(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2022 年 1 月 27 日 (27.01.2022)

WIPO | PCT

(10) 国际公布号
WO 2022/017454 A1

(51) 国际专利分类号:
H04L 12/911 (2013.01)

(21) 国际申请号: PCT/CN2021/107828

(22) 国际申请日: 2021 年 7 月 22 日 (22.07.2021)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202010716127.9 2020年7月23日 (23.07.2020) CN

(71) 申请人: 华 为 技 术 有 限 公 司 (**HUAWEI TECHNOLOGIES CO., LTD.**) [CN/CN]; 中国广东省深圳市 龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 廖志坚 (**LIAO, Zhijian**); 中国广东省深圳市 龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 包锦程 (**BAO, Jincheng**); 中国广东省深圳市 龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

WO 2022/017454 A1

(54) **Title:** MESSAGE PROCESSING METHOD, NETWORK DEVICE AND RELATED DEVICE

(54) 发明名称: 一种报文处理方法、网络设备以及相关设备

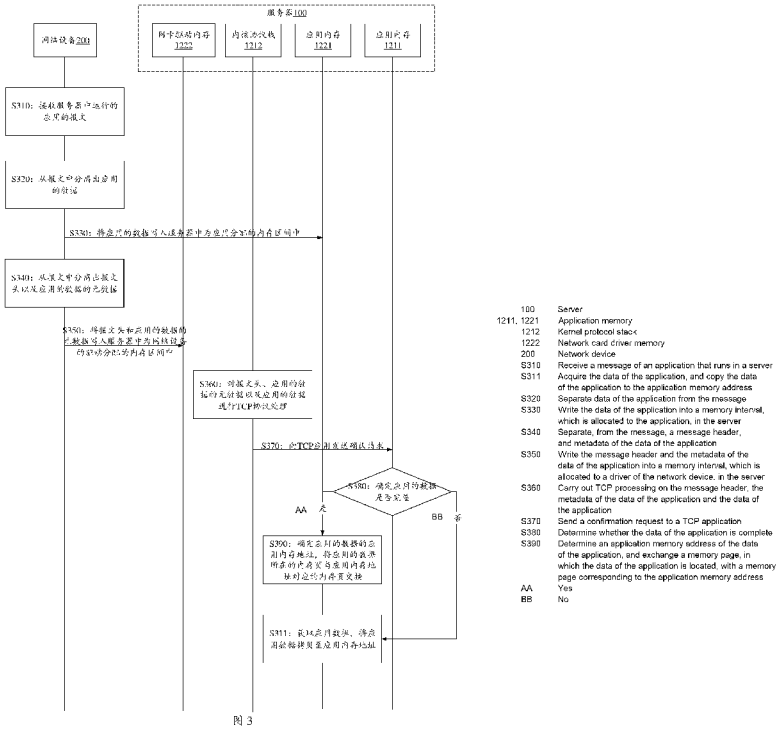

图 3

(57) **Abstract:** Provided are a message processing method, a network device and a related device. The method comprises the following steps: a network device receiving a message of an application that runs in a server, separating data of the application from the message, and writing the data of the application into a memory interval, which is allocated to the application, in the server. Therefore, during the process of the server processing the message, there is no need to repeatedly copy the data of the application, thereby reducing the memory occupancy rate during the message processing process and improving the message processing efficiency.

[见续页]

ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) **指定国**(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要：本申请提供了一种报文处理的方法、网络设备以及相关设备，该方法包括以下步骤：网络设备接收服务器中运行的应用的报文，从报文中分离出应用的数据，将应用的数据写入服务器中为应用分配的内存区间中，使得服务器处理该报文的过程中，无需重复拷贝应用的数据，减少报文处理过程中的内存占用率，提高报文处理效率。

一种报文处理方法、网络设备以及相关设备

技术领域

本申请涉及通信领域，尤其涉及一种报文处理方法、网络设备以及相关设备。

背景技术

传输控制协议(Transmission Control Protocol, TCP)是互联网核心协议之一，由于TCP协议可以保证数据通信的完整性和可靠性，被广泛应用在对准确性要求相对较高的场景下，比如文件传输场景中，TCP应用可以基于TCP协议组中的文件传输协议（File Transfer Protocol，FTP）、超文本传输协议（Hyper Text Transfer Protocol，HTTP）实现文件传输的功能；再比如发送或接收邮件的场景中，TCP应用可以基于TCP协议组中的简单邮件传输协议（Simple Mail Transfer Protocol，SMTP）或者交互邮件访问协议（Interactive Mail Access Protocol，IMAP）实现邮件收发的功能。

但是，基于TCP协议进行数据通信的服务器在接收TCP报文时，往往是与其相连的外接网卡首先接收到该报文，外接网卡将该报文写入服务器的网卡驱动内存后，TCP应用再从网卡驱动内存中获取报文中的应用的数据，然后将该应用的数据拷贝至应用内存，多余的拷贝步骤导致TCP报文处理的内存占用率高，处理效率受到限制。

发明内容

本申请提供了一种报文处理的方法、网络设备及相关设备，能够减少报文处理过程中的内存占用率，提高报文处理效率。

第一方面，提供了一种报文处理的方法，该方法应用于网络设备，该网络设备连接至服务器，该方法包括以下步骤：接收服务器中运行的应用的报文，从报文中分离出应用的数据，将应用的数据写入服务器中为应用分配的内存区间中。

实施第一方面描述的方法，网络设备在将报文写入服务器之前，先从报文中分离出应用的数据，然后将应用的数据写入应用内存，整个报文处理的过程无需重复拷贝应用的数据，降低了报文处理过程中的内存占用率，提高报文处理效率。

在一可能的实现方式中，该方法还包括：从报文中分离出报文头及应用的数据的元数据；将报文头和应用的数据的元数据存储至服务器为网络设备的驱动分配的内存区间中。

实施上述实现方式，网络设备从报文中分离出应用的数据后，再将报文头以及应用的数据的元数据也从报文中分离出，然后将应用的数据写入应用内存，将报文头以及应用的数据的元数据写入网卡驱动内存，这样，服务器的TCP应用可根据网卡驱动内存中的元数据确定应用的数据应存储的应用内存地址后，将应用的数据所在的内存页与该应用内存地址对应的内存页交换即可完成一次数据通信的过程，该方法避免了报文由网络设备传输至TCP应用内存页的过程中多次拷贝情况的发生，提高报文处理效率，降低内存占用率。

在一可能的实现方式中，从报文中分离出应用的数据，包括：根据定界模板从报文中分离出应用的数据，定界模板定义了应用的数据与报文中其他数据的分离规则。

应理解，相同应用的元数据是按照统一规则生成的，因此相同应用的定界模板相同，不同应用的定界模板可能相同也可能不同，定界模板也可根据元数据生成的规则确定。在应用启动后，应用可向网络设备下发与其应用类型对应的定界模板。

可选地，如果 TCP 应用的元数据长度统一为 L1，而报文的报文头长度也是固定的，那么报文头之后长度为 L1 的数据即为元数据，根据元数据确定应用的数据长度 L2，那么元数据之后长度为 L2 的数据即为应用的数据，从而将应用的数据从报文中分离。

可选地，如果 TCP 应用的元数据统一以某个分隔符结尾，比如分隔符可以是换行符、空格符、冒号等等，那么定界模板可以根据该分隔符获得元数据，再根据元数据描述的应用的数据长度，将应用的数据从报文中分离。同理，报文的报文头长度是固定的，那么报文头之后、分隔符之前的数据即为元数据，根据该元数据确定应用的数据长度 L2 后，元数据之后长度为 L2 的数据即为应用的数据，从而将应用的数据从报文中分离。

实施上述实现方式，TCP 应用根据元数据的生成规则生成定界模板后，将定界模板下发至网络设备，使得网络设备可以根据定界模板对报文进行拆分，将应用的数据从报文中分离，并将其写入应用内存，避免了报文写入服务器之后，再将应用的数据拷贝入应用内存的冗余步骤，且降低了报文处理过程中的内存占用，提高报文处理效率。

在一可能的实现方式中，接收服务器中运行的应用的报文包括：将属于同一个数据流的多个子报文聚合为报文，其中，属于同一个数据流的多个子报文的源网际互联协议 IP 地址和目的 IP 地址相同。

实施上述实现方式，网络设备将属于同一个数据流的多个报文聚合后，使得待处理的报文数量减少，网络设备将报文写入服务器的次数也随之减少，提高网络设备的报文处理效率。

在一可能的实现方式中，从报文中分离出应用的数据之前，方法还包括：确定报文包括一个数据流中的完整数据。

实施上述实现方式，对于支持 TCP 减负引擎(TCP Offload Engine. TOE)功能的网络设备，可在网络设备内部提前进行 TCP 协议处理，比如乱序处理、拥塞控制、重传等等，协议处理后的报文包括一个数据流中的完整数据。可以理解的，网络设备确定报文包括一个数据流中的完整数据之后，此时报文不会出现失序、重复等情况，此时再使用定界模板对报文进行拆分，可使得报文拆分的准确率提高。

第二方面，提供了一种报文处理的方法，该方法应用于服务器，该服务器与网络设备连接，该方法包括以下步骤：服务器接收网络设备发送的应用的数据，将其存储至服务器为应用分配的内存区间中，接收网络设备发送的报文头以及应用的数据的元数据，将其存储至服务器为网络设备的驱动分配的内存区间中，然后服务器确定服务器中为应用分配的内存区间中的应用的数据是否完整，并在该应用数据完整的情况下，根据应用的数据的元数据确定应用的数据的应用内存地址，将该地址对应的内存页与应用的数据所在的内存页交换，使得应用的数据存入该应用内存地址。

具体实现中，内存页交换可以是将指向应用的数据的指针与指向该应用内存地址的指针进行交换，或者将应用的数据所在的内存页的虚拟地址替换成该应用内存地址，本申请不对内存页交换的具体实现方式进行限定。

实施第二方面描述的方法，服务器中运行的 TCP 应用根据元数据，确认应用的数据的应用内存地址，然后将该应用内存地址对应的应用内存页与应用的数据所在的内存页进行交换，从而避免了多次拷贝应用的数据造成的资源浪费，提高报文处理效率。

在一种可能的实现方式中，服务器根据元数据确定应用的数据的应用内存地址之前，还

可根据定界模板，确定网络设备写入服务器的报文头、应用的数据的元数据以及元数据是同一个数据流的完整数据，确定是完整数据的情况下，再根据元数据确定应用的数据的应用内存地址，确定不是完整数据的情况下，服务器根据网络设备写入服务器的报文头、应用的数据的元数据以及元数据，获取应用的数据和应用的数据的应用内存地址，再将应用的数据拷贝至应用内存地址。

其中，由于网络设备接收到的报文可能存在乱序、重复等情况，此时网络设备将应用的数据从报文中分离时，可能会出现分离出的应用的数据不完整的情况，因此在网络设备将应用的数据写入服务器之后，应用可以根据定界模板再次确认应用的数据是否完整，这里的定界模板与网络设备拆分报文时使用的定界模板相同。举例来说，TCP 应用可根据定界模板获取元数据，比如根据分隔符确定元数据，或者根据元数据的固定长度确定元数据，然后根据元数据获取应用的数据的长度，如果该长度与被写入应用内存中的应用的数据长度相同，表示网络设备对报文拆分正确，相反地，如果该数据长度与被写入应用内存中的应用的数据的长度不同，表示网络设备对报文拆分错误，应理解，上述 TCP 应用根据定界模板确定网络设备是否正确拆分报文的过程用于举例说明，本申请不对此进行限定。

实施上述实现方式，服务器使用定界模板重新对报文头、应用的数据的元数据以及应用的数据进行拆分，从而避免了由于网络设备拆分错误导致最终写入应用内存地址的数据不完整情况的发生，提高数据传输的可靠性。

第三方面，提供了一种网络设备，网络设备连接至服务器，网络设备包括：接收单元，用于接收服务器中运行的应用的报文；分离单元，用于从报文中分离出应用的数据；写入单元，用于将应用的数据写入服务器中为应用分配的内存区间中。

在一种可能的实现方式中，分离单元还用于从报文中分离出报文头及应用的数据的元数据；写入单元还用于将报文头和应用的数据的元数据存储至服务器为网络设备的驱动分配的内存区间中。

在一种可能的实现方式中，分离单元用于根据定界模板从报文中分离出应用的数据，定界模板定义了应用的数据与报文中其他数据的分离规则。

在一种可能的实现方式中，接收单元用于将属于同一个数据流的多个子报文聚合为报文，其中，属于同一个数据流的多个子报文的源网际互联协议 IP 地址和目的 IP 地址相同。

在一种可能的实现方式中，网络设备还包括确定单元，确定单元用于在分离单元从报文中分离出应用的数据之前，确定报文包括一个数据流中的完整数据。

第四方面，提供了一种服务器，该服务器与网络设备连接，该服务器包括：应用模块，用于接收网络设备发送的应用的数据，将其存储至服务器为应用分配的内存区间中，网卡驱动，用于接收网络设备发送的报文头以及应用的数据的元数据，将其存储至服务器为网络设备的驱动分配的内存区间中，内核协议栈，用于确定服务器中为应用分配的内存区间中的应用的数据是否完整，应用模块还用于在该应用数据完整的情况下，根据应用的数据的元数据确定应用的数据的应用内存地址，将该地址对应的内存页与应用的数据所在的内存页交换，使得应用的数据存入该应用内存地址。

在一种可能的实现方式中，应用模块还用于在交换单元根据元数据确定应用的数据的应用内存地址之前，根据定界模板，确定网络设备写入服务器的报文头、应用的数据的元数据以及元数据是同一个数据流的完整数据，确定是完整数据的情况下，再根据元数据确定应用的数据的应用内存地址，确定不是完整数据的情况下，服务器根据网络设备写入服务器的报文头、应用的数据的元数据以及元数据，获取应用的数据和应用的数据的应用内存地址，再

将应用的数据拷贝至应用内存地址。

第五方面，提供了一种报文处理系统，包括服务器和网络设备，其中，服务器用于实现如第二方面或第二方面任一种可能的实现方式中描述的方法的操作步骤，网络设备用于实现如第一方面或第一方面任一种可能的实现方式中描述的方法的操作步骤。

第六方面，提供了一种计算机程序产品，包括计算机程序，当计算机程序被计算设备读取并执行时，实现如第一方面或第二方面所描述的方法。

第七方面，提供了一种计算机可读存储介质，包括指令，当指令在计算设备上运行时，使得计算设备实现如第一方面或第二描述所描述的方法。

第八方面，提供了一种网络设备，包括处理器和通信接口，该通信接口用于接收报文，该处理器用于执行如第一方面描述的方法以对报文进行处理。

第九方面，提供了一种服务器，包括处理器和存储器，处理器执行存储区中的代码实现如第二方面描述的方法。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍。

图 1 是本申请提供的一种报文处理系统的结构示意图；

图 2 是相关技术中的一种报文处理方法的步骤流程示意图；

图 3 是本申请提供的一种报文处理方法的步骤流程示意图；

图 4 是本申请提供的一种应用场景下从报文中分离应用的数据的流程示意图；

图 5 是本申请提供的一种报文聚合的步骤流程示意图；

图 6 是本申请提供的报文处理方法在一应用场景下的步骤流程示意图；

图 7 是本申请提供的一种报文处理方法的步骤流程示意图；

图 8 是本申请提供的一种网络设备的结构示意图；

图 9 是本申请提供的一种网络设备的硬件结构示意图。

具体实施方式

首先，对本申请涉及的部分术语进行解释说明。值得注意的，本申请的实施方式部分使用的术语仅用于对本申请的具体实施例进行解释，而非旨在限定本申请。

协议栈(Protocol Stack)：又称协议堆叠，是计算机网络协议套件的一个具体地软件实现。协议套件中的一个协议通常只为一个目的而设计的，这样可以使得设计更容易。因为每个协议模块通常都要和上下两个其他协议模块通信，它们通常可以想象成是协议栈中的层。最低级的协议总是描述与硬件的物理交互。举例来说，三台电脑分别为 A、B 以及 C，其中，电脑 A 和电脑 B 都有无线电设备，可以通过网络协议 IEEE802.11 通信，电脑 B 和电脑 C 通过电缆连接来交换数据，比如以太网，这样，电脑 A 与电脑 C 之间的数据通信只能通过电脑 B 来传输，而无法直接传输，为了解决这一问题，可以在两个协议之上建立一个新的协议，比如 IP 协议，这样就形成了两个协议栈，实现了电脑 A 和电脑 C 之间的数据通信。

TCP：TCP 提供一种面向连接的可靠的字节流服务。TCP 将用户数据打包构成待处理报文段，它发送数据时启动一个定时器，另一端接收到数据后进行确认，然后对失序的数据重新排列，丢弃重复的数据，因此基于 TCP 的数据通信具有较高的安全性和可靠性，被广泛应

用在对准确性要求相对较高的场景下，比如文件传输场景中，TCP 应用可以基于 TCP 协议组中的文件传输协议(File Transfer Protocol, FTP)、超文本传输协议(Hyper Text Transfer Protocol, HTTP)实现文件传输的功能；再比如发送或接收邮件的场景中，TCP 应用可以基于 TCP 协议组中的简单邮件传输协议(Simple Mail Transfer Protocol, SMTP)或者交互邮件访问协议(Interactive Mail Access Protocol, IMAP)实现邮件收发的功能。应理解，上述 TCP 应用的类型仅用于举例说明，本申请提供的报文处理方案适用于任何 TCP 应用，本申请不对 TCP 应用进行限定。

直接内存存取(Direct Memory Access, DMA)：设备直接与计算机内存进行报文处理，实现了 DMA 设备直接访问服务器内存，缩短了报文处理路径，不仅提升服务器的 IO 性能，也降低了 CPU 的负载压力。

内存页：将内存的地址空间人为地划分为大小相等的若干份，一份对应一个内存页，处理器以页为单位对内存进行写入和读取。

TOE：TOE 一般由软硬两部分组件构成，将传统的 TCP/IP 协议栈的功能进行延伸，把网络数据流量的 TCP 协议处理工作全部转移到网卡的集成硬件中进行，服务器只承担 TCP/IP 应用的处理任务，从而减轻了服务器的处理压力。

首先，对本申请适用的应用场景进行解释说明。

图 1 是一种与网络相连的服务器的结构示意图，其中，该服务器 100 与网络设备 200 相连，网络设备 200 与网络 300 相连，当网络 300 中的其他服务器向该服务器 100 发送待处理报文时，网络设备 200 首先接收到该待处理报文，然后将该待处理报文发送至服务器 100，服务器 100 对该待处理报文进行处理后，完成一次报文处理。

网络设备 200 是一个使得服务器 100 与网络 300 相连接的硬件设备，具体可以是网卡（Nic），也可以是 TOE 网卡，本申请不对此具体限定。其中，一个服务器 100 可以与一个或者多个网络设备 200 相连，图 1 以服务器 100 与一个网络设备 200 相连为例进行了说明，本申请不对此进行限定。

服务器 100 是通用的物理服务器，例如，ARM 服务器或者 X86 服务器，服务器 100 包括处理器 110 和内存 120，其中，处理器 110 和内存 120 通过内部总线 130 相互连接，内部总线 130 可以是外设部件互连标准(Peripheral Component Interconnect, PCI)总线或扩展工业标准结构(Extended Industry Standard Architecture, EISA)总线等。需要说明的，图 1 仅仅是服务器 100 的一种可能的实现方式，实际应用中，服务器 100 还可以包括更多或更少的部件，这里不作限制。

处理器 110 可以由至少一个通用处理器构成，例如中央处理器(Central Processing Unit, CPU)，或者 CPU 和硬件芯片的组合。上述硬件芯片可以是专用集成电路(Application-Specific Inegrated Circuit, ASIC)、可编程逻辑器件(Programmable Logic Device, PLD)或其组合。上述 PLD 可以是复杂可编程逻辑器件(Complex Programmable Logic Device, CPLD)、现场可编程逻辑门阵列(Field-Programmable Gate Array，FPGA)、通用阵列逻辑(Generic Array Logic, GAL)或其任意组合。处理器 110 执行各种类型的数字存储指令，例如存储在内存 120 中的软件或者固件程序，它能使服务器 1 提供较宽的多种服务。

内存 120 可以是易失性存储器(Volatile Memory)，例如随机存取存储器(Random Access Memory, RAM)、动态随机存储器（Dynamic RAM，DRAM）、静态随机存储器（Static RAM，SRAM）、同步动态随机存储器（Synchronous Dynamic RAM，SDRAM）、双倍速率同步动态

随机存储器（Double Data Rate RAM，DDR）、高速缓存（Cache）等等，内存 120 还可以包括上述种类的组合。内存 120 包括程序代码 121，其中，程序代码 121 可以包括一个或多个软件模块，比如图 1 所示的 TCP 应用 1211、内核协议栈 1212 以及网卡驱动 1213 的代码，TCP 应用 1211 是基于 TCP 协议实现各种功能的应用模块，内核协议栈 1212 可以理解为是操作系统的一部分，主要用于处理内核协议栈中的报文，网卡驱动 1213 是一种可以使 CPU 控制和使用网络设备 200（比如 Nic 或者 TOE 网卡）的特殊程序，相当于网络设备 200 的硬件接口，操作系统通过该接口可以控制网络设备 200。内存 120 还包括应用内存 1221 和网卡驱动内存 1222，其中，网卡驱动内存 1222 是网卡驱动 1213 向内存 120 申请的一段内存，内核协议栈 1212 可对网卡驱动内存 1222 中的数据进行处理，应用内存 1221 是 TCP 应用 1211 向内存 120 申请的一段内存，TCP 应用 1211 可对应用内存 1221 中的数据进行处理。

下面结合图 2，对相关技术中图 1 所示的服务器 100 接收并处理来自网络 300 的 TCP 报文的具体流程进行解释说明。

当其他服务器向该服务器 100 发送数据时，网络设备 200 最先接收到来自网络 300 的 TCP 报文，TCP 报文包括报文头、元数据以及应用的数据，其中，如图 2 所示，报文头位于元数据之前，元数据位于应用的数据之前。

其中，报文头至少包括四元组（源 IP 地址、目的 IP 地址、源端口、目的端口），应理解，同一个 TCP 流(Stream)的待处理报文，其报文头中的四元组是相同的。

元数据至少包括应用的数据的长度和应用的数据的控制信息，其中，控制信息用于供 TCP 应用确定该应用的数据的应用内存地址，举例来说，控制信息可以包括上下文信息，TCP 应用接收到报文后，TCP 应用可根据上下文信息，确定该报文中的应用的数据 D2 的上下文数据 D1 和 D3 所在的应用内存地址分别为 Add1 和 Add3，从而确定应用的数据的应用内存地址为 Add2，其中，Add1、Add2 和 Add3 是一段连续的内存。具体实现中，TCP 应用根据元数据确定的应用内存地址可以是应用内存 1221 中的某个内存页的地址，本申请不对元数据和根据元数据确定的应用内存地址的具体形式进行限制。

应用的数据是其他服务器向服务器 100 发送的原始数据，即负载（Payload）。应理解，通常在数据传输时，为了使数据传输更加可靠，将会把原始数据的头部和/或尾部增加一定的辅助信息，比如数据量的大小、校验位等等，使得原始数据在传输过程中不易丢失，原始数据加上辅助信息就形成了传输通道的基本传输单元，也就是数据帧、数据包或者 TCP/IP 报文等等，其中的原始数据即为应用的数据。

应理解，由于每个 TCP 报文的大小是固定的，一般为 1480 字节左右，如果要一次性发送大量数据，数据需要进行分片处理后，分成多个报文进行传输，比如一个 10MB 的文件，需要发送 7100 多个报文，发送侧服务器发送报文的时候，TCP 协议为每个报文编号(Sequence Number, SEQ)，以便接收侧服务器按照 SEQ 顺序将收到的多个报文还原出 10M 的原文件。图 2 以应用的数据 1 被分片为 3 个应用的数据（应用的数据 1A、应用的数据 1B 以及应用的数据 1C）为例进行了说明，其中，报文 1 包括报文头、元数据 1 以及应用的数据 1A，报文 2 包括报文头以及应用的数据 1B，报文 3 包括报文头、应用的数据 1C、元数据 2 以及应用的数据 2，元数据 1 包括应用的数据 1（应用的数据 1A、应用的数据 1B 以及应用的数据 1C）的控制信息，TCP 应用可以根据该控制信息，确定应用的数据 1 的应用内存地址，元数据 2 包括应用的数据 2 的控制信息，TCP 应用可以根据该控制信息确定应用的数据 2 的应用内存地址。假设报文 1、报文 2 以及报文 3 属于同一个 TCP 流，那么报文 1、报文 2 以及报文 3

的报文头的四元组相同，但是报文 1、报文 2 以及报文 3 中应用的数据 1 的 SEQ 不同。

在上述应用场景中，如图 2 所示，当网络设备 200 接收到报文 1、报文 2 以及报文 3 之后，图 1 所示的服务器接收并处理来自网络 300 的 TCP 报文的具体流程包括以下步骤：

步骤 1、网络设备 200 通过 DMA 技术将报文 1、报文 2 以及报文 3 写入服务器 100 的网卡驱动内存 1222。

具体实现中，网卡驱动内存 1222 是网卡驱动 1213 向服务器 100 申请的一段内存，网络设备 200 将报文 DMA 到网卡驱动内存 1222 之后，可向内核协议栈发送协议处理请求，该协议处理请求中包括该报文的地址。

步骤 2、内核协议栈 1212 对网卡驱动内存 1222 中的报文进行 TCP 协议处理。

具体地，内核协议栈 1212 接收到网卡驱动发送的协议处理请求之后，内核协议栈 1212 可根据协议处理请求中的报文的地址，对该报文进行 TCP 协议处理，以确保数据通信的完整性。

具体实现中，由于报文头的长度和格式是固定的，比如报文头的长度为 20kb，内核协议栈 1212 可以从该网卡内存页的头部开始读取 20kb 的数据，从而获得该报文的报文头，然后根据该报文头完成拥塞控制、乱序处理、重传等等 TCP 协议处理步骤，对失序的数据重新排列，丢弃重复的数据，保证该报文头对应的 TCP 流中的全部报文都已写入服务器中。

例如图 2 所示，内核协议栈 1212 对报文头进行 TCP 协议处理，根据报文头中的 SEQ 确定报文 1~3 是否都已被写入网卡驱动内存 1222，如果报文 2 遗失，内核协议栈 1212 可以向网络设备发送重传报文 2 的请求，在确认没有报文遗失的情况下，向 TCP 应用发送应用处理请求，该应用处理请求中包括上述报文的地址。

步骤 3、TCP 应用对网卡驱动内存 1222 中的元数据进行处理，确定应用的数据的应用内存地址。

具体地，每个应用的数据的元数据是按照固定格式生成的，可根据元数据的格式特点来获取元数据。如果元数据的长度是固定的，TCP 应用接收到内核协议栈发送的应用处理请求后，首先根据元数据的长度，比如 40kb，从报文头之后读取 40kb 的数据至元数据内存，该 40kb 的数据即为元数据，然后再对元数据进行解析，确定应用的数据的长度和应用内存地址；如果元数据的尾部存在作为分隔符的特殊符号，比如换行符、空格符、冒号等等，TCP 应用接收到内核协议栈发送的应用处理请求后，首先确定分隔符的位置，然后将报文头之后、分隔符之前的数据读取至元数据内存，再对其进行解析，确定应用的数据的长度和应用内存地址。其中，该元数据内存可以是 TCP 应用提前向内存申请的一段用于临时存储元数据的内存，TCP 应用对应用元数据内存中的元数据进行解析后，可以将内存中的元数据删除，释放出存储空间。应理解，图 3 仅用于举例说明，具体实现中，元数据还可以包括更多的内容，这里不一一举例说明。

步骤 4、TCP 应用将网卡驱动内存 1222 中的应用的数据拷贝入应用内存地址。

仍以图 3 为例，TCP 应用根据元数据确认应用的数据的长度为 1400kb、应用的数据的应用内存地址为 Add1 后，可以将 40kb 元数据之后的 1400kb 数据拷贝至 Add1，该 1400kb 数据即为 TCP 应用所需的应用的数据。

综上可知，当服务器 100 外接的网络设备 200 接收到 TCP 报文后，网络设备 200 将报文 DMA 到服务器提前为网卡驱动划分的网卡驱动内存 1222 中，服务器 100 的内核协议栈 1212 先对网卡驱动内存 1222 中的报文进行处理后，再将报文中的应用的数据拷贝至应用内存 1221 中。简单来说，报文需要先写入网卡内存，再由 TCP 应用从网卡内存中拷贝报文中的应用的

数据至应用内存，多余的拷贝步骤导致TCP报文处理的内存占用率高，处理效率受到限制。

为了解决上述TCP报文处理时内存占用率高、处理效率受限的问题，本申请提供了一种报文处理方法。如图3所示，该方法包括以下步骤：

S310：网络设备200接收服务器100中运行的应用的报文。

在一实施例中，上述应用的报文可以是网络设备200接收到的一个报文，比如图2中的报文1、或者报文2、或者报文，也可以是由网络设备200将接收到的属于同一个数据流（比如TCP流）的多个子报文进行报文聚合处理后得到的，其中，属于同一个数据流的多个子报文的源IP地址和目的IP地址相同。网络设备200可以根据每个报文的TCP头中的四元组确定同一个TCP流的多个报文，其中，同一个TCP流中的多个报文的四元组相同，然后将属于同一个TCP流的多个报文聚合为一个报文，该报文也是由TCP报文头、元数据和应用的数据构成，其中，每个元数据后面紧接着与其对应的应用的数据，每个应用的数据后面紧接着下一个应用的数据的元数据。

假设网络设备200接收到了如图2所示的报文1~3，且报文1~报文3是属于同一个TCP流的三个报文，网络设备200对该报文1~3进行聚合获得的报文0可以如图5所示，其中，应用的数据1包括了报文1中的应用的数据1A、报文2中的应用的数据1B以及报文3中的应用的数据1C。具体实现中，网络设备200可以通过大量接收卸载(Large Receive Offload, LRO)、接收方向聚合(Receive Side Coalescing, RSC)等算法实现报文聚合，这里不作具体限定。可以理解的，网络设备200将属于同一个TCP流的多个报文聚合后，可以减少网络设备200将报文写入服务器的次数，提高网络设备200的报文处理效率。

S320：网络设备200从报文中分离出应用的数据。

在一实施例中，网络设备200可根据定界模板从报文中分离出应用的数据，其中，该定界模板定义了应用的数据与报文中其他数据的分离规则。应理解，由于网络设备200接收到的报文是未经过TCP协议处理的报文，因此该报文可能是乱序、或者丢包的报文，该种情况下使用定界模板对报文进行拆分，可能会出现拆分错误的情况，也就是拆分出的应用的数据不完整的情况，比如只包括部分应用的数据，或者除了应用的数据还包括其他数据的情况。在网络设备200对报文拆分正确的情况下，分离出的应用的数据只包括完整的应用的数据，剩余报文只包括完整的报文头和完整元数据。

其中，相同TCP应用的元数据是按照统一规则生成的，相同TCP应用的定界模板相同，不同TCP应用的定界模板可能相同也可能不同，因此在步骤S310之前，TCP应用启动时，可以向网络设备200下发与其应用类型对应的定界模板。具体地，TCP应用可向网络设备200的驱动（比如网卡驱动）发送接口调用请求，网络设备200的驱动响应于该请求，向TCP应用提供网络设备200的接口，TCP应用调用该接口将定界模板下发至网络设备200。

为了便于本申请更好地被理解，示例性地，下面对两种元数据格式对应的定界模板进行举例说明。

第一种定界模板：如果TCP应用的元数据长度统一为L1，而报文的报文头长度也是固定的，那么报文头之后长度为L1的数据即为元数据，根据元数据确定应用的数据长度L2，那么元数据之后长度为L2的数据即为应用的数据，从而将应用的数据从报文中分离。

例如，图4是一种报文格式的示意图，在图4所示的例子中，应用1的应用的数据的元数据的长度为40kb，其中，前10kb描述了该应用的数据的长度，比如应用的数据长度为1400kb，后30kb描述了该应用的数据的控制信息。那么TCP应用接收到内核协议栈发送的

应用处理请求后，首先根据元数据的长度40kb，从报文头之后读取40kb的数据至应用内存，该40kb的数据即为应用的数据1的元数据，然后再根据该元数据的前10kb内容得到应用的数据长度为1400kb，再根据该元数据的后30kb中的控制信息，确定该应用的数据的应用内存地址Add1。应理解，图4仅用于举例说明，具体实现中，元数据还可以包括更多的内容，这里不一一举例说明。

第二种定界模板：如果TCP应用的元数据统一以某个分隔符结尾，比如分隔符可以是换行符、空格符、冒号等等，那么定界模板可以根据该分隔符获得元数据，再根据元数据描述的应用的数据长度，将应用的数据从报文中分离。同理，报文的报文头长度是固定的，那么报文头之后、分隔符之前的数据即为元数据，根据该元数据确定应用的数据长度L2后，元数据之后长度为L2的数据即为应用的数据，从而将应用的数据从报文中分离。

应理解，上述两种定界模板仅用于举例说明，具体实现中，不同TCP应用的定界模板可以根据该TCP应用的元数据的格式来确定，本申请不对此进行具体限定。

值得注意的是，如果报文是未经过报文聚合处理的报文时，比如图5实施例中的报文1~报文3，使用的定界模板对该类报文进行拆分之前，需要先识别报文头，确定属于同一个TCP流的多个报文，然后再使用定界模板，对同一个TCP流中的多个报文进行拆分，从同一个TCP流中每个报文中分离出应用的数据，然后将同一个TCP流中分离出的应用的数据全部写入网卡驱动内存1222，报文头和元数据全部写入应用内存1221。

仍以图5所示的3个报文为例，假设元数据长度固定为40kb，每个报文的总长度为1400kb，网络设备200没有对报文1~3进行聚合，在该应用场景下，使用上述第一种定界模板拆分报文时，网络设备200可先根据报文头确定报文1~3属于同一个TCP流，并且该TCP流中报文的读取顺序为报文1、报文2和报文3，然后读取报文1中的元数据，确定应用的数据1的长度为3000KB之后，读取报文1中的应用的数据1A(假设应用的数据长度为1000KB)，此时应用的数据1还有2000KB未读取，可从报文2再读应用的数据1B（假设应用的数据长度为1400KB)，此时应用的数据1还有600KB未读取，可继续从报文3读取应用的数据1C（假设应用的数据长度为600KB)，从而获得长度为3000KB的应用的数据1，继续读取40kb数据即可获得应用的数据2的元数据，以此类推，最后将元数据1以及报文头写入网卡驱动缓内存，将应用的数据1A、应用的数据1B以及应用的数据1C写入应用内存。第二种定界模板的读取方法类似，这里不再重复赘述。

S330：网络设备200将应用的数据写入服务器100中为应用分配的内存区间中，即图1所示的应用内存1221。

具体实现中，网络设备200可通过DMA技术将分离出的应用的数据写入应用内存1221。其中，应用的数据的存储地址可以是服务器的网卡驱动事先向网络设备200发送的，该存储地址是TCP应用根据应用内存1221的空闲情况确定后，向网卡驱动发送的地址。

S340：网络设备200从报文中分离出报文头以及应用的数据的元数据。

可以理解的，报文包括报文头、应用的数据以及元数据，步骤S320使用定界模板分离出应用的数据后，也可分离出报文头以及应用的数据的元数据，定界模板的具体描述可参考前述内容的步骤S320，这里不重复赘述。

S350：网络设备200将报文头和应用的数据的元数据写入服务器100中为网络设备200的驱动分配的内存区间中，即图1所示的网卡驱动内存1222。

具体实现中，网络设备200可通过DMA技术将报文头和应用的数据的元数据写入网卡驱动内存1222。其中，报文头和应用的数据的元数据的存储地址可以是服务器的网卡驱动事

先向网络设备200发送的，该存储地址是网卡驱动根据网卡驱动内存1222的空闲情况确定的。

应理解，步骤S340~步骤S350可以是与步骤S320~步骤S330同时发生的，也可以是先后发生的，本申请不作具体限定。

S360:服务器100的内核协议栈对报文头、应用的数据的元数据以及应用的数据进行TCP协议处理，其中，TCP协议处理用于使得报文包括一个数据流中的完整数据。

具体地，内核协议栈可以对报文头进行乱序处理、拥塞处理、重传等等，如果出现乱序，则根据报文头对应用的数据进行重新排列，如果出现数据重复，则根据报文头丢弃应用的数据中重复的数据，使得协议处理后的报文头、应用的数据以及应用的数据的元数据包括同一个TCP数据流中的全部数据，确保该TCP流中的完整元数据和完整应用的数据都已被写入服务器，提高报文处理的可靠性。

仍以图5为例，假设报文1~3同属一个TCP流，如果网络设备先接收到了报文1和报文2，还未接收到报文3时，网络设备将报文1和报文2聚合为待处理报文11，并通过TCP应用事先下发至网络设备的定界模板，将应用的数据X2从待处理报文11中分离，然后将该应用的数据X2写入应用内存1221，将剩余报文X1写入网卡驱动内存1222，内核协议栈从剩余报文Y1中获取TCP报文头，确定该TCP流中包括3个报文，而写入服务器的应用的数据X2和剩余报文Y1只包括报文1和报文2，因此报文3需要重传，内核协议栈可以向网络设备200发送重传报文3的请求，网络设备200响应于该请求，在接收到报文3时，根据定界模板将报文3中的应用的数据X2分离出来，然后将应用的数据X2写入应用内存1221，剩余报文Y2写入网卡驱动内存1222，使得服务器100获得该TCP流的全部报文。应理解，上述举例仅用于说明，并不能构成具体限定。

需要说明的，内核协议栈的数据结构通常为套接字缓存(Soket Buffer, SKB)结构，该结构可以通过挂多个指针项的形式，对多个不同地址的数据进行处理，比如处理报文1的SKB1的指针1指向网卡驱动内存1222，这样，内核协议栈可以通过指针1对网卡驱动内存中的报文头进行TCP协议处理，同时，SKB1的指针2还可以指向应用内存1221，通过指针2监控应用内存1221中的数据，确定应用的数据已写入该内存中，并在报文头出现数据重复和乱序等情况下，对应用的数据进行相应的调整，如果出现报文漏发或者报文乱序的情况，可以向网络设备发送重传请求，直至全部报文都已写入服务器。步骤S330未描述的内容可以参考前述图2实施例中的步骤2，这里不重复赘述。

S370：内核协议栈向TCP应用发送确认请求，该确认请求中携带有应用的数据的元数据在网卡驱动内存中的地址以及应用的数据在应用内存中的地址信息。

S380：TCP应用确定应用的数据是否是同一个TCP流中完整的应用的数据，如果是完整的应用的数据的情况下，执行步骤S390，如果不是完整的应用的数据的情况下执行步骤S311。

其中，TCP应用可以根据定界模板再次确认应用的数据是否完整，这里的定界模板与步骤S310处网络设备200使用的定界模板相同。举例来说，TCP应用可根据定界模板获取元数据，比如根据分隔符确定元数据，或者根据元数据的固定长度确定元数据，然后根据元数据获取应用的数据的长度，如果该长度与被写入应用内存中的应用的数据长度相同，表示网络设备200在步骤S310处拆分正确，即可执行步骤S390，相反地，如果该数据长度与被写入应用内存中的应用的数据的长度不同，表示网络设备200在步骤S310处拆分错误，即可执行步骤S311。应理解，上述TCP应用根据定界模板确定网络设备200是否对报文拆分正确的过程用于举例说明，本申请不对此进行限定。

S390：TCP应用根据元数据确定应用的数据的应用内存地址，将该应用内存地址对应的

内存页与应用的数据所在的内存页进行交换。

具体实现中，内存页交换可以是将指向应用的数据的指针与指向应用内存地址的指针进行交换，或者将应用的数据所在的内存页的虚拟地址替换成该应用内存地址，本申请不对内存页交换的具体实现方式进行限定。

可以理解的，TCP 应用根据元数据，确认应用的数据的应用内存地址，然后将该应用内存地址对应的应用内存页与应用的数据所在的内存页进行交换，从而避免了多次拷贝应用的数据造成的资源浪费，提高报文处理效率。

值得注意的，如果应用的数据的存储地址与根据元数据确定的应用内存地址相同，那么可以不执行步骤 S390，从而进一步提升报文处理效率。

S311：TCP 应用从获取应用的数据的元数据以及应用的数据，对元数据进行解析后，确定应用的数据的应用内存地址，然后将应用的数据拷贝入该应用内存地址。具体可以参考图 1 实施例中的步骤 3-步骤 4，这里不重复赘述。

在一实施例中，TCP 应用记录步骤 S380 的确认结果，也就是网络设备 200 是否拆分正确的确认结果，如果连续拆分错误的次数超过阈值，TCP 应用可以向网络设备 200 发送重新定界请求，该重新定界请求包括拆分错误的报文所属的 TCP 流的信息，网络设备 200 响应于该请求，重新对该 TCP 流中的全部报文进行聚合和拆分，从而避免了连续拆分错误的场景下，TCP 应用重复执行步骤 S370~步骤 S311 这一情况的发生，进一步提高报文处理效率，降低服务器处理报文的内存占用率。

具体实现中，TCP 应用向网络设备 200 发送重新定界请求时，可以首先向网卡驱动发送接口调用请求，网卡驱动响应于该请求，将网卡的接口提供给 TCP 应用，TCP 应用调用该接口下发同步命令（SYNC），其中，SYNC 命令用于将内存缓冲区中的数据立即强制写入网络设备 200，使得网络设备 200 执行重启定界操作，从而避免了连续报文拆分错误的场景下，TCP 应用重复执行步骤 S370~步骤 S11 这一情况的发生，进一步提高报文处理效率，降低服务器处理报文的内存占用率。

仍以图 2 所示的应用场景为例，网络设备 200 接收到了报文 1~3，假设报文 1~3 属于同一个 TCP 流，网络设备 200 对该报文 1~3 进行报文聚合，获得如图 5 所示的报文 0 之后，使用上述步骤 S310~步骤 S311 对该报文 1~3 进行处理的具体流程可以如图 6 所示。

如图 6 所示，首先，网络设备 200 根据定界模板，通过步骤 S320 和步骤 S340 将应用的数据从报文 0 中分离出来，将报文头和元数据从报文 0 中分离出来，如图 6 所示，应用的数据 1 和应用的数据 2 作从报文 0 中分离出来，报文头、元数据 1 和元数据 2 从报文 0 中分离出来。参考前述内容可知，定界模板根据元数据的格式确定，比如元数据的长度是 L1，网络设备 200 从报文头之后读取长度为 L1 的数据，获得元数据 1，然后根据元数据确定应用的数据长度 L2，再从元数据 1 之后读取长度为 L2 的数据，获得应用的数据 1，然后再从应用的数据 1 之后读取长度为 L1 的数据，获得元数据 2，根据元数据 2 确定应用的数据 2 的长度 L3 后，最后从元数据 2 之后读取长度为 L3 的数据，获得应用的数据 2。应理解，上述举例仅用于说明，并不能构成具体限定。

其次，网络设备 200 通过步骤 S330 和步骤 S350 将应用的数据 DMA 至服务器的网卡驱动内存 1222，将报文头和元数据 DMA 至服务器的应用内存 1211，其中，应用的数据 1 写入应用内存页 Y1，应用的数据 2 写入应用内存页 Y2。需要说明的，网络设备 200 在将应用的数据写入应用内存 1221 时，可以如图 6 所示，将不同的应用的数据写入不同的应用内存页，也可以将全部应用的数据写入同一个应用内存页，也就是将应用的数据 1 和应用的数据 2 都

写入应用内存页 Y1 或者应用内存页 Y2，具体可以根据 TCP 应用的处理逻辑确定，本申请不对此进行限定。

最后，服务器 100 执行步骤 S360，对网卡驱动内存 1222 中的报文头进行处理，比如乱序处理、重传等等，对失序的数据重新排列，丢弃重复的数据，确保该 TCP 流中的完整元数据和完整应用的数据都已被写入服务器之后，执行步骤 S370 向 TCP 应用发送确认请求，TCP 应用响应于该确认请求，根据定界模板执行步骤 S380 确定写入应用内存的应用的数据是同一个 TCP 流中的完整的应用数据之后，TCP 应用执行步骤 S390，根据元数据 1 确定应用的数据 1 的应用内存地址对应的应用内存页 Y3，然后将应用的数据 1 所在的应用内存页 Y1 与该应用内存页 Y3 进行交换，根据元数据 2 确定应用的数据 2 的应用内存页 Y4，将应用的数据 2 所在的应用内存页 Y2 与该应用内存页 Y4 进行交换，从而完成一次报文的处理过程。该过程无需重复拷贝应用的数据，使得服务器处理 TCP 报文的内存占用率降低，提高报文处理效率。

参考前述内容可知，为了降低服务器的处理压力，部分网络设备 200 支持 TOE 功能，该功能使得在网络设备 200 接收到报文后可先对报文进行 TCP 协议处理，然后再将处理后的完整的 TCP 流写入服务器 100。对于该类 TOE 网卡使用本申请提供的报文处理方法的具体流程可以如图 7 所示。

S410：网络设备 200 对服务器 100 中运行的应用的待处理报文进行 TCP 协议处理，比如乱序处理、重传、拥塞处理等等，使得待处理报文包括一个数据流中的完整数据，具体描述可参考前述内容中的步骤 S360，这里不重复赘述，其中，该网络设备 200 可以是支持 TOE 功能的 TOE 网卡。

具体实现中，上述服务器 100 中运行的应用的报文可以是单个子报文，比如图 5 实施例中的报文 1~报文 3，也可以是网络设备 200 对接收到的多个子报文进行报文聚合后，获得的待处理报文，比如图 5 实施例中的报文 0。值得注意的是，网络设备 200 可对接收到的所有报文进行 TCP 协议处理后，将属于同一个 TCP 流的多个子报文聚合为报文，也可以将接收到的多个子报文聚合为待处理报文后，再对待处理报文进行 TCP 协议处理，本申请不对此进行限定。其中，TCP 协议处理的具体描述可以参考前述内容中的步骤 2、步骤 S330 等等，这里不重复赘述。

S420：网络设备 200 从报文中分离出应用的数据。

具体实现中，网络设备 200 可根据定界模板从报文中分离出应用的数据，其中，定界模板是服务器 100 中的 TCP 应用在步骤 S410 之前，通过调用网卡驱动的接口向网络设备 200 下发的模板，定界模板的描述可以参考前述内容中的步骤 S310，这里不重复赘述。

应理解，由于网络设备 200 支持 TOE 功能，在步骤 S420 已对报文进行了 TCP 协议处理，因此报文不会存在失序、重复等情况，使用定界模板对从报文中分离出的应用的数据，不会出现拆分错误的情况。

S430：网络设备 200 将应用的数据写入服务器中为应用分配的内存区间中，具体可以是应用内存 1221，该步骤的描述可以参考前述内容的步骤 S330，这里不重复赘述。具体实现中，网络设备 200 可通过 DMA 技术将应用的数据写入应用内存 1221。

S440：从报文中分离出报文头以及应用的数据的元数据。该步骤的具体描述可参考前述内容的步骤 S340，这里不重复赘述。

S450：将报文头以及应用的数据的元数据写入服务器为网络设备的驱动分配的内存区间

中个，具体可以是网卡驱动内存 1222。具体实现中，网络设备 200 可通过 DMA 技术将报文头以及应用的数据的元数据写入网卡驱动内存 1222。该步骤的描述可参考前述内容的步骤 S350，这里不重复赘述。

S460：内核协议栈向 TCP 应用发送交换请求，其中，该交换请求包括分离得到的应用的数据的地址以及元数据的地址。具体可参考前述内容中的步骤 S370，这里不重复赘述。

S470：TCP 应用根据元数据确定应用的数据的应用内存地址，将应用的数据所在的内存页和该应用内存地址对应的内存页交换。具体地，TCP 应用可以根据交换请求中元数据的地址获取元数据，然后根据元数据确定应用的数据的应用内存地址，其中，根据元数据确定应用的数据的应用内存地址以及内存页交换的具体描述可以参考签署内容中的步骤 S390，这里不重复赘述。

应理解，如果该应用内存地址与分离出的应用的数据所在的内存地址相同，那么可以不再进行内存页交换，进一步提升报文处理的效率。

可选地，TCP 应用可在根据元数据确定应用的数据的应用内存地址之前，通过定界模板二次确认网络设备 200 是否拆分正确，在拆分正确的情况下，再根据元数据确定应用的数据的应用内存地址，将应用的数据所在的内存页和该应用内存地址对应的内存页交换，从而避免由于其他原因比如 DMA 出错导致应用的数据不完整，内存页交换后 TCP 应用得到错误数据情况的发生，进一步提高本申请提供的报文处理方法的可靠性。

综上可知，本申请提供的报文处理方法，服务器提前向网络设备下发定界模板，使得网络设备在向服务器发送报文之前，先通过定界模板从报文中拆分出应用的数据和元数据，并将元数据写入网卡驱动内存，将应用的数据写入应用内存，使得服务器的 TCP 应用根据网卡驱动内存中的元数据确定应用的数据的应用内存地址后，将应用的数据所在的内存页与该应用内存地址对应的内存页交换即可完成一次数据通信的过程，该方法避免了报文由网络设备传输至 TCP 应用内存页的过程中多次拷贝情况的发生，提高报文处理效率，降低内存占用率。

上述详细阐述了本申请实施例的方法，为了便于更好的实施本申请实施例上述方案，相应地，下面还提供用于配合实施上述方案的相关设备。

图 8 是本申请提供的一种网络设备 200 的结构示意图，该网络设备 200 应用于如图 1 所示的报文处理系统，其中，网络设备 200 与服务器 100 连接，如图 8 所示，网络设备 200 可包括接收单元 810、分离单元 820 以及写入单元 830。

接收单元 810 用于接收服务器中运行的应用的报文，具体实现方式，请参考上述图 3 所示实施例中步骤 S310 以及上述图 4 实施例中步骤 S410 的详细描述，这里不重复赘述；

分离单元 820 用于从报文中分离出应用的数据，具体实现方式，可参考上述图 3 实施例中步骤 S320 以及图 4 实施例中步骤 S420 的详细描述，这里不重复赘述；

写入单元 830 用于将应用的数据写入服务器中为应用分配的内存区间中，具体实现方式，可参考上述图 3 实施例的步骤 S330 以及图 4 实施例的步骤 S430 的详细描述，这里不重复赘述。

在一实施例中，分离单元 820 还用于从报文中分离出报文头及应用的数据的元数据；写入单元还用于将报文头和应用的数据的元数据存储至服务器为网络设备的驱动分配的内存区间中，具体实现方式，可参考上述图 3 实施例中步骤 S340~步骤 S350，以及图 4 实施例中步骤 S440~步骤 S450 的详细描述，这里不重复赘述。

在一实施例中，分离单元 820 用于根据定界模板从报文中分离出应用的数据，定界模板

定义了应用的数据与报文中其他数据的分离规则，具体实现方式，可参考上述图 3 实施例中的步骤 S310，以及图 4 实施例中的步骤 S420 中关于网络设备 200 使用定界模板将应用的数据从报文中分离这一步骤的详细描述，这里不重复赘述。

在一实施例中，接收单元 810 用于将属于同一个数据流的多个子报文聚合为报文，其中，属于同一个数据流的多个子报文的源网际互联协议 IP 地址和目的 IP 地址相同，具体实现方式，可参考上述图 3 实施例中的步骤 S310 以及图 4 实施例中的步骤 S410 关于报文聚合步骤的详细描述，这里不重复赘述。

在一实施例中，当网络设备 200 是支持 TOE 功能的网卡时，网络设备 200 还包括确定单元 840，确定单元 840 用于在分离单元从报文中分离出应用的数据之前，确定报文包括一个数据流中的完整数据。具体可以对报文进行 TCP 协议处理，比如拥塞控制、重传、乱序处理等等，具体实现方式，可参考上述图 4 实施例中的步骤 S410，这里不重复赘述。

应理解，图 8 所示的网络设备 200 的内部的单元模块也可以有多种划分，各个模块可以是软件模块，也可以是硬件模块，也可以是部分软件模块部分硬件模块，本申请不对其进行限制，图 8 是一种示例性的划分方式，本申请不作具体限定。

可以理解的，本申请提供的网络设备可提前接收由服务器下发的定界模板，使得网络设备在向服务器发送报文之前，先通过定界模板从报文中拆分出应用的数据和元数据，并将元数据写入网卡驱动内存，将应用的数据写入应用内存，使得服务器的 TCP 应用根据网卡驱动内存中的元数据确定应用的数据的应用内存地址后，将应用的数据所在的内存页与该应用内存地址对应的内存页交换即可完成一次数据通信的过程，该方法避免了报文由网络设备传输至 TCP 应用内存页的过程中多次拷贝情况的发生，提高报文处理效率，降低内存占用率。

图 9 是本申请提供的一种网络设备 200 的硬件结构示意图，如图 9 所示，网络设备 200 包括：处理器 910、通信接口 920 以及存储器 930。其中，处理器 910、通信接口 920 以及存储器 930 可以通过内部总线 940 相互连接，也可通过无线传输等其他手段实现通信。本申请实施例以通过总线 940 连接为例，总线 940 可以是外设部件互连标准（Peripheral Component Interconnect, PCI）总线或扩展工业标准结构（Extended Industry Standard Architecture, EISA）总线等。总线 940 可以分为地址总线、数据总线、控制总线等。为便于表示，图 9 中仅用一条粗线表示，但并不表示仅有一根总线或一种类型的总线。

处理器 910 可以由至少一个通用处理器构成，例如中央处理器（Central Processing Unit，CPU），或者 CPU 和硬件芯片的组合。上述硬件芯片可以是专用集成电路（Application-Specific Integrated Circuit，ASIC）、可编程逻辑器件（Programmable Logic Device，PLD）或其组合。上述 PLD 可以是复杂可编程逻辑器件（Complex Programmable Logic Device，CPLD）、现场可编程逻辑门阵列（Field-Programmable Gate Array，FPGA）、通用阵列逻辑（Generic Array Logic，GAL）或其任意组合。处理器 910 执行各种类型的数字存储指令，例如存储在存储器 930 中的软件或者固件程序，它能使网络设备 200 提供较宽的多种服务。

存储器 930 用于存储程序代码，并由处理器 910 来控制执行，以执行上述图 1-图 7 中任一实施例中网络设备 200 的处理步骤。程序代码中可以包括一个或多个软件模块。这一个或多个软件模块可以为图 8 所示实施例中提供的软件模块，如接收单元，分离单元以及写入单元，其中，接收单元，可用于接收服务器中运行的应用的报文；分离单元，可用于从报文中分离出应用的数据；写入单元，用于可将应用的数据写入服务器中为应用分配的内存区间中，具体可用于执行前述方法的步骤 S310-步骤 S350、步骤 S410-步骤 S450 及其可选步骤，还可

以用于执行图 2-图 7 实施例描述的其他由网络设备 200 执行的步骤，这里不再进行赘述。

存储器 930 可以包括易失性存储器（Volatile Memory），例如随机存取存储器（Random Access Memory，RAM）；存储器 1030 也可以包括非易失性存储器（Non-Volatile Memory），例如只读存储器（Read-Only Memory，ROM）、快闪存储器（Flash Memory）、硬盘（Hard Disk Drive，HDD）或固态硬盘（Solid-State Drive，SSD）；存储器 930 还可以包括上述种类的组合。存储器 930 可以存储有程序代码，具体可以包括用于供处理器 910 执行图 2-图 7 实施例描述的其他步骤的程序代码，这里不再进行赘述。

通信接口 920 可以为有线接口（例如以太网接口），可以为内部接口（例如高速串行计算机扩展总线（Peripheral Component Interconnect express, PCIe）总线接口）、有线接口（例如以太网接口）或无线接口（例如蜂窝网络接口或使用无线局域网接口），用于与其他服务器或模块进行通信，具体实现中，通信接口 920 可用于接收报文，以供处理器 910 对该报文进行处理。

需要说明的，图 9 仅仅是本申请实施例的一种可能的实现方式，实际应用中，网络设备还可以包括更多或更少的部件，这里不作限制。关于本申请实施例中未示出或未描述的内容，可参见前述图 2-图 7 实施例中的相关阐述，这里不再赘述。

本申请实施例还提供一种计算机可读存储介质，计算机可读存储介质中存储有指令，当其在处理器上运行时，图 2-图 7 所示的方法流程得以实现。

本申请实施例还提供一种计算机程序产品，当计算机程序产品在处理器上运行时，图 2-图 7 所示的方法流程得以实现。

上述实施例，可以全部或部分地通过软件、硬件、固件或其他任意组合来实现。当使用软件实现时，上述实施例可以全部或部分地以计算机程序产品的形式实现。计算机程序产品包括至少一个计算机指令。在计算机上加载或执行计算机程序指令时，全部或部分地产生按照本发明实施例的流程或功能。计算机可以为通用计算机、专用计算机、计算机网络、或者其他可编程装置。计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输，例如，计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线（例如同轴电缆、光纤、数字用户线（Digital Subscriber Line, DSL））或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、服务器或数据中心进行传输。计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含至少一个可用介质集合的服务器、数据中心等数据存储设备。可用介质可以是磁性介质（例如，软盘、硬盘、磁带）、光介质（例如，高密度数字视频光盘（Digital Video Disc, DVD）、或者半导体介质。半导体介质可以是 SSD。

以上，仅为本发明的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到各种等效的修改或替换，这些修改或替换都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应以权利要求的保护范围为准。

权 利 要 求 书

1、一种报文处理方法，其特征在于，所述方法应用于网络设备，所述网络设备连接至服务器，所述方法包括：

接收服务器中运行的应用的报文；

从所述报文中分离出所述应用的数据；

将所述应用的数据写入所述服务器中为所述应用分配的内存区间中。

2、根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

从所述报文中分离出报文头及所述应用的数据的元数据；

将所述报文头和所述应用的数据的元数据存储至所述服务器为所述网络设备的驱动分配的内存区间中。

3、根据权利要求 1 或 2 所述的方法，其特征在于，

所述从所述报文中分离出所述应用的数据，包括：

根据定界模板从所述报文中分离出所述应用的数据，所述定界模板定义了所述应用的数据与报文中其他数据的分离规则。

4、根据权利要求 1 至 3 任一权利要求所述的方法，其特征在于，

所述接收服务器中运行的应用的报文包括：将属于同一个数据流的多个子报文聚合为所述报文，其中，所述属于同一个数据流的多个子报文的源网际互联协议 IP 地址和目的 IP 地址相同。

5、根据权利要求 1 至 4 任一权利要求所述的方法，其特征在于，从所述报文中分离出所述应用的数据之前，所述方法还包括：

确定所述报文包括一个数据流中的完整数据。

6、一种网络设备，其特征在于，所述网络设备连接至服务器，所述网络设备包括：

接收单元，用于接收服务器中运行的应用的报文；

分离单元，用于从所述报文中分离出所述应用的数据；

写入单元，用于将所述应用的数据写入所述服务器中为所述应用分配的内存区间中。

7、根据权利要求 6 所述的网络设备，其特征在于，

所述分离单元还用于从所述报文中分离出报文头及所述应用的数据的元数据；

所述写入单元还用于将所述报文头和所述应用的数据的元数据存储至所述服务器为所述网络设备的驱动分配的内存区间中。

8、根据权利要求 6 或 7 所述的网络设备，其特征在于，

所述分离单元用于根据定界模板从所述报文中分离出所述应用的数据，所述定界模板定义了所述应用的数据与报文中其他数据的分离规则。

9、根据权利要求 6 至 8 任一权利要求所述的网络设备，其特征在于，

所述接收单元用于将属于同一个数据流的多个子报文聚合为所述报文，其中，所述属于同一个数据流的多个子报文的源网际互联协议 IP 地址和目的 IP 地址相同。

10、根据权利要求 6 至 9 任一权利要求所述的网络设备，其特征在于，所述网络设备还包括确定单元，所述确定单元用于在所述分离单元从所述报文中分离出所述应用的数据之前，确定所述报文包括一个数据流中的完整数据。

11、一种网络设备，其特征在于，包括处理器和通信接口，所述通信接口用于接收报文，所述处理器用于执行权利要求 1 至 5 任一权利要求所述的方法以对报文进行处理。

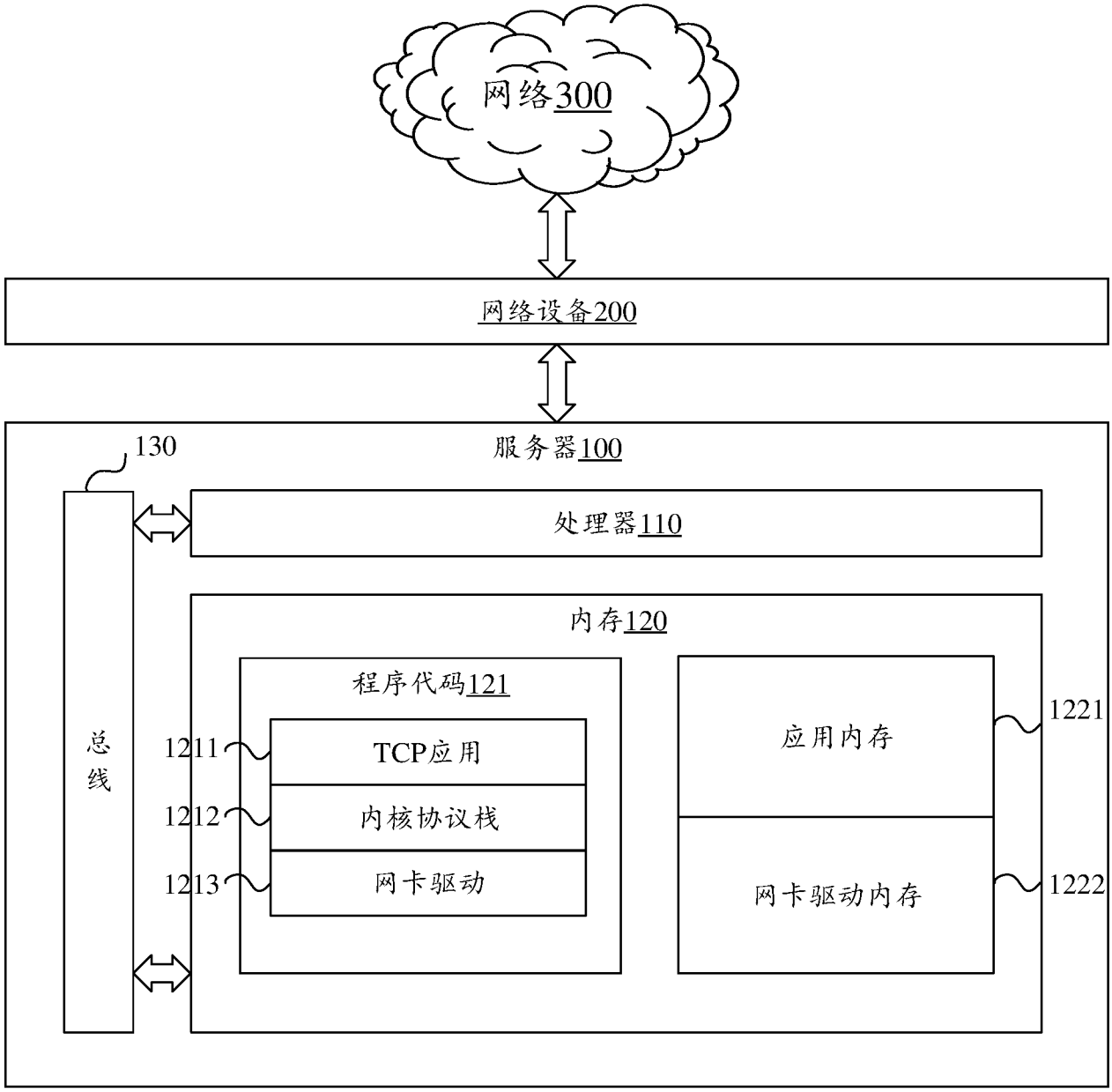
网络300
网络设备200
服务器100
130
处理器110
内存120
程序代码121
1211
TCP应用
1212
内核协议栈
1213
网卡驱动
总线
应用内存
1221
网卡驱动内存
1222

图 1

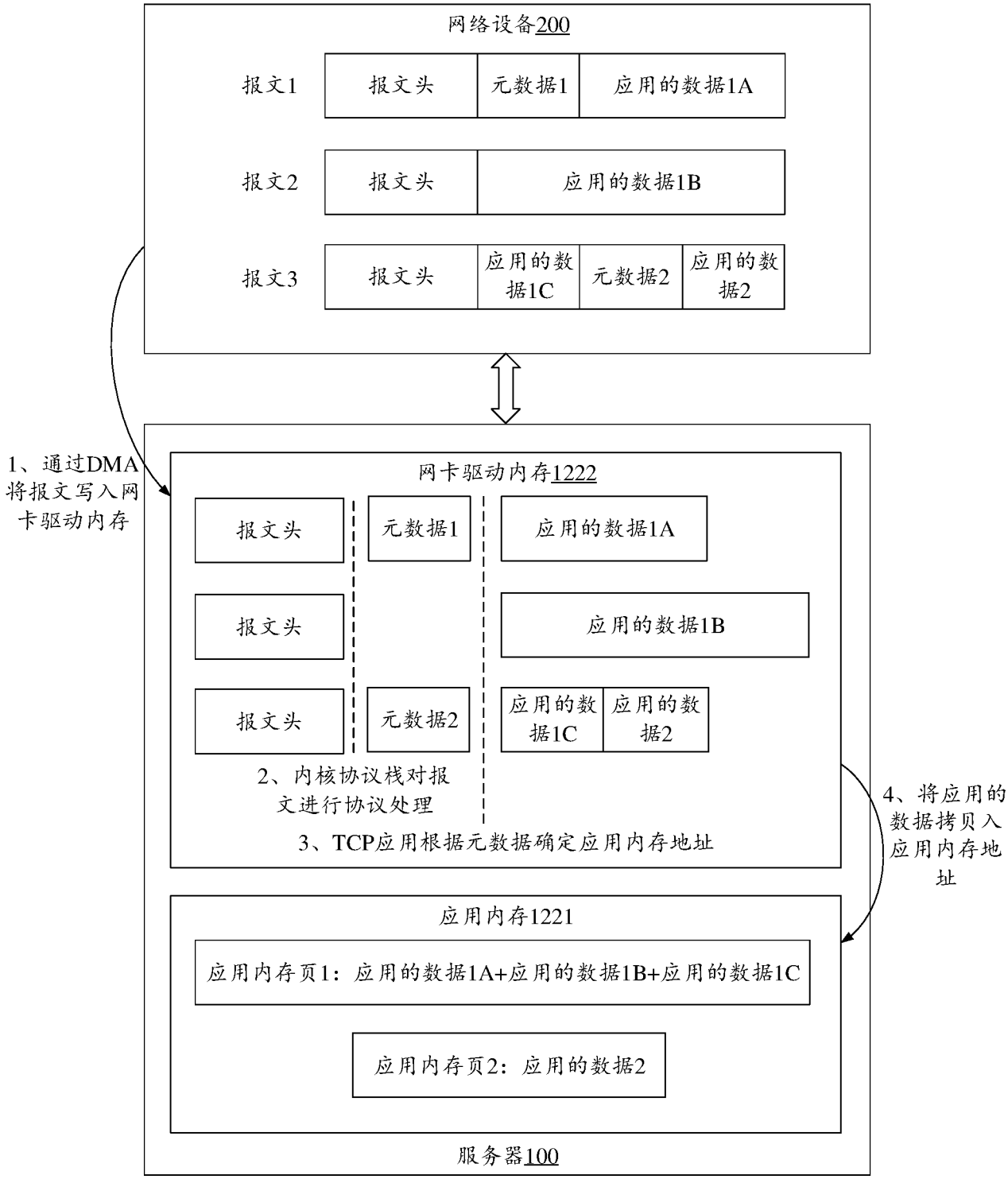
网络设备200
报文1
报文头
元数据1
应用的数据1A
报文2
报文头
应用的数据1B
报文3
报文头
应用的数据1C
元数据2
应用的数据2
1、通过DMA将报文写入网卡驱动内存
网卡驱动内存1222
报文头
元数据1
应用的数据1A
报文头
应用的数据1B
报文头
元数据2
应用的数据1C
应用的数据2
2、内核协议栈对报文进行协议处理
3、TCP应用根据元数据确定应用内存地址
4、将应用的数据拷贝入应用内存地址
应用内存1221
应用内存页1：应用的数据1A+应用的数据1B+应用的数据1C
应用内存页2：应用的数据2
服务器100

图 2

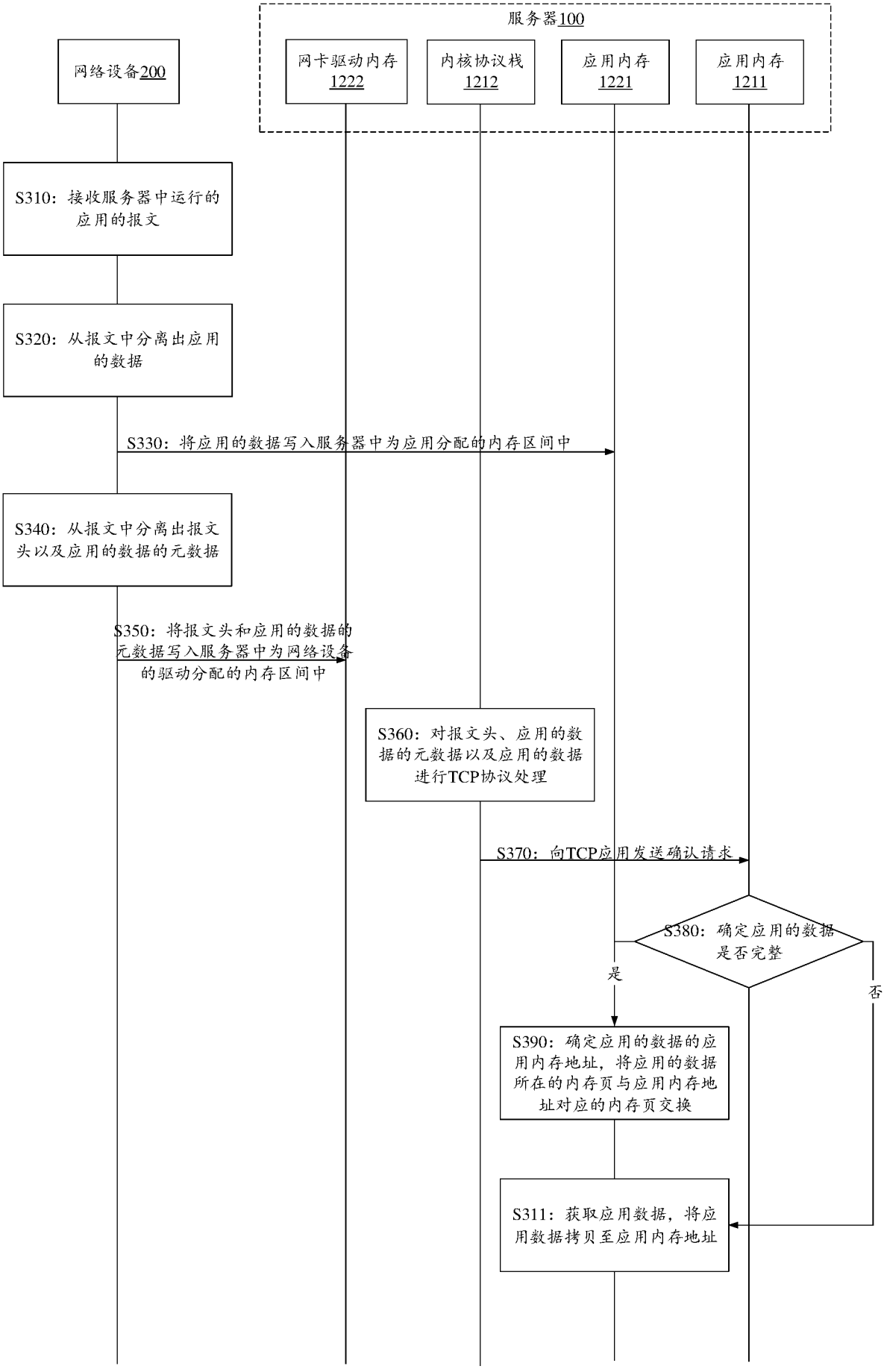
服务器100
网络设备200
网卡驱动内存 1222
内核协议栈 1212
应用内存 1221
应用内存 1211
S310：接收服务器中运行的应用的报文
S320：从报文中分离出应用的数据
S330：将应用的数据写入服务器中为应用分配的内存区间中
S340：从报文中分离出报文头以及应用的数据的元数据
S350：将报文头和应用的数据的元数据写入服务器中为网络设备的驱动分配的内存区间中
S360：对报文头、应用的数据的元数据以及应用的数据进行TCP协议处理
S370：向TCP应用发送确认请求
S380：确定应用的数据是否完整
是
否
S390：确定应用的数据的应用内存地址，将应用的数据所在的内存页与应用内存地址对应的内存页交换
S311：获取应用数据，将应用数据拷贝至应用内存地址

图 3

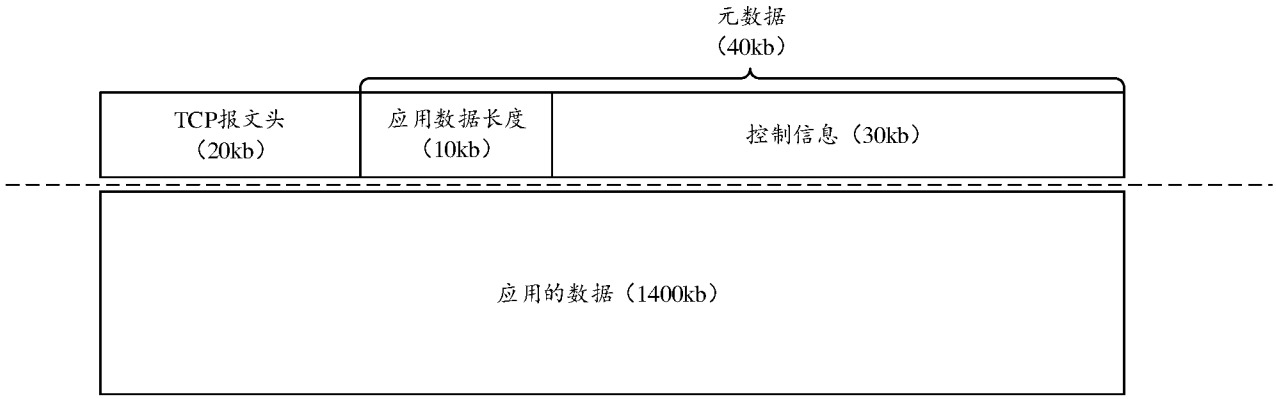
元数据
（40kb）
TCP报文头
（20kb）
应用数据长度
（10kb）
控制信息（30kb）
应用的数据（1400kb）

图 4

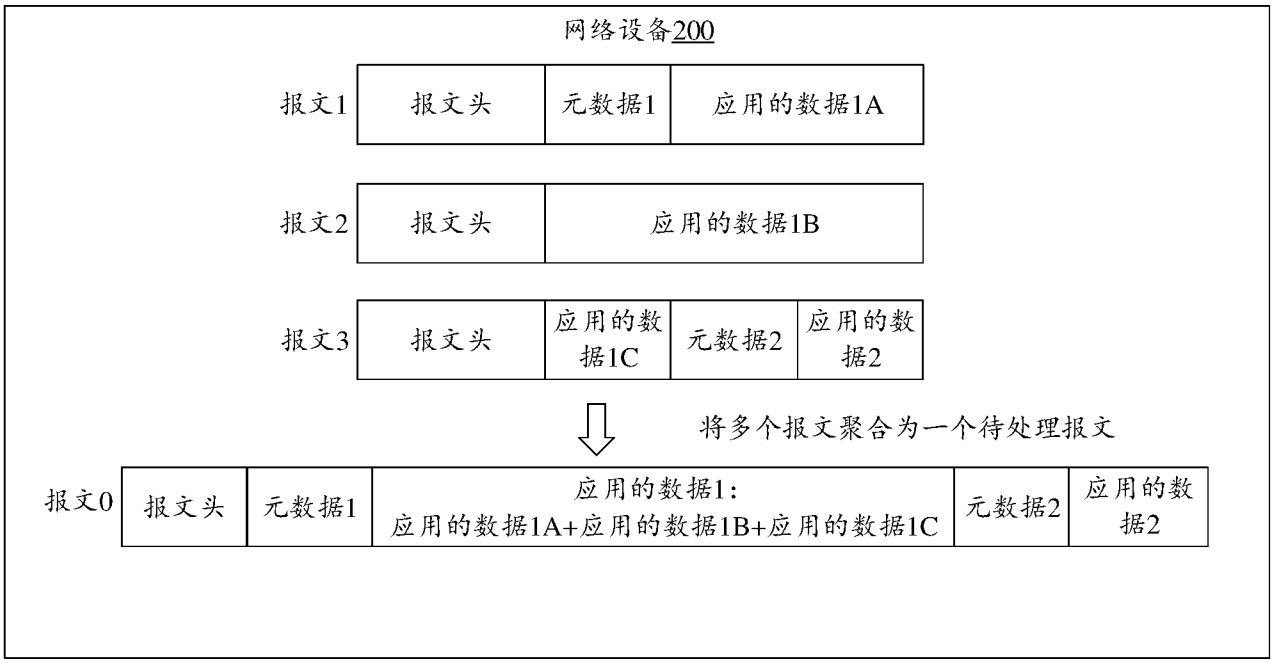
网络设备200
报文1
报文头
元数据1
应用的数据1A
报文2
报文头
应用的数据1B
报文3
报文头
应用的数据1C
元数据2
应用的数据2
将多个报文聚合为一个待处理报文
报文0
报文头
元数据1
应用的数据1：
应用的数据1A+应用的数据1B+应用的数据1C
元数据2
应用的数据2

图 5

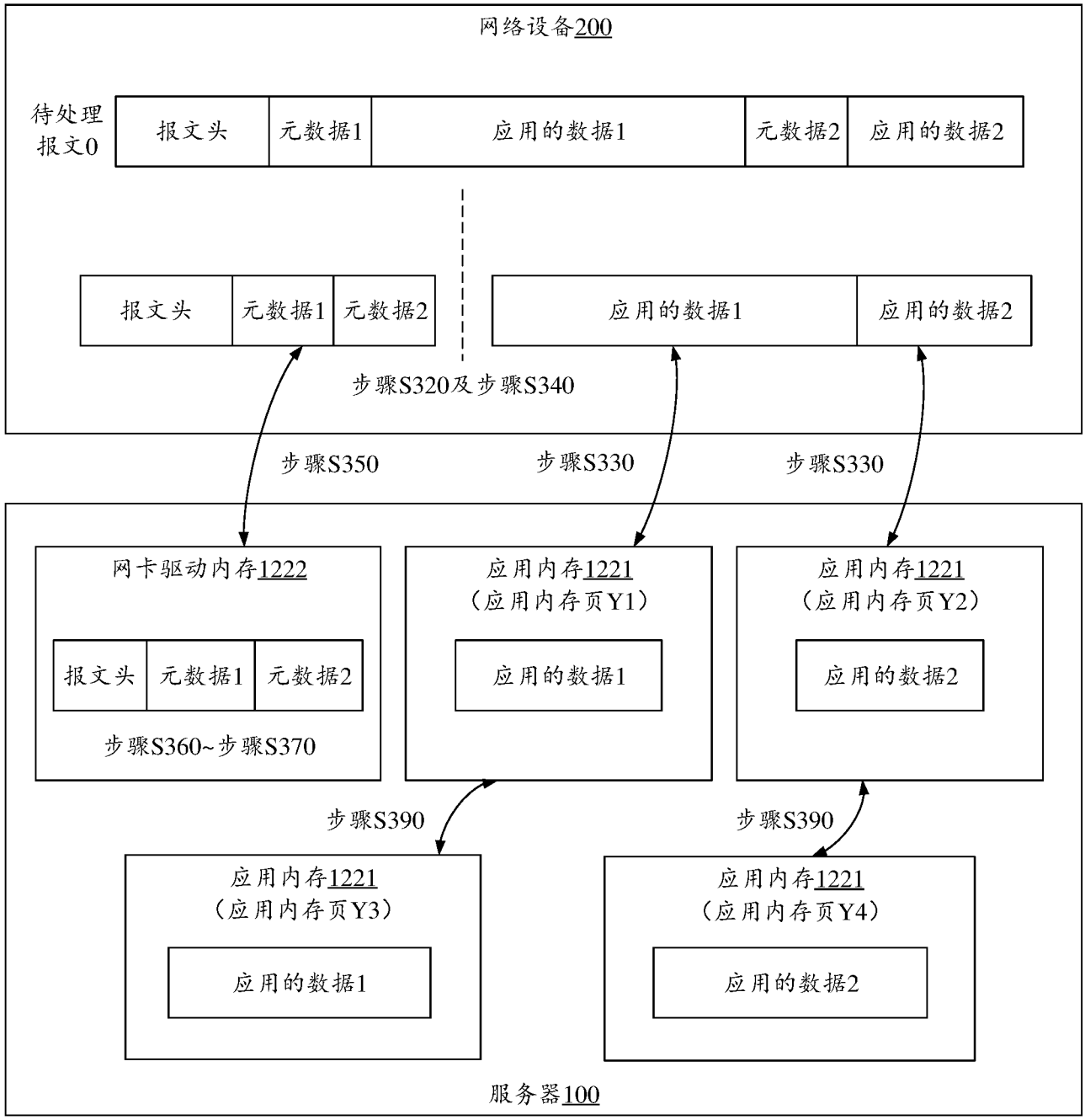
网络设备200
待处理报文0
报文头
元数据1
应用的数据1
元数据2
应用的数据2
报文头
元数据1
元数据2
应用的数据1
应用的数据2
步骤S320及步骤S340
步骤S350
步骤S330
步骤S330
网卡驱动内存1222
报文头
元数据1
元数据2
步骤S360~步骤S370
应用内存1221
（应用内存页Y1）
应用的数据1
应用内存1221
（应用内存页Y2）
应用的数据2
步骤S390
步骤S390
应用内存1221
（应用内存页Y3）
应用的数据1
应用内存1221
（应用内存页Y4）
应用的数据2
服务器100

图 6

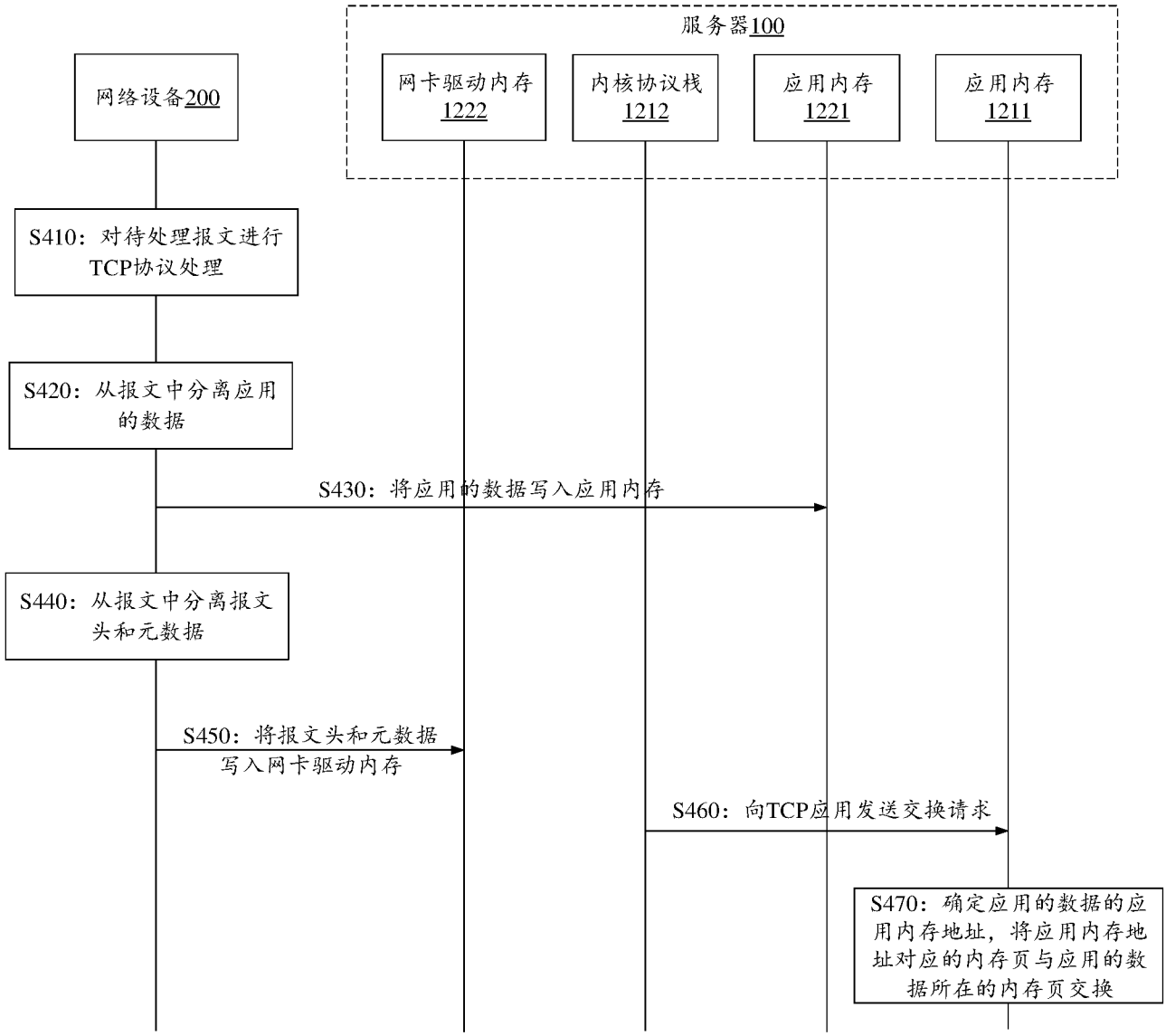
服务器100
网络设备200
网卡驱动内存 1222
内核协议栈 1212
应用内存 1221
应用内存 1211
S410：对待处理报文进行TCP协议处理
S420：从报文中分离应用的数据
S430：将应用的数据写入应用内存
S440：从报文中分离报文头和元数据
S450：将报文头和元数据写入网卡驱动内存
S460：向TCP应用发送交换请求
S470：确定应用的数据的应用内存地址，将应用内存地址对应的内存页与应用的数据所在的内存页交换

图 7

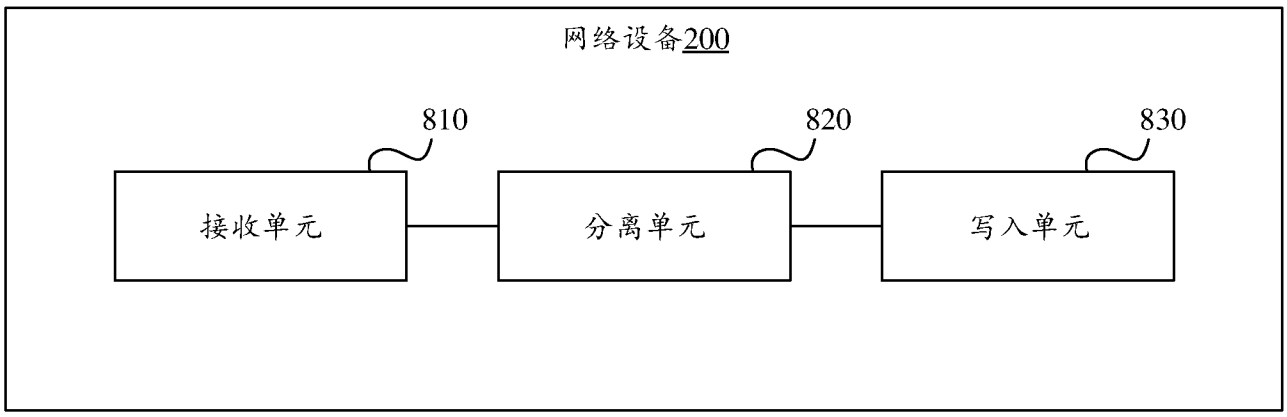
网络设备200
810
820
830
接收单元
分离单元
写入单元

图 8

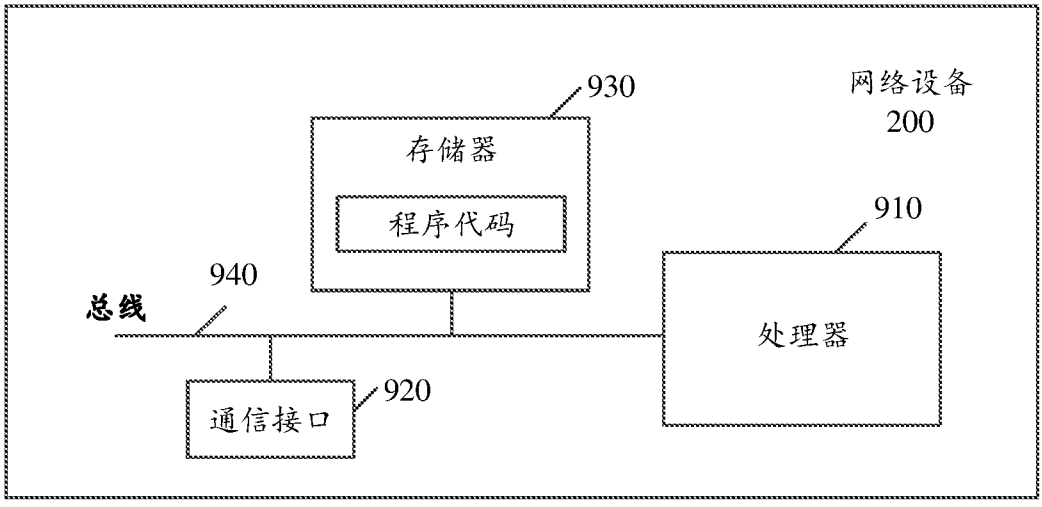

图 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2021/107828

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/911(2013.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L; H04W; G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, WPI, EPODOC, CNKI: 报文, 服务器, TCP, 应用, 内存, 缓存, 缓冲, 存储, 储存, 网卡, 网络适配, 网络接口卡, 驱动, 拷贝, 复制, 原始数据, 负载, Payload, NIC, TOE, 直接内存存取, Direct Memory Access, DMA, copy, 元数据, metedata, 报头, 报文头, 头部, 内核, server, APP+, cache, memory, buffer, storage, adapter, driver, header

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 107547623 A (H3C CLOUD COMPUTER TECHNOLOGY CO., LTD.) 05 January 2018 (2018-01-05) description, paragraphs [0031]-[0047] and figures 1-4	1-11
A	CN 104506379 A (RUN TECHNOLOGIES CO., LTD. BEIJING) 08 April 2015 (2015-04-08) entire document	1-11
A	CN 101616194 A (UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA) 30 December 2009 (2009-12-30) entire document	1-11
A	US 9952979 B1 (CAVIUM, INC.) 24 April 2018 (2018-04-24) entire document	1-11
A	US 2011258337 A1 (ZTE CORPORATION) 20 October 2011 (2011-10-20) entire document	1-11
A	US 2018219805 A1 (JUNIPER NETWORKS, INC.) 02 August 2018 (2018-08-02) entire document	1-11

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 October 2021	**27 October 2021**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088** **China**	
Facsimile No. **(86-10)62019451**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2021/107828

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 107547623 A	05 January 2018	None	
CN 104506379 A	08 April 2015	None	
CN 101616194 A	30 December 2009	None	
US 9952979 B1	24 April 2018	None	
US 2011258337 A1	20 October 2011	WO 2010015142 A1	11 February 2010
		EP 2312807 A1	20 April 2011
		PL 2312807 T3	30 April 2019
		RU 2011107517 A	10 September 2012
		CN 101340574 A	07 January 2009
US 2018219805 A1	02 August 2018	EP 3355526 A1	01 August 2018
		CN 108377213 A	07 August 2018

国际检索报告

国际申请号
PCT/CN2021/107828

A. 主题的分类

H04L 12/911(2013.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

H04L; H04W; G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

CNPAT,WPI,EPODOC,CNKI: 报文，服务器，TCP，应用，内存，缓存，缓冲，存储，储存，网卡，网络适配，网络接口卡，驱动，拷贝，复制，原始数据，负载，Payload，NIC，TOE，直接内存存取，Direct Memory Access，DMA，copy，元数据，metedata，报头，报文头，头部，内核，server，APP+, cache, memory, buffer, storage, adapter, driver, header

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
X	CN 107547623 A (新华三云计算技术有限公司) 2018年 1月 5日 (2018-01-05) 说明书第[0031]-[0047]段及图1-4	1-11
A	CN 104506379 A (北京锐安科技有限公司) 2015年 4月 8日 (2015-04-08) 全文	1-11
A	CN 101616194 A (中国科学技术大学) 2009年 12月 30日 (2009-12-30) 全文	1-11
A	US 9952979 B1 (CAVIUM, INC.) 2018年 4月 24日 (2018-04-24) 全文	1-11
A	US 2011258337 A1 (ZTE CORPORATION) 2011年 10月 20日 (2011-10-20) 全文	1-11
A	US 2018219805 A1 (JUNIPER NETWORKS, INC.) 2018年 8月 2日 (2018-08-02) 全文	1-11

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2021年 10月 19日	2021年 10月 27日
ISA/CN的名称和邮寄地址 中国国家知识产权局(ISA/CN) 中国 北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 罗啸 电话号码 86-(10)-53961774

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2021/107828

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	107547623	A	2018年 1月 5日	无			
CN	104506379	A	2015年 4月 8日	无			
CN	101616194	A	2009年 12月 30日	无			
US	9952979	B1	2018年 4月 24日	无			
US	2011258337	A1	2011年 10月 20日	WO	2010015142	A1	2010年 2月 11日
				EP	2312807	A1	2011年 4月 20日
				PL	2312807	T3	2019年 4月 30日
				RU	2011107517	A	2012年 9月 10日
				CN	101340574	A	2009年 1月 7日
US	2018219805	A1	2018年 8月 2日	EP	3355526	A1	2018年 8月 1日
				CN	108377213	A	2018年 8月 7日