

(12) 发明专利

(10) 授权公告号 CN 101253745 B

(45) 授权公告日 2011.06.22

(21) 申请号 200680026247.4

(22) 申请日 2006.07.18

(30) 优先权数据

60/700,544 2005.07.18 US

(85) PCT申请进入国家阶段日

2008.01.18

(86) PCT申请的申请数据

PCT/IB2006/004098 2006.07.18

(87) PCT申请的公布数据

W02007/069095 EN 2007.06.21

(73) 专利权人 博通以色列研发公司

地址 以色列拉马特甘市

专利权人 埃利泽·阿朗

拉弗·沙洛姆

谢伊·米兹拉希

达夫·赫什菲尔德

阿维弗·格林伯格

阿萨夫·格伦费尔德

埃利泽·塔米尔

盖伊·科里姆

奥里·哈尼格比

(72) 发明人 埃利泽·阿朗 拉弗·沙洛姆

谢伊·米兹拉希 达夫·赫什菲尔德
 阿维弗·格林伯格
 阿萨夫·格伦费尔德
 埃利泽·塔米尔 盖伊·科里姆
 奥里·哈尼格比

(74) 专利代理机构 深圳市顺天达专利商标代理有限公司 44217
 代理人 蔡晓红 王小青

(51) Int. Cl.

H04L 29/06(2006.01)

H04L 12/56(2006.01)

(56) 对比文件

WO 2004021150 A2, 2004.03.11,
 WO 03021436 A2, 2003.03.13,
 WO 2004112350 A1, 2004.12.23,
 CN 1494293 A, 2004.05.05,
 CN 1520112 A, 2004.08.11,

审查员 袁堃

权利要求书 2 页 说明书 13 页 附图 10 页

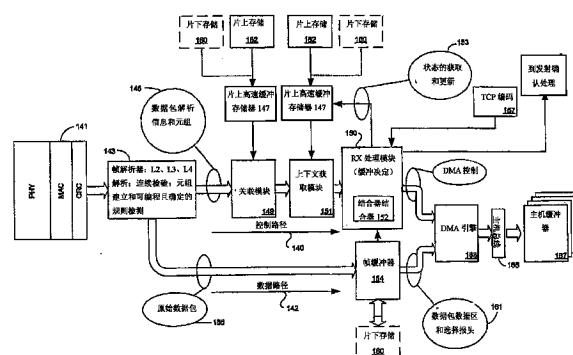
(54) 发明名称

用于透明 TCP 卸载的方法和系统

(57) 摘要

本发明涉及一种透明传输控制协议 (TCP) 卸载的方法和系统，所述方法的各个方面包括：在网络接口卡处理器中收集 TCP 段，但不将状态信息转移到主机系统。收集的 TCP 段缓存在结合器中。结合器可检验与收集的 TCP 段相关的网络流在流查找表 (FLT) 中具有入口。当 FLT 已满时，结合器可关闭当前入口，并将网络流指派到可用入口。结合器也可更新 FLT 中的信息。当终止 TCP 段收集的事件发生时，结合器可基于收集的 TCP 段生成单个聚合的 TCP 段。可将聚合的 TCP 段和状态信息传送到主机以进行处理。

CN 101253745 B



1. 一种处理网络信息的网络处理的方法,其特征在于,该方法包括:

通过网络接口卡(NIC)处理器收集接收的用于特定网络流的至少一个传输控制协议(TCP)段,且在每次所述至少一个传输控制协议段被接收时,不将关于所述特定网络流的状态信息转移到主机系统;

在终止事件发生后,生成新的传输控制协议段,所述新的传输控制协议段包括所述收集的至少一个传输控制协议段;以及

将所述生成的新的传输控制协议段和所述新的传输控制协议段的状态信息传送到所述主机系统以进行处理;

所述方法进一步包括通过聚合所述特定网络流的多个所述收集的传输控制协议段的至少一部分,生成所述新的传输控制协议段。

2. 根据权利要求1所述的方法,其特征在于,进一步包括为流查找表(FLT)中的所述特定网络流更新生成的流查找表入口中的信息。

3. 根据权利要求2所述的方法,其特征在于,所述流查找表入口包括以下中的至少一个:

元组;

传输控制协议序列号;

传输控制协议确认号;以及

传输控制协议有效载荷长度;

所述元组包括:

网络协议(IP)源地址;

网络协议目的地地址;

源传输控制协议端口;以及

目的地传输控制协议端口。

4. 根据权利要求1所述的方法,其特征在于,进一步包括生成所述生成的新传输控制协议段的传输控制协议报头,所述传输控制协议报头指示与所述收集的至少一个传输控制协议段相关的传输控制协议有效载荷字节的总数。

5. 根据权利要求1所述的方法,其特征在于,进一步包括生成所述生成的新传输控制协议段的传输控制协议报头,所述传输控制协议报头指示所述收集的至少一个传输控制协议段的第一时戳选择。

6. 一种处理网络信息的网络处理的系统,其特征在于,该系统包括:

网络接口卡(NIC)处理器,用于收集接收用于特定网络流的至少一个传输控制协议(TCP)段,且在每次所述至少一个传输控制协议段被接收时,不将关于所述特定网络流的状态信息转移到主机系统;

所述网络接口卡处理器,用于在终止事件发生后,生成新的传输控制协议段,所述新的传输控制协议段包括所述收集的至少一个传输控制协议段;以及

所述网络接口卡处理器,用于将所述生成的新的传输控制协议段和所述新的传输控制协议段的状态信息传送到所述主机系统以进行处理;

所述网络接口卡处理器,用于通过聚合所述特定网络流的多个所述收集的传输控制协议段的至少一部分,生成所述新的传输控制协议段。

7. 根据权利要求 6 所述的系统，其特征在于，所述网络接口卡处理器，用于为流查找表(FLT) 中的所述特定网络流更新生成的流查找表入口中的信息。

8. 根据权利要求 7 所述的系统，其特征在于，所述流查找表入口包括以下中的至少一个：

元组；

传输控制协议序列号；

传输控制协议确认号；以及

传输控制协议有效载荷长度；

所述元组包括：

网络协议 (IP) 源地址；

网络协议目的地地址；

源传输控制协议端口；以及

目的地传输控制协议端口。

用于透明 TCP 卸载的方法和系统

技术领域

[0001] 本发明涉及 TCP 数据和相关的 TCP 信息的处理。更具体地说，本发明的实施例涉及用于透明 TCP 卸载 (transparent TCP offload) 的方法和系统。

背景技术

[0002] 当前多种用于降低 TCP/IP 栈处理功率的方法。在 TCP 卸载引擎 (TOE) 中，卸载引擎执行全部或大部分 TCP 处理，这些 TCP 处理出现在数据流的上层。这个方法存在各种缺陷。TTOE 与操作系统紧密相联，因此其解决方案需要依赖操作系统，并可能需要对操作系统进行改变以实现对其的支持。TTOE 可能需要采用某种手动配置的肩并肩的栈处理方案，这可由其应用完成，例如，为加速连接明确地指定套接字地址族。TTOE 也可需要某些由 IT 管理员完成的手动配置，例如，为加速连接明确地指定 IP 子网络地址以选择卸载哪一个 TCP 流，并且由于卸载引擎需要执行 TCP 数据包处理，使得它非常的复杂。

[0003] 大分段卸载 (large-segment-offload, LSO) / 传输段卸载 (transmit segment offload, TSO) 可通过减少传输数据包处理来降低所要求的主机处理功率。在这个方法中，主机将比最大传输单元 (the maximum transmission unit, MTU) 更大的传输单元发送到 NIC，且 NIC 依照 MTU 将它们切成段。因为主机处理部分与所传输单元的数量线性相关，这将降低所要求的主机处理功率。当其有效地降低传输数据包处理时，LSO 并不有助于接收数据包处理。另外，对于由主机发送的每个大传输单元，主机将从远端接收多个 ACK，每个 ACK 用于每个 MTU 大小的段。所述多个 ACK 需要消耗有限的且昂贵的带宽，因此将降低吞吐量和效率。

[0004] 在大接收卸载 (large receive offload, LRO) 中，在无状态 (stateless) 接收卸载机制中，可依照哈希函数 (hash function) 将 TCP 流划分成多个硬件队列，这个划分将保证特定的 TCP 流将一直直接流入同一硬件队列。针对每个硬件队列，该机制利用中断结合 (interrupt coalescing) 来扫描队列并将属于同一 TCP 流的队列上的后续数据包聚合 (aggregate) 到单个大接收单元。

[0005] 虽然该机制并不需要来自 NIC 的任何附加硬件和多个硬件队列，但是其可能有不同性能限制。例如，如果流数大于硬件队列数，多个流将掉入同一队列，导致没有用于该队列的 LRO 聚合。如果流数大于硬件队列数的两倍，在任何流上均不执行 LRO 聚合。该聚合将限于在一个中断周期中主机可用的数据包的数量。如果中断周期短，流数并不小，主机 CPU 可用于在每个流上聚合的数据包的数量较小，即使硬件队列的数量较大，也将导致 LRO 聚合受限或不进行聚合。LRO 聚合可在主机 CPU 上执行，这将导致附加处理。驱动器可将 TCP 栈转交到缓冲器的连接表中，所述缓冲器包括报头缓冲器和紧跟其后的一系列数据缓冲器，与所有的数据均连续地传输到一个缓冲器相比，这将需要更多的处理。

[0006] 比较本发明后续将要结合附图介绍的系统的各个特征，当前和传统技术的其它局限性和弊端对于本领域的普通技术人员来说是显而易见的。

发明内容

[0007] 本发明提供了一种用于透明 TCP 卸载系统和 / 或方法，结合至少一幅附图进行了充分的展现和描述，并在权利要求中得到了更完整的阐述。

[0008] 本发明的各种优点、各个方面和创新特征，以及其中所示例的实施例的细节，将在以下的描述和附图中进行详细介绍。

附图说明

[0009] 图 1A 是根据本发明实施例的用于透明 TCP 卸载的典型系统的框图；

[0010] 图 1B 是根据本发明实施例的用于透明 TCP 卸载的又一典型系统的框图；

[0011] 图 1C 是根据本发明实施例的用于透明 TCP 卸载的典型系统的可选实施例的示意图；

[0012] 图 1D 是根据本发明实施例的用于处理透明 TCP 卸载的典型系统的框图；

[0013] 图 1E 是根据本发明实施例的用于帧接收和放置 (placement) 的典型步骤流程图；

[0014] 图 1F 示出了根据本发明实施例的即将被聚合和无序接收的 TCP/IP 帧的典型顺序；

[0015] 图 2A 示出了根据本发明实施例的即将被聚合和有序接收的 TCP/IP 帧的典型顺序；

[0016] 图 2B 示出了根据本发明实施例的典型的聚合的 TCP/IP 帧，所述聚合的 TCP/IP 帧从按照图 2A 中的 TCP 帧的序列信息中生成；

[0017] 图 2C 是根据本发明实施例的用于当数据包 P3 和数据包 P4 未按照传输顺序到达时，用于处理无序数据的典型步骤示意图；

[0018] 图 2D 是根据本发明实施例的典型透明 TCP 卸载的状态图；

[0019] 图 3 是根据本发明实施例的透明 TCP 卸载的典型步骤流程图。

具体实施方式

[0020] 本发明公开了一种用于透明 TCP 卸载的系统和方法。该方法和系统的各个方面可包括：结合器 (coalescer)，用于收集网络接口卡 (NIC) 中的一个或多个 TCP 段，而不将这些 TCP 段中的每一个的状态信息单独传输到主机系统。可将收集的 TCP 段临时缓存在结合器中。结合器可校验与收集的 TCP 段相关的网络流在流查找表中 (FLT) 有入口 (entry)。在 FLT 已满的例子中，结合器可关闭当前入口，并将该网络流分配到可用入口。结合器可更新 FLT 中的信息。当发生终止 TCP 段收集的事件时，结合器可基于收集的 TCP 段生成单个聚合的 TCP 段。该单个聚合的 TCP 段可包括多个 TCP 段，可看作是大接收段。可将该聚合的 TCP 段和状态信息传送到主机系统用于进一步处理。

[0021] 在传统的处理中，所述接收到的多个 TCP 段中的每一个将由主机系统的中主机处理器单独处理。在接收器侧的协议处理和数据放置方面，TCP 处理需要大量的 CPU 处理功率。当前的处理系统和方法包括将 TCP 状态转移到专用的硬件（如 NIC），其中需要对主机 TCP 栈和 / 或底层硬件作重大修改。

[0022] 图 1A 是根据本发明实施例的用于透明 TCP 卸载的典型系统的框图。因此，图 1A 中

的系统可用于处理传输控制协议 (TCP) 数据报或数据包的透明 TCP 卸载。参照图 1A, 该系统可包括:如 CPU 102、存储控制器 104、主机存储器 106、主机接口 108、网络子系统 110 和以太网 112。网络子系统 110 可包括:如透明 TCP- 激活以太网控制器 (TTEEC) 或透明 TCP 卸载引擎 (TTOE) 114。网络子系统 110 可包括:如网络接口卡 (NIC)。主机接口 108 可以是, 如周边元件扩展接口 (peripheral component interconnect, PCI)、PCI-X、PCI-Express、ISA、SCSI 或其它类型的总线。存储控制器 106 可与 CPU 102、存储器 106 和主机接口 108 相连。主机接口 108 可通过 TTEEC/TTOE114 与网络子系统 110 相连。

[0023] 图 1B 是根据本发明实施例的用于透明 TCP 卸载的又一典型系统的框图。参照图 1B, 该系统可包括:如 CPU 102、主机存储器 106、专用存储器 116 和芯片组 118。芯片组 118 可包括:如网络子系统 110 和存储控制器 104。芯片组 118 可与 CPU 102、主机存储器 106、专用存储器 116 和以太网 112 相连。芯片组 118 的网络子系统 110 可与以太网 112 相连。网络子系统 110 可包括:如与以太网 112 相连的 TTEEC/TTOE 114。例如, 网络子系统 110 可通过有线和 / 或无线连接与以太网 112 通信。例如, 该无线连接可以是由 IEEE 802.11 标准支持的无线本地局域网 (WLAN) 连接。网络子系统 110 可包括:如片上 (on-chip) 存储器 113。专用存储器 116 可为上下文 (context) 和 / 或数据提供缓冲区。

[0024] 网络子系统 110 可包括处理器, 如结合器 111。结合器 111 可包括合适的逻辑、电路和 / 或代码, 用于处理 TCP 数据的收集和结合。在这一点上, 结合器 111 可使用流查找表 (FLT) 以保持与当前网络流相关的信息, 为该当前网络流收集 TCP 段, 用于聚合。例如, FLT 可存储到网络子系统 110 中。例如, FLT 可包括至少一个:源 IP 地址、目标 IP 地址、源 TCP 地址、目标 TCP 地址。在本发明的可选实施例中, 至少两个不同的表 (例如, 包括 4- 元组 (tuple) 查找的表) 可用于根据数据包的流来分类输入数据包。例如, 4- 元组查找表可包括以下中的至少一个:例如源 IP 地址、目标 IP 地址、源 TCP 地址、目标 TCP 地址。流上下文表可包括用于聚合的状态变量, 如 TCP 序列数字。

[0025] FLT 可包括至少一个主机缓冲器或存储器地址, 该地址包括:例如, 用于非连续存储器的分散集中列表 (SGL)、累积确认 (cumulative acknowledgment, ACK)、TCP 报头和选项的备份、IP 报头和选项的备份、以太网报头的备份、和 / 或积聚的 TCP 标记。当发生终止事件时, 结合器 111 可用于从聚合的或收集的 TCP 段生成单个聚合的 TCP 段。例如, 单个聚合的 TCP 段可被传递到主机存储器 106。

[0026] 尽管图中示出如 CPU 和以太网, 但本发明并不限于这些例子, 并可分别使用任何类型的处理器、任何类型的数据链路层或物理媒介。因此, 虽然如图所示, 图 1A 中的 TTEEC 或 TTOE 114 与以太网 112 相连, 其可适用于任何类型的数据链路层或物理媒介。此外, 本发明也可构思图 1A-B 中示出的元件的不同程度的结合和分割。例如, TTEEC/TTOE 114 可以是与嵌在底板上的芯片组 118 分开的集成芯片, 或者嵌置在 NIC 中。类似地, 结合器 111 可以是与嵌在底板上的芯片组 118 分开的集成芯片, 或者嵌置在 NIC 中。另外, 专用存储器 116 可集成在图 1B 中的芯片组 118 中或网络子系统 110 中。

[0027] 图 1C 是根据本发明实施例的用于透明 TCP 卸载的典型系统的可选实施例的框图。参照图 1C, 示出了主机处理器 124、主机存储器 / 缓冲器 126、软件算法模块 134 和 NIC 模块 128。NIC 模块 128 可包括 NIC 处理器 130, 如结合器 131 和简化的 NIC 存储器 / 缓冲器模块 132 之类的处理器。例如, NIC 模块 128 可通过有线和 / 或无线连接与外部网络通信。

例如,无线连接可为由 IEE 802.11 标准支持的无线局域网 (WLAN) 连接。

[0028] 结合器 131 可为位于数据包接收路径中的专用处理器或硬件状态机。主机 TCP 栈可包括可用于管理 TCO 协议处理的软件,并可为操作系统的一部分,该操作系统可以是微软视窗 (Microsoft windows) 或免费多用户多任务操作系统 (Linux)。结合器 131 可包括合适的逻辑、电路和 / 或代码,用于累积或结合 TCP 数据。在这一点上,结合器 131 可使用流查找表 (FLT) 以保持与当前网络流相关的信息,为该当前网络流收集 TCP 段,用于聚合。例如,FLT 可存储到简化的 BIC 存储器 / 缓冲器模块 132。当发生终止事件时,结合器 131 可用于从聚合的或收集的 TCP 段生成单个聚合的 TCP 段。例如,单个聚合的 TCP 段可被传送到主机存储器 / 缓冲器 126。

[0029] 根据本发明的某个实施例,为主机提供了用于 TCP 处理的单个聚合 TCP 段,显著地降低了主机 124 的处理开销。此外,因为没有 TCP 状态信息的转移,专用硬件 (如 NIC 128) 可通过结合或积累多个接收到的 TCP 段来协助处理接收到的 TCP 段,以减少每个数据包的处理开销。

[0030] 在传统的 TCP 处理系统中,在 TCP 连接的第一段到来之前,应了解关于该 TCP 连接的某些信息。根据本发明的各种实施例,并不需要在第一 TCP 段到来之前了解 TCP 连接,因为 TCP 状态或上下文信息还是由主机 TCP 栈单独管理,且在任何给定时间,在硬件栈和软件栈之间并没有状态信息的转移。

[0031] 在本发明的一个实施例,可从主机栈方面 (perspective) 提供无状态卸载机制,同时从卸载设备方面提供有状态 (state full) 卸载机制。当将所述卸载机制与 TTOE 进行比较时,达到相当的性能增益 (comparable performance gain)。透明 TCP 卸载 (TTO) 通过允许主机系统处理大于 MTU 的接收和发送数据单元,降低了主机的处理功率。在本发明的一个典型实施例中,对 64KB 的处理数据单元 (PDU) 而不是 1.5KB 的 PDU 进行处理,以显著地降低数据包速率,进而降低用于数据包处理的主机处理功率。

[0032] 在 TTO 中,不能在主机操作系统和包括 TTO 引擎的 NIC 中使用同步交换 (handshake)。在识别新的流并用于卸载的过程中,TTO 引擎可自主运行。在发射侧的卸载可与 LSO 类似,在此主机发送大传输单元,且 TTO 引擎根据最大段大小 (maximum segment size, MSS) 将这些大传输单元切割成较小传输数据包。

[0033] 接收侧的透明 TCP 卸载可通过以下来完成:聚合同一流的多个接收到的数据包,并将它们传送到主机,就像它们在一个数据包 (当接收到的是数据的数据包时,其可为较大的数据包,当接收到的是 ACK 数据包时,其可为聚合的 ACK 数据包) 中接收一样。主机中的处理可与接收到的较大的数据包的处理类似。在 TCP 流聚合的情况下,可定义一些规则以确定是否要聚合数据包。可设置聚合规则以在不增加往返行程时间 (round trip time) 的情况下,允许尽可能多的聚合。这样是否聚合的决定取决于接收到的数据和在没有延时的情况下将该数据传送到主机的重要性。使用用于卸载决定的传输信息,该聚合可与发送 - 接收耦合一起执行,在此发射器和接收器耦联,且该流可当作双向流处理。可在每个流中获取 TTO 中的接收卸载中的上下文信息。在这一点上,对于每个接收到的数据包,下一包头 (packet header) 可用于检测其所属的流,且该数据包更新该流的上下文。

[0034] 当发射器和接收器耦联时,可搜索发送的网络数据包和接收的网络数据包,以确定这些数据包属于哪一个特定的网络流。发送的网络数据包可用于更新所述流的上下文,

这些上下文可用于接收卸载。

[0035] 图 1D 是根据本发明实施例的用于处理透明 TCP 卸载的典型系统的框图。参照图 1D, 示出了输入数据包帧 141、帧解析器 143、关联模块 149、上下文获取模块 151、多个片上高速缓冲存储器 147、多个片下 (off-chip) 存储模块 160、多个片上存储模块 162、RX 处理模块 150、帧缓冲器 154、DMA 引擎 163、TCP 编码模块 157、主机总线 165、多个主机缓冲器 167。RX 处理模块 150 可包括结合器 152。

[0036] 帧解析器 143 可包括合适的逻辑、电路和 / 或代码, 可激活 L2 以太网处理, 包括 : 如输入帧的地址滤波、帧验证和错误检测。不同于普通的以太网控制器, 处理的下一步可包括 : 如帧解析器 143 中的 L3 (如 IP 处理) 和 L4 (如 TCP 处理)。例如, 通过在结合的 TCP/IP 流上处理通信量, TTEEC 114 可降低主机 CPU 102 的使用和存储器带宽。例如, TTEEC 114 可基于数据包解析信息和元组 145 确定输入数据包所属的协议。如果该协议是 TCP, 接着 TTEEC 114 将检测该数据包是否对应卸载 TCP 流, 例如, 对于该 TCP 流, 至少一部分 TCP 状态信息可由 TTEEC 114 保存。如果该数据包对应卸载连接, 接着 TTEEC 114 可指示该帧的数据有效载荷部分的数据动作。有效载荷数据的目的地由该帧中的流状态信息和方向信息共同确定。例如, 该目的地可以是主机存储器 106。最后, TTEEC 114 可更新其内部 TCP 和流状态的更高级别, 而不需与主机 TCP 栈的连接状态协调, 并可从其内部流状态获得主机缓冲器地址和长度。

[0037] 接收系统构架可包括 : 如控制路径处理 140 和数据动作引擎 142。图 1D 的上部示出的控制路径上的系统元件可设计为处理各种处理过程, 这些处理过程用于完成如具备最大灵活性、效率和目标网速的 L3/L4 或更高级的处理。例如, 这些处理过程的结果可包括一个或多个数据包识别卡, 这些数据包识别卡可用于提供携带与帧有效载荷数据相关的信息的控制结构。当在不同的模块中处理这些数据包时, 可以在 TTEEC 114 中生成这些结果。数据路径 142 可在控制处理完成的基础上转移帧的有效载荷数据部分或原始的数据包 155, 例如从片上数据包帧缓冲器 154 转移到直接内存存取 (DMA) 引擎 163, 接着通过由处理选定的主机总线 165 转移到主机缓冲器 167。数据路径 142 到 DMA 引擎可包括数据包数据区和任意包头 161。

[0038] 例如, 接收系统可完成下列一个或多个 : 解析 TCP/IP 报头 145、在关联模块 149 中关联帧和 TCP/IP 数据流、在上下文获取模块 151 中获取 TCP 流上下文、在 RX 处理模块 150 中处理 TCP/IP 报头、确定报头 / 数据边界并更新状态 153、将数据映射到主机缓冲器、并通过 DMA 引擎 163 将数据转移到主机缓冲器 167。可通过 DMA 引擎 163 在芯片上消耗掉报头或者将其转移到主机缓冲器 167。

[0039] 数据包帧缓冲器 154 可为接收系统构架中的任意模块。例如, 出于同一目的, 其可用于在传统的 L2 NIC 中使用先进先出 (FIFO) 数据结构, 或用于存储针对附加处理的高层通信量。接收系统中的数据包帧缓冲器 154 并不限于单个示例。当完成控制路径 140 处理后, 数据路径 142 可在数据处理级中一次或多次存储数据。

[0040] 在本发明的一个典型实施例中, 所述的结合操作的至少一部分针对图 1B 中的结合器 111 和 / 或针对图 1C 中的结合器 131, 并可在图 1D 中的 RX 处理模块 150 中的结合器 152 中执行。在这个例子中, TCP 数据的缓冲或存储可由如帧缓冲器 154 执行。此外, 结合器 152 使用的 FLT 可使用如片下存储 160 和 / 或片上存储 162 来实现。

[0041] 在本发明的一个实施例中，在流的生命周期中(lifetime)，可在某个点检测到新的流。当新的流被检测到时，其状态是未知的，且第一数据包可用于更新流状态直到已知该流是整齐的。执行 TTO 的设备也可支持其它的卸载类型，如 TOE、RDMA 或 iSCSI 卸载。在这种情况下，可将用于 TTO 的 FLT 与用于其它卸载类型的连接搜索共享，在指示流的卸载类型的 FLT 中，这些卸载类型具有各自的入口。属于其它卸载类型的流的数据包可不属于自己 TTO 的可选类型。一旦检测到新的流，该流可始于基本初始化上下文。可在 FLT 中创建具有流 ID 的入口。

[0042] 在本发明的另一实施例中，同一流的多个段可在 TTO 中聚合以达到接收聚合长度(receive aggregation length)，向主机提供用于处理的更大段，如果聚合是被允许的，可将接收到的数据包放置在主机存储器 126 中，但并不将其传送到主机。可替换地，主机处理器 124 可更新数据包所属流的上下文。如果没有未传送的在先聚合的数据包，新的输入数据包可直接将该数据包单独传送或可作为表示该数据包和在先接收到的数据包的单个数据包。在本发明的另一实施例中，数据包未被传送但可更新流的上下文。

[0043] 如果至少一个下列事件在 TCP 水平发生，可发生终止事件且该数据包可不聚合：(1) 当其从接收到的序号(SN)和流的上下文中生成时，该数据包不是有序的；(2) 检测到至少一个数据包具有 TCP 标记而不是 ACK 标记(如 PUSH 标记)；(3) 检测到具有选择性确认(SACK)信息的至少一个数据包；或(4) 所接收的该 ACK SN 大于传送的 ACK SN，并请求停止聚合。类似地，如果至少一个下列事件在 IP 水平发生，可发生终止事件且该数据包可不聚合：(1) IP 报头中的服务(TOS)域不同于聚合的在先数据包的 TOS 域；或接收到的数据包是 IP 分段(fragment)。

[0044] 当将多个数据包聚合到单个数据包时，聚合的数据包头可包括其包括的所有单个数据包的聚合报头。在本发明的一个典型实施例中，用于聚合的多个 TCP 规则可如下。例如，(1) 在聚合的报头中的 SN 是第一或最旧的数据包的 SN；(2) ACK SN 是最后或最新的段的 SN；(3) 聚合的报头的长度是所有聚合的数据包的长度的总和；(4) 聚合的报头中的窗口是在最后或最新聚合的数据包中接收到的窗口；(5) 聚合的报头中的时戳(TS)是在第一或最旧聚合的数据包中接收到的 TS；(6) 聚合的报头中的时戳回波显示(TS-echo)是在第一或最旧聚合的数据包中接收到的 TS-echo；且(7) 在聚合的报头中接收到的检验和是所有聚合的数据包的聚合检验和。

[0045] 在本发明的典型实施例中，可提供多个 IP 域聚合规则。如：(1) 聚合的报头中的 SN 是第一或最旧的数据包的 SN；(2) 该 ACK SN 是最后或最新的段的 SN；(3) 聚合的报头的长度是所有聚合的数据包的长度的总和；(4) 聚合的报头中的窗口是在第一或最旧的数据包中接收到的窗口；(5) 聚合的报头中的时戳(TS)是在第一或最旧的数据包中接收到的 TS；(6) 聚合的报头中的时戳禁止(TS-echo)是在第一或最旧的数据包中接收到的 TS-echo；且(7) 在聚合的报头中接收到的检验和是所有聚合的数据包的检验和的收集。

[0046] 在本发明的一个典型实施例中，可提供多个 IP 域聚合规则。例如，(1) 聚合的报头的 TOS 是所有的聚合的数据包的 TOS；(2) 聚合的报头的生存时间(TTL)是所有输入 TTL 中的最小值；(3) 聚合的报头的长度是聚合的数据包的长度和；(4) 对于聚合的数据包，聚合的报头的分段偏移量(fragment offset)可为 0；且(5) 聚合的报头的数据包 ID 是最后接收到的 ID。

[0047] 可将接收到的数据包聚合，直到由于终止事件发生，使得接收到的数据包不能被聚合，或者该流超时 (timeout) 届满，或聚合的数据包超过 RAL。可这样执行所述超时，当流上的第一数据包用于放置而非发送时，将所述超时设置为一个值，超时聚合值。其后的聚合的数据包可不改变该超时。当由于超时期满将要发送该数据包时，可取消超时并在不会被发送的下一第一数据包中重新设置超时。然而，本发明的其它实施例可通过周期性扫描所有流以提供超时。

[0048] 在本发明的典型实施例中，接收到的 ACK SN 可用来确定聚合纯 ACK 的规则或确定由于接收到的 ACK SN 而停止聚合具有数据的数据包的规则。可从不将复制的纯 ACK 聚合。当复制的纯 ACK 被接收时，它们可使在先聚合的数据包被发送，且该纯 ACK 将迅速被分开发送。接收到的 ACK SN 也可用于停止聚合并将半连接 (pending) 聚合数据发送到主机 TCP/IP 栈。

[0049] 在本发明的典型实施例中，可根据 ACK SN 提供用于停止聚合的多个规则。例如，(1) 如果还没有发送经确认的 (ACKed) 字节数，则考虑接收到的段和未发送的在先段超过门限值，如在字节中的接收确认字节聚合 (ReceiveAckedBytesAggregation)；或 (2) 自第一数据包的到达的时间超过门限值，如超时确认聚合 (TimeoutAckAggregation)，该数据包可使接收到的 ACK SN 往前。为了这个目的，每个流可能需要第二计时器或其它的机制，如对这些流执行周期性扫描。

[0050] 在本发明的又一典型实施例中，如果至少一个下列事件发生，可从主机存储器中移除这些流：(1) 在接收侧检测到重设 (RST) 标记；(2) 在接收侧检测到结束 (FIN) 标记；(3) 在预定时间 (如终止无活动时间, TerminateNoActivityTime) 的流上无接收活动；(4) 接收方向的保持活动 (KeepAlive) 数据包未经确认。可使用最近最少使用 (least recently used, LRU) 的高速缓冲存储器来替代超时规则，以从主机存储器中移除这些流。

[0051] 图 1E 是根据本发明实施例的用于帧接收和放置 (placement) 的典型步骤流程图。参照图 1D 和图 1E，在步骤 180 中，网络子系统 110 可从例如以太网 112 接收帧。在步骤 182 中，帧解析器可对该帧进行解析，以找出如 L3 报头和 L4 报头。该帧解析器可对 L2 报头进行处理，以为 L3 报头（如 IP 版本 4(IPv4) 报头或 IP 版本 6(IPv6) 报头）处理作准备。IP 报头版本域可确定该帧是否携带 IPv4 数据报或 IPv6 数据报。

[0052] 例如，如果 IP 报头版本域携带值为 4，那么该帧可携带 IPv4 数据报。例如，如果 IP 报头版本域携带 6 个值，那么该帧可携带 IPv6 数据报。可从选取 IP 报头域，获得如 IP 源 (IP SRC) 地址、IP 目的地 (IP DST) 地址和 IPv4 报头“协议”域或 IPv6 “下一报头”。如果 IPv4 “协议”报头域或 IPv6 “下一报头”报头域携带值为 6，那么下一报头可为 TCP 报头。

[0053] 接下来发生剩余的 IP 处理，其方式可类似于在传统的现成 (off-the-shelf) 软件栈中的处理。该执行可从使用嵌置的处理器上的固件，到使用专用的有限状态机 (FAM) 或处理器和状态机的混合体。例如，该执行可随着由一个或多个处理器、状态机或它们的混合体进行的处理的多个阶段而改变。IP 处理可包括，但不限于：提取关于如长度、有效性和分段的信息。也可对定位 TCP 报头进行解析和处理。例如，对 TCP 报头的解析可提取与接收帧相关的特定网络流的源端口和目的端口的信息。

[0054] TCP 处理可分为多个附加处理阶段。在步骤 184 中，该帧可与端到端 TCP/IP 连接相关。在 L2 处理后，在一个实施例中，本发明可提供经校验的 TCP 检验和。例如，该流可由

下列 4- 元组中的一部分定义 :IP 源地址 (IP_SRCaddr) ;IP 目的地地址 (IP_DST_addr) ;TCP 源端口号 (TCP_SRC) ; 和 TCP 目的地端口号 (TCP_DST)。根据相关的 IP 地址的选择, 该处理可用于 IPv4 或 IPv6。

[0055] 如步骤 812 中帧解析的结果, 可完全地提取该 4- 元组。关联硬件 (association hardware) 可将接收到的 4- 元组与存储在 TTEEC/TTOE 中的 4- 元组清单进行比较。TTEEC/TTOE 114 可保存元组清单, 该元组可表示如由 TTEEC/TTOE 114 管理的聚合流或卸载连接。对于片上和片下选项, 用于存储关联信息的存储器资源是价格昂贵的。因此, 可能并非全部的关联信息都位于芯片上。可采用高速缓冲存储器将最有效的连接存储在芯片上。如果找到匹配的 4- 元组, TTEEC/TTOE 114 可采用匹配的 4- 元组管理特定的 TCP/IP 流。

[0056] 在步骤 186 中, 可获取 TCP 流上下文。在步骤 188 中, 可对 TCP/IP 报头进行处理。在步骤 192 中, 可确定报头 / 数据边界。在步骤 192 中, 结合器可收集或积聚与特定网络流相关的多个帧。例如, 与 TCP/IP 连接相关的收集的 TCP 段和收集的信息可用于生成 TCP/IP 帧, 该帧包括单个聚合的 TCP 段。在步骤 194 中, 当发生终止事件时, 该处理可进行到步骤 196。该终止事件可以是突发事件 (incident)、事例 (instance) 和 / 或信号, 它们可向结合器指示, 完成 TCP 段的收集和积聚以及单个集合的 TCP 段可传送到主机系统以用于处理。在步骤 198 中, 可将对应于单个集合的 TCP 段的有效载荷数据映射到主机缓冲器。在步骤 198 中, 可将来自单个集合的 TCP 段的数据转移到主机缓冲器中。回到步骤 194, 当并没有终止事件发生时, 该处理可进行到步骤 180 且可对下一接收帧进行处理。

[0057] 图 1F 示出了根据本发明实施例的即将被聚合和非顺序接收的 TCP/IP 帧的典型顺序。参照图 1F, 示出了第一 TCP/IP 帧 202、第二 TCP/IP 帧 204、第三 TCP/IP 帧 206 和第四 TCP/IP 帧 208。示出的每个 TCP/IP 帧可包括以太网报头 200a、IP 报头 200b、TCP 报头 200c 和 TCP 选项 200d。虽然在图 1F 中未示出, 但是每个 TCP/IP 帧还可包括有效载荷部分, 该有效载荷部分包括 TCP 段, 该 TCP 段中包括被传输的数据。以太网报头 200a 具有 TCP/IP 帧的值 enet_hdr。IP 报头 200b 可包括多个域 (field)。在这一点上, IP 报头 200b 可包括域 IP_LEN, 其可用于指示帧中的字节数。在这个例子中, 第一 TCP/IP 帧 202、第二 TCP/IP 帧 204、第三 TCP/IP 帧 206 和第四 TCP/IP 帧 208 中的每一个都具有 1448 个 TCP 有效载荷数据包。

[0058] IP 报头 200b 也可包括识别域 (identification field) ID, 其可用于识别帧。在这个例子中, 对于第一 TCP/IP 帧 202, ID = 100, 对于第二 TCP/IP 帧 204, ID = 101, 对于第三 TCP/IP 帧 206, ID = 103, 对于第四 TCP/IP 帧 208, ID = 102。IP 报头 200b 可包括附加域, 如 IP 报头检验和域 ip_csm、源域 ip_src、和目的域 ip_dest。在这个例子中, 对于所有的帧, ip_src 和 ip_dest 的值可以相同; 而对于第一 TCP/IP 帧 202, IP 报头检验和域的值为 ip_csm0, 对于第二 TCP/IP 帧 204, IP 报头检验和域的值为 ip_csm1, 对于第三 TCP/IP 帧 206, IP 报头检验和域的值为 ip_csm3, 对于第四 TCP/IP 帧 208, IP 报头检验和域的值为 ip_csm2。

[0059] TCP 报头 200c 可包括多个域。例如, TCP 报头 200c 可包括源端口域 srcprt、目的地端口域 dest_prt、TCP 序列域 SEQ、确认域 ACK、标记域 FLAG、广播窗口域 WIN、和 TCP 报头检验和域 tcp_csm。在这个例子中, 对于所有的帧, src_prt、dest_prt、FLAG 和 WIN 的值相同。对于第一 TCP/IP 帧 202, SEQ = 100, ACK = 5000 且 TCP 报头检验和域为 tcp_csm0。

对于第二 TCP/IP 帧 204, SEQ = 1548, ACK = 5100 且 TCP 报头检验域为 tcp_csm1。对于第三 TCP/IP 帧 206, SEQ = 4444, ACK = 5100 且 TCP 报头检验域为 tcp_csm3。对于第四 TCP/IP 帧 208, SEQ = 2996, ACK = 5100 且 TCP 报头检验域为 tcp_csm2。

[0060] TCP 选项 200d 可包括多个域。例如, TCP 选项 200d 可包括与 TCP 帧关联的时戳指示器,也叫做时戳。在这个例子中,对于第一 TCP/IP 帧 202,其时戳指示器的值为 timeatamp0,对于第二 TCP/IP 帧 204,其时戳指示器的值为 timeatamp1,对于第三 TCP/IP 帧 206,其时戳指示器的值为 timeatamp3,且对于第四 TCP/IP 帧 208,其时戳指示器的值为 timeatamp2。

[0061] 图 1F 中示出的 TCP/IP 帧的典型序列被无序接收,该无序接收是相对于网络子系统 110 的传输顺序。IP 报头 200b 中的 ID 域包括的信息和 / 或 TCP 选项 200d 中的时戳中包括的信息可指示第三 TCP/IP 帧 206 和第四 TCP/IP 帧 208 可按照与传输顺序不同的顺序被接收。在这个例子中,可在第二 TCP/IP 帧 204 之后且在第三 TCP/IP 帧 206 之间发送 TCP/IP 帧 208。结合器,如图 1B-1E 中所示的结合器可获得来自 TCP/IP 帧的信息,并通过结合接收到的信息生成单个 TCP/IP 帧。在这一点上,结合器可使用 FLT 存储和 / 或更新从 TCP/IP 帧接收到的至少一部分信息。该结合器可使用可用存储器以存储或缓存聚合的 /IP 帧的有效载荷。

[0062] 图 2A 示出了根据本发明实施例的即将被聚合和顺序接收的 TCP/IP 的典型序列。参照图 2A,示出了第一 TCP/IP 帧 202、第二 TCP/IP 帧 204、第三 TCP/IP 帧 206 和第四 TCP/IP 帧 208。示出的每个 TCP/IP 帧可包括以太网报头 200a、IP 报头 200b、TCP 报头 200c 和 TCP 选项 200d。虽然在图 2A 中未示出,但是每个 TCP/IP 帧还可包括有效载荷部分,该有效载荷部分包括 TCP 段,该 TCP 段中包括将要被传输的数据。以太网报头 200a 具有 TCP/IP 帧的值 enet_hdr。IP 报头 200b 可包括多个域 (field)。在这一点上,IP 报头 200b 可包括域 IP_LEN,其可用于指示帧中的字节数。在这个例子中,第一 TCP/IP 帧 202、第二 TCP/IP 帧 204、第三 TCP/IP 帧 206 和第四 TCP/IP 帧 208 中的每一个都具有 1448 个 TCP 有效载荷数据包。

[0063] IP 报头 200b 也可包括识别域 ID,其可用于识别帧。在这个例子中,对于第一 TCP/IP 帧 202, ID = 100,对于第二 TCP/IP 帧 204, ID = 101,对于第三 TCP/IP 帧 206, ID = 102,对于第四 TCP/IP 帧 208, ID = 103。IP 报头 200b 可包括附加域,如 IP 报头检验域 ip_csm、源域 ip_src、和目的地域 ip_dest。在这个例子中,对于所有的帧, ip_src 和 ip_dest 的值可以相同;而对于第一 TCP/IP 帧 202, IP 报头检验域的值为 ip_csm0,对于第二 TCP/IP 帧 204, IP 报头检验域的值为 ip_csm1,对于第三 TCP/IP 帧 206, IP 报头检验域的值为 ip_csm2,对于第四 TCP/IP 帧 208, IP 报头检验域的值为 ip_csm3。

[0064] TCP 报头 200c 可包括多个域。例如, TCP 报头 200c 可包括源端口域 srcprt、目的地端口域 dest_prt、TCP 序列域 SEQ、确认域 ACK、标记域 FLAG、广播窗口域 WIN、和 TCP 报头检验域 tcp_csm。在这个例子中,对于所有的帧, src_prt、dest_prt、FLAG 和 WIN 的值相同。对于第一 TCP/IP 帧 202, SEQ = 100, ACK = 5000 且 TCP 报头检验域为 tcp_csm0。对于第二 TCP/IP 帧 204, SEQ = 1548, ACK = 5100 且 TCP 报头检验域为 tcp_csm1。对于第三 TCP/IP 帧 206, SEQ = 4444, ACK = 5100 且 TCP 报头检验域为 tcp_csm2。对于第四 TCP/IP 帧 208, SEQ = 2996, ACK = 5100 且 TCP 报头检验域为 tcp_csm3。

[0065] TCP 选项 200d 可包括多个域。例如, TCP 选项 200d 可包括与 TCP 帧关联的时戳指示

器,也叫做时戳。在这个例子中,对于第一 TCP/IP 帧 202,其时戳指示器的值为 timeatamp0,对于第二 TCP/IP 帧 204,其时戳指示器的值为 timeatamp1,对于第三 TCP/IP 帧 206,其时戳指示器的值为 timeatamp2,且对于第四 TCP/IP 帧 208,其时戳指示器的值为 timeatamp3。

[0066] 图 2A 中示出的 TCP/IP 帧的典型序列被无序接收,该无序接收是相对于网络子系统 110 的传输顺序。结合器,如图 1B-1E 中所示的结合器可获得来自 TCP/IP 帧的信息,并通过结合接收到的信息生成单个 TCP/IP 帧。在这一点上,结合器可使用 FLT 存储和 / 或更新从 TCP/IP 帧接收到的至少一部分信息。该结合器可使用可用存储器以存储或缓存聚合的 /IP 帧的有效载荷。

[0067] 图 2B 示出了根据本发明实施例的典型的聚合的 TCP/IP 帧,所述聚合的 TCP/IP 帧从图 2A 的 TCP 帧序列中的信息中生成。参照图 2B,示出了单个 TCP/IP 帧 210,其是由结合器从图 2A 中接收到的 TCP/IP 帧序列中生成的。该 TCP/IP 帧 210 可包括以太网报头 200a、IP 报头 200b、TCP 报头 200c 和 TCP 选项 200d。TCP/IP 帧 210 也可包括有效载荷,该有效载荷包括 TCP 段,该 TCP 段包括所接收的 TCP/IP 帧的实际数据 (actual data)。TCP/IP 帧 210 中的以太网报头 200a、IP 报头 200b、TCP 报头 200c 和 TCP 选项 200d 中的域可与图 2A 中的 TCP/IP 帧中的域基本相似。对于 TCP/IP 帧 210,有效载荷中的数据包的总数是 IP_LEN = 5844,其对应于图 2A 中的所有的四个 TCP/IP 帧的数据包数的总和 (1448*4+52)。对于 TCP/IP 帧 210, ID 值 = 100,这对应于第一 TCP/IP 帧 202 的 ID 值。此外,时戳指示器的值为 timeatamp0,这对应于第一 TCP/IP 帧 202 的时戳指示器。例如,可将 TCP/IP 帧 210 传输或转移到主机系统用于 TCP 处理。

[0068] 图 2C 是根据本发明实施例的用于当数据包 P3 和数据包 P4 未按照传输顺序到达时,用于处理无序数据的典型步骤示意图。参照图 2C,如实际接收 RX 通信量图 200 所示,在数据包 P2 到达之前,数据包 P3 和 P4 可相对于按彼此顺序到达 NIC 128。数据包 P3 和 P4 可分别对应于 TCP 传输序列中的岛 (isle) 211 中的第四和第五数据包。在这种情况下,在实际接收 RX 通信量图 200 中,在数据包 P1 的结束和数据包 P3 的开始之间存在间隙或时间间隔。如 TCP 接收序列空间 202 中所示,TCP 传输序列中的第一脱节部分 (disjoint portion) 可能是由于数据包 P3 和 P4 在岛 213(包括数据包 P0 和 P1) 之后到达。岛 211 的最右边部分 rcv_nxt_R 可重新表示为 (rcv_nxt_L+(岛的长度)),在此, rcv_nxt_L 是岛 211 的最左边部分,且岛的长度为数据包 P3 和 P4 的长度和。

[0069] 图 2D 是根据本发明实施例的典型透明 TCP 卸载的状态图。参照图 2D,示出了多个典型流状态,也就是,顺序状态 226、无序状态 (000) 224 或未知状态 222。在过渡状态 (transition state) 228,可对流进行未知状态 222 检测,对于该流未检测到 3- 向同步交换 (3-way handshake) 或是在该流的生命周期中的除初始化阶段以外的其它时间点检测到 3- 向同步交换。

[0070] TCP 3- 向同步交换始于同步 (SYN) 段,该同步段包括由第一主机选择和发送的初始发送序列号 (initial send sequence number)。该序列号可为数据包中起始序列号 (starting sequence number),也可是针对该段中的被发送的数据的每个字节的增量。当第二主机接收具有序列号的 SYN 时,其可在序列号域和确认域发送具有其自身完全独立的 ISN 号的 SYN。确认 (ACK) 域可告知接收器其数据已经在另一端被接收,且预计数据字节的下一段将要发送,且其可被称做 SYN-ACK。当第一主机接收该 SYN-ACK 段时,其可发送包含

下一序列号的 ACK 段,该 ACK 段被称为前向确认 (forward acknowledgement),并由第二主机接收。该 ACK 段可由正在设置的 ACK 域识别。可将在某个时间范围内未被确认的段再次发送。

[0071] 当流是被透明 TCP 卸载的时候,其不会从顺序状态 226 和 000 状态 224 转移到未知状态 222,除非其被清除和检测。在过渡状态 230,状态图可追踪无序岛序列号边界,如使用图 2C 中示出的参数 rcv_nxt_R 和 rcv_nxt_L 。第一入口处段 (ingress segment) 可称做岛,如岛 213(图 2C) 且排序状态可设置为 000 状态 224。岛的最右边部分 rcv_nxt_R 可表示为 ($rcv_nxt_L + (\text{岛的长度})$),在此, rcv_nxt_L 是岛的最左边部分,岛的长度是岛中数据包的长度和。

[0072] 在过渡状态 232,当发射器和接收器非耦合时,NIC 128 可能不具有本地栈确认信息。只要岛长度大于门限值,排序状态就可从 000 状态 224 修正到有序状态 226。由于在未接收到 ACK 时,远端不会发送多于这个值的数据,可根据本地主机栈最大接收窗口设置门限值。

[0073] 在过渡状态 234,如果检测到新的流具有 TCP 3- 向同步交换,可将初始排列状态设置成有序状态 226。在过渡状态 236,可根据下列算法,将 rcv_nxt_R 用于检查入口处数据包的状态。

```
[0074] if(in_packet_sn == rcv_nxt_R)// 当岛是增加的更新的 rcv_nxt_L
[0075]     rcv_nxt_R = in_packet_sn+in_packet_len
[0076] 在过渡状态 238,如果岛的长度不等于 rcv_nxt_R,排序状态可从有序状态 226 修正到 000 状态 224。可根据下列算法,将 rcv_nxt_R 用于检查入口处数据包的状态。
[0077] if(in_packet_sn != rcv_nxt_R)
[0078]     rcv_nxt_L = in_packet_sn
[0079]     rcv_nxt_R = in_packet_sn+in_packet_len
[0080]     change state to 000 224。
```

[0081] 在过渡状态 240,可在 000 状态 224 过程中使用下列典型算法追踪最高 000 岛的边界的每个入口处数据包。

```
[0082] if(in_packet_sn == rcv_nxt_R)// 当岛是增加的更新的 rcv_nxt_L
[0083]     rcv_nxt_R = in_packet_sn+in_packet_len
[0084] else_if(in_packet_sn > rcv_nxt_R)// 当新的更高岛生成时
[0085]     rcv_nxt_R = in_packet_sn+in_packet_len
[0086]     rcv_nxt_L = in_packet_sn
```

[0087] 图 3 是根据本发明实施例的透明 TCP 卸载的典型步骤流程图。参照图 3,在步骤 302 中,针对每个接收到的数据包,结合器 131 可通过检查协议报头,将其分类成非 -TCP 和 TCP 数据包。在步骤 304 中,对于非 TCP 数据包或不具有正确检验和的数据包,结合器 131 继续对其进行处理而不发生改变。在步骤 306 中,结合器 131 可计算有效载荷的 TCP 检验和。在步骤 308 中,对于具有有效检验和的 TCP 数据包,结合器 131 先使用元组检索流查找表 (FLT),以确定是否该数据包属于结合器 131 已知的连接,上述元组包括 IP 源地址、IP 目的地地址、源 TCP 端口和目标 TCP 端口。

[0088] 在步骤 310 中,在该检索失败的例子中,该数据包属于结合器 131 未知的连接。结

合器 131 可确定是否有 TCP 有效载荷。如果没有 TCP 有效载荷（如纯 TCP ACK），结合器 131 可停止进一步的处理，并允许通过常用处理路径处理该数据包，并在 FLT 中增加入口。在步骤 312 中，如果没有 TCP 有效载荷且该连接并未在 FLT 中，结合器 131 可在 FLT 中为该连接创建新的入口。该操作可包括当 FLT 已满时在 FLT 中引退入口。FLT 引退可迅速停止任何进一步的结合，并向主机 TCP 栈提供任何聚合 TCP 段的指示。

[0089] 在步骤 314 栈，如果新创建 / 替换的 FLT 入口、以及元组、TCP 序列号、TCP 确认号、TCP 有效载荷的长度以及时戳出现，它们将被记录。在步骤 316，TCP 有效载荷前的任何报头部可放置到缓冲器（报头缓冲器）中，然而该 TCP 有效载荷将被放置到另一缓冲器（有效载荷缓冲器）中。该信息也可保存在 FLT 中并起动计时器。在步骤 318 中，可在结合器 131 中临时收集报头和有效载荷直到任一下列典型终止事件发生：

- [0090] a. TCP 标记包括 PSH(强迫推送)、FIN 或 RST 字节。
- [0091] b. TCP 有效载荷量超过门限值或最大 IP 数据报大小。
- [0092] c. 计时器期满。
- [0093] d. FLT 表已满且将一个当前网络流入口替换成与新的网络流相关的入口。
- [0094] e. 检测到包括同一元组的第一 IP 分段。
- [0095] f. 发送窗口大小改变。
- [0096] g. TCP 确认 (ACK) 号的改变超过 ACK 门限值。
- [0097] h. 复制 ACK 的数量超过复制 ACK 门限值。
- [0098] i. 可选择的 TCP 确认 (SACK)。

[0099] 在这一点上，可将 PSH 比特称作控制比特，其可指示包括数据的段必须被推进到接收用户。FIN 比特可称作控制比特，其可指示发送器将不再发送更多数据或控制占用序列空间。RST 比特可当作控制比特，其可在接收器应删除连接而不进行进一步的交互时，指示重设操作。ACK 比特可称作控制比特，其可指示段的确认域指定该段的发送器预计接收的下一序列号，从而确认收到所有在先序列号。

[0100] 在步骤 320 中，当这些事件中的任一发生时，结合器 131 可修正具有 TCP 有效载荷的新总量的 TCP 报头，并向普通 TCP 栈指示较大和单个的 TCP 段，以及集合的 TCP 段的总量和 / 或第一时戳选项。在步骤 322 中，当较大和单个 TCP 段到达主机 TCP 栈，主机 TCP 栈将其当作任何普通输入帧进行处理。

[0101] 将要定位在 NIC 上的硬件栈适合从线路上取走数据包，并独立于在主机处理器上运行的 TCO 栈将它们积聚或聚合。例如，多个接收到的数据包的数据部分在主机存储器上积聚，直到创造出单个较大的 TCP 接收数据包（如 8K）。一旦这个单个较大的 TCO 接收数据包生成，其将被转移到主机以进行处理。在这一点上，当该硬件栈承认该接收到的 TCP 数据包时，其适合建立状态和上下文信息。这显著地降低了与 TCP 栈处理相关的计算密集型任务。虽然在主机存储器中积聚多个接收到的数据包的数据部分，但是该数据依然受 NIC 的控制。

[0102] 虽然示出了单个 TCP 连接的处理，本发明并不限于此。因此本发明的各种实施例可在多个物理网络端口上的多个 TCP 连接提供支持。

[0103] 本发明的又一实施例可提供一种机器可读存储。其内存储的计算机程序包括至少一个代码段，所示至少一个代码段由机器执行而使得所述机器执行上述步骤，以用于透明

TCP 卸载。

[0104] 因此，本发明可以通过硬件、软件，或者软、硬件结合来实现。本发明可以在至少一个计算机系统中以集中方式实现，或者由分布在几个互连的计算机系统中的不同部分以分散方式实现。任何可以实现方法的计算机系统或其它设备都是可适用的。常用软硬件的结合可以是安装有计算机程序的通用计算机系统，通过安装和执行程序控制计算机系统，使其按方法运行。

[0105] 本发明还可以通过计算机程序产品进行实施，程序包含能够实现本发明方法的全部特征，当其安装到计算机系统中时，可以实现本发明的方法。本文件中的计算机程序所指的是：可以采用任何程序语言、代码或符号编写的一组指令的任何表达式，该指令组使系统具有信息处理能力，以直接实现特定功能，或在进行下述一个或两个步骤之后实现特定功能：a) 转换成其它语言、编码或符号；b) 以不同的格式再现。

[0106] 虽然本发明是通过具体实施例进行说明的，本领域技术人员应当明白，在不脱离本发明范围的情况下，还可以对本发明进行各种变换及等同替代。另外，针对特定情形或材料，可以对本发明做各种修改，而不脱离本发明的范围。因此，本发明不局限于所公开的具体实施例，而应当包括落入本发明权利要求范围内的全部实施方式。

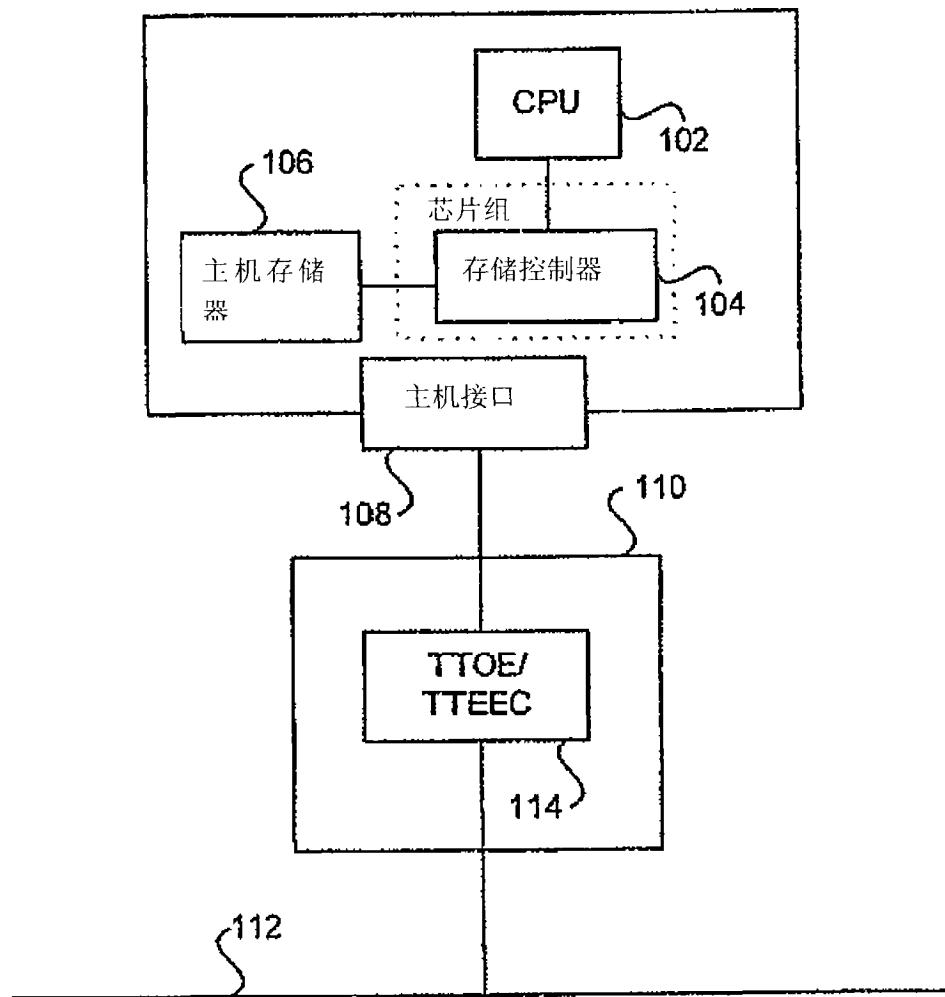


图 1A

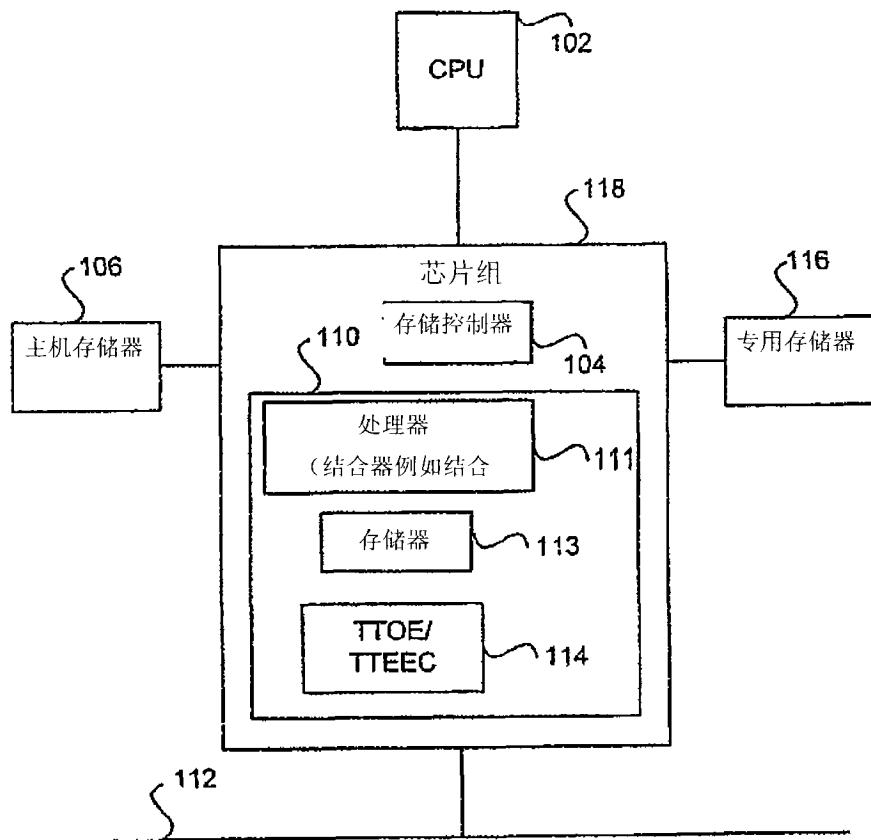


图 1B

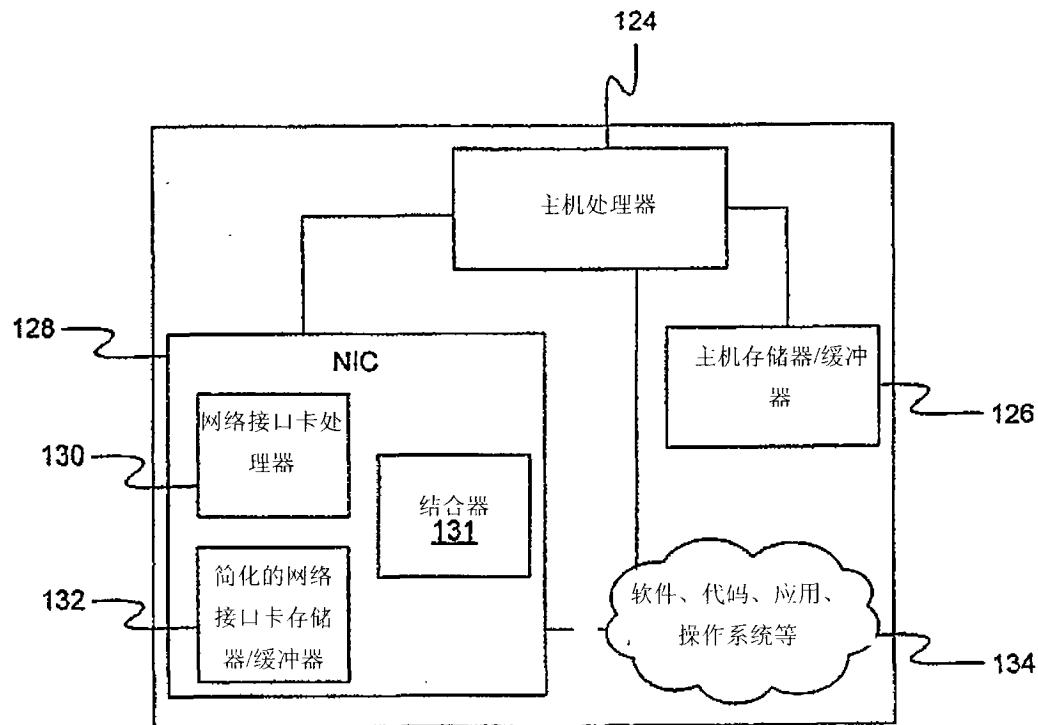


图 1C

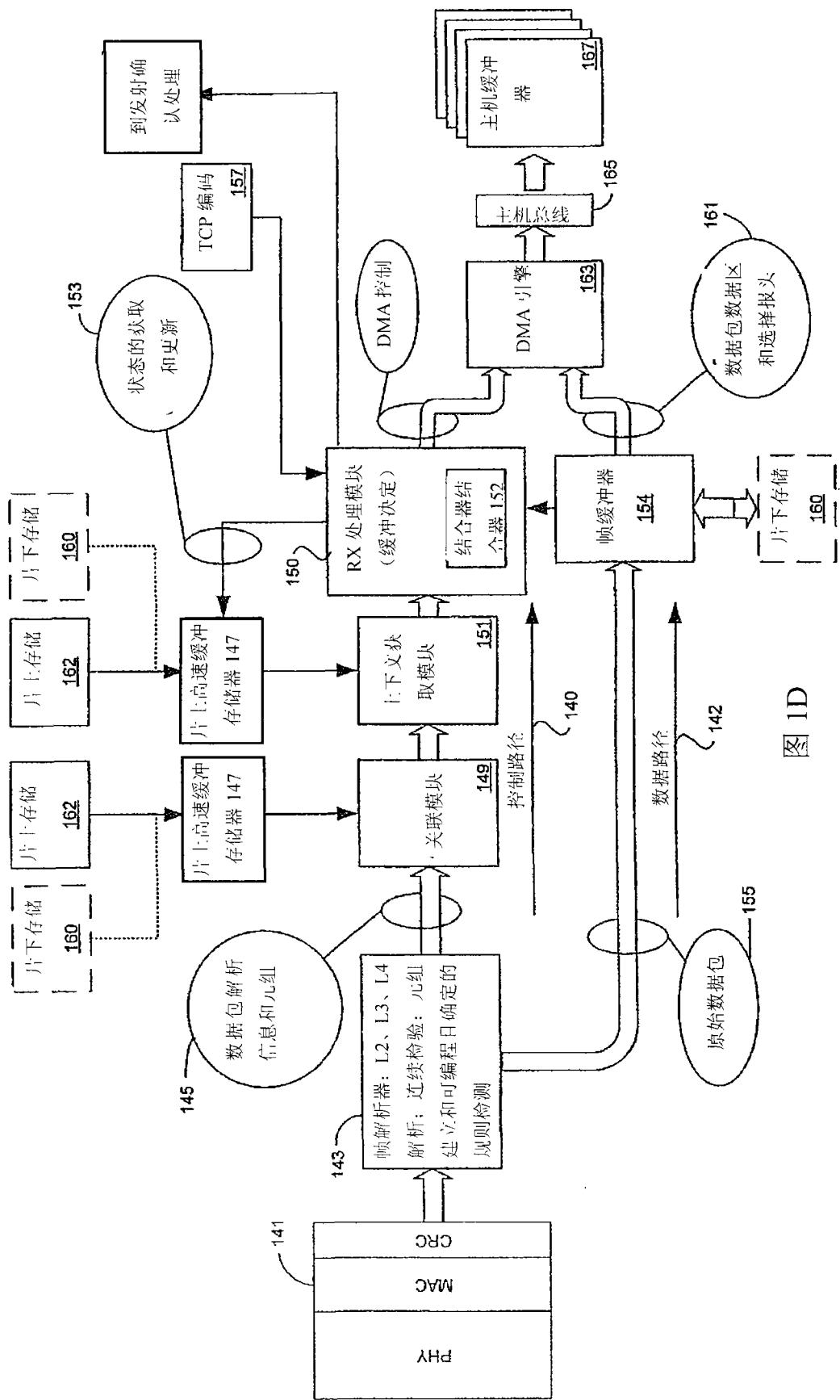


图 1D

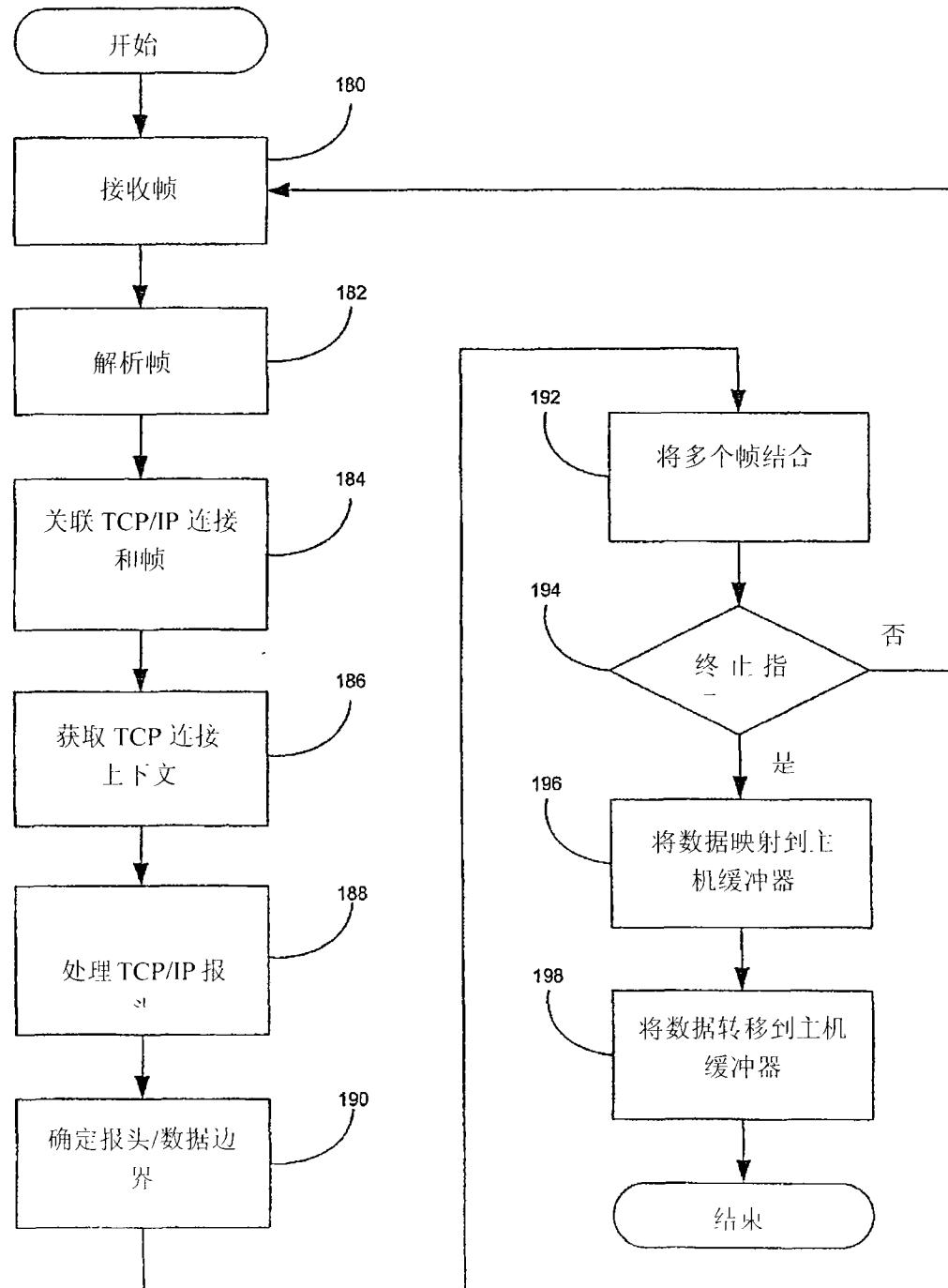


图 1E

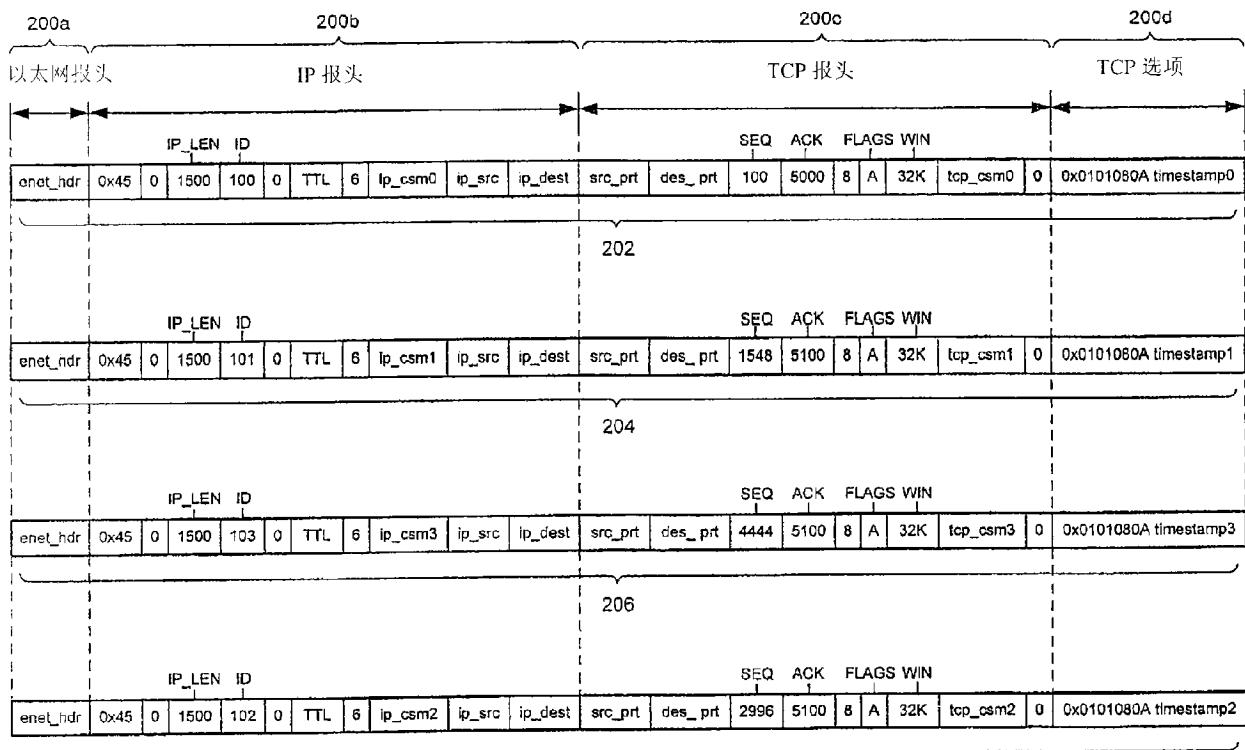


图 1F

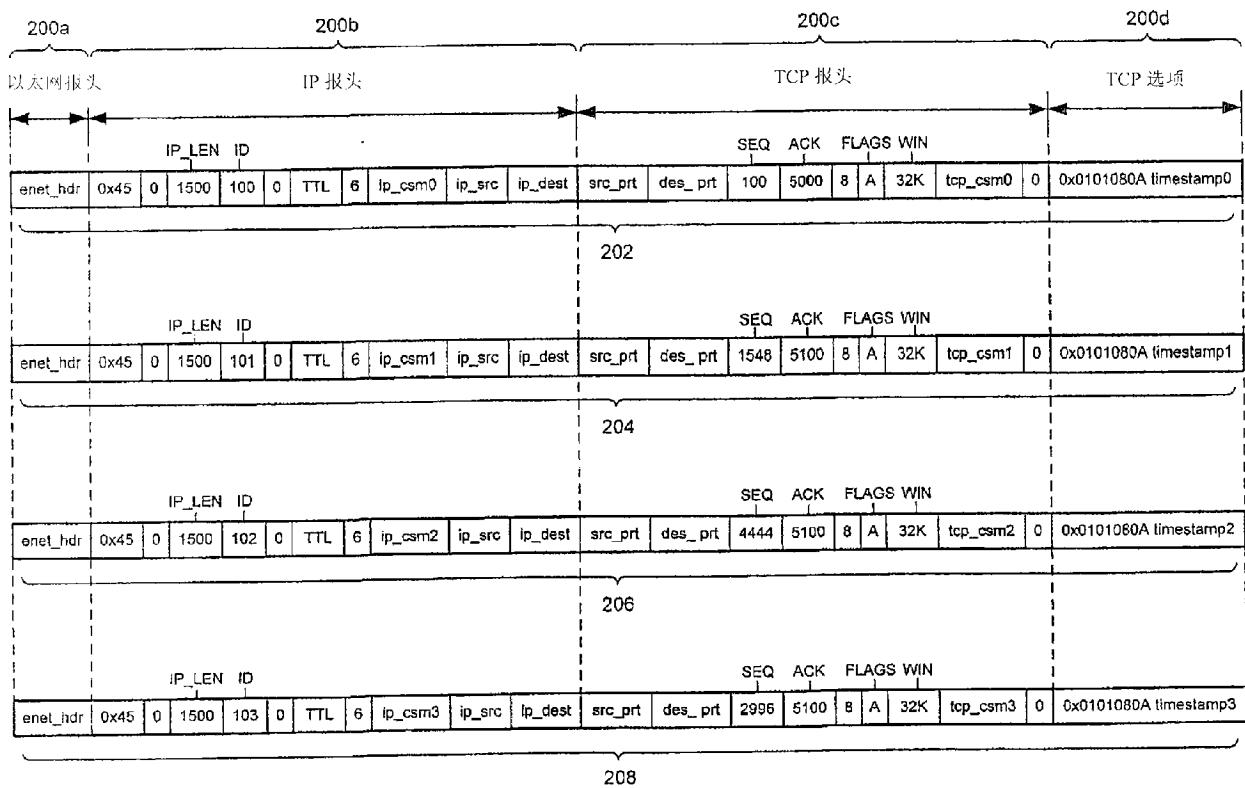


图 2A

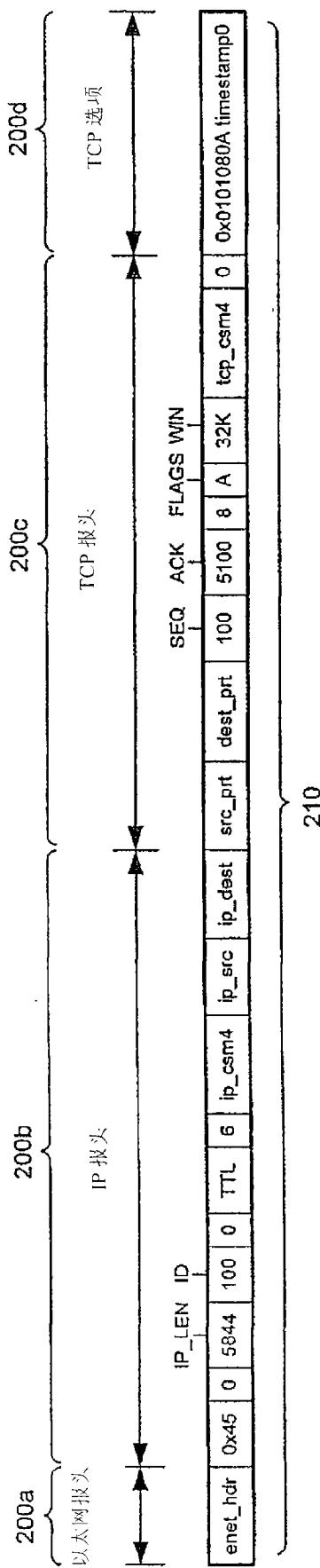


图 2B

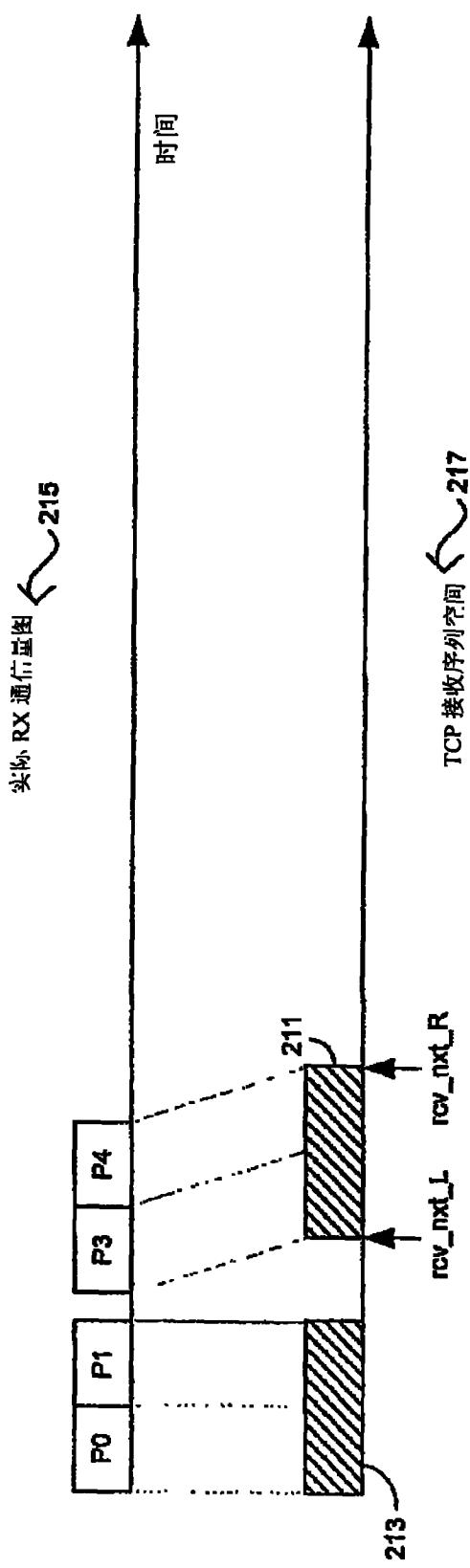


图 2C

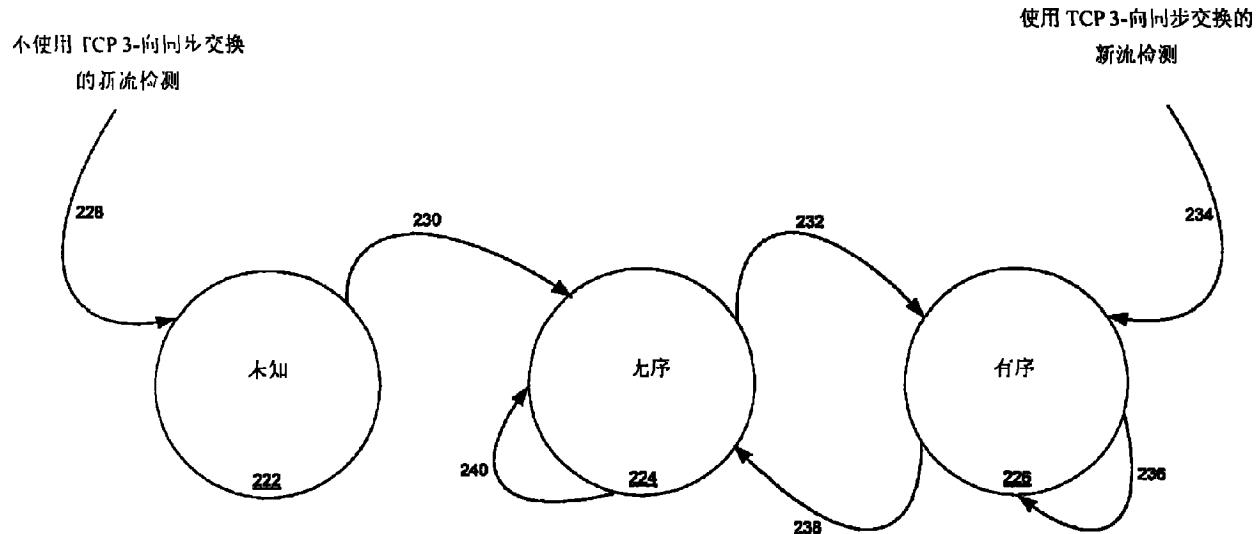


图 2D

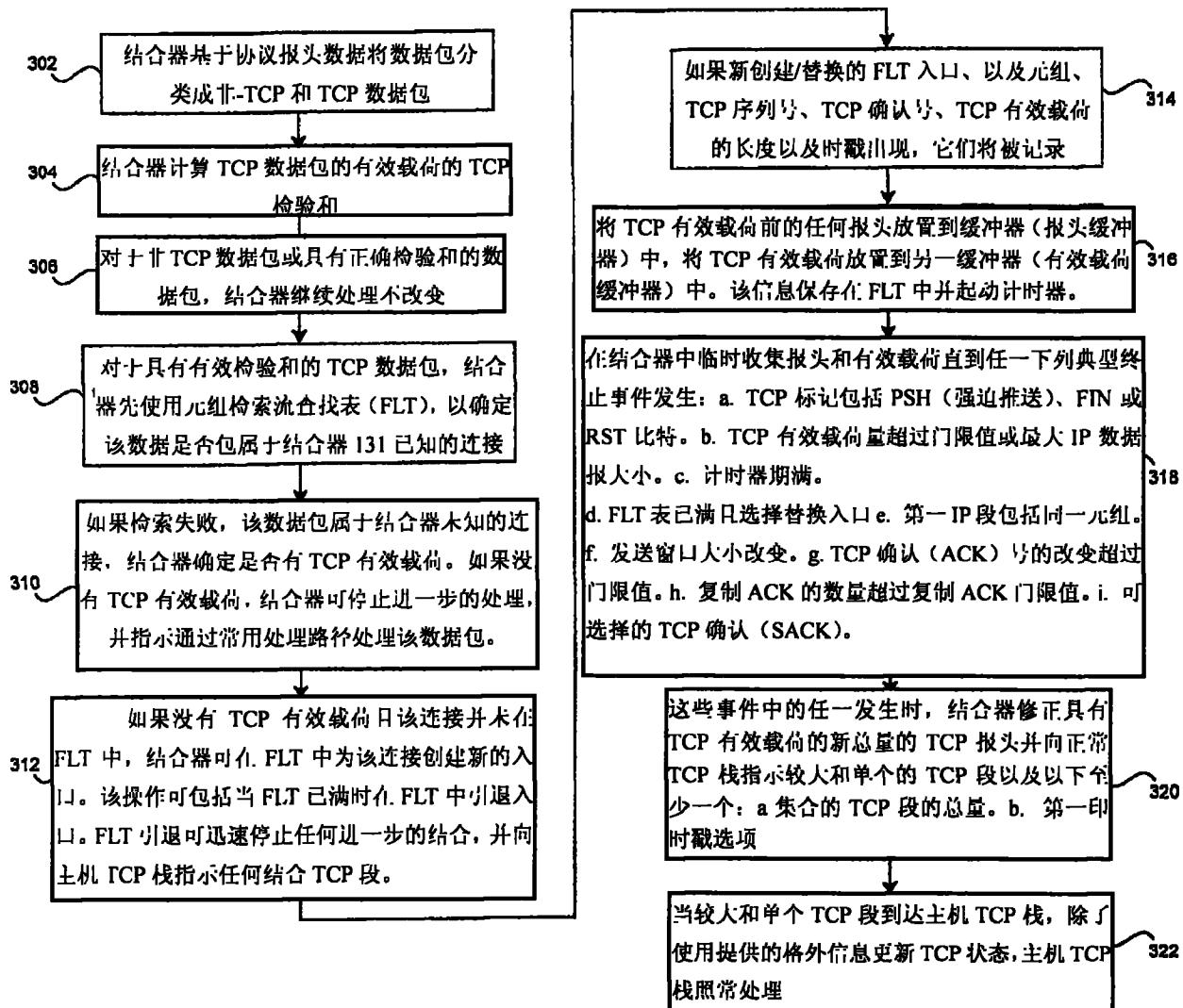


图 3