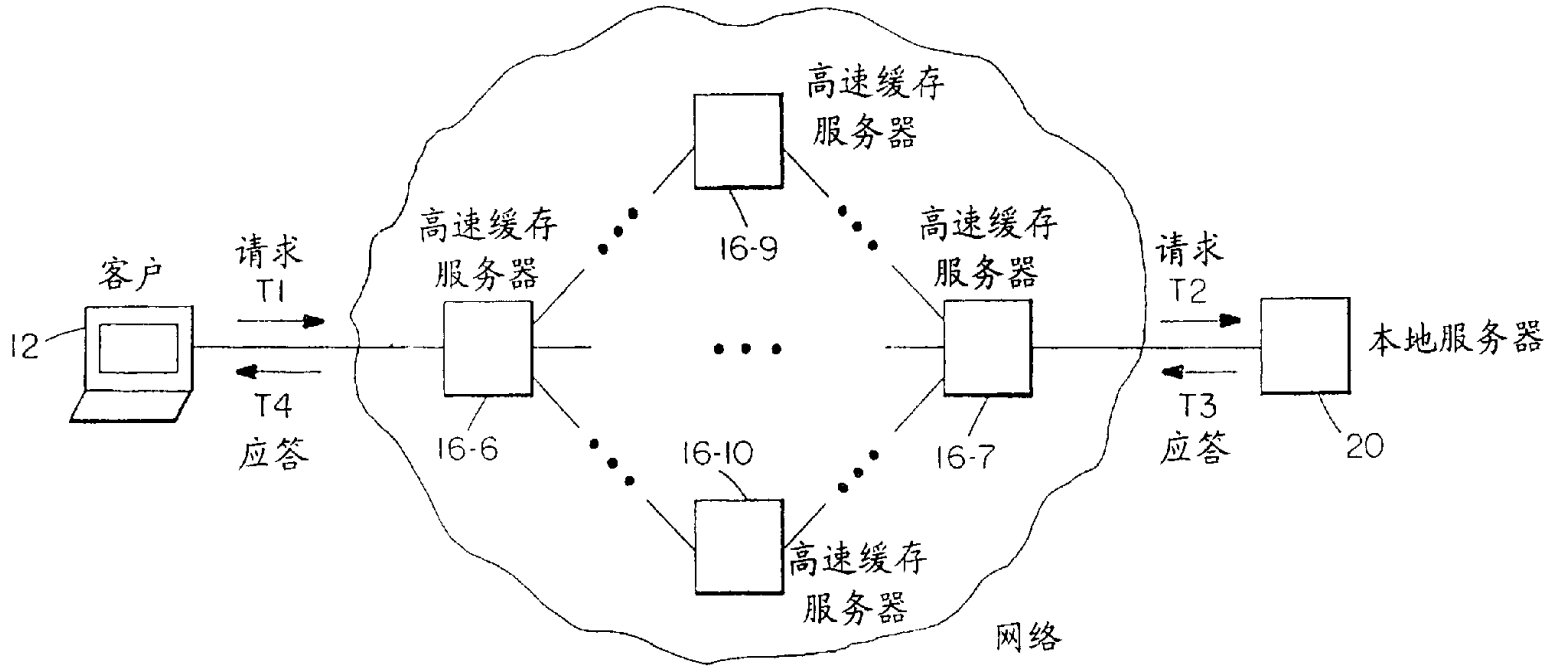

高速缓存
服务器
高速缓存
服务器
高速缓存
服务器
客户
请求
T1
12
T4
应答
16-9
16-6
16-10
16-7
高速缓存
服务器
网络
请求
T2
T3
应答
本地服务器
20

图 10