



(12) 发明专利申请

(10) 申请公布号 CN 115190071 A

(43) 申请公布日 2022. 10. 14

(21) 申请号 202110361730.4

(22) 申请日 2021.04.02

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 喻径舟 刘杨 杨斐然 郭玲波 邢豫盛 周立伟

(74) 专利代理机构 深圳市深佳知识产权代理事务所(普通合伙) 44285 专利代理师 常忠良

(51) Int. Cl.

H04L 45/745 (2022.01)

H04L 45/74 (2022.01)

H04L 45/748 (2022.01)

H04L 61/5007 (2022.01)

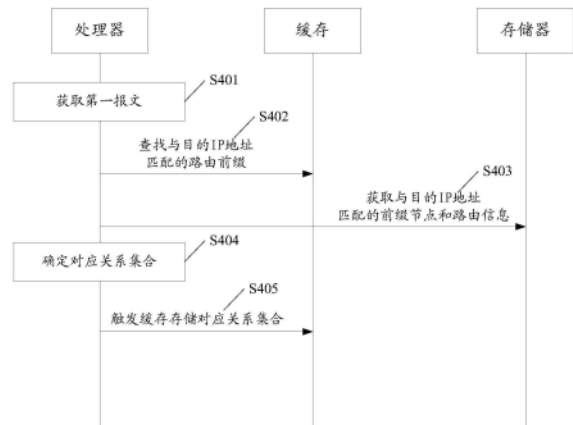
权利要求书3页 说明书22页 附图9页

(54) 发明名称

一种缓存方法及集成电路

(57) 摘要

本申请实施例提供了一种缓存方法及集成电路,能够避免虚假命中。所述缓存方法包括:获取第一报文,所述第一报文包括目的互联网协议IP地址;从缓存中查找与所述目的IP地址匹配的路由前缀;响应于所述缓存中未存储所述路由前缀,根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;响应于所述前缀节点不为叶子节点,所述处理器确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度;根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。



1. 一种缓存方法,其特征在于,所述方法包括:

处理器获取第一报文,所述第一报文包括目的互联网协议IP地址;

所述处理器从缓存中查找与所述目的IP地址匹配的路由前缀;

响应于所述缓存中未存储所述路由前缀,所述处理器根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;

响应于所述前缀节点不为叶子节点,所述处理器确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度,所述尾节点为所述树结构的查找路径上的最后一个节点;

所述处理器根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;

所述处理器触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。

2. 根据权利要求1所述的方法,其特征在于,所述第一路由前缀的长度大于或等于所述第一前缀匹配长度。

3. 根据权利要求1或2所述的方法,其特征在于,所述处理器根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点包括:

所述处理器根据所述目的IP地址查找与路由表对应的虚拟树结构,确定与所述目的IP地址最长匹配的虚拟前缀;

所述处理器根据所述目的IP地址查找与所述虚拟前缀对应的子树结构,确定与所述目的IP地址匹配的前缀节点,所述虚拟前缀为所述子树结构的根节点;

所述第一前缀匹配长度等于第一长度和第二长度之和,所述第一长度为所述虚拟树结构的根节点到所述虚拟前缀的长度,所述第二长度为所述虚拟前缀到尾节点之间的长度,所述尾节点为所述子树结构的查找路径上的最后一个节点。

4. 根据权利要求1或2所述的方法,其特征在于,所述方法还包括:

响应于所述前缀节点不为叶子节点,所述处理器确定第二前缀匹配长度,所述第二前缀匹配长度等于所述树结构的根节点到独子节点之间的长度,所述独子节点为所述前缀节点到所述尾节点之间的只具有左孩子节点或只具有右孩子节点的节点;

所述处理器根据所述第二前缀匹配长度和所述目的IP地址,确定第二路由前缀,所述第二路由前缀的长度大于所述第二前缀匹配长度;

所述处理器触发所述缓存存储第二对应关系,所述第二对应关系为所述第二路由前缀和所述前缀节点对应的路由信息。

5. 根据权利要求1-4任一项所述的方法,其特征在于,所述方法还包括:

所述处理器触发所述缓存存储所述第一路由前缀的类型,所述第一路由前缀的类型用于指示在所述第一路由前缀被命中时,根据所述第一对应关系在所述缓存中获取与所述第一路由前缀对应的路由信息。

6. 根据权利要求3所述的方法,其特征在于,所述方法还包括:

所述处理器触发所述缓存存储第三对应关系,所述第三对应关系为所述虚拟前缀、所述虚拟前缀的类型以及所述子树结构的信息之间的对应关系,所述虚拟前缀的类型用于指示在所述虚拟前缀被命中时,根据所述子树结构的信息查找所述子树结构。

7. 根据权利要求1-6任一项所述的方法,其特征在于,所述方法还包括:

所述处理器触发所述缓存存储所述第一路由前缀的长度。

8. 根据权利要求2所述的方法,其特征在于,所述第一路由前缀的长度等于所述第一前缀匹配长度加1。

9. 根据权利要求1-8任一项所述的方法,其特征在于,所述第一路由前缀的长度小于所述目的IP地址的长度。

10. 根据权利要求1-9任一项所述的方法,其特征在于,所述目的IP地址包括所述第一路由前缀。

11. 根据权利要求1-10任一项所述的方法,其特征在于,所述缓存至少包括以下其中一种:

三态内容寻址存储器TCAM、寄存器、线卡板和晶粒。

12. 一种集成电路,其特征在于,所述集成电路包括接口电路和控制电路;

其中,所述接口电路,用于获取第一报文,所述第一报文包括目的互联网协议IP地址;

所述控制电路,用于从缓存中查找与所述目的IP地址匹配的路由前缀;响应于所述缓存中未存储所述路由前缀,根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;响应于所述前缀节点不为叶子节点,确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度,所述尾节点为所述树结构的查找路径上的最后一个节点;根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。

13. 根据权利要求12所述的集成电路,其特征在于,所述第一路由前缀的长度大于或等于所述第一前缀匹配长度。

14. 根据权利要求12或13所述的集成电路,其特征在于,

所述控制电路,用于根据所述目的IP地址查找与路由表对应的虚拟树结构,确定与所述目的IP地址最长匹配的虚拟前缀;根据所述目的IP地址查找与所述虚拟前缀对应的子树结构,确定与所述目的IP地址匹配的前缀节点,所述虚拟前缀为所述子树结构的根节点;所述第一前缀匹配长度等于第一长度和第二长度之和,所述第一长度为所述虚拟树结构的根节点到所述虚拟前缀的长度,所述第二长度为所述虚拟前缀到尾节点之间的长度,所述尾节点为所述子树结构的查找路径上的最后一个节点。

15. 根据权利要求12或13所述的集成电路,其特征在于,

所述控制电路,还用于响应于所述前缀节点不为叶子节点,确定第二前缀匹配长度,所述第二前缀匹配长度等于所述树结构的根节点到独子节点之间的长度,所述独子节点为所述前缀节点到所述尾节点之间的只具有左孩子节点或只具有右孩子节点的节点;根据所述第二前缀匹配长度和所述目的IP地址,确定第二路由前缀,所述第二路由前缀的长度大于所述第二前缀匹配长度;触发所述缓存存储第二对应关系,所述第二对应关系为所述第二路由前缀和所述前缀节点对应的路由信息。

16. 根据权利要求12-15任一项所述的集成电路,其特征在于,

所述控制电路,还用于触发所述缓存存储所述第一路由前缀的类型,所述第一路由前缀的类型用于指示在所述第一路由前缀被命中时,根据所述第一对应关系在所述缓存中获取与所述第一路由前缀对应的路由信息。

17. 根据权利要求14所述的集成电路,其特征在于,

所述控制电路,还用于触发所述缓存存储第三对应关系,所述第三对应关系为所述虚拟前缀、所述虚拟前缀的类型以及所述子树结构的信息之间的对应关系,所述虚拟前缀的类型用于指示在所述虚拟前缀被命中时,根据所述子树结构的信息查找所述子树结构。

18. 根据权利要求12-17任一项所述的集成电路,其特征在于,

所述控制电路,还用于触发所述缓存存储所述第一路由前缀的长度。

19. 根据权利要求13所述的集成电路,其特征在于,所述第一路由前缀的长度等于所述第一前缀匹配长度加1。

20. 根据权利要求12-19任一项所述的集成电路,其特征在于,所述第一路由前缀的长度小于所述目的IP地址的长度。

21. 根据权利要求12-20任一项所述的集成电路,其特征在于,所述目的IP地址包括所述第一路由前缀。

22. 根据权利要求12-21任一项所述的集成电路,其特征在于,所述缓存至少包括以下其中一种:

三态内容寻址存储器TCAM、寄存器、线卡板和晶粒。

23. 一种芯片,其特征在于,所述芯片包括存储器和处理器,存储器用于存储指令或程序代码,处理器用于从存储器中调用并运行所述指令或程序代码,以执行如权利要求1-11任一项所述的缓存方法。

24. 一种设备,其特征在于,所述设备包括处理器芯片和存储器,存储器用于存储指令或程序代码,处理器芯片用于从存储器中调用并运行所述指令或程序代码,以执行如权利要求1-11任一项所述的缓存方法。

25. 一种计算机可读存储介质,其特征在于,包括指令、程序或代码,当其在计算机上执行时,使得所述计算机执行如权利要求1-11任一项所述的缓存方法。

一种缓存方法及集成电路

技术领域

[0001] 本申请涉及通信技术领域,尤其涉及一种缓存方法及集成电路。

背景技术

[0002] 路由器或交换机等网络设备在转发报文时,可以获取报文中携带的目的互联网协议(Internet Protocol, IP)地址,并从路由表中查找与该目的IP地址对应的路由信息,以便根据查找到的路由信息转发报文。路由表为记载路由前缀与路由信息的表项,预先存储在网络设备中。路由前缀用于确定与目的IP地址相匹配的路由信息,例如可以是IP地址的前 n 位, n 为路由前缀的长度。路由信息可以是网络接口的标识,表示网络设备需要通过哪个网络接口转发报文。

[0003] 在查找路由表时,判断目的IP地址与路由前缀是否匹配。例如,假设某个路由前缀的长度为 m ,且目的IP地址的前 m 位与该路由前缀一致,可以认为该目的IP地址与该路由前缀相匹配,对应的路由信息即为该路由前缀对应的路由信息。当目的IP地址同时与两个或两个以上的路由前缀相匹配时,可以根据最长前缀匹配(Longest Prefix Match, LPM)原则确定路由前缀。即,可以将与目的IP地址相匹配的多个路由前缀中长度最长的路由前缀的路由信息作为目的IP地址对应的路由信息。

[0004] 由于查找路由表需要比较路由前缀与目的IP地址是否匹配,网络设备需要查找路由表。显然,调用路由表的速度越快,网络设备转发报文的速度也就越快。目前,为了提高网络设备调用路由表的速度,可以将部分路由前缀和路由信息的对应关系存储在缓存中。但是这种方法往往存在虚假命中的情况。

发明内容

[0005] 本申请实施例提供了一种缓存方法及集成电路,通过在缓存中记录长度大于路由表中与目的IP地址的第一路由前缀与路由信息之间的对应关系,避免了虚假命中的发生。

[0006] 第一方面,本申请实施例提供了一种缓存方法,该方法可以应用于网络设备中的处理器。在执行缓存方法时,处理器可以先获取第一报文,并确定第一报文中携带的目的IP地址。接着,处理器可以从缓存中查找与该目的IP地址相匹配的路由前缀。如果缓存中没有存储与该目的IP地址相匹配的路由前缀,处理器可以根据目的IP地址从其他存储器中查找路由表对应的树结构,从而确定树结构中与该目的IP地址相匹配的前缀节点,即与目的IP地址相匹配的路由前缀在路由表对应的树结构中的节点。当该前缀节点不为树结构中的叶子节点,即该前缀节点具有子节点时,处理器可以将树结构的根节点到尾节点的长度确定为第一前缀匹配长度。其中,尾节点为树结构中查找路径上的最后一个节点。查找路径为路由表对应的树结构中从根节点沿目的IP地址对应的树结构到路由表对应的树结构中叶子节点的路径。在得到第一前缀匹配长度之后,处理器可以根据第一前缀匹配长度后,处理器可以根据目的IP地址和第一前缀匹配长度确定第一路由前缀,并触发缓存存储第一对应关系,第一对应关系为第一路由前缀和与前缀节点相对应的路由信息之间的对应关系。也就

是说,在前缀节点不为叶子节点的情况下,对应关系中的路由前缀和路由表中记载的路由前缀不同。即,当路由表中存在包括且长度大于目的IP地址对应的路由前缀的其他路由前缀时,处理器可以根据第一前缀匹配长度确定新的第一路由前缀,并在缓存中存储该新的路由前缀和路由信息的对应关系。这样,第一路由前缀的长度可以大于存储器路由表中与目的IP地址相匹配的路由前缀的长度。如此,相较于现有技术,可以起到减小虚假命中出现的概率的作用。

[0007] 在一种可能的实现中,第一路由前缀的长度可以大于第一前缀匹配长度。容易理解的是,第一路由前缀的长度越长,出现虚假命中的概率也就越小。当第一路由前缀的长度大于第一前缀匹配长度时,出现虚假命中的概率为0。

[0008] 在一种可能的实现中,当目的IP地址在路由表对应的树结构中尾节点为叶子节点时,第一路由前缀的长度可以等于第一前缀匹配长度。

[0009] 在一种可能的实现中,路由表对应的树结构可以被划分为虚拟树结构和多个子树结构。处理器可以通过虚拟树结构和子树结构确定与目的IP地址匹配的前缀节点。其中,虚拟树结构可以包括路由表对应的树结构中任意多个节点,多个子树结构中每个子树结构的根节点分别对应虚拟树结构中一个节点。具体的,处理器可以先根据目的IP地址查找与路由表对应的虚拟树结构,确定与目的IP地址最长匹配的虚拟前缀。接着,处理器可以根据该虚拟前缀确定根节点对应的路由前缀为该虚拟前缀的子树结构,并从子树结构中查找与目的IP地址匹配的前缀节点。那么,假设第一长度为虚拟书结构的根节点到虚拟前缀的长度,第二长度为从虚拟前缀到子树结构的查找路径上尾节点的长度,那么第一长度与第二长度之和为前述第一前缀匹配长度。如此,将路由表对应的树结构划分为虚拟树结构和多个子树,可以提高路由前缀的查找效率。

[0010] 在一些可能的实现中,处理器还可以存储第二路由前缀和路由信息的对应关系。其中没地儿路由前缀为转发路径上独子节点对应的路由前缀,独子节点为只具有左孩子节点或右孩子节点的子节点。具体的,当与目的IP地址匹配的前缀节点不为叶子节点时,处理器可以根据目的IP地址的转发路径确定转发路径上的独子节点,并根据肚子节点确定第二前缀匹配长度。第二前缀匹配长度等于从根节点到该独子节点的长度。在确定第二前缀匹配长度后,处理器可以根据第二前缀匹配长度确定第二路由前缀,从而存储第二对应关系。

[0011] 在一些可能的实现中,处理器还可以触发缓存存储第一路由前缀的类型。那么在处理器接收到其他报文后,如果该报文的IP地址命中第一路由前缀,处理器可以根据第一路由前缀的类型的指示根据第一对应关系从缓存中获取与第一路由前缀对应的路由信息。

[0012] 在一些可能的实现中,当处理器通过虚拟树结构和子树结构确定前缀节点时,处理器可以触发缓存存储第三对应关系,该第三对应关系为虚拟前缀、虚拟前缀的类型和子树结构的信息之间的对应关系。相似地,如果该报文的IP地址命中虚拟前缀,处理器可以根据虚拟前缀的类型的指示,从子树结构的信息中查找子树结构,从而确定与第一路由前缀对应的路由信息。

[0013] 在一些可能的实现中,考虑到最大长度匹配原则,处理器还可以触发缓存存储第一路由前缀的长度。

[0014] 在一些可能的实现中,所述第一路由前缀的长度等于所述第一前缀匹配长度加1。

[0015] 在一些可能的实现中,所述第一路由前缀的长度小于所述目的IP地址的长度。

[0016] 在一些可能的实现中,所述目的IP地址包括所述第一路由前缀。

[0017] 在一些可能的实现中,所述缓存至少包括以下其中一种:三态内容寻址存储器(Ternary Content Addressable Memory,TCAM)、寄存器、线卡板和晶粒。

[0018] 第二方面,本申请实施例提供了一种集成电路,所述集成电路应用于处理器,包括接口电路和控制电路;其中,所述接口电路,用于获取第一报文,所述第一报文包括目的互联网协议IP地址;所述控制电路,用于从缓存中查找与所述目的IP地址匹配的路由前缀;响应于所述缓存中未存储所述路由前缀,根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;响应于所述前缀节点不为叶子节点,确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度,所述尾节点为所述树结构的查找路径上的最后一个节点;根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。

[0019] 在一些可能的实现中,所述第一路由前缀的长度大于或等于所述第一前缀匹配长度。

[0020] 在一些可能的实现中,所述控制电路,用于根据所述目的IP地址查找与路由表对应的虚拟树结构,确定与所述目的IP地址最长匹配的虚拟前缀;根据所述目的IP地址查找与所述虚拟前缀对应的子树结构,确定与所述目的IP地匹配的前缀节点,所述虚拟前缀为所述子树结构的根节点;所述第一前缀匹配长度等于第一长度和第二长度之和,所述第一长度为所述虚拟树结构的根节点到所述虚拟前缀的长度,所述第二长度为所述虚拟前缀到尾节点之间的长度,所述尾节点为所述子树结构的查找路径上的最后一个节点。

[0021] 在一些可能的实现中,所述控制电路,还用于响应于所述前缀节点不为叶子节点,确定第二前缀匹配长度,所述第二前缀匹配长度等于所述树结构的根节点到独子节点之间的长度,所述独子节点为所述前缀节点到所述尾节点之间的只具有左孩子节点或只具有右孩子节点的节点;根据所述第二前缀匹配长度和所述目的IP地址,确定第二路由前缀,所述第二路由前缀的长度大于所述第二前缀匹配长度;触发所述缓存存储第二对应关系,所述第二对应关系为所述第二路由前缀和所述前缀节点对应的路由信息。

[0022] 在一些可能的实现中,所述控制电路,还用于触发所述缓存存储所述第一路由前缀的类型,所述第一路由前缀的类型用于指示在所述第一路由前缀被命中时,根据所述第一对应关系在所述缓存中获取与所述第一路由前缀对应的路由信息。

[0023] 在一些可能的实现中,所述控制电路,还用于触发所述缓存存储第三对应关系,所述第三对应关系为所述虚拟前缀、所述虚拟前缀的类型以及所述子树结构的信息之间的对应关系,所述虚拟前缀的类型用于指示在所述虚拟前缀被命中时,根据所述子树结构的信息查找所述子树结构。

[0024] 在一些可能的实现中,所述控制电路,还用于触发所述缓存存储所述第一路由前缀的长度。

[0025] 在一些可能的实现中,所述第一路由前缀的长度等于所述第一前缀匹配长度加1。

[0026] 在一些可能的实现中,所述第一路由前缀的长度小于所述目的IP地址的长度。

[0027] 在一些可能的实现中,所述目的IP地址包括所述第一路由前缀。

[0028] 在一些可能的实现中,所述缓存至少包括以下其中一种:TCAM、寄存器、线卡板和晶粒。

[0029] 第三方面,本申请实施例提供了一种缓存装置,所述缓存装置包括获取单元和处理单元。其中,所述获取单元,用于获取第一报文,所述第一报文包括目的互联网协议IP地址。所述处理单元,用于从缓存中查找与所述目的IP地址匹配的路由前缀;响应于所述缓存中未存储所述路由前缀,根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;响应于所述前缀节点不为叶子节点,确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度,所述尾节点为所述树结构的查找路径上的最后一个节点;根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。

[0030] 第四方面,本申请实施例提供了一种芯片,所述芯片包括存储器和处理器,存储器用于存储指令或程序代码,处理器用于从存储器中调用并运行所述指令或程序代码,以执行如前述第一方面所述的缓存方法。

[0031] 第五方面,本申请实施例提供了一种设备,所述设备包括处理器芯片和存储器,存储器用于存储指令或程序代码,处理器芯片用于从存储器中调用并运行所述指令或程序代码,以执行如前述第一方面所述的缓存方法。

[0032] 第六方面,本申请实施例提供了一种计算机可读存储介质,包括指令、程序或代码,当其在计算机上执行时,使得所述计算机执行如前述第一方面所述的缓存方法。

附图说明

[0033] 图1为本申请实施例提供的处理器的一种结构示意图;

[0034] 图2为本申请实施例提供的路由前缀对应的树结构的一种结构示意图;

[0035] 图3-a为本申请实施例提供的网络系统300的一种结构示意图;

[0036] 图3-b为本申请实施例提供的网络设备350的一种结构示意图;

[0037] 图3-c为本申请实施例提供的网络设备350的另一种结构示意图;

[0038] 图3-d为本申请实施例提供的网络设备350的又一种结构示意图;

[0039] 图3-e为本申请实施例提供的网络设备360的一种结构示意图;

[0040] 图3-f为本申请实施例提供的网络设备的芯片370的一种结构示意图;

[0041] 图3-g为本申请实施例提供的虚拟路由器380的一种结构示意图;

[0042] 图4为本申请实施例提供的一种缓存方法的流程示意图;

[0043] 图5为本申请实施例提供的路由表对应的树结构的一种结构示意图;

[0044] 图6为本申请实施例提供的子树结构的一种结构示意图;

[0045] 图7为本申请实施例提供的路由表对应的树结构的一种结构示意图;

[0046] 图8为本申请实施例提供的集成电路800的一种结构示意图;

[0047] 图9为本申请实施例提供的缓存装置900的一种结构示意图;

[0048] 图10为本申请实施例提供的芯片1000的一种结构示意图;

[0049] 图11为本申请实施例提供的一种设备1100的结构示意图;

[0050] 图12为本申请实施例提供的一种设备1200的结构示意图。

具体实施方式

[0051] 下面结合附图对传统技术和本申请实施例提供缓存方法进行介绍。

[0052] 首先对包括缓存的设备的基本结构进行介绍。参见图1,该图为设备100的一种结构示意图。该设备100包括处理器110和存储器120,处理器110包括处理单元111和缓存112。其中,缓存112可以用于从存储器120中加载数据,处理单元111可以用于从缓存112中读取数据,并对数据进行处理。可选地,当处理器110封装为芯片,该芯片可以被称为中央处理器(central processing unit,CPU)。

[0053] 缓存,全称高速缓冲存储器(cache),大多由数据交换速度较快的存储设备组成。在本申请实施方式中,缓存也可以被称为暂时内存(temporary memory)或者暂时存储器(temporary storage)。缓存分别与处理器内部的处理单元和处理器外部的存储器连接,可以存储存储器中存储的部分或全部数据。

[0054] 存储器可以分为外存和内存,外存例如是硬盘(Hard Disk Drive,HDD)等存储设备,内存例如是同步动态随机存取存储器(synchronous dynamic random-access memory,SDRAM)或双倍速度同步动态随机存储器(Double Data Rate SDRAM,DDR)等存储设备。而缓存例如是静态存储器(Static Random Access Memory,SRAM)等存储设备。在与处理器进行数据交换的过程中,SRAM的数据交换速度大于DDR(或SDRAM)的数据交换速度,而DDR的数据交换速度又大于硬盘的数据交换速度。对于当网络设备通过缓存存储部分或全部路由表时,可以先在缓存中创建一个内容为空的路由表。本申请实施例中,缓存中存储的路由表可以被称为缓存路由表,存储器中存储的路由表可以被称为存储器路由表。

[0055] 在处理器确定目的IP地址与存储器路由表中某个路由前缀相匹配时,处理器可以将该路由前缀和路由信息存储在缓存路由表中。这样,当网络设备接收到新报文,且新报文的IP地址与缓存路由表中已存储的路由前缀相匹配时,处理器可以从缓存路由表中获取对应的路由信息。这样,由于处理器与缓存的数据交换速度较快,提高了路由表的查找速度。当IP地址的前n位分别与该路由前缀一致时,可以称该IP地址与该路由前缀相匹配

[0056] 可选地,缓存路由表可以包括缓存标签(Cache Tag)项和缓存数据(Cache Data)项。其中,缓存标签项可以包括路由前缀,缓存数据项可以用于存储路由信息。相应的,缓存标签项与缓存数据项之间的对应关系可以表示路由前缀与路由信息之间的对应关系。

[0057] 考虑到网络设备需要根据最长匹配原则确定与目的IP地址相匹配的路由前缀,缓存标签项还可以包括路由前缀的长度(length,以下简称len)。这样,当目的IP地址同时与缓存中的路由表中多个路由前缀相匹配时,处理器可以将长度最长的路由前缀确定为与目的IP地址相匹配的路由前缀。

[0058] 举例说明。假设网络设备具有网络接口A、网络接口B,且网络设备的存储器中记载了路由前缀0001010*和网络接口A的对应关系,以及路由前缀0*和网络接口B的对应关系。

[0059] 如果网络设备接收到了报文M,且报文M的目的IP地址为0001010010100010。网络设备的处理器可以先从存储器路由表中查找与该目的IP地址相匹配的路由前缀和路由信息。在确定目的IP地址0001010010100010与路由前缀0001011*相匹配之后,处理器可以将路由前缀0001010*、路由前缀0001010*的长度和路由前缀0001010*对应的路由信息(即网络接口A)存储在缓存路由表中。该缓存路由表可以如表1所示。

[0060] 表1

[0061]	Cache tag	Cache data
	0001010*,len=7	A

[0062] 其中,表1的第一行表示路由前缀000101101*与网络接口A相对应,那么目的IP地址第一位为0的报文可以通过网络接口A转发,例如目的IP地址为0001010010100010的报文可以通过网络接口A转发。

[0063] 如果网络设备又接收到报文N,报文N的目的IP地址为0001011010010001。由于该目的IP地址的前7位为0001011,与路由前缀0001010*不一致,网络设备的处理器可以从存储器路由表中查找与目的IP地址0001011010010001相匹配的路由前缀。在确定目的IP地址0001011010010001与路由前缀0*相匹配后,处理器可以将路由前缀0*、路由前缀0*的长度和路由前缀0*对应的路由信息(即网络接口A)存储在缓存路由表中。该缓存路由表可以如表2所示。

[0064] 表1

[0065]	Cache tag	Cache data
	0001010*,len=7	A
	0*,len=1	B

[0066] 如果网络设备又接收到了一条目的IP地址为0001011011110100的报文,由于该目的IP地址的第一位与路由前缀0*一致,且目的IP地址的前七位与路由前缀0001010*不一致,处理器可以确定该报文与路由前缀0*相对应,从而通过网络接口B转发该报文。显然,由于处理器从缓存中获取数据的速度较快,将部分或全部路由前缀和路由信息的对应关系存储在缓存路由表中可以提高网络设备转发报文的速度。

[0067] 但是,由于缓存路由表中的信息是随着报文的转发而逐渐增加的,在这个过程中,缓存路由表和存储器路由表并不一致。那么,如果两个相似度较高的路由中较长的路由前缀未被存入缓存路由表,而较短的路由前缀已被存入缓存路由表,处理器有可能错误地将较短的路由前缀确定为与报文相匹配的路由前缀,从而根据错误的路由信息转发报文,这种情况被称为虚假命中。

[0068] 例如,假设某个报文的目的IP地址与路由前缀X和路由前缀Y相匹配,且路由前缀X已被存储缓存路由表,路由前缀Y未被存入缓存路由表。如果路由前缀X的长度短于路由前缀Y的长度,该报文理论上对应的路由信息应当是路由前缀Y对应的路由信息。但是,处理器在接收到报文后优先从缓存路由表中查找与目的IP地址相匹配的路由前缀。由于缓存路由表包括路由前缀X,处理器可以能够从缓存路由表中找到与目的IP地址相匹配的路由前缀X,从而利用路由前缀X对应的路由信息转发报文。可见,由于缓存路由表与存储器路由表中存储的路由前缀不同,网络设备在转发目的IP地址与多个路由前缀相匹配的报文时可能出现虚假命中的情况。

[0069] 仍然以网络设备接收报文M和报文N为例进行说明。假设网络设备先接收到了报文N,那么缓存路由表可以如表3所示:

[0070] 表3

[0071]	Cache tag	Cache data
	0*,len=1	B

[0072] 如果网络设备再次接收到了报文M,由于报文M的目的IP地址为0001010010100010,其第一位为0,与缓存路由表中记录的路由前缀0*一致,处理器可以认为报文M与路由前缀0*相匹配,从而利用网络接口B转发报文M。显然,报文M实际对应的网络接口应当是网络接口A。由于缓存路由表中包括的信息不全,存在虚假命中的情况。

[0073] 为了解决上述问题,本申请实施例提供了一种缓存方法,该方法可以在缓存中记录长度大于路由表中与目的IP地址的第一路由前缀与路由信息之间的对应关系,避免了虚假命中的发生。

[0074] 以下,首先对本申请涉及的术语进行简单介绍:

[0075] 树结构:树(tree)结构是一种非线性的数据结构。一个树结构可以包括多个节点,每个节点可以用于存储数据。树结构的各个节点之间的连接关系表示各个节点存储的数据之间的关系,例如可以表示数据的顺序关系或从属关系等。树结构中最顶端的节点被称为根节点。一个树结构可以包括多个子树结构,每个子树结构又可以包括一个或多个子树结构。每个子树结构最顶端的节点可以被称为该子树结构的根节点。典型的树结构可以包括二叉树、三叉树和四叉树。

[0076] 在本申请实施例中,路由前缀可以通过二叉树等树结构表示,通过树结构中相邻两层节点之间的连接关系体现路由前缀每一位的具体值。相似地,路由表也可以通过相似的规则通过树结构表示。例如,可以将路由表中每条路由前缀用树结构表示,再将这些树结构的根节点叠加在一起,最终得到的树结构即为路由表的对应的树结构。为了区分路由表中各个路由前缀,可以将每个路由前缀最后一位对应的节点标记为前缀节点。与路由表对应的树结构可以用于表述路由表中前缀节点的分布。

[0077] 在本申请实施例中,路由前缀和路由表可以通过二叉树表示,也可以通过四叉树或十六叉树表示。下面对二叉树进行介绍。需要说明的是,后文为了便于理解以路由表的树结构为二叉树为例进行说明,并不代表本申请全部实施例中路由表的树结构均为二叉树。当路由前缀通过四叉树结构表示时,每个子节点对应路由前缀中的两位数字,例如根节点的四个子节点可以分别表示“00”、“01”、“10”和“11”。如果路由表以树结构表示,该树结构可以拆分为一个虚拟树结构和多个子树结构。关于虚拟树结构的介绍可以参加下文,这里不再赘述。

[0078] 二叉树(binary tree):二叉树是指树结构中节点的度不大于2的有序树。即,二叉树具有一个根节点,且二叉树中任意一个节点最多具有两个子节点,当一个节点具有两个子节点时,左侧的子节点可以被称为左孩子节点,右侧的子节点可以被称为右孩子节点。二叉树中任意一个节点。一个二叉树可以包括多个子二叉树。

[0079] 当路由前缀通过二叉树表示时,从根节点的子节点开始,每一位IP地址对应二叉树的一层。如果IP地址的某一位为0,则二叉树在该层向左侧延伸,如果路由前缀的某一位为1,则二叉树该层向右侧延伸。也就是说,如果路由前缀的第一位为0,那么该路由前缀对应的二叉树的根节点具有左孩子节点,且不具有右孩子节点。

[0080] 例如路由前缀10010*对应的树结构可以如图2所示。其中,节点2为根节点1的右孩子节点,连接表示IP地址的第一位为1;节点3为节点2的左孩子节点,连接表示IP地址的第二位为0;节点4为节点3的左孩子节点,连接表示IP地址的第三位为0;节点5为节点4的右孩子节点,连接表示IP地址的第四位为1;节点6为节点5的左孩子节点,连接表示IP地址的第

五位为0。

[0081] 同理,当路由表通过二叉树进行表示时,路由表可以如图5所示。关于这部分内容可以参见后文,这里不再赘述。

[0082] 独子节点:独子节点为二叉树中仅具有左孩子节点或仅具有右孩子节点的节点。

[0083] 叶子节点:叶子节点为树结构中不具有子节点的节点。

[0084] 前缀节点:前缀节点为路由前缀在路由表对应的树结构中对应的节点。

[0085] 查找路径:查找路径为路由表对应的树结构中从根节点沿目的IP地址对应的树结构到路由表对应的树结构中叶子节点的路径。

[0086] 尾节点:尾节点为查找路径中最后一个节点。

[0087] 本申请实施例提供的方法可以应用于图3-a所示的系统300。该系统300包括网络设备310、网络设备320、网络设备330和网络设备340。

[0088] 在本申请实施例中,网络设备320包括处理器321、缓存322和存储器323。其中,处理器321是网络设备320内部具有数据处理能力的单元,例如可以是处理器内核(core)中用于数据处理的单元,缓存322可以是SRAM等高速存储设备,存储器323可以包括内存和外存中的任意一种或多种。

[0089] 需要说明的是,图3-a中缓存320独立于处理器321,并不表示本申请实施例中提供的缓存必然封装在处理器中。在一些可能的实现方式中,缓存也可以是封装在处理器内部的存储设备。另外,在图1所示实施例中,网络设备320包括一个处理器321和一个缓存322,并不代表本申请实施例中提供的设备只能包括一个处理器和一个缓存。在一些可能的实现方式中,设备可以包括一个或多个处理器,处理器可以包括一个或多个处理单元,以及一个或多个缓存。其中,一个缓存可以对应一个处理单元或多个处理单元,一个处理单元也可以对应一个或多个缓存。本申请实施例对处理器的具体架构不做限定。

[0090] 在一些可能的实现方式中,网络设备可能具有多级缓存结构,即利用多种存储介质构建多个等级不同的缓存,等级越高的缓存交换数据的速度越快,大小越小。那么,对于具有多级缓存结构的计算机,本申请实施例中的缓存可以是一级缓存(一级缓存为等级最高的缓存),存储器可以包括等级低于一级缓存的其他缓存,例如二级缓存、三级缓存等。即,在本申请实施例中,缓存为数据交换速度最快的存储设备,存储器为数据交换速度低于缓存的其他所有存储设备的统称。

[0091] 在本申请实施例中,为了便于路由表项的快速查找,缓存还可以包括三态内容寻址存储器(Ternary Content Addressable Memory,TCAM)、寄存器、线卡版和晶粒等硬件模块。

[0092] 下面先对本申请实施例提供的缓存方法的一些应用场景进行介绍。需要说明的是,本申请实施例提供的应用场景仅仅是一些典型的应用场景,不代表本申请实施例提供的缓存方法只能用于这些应用场景。

[0093] 参见图3-b,该图为本申请实施例提供的一种网络设备的结构示意图。在图3-b所示的实施例中,网络设备350包括处理器351、TCAM352和存储器353。其中,处理器351分别与TCAM352和存储器353连接。存储器353可以用于存储主路由表;TCAM352可以作为缓存用于存储缓存路由表,或根据目的IP地址查找对应的路由信息;处理器351可以用于根据目的IP地址向TCAM352中存储路由前缀和路由信息,或根据TCAM352查找到的路由信息转发报文。

[0094] 可选地,当网络设备具有多个处理器时,每个处理器可以对应各自的TCAM,多个处理器可以对应同一个存储器。具体地,参见图3-c,当网络设备350还包括处理器354时,网络设备350还可以包括TCAM355。其中,处理器351分别与TCAM355和存储器353连接。TCAM355可以作为缓存用于存储处理器354所用的路由表。相似的,TCAM352可以用于存储处理器351所用的路由表。这样,每个处理器都具有相对独立的TCAM,用于存储自身所需的路由表,无需存储全部的路由表。如此,既节省了存储空间,又因为TCAM中存储的内容较少,提高了处理器读取路由表的速度。需要说明的是,在图3-c所示的实施例中,处理器351从TCAM352中读取数据的速度可以不快于处理器351从存储器353中读取数据的速度。

[0095] 关于TCAM352的工作方式可以参见后文表7所在实施例的介绍,这里不再赘述。

[0096] 可选地,前述TCAM所起到的作用可以通过并行比较逻辑电路和寄存器实现。具体地,参见图3-d,图3-b所示实施例中TCAM352的作用可以通过并行比较逻辑电路356和寄存器357实现。其中,处理器351与并行比较逻辑电路356连接,并行比较逻辑电路356与寄存器357连接。其中,寄存器357用于存储路由表,并行比较逻辑电路用于确定与目的IP地址相对应的路由信息。

[0097] 可选地,在图3-b、图3-c和图3-d所示的实施例中,处理器351和/或处理器352可以指处理器内核(core)。

[0098] 参见图3-e,该图为本申请实施例提供的一种网络设备的结构示意图。在图3-e所示的实施例中,网络设备360包括线卡板361、线卡板362、线卡板363、交换网板364和交换网板365。其中,交换网板364分别与线卡板361、线卡板362和线卡板363连接,交换网板365分别与线卡板361、线卡板362和线卡板363连接。

[0099] 在图3-e所示的实施例中,每块线卡板可以包括一个缓存,用于存储该线卡板需要使用的缓存路由表。例如,线卡板361包括缓存361-1,用于存储线卡板361需要使用的缓存路由表;线卡板362包括缓存362-1,用于存储线卡板362需要使用的缓存路由表;线卡板363包括缓存363-1,用于存储线卡板363需要使用的缓存路由表。交换网板可以存储包含全部路由信息的路由表,即前述主路由表。主路由表可以存储在交换网板364和/或交换网板352中,也可以拆分为两部分并分别存储在交换网板364和交换网板352中。可选地,主路由表可以存储在交换网版的缓存中。

[0100] 参见图3-f,该图为本申请实施例提供的一种网络设备的芯片的结构示意图。在图3-f所示的实施例中,芯片370包括晶粒(die)371、晶粒372、晶粒373和晶粒374。其中,晶粒371分别与晶粒372和晶粒374连接,晶粒373分别与晶粒372和晶粒374连接。当然,在一些其他的实现方式中,芯片中的晶粒还可以以其他方式连接。

[0101] 芯片中任意一个晶粒中可以包括缓存。例如晶粒371可以包括缓存371-1,晶粒372可以包括缓存372-1,晶粒373可以包括缓存373-1,晶粒374可以包括缓存374-1。缓存371-1、缓存372-1、缓存373-1和缓存374-1中任意一个或多个缓存可以用于存储缓存路由表,任意一个或多个缓存可以用于存储主路由表。

[0102] 在实际的应用场景中,多个网络设备可以作为一个虚拟路由器转发报文。例如,参见图3-g,该图为本申请实施例提供的一种虚拟路由器的结构示意图。在图3-g所示的实施例中,虚拟路由器380包括网络设备381、网络设备382、网络设备383、网络设备384和网络设备385。其中,网络设备383分别与网络设备381、网络设备382、网络设备384和网络设备385

连接。在虚拟路由器380中,网络设备385可以用于存储主路由表,网络设备381、网络设备382、网络设备384和网络设备385可以分别存储各自所需要的缓存路由表。可选地,网络设备383可以将主路由表存储在网络设备383的缓存中。

[0103] 参见图4,该图4为本申请实施例提供的一种缓存方法的数据交互图。本申请实施例提供的缓存方法包括如下步骤:

[0104] S401:处理器获取第一报文。

[0105] 在本申请实施例中,处理器,可以是网络设备中的处理器。例如,处理器可以是图3-a中网络设备320中处理器321,也可以至处理器321中用于对数据进行处理的处理单元。处理器可以通过网络接口接收其他设备发送的第一报文,例如,当处理器为图3-a中处理器321时,处理器321可以接收网络设备310发送的第一报文。当然,在一些可能的实现中,处理器也可以接收终端设备发送的第一报文。当然,下文所述的缓存可以是网络设备中的缓存,例如可以是图3-a中的缓存322。存储器可以是网络设备中的存储器。例如可以是图3-a中的存储器323。

[0106] 第一报文的的目的设备的IP地址被称为第一报文的的目的IP地址,表示第一报文需要被发送到哪个设备。那么,在接收到第一报文后,处理器可以对第一报文进行解析,从而确定第一报文的的目的IP地址。

[0107] S402:处理器从缓存中查找与目的IP地址匹配的路由前缀。

[0108] 在确定第一报文的的目的地址后,处理器可以从缓存中查找与目的IP地址相匹配的路由前缀,例如处理器可以从缓存路由表中查找与目的IP地址相匹配的路由前缀。当缓存路由表中存在多条与目的IP地址相匹配的路由前缀时,处理器可以将长度最长的路由前缀确定为与目的IP地址相匹配的路由前缀。在一些可能的实现中,处理器还可以将缓存路由表转换为对应的二叉树或多叉树等树结构,并通过树结构确定与目的IP地址相匹配的路由前缀。

[0109] 如果处理器能够从缓存中找到与目的IP地址相匹配的路由前缀,处理器可以从缓存中获得该目的IP地址对应路由信息,从而通过该路由信息转发第一报文。如果处理器无法从缓存中找到与目的IP地址相匹配的路由前缀,说明缓存路由表中未存储目的IP地址对应的路由前缀。那么处理器可以从存储器中查找与目的IP地址匹配的路由前缀和路由信息,并根据查找到的路由前缀和路由信息更新缓存路由表。

[0110] 可选地,更新缓存路由表之前,处理器可以先判断缓存路由表中是否存在用于存储路由前缀和路由信息的存储空间。如果缓存的存储空间已满,处理器可以不更新缓存路由表,或者将缓存中已存储的路由前缀和路由信息删除,从而利用新的存储空间存储新的路由前缀和路由信息。

[0111] 下面介绍在从缓存中不包括与目的IP地址相匹配的路由前缀的情况下,处理器更新缓存路由表的方法。

[0112] S403:响应于所述缓存未存储所述路由前缀,所述处理器从存储器中查找与所述目的IP地址相对应的前缀节点和路由信息。

[0113] 如果缓存中未存储与目的IP地址相对应的路由前缀,处理器可以从存储器中查找与目的IP地址相对应的路由前缀和路由信息。可选地,处理器可以通过与路由表对应的树结构确定与目的IP地址相对应的路由前缀。那么,处理器可以从与路由表对应的树结构中

确定与目的IP地址对应的前缀节点,并获取该前缀节点的路由信息。

[0114] 下面介绍处理器确定前缀节点的方法。

[0115] 假设存储器中存储的路由表如表4所示

[0116] 表5

路由前缀	路由信息
0*	网络接口X
11*	网络接口A
0000*	网络接口B
0001010*	网络接口C
00010111*	网络接口D
0001011111*	网络接口E
00010110000*	网络接口F
00010110010*	网络接口G

[0118] 该路由表对应的属性结构的示意图可以如图5所示。其中,图5中各个节点内的数字表示该节点的编号。那么,其中节点2为对应路由前缀0*的前缀节点;节点2为前缀节点,对应路由前缀11*;节点7为对应路由前缀0000*的前缀节点;节点11为对应路由前缀0001010*的前缀节点;节点14为对应路由前缀00010111*的前缀节点;节点19为对应路由前缀0001011111*的前缀节点;节点20为对应路由前缀00010110000*的前缀节点;节点21为对应路由前缀00010110010*的前缀节点。

[0119] 如果第一报文的目的IP地址为0001011010010001,由于该目的IP地址仅与路由前缀0*相匹配,处理器可以确定与目的IP地址对应的前缀节点为节点2,对应的路由信息为网络接口X。如果第一报文的目的IP地址为0001010010100010,由于该目的IP与路由前缀0*和路由前缀0001010*相匹配,且路由前缀0001010*的长度大于路由前缀0*的长度,处理器可以确定与目的IP地址对应的前缀节点为节点11,对应的路由信息为网络接口C。

[0120] 在实际的应用场景中,路由表中存储的对应关系的数量往往较多,对应的树结构也较为复杂。那么,为了快速找到与目的IP地址对应的前缀节点,可以将路由表对应的树结构拆分为多个子树结构,这些子树结构的根节点组成的树结构被称为虚拟树结构。每个子树结构的根节点对应一个虚拟前缀。子树结构中各个节点代表的路由前缀可以根据虚拟前缀和该节点在虚拟树结构中的位置确定。例如,假设某个子树结构的虚拟前缀为1000*,那么该子树结构的根节点的右孩子节点对应的路由前缀即为10001*。

[0121] 在查找目的IP地址对应的路由前缀时,处理器可以先从虚拟树结构中查找与目的IP地址最长匹配的虚拟前缀,从与目的IP地址相匹配的虚拟前缀中确定长度最长的虚拟前缀。接着,处理器可以从该虚拟前缀对应的子树结构中确定与目的IP地址匹配的前缀节点。如果虚拟前缀对应的子树中不存在与目的IP地址相匹配的前缀节点,处理器可以将该虚拟前缀对应的前缀节点确定为目的IP地址的前缀节点。

[0122] 举例说明。对于图5所示的二叉树,可以将其划分为三个子树结构。这三个子树结构的根节点分别为根节点1、节点6和节点12,虚拟前缀分别为*、00*和0001011*。那么,当用表格的形式表示该虚拟树结构时,该表格可以如表6所示。三个子树可以如图6所示。

[0123] 表6

[0124]	虚拟前缀	前缀长度	子树信息
	*	0	子树0
	00*	2	子树1
	0001011*	7	子树2

[0125] 其中,表6的第一行表示长度为0的虚拟前缀为*与子树0对应,即目的IP地址不与表6中其他虚拟前缀匹配的情况下,处理器可以从子树0中确定与目的IP地址确定的前缀节点。表6的第二行表示长度为2的虚拟前缀00*与子树1对应。表6的第三行表示长度为7的虚拟前缀0001011*与子树2对应。

[0126] 在接收到目的IP地址为0001010010100010的第一报文时,由于该目的IP地址与虚拟前缀00*最长匹配,处理器可以从子树1中查找与目的IP地址匹配的前缀节点。由于子树1中节点11的路由前缀为0001010*与目的IP地址相匹配,且节点11为前缀节点,那么处理器可以将路由前缀0001010*确定为与目的IP地址0001010010100010相匹配的路由前缀,节点11为对应的前缀节点。

[0127] 在接收到目的IP地址为0001011010010001的第一报文时,由于该目的IP地址与虚拟前缀00*最长匹配,处理器可以从子树1中查找与目的IP地址匹配的前缀节点。由于子树1中不具有目的IP地址相匹配的前缀节点,处理器可以将虚拟前缀00*对应的前缀节点确定为目的IP地址对应的前缀节点。那么处理器可以确定与目的IP地址0001011010010001匹配的前缀节点为节点2,匹配的路由前缀为0*。

[0128] S404:所述处理器根据所述前缀节点和路由信息确定对应关系集合。

[0129] 在确定与目的IP地址相对应的前缀节点和路由信息后,处理器可以根据前缀节点和路由信息确定对应关系集合。其中,对应关系集合可以包括前缀节点的路由前缀与路由信息之间的对应关系,也可以包括第一路由前缀与路由信息之间的对应关系。在本申请实施例中,第一路由前缀与路由信息之间的对应关系可以被称为第一对应关系。具体地,当与目的IP地址匹配的前缀节点为叶子节点时,对应关系可以包括前缀节点的路由前缀与路由信息之间的对应关系;当与目的IP地址匹配的前缀节点不为叶子节点时,对应关系可以包括第一对应关系。有关第一路由前缀的介绍可以参见下文。

[0130] 下面分别对这两种情况进行介绍。

[0131] 在第一种可能的实现中,与目的IP地址匹配的前缀节点为叶子节点,即查找路径的尾节点为前缀节点,那么处理器可以确定对应关系为该前缀节点的路由前缀和路由信息之间的对应关系。

[0132] 由于叶子节点为不具有子节点的节点,该前缀节点也不具有子节点。也就是说,在路由表所记载的一个或多个路由前缀中,不存在前n位与该路由前缀一致,且长度大于n的其他路由前缀。其中,n为该路由前缀的长度。

[0133] 仍以图5为例进行说明,假设目的IP地址为0001010010100010,那么与该目的IP地址相匹配的前缀节点为节点11。节点11作为叶子节点,并没有子节点。如果节点11具有左孩子节点或右孩子节点,那么节点11的左孩子节点对应的路由前缀为00010100*,节点11的右孩子节点对应队列路由前缀为00010101*。由于这两个子节点并不存在,说明路由表不包括00010100*和00010101*两个路由前缀,自然也不包括其他前7位为0001010且长度大于7的路由前缀。

[0134] 可见,由于路由表中不存在前n位与前缀节点的路由前缀一致,且长度大于n的其他路由前缀(n为该路由前缀的长度)。自然,与该前缀节点的路由前缀相匹配的目的IP地址不可能与其他长度大于n的路由前缀匹配。因此,在前缀节点的路由前缀存储到缓存路由表之后,不会出现虚假命中的情况,即不会出现目的IP地址本应匹配一个长度大于n的路由前缀,却因缓存路由表中存储的路由前缀不全匹配到该路由前缀的情况。

[0135] 那么,处理器可以从存储器中获取前缀节点的路由前缀对应的路由信息,以便在后续步骤中将该路由前缀和路由信息之间的对应关系存储到缓存。

[0136] 在第二种可能的实现中,与目的IP地址匹配的前缀节点不为叶子节点,即查找路径的尾节点不为前缀节点,那么对应关系集合可以包括第一对应关系。在确定第一对应关系时,处理器可以先确定第一路由前缀。其中,第一路由前缀与目的IP地址相匹配,且其长度是根据第一前缀匹配长度确定的,第一前缀匹配长度为从树结构的根节点到查找路径的尾节点的长度,即查找路径的尾节点到根节点之间的节点的数量加1。

[0137] 在本申请实施例中,第一路由前缀的长度可以大于第一前缀匹配长度,也可以等于第一前缀匹配长度,还可以小于第一前缀匹配长度。下面分别进行介绍。

[0138] 首先介绍第一路由前缀大于第一前缀匹配长度的情况。

[0139] 在路由表对应的树结构中,由于目的IP地址对应的前缀节点不为叶子节点,说明该前缀节点的子树中包括其他前缀节点,即缓存路由表中存在包括目的IP地址对应的路由前缀的其他路由前缀,且这些路由前缀与目的IP地址并不匹配。因此,为了避免虚假命中,处理器可以将路由表对应的树结构以外,且与目的IP地址相匹配的路由前缀确定为第一路由前缀。

[0140] 具体地,处理器可以从目的IP地址中选取长度大于第一前缀匹配长度的部分作为第一路由前缀。其中,第一前缀匹配长度为从树结构的根节点到目的IP地址的查找路径的尾节点的长度,即该尾节点对应的路由前缀的长度。由于目的IP地址对应的前缀节点不为叶子节点,查找路径中的尾节点不为前缀节点,说明目的IP地址对应的树结构与转发表对应的树结构从尾节点的位置开始出现差异。因此,从目的IP地址中选取长度大于第一前缀匹配长度的部分作为第一路由前缀,可以确保第一路由前缀对应的尾节点在转发表对应的树结构以外,即第一路由前缀与路由表对应的树结构中任意一条路由前缀均不重合。这样,路由表中没有记载第一路由前缀,自然也不可能记载长度长于第一路由前缀且包括第一路由前缀的其他路由前缀。也就是说,假设第一路由前缀的长度为n,那么路由表中不存在前n位与目的IP地址一致,且长度大于n的其他路由前缀。自然,包括第一路由前缀的任意一个IP地址不可能与目的路由表中长度大于n的路由前缀相匹配。因此,在前缀节点的路由前缀存储到缓存路由表之后,不会出现虚假命中的情况,即不会出现目的IP地址本应匹配一个长度大于n的路由前缀,却因缓存路由表中存储的路由前缀不全匹配到该路由前缀的情况。

[0141] 仍以图5为例进行说明。假设目的IP地址为0001011010010001,与该目的IP地址相匹配的路由前缀为0*,该目的IP地址在树结构中的查找路径为“1-2-4-6-8-9-10-12-13”。其尾节点为节点13,第一前缀匹配长度为8。将该目的IP地址在路由表以外的部分通过树结构表示,得到的树结构可以如图7所示。在图5所示实施例的基础上,图7所示的树结构新增了节点22、节点23、节点24、节点25、节点26、节点27、包节点28和节点29。这些节点分别对应目的IP地址的第9-16位。由于这些节点在路由表对应的树结构中并不存在,这些节点又可

以被称为虚拟节点。

[0142] 由于虚拟节点位于目的IP地址对应的树结构在路由表的树结构以外的部分,虚拟节点对应的路由前缀大于第一前缀匹配长度。因此,在确定第一路由前缀时,处理器可以将任意一个虚拟节点对应的路由前缀确定为第一路由前缀。例如处理器可以将直接与尾节点相连的虚拟节点对应的路由前缀确定为第一路由前缀。在图6所示的实施例中,处理器可以将节点22对应的路由前缀000101101*确定为第一路由前缀,该第一路由前缀的长度为第一前缀匹配长度加一。当然,在一些可能的实现方式中,处理器还可以将其他任意一个虚拟节点对应的路由前缀确定为第一路由前缀。例如可以将节点24对应的路由前缀00010110100*确定为第一路由前缀。

[0143] 根据前文介绍可知,在确定与目的IP地址匹配的前缀节点时,处理器可以先根据虚拟树结构确定目的IP地址对应的虚拟前缀和子树结构,再根据子树结构确定与目的IP地址匹配的前缀节点。相似的,处理器也可以通过虚拟树结构确定第一路由前缀。

[0144] 具体地,假设第一长度为虚拟树结构的根节点到与目的IP地址对应的虚拟前缀的长度,即与目的IP地址匹配的虚拟前缀的长度,第二长度为该虚拟前缀到查找路径的尾节点之间的长度,即与目的IP地址对应的子树结构中根节点到尾节点的长度。由于目的IP地址的查找路径为虚拟树结构的根节点到虚拟前缀对应的子树结构的根节点,再从该子树结构的根节点到尾节点,那么第一长度与第二长度之和为第一前缀匹配长度。在确定第一路由前缀时,处理器可以将虚拟前缀确定为第一路由前缀的前半部分,再从目的IP地址中选择长度大于第二长度的部分确定为第一路由前缀的后半部分。

[0145] 以图6所示的实施例为例进行说明,假设目的IP地址为0001011010010001,那么与目的IP地址匹配的虚拟前缀为节点12对应的虚拟前缀,即0001011*。处理器可以将第一路由前缀的前7位确定为0001011*。由于目的IP地址在子树2中的尾节点为节点13,节点12到节点13的长度为1,即第二长度为1,处理器可以从目的IP地址的第8位开始截取长度大于1的部分作为第一路由前缀的后半部分。例如,处理器可以将目的IP地址的第8位和第9位作为第一路由前缀的后半部分,得到的第一路由前缀为000101101*。

[0146] 下面介绍第一路由前缀的长度等于第一前缀匹配长度的情况。

[0147] 在实际的应用场景中,由于路由表的修改与树结构的修改可能并不同步,树结构中的叶子节点可能并不是前缀节点。那么,当与目的IP地址对应的前缀节点不为叶子节点,且目的IP地址在树结构中的查找路径的尾节点为叶子节点,处理器可以将该叶子节点对应的路由前缀确定为第一路由前缀。这样,第一路由前缀的长度还可以等于第一前缀匹配长度。

[0148] 具体的,在路由表中某个路由前缀被删除以后,树结构中该路由前缀对应的节点(后称节点A)可能还未被删除。但是由于路由表已被修改,节点A并不为前缀节点。如果节点A不具有子节点,那么节点A即为树结构中不为前缀节点的叶子节点。

[0149] 例如,在图5所示实施例中,假设路由前缀0000*从路由表中被删除,但是树结构中可能仍然保留有节点7。那么该节点7为非前缀节点的叶子节点。

[0150] 在接收到第一报文后,如果第一报文的IP地址为0000101010101010,那么与该目的IP地址相匹配的路由前缀为0*,对应的前缀节点为节点2,查找路径的尾节点为节点7,节点7的路由前缀为0000*,第一前缀匹配长度为4。那么,处理器可以将节点7的路由前缀

0000*确定为第一路由前缀。可见,在这种情况下,第一路由前缀的长度与第一前缀匹配长度一致。

[0151] 下面介绍第一路由前缀的长度小于第一前缀匹配长度的情况。

[0152] 在一些可能的实现中,第一路由前缀的长度也可以小于第一前缀匹配长度,且大于前缀节点对应的路由前缀的长度,即存储器路由表中与目的IP地址相匹配的路由前缀的长度。也就是说,第一路由前缀在树结构中对应的节点为查找路径中前缀节点下层,尾节点上层的节点。这样,虽然第一路由前缀的长度小于第一前缀匹配长度,但是由于第一路由前缀的长度仍然大于存储器路由表中与目的IP地址相匹配的路由前缀的长度,那么相较于现有技术,将第一路由前缀和路由信息的对应关系存储到缓存,仍然可以起到减小虚假命中出现的概率的作用。容易理解的是,第一路由前缀的长度越长,出现虚假命中的概率也就越小。当第一路由前缀的长度大于第一前缀匹配长度时,出现虚假命中的概率为0。

[0153] 举例说明。假设目的IP地址为0001011010010001,那么与目的IP地址匹配的前缀节点为节点2,对应的路由前缀的长度为1,尾节点为节点12,对应的第一前缀匹配长度为8。那么,处理器可以从目的IP地址的前m位作为第一路由前缀,其中,m为大于1小于8的正整数。例如,处理器可以将路由前缀00010110*确定为第一路由前缀。

[0154] 上面介绍了处理器确定第一路由前缀的方法,在确定第一路由前缀后,处理器可以根据与目的IP地址相匹配的路由前缀,从存储器存储的路由表中查找路由信息,进而将该路由信息和第一路由前缀之间的对应关系确定为第一对应关系。

[0155] 可选地,当处理器通过虚拟树结构确定与目的IP地址相匹配的前缀节点时,对应关系集合中还可以包括第三对应关系。其中,第三对应关系可以包括虚拟前缀、虚拟前缀的类型和子树结构的信息。其中,虚拟前缀为与目的IP地址相匹配的虚拟前缀,子树结构的信息用于表示目的IP地址对应哪个子树。虚拟前缀的类型用于指示在虚拟前缀被命中时,根据子树结构的信息查找对应的子树结构。

[0156] 也就是说,在接收到新的报文后,如果该报文的目的是IP地址与缓存中存储的虚拟前缀相匹配,处理器可以根据虚拟前缀的类型需要查找与虚拟前缀对应的子树结构,从而根据第三对应关系确定与虚拟前缀对应的子树结构,进而根据子树结构确定与目的IP地址相匹配的路由前缀。

[0157] 为了提高查找效率,在一些可能的实现方式中,对应关系集合还可以包括第二对应关系。第二对应关系为第二路由前缀和路由信息之间的对应关系。其中,路由信息为第一报文的目的是IP地址对应的路由前缀的路由信息。

[0158] 下面介绍处理器确定第二路由前缀的方法。

[0159] 在确定目的IP地址对应的前缀节点不为叶子节点后,处理器可以根据树结构确定查找路径中前缀节点以后的上所有的独子节点,即仅具有左孩子节点或右孩子节点的节点。接着,处理器可以将该独子节点对应的路由前缀的长度确定为第二前缀匹配长度。即第二前缀匹配长度等于树结构的根节点到该独子节点之间的长度。接着,处理器可以根据第二前缀匹配长度和目的IP地址确定第二路由前缀。假设第二前缀匹配长度为n,处理器可以将目的IP地址的前n为确定为第二路由前缀。

[0160] 由于独子节点为目的IP地址的查找路径中前缀节点后的节点,说明任意一个独子节点对应的路由前缀均不在缓存路由表中,且与独子节点相匹配的报文对应的前缀节点均

为目的IP地址的前缀节点。因此,与这些独子节点相匹配的IP地址的路由前缀均为前缀节点的路由前缀。又因为独子节点位于目的IP地址的查找路径上,目的IP地址中包括每个独子节点的路由前缀。因此,根据独子节点确定第二前缀匹配长度,再根据第二前缀匹配长度从目的IP地址中选取第二路由前缀,得到的第二路由前缀与目的IP地址的前缀节点对应的路由信息相匹配。因此,处理器可以记录第二路由前缀与路由信息之间的对应关系记录为第二对应关系,并加入对应关系集合以便缓存存储。

[0161] 仍以图5所示实施例为例进行说明。在接收到了目的IP地址为0001011010010001的第一报文后,处理器可以确定前缀节点为节点2。尾节点为节点13,前缀节点与尾节点之间的独子节点包括节点4、节点8和节点9。其中节点4对应的第二前缀匹配长度为2、节点8的第二前缀匹配长度为4、节点9的第二前缀匹配长度为5。那么,处理器可以将目的IP地址的前2位、前4位和前5位分别确定为第二路由前缀,即路由前缀00*,路由前缀0001*和路由前缀00010*均为第二路由前缀。接着,处理器可以将这三条路由前缀和路由前缀0*对应的路由信息“网络接口X”之间的对应关系确定为第二对应关系。

[0162] S405:所述处理器触发所述缓存存储所述对应关系集合。

[0163] 在得到对应关系集合后,处理器可以触发缓存存储对应关系集合。可选地,考虑到最长前缀匹配原则,处理器还可以触发缓存存储第一路由前缀的长度。

[0164] 另外,当对应关系集合中包括第一对应关系时,处理器还可以触发缓存存储第一路由前缀的类型。该第一路由前缀的类型用于指示处理器在第一路由前缀被命中时,根据第一对应关系在缓存中获取与第一路由前缀对应的路由信息。也就是说,在接收到新的报文后,如果该报文的IP地址与缓存中存储的第一路由前缀相匹配,处理器可以根据第一路由前缀的类型从缓存路由表中查找第一路由前缀对应的路由信息,无需从存储器中查找第一路由前缀对应的路由信息。

[0165] 下面对TCAM目的IP地址查找对应的路由信息的过程进行介绍。

[0166] 假设作为缓存路由表的TCAM中存储了路由前缀000101101*和网络接口X之间的对应关系,以及路由前缀0001011*和路由前缀0000*和网络接口B之间的对应关系。那么TCAM中存储的表项可以如表7所示。

[0167] 表7

[0168]

TCAM key	TCAM mask	TCAM AD
000101101*	1111111110000000	网络接口X
0000*	1111000000000000	网络接口B

[0169] 其中,表7中表头的第一项为TCAM掩码(TCAM key),用于记录待匹配的路由前缀。表7中表头的第一项为TCAM掩码(TCAM mask),用于检测目的IP地址与TCAM关键词的匹配程度。表7中表头的第一项为TCAM关联数据(Associated Data,AD),用于记录与TCAM关键词对应的路由信息。

[0170] 表7中第一行表示TCAM可以通过TCAM掩码1111111110000000判断报文的IP地址是否与TCAM关键词000101101*匹配,并在确定目的IP地址TCAM关键词000101101*匹配后通过网络接口X发送。表7中第二行表示TCAM可以通过TCAM掩码1111000000000000判断报文的IP地址是否与TCAM关键词0000*匹配,并在确定目的IP地址TCAM关键词0000*匹配后通过网络接口B发送。

[0171] 在判断报文的目的IP地址是否与TCAM关键词匹配时,TCAM可以将TCAM掩码与目的IP地址逐位进行“与”运算,并将TCAM关键词和TCAM掩码也逐位进行“与”运算。最后再逐位比较两次运算得到的结果是否完全一致。若两次运算得到的结果完全一致,TCAM可以确定目的IP地址与TCAM关键词相匹配,从而根据TCAM关联数据中记载的路由信息发送该报文。

[0172] 举例说明。假设网络设备接收到了目的IP地址为0001011010010001的报文,处理器可以将目的IP地址0001011010010001分别与TCAM关键词1111111110000000和TCAM掩码1111000000000000进行逐位“与”运算。

[0173] 在进行逐位“与”运算时,如果目的IP地址与TCAM掩码在某一位上的值均为1,那么得到结果为1,若否,得到的结果为0。例如,目的IP地址的第一位为0,TCAM掩码1111111110000000的第一位为1,二者进行“与”运算得到的结果为0。目的IP地址的第四位为1,TCAM掩码1111111110000000的第四位为1,二者进行“与”运算得到的结果为1。

[0174] 依此类推,目的IP地址0001011010010001与TCAM掩码1111111110000000进行逐位“与”运算得到的结果为0001011010000000。目的IP地址0001011010010001与TCAM掩码1111000000000000进行逐位“与”运算得到的结果为0001000000000000。同样,TCAM关键词000101101*与TCAM掩码1111111110000000进行逐位“与”运算得到的结果为0001011010000000。TCAM关键词0000*与TCAM掩码1111111110000000进行逐位“与”运算得到的结果为0000000000000000。

[0175] 接着,由于目的IP地址运算得到的结果与TCAM关键词000101101*运算得到的结果完全一致,TCAM可以确定目的IP地址与路由前缀000101101*相匹配,从而通过网络接口X发送该报文。由于根据目的IP地址运算得到的结果与根据TCAM关键词0000*运算得到的结果不一致,TCAM可以确定目的IP地址与路由前缀0000*不匹配,从而不通过网络接口B发送该报文。

[0176] 参见图8,本申请实施例还提供了一种集成电路800,该集成电路800可以实现图4所示实施例中处理器的功能。该集成电路包括接口电路801和控制电路802。所述控制电路801与接口电路802通过通信线路803连接。接口电路802用于使得芯片800通过通信链路,与其它设备相连。其中,所述接口电路601用于实现图4所示实施例中的S401、S402、S403和S405,所述控制电路802用于实现图4所示实施例中S402、S403、S404和S405。

[0177] 具体的,所述接口电路801,用于获取第一报文,所述第一报文包括目的互联网协议IP地址。

[0178] 所述控制电路802,用于从缓存中查找与所述目的IP地址匹配的路由前缀;响应于所述缓存中未存储所述路由前缀,根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;响应于所述前缀节点不为叶子节点,确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度,所述尾节点为所述树结构的查找路径上的最后一个节点;根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。

[0179] 具体执行过程请参考上述图4所示实施例中相应步骤的详细描述,这里不再一一赘述。

[0180] 可选地,本申请实施例中提供的集成电路800还可以包括缓存。该集成电路800中

的接口电路801和控制电路802可以为一个或多个。

[0181] 示例性的,该集成电路600可以是FPGA,可以是ASIC,还可以是系统芯片(system on chip,SoC),还可以是CPU,还可以是NP,还可以是数字信号处理电路(digital signal processor,DSP),还可以是微控制器(micro controller unit,MCU),还可以是可编程控制器(programmable logic device,PLD)或其他集成芯片。

[0182] 参见图9,本申请实施例还提供了一种装置900,该缓存装置900可以实现图4所示实施例中处理器的功能。该缓存装置900包括获取单元901和处理单元902。其中,获取单元901用于实现图4所示实施例中的S401,处理单元402用于实现图4所示实施例中的S402、S403、S404和S405。

[0183] 具体的,获取单元901,用于获取第一报文,所述第一报文包括目的互联网协议IP地址。

[0184] 处理单元902,用于从缓存中查找与所述目的IP地址匹配的路由前缀;响应于所述缓存中未存储所述路由前缀,根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;响应于所述前缀节点不为叶子节点,确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度,所述尾节点为所述树结构的查找路径上的最后一个节点;根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。

[0185] 具体执行过程请参考上述图4所示实施例中相应步骤的详细描述,这里不再一一赘述。

[0186] 需要说明的是,本申请实施例中对单元的划分是示意性的,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式。本申请实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。例如,上述实施例中,获取单元和处理单元可以是同一个单元,也不同的单元。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0187] 参见图10,本申请实施例还提供了一种芯片1000,该芯片1000可以实现图4所示实施例中处理器的功能。该芯片包括存储器1001和处理器1002。其中,存储器1001用于存储指令或程序代码,处理器1002用于从存储器1001中调用并运行指令或程序代码,以实现图4所示实施例中的S401、S402、S403、S404和S405。

[0188] 可选地,所述处理器1002可以是图8所示集成电路800。所述处理器1002可以包括缓存。当处理器1002不包括缓存时,所述存储器1001可以包括缓存。

[0189] 图11是本申请实施例提供的一种设备1100的结构示意图。上文中处理器所在的设备可以通过位于图11所示的设备来实现。参见图11,该设11900包括至少一个处理器芯片1101和至少一个存储器1102。可选地,该设备1100还可以包括通信总线1103以及至少一个网络接口1104。

[0190] 处理器芯片1101可以是一个通用中央处理器(central processing unit,CPU)、特定应用集成电路(application-specific integrated circuit,ASIC)或一个或多个用于控制本申请方案程序执行的集成电路(integrated circuit,IC)。该处理器芯片901可以用于实现本申请实施例中提供的数据缓存方法。

[0191] 比如,当图1中的网络设备通过图11所示的设备来实现时,该处理器可以用于,获取第一报文,所述第一报文包括目的互联网协议IP地址;从缓存中查找与所述目的IP地址匹配的路由前缀;响应于所述缓存中未存储所述路由前缀,根据所述目的IP地址查找与路由表对应的树结构,确定与所述目的IP地址匹配的前缀节点;响应于所述前缀节点不为叶子节点,确定第一前缀匹配长度,所述第一前缀匹配长度等于所述树结构的根节点到尾节点之间的长度,所述尾节点为所述树结构的查找路径上的最后一个节点;根据所述第一前缀匹配长度和所述目的IP地址,确定第一路由前缀;触发所述缓存存储第一对应关系,所述第一对应关系为所述第一路由前缀和与所述前缀节点对应的路由信息。

[0192] 可选地,处理器芯片1101可以是图10所示实施例中芯片1000,也可以是图8所示实施例中集成电路800。当处理器芯片1001不包括缓存时,所述存储器1102可以包括缓存、内存和外存;当处理器芯片1101包括缓存时,所述存储器1102可以包括内存和外存。

[0193] 存储器1102可以是只读存储器(read-only memory,ROM)或可存储静态信息和指令的其它类型的静态存储设备,存储器902还可以是随机存取存储器(random access memory, RAM)或者可存储信息和指令的其它类型的动态存储设备,也可以是只读光盘(compact disc read-only Memory,CD-ROM)或其它光盘存储、光碟存储(包括压缩光碟、激光光碟、光碟、数字通用光碟、蓝光光碟等)、磁盘存储介质或者其它磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其它介质,但不限于此。存储器1102可以是独立存在,通过通信总线1103与处理器1101相连接。存储器1102也可以和处理器1101集成在一起。可选地,存储器1102可以包括缓存。

[0194] 可选地,存储器1102用于存储执行本申请方案的程序代码或指令,并由处理器1101来控制执行。处理器1101用于执行存储器1102中存储的程序代码或指令。程序代码中可以包括一个或多个软件模块。可选地,处理器1101也可以存储执行本申请方案的程序代码或指令,在这种情况下处理器1101不需要到存储器1102中读取程序代码或指令。

[0195] 可选地,存储器1102可以用于实现图1所示的设备中存储器120的作用。

[0196] 通信总线1103用于在处理器芯片1101、网络接口1104和存储器1102之间传送信息。

[0197] 网络接口1104可以为收发器一类的装置,用于与其它设备或通信网络通信,通信网络可以为以太网、无线接入网(RAN)或无线局域网(wireless local area networks, WLAN)等。在本申请实施例中,网络接口904可以用于接收分段路由网络中的其他节点发送的报文,也可以向分段路由网络中的其他节点发送报文。网络接口1104可以为以太网接口(ethernet)接口、快速以太网(fast ethernet,FE)接口或千兆以太网(gigabit ethernet,GE)接口等。

[0198] 在具体实现中,作为一种实施例,设备1100可以包括多个处理器芯片,例如图9中所示的处理器芯片1101和处理器芯片1105。这些处理器芯片中的每一个可以是一个单核(single-CPU)处理器,也可以是一个多核(multi-CPU)处理器。可选地,图4所示的缓存方法中不同步骤可以由不同的处理器芯片执行。这里的处理器芯片可以指一个或多个设备、电路、和/或用于处理数据(例如计算机程序指令)的处理核。

[0199] 图12是本申请实施例提供的一种设备1200的结构示意图。图1中的设备100可以通过图12所示的设备来实现。参见图12所示的设备结构示意图,设备1200包括主控板和一个

或多个接口板。主控板与接口板通信连接。主控板也称为主处理单元(main processing unit,MPU)或路由处理卡(route processor card),主控板包括CPU和存储器,主控板负责对设备1000中各个组件的控制和管理,包括路由计算、设备管理和维护功能。接口板也称为线处理单元(line processing unit,LPU)或线卡(line card),用于接收和发送报文。在一些实施例中,主控板与接口板之间或接口板与接口板之间通过总线通信。在一些实施例中,接口板之间通过交换网板通信,在这种情况下设备1200也包括交换网板,交换网板与主控板、接口板通信连接,交换网板用于转发接口板之间的数据,交换网板也可以称为交换网板单元(switch fabric unit,SFU)。接口板包括CPU、存储器、转发引擎和接口卡(interface card,IC),其中接口卡可以包括一个或多个网络接口。网络接口可以为Ethernet接口、FE接口或GE接口等。CPU与存储器、转发引擎和接口卡分别通信连接。存储器用于存储转发表。转发引擎用于基于存储器中保存的转发表转发接收到的报文,如果接收到的报文的地址为设备1200的IP地址,则将该报文发送给主控板或接口板的CPU进行处理;如果接收到的报文的地址不是设备1200的IP地址,则根据该目的地查转发表,如果从转发表中查找到该目的地址对应的下一跳和出接口,将该报文转发到该目的地址对应的出接口。转发引擎可以是网络处理器(network processor,NP)。接口卡也称为子卡,可安装在接口板上,负责将光电信号转换为数据帧,并对数据帧进行合法性检查后转发给转发引擎处理或接口板CPU。在一些实施例中,CPU也可执行转发引擎的功能,比如基于通用CPU实现软转发,从而接口板中不需要转发引擎。在一些实施例中,转发引擎可以通过ASIC或现场可编程门阵列(field programmable gate array,FPGA)实现。在一些实施例中,存储转发表的存储器也可以集成到转发引擎中,作为转发引擎的一部分。

[0200] 可选地,该芯片系统中的处理器可以为一个或多个。该处理器可以通过硬件实现也可以通过软件实现。当通过硬件实现时,该处理器可以是逻辑电路、集成电路等。当通过软件实现时,该处理器可以是一个通用处理器,通过读取存储器中存储的软件代码来实现。可选地,该芯片系统中的存储器也可以为一个或多个。该存储器可以与处理器集成在一起,也可以和处理器分离设置,本申请并不限定。示例性的,存储器可以是非瞬时性处理器,例如只读存储器ROM,其可以与处理器集成在同一块芯片上,也可以分别设置在不同的芯片上,本申请对存储器的类型,以及存储器与处理器的设置方式不作具体限定。

[0201] 示例性的,该芯片系统可以是FPGA,可以是ASIC,还可以是系统芯片(system on chip,SoC),还可以是CPU,还可以是NP,还可以是数字信号处理电路(digital signal processor,DSP),还可以是微控制器(micro controller unit,MCU),还可以是可编程控制器(programmable logic device,PLD)或其他集成芯片。

[0202] 应理解,上述方法实施例中的各步骤可以通过处理器中的硬件的集成逻辑电路或者软件形式的指令完成。结合本申请实施例所公开的方法步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0203] 本申请实施例还提供了一种计算机可读存储介质,包括指令,当其在计算机上运行时,使得计算机执行以上方法实施例提供的、由处理器执行的缓存方法。

[0204] 本申请实施例还提供了一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行以上方法实施例提供的、由处理器执行的缓存方法。

[0205] 本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”、“第三”、“第

四”等(如果存在)是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0206] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0207] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑模块划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0208] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要获取其中的部分或者全部单元来实现本实施例方案的目的。

[0209] 另外,在本申请各个实施例中的各模块单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件模块单元的形式实现。

[0210] 所述集成的单元如果以软件模块单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0211] 本领域技术人员应该可以意识到,在上述一个或多个示例中,本发明所描述的功能可以用硬件、软件、固件或它们的任意组合来实现。当使用软件实现时,可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质,其中通信介质包括便于从一个地方向另一个地方传送计算机程序的任何介质。存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0212] 以上所述的具体实施方式,对本发明的目的、技术方案和有益效果进行了进一步详细说明,所应理解的是,以上所述仅为本发明的具体实施方式而已。

[0213] 以上所述,以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前

述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

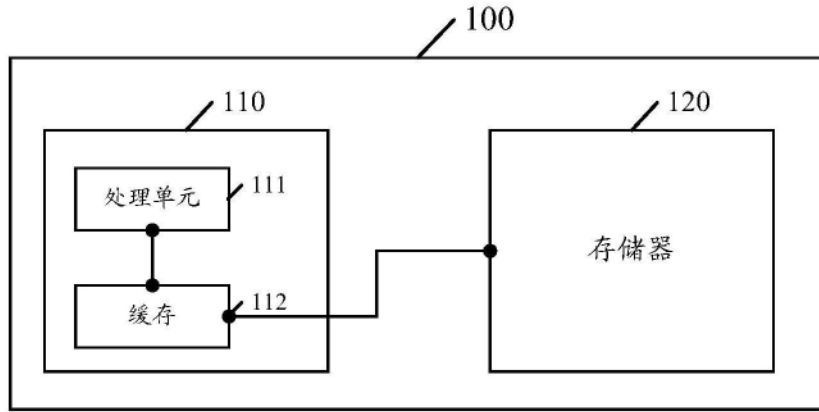


图1

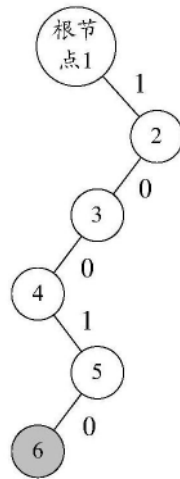


图2

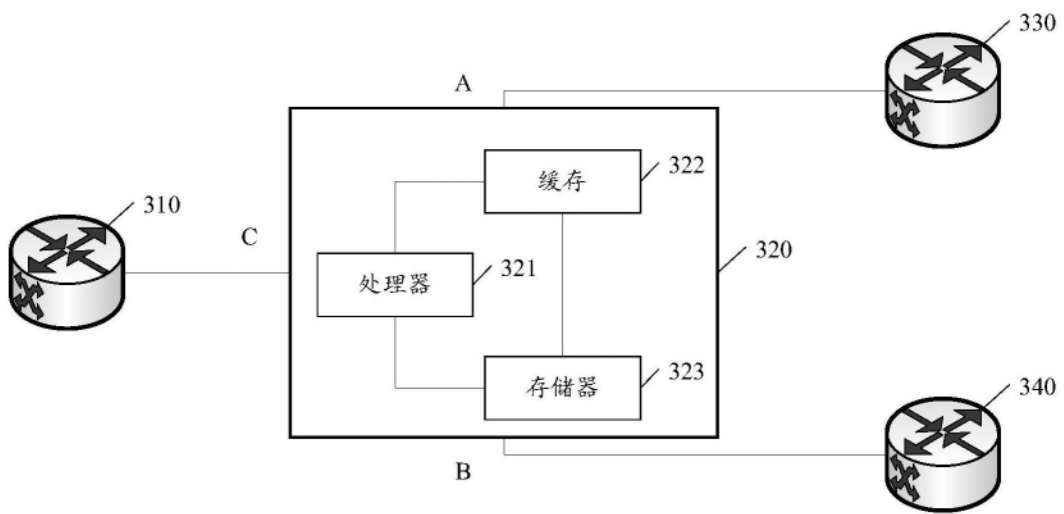


图3-a

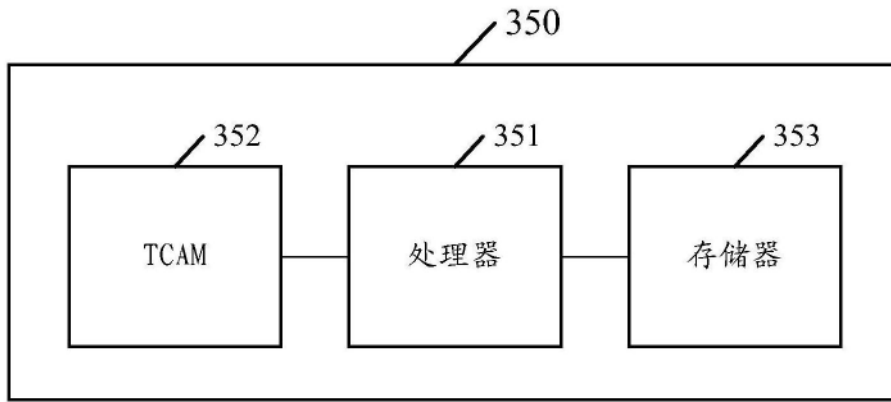


图3-b

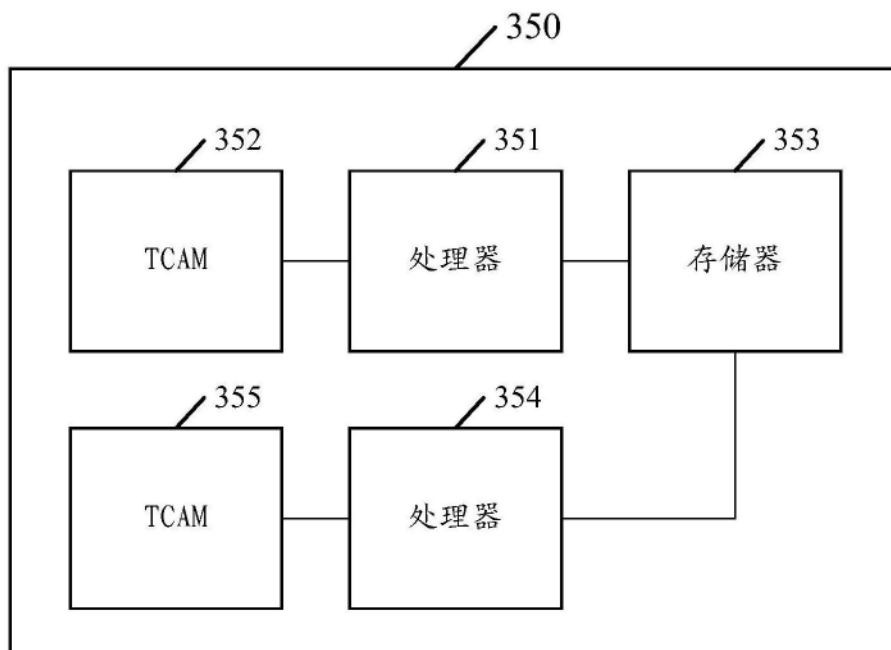


图3-c

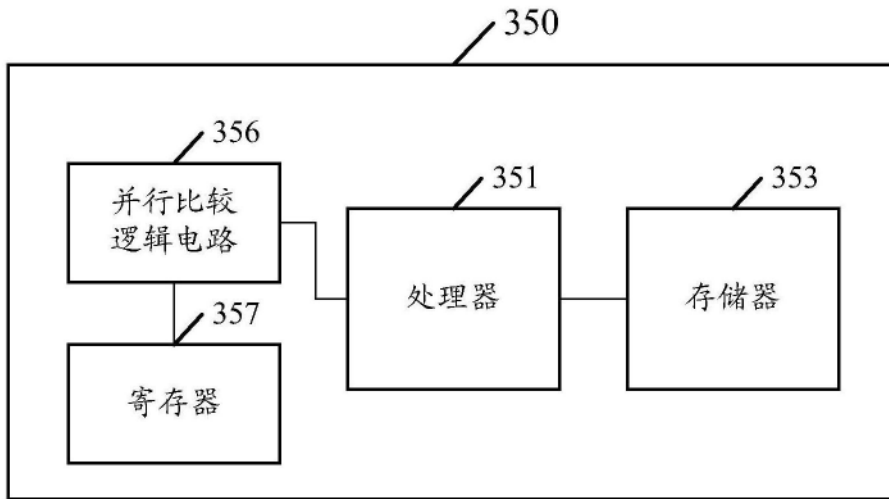


图3-d

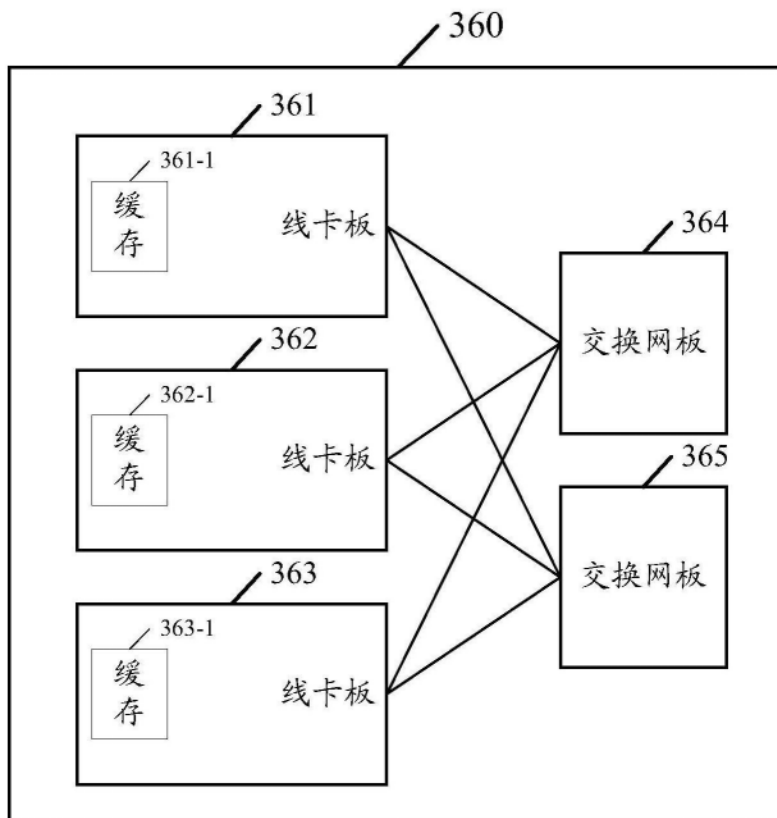


图3-e

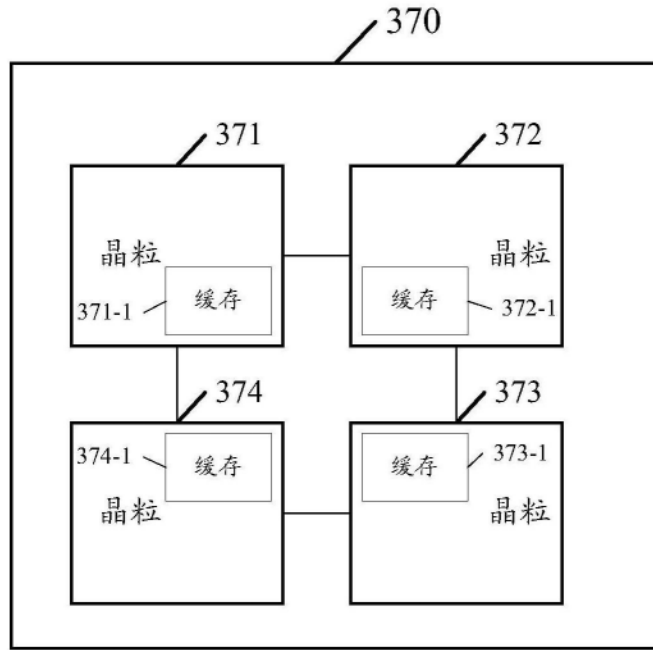


图3-f

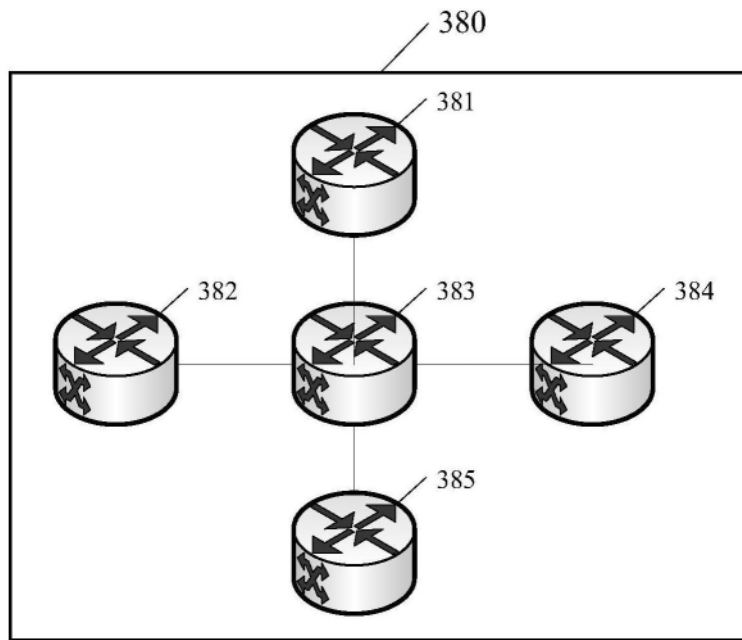


图3-g

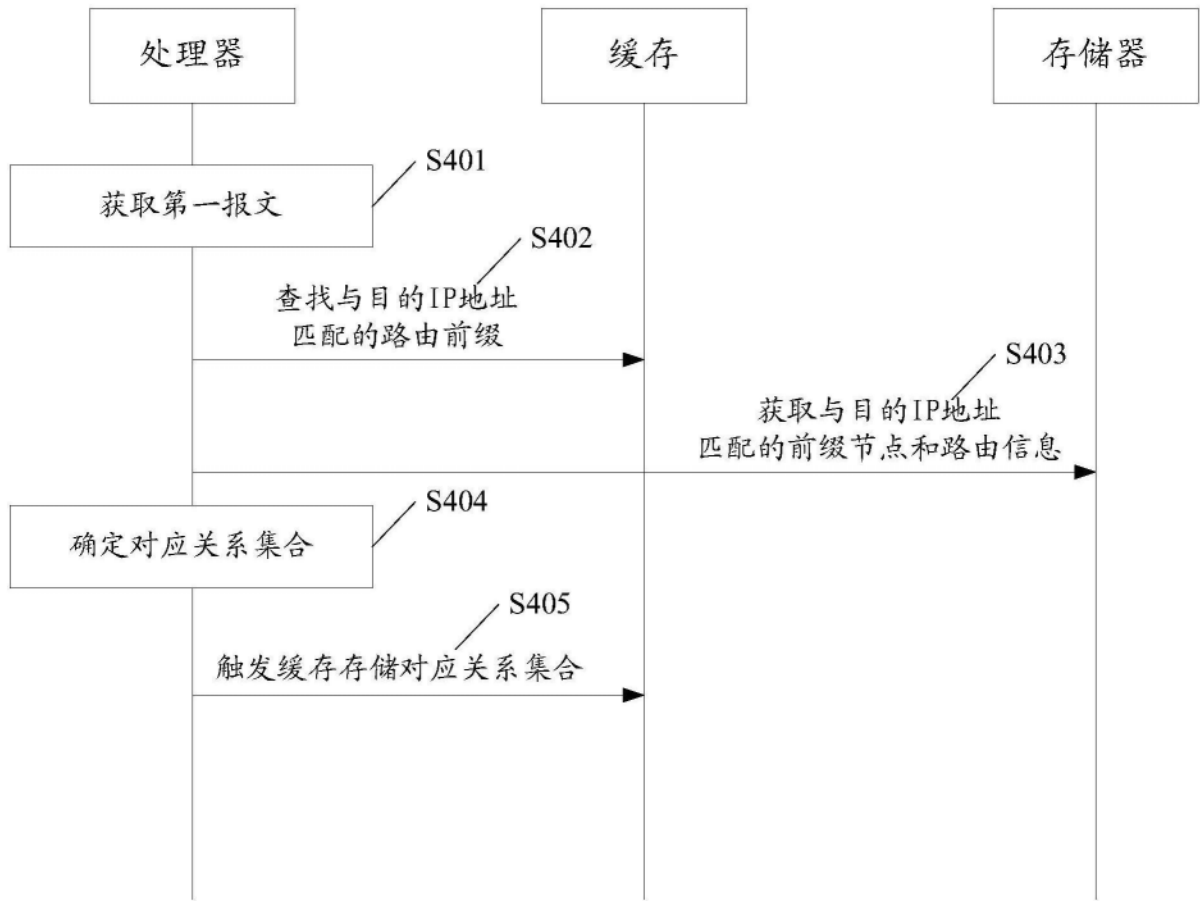


图4

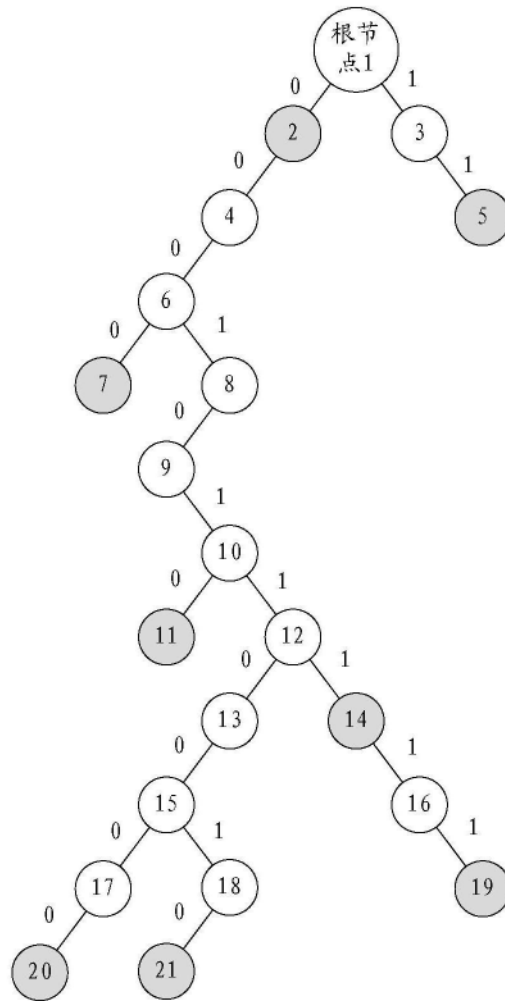


图5

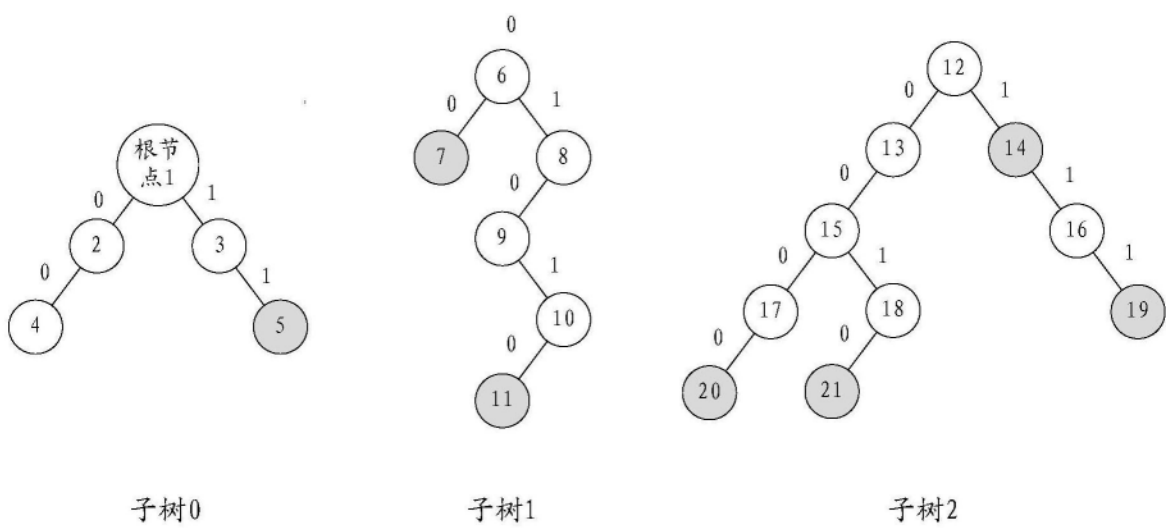


图6

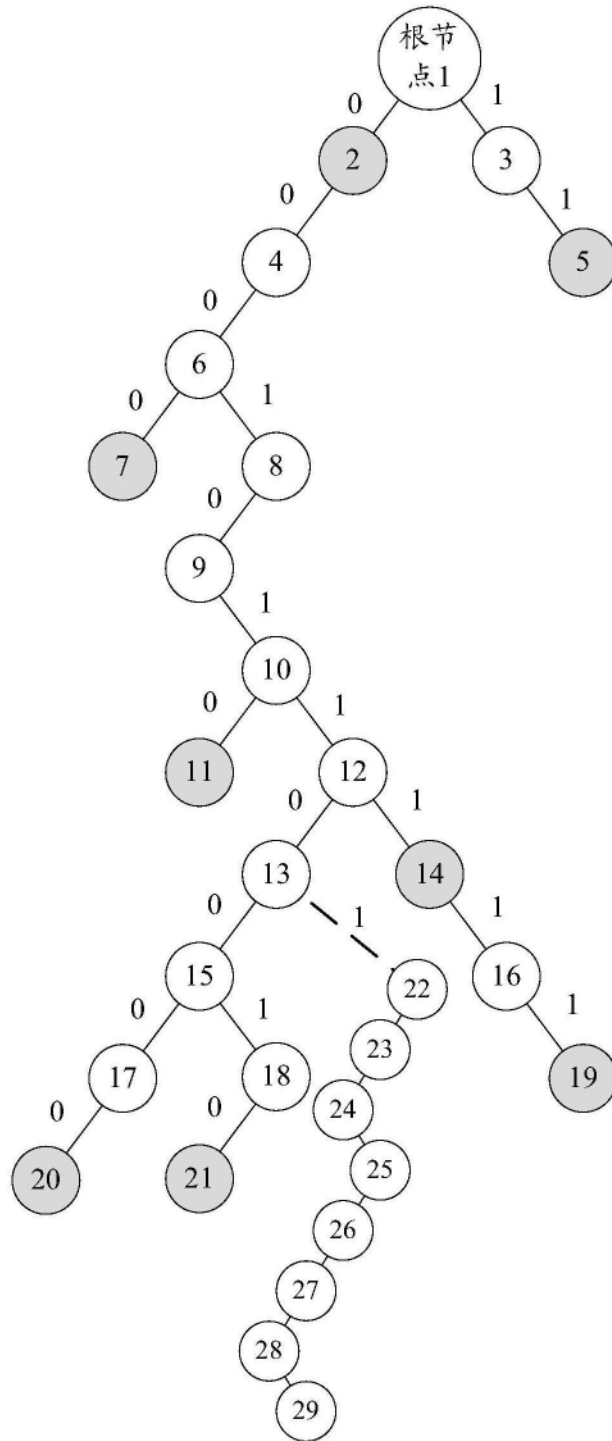


图7

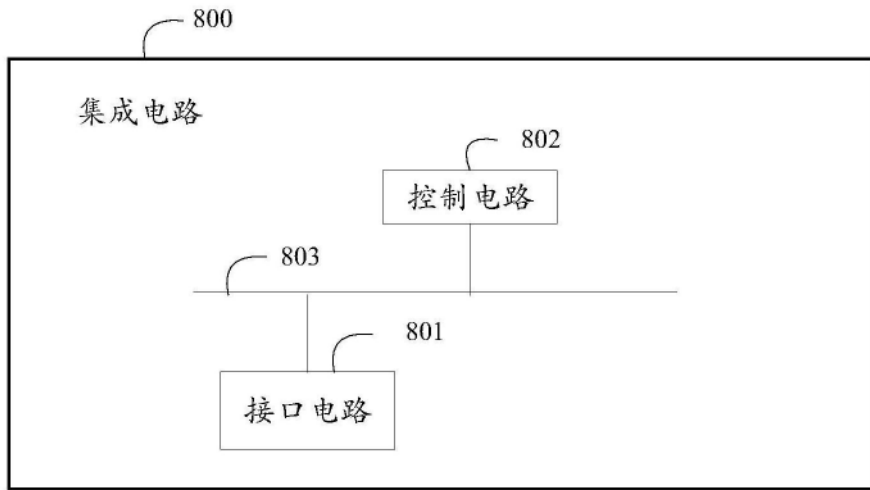


图8

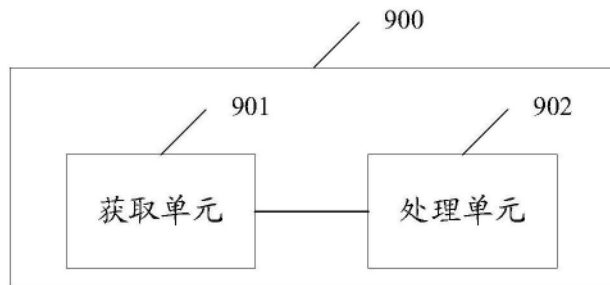


图9

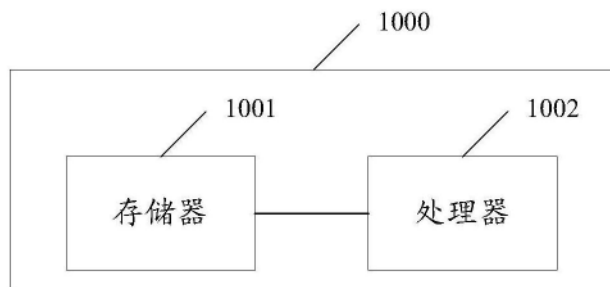


图10

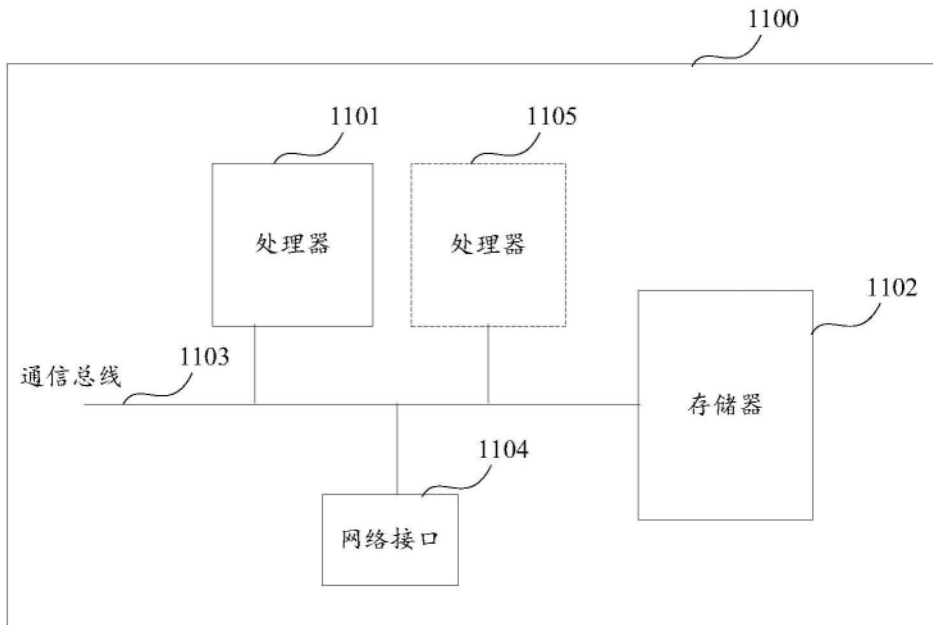


图11

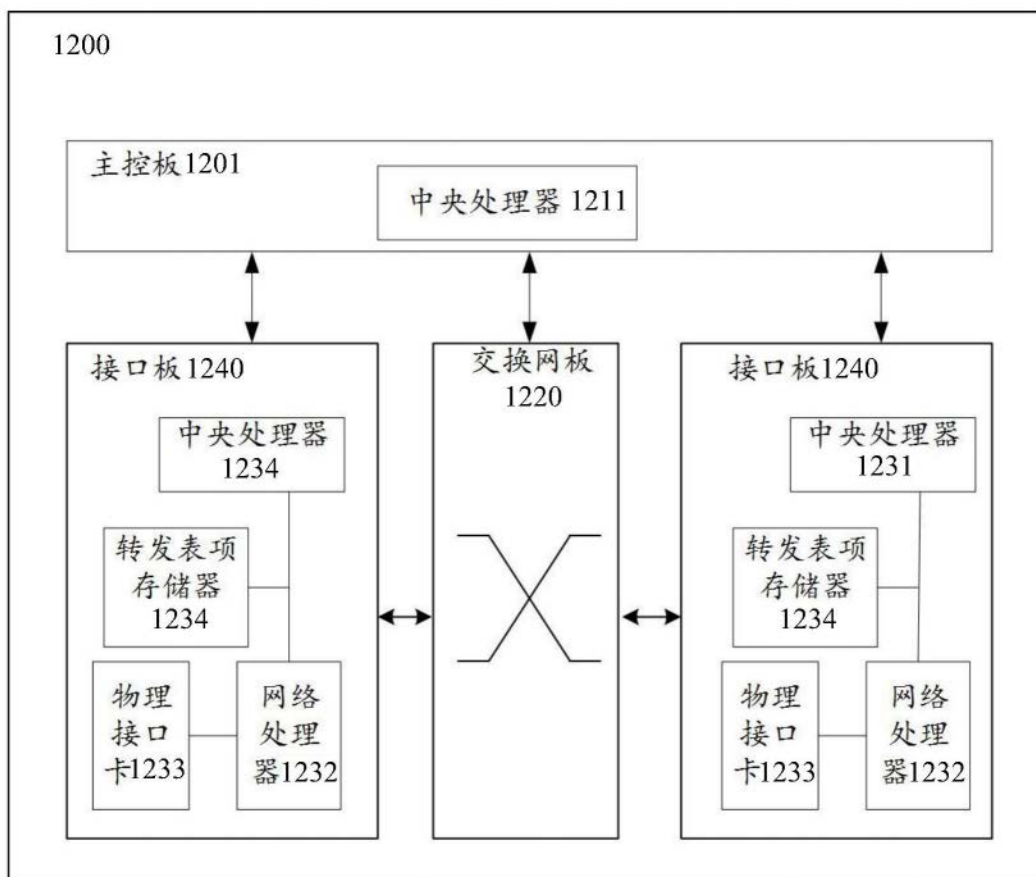


图12