[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 12/56 (2006.01)

[12] 发明专利说明书

专利号 ZL 03820350.2

[45] 授权公告日 2008 年 9 月 24 日

[11] 授权公告号 CN 100421417C

[22] 申请日 2003.8.29 [21] 申请号 03820350.2

[30] 优先权

[32] 2002. 8.30 [33] US [31] 60/407,165

[32] 2002. 9. 6 [33] US [31] 60/408,617

[32] 2003. 3.20 [33] US [31] 60/456,260

[32] 2003. 3.20 [33] US [31] 60/456,265

[86] 国际申请 PCT/US2003/027231 2003.8.29

[87] 国际公布 WO2004/021627 英 2004.3.11

[85] 进入国家阶段日期 2005.2.28

[73] 专利权人 美国博通公司

地址 美国加州尔湾市

[72] 发明人 尤里·埃尔朱 弗兰克·凡

史蒂夫·林赛

斯科特·S·麦克丹尼尔

[56] 参考文献

EP0905938A2 1999.3.31

US2002/0091844A1 2002.7.11

审查员 熊金安

[74] 专利代理机构 深圳市顺天达专利商标代理有限公司

代理人 蔡晓红 易 钊

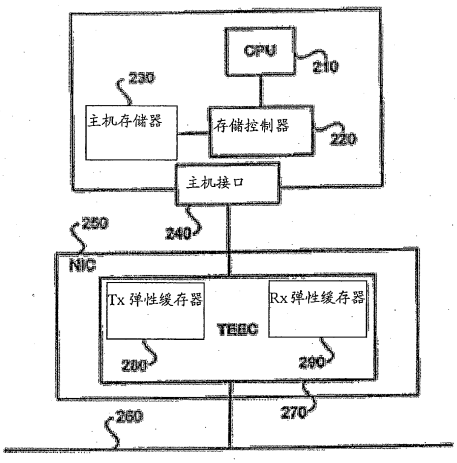
权利要求书 4 页 说明书 23 页 附图 15 页

[54] 发明名称

TCP 卸载的系统和方法

[57] 摘要

本发明中，在 TEEC 上接收输入 TCP 分组，并由所述 TEEC 处理所述输入分组的至少一部分一次，而不必由所述 TEEC 进行任何重组和/或重发。至少一部分所述输入 TCP 分组可缓存在所述 TEEC 的至少一个内部弹性缓存器中。 所述内部弹性缓存器可包括接收内部弹性缓存器和/或发送内部弹性缓存器。 因此，至少一部分所述输入 TCP 分组可缓存在所述接收内部弹性缓存器中。 至少一部分所述处理过的输入分组可放进一个主机存储器的一部分中，以便由主机处理器或 CPU 进行处理。 此外，至少一部分所述处理过的输入 TCP 分组可 DMA 传送到所述所述主机存储器的一部分中。



1.一种用于卸载 TCP 处理的系统，所述系统包括：

一个主机；

一个连接到所述主机的网络接口卡，所述网络接口卡包括，

一个 TCP 使能的以太网控制器，所述以太网控制器包括，

至少一个内部弹性缓存器，其中所述以太网控制器处理一个输入 TCP 分组一次，并将至少一部分所述输入 TCP 分组临时缓存到所述内部弹性缓存器中，所述处理在没有重组的情况下发生。

2.根据权利要求 1 所述的系统，其特征在于，所述至少一个内部弹性缓存器包括接收内部弹性缓存器和发送内部弹性缓存器中的至少一个。

3.根据权利要求 2 所述的系统，其特征在于，当所述至少一个内部弹性缓存器包括接收内部弹性缓存器时，所述输入 TCP 分组的所述至少一部分被临时缓存在所述接收内部弹性缓存器中。

4.根据权利要求 2 所述的系统，其特征在于，当所述至少一个内部弹性缓存器包括发送内部弹性缓存器时，要被传输的一个 TCP 分组的至少一部分被临时缓存在所述发送内部弹性缓存器中。

5.根据权利要求 1 所述的系统，其特征在于，所述以太网控制器将至少一部分所述输入 TCP 分组数据放进一个主机存储器的至少一部分中。

6.根据权利要求 1 所述的系统，其特征在于，所述网络接口卡只利用所述至少一个内部弹性缓存器来临时缓存所述输入 TCP 分组的所述至少一部分。

7.根据权利要求 1 所述的系统，其特征在于，无序的 TCP 分组不在以太

网控制器缓存器中进行存储、重新排序和重组中的至少一个操作。

8.根据权利要求 1 所述的系统，其特征在于，所述网络接口卡不需要专用的存储器来对失序的 TCP 分组重新排序。

9.根据权利要求 1 所述的系统，其特征在于，所述网络接口卡不需要专用的存储器来组合并重新排序在 IP 层分段上的 IP 分组。

10.根据权利要求 1 所述的系统，其特征在于，所述以太网控制器通过执行单一拷贝操作，至少将来自所述输入 TCP 分组的数据放进一个主机存储器中可得的缓存器的最高层中。

11.根据权利要求 1 所述的系统，其特征在于，所述以太网控制器将至少一部分处理过的输入 TCP 分组 DMA 传送到一个主机存储器的至少一部分中。

12.根据权利要求 1 所述的系统，其特征在于，所述网络接口卡不需要一个 TOE 专用存储器以用于分组重发和分组重组中的至少一个。

13.根据权利要求 1 所述的系统，其特征在于，所述以太网控制器将至少一部分处理过的输入 TCP 分组放进一个主机存储器的主机缓存器中，以便进行重组。

14.根据权利要求 1 所述的系统，其特征在于，所述以太网控制器包括一个单芯片，其中集成了所述至少一个内部弹性缓存器。

15.根据权利要求 1 所述的系统，其特征在于，所述以太网控制器包括一个单芯片，其中集成了所述至少一个内部弹性缓存器，并且没有用于进行分组

重发、分组重组和分组重新排序中的至少一种的内部缓存器和外部缓存器接口。

16. 一种用于卸载 TCP 处理的方法，所述方法包括：

在以太网控制器上接收输入 TCP 分组；

通过所述以太网控制器将至少一部分所述输入分组处理一次而不进行重组；以及

将所述输入 TCP 分组的所述至少一部分临时缓存于所述以太网控制器的至少一个内部弹性缓存器中。

17. 根据权利要求 16 所述的方法，其特征在于，所述至少一个内部弹性缓存器包括接收内部弹性缓存器和发送内部弹性缓存器中的至少一个。

18. 根据权利要求 17 所述的方法，其特征在于，当所述至少一个内部弹性缓存器包括接收内部弹性缓存器时，所述方法还包括在所述接收内部弹性缓存器中临时缓存所述输入 TCP 分组的所述至少一部分的步骤。

19. 根据权利要求 16 所述的方法，其特征在于，还包括将至少一部分处理过的输入分组放进一个主机存储器的至少一部分中。

20. 根据权利要求 16 所述的方法，其特征在于，所述放置步骤还包括通过执行单一拷贝操作，将至少一部分处理过的输入 TCP 分组放进一个主机存储器中可得的缓存器的最高层。

21. 根据权利要求 16 所述的方法，其特征在于，还包括将至少一部分处理过的输入 TCP 分组 DMA 传送到一个主机存储器的至少一部分中。

22. 根据权利要求 16 所述的方法，其特征在于，将多个分组临时缓存在

所述至少一个内部弹性缓存器中不是为了重组和重发中的至少一个。

23.根据权利要求 16 所述的方法，其特征在于，还包括将至少一部分处理过的输入 TCP 分组放进主机存储器的主机缓存器中，以便进行处理。

24.根据权利要求 16 所述的方法，其特征在于，所述以太网控制器包括一个单芯片，其中集成了所述至少一个内部弹性缓存器。

TCP 卸载的系统和方法

技术领域

本发明涉及 TCP 数据和相关的 TCP 信息的处理。更具体地说，涉及 TCP/IP 卸载的方法和系统。

背景技术

传输控制协议/因特网协议(TCP/IP)是一种已广泛用于通信的协议。传统的网络接口卡(NIC)一般包含适于对从传输介质接收到的分组信息进行处理的专用处理器或加速器。在示范性的网络接口卡中，数据的接收包括在数据被拷贝到终端目标站例如应用程序缓存器之前，在多数通信层中的分组数据处理。但是，在 TCP 段中通信的分组数据的接收、缓存、处理和存储会消耗接收器中大量的主机处理能力和存储器带宽。由于当今的高速通信系统已约有吉(G)比特的速度，因此这些传统的网络接口卡是低效的并且无法处理这种高速通信。

TCP 分段是一种可允许处理非常小的 TCP 部分以便将其卸载到网络接口卡(NIC)中的技术。因此，支持 TCP 分段的 NIC 并不真正包含整个传输控制处理卸载引擎。相反，支持 TCP 分段的 NIC 只具有将出站的 TCP 块分段成具有物理介质可支持的大小的分组的能力。每个出站 TCP 块小于允许的 TCP 窗体大小。例如，支持 TCP 分段的以太网接口卡可将 4KB 的 TCP 数据块分段成 3 个以太网分组。最大的以太网分组的大小为 1518 个字节，其中包括报头和报尾 CRC。

支持 TCP 分段的设备跟踪某个 TCP 状态信息，比如与卸载 NIC 正分段的数据相关的 TCP 序列号。但是，支持 TCP 分段的设备不会跟踪任何与入站业务量相关的状态信息、或者任何与支持 TCP 应答或流控制相关的状态信息。在已建立的状态中支持整个 TCP 卸载的 NIC 负责处理 TCP 流控制和负责处理输入 TCP 应答，以及产生输入数据的出站 TCP 应答。

TCP 分段可视为 TCP 卸载的一个子集。TCP 分段允许协议堆栈或操作系统

将还没有分段成单个 TCP 分组的 TCP 数据块形式的信息传输到设备驱动器。数据块可为 4K 字节或 16K 字节。与设备驱动器相关联的网络适配器可获得 TCP 数据块、将得到的 TCP 数据块分段成 1518 字节的以太网分组，以及更新每个按递增方式创建的分组中的某些字段。例如，网络适配器可通过使每个分组的 TCP 序列号递增，更新每个 TCP 分组相应的 TCP 序列号。在另一个示例中，每个分组的 IP 识别（IP ID）字段和标志字段也将得到更新。TCP 分段的一个局限性在于 TCP 分段仅可在小于 TCP 窗体大小的数据块上进行。这是由于实现 TCP 分段的设备不会影响到 TCP 流控制。因此，实现 TCP 流控制的设备只控制出站 TCP 分组的分段。

TCP 分段设备不检查输入的分组，同样地不会影响流控制。任何接收到的应答分组被传输到主机进行处理。因此，用于流控制的应答分组不由 TCP 分段设备处理。而且，TCP 分段设备不执行拥塞控制或流启动，并且不计算或修改任何传回到操作系统和/或主系统处理器的变量。

TCP 分段的另一局限性在于 TCP 分段跟踪的信息只是与 TCP 数据生命期相关的信息。例如，TCP 分段设备可跟踪 TCP 分段的数量，但不跟踪 TCP 应答(ACK)的数量。因此，TCP 分段设备只跟踪与相应 TCP 数据相关的信息的最小子集。这就限制了 TCP 分段设备的能力和/或功能。TCP 分段的还有一个局限性是 TCP 分段设备不会将 TCP 处理的信息传回到操作系统和/或主处理器。由于没有这种反馈，因此便限制了可由操作系统和/或主系统处理器实现的 TCP 处理。

与 TCP 分段相关联的其它局限性在 2003 年 8 月 29 日提交的美国专利申请号为_____（代理人档案号为 No. 13785US02）的专利申请中进行阐述，本专利中以之作为参考。

由于 TCP 段的处理会消耗大量的主机处理能力和存储器带宽，为减轻主机资源的消耗，可从主机卸载一些 TCP 处理，如图 1 所示。图 1 说明了传统的卸载系统。参考图 1，系统可包括 CPU 10、存储控制器 20、主机存储器 30、主机接口 40、网路接口卡(NIC)50 和以太网 60。NIC 50 包括 TCP 卸载引擎(TOE)70、发送帧缓存器 80 和接收帧缓存器 90。CPU 10 与存储控制器 20 相连。存储控制器 20 分别与主机存储器 30 和主机接口 40 相连。主机接口 40 通过

TOE 70 与 NIC 50 相连。TOE 70 分别与发送帧缓存器 80、接收帧缓存器 90 和以太网 60 相连。

工作时，由 NIC50 接收来自以太网 60 的输入帧。TOE70 在接收帧缓存器 90 中处理并存储帧。当主机存储器 30 中的缓存器可用并且已存储了足够多的帧时，TOE70 接收存储在接收缓存器 90 中的帧，并通过主机接口 40 和存储控制器 20 将帧传输到主机存储器 30。由主机输出的帧先传输到 TOE70，然后由 TOE70 将它们存储在发送帧缓存器 80 中。传输时，TOE70 取回存储在发送帧缓存器 80 中的帧，并通过以太网 60 传输这些帧。对于高速网络比如 10G 比特/秒的以太网(GbE)而言，附加的数据拷贝会增加计算机或主机的存储子系统上不必要的负担。大多数商业购得的服务器或主计算机的存储子系统成为了瓶颈，由此而妨碍了系统支持高数据速率比如 10G 比特的网络业务量。因为当今大多数应用软件中使用的 TCP/IP 协议在传输协议中占主导地位，因此它将有利地用于减轻处理过程的负担，以实现例如与同等机器通信时可升级的低级 CPU 的使用。

TCP/IP 在 IP 层采用报文服务。在路由器或开关拥塞的正常操作环境下，IP 报文可能会被丢弃，从而导致通向接收器时的报文流中存在一个“洞”。因此，接收器将以无序的方式接收报文。分组丢失还会导致，例如其它较少次数的传输误差。处理这一问题常用的方法是缓存成功接收的报文，同时等待以得到丢失的报文或从发送源重发的报文。重发可由发送器或接收器触发。在假设为高性能配置的情况下，TCP 协议允许每次连接的完整报文 TCP 窗口从发送器闪式传输到接收器。例如报文可包含 64K 字节数据。许多应用软件采用由大量接收器支持的 TCP 连接，例如 1000 到 100000 个 TCP 连接。在更高网络速度比如 1G 比特/秒或更高时，每当出现了报文丢失便将会为丢弃或耗尽输送管或一部分接收的数据流而使得网络变得低效。因为拥塞窗口大小减少，并且它必须要逐渐增加窗口直到其大小等于接收器的公开窗口大小为止，因此 TCP 带宽探测方法比如可能在连接启动时或检测到拥塞时触发的缓慢起动和/或避免拥塞会导致耗费宝贵的时间，故而它是低效的。因此，典型的 TCP 实现留出大缓存比如 64MB 到 6.4GB 的缓存来处理这些问题。使用大缓存来重组 TCP/IP 数据或

IP 片断。根据连接带宽和 TCP 连接上网络延时的乘积确定缓存器深度。因此这一结构易受 LAN 或 WAN 配置的影响，在这点上，用于高延时 WAN 配置的介质带宽的缓存器要比用于低延时高速 LAN 配置的介质带宽的多。

图 1 中说明的 TCP 卸载结构也称为存储和转发的方法。它增加用来存储 NIC50 的缓存器 80 和 90 中数据的等待时间，以管理缓存器 80 和 90、按顺序检索缓存器 80 和 90 外部的信息，并将它们传输到主存储器 30。在接收期间，接收的分组可存储在接收帧缓存器 90 中，并在其中进行处理。当分组以无序方式到达时，不是丢弃先前接收到的相关分组，而是缓存接收的分组，直到接收到丢失的分组为止。随后重组或重新排序接收到的丢失分组和无序分组。接着，处理组合或重新排序的分组，以确定这些分组应该放于主机系统上的什么位置中。一旦确定了组合分组的放置，便将该组合的分组传送到主机，分组在主机中存储以便用于处理。这一分别缓存、处理、重组或重新排序、处理和放置要求大量的存储器，并消耗大量的处理资源。

类似需要考虑的事项同样适用于发送侧。TCP 发送器维护拥有所有作为 TCP“窗口”部分传输的数据的发送帧缓存器 80。一旦远端应答接收到数据，发送器释放发送帧缓存器 80，TCP 窗口边缘移到右侧。发送帧缓存器 80 的大小与接收帧缓存器 90 的大小相似，因为没有被应答的未解决的数据在缓存器中，因此万一远端接收器没有接收到一个或多个报文，则允许发送器进行重新传输。与接收侧类似，它也是存储和转发结构。

此外，对于这一领域的普通技术人员来说，通过将这样的系统和如参考附图在本申请的其余部分中所阐述的本发明一些方面进行比较，常规和传统方法的局限性和缺点将是显而易见的。

发明内容

本发明包括 TCP 卸载的系统和方法上。所述系统可包括一个主机，所述主机包括主机存储器和与主机相连的网络接口卡(NIC)。所述 NIC 包括至少一个 TCP 使能的以太网控制器(TEEC)。所述 TEEC 包括至少一个内部弹性缓存器。因此，TEEC 包括接收内部弹性缓存器和/或发送内部弹性缓存器。TEEC 可配置

成处理输入 TCP 分组一次，而不用进行任何重组。因此，TEEC 可处理输入的 TCP 数据分组一次，并临时缓存内部弹性缓存器中至少一部分输入 TCP 分组，而不用将 TCP 分组数据与从同一流的相邻分组中的 TCP 数据进行组合。至少一部分输入 TCP 分组可暂时在接收内部弹性寄存器中缓存。在一些类似的方式中，至少一部分要传输的 TCP 分组可暂时缓存在发送内部弹性缓存器中。

TEEC适于将至少一部分输入TCP分组数据放到一个主机存储器的至少一部分中。通过执行单一拷贝操作，TEEC 便可至少将输入 TCP 分组的数据部分放到主机存储器中可用的缓存器的最高层。TEEC 可将至少一部分处理过的输入 TCP 分组 DMA 传送到一个主机存储器的至少一部分中。TEEC 还可将至少一部分处理过的输入 TCP 分组放到主机存储器中的主机缓存器中，以便进行重组。TEEC 可以是其中集成了至少一个内部弹性缓存器的单芯片。因此，接收内部弹性缓存器和发送内部弹性缓存器都集成在 TEEC 中。

卸载 TCP 处理的方法可包括在 TEEC 上接收输入 TCP 分组并由 TEEC 将至少一部分输入分组处理一次，而不需要通过 TEEC 进行任何重组或重发。可将至少一部分输入 TCP 分组缓存在 TEEC 的至少一个内部弹性缓存器中。内部弹性缓存器可包括一个接收内部弹性缓存器和/或一个发送内部弹性缓存器。可将至少一部分输入 TCP 分组缓存在接收内部弹性缓存器中。可将至少一部分处理过的输入 TCP 分组放在主机存储器的一部分中。因此，通过执行单一拷贝操作，可将至少一部分处理过的输入 TCP 分组放在主机存储器中可用的缓存器的最高层中。可将至少一部分处理过的输入 TCP 分组 DMA 传送到主机存储器的一部分中。

根据本发明的一个方面，在内部弹性缓存器中临时缓存的 TCP 分组不包括用于重组的分组和用于重发的分组。可由主机处理器或 CPU 将一部分处理过的输入 TCP 分组放在位于主机存储器中的主机缓存器中，以便进行处理。TEEC 可以是含有至少一个内部弹性缓存器的单芯片。而且，接收内部弹性缓存器和接收内部弹性缓存器均可集成在此芯片中。

本发明的另一实施例还提供了一种机器可读存储装置，其上存储了含有至少一个用于提供 TCP 卸载的代码部分的计算机程序。所述至少一个代码部分可

由机器执行，使得机器可执行以上所描述的 TCP 卸载的步骤。

根据以下描述和附图，将更清楚地明白本发明的这些和其它优点、方面和创新特征以及其中所示实施例的细节。

附图说明

图 1 示出一个了传统 TCP 卸载系统。

图 2 是根据本发明的实施例可用于以流过方式处理 TCP/IP 报文的示范性系统的框图。

图 3 示出了本发明一个实施例中 TCP 卸载系统的一个示范性接收系统。

图 4 示出了本发明一个实施例中 IPv4 的 IP 报文报头。

图 5 示出了本发明一个实施例中 IPv6 的 IP 报文报头。

图 6 是本发明中 TCP 报头格式的实施例。

图 7 示出了本发明一个实施例中示范性有效负荷选择。

图 8A 示出一个芯片组，其中 TEEC 是一个单芯片或单芯片的一部分。

图 8B 示出了本发明一个实施例中包括图 8A 中的 TEEC 和专用元组和/或内容存储器的 NIC。

图 9 示出了本发明一个实施例中可将输入分组数据绘制和拷贝到主机常驻缓存器或缓存器组中的一个系统。

图 10 示出了本发明中的一个示范性的发送路径。

图 11 示出了本发明中的一个示范性的帧接收步骤的流程图。

图 12 示出了本发明中的一个示范性接收系统实施例的框图。

图 13 是本发明一个实施例中接收系统的示范性实施例的框图。

图 14 是本发明一个实施例中接收系统的示范性实施例的框图。

具体实施方式

本发明包括 TCP 卸载的系统和方法。该方法中可包括在 TEEC 上接收输入 TCP 分组，并由 TEEC 将至少一部分输入 TCP 分组处理一次，而不需要通过 TEEC 进行任何重组或重发。可将至少一部分输入 TCP 分组缓存在 TEEC 的至少一个

内部弹性缓存器中。内部弹性缓存器包括一个接收内部弹性缓存器和/或一个发送内部弹性缓存器。因此，可将至少一部分输入 TCP 分组缓存在接收内部弹性缓存器中。可将至少一部分处理过的输入分组放置到主机存储器的一部分中。因此，通过执行单一拷贝操作，便可将至少一部分处理过的输入 TCP 分组放置在主机存储器中可用的缓存器的最高层中。而且，可将至少一部分处理过的输入 TCP 分组 DMA 传递到主机存储器的一部分中。

根据本发明的一个实施例，可将无序的 TCP 分组存储在小的内部弹性缓存器中。弹性缓存器可以是例如 64 KB 的片内分组缓存器，该缓存器用于提供与例如用于分组重排序、重组和/或重发的多兆字节存储器相反的弹性。根据本发明的各种实施例的弹性缓存器一般可通过 NIC 临时缓存至少一部分输入 TCP 分组至其中。此外，根据本发明的一个实施例的 NIC 将不包括用来重排序或重组无序 TCP 分组或 IP 段的专用存储器。而且，根据本发明的一个实施例，NIC 将不包括用来分组重发和/或分组重组的大 TOE 专用存储器。因此，不需要由 TCP 使能的以太网控制器 (TEEC)进行分组重组和/或分组重发缓存。

传输控制协议/因特网协议(TCP/IP)是网络和基于因特网数据传输的主要协议。TCP/IP 的使用已经超出了应用到应用通信和基于文件存储比如网络文件系统(NFS)和公共网络文件系统(CIFS)的范围，而扩展到基于块的网络存储比如因特网小型计算机系统接口(iSCSI)。TCP/IP 还可用来在传输层采用远程 DMA(RDMA)协议进行聚类/进程间通信(IPC)。

以有线速度的处理 TCP/IP 会完全耗尽例如 1GHz 处理器。使用 TCP 使能以太网控制器(TEEC)可提供例如以下优点中的一个或多个：降低的主机 CPU 使用，例如，从当运行 TCP/IP 应用软件的大约 100%减少到少于大约 10%；更少的数据拷贝；以及更少的中断和内容转换，这些都将为应用软件的处理释放主机 CPU 和系统。这些优点在更高速时变得更加明显。从系统的角度来看，使用 TEEC NIC 可提供比使用专用处理器或其基本部件和其相关的 TCP 处理系统进行处理更好的投资回报(ROI)，即使有些的平均卖价(ASP)费用高于通常的 GbE NIC 也如此。也

本发明的某些方面可提供通过为 TEEC 的发送路径和接收路径提供最小量

存储器，以流过方式来处理 TCP/IP 报文。现有卸载系统采用的各种缓存、处理、重组或重新排序、处理和放置的方法，例如图 1，在接收和发送侧需要大量的存储器和消耗大量的处理资源。但是，根据本发明的一个实施例和参考图 2，发送弹性缓存器 280 和接收弹性缓存器 290 按提供的流过设计方式操作。因此，TEEC270 的发送弹性缓存器 280 和接收弹性缓存器 290 适于临时缓存接收分组，并用于提供弹性以适应例如以太网接口与计算机的主机接口如 PCI 接口之间的可变数据速率。因此，TEEC 270 和其相关的接收弹性缓存器 290 可以这样操作，使得接收到的分组在接收弹性缓存器 290 中临时缓存，在主机存储器 30 中处理和放置。这一流过处理不需要接收弹性缓存 290 中无序分组的重组或重新排序。因此，使由现有的卸载系统所采用的各种缓存、处理、重组或重新排序、处理和放置的方法对处理和放置而言减少到最小。

根据本发明的一个方面，TEEC 270 不需要用来组合和/或重新排序在 IP 层处分段的 IP 分组的专用存储器。因此，在 TEEC 缓存中不存储、重新排序和/或组合无序的 TCP 分组。因此，NIC 可包括其中集成了至少一个内部弹性缓存器的单芯片，它没有用于分组重发、分组重组和分组重新排序的内部缓存器或与外部缓存器的接口。

图 2 是根据本发明的一个实施例，以流过方式处理 TCP/IP 报文的示范性系统的框图。参照图 2，本系统可包括例如 CPU210、存储控制器 220、主机存储器 230、主机接口 240、网络接口卡(NIC) 250 和以太网 260。虽然举例说明了例如 CPU 210 和以太网 260，但是本发明并不局限于此，并且可以使用例如任何类型的处理器和任何类型数据链层或物理介质。NIC250 可包括例如 TEEC 270、发送弹性缓存器 280 和接收弹性缓存器 290。发送弹性缓存 280 和接收弹性缓存器 290 可以是内部弹性缓存器。

虽然举例说明的是以太网 260 的控制器，但是 TEEC270 可以是任何类型数据链路层或任何类型的物理介质的控制器。在本发明的一个实施例中， TEEC 270 可实现 TOE 的至少一些功能。主机接口 240 可以是例如周边元件扩展接口(PCI)、PCI-X、ISA、SCSI 或另一类型总线。存储控制器 230 可分别与 CPU 220、存储器 230 和主机接口 240 相连。主机接口 240 可通过 TEEC270 与 NIC250 相

连。最后，TEEC270 可与以太网 260 相连。

运行中，在接收侧，NIC250 从以太网 260 中接收分组或帧。通常， TEEC270 例如会解析和处理报头，并且将接收到的分组临时缓存到接收弹性缓存器 290 的特定位置中。因此，TEEC 可“闪式”处理每个输入分组。例如基于控制信息、报头信息和/或有效负载信息 以及接收到的分组可确定和/或断定放置信息。一旦确定了接收到的分组的放置信息，TEEC 270 便可将接收到的分组传输到主机，在主机中将接收到的分组存储在主机存储器 230 中以便进行处理。

在本发明的一个方面中，至少一部分接收到的分组通过 TEEC270 进行处理并在接收弹性缓存器 290 中进行排列。将接收到分组的排列部分从接收弹性缓存器 290 DMA 传输到主机存储器 230。因此，TEEC270 可包括适当的 DMA 硬件和/或代码，它适于将接收到的分组通过主机接口 240 从接收弹性缓存器 290 直接传输到主机存储器 230。因此，可将分组在以太网 260 电缆上传输、“闪式”处理和在接收弹性缓存器 290 中临时缓存。由于在 NIC 250 中进行“闪式”处理和临时缓存，因此在 NIC 250 上不进行分组的重组或重新排序。

在本发明的另一方面中，本系统还可处理如可在例如帧延时或帧丢失期间出现的无序帧。例如，TEEC270 可管理漏洞或漏洞组，直到接收到正确的数据为止。在发送路径上，可在 NIC250 的发送弹性缓存器 280 中进行传输。因此，传输到以太网接口的 TCP 数据可从主机存储器 230 传输，并在发送弹性缓存器 280 中临时缓存。TEEC270 可完成“闪式”传输。TEEC270 可从主机中取到传输 TCP 数据，在发送弹性缓存器中临时缓存取到的数据然后为传输处理数据。通过格式化以及附加更高级协议报头和错误校正代码，将数据构建到一个或多个以太网分组中。传输之后，数据可在例如 TEEC 270 所拥有的主机上保留。在本发明的一个方面中，没有拷贝的分组或突出的 TCP 传输数据本地存储在 TEEC 270 上，以助于重发。因此，TEEC 270 可适于通过从主机存储器 230 中再次取得数据以及在发送弹性缓存器 280 中临时缓存取得的数据，以助于重发，。一旦数据得到远端对等体的应答，发送器的主机缓存器便可对于它们的原所有者比如应用软件或 ULP 所利用。

根据本发明，在接收侧，不像图 1 中的 TOE 70，TEEC 270 没有用来重新

排序 TCP 业务量以处理例如无序接收的 TCP 段的专用外部存储器。而且，TEEC 270 不适于重组或重新排序无序接收的 TCP 段。然而，TEEC 270 可适于处理无序段，而不需利用外部专用存储器，也不需在将 TCP 段 DMA 传送到主机存储器 230 中之前重新排序帧。在本发明的另一方面中，TEEC 270 使用的内部存储器比传统的 TOE 70 使用的存储器小。这中由 TEEC 270 使用的更小的内部存储器提供了弹性，并可例如用来计算“闪式”处理时的内部延时。TEEC 270 的更小的内部存储器也可在缓存接收到的含有不足的放置信息的帧时提供弹性。考虑到 TEEC 270 的更小的存储器的大小，因此它不会被用来缓存分组，在其它情况下，该分组可在没有得到应答 (ACK)时便被重发。

在本发明的某些方面中，主机存储器 230 可用来例如重组接收业务量或用于传输行为和转发行为。这样对于 TEEC 270 而言便消除了拥专用外部存储器的必要。例如，这样可减少以下部分中的一个或多个：成本、复杂性、覆盖域和能量消耗。另外，这样可消除或降低带宽延时乘积相关性。主机存储器 30 一般比任何经济可行地附加于 TEEC 的存储器更大并且更容易扩展。当启动应用软件和 TCP 时，它也可因为是数据源和目的地而体现了机器缓存数据能力的限制。同时它也可体现了单块软件堆栈的限制。

本发明的某一实施例与传统的卸载引擎相比还提供了虚拟的无缓存和简化的缓存结构。在此方面，虽然不是真正的无缓存设计，但是在与图 1 中的传统 TOE 70 相比时， TEEC 270 内部存储器的大小是相当小的。这些结构设计用于维护小量的存储器比如 TEEC 270 上的 FIFO。对于 FIFO 来说，FIFO 将提供弹性并代替了 TEEC 或 NIC 上专用外部存储器的需要。因此，这可有助于 TCP 段的“闪式”处理。段的“闪式”处理可视为是可移动 TCP 段的“单触式方法”，例如在正在处理协议层时将 TCP 段移入可用缓存的最高层。例如，因为 L5 或更高的应用缓存器的使用可节省额外的拷贝处理步骤，所以层 5 (L5) 或更高的应用缓存器可提供比专用 L4 TCP 缓存器或普通的 L4 TCP 缓冲器更好的性能。它还可消除或减少例如在发送路径上使用任何中间缓存器的需要。

在 TEEC 的“单触式”处理期间，TEEC 的虚拟无缓存器结构或简化的缓存器结构可扩展到特定 TEEC 所支持的相同高度的处理水平。如果可向 TEEC 系统

提供缓存信息和协议解析信息，则数据可直接放置在 L5 或更高的缓存器中。虚拟无缓存器或简化的缓存器结构可支持并行操作，如 TCP 层 2 (L2)、层 4 (L4) 和层 5 (L5) 之间的灵活转换。

在本发明的一方面中，不像许多传统的 TOE 设备，TEEC 75 可作为纯以太网控制器并提供整套 L2 服务。TEEC 75 还可作为纯 TEEC，或者可例如有在 L2 上的一些业务量，如非 TCP 以太网业务量，以及在 L4 上的一些业务量。对于 L2 业务量而言，可提供 L2 服务，比如以太网地址对照和 CRC 计算。对于 L4 业务量，可提供另外的服务包括例如设备上的 TCP/IP 处理。TEEC 75 还可作为纯 L5 或更高的使能控制器。任一处理级还可用于经过 TEEC 75 的不同连接的任一组合中。为了有助于硬件对它的管理，业务量混合上没有限制，也不需要外部软件干涉。根据本发明的一些实施例只针对 L4 服务。这种 TCP/IP 处理的创新方法不需要依赖以太网，并可应用于任何其它的 L1/L2 接口。

图 12、图 13 和图 14 均为说明根据本发明的一个实施例的接收系统的实施例的框图。参照图 12-14，在每个所示的实施例中，可采用管道处理，并且可将信息分到两条通路中：控制处理通路和数据移动通路。参照图 12，以太网与第一处理元件 300 相连。第一处理元件 300 可提供例如 L1/L2 处理。可解析输入信息，而且至少一部分输入信息可通过处理元件 310 和 DMA 引擎 320 引入到控制处理通路。输入信息的另一部分例如有效负荷数据，可通过存储元件 330 和 DMA 引擎 320 引入到数据移动通路中。

处理元件 310 还可处理来自处理元件 300 的所接收的控制信息。在一个实施例中，处理元件 310 可适于执行 L4/L5 或更高级的处理。例如记录在本地存储器 340 中的内容信息可由处理元件 310 进行访问。处理元件 310 可得到记录在本地存储器 340 中的内容信息， 以及从先前的处理元件 300 中接收的控制信息，并在将合并的信息传输到 DMA 引擎 320 之前处理并合并信息。在将数据或合并的信息直接存储到一个或多个主机缓存器中之前，DMA 引擎 320 可将控制通路中的控制信息与存储在存储元件 330 中的数据合并。

图 13 和图 14 还显示了根据本发明其它多级配置。具体而言，图 13 显示了多个控制处理级，每个控制级都能访问内容信息。内容信息可包括与 TCP

连接状态有关的 TCP 连接的信息，该信息通常包括用来表现 TCP 连接特征的连接状态信息。图 14 显示了多个控制处理级和存储级。虽然图 14 中只显示了单个内容部分，但是可将多个处理元件与各个内容部分或共同的内容部分相连。在其它结构中，级间处理元件可与存储级相连，存储级中的数据是处理数据和/或对应于数据控制信息中的因子。

图 3 说明了根据本发明的实施例的 TCP 卸载系统的示范性接收系统。输入帧可经受 L2 比如包括如地址过滤、帧有效性校验和错误监测的以太网处理。不像普通的以太网控制器，处理的下一级可包括例如 L3 比如 IP 处理，以及 L4 比如 TCP 处理。TEEC 可减少主机 CPU 的利用率和存储器带宽，例如通过处理硬件上的业务量卸载 TCP/IP 连接。TEEC 可监测例如输入分组属于的协议。如果协议是 TCP，则 TEEC 可监测分组是否与卸载的 TCP 连接对应，例如，可由 TEEC 保存的至少一些 TCP 状态信息的连接。一旦连接与分组或帧关联，则可达到任何处理的更高级比如 L5 或更高。如果分组与一卸载的连接对应，则 TEEC 可引导帧的数据有效负荷部分的数据移动。可根据结合帧内的方向信息的连接状态信息来确定有效负荷数据的目标位置。例如目标位置可以是主机存储器。最后，TEEC 可更新它的内部 TCP 和连接状态的更高级，并可根据它的内部连接状态得到主机缓存器地址和长度。

接收系统结构可包括例如控制通路处理和数据移动引擎。如图 3 上面部分所示的控制通路上的系统组件可用以处理不同的处理级，这些处理级用于完成例如具有最大的灵活性和最高的效率以及最大的目标有线速度的 L3/L4 或更高级的处理。级处理的结果包括例如一个或个多分组识别卡(PID_C)，它可提供可承载与帧有效负荷数据相关的信息的控制结构。当在不同的块中处理分组时，这可在 TEEC 中产生。图 3 下部所示的数据移动系统可将帧的有效负荷数据部分从例如片内分组缓存器向前移动并在完成了控制处理时，将其移动到直接存储器存取(DMA)引擎，并随后移动到经由处理而选择的主机缓存器中。

接收系统可完成例如以下操作中的一个或多个：解析 TCP/IP 报头；将帧与端到端 TCP/IP 连接相关联； 取得 TCP 连接内容；处理 TCP/IP 报头；确定报头/数据边界；将数据映射到主机缓存器；以及将数据通过 DMA 引擎传送到

这些缓存器中。报头可在片内被处理或经由 DMA 引擎被传送到主机中。

分组缓存器是接收系统结构中的一块。它用于与例如在传统的 L2 NIC 中所采用的先入先出(FIFO) 数据结构相同的用途，或用来存储更高级的业务量以便进行其它处理。

接收系统中的分组缓存器并不局限于单一的实例。当执行完控制通路处理时，数据通路可根据例如协议要求，一次或多次存储数据处理级之间的数据。

图 11 是本发明一个实施例中帧接收的示范性步骤的流程图。参照图 3 和图 11，在步骤 100 中，NIC50 可从例如以太网 60 中接收帧。在步骤 110 中，帧解析器可解析帧，例如，找出 L3 和 L4 报头。帧解析器可处理 L2 的报头，将其带到 L3 的报头上，例如 IP 版本 4(IPv4)的报头或 IP 版本 6(IPv6)的报头。IP 报头版本字段可确定帧是否承载了 IPv4 报文或 IPv6 报文。图 4 说明了根据本发明的一个实施例的 IPv4 的 IP 报文。图 5 说明根据本发明的一个实施例的 IPv6 的 IP 报文。例如，如果 IP 报头版本字段承载的值为 4，则该帧可承载 IPv4 报文。如果，例如，IP 报头版本字段承载的值为 6，则该帧可承载 IPv6 报文。可提取 IP 报头字段，因此得到例如 IP 源(IP SRC)地、IP 目标(IP DST) 地址和 IPv4 报头“协议”字段或 IPv6 的“下一报头”。如果 IPv4 的“协议”报头字段或 IPv6 的“下一报头”报头字段承载的值为 6，则随后的报头可以是 TCP 报头。将解析的结果添加到 PID_C 中， PID_C 与 TEEC 内部的分组一起传播。

随后，以与传统现成的软件堆栈中的处理相似的方式处理余下的 IP。实现可随嵌入式处理器固件或可能速度快点的专用的有限状态机或处理器和状态机的混合的使用而而变化。实现可随，例如通过一个或多个处理器、状态机或混合的多个处理级而改变。IP 处理可包括例如提取与如长度、有效性、分段等相关的信息。还可解析和处理所定位的 TCP 的报头。图 6 是说明根据本发明的一个实施例的 TCP 报头格式的图。TCP 报头的解析可提取与例如源端口和目的端口相关的信息。

TCP 处理可分为多个附加处理级。在步骤 120 中，帧与端到端的 TCP/IP 连接相关联。在一个实施例中，在 L2 处理后，本发明可提供验证 TCP 校验和。

例如通过以下 5 元组来唯一地确定端到端连接：IP 源地址(IP SRC addr)、IP 目的地址(IP DST addr)、IP 协议之上的 L4 协议(例如，TCP、UDP 或其它高层协议)、TCP 源端口数量(TCP SRC)和 TCP 目的端口数量(TCP DST)。在选择相关的 IP 地址的情况下，该处理可适用于 IPv4 或 IPv6。

作为步骤 110 中帧解析的结果，可完全提取出 5 元组，并可在 PIC 中得到。关联硬件将接收 5 元组与存储 TEEC 中的 5 元组列表进行比较。TEEC 保留元组表示列表，例如，先前操作的卸载连接或 TEEC 管理的卸载连接。存储联合信息使用的存储器资源对于片内和片外可选是高成本的。因此，可能不是所有的联合信息都驻留在芯片上。使用高速缓冲存储器存储芯片上最活跃的连接。如果发现相匹配，则 TEEC 管理有匹配 5 元组的特定 TCP/IP 连接。

图 7 根据本发明的一个实施例，说明了示范性的有效负荷选择。当不匹配时，则根据例如一个或更多卸载方案选择、依据本发明的图 7 中说明的实施例，管理 TCP 连接。

使用嵌入式处理器固件，或可能速度快点的专用、有限状态机，或处理器和状态机的混合，TCP 处理执行大不相同。执行会随着通过一个或多个处理器、状态机或混合的多个处理级而不同。TCP 处理可包括，例如，提取与如长度、有效性、分段等相关的信息。还可解析和处理所定位的 TCP 的报头。图 6 是根据本发明的一个 TCP 报头的实施例。

随后，以与传统现成的软件堆栈中的处理相似的方式进行任何更高级的处理，比如 L5 及更高级处理。实现可随嵌入式处理器固件或可能速度快点的专用的有限状态机或处理器和状态机的混合的使用而有所变化。实现会随通过一个或多个处理器、状态机或混合的多个处理级而变。更高级的处理可包括例如提取例如与帧有关的安全信息、放置信息和缓存管理信息。更高级的处理并不仅限于这些操作。

参照图 7，选项 A 包括单块软件堆栈和硬件堆栈。硬件堆栈提供了例如对于所有通过硬件堆栈处理的帧和那些通过单块集成电路软件堆栈管理的帧的标准 L2 帧处理。硬件堆栈向某些连接提供了例如更高级的卸载服务，而单块软件堆栈向其它连接提供了例如更高级的卸载服务器。当不匹配时，硬件可假

设单块软件堆栈管理连接。但是，这不排除保留与例如 TEEC 内或利用适合连接的软件驱动的特定 TCP 连接相关的统计数字，以便将来从单块软件堆栈卸载到硬件堆栈中。后台任务会占用大多数使用过的连接，并在硬件堆栈上将连接推入卸载状态。

同样参照图 7，选项 B 包括单块软件堆栈、软件卸载堆栈和硬件堆栈。硬件堆栈提供了，例如，对所有通过硬件堆栈处理的帧和那些通过软件堆栈管理的帧的标准 L2 帧处理。硬件堆栈例如向某些连接提供更高级的卸载服务。软件卸载堆栈例如向另一组连接提供更高级的卸载服务，而单块软件堆栈例如向另一组连接提供了更高级的卸载服务。当在硬件堆栈中不匹配时，硬件还检查看软件卸载堆栈是否管理连接。当软件卸载堆栈中管理连接时，可将帧转发到软件卸载堆栈，该堆栈可在处理连接的同时维护准备用于硬件卸载的数据结构。统计数字可一直保留在这些连接中。如果连接被确定是高使用率的，则它可被直接卸载到硬件。如果软件卸载堆栈不能处理该连接，则可将该连接传递到单块软件堆栈。软件卸载堆栈如硬件堆栈一样可处理帧。因此，从单块软件堆栈的角度，硬件堆栈和软件卸载堆栈的结合可处理所有已经卸载的连接。

在步骤 130 中，TCP 连接内容可从如内容存储器中取得。内容信息包括如在存储数据的主机中用来处理帧和缓存信息的 TCP 变量。图 8A-B 根据本发明，说明了一些元组存储位置和/或内容信息示范性实施例。除了片内存储器，可使用外部存储器资源扩充容量。

图 8A 说明了示范性芯片组，其中 TEEC 是一单芯片或单芯片的一部分。TEEC 75 可从位于主机存储器 30 中的元组和/或内容缓存器中取得元组和/或内容信息。TEEC75 还可从与芯片集 55 相连的专用元组和/或内容存储器 35 中取得元组和/或内容信息。

图 8B 根据本发明的一个实施例，说明了 NIC，它包括例如图 8A 的 TEEC 和专用元组和/或内容存储器。TEEC 75 可从位于主机存储器 30 中的元组和/或内容缓存器中取得元组和/或内容信息。TEEC 75 还可专用元组和/或内容存储器 35 中取得元组和/或内容信息，存储器 35 同样在 NIC 50 上并与 TEEC 75 相连。

在步骤 140 中，处理 TCP/IP 报头。将数据从帧解析器移动到帧缓存器，实现一些 IP 和 TCP 帧的有效检查例如 IPv4 报头校验和与 TCP 校验和。在 PID_C 中登记结果。TCP/IP 报头，将从内容存储器中取得的内容和至今在 PID_C 中产生的信息提供给接收器处理块，此处理块包括一个或多个处理器和/或有限状态机。接收器处理块可利用例如内容信息来完善对于帧的附加 TCP/IP 处理，这种处理包括例如更新 TCP 状态变量或重新设置 RFC 79 中所述的定时器。接收器处理块还可使用部分由先前存储在 PID_C 中的帧解析器和联合块提供的结果。如果处理实现没有错误，则数据可被映射到主机缓存器用来存储。数据在主机存储器的缓存器中成功存储后，接收器处理块可为连接传输发送侧将来传输的 TCP 应答信号。

在步骤 150 中，可确定报头/数据的边界。控制路径中的处理结果可确定作为报头处理的分组部分和作为数据或有效负荷处理的分组部分之间的边界。虽然可将数据移到主机缓存器中，报头会被 TEEC 耗尽或移到单独的主机缓存器中以统计、调试或进一步的处理。

在步骤 160 中，属于特定 5 元组连接的接收到分组中的数据可映射到为那个特定连接分配的主机常驻缓存器中。分配的缓存器已经通过应用软件或协议处理层(例如 TCP 层)预分配。在一个示例中，分配的缓存器是临时缓存器。图 9 根据本发明的实施例，说明了可将输入分组数据映射和复制到主机常驻缓存器或缓存器组中的系统实施例。在一个示例中，TEEC 可直接将数据复制到主机缓存器中，而最初不需要保留位于 NIC 上的数据。

主机可通过使用描述每个有例如主机存储器的物理地址和字节长度的缓存器的列表结构来描述缓存器。主机还可以其它方式来描述缓存器，例如通过页表结构。TEEC 可读取缓存器信息，并可构造输入分组的 TCP 序列号和主机缓存器之间的映射。特定的 TCP 序列号可被映射到例如特定缓存器的初始点或特定缓存器的一些偏移量中。当缓存器被分配到卸载连接，可初始化映射。当接收到分组，根据例如长度和 TCP 序列号，将分组与缓存器映射信息比较。根据比较，一个分组可被映射到一个或多个缓存器中。依次对 DMA 引擎发出一个或多个指令，以将分组数据移入主机缓存器或缓存器中。

TEEC 可将在 TCP 报头分组中传送的 TCP 序列号的第一有效负荷字节映射到 TEEC 的主机常驻缓存器中的偏移量。在步骤 170 中，TEEC 可将 TCP 段数据直接存放入主机缓存器，例如，TCP 缓存器和预置应用软件缓存器。TCP 数据可在例如主机存储器中重新组合，而不需要在 TEEC 上局部复制。它还可节省在主机上复制数据、保留完全的 CPU 周期和保存存储器子系统中的带宽。因此，可支持零复制操作。

当 TCP 连接被卸载到 TEEC 时，它可包括例如象 RCV_NXT 映射的支撑物、下一个期望字节的 TCP 序列号、特定的主机基地址和缓存器中的偏移量。当每次多个字节被 TCP 接收并认可时，调节变量 RCV_NXT。对应于 RCV_NXT 的缓存地址是缓存器[I],NXT_addr，并将其调节为指向第一缓存器中可用的第一字节。当将缓存加入到现有列表的尾部时，将可用缓存器调到最大。当缓存器完全耗尽时，通过缓存器所有者将缓存器返回主机用来消耗。当缓存器返回到主机时，因为 PUSH 位在输入 TCP 段中进行设置，调节 RCV_NXT 映射使其指向下一个缓存器的第一字节。当每次耗尽列表并且新的缓存器分配给卸载的 TCP 连接时，可重复此行为。

先卸载连接然后分配缓存器。映射任一 TCP 段到主机缓存器的处理可由计算它的 TCP 序列号和 RCV_NXT 值之间的增量(例如差值)开始。然后将增量加入基值，偏移量加入第一缓存器(例如缓存器[I].NXT_addr)。如果增量超出第一缓存器的长度，那么加入第二缓存器的长度。持续此处理直到在缓存器中建立 TCP 序列号映射。计算可考虑例如预置缓存器的变量大小。然后 TEEC 确定识别的缓存器是否对整个 TCP 段有足够的存储空间或段是否溢出到下一个缓存器。根据决定和计算结果，一系列 DMA 命令以接收的 TCP 段、主机地址和长度中偏移量产生。当帧中数据需要在超出 TEEC 拥有的当前列表的缓存器中存储时，TEEC 要么放弃帧并不对同等 TCP 应答，要么临时存储直到能从主机得到另一缓存器。

对于接收的 TCP 段，如以下部分的说明执行一个或多个处理地址。在一个实施例中，当 TCP 序列空间限制大约为 2^{32}-1 字节时，TCP 序列空间的操作是取模 32。但是，可实现其它多模结构。

根据本发明，以下伪代码描述了将数据从 TCP 段移到缓存器列表处理的实施例。为了简化，处理 PUSH 位或在当前缓存器列表中没有空间的代码被省略。

1./*检查 TCP 序列号范围(TCP 有效负荷的第一字节的 TCP Seq#，最后字节的 TCP 序列#)是否在 RCV 窗口(在 RCVNXT 和 RCVNXT+TCP 窗之间)内*/

1A./*如果复制帧(之前已接收了所有的字节)则丢弃该帧*/

if TCP Sequence # of last Byte < RCVNXT then drop frame

1B. check that RCVNXT < TCP Sequence # of first Byte < (RCVNXT + TCP window)

1 C. check that RCVNXT < TCP Sequence # of last Byte < (RCVNXT + TCP window)

1 D./* 如果之前已接收了一些字节，则不复制这些字节*/

if TCP Sequence # of first Byte < RCVNXT then TCP Sequence # of first Byte = RCVNXT ;

2./*找到缓存器列表中正确的输入项*/

Segment Length = TCP Sequence # of last byte of TCP payload- TCP Sequence # of first byte ;

/*增量保持 TCP 序列号的差异到帧的第一字节的位置。它也是缓存器空间到应当用于存储它的第一字节之间的距离*/

Delta = (TCP—Sequence # of first Byte-RCV NXTfrom context)

/*支撑物增量*/

i=0; /*用于动态指向对应于 RCVNXT 的缓存器*/

/主机缓存器列表中的一些字节。缓存器[0]可能已经使用。需要算出剩下的/

if (Delta < (Host Buffer List. Buffer [0]. length- (Host Buffer List. Buffer [0]. NXTAddr- Host Buffer List. Buffer [0]. Phy~Addr)))

{Delta+= (Host-Buffer List. Buffer [0]. NXTAddr- Host Buffer List.

Buffer [0]. Phy~Addr) ;

I else { Delta-= (Host-Buffer-List. Buffer [0]. length- (Host Buffer List. Buffer [0]. NXTAddr- Host Buffer List. Buffer [0]. Phy~Addr)) ;

5. Do while {Delta-Host-Buffer-List. Buffer [i]. length > 0} { Delta-= Host Buffer List. Buffer [i]. length ; j++ ; } ; }

6./*变量 i 指向第一缓存器后的第 i 个缓存器，其中数据公布应当在此处开始。增量为此缓存器中的偏移量*/

7. Bytes to DMA = Segment length ;

8./*DMA 到第一缓存器中，DMA 数据语法(来自地址，发往地址，长度)*/

DMA Data (TCP Sequence of first byte, Host Buffer List. Buffer [i]. Phy~Address+ Delta, Host Buffer List. Buffer [i]. length-Delta)

10./*判断缓存器满具有以下语法(所写的第一字节，长度)，并且如果是满的话，则返回结果 1*/

if (buff full = is buffer- Full (Host Buffer List. Buffer [i]. Phy~Address+ Delta, Host Buffer List. Buffer [i]. length-Delta)) then return buffer to owner ();

11. BytestoDMA-= HostBufferList. Buffer [i]. length-Delta ;

/*已经 DMA 到第一缓存器中的字节*/

12. StartTCPSeq = TCP Sequence of first byte + (Host Buffer List. Buffer [i]. length-Delta);

/*如果需要的话 DMA 到下一缓存器中*/

13. Dowhile {BytestoDMA > 0}

14. {

if (Bytes to DMA > Host Buffer List. Buffer [i]. Length) DMA data (StartTCPSeq, HostBufferList. Buffer [i]. Phy~Address, Host Buffer List. Buffer [i]. Length)

else DMA data (StartTCPSeq, Host BufFer List. Buffer [i].

```
PhyAddress, BytestoDMA) ;
    BytestoDMA-= HostBufferList. Buffer [i]. length ;
    Start TCP Seq += Host Buffer List. Buffer [i]. length i++ ;
    If i > maxbuffers then goto no more buffers ;
```

当缓存器沿着 TCP 窗口移动到右侧消耗时，可更新基值序列号和主机缓存器信息列表。

图 10 根据本发明的一个实施例，说明了典型的发送路径。TEEC 可包括例如物理层 (PHY) 180、MAC 层 190、报头构造器 200、内容预取 210、定时器 220，、传输处理器 230，数据和控制 块 240、应答块 250、调度程序 260 和 DMA 引擎 270。其组成连接在图 10 中说明。定时器 220 可包括例如 TCP 状态码传输和传输定时器。调度程序 260 适于例如开窗和/或传输判决。DMA 引擎 270 可包括例如 XSUM 块 280 或其它数据特定处理。这个可包括将数据插入到由主机提供的数据，并计算 CRC 值。数据处理并不仅限于这些功能。

在发送路径上，支持 L4 和更高级可包括附加的复杂度和功能。传输 可包括执行例如一个或多个以下功能：安排传输流、通过 DMA 传输数据、取得内容、传输处理、增加 L5 或更高级 和 TCP/IP 报头，以及适当的添加到所有这些报头域中、提供定时器、以及 L2 传输。

调度程序 260 可确定下一个服务流。调度程序 260 还可处理多路 L2 和 L4 和更高级的业务量。对于 L4 和更高级的业务量，安排特定 TCP/IP 流的传输判决可根据例如以下因素中的一个或多个：主机侧传输有效数据；比如当远端 TCP 连接已经关闭 TCP 窗口时的远端缓存器状态；防止以太网介质上 TCP 连接的大量潜在的时间竞争造成资源缺乏；来自接收侧的有效性的 TCP 应答；转发代表 TCP 连接信息的需要；以及从主机传输到 TEEC 中流优先级或服务质量(QoS) 信息。

利用一些或所有以上确定的信息或其它信息，调度程序 260 可取下一个流传输。调度程序 260 可从内容信息中取出指向下一个主机常驻缓存器的指针。调度程序还可 260 给 DMA 引擎 270 编程以得到数据和存储数据，例如在弹性缓存器 281 中。虽然显示了弹性缓存器 281，但是根据本发明不同的实

施例，本发明并不限于此，并且可利用片内 FIFO 缓存器或其它适当的存储器或缓存器设备存储数据。

DMA 引擎 270 可从主机 缓存器或缓存器将数据传输到例如片内、发送侧 FIFO 缓存器。可在正传输的数据上计算 IP 校验和(IPv4) 和 TCP 校验和。计算的实现与数据移动同时进行。更高级数据处理可在此集进行。

例如通过内容预取 210 从中央内容源取得流内容。访问中央内容源可增加它的所有用户中的同步机械装置的有效性，以保证数据的完整性和一致性。在非期望执行冲突最小时，同步机械装置非常有效。选择流的内容可被提供给传输处理器 230，比如 CPU 和/或有限状态机(FSM)。

传输处理器 230 适于例如执行 TCP/IP 和更高级的代码、更新内容和产生 TCP/IP 和更高级报头 变量以取代报头。可存储更新的内容。此级的处理可由一个或多个包括一个或多个处理器、状态机或混合处理器的级来实现。

报头构造器 200 可使用传输处理器 230 产生的报头变量，可产生 TCP/IP 和更高级报头，并可将 TCP/IP 和更高级报头附加到数据前以传输。使用部分从 DMA 引擎 270 得到的校验和结果，报头构造器 200 可确定校验和域，并可将他们放置到各个报头。传输处理并不仅限于此特定数量级处理，并可进行不同级的最佳处理。

定时器 220 可由传输处理器 230 提供，并更新未来的定时器事件列表。当 L4 和更高级的处理完成时，L2 处理和传输可遵循传统的以太网控制器的一般执行步骤。

重发事件类似于标准传输，除了例如可从如前所述的主机缓存器或其它临时控制缓存器中取得的要重发的数据之外。可计算这一数据的地址。主机缓存器地址的计算会更加复杂。接收路径部分描述的同一映射功能可用于重发 TCP 序列号范围。一旦确定了缓存器地址，便会进行以上描述的余下的传输处理。

本发明的一个或多个实施例可具有以下所述优点中的一个或多个。

印脚的减少使得能够不需要外部存储器单片执行。印脚减少可提供与传统非卸载以太网控制器相似大小，因此可允许在受限于 LAN 部件实时状态分配

的服务器和客户机的母板上集成。换句话说，可用来解决母板上网络(LOM)应用软件。这是客户机和服务器不断紧缩形式因素的有利面。

由于可不必与外部存储器接口，因此该解决方案的成本可以降低。不仅可节省外部存储器的成本，而且 TEEC 也可变得更便宜。可不需要或减少对可接口存储器和用于 I/O 缓存器的机器的需要，便可驱动此机器。同时，这可允许具有更少的管脚和更高的性能的更小封装。

功率和热量考虑可能是对于数据中心的发展的基本抑致因素。通过消除或简化外部存储器，TEEC 可降低其功耗。由于要耗散的热量较少，因此可提供更紧凑的服务器。

可节约与数据的临时缓存相关联的延时。一些应用如分布式数据库、聚类、高性能计算机(HPC)、服务质量(QoS)应用以及其它应用可得益于大量节约的延时。

可能没有存储器和 TEEC 的速度耦合。当外部存储器附加到 TEEC 的结构上时，该存储器的速度和宽度可能会影响内部结构。此影响对于更改的有线速度来说甚至更大。由于没有可能进行外部连接，因此大大简化了内部结构。

这可减少由 NIC 上 TEEC 所用的存储器，只用传统的 FIFO 缓存器用于匹配有线、缺少内部处理以及主机总线速度。这种结构的存储器需求不可通过若干连接来扩展，并且可能对 LAN 或 WAN 配置不敏感。成本和大小受带宽距离乘积的影响较少，这种问题在大型(环球)快速网络中可能会更加恶化。

因此，本发明可实现于硬件、软件或硬件和软件的组合。本发明可以集中方式实现于一个计算机系统中，或以分布的方式实现于分布在几个互连接的计算机系统之间的不同单元中。任何种类的计算机系统或其适于实施这里所述的方法的装置均是适合的。硬件和软件的典型组合可以是带有计算机程序的通用计算机系统，该程序在加载并被执行时，可控制所述计算机系统，使得它可实施这里所述的方法。

本发明的一些部分还可嵌入到计算机程序产品当中，该产品包括了实现这里所述的方法的所有特征，并且在载入计算机系统时可实施这些方法。本文中的计算机程序意指用任何语言、代码或注释的指令集的任何表达，这些指令

用于使系统具有信息处理能力，以便或者直接或者在结合了以下二者之后执行特定功能：a)转换为另一种语音代码或注释；b)用不同的材料形式进行复制。

虽然本发明是参考特定实施例进行描述的，但对于本领域的技术人员而言，在不背离本发明的范围的情况下，可进行各种等效变化并可用等效物替代。此外，在不背离本发明范围的情况下，可进行多种修改，以适应特定条件或本发明示教的内容。

因此，本发明不限于所公开的特定实施例，本发明将包括落于权利要求范围中的所有的实施例。

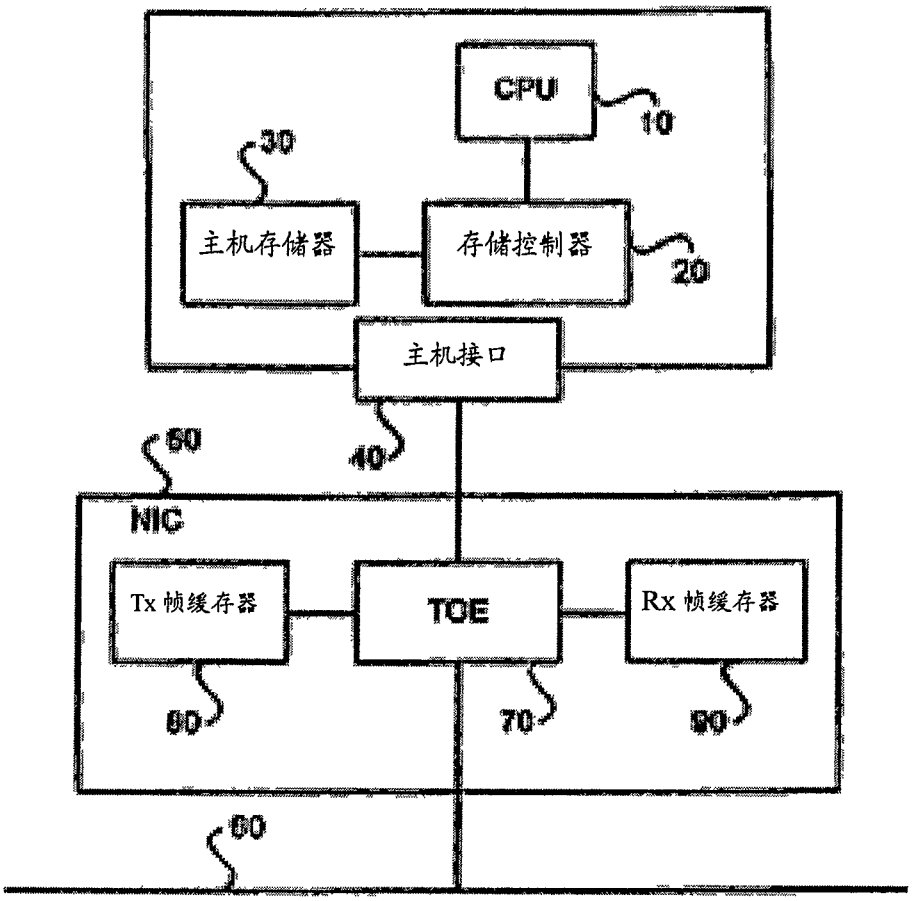

CPU
10
30
主机存储器
存储控制器
20
主机接口
40
50
NIC
Tx 帧缓存器
TOE
Rx 帧缓存器
80
70
90
60

图 1

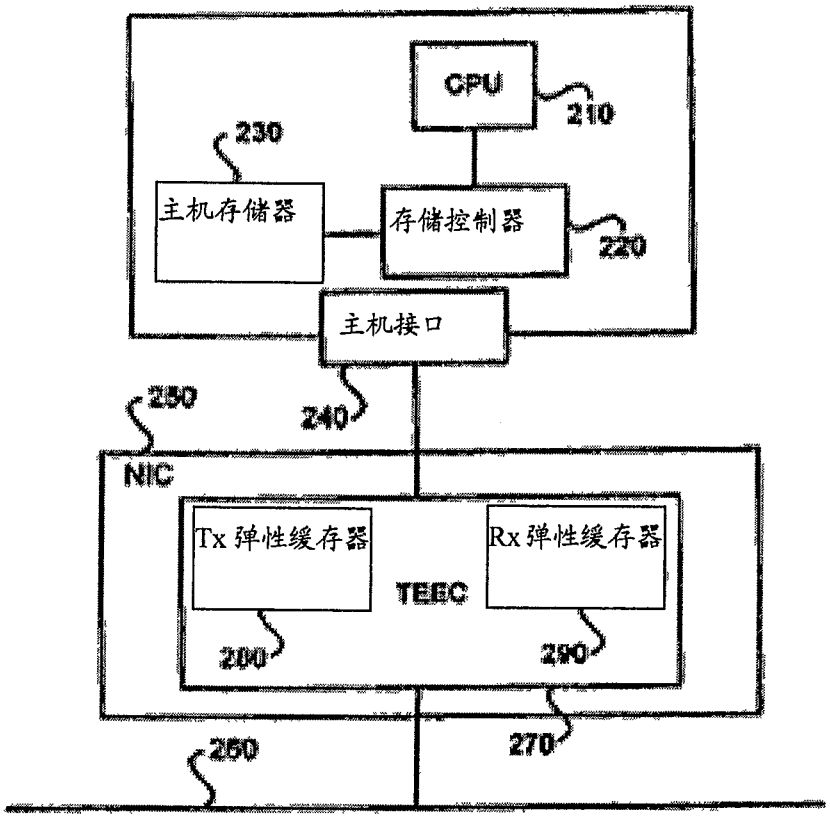

CPU
210
230
主机存储器
存储控制器
220
主机接口
240
250
NIC
Tx 弹性缓存器
Rx 弹性缓存器
TEBC
280
290
270
260

图 2

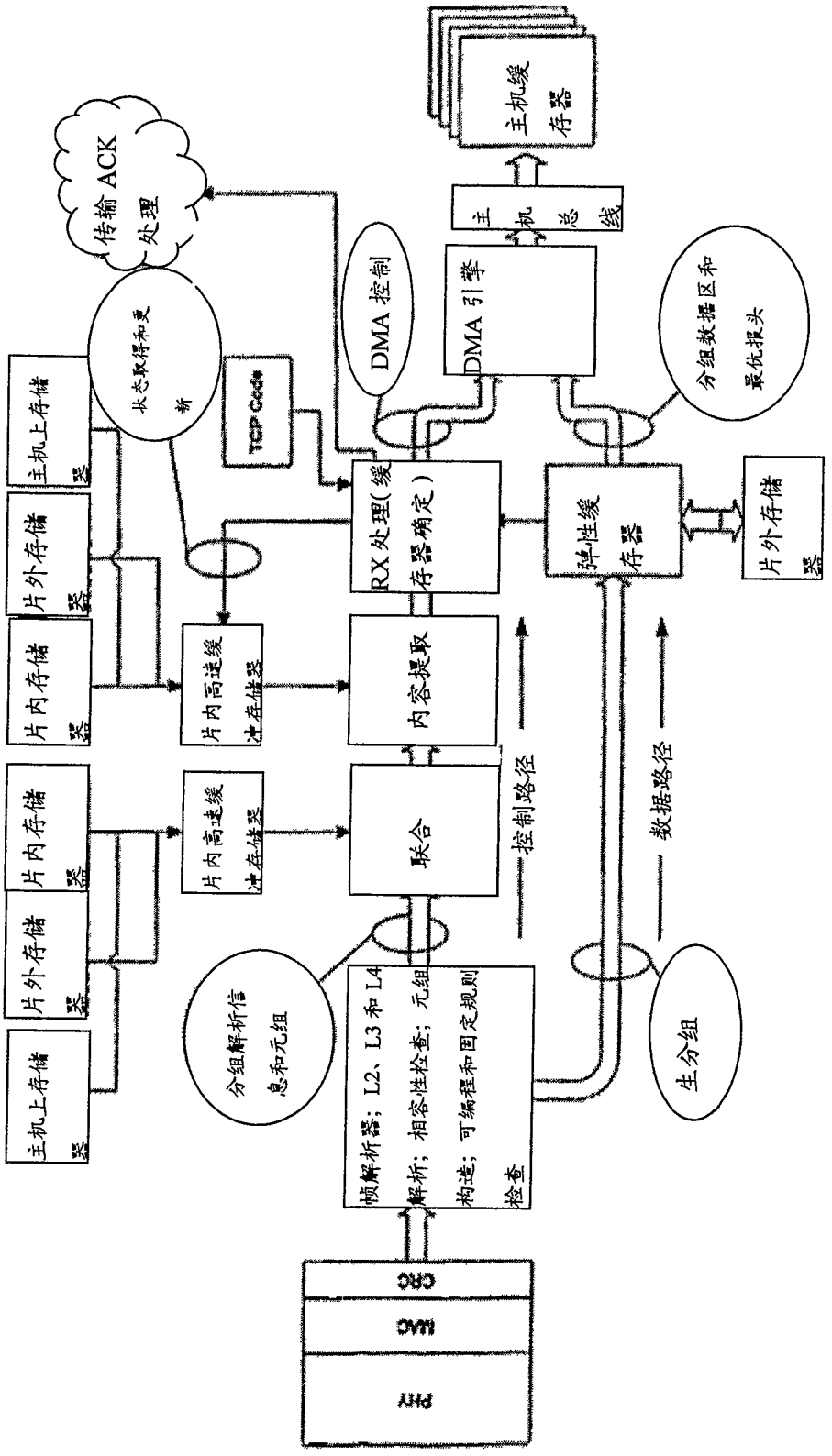

主机缓存器
主机总线
DMA 引擎
DMA 控制
分组数据区和最优报头
传输 ACK 处理
状态取得和更新
TCP Code
主机上存储器
片外存储器
片内存储器
片内高速缓冲存储器
RX 处理(缓存器确定)
内容提取
联合
弹性缓存器
片外存储器
控制路径
数据路径
分组解析信息和元组
帧解析器；L2、L3 和 L4 解析；相容性检查；元组构造；可编程和固定规则检查
生分组
PHY
MAC
CRC

图 3

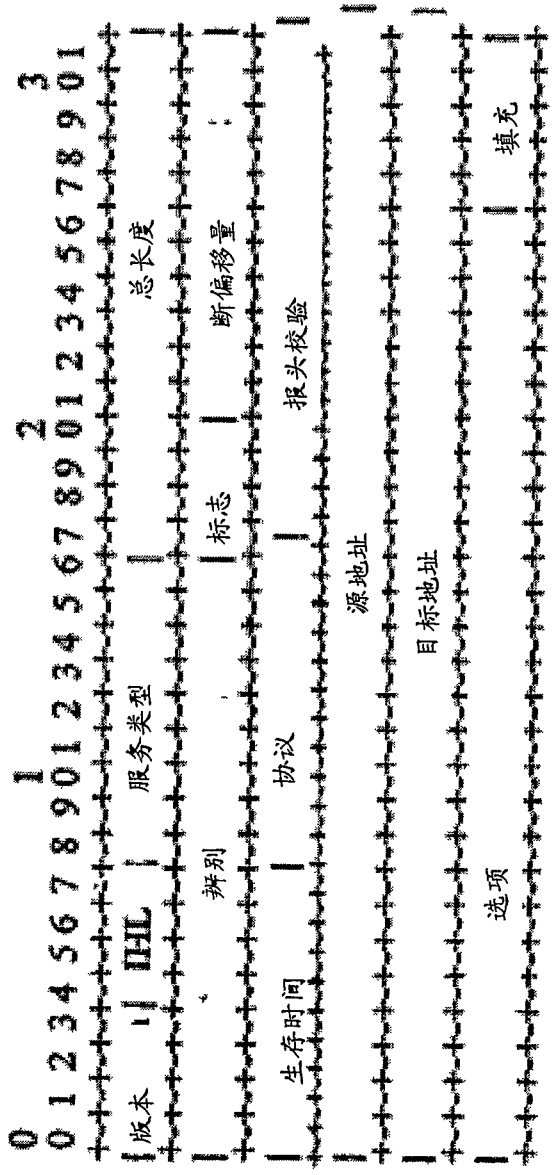

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
版本
IHL
服务类型
总长度
辨别
标志
断偏移量
生存时间
协议
报头校验
源地址
目标地址
选项
填充

图 4

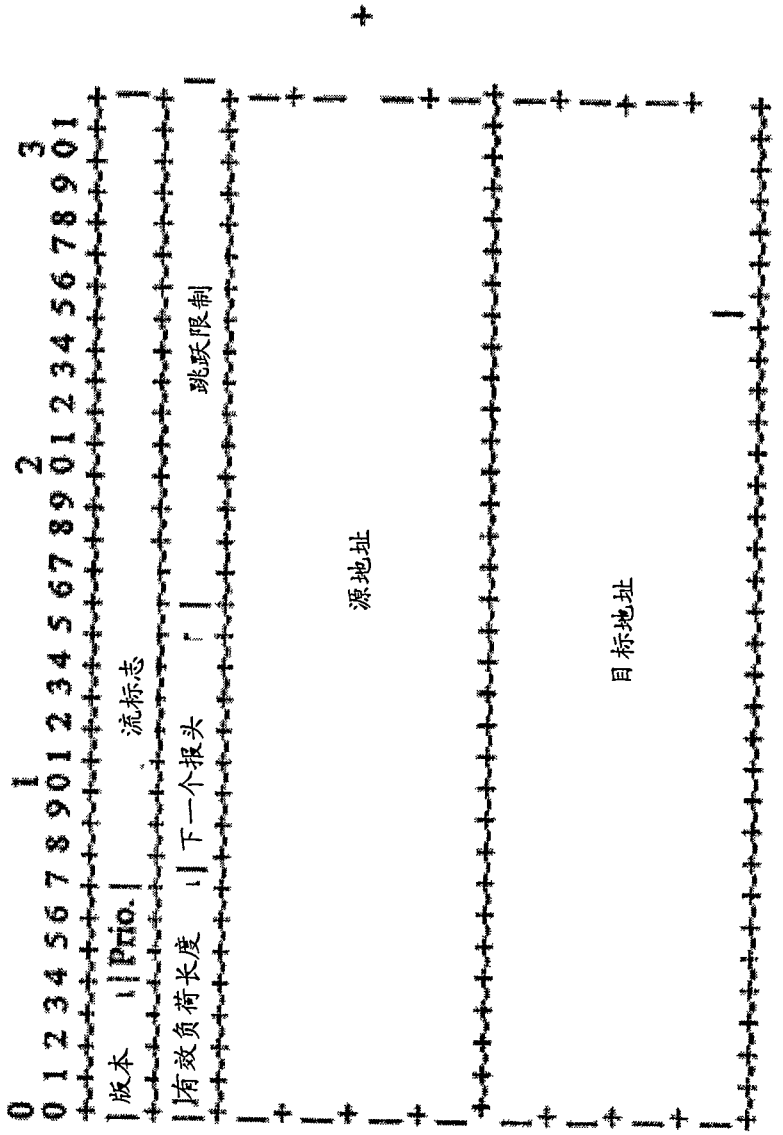

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
版本
Prio.
流标志
有效负荷长度
下一个报头
跳跃限制
源地址
目标地址

图 5

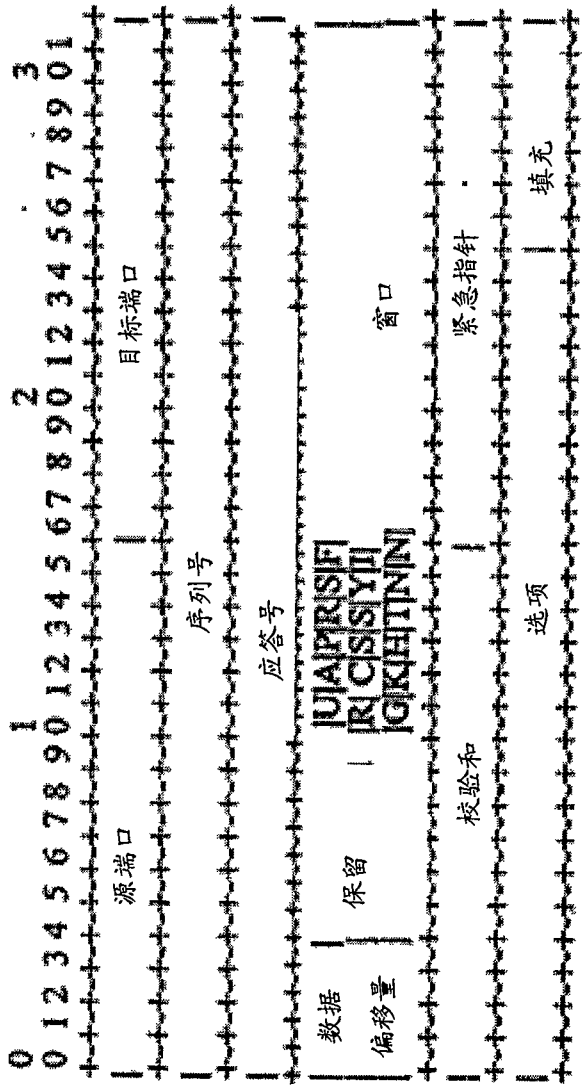

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
源端口
目标端口
序列号
应答号
数据偏移量
保留
窗口
校验和
紧急指针
选项
填充

图 6

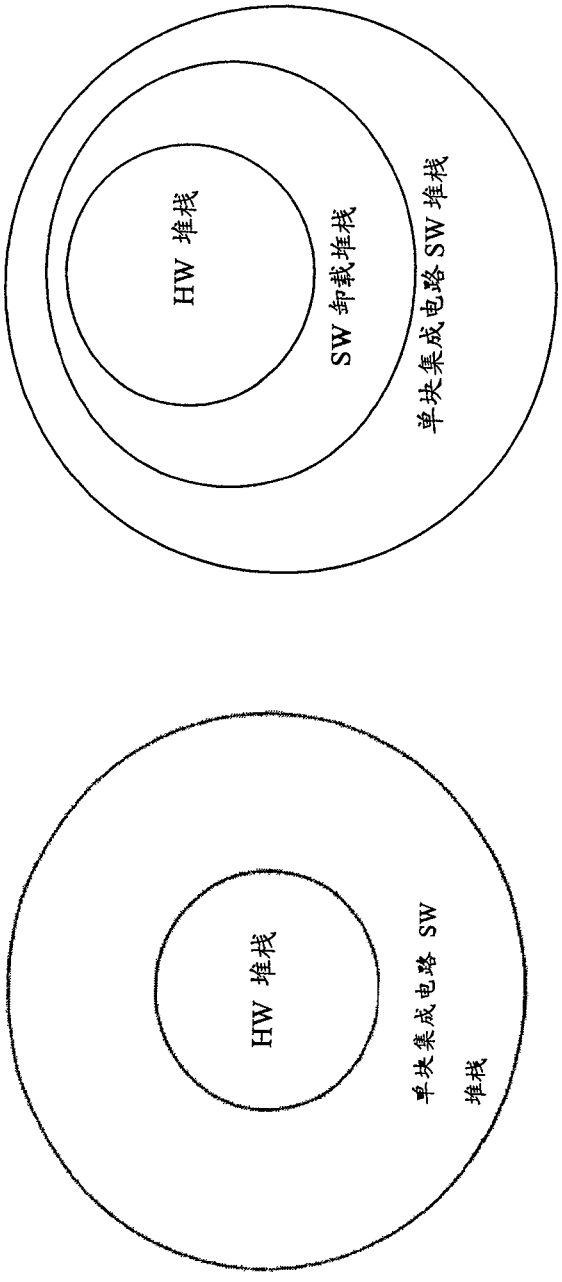


图 7

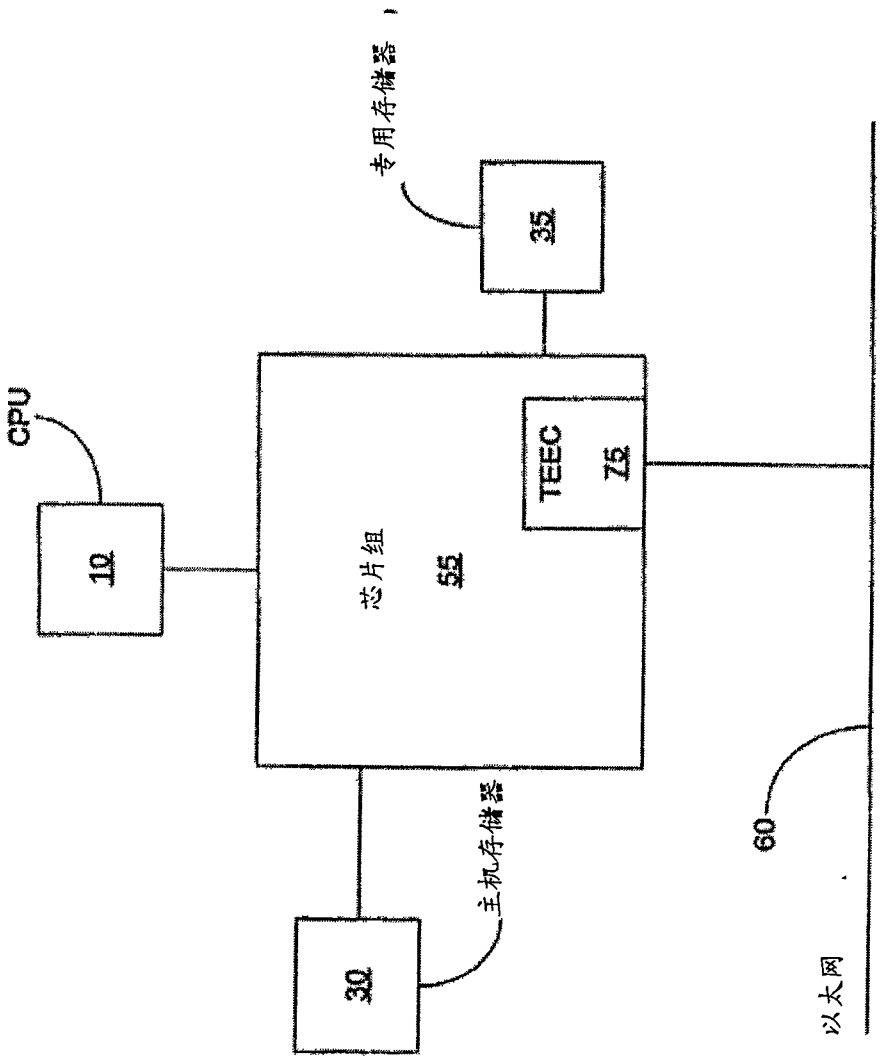


图 8A

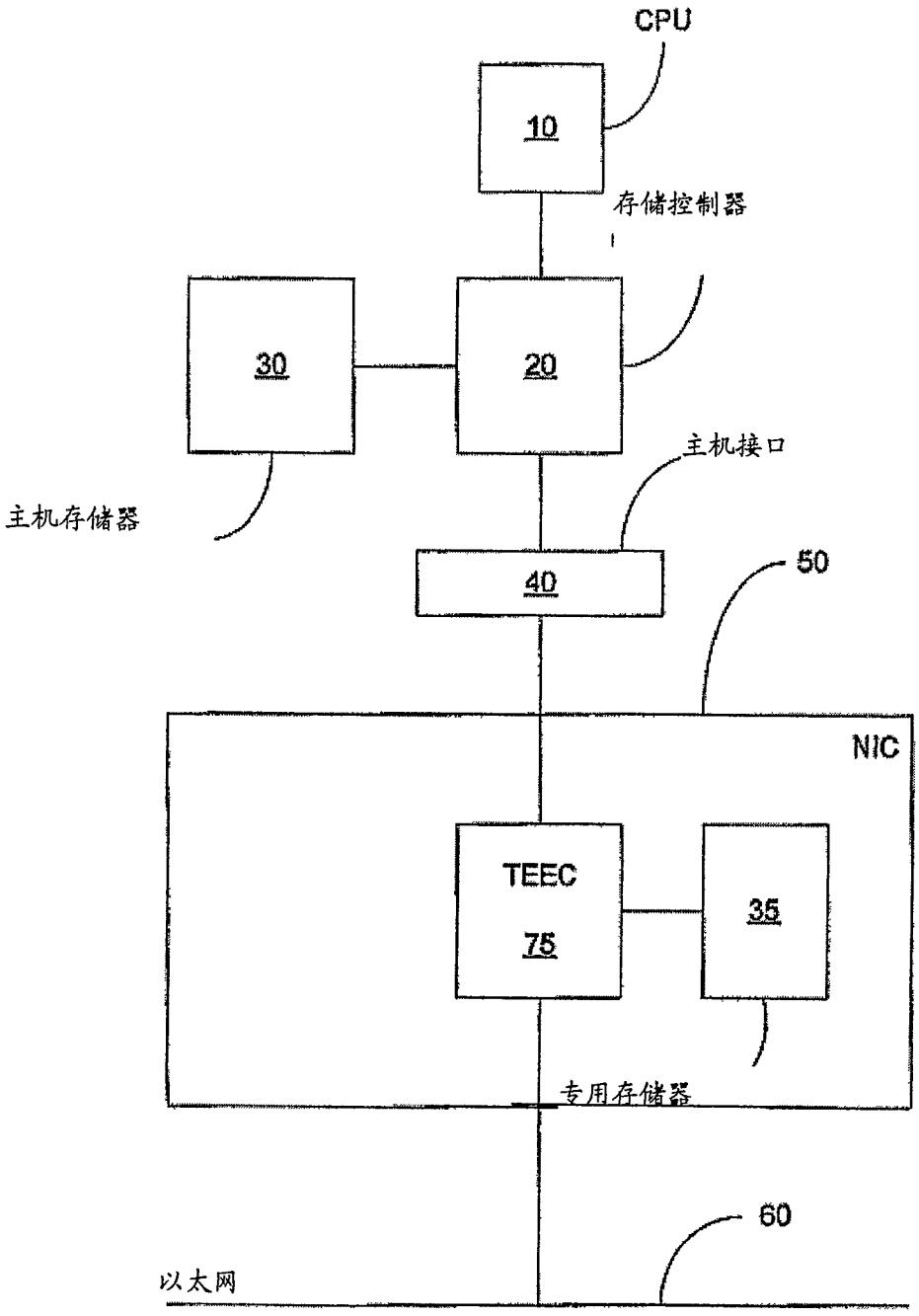


图 8B

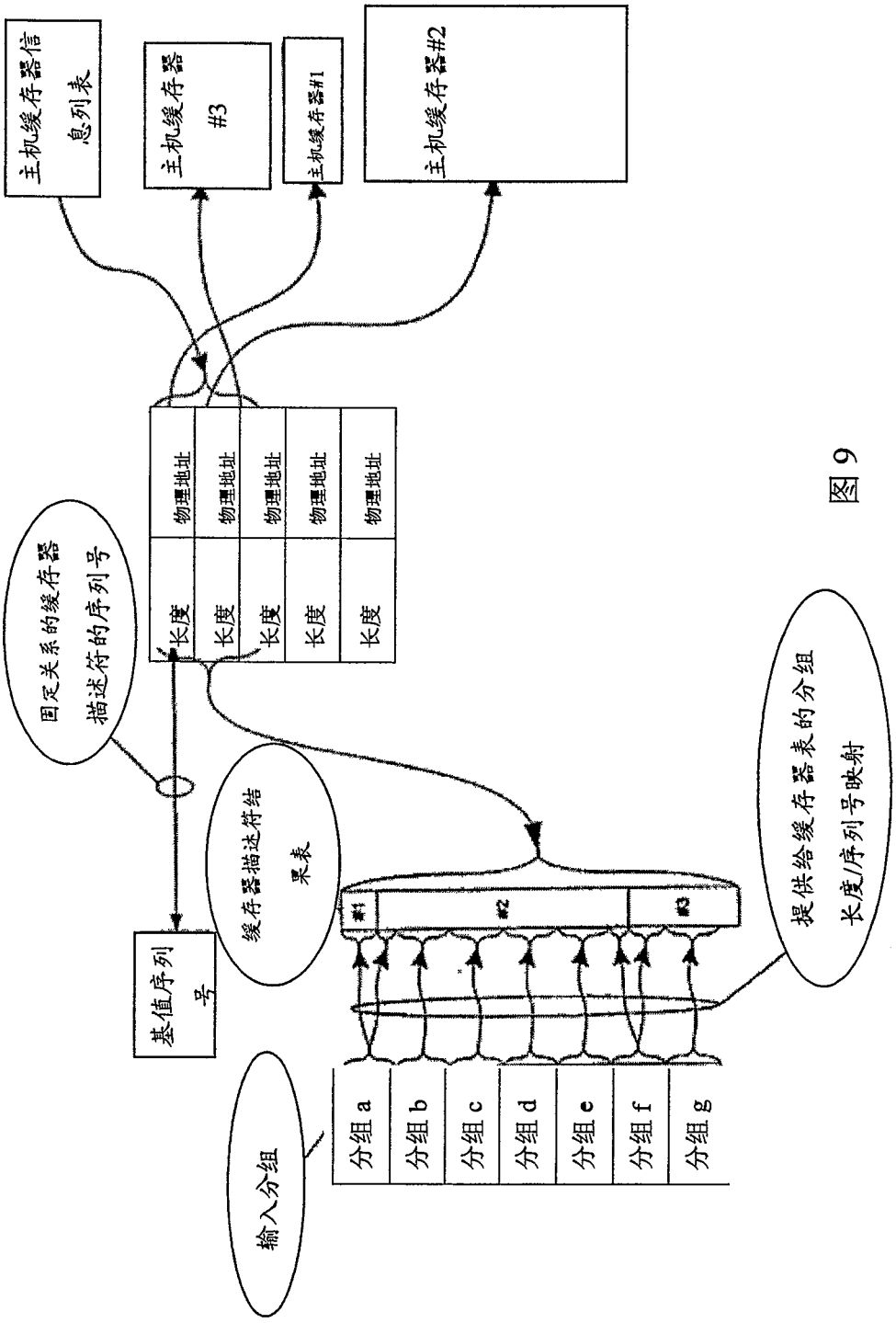

主机缓存器信息列表
主机缓存器#3
主机缓存器#1
主机缓存器#2
固定关系的缓存器描述符的序列号
长度
物理地址
基值序列号
缓存器描述符结果表
输入分组
分组 a
分组 b
分组 c
分组 d
分组 e
分组 f
分组 g
提供给缓存器表的分组长度/序列号映射

图 9

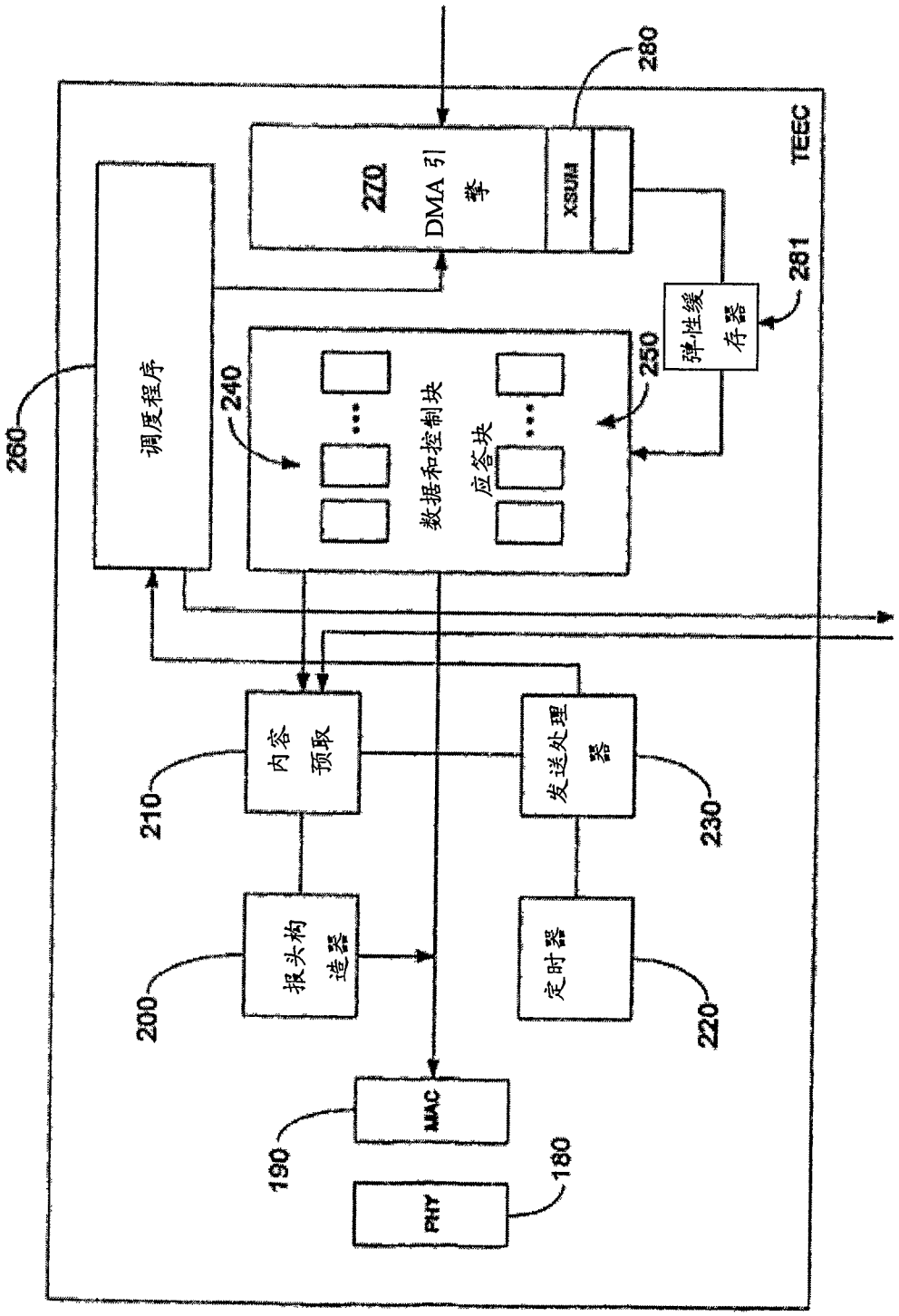

TEEC
280
270
DMA 引擎
XSUM
281
弹性缓存器
250
240
数据和控制块
应答块
260
调度程序
210
内容预取
发送处理器
230
200
报头构造器
定时器
220
190
MAC
180
PHY

图 10

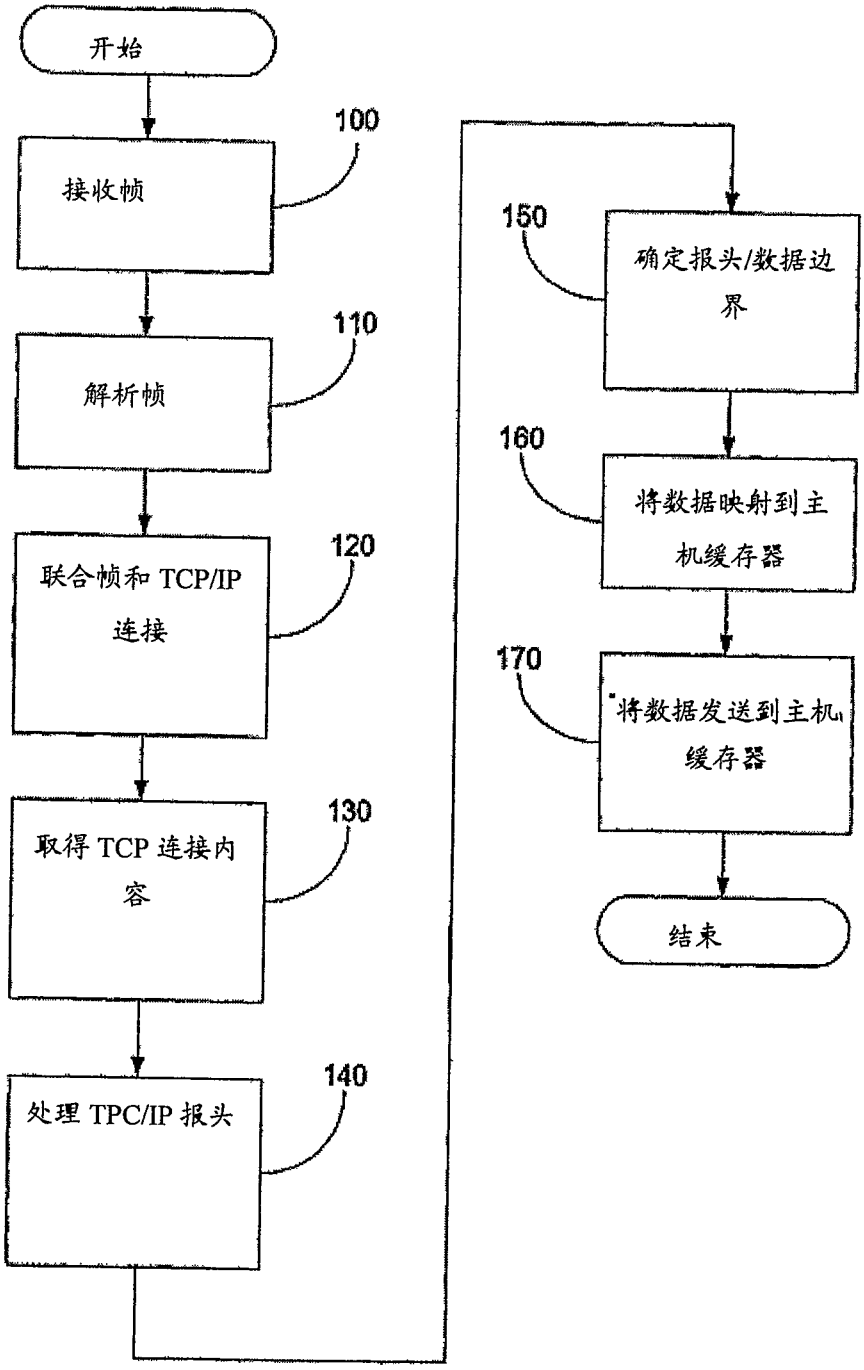

开始
100
接收帧
110
解析帧
120
联合帧和 TCP/IP 连接
130
取得 TCP 连接内容
140
处理 TPC/IP 报头
150
确定报头/数据边界
160
将数据映射到主机缓存器
170
将数据发送到主机缓存器
结束

图 11

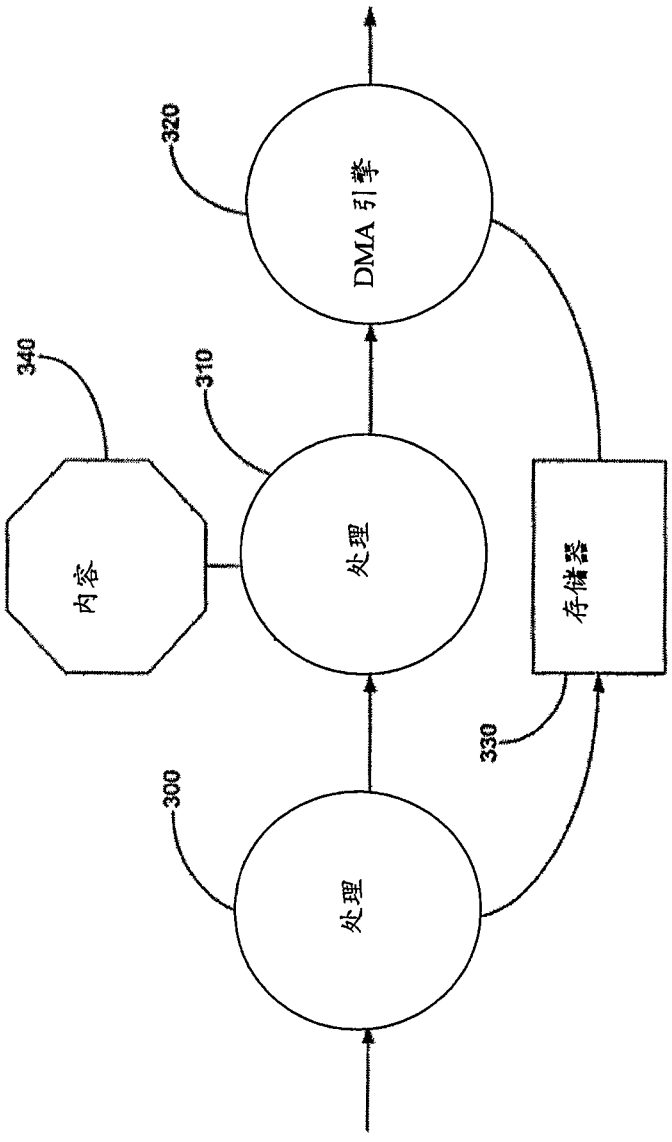


图 12

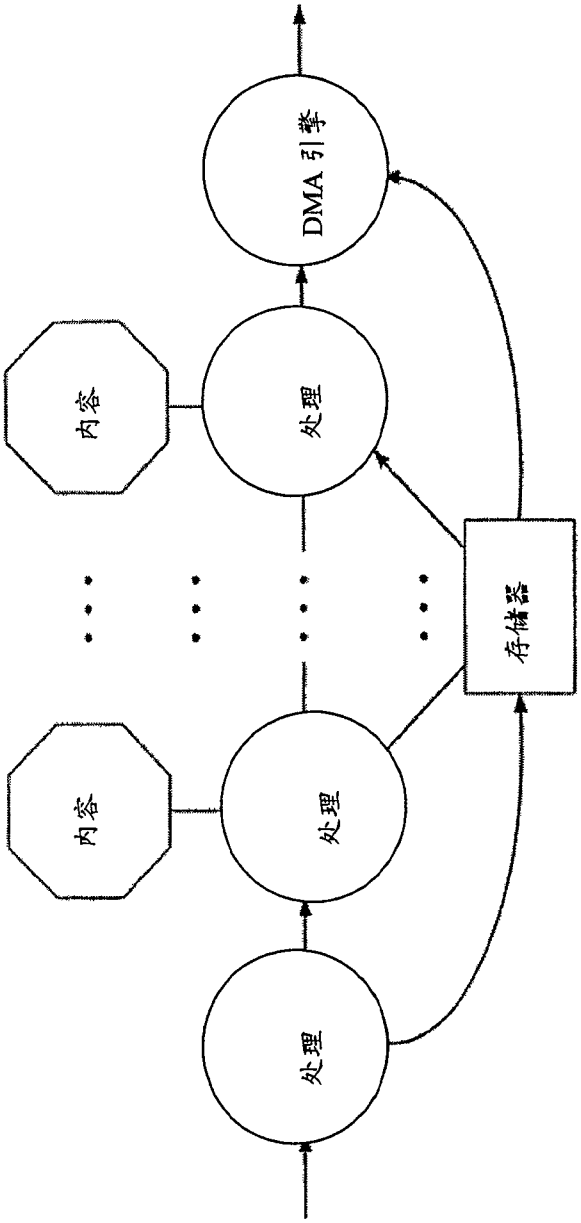


图 13

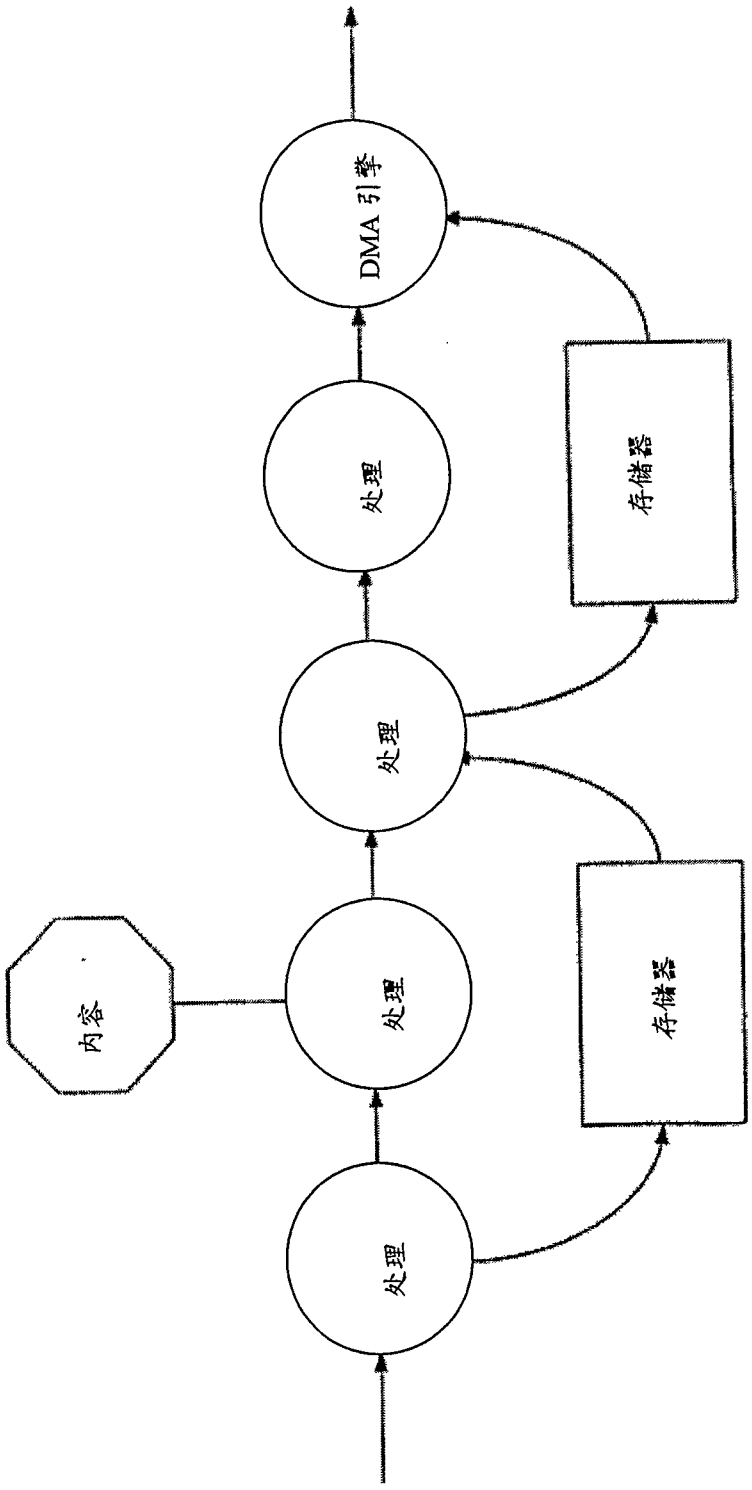


图 14