

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 29/06 (2006.01)

H04L 12/56 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510083698.9

[43] 公开日 2006年11月15日

[11] 公开号 CN 1863197A

[22] 申请日 2005.6.30

[21] 申请号 200510083698.9

[30] 优先权

[32] 2005.5.10 [33] US [31] 11/126,464

[71] 申请人 美国博通公司

地址 美国加州

[72] 发明人 吉汉·卡若古 尤里·伊莱泽

詹姆士·D·贝内特

[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

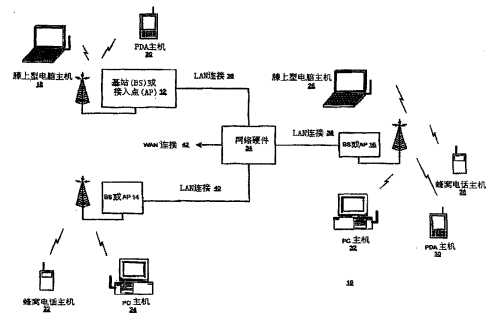
权利要求书 3 页 说明书 42 页 附图 13 页

[54] 发明名称

通过直接数据布置处理无线系统内乱序分段的方法及系统

[57] 摘要

一种处理无线系统内乱序分段的方法及系统包括下列至少之一：将无线网络处理器接收的第一 TCP 分段放置于主机缓冲区内，在 TCP 序列数量与相应缓冲区地址间具有一个映射。确定无线网络处理器接收的第二 TCP 分段是否为有序 TCP 分段或乱序 TCP 分段中之一。如果所述第二接收的 TCP 分段为乱序 TCP 分段，则与至少所述第二 TCP 分段相关的控制信息被存储于所述无线网络处理器上。所述乱序 TCP 分段被放置于所述主机缓冲区的一部分内。



1、一种处理无线系统内乱序TCP分段的方法，其特征在于，包括：

下列至少之一：将无线网络处理器接收的第一TCP分段放置于主机缓冲区内，映射所述第一TCP分段的TCP序列数量及相应缓冲区地址，确定第二接收的TCP分段是否为有序TCP分段或乱序TCP分段中之一；及

如果所述无线网络处理器接收的所述第二TCP分段为乱序TCP分段，则：

将与至少所述第二TCP分段相关的控制信息存储于所述无线网络处理器上；及

将所述乱序TCP分段放置于所述主机缓冲区的一部分内。

2、根据权利要求1所述的方法，其特征在于，进一步包括：记录所述无线网络处理器上被所述第一接收的TCP分段的结尾占用的至少一个缓冲区地址，所述缓冲区为TCP缓冲区、ULP缓冲区及应用缓冲区之一。

3、根据权利要求2所述的方法，其特征在于，进一步包括：如果所述第二接收的TCP分段为有序TCP分段，则将所述有序TCP分段相邻于所述第一接收的TCP分段放置于所述主机缓冲区内。

4、根据权利要求3所述的方法，其特征在于，进一步包括：将所述有序TCP分段从与所述第一接收的TCP分段的所述结尾占用的所述被记录的地址相邻的地址开始放置于所述主机缓冲区内。

5、一种处理无线系统内乱序TCP分段的系统，其特征在于，包括：

将无线网络处理器接收的第一TCP分段放置于主机缓冲区内，在TCP序列数量与相应缓冲区地址间具有一个映射，确定无线网络处理器接收的第二TCP分段是否为有序TCP分段或乱序TCP分段中之一；及

如果所述第二接收的TCP分段为乱序TCP分段，则：

所述至少一个处理器将与至少所述第二TCP分段相关的控制信息存储于所述无线网络处理器上；及

所述至少一个处理器将所述乱序TCP分段放置于所述主机缓冲区的一部分内。

6、根据权利要求5所述的系统，其特征在于，所述至少一个处理器记录所述无线网络处理器上被所述第一接收的TCP分段的结尾占用的至少一个缓冲区地址，所述缓冲区为TCP缓冲区、ULP缓冲区及应用缓冲区之一。

7、根据权利要求6所述的系统，其特征在于，如果所述第二接收的TCP分段为有序TCP分段，则所述至少一个处理器将所述有序TCP分段相邻于所述第一接收的TCP分段放置于所述主机缓冲区内。

8、根据权利要求7所述的系统，其特征在于，所述至少一个处理器将所述有序TCP分段从与所述第一接收的TCP分段的所述结尾占用的所述被记录的地址相邻的地址开始放置于所述主机缓冲区内。

9、根据权利要求8所述的系统，其特征在于，所述至少一个处理器执行下列至少之一：

通过所述无线网络处理器记录下列数据指示至少之一：

所述有序TCP分段的结尾占用的结尾缓冲区地址；

开始缓冲区地址加上所述有序TCP分段的数据部分的长度；及

跟随所述有序TCP分段的结尾的一个字节的缓冲区地址；及

在所述无线网络处理器上存储与至少一个所述指示数据相关的控制信息。

10、根据权利要求5所述的系统，其特征在于，所述至少一个处理器执行下列至少之一：

在所述无线网络处理器上记录至少一个数据指示：

所述放置的乱序TCP分段的开始所占用的地址，及所述放置的乱序TCP

分段的结尾所占用的地址；及

所述的加上所述放置的乱序TCP分段占用的长度的开始缓冲区地址；及
在所述无线网络处理器上存储与所述网络处理器上的所述指示数据相关的
控制信息。

通过直接数据布置处理无线系统内乱序分段的方法及系统

技术领域

本发明的某些实施例涉及通信过程中的信息分段。更具体地，本发明的某些实施例涉及通过直接数据布置处理无线系统内乱序分段的方法及系统。

背景技术

IEEE 802.11标准提供用于提供多种功能的公共媒介访问控制（medium access control，简称MAC）层，其支持基于无线局域网（WLAN）的802.11。该MAC层适用于在一个共享无线通信信道上推动及管理接入点（AP）及站点（STA）间的通信。该MAC层适用于处理多种功能，如扫描、验证、联合、节能及存储碎片。由802.11 MAC提供的可选功能包括加密及RTS/CTS信息交换。

该802.11标准包括被动扫描模式及主动扫描模式。在被动扫描模式中，一个如无线收发机或NIC这样的无线站点通过在信道或一连串信道上监听接入点而搜索服务。被动地扫描的无线站点没有传输。802.11标准内，被动地扫描被定义为强制性扫描模式，主动扫描被定义为可选扫描模式。在主动扫描模式中，每一个无线收发机或NIC会发送试图引出探测反应帧的探测帧，以扫描单独的信道来定位接入点。试探性结合的最佳接入点决定于无线站点接收的来自每个接入点的信号的物理性能，与在扫描操作期间所接收的各种其它信号相结合，这些信号如接入点支持的速率、QOS性能、当前负荷因数及其它特性的支持。一个接入点可以周期性地发送一个信标帧，该信标帧可以在扫描期间被无线收发机或STA接收器接收。该信标帧包括相应AP的信号强度信息，同如服务集标识符（Service Set

Identifier, SSID) 这样的接入点特别信息, 及由接入点所支持的数据率。该无线 STA 基于从一个或多个接入点接收的接入点特别信息可以确定将要连接哪一个接入点。在该可选主动扫描模式期间, 无线 STA 可以发送一个探测帧, 及所有接收该探测帧的接入点可以响应于其相应的特别信息, 该特别信息如 SSID、信号强度及支持的数据率。主动扫描允许无线 STA 从一个或多个接入点平均地接收较快的响应, 而无需等待信标帧频的传输。由于探测帧被发送及来自响应 AP 的响应帧被接收, 主动扫描的一个缺点是其强加了额外的网络消耗。另外, 该执行主动扫描的无线 STA 可干扰网络的正常通信, 因为该扫描 STA 已切换至具有很少关于当前信道状态的信息的当前信道。

该 802.11 标准提供了开放系统验证方法及共享密钥验证方法, 以对如无线 STA 这样的网络实体的身份进行验证。在 802.11 标准中, 开放系统验证方法被指定为强制性的, 而共享密钥验证方法被指定为可选的。通过开放系统验证, 无线 STA 可通过发送一个验证请求至一个接入点而开始验证。在响应方面, 该接入点可以用一个验证响应进行回答, 以认可或否认验证。验证的认可或否认在一个帧内的状态码域中被指示。对于可选的共享密钥验证, 基于一个验证设备如无线 STA 是否具有适当的有线等效协议 (wired equivalent privacy, 简称 WEP) 密钥而使验证受到影响。在这个方面, 该无线 STA 可以发送一个验证请求至接入点, 及该接入点可以通过在被发送至无线 STA 的响应帧内引入质询文本而进行响应。该无线 STA 被配置成使用其 WEP 密钥加密该质询文本, 及该加密的质询文本然后被从 STA 传送至该接入点。一旦接收了该加密的质询文本, 该接入点被用于解密该加密的质询文本及将其与原始文本进行对比。如果解密的文本与原始文本的对比表示二者间相互匹配, 则该接入点假定该无线 STA 具有正确的有线等效协议密钥。因此, 该接入点会发送一个验证帧至该无线 STA, 表明如果匹配则服务认可, 如果匹配失败则服务否认。

在无线STA接入无线媒介及被验证之后，该无线STA在数据通信开始之前必须要与接入点联系。联系允许如同步及接入点与无线STA间的重要信息的交换这样的任务。例如，在联系期间，关联的数据率可被从一个接入点传送至一个无线STA。无线STA可用于通过传送一个联系请求而启动联系，该联系请求包括的信息如支持的数据率、可选性能支持、安全性能支持、其它可选特性支持及SSID信息。在响应中，接入点可传送包括联系身份识别（ID）及其它接入点特别信息的联系响应帧。一旦该无线STA及接入点完成了联系，该无线STA然后开始与接入点进行数据帧的通信。

如果可用，该802.11标准提供一个可由用户开启或禁止的可选节能模式。如果被执行，该节能模式允许用户激活无线STA以适当地打开或关闭其无线收发机，以保存电池能量。例如，当不需要无线收发机或STA传送信息时，该无线STA可以关闭其无线收发机。在当该节能模式被打开的例子中，无线STA可通知可能的接入点其可进入休眠状态。在每个帧标题中的状态位可被用于表示无线STA的节能模式。在这点上，节能模式在该位为高电平时被打开，该位为低电平时被关闭。接入点被配置成记录表示其意图进入或退出该节能模式的每个无线站点。其允许该接入点缓冲那些当其在休眠模式时已经指示其可能进入休眠模式的可能性的无线STA的数据包，避免缓冲那些用于已经指示其试图退出（或不进入）休眠模式的无线STA的数据包。该已经进入休眠模式的STA可以周期性地从休眠状态醒来，并检测接入点是否具有缓冲的数据或新数据是否正在等候被传送。

该802.11协议提供对可被用于传输同步及时间限制服务的两个不同媒介访问控制（MAC）机制的支持。该第一机制为分布式协调功能（distributed coordination function，简称DCF），及第二机制为点协调功能（point coordination function，简称PCF）。分布式协调功能利用尽力服务促进信息的通信，其中具有要传送信息的接入设备具有同等的机会以传送信息。点协调功能可用于传送时

间敏感或潜在敏感信息。在这方面，点协调功能利用轮询机制，其可由扮演中央协调站（Point Coordinator, PC）角色的接入点（AP）控制。

在传输帧之前，站点被要求首先获得至共享无线媒介的访问。该802.11标准定义了一个媒介访问的分布协调功能（DCF）类型及媒介访问的点协调功能（PCF）类型。媒介访问的DCF类型是强制性的，且其用具有碰撞避免（CSMA/CA）协议的载波侦听多路访问。当无线STA试图发送帧时，DCF允许多个无线STA竞争访问无线媒介。无线STA可利用二进制补偿机制提供公平的媒介访问机制。因此，无线STA在试图访问媒介前会补偿一个随机数量的媒介空闲时间。

MAC层利用网络分配向量（network allocation vector, NAV）确保对媒介的公平访问。该NAV是一个计算器，其驻留地每个无线站点处，且代表一个帧或一组帧序列将要求传送包含于其中的数据的时间数量。为了允许公平访问媒介，MAC层检测网络分配向量（NAV）的值。当无线STA的NAV为0且任何补偿已被完成时，其被允许发送帧。在被允许发送帧之前，基于帧的长度及帧的数据率，站点被要求确定其需传送当前帧加上任何为相同帧序列部分的序列帧的时间数量。该站点会将这个确定的时间放置于要被传送的帧的帧标题的持续时间域中。当无线STA接收到该帧时，该时间从接收帧的持续时间域被获得，并被用于为其NAV确定相应的值。

被用于DCF的随机补偿计时器可被STA用于确定或检测该媒介是否是可访问的或繁忙的。如果确定了该媒介是繁忙的，STA在访问该媒体的另一试图做出前必须等待一个随机产生的时间段。这可确保一个公平的访问机制，并减少在相同时间上多个基站在媒介上传送数据的可能性。通过补偿强加的随机延迟，防止了多个无线STA在一个单独传输完结的空闲时同时感应媒介及随后同时试图传输，其会导致碰撞。因此，该随机补偿计时器显著地减少了碰撞的次数，因此重

发的次数也减少了，这当主动无线STA数量增加时是极其重要的。

当无线STA传送数据时其不会监听碰撞，因为当传输数据时，其不能使其接收器打开。这意味着无论何时接收的帧内没有错误被探测时，正接收的无线STA不得不发送肯定应答（ACK）。如果在一个确定的时间段过去后正传送的STA没有接收到ACK，该正传送的STA会自动地假定碰撞发生，且会在其自己的协定上重发该帧。该802.11标准通过可选的点协调功能（PCF）提供数据帧的时间限制传送。在可选的点协调功能中，在争用空闲时间期间通过轮询，一个接入点可以每一站点为基础而准许访问媒介。在这个方面上，无线STA在被允许传送帧之前必须被轮询。PCF通信业务可于可选的争用或DCF周期间被发送。在这个方面上，接入点可基于一个轮询表而轮询无线STA，并在无线STA使用DCF的周期内转换至一个轮询周期。这可允许同步操作模式，也可允许异步操作模式。例如，同步操作模式可被用于基于视频的应用，及异步操作模式可被用于支持浏览或消息应用。

无线STA适合于在每个帧的传送之前使用一个共同WEP密钥而加密每个帧的可效载荷。接收到加密的帧，正接收的无线STA或接入点会使用共同WEP密钥解密该接收的加密的帧。有多种不同尺寸的共同WEP密钥可用及每一个适于提供加密的变化强度。该协议也支持另外的安全方案。

包括发送请求(Request To Send, RTS)及清除发送(Clear To Send, RTS/CTS)的交换信号被接入点或STA用于控制通过RTS/CTS激活的STA访问及使用无线媒介。该无线STA可建立最大帧长度且无论何时该最大帧长度被超过，RTS/CTS都可自动被使用。无论何时无线STA激活RTS/CTS交换机制，该无线STA都会在其传送一个数据帧之前传送一个RTS帧至接入点或另一个STA。响应地，该接入点或其它的STA会传送一个CTS帧，表明了该无线STA可以传送数据帧。关于CTS帧，接入点或STA可以从RTS帧的帧标题内的持续时间域改变持续时间值，并将

这个改变的值放进CTS帧的帧标题内的持续时间域。这会阻碍其它站点的传送，直到启动PTS传送该数据帧的该无线STA已经完成了传送数据帧，且已经有机会接收ACK帧。

在传送期间，传送较小的信息片段比较长的数据片段更有效。这些较小的信息片段可被作为片段（fragment）。例如，一个包括L2标题信息、L3标题信息、L4标题信息、ULP信息及有效载荷数据的帧可被分成多个片段，其中所有的L3、L4、L5标题均在一个单独的片段内，该ULP信息及有效载荷数据的一部分可在另一个片段内，及剩余的有效载荷数据部分可分至其它多个片段内。如果该片段发生于802.11层内，与该标准相符合，该发送机不会乱序地发送这些片段。在这个方面上，该发送机可不开始后续的（n+i）片段的发送，其中I大于1，直至第n个或在先片段已被成功地发送。因此，乱序（out-of-order，简称OOO）片段不会在这种情况下发生。

媒介访问的标准分布协调功能就带宽利用来说是无效的，尤其在更高的特理层（PHY）速度上，例如，54Mbps或更高。该DCF可适用于解决如典型地与一些无线链接相关的网络拥挤及高包错误率（packet error rate，简称PER）这样的问题。该DCF也可按指数规律地增加补偿及肯定确认（positive acknowledgments，简称PACKs）。每一个MAC协议数据单元（MAC protocol data unit，简称MPDU）的补偿时间可为重发送按指数规律地增加，每个MPDU的PACK可在高物理层（PHY）速度上呈现带宽使用无效。该RTS/CTS机制在与规则DCF协同被使用时，可以减小效能，甚至可导致很少被使用。例如，在没有RTS/CTS被使用的情况下，传送以在54Mbps的速度传送一个包括MAC标题的1500字节帧要用248微秒。当PACK以24Mbps的速度传送时，传送平均补偿、PACK及短帧间空间（short interframe space，简称SIFS）的总和用130微秒。前置传输时间比数据传输时间的一半还多。

分布协调功能不是最大带宽有效传输机制。该802.11标准为MAC协议数据单元（MSDU）定义了一个分裂方法，称为存储碎片。在这个方面，MAC服务数据单元（MSDU）可于MAC级上被分成多个较小的MPDU。单独的MPDU包括一个可被以“猝发”形式被传送的MSDU，其中该帧间空间为一个SIFS，及PACK帧跟随每个片段的传输。因此，一个存储碎片下的典型帧交换序列为数据-短帧间空间-肯定确认-短帧间空间-数据-短帧间空间-肯定确认（DATA-SIFS-PACK-SIFS-DATA-SIFS-PACK），例如，在开始时具有一个可选的RTS/CTS交换。然而，在802.11标准中，存储碎片被定义为一种反对具有高包错误率（PER）的不可靠的无线链接方法。在可靠的无线链接中，即，那些具有低包错误率的无线链接中，由于在MPDU及PACK间的每个MPDU及SIFS间隔上引进了MAC标题，存储碎片可降低MAC效能。

为了处理各种服务质量（QoS）问题，如特别的QoS及MAC有效性的确保传送，要求更有效的带宽配置及使用机制。IEEE 802.11e起草标准定义了块肯定应答规则，其排除了为每个MPDU单独地传送肯定应答（PACK）的需要。该块肯定应答方案允许多个片段和/或帧在MAC层上被传送，而没有为每个片段发布一个单独的肯定应答。然而，这个块PACK机制引进了包括块PACK请求帧及块PACK响应帧的额外的系统开销。在块肯定应答方案下，一个传送机在块PACK响应被接收到之前会发送参数化数量的帧，确认块中帧的接收。MAC层应答机制尽管如此，但当乱序TCP片段被传送的情况下，也相应地会导致乱序段。IEEE 802.11e起草标准还定义了分配特别设备的时间的争用空闲周期，其中帧可与间隔的SIFS周期一起被传送，而不是在先描述的被一个单独的无线STA于传送的帧间的补偿间隔。这个配置的机制是复杂的，并包括涉及轮询机制的额外开销。

通过结合附图的本申请的其余部分的阐述，将本发明一些方面与这种系统进行对比，常用的和传统的方法的进一步限制及缺点对本领域的普通技术人员来说

是显而易见的。

发明内容

本发明提供一种通过直接数据布置处理无线系统内乱序分段的方法和/或系统，结合至少一个附图充分地展示和/或描述，并更完整地于权利要求中提出。

根据本发明的一个方面，提供一种处理无线系统内乱序TCP分段的方法，包括：

下列至少之一：将无线网络处理器接收的第一TCP分段放置于主机缓冲区内，映射所述第一TCP分段的TCP序列数量及相应缓冲区地址，确定第二接收的TCP分段是否为有序TCP分段或乱序TCP分段中之一；及

如果所述无线网络处理器接收的所述第二TCP分段为乱序TCP分段，则：

将与至少所述第二TCP分段相关的控制信息存储于所述无线网络处理器上；及

将所述乱序TCP分段放置于所述主机缓冲区的一部分内。

优选地，所述方法进一步包括记录所述无线网络处理器上被所述第一接收的TCP分段的结尾占用的至少一个缓冲区地址，所述缓冲区为TCP缓冲区、ULP缓冲区及应用缓冲区之一。

优选地，所述方法进一步包括如果所述第二接收的TCP分段为有序TCP分段，则将所述有序TCP分段相邻于所述第一接收的TCP分段放置于所述主机缓冲区内。

优选地，所述方法进一步包括将所述有序TCP分段从与所述第一接收的TCP分段的所述结尾占用的所述被记录的地址相邻的地址开始放置于所述主机缓冲区内。

优选地，该方法进一步包括下列至少之一：

通过所述无线网络处理器记录下列数据指示至少之一：

所述有序TCP分段的结尾占用的结尾缓冲区地址；

开始缓冲区地址加上所述有序TCP分段的数据部分的长度；及

跟随所述有序TCP分段的结尾的一个字节的缓冲区地址；及

在所述无线网络处理器上存储与至少一个所述指示数据相关的控制信息。

优选地，该方法进一步包括下列至少之一：

在所述无线网络处理器上记录下列数据指示至少之一：

所述放置的乱序TCP分段的开始所占用的地址，及所述放置的乱序TCP分段的结尾所占用的地址；及

所述的加上所述放置的乱序TCP分段占用的的长度的开始缓冲区地址；

在所述无线网络处理器上存储与所述网络处理器上的所述指示数据相关的控制信息。

优选地，所述方法进一步包括通过所述无线网络处理器接收第三乱序TCP分段。

优选地，所述方法进一步包括通过所述无线网络处理器确定所述主机缓冲区中的所述第三接收的乱序TCP分段的放置。

优选地，所述方法进一步包括

如果所述第三接收的乱序TCP分段是与所述放置的乱序TCP分段有序的，

则：

将所述第三乱序TCP分段相邻于所述放置的乱序TCP分段放置于所述主机缓冲区内；及

下列至少之一：

将所述放置的乱序TCP分段所占用的开始地址调整为结合所述第三乱序TCP分段及所述乱序TCP分段的开始地址；及

将所述放置的乱序TCP分段所占用的开始地址调整为结合所述第三乱序TCP分段及所述乱序TCP分段的结尾地址。

优选地，所述方法进一步包括，如果所述第三接收的乱序分段是关于所述放置的乱序分段乱序的，则将所述第三乱序TCP分段放置于在所述主机缓冲区内一个地址上，该地址位于所述放置的乱序TCP分段之前及所述放置的乱序TCP分段之后之一。

优选地，所述方法进一步包括下列至少之一：

在所述无线网络处理器上记录至少一个位置数据表示：

所述第三接收的TCP分段被放置于所述无线网络处理器上的主机缓冲区中的位置的开始地址及结尾地址；

开始地址加上所述第三接收的TCP分段被放置于所述无线网络处理器上的主机缓冲区处的长度；及

跟随所述第三接收的TCP分段被放置于所述无线网络处理器上的主机缓冲区处的结尾的字节；及

在所述无线网络处理器上存储与至少所述第三接收的TCP分段被放置处的所述位置数据相关的控制信息。

优选地，所述方法进一步包括在所述主机缓冲区内所述第一接收的TCP分段的所述放置之上将TCP窗口调整至所述最后有序接收的TCP分段的结尾。

优选地，所述方法进一步包括下列之一：

将所述TCP窗口的左边缘调整至紧接着与最后有序在先接收的TCP分段的结尾相邻放置的乱序TCP分段的结尾的下一个字节；及

将所述TCP窗口的左边缘调整至紧接着与最后有序在先接收的TCP分段的结尾相邻放置的多个乱序TCP窗口的最后一个的结尾的下一个字节。

优选地，所述方法进一步包括将所述TCP窗口的左边缘调整至紧接着与所述

第一接收的TCP分段相邻放置的最后有序TCP分段的结尾的下一个字节。

优选地，所述方法进一步包括将与至少所述TCP窗口的位置相关的控制信息存储在所述无线网络处理器上。

优选地，所述方法进一步包括每当所述TCP窗口被调整时，更新所述与至少所述TCP窗口的位置相关的控制信息。

优选地，所述方法进一步包括：

在第一缓冲区内计算第一TCP分段的数据的第一字节的位置的缓冲区地址；

基于相应于所述第二TCP分段的TCP分段数量的缓冲区地址于第二TCP分段内计算所述第一数据字节的缓冲区地址；及

计算从相应于所述第一TCP分段的所述TCP分段数量至所述第二TCP分段的第一字节的偏移量。

优选地，所述方法进一步包括确定所述计算的偏移量是否大于所述第一缓冲区的长度。

优选地，所述方法进一步包括如果所述计算的偏移量大于所述第一缓冲区的长度，则从所述计算的偏移量中减去所述第一缓冲区的长度由此得出一个新的偏移量。

优选地，所述方法进一步包括从所述新的偏移量中减去第二缓冲区的长度。

优选地，所述方法进一步包括如果所述第二缓冲区的减去导致所述新的偏移量小于所述第二缓冲区的长度，在所述第一缓冲区及所述第二缓冲区中缓冲所述第二TCP分段的数据。

优选地，所述方法进一步包括如果所述新的偏移量大于所述第二缓冲区的长度，减去其它缓冲区的长度，直到以下之一：

所述缓冲区被用尽；及

剩余的偏移量小于当前缓冲区的长度。

优选地，所述方法进一步包括如果剩余的偏移量小于所述当前缓冲区的长度，在所述第二缓冲区、所述第三缓冲区及所述其它缓冲区内缓冲所述第二TCP分段的数据。

优选地，所述方法进一步包括如果缓冲区被用尽，丢弃所述第二TCP分段。

根据本发明的一个方面，提供一种处理无线系统内乱序TCP分段的系统，该系统包括：

至少一个处理器用于执行下列至少之一：

将无线网络处理器接收的第一TCP分段放置于主机缓冲区内，在TCP序列数量与相应缓冲区地址间具有一个映射，确定无线网络处理器接收的第二TCP分段是否为有序TCP分段或乱序TCP分段中之一；及

如果所述第二接收的TCP分段为乱序TCP分段，则：

所述至少一个处理器将与至少所述第二TCP分段相关的控制信息存储于所述无线网络处理器上；及

所述至少一个处理器将所述乱序TCP分段放置于所述主机缓冲区的一部分内。

优选地，所述至少一个处理器记录所述无线网络处理器上被所述第一接收的TCP分段的结尾占用的至少一个缓冲区地址，所述缓冲区为TCP缓冲区、ULP缓冲区及应用缓冲区之一。

优选地，如果所述第二接收的TCP分段为有序TCP分段，则所述至少一个处理器将所述有序TCP分段相邻于所述第一接收的TCP分段放置于所述主机缓冲区内。

优选地，所述至少一个处理器将所述有序TCP分段从与所述第一接收的TCP分段的所述结尾占用的所述被记录的地址相邻的地址开始放置于所述主机缓冲区内。

优选地，所述至少一个处理器执行下列至少之一：

通过所述无线网络处理器记录下列数据指示至少之一：

所述有序TCP分段的结尾占用的结尾缓冲区地址；

开始缓冲区地址加上所述有序TCP分段的数据部分的长度；及

跟随所述有序TCP分段的结尾的一个字节的缓冲区地址；及

在所述无线网络处理器上存储与至少一个所述指示数据相关的控制信息。

优选地，所述至少一个处理器执行下列至少之一：

在所述无线网络处理器上记录至少一个数据指示：

所述放置的乱序TCP分段的开始所占用的地址，及所述放置的乱序TCP分段的结尾所占用的地址；及

所述的加上所述放置的乱序TCP分段占用的长度的开始缓冲区地址；

在所述无线网络处理器上存储与所述网络处理器上的所述指示数据相关的控制信息。

优选地，所述缓冲区地址TCP、ULP及驱动器至少之一提供。

优选地，所述至少一个处理器为TCP卸载引擎（TCP offload engine，简称TOE）、RISC处理器及有限状态机（finite state machine，简称FSM）之一，其中每一个被集成于所述无线网络处理器中。

本发明的这些及其它优点、方面及新颖性特征同所举的实施例的细节将从下面的描述及附图中被更全面地理解。

附图说明

图1A为根据本发明的实施例的无线通信系统的结构示意图。

图1B为根据本发明的实施例的用于与处理无线系统内乱序分段有关的一示例性系统的结构示意图。

图1C为根据本发明的实施例的用于与处理无线系统内乱序分段有关的另一示例性系统的结构示意图。

图1D为根据本发明的实施例的用于处理乱序（OOO）TCP分段的示例性系统的可选实施例的结构示意图。

图1E为根据本发明的实施例的以一种直通方式处理乱序（OOO）传输控制协议（TCP）数据报的系统的结构示意图。

图1F为根据本发明的实施例的帧接收的示例性步骤的流程图。

图1G为根据本发明的实施例的示例性传输路径的结构示意图。

图2A为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内接收的TCP分段的结构示意图。

图2B为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。

图3为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内多个接收的乱序TCP分段的结构示意图。

图4为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的有序TCP分段的结构示意图。

图5为根据本发明的实施例的插入第一漏洞的一部分且用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。

图6为根据本发明的实施例的插入第一漏洞的剩余部分且用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。

图7A为根据本发明的实施例的完全插入一个漏洞且用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。

图7B为根据本发明的实施例的示例性的更新变量的结构示意图。

图8为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的

TCP序列空间内及配置前的示例性变量及缓冲区的状态的新接收的有序TCP分段的结构示意图。

图9为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内及描述随后配置的示例性变量及缓冲区的各种状态的新接收的有序TCP分段的结构示意图。

图10为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间及其缓冲内的新接收的有序TCP分段的处理的结构示意图。

具体实施方式

本发明的某些实施例可被发现于通过直接数据布置处理无线系统内乱序片段（fragment）或分段（segment）的方法及系统中。

图1A为根据本发明的实施例的无线通信系统的结构示意图。参考图1A，所示的通信系统10包括多个基站和/或接入点12-16，多个无线通信设备18-32及网络硬件元件34。该无线通信设备18-32可为膝上型主机电脑18及26、个人数字助理主机20及30、个人计算机（PC）主机24及32和/或蜂窝电话主机22及28。

例如，基站和/或接入点12-16可通过局域网（local area network，简称LAN）连接36、38及40而被操作地连接至网络硬件34。该网络硬件34如路由器、开关、电桥、调制解调器或系统控制器可为通信系统10提供广域网连接42。每个基站和/或接入点12-16可具有一个相关的天线或天线阵列，以在其区域内与无线通信设备进行通信。典型地，无线通信设备向一特定基站和/或接入点12-16注册，以从该通信系统10接收服务。对于直接连接，例如点对点通信，无线通信设备可通过一个或多个分配的信道直接地通信。一般，基站被用于蜂窝电话系统或相似类型的系统，而接入点被用于家庭内或室内无线网络。无论哪种特定类型的通信系统，每个无线通信设备包括内置无线电设备和/或与无线电设备相连接。

图1B为根据本发明的实施例的用于与处理无线系统内乱序分段相关的示例

性系统的结构示意图。参照图1B，该系统包括处理器102、存储控制器104、主机存储器106、主机界面108、无线网络界面处理系统110及天线112。在本发明的一个实施例中，存储控制器104可集成至控制器102。无线网络界面处理系统110可为一个无线网络处理器110。

无线网络处理器110包括如TCP卸载引擎（TCP Offload Engine, TOE）114或集成的无线网络处理器芯片或芯片组。该无线网络处理器110包括如无线网络界面卡（WNIC）和/或无线网络界面处理逻辑、电路和/或编码。例如，该无线网络界面处理逻辑包括单片无线网络芯片或处理器或芯片组以用于提供相似类型的功能。在本发明的一个实施例中，该无线网络界面卡可通过主机界面108而被连接至主机105。主机界面108可为如通用串行总线（universal serial bus, 简称USB）、外围部件互连（peripheral component interconnect, 简称PCI）、PCI-X、ISA、SCSI或其它类型的总线。该存储控制器106可被连接至处理器104、存储器106及主机界面108。图1B的系统可根据本发明的各个方面而以用于直通式方式处理乱序（OOO）传输控制协议（TCP）数据报。

图1C为根据本发明的实施例的用于与处理无线系统内乱序分段有关的另一示例性系统的结构示意图。参照图1C，该系统包括如处理器102、主机存储器106、专用存储器116及芯片组118。该芯片组118包括如无线网络处理器110。该芯片组118可被连接至CPU102、主机存储器106及专用存储器116及天线112。该芯片组118的无线网络处理器110包括TOE，且可被连接至天线112。该专用存储器116可为文本和/或数据提供缓冲。图1C的系统可用于处理乱序（OOO）传输控制协议（TCP）数据报，例如，根据本发明的各个方面以直通式方式。

该专用存储器116为文本和/或数据提供缓冲。存在多个选择以存储文本缓冲。这些选择包括于TOE114内存储文本，于专用存储器116内存储文本及于主机存储器106内存储文本。相应地，图1A、图1B、图1C及图1D中所示的各个实施

例配置拥有在通过这些存储选择中处理文本存储的能力。

虽然本发明举例示出了各种实施例，例如，CPU及包括天线的无线界面，本发明不必限于这些例子。相应地，本发明的各种实施例可分别地采用，例如，任何类型的电路和/或处理器及任何类型的数据链接层或物理媒介。相应地，虽然示例的为连接至图1B的天线112或TOE 114可被用于处理各种类型的数据链接层或物理媒介。而且，本发明还考虑到在图1B-C中所示元件间不同程度的集成及分离。例如，TOE114可为从嵌入至一个电路板的芯片组118上分离的集成芯片或可被嵌入于WNIC中。另外，专用存储器116可被集成至芯片组118或被集成至图1C的无线网络处理器110。

图1D为根据本发明的实施例的可用于处理乱序（OOO）TCP分段的示例性系统的可选实施例。参照图1D，其示出了主机处理器124、主机存储器/缓冲区126、软件算法模块134及无线网络处理模块128。无线网络处理模块128包括网络处理器130及简化的无线网络处理器存储器/缓冲区模块132。

根据本发明的实施例，软件算法模块134包括可被用于控制主机处理器124和/或无线网络处理器（WNP）128的代码，以处理乱序（OOO）TCP分段。乱序TCP分段可被语法分析或TCP文本信息可获得。主机处理器124可决定标题信息及有效载荷的边界，及该有效载荷数据可被映射至主机存储器/缓冲区126。映射之后，该有效载荷可被传送至主机存储器/缓冲区126。

图1E为根据本发明的实施例的以一种直通方式处理乱序（OOO）传输控制协议（TCP）数据报的系统的结构示意图。参照图1E，其示出了物理层（PHY）71、媒介访问控制（MAC）模块72、CRC模块73、DMA引擎、主机总线75、主机缓冲模块76、控制路径78、数据路径78、帧缓冲区83、帧语法分析模块84。图1E进一步包括多个存储选择，其包括芯片内高速缓冲存储器模块79a、79b、主机上存储模块80a、80b、芯片外存储器81a、81b，81c及芯片内存储器82a、82b。

一般，输入帧可以被L2处理，包括如地址滤波、帧有效性及错误探测。一个输入帧在由PHY 71、MAC 72及CRC模块72处理之后，或被发送至帧语法分析模块84，以进行语法分析。帧语法分析模块84可被用于分析来自帧的控制信息及实际的有效载荷数据。帧语法分析模块84可被用于促进L2、L3、L4标题信息的语法分析、相容性检测、多元组查找及可编程和固定规则检测。在帧语法分析模块84完成语法分析后，所得的控制信息可通过控制路径77被传送以进行处理，及有效载荷数据和/或原始打包数据可通过数据路径78被传送以进行处理。该原始打包数据包括可选标题信息。该经语法分析的有效载荷打包数据可于帧缓冲模块内被缓冲。在本发明的一个可选实施例中，经语法分析的有效载荷打包数据的至少一部分可被存储于芯片外存储模块，如芯片外存储器81c。在这个方面，进出帧缓冲区的原始打包信息和/或有效载荷数据可被移至该芯片外存储器。DMA引擎74可将出帧缓冲区的DMA数据移入主机缓冲区模块76内的缓冲区。

与普通以太网控制器不同，处理的下一个阶段包括如IP处理这样的L3及如TCP处理这样的L4。该无线网络处理器110可减少主机CPU的使用及存储器带宽，例如，通过在硬件卸载TCP/IP连接上处理通信业务。该无线网络处理器110可探测如输入数据包所属的协议。对于TCP，无线网络处理器110可探测该数据包是否相应于一个卸载TCP连接，例如，对于一个连接，至少一些TCP状态信息可被无线网络处理器110保持。一旦连接已经与数据包或帧的连接，如L5或其上的任何更高级别的处理可被获得。如果该数据包相应于一个卸载连接，那么该无线网络处理器110可指引所述帧的数据有效载荷部分的数据移动。有效载荷数据的目的地可从与帧内方向信息相结合的连接状态信息确定。如该目的地可为主机存储器。最后，该无线网络处理器110更新其内部TCP及连接状态的更高级别，且可获得来自其内部连接状态的主机缓冲区地址及长度。

该控制路径77内的系统元件可被用于处理各种用于以最大适应性及效率及

目标有线速度完成如L3/L4或高级处理的处理阶段。这些元件包括联系模块85，文本获取模块86，接收（Rx）处理模块87，TCP代码88及高速缓冲存储器模块。这些处理阶段的结果包括如一个或多个数据包识别卡（PID-Cs），其可提供与帧有效载荷数据相关联的承载信息的控制结构。当在各种模块内处理数据包时，其已经发生于无线网络处理器110内。该接收处理模块包括适合的逻辑、电路和/或代码，可被用于产生被用于控制DMA引擎74的缓冲控制信息。

在帧语法分析模块84分析来自输入帧的TCP/IP标题后，联系模块85将该帧与端到端TCP/IP连接联系。该文本获取模块86被用于获取TCP连接文本及处理TCP/IP标题。标题和/或数据边界可被确定，数据可被映射至主机缓冲区模块76内的一个或多个主机缓冲区。该DMA引擎74可被用于通过主机总线75将数据DMA传送至主机缓冲区模块76内的缓冲区。该标题可在芯片内被消耗或通过DMA引擎被传送至主机。

帧缓冲区83可为接收系统结构内的可选模块。其为了相同的目的而被使用、如先进先出（FIFO）数据结构被用于传统L2 NIC或为传统的处理存储更高层通信业务。接收系统内的该帧缓冲区83不被限于一个单独的例子，且相应地，可为该帧缓冲区83的多个例子。在本发明的一个实施例中，FIFO的一个例子可被用于多个连接。当控制路径77处理语法分析的控制信息的处理时，根据协议要求，该数据路径78可一次或多次在数据处理阶段之间存储相应的数据。

图1F为根据本发明的实施例的帧接收的示例性步骤的流程图。参照图1E及图1F，在步骤191中，无线网络处理器110从如天线112接收一个帧。在帧192中，该帧语法分析模块84可对该帧进行语法分析，如，发现L3及L4标题信息。该帧语法分析模块84可处理引导L3标题信息的L2标题信息，如第4版本IP（IPv4）标题或第6版本IP（IPv6）标题。该IP标题版本域可被用于确定该帧是否承载IPv4数据报或IPv6数据报。

例如，如果IP标题版本域承载值为4，那么该帧可承载IPv4数据报。如果，例如，该IP标题版本域承载值为6，那么该帧可承载IPv6数据报。该IP标题域可被提取，因此获得如该IP源（IP SRC）地址，该IP目的（IP DST）地址，及该IPv4标题“协议”域或该IPv6“下一个标题”。如果该IPv4标题“协议”域或该IPv6“下一个标题”标题域承载值为6，那么接下来的标题可为TCP标题。语法分析的结果可被加到PID_C及该PID_C可在无线网络处理器110内与数据包一起传输。

其余该IP测试的处理可以一种与传统的现有软件栈内的处理相似的方式而随后发生。执行可从嵌入式处理器上的韧件的使用到潜在地更快的专用的有限状态机，或处理器与状态机的结合的范围内变化。该执行可如通过一个或更多的处理器、状态机或混合电路，随处理的多个阶段一起变化。该IP处理包括但不限于提取与如长度、有效性及存储碎片相关的信息。该定位的TCP标题信息也可被语法分析和处理。该TCP标题信息的语法分析可提取与如源端口及目的端口相关的信息。

该TCP处理可被分成多个附加的处理阶段。在步骤193中，该帧可与一端到端TCP/IP连接相联系。在L2处理之后，在一个实施例中，本发明可提供TCP校验和被验证。该端到端连接可被通过例如如下5个多元组而被唯一地定义：IP源地址（IP SRC addr）；IP目的地址（IP DST addr）；如TCP、UDP或其它上层协议这样的IP协议之上的L4协议；TCP源端口数量（TCP SRC）；及TCP目的端口数量（TCP DST）。该处理可适用于具有相关IP地址选择的IPv4或IPv6。在步骤194中，TCP连接文本可被获取。在步骤195中，该TCP/IP标题可被处理。在步骤196中，确定标题信息及有效载荷数据的边缘。其可包括TCP选择及任何上层协议（ULP）标题信息。TCP选择或ULP的另外处理可先于有效载荷映射及布置。在步骤197中，有效载荷数据可被映射至一个或更多的主机缓冲区。在步骤198中，有效载荷数据可被传输至它们被映射到的主机缓冲区。

作为在步骤192帧语法分析的结果，该5个多元组可被完全地提取，且可于PID_C内获得。联系硬件可将该接收到的5个多元组与存储于无线网络处理器110内的5个多元组的列表进行对比。该无线网络处理器110可保留多元组的列表，其代表着如前一个处理过的卸载连接或正被无线网络处理器110管理的卸载连接。该用于存储联系信息的存储器资源对于芯片内及芯片外的选择来说是昂贵的。因此，不是所有的联系信息均被收容于芯片上是有可能的。高速缓冲存储器可被用于芯片上的存储最有效的主动连接。如果发现了一个匹配，那么该无线网络处理器110可管理与该匹配的5多元组相连接的特定TCP/IP。

图1G为根据本发明的实施例的示例性传输路径。参照图1G，该无线网络处理器110包括如物理层（PHY）180、MAC层190、标题生成器200、文本预取210、计时器模块220、传送处理器230、数据及控制模块240、确认模块250、程序安排260及DMA引擎270。该计时器模块220包括如TCP状态传送及重传送计时器。该程序安排260被用于进行如至具有传输、共享及请求访问媒介、窗口和/或重传送判断的多连接的下一个连接的选择。该DMA引擎270包括如XSUM模块280或其它数据特别处理。例如，用于如iSCSI协议要求的CRC。其包括将数据插入由主机提供的数据中及计算CRC值。

在传输路径上，对L4及更高级别的支持包括附加的复杂性及功能性。传输包括执行如下的一个或更多：安排传输流程；通过DMA传送数据；获取文本；传送处理；将L5或更高与TCP/IP标题相加；提供计时器；及L2传送。

程序安排260可用于确定接下来应进行哪一个流程。该程序安排260也可被用于处理通信业务的L2、L4及更高级别的复用。关于通信业务的L4及更高级别，安排一个特别的TCP/IP流程以用于传输的决定依赖于如下因素的一个或更多：在主机侧传输的数据有效性；远端缓冲状态如当远程TCP连接没关闭TCP窗口时；防止媒介上的潜在大量TCP连接竞争时间中的资源缺乏；来自接收侧的TCP确认

的有效性；及在TCP连接的利益上重传送信息的需要。

利用上面识别的信息或其它信息的一些或全部，程序安排260可选择下一个要被传输的流程。在本发明的一个实施例中，程序安排260包括适合的逻辑，电路和/或代码，以用于选取来自文本信息的至下一个主机常驻缓冲区的指示器。该程序安排260还编程DMA引擎270以如在芯片内FIFO缓冲区中得到数据并存储数据。

该DMA引擎270包括适合的逻辑，电路和/或代码，以用于将数据从主机缓冲区或缓冲区传送至如芯片内传送侧FIFO缓冲区。如IPv4校验和/或TCP校验这样的IP校验可为正被传送的数据被计算。该计算可与数据移动同时被执行。更高级的数据处理也可在该阶段被执行。

通过文本预取210从中心文本源中可获取流程文本。访问该中心文本源可增加其所有消耗装置中的锁定机构的有效性以确保数据的完整性及相干性。锁定机构可有效地最小化不渴望性能影响。该选择的流程文本可被提供至如CPU和/或有限状态机（FSM）这样的传送处理器230。

该传送处理器230可被用于如执行TCP/IP及更高代码级，以更新文本及产生TCP/IP及更高级标题变量以被放置于标题内。该更新的文本被存储。这个阶段的处理通过一个或更多包括一个或更多处理器、状态机或混合处理器的阶段而被执行。

标题生成器200可被使用应用通过传送处理器230产生标题变量，及可产生TCP/IP及更高级别的标题信息和/或报尾，及可将该TCP/IP及更高级别的标题附着于要被传送的数据的前面。通过使用自DMA引擎270获得的部分校验和结果，该标题生成器200可最后确定该校验和域及可将它们安置于各自的标题内。传送处理不限于特定数量的阶段和处理，且可作为在不同级的最佳处理阶段被执行。

该计时器220被传送处理器230提供，且可更新其未来计时事件表。当该L4

及更高级的处理被完成时，该L2处理及传输可通过传统的无线控制器而遵从传统步骤执行。

重传送事件与正常传送相似，除了要发送的数据如以前从主机缓冲区获取或从任何其它暂时的支持缓冲区获取。这个数据的地址可被计算。主机缓冲区地址的计算可更复杂。描述于接收路径段的相同映射功能可被用于重传送TCP序列数值范围。一旦缓冲区地址被确定，传输处理的剩余部分可按如上所述而发生。

对于接收器上的数据重新装配，一块接收的帧包括乱序信息是可能的。在这个方面，接收块内的一个或更多的TCP分段可能是遗漏的。一个遗漏的TCP段可被认作为一个漏洞（hole）。具有遗漏的TCP分段或漏洞的控制信息或数据的总和也可被存储或保留于无线网络处理器110或网络系统50。其可提供一个直通式结构，该结构可在无线NIC或无线网络处理器上要求最小的资源。在这个方面，该无线网络处理器不需要支持缓冲区用于存储或缓冲乱序TCP分段，且能够利用源有效可升级存储器管理乱序TCP分段，因为乱序TCP分段可被放置或至少存储于暂时的缓冲区内。在本发明的一个方面中，如FIFO这样的暂时缓冲区或主机缓冲区可被用于暂时性地缓冲TCP分段。通过这种方法，数据要求的存储器不会如典型的传统系统那样与连接数量、每个连接的漏洞的数量或TCP连接带宽延迟产品数量成比例。相应地，被定位以容置控制数据的附加空间与漏洞的数量成比例，不与乱序的TCP分段或帧的数量成比例。

根据本发明的一个方面，示例性的TCP控制板状态的分析连同其提供给用户的信息及服务，显示了稍微多一些的计算，直通式无线或无线网络处理器可支持乱序TCP分段而不用为on-NIC数据缓冲区存储乱序TCP分段。如FIFO或芯片内存存储器的其它组织这样的暂时性的缓冲可被利用。相应地，无论何时数据符合TCP规则或被适当地拒绝，乱序TCP分段一到达被充分地处理以允许数据的布置。在这个方面，被适当地拒绝的数据如非法的或失效的数据。非法的或失效的数据包

括已经超出相应的TCP窗口或没有在相应的TCP窗口内被接收到的数据。

图2A为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内接收的TCP分段的结构示意图。参照图2A，其示出了在TCP序列空间及TCP窗口282内的一个接收的TCP分段281。TCP接收窗口（RCV.NXT）左边界右边的数据表示为L，及TCP接收窗口（RCV.NXT+RCV_WIND）右边界左边的数据表示为R，其根据TCP协议而被接收。

图2B为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。参照图2B，其示出了TCP序列空间及TCP窗口285内的有序TCP分段283、乱序TCP分段284。

在本发明的一个实施例中，新接收的乱序TCP片段284产生位于有序TCP分段283与新接收的乱序TCP分段284间的第一漏洞。该第一漏洞的开始由一个变量Hole_1_Start表示，该第一漏洞的结尾由一个变量Hole_1_End表示。新接收的乱序TCP分段284的最左边部分的位置可被赋值一个变量First_TCP_Seq，新接收的乱序TCP分段284的最右边部分的位置可被赋值一个变量Last_TCP_Seq。新接收的乱序TCP分段284的最左边部分的位置还可被赋值一个变量如OOO_List[0].Begin。新接收的乱序TCP分段284的最右边部分的位置还可被赋值一个变量如OOO_List[0].End。一般，变量OOO_List[0].Begin及变量OOO_List[0].End提供了一个可被用于跟踪乱序TCP分段的列表。相应地，在本发明的一个方面中，被用于跟踪乱序TCP分段的列表可在一个阵列中被执行。而且，本发明不只限于此，而其它数据结构可被利用而不脱离本发明的各种方面。

在TCP分段是乱序的例子中，在一个主机处理器或专用的无线网络界面卡上运行的共同TCP堆栈或者落入该TCP分段或者缓冲该TCP分段，直到TCP序列空间的漏洞被堵上。堵上后，该作为乱序TCP分段被接收的一个或多个TCP分段可被处理。

尽管如此，通过直通式结构，希望TCP分段的TCP处理具有最小的或真实的无芯片内缓冲。在这个方面，在网络被接收时，有效的TCP分段可被处理并被置于主机缓冲区内。有效的TCP分段是那些可以落入TCP窗口内的分段。在这种情况下，接收器可以获得和/或监视与乱序TCP分段相关的数据，且保持TCP控制空间的计算以使漏洞可被堵上并加倍阻止。对于TCP分段空间内的每个漏洞，接收器可保持例如其开始及结尾点的TCP序列数量或开始及长度。我们用一个作为例子的方法举例说明了关于伪码的另一个方法。其后接收的任何新的TCP分段可被检测以注意其是否为有序的，相邻于一个漏洞或是否为新的乱序TCP分段。漏洞的边缘或边界可被相应地移动。该接收处理遵从TCP协议，其中，复制的字节不被接收及落入窗口外的数据不被接收。例如，变量First_TCP_Seq及Last_TCP_Seq可被用于保持第一漏洞的轨迹，该第一漏洞由图2B的TCP序列空间上的新的乱序分段284的布置而产生。

图3为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内多个接收的乱序TCP分段的结构示意图。参照图3，其示出了有序TCP分段302、在先乱序TCP分段304及新的乱序TCP分段306。该第一漏洞被配置于图2B所示的有序TCP分段302和在先乱序TCP分段304之间。由于新的乱序TCP分段306的布置，第二漏洞被配置于在先乱序TCP分段304与新的乱序TCP分段306之间。第二漏洞的开始可由如变量Hole_2_Start表示，该第二漏洞的结尾由如一个变量Hole_2_End表示。新的乱序TCP分段306与第二漏洞的结尾相邻的部分Hole_2_End可被赋值一个变量如OOO_List[1].Begin。分段306的结尾可被赋值一个变量OOO_List[1].End。

图4为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的有序TCP分段的结构示意图。参照图4，其示出了一个有序分段404、新的有序分段、在先乱序分段406及新的乱序分段408，所有的这

些被置于一个TCP序列空间内。新接收的有序分段404与有序分段402相邻，根据本发明的一个方面，因为该新的有序TCP分段关于该有序分段402是有序的，相关于第一漏洞和/或第二漏洞的变量不必被改变。然而TCP窗口的左边缘可被更新。另外，其也可造成TCP窗口的右边缘的更新。

图5为根据本发明的实施例的插入第一漏洞的一部分且用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。参照图5，其示出了有序分段502、新的乱序分段506及乱序分段508及在先乱序分段510，所有的这些被置于一个TCP序列空间内。

新接收的有序分段506与乱序分段508相邻且堵上了第一漏洞的一部分。相应地，在本发明的一个方面中，由于该新接收的有序分段506堵上了第一漏洞的一部分，变量OOO_List[1].Begin的开始可被调整以反映相邻于乱序分段508的新接收的有序分段506的位置。然而，由于新接收的有序分段506与乱序段508相邻，变量OOO_List[1].End不必被改变。当新的乱序帧被接收时，该第一漏洞收缩可被要求进行漏洞管理的信息的数量中没有改变的尺寸。

图6为根据本发明的实施例的插入第一漏洞的剩余部分且用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。参照图6，其示出了合并的TCP分段602、新的乱序分段606、合并的TCP分段610及乱序分段612，所有的这些被置于一个TCP序列空间内。新接收的乱序段606在合并的TCP片段602与合并的TCP片段610之间。在这个方面，新的乱序分段606堵上了第一漏洞的剩余部分。

一些如TCP上的RDMA的ULP不被TCP内的漏洞开启，及完成ULP处理仍然是可能的。对于一些ULP，当在TCP分段内有一个漏洞时，这可激活一些不可能的ULP处理。这个处理可于芯片内或主机上被执行。在另一种情况下，TCP有效载荷的布置可被执行。

接着第一漏洞的堵上,在新的第一漏洞处第二漏洞被指示,与第二漏洞相关的参数被指示与该新的或当前的第一漏洞相关。例如,先前与第二漏洞相关的变量OOO_List[1].Begin被调整或被指示为变量OOO_List[0].Begin。先前与第二漏洞相关的变量OOO_List[1].End被调整或被指示为变量OOO_List[0].End。最后,TCP窗口614的开始从在先第一漏洞的开始移动至新的或当前第一漏洞的开始。在这个方面,变量TCP Seq=RCV_NXT的移动方式为,变量TCP Seq=RCV_NXT开始于新的或当前第一漏洞的开始,并延伸至TCP序列空间的右边。如果附加的缓冲区对于接收器来说是有用的,并已经通过TCP协议被通告至远端对等处,则变量TCP Seq=RCV_NXT+REV_WIND可通过等于TCP窗口614的尺寸的数值而被延伸至右边。在本发明的一个方面中,由于新的乱序帧完全地堵上了第一漏洞,漏洞管理的要求较少的信息被存储。相应地,一个漏洞描述符可被移除,且窗口左边缘移动至右边。

一般,被接收及被确定为乱序的新的TCP分段或帧可被用于缩减与任何存在的漏洞相邻的它们所占用的TCP序列数量空间内的任何漏洞。如果新的帧堵上了两个相邻漏洞间的空间,那么该漏洞被合并进一个漏洞。相应地,通过附加参考图5,图6示出了当一个漏洞由一个乱序TCP分段完全堵上时的TCP分段的合并。另外,图6举例示出了当一个漏洞被完全地堵上时,TCP窗口614的左边缘的移动及可选的右边缘的移动。在这个方面,通过参考图5及图6,TCP窗口614的最左侧边缘从有序TCP分段502(图5)的最右边缘移动至在先乱序TCP分段610(图6)的最右侧边缘,该在先乱序分段现在是最后有序分段。新的有序TCP分段502的最右边缘与合并的TCP分段602是同步的。

图7A为根据本发明的实施例的完全插入一个漏洞且用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内新接收的乱序TCP分段的结构示意图。参照图7A,其示出了第一漏洞、第二漏洞、有序TCP分段702、在先乱序TCP分段704、

新的乱序TCP分段706及在先乱序TCP分段708。除了图7A，图3也被用于作为图7A的示例的参考。

相应地，图7A的第一漏洞配置于有序TCP分段702与在先乱序TCP分段704之间。第二漏洞配置于在先乱序TCP分段704与在先乱序TCP分段708之间。该新的乱序TCP分段与在先乱序TCP分段704与在先乱序TCP分段708均相邻。在这一点上，乱序TCP分段706完全堵上了第二漏洞。在本发明的一个方面中，由于第二漏洞被完全堵上了，与第二和/或第一漏洞相关的参数可被更新以反映第二漏洞的缺席。例如，由于第二漏洞不再存在，先前与第二漏洞相关的变量OOO_List[1].Begin不再被要求。由于漏洞的右边缘由变量OOO_List[1].End表示，先前与该结尾第一漏洞相关的变量OOO_List[0].End保留一个作废的值。因此，由变量OOO_List[1].End表示的数值现在被复制到变量OOO_List[0].End。最后，由于不再有第二漏洞，变量OOO_List[1].End被移除。相应地，三个相邻的TCP分段704、706及708具有相应的变量OOO_List[0].Begin及OOO_List[0].End，其中变量OOO_List[0].Begin表示在先乱序TCP分段704的最左侧部分，变量OOO_List[0].End表示乱序TCP分段708的最右侧部分。从另一方面，三个相邻的TCP分段704、706及708可被认为是一个具有相应的变量OOO_List[0].Begin及OOO_List[0].End的单独的乱序TCP分段710。虽然新的分段能部分地或完全地交叠一个或多个在先接收的分段，为了简化这里举例说明。然而，这种方案被包含于下面的伪码中。

根据本发明的一个实施例，直通式无线网络处理器可如所希望地管理TCP接收窗口内的许多漏洞。直通式无线网络处理器支持的漏洞的最佳数量依赖于如与一个特定的连接相关的典型的下降出错位组（drop syndrome）。例如，该连接可为WAN连接，其中该下降主要由于大窗口内路由拥挤而潜在地产生很多漏洞。该连接还可为LAN连接，该LAN连接具有例如单独下降应有的切换，及在这种

情况下，逐条链路控制或拥挤避免会花掉一些时间来开始机能。

由于如漏洞数量及漏洞中分段或帧的数量这样的下降的类型可以变化，可使用可编程的漏洞数量。每一个漏洞要求可被用于保持其开始及结尾点的状态变量。在直通式无线网络处理器支持TCP之上的上层协议（ULP）的情况下，每一个洞要求更多的状态变量。一旦一个漏洞被堵上，上层协议的特别动作会被要求，且接收器获得更多关于由TCP控制板中上层协议期望的行为的信息。例如，该上层协议被通知消息的完成。然而，在这种情况下，数据及控制板间的基本的间隔会允许直通式无线网络处理器完成实质上没有数据的缓冲或没有乱序TCP分段的缓冲的布置。

在本发明的实施例中，数据的布置通过将TCP序列空间映射至缓冲区空间而被执行。在这个方面，该缓冲区空间可被看作为一个线性空间。相应地，该缓冲区的第一个字节可相应于TCP序列空间内的特定的TCP序列值。因此，当与第一字节的序列值比较时，TCP分段内的其它字节被放置于缓冲区的一个偏移上，相应于其TCP序列空间内的 Δ （delta）或偏移。换句话说，TCP分段或TCP帧的第一字节的开始地址或位置可被作为一个参考点而操作，且任何其它字节可作为从TCP序列空间内的第一字节的偏移而被定位。在本发明的另一方面，存在有高层协议特别传输，其被用于将TCP序列数量的一定范围映射至缓冲空间内的特别缓冲区域。然而，这样不会改变本发明的各个方面所揭示的基本映射。

图7B为根据本发明的实施例的示例性的更新变量的结构示意图。参照图7B，其示出了有序分段752、新的乱序分段754、乱序分段756及在先乱序分段758。同样示于图7B，缓冲区空间762为有序分段752a及新的乱序分段754a。该新的乱序分段754a跨越第一缓冲区（i）及连续的缓冲区（i+1）。缓冲区空间762具有缓冲区长度L并被分隔入缓冲区（i）、缓冲区（i+1）、…、缓冲区（n）。缓冲区空间762具有一个基本地址（Phy_addr），例如，其位于偏移量为0处。该基本地址

可为任何物理地址，且为了方便，可被表示为偏移量0。在其当前状态下，有序TCP分段被配置于在基本地址开始，偏移量0处的缓冲区空间762。变量Nxt_addr指示下一个地址，在此数据被配置于该缓冲区空间762内，且在这个例子中，该变量为超出有序分段752a的结尾一个字节。根据本发明的一个实施例，接收的乱序TCP分段于缓冲区空间762内跨越多个缓冲区。例如，接收的乱序TCP分段可跨越缓冲区(i)及缓冲区(i+1)。该缓冲区长度至少等于TCP窗口尺寸。

参照图7B，为了反映变化，新接收的乱序分段754变成乱序分段列表内的第一个成员，且其相关的变量被相应地更新。以一个有些相似的方法，与乱序分段756相关的变量也被更新，以反映出乱序分段756为乱序分段列表内的第二个成员。在先地，乱序分段756为乱序分段列表内的第一TCP分段。最后，乱序TCP分段758成为乱序分段列表内的第三个成员，且其相关变量被相应地更新。在先地，乱序TCP分段758为乱序段列表内的第二TCP分段。下表举例说明了新接收的乱序TCP分段754、乱序TCP分段756及乱序TCP分段758的先前的及更新的变量。

变量	先前值	当前值
新的OOO分段 754	- -	OOO_List[0].Begin OOO_List[0].End
OOO分段 756	OOO_List[0].Begin OOO_List[0].End	OOO_List[1].Begin OOO_List[1].End
OOO分段: 758	OOO_List[1].Begin OOO_List[1].End	OOO_List[2].Begin OOO_List[2].End

图8为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间及配置前的示例性变量及缓冲区的状态内新接收的有序TCP分段的结构示意图。当数据被按顺序接收时，示例性的被用控制TCP状态及缓冲区空间的变量被示于图8中。这些变量被用于下面提出的伪码中。参照图8，其示出了有序TCP分段802、新的有序TCP分段804、乱序分段806、乱序分段808及TCP窗

口810, 所有这些均位于TCP序列空间内。新接收的有序TCP分段80处于与有序分段802相邻, 堵上了第一漏洞的最左边缘, 且与TCP窗口810的左边缘在一行上。相应地, 在本发明的一个方面中, 乱序TCP分段806的最左边缘可被表示为变量`OOO_List[0].Begin`及乱序TCP分段806的最右边缘可被表示为变量`OOO_List[0].End`。以一个有些相似的方法, 在先乱序TCP分段808的最左边缘可被表示为变量`OOO_List[1].Begin`及乱序TCP分段808的最右边缘可被表示为变量`OOO_List[1].End`。

图8也示出了相应的缓冲区空间812及有序TCP分段802a。该缓冲区空间812的缓冲区长度为L及该缓冲区空间可被分段成缓冲区(i)、缓冲区(i+1)、...、缓冲区(n)。有序TCP分段802a相应于TCP序列空间内的有序TCP分段802。缓冲区空间812具有一个基本地址(`Phy_addr`), 例如, 其位于偏移量为0处。该基本地址可为任何物理地址, 且为了方便, 可被表示为偏移量0。在其当前状态下, 有序TCP分段被配置于在基本地址开始, 偏移量0处的缓冲区空间812。变量`Nxt_addr`指示下一个地址, 在此数据被配置于该缓冲区空间812内, 且在这个例子中, 该变量为超出有序分段802a的结尾的一个字节。根据本发明的一个实施例, 接收的乱序TCP分段于缓冲区空间812内跨越出多个缓冲区。例如, 接收的乱序TCP分段可跨越出缓冲区(i)及缓冲区(i+1)。该缓冲区长度至少等于TCP窗口尺寸。

图9为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间内及描述随后配置的示例性变量及缓冲区的各种状态的新接收的有序TCP分段的结构示意图。当数据被按顺序接收时, 图8中用于控制TCP状态及缓冲区空间的示范性的变量也被描述于图9中。

参照图9, 其示出了有序TCP分段902、新的有序TCP分段904及TCP窗口910, 所有的这些均位于TCP序列空间内。该新接收的有序TCP分段904与有序TCP分段

902相邻放置，堵上了第一漏洞的最左边缘，且与TCP窗口910的左边缘在一行上。

在本发明的一个方面中，由于新接收的有序TCP分段904是有序的，该新接收的有序TCP分段904被与有序TCP分段902相邻放置。因此，TCP窗口910被调整以使新的有序TCP分段904的最右边缘成于TCP窗口910的最左边缘。在这个方面，该窗口从有序TCP分段902的最右边缘滑动至有序TCP分段904的最右边缘。

同样如图9，其示出了缓冲区长度为L的相应的缓冲区空间912、有序TCP分段902a及新接收的有序TCP分段904a。该有序TCP分段902a相应于有序TCP分段902，该TCP分段904a相应于新接收的有序TCP分段904。在这个方面，有序TCP分段902a可被认为在先配置的有序TCP分段。缓冲区空间912具有一个基本地址(Phy_addr)，其位于偏移量为0处。在其当前状态下，新接收的有序TCP分段904a可与在先配置的开始于图8所示的Nxt_addr位置处的有序TCP分段902a的相邻配置。由于新接收的有序TCP分段904a相邻有序TCP分段902a放置，那么，在这种情况下，在数据被配置之后，下一个地址(Nxt_addr)及该缓冲区偏移被定位于新接收的有序TCP分段904a的最右边缘上。下一个被接收的有序分段将被配置于Nxt_addr，其在这种情况下，其指示缓冲区空间912内的有序TCP分段的结尾。在本发明的一个方面中，新的有序TCP分段904a跨越包括缓冲区空间912内的缓冲区(i)及缓冲区(i+1)的多个缓冲区。在其它的例子中，新的有序TCP分段904a可以跨越多个TCP分段。既然缓冲区(i)是完全充满的，缓冲区(i)对于ULP、本申请或任何提供该缓冲区开始的人来说均是完整的。

图10为根据本发明的实施例的用于与乱序TCP分段或帧的数据布置有关的TCP序列空间及其缓冲内的新接收的有序TCP分段的处理的结构示意图。参照图10，其示出了有序TCP分段1002及新的乱序TCP分段1004。缓冲区空间1012内的有序TCP分段1002a映射至TCP分段空间内的有序TCP分段1002。以一个有些相似的方法，缓冲区空间1012内的新接收的乱序TCP分段1004a映射至TCP分段空间

内的乱序TCP分段1004。当数据被按顺序接收时，被用于控制TCP状态及缓冲区空间的图8所示的示例性变量也同样被示于图10中。参照图10，其示出了有序TCP分段1002、新的乱序TCP分段1004，所有这些被置于一个TCP序列空间内。新接收的乱序TCP分段1004位于第一漏洞的一部分内。

图10还包括缓冲区长度为L的相应的缓冲区空间1012、有序TCP分段1002a及新接收的乱序TCP分段1004a。该有序TCP分段1002a相应于有序TCP分段1002，及该新接收的有序TCP分段1004a相应于新接收的有序TCP分段1004。缓冲区空间1012具有一个基本地址（Phy_addr），其位于偏移量为0处。在其当前状态下，新接收的有序TCP分段1004a可被配置于被指示为缓冲区偏移位置的开始处。在这个方面，由于新接收的有序TCP分段不是有序TCP分段，其不会被配置于Nxt_addr。在一个新的有序TCP分段被接收的情况下，其与开始于Nxt_addr的有序TCP分段1002a的相邻配置。作为代代替，其可被配置于等于TCP空间内的数据的偏移内。这个TCP序列空间 Δ 可被用于计算缓冲区内的偏移。

新接收的有序TCP分段904a可与在先配置的开始于图8所示的Nxt_addr位置处的有序TCP分段902a的相邻配置。由于新接收的有序TCP分段904a相邻有序TCP分段902a放置，那么，在这种情况下，在数据被配置之后，下一个地址（Nxt_addr）及该缓冲区偏移被定位于新接收的有序TCP分段904a的最右边缘上。下一个被接收的有序分段将被配置于Nxt_addr，其在这种情况下，其指示缓冲区空间912内的有序TCP分段的结尾。在本发明的一个方面中，新的有序TCP分段904a跨越包括缓冲区空间912内的缓冲区（i）及缓冲区（i+1）的多个缓冲区。在其它例子中，新的有序TCP分段904a可以跨越多个TCP分段。既然缓冲区（i）是完全充满的，缓冲区（i）对于ULP、本申请或任何提供该缓冲区开始的人来说均是完整的。

根据本发明的一个实施例，可被假定主机提供了充足的缓冲区，以使缓冲区

的总长度总是大于或等于TCP窗口尺寸。然而，在由TCP、ULP或本申请没有的缓冲器不够大的缓冲区的例子中，那么接收器可决定不接收没有缓冲区的乱序TCP分段。在这个方面，该接收器会丢弃那么乱序分段。接收器的另一个选择是使用通过其驱动器、TCP或任何其它机构提供的其它缓冲区，并作为一个数据暂时的存储，并继而将其复制至配置的缓冲区。相应地，该处理可发生于由TCP、ULP、驱动器、任何其它机构所配置的缓冲区内或任何其结合内。例如，布置可以开始于由该驱动器提供的暂时的缓冲区内，且在ULP服从的缓冲区内继续。接下来，该放置于由驱动器提供的缓冲区内文本可被硬件或任何软件实体复制，该硬件或任何软件实体如驱动器、通信栈和/或指示的缓冲区的中间设备。

根据本发明的另一个实施例，下面的伪码举例说明了一个用于将数据从TCP分段移动至缓冲区列表的示例性的处理。为了简化，覆盖处理PUSH位或“当前缓冲区列表内无空间”的伪码被省略。然而，伪码的省略不会限制本发明。同样，该伪码假定一个没有接收的TCP窗口更新的静态的TCP情形。然而，增加代码而不会偏离本发明的各种实施例。另外，TCP序列数量空间的回绕不包括于伪码内。该伪码同样假定当当前帧在被处理时，没有其它帧被处理而。在这个方面，伪码假定中断是无效的。该伪码如下所示：

```

/*整体初始化*/
Number_of_holes = 0;
Number_of_holes = 0;
OOO_Data {
  Begin Int32;
  ULP_Event_present Boolean; /*在OOO分段内的ULP事件的出现的变量保持标识符*/
  ULP_Event_TCP_Sequence_number Int32; /*ULP事件的TCP序列数量*/
} OOO_list[i]; /*这列结构保持接收O-O-O的分段的TCP序列数量的标识符。创

```

造最大数的量被支持漏洞的列表及将所有变量初始化为0*/

/*RCV_NXT为期望被接收的下一个字节的TCP序列数量*/

1. /*TCP序列数量（TCP有效载荷的第一个字节的TCP序列#为First_TCP_Seq，最后一个字节的TCP序列#为End_TCP_Seq）被检测出在TCV窗口的范围内（在RCV_NXT与RCV_NXT+ RCV_Window之间）*/

1A. /*如果窗口（失效的或复制的）外边的帧丢弃该帧*/

```
if ((First_TCP_Seq > (RCV_NXT + RCV_window)) || (Last_TCP_Seq <
RCV_NXT)) then drop_frame();
```

1B. /*如果一些字节之前已被接收，忽略复制的字节*/

```
if First_TCP_Seq < RCV_NXT then First_TCP_Seq = RCV_NXT;
```

1C. /*在一些字节在窗口之外的情况下，该接收器可丢弃整个分段。这里该接收器只丢弃禁用的字节*/

```
if Last_TCP_Seq > (RCV_NXT + RCV_window) then Last_TCP_Seq =
(RCV_NXT + RCV_window);
```

1D. /*在最后的数为O-O-O的情况下；进行计算。其可为一个新的漏洞或堵上一个漏洞或与一个漏洞相邻*/

```
if First_TCP_Seq > RCV_NXT then Number_of_holes = Hole_accounting();
```

1E. /*在数据为有序的情况下；其为堵上一个漏洞*/

```
if (First_TCP_Seq == RCV_NXT) RCV_NXT = Last_TCP_Seq + 1;
```

/*RCV_Window的更新在这个编码范围之外*/

```
{
```

```
if ((Number_of_Holes > 0) && (Last_TCP_Seq + 1 ==
OOO_list[1].Begin)) then
```

```
{
```

/*移除元素I，接下来的元素的索引减1*/

```
Remove_OOO_list_element(1);
```

```
Number_of_Holes--;
```

```
    }
}
```

2. /*在缓冲区列表中找到最边的入口。放置对于有序、乱序或部分漏洞问题是不在意的*/

```
Segment_Length = Last_TCP_Seq - First_TCP_Seq;
```

/*TCP_Delta支持TCP序列数量与帧的位置第一字节的差。其还为缓冲区空间至应该被用于存储的第一字节的距离*/

```
TCP_Delta = (First_TCP_Seq - RCV_NXT) /*Delta来自TCP空间内的锚点*/
```

```
i=0; /*被用于动态地指示相应于RCV_NXT的缓冲区*/
```

/*Host_Buffer_List.Buffer[0]已经被使用。需要示出有多少是左边的*/

```
if (TCP_Delta < (Host_Buffer_List.Buffer[0].length -
```

```
(Host_Buffer_List.Buffer[0].NXT_Addr -
```

```
Host_Buffer_List.Buffer[0].Phy_Addr)))
```

```
{
```

```
    Buffer_Offset = TCP_Delta + (Host_Buffer_List.Buffer[0].NXT_Addr -
```

```
Host_Buffer_List.Buffer[0].Phy_Addr);
```

```
}
```

```
else
```

```
{
```

```
    Buffer_Offset = TCP_Delta - (Host_Buffer_List.Buffer[0].length -
```

```
(Host_Buffer_List.Buffer[0].NXT_Addr -
```

```
Host_Buffer_List.Buffer[0].Phy_Addr));
```

```
i = 1;
```

```
    Do while { Buffer_Offset - (Host_Buffer_List.Buffer[i].length > 0)
```

```
{
```

```
        Buffer_Offset -= Host_Buffer_List.Buffer[i].length;
```



```

i++;
}
i--;
}

```

/*第一次缓冲后,该变量i指向第i个缓冲区,其为数据配置的开始处。Buffer_Offset支持该偏移进入这个缓冲区*/

3. /*DMA进入第一个缓冲区, DMA_Data的语法 (从地址至地址, 长度)

第一个缓冲区是特别的, 因为放置可以开始于缓冲区内侧的任何地方*/

```

DMA_Data (@First_TCP_Seq, Host_Buffer_List.Buffer[i].Phy_Address+
Buffer_Offset, Min((Host_Buffer_List.Buffer[i].length-Buffer_Offset),
Segment_Length));

```

5. /*完成ULP第一缓冲区如果数据是有序的正好在缓冲区内的第一个可得字节, 及缓冲区目前是充分的, 及其内无漏洞及所有接续它的缓冲区 (如相关于低TCP序列数量的数据) 对于ULP已经是完成的。注意第一缓冲区包括TCP窗口如有序TCP数据的左边缘*/

```

if (l == 0 &&
(Buffer_Offset == Host_Buffer_List.Buffer[0].NXT_Addr) &&
Segment_Length >= (Host_Buffer_List.Buffer[0].length -
(Host_Buffer_List.Buffer[0].NXT_Addr -
Host_Buffer_List.Buffer[0].Phy_Addr)) &&
(OOO_list[0].Begin > 0) &&
(OOO_list[0].Begin > (First_TCP_Seq +
Min((Host_Buffer_List.Buffer[0].length-Buffer_Offset), Segment_Length))
{
Indicate_buffer_to_ULP(Host_Buffer_List.Buffer[0]);
remove_buffer_from_list (Host_Buffer_List.Buffer[0]); /* following buffer is

```

now the first */

} /*接下来的缓冲区为当前的第一个*/

6. /*保持TCP变量。为简单仅处理RCV_NXT。如果数据是有序的且一个缓冲区可获得，其递增*/

```
if( TCP_Delta == 0) RCV_NXT += Segment_Length;
```

7. /*处理下一个缓冲区（为简化跳过对缓冲区可用性的检测）。引入临时变量Bytes_to_DMA及Start_TCP_Seq。注意i指向接下来的缓冲区的第一个缓冲区数据已被存储。在有序接收的情况中，如果第一缓冲区（i=0）已被完成且有更多的字节要被存储，i=0但指向原来的第二个缓冲区*/

```
Bytes_to_DMA = Segment_Length - Min((Host_Buffer_List.Buffer[i].length-
Buffer_Offset), Segment_Length);/*已被DMA移至第一缓冲区的字节*/
```

```
Start_TCP_Seq = First_TCP_Seq + Min((Host_Buffer_List.Buffer[i].length-
Buffer_Offset), Segment_Length);
```

8. /*如果需要，DMA进入下一个缓冲区*/

```
Do while ( Bytes_to_DMA > 0)
```

```
{
```

```
    DMA data (@Start_TCP_Seq,
```

```
Host_Buffer_List.Buffer[i].Phy_Address,
```

```
Min((Host_Buffer_List.Buffer[i].length, Bytes_to_DMA)
```

```
    Bytes_to_DMA -= Min((Host_Buffer_List.Buffer[i].length,
```

```
Bytes_to_DMA;
```

```
    Start_TCP_Seq += Min((Host_Buffer_List.Buffer[i].length,
```

```
Bytes_to_DMA
```

```
    i++;
```

```
    if i > max_buffers then no_more_buffers(); /*退出无可得的缓冲区*/
```

```
}
```

9. /*计算窗口内接收的乱序TCP数据的规则。建构O-O-O数据管理结构，返回漏

洞的数量*/

int32

Hole_accounting()

{

/*检测无复制的O-O-O数据。数据可于缓冲区内重写(当内容通过要被识别的TCP确认时)或保留DMA循环可以写入缓冲区正好该接收数据没有被先前接收*/

l = 0;

Do while (Number_of_Holes > 1)

{

/*为了伪码的简化,一个TCP分段交叠超过一个漏洞的情况此处没被讨论。然而,其为下述编码的简单推论*/

/*如果新漏洞在元素I的左侧,分配空间对其进行管理*/

If ((Last_TCP_Seq+1) < OOO_list[l].Begin) then

{

/*在原始元素I之前插入元素I,将原始I+1及子序列元素1上向推*/

Insert_OOO_list_element(l);

OOO_list[l].Begin = First_TCP_Seq;

OOO_list[l].End = Last_TCP_Seq;

return (Number_of_Holes++);

}

If ((Last_TCP_Seq+1) == OOO_list[l].Begin) then OOO_list[l].Begin =

First_TCP_Seq;

Else If ((First_TCP_Seq < (OOO_list[l].Begin) && (Last_TCP_Seq >

(OOO_list[l].Begin) && (Last_TCP_Seq <= (OOO_list[l].End))) then

Last_TCP_seq = (OOO_list[l].Begin

Else If ((First_TCP_Seq >= (OOO_list[l].Begin) && (Last_TCP_Seq <=

(OOO_list[l].End)) then drop_frame();

Else If ((First_TCP_Seq >= (OOO_list[l].Begin) && (First_TCP_Seq <

```

(OOO_list[I].End) && (Last_TCP_Seq > (OOO_list[I].End) then
First_TCP_seq = (OOO_list[I].End
/*这个数据与存在的漏洞的右边相邻？移动漏洞的边缘。现在漏洞被堵上
吗？*/
    If ((First_TCP_Seq == OOO_list[I].End+1)      then OOO_list[I].End
= Last_TCP_Seq;
    If ((OOO_list[I].End +1) == OOO_list[I+1].Begin) then
    {
    OOO_list[I].End = OOO_list[I+1].End;
/*移除元素I+1，（将其变量重新设置为0）接下来的元素的索引减1*/
    Remove_OOO_list_element(I+1);
    return (Number_of_Holes--);
    }
/*漏洞现在从左边被堵上吗？*/
    If (OOO_list[I].Begin == OOO_list[I-1].End +1) then
    {
    OOO_list[I-1].End = OOO_list[I].End;
/*移除元素I，接下来的元素的索引减1*/
    Remove_OOO_list_element(I);
    return (Number_of_Holes--);
    }
/*如果新的漏洞在元素I的左侧，分配空间以管理*/
    If ((First_TCP_Seq+1) > OOO_list[I].End) && (Last_TCP_Seq <
OOO_list[I+1])) then
    {
/*在原始元素I之后插入元素I+1，将原始I+1及子序列元素1上向推*/
    Insert_OOO_list_element(I+1);

```

```
        OOO_list[i+1].Begin = First_TCP_Seq;
        OOO_list[i+1].End = Last_TCP_Seq;
        return (Number_of_Holes++);
    }
    i++;
}
}
```

本发明的另一个实施例提供了一种机器可读存储器，其中存储有计算机程序，该计算机程序具有可于无线系统内提供处理乱序TCP分段的数据的至少一个代码段。该至少一个代码段可通过机器执行，因此造成该机器执行如此处所描述的乱序（OOO）TCP分段的处理。

相应地，本发明可于硬件、软件或硬件与软件的结合中而被实现。本发明可于至少一个计算机系统的集中形式中而被实现，或以一个其不同的元件分散交叉于几个相互连接的计算机系统上的分布式实现。任何类型的计算机系统或其它设备被用于执行此处所述的方式均是适合的。一种典型的硬件与软件的结合为包含有计算机程序通用计算机系统，当被登陆及被执行时，其控制该计算机系统，以使其执行此处所述的方法。

本发明还可被嵌入一个计算机产品中，其包括所有的可执行此处所述方法的特征，当登陆到一个计算机系统中时，这些方法可被执行。本文所述的计算机系统意味着任何表达，以任何语言、代码或国家，一组指令可造成一个具有信息处理能力的系统或者直接地执行一个特别功能，或者以下面的二者或二者之一执行：a) 转换成另一种语言、代码或国家；b) 以不同的物质形式重新产生。

当本发明通过参考某些实施例而被描述时，其会被本领域的普通技术人员所理解的是可作出各种改变及等同的替代而不脱离本发明的范围。另外，可作出很多修改以适合本发明的技术的特别情况或材质而不脱离其范围。因此，其意味着

本发明不会被限制于所揭示的特别实施例,但是本发明可包括落入权利要求的范围内的所有实施例。

通过参考而交叉参考相关的申请/结合

本申请参考、主张优先权及要求系列号为_____ (代理机构案件号为NO.15774US02) 名称为_____美国专利申请的益处。

本申请参考:

系列号为_____ (代理机构案件号为NO.15064US02) 名称为_____的美国专利申请; 及

系列号为_____ (代理机构案件号为NO.1378US02) 名称为_____的美国专利申请。

上面所述的申请通过参考其全文而被结合于此处。

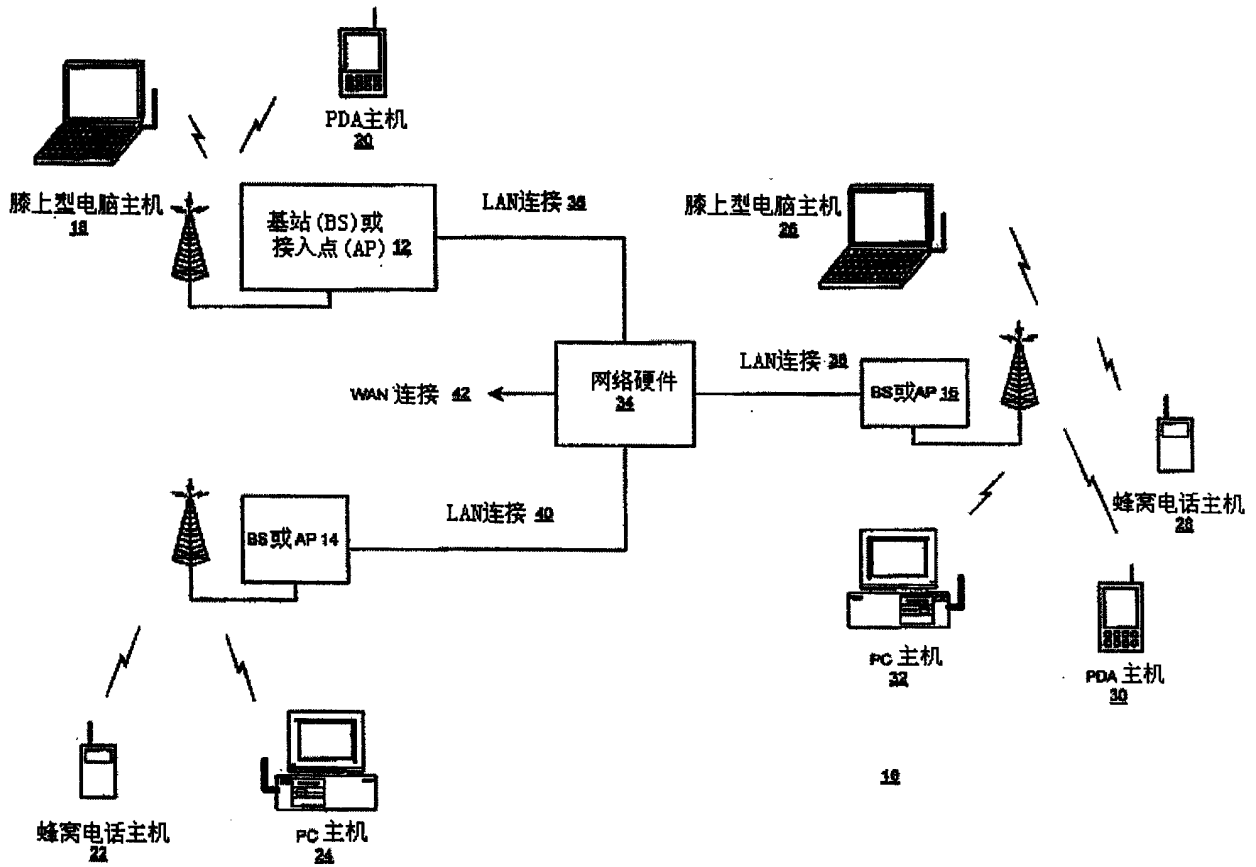


图 1A

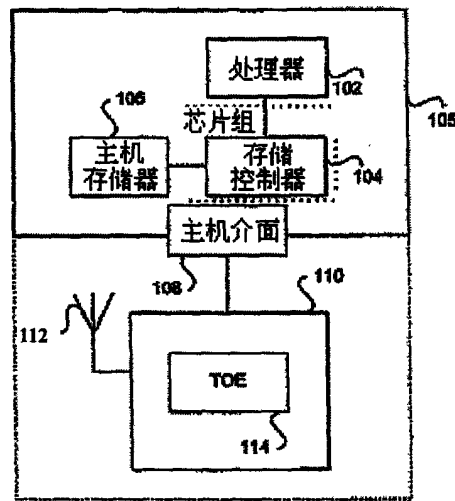


图 1B

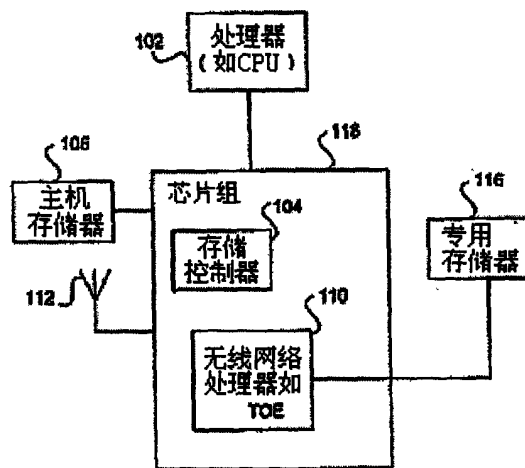


图 1C

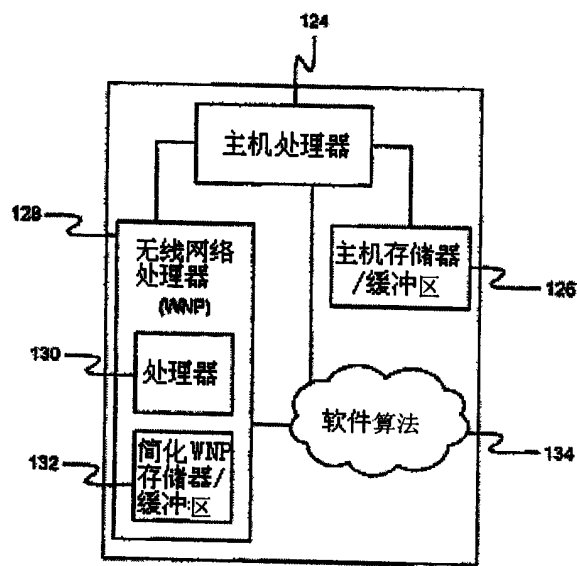


图 1D

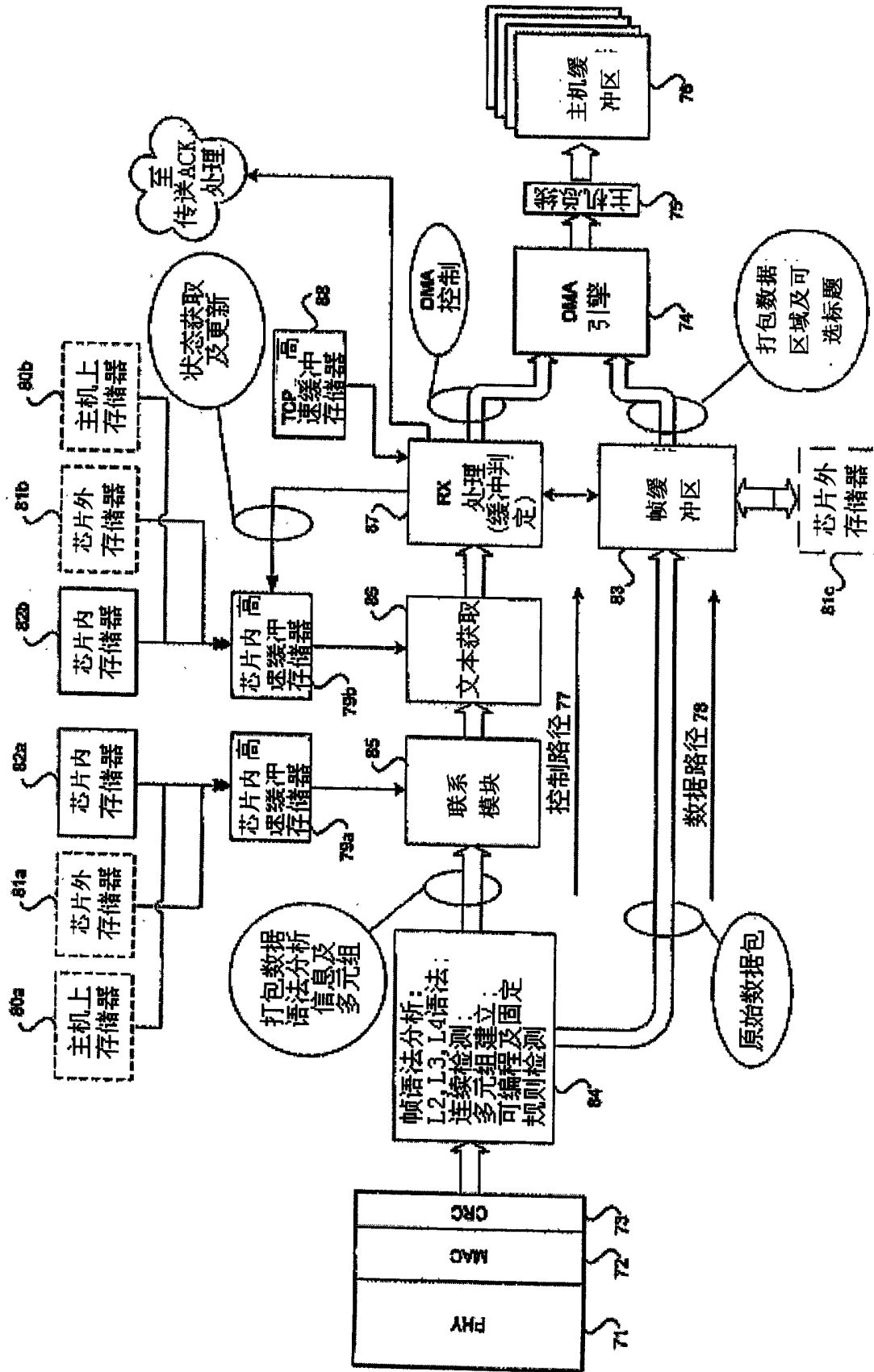


图 1E

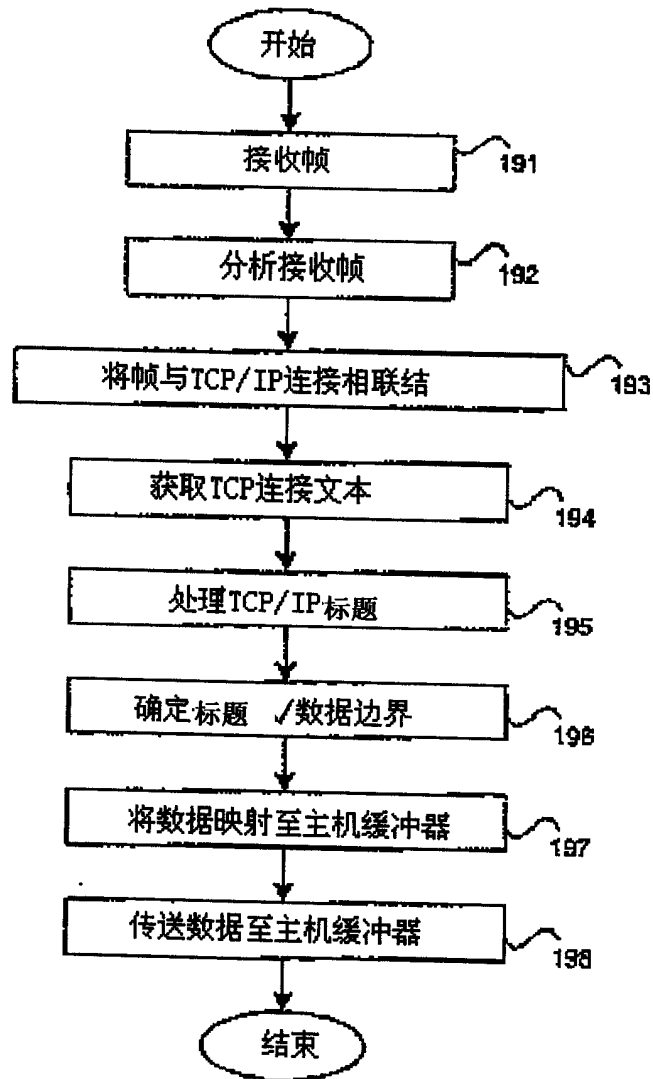


图 1F

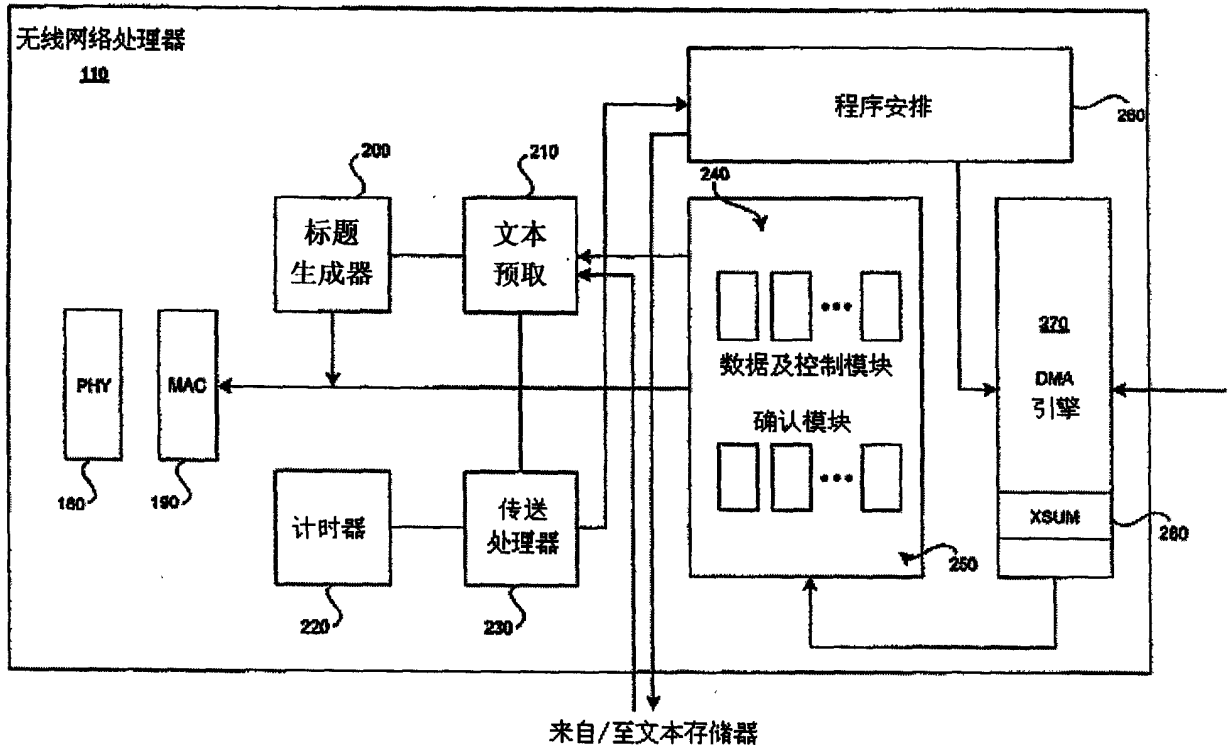


图 1G

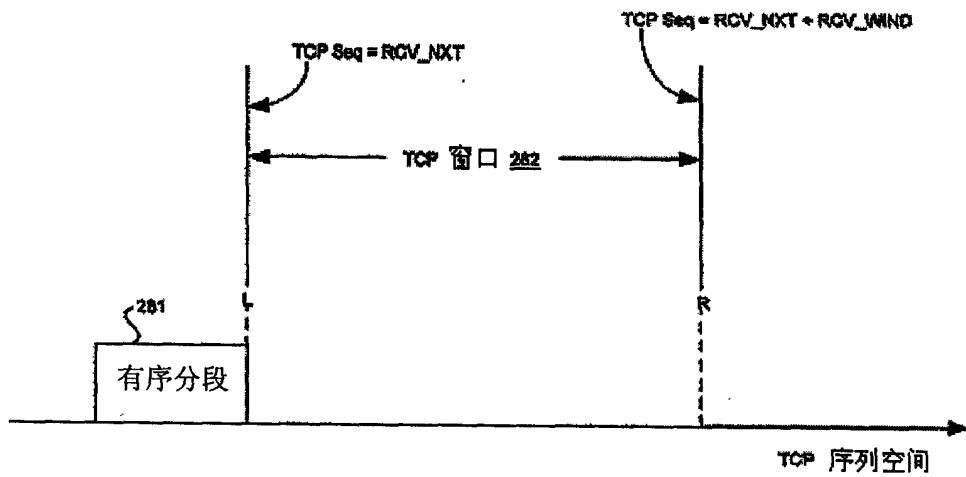


图 2A

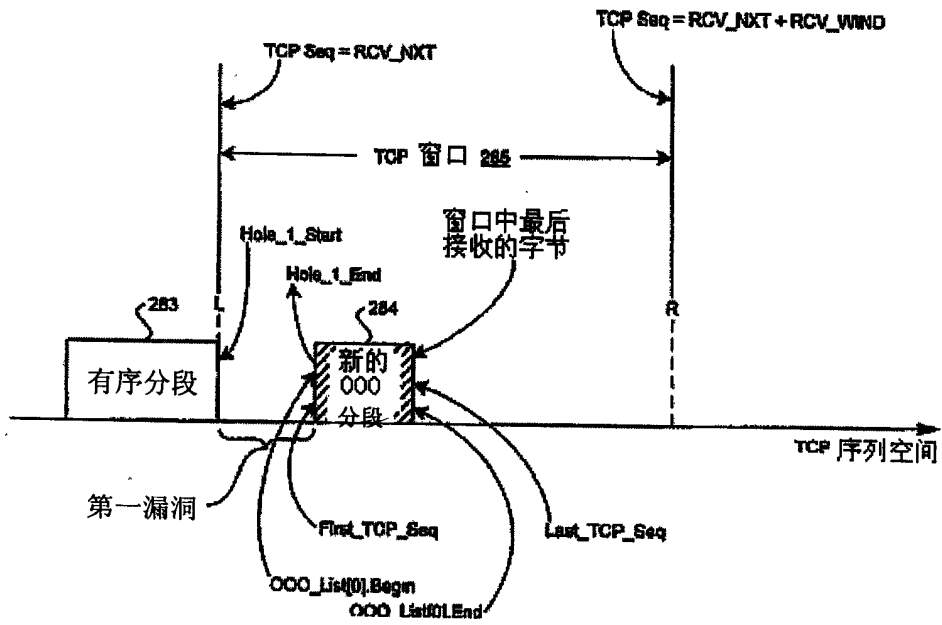


图 2B

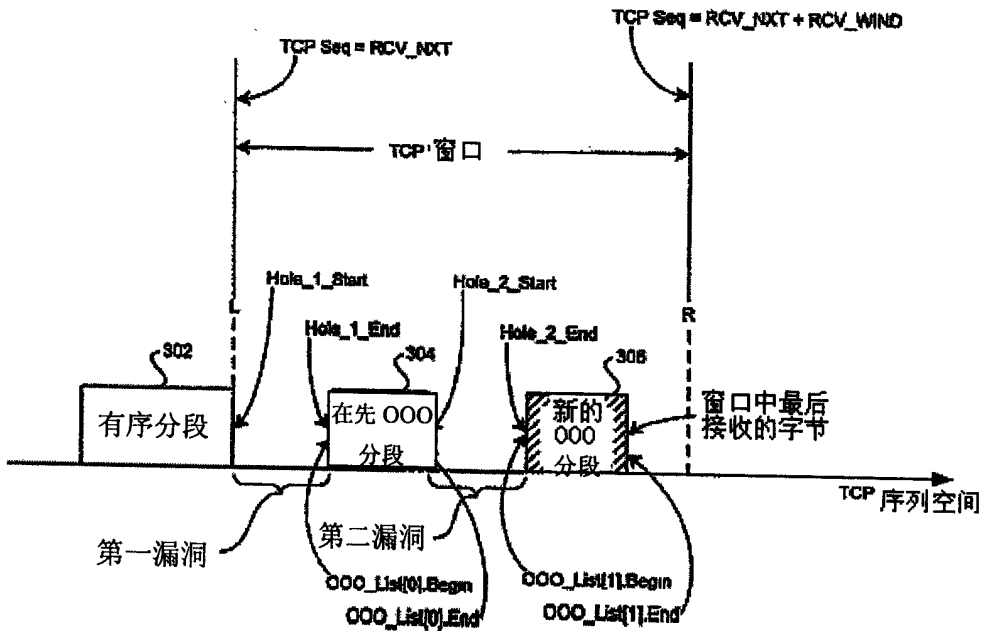


图 3

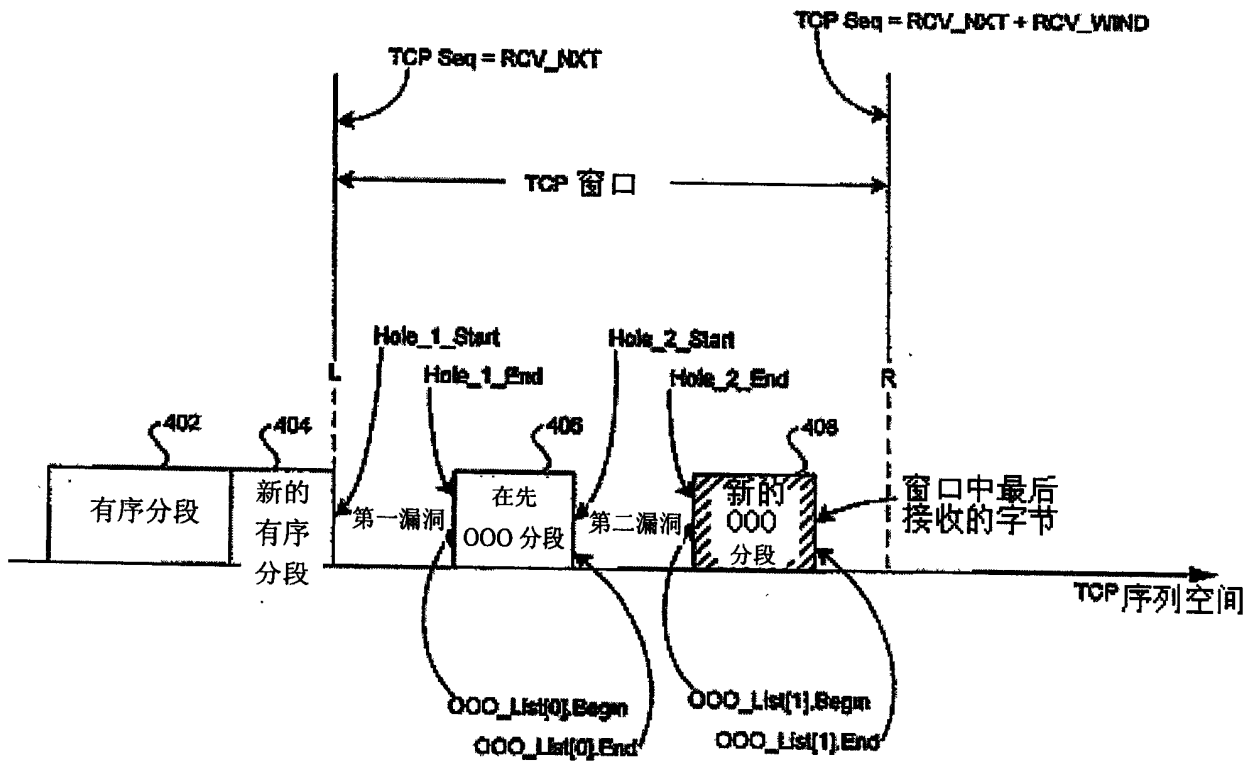


图 4

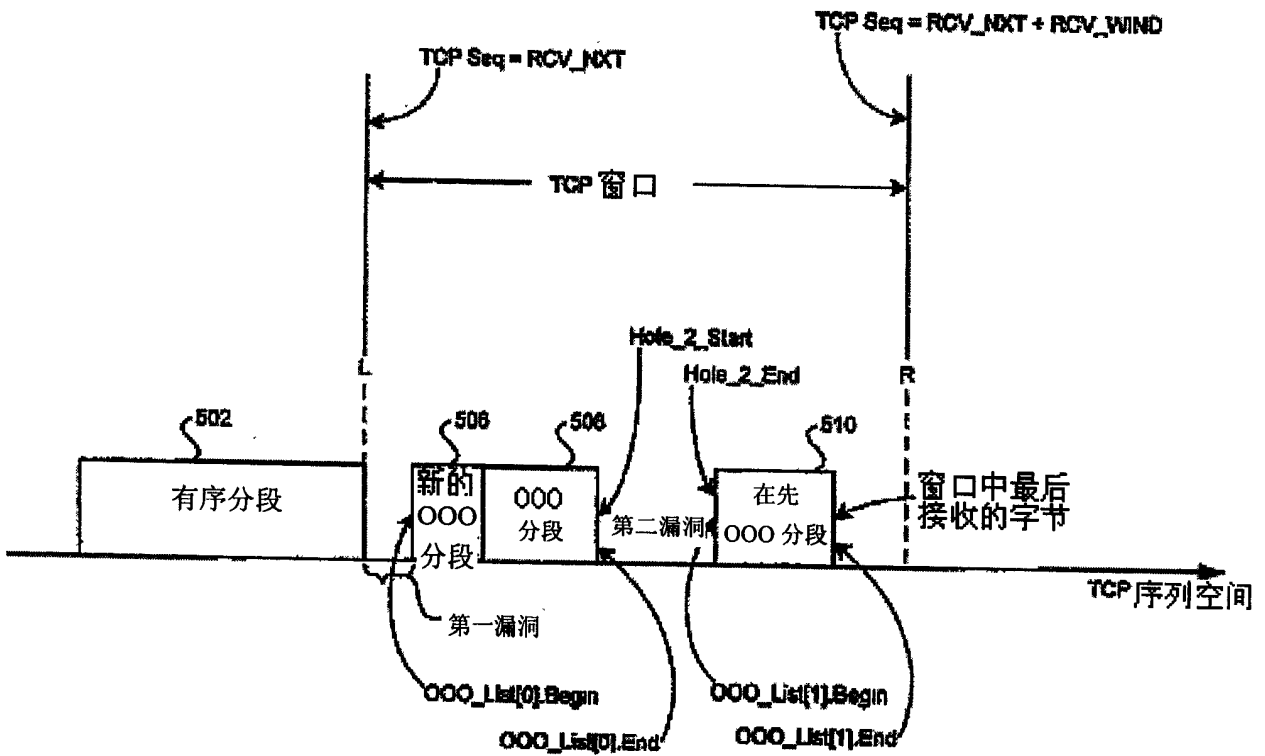


图 5

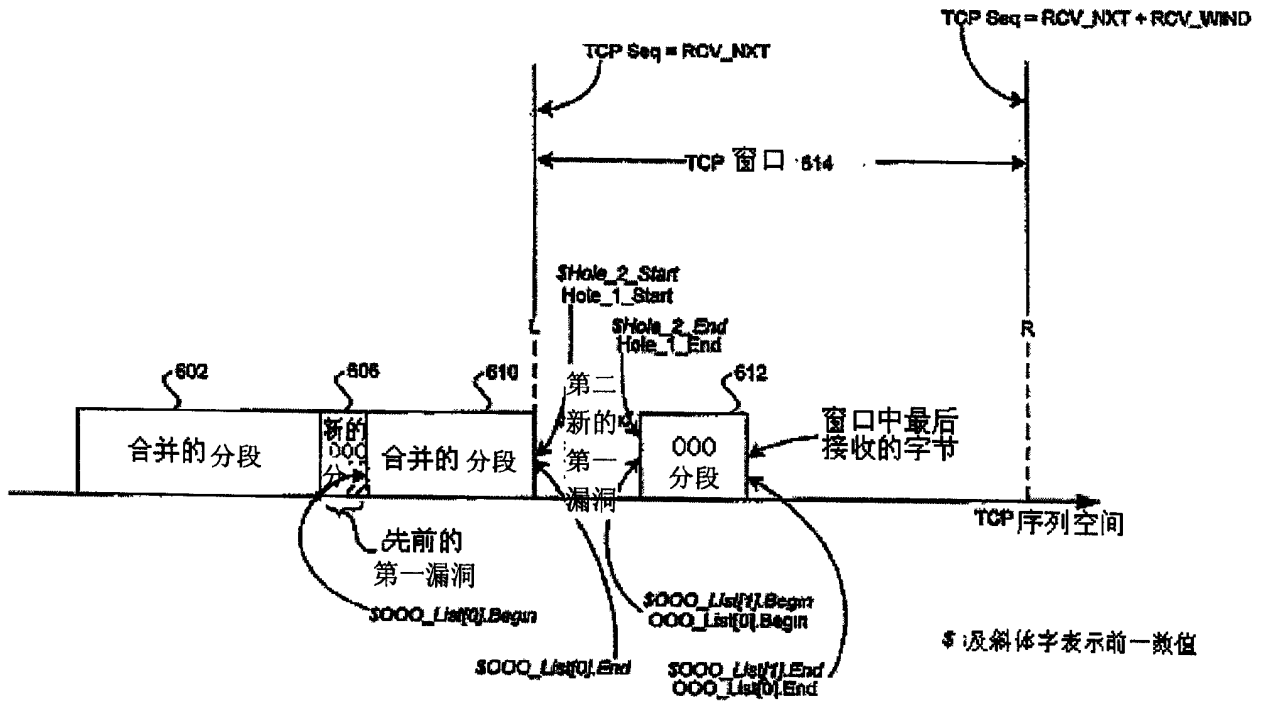


图 6

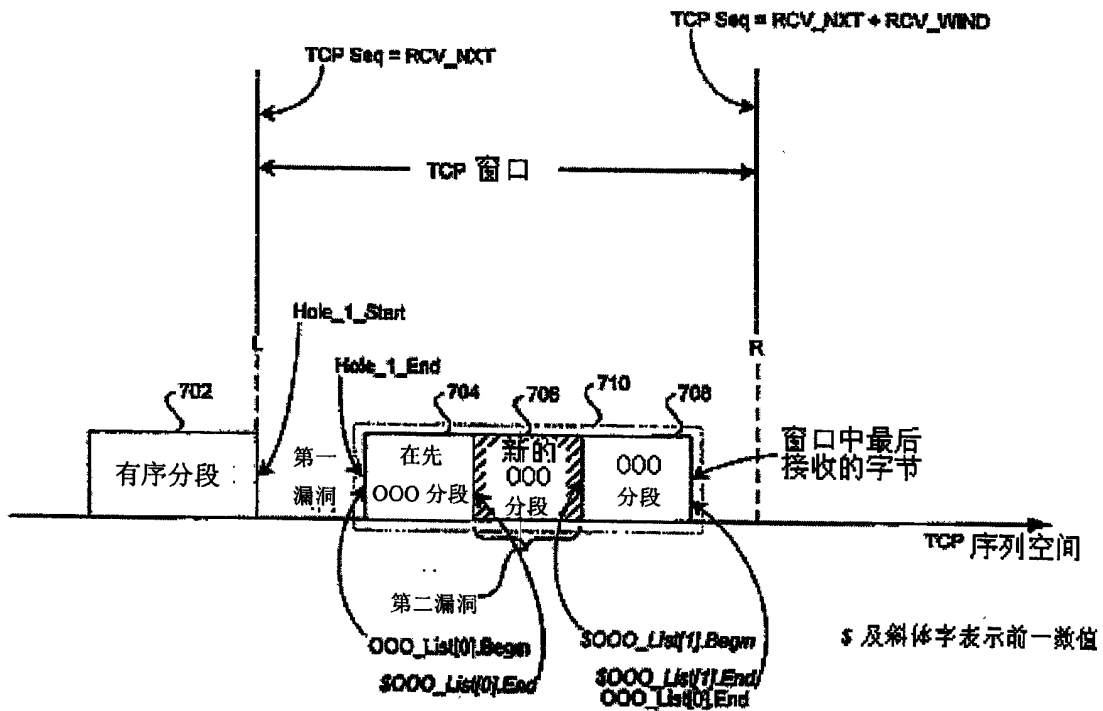


图 7A

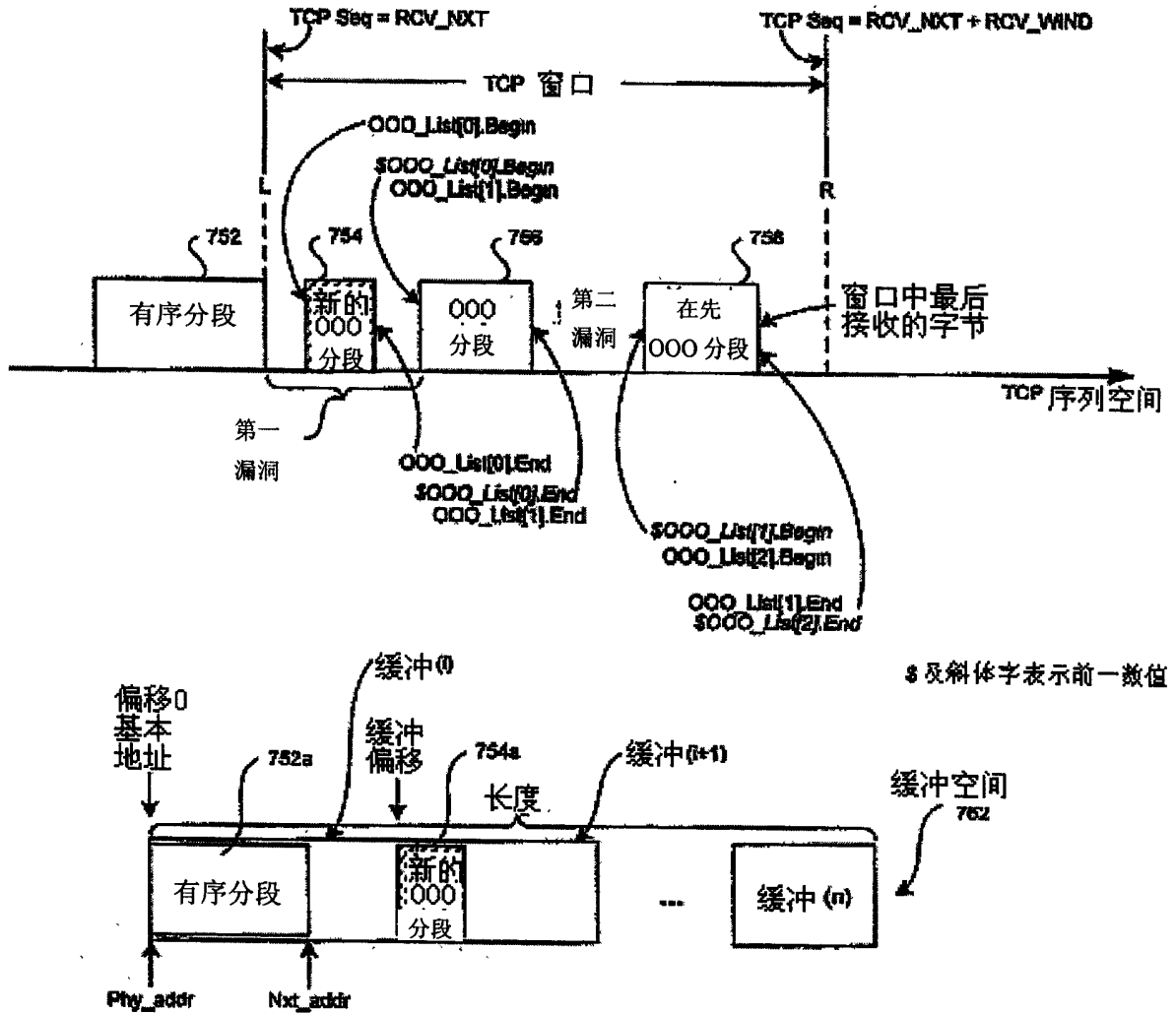


图 7B

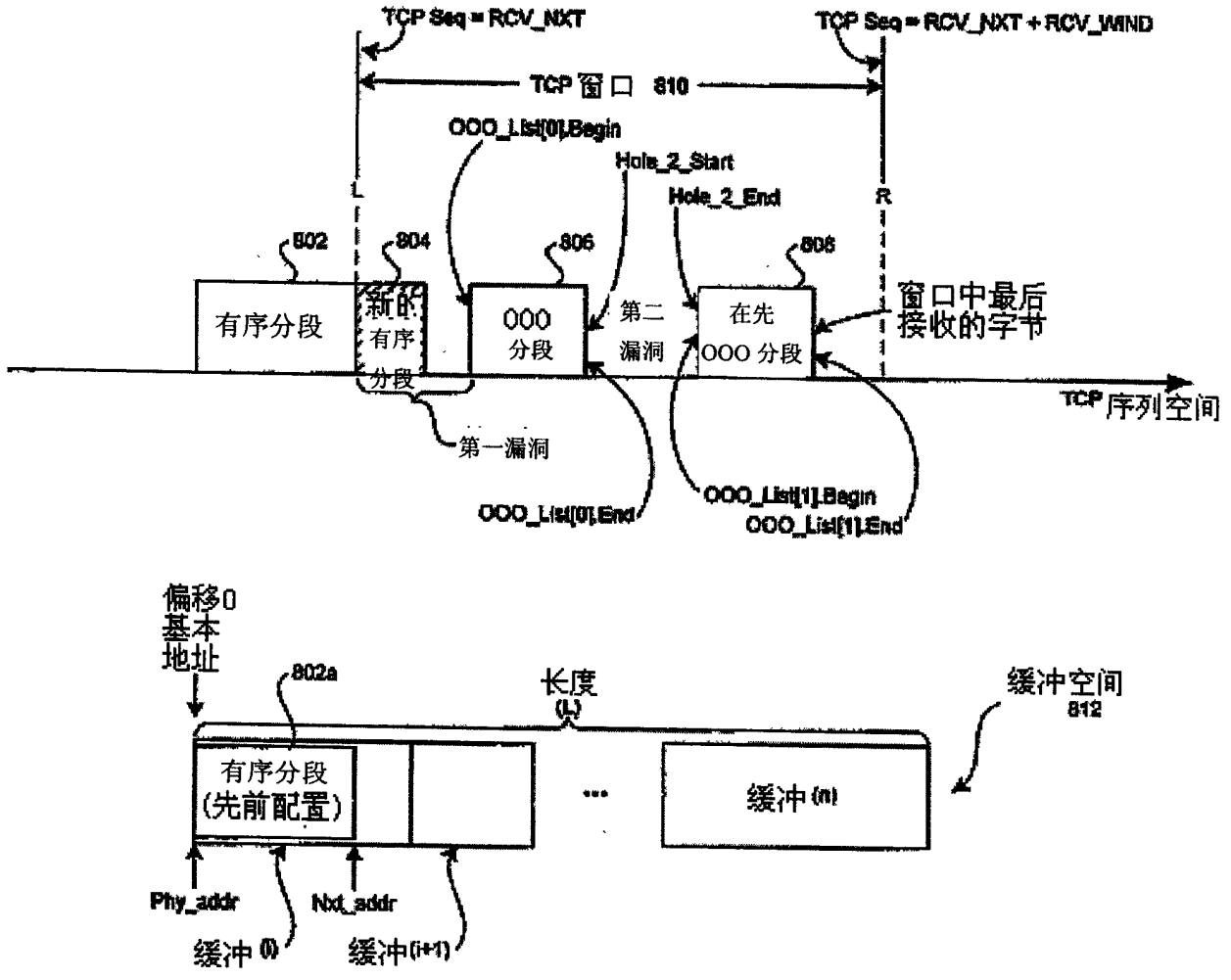


图 8

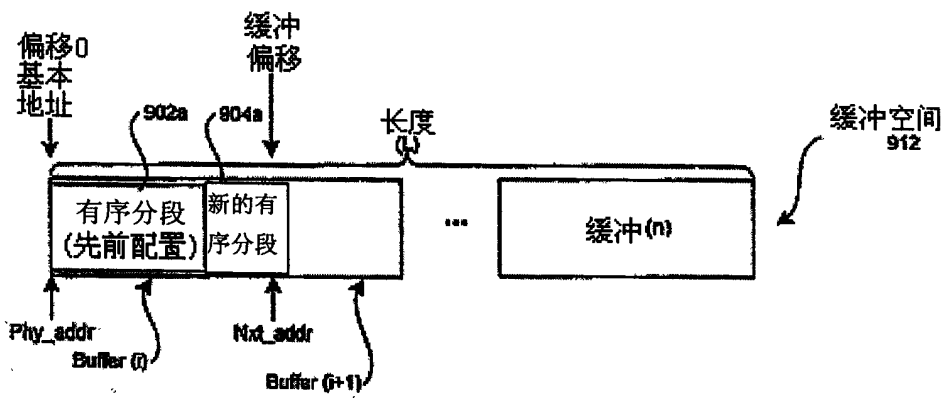
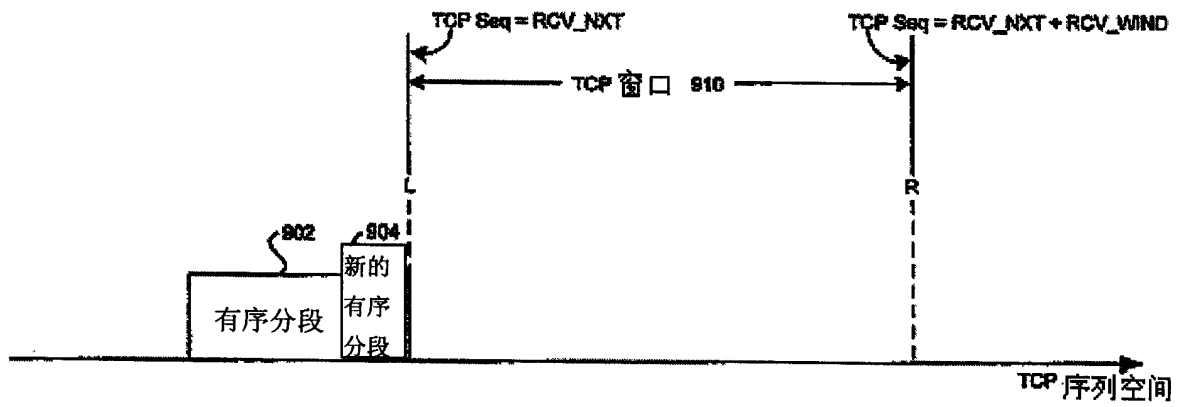


图 9

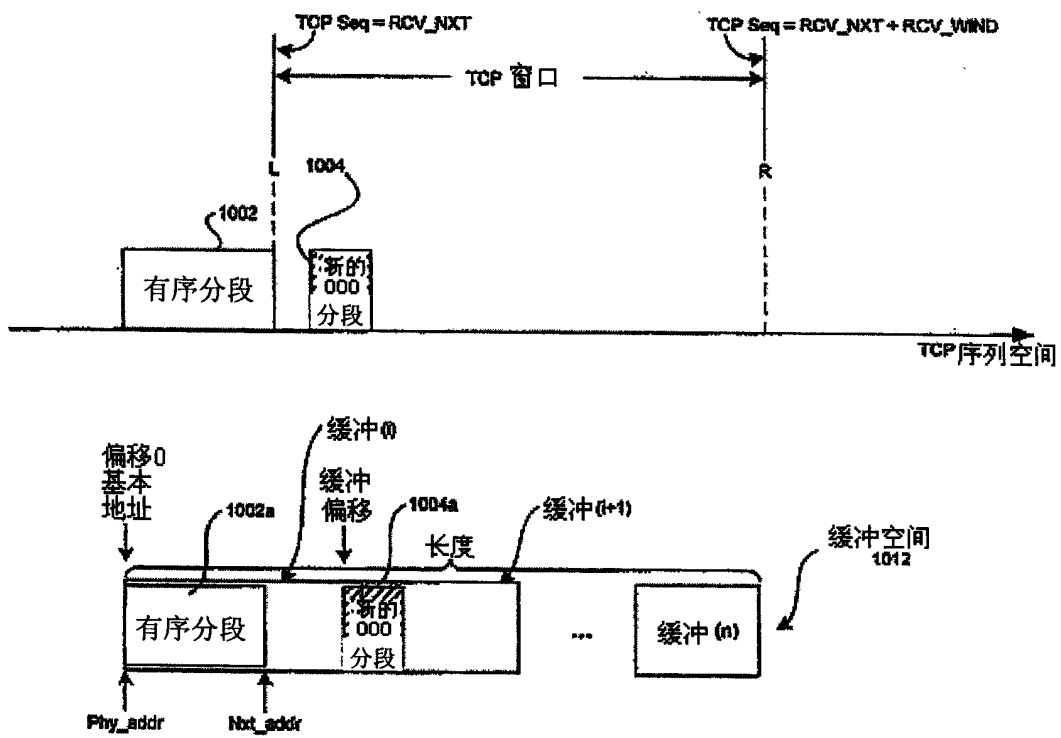


图 10