拟申请类型（发明/实用新型）：发明

申请人名称：中南民族大学

申请人的社会信用代码：12100000420000496K

申请人地址：湖北省武汉市洪山区民族大道中南民族大学计算机科学学院

申请人邮政编码：430074

发明人姓名：侯睿，胡青，邢光林

第一发明人身份证号：530103197707030650

联系人电话：15071162550

联系人邮箱：121784399@qq.com

是否提前公开：提前公开

本申请在提交前其技术方案是否还未以任何形式公开（包括网络公开，会议论文，摘要，文章或毕业论文等）：未以任何形式公开。

**权利要求书**

1、一种命名数据网络中基于速率的多路径感知拥塞控制方法，其特征在于，包括以下步骤：

步骤1、路由节点根据命名数据网络名字前缀的不同将兴趣包分为不同的数据流， 并根据公式计算更新Interest流和Data流的平均发送速率；

步骤2、将Data流的发送速率反馈至消费者；

步骤3、路由节点根据FIB表和各个接口的拥塞程度，将流量分流到不同的接口上；

步骤4、消费者更新Interest包的发送速率。

2、根据权利要求1所述的一种命名数据网络中基于速率的多路径感知拥塞控制方法，其特征在于，所述的步骤1包括以下步骤：

步骤1.1、根据命名数据网络名字前缀的不同将兴趣包分为不同的数据流；

步骤1.2、计算Interest包的发送速率：



公式中，I(t)为t时间内每个数据流的Interest包的平均发送速率，η为目标链路利用率，α和β为控制稳定性和性能的参数。Ni(t)表示t时刻通过接口的Interest流数量，Nd(t)为t时刻通过接口的Data流数量。其中C为链路可用带宽，y(t)为接口在时间段(t−d, t)已使用的带宽。q(t)表示时间t时缓冲队列的长度，d是消费者发送Interest包到匹配Data包到达路由节点之间的平均响应时间。平均响应时间采用指数加权移动平均（Exponential Weighted Moving Average,EWMA）计算:d=θ×dsample+(1−θ)×d，其中θ为加权因子。s为Data包与Interest包的平均大小比。

利用Interest流的平均发送速率I(t)，计算出Data流的平均发送速率:



步骤1.3、如果接口内置队列，并且占用率超过阈值π，本方法将立即更新平均发送速率。由于步骤1.2已将链路带宽分配给所有通过的数据流，因此没有可用带宽供新的数据流使用。当接口接收到新的数据流时，这可能会显著超出范围并导致丢包。为了避免这个问题，需要降低其他流的平均发送速率。当新的Interest流到达时，本方法通过以下公式更新Interest包平均发送速率:



当新的Data流到达时，使用以下公式更新Data包的平均发送速率:



3、根据权利要求2所述的一种命名数据网络中基于速率的多路径感知拥塞控制方法，其特征在于，所述的步骤2包括以下步骤：

步骤2.1、当接口接收到新的Interest包时，计算新Interest流的平均发送速率。在路由节点中，每个接口记录Interest流和Data流的平均发送速率I(t)和 D(t)，同时记录每个Interest流允许的速率A(t)。当接口从上游路由节点接收到上游路由节点允许的最小发送速率B(t)的Data包时，接口将A(t)更新为I(t)×s和B(t)之间的较小值；

步骤2.2、当路由节点转发Data包时，路由节点需要计算并告知下游路由节点Data流f的允许发送速率Rf (t)：



其中L为Interest流f在路由节点FIB表项中的转发接口可用列表，用于将Data流f拉回。在转发Data包之前，路由节点将Data包中的反馈值B(t)替换为Rf (t)，具有反馈值的Data包将根据PIT的记录转发到相应的接口。同时，由于每个接口都有自己允许的发送速度D(t)，为了保证Data包的接收速率不大于D(t)，接口会检查并将Data包的反馈值更新为B(t)与D(t)之间的较小值；

步骤2.3、Interest包携带一个反馈值F(t)。当路由节点将一个Interest包转发到一个接口时，该接口将Interest包的反馈值更新为F(t)和I(t)×s之间的较小值；

步骤2.4、当接口接收到下游路由节点的Interest包时，接口也需要更新Interest包的反馈值，从而保证接收到的Data包的速率不超过接口发送Data流的平均速率：

F(t)=min{F(t)，D(t)}。

步骤2.5、路由节点会对每个Interest流维护一个输入接口列表，记录所有接收到属于该Interest流的Interest包的接口，并保留每个接口的权值W，通过EWMA计算:W=δ×W +(1−δ) ×F(t)。路由节点在接收到Data流f的Data包时，根据Data包数据内容名称，会得到相应的Interest流的输入接口列表L+。如果L+中有N个接口，多路径反馈机制采用最大最小公平分配原则，根据每个接口的权值W，将反馈速率Rf (t)分配给每个接口，具体如下所示：





当Data包转发到接口h时，将Data包的反馈值B(t)设为eh。

4、根据权利要求3所述的一种命名数据网络中基于速率的多路径感知拥塞控制方法，其特征在于，所述的步骤3包括以下步骤：

步骤3.1、本方法的多路径转发策略将A(t)的值设为接口的转发权值，根据轮盘选择算法选择可用的转发接口。当路由节点接收到Interest流f的请求时，如果FIB表项有N个可用转发接口，则选择可用转发接口j的概率可通过如下公式计算得出：



步骤3.2、接口允许的转发速率越大，被选中的概率越大。在得到每个接口的选中概率后，可以进一步计算每个接口的累积概率：



根据累积概率，可以确定最终的转发接口。可以使用随机函数生成0到1之间的随机值r，如果r≤q1，则选择第一个接口。否则，当qk−1<r≤qk(2≤k≤N)时，选择第一个k接口转发Interest包。

5、根据权利要求3所述的一种命名数据网络中基于速率的多路径感知拥塞控制方法，其特征在于，所述的步骤4包括以下步骤：

步骤4.1、当消费者收到带有新反馈速率的Data包时，会更新Interest包的发送速率。

步骤4.2、当路由节点因为缓冲区溢出而丢包时，它会发送一个NACK包给消费者。当接收到NACK时，消费者需要重新发送丢失的Interest包，并根据NACK包中的反馈值更新Interest包发送速率。

步骤4.3、消费者还将为每个Interest包维护一个RTO。由于多源、多路径和路由节点缓存，NDN中RTT的测量并不准确。在本方法中，为每个Interest包设置一个相对较大的RTO值。

**说明书**

**一种命名数据网络中基于速率的多路径感知拥塞控制方法**

**技术领域**

本发明涉及计算机网络数据拥塞控制领域，具体是一种命名数据网络NDN（Named Data Networking）中基于速率的多路径感知拥塞控制方法。

**背景技术**

NDN是一种非常有前途的未来网络体系结构，它与TCP/IP完全不同。在NDN中，没有IP地址，所有数据内容都用唯一的数据内容名称命名。通信由消费者发起，消费者通过发送带有唯一所需数据内容名称的Interest包向网络请求所需数据内容。路由节点根据数据内容名称将Interest包转发给相应的节点，这个节点可以是拥有数据内容的生产者，也可以是一个缓存数据内容的路由节点。在TCP/IP中，有两种实现拥塞控制的方案:分别是基于滑动窗口的拥塞控制机制和基于速率的拥塞控制机制。基于滑动窗口的拥塞控制机制已经得到了广泛的研究和部署，它利用一个滑动窗口来限制数据包的发送，并根据拥塞指示(如接收正常ACK报文、数据包超时和乱序消息)来调整窗口大小。基于速率的拥塞控制是TCP/IP拥塞控制的后来者，它计算路由节点中每个数据流的发送速率，并将其反馈给端点。但是基于速率的协议对于TCP/IP并不友好，并且与基于滑动窗口的机制不兼容，因此在实际应用中并不广泛。

幸运的是，NDN是一种新的网络架构。到目前为止，在NDN中还没有标准的拥塞控制机制，也没有像TCP/IP中基于滑动窗口的拥塞控制机制那样强大的拥塞控制基础，这给基于速率的拥塞控制机制带来了新的机遇。传统的基于滑动窗口的拥塞控制机制在接收到正常的ACK报文时增大拥塞窗口，并在检测到拥塞时通过测量RTT、超时和无序数据交付来缩小拥塞窗口。但在NDN中，Data包可以从多个源、在网络缓存中通过多条路径进行检索，从而使RTT发生变化，使得RTO的值难以计算，并且来自不同缓存的Data包可能会无序到达。这使得基于RTT的超时和无序的数据交付成为不可靠的拥塞指标。然而，基于速率的拥塞控制机制并不依赖于这些指标。

本方法采用基于速率的方案，探索了RCP和NDN的多路径转发特性的结合，设计、实现和评估了一种基于速率的多路径感知拥塞控制机制，它允许路由节点公平地反馈速率信息并避免拥塞，同时提出一种多路径转发策略，通过充分利用所有可用路径上的网络资源来实现更高的总吞吐量。

**发明内容**

本发明的目的在于针对现有技术中存在的上述问题，提出一种基于速率的多路径感知拥塞控制机制，更快地检测和响应拥塞状态，实现高链路利用率下的低丢包率和快速数据传输。

本发明的上述目的通过以下的技术方案实现：

一种基于速率的多路径感知拥塞控制机制，包括以下步骤：

步骤1、路由节点根据命名数据网络名字前缀的不同将兴趣包分为不同的数据流， 并根据公式计算更新Interest流和Data流的平均发送速率；

步骤2、将Data流的发送速率反馈至消费者；

步骤3、路由节点根据FIB表和各个接口的拥塞程度，将流量分流到不同的接口上；

步骤4、消费者更新Interest包的发送速率。

如上所述的步骤1包括以下步骤：

步骤1.1、根据命名数据网络名字前缀的不同将兴趣包分为不同的数据流；

步骤1.2、计算Interest包的发送速率：



公式中，I(t)为t时间内每个数据流的Interest包的平均发送速率，η为目标链路利用率，α和β为控制稳定性和性能的参数。Ni(t)表示t时刻通过接口的Interest流数量，Nd(t)为t时刻通过接口的Data流数量。其中C为链路可用带宽，y(t)为接口在时间段(t−d, t)已使用的带宽。q(t)表示时间t时缓冲队列的长度，d是消费者发送Interest包到匹配Data包到达路由节点之间的平均响应时间。平均响应时间采用指数加权移动平均（Exponential Weighted Moving Average,EWMA）计算:d=θ×dsample+(1−θ)×d，其中θ为加权因子。s为Data包与Interest包的平均大小比。

利用Interest流的平均发送速率I(t)，计算出Data流的平均发送速率:



步骤1.3、如果接口内置队列，并且占用率超过阈值π，本方法将立即更新平均发送速率。由于步骤1.2已将链路带宽分配给所有通过的数据流，因此没有可用带宽供新的数据流使用。当接口接收到新的数据流时，这可能会显著超出范围并导致丢包。为了避免这个问题，需要降低其他流的平均发送速率。当新的Interest流到达时，本方法通过以下公式更新Interest包平均发送速率:



当新的Data流到达时，使用以下公式更新Data包的平均发送速率:



如上所述的步骤2包括以下步骤：

步骤2.1、当接口接收到新的Interest包时，计算新Interest流的平均发送速率。在路由节点中，每个接口记录Interest流和Data流的平均发送速率I(t)和 D(t)，同时记录每个Interest流允许的速率A(t)。当接口从上游路由节点接收到上游路由节点允许的最小发送速率B(t)的Data包时，接口将A(t)更新为I(t)×s和B(t)之间的较小值；

步骤2.2、当路由节点转发Data包时，路由节点需要计算并告知下游路由节点Data流f的允许发送速率Rf (t)：



其中L为Interest流f在路由节点FIB表项中的转发接口可用列表，用于将Data流f拉回。在转发Data包之前，路由节点将Data包中的反馈值B(t)替换为Rf (t)，具有反馈值的Data包将根据PIT的记录转发到相应的接口。同时，由于每个接口都有自己允许的发送速度D(t)，为了保证Data包的接收速率不大于D(t)，接口会检查并将Data包的反馈值更新为B(t)与D(t)之间的较小值；

步骤2.3、Interest包携带一个反馈值F(t)。当路由节点将一个Interest包转发到一个接口时，该接口将Interest包的反馈值更新为F(t)和I(t)×s之间的较小值；

步骤2.4、当接口接收到下游路由节点的Interest包时，接口也需要更新Interest包的反馈值，从而保证接收到的Data包的速率不超过接口发送Data流的平均速率：

F(t)=min{F(t)，D(t)}。

步骤2.5、路由节点会对每个Interest流维护一个输入接口列表，记录所有接收到属于该Interest流的Interest包的接口，并保留每个接口的权值W，通过EWMA计算:W=δ×W +(1−δ) ×F(t)。路由节点在接收到Data流f的Data包时，根据Data包数据内容名称，会得到相应的Interest流的输入接口列表L+。如果L+中有N个接口，多路径反馈机制采用最大最小公平分配原则，根据每个接口的权值W，将反馈速率Rf (t)分配给每个接口，具体如下所示：





当Data包转发到接口h时，将Data包的反馈值B(t)设为eh。

如上所述的步骤3包括以下步骤：

步骤3.1、本方法的多路径转发策略将A(t)的值设为接口的转发权值，根据轮盘选择算法选择可用的转发接口。当路由节点接收到Interest流f的请求时，如果FIB表项有N个可用转发接口，则选择可用转发接口j的概率可通过如下公式计算得出：



步骤3.2、接口允许的转发速率越大，被选中的概率越大。在得到每个接口的选中概率后，可以进一步计算每个接口的累积概率：



根据累积概率，可以确定最终的转发接口。可以使用随机函数生成0到1之间的随机值r，如果r≤q1，则选择第一个接口。否则，当qk−1<r≤qk(2≤k≤N)时，选择第一个k接口转发Interest包。

如上所述的步骤4包括以下步骤：

步骤4.1、当消费者收到带有新反馈速率的Data包时，会更新Interest包的发送速率。

步骤4.2、当路由节点因为缓冲区溢出而丢包时，它会发送一个NACK包给消费者。当接收到NACK时，消费者需要重新发送丢失的Interest包，并根据NACK包中的反馈值更新Interest包发送速率。

步骤4.3、消费者还将为每个Interest包维护一个RTO。由于多源、多路径和路由节点缓存，NDN中RTT的测量并不准确。在本方法中，为每个Interest包设置一个相对较大的RTO值。

本发明相对于现有技术具有以下有益效果：

由于NDN中存在多源、多路径、PIT聚合等问题，使得Interest包的RTT值难以精确计算，导致传统的端到端拥塞控制算法不适用于NDN。本发明是基于速率的拥塞控制机制更新路由节点的发送速率，从而避免了RTT难以估计和无序数据包到达不能作为拥塞信号的问题，通过监控接口队列长度和流量数量来实现拥塞控制，在ndnSIM2.0中的实验结果表明，本发明在NDN网络中有效且公平，所提出的多路径转发策略能够充分利用可用链路带宽，实现流量负载均衡。

**附图说明**

图1为哑铃拓扑。

图2为Interest流的影响。

图3为多路径拓扑。

图4为链路利用率。

图5为丢包情况。

图6为公平场景拓扑。

图7为缓存和多路径反馈的影响。

图8为RMCM、IRCF和IRPI在不同场景下的公平性。

图9为多路径转发: RMCM、CF和PI。

**具体实施方式**

为了便于本领域普通技术人员理解和实施本发明，下面结合实施例对本发明作进一步的详细描述，应当理解，此处所描述的实施示例仅用于说明和解释本发明，并不用于限定本发明。

一种命名数据网络中基于速率的多路径感知拥塞控制方法，包括以下步骤：

步骤1、 路由节点根据命名数据网络名字前缀的不同将兴趣包分为不同的数据流， 并根据公式计算更新Interest流和Data流的平均发送速率；

步骤1具体包括以下步骤：

步骤1.1、根据命名数据网络名字前缀的不同将兴趣包分为不同的数据流；

步骤1.2、在NDN中，路由节点将Interest包和Data包转发到其接口，导致Interest流和Data流同时流出接口，这促使本方法在为数据流分配带宽时考虑Interest流的影响。如果接口不能及时转发，Data包将会在路由节点排队，为了清空队列【什么队列，队列与Interest流、Data流、Interest包和Data包的关系？】并将可用链路带宽分配给每个接口的数据流，使用如下公式来计算Interest包的发送速率：【针对每个接口？】

【对于同一个对象，全文引用同一个名称定义，不要用不同的角度的定义对同一个对象进行说明，不清楚上述记载的“Interest包”、“Interest流”、“数据流”之间的关系？不清楚上述记载的“Data包”、“Data流”、“数据流”之间的关系？是否可以直接以“Interest包”和“Data包”这两个概念进行说明？

步骤1.1中的将兴趣包分为不同的数据流与步骤1.2中提到的Data流是同一个对象吗？

尽量不要用“数据流”这种定义方式，因为不清楚是兴趣包的还是数据包的】



公式中，I(t)为t时间内每个数据流的Interest包的平均发送速率【的含义？】，η为目标链路利用率，α和β为控制稳定性和性能的参数。Ni(t)表示t时刻通过接口的Interest流数量，Nd(t)为t时刻通过接口的Data流数量。其中C为链路可用带宽，y(t)为接口在时间段(t−d, t)已使用的带宽。q(t)表示时间t时缓冲队列的长度，d是消费者发送Interest包到匹配Data包到达路由节点之间的平均响应时间。平均响应时间采用指数加权移动平均（Exponential Weighted Moving Average,EWMA）计算:d=θ×dsample+(1−θ)×d，其中θ为加权因子。s为Data包与Interest包的平均大小比。

利用Interest流的平均发送速率I(t)，计算出Data流的平均发送速率:



步骤1.3、在每个周期d中，每个接口通过步骤1.2来更新Interest流和Data流的平均发送速率。但是由于缓存的多样性，如果队列已经建立，那么在下一个周期之前没有必要更新这些速率。因此，如果接口内置队列【“队列已经建立”和“接口内置队列”的区别？】，并且占用率超过阈值π，本方法将立即更新平均发送速率。由于步骤1.2已将链路带宽分配给所有通过的数据流，因此没有可用带宽供新的数据流使用。当接口接收到新的数据流时，这可能会显著超出范围并导致丢包。为了避免这个问题，需要降低其他流的平均发送速率。当新的Interest流到达时，本方法通过以下公式更新Interest包平均发送速率:【前面调整“流”的发送速率，此处公式又是“包”的速率】



当新的Data流到达时，使用以下公式更新Data包的平均发送速率:



步骤2、将Data流的发送速率反馈至消费者；

步骤2具体包括以下步骤：

步骤2.1、在NDN中，不同的Data流可能有不同的Data包大小。反馈Interest流发送速率会导致相同发送速率的Interest流会拉回不同速率的Data流。因此，本方法反馈Data流发送速率，以保持不同Data包大小的消费者之间的公平性。

当接口接收到新的Interest包时，计算新Interest流的平均发送速率。在路由节点中，每个接口记录Interest流和Data流的平均发送速率I(t)和 D(t)，同时记录每个Interest流允许的速率A(t)。当接口从上游路由节点接收到上游路由节点允许的最小发送速率B(t)的Data包时，接口将A(t)更新为I(t)×s和B(t)之间的较小值【较大值？】；

步骤2.2、当路由节点转发Data包时，路由节点需要计算并告知下游路由节点Data流f的允许发送速率Rf (t)【最大允许，最小允许？】：



其中，【的含义？】L为Interest流f在路由节点FIB表项中的转发接口可用列表，用于将Data流f拉回。在转发Data包之前，路由节点将Data包中的反馈值B(t)替换为Rf (t)，具有反馈值的Data包将根据PIT的记录转发到相应的接口。同时，由于每个接口都有自己允许的发送速度D(t)【符号表达应该与接口关联】，为了保证Data包的接收速率不大于D(t)，接口会检查并将Data包的反馈值更新为B(t)与D(t)之间的较小值；

步骤2.3、在NDN中，请求相同数据内容的Interest包可能会到达不同的接口，导致Data流的发送速率必须反馈给多个接口。为了避免这个问题,本方法提出了一个多路径反馈机制, 在保证数据传输的公平性的前提下，向多个接口分配Data流的发送速率。Interest包也将携带一个反馈值F(t)。当路由节点将一个Interest包转发到一个接口时，该接口将Interest包的反馈值更新为F(t)和I(t)×s【含义？】之间的较小值；

步骤2.4、当接口接收到下游路由节点的Interest包时，接口也需要更新Interest包的反馈值，从而保证接收到的Data包的速率不超过接口发送Data流的平均速率：

F(t)=min{F(t)，D(t)}。

步骤2.5、路由节点会对每个Interest流维护一个输入接口列表，记录所有接收到属于该Interest流的Interest包的接口，并保留每个接口的权值W，通过EWMA计算:W=δ×W +(1−δ) ×F(t)【各个公式含义】。路由节点在接收到Data流f的Data包时，根据Data包数据内容名称，会得到相应的Interest流的输入接口列表L+。如果L+中有N个接口，多路径反馈机制采用最大最小公平分配原则，根据每个接口的权值W，将反馈速率Rf (t)分配给每个接口，具体如下所示：





当Data包转发到接口h时，将Data包的反馈值B(t)设为eh。

当具有相同数据内容名称的Interest包聚合到一个PIT条目时，该机制不能立即充分利用链路带宽，但由于步骤1是周期性执行的，所以可以在几个间隔d后利用可用带宽。在没有PIT聚合的情况下，该机制能很好地防止上游链路拥塞。

步骤3、路由节点根据FIB表和各个接口的拥塞程度，将流量分流到不同的接口上；

步骤3具体包括以下步骤：

步骤3.1、本方法的多路径转发策略将A(t)的值设为接口的转发权值，根据轮盘选择算法选择可用的转发接口。当路由节点接收到Interest流f的请求时，如果FIB表项有N个可用转发接口，则选择可用转发接口j的概率可通过如下公式计算得出：



步骤3.2、接口允许的转发速率越大，被选中的概率越大。在得到每个接口的选中概率后，可以进一步计算每个接口的累积概率：



根据累积概率，可以确定最终的转发接口。可以使用随机函数生成0到1之间的随机值r，如果r≤q1，则选择第一个接口。否则，当qk−1<r≤qk(2≤k≤N)时，选择第一个k接口转发Interest包。

步骤4、消费者更新Interest包的发送速率。

步骤4.1、在本方法中，消费者发送Interest包的速率取决于Data包中所携带的反馈信息。在初始阶段，消费者使用慢启动发送Interest包。一旦有Data包到达，消费者从Data包中提取反馈速率，并根据自己的Data包大小转换成Interest包的发送速率。当消费者收到带有新反馈速率的Data包时，会更新Interest包的发送速率。

步骤4.2、当路由节点因为缓冲区溢出而丢包时，它会发送一个NACK包给消费者。路由节点也通过NACK包将传输速率反馈给消费者，这与Data包中的反馈模式相同。当接收到NACK时，消费者需要重新发送丢失的Interest包，并根据NACK包中的反馈值更新Interest包发送速率。

步骤4.3、消费者还将为每个Interest包维护一个RTO。由于多源、多路径和路由节点缓存，NDN中RTT的测量并不准确。在本方法中，为每个Interest包设置一个相对较大的RTO值。虽然这样会导致消费者对丢包的响应变慢，但是本方法可以保持队列长度很小甚至为零，从而避免丢包。

最后，将对本发明方法（RMCM）进行性能仿真分析。采用的仿真平台为ndnSIM2.0，ndnSIM2.0是基于ns-3网络仿真软件用C++编写实现CCNx协议的NDN仿真模块，实现了基本的网络协议、路由转发策略以及数据包节点缓存等功能，能够仿真多样化的部署场景，支持大范围的NDN实验； 步骤1中有三个参数，η为目标链路利用率，在所有的实验中其都被设置为0.9。α和β为控制稳定性和性能的参数，通过实验发现，当α值小于0.3，β值在0.5到1.0之间时，RMCM可以很好地工作。因此，在接下来的实验中，将α设置为0.2，β设置为0.9。

1、RMCM的有效性

（1）单路径场景：本发明（RMCM）首先利用图1所示的哑铃拓扑来评估在公式4.1中考虑Interest流的效果。Consumer1从Producer1检索数据内容，Consumer2延迟5s开始，向Producer2检索数据内容。

实验结果如图2所示。在不考虑Interest流的场景里, Router1和Router2中的队列是在Consumer2在5s的时间点开始请求Data包时建立的，并在Consumer2在20s的时间点停止之前在Router2中保持平均100个Data包的长度。这说明了在步骤1中考虑Interest流的影响的必要性。

（2）多路径场景：接下来，实验评估了本发明（RMCM）在多源多路径场景中的有效性。如图3所示，有两个数据源，Producer1和Producer2，存储相同的数据内容。Consumer从这两个数据源中检索数据内容。

图4和图5显示了不同链路的利用率和不同节点中的丢包情况，在三个瓶颈链路中，数据流的平均速率都大于4Mbps，接近最佳值4.5 Mbps（η为0.9）。在实验过程中，没有发生数据包丢失。实验结果表明，RMCM能够有效地支持多径场景。

（3）缓存和多路径反馈场景：在此场景中，评估了缓存的效果和多路径反馈机制的性能，使用C1和C2来检索存储在P1节点中的相同的数据内容，具体如图6所示。C2从8s的时间点开始，在25s的时间点停止。R1的缓存区大小为5000，采取的缓存策略为LRU。每个接口的缓冲区大小为100。分别进行了有多径反馈机制和无多径反馈机制的实验。

实验结果如图7所示。在8s的时间点之前，C1是网络拓扑中唯一的消费者，以接近4.5Mbps的速率检索数据内容。在8s的时间点，C2开始请求数据，请求命中R1处的缓存，导致18Mbps的更高的带宽。大约在9s左右的时间点，C2耗尽R1处的缓存，并加入C1从节点P1处检索数据内容，致使大量的请求进入节点P1，这导致P1中的队列不断增加。

同样的结果也可以在12s到14s的时间点之间发现。在采用多径反馈机制的实验结果中，经过14s的时间点之后，C1和C2以相同的发送速率从P1中检索数据内容，链路R1-P1的发送速率接近于最优值9Mbps。在此期间，节点P1和R1的队列长度为零。在没有多径反馈机制的实验结果中，在13s到18s的时间点之间，C1从R1的缓存中检索数据内容，导致C2以9Mbps的发送速率从P1中检索数据内容。但是在19s之后，C1加入C2从P1检索数据内容。这就是没有多路径反馈机制的问题开始显现的地方:R1向C1和C2反馈9Mbps的发送速率。因为瓶颈链路C1-R1的带宽为5Mbps，C1以4.5Mbps的速率检索数据内容，而C2以9Mbps的速率检索数据内容。但是在R1-P1链路上，带宽只有10Mbps，小于13.5Mbps，从而导致了大量的丢包。基于以上的分析，可以发现本发明（RMCM）能够适应NDN中的网内缓存，多径反馈机制使得本发明（RMCM）能够保持公平性，避免拥塞的发生。

2、RMCM的公平性

实验通过与另外两个拥塞控制算法IRCF和IRPI的比较来评估RMCM的公平性。IRCF是一种多路径拥塞控制算法，结合了远程自适应主动队列管理RAAQM和多路径转发策略CF。IRPI是RAAQM和另一种多路径转发策略PI的结合。本实验在图6所示的拓扑中使用了以下三种场景：

Equal：四个消费者具有相同的RTT和相同大小的Data包。

Diff\_RTT：四个消费者具有不同的RTT和相同大小的Data包。从C1、C2、C3和C4到R1的RTT分别为1ms、10ms、20ms和40ms。

Diff\_DS：四个消费者具有相同的RTT，但Data包大小不同，分别为1KB、2KB、3KB和4KB。

实验结果如图8所示。可以明显观察到，与其他两种算法相比，本发明（RMCM）能够保证消费者在不同场景下的公平性。在相等和不同的RTT场景中，不同消费者的Data包的接收速率几乎相同。在Diff\_DS场景中其差别也不大，Data包大小较大的消费者的速率略大，但与其他两种算法相比，四个消费者之间的速率差异可以忽略不计。此外，还可以发现本发明（RMCM）比其他两种算法具有更好的稳定性。

3、多路径感知转发的性能

以上仿真结果已经证明了本发明（RMCM）的有效性和公平性。接下来将对本发明（RMCM）中多路径转发的性能进行验证，如图6所示，本文选取了R2-P1、R2-P2和R2-P3这三条路径来实现三种不同的场景，在这里，R2充当消费者，从三个生产者那里获取数据内容。三种情况如下：

Equal：R2-P1、R2-P2和R2-P3这三条路径具有相同的带宽，均为10Mbps，RTT也相同，均为10ms。

Diff\_RTT：三条路径具有相同的带宽，均为10Mbps，但RTT不同，分别为10ms（端口257）、50ms（端口258）和100ms（端口259）。

Diff\_BW：三条路径具有相同的RTT，均为10ms，但带宽不同，分别为10 Mbps （端口257）、20 Mbps （端口258）、40 Mbps （端口259）。

本文将本发明（RMCM）同另外两种多路径转发策略CF和PI进行了对比。其中PI 优先选择待定兴趣包最少的接口进行转发，而 CF 在PI 的基础上计算每个接口的权重，并使用加权轮询算法选择转发接口。在每个算法都稳定之后的情况下，再对每个端口的传输流量进行统计。

实验结果如图9所示。在Equal场景下，三种转发策略的效果都很好，即三个端口接收的流量基本相同。在Diff\_RTT场景下，PI和CF都更倾向于RTT比较低的端口。在Diff\_BW场景下，PI和CF不能有效地使用更高带宽的端口。理想的最优比率是14%:28%:57%，但实际只达到了18%:36%:45%，从而降低了总吞吐量。在Diff\_RTT和Diff\_BW两种场景下， CF和PI的转发策略更倾向于RTT更低或带宽更高的路径。由于本发明的多路径转发直接响应可用带宽，所以在任何情况下都可以达到预期的转发分流比，这使得本发明（RMCM）能够快速获得更高的整体吞吐量。

需要指出的是，本发明中所描述的具体实施例仅是对本发明精神作举例说明。本发明所属技术领域的技术人员可以对所描述的具体实施例作各种各样的修改或补充或采用类似的方式替代，但并不会偏离本发明的精神或超越所附权利要求书所定义的范围。

**说明书附图**



图1

(1) (2)



(3) (4)



图2

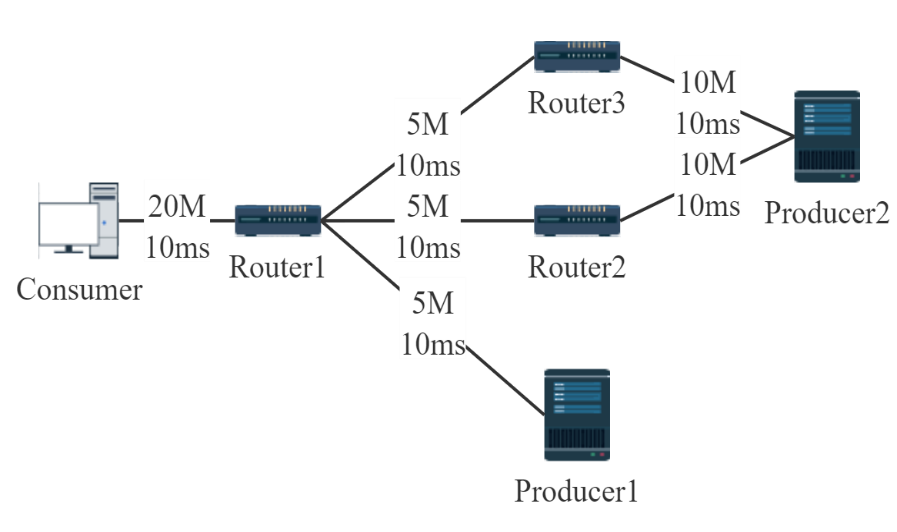


图3



图4



图5

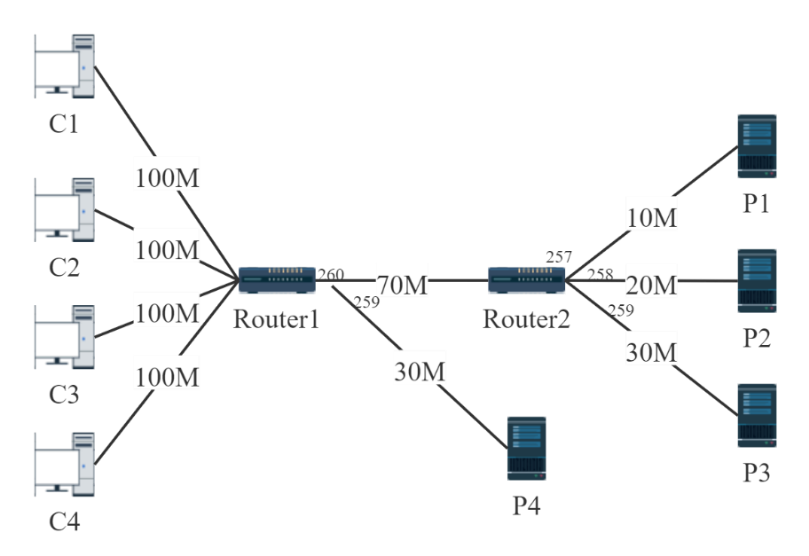


图6

(1) (2)



(3) (4)



(5) (6)



图7

(1) (4) (7)



(2) (5) (8)



(3) (6) (9)



图8

(1) (2) (3)



图9

**说明书摘要**

本发明公开了一种命名数据网络中基于速率的多路径感知拥塞控制方法。将结合速率控制协议（RCP）和NDN的多路径转发特性，设计、实现和评估一种基于速率的多路径拥塞控制方法，该方法计算每个数据流的传输速率，并将其反馈给消费者，消费者从而相应地控制Interest包的发送速率，同时提出一种多路径转发策略，通过充分利用所有可用路径上的网络资源来实现更高的总吞吐量。

**摘要附图**



图1