

多链路通信方法及装置

本申请要求于 2022 年 01 月 23 日提交中国专利局、申请号为 202210076088.X、申请名称为“多链路通信方法及装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中，本申请要求于 2022 年 01 月 28 日提交中国专利局、申请号为 202210109328.1、申请名称为“多链路通信方法及装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及通信技术领域，尤其涉及一种多链路通信方法及装置。

背景技术

下一代无线局域网（wireless local area network, WLAN）标准朝着不断提高吞吐率的方向进行发展和演进，WLAN 系统标准主要在电气与电子工程师协会（institute of electrical and electronics engineers, IEEE）802.11 标准组中进行研究和讨论。如将极高吞吐率（extremely high throughput, EHT）作为目标的关键技术可以包括多链路（multi-link, ML）通信。基于该多链路通信，多链路设备可以同时 2.4GHz、5GHz 以及 6GHz 频段上进行通信，从而选择最佳的频段，保证其通信质量。

在多链路通信的场景中，接入点多链路设备可以沿多条链路向站点多链路设备发送对应同一个业务标识符的数据包。同时，站点多链路设备在每条链路上维护本地记分板时，不同链路的站点在现有的块确认机制中可能无法完成对所接收的聚合媒介接入控制层协议数据单元（aggregation medium access control protocol data unit, A-MPDU）中所有 MPDU 的接收情况的正确反馈。

因此，在多链路通信的场景中，如何保障不同链路的接收设备正确反馈接收到的 A-MPDU 中所有 MPDU 的接收情况亟待解决。

发明内容

本申请提供一种多链路通信方法及装置，可以有效保证接收端正确反馈其接收到的 A-MPDU 中所有 MPDU 的接收情况。

第一方面，本申请实施例提供一种多链路通信方法，所述方法包括：接收端接收第一聚合的媒介接入控制层协议数据单元 A-MPDU；所述接收端获取本地记录的记分板上下文，在所述第一 A-MPDU 满足如下任一项或多项条件时，根据所述第一 A-MPDU 的序列号 SN 刷新所述记分板上下文：第一条件包括：隶属于所述接收端的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$ ；第二条件包括：SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的任何一个 MPDU 都不是通过所述某个 STA 接收的；或者，SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的所有 MPDU 都是由隶属于所述接收端的其他 STA 接收的；第三条件包括：不存在任意一个在接收到所述第一 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU；或者，在所述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应；或者，在所述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应。

在一种可能的实现方式中，在所述第一 A-MPDU 还满足以下第四条件：所述第四条件包括：在所述第一 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足所述第一条件、所述第二条

件和所述第三条件；或者，隶属于所述接收端的某个 STA 接收到的具有 SN 的 MPDU 是所述第一 A-MPDU 中满足所述第一条件、所述第二条件和所述条件的第一个 MPDU；或者，隶属于所述接收端的某个 STA 的计分板上下文还没有基于所述第一 A-MPDU 中的 MPDU 进行刷新。

在一种可能的实现方式中，所述接收端在每条链路上都有一个单独的计分板上下文控制，所述链路为所述接收端与发送端之间的链路。

在一种可能的实现方式中，隶属于所述接收端的 STA 具有使用重新排序的缓冲区信息更新计分板上下文的能力。

在一种可能的实现方式中，所述第一 A-MPDU 中的 SN 属于同一个业务标识 TID。

通过本申请实施例提供的方法，可使得接收端能够合理有效地刷新本地计分板上下文 (local scoreboard)。

可理解，关于第一方面或任一种可能的实现方式的具体说明，可以参考下文关于第三种计分板上下文控制操作至第十种计分板上下文控制操作的描述，以及下文图 4 所示的方法等，这里不再一一详述。

第二方面，本申请实施例提供一种多链路通信方法，所述方法包括：发送端发送第二 A-MPDU，所述 A-MPDU 包括指示信息，所述指示信息用于指示接收端是否刷新所述接收端的计分板上下文。

在一种可能的实现方式中，所述指示信息用于指示用于传输所述第二 A-MPDU 的链路对应的 STA 是否刷新所述 STA 的计分板上下文。

在一种可能的实现方式中，所述指示信息承载于 MPDU 分隔符，或者高吞吐率控制字段中。

可理解，关于第二方面或任一种可能的实现方式的具体说明，可以参考下文图 5 所示的方法，以及图 6a 至图 6d 所示的描述，这里不再一一详述。

第三方面，本申请实施例提供一种通信装置，用于执行第一方面或第一方面的任意可能的实现方式中的方法。该通信装置包括具有执行第一方面或第一方面的任意可能的实现方式中的方法的单元。

示例性的，该通信装置可以为发送端或发送端中的芯片等。

第四方面，本申请实施例提供一种通信装置，用于执行第二方面或第二方面的任意可能的实现方式中的方法。该通信装置包括具有执行第二方面或第二方面的任意可能的实现方式中的方法的单元。

示例性的，该通信装置可以为接收端或接收端中的芯片等。

在第三方面或第四方面中，上述通信装置可以包括收发单元和处理单元。对于收发单元和处理单元的具体描述还可以参考下文示出的装置实施例。

第五方面，本申请实施例提供一种通信装置，该通信装置包括处理器，用于执行上述第一方面或第一方面的任意可能的实现方式所示的方法。或者，该处理器用于执行存储器中存储的程序，当该程序被执行时，上述第一方面或第一方面的任意可能的实现方式所示的方法被执行。

在一种可能的实现方式中，存储器位于上述通信装置之外。

在一种可能的实现方式中，存储器位于上述通信装置之内。

本申请实施例中，处理器和存储器还可以集成于一个器件中，即处理器和存储器还可以被集成在一起。

在一种可能的实现方式中，通信装置还包括收发器，该收发器，用于接收信号或发送信号。示例性的，该收发器还可以用于接收 A-MPDU 等。

本申请实施例中，该通信装置可以为发送端或发送端中的芯片等。

第六方面，本申请实施例提供一种通信装置，该通信装置包括处理器，用于执行上述第二方面或第二方面的任意可能的实现方式所示的方法。或者，处理器用于执行存储器中存储的程序，当该程序被执行时，上述第二方面或第二方面的任意可能的实现方式所示的方法被执行。

在一种可能的实现方式中，存储器位于上述通信装置之外。

在一种可能的实现方式中，存储器位于上述通信装置之内。

在本申请实施例中，处理器和存储器还可以集成于一个器件中，即处理器和存储器还可以被集成在一起。

在一种可能的实现方式中，通信装置还包括收发器，该收发器，用于接收信号或发送信号。示例性的，该收发器可以用于发送 A-MPDU。

本申请实施例中，该通信装置可以为接收端或接收端中的芯片等。

第七方面，本申请实施例提供一种通信装置，该通信装置包括逻辑电路和接口，所述逻辑电路和所述接口耦合；所述接口，用于输入第一 A-MPDU；所述逻辑电路，用于根据所述第一 A-MPDU 刷新本地计分板上下文。

可选的，所述接口，还用于输出 BA 帧。

可选的，该通信装置还包括存储器，所述存储器用于存储第三种计分板上下文控制操作至第十种计分板上下文控制操作中的任一项或多项。

第八方面，本申请实施例提供一种通信装置，该通信装置包括逻辑电路和接口，所述逻辑电路和所述接口耦合；所述逻辑电路，用于生成第一 A-MPDU；所述接口，用于输出所述第一 A-MPDU。

在本申请的另一些实施例中，接口，用于输出第二 A-MPDU，该第二 A-MPDU 包括指示信息。

第九方面，本申请实施例提供一种计算机可读存储介质，该计算机可读存储介质用于存储计算机程序，当其在计算机上运行时，使得上述第一方面或第一方面的任意可能的实现方式所示的方法被执行。

第十方面，本申请实施例提供一种计算机可读存储介质，该计算机可读存储介质用于存储计算机程序，当其在计算机上运行时，使得上述第二方面或第二方面的任意可能的实现方式所示的方法被执行。

第十一方面，本申请实施例提供一种计算机程序产品，该计算机程序产品包括计算机程序或计算机代码，当其在计算机上运行时，使得上述第一方面或第一方面的任意可能的实现方式所示的方法被执行。

第十二方面，本申请实施例提供一种计算机程序产品，该计算机程序产品包括计算机程序或计算机代码，当其在计算机上运行时，使得上述第二方面或第二方面的任意可能的实现方式所示的方法被执行。

第十三方面，本申请实施例提供一种计算机程序，该计算机程序在计算机上运行时，上述第一方面或第一方面的任意可能的实现方式所示的方法被执行。

第十四方面，本申请实施例提供一种计算机程序，该计算机程序在计算机上运行时，上述第二方面或第二方面的任意可能的实现方式所示的方法被执行。

第十五方面，本申请实施例提供一种无线通信系统，该无线通信系统包括发送端和接收端，所述发送端用于执行上述第一方面或第一方面的任意可能的实现方式所示的方法，所述接收端用于执行上述第二方面或第二方面的任意可能的实现方式所示的方法。

第一方面至第十五方面涉及的 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$ 是以 $\text{WinSize}_R = 1024$ 为例示出的。然而，本申请涉及的 WinSize_R 还可以是 64、128、256 或 512，因此，第一方面至第十五方面涉及的 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$ 还可以理解为： $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R$ 。这里不再一一列举。

附图说明

图 1a 至图 1c 是本申请实施例提供的一种多链路通信的场景示意图；
图 2 是本申请实施例提供的一种多链路通信方法的流程示意图；
图 3a 至图 3f 是本申请实施例提供的一种计分板上下文控制操作的示意图；
图 4 是本申请实施例提供的一种多链路通信方法的流程示意图；
图 5 是本申请实施例提供的另一种多链路通信方法的流程示意图；
图 6a 是本申请实施例提供的一种 MPDU 子帧的结构示意图；
图 6b 是本申请实施例提供的一种聚合控制字段的格式示意图；
图 6c 是本申请实施例提供的一种聚合控制字段的格式示意图；
图 6d 是本申请实施例提供的一种命令和状态类型控制信息的帧结构示意图；
图 7 至图 9 是本申请实施例提供的一种通信装置的结构示意图；
图 10a 和图 10b 是本申请实施例提供的一种计分板上下文控制操作的示意图。

具体实施方式

为了使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请作进一步地描述。

本申请的说明书、权利要求书及附图中的术语“第一”和“第二”等仅用于区别不同对象，而不是用于描述特定顺序。此外，术语“包括”和“具有”以及它们的任何变形，意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备等，没有局限于已列出的步骤或单元，而是可选地还包括没有列出的步骤或单元等，或可选地还包括对于这些过程、方法、产品或设备等固有的其它步骤或单元。

在本文中提及的“实施例”意味着，结合实施例描述的特定特征、结构或特性可以包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例，也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员可以显式地和隐式地理解的是，本文所描述的实施例可以与其它实施例相结合。

在本申请中，“至少一个（项）”是指一个或者多个，“多个”是指两个或两个以上，“至少两个（项）”是指两个或三个及三个以上，“和/或”，用于描述关联对象的关联关系，表示可以存在三种关系，例如，“A 和/或 B”可以表示：只存在 A，只存在 B 以及同时存在 A 和 B 三种情况，其中 A、B 可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项（个）”或其类似表达，是指这些项中的任意组合。例如，a、b 或 c 中的至少一项（个），可以表示：a，b，c，“a 和 b”，“a 和 c”，“b 和 c”，或“a 和 b 和 c”。

本申请提供的多链路通信方法可以应用于无线通信系统。该无线通信系统可以包括

WLAN 或蜂窝网等。该方法可以由无线通信系统中的通信设备或通信设备中的逻辑电路或处理器实现等，该通信设备可以是一种支持多条链路并行进行传输的无线通信设备，例如，称为多链路设备（multi-link device, MLD）或多频段设备（multi-band device）等。比如，在无线局域网中，该通信设备支持采用 IEEE 802.11 系列协议进行通信，IEEE 802.11 系列协议包括：802.11be, 802.11be 下一代, 802.11ax, 或 802.11a/b/g/n/ac 等，这里不再一一列举。

以下先对本申请所涉及的多链路设备进行说明。

多链路通信的核心思想是支持 IEEE 802.11 标准的 WLAN 设备如 EHT 设备等拥有在多个频段上发送和/或接收的能力，从而可以使用更大的带宽进行传输，进而提升吞吐率。如多频段主要包括但不限于 2.4GHz Wi-Fi 频段、5GHz Wi-Fi 频段或 6GHz Wi-Fi 频段。示例性的，在每一个频段上的接入和/或传输称为一个链路，从而多个频段上的接入和/或传输便称为多链路。例如，支持多链路通信的设备称为多链路设备（multi-link device, MLD）。图 1a 是本申请实施例提供的一种多链路通信的场景示意图，如 MLD 设备中存在多个接入点（access point, AP）或站点（station, STA），从而组成 AP 多链路设备（AP multi-link device, AP MLD）或 non-AP 多链路设备（non-AP multi-link device, non-AP MLD）。MLD 之间的通信为多链路通信，如图 1a 中的链路 1 和链路 2 组成了多链路。

多链路设备包括一个或多个隶属的站点，隶属的站点是逻辑上的站点，可以工作在一条链路或一个频段或一个信道上等。该隶属的站点可以为接入点（access point, AP）或非接入点站点（non-access point station, non-AP STA）。一般的，可以将隶属的站点为 AP 的多链路设备称为多链路 AP 或多链路 AP 设备或 AP MLD。隶属的站点为 non-AP STA 的多链路设备可以称为多链路 STA 或多链路 STA 设备或 STA 多链路设备（STA multi-link device），或者，隶属的站点为 non-AP STA 的多链路设备称为多链路 non-AP 或多链路 non-AP 设备或 non-AP MLD 等。下文将隶属的站点为 AP 的多链路设备称为 AP MLD，将隶属的站点为 non-AP STA 的多链路设备称为 non-AP MLD。AP MLD 中隶属的 AP 为一个或多个；STA MLD 中隶属的 STA 为一个或多个。

多链路设备可以遵循 802.11 系列协议实现无线通信，例如，遵循极高吞吐率（extremely high throughput, EHT）的多链路设备，或遵循基于 802.11be 或兼容支持 802.11be 的多链路设备，实现与其他设备的通信。

多链路设备（这里既可以是 non-AP MLD，也可以是 AP MLD）为具有无线通信功能的通信装置。该通信装置可以为一个整机的设备，还可以是安装在整机设备中的芯片或处理系统等，安装这些芯片或处理系统的设备可以在这些芯片或处理系统的控制下，实现本申请实施例的方法和功能。例如，本申请实施例中的多链路设备具有无线收发功能，可以支持 802.11 系列协议，可以与 AP 多链路设备或 non-AP 多链路设备进行通信。例如，non-AP 多链路设备是允许用户与 AP 通信进而与 WLAN 通信的任何用户通信设备。例如，non-AP 多链路设备可以为平板电脑、桌面型、膝上型、笔记本电脑、超级移动个人计算机（ultra-mobile personal computer, UMPC）、手持计算机、上网本、个人数字助理（personal digital assistant, PDA）、手机等可以联网的用户设备，或物联网中的物联网节点，或车联网中的车载通信装置等。non-AP 多链路设备还可以为上述这些终端中的芯片和处理系统。AP 多链路设备可以为 non-AP 多链路设备提供服务的装置，可以支持 802.11 系列协议。例如，AP 多链路设备可以为通信服务器、路由器、交换机、网桥等通信实体，或，AP 多链路设备可以包括各种形式的宏基站，微基站，中继站等，当然 AP 多链路设备还可以为这些各种形式的设备中的芯片和

处理系统。其中，802.11 协议可以为支持 802.11be 或兼容 802.11be 的协议。

可理解的，多链路设备（这里既可以是 non-AP MLD，也可以是 AP MLD）可以支持高速率低时延的传输，随着无线局域网应用场景的不断演进，多链路设备还可以应用于更多场景中，比如为智慧城市中的传感器节点（比如，智能水表，智能电表，智能空气检测节点），智慧家居中的智能设备（比如智能摄像头，投影仪，显示屏，电视机，音响，电冰箱，洗衣机等），物联网中的节点，娱乐终端（比如 AR，VR 等可穿戴设备），智能办公中智能设备（比如，打印机，投影仪等），车联网中的车联网设备，日常生活场景中的一些基础设施（比如自动售货机，商超的自助导航台，自助收银设备，自助点餐机等）。本申请实施例中对于多链路设备的具体形式不做限定，在此仅是示例性说明。

结合以上所示的多链路设备，图 1b 是本申请实施例提供的一种多链路通信的场景示意图。如图 1b 所示，AP MLD 包括 AP1，AP2，…，APn，non-AP MLD 包括 STA1，STA2，…，STAn。这里所示的 n 为正整数。AP MLD 和 non-AP MLD 可以采用链路 1，链路 2，…，链路 n 并行进行通信。non-AP MLD 中的 STA1 与 AP MLD 中的 AP1 建立关联关系，non-AP MLD 中的 STA2 与 AP MLD 中的 AP2 建立关联关系，non-AP MLD 中的 STAn 与 AP MLD 中的 APn 建立关联关系等。由此，non-AP MLD 中的一个或多个 STA 与 AP MLD 中的一个或多个 AP 之间建立关联关系之后便可以通信。

图 1c 是本申请实施例提供的一种多链路通信的场景示意图。如图 1c 所示，包括至少一个 AP 和至少一个 STA，图 1c 示出的是两个 STA，如 STA1、STA2，以及一个 non-AP MLD。例如，该 non-AP MLD 可以通过两个链路与 AP 通信，该 non-AP MLD 包括两个隶属的 STA，图 1c 未示出。又例如，STA2 或 STA3 可以通过一个链路与 AP 通信。也就是说，图 1c 所示的系统中既包括多链路通信，也包括单链路通信。

本申请所提供的方法可以适用于但不限于：单用户的上/下行传输、多用户的上/下行传输、车与任何事物（vehicle-to-everything，V2X，X 可以代表任何事物）、设备到设备（device-to-device，D2D）。例如，该 V2X 可以包括：车辆到车辆（vehicle to vehicle，V2V），车辆与基础设施（vehicle to infrastructure，V2I）、车辆与行人之间的通信（vehicle to pedestrian，V2P）或车辆与网络（vehicle to network，V2N）通信等。

本申请下文所示的发送端可以理解为使用块确认（block acknowledge，BA）机制发送数据的 MLD，该发送端也可以称为发送 MLD（originator MLD），接收数据的一方称为接收端，该接收端还可以称为接收 MLD（recipient MLD）。示例性的，发送端可以是 non-AP MLD，接收端可以是 AP MLD。又如，发送端可以是 AP MLD，接收端可以是 non-AP MLD。发送端与接收端之间可以包括多条链路，其中每条链路供一个隶属于发送 MLD 的 STA（a STA affiliated with originator MLD）和一个隶属于接收 MLD 的 STA（a STA affiliated with recipient MLD）进行通信。

图 2 是本申请实施例提供的一种多链路通信方法的流程示意图。该多链路通信方法是基于块确认（block acknowledge，BA）机制实现的，该 BA 机制可以将多个“确认（acknowledgment）”汇总到一个帧，以此提高信道效率。如图 2 所示，该方法包括：

201、发送端向接收端发送块确认（add BA，ADDBA）请求（request）帧。对应的，接收端接收该 ADDBA 请求帧。

202、接收端向发送端发送 ADDBA 响应（response）帧。对应的，发送端接收该 ADDBA 响应帧。

通过上述步骤 201 和步骤 202, 发送端和接收端之间便可以称为成功建立 BA 协议, 或者成功建立 BA 机制, 或者成功建立 BA 会话。

203、发送端向接收端发送聚合的媒介接入控制层协议数据单元 (aggregated MPDU, A-MPDU)。对应的, 接收端接收该 A-MPDU。

可理解, 基于发送端与接收端之间建立的 BA 会话, 因此当发送端需要向接收端发送多个媒介接入控制 (medium access control, MAC) 层协议数据单元 (MAC protocol data unit, MPDU) 时, 该发送端可以将上述多个 MPDU 以聚合方式变成一个 A-MPDU。也就是说, 接收端在接收到 A-MPDU 之后, 会对 A-MPDU 进行解聚合控制 (deaggregation control), 获得多个 MPDU。示例性的, A-MPDU 中的每个 MPDU 会有一个序列号 (sequence number, SN), 该 SN 用于表示 A-MPDU 中的每个 MPDU 的顺序。

示例性的, 一个 A-MPDU 中所包括的 MPDU 的最大数量可以是 1024。例如, 记分板的窗口大小是可以变化的, 其可以通过 ADDBA 请求帧和 ADDBA 响应帧交互中协商确定。例如, 该记分板的窗口大小可以是 64, 128, 256, 512, 1024 等。一个 A-MPDU 聚合的数据不超过记分板窗口的大小, 但是可以小于记分板窗口的大小。

204、发送端向接收端发送 BA 请求 (BA request, BAR) 帧。对应的, 接收端接收该 BAR。

可理解, 步骤 204 是以发送端发送 BA 请求帧为例示出的。可选的, 发送端还可以在发送的 A-MPDU 中设置确认原则 (ACK policy) 为隐含的 BAR (implicit BAR), 由此, 发送端可以不需要单独发送 BAR 帧。

205、接收端基于 BAR 向发送端发送 BA 帧, 该 BA 用于对 A-MPDU 中的 MPDU 的接收情况做出确认。对应的, 发送端接收该 BA 帧。

通常来讲, 接收端需要维护一个块确认记录 (BA record)。该 BA 记录包含一个按序列号索引的位图: 一个 12 位无符号整数的开始序号 (可以用 WinStart_R 表示), 该开始序号可以用于表示位图中最低的序列号位置; 当前传输窗口中的最高序号 (可以用 WinEnd_R 表示); 最大窗口尺寸 (如可以用 WinSize_R 表示)。为便于描述, 下文将以接收端的记分板包括 WinStart_R 、 WinSize_R 和 WinEnd_R 三个参数定义为例说明本申请实施例提供的方法。其中, WinStart_R 表示记分板的开始序号, WinSize_R 表示记分板的窗口尺寸 (如可以为 1024), WinEnd_R 表示记分板的结束序号 (记分板当前的结束序号)。

示例性的, 在基于 A-MPDU 获得多个 MPDU 之后, 接收端对每个 MPDU 进行记分板上下文控制 (scoreboard context control) 操作, 并在记分之后将接收到的 MPDU 递交到接收重排序缓冲区 (receive reordering buffer), 按照 SN 的顺序对 MPDU 进行排序, 将接收正确的 MPDU 向上递交。如果在重排序过程中出现一个未成功接收的 MPDU, 则将该 MPDU 对应的 SN 记做 WinStart_B , 并向上层递交该 SN 之前的按照顺序且成功接收的所有 MPDU。对于该 SN 之后的 MPDU, 接收端会在接收该 SN 对应的 MPDU 之后才能将其以及之后的接收正确的 MPDU 向上递交。

图 3f 是本申请实施例提供的一种记分板上下文控制操作的示意图。示例性地, 如图 3f 所示, 接收端在解聚合一个 A-MPDU 之后得到 SN 分别是 102、103、105 和 100 等四个 MPDU, 则接收端可以在记分板上下文中与 SN 对应的比特位置上记一个 1, 该 “1” 用于表示 SN 对应的 MPDU 被正确接收, 并基于记分结果形成比特位图 (bitmap), 将该比特位图放在 BA 帧中, 作为对应的一个 A-MPDU 的确认。在图 3f 中, $\text{WinStart}_R=98$, $\text{WinSize}_R=12$, $\text{WinEnd}_R=109$ 。

一个 BA 记录的序列号空间可以包括 4096 个 SN, 记分板的窗口可以在该序列号空间中移动。随着 BA 会话的建立, 记分板可以初始化。如 WinStart_R 可以被设置为 ADDBA 请求帧

提供的起始序列号 (start sequence number, SSN)。当一个 MPDU 到达时, 如果该 MPDU 的 SN 落在记分板所表示的空间内, 接收端将使用 SN 索引记分板, 并记录其正确接收。如果 SN 在记分板表示的空间之外, 但在 WinEnd_R 到 WinStart_R+2^{11} 的范围内 (序列号空间的一半以内), 接收端将向右移动记分板的窗口 (可以理解为: 接收端更新 WinEnd_R , 使 $\text{WinEnd}_R=\text{该 SN}$) 直到记分板的窗口的最右边缘包含新的 SN。当 BAR 帧到达时, 记分板的窗口向右移动, 使 WinStart_R 等于 BAR 帧提供的 SSN, 并返回带有记分板的记录内容的 BA 帧。

示例性的, 接收端接收带有 SN 的 MPDU 时, 会检查该 MPDU 是否有对应的 BA 会话的记分板记录。其中, 该 BA 会话是由发送地址 (transmission address, TA) 和 TID 标识的。如果没有记分板记录, 接收端为这个 BA 会话创建一个记分板, 也许可以重用另一个会话的内存。

需要说明的是, 本申请实施例所示的多链路通信的场景中, 发送端与接收端之间的每条链路均会维护一个各自的记分板上下文控制, 且还有一个 MLD 的公共记分板。每条链路上的设备 (如 MLD 中的每个 STA, 或者, MLD 中的每个 AP) 会使用各自的记分板上下文控制反馈该条链路上的 MPDU 的接收情况。公共记分板可以用于记录其他链路上的记分板的参数信息, 例如, 每个链路上的记分板的起始序号和/或结束序号, 或者记录综合所有链路的记分板的信息 (如 WinStart_B)。

以下结合具体的记分板上下文控制操作说明本申请实施例提供的方法。

示例性的, SN 的范围是 0-4095。当 SN 为 4095 时, 如果再加 1, 则 4096 就会变成 0。因此, 本申请下文所示的各个例子中的 SN 的数值均是以对 4096 取模得到的。如接收端可以对 SN 以模 4096 进行操作, 每个 SN 空间可以由一个模 4096 计数器表示。

图 3a 至图 3c 所示的示意图中, 发送端可以通过两条链路, 如第一链路和第二链路与接收端通信。为便于描述, 下文将隶属于接收端的第一链路上的 STA 称为第一 STA, 隶属于接收端的第二链路上的 STA 称为第二 STA。即第一 STA 和第二 STA 隶属于同一个 MLD。

第一种记分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的记分板上下文控制;

隶属于该接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新记分板上下文;

当隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R+2^{11} \leq \text{SN} < \text{WinStart}_R$, 且不满足 $\text{WinStart}_B+2^{11} \leq \text{SN} < \text{WinStart}_B$;

则该某个 STA 可以认为其接收到的 MPDU 的 SN 满足 $\text{WinEnd}_R < \text{SN} < \text{WinStart}_R+2^{11}$, 该某个 STA 可以更新其记分板上下文。

图 3a 是本申请实施例提供的一种记分板上下文控制操作的示意图。举例来说, 同一个 TID 对应四个 A-MPDU, 每个 A-MPDU 的聚合长度都为 1024, 即每个 A-MPDU 均包括 1024 个 MPDU, 每个 A-MPDU 的确认策略是立即块确认。第一 STA 接收第一个 A-MPDU (如图 3a 所示的 A-MPDU1), SN 为 0-1023; 第二 STA 接收第二个 A-MPDU (如图 3a 所示的 A-MPDU2), SN 为 1024-2047, 和第三个 A-MPDU (如图 3a 所示的 A-MPDU3), SN 为 2048-3071; 第一 STA 接收第四个 A-MPDU (如图 3a 所示的 A-MPDU4), SN 为 3072-4095。可理解, 关于图 3b 所示的各个 A-MPDU 和 SN 的关系的说明可以参考图 3a, 下文不再一一详述。

当第一 STA 根据第一个 A-MPDU 的 SN 对其记分板上下文更新之后, 该第一 STA 的记分板 (也可以称为第一链路的记分板) 的 $\text{WinStart}_R=0$, $\text{WinEnd}_R=1023$ 。以及, 第二个 STA 根据第二个 A-MPDU 和第三个 A-MPDU 的 SN 对其记分板上下文更新之后, 该第二个 STA 的记分板 (也可以称为第二链路的记分板) 的 $\text{WinStart}_R=2048$, $\text{WinEnd}_R=3071$ 。同时, 第一

STA 的计分板和第二 STA 的计分板中的 $\text{WinStart}_B=3072$ 。

根据上述第一种计分板上下文控制操作,当第一 STA 接收到的第四个 A-MPDU 的 SN 满足 $0+2048 \leq \text{SN} < 0$, 且 SN 不满足 1024 (根据 4096 对 3072 与 2048 之和取模得到) $\leq \text{SN} < 3072$ 时, 第一 STA 可以根据第四个 A-MPDU 的 SN 更新其计分板上下文。由于第四个 A-MPDU 的 SN 为 3072-4095, 因此如果第一 STA 均成功接收第四个 A-MPDU 中所有 MPDU, 则更新第一 STA 的计分板 $\text{WinStart}_R=3072$, $\text{WinEnd}_R=4095$ 。

然而, 当第一 STA 和第二 STA 接收到的 A-MPDU 如图 3b 所示, 当第一 STA 接收到 SN 为 0-1023 的第五个 A-MPDU (如图 3b 所示的 A-MPDU5) 时, 根据上述第一种计分板上下文控制操作, 第一 STA 不能正确反馈第五个 A-MPDU 中所有 MPDU 的接收情况。

为改善上述问题, 本申请实施例还提供了第二种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于该接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

当隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$;

隶属于该接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

则该某个 STA 可以刷新 (或冲刷) 该某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

以上所示的接收 MLD 在每条链路上都有一个单独的计分板上下文控制, 也可以理解为: 隶属于接收 MLD 的每个 STA 各自维护一个计分板上下文控制。

需要说明的是, 本申请实施例所示的某个 STA 刷新其计分板上下文指的是该某个 STA 将其计分板上下文进行初始化, 或者, 将其计分板上下文清零处理, 或者, 清除该计分板上下文中已经记录的内容, 或者, 清除该计分板上下文的缓存等。更新该某个 STA 的计分板上下文指的是该某个 STA 可以基于计分板上下文中已有的内容进行更新。

以图 3b 为例, 第一 STA 接收到的第五个 A-MPDU 的 SN 满足 $0 \leq \text{SN} < 1023$, 第二 STA 的计分板上下文 (即其他 STA 的计分板) 接收到的 MPDU 的 SN 满足 $3072 \leq \text{SN} < 4096$, 因此, 第一 STA 可以刷新该第一 STA 计分板并更新第一 STA 的计分板上下文。也就是说, 该第一 STA 可以将其在接收到第五个 A-MPDU 之前的记录清除掉, 从而利用第五个 A-MPDU 的 SN 对初始化后的计分板上下文进行更新。由此, 如果第一 STA 均成功接收第五个 A-MPDU 中所有 MPDU, 则刷新并更新后的第一 STA 的计分板 $\text{WinStart}_R=0$, $\text{WinEnd}_R=1023$ 。

又一种场景中, 当第一 STA 和第二 STA 接收到的 A-MPDU 如图 3c 所示, 在第一 STA 未接收到第六个 A-MPDU (如图 3c 所示的 A-MPDU6) 之前, 第一 STA 的计分板上下文中的 $\text{WinStart}_R=0$, $\text{WinEnd}_R=1023$ 。当接收端接收到第六个 A-MPDU 时, 由于 SN 为 1024-1500, 该第六个 A-MPDU 的 SN 既不属于上述第一种计分板上下文控制操作, 也不属于上述第二种计分板上下文控制操作。由此, 第一 STA 就无法获知到底该如何更新其计分板上下文。也可以理解为: 当接收端接收到第六个 A-MPDU 时, 上述第二种计分板上下文控制操作中并没有考虑当第一 STA 接收到的 MPDU 的 SN 处于 $\text{WinEnd}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$ 这个范围。如果直接将第一 STA 的计分板中的 WinEnd_R 移动到 1500 的位置, WinStart_R 在 477 (即 1500-1023), 则在反馈 BA 帧的时候会把上一轮的 477~1023 的状态信息错误地反馈回去。

为改善上述问题, 本申请实施例提供了第三种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

则该某个 STA 可以刷新该某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

(If the following conditions are true: a recipient MLD has a separate scoreboard context control in each link; The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context; a STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$; the other STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$; the STA shall flush the scoreboard context and update the scoreboard context.)

基于上述第三种计分板上下文控制操作, 如以图 3c 为例, 在第一 STA 未接收到第六个 A-MPDU 之前, 该第一 STA 中计分板上下文中 $\text{WinStart}_R=0$, $\text{WinEnd}_R=1023$ 。当该第一 STA 接收到第六个 A-MPDU 时, 由于该第六个 A-MPDU 的 SN 满足: $0 \leq \text{SN} < 0 + 2^{11}$, 其他 STA (如图 3c 所示的第二 STA) 已经接收到的 A-MPDU 的 SN 满足: $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ ($\text{WinStart}_B=1024$)。因此, 该第一 STA 可以刷新其计分板上下文并更新计分板上下文, 更新后的计分板上下文中 $\text{WinStart}_R=477$, $\text{WinEnd}_R=1500$ 。

结合以上所示的第一种计分板上下文控制操作和第二种计分板上下文控制操作, 隶属于接收 MLD 的 STA 每获得一个 MPDU 都可以刷新计分板上下文。这种方式会存在不恰当地冲刷本地计分板的问题。

假设 STA 接收到了一个 A-MPDU, 该 A-MPDU 包含第一 MPDU 和第二 MPDU, 则根据第一 MPDU 的 SN 可能会冲刷一次本地计分板上下文, 然后再根据第二 MPDU 冲刷本地计分板上下文的时候又将还未反馈 BA 帧的与第一 MPDU 的 SN 相关的计分板上下文给冲刷掉。因为 STA 根据第一 MPDU 的 SN1 冲刷本地计分板上下文之后, 该 STA 还未反馈 BA 帧, 该情况下, 如果再根据第二 MPDU 冲刷本地计分板上下文, 会导致 STA 无法正确反馈 BA 帧。

为改善上述问题, 本申请实施例提供了第四种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

在当前接收到的 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述所有条件 (no other frame with a SN in current A-MPDU that already fulfills all above conditions); 或者, 该某个 STA 接收到的上述具有 SN 的 MPDU 是 A-MPDU 中满足以上条件的第一个 MPDU (it is the first frame with a SN in current A-MPDU that fulfills all above conditions); 或者, 该某个 STA 的计分板上下文还没有基于当前接收到的 A-MPDU 中的 MPDU 进行刷新 (the scoreboard is not flushed base on a frame in current A-MPDU);

则该某个 STA 可以刷新该某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。(If the following conditions are true: a recipient MLD has a separate scoreboard context control in each

link, The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context; a STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$; the other STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$; no other frame with a SN in current A-MPDU that already fulfills all above conditions; the STA shall flush the scoreboard context and update the scoreboard context.)

也就是说,在第四种计分板上下文控制操作中,当接收端获取到 A-MPDU,且该 A-MPDU 包括的多个 MPDU 都是发送给某个 STA 时,且该多个 MPDU 的 SN 都满足 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$,则该某个 STA 可以基于该 A-MPDU 中的第一个满足除了最后一条以外所有条件的 MPDU 对该某个 STA 的计分板上下文进行刷新;也可以称为该某个 STA 不可以基于该 A-MPDU 中的非第一个满足除了最后一条以外所有条件的 MPDU 的 SN 对其计分板上下文进行刷新。

以图 3d 所示的示意图为例,当第一 STA 在接收到 SN 为 500~1023 和 SN 为 0~499 的两个 A-MPDU 之后,由于这两个 A-MPDU 都满足上述第二种计分板上下文控制操作,因此该第一 STA 会分别进行两次冲刷。如当在接收到 SN 为 0~499 的 A-MPDU 之后冲刷本地计分板上下文的时候会把还没有回复 BA 帧的 SN 为 500~1023 的 BA 记录也冲刷掉。可理解,图 3d 所示的第一个确认原则 (ACK policy) 用于指示第一 STA 暂时不需要反馈 BA 帧,图 3d 所示的第二个确认原则 (也可以称为隐式的 BAR) 用于指示第一 STA 需要反馈 BA 帧。

为改善上述问题,本申请实施例还提供了第五种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU;或者,在当前接收到的 A-MPDU 之前接收的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应 (no any frame that are carried in any A-MPDU that received before current A-MPDU is not responded by a BA);或者,在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应 (all frames that are carried in the A-MPDUs that received before current A-MPDU has been responded by BAs);

则该某个 STA 可以刷新某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

(If the following conditions are true: a recipient MLD has a separate scoreboard context control in each link, The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context. a STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$; the other STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$; no any frame that are carried in any A-MPDU that received before current A-MPDU is not responded by a BA; the STA shall flush the scoreboard context and update the scoreboard context.)

也就是说,在第五种计分板上下文控制操作中,当某个 STA 先后接收到两个 A-MPDU 时,当第一个 A-MPDU 中的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$,则该某个 STA 可以

刷新其计分板上下文。在该某个 STA 还未反馈第一个 A-MPDU 的 BA 帧时,即使第二个 A-MPDU 中的 MPDU 的 SN 也满足上述条件,该某个 STA 将不会刷新其计分板上下文,但可以更新其计分板上下文。

以图 3e 所示的示意图为例,当第一 STA 接收到 SN 为 200~299 的 A-MPDU 之后,其满足上述第二种计分板上下文控制操作,该第一 STA 中的计分板上下文被刷新。然而,图 3e 所示的场景中,根据第一个 A-MPDU、第三个 A-MPDU 和第四个 A-MPDU 之间的 SN 关系,该第一 STA 是没有必要进行冲刷的。

为改善上述问题,本申请实施例提供了第六种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$;

SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的任何一个 MPDU 都不是通过该某个 STA 接收的 (none of a frame with SN that is $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ is (was) received by the STA); 或者, SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的所有 MPDU 都是由隶属于接收 MLD 的其他 STA 接收的 (all frames with SN that are $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ are (were) received by the other STA(s)); 需要说明一下,这里 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的 MPDU 是指针对当前重新排序的缓冲区的 WinStart_B 而言的,并不考虑 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的范围,但是重新排序的缓冲区内上一轮或者更早接收的 MPDU。

则该某个 STA 可以刷新某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

(If the following conditions are true: a recipient MLD has a separate scoreboard context control in each link, The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context; a STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$; none of a frame with SN that is $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ is (was) received by the STA; the STA shall flush the scoreboard context and update the scoreboard context.)

也就是说,在第六种计分板上下文控制操作中,当某个 STA 接收到 A-MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R$, 且 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的 MPDU 都不是该某个 STA 接收的,则该某个 STA 可以刷新其计分板上下文。

如以图 3e 为例,在第一 STA 未接收到第二个 A-MPDU 和第三个 A-MPDU 之前,该第一 STA 中的计分板上下文中 $\text{WinStart}_R = 199 - 1023 + 4096 = 3272$, $\text{WinEnd}_R = 199$, $\text{WinStart}_B = 200$ 。当该第一 STA 接收到第三个 A-MPDU 时,由于该第三个 A-MPDU 的 SN 为 300~399,不满足上述条件,因此该不刷新其计分板上下文。对于第四个 A-MPDU 类似,不再一一列举。

可理解,以上所示的第三种计分板上下文控制操作至第六种计分板上下文控制操作可以单独实现,也可以相互结合。示例性的,上述第三种计分板上下文控制操作可以与第四种计分板上下文控制操作结合,如得到第七种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

在当前接收到的 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述条件; 或者, 该某个 STA 接收到的上述具有 SN 的 MPDU 是 A-MPDU 中满足以上条件的第一个 MPDU; 或者, 该某个 STA 的计分板上下文还没有基于当前接收到的 A-MPDU 中的 MPDU 进行刷新;

则该某个 STA 可以刷新该某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

由此, 该第七种计分板上下文控制操作不仅可以改善上文如图 3c 所示的问题, 还可以改善上文所示的对每个 MPDU 都刷新计分板上下文的问题。

示例性的, 上述第三种计分板上下文控制操作可以与第五种计分板上下文控制操作结合, 如得到第八种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU; 或者, 在当前接收到的 A-MPDU 之前接收的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应; 或者, 在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应;

则该某个 STA 可以刷新该某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

由此, 该第八种计分板上下文控制操作不仅可以改善上文如图 3c 所示的问题, 还可以改善上文如图 3d 所示的问题。

可理解, 本申请实施例所示的第四种计分板上下文控制操作和第五种计分板上下文控制操作也可以结合, 这里不再一一详述。

示例性的, 上述第三种计分板上下文控制操作、第四种计分板上下文控制操作与第五种计分板上下文控制操作结合, 如得到第九种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

在当前接收到的 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述所有条件; 或者, 该某个 STA 接收到的上述具有 SN 的 MPDU 是当前接收到的 A-MPDU 中满足以上条件的第一个 MPDU; 或者, 该某个 STA 的计分板上下文还没有基于当前接收到的 A-MPDU 中的 MPDU 进行刷新;

不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU; 或者, 在当前接收到的 A-MPDU 之前接收的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应; 或者, 在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应;

则该某个 STA 可以刷新某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。
由此, 该第九种计分板上下文控制操作可以有效改善上文如图 3c 至图 3d 所示的问题。

示例性的, 上述第三种计分板上下文控制操作、第四种计分板上下文控制操作、第五种计分板上下文控制操作和第六种计分板上下文控制操作结合, 如得到第十种计分板上下文控制操作:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$;

该某个 STA 接收到的 MPDU 中没有任何 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$; 或者, 隶属于接收 MLD 的其他 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU; 或者, 在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应; 或者, 在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应;

可理解, 以上所示的各个条件, 在 STA 执行时, 不限定先后顺序。

在当前接收到的 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述所有条件; 或者, 该某个 STA 接收到的上述具有 SN 的 MPDU 是 A-MPDU 中满足以上条件的第一个 MPDU; 或者, 该某个 STA 的计分板上下文还没有基于当前接收到的 A-MPDU 中的 MPDU 进行刷新;

则该某个 STA 可以刷新某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

由此, 该第十种计分板上下文控制操作可以有效改善上文如图 3c 至图 3e 所示的问题。

示例性的, 对于上文所示的第十种计分板上下文控制操作还可以理解为:

If the following conditions are true:

a recipient MLD has a separate scoreboard context control in each link,

The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context.

a STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$

none of a frame with SN that is $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ is (was) received by the STA;

no any frame that are carried in another A-MPDU that received before current A-MPDU is not responded by a BA;

no other frame with a SN in current A-MPDU that already fulfills all above conditions.

the STA shall flush the scoreboard context and update the scoreboard context.

可理解, 图3a至图3e均是以一个A-MPDU包括多个MPDU为例示出的, 每一个SN可以对应一个MPDU。可理解, 本申请以上所示的各个A-MPDU的SN均是以属于同一个TID为例示出的。

结合以上所示的第三种至第十种计分板上下文控制操作, 本申请实施例提供了一种多链路通信方法, 该方法中所涉及的发送端和接收端的说明可以参考上文, 这里不再一一详述。如图 4 所示, 该多链路通信方法包括:

401、发送端向接收端发送第一 A-MPDU，接收端接收该第一 A-MPDU。

可选的，发送端与接收端之间可以包括一条链路。

可选的，发送端与接收端之间可以包括至少两个链路。

可选的，第一 A-MPDU 可以包括一个 MPDU。可选的，该第一 A-MPDU 可以包括至少两个 MPDU。

可理解，由于 A-MPDU 是承载在 PPDU 中，通常对于一个接收站点来说，一个 PPDU 中只承载一个 A-MPDU，因此本申请实施例所示的 A-MPDU 还可以理解为 PPDU，本申请实施例对此不作限定。

402、接收端获取本地记录的计分板上下文，在第一 A-MPDU 满足如下任一项或多项条件时，根据该第一 A-MPDU 的 SN 刷新该计分板上下文。

可理解，本申请实施例所示的本地记录的计分板上下文可以是初始化的计分板上下文，也可以是接收端根据其在接收第一 A-MPDU 之前已经接收到的 A-MPDU 更新后的计分板上下文，或者，根据其在接收第一 A-MPDU 之前已经接收到的 A-MPDU 刷新后的计分板上下文等，这里不再一一列举。

第一 A-MPDU 满足的条件包括：

第一条件、隶属于该接收端的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$ 。

第二条件、SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的任何一个 MPDU 都不是通过所述某个 STA 接收的；或者，SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的所有 MPDU 都是由隶属于所述接收端的其他 STA 接收的。

第三条件、不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU；或者，在上述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应；或者，在上述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应。

第四条件、在上述第一 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述所有条件(如上文所示的第一条件至第三条件)；或者，隶属于上述接收端的某个 STA 接收到的具有 SN 的 MPDU 是上述第一 A-MPDU 中满足以上条件的第一个 MPDU；或者，隶属于上述接收端的某个 STA 的计分板上下文还没有基于上述第一 A-MPDU 中的 MPDU 进行刷新。

可理解，本申请实施例对于 STA 执行上述第一条件至第三条件的先后顺序不作限定。

可选的，第一 A-MPDU 满足以上所示的第一条件至第四条件中的任一项条件时，该第一 A-MPDU 所满足的其他条件，可以参考上文所示的第三种计分板上下文控制操作至第六种计分板上下文控制操作中的相关描述。

可选的，第一 A-MPDU 满足以上所示的第一条件至第四条件中的任两项条件时，该第一 A-MPDU 所满足的其他条件，可以参考上文所示的第六种计分板上下文控制操作至第九种上下文控制操作中的相关描述。

可选的，当第一 A-MPDU 满足以上所示的第一条件至第四条件时，该第一 A-MPDU 所满足的其他条件可以参考上文所示的第十种计分板上下文控制操作的相关描述。

可理解，关于第一 A-MPDU 所满足的条件还可以参考上文所示的第三种计分板上下文控制操作至第十种上下文控制操作的说明，这里不再一一详述。

403、接收端向发送端发送 BA 帧，该 BA 帧用于对第一 A-MPDU 中的 MPDU 的接收情况做出确认。

可理解, 本申请实施例未示出的 ADDBA 请求帧、ADDBA 响应帧以及 BAR 帧的说明, 可以参考上文所示的图 2, 这里不再一一详述。

通过本申请实施例提供的方法, 可使得接收端能够合理有效地刷新本地计分板上下文 (local scoreboard)。

图 4 所示的多链路通信方法中, 接收端可以根据上文所示的第三种计分板上下文控制操作至第十种计分板上下文控制操作为例刷新本地计分板上下文, 或者, 更新本地计分板上下文。本申请实施例还提供了一种多链路通信方法, 该方法中, 发送端可以通过指示信息指示接收端是否属性本地计分板上下文, 有效减少了接收端的复杂度。

图 5 是本申请实施例提供的另一种多链路通信方法的流程示意图, 该方法中涉及的发送端和接收端的说明可以参考上文, 这里不再一一详述。如图 5 所示, 该多链路通信方法包括:

501、发送端向接收端发送第二 A-MPDU, 该第二 A-MPDU 包括指示信息, 该指示信息用于指示接收端是否刷新该接收端中的计分板上下文。对应的, 接收端接收该第二 A-MPDU。

可理解, 图 4 所示的方法中接收端中的 STA 有能力使用重新排序的缓冲区信息更新计分板上下文。图 5 所示的方法中接收端中的 STA 可以有能力使用重新排序的缓冲区信息更新计分板上下文, 也可以没有能力使用重新排序的缓冲区信息更新计分板上下文。

例如, 指示信息设置为 0 时, 则可以表示接收端不需要刷新本地计分板上下文。又例如, 指示信息设置为 1 时, 则表示接收端必须刷新本地计分板上下文, 也可以隐含的表示接收端有能力刷新本地计分板上下文。一般的, 第二 A-MPDU 会通过某一个链路来传输, 在这个链路上接收 A-MPDU 的 STA 是唯一的, 因此与该用于传输第二 A-MPDU 的链路对应的 STA 可以根据指示信息确定是否刷新本地计分板上下文。

图 6a 是本申请实施例提供的一种 MPDU 子帧的结构示意图。如图 6a 所示, 指示信息可以承载于 MPDU 分隔符 (delimiter) 中的预留 (reserved) 字段中, 或者, 承载于高吞吐率 (high throughput, HT) 控制 (HT control) 字段中。

该 HT 控制字段可以携带在 MPDU (例如, 服务质量 (quality of service, QoS) 数据 (data) 帧, 服务质量空 (QoS Null) 帧和管理帧) 中, 用于携带一些控制信息。HT 控制字段的长度为 4 个字节。HT 控制字段有三种类型, 通过 B0 和 B1 比特进行区分。如表 1 所示, 当 B0=0 时, 对应的 HT 控制字段为高吞吐率 (high throughput, HT) 类型的 HT 控制字段; 当 B0=1, B1=0 时, 对应的 HT 控制字段为非常高吞吐量 (very high throughput, VHT) 类型的 HT 控制字段; 当 B0=1, B1=1 时, 对应的 HT 控制字段为高效率 (high efficiency, HE) 类型的 HT 控制字段。

表 1

变体	B0	B1	B2-B29	B30	B31
HT	0		HT 控制字段中间部分	AC 限制条件	RDG/更多的 PPDU
VHT	1	0	VHT 控制字段中间部分	AC 限制条件	RDG/更多的 PPDU
HE	1	1	聚合控制字段 A-Control		

其中, HE 类型的 HT 控制字段中的 B2-B31 部分称作聚合控制 (A-control) 字段 (subfield)。图 6b 是本申请实施例提供的一种聚合控制字段的格式示意图, 该聚合控制字段中包括控制列表 (Control List) 和填充 (Padding) 部分。其中, 控制列表的比特长度是可变的, 其可以包括一个或者多个控制子字段 (control subfield); 填充部分包括 0 个或更多的比特。如图 6c 所

示的本申请实施例提供的聚合控制字段中的一个控制子字段的格式示意图，一个控制子字段包括一个 4 比特的控制标识（control ID）和控制信息（control Information）。该控制标识用于标识控制信息。控制标识和其对应的控制信息的长度如表 2 所示：

表 2

控制标识 取值	含义	控制信息的长 度（比特）	控制信息的内容
0	触发的响应调度（Triggered response scheduling, TRS）	26	参见 9.2.4.6a.1（TRS 控制）
1	操作模式（operating mode, OM）	12	参见 9.2.4.6a.2（OM 控制）
2	高效率链路适应（HE link adaptation, HLA）	26	参见 9.2.4.6a.3（HLA 控制）
3	缓存状态报告（buffer status report, BSR）	26	参见 9.2.4.6a.4（BSR 控制）
4	上行功率净空（UL power headroom, UPH）	8	参见 9.2.4.6a.5（UPH 控制）
5	带宽查询报告（Bandwidth query report, BQR）	10	参见 9.2.4.6a.6（BQR 控制）
6	命令和状态（Command and status, CAS）	8	参见 9.2.4.6a.7（CAS 控制）
7	极高吞吐率操作模式（EHT operating mode, EHT OM）	6	参见 9.2.4.6a.8（EHT OM 控制）
8	单个响应调度（Single response scheduling, SRS）	10	参见 9.2.4.6a.9（SRS 控制）
10	接入点辅助请求（AP assistance request, AAR）	20	参见 9.2.4.6a.10（AAR 控制）
9,11-14	保留		
15	单纯地需要扩张（Ones need expansion surely, ONES）	26	设置为全 1

可理解，图 6a 所示的其他字段，如 MPDU 分隔符包括的帧结尾（end of frame, EOF）、预留（reversed）、MPDU 长度（length）、循环冗余校验（cyclic redundancy check, CRC）和分隔符签名（delimiter signature），以及 MPDU 中包括的帧控制（frame control）、长度/标识符（duration/ID）、地址 1（address 1）至地址 4（address 4）、序列控制（sequence control）、QoS 控制（QoS control）和 HT 控制（HT control），以及填充（padding）等的说明可以参考相关标准或协议，这里不再一一详述。

基于上文介绍，本申请实施例提供了两种可以在 HE 类型的 HT 控制字段中携带指示信息。

第一种方法是使用 CAS 类型控制信息中的预留（reversed）比特实现指示信息。该 CAS 类型控制信息的帧结构如图 6d 所示。如图 6d 所示，该 CAS 类型控制信息可以包括限制接入（access category, AC）类型（AC constraint）、反向授权（reverse direct grant, RDG）、参数

化空间复用传输 (parameterized spatial reuse transmission, PSRT) PPDU 和预留 (reserved)。可理解, 关于图 6d 的其他字段的相关描述可以参考相关标准或协议, 这里不再一一详述。

第二种方法是使用一个目前保留的控制标识 (control ID) 来建立一个新的控制类型 (如图 6c 所示), 然后通过该新控制类型的控制信息来进行指示。例如, 指示信息的比特长度可以为 1 比特。

502、接收端根据指示信息刷新计分板上下文。

可理解, 接收端也可以根据指示信息不刷新计分板上下文。如接收端可以更新计分板上下文。

503、接收端向发送端发送 BA 帧, 发送端接收该 BA 帧。

可理解, 本申请实施例未示出的 ADDBA 请求帧、ADDDBA 响应帧以及 BAR 帧的说明, 可以参考上文所示的图 2, 这里不再一一详述。

本申请实施例提供的方法, 发送端通过在 A-MPDU 中携带指示信息, 从而指示接收端刷新计分板上下文, 可有效减少接收端的复杂度, 提高接收端刷新计分板上下文的效率。

本申请实施例还提供了一种多链路通信方法, 这种实施方式中, 接收端中的 STA 可以有能力使用重新排序的缓冲区信息更新计分板上下文, 也可以没有能力使用重新排序的缓冲区信息更新计分板上下文。

本申请实施例中, 发送端在传输的数据帧 (即 A-MPDU) 中包括第一字段, 该第一字段用于指示当前 SN 是第几轮传输。该第一字段可以称为 SN 轮数字段 (Round of SN subfield), 该字段初始值设置为 0。当数据帧的 SN 一旦超过 SN 最大值 (4095) 就重新变为 0 之后, SN 轮数字段的取值加 1。举例来说, 假设 SN 轮数字段包括 5 个比特, 则其取值可以是 0-31。如当 SN 轮数字段设置为 31 之后, 如果再次加 1 就变回 0。

由此, 接收端在接收到一个数据帧之后, 如果满足以下条件, 就可以在更新本地计分板上下文前先对该本地计分板上下文进行刷新:

如果根据该数据帧的 SN 更新本地计分板上下文后, 更新后的本地计分板上下文至少部分会覆盖更新之前的范围:

SN 轮数字段的数值与上一次接收到的数据帧携带的 SN 轮数字段的数值不同。

示例性的, SN 轮数字段可以包含于如图 5 所示的新建的 HE 类型的 HT 控制类型 (通过表 2 所示的第一列中没有使用的 ID 新建的控制类型, 如图 6c 所示)。对于该 SN 轮数字段的比特长度, 本申请实施例不作限定。例如, 可以为 5 个比特等。示例性的, SN 轮数字段可以通过新建的 A-control 类型来承载, 或者承载在 CAS 控制类型的 Reserved 比特中, 如可以参考如图 5 所示的指示信息的承载位置。

可理解, 如果 SN 轮数字段长度有限 (如 5 个比特), 则有可能增加 32 次之后回复到原来的数值, 这样接收端可能无法判断 SN 轮数数值是否发生过变化。但是只要保证 SN 轮数字段长度足够长, 发生该情况的概率就会非常低, 可以将其作为一个小概率事件而忽略。

本申请实施例还提供了一种计分板上下文控制操作, 如下所示:

如果隶属于接收 MLD 中的一个 STA 不能使用重新排序缓冲区信息更新其记分板上下文, 且接收 MLD 在每条链路上都有一个单独的计分板上下文控制, 则 STA 应实施部分状态操作 (指的是 BA 机制的部分状态操作), 并应在以下定义的时间段内丢弃临时记录:

在发送 BA 之后, 该 BA 和已确认的 A-MPDU 在同一个传输机会 (transmission opportunity,

TXOP) 中), 并且在处理在同一个链路上接收到来自 initiator MLD 的属于同一个 TID 的下一个 QoS 数据帧的记分板上下文之前;

假如 BA 没有在当前 TXOP 结束之前发送, 则在一个新 TXOP 内在同一个链路上接收到来自 initiator MLD 的下一个 QoS 数据帧之后, 并且更新的记分板上下文之前。(If a STA affiliated with a recipient MLD in a link is not capable of using reordering buffer information to update its scoreboard context and the recipient MLD has a separate scoreboard context control in each link, the STA shall implement the partial-state operation and should discard the temporary record in the following defined time periods:

After sending a BA where the BA and the acknowledged A-MPDU(s) are in one TXOP and before processing the scoreboard context of the next received the QoS Data frame of the TID from the initiator MLD in the link if BA is transmitted;

Immediate before processing the scoreboard context of the next received the QoS Data frame of the TID from the initiator MLD in the link in a new TXOP if BA is not transmitted at the end of the current TXOP (也即 After received the next QoS Data frame of the TID from the initiator MLD in the link in a new TXOP and before processing the scoreboard context, if BA is not transmitted at the end of the current TXOP)) .

可理解, 以上所示的各个实施例中, 其中一个实施例未详细描述的实现方式, 还可以参考其他实施例。

如上文记载: 记分板的窗口大小可以是 64, 128, 256, 512, 1024 等。因此, 上述所示的第三种记分板上下文控制操作至第十种记分板上下文控制操作中涉及的 2^{11} 均是以 $\text{WinSize}_R=1024$ 为例示出的。

当记分板的窗口大小不是 1024 时, 以上所示的第三种记分板上下文控制操作至第十种记分板上下文控制操作中的至少一项中涉及的 2^{11} 也可以随着变化。

示例性的, 以第三种记分板上下文控制操作为例, 实际上当接收到一个 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + Q * \text{WinSize}_R$ 范围内的帧时会需要冲刷其记分板, 如 Q 等于 2, 或者, Q 还可以等于 3 等。为便于描述, 下文将以 $Q=2$ 为例说明本申请提供的方法。

当 $Q=2$ 时, 因为当 $\text{WinStart}_R + 2 * \text{WinSize}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$ 时, 第一 STA 会更新 WinStart_R 和 WinEnd_R , 而记分板 (新的 $[\text{WinStart}_R, \text{WinEnd}_R]$) 与更新前的记分板范围 (旧的 $[\text{WinStart}_R, \text{WinEnd}_R]$) 不重叠, 记分板上不会遗留旧数据帧的接收记录; 也就是不会把旧的数据的记录通过 BA 反馈给发送站点。

可理解, 如果考虑到临界点, 则以上所示的 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$ 还可以理解为: $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11} - 1$ 。以及下文所示的 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R$ 也可以理解为: $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$, 或者 $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R + \text{WinSize}_R$ 。也就是说, 如果考虑到数值的取整或公式的简洁等, 本申请所涉及的 SN 所满足的范围可以基于一定的范围 (如在 -2 到 +2 的范围) 进行调整, 这里不再一一列举。

为便于描述, 下文将以 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$ 为例进行说明。

以 $Q=2$ 为例, 则第三种记分板上下文控制操作还可以理解为:

接收 MLD 在每条链路上都有一个单独的记分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新记分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

则该某个 STA 可以刷新该某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

(If the following conditions are true: a recipient MLD has a separate scoreboard context control in each link; The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context; a STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_R \leq \text{SN} < \text{WinEnd}_R + \text{WinSize}_R$; the other STA affiliated with the MLD receives a frame with SN that is $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$; the STA shall flush the scoreboard context and update the scoreboard context.)

针对上述第三种上下文控制操作, 图 10a 是以 $\text{WinSize}_R=256$ 为例的一个流程, 在第一 STA 未接收到第六个 A-MPDU (如图 10a 所示的 A-MPDU6) 之前, 第一 STA 的计分板上下文中的 $\text{WinStart}_R=0$, $\text{WinEnd}_R=255$ 。当接收端接收到第六个 A-MPDU 时, 由于 SN 为 256-400, 该第六个 A-MPDU 的 SN 既不属于上述第一种计分板上下文控制操作, 也不属于上述第二种计分板上下文控制操作。由此, 第一 STA 就无法获知到底该如何更新其计分板上下文。也可以理解为: 当接收端接收到第六个 A-MPDU 时, 上述第二种计分板上下文控制操作中并没有考虑当第一 STA 接收到的 MPDU 的 SN 处于 $\text{WinEnd}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$ 这个范围。如果直接将第一 STA 的计分板中的 WinEnd_R 移动到 400 的位置, WinStart_R 在 145 (即 400-255), 则在反馈 BA 帧的时候会把上一轮的 145~255 的状态信息错误地反馈回去。

而在图 10b 中, 在第一 STA 未接收到第六个 A-MPDU (如图 10b 所示的 A-MPDU6) 之前, 第一 STA 的计分板上下文中的 $\text{WinStart}_R=0$, $\text{WinEnd}_R=255$ 。当接收端接收到第六个 A-MPDU 时, 由于 SN 为 512-767, 该第六个 A-MPDU 的 SN 既不属于上述第一种计分板上下文控制操作, 也不属于上述第二种计分板上下文控制操作。由此, 第一 STA 就无法获知到底该如何更新其计分板上下文。也可以理解为: 当接收端接收到第六个 A-MPDU 时, 直接将第一 STA 的计分板中的 WinEnd_R 移动到 767 的位置, WinStart_R 在 512, 则在反馈 BA 帧的时候不会把上一轮的 0~255 范围内的任何一个帧的状态信息错误地反馈回去。因此针对第六个 A-MPDU 不需要冲刷计分板。可理解, 图 10b 所示的例子中省略号省略的是其他未示出的 A-MPDU。

以 $Q=2$ 为例, 则第七种计分板上下文控制操作还可以理解为:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

在当前接收到的 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述条件; 或者, 该某个 STA 接收到的上述具有 SN 的 MPDU 是 A-MPDU 中满足以上条件的第一个 MPDU; 或者, 该某个 STA 的计分板上下文还没有基于当前接收到的 A-MPDU 中的 MPDU 进行刷新;

则该某个 STA 可以刷新该某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

以 $Q=2$ 为例, 则第八种计分板上下文控制操作还可以理解为:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU; 或者, 在当前接收到的 A-MPDU 之前接收的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应; 或者, 在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应;

则该某个 STA 可以刷新某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

以 $Q=2$ 为例, 则第九种计分板上下文控制操作还可以理解为:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$;

隶属于接收 MLD 的其他 STA 已经接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

在当前接收到的 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述所有条件; 或者, 该某个 STA 接收到的上述具有 SN 的 MPDU 是当前接收到的 A-MPDU 中满足以上条件的第一个 MPDU; 或者, 该某个 STA 的计分板上下文还没有基于当前接收到的 A-MPDU 中的 MPDU 进行刷新;

不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU; 或者, 在当前接收到的 A-MPDU 之前接收的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应; 或者, 在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应;

则该某个 STA 可以刷新某个 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

以 $Q=2$ 为例, 则第十种计分板上下文控制操作还可以理解为:

接收 MLD 在每条链路上都有一个单独的计分板上下文控制;

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文;

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$;

该某个 STA 接收到的 MPDU 中没有任何 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$; 或者, 隶属于接收 MLD 的其他 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$;

不存在任意一个在当前接收到的 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU; 或者, 在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中没有任

意一个 MPDU 没有基于 BA 响应；或者，在当前接收到的 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应；

可理解，以上所示的各个条件，在 STA 执行时，不限定先后顺序。

在当前接收到的 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足上述所有条件；或者，该某个 STA 接收到的上述具有 SN 的 MPDU 是 A-MPDU 中满足以上条件的第一个 MPDU；或者，该某个 STA 的计分板上下文还没有基于当前接收到的 A-MPDU 中的 MPDU 进行刷新；

则该某个 STA 可以刷新该 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

可理解，以上所示的计分板上下文控制操作的说明仅为示例，本申请上文涉及的 2^{11} 与计分板窗口之间的关系这里不再一一详述。示例性的，将 SN 所满足的条件与计分板窗口的大小结合，则上文所示的 2^{11} 均可以替换为 $2 * \text{WinSize}_R$ 。

结合以上所示的第三种计分板上下文控制操作和第六种计分板上下文控制操作，则本申请还提供了第十一种计分板上下文控制操作：

接收 MLD 在每条链路上都有一个单独的计分板上下文控制；

隶属于接收 MLD 的 STA 都可以使用重新排序的缓冲区信息更新计分板上下文；

隶属于该接收 MLD 的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2 * \text{WinSize}_R - 1$ ；

该某个 STA 接收到的 MPDU 中没有任何 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ ；或者，隶属于接收 MLD 的其他 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ ；

则该某个 STA 可以刷新该 STA 的计分板上下文并更新该某个 STA 的计分板上下文。

以下将介绍本申请实施例提供的通信装置。

本申请根据上述方法实施例对通信装置进行功能模块的划分，例如，可以对应各个功能划分各个功能模块，也可以将两个或两个以上的功能集成在一个处理模块中。上述集成的模块既可以采用硬件的形式实现，也可以采用软件功能模块的形式实现。需要说明的是，本申请中对模块的划分是示意性的，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式。下面将结合图7至图9详细描述本申请实施例的通信装置。

图7是本申请实施例提供的一种通信装置的结构示意图，如图7所示，该通信装置包括处理单元701、收发单元702。

在本申请的一些实施例中，该通信装置可以是上文示出的发送端。即图7所示的通信装置可以用于执行上文方法实施例中由发送端执行的步骤或功能等。

例如，收发单元702，用于发送ADDBA请求帧，以及接收ADDBA响应帧。

又例如，收发单元702，还用于发送第一A-MPDU。

又例如，收发单元702，还用于发送BAR帧，以及接收BA帧。

又例如，收发单元702，还用于发送第二A-MPDU。

可理解，本申请实施例示出的收发单元和处理单元的具体说明仅为示例，对于收发单元和处理单元的具体功能或执行的步骤等，可以参考上述方法实施例（如包括图4和图5等），这里不再详述。

复用图7，在本申请的另一一些实施例中，该通信装置可以是上文示出的接收端。即图7所示的通信装置可以用于执行上文方法实施例中由接收端执行的步骤或功能等。

例如，收发单元702，用于接收ADDBA请求帧，以及发送ADDBA响应帧。

又例如，收发单元702，还用于接收第一A-MPDU。

又例如，处理单元701，用于获取本地记录的计分板上下文，并在第一A-MPDU满足本申请实施例上文所示的如第三种计分板上下文控制操作至第十种计分板上下文控制操作中的至少一项时，根据该第一A-MPDU的SN刷新计分板上下文。

又例如，收发单元702，还用于接收BAR帧，以及发送BA帧。

又例如，收发单元702，还用于接收第二A-MPDU；处理单元701，还用于根据该第二A-MPDU中的指示信息刷新计分板上下文。

可理解，关于第三种计分板上下文控制操作至第十种计分板上下文控制操作等的具体说明可以参考上文所示的方法实施例，这里不再一一详述。

可理解，本申请实施例示出的收发单元和处理单元的具体说明仅为示例，对于收发单元和处理单元的具体功能或执行的步骤等，可以参考上述方法实施例（如包括图4和图5等），这里不再详述。

以上介绍了本申请实施例的发送端和接收端，以下介绍所述发送端和接收端可能的产品形态。应理解，但凡具备上述图7所述的发送端的功能的任何形态的产品，或者，但凡具备上述图7所述的接收端的功能的任何形态的产品，都落入本申请实施例的保护范围。还应理解，以下介绍仅为举例，不限制本申请实施例的发送端和接收端的产品形态仅限于此。

图7所示的通信装置中，处理单元701可以是一个或多个处理器，收发单元702可以是收发器，或者收发单元702还可以是发送单元和接收单元，发送单元可以是发送器，接收单元可以是接收器，该发送单元和接收单元集成于一个器件，例如收发器。本申请实施例中，处理器和收发器可以被耦合等，对于处理器和收发器的连接方式，本申请实施例不作限定。

如图8所示，该通信装置80包括一个或多个处理器820和收发器810。

示例性的，当该通信装置用于执行上述发送端执行的步骤或方法或功能时，例如，收发器810，用于发送ADDBA请求帧，以及接收ADDBA响应帧。又例如，收发器810，还用于发送第一A-MPDU。又例如，收发器810，还用于发送BAR帧，以及接收BA帧。又例如，收发器810，还用于发送第二A-MPDU。

示例性的，当该通信装置用于执行上述接收端执行的步骤或方法或功能时，例如，收发器810，用于接收ADDBA请求帧，以及发送ADDBA响应帧。又例如，收发器810，还用于接收第一A-MPDU。又例如，处理器820，用于获取本地记录的计分板上下文，并在第一A-MPDU满足本申请实施例上文所示的如第三种计分板上下文控制操作至第十种计分板上下文控制操作中的至少一项时，根据该第一A-MPDU的SN刷新计分板上下文。又例如，收发器810，还用于接收BAR帧，以及发送BA帧。又例如，收发器810，还用于接收第二A-MPDU；处理器820，还用于根据该第二A-MPDU中的指示信息刷新计分板上下文。

可理解，对于处理器和收发器的具体说明还可以参考图7所示的处理单元和收发单元的介绍，这里不再赘述。

在图8所示的通信装置的各个实现方式中，收发器可以包括接收机和发射机，该接收机用于执行接收的功能（或操作），该发射机用于执行发射的功能（或操作）。以及收发器用于通过传输介质和其他设备/装置进行通信。

可选的，通信装置80还可以包括一个或多个存储器830，用于存储程序指令和/或数据。存储器830和处理器820耦合。本申请实施例中的耦合是装置、单元或模块之间的间接耦合

或通信连接，可以是电性，机械或其它的形式，用于装置、单元或模块之间的信息交互。处理器 820 可能和存储器 830 协同操作。处理器 820 可以执行存储器 830 中存储的程序指令。可选的，上述一个或多个存储器中的至少一个可以包括于处理器中。

本申请实施例中不限定上述收发器 810、处理器 820 以及存储器 830 之间的具体连接介质。本申请实施例在图 8 中以存储器 830、处理器 820 以及收发器 810 之间通过总线 840 连接，总线在图 8 中以粗线表示，其它部件之间的连接方式，仅是进行示意性说明，并不引以为限。所述总线可以分为地址总线、数据总线、控制总线等。为便于表示，图 8 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。

在本申请实施例中，处理器可以是通用处理器、数字信号处理器、专用集成电路、现场可编程门阵列或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等，可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成，或者用处理器中的硬件及软件模块组合执行完成等。

本申请实施例中，存储器可包括但不限于硬盘（hard disk drive, HDD）或固态硬盘（solid-state drive, SSD）等非易失性存储器，随机存储记忆体（Random Access Memory, RAM）、可擦除可编程只读存储器（Erasable Programmable ROM, EPROM）、只读存储器（Read-Only Memory, ROM）或便携式只读存储器（Compact Disc Read-Only Memory, CD-ROM）等等。存储器是能够用于携带或存储具有指令或数据结构形式的程序代码，并能够由计算机（如本申请示出的通信装置等）读和/写的任何存储介质，但不限于此。本申请实施例中的存储器还可以是电路或者其它任意能够实现存储功能的装置，用于存储程序指令和/或数据。

处理器 820 主要用于对通信协议以及通信数据进行处理，以及对整个通信装置进行控制，执行软件程序，处理软件程序的数据。存储器 830 主要用于存储软件程序和数据。收发器 810 可以包括控制电路和天线，控制电路主要用于基带信号与射频信号的转换以及对射频信号的处理。天线主要用于收发电磁波形式的射频信号。输入输出装置，例如触摸屏、显示屏，键盘等主要用于接收用户输入的数据以及对用户输出数据。

当通信装置开机后，处理器 820 可以读取存储器 830 中的软件程序，解释并执行软件程序的指令，处理软件程序的数据。当需要通过无线发送数据时，处理器 820 对待发送的数据进行基带处理后，输出基带信号至射频电路，射频电路将基带信号进行射频处理后将射频信号通过天线以电磁波的形式向外发送。当有数据发送到通信装置时，射频电路通过天线接收到射频信号，将射频信号转换为基带信号，并将基带信号输出至处理器 820，处理器 820 将基带信号转换为数据并对该数据进行处理。

在另一种实现中，所述的射频电路和天线可以独立于进行基带处理的处理器而设置，例如在分布式场景中，射频电路和天线可以与独立于通信装置，呈拉远式的布置。

可理解，本申请实施例示出的通信装置还可以具有比图8更多的元器件等，本申请实施例对此不作限定。以上所示的处理器和收发器所执行的方法仅为示例，对于该处理器和收发器具体所执行的步骤可参照上文介绍的方法。

在另一种可能的实现方式中，图7所示的通信装置中，处理单元701可以是一个或多个逻辑电路，收发单元702可以是输入输出接口，又或者称为通信接口，或者接口电路，或接口等等。或者收发单元702还可以是发送单元和接收单元，发送单元可以是输出接口，接收单元可以是输入接口，该发送单元和接收单元集成于一个单元，例如输入输出接口。如图9所示，图

9所示的通信装置包括逻辑电路901和接口902。即上述处理单元701可以用逻辑电路901实现，收发单元702可以用接口902实现。其中，该逻辑电路901可以为芯片、处理电路、集成电路或片上系统（system on chip, SoC）芯片等，接口902可以为通信接口、输入输出接口、管脚等。示例性的，图9是以上述通信装置为芯片为例出的，该芯片包括逻辑电路901和接口902。

本申请实施例中，逻辑电路和接口还可以相互耦合。对于逻辑电路和接口的具体连接方式，本申请实施例不作限定。

示例性的，当通信装置用于执行上述发送端执行的方法或功能或步骤时，例如，接口902，用于发送ADDBA请求帧，以及接收ADDBA响应帧。又例如，接口902，还用于发送第一A-MPDU。又例如，接口902，还用于发送BAR帧，以及接收BA帧。又例如，接口902，还用于发送第二A-MPDU。

示例性的，当通信装置用于执行上述接收端执行的方法或功能或步骤时，例如，接口902，用于接收ADDBA请求帧，以及发送ADDBA响应帧。又例如，接口902，还用于接收第一A-MPDU。又例如，逻辑电路901，用于获取本地记录的记分板上下文，并在第一A-MPDU满足本申请实施例上文所示的如第三种记分板上下文控制操作至第十种记分板上下文控制操作中的至少一项时，根据该第一A-MPDU的SN刷新记分板上下文。又例如，接口902，还用于接收BAR帧，以及发送BA帧。又例如，接口902，还用于接收第二A-MPDU；逻辑电路901，还用于根据该第二A-MPDU中的指示信息刷新记分板上下文。

可理解，本申请实施例示出的通信装置可以采用硬件的形式实现本申请实施例提供的方法，也可以采用软件的形式实现本申请实施例提供的方法等，本申请实施例对此不作限定。

对于图9所示的各个实施例的具体实现方式，还可以参考上述各个实施例，这里不再详述。

本申请实施例还提供了一种无线通信系统，该无线通信系统包括发送端和接收端，该发送端和该接收端可以用于执行前述任一实施例中的方法（如图4和图5等）。

此外，本申请还提供一种计算机程序，该计算机程序用于实现本申请提供的方法中由发送端执行的操作和/或处理。

本申请还提供一种计算机程序，该计算机程序用于实现本申请提供的方法中由接收端执行的操作和/或处理。

本申请还提供一种计算机可读存储介质，该计算机可读存储介质中存储有计算机代码，当计算机代码在计算机上运行时，使得计算机执行本申请提供的方法中由发送端执行的操作和/或处理。

本申请还提供一种计算机可读存储介质，该计算机可读存储介质中存储有计算机代码，当计算机代码在计算机上运行时，使得计算机执行本申请提供的方法中由接收端执行的操作和/或处理。

本申请还提供一种计算机程序产品，该计算机程序产品包括计算机代码或计算机程序，当该计算机代码或计算机程序在计算机上运行时，使得本申请提供的方法中由发送端执行的操作和/或处理被执行。

本申请还提供一种计算机程序产品，该计算机程序产品包括计算机代码或计算机程序，当该计算机代码或计算机程序在计算机上运行时，使得本申请提供的方法中由接收端执行的操作和/或处理被执行。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结

合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另外，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口、装置或单元的间接耦合或通信连接，也可以是电的，机械的或其它的形式连接。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本申请实施例提供的方案的技术效果。

另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以是两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

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

以上所述，仅为本申请的具体实施方式，但本申请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申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以所述权利要求的保护范围为准。

权 利 要 求 书

1.一种多链路通信方法,其特征在于,所述方法包括:

接收端接收第一聚合的媒介接入控制层协议数据单元 A-MPDU;

所述接收端获取本地记录的计分板上下文,在所述第一 A-MPDU 满足如下任一项或多项条件时,根据所述第一 A-MPDU 的序列号 SN 刷新所述计分板上下文;

第一条件包括:隶属于所述接收端的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$;

第二条件包括:SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的任何一个 MPDU 都不是通过所述某个 STA 接收的;或者,SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的所有 MPDU 都是由隶属于所述接收端的其他 STA 接收的;

第三条件包括:不存在任意一个在接收到所述第一 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU;或者,在所述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应;或者,在所述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应。

2.根据权利要求 1 所述的方法,其特征在于,在所述第一 A-MPDU 还满足以下第四条件:

所述第四条件包括:在所述第一 A-MPDU 中没有其他具有 SN 的 MPDU 已经满足所述第一条件、所述第二条件和所述第三条件;或者,隶属于所述接收端的某个 STA 接收到的具有 SN 的 MPDU 是所述第一 A-MPDU 中满足所述第一条件、所述第二条件和所述第三条件的第一个 MPDU;或者,隶属于所述接收端的某个 STA 的计分板上下文还没有基于所述第一 A-MPDU 中的 MPDU 进行刷新。

3.根据权利要求 1 或 2 所述的方法,其特征在于,所述接收端在每条链路上都有一个单独的计分板上下文控制,所述链路为所述接收端与发送端之间的链路。

4.根据权利要求 1-3 任一项所述的方法,其特征在于,隶属于所述接收端的 STA 具有使用重新排序的缓冲区信息更新计分板上下文的能力。

5.根据权利要求 1-4 任一项所述的方法,其特征在于,所述第一 A-MPDU 中的 SN 属于同一个业务标识 TID。

6.一种通信装置,其特征在于,包括:

收发单元,用于接收第一聚合的媒介接入控制层协议数据单元 A-MPDU;

处理单元,用于获取本地记录的计分板上下文,在所述第一 A-MPDU 满足如下任一项或多项条件时,根据所述第一 A-MPDU 的序列号 SN 刷新所述计分板上下文;

第一条件包括:隶属于所述通信装置的某个 STA 接收到的 MPDU 的 SN 满足 $\text{WinStart}_R \leq \text{SN} < \text{WinStart}_R + 2^{11}$;

第二条件包括:SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的任何一个 MPDU 都不是通过所述某个 STA 接收的;或者,SN 满足 $\text{WinStart}_B - \text{WinSize}_R \leq \text{SN} < \text{WinStart}_B$ 的所有 MPDU 都是由隶属于所述通信装置的其他 STA 接收的;

第三条件包括:不存在任意一个在接收到所述第一 A-MPDU 之前接收的另一个 A-MPDU 中的没有基于 BA 响应的 MPDU;或者,在所述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中没有任意一个 MPDU 没有基于 BA 响应;或者,在所述第一 A-MPDU 之前接收到的任意一个 A-MPDU 中承载的所有 MPDU 已基于 BA 响应。

- 7.根据权利要求6所述的装置,其特征在于,所述第一A-MPDU还满足以下第四条件:
所述第四条件包括:在所述第一A-MPDU中没有其他具有SN的MPDU已经满足所述第一条件、所述第二条件和所述第三条件;或者,隶属于所述通信装置的某个STA接收到的具有SN的MPDU是所述第一A-MPDU中满足所述第一条件、所述第二条件和所述第三条件的第一个MPDU;或者,隶属于所述通信装置的某个STA的计分板上下文还没有基于所述第一A-MPDU中的MPDU进行刷新。
- 8.根据权利要求6或7所述的装置,其特征在于,所述通信装置在每条链路上都有一个单独的计分板上下文控制,所述链路为所述通信装置与发送端之间的链路。
- 9.根据权利要求6-8任一项所述的装置,其特征在于,隶属于所述通信装置的STA具有使用重新排序的缓冲区信息更新计分板上下文的能力。
- 10.根据权利要求6-9任一项所述的装置,其特征在于,所述第一A-MPDU中的SN属于同一个业务标识TID。
- 11.一种通信装置,其特征在于,包括处理器和存储器;
所述存储器用于存储指令;
所述处理器用于执行所述指令,以使权利要求1至5任一项所述的方法被执行。
- 12.一种通信装置,其特征在于,包括逻辑电路和接口,所述逻辑电路和接口耦合;
所述接口用于输入和/或输出代码指令,所述逻辑电路用于执行所述代码指令,以使权利要求1至5任一项所述的方法被执行。
- 13.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质用于存储计算机程序,当所述计算机程序被执行时,权利要求1至5任一项所述的方法被执行。
- 14.一种计算机程序,其特征在于,所述计算机程序被执行时,权利要求1至5任一项所述的方法被执行。