(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102984290 B
(45)授权公告日 2015.08.19

(21)申请号 201210502601.3
(22)申请日 2009.02.23
(30)优先权数据
12/043080 2008.03.05 US
(62)分案原申请数据
200980107737.0 2009.02.23
(73)专利权人 索尼电脑娱乐公司
地址 日本东京都
(72)发明人 竹田丰
(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 刘金凤 王忠忠
(51)Int.Cl.
H04L 29/12(2006.01)
(56)对比文件
US 2004/0139228 A1,2004.07.15, 全文.
WO 2007/041417 A1,2007.04.12, 全文.
CN 1998218 A,2007.07.11, 全文.
审查员 张颖浩

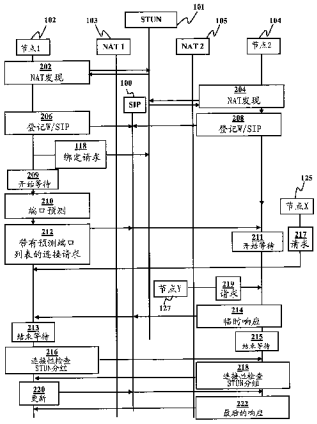
权利要求书3页 说明书10页 附图5页

(54)发明名称

用于多重同时连接的对称网络地址转换器的穿越

(57)摘要

公开了对于对称NAT后面的节点在NAT穿越期间的多重连接的处理。通过在端口预测之后使临界时间窗口串行化,对称NAT穿越期间连接失败的可能性可以得以减小。一旦已经开始了对于第一连接的端口预测,对于后续连接的端口预测可以被延迟直到对于第一连接的连接性检测已经开始。对于到不同节点的多重同时连接,可以重复此过程以处理NAT穿越。



CN 102984290 B

1. 一种用于网络上位于第一对称网络地址转换器(NAT)后面的第一节点和两个或更多其它节点之间的端到端连接的方法，所述方法包括：

a)从第一节点向第二节点发送包含在所述第一对称网络地址转换器上的预测的传输地址的列表的连接请求消息，其中利用第一节点针对第一节点和第二节点之间的通信会话构建预测的传输地址的列表；

b)在第一节点处接收对所述连接请求消息的临时响应；

c)使用预测的传输地址来执行第一节点和第二节点之间的连接性的检查；以及

d)延迟用于第一节点和第三节点之间的通信的端口预测，直到 c)已经开始之后。

2. 如权利要求 1 所述的方法，还包括：

e)在 d)之后，执行用于第一节点和第三节点之间的通信的端口预测，其中所述第一节点构建在所述第一对称网络地址转换器上的预测的传输地址的列表；

f)从所述第一节点向第三节点发送包含预测的传输地址的所述列表的连接请求消息；

g)使用所述预测的传输地址执行所述第一节点和所述第三节点之间的连接性的检查。

3. 如权利要求 2 所述的方法，还包括：h)延迟用于所述第一节点和第四节点之间的通信的端口预测，直到 f)之后。

4. 如权利要求 1 所述的方法，其中，执行连接性检查包括：从所述第二节点发送用于 NAT 的会话穿越公用程序(STUN)分组到由所述第一节点在所述连接请求消息中提供的所述一个或多个传输地址。

5. 如权利要求 4 所述的方法，其中，执行所述连接性检查还包括：从所述第一节点向所述第二节点发送 STUN 分组响应，其中所述 STUN 分组响应包括在所述第一对称网络地址转换器上的外部端口的传输地址，从所述第二节点发送的 STUN 分组之一通过所述第一对称网络地址转换器到达所述第一节点。

6. 如权利要求 1 所述的方法，还包括：

响应于来自第二节点的、指示第二节点针对连接到另一节点而在排队等待的消息而在 a)之后但在 b)之前等待时间段 T_w；

取消第一节点和第二节点之间的通信会话；以及

重复 a)、b)、c)、d)，

其中时间量 T_w 大于或等于零且小于第一节点和第二节点之间的连接失败的超时。

7. 如权利要求 1 所述的方法，其中 c)包括使用来自所述预测的传输地址的列表的传输地址从所述第一节点向所述第二节点发送一个或多个测试分组。

8. 如权利要求 7 所述的方法，其中 d)包括延迟用于所述第一节点和所述第三节点之间的通信的端口预测，直到所述一个或多个测试分组中的第一个已经被发送之后。

9. 一种用于网络上位于第一对称网络地址转换器(NAT)后面的第一节点和两个或更多其它节点之间的端到端连接的设备，所述设备包括：

用于从第一节点向第二节点发送包含在所述第一对称网络地址转换器上的预测的传输地址的列表的连接请求消息的装置，其中利用第一节点针对第一节点和第二节点之间的通信会话构建预测的传输地址的列表；

用于在第一节点处接收对所述连接请求消息的临时响应的装置；

用于使用预测的传输地址来执行第一节点和第二节点之间的连接性的检查的装置；以及

用于延迟用于第一节点和第三节点之间的通信的端口预测，直到第一节点和第二节点之间的连接性的检查已经开始之后的装置。

10. 一种用于在网络上位于第一对称网络地址转换器（NAT）后面的第一节点和两个或更多其它节点之间的端到端连接的方法，该方法包括：

a）在第一节点处接收对发送到第二节点的连接请求消息的临时响应，其中，所述连接请求消息包含在所述第一对称网络地址转换器上的预测的传输地址的列表，其中利用第一节点针对第一节点和第二节点之间的通信会话构建预测的传输地址的列表；

b）使用所述预测的传输地址执行第一节点和第二节点之间的连接性的检查；以及

c）延迟用于第一节点和第三节点之间的通信的端口预测，直到第一节点和第二节点之间的连接性的检查已经开始之后。

11. 如权利要求 10 所述的方法，还包括：

d）在 c）之后，执行用于第一节点和第三节点之间的通信的端口预测，其中第一节点构建在第一对称网络地址转换器上的预测的传输地址的列表；

e）从第一节点向第三节点发送包含预测的传输地址的列表的连接请求消息；

f）使用所述预测的传输地址执行所述第一节点和所述第三节点之间的连接性的检查。

12. 如权利要求 11 所述的方法，还包括：

g）延迟用于第一节点和第四节点之间的通信的端口预测，直到 e）已经开始之后。

13. 如权利要求 10 所述的方法，其中执行连接性的检查包括：从第二节点发送用于 NAT 的会话穿越公用程序（STUN）分组到所述连接请求消息中的所述预测的传输地址的列表中的一个或多个传输地址。

14. 如权利要求 13 所述的方法，其中执行连接性的检查还包括：从第一节点向第二节点发送 STUN 分组响应，其中所述 STUN 分组响应包括所述第一对称网络地址转换器上的外部端口的传输地址，从所述第二节点发送的 STUN 分组之一通过所述第一对称网络地址转换器到达所述第一节点。

15. 如权利要求 10 所述的方法，还包括：

响应于来自第二节点的、指示第二节点针对连接到另一节点而在排队等待的消息而在 a）之前等待时间段 T_w；

取消第一节点和第二节点之间的通信会话；以及

重复 a）、b）和 c），其中时间量 T_w 大于或等于零且小于第一节点和第二节点之间的连接失败的超时。

16. 如权利要求 10 所述的方法，其中执行第一节点和第二节点之间的连接性的检查包括：使用来自预测的传输地址的所述列表的传输地址从第一节点向第二节点发送一个或多个测试分组。

17. 如权利要求 16 所述的方法，其中延迟用于第一节点和第三节点之间的通信的端口预测包括：延迟用于第一节点和第三节点之间的通信的端口预测，直到所述一个或多个测试分组中的第一个已被发送之后。

18. 一种用于网络上位于第一对称网络地址转换器（NAT）后面的第一节点和两个或更

多其它节点之间的端到端连接的设备，所述设备包括：

用于接收对从第一节点发送到第二节点的连接请求消息的临时响应的装置，其中所述连接请求消息包含在所述第一对称网络地址转换器上的预测的传输地址的列表，其中利用第一节点针对第一节点和第二节点之间的通信会话构建预测的传输地址的列表；

用于利用所述预测的传输地址的列表执行第一节点和第二节点之间的连接性的检查的装置；以及

用于延迟用于所述第一节点和第三节点之间的通信的端口预测，直到第一节点和第二节点之间的连接性的检查已经开始之后的装置。

19. 如权利要求 18 所述的设备，还包括：

用于在所述延迟之后，执行用于第一节点和第三节点之间的通信的端口预测的装置，其中第一节点构建在第一对称网络地址转换器上的预测的传输地址的列表；

用于从第一节点向第三节点发送包含预测的传输地址的列表的连接请求消息的装置；

用于使用所述预测的传输地址执行所述第一节点和所述第三节点之间的连接性的检查的装置。

20. 如权利要求 19 所述的设备，还包括：

用于延迟用于第一节点和第四节点之间的通信的端口预测，直到将包含预测的传输地址的列表的连接请求消息从第一节点发送到第三节点之后的装置。

21. 如权利要求 18 所述的设备，其中所述设备被配置为：通过从第二节点发送用于 NAT 的会话穿越公共程序（STUN）分组到所述连接请求消息中的所述预测的传输地址的列表中的一个或多个传输地址，来执行第一节点和第二节点之间的连接性的检查。

22. 如权利要求 21 所述的设备，其中所述设备被配置为：通过从第一节点向第二节点发送 STUN 分组响应来执行连接性的检查，其中所述 STUN 分组响应包括所述第一对称网络地址转换器上的外部端口的传输地址，从第二节点发送的所述 STUN 分组之一通过所述第一对称网络地址转换器到达第一节点。

23. 如权利要求 18 所述的设备，还包括用于响应于来自第二节点的、指示第二节点针对连接到另一节点而在排队等待的消息而在接收到所述临时响应之前等待时间段 T_w 的装置；

用于取消第一节点和第二节点之间的通信会话的装置；以及

用于重复所述接收、执行以及延迟的装置，其中时间量 T_w 大于或等于零且小于第一节点和第二节点之间的连接失败的超时。

24. 如权利要求 18 所述的设备，其中所述设备被配置为：通过使用来自预测的传输地址的所述列表的传输地址从第一节点向第二节点发送一个或多个测试分组，来执行第一节点和第二节点之间的连接性的检查。

25. 如权利要求 24 所述的设备，其中所述设备被配置为：通过延迟用于第一节点和第三节点之间的通信的端口预测、直到所述一个或多个测试分组中的第一个已经被发送之后，来延迟用于第一节点和第三节点之间的通信的端口预测。

用于多重同时连接的对称网络地址转换器的穿越

[0001] 本申请是于 2010 年 9 月 3 日提交的、申请日为 2009 年 2 月 23 日、发明名称为“用于多重同时连接的对称网络地址转换器的穿越”、申请号为 200980107737.0 的中国申请的分案申请。

[0002] 优先权声明

[0003] 本申请要求于 2008 年 3 月 5 日提交的共同转让的美国专利申请 12/043,080 的优先权的权益，其全部公开内容在此引入以供参考。

技术领域

[0004] 本发明的实施例涉及计算机网络，以及更具体涉及在计算机网络上跨对称网络地址转换器的端到端通信。

背景技术

[0005] 带有 NAT（网络地址转换，Network Address Translation）特征的路由器的使用能够干扰从外部网络接入内部网络。这对于诸如在因特网上的语音通信（称为 VoIP）和 / 或在线游戏等等之类的端到端应用而言，是个特别问题。NAT 是因特网标准，其使得局域网（LAN）对内部流量使用一组私有 IP 地址而对外部流量使用第二组全局 IP 地址。具有 NAT 能力的节点常常被称为“NAT 盒”。

[0006] NAT（字面上）在这两个网络之间转换网络（IP）地址。网络地址端口转换（Network Address Port Translation, NAPT）不仅转换 IP 地址而且还转换传输层协议的端口号。虽然 NAT/NAPT 具有其良好属性，但也存在显著的副作用。如果转换是动态执行的，那么外部网络中的节点就没有办法提前知道 NAT 上的 IP 地址（和端口号）以到达内部网络中的节点。不幸的是，这是目前市场上所配置的住宅路由器和 SOHO 路由器中的 NAT 的最普遍的行为。

[0007] NAT 一般可以分类为完全锥形（Full Cone），受限锥形（Restricted Cone），端口受限锥形（Port Restricted Cone）或对称。完全锥形 NAT 将来自相同的内部 IP 地址和端口的所有请求都映射到相同的外部 IP 地址和端口。此外，任何外部主机可以通过把分组发送到所映射的外部地址，来把分组通过完全锥形 NAT 发送到内部主机。在受限锥形 NAT 中，所有来自相同的内部 IP 地址和端口的请求被映射到相同的外部 IP 地址和端口。不像完全锥形 NAT，具有 IP 地址 X 的外部主机只有当内部主机先前已经向 IP 地址 X 发送了分组时才能够向该内部主机发送分组。端口受限锥形 NAT 类似于受限锥形 NAT，但是该限制还包括端口号。具体地，外部主机只有当内部主机先前已经向 IP 地址 X 和端口 P 发送了分组时才能够向该内部主机发送具有源 IP 地址 X 和源端口 P 的分组。

[0008] 在对称 NAT 中来自相同的内部 IP 地址和端口，到特定目的地 IP 地址和端口的所有请求被映射到相同的外部 IP 地址和端口。如果该相同的主机发送带有相同源地址和端口的分组但是到不同的目的地，则使用不同的映射。此外，仅有接收分组的外部主机才能够向回发送 UDP 分组到该内部主机。对称 NAT 往往是要穿越的最难以解决的 NAT 类型。一种用于对称 NAT 穿越的技术被称作“端口预测”，其在美国专利申请公开 20070076729A1 中予

以详细描述，在此将其引入以供参考。在此对称 NAT 穿越类型中，第一节点位于对称的第一 NAT 的后面和第二节点位于第二 NAT 的后面。第一节点在第一 NAT 上构建预测的传输地址列表并把包含该预测的传输地址列表的消息发送给第二节点。利用第二节点使用该预测的传输地址执行连接性检查。

[0009] 经估计，18% 的 NAT 是对称的，而且在不进行端口预测的情况下预计有 10% 以上的连接失败率。一些涉及 NAT 穿越的应用可能需要高达 64 个同时连接。对于这样的应用而言，不清楚端口预测是否能够可靠地工作。

[0010] 它处于本发明的实施例所出现的上下文之内。

附图说明

[0011] 通过考虑结合附图的下面详细描述，会很容易理解本发明的教导，其中：

[0012] 图 1A-1B 是图示使用端口预测的 NAT 穿越的消息序列图。

[0013] 图 2 是图示依据本发明的实施例的修改的端口预测的时序图。

[0014] 图 3 是图示依据本发明的实施例的用于在两个节点之间的 NAT 穿越的系统的示意图。

[0015] 图 4 是图示依据本发明的实施例的在两个节点之间的 NAT 穿越的方法的流程图。

[0016] 图 5 是三个节点在网络上通信的示意图，图示了同时预测的问题。

[0017] 图 6 是依据本发明的实施例的被配置成执行端口预测的节点的示意图。

具体实施方式

[0018] 虽然为了举例说明的目的，接下来的详细描述包括许多具体细节，但是任何本领域技术人员将理解对下述细节的许多改进和变化都处于本发明的范围之内。因此，在不对所要求保护的发明的普遍性有任何损失的情况下以及在不对所要求保护的发明施加限制的情况下，阐述下面描述的本发明的实施例的示例。

[0019] Ⅱ．概要

[0020] 端口预测技术用于穿越对称 NAT（网络地址转换器）。如果对称 NAT 后面的节点每次正试着连接一些远程节点，则预测可能失败的概率较高。本发明的实施例通过使新的请求挂起直到先前的请求已经到达连接性检查阶段来避免由同时端口预测的情况所引起的连接失败。这允许节点通过对称 NAT 的多重同时连接而不导致关于端口预测的问题。

[0021] Ⅲ．术语

[0022] 如这里所使用的，下面术语具有在下表Ⅰ中所示出的含义。

[0023] 表Ⅰ

[0024] Natted（NAT 后面）节点位于 NAT 后面的节点。

[0025] 客户端-服务器所有客户端连接到中央服务器的网络架构模型。（例如，Web（HTTP）服务等）

[0026] 端到端不依靠中央服务器的网络架构。每个客户端（端点）直接连接到其它端点。（例如，DHT、VoIP、文件共享系统等）

[0027] 传输地址一组 IP 地址和端口号。

[0028] 网络语音电话业务 (VoIP) 使得在因特网或任何其它 IP 网络上会话能够路由的技

术。

[0029] 首字母缩略词

[0030] 如这里所使用的，下面首字母缩略词具有在下表II中所示出的含义。

[0031] 表II

[0032]	DHT	分布式哈希表(例如，Chord)
[0033]	HTTP	超文本传输协议
[0034]	IETF	因特网工程任务组
[0035]	LAN	局域网
[0036]	NAT	网络地址转换器(RFC 3022)
[0037]	NAPT	网络地址端口转换(RFC 3022)
[0038]	SIP	会话发起协议(RFC 3261)
[0039]	SSP	会话建立协议
[0040]	SOHO	小型办公室和家庭办公室
[0041]	STUN	用于NAT的会话穿越的公用程序(先前被称为“UDP通过NAT的简单穿越”)
[0042]	TURN	使用中继NAT的穿越
[0043]	VoIP	网络语音电话业务。

[0044] IV．问题分析

[0045] 端口预测失败的基本问题可以关于图1A和图1B来理解。图1A图示了当第一节点2位于对称NAT（未示出）后面时的NAT穿越的基本流。为了发起与第二节点4的通信，第一节点2发送绑定请求10到STUN服务器6。在响应中，STUN服务器6向回发送来自对称NAT的地址信息12。使用地址信息12和端口预测，第一节点2能够构建包括端口号的候选地址列表。第一节点2经由信令服务器8发送带有该列表的连接请求14给节点4。第二节点4也可以发送绑定请求16到STUN服务器6以及在响应中接收地址信息18。使用地址信息18，第二节点4可以生成其自己的候选列表和在临时响应20中发送该列表给第一节点2。第一和第二节点于是可以通过发送检查分组22、24来执行连接性检查。

[0046] 注意，对于第一节点2而言，临界时间窗T_{C1}存在于绑定请求10的发送和检查分组中的第一个检查分组22的发送之间。在此时间期间由第一节点2执行的端口预测是基于由STUN服务器4响应于绑定请求10而发送的地址信息12。如果该地址信息在此临界时间期间改变，则端口预测可能就失败了。如果第一节点2发起另一个与第三节点(未示出)通信的绑定请求，那么该地址信息可能改变。对于第二节点4而言，类似的临界时间窗T_{C2}存在于来自第一节点2的连接请求14的接收和第一检查分组24的发送之间存在。

[0047] 总的来说，当发送了第一组连接性检查分组时，对于NAT穿越的临界时间窗终止。参照附图1A，当发送了第一组检查分组22时，对于第一节点2的临界时间窗T_{C1}终止。当发送了第一组检查分组24时，对于第二节点4的临界时间窗T_{C2}终止。

[0048] 关于图1A所描述的对称NAT穿越对于在一个节点和另一个节点之间的通信通常具有良好效果。如果节点试图与不同的节点进行多重同时连接，则可能会出现意料之外的问题。例如，在多用户在线游戏中，这样的情况可能出现。图1B图示了运行在对称NAT后

面的节点上的单个应用 30 的多重通信会话的问题。作为示例，应用 30 可以发起试图与节点 A 连接的第一会话 32。第一会话 32 协商由绑定请求开始的连接和接下来的临时响应的接收开始连接性检查。然而，假设在临界时间窗T_c期间应用 30 发起了试图与节点 B 通信的第二会话 34。作为对于第二会话 34 的协商的一部分，新的 NAT 绑定请求将被发送到 STUN 服务器。这改变了 NAT 上的端口分配。不幸的是，第一会话 32 依赖于针对其与节点 A 通信的端口预测的先前端口分配。结果，对于第一会话 32 的端口预测可能失败以及在某预定的超时时间段已经过去了之后第一会话将超时。虽然现有的端口预测模式易于受到显而易见的端口预测失败，但是如上描述的该问题的特定本质，通常未被识别出来。

[0049] Ⅳ．解决方案

[0050] 本发明的实施例在每次建立多个连接时通过在 NAT 穿越的协商阶段期间使“临界窗”串行化，来克服多重端口预测的问题。注意，诸如多用户在线游戏应用之类的某些应用能够支持多个端到端连接的建立。

[0051] 依据本发明的实施例，临界时间窗的串行化可以被实现在试图在网络上发起与两个或更多其它节点的端到端连接的、对称 NAT 后面的节点上。特别地，第一节点可以执行针对由第一节点发起的与第二节点的通信会话的端口预测，和在对称 NAT 上构建预测的传输地址列表，以及接着发送包含该预测的传输地址列表的 CONNECTION REQUEST(连接请求)消息给第二节点。在接收到对该 CONNECTION REQUEST 消息的临时响应时，第一节点就可以使用该预测的传输地址，例如通过发送测试分组来执行第一节点和第二节点之间的连接性的检查。，第一节点可以通过延迟针对第一节点和第三节点之间的通信的端口预测直到该连接性检查已经开始之后来使临界时间窗串行化。

[0052] 在类似的方式中，临界时间窗的串行化可以被实现在对称 NAT 后面的节点上，其在与一个或更多其它节点协商 NAT 穿越时接收 CONNECTION REQUEST 以在网络上发起端到端连接。当这样的节点从另一节点接收 CONNECTION REQUEST 消息时，它可以执行端口预测并针对该 CONNECTION REQUEST 消息发送带有预测的地址列表的临时响应以及执行连接性的检查。对于与另外节点的通信的端口预测可以被延迟直到该连接性检查已经开始之后。

[0053] 具体地，如图 2 的时序图所示，第一节点 2 可以同时发起或接收 N 个连接请求R_1，R_2，…R_N。可以如关于图 1A 所描述的那样处理第一请求R_1。然而后续请求R_2，…R_N以它们被生成或接收的次序的串行方式来排队。具体地，后续请求中的每一个在开始端口预测之前等待，直到针对每个先前请求的临界时间窗T_c已经过去。

[0054] 如上所论述的，如果连接不能在超时时间段T_O之内建立，则该连接可能失败。因此，最好超时时间段T_O足够长以使所有 N 个连接都能够被建立。作为示例，假设对于每个连接的所有临界时间窗T_c是一些低于连接的某最大数的，超时时间段T_O应该低于$N_{max}T_c + T_{conn}$，其中T_{conn}是用于连接性检查的时间和N_{max}是可能同时连接的某最大数。

[0055] 如图 3 所示，第一节点 102 驻留于经由第一 NAT 103 在物理上与公共网络 PNW(例如，因特网)连接的第一私有网络 NW1 中。类似地，第二节点 104 驻留于第二私有网络 NW2 中且能够经由第二 NAT 105 接入相同的公共网络。私有网络 NW1 和 NW2 均可以包括除了

第一和第二节点 102、104 以及第一和第二 NAT 103、105 之外的其它节点和其它 NAT。其它私有网络也可以被连接到公共网络 PNW。公共网络 PNW 包括 SIP 代理服务器 100，和 STUN 服务器 101。第一节点 102 具有仅在第一 NAT 103 后面的第一私有网络 NW1 中有效的私有 IP 地址。第一节点 102 具有全局唯一且可路由的 IP 地址，以及第一 NAT 103 在公共网络和私有网络之间执行 IP 地址和端口转换。类似地，第二节点 104 具有仅在第二 NAT 105 后面的第二私有网络 NW2 中有效的私有 IP 地址。第二节点 104 具有全局唯一且可路由的 IP 地址，以及第二 NAT 105 在公共网络和私有网络之间执行 IP 地址和端口转换。

[0056] 节点 102、104 例如可以是服务器主机，诸如音频 / 视频 (A/V) 聊天，多媒体流设备，文件共享节点，在线游戏模块等等。每个节点 102、104 可以是通用计算机，当运行诸如用于实现图 6 的方法步骤的指令之类的指令时变成特殊用途的计算机。作为示例，节点 102 和 104 可以是 SIP 用户代理并能够通过 SIP 代理服务器 100 彼此发送和接收消息。节点 102 和 104 也可以是 STUN 客户端。因此，第一节点 102 能够通过使用 STUN 协议与 STUN 服务器 101 通信，来发现在第一节点 102 和 STUN 服务器 101 之间的路径中的 NAT 的存在及类型。第二节点 104 类似地能够发现第二节点 104 和 STUN 服务器 101 之间的 NAT 的存在及类型。

[0057] 通过同时参照图 3 的框图和图 4 的流程图，可以理解依据本发明的实施例的 NAT 穿越的方法。首先，每个节点 102、104 可以使用 STUN 协议执行 NAT 发现，如在图 4 的 202、204 处所指示的。在所图示的示例中，第一节点 102 发现它在第一 NAT 103 后面且第一 NAT 103 的类型是对称的。第二节点 104 发现它还在第二 NAT 105 后面而且第二 NAT 105 的类型例如是端口受限锥形。于是，两个节点 102、104 可以把它们自己登记到 SIP 代理服务器 100 中以加入网络，如在 206、208 处所指示的。这建立了到 SIP 代理服务器 100 的第一和第二信令路径 116、117。一旦建立，SIP 代理服务器 100 就能够分别通过第一和第二信令路径 116，117 转发消息到第一和第二节点 102、104。

[0058] 作为示例，第一节点 102 可能希望建立到第二节点 104 的端到端连接。第一节点 102 为新的端到端会话分配本地端口 107。然后第一节点 102 通过从本地端口 107 发送绑定请求 118 到 STUN 服务器 101 来获得外部端口 112。绑定请求的发送开始了对于第一节点 102 的临界时间窗。由于第一节点 102 知道第一 NAT 103 是存在的且是对称类型的，它可以执行如在 210 处所指示的端口预测和构建传输地址 107、112、113 和 114 的列表。该列表可以被放于新的 CONNECT REQUEST 消息中。在优选实施例中，第一节点 102 可以不发送关于第一 NAT 103 的信息。此外，第一节点 102，而不是第二节点 104，可以执行端口预测。另外，发送包含传输地址的 CONNECT REQUEST 消息与现有的 ICE 方法很好的兼容。

[0059] 为了防止后续的与另一节点连接的绑定请求干扰端口预测，第一节点 102 在发送该绑定请求 118 时可以开始等待时间段，如在 209 处所指示的。在等待时间段期间，用于与其它节点通信的新的绑定请求可以被临时挂起并以它们源起的次序排队。

[0060] 在步骤 212 中，第一节点 102 通过本地端口 106 和已经建立的路径 116 发送带有该传输地址列表的 CONNECT REQUEST 消息给 SIP 代理服务器 100。SIP 代理服务器 100 在消息中发现最终目的地是第二节点 104，并通过已经建立的路径 117 和端口 111 转发该 CONNECT REQUEST 消息，以及最后该消息到达本地端口 108 上的第二节点 104。在接收到该 CONNECT REQUEST 时，第二节点 104 可以为未来的端到端会话分配本地端口 109，以及然后第二节点 104 通过从本地端口 109 发送绑定消息 119 到 STUN 服务器 101 可以获得外部端口 115。为

了防止后续的与另一节点连接的绑定请求干扰端口预测，第二节点 104 在接收该 CONNECT REQUEST 消息时可以开始等待时间段，如在 211 处所指示的。在等待时间段期间，第二节点 104 可以把用于与其它节点通信的新的绑定请求临时挂起并以它们源起的次序对它们进行排队。具体地，节点 1 102 可以把来自节点 X 125 的请求 217 以接收该请求的次序放到队列中。节点 2 104 可以类似地对来自节点 Y 127 的请求 219 进行排队。

[0061] 由于第二节点 104 知道第二 NAT 105 不是对称的，在 214，它把本地端口 109 和外部端口 115 放到新的临时响应消息中并经由 SIP 代理服务器 100 和第一和第二信令路径 117、116 把它向回发送给第一节点 102。该临时响应的传输终止了传输交换阶段并开始了连接性检查阶段。在这个阶段，第一和第二节点 102、104 可以安全发起用于与其它节点通信的新的绑定请求。因此，针对第一和第二节点 102、104 的等待时间段可以结束，如分别在 213、215 处所指示。

[0062] 为了检查连接性，在步骤 216、218，节点 102 和 104 二者可以开始从它们的本地端口 107、109 发送 STUN 分组到从其它节点获得的传输地址以检查连接性。当第一节点 102 发送 STUN 分组 120 时，第一 NAT 103 分配新的外部端口 113，以及然后分组 120 到达第二 NAT 105 上的外部端口 115。第一少数分组可能在外部端口 115 处被丢弃，这是因为第二 NAT 105 是端口受限锥形 NAT 和第二节点 104 可能还尚未从本地端口 109 发送分组到第一 NAT 103 上的本地端口 113。第二节点 104 也发送 STUN 分组 121、122、123 到所获得的传输地址 112、113 和 114。到达端口 112 处的分组 121 被丢弃，这是因为第一 NAT 103 是对称 NAT 和端口 112 被专门分配给到 STUN 服务器 101 的会话。到达 114 处的 STUN 分组 123 也被丢弃，这是因为不存在由第一 NAT 103 分配的这样的外部端口。到达 113 处的 STUN 分组 122 被第一 NAT 103 转发到本地端口 107。第一节点 102 于是发送响应到第二节点 104 以及第二节点 104 在接收到响应时发现它具有到端口 113 的连接性。从第一节点 102 上的本地端口 107 发送到第二 NAT 105 上的外部端口 115 的 STUN 分组最终在第二节点 104 的本地端口 109 处被接收。第二节点 104 于是向回发送响应到第一节点 102。

[0063] 在接收到响应消息时，在步骤 220，第一节点 102 经由 SIP 代理服务器 100 发送 UPDATE（更新）消息给第二节点 104 以告诉第二节点 104：第一节点 102 找到了到外部端口 115 的连接性。这触发了第二节点 104 在步骤 222 发送最后的响应消息给第一节点 102 来完成连接建立过程。

[0064] 代替将 NAT 类型放在消息中以及使用第二节点 104 进行预测的是，由第一节点 102 进行预测和把预测的外部端口 113、114 连同从 STUN 服务器 101 获取的外部端口 112 一起放到新的 CONNECT REQUEST 消息中。因此，第一节点 102 不把关于第一 NAT 103 的信息提供给第二节点 104。这样的 ICE 方法的使用完全消除了现有技术（US2004/0139228）中所进行的“突发（break-out）”分组的复杂的 NAT 组合逻辑。相反，本发明的实施例能够通过执行基本上等价于“突发分组”的连接性检查来达到相同结果。因此，本发明的实施例允许已经使用 ICE 方法的系统通过简单的把预测的传输地址加到连接性检查列表来增加对称 NAT 穿越能力。

[0065] 如上所描述的，第一节点 102，即，试图发起与第二节点 104 通信的节点，执行端口预测和把预测的端口放在 CONNECT REQUEST 消息中。存在许多执行端口预测的技术。例如，运用下面的测试利用端口分配规则发现过程可以实现端口预测。第一节点

102 在不在 CHANGE-REQUEST（改变－请求）属性中设置任何标志的情况下以及在没有 RESPONSE-ADDRESS（响应－地址）属性的情况下，发送 STUN 绑定请求给 STUN 服务器 101。这促使 STUN 服务器 101 把响应发回该请求所来自的地址和端口。这个测试被应用于 IP 地址和端口的不同结合以便领会 NAT 103 的端口分配特性。STUN 服务器 101 使用在下表 III 中所示出的两个不同的 IP 地址，C_A 和 D_A 以及两个不同的端口 C_P 和 D_P。

[0066] 表 III

[0067]

	目的地	
	IP 地址	端口
TRY-1	D_A	D_P
TRY-2	D_A	C_P
TRY-3	C_A	D_P
TRY-4	C_A	C_P

[0068] 从表 III 可以看出，在这个示例中该测试每本地端口执行四次（例如，从 TRY-1 到 TRY-4）。所有的测试必须从相同的本地端口来进行。第一节点 102 从响应中获得四个映射地址。分析这些四个映射地址以确定端口分配规则和端口增量值△ P 以及估计一致性。为了寻求一致性，该过程可以被执行多次，优选使用不同的本地端口，所述本地端口不具有与其相关联的 NAT 绑定。通过查看从映射地址获得的端口号，能够确定端口分配规则。如果对于具有不同端口号的连续目的地，所有的端口号是递增的，则该端口分配规则被称为“端口敏感的（port sensitive）”。如果来自具有相同 IP 地址（例如，从 TRY-1 到 TRY-2 和从 TRY-3 到 TRY-4）的连续目的地的端口增量大小总是 0，而来自具有不同 IP 地址（例如，从 TRY-2 到 TRY-3）的连续目的地的端口增量大小不是 0，则端口分配规则被称为“地址敏感的”。如果所获得的映射地址的所有端口号相同，则 NAT 103 是“锥形 NAT”。

[0069] △P 值可以被如下确定。对于地址敏感的分配，△P 值等于目的地端口不同（例如，对于 TRY-2 和 TRY-3）的连续尝试之间的端口增量大小。从如在表 IV 中所示的另一本地端口可以重复该过程。在这个示例中，TRY-1 到 TRY-4 如在表 III 中，而 TRY-5-TRY 8 继续表 III 的模式目的地 IP 地址和端口号。

[0070] 表 IV

[0071]

TRY（尝试）	映射地址	△ P
1	67.105.12.10:49152	
2	67.105.12.10:49152	0
3	67.105.12.10:49154	2
4	67.105.12.10:49154	0
5	67.105.12.10:49156	2
6	67.105.12.10:49156	0
7	67.105.12.10:49158	2
8	67.105.12.10:49158	0

[0072] 注意，从表 IV 能够看出对于连续尝试，目的地 IP 地址是相同的，在对应的映射地址中端口号都是相同的。从这能够确定该端口分配规则是“地址敏感的”。此外，能够看出△ P 值等于 TRY-2 和 TRY-3 之间的端口增量且还等于 TRY-4 和 TRY-5 之间以及 TRY-6 和 TRY-7 之间的端口增量大小。

[0073] 对于端口敏感的分配，△ P 的值是从测试 TRY-[N+1] 和 TRY-[N] 获得的映射地址的邻接端口号之间的差。在第一节点 102 不能发现用于△ P 确定的该端口增量大小一致性的情况下，该应用可以包括基于统计观察确定△ P 值的算法，或者决定放弃获取有效△ P 的算法。

[0074] 如果第二 NAT 105 不是对称的，则它可能足以执行端口预测，例如，作为对于只是第一节点 102 的 NAT 发现步骤 202 的一部分。在第二 NAT 105 也是对称 NAT 的情况下，第二节点 104 可以执行类似的端口预测以作为其 NAT 发现阶段 204 的一部分。

[0075] 注意，在本发明的实施例中，节点可以被配置成使用于 NAT 穿越的临界时间窗串行化，为了到其它节点的多重连接，用于 NAT 穿越的临界时间窗可以被串行化。如上所述的串行化临界时间窗可能会引起隐患，在这里被称为对称节点连接环（Symmetric Node Connection Loop）。注意，对于不在对称 NAT 后面的节点，对连接请求进行排队以串行化临界时间窗通常不是必需的。然而，如果对于处于对称 NAT 后面的三个或者更多节点来执行临界时间窗串行化，则可能发生死锁（lockup）。参考图 5 可以理解此隐患的性质，在图 5 中三个节点 A、B 和 C 都位于对称 NAT 后面。如果节点 A 试着联系节点 B 而同时节点 B 正试着联系节点 A，则在这三个节点之间可能就创建了发起依赖环，其可能导致死锁，在该死锁中没有节点能够成功发起端到端通信。

[0076] 在图 5 中所描绘的示例中，如果所有三个节点被配置成执行串行化的临界时间窗，例如，如关于图 2 所讨论的，节点 A 可能等待节点 B 发起与节点 C 的通信，而节点 B 等待节点 C 发起与节点 A 的通信，以及节点 C 等待节点 A 发起与节点 B 的通信。在这样的情况下，所有三个节点可能同时等待，这创建了死锁情形。为了克服这种问题，可以执行发起一事务超时（initiate-transacation timeout）。

[0077] 作为示例，节点 C 可以发送“QUEUED（排队）”消息 Q 给节点 B。QUEUED 消息 Q 指示节点 C 位于对称 NAT 后面以及正等待与另一节点的连接。节点 B 可能等待了预定的等待时间T_w，于是取消其与节点 C 的发起，并接着在处理来自节点 A 的连接请求之后随后再发起与节点 C 的连接。等待时间T_w可以大于或等于零以及小于节点 B 放弃与节点 C 的通信之前通常会等待的超时时间。更具体地，C 发送“QUEUED”消息 Q 给 B，这是因为 C 知道 C 位于对称 NAT 的后面且该请求不会被立即处理而是在排队。响应于该“QUEUED”消息，节点 B 可能等待相对较短的时间时间段（例如 3 秒），接着取消（推迟）到节点 C 的连接过程，处理在队列中的其它请求，于是最后再次发起与 C 的连接。为了防止节点 B 无限期地再试着发起与节点 C 的连接，可以限制对相同的节点的再次尝试的次数，例如，高达 3 次，在该次数之后，节点 B 将完全放弃。在这种情况下运行在节点 B 上的应用会接收到连接错误。

[0078] 作为死锁的替代解决方案，以上描述的技术可以按照下列的方式予以修改。可以仅对于外出的连接请求，使临界时间窗串行化，并且可以执行稍微较深的端口预测。例如，代替基于端口增量△ P 来生成端口预测，另外的端口预测可以基于 2 △ P，3 △ P…M △ P

来生成，其中 M 是整数，在这里称为预测的“深度”。作为示例，假设确定了对于特定的对称 NAT，端口增量△ P=1。基于△ P 的这个值，基于深度 M=1 的端口预测的候选列表可能像这样：

[0079] 候选 1: type（类型）=stun, 202.10.9.20:9021（预测的）

[0080] 候选 2: type=local（本地）, 192.168.1.2:3658

[0081] 从 M=1 到 M=3 加深该预测可以产生如下的候选列表：

[0082] 候选 1: type=stun, 202.10.9.20:9021（预测的）

[0083] 候选 2: type=stun, 202.10.9.20:9022（预测的）

[0084] 候选 3: type=stun, 202.10.9.20:9023（预测的）

[0085] 候选 2: type=local, 192.168.1.2:3658

[0086] 不像“QUEUED”信令消息的使用，之前的解决方案可能不能解决每个可能的死锁情形，但是它实现起来更简单。此外，由于连接过程更可能在呼叫侧并行出现，所以使外出的连接串行化而不是使进入的连接串行化能够避免最可能的死锁情形。由于用于进入的连接的临界时间窗不被串行化，所以端口预测可能失败。增加端口预测的深度，例如，从 M=+1 到 M=+3 增加了预测的候选的数目，这减少了端口预测失败的可能性。

[0087] 以下关于图 3 和图 4 所图示的示例与使用 SIP（会话发起协议；RFC3261）协议的端到端 UDP 会话建立有关。然而，本发明的实施例，可应用于任何其它信令协议，该信令协议允许端点在端到端直接连接建立之前，经由公共网络上的代理服务器发送和接收至另一节点的信号。这样的信令协议可以包括但不局限于 H.323、MGCP、HTTP、XMPP 等。

[0088] NAT 穿越算法可以以软件或者硬件或者两者的组合来实现。作为示例，图 6 描绘了实现这样的算法的计算机设备 400。该设备 400 可以包括处理器模块 401 和存储器 402。处理器模块 401 可以包括单个处理器或多个处理器。作为单个处理器的示例，处理器模块 401 可以包括来自因特尔的 Pentium® 微处理器或类似的与因特尔兼容的微处理器。作为多处理器模块的示例，处理器模块 401 可以包括单元处理器。

[0089] 存储器 402 可以是集成电路的形式，例如，RAM、DRAM、ROM，等等。存储器 402 也可以是主存储器或单元处理器的协同处理器单元的本地存储器。包括以上所描述的帧重构算法的计算机程序 403 可以以能够在处理器模块 401 上执行的处理器可读指令的形式存储在存储器 402 中。处理器模块 401 可以包括存储来自程序 403 的指令的一个或多个寄存器 405。程序 403 的指令可以包括用于在网络上端到端连接的方法的步骤，例如，如上面关于图 2、图 3 和图 4 所描述的。具体地，程序 403 可以对如上所述的用于到其它节点的连接的请求进行排队以防止多重连接尝试干扰端口预测。该程序可以生成存储在存储器 402 中的连接队列 407。可以把连接队列 407 实现为表或其它数据结构。连接队列 407 可以存储发起特定连接的时间以便针对另一先前发起的队列的临界时间窗已经过去之后，可以从队列 407 中释放连接。

[0090] 程序 403 可以用任何合适的处理器可读语言来编写，所述语言例如，C、C++、JAVA、Assembly、MATLAB、FORTRAN 和许多其它语言。该设备还可以包括众所周知的支持功能 410，诸如输入 / 输出（I/O）元件 411，电源供应（P/S）412，时钟（CLK）413 和高速缓存器 414。该设备 400 任选地可以包括大容量存储装置 415，诸如盘驱动器、CD-ROM 驱动器、带驱动器等用以存储程序和 / 或数据。该设备 400 任选地还可以包括显示单元 416 和用户接口单元以

便于该装置和用户之间的交互。显示单元 416 可以是以显示文本、数字、图像符号或图像的阴极射线管(CRT)或者平板屏幕的形式。显示单元 416 还可以包括扬声器或其它产生可听得见声音的音频换能器。用户接口 418 可以包括键盘、鼠标、操纵杆、光笔、麦克风或者可与图形用户接口(GUI)结合使用的其它装置。该设备 400 还可以包括网络接口 420 以使得该装置能够在诸如因特网之类的网络上与其它装置通信。这些组件可以实现为硬件、软件或固件或者这些中的两个或多个的某组合。

[0091] V. 结论

[0092] 在实现 NAT 穿越特征的端到端库中,使用运行在该端到端库顶部上的连接性检查工具对本发明的实施例进行了测试。该连接的两个端处的节点都位于对称 NAT 的后面。在 PlayStation 3 开放站上出现的端到端库节点试图连接到运行在 Linux PC 上的 64 个其它节点。对于进入和外出连接这二者的临界时间窗被串行化。所有 64 个连接被成功建立。

[0093] 虽然上面是对本发明的优选实施例的完整描述,但是使用各种替代物、修改和等同物是可能的。因此,本发明的范围不参照以上的描述予以确定,相反,而是应该参照附带的权利要求连同它们的等同物的全部范围一起予以确定。在此所描述的无论是否优选的任何特征可以与于此所描述的无论是否优选的其它特征进行组合。在随后的权利要求中,不定冠词“一”或者“一个”指的是该冠词后面的一个或更多项目的量,除了另外特被声明的地方。附带的权利要求不能被解释为包括装置加功能限制,除非使用短语“用于…的装置”来将这样的限制明确记载在给定的权利要求中。

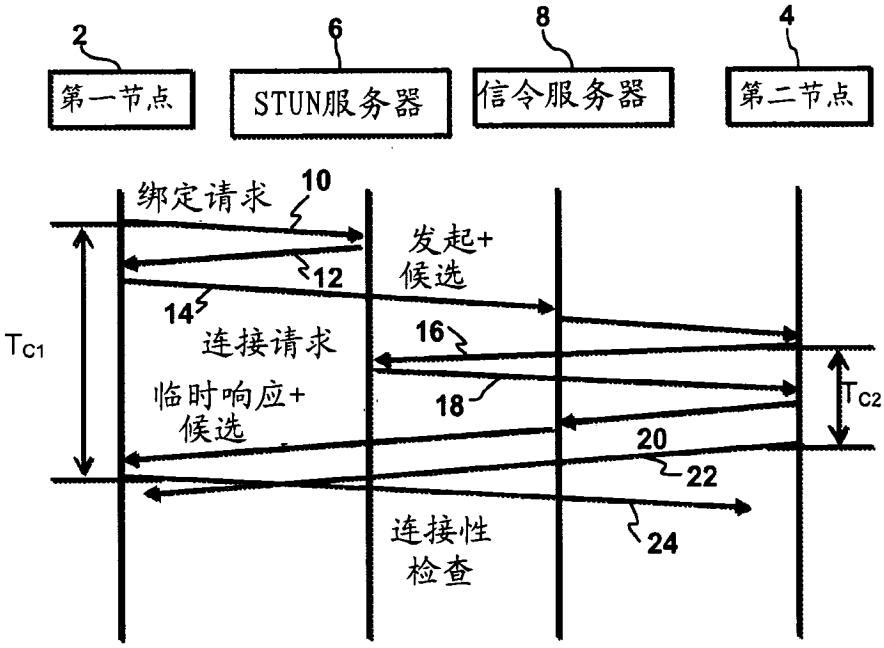

2
第一节点
6
STUN服务器
8
信令服务器
4
第二节点
绑定请求
10
发起+
候选
12
14
连接请求
16
T_{C1}
临时响应+
候选
18
T_{C2}
20
22
连接性
检查
24

图 1A

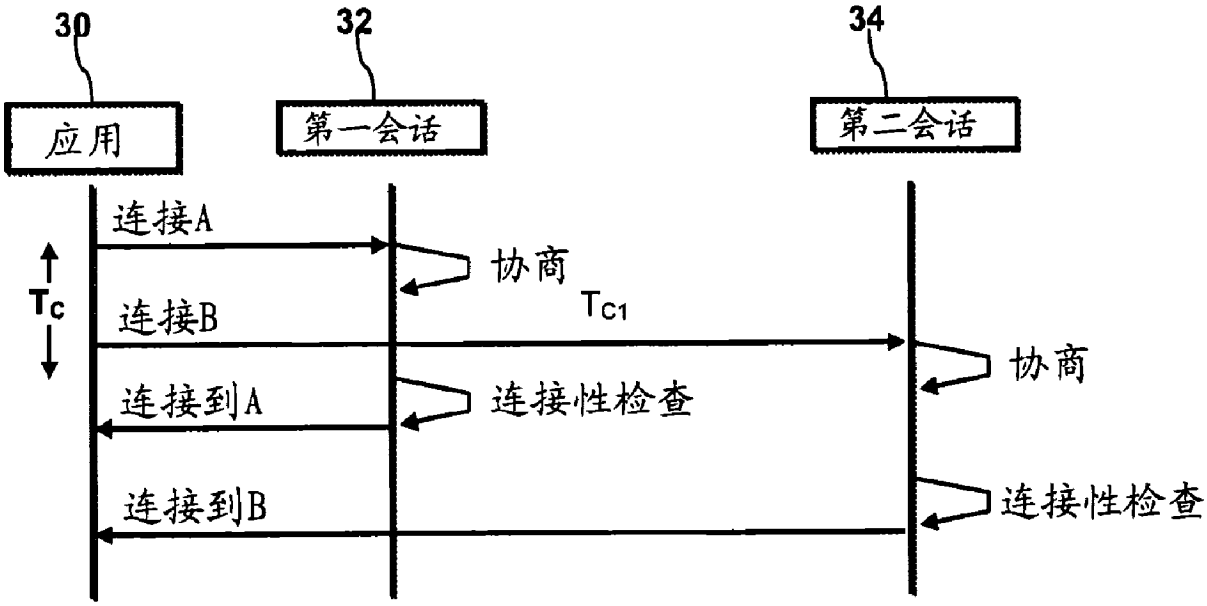

30
应用
32
第一会话
34
第二会话
连接A
协商
T_C
连接B
T_{C1}
协商
连接到A
连接性检查
连接到B
连接性检查

图 1B

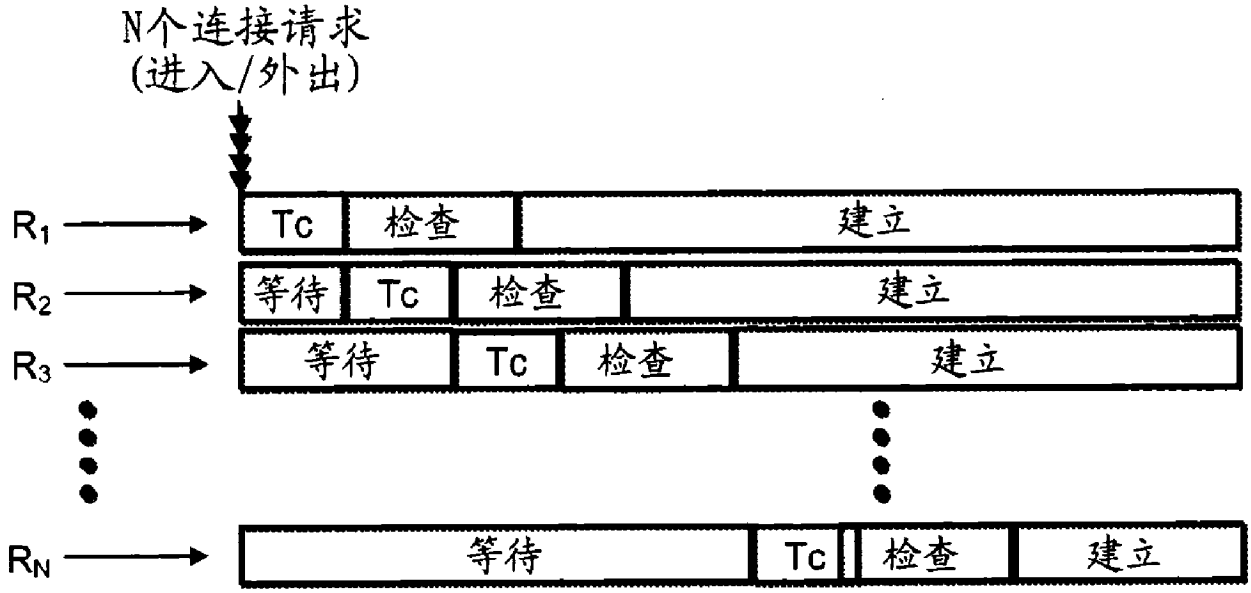

N个连接请求
(进入/外出)
R1
R2
R3
RN
Tc
检查
建立
等待
Tc
检查
建立
等待
Tc
检查
建立
等待
Tc
检查
建立

图 2

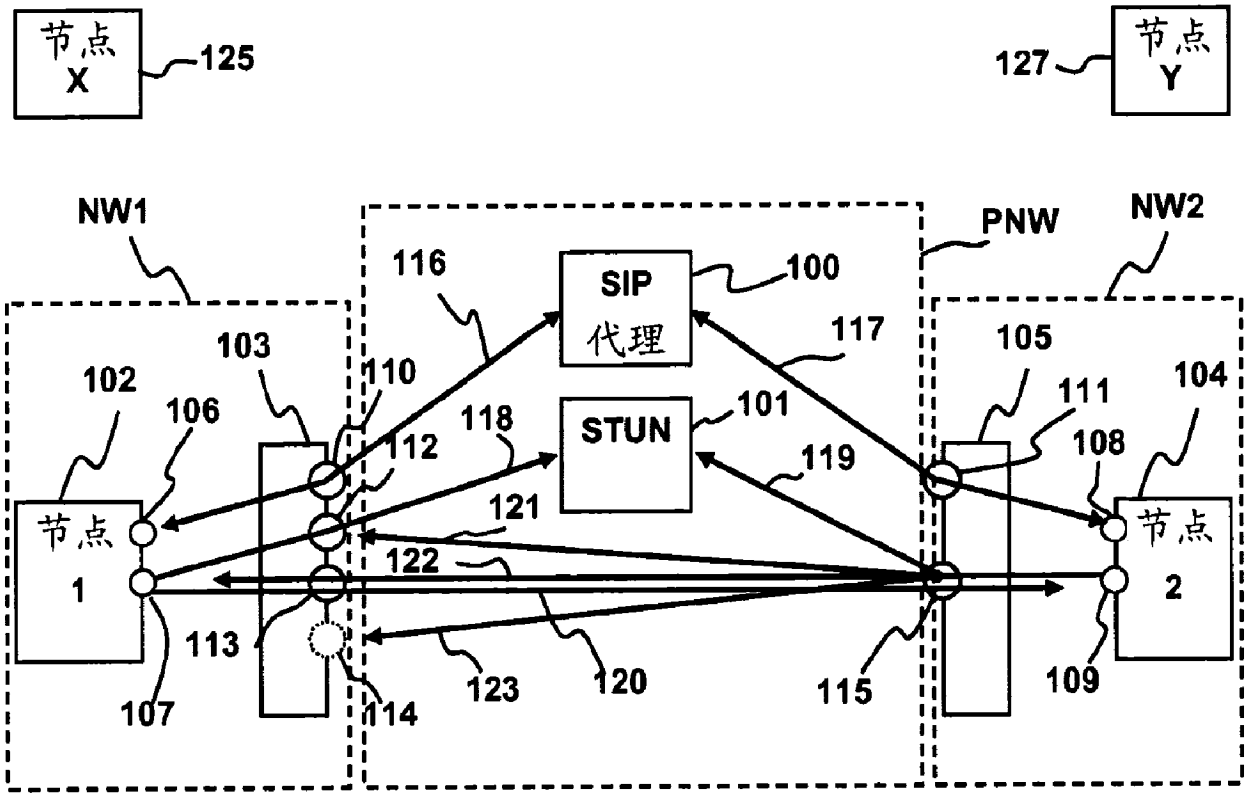

节点
X
125
127
节点
Y
NW1
PNW
NW2
SIP
代理
100
STUN
101
116
117
103
110
102
106
112
118
105
111
104
108
119
121
122
节点
1
节点
2
113
107
114
123
120
115
109

图 3

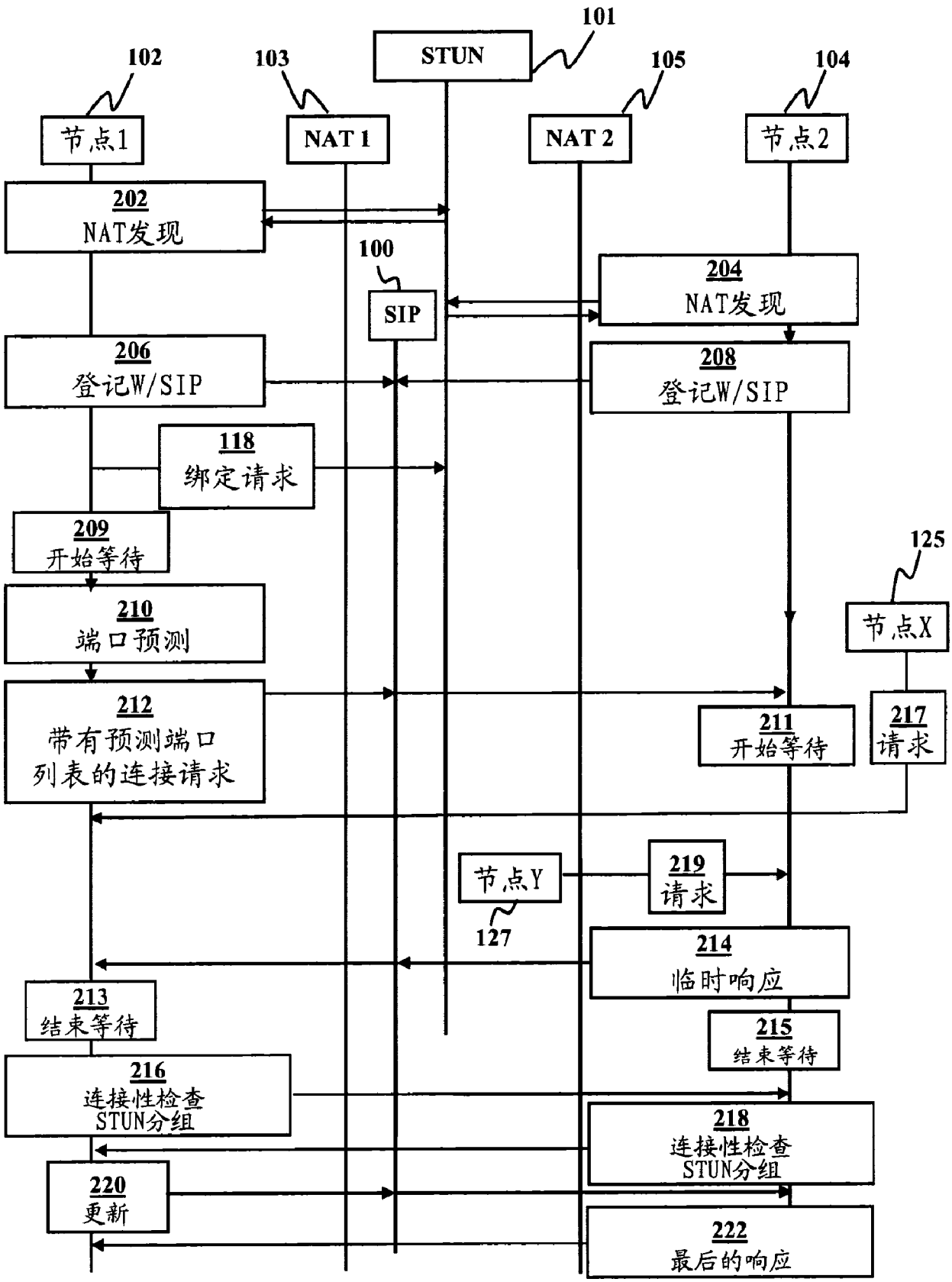

101
STUN
102
103
105
104
节点1
NAT 1
NAT 2
节点2
202
NAT发现
100
SIP
204
NAT发现
206
登记W/SIP
208
登记W/SIP
118
绑定请求
209
开始等待
125
210
端口预测
节点X
212
带有预测端口
列表的连接请求
211
开始等待
217
请求
节点Y
219
请求
127
214
临时响应
213
结束等待
215
结束等待
216
连接性检查
STUN分组
218
连接性检查
STUN分组
220
更新
222
最后的响应

图 4

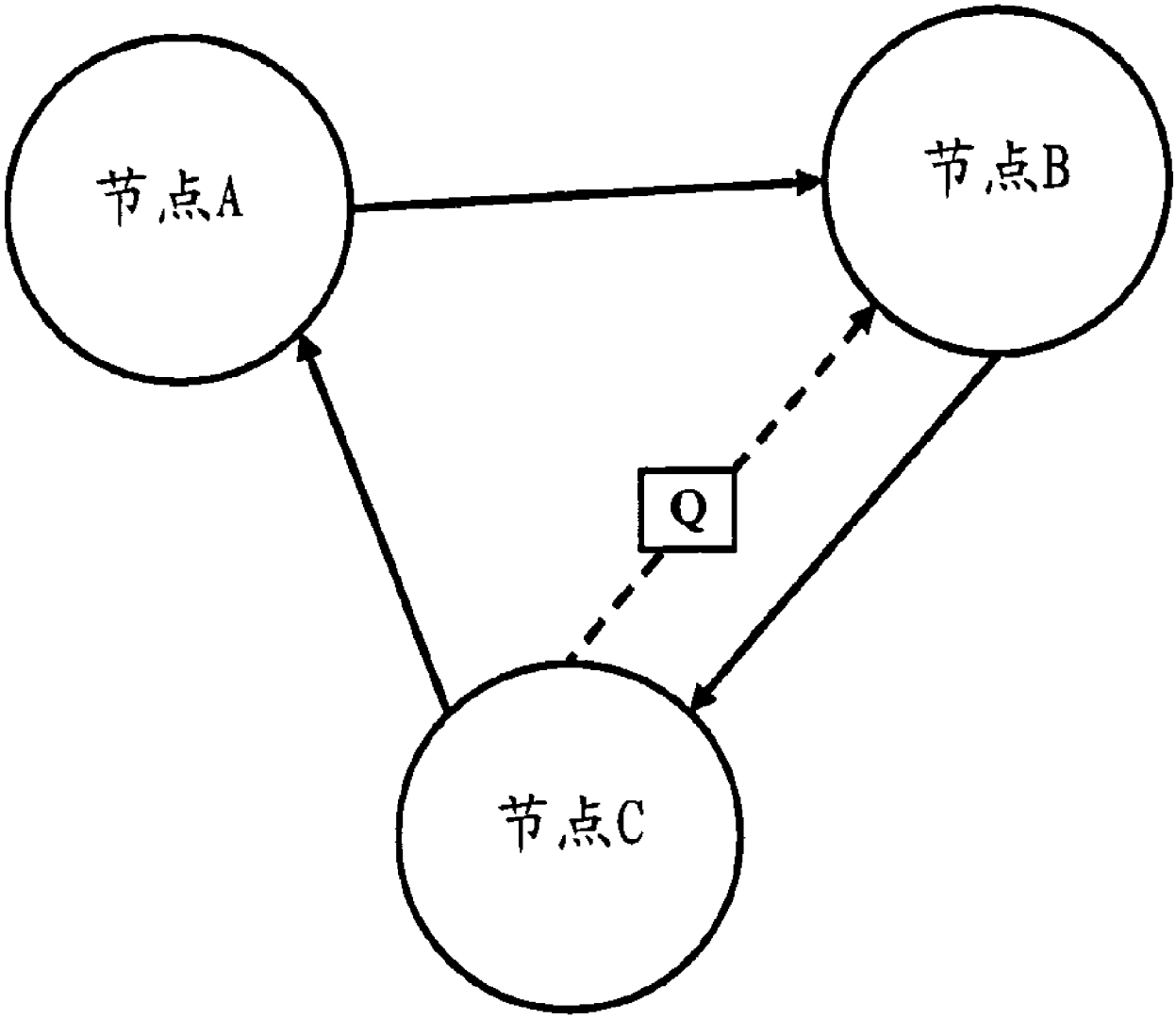


图 5

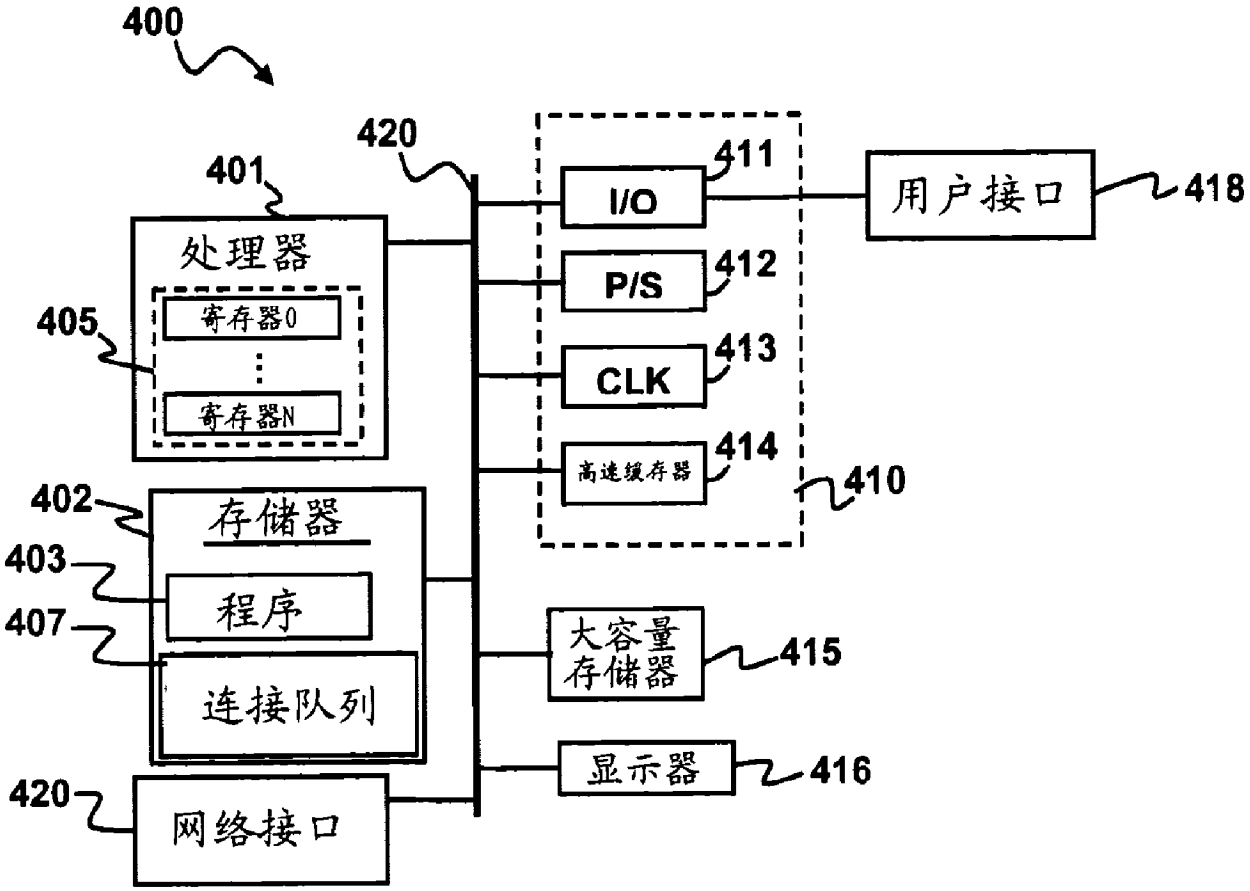

400
420
401
处理器
寄存器0
405
寄存器N
402
存储器
403
程序
407
连接队列
420
网络接口
I/O
411
用户接口
418
P/S
412
CLK
413
高速缓存器
414
410
大容量
存储器
415
显示器
416

图 6