说明书

发明名称: 报文发送方法、数据处理器、计算机设备及存储介质

[0001] 相关申请

[0002] 本申请要求2024年02月27日申请的，申请号为2024102177480，名称为“报文发送方法、数据处理器、计算机设备及存储介质”的中国专利申请的优先权，在此将其全文引入作为参考。

技术领域

[0003] 本申请涉及计算机技术领域，特别是涉及一种报文发送方法、数据处理器、计算机设备、存储介质和计算机程序产品。

背景技术

[0004] 远程直接内存访问(Remote Direct Memory Access，简称RDMA)网卡需要使用拥塞控制算法来检测和缓解网络中的拥塞，目前主流的拥塞控制算法是数据中心量化拥塞通知(Data Center Quantized Congestion Notification，简称DCQCN)算法。DCQCN算法遵循RP-CP-NP模型，数据中心包括数据发送服务器集群和数据存储服务器集群，数据发送服务器集群的网卡可以称为反应点(Reaction Point，简称RP)，数据存储服务器集群的网卡可以称为通知点(Notification Point，简称NP)，数据发送服务器集群和数据存储服务器集群中间的交换机可以称为拥塞点(Congestion Point，CP)。对于数据发送服务器集群的网卡，硬件报文处理与DCQCN算法执行在逻辑上可以看成是彼此独立的模块，可分别称为报文处理模块和算法执行模块。

[0005] 传统技术中，报文处理模块在收到拥塞通知报文(Congestion Notification Packet，简称CNP)后，向算法执行模块上报一个CNP事件，算法执行模块运行DCQCN算法的降速处理逻辑，将降速处理后的报文发送速率作为响应反馈给报文处理模块。另外，算法执行模块将“一段时间内没有收到CNP事件”作为自触发事件，事件触发后运行DCQCN算法的升速处理逻辑，将升速处理后的报文发送速率作为响应反馈给报文处理模块。报文处理模块在收到算法执行模块反馈的速率后，以该速率约束对应流的报文发送行为。

[0006] 然而，将速率作为响应反馈给报文处理模块的方式，在算法执行模块响应性能抖动时，容易导致异常拥塞。

发明内容

[0007] 基于此，根据本申请的各种实施例，提供一种报文发送方法、数据处理器、计算机设备、存储介质和计算机程序产品。

[0008] 第一方面，本申请提供了一种报文发送方法，应用于数据处理器，数据处理器包括报文处理模块和算法执行模块，方法包括：

[0009] 报文处理模块在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件；

[0010] 算法执行模块对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块；

[0011] 报文处理模块基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。

[0012] 在一些实施例中，基于当前令牌数量发送待处理数据流对应的报文，包括：

[0013] 判断当前令牌数量是否大于或者等于预设门限值；

[0014] 若当前令牌数量大于或者等于预设门限值，发送待处理数据流对应的当前待发送报文，根据当前待发送报文的大小更新当前令牌数量；并判断待处理数据流是否仍有报文需要发送；在待处理数据流仍有报文需要发送的情况下，返回判断当前令牌数量是否大于或者等于预设门限值的步骤；在待处理数据流没有报文需要发送的情况下，调度待调度流队列中待处理数据流的下一数据流；及

[0015] 若当前令牌数量小于预设门限值，则将待处理数据流放入待调度流队列的队尾，等待下一次调度。

[0016] 在一些实施例中，报文发送方法还包括：

[0017] 根据当前待发送报文的大小更新单次突发报文大小；及

[0018] 在确定待处理数据流仍有报文需要发送的情况下，继续判断单次突发报文大小是否达到单次突发的数量上限；在未达到的情况下，返回判断当前令牌数量是

否大于或者等于预设门限值的步骤；在达到的情况下，调度待调度流队列中待处理数据流的下一数据流。

[0019] 在一些实施例中，将报文发送速率映射为目标令牌数量，包括：

[0020] 基于令牌事件中携带的令牌事件时间戳、预先设置的令牌基准时间戳以及令牌生成时间段，确定令牌事件对应的令牌请求序号；

[0021] 判断令牌请求序号是否大于令牌发放序号；

[0022] 在大于的情况下，确定还未针对令牌请求序号发放过令牌，则基于报文发送速率和令牌生成时间段，确定目标令牌数量，并基于令牌请求序号对令牌发放序号进行更新；及

[0023] 在小于或者等于的情况下，确定已针对令牌请求序号发放过令牌，则将预设值作为目标令牌数量。

[0024] 在一些实施例中，将报文发送速率映射为目标令牌数量之前还包括：

[0025] 基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于时间周期的升速调节机制的触发条件，在满足的情况下，采用基于时间周期的升速调节机制对报文发送速率进行升速调节；

[0026] 在一些实施例中，将报文发送速率映射为目标令牌数量之前还包括：

[0027] 基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于已发报文大小的升速调节机制的触发条件，在满足的情况下，采用基于已发报文大小的升速调节机制对报文发送速率进行升速调节。

[0028] 在一些实施例中，基于时间周期的升速调节机制的触发条件为：

$M \times T \leqslant t = ts_current - ts_last_t_increase \leqslant (M+1) \times T$

[0029] 其中，T为预设的升速时间周期，ts_current为令牌事件时间戳，ts_last_t_increase为上一次针对待处理数据流采用第一升速调节机制进行升速调节时的时间戳，M为正整数；

[0030] 在一些实施例中，采用基于时间周期的升速调节机制对报文发送速率进行升速调节，包括：

[0031] 采用基于时间周期的升速调节机制对报文发送速率连续升速M次。

[0032] 在一些实施例中，基于已发报文大小的升速调节机制的触发条件为：

$N \times B \leqslant b \leqslant (N+1) \times B$

[0033] 其中，B为预设的升速报文大小，b为令牌事件时间戳与上一次针对待处理数据流采用第二升速调节机制进行升速调节时的时间戳之间，已发送的待处理数据流的报文大小，N为正整数；

[0034] 在一些实施例中，采用基于已发报文大小的升速调节机制对报文发送速率进行升速调节，包括：

[0035] 采用基于已发报文大小的升速调节机制对报文发送速率连续升速N次。

[0036] 在一些实施例中，将报文发送速率映射为目标令牌数量，包括：

[0037] 若待处理数据流被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；

[0038] 若还未针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，从全额令牌数量中确定发放令牌数量，将发放令牌数量作为目标令牌数量；若是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量，并取消待处理数据流的饥饿模式标记；及

[0039] 若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，将预设值作为目标令牌数量；若是饥饿类型，取消待处理数据流的饥饿模式标记，并基于全额令牌数量和发放令牌数量确定保留令牌数量，将保留令牌数量作为目标令牌数量。

[0040] 在一些实施例中，报文发送方法还包括：

[0041] 若待处理数据流未被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；

[0042] 若还未针对令牌请求序号发放过令牌，基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量；及

[0043] 若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为已获得令牌的队列类型，若是，将预设值作为目标令牌数量；若否，将待

处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并将预设值作为目标令牌数量。

[0044] 第二方面，本申请还提供了一种数据处理器，包括：

[0045] 报文处理模块，用于在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件；

[0046] 算法执行模块，用于对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块；

[0047] 报文处理模块，还用于基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。

[0048] 在一些实施例中，报文处理模块，具体用于判断当前令牌数量是否大于或者等于所述预设门限值；若当前令牌数量大于或者等于所述预设门限值，发送待处理数据流对应的当前待发送报文，根据所述当前待发送报文的大小更新当前令牌数量；并判断待处理数据流是否仍有报文需要发送；在待处理数据流仍有报文需要发送的情况下，返回判断当前令牌数量是否大于或者等于所述预设门限值的步骤；在待处理数据流没有报文需要发送的情况下，调度待调度流队列中待处理数据流的下一数据流；若当前令牌数量小于所述预设门限值，则将所述待处理数据流放入待调度流队列的队尾，等待下一次调度。

[0049] 在一些实施例中，报文处理模块，还用于根据所述当前待发送报文的大小更新单次突发报文大小；在确定待处理数据流仍有报文需要发送的情况下，继续判断单次突发报文大小是否达到单次突发的数量上限；在未达到的情况下，返回判断当前令牌数量是否大于或者等于所述预设门限值的步骤；在达到的情况下，调度待调度流队列中待处理数据流的下一数据流。

[0050] 在一些实施例中，算法执行模块，具体用于基于令牌事件中携带的令牌事件时间戳、预先设置的令牌基准时间戳以及令牌生成时间段，确定令牌事件对应的令牌请求序号；判断令牌请求序号是否大于令牌发放序号；在大于的情况下，确定还未针对令牌请求序号发放过令牌，则基于报文发送速率和令牌生成时间段，确定目标令牌数量，并基于令牌请求序号对令牌发放序号进行更新；在小

于或者等于的情况下，确定已针对令牌请求序号发放过令牌，则将预设值作为目标令牌数量。

[0051] 在一些实施例中，算法执行模块，具体用于基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于时间周期的升速调节机制的触发条件，在满足的情况下，采用基于时间周期的升速调节机制对报文发送速率进行升速调节；和/或；基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于已发报文大小的升速调节机制的触发条件，在满足的情况下，采用基于已发报文大小的升速调节机制对报文发送速率进行升速调节。

[0052] 在一些实施例中，基于时间周期的升速调节机制的触发条件为：

$M \times T \leqslant t = ts_current - ts_last_t_increase \leqslant (M+1) \times T$

[0053] 其中，T为预设的升速时间周期，ts_current为令牌事件时间戳，ts_last_t_increase为上一次针对待处理数据流采用第一升速调节机制进行升速调节时的时间戳，M为正整数；

[0054] 算法执行模块，具体用于采用基于时间周期的升速调节机制对报文发送速率连续升速M次。

[0055] 在一些实施例中，基于已发报文大小的升速调节机制的触发条件为：

$N \times B \leqslant b \leqslant (N+1) \times B$

[0056] 其中，B为预设的升速报文大小，b为令牌事件时间戳与上一次针对待处理数据流采用第二升速调节机制进行升速调节时的时间戳之间，已发送的待处理数据流的报文大小，N为正整数；

[0057] 算法执行模块，具体用于采用基于已发报文大小的升速调节机制对报文发送速率连续升速N次。

[0058] 在一些实施例中，算法执行模块，具体用于若待处理数据流被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，从全额令牌数量中确定发放令牌数量，将发放令牌

数量作为目标令牌数量；若是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量，并取消待处理数据流的饥饿模式标记；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，将预设值作为目标令牌数量；若是饥饿类型，取消待处理数据流的饥饿模式标记，并基于全额令牌数量和发放令牌数量确定保留令牌数量，将保留令牌数量作为目标令牌数量。

[0059] 在一些实施例中，算法执行模块，具体用于若待处理数据流未被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为已获得令牌的队列类型，若是，将预设值作为目标令牌数量；若否，将待处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并将预设值作为目标令牌数量。

[0060] 第三方面，本申请还提供了一种计算机设备，包括第二方面提供的数据处理器。

[0061] 第四方面，本申请还提供了一种计算机可读存储介质，其上存储有计算机程序，计算机程序被处理器执行时实现以下步骤：

[0062] 报文处理模块在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件；

[0063] 算法执行模块对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块；

[0064] 报文处理模块基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。

[0065] 第五方面，本申请还提供了一种计算机程序产品，包括计算机程序，该计算机程序被处理器执行时实现以下步骤：

[0066] 报文处理模块在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件；

[0067] 算法执行模块对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块；

[0068] 报文处理模块基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。

[0069] 本发明的一个或多个实施例的细节在下面的附图和描述中提出。本发明的其他特征、目的和优点将从说明书、附图以及权利要求书变得明显。

附图说明

[0070] 为了更清楚地说明本申请实施例或传统技术中的技术方案，下面将对实施例或传统技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据公开的附图获得其他的附图。

[0071] 图1为一个实施例中报文发送方法的流程示意图；

[0072] 图2为另一个实施例中报文发送方法的流程示意图；

[0073] 图3为一个实施例中报文处理模块基于令牌数量的工作流程示意图；

[0074] 图4为一个实施例中算法执行模块的令牌发放机制的流程示意图；

[0075] 图5为一个实施例中算法执行模块的防饥饿处理机制的流程示意图；

[0076] 图6为一个实施例中数据处理器的结构框图；

[0077] 图7为一个实施例中计算机设备的内部结构图。

具体实施方式

[0078] 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

[0079] 对于RDMA网卡来说，目前主流的拥塞控制算法是DCQCN算法，DCQCN算法遵循RP-CP-NP模型，对于RP侧的网卡，算法执行模块通常将速率作为响应反馈给报文处理模块，报文处理模块在收到算法执行模块反馈的速率后，以该速率约束对应流的报文发送行为。这种方式在算法执行模块响应性能抖动时，容易导致异常拥塞。为此，本申请提供一种报文发送方法，对上述DCQCN算法进行了改进，本申请提供的报文发送方法可应用于RP侧的网卡，算法执行模块不是将速率直接反馈给报文处理模块，而是将速率映射为令牌数量反馈给报文处理模块，报文处理模块以令牌数量约束对应流的报文发送行为。可以避免算法执行模块接收到CNP事件后，降速处理逻辑的运行性能抖动时，无法及时将降速处理后的报文发送速率反馈给报文处理模块，而报文处理模块仍然以较高速率发送报文等情况引起的异常拥塞。下面结合具体实施例对本申请提供的报文发送方法进行详细说明。

[0080] 如上文描述，本申请提供的报文发送方法可应用于RP侧的数据处理器，RP侧的数据处理器可以为RDMA数据处理器，或者其他能够支持本申请提供的报文发送方法得以实现的数据处理器，本申请实施例对此不做限定，本申请所述数据处理器(Data Processing Unit,DPU)可以为网卡、智能网卡、或者芯片等。

[0081] 在一个示例性的实施例中，如图1所示，提供了一种报文发送方法，以该方法应用于RDMA数据处理器为例进行说明，RDMA数据处理器中硬件报文处理与DCQCN算法执行在逻辑上可以看成是彼此独立的模块，可分别称为报文处理模块和算法执行模块。本申请实施例提供的报文发送方法，包括以下步骤102至步骤106。其中：

[0082] 步骤102，报文处理模块在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件。

[0083] 可选的，本申请实施例中的待处理数据流为活跃流，活跃流为有报文需要发送的流。需要报文处理模块处理的所有活跃流均存放在待调度活跃流队列中，报文处理模块按照从队首到队尾的顺序，依次调度该队列中的活跃流，报文处理模块调度到一条活跃流后，可将该活跃流作为待处理数据流。

[0084] 其中，报文处理模块可为每条活跃流维护一个令牌数量，报文处理模块调度到一条活跃流后，将该活跃流作为待处理数据流，可查询待处理数据流的当前令牌数量，并判断待处理数据流的当前令牌数量是否小于预设门限，在小于预设门限的情况下，获取当前时刻的时间戳，将该时间戳作为令牌事件时间戳，基于令牌事件时间戳、待处理数据流的流标识、待处理数据流的报文队列类型等信息，生成令牌事件，将令牌事件发送给算法执行模块。

[0085] 其中，待处理数据流的报文队列类型可能为数据发送队列(data_send queue，简称SQ)，也可能为数据响应队列(data_response queue，简称RQ)。

[0086] 如上文描述，报文处理模块在待处理数据流的当前令牌数量小于预设门限的情况下，会向算法执行模块发送令牌事件；在待处理数据流的当前令牌数量大于或者等于预设门限的情况下，报文处理模块可发送待处理数据流对应的当前待发送报文，根据当前待发送报文的大小更新当前令牌数量以及单次突发报文大小，并判断待处理数据流是否仍有报文需要发送，在待处理数据流仍有报文需要发送的情况下，继续判断单次突发报文大小是否达到单次突发的数量上限；在未达到的情况下，返回判断当前令牌数量是否大于或者等于预设门限值的步骤；在达到的情况下，或者待处理数据流没有报文需要发送的情况下，调度待调度活跃流队列中待处理数据流的下一活跃流，在本申请实施例中，待调度活跃流队列也称为待调度流队列，调度活跃流队列中的活跃流也称为数据流。

[0087] 其中，预设门限值可根据实际情况灵活设定，示例性的，可以设置为1个最大传输单元(Maximum Transition Unit，简称MTU)的大小，例如1KB。本申请实施例中的令牌数量可以理解为令牌字节数。

[0088] 步骤104，算法执行模块对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块。

[0089] 其中，算法执行模块接收到令牌事件后，可对令牌事件进行解析得到待处理数据流的流标识以及令牌事件时间戳。算法执行模块可基于待处理数据流的流标识查询待处理数据流的报文发送速率，并基于待处理数据流的流标识以及令牌事件时间戳，判断是否需要对该报文发送速率进行升速调节。

[0090] 在无需进行升速调节的情况下，将该报文发送速率映射为目标令牌数量；将该目标令牌数量返回给报文处理模块。在需要进行升速调节的情况下，对该报文发送速率进行升速调节，将升速调节后的报文发送速率映射为目标令牌数量，将该目标令牌数量返回给报文处理模块。

[0091] 需要说明的是：报文处理模块在收到待处理数据流的CNP报文后，可向算法执行模块上报一个CNP事件，算法执行模块接收到CNP事件后，可对待处理数据流的报文发送速率进行降速调节。也就是说，随着待处理数据流的令牌事件的到来，或者待处理数据流的CNP事件的到来，算法执行模块可能会对待处理数据流的报文发送速率进行更新，在做映射处理时，是对最新的报文发送速率进行映射，得到目标令牌数量。

[0092] 步骤106，报文处理模块基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。

[0093] 其中，报文处理模块接收到目标令牌数量后，将待处理数据流的当前令牌数量加上目标令牌数量，从而实现对当前令牌数量的更新，更新后基于当前令牌数量发送待处理数据流对应的报文。

[0094] 可选的，待处理数据流需要发送的报文可能有N个，N个报文有发送顺序，报文处理模块可按照该顺序依次发送每个报文，直到N个报文全部发完为止。对于每个报文，报文处理模块将该报文发送给交换机，经由交换机发送给NP侧的数据处理器。交换机接收到报文后执行显式拥塞通知(Explicit Congestion Notification，简称ECN)标记机制，NP侧的数据处理器接收到交换机发送的报文后，执行CNP报文上报机制，这两个机制的详细内容可参见现有DCQCN算法，本申请实施例在此不再赘述。

[0095] 参见图2所示，本申请实施例提供的报文发送方法，整个过程大致为，报文处理模块在待处理数据流的当前令牌数量不足的情况下，向算法执行模块发送令牌事件；算法执行模块接收到令牌事件后，判断是否需要对待处理数据流的报文发送速率进行升速调节，在需要进行升速调节的情况下，对该报文发送速率进行升速调节，将升速调节后的报文发送速率映射为目标令牌数量返回给报文处理模块。报文处理模块在收到待处理数据流的CNP报文后，可向算法执行模

块上报CNP事件。算法执行模块接收到CNP事件后，可对待处理数据流的报文发送速率进行降速调节。

[0096] 上述实施例中，报文处理模块在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件；算法执行模块对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块；报文处理模块基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。算法执行模块不是将速率直接反馈给报文处理模块，而是将速率映射为令牌数量反馈给报文处理模块，报文处理模块以令牌数量约束对应流的报文发送行为；可以避免算法执行模块响应性能抖动时导致的异常拥塞，比如算法执行模块接收到CNP事件后，降速处理逻辑的运行性能存在抖动，无法及时将降速处理后的报文发送速率反馈给报文处理模块，而报文处理模块仍然以较高速率发送报文，这样可能导致网络拥塞，本申请提出的令牌机制可以避免这种拥塞。

[0097] 在一些实施例中，报文处理模块对当前令牌数量更新后，可通过如下方式发送待处理数据流对应的报文，即，上述基于当前令牌数量发送待处理数据流对应的报文的实现，包括如下步骤：

[0098] 判断当前令牌数量是否大于或者等于预设门限值；若当前令牌数量大于或者等于预设门限值，发送待处理数据流对应的当前待发送报文，根据当前待发送报文的大小更新当前令牌数量；并判断待处理数据流是否仍有报文需要发送；在待处理数据流仍有报文需要发送的情况下，返回判断当前令牌数量是否大于或者等于预设门限值的步骤；在待处理数据流没有报文需要发送的情况下，调度待调度流队列中待处理数据流的下一数据流；若当前令牌数量小于预设门限值，则将待处理数据流放入待调度流队列的队尾，等待下一次调度。

[0099] 如上文描述，待处理数据流需要发送的报文可能有N个，N个报文有发送顺序，如果已经发送M个，那么可将N个报文中第M+1个报文作为当前待发送报文。

[0100] 其中，在发送待处理数据流对应的当前待发送报文后，可使用当前令牌数量减去当前待发送报文的大小，从而实现对当前令牌数量的更新。

[0101] 在一些实施例中，在发送待处理数据流对应的当前待发送报文后，除了根据当前待发送报文的大小更新当前令牌数量，还可根据当前待发送报文的大小更新单次突发报文大小，再判断待处理数据流是否仍有报文需要发送，在待处理数据流仍有报文需要发送的情况下，继续判断单次突发报文大小是否达到单次突发的数量上限；在未达到的情况下，返回判断当前令牌数量是否大于或者等于预设门限值的步骤；在达到的情况下，调度待调度流队列中待处理数据流的下一数据流。

[0102] 其中，在发送待处理数据流对应的当前待发送报文后，可使用单次突发报文大小加上当前待发送报文的大小，从而实现对单次突发报文大小的更新。

[0103] 其中，待处理数据流需要发送的报文有N个的情况下，报文处理模块可判断N个报文是否全部发完，在未发完的情况下，确定待处理数据流仍有报文需要发送。

[0104] 示例性的，图3示出了报文处理模块基于令牌数量的工作流程示意图。报文处理模块从待调度活跃流队列中调度活跃流，将该活跃流作为待处理数据流，判断待处理数据流的当前令牌数量token_p是否大于或者等于预设门限token_thresh，在大于或者等于的情况下，发送待处理数据流对应的当前待发送报文，并使用当前令牌数量token_p减去当前待发送报文的大小packet_size，从而实现对当前令牌数量的更新；然后使用单次突发报文大小burst_size加上当前待发送报文的大小packet_size，从而实现对单次突发报文大小的更新。然后判断当前流(待处理数据流)是否仍有报文需要发送，在仍有报文需要发送的情况下，继续判断单次突发报文大小burst_size是否达到单次突发的数量上限max_burst_size，在未达到的情况下，返回判断待处理数据流的当前令牌数量token_p是否大于或者等于预设门限token_thresh的步骤；在达到的情况下，或者当前流没有报文需要发送的情况下，调度下一活跃流。在待处理数据流的当前令牌数量token_p小于预设门限token_thresh的情况下，向算法执行模块发送令牌事件，算法执行模块向报文处理模块反馈目标令牌数量token_c，报文处理模块接收到目标令牌数量token_c后，将

待处理数据流的当前令牌数量token_p加上目标令牌数量token_c，从而实现对当前令牌数量的更新，并判断待处理数据流的当前令牌数量token_p是否大于或者等于预设门限token_thresh，在大于或者等于的情况下，跳转到发送当前待发送报文的步骤，在小于的情况下，挂起当前流，等待下一次调度。

[0105] 上述实施例中，提供了报文处理模块如何基于令牌数量进行报文发送的具体逻辑，该逻辑涉及到的计算包括当前令牌数量的更新、单次突发报文大小的更新，这种更新是简单的加减法运算，现有技术中报文处理模块基于算法执行模块反馈的速率进行报文发送时，需要进行复杂的乘除法运算，相比之下，本申请实施例的方法硬件实现简单，可扩展性强，不易出错，稳定性好。

[0106] 在一些实施例中，算法执行模块可通过如下方式将报文发送速率映射为目标令牌数量：基于令牌事件中携带的令牌事件时间戳、预先设置的令牌基准时间戳以及令牌生成时间段，确定令牌事件对应的令牌请求序号；判断令牌请求序号是否大于令牌发放序号；在大于的情况下，确定还未针对令牌请求序号发放过令牌，则基于报文发送速率和令牌生成时间段，确定目标令牌数量，并基于令牌请求序号对令牌发放序号进行更新；在小于或者等于的情况下，确定已针对令牌请求序号发放过令牌，则将预设值作为目标令牌数量。

[0107] 其中，令牌生成时间段为用户可配置参数。

[0108] 其中，算法执行模块可采用如下公式计算令牌事件对应的令牌请求序号：

token_seq＝(ts_current−ts_token__base)/token__int erval

[0109] 其中，token_seq表示令牌事件对应的令牌请求序号，ts_current表示令牌事件时间戳，ts_token_base表示令牌基准时间戳，token_int erval表示令牌生成时间段。需要说明的是：在上述公式中除法运算的结果为小数时，取小数的整数部分作为令牌事件对应的令牌请求序号。

[0110] 可选的，算法执行模块可采用如下公式确定目标令牌数量：

token c＝rate*token_int erval

[0111] 其中，token_c表示目标令牌数量，rate表示报文发送速率，token_int erval表示令牌生成时间段。

[0112] 可选的，算法执行模块还可采用如下公式确定目标令牌数量：

token_c＝rate*(ts_current−ts_last)

[0113] 其中，token_c表示目标令牌数量，rate表示报文发送速率，ts_current表示令牌事件时间戳，ts_last表示上次接收到的待处理数据流的令牌事件中携带的令牌事件时间戳。

[0114] 其中，预设值可以根据实际情况灵活设定，示例性的，预设值可以为0。

[0115] 示例性的，图4示出了算法执行模块的令牌发放机制的流程示意图。算法执行模块计算令牌事件对应的令牌请求序号token_seq，然后判断该令牌请求序号token_seq是否大于令牌发放序号assign_seq，在小于或者等于的情况下，通过上述公式计算目标令牌数量token_c，并将令牌请求序号token_seq作为新的令牌发放序号assign_seq，从而实现令牌发放序号assign_seq的更新，进一步将目标令牌数量token_c反馈给报文处理模块；在大于的情况下，确定目标令牌数量token_c为0，并将目标令牌数量token_c反馈给报文处理模块。

[0116] 示例性的，假设算法执行模块前一次接收到待处理数据流的令牌事件中，令牌事件时间戳为1s，令牌生成时间段为10s，通过上面的公式可以计算得到该次令牌事件对应的令牌请求序号token_seq为0，对令牌发放序号assign_seq更新后，令牌发放序号assign_seq为0。假设算法执行模块本次接收到待处理数据流的令牌事件中，令牌事件时间戳为5s，通过上面的公式可以计算得到本次令牌事件对应的令牌请求序号token_seq还是0，由于与令牌发放序号assign_seq相等，确定目标令牌数量token_c为0，并将目标令牌数量token_c反馈给报文处理模块，即不针对本次令牌事件发放令牌。

[0117] 上述实施例中，算法执行模块接收到令牌事件后，计算令牌事件对应的令牌请求序号，并判断是否已针对该令牌请求序号发放过令牌，在未发放过令牌的情况下，一次性发放所有令牌，即，将报文发送速率和令牌生成时间段的乘积作为目标令牌数量；在已发放过令牌的情况下，则不针对本次令牌事件发放令牌。这种对短时间内令牌发放数量进行控制的方式，可防止报文处理模块短时间内获得过多令牌数量来进行报文发送引起的网络拥塞。

[0118] 在一些实施例中，将报文发送速率映射为目标令牌数量之前还包括：基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是

否满足基于时间周期的升速调节机制的触发条件，在满足的情况下，采用基于时间周期的升速调节机制对报文发送速率进行升速调节；和/或；基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于已发报文大小的升速调节机制的触发条件，在满足的情况下，采用基于已发报文大小的升速调节机制对报文发送速率进行升速调节。

[0119] 其中，算法执行模块可将令牌事件作为隐式升速信号，在接收到令牌事件后，一方面，基于待处理数据流的流标识，查询待处理数据流的报文发送速率；另一方面，基于待处理数据流的流标识以及令牌事件时间戳，判断是否需要对该报文发送速率进行升速调节。

[0120] 可选的，算法执行模块可以执行的升速调节包括两种机制，分别为基于时间周期的升速调节机制，以及基于已发报文大小的升速调节机制。

[0121] 可选的，算法执行模块接收到令牌事件后，一方面可基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于时间周期的升速调节机制的触发条件，在满足的情况下，采用基于时间周期的升速调节机制对报文发送速率进行升速调节；另一方面可基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于已发报文大小的升速调节机制的触发条件，在满足的情况下，采用基于已发报文大小的升速调节机制对报文发送速率进行升速调节。或者执行上述两方面逻辑其中之一，本申请实施例对此不做限定。

[0122] 上述实施例中，算法执行模块接收到令牌事件后，将令牌事件作为隐式升速信号，进一步基于待处理数据流的流标识以及令牌事件时间戳，判断是否真的需要对该报文发送速率进行升速调节，在确定需要进行升速调节的情况下，对待处理数据流的报文发送速率进行升速调节。和现有技术中算法执行模块将“一段时间内没有收到CNP事件”作为自触发事件，事件触发后运行DCQCN算法的升速处理逻辑相比，本申请中算法执行模块依赖报文处理模块的事件上报触发算法计算逻辑，接口简单，实现便捷。不需要算法执行模块的算法核具备时间戳计时器和复杂任务调度能力，实现成本更低，扩展性更好。

[0123] 基于时间周期的升速调节机制的触发条件可以为：

$ts_current - ts_last_t_increase \geq T$

[0124] 其中，T为预设的升速时间周期，ts_current为令牌事件时间戳，ts_last_t_increase为上一次针对待处理数据流，采用基于时间周期的升速调节机制进行升速调节时的时间戳。

[0125] 一种可能的情形是，令牌事件中携带的令牌事件时间戳，和上一次针对待处理数据流采用基于时间周期的升速调节机制进行升速调节时的时间戳之间，可能相隔多个预设的升速时间周期，因此，基于时间周期的升速调节机制的触发条件可以为：

$M \times T \leq t = ts_current - ts_last_t_increase \leq (M+1) \times T$

[0126] 其中，M为正整数。上述情形下，由于前后两个时间戳之间相隔的时长大于等于M个预设的升速时间周期T，算法执行模块可采用基于时间周期的升速调节机制，对报文发送速率连续进行M次升速调节，即连续升速M次。

[0127] 基于已发报文大小的升速调节机制的触发条件可以为：

$b \geq B$

[0128] 其中，B为预设的升速报文大小，b为令牌事件时间戳与上一次针对待处理数据流，采用基于已发报文大小的升速调节机制进行升速调节时的时间戳之间，已发送的待处理数据流的报文大小。

[0129] 一种可能的情形是，令牌事件中携带的令牌事件时间戳，和上一次针对待处理数据流采用基于已发报文大小的升速调节机制，进行升速调节时的时间戳之间，已发送的待处理数据流的报文大小可能为预设的升速报文大小的若干倍，因此，基于已发报文大小的升速调节机制对应的调节触发条件可以为：

$N \times B \leq b \leq (N+1) \times B$

[0130] 其中，N为正整数。上述情形下，由于前后两个时间戳之间已发送的待处理数据流的报文大小大于等于N个预设的升速报文大小，算法执行模块可采用基于已发报文大小的升速调节机制，对报文发送速率连续进行N次升速调节，即连续升速N次。

[0131] 上述实施例中，提供了基于时间周期的升速调节机制的触发条件，以及基于已发报文大小的升速调节机制的触发条件，和现有技术中的自触发条件“一段时

间内没有收到CNP事件”相比，算法执行模块不需要算法执行模块的算法核具备时间戳计时器和复杂任务调度能力，实现成本更低，扩展性更好。

[0132] 报文处理模块发送给算法执行模块的令牌事件中，不仅携带待处理数据流的流标识和令牌事件时间戳，还携带有待处理数据流的报文队列类型，待处理数据流的报文队列类型可能为SQ，也可能为RQ，这两种类型分别调度。SQ与RQ可能会因为硬件的特殊调度模式而产生饥饿现象，比如每次SQ都取走全部令牌，而RQ请求不到令牌，导致报文队列类型为RQ的流始终无法发送报文。为解决该饥饿现象，本申请实施例提出，可以采用以下逻辑来将报文发送速率映射为目标令牌数量。

[0133] 在一些实施例中，先判断待处理数据流是否被标记为饥饿模式，若待处理数据流被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，从全额令牌数量中确定发放令牌数量，将发放令牌数量作为目标令牌数量；若是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量，并取消待处理数据流的饥饿模式标记；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，将预设值作为目标令牌数量；若是饥饿类型，取消待处理数据流的饥饿模式标记，并基于全额令牌数量和发放令牌数量确定保留令牌数量，将保留令牌数量作为目标令牌数量。

[0134] 其中，算法执行模块可采用上述实施例提供的公式，计算令牌事件对应的令牌请求序号，在得到令牌请求序号后，判断令牌请求序号是否大于令牌发放序号，在大于的情况下，确定还未针对令牌请求序号发放过令牌，在小于或者等于的情况下，确定已针对令牌请求序号发放过令牌。

[0135] 其中，饥饿类型随着不断有令牌事件到来会发生相应的变更，在确定还未针对令牌请求序号发放过令牌的情况下，可判断令牌事件中携带的报文队列类型和

当前的饥饿类型是否一致，若不一致，说明令牌事件中携带的报文队列类型不是饥饿类型，若一致，说明令牌事件中携带的报文队列类型是饥饿类型。

[0136] 其中，在确定还未针对令牌请求序号发放过令牌，且令牌事件中携带的报文队列类型不是饥饿类型的情况下，算法执行模块可采用如下公式确定目标令牌数量：

token_c＝rate*token_int erval/2

[0137] 其中，token_c表示目标令牌数量，rate表示报文发送速率，token_int erval表示令牌生成时间段，rate*token_int erval表示全额令牌数量，也就是说，从全额令牌数量中取一半作为目标令牌数量，剩下的数量作为保留令牌数量。

[0138] 其中，在确定还未针对令牌请求序号发放过令牌，且令牌事件中携带的报文队列类型是饥饿类型的情况下，算法执行模块可采用如下公式确定目标令牌数量：

token_c＝rate*token_int erval

[0139] 其中，token_c表示目标令牌数量，rate表示报文发送速率，token_int erval表示令牌生成时间段，也就是说，将全额令牌数量作为目标令牌数量。

[0140] 其中，在确定已针对令牌请求序号发放过令牌的情况下，仍要判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，可确定目标令牌数量为0，即不发放令牌；若是饥饿类型，则取消待处理数据流的饥饿模式标记，并将保留令牌数量作为目标令牌数量，结合上述实施例的描述，保留令牌数量是用全额令牌数量减去发放令牌数量得到的。

[0141] 在一些实施例中，若待处理数据流未被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为已获得令牌的队列类型，若是，将预设值作为目标令牌数量；若否，将待处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并将预设值作为目标令牌数量。

[0142] 同样的，在待处理数据流未被标记为饥饿模式的情况下，算法执行模块可采用上述实施例提供的公式，计算令牌事件对应的令牌请求序号，在得到令牌请求序号后，判断令牌请求序号是否大于令牌发放序号，在大于的情况下，确定还未针对令牌请求序号发放过令牌，在小于或者等于的情况下，确定已针对令牌请求序号发放过令牌。

[0143] 其中，在确定还未针对令牌请求序号发放过令牌的情况下，算法执行模块可采用如下公式确定目标令牌数量：

token_c＝rate*token_int erval

[0144] 其中，token_c表示目标令牌数量，rate表示报文发送速率，token_int erval表示令牌生成时间段，也就是说，将全额令牌数量作为目标令牌数量。

[0145] 其中，在确定已针对令牌请求序号发放过令牌的情况下，继续判断已发放令牌所对应的令牌事件中报文队列类型，和本次令牌事件中的报文事件类型是否一致，若一致，说明令牌事件中携带的报文队列类型为已获得令牌的队列类型，确定目标令牌数量为0；若不一致，说明令牌事件中携带的报文队列类型不是已获得令牌的队列类型，将待处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并确定目标令牌数量为0。

[0146] 示例性的，图5示出了算法执行模块的防饥饿处理机制的流程示意图。算法执行模块接收到令牌事件后，对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；基于上述实施例的逻辑判断是否需要对该报文发送速率进行升速调节，在需要的情况下，对该报文发送速率进行升速调节，详细过程参见前述实施例的描述，在此不再赘述，升速调节后再执行图5所示防饥饿处理机制；在不需要的情况下，直接执行图5所示防饥饿处理机制。参见图5所示，首先判断待处理数据流当前是否被标记为饥饿模式；若被标记为饥饿模式，判断是否已针对令牌事件对应的令牌请求序号发放过令牌；若未发放，判断事件请求queue是否为饥饿queue，即，判断令牌事件中携带的报文队列类型是否为饥饿类型；若不是饥饿类型，则发放一半可用令牌，保留另一半，即，将token_c＝rate*token_int erval/2作为目标令牌数量，将token_r作为保留令牌数量；若是饥饿类型，则一次性发放所有令牌，即，将toke

n_c＝rate*token_int erval作为目标令牌数量，并取消待处理数据流的饥饿模式标记；若已发放，仍判断判断事件请求queue是否为饥饿queue；若是饥饿类型，本次不发放令牌，即，确定目标令牌数量token_c为0；若不是饥饿类型，发放保留令牌数量，即，将token_r作为目标令牌数量；以上为待处理数据流被标记为饥饿模式情况下的执行逻辑。若待处理数据流未被标记为饥饿模式，仍判断是否已针对令牌事件对应的令牌请求序号发放过令牌；若未发放，则一次性发放所有令牌，即，将token_c＝rate*token_int erval作为目标令牌数量；若已发放，判断事件请求queue是否为已获取令牌queue，即，判断令牌事件中携带的报文队列类型是否为已获得令牌的队列类型；若是，本次不发放令牌，即，确定目标令牌数量token_c为0；若否，标记饥饿模式与饥饿queue，即，将待处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并确定目标令牌数量token_c为0。

[0147] 上述实施例中，提供了算法执行模块的防饥饿处理机制，基于该机制进行令牌发放，可以避免SQ和RQ中任何一种类型一直处于饥饿状态因而得不到令牌无法发送报文的问题，提升了报文发送及时性。

[0148] 应该理解的是，虽然如上的各实施例所涉及的流程图中的各个步骤按照箭头的指示依次显示，但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明，这些步骤的执行并没有严格的顺序限制，这些步骤可以以其它的顺序执行。而且，如上的各实施例所涉及的流程图中的至少一部分步骤可以包括多个步骤或者多个阶段，这些步骤或者阶段并不必然是在同一时刻执行完成，而是可以在不同的时刻执行，这些步骤或者阶段的执行顺序也不必然是依次进行，而是可以与其它步骤或者其它步骤中的步骤或者阶段的至少一部分轮流或者交替地执行。

[0149] 基于同样的发明构思，本申请实施例还提供了一种用于实现上述所涉及的报文发送方法的数据处理器。该数据处理器所提供的解决问题的实现方案与上述方法中所记载的实现方案相似，故下面所提供的一个或多个数据处理器实施例中的具体限定可以参见上文中对于报文发送方法的限定，在此不再赘述。

[0150] 在一个示例性的实施例中，如图6所示，提供了一种数据处理器，包括：

[0151] 报文处理模块601，用于在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件；

[0152] 算法执行模块602，用于对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块；

[0153] 报文处理模块601，还用于基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。

[0154] 在一些实施例中，报文处理模块601，具体用于判断当前令牌数量是否大于或者等于预设门限值；若当前令牌数量大于或者等于预设门限值，发送待处理数据流对应的当前待发送报文，根据当前待发送报文的大小更新当前令牌数量；并判断待处理数据流是否仍有报文需要发送；在待处理数据流仍有报文需要发送的情况下，返回判断当前令牌数量是否大于或者等于预设门限值的步骤；在待处理数据流没有报文需要发送的情况下，调度待调度流队列中待处理数据流的下一数据流；若当前令牌数量小于预设门限值，则将待处理数据流放入待调度流队列的队尾，等待下一次调度。

[0155] 在一些实施例中，报文处理模块601，还用于在发送待处理数据流对应的当前待发送报文后，根据当前待发送报文的大小更新单次突发报文大小；在确定待处理数据流仍有报文需要发送的情况下，继续判断单次突发报文大小是否达到单次突发的数量上限；在未达到的情况下，返回判断当前令牌数量是否大于或者等于预设门限值的步骤；在达到的情况下，调度待调度流队列中待处理数据流的下一数据流。

[0156] 在一些实施例中，算法执行模块602，具体用于基于令牌事件中携带的令牌事件时间戳、预先设置的令牌基准时间戳以及令牌生成时间段，确定令牌事件对应的令牌请求序号；判断令牌请求序号是否大于令牌发放序号；在大于的情况下，确定还未针对令牌请求序号发放过令牌，则基于报文发送速率和令牌生成时间段，确定目标令牌数量，并基于令牌请求序号对令牌发放序号进行更新；在小于或者等于的情况下，确定已针对令牌请求序号发放过令牌，则将预设值作为目标令牌数量。

[0157] 在一些实施例中，算法执行模块602，具体用于基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于时间周期的升速调节机制的触发条件，在满足的情况下，采用基于时间周期的升速调节机制对报文发送速率进行升速调节；和/或；基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于已发报文大小的升速调节机制的触发条件，在满足的情况下，采用基于已发报文大小的升速调节机制对报文发送速率进行升速调节。

[0158] 在一些实施例中，基于时间周期的升速调节机制的触发条件为：

$M \times T \leqslant t = ts_current - ts_last_t_increase \leqslant (M+1) \times T$

[0159] 其中，T为预设的升速时间周期，ts_current为令牌事件时间戳，ts_last_t_increase为上一次针对待处理数据流采用第一升速调节机制进行升速调节时的时间戳，M为正整数；

[0160] 算法执行模块602，具体用于采用基于时间周期的升速调节机制对报文发送速率连续升速M次。在一些实施例中，基于已发报文大小的升速调节机制的触发条件为：

$N \times B \leqslant b \leqslant (N+1) \times B$

[0161] 其中，B为预设的升速报文大小，b为令牌事件时间戳与上一次针对待处理数据流采用第二升速调节机制进行升速调节时的时间戳之间，已发送的待处理数据流的报文大小，N为正整数；

[0162] 算法执行模块602，具体用于采用基于已发报文大小的升速调节机制对报文发送速率连续升速N次。

[0163] 在一些实施例中，算法执行模块602，具体用于若待处理数据流被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，从全额令牌数量中确定发放令牌数量，将发放令牌数量作为目标令牌数量；若是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量，并取消待处理数

据流的饥饿模式标记；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，将预设值作为目标令牌数量；若是饥饿类型，取消待处理数据流的饥饿模式标记，并基于全额令牌数量和发放令牌数量确定保留令牌数量，将保留令牌数量作为目标令牌数量。

[0164] 在一些实施例中，算法执行模块602，具体用于若待处理数据流未被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为已获得令牌的队列类型，若是，将预设值作为目标令牌数量；若否，将待处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并将预设值作为目标令牌数量。

[0165] 上述数据处理器中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中，也可以以软件形式存储于计算机设备中的存储器中，以便于处理器调用执行以上各个模块对应的操作。

[0166] 在一个示例性的实施例中，还提供一种计算机设备，包括上述数据处理器。

[0167] 在一个示例性的实施例中，提供了一种计算机设备，该计算机设备可以是服务器，其内部结构图可以如图7所示。该计算机设备包括处理器、存储器、输入/输出接口(Input/Output，简称I/O)和通信接口。其中，处理器、存储器和输入/输出接口通过系统总线连接，通信接口通过输入/输出接口连接到系统总线。其中，该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质和内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库用于存储预设门限值、报文发送速率等数据。该计算机设备的输入/输出接口用于处理器与外部设备之间交换信息。该计算机设备的通信接口用于与外部的终端通过网络连接通信。该计算机

设备的通信接口可以使用本申请实施例提供的数据处理器。该计算机程序被处理器执行时以实现一种报文发送方法。

[0168] 本领域技术人员可以理解，图7中示出的结构，仅仅是与本申请方案相关的部分结构的框图，并不构成对本申请方案所应用于其上的计算机设备的限定，具体的计算机设备可以包括比图中所示更多或更少的部件，或者组合某些部件，或者具有不同的部件布置。

[0169] 在一个实施例中，提供了一种计算机可读存储介质，其上存储有计算机程序，计算机程序被处理器执行时实现以下步骤：

[0170] 报文处理模块在待处理数据流的当前令牌数量小于预设门限值的情况下，向算法执行模块发送令牌事件；

[0171] 算法执行模块对令牌事件进行解析，得到待处理数据流的流标识，基于流标识，查询待处理数据流的报文发送速率；将报文发送速率映射为目标令牌数量；将目标令牌数量返回给报文处理模块；

[0172] 报文处理模块基于目标令牌数量对当前令牌数量进行更新，并基于当前令牌数量发送待处理数据流对应的报文。

[0173] 在一个实施例中，计算机程序被处理器执行时还实现以下步骤：判断当前令牌数量是否大于或者等于预设门限值；若当前令牌数量大于或者等于预设门限值，发送待处理数据流对应的当前待发送报文，根据当前待发送报文的大小更新当前令牌数量；并判断待处理数据流是否仍有报文需要发送；在待处理数据流仍有报文需要发送的情况下，返回判断当前令牌数量是否大于或者等于预设门限值的步骤；在待处理数据流没有报文需要发送的情况下，调度待调度流队列中待处理数据流的下一数据流；若当前令牌数量小于预设门限值，则将待处理数据流放入待调度流队列的队尾，等待下一次调度。

[0174] 在一个实施例中，计算机程序被处理器执行时还实现以下步骤：在发送待处理数据流对应的当前待发送报文后，根据当前待发送报文的大小更新单次突发报文大小；在确定待处理数据流仍有报文需要发送的情况下，继续判断单次突发报文大小是否达到单次突发的数量上限；在未达到的情况下，返回判断当前令

牌数量是否大于或者等于预设门限值的步骤；在达到的情况下，调度待调度流队列中待处理数据流的下一数据流。

[0175] 在一个实施例中，计算机程序被处理器执行时还实现以下步骤：

[0176] 基于令牌事件中携带的令牌事件时间戳、预先设置的令牌基准时间戳以及令牌生成时间段，确定令牌事件对应的令牌请求序号；判断令牌请求序号是否大于令牌发放序号；在大于的情况下，确定还未针对令牌请求序号发放过令牌，则基于报文发送速率和令牌生成时间段，确定目标令牌数量，并基于令牌请求序号对令牌发放序号进行更新；在小于或者等于的情况下，确定已针对令牌请求序号发放过令牌，则将预设值作为目标令牌数量。

[0177] 在一个实施例中，计算机程序被处理器执行时还实现以下步骤：基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于时间周期的升速调节机制的触发条件，在满足的情况下，采用基于时间周期的升速调节机制对报文发送速率进行升速调节；和/或；基于令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于已发报文大小的升速调节机制的触发条件，在满足的情况下，采用基于已发报文大小的升速调节机制对报文发送速率进行升速调节。

[0178] 在一个实施例中，基于时间周期的升速调节机制的触发条件为：

$M \times T \leqslant t = ts_current - ts_last_t_increase \leqslant (M+1) \times T$

[0179] 其中，T为预设的升速时间周期，ts_current为令牌事件时间戳，ts_last_t_increase为上一次针对待处理数据流采用第一升速调节机制进行升速调节时的时间戳，M为正整数；

[0180] 计算机程序被处理器执行时还实现以下步骤：

[0181] 采用基于时间周期的升速调节机制对报文发送速率连续升速M次。

[0182] 在一个实施例中，基于已发报文大小的升速调节机制的触发条件为：

$N \times B \leqslant b \leqslant (N+1) \times B$

[0183] 其中，B为预设的升速报文大小，b为令牌事件时间戳与上一次针对待处理数据流采用第二升速调节机制进行升速调节时的时间戳之间，已发送的待处理数据流的报文大小，N为正整数；

[0184] 计算机程序被处理器执行时还实现以下步骤：

[0185] 采用基于已发报文大小的升速调节机制对报文发送速率连续升速N次。

[0186] 在一个实施例中，计算机程序被处理器执行时还实现以下步骤：

[0187] 若待处理数据流被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，从全额令牌数量中确定发放令牌数量，将发放令牌数量作为目标令牌数量；若是饥饿类型，则基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量，并取消待处理数据流的饥饿模式标记；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，将预设值作为目标令牌数量；若是饥饿类型，取消待处理数据流的饥饿模式标记，并基于全额令牌数量和发放令牌数量确定保留令牌数量，将保留令牌数量作为目标令牌数量。

[0188] 在一个实施例中，计算机程序被处理器执行时还实现以下步骤：

[0189] 若待处理数据流未被标记为饥饿模式，确定令牌事件对应的令牌请求序号，判断是否已针对令牌请求序号发放过令牌；若还未针对令牌请求序号发放过令牌，基于报文发送速率和令牌生成时间段，确定全额令牌数量，将全额令牌数量作为目标令牌数量；若已针对令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为已获得令牌的队列类型，若是，将预设值作为目标令牌数量；若否，将待处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并将预设值作为目标令牌数量。

[0190] 在一个实施例中，提供了一种计算机程序产品，包括计算机程序，该计算机程序被处理器执行时实现上述实施例提供的报文发送方法。

[0191] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机程序来指令相关的硬件来完成，所述的计算机程序可存储于一非易失性计算机可读取存储介质中，该计算机程序在执行时，可包括如上述各方法的实施例的流程。其中，本申请所提供的各实施例中所使用的对存储器、

数据库或其它介质的任何引用，均可包括非易失性和易失性存储器中的至少一种。非易失性存储器可包括只读存储器(Read-Only Memory，ROM)、磁带、软盘、闪存、光存储器、高密度嵌入式非易失性存储器、阻变存储器(ReRAM)、磁变存储器(Magnetoresistive Random Access Memory，MRAM)、铁电存储器(Ferroelectric Random Access Memory，FRAM)、相变存储器(Phase Change Memory，PCM)、石墨烯存储器等。易失性存储器可包括随机存取存储器(Random Access Memory，RAM)或外部高速缓冲存储器等。作为说明而非局限，RAM可以是多种形式，比如静态随机存取存储器(Static Random Access Memory，SRAM)或动态随机存取存储器(Dynamic Random Access Memory，DRAM)等。本申请所提供的各实施例中所涉及的数据库可包括关系型数据库和非关系型数据库中至少一种。非关系型数据库可包括基于区块链的分布式数据库等，不限于此。本申请所提供的各实施例中所涉及的处理器可为通用处理器、中央处理器、图形处理器、数字信号处理器、可编程逻辑器、基于量子计算的数据处理逻辑器等，不限于此。

[0192] 以上所述实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

[0193] 以上所述实施例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对申请专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请专利的保护范围应以所附权利要求为准。

权利要求书

[权利要求 1] 一种报文发送方法，其特征在于，应用于数据处理器，所述数据处理器包括报文处理模块和算法执行模块，所述方法包括：

所述报文处理模块在待处理数据流的当前令牌数量小于预设门限值的情况下，向所述算法执行模块发送令牌事件；

所述算法执行模块对所述令牌事件进行解析，得到待处理数据流的流标识，基于所述流标识，查询待处理数据流的报文发送速率；将所述报文发送速率映射为目标令牌数量；将所述目标令牌数量返回给所述报文处理模块；及

所述报文处理模块基于所述目标令牌数量对所述当前令牌数量进行更新，并基于当前令牌数量发送所述待处理数据流对应的报文。

[权利要求 2] 根据权利要求1所述的方法，其特征在于，所述基于当前令牌数量发送所述待处理数据流对应的报文，包括：

判断当前令牌数量是否大于或者等于所述预设门限值；

若当前令牌数量大于或者等于所述预设门限值，发送待处理数据流对应的当前待发送报文，根据所述当前待发送报文的大小更新当前令牌数量；并判断待处理数据流是否仍有报文需要发送；在待处理数据流仍有报文需要发送的情况下，返回判断当前令牌数量是否大于或者等于所述预设门限值的步骤；在待处理数据流没有报文需要发送的情况下，调度待调度流队列中待处理数据流的下一数据流；及

若当前令牌数量小于所述预设门限值，则将所述待处理数据流放入待调度流队列的队尾，等待下一次调度。

[权利要求 3] 根据权利要求2所述的方法，其特征在于，所述方法还包括：

根据所述当前待发送报文的大小更新单次突发报文大小；及

在确定待处理数据流仍有报文需要发送的情况下，继续判断单次突发报文大小是否达到单次突发的数量上限；在未达到的情况下，返

回判断当前令牌数量是否大于或者等于所述预设门限值的步骤；在达到的情况下，调度待调度流队列中待处理数据流的下一数据流。

[权利要求 4] 根据权利要求1所述的方法，其特征在于，所述将所述报文发送速率映射为目标令牌数量，包括：

基于所述令牌事件中携带的令牌事件时间戳、预先设置的令牌基准时间戳以及令牌生成时间段，确定所述令牌事件对应的令牌请求序号；

判断所述令牌请求序号是否大于令牌发放序号；

在大于的情况下，确定还未针对所述令牌请求序号发放过令牌，则基于所述报文发送速率和所述令牌生成时间段，确定目标令牌数量，并基于所述令牌请求序号对所述令牌发放序号进行更新；及

在小于或者等于的情况下，确定已针对所述令牌请求序号发放过令牌，则将预设值作为目标令牌数量。

[权利要求 5] 根据权利要求1所述的方法，其特征在于，所述将所述报文发送速率映射为目标令牌数量之前还包括：

基于所述令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于时间周期的升速调节机制的触发条件，在满足的情况下，采用基于时间周期的升速调节机制对所述报文发送速率进行升速调节。

[权利要求 6] 根据权利要求1所述的方法，其特征在于，所述将所述报文发送速率映射为目标令牌数量之前还包括：

基于所述令牌事件中携带的令牌事件时间戳和待处理数据流的流标识，判断待处理数据流是否满足基于已发报文大小的升速调节机制的触发条件，在满足的情况下，采用基于已发报文大小的升速调节机制对所述报文发送速率进行升速调节。

[权利要求 7] 根据权利要求5所述的方法，其特征在于，基于时间周期的升速调节机制的触发条件为：

$M \times T \leqslant t = ts_current - ts_last_t_increase \leqslant (M+1) \times T$

其中，T为预设的升速时间周期，ts_current为令牌事件时间戳，ts_last_t_increase为上一次针对待处理数据流采用第一升速调节机制进行升速调节时的时间戳，M为正整数。

[权利要求 8] 根据权利要求7所述的方法，其特征在于，所述采用基于时间周期的升速调节机制对所述报文发送速率进行升速调节，包括：

采用基于时间周期的升速调节机制对所述报文发送速率连续升速M次。

[权利要求 9] 根据权利要求6所述的方法，其特征在于，基于已发报文大小的升速调节机制的触发条件为：

$N \times B \leqslant b \leqslant (N+1) \times B$

其中，B为预设的升速报文大小，b为令牌事件时间戳与上一次针对待处理数据流采用第二升速调节机制进行升速调节时的时间戳之间，已发送的待处理数据流的报文大小，N为正整数。

[权利要求 10] 根据权利要求9所述的方法，其特征在于，所述采用基于已发报文大小的升速调节机制对所述报文发送速率进行升速调节，包括：

采用基于已发报文大小的升速调节机制对所述报文发送速率连续升速N次。

[权利要求 11] 根据权利要求1所述的方法，其特征在于，所述将所述报文发送速率映射为目标令牌数量，包括：

若待处理数据流被标记为饥饿模式，确定所述令牌事件对应的令牌请求序号，判断是否已针对所述令牌请求序号发放过令牌；

若还未针对所述令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，则基于所述报文发送速率和所述令牌生成时间段，确定全额令牌数量，从所述全额令牌数量中确定发放令牌数量，将所述发放令牌数量作为目标令牌数量；若是饥饿类型，则基于所述报文发送速率和所述令牌生成时间段，确定全额令牌数量，将所述全额令牌数量作为目标令牌数量，并取消待处理数据流的饥饿模式标记；及

若已针对所述令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为饥饿类型，若不是饥饿类型，将预设值作为目标令牌数量；若是饥饿类型，取消待处理数据流的饥饿模式标记，并基于所述全额令牌数量和所述发放令牌数量确定保留令牌数量，将所述保留令牌数量作为目标令牌数量。

[权利要求 12] 根据权利要求11所述的方法，其特征在于，所述方法还包括：

若待处理数据流未被标记为饥饿模式，确定所述令牌事件对应的令牌请求序号，判断是否已针对所述令牌请求序号发放过令牌；

若还未针对所述令牌请求序号发放过令牌，基于所述报文发送速率和所述令牌生成时间段，确定全额令牌数量，将所述全额令牌数量作为目标令牌数量；及

若已针对所述令牌请求序号发放过令牌，判断令牌事件中携带的报文队列类型是否为已获得令牌的队列类型，若是，将预设值作为目标令牌数量；若否，将待处理数据流标记为饥饿模式，将令牌事件中携带的报文队列类型标记为饥饿类型，并将预设值作为目标令牌数量。

[权利要求 13] 一种数据处理器，其特征在于，包括：

报文处理模块，用于在待处理数据流的当前令牌数量小于预设门限值的情况下，向所述算法执行模块发送令牌事件；及

算法执行模块，用于对所述令牌事件进行解析，得到待处理数据流的流标识，基于所述流标识，查询待处理数据流的报文发送速率；将所述报文发送速率映射为目标令牌数量；将所述目标令牌数量返回给所述报文处理模块；

所述报文处理模块，还用于基于所述目标令牌数量对所述当前令牌数量进行更新，并基于当前令牌数量发送所述待处理数据流对应的报文。

[权利要求 14] 一种计算机设备，其特征在于，包括权利要求13所述数据处理器。

[权利要求 15] 一种计算机可读存储介质，其上存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现权利要求1至12中任一项所述的方法的步骤。

[权利要求 16] 一种计算机程序产品，包括计算机程序，其特征在于，该计算机程序被处理器执行时实现权利要求1至12中任一项所述的方法的步骤。