

一种基于缓存交互的命名数据网络拥塞控制算法^①

石珊珊^{②***} 任勇毛^{*} 李俊^{③*} 李灵玲^{*} 智江^{***}

(^{*} 中国科学院计算机网络信息中心 北京 100190)

(^{**} 中国科学院大学 北京 100049)

摘 要 研究了命名数据网络(NDN)的拥塞控制。为了解决突发流量问题和提高吞吐量及网络资源利用率,考虑了路由器缓冲区大小与拥塞控制机制的相互影响以及 NDN 内部署缓存这一重要特性,提出了一种基于缓存交互的 NDN 拥塞控制算法。该算法通过利用 NDN 中的路由器缓存,在逻辑上动态扩充缓冲区大小并控制 Data 包的发送速率,同时与现有的 NDN 拥塞控制算法相结合,动态调整 Interest 包发送速率阈值,以平滑突发流量,缓解网络拥塞。基于 ndnSIM 的仿真实验结果表明,该算法能有效提高 NDN 的传输效率、吞吐量和网络资源利用率。

关键词 命名数据网络(NDN), 拥塞控制, 缓存, 缓冲队列大小, 动态阈值(DT), 突发流量

0 引 言

命名数据网络(named data networking, NDN)^[1,2]作为一种以信息为中心的网络(information-centric networking, ICN)^[3]体系结构,是未来互联网体系结构研究的一个热点方向。体系结构的革新使 NDN 与传统 TCP/IP 网络相比,表现出新的传输模式和传输特点,因此,NDN 拥塞控制机制是 NDN 体系结构中有待研究的关键技术之一。目前 NDN 拥塞控制机制^[4]主要考虑 Interest 包和 Data 包一对一的传输特点,通过在接收端控制 Interest 的发送速率控制 Data 包的发送速率。当网络发生拥挤时,通过隐式或显式方式通知接收端调整 Interest 包的发送速率。其中,隐式拥塞控制代表算法如 ICP^[5],通过借鉴 TCP 协议的加性增长乘性减少(AIMD)算法,在接收端根据往返时间(RTT)的值调整 Interest 的发送速率,以及根据重传超时时间(RTO)判断拥塞的发

生,但很难估计 RTO 的值^[6]。显式拥塞控制算法通常在中间节点检测网络拥塞状态,并显式反馈给接收端,接收端据此调整 Interest 包发送速率,以控制 Data 包返回速率,主要的算法有 CHoPCoP^[7], ECP^[8]和 ECN-based^[9]等。

虽然命名数据网络(NDN)拥塞控制方面的研究已有了一些成果,但仍不成熟。一方面未充分利用 NDN 中间节点部署缓存这一重要特点。另一方面也未充分考虑网络层面对拥塞的控制以及突发流量的问题,比如网络供给、流量整形等。在数据中心网络中普遍存在 micro-burst^[10]、突发流量^[11]问题,针对此问题通常采用大缓冲区将部分流量缓存下来,暂缓发送,从而减小丢包率,防止数据包重传。同时采用动态阈值(dynamic threshold, DT)算法^[12]或改进的增强动态阈值(enhancing dynamic threshold, EDT)算法^[10]进行缓冲队列管理,从而有效地吸收突发流量。基于以上考虑,本文从 NDN 中间节点部署缓存这一特点出发,针对突发流量问题,提出了

① 973 计划(2012CB315803)和中国科学院计算机网络信息中心“一三五”计划(CNIC _ PY-1401)资助项目。
② 女,1988 年生,硕士生;研究方向:信息中心网络拥塞控制;E-mail: shishanshan@ cstnet. cn
③ 通讯作者, E-mail: lijun@ cnic. cn
(收稿日期:2015-12-29)

一种基于缓存交互的命名数据网络拥塞控制算法。将该算法作为独立的组件,与现有的基于端或基于 hop-by-hop 的流量整形算法相结合,在逻辑上扩充缓冲队列,可以减小丢包,平滑突发流量。实验结果表明,引入基于缓存交互 (content store-based, CS-based) 模块的拥塞控制算法可以更好地吸收突发流量,缓解突发流量带来的不稳定性,具有更高的传输效率、网络吞吐量以及资源利用率。

1 算法设计

1.1 算法设计动机与原理

本节首先论述缓冲区大小与拥塞控制的相互影响,然后再从理论上简单论述使用缓存 (content store, CS) 在逻辑上扩充输出队列的合理性,即利用 NDN 缓存的优势,从逻辑上扩充输出缓冲队列,以平滑短暂的突发性流量,减小丢包率,从而提高吞吐量和资源利用率。

1.1.1 缓冲区大小与拥塞控制的相互影响

网络中路由器都配有缓冲区,当网络发生拥塞或遭遇突发流量时,缓冲区可以缓存网络中的数据包,减小丢包率,防止数据包重传,但缓冲队列过长则会增加排队时延^[13]。因此,路由器缓冲区大小与 TCP 拥塞控制算法的动态性密切相关。同时路由器端口队列的建立依赖于可用缓冲区空间大小和 TCP 流量的突发性^[14],而 TCP 流量大小又依赖于流量经过的链路的拥塞程度。DT 算法^[12]通过管理输出队列的共享缓冲区改善系统损耗以及公平性,提高网络传输性能。队列延迟控制 (controlling queue delay, CoDel)^[15]考虑到目前网络缓冲区的不断增大、延迟敏感应用的广泛发展以及流媒体业务的不断兴起,对主动队列管理机制 (AQM) 进行改进,可适应链路的动态变化以及突发流量的情况。因此,本文在 NDN 中也将拥塞控制机制与缓冲区和缓冲队列管理进行综合考虑,同时利用 NDN 缓存优势,以应对网络突发流量,提高网络性能。

1.1.2 动态扩充缓冲队列的合理性分析

动态阈值 (DT) 算法^[12]认为输出队列的阈值与空闲缓冲区大小成正比,并为所有端口配置相同大

小的队列阈值,即分配相同的缓冲资源。用 $T(t)$ 表示队列控制阈值,其计算公式如下:

$$T(t) = \alpha(B - Q(t)) = \alpha(B - \sum_i Q^i(t)) \quad (1)$$

式中, $Q^i(t)$ 表示端口 i 在 t 时刻的输出队列长度, B 表示系统共享缓冲区大小, α 为控制参数。当队列长度 $Q(t)$ 等于或超过控制阈值 $T(t)$ 时,将不再允许数据包进入队列,即进行丢包。根据上述缓冲资源分配公式,我们结合 NDN 缓存构造如下缓冲资源分配公式:

$$\begin{aligned} T(t) &= \alpha(C + B - \bar{q}(t)) \\ &= \alpha(C + B - \sum_i \bar{q}^i(t)) \end{aligned} \quad (2)$$

式中,我们增加了 C ,表示为扩充输出缓冲队列借用缓存的容量,这里 $C \geq 0$,表示缓存逻辑扩充缓冲队列算法可作为独立模块。当 $C = 0$ 时,表示不使用该模块,当 $C > 0$ 时,表示使用该模块。 $q(t)$ 表示瞬时队列长度平均值, $\bar{q}(t)$ 的计算公式见 1.4 节。由于队列阈值 $T(t)$ 与空闲缓冲区成正比,因此 $C + B$ 扩大了共享缓冲区,则空闲缓冲区增大,队列阈值 $T(t)$ 增大,从而减小了丢包可能性,防止数据包重传。

同时,DT 算法为吸收突发流量,保持系统稳定性,会预留一部分缓冲区,用 Ω 表示,并给出如下公式:

$$Q = S \cdot T + \Omega \quad (3)$$

式中, Q 为稳定状态下系统总的缓冲区占用量, T 为稳定状态下的队列阈值, S 为工作的端口数。增强动态阈值 (EDT) 算法^[10]通过理论和实验论证,由于 DT 算法预留部分缓冲资源,当发生丢包时,系统中仍有部分缓冲区未使用。同时,DT 算法为确保输出端口缓冲区公平分配,即使突发流大小远远小于缓冲区大小时,数据包也会被丢掉,造成了缓冲资源的浪费。因此,我们结合 NDN 缓存给出如下公式:

$$Q = (S \cdot T + \Omega) + \Omega_c \quad (4)$$

式中,我们增加了 Ω_c 表示通过缓存 (CS) 在逻辑上增加的预留缓冲区。同上, $\Omega_c \geq 0$,表示缓存逻辑扩充缓冲队列算法可作为独立模块,当 $\Omega_c = 0$ 时,表示不使用该模块,当 $\Omega_c > 0$ 时,表示使用该模块。可以看出,使用缓存 Ω_c 代替 Ω 作为预留缓冲资源,可以提高原有缓冲区的利用率,防止丢包,更好地吸收突

发流。

需要指出的是,本文仅初步论证了考虑到 NDN 缓存的天然特性,在逻辑上动态扩充缓冲队列的有效性,目的在于利用 NDN 路由器缓存对缓冲区管理和拥塞控制提供一种参考,而对于缓冲区大小初始值设置,以及利用的缓存大小的最优值还未做进一步探讨。

1.2 基于缓存交互的拥塞控制模型

本节给出的 NDN 中基于缓存交互的拥塞控制机制模型如图 1 所示。缓存交互拥塞控制算法运行在 NDN 路由器上。该模型在原有 NDN 路由模型的基础上增加了数据表(data table, DT)和基于缓存的速率调整(content store-based rate adjusting, CRA)两个模块,当遇到突发流量导致缓冲队列平均队列长度瞬间增大时,CRA 和 DT 可控制 Data 包的转发,将 Data 包暂存于 CS 中,并延迟 Data 包的转发,实现在逻辑上扩充输出缓冲队列,从而平滑突发流量,减小丢包数,提高网络吞吐量。同时该算法与已有的基于接收端的拥塞控制算法进行协商,决定是否使用该算法,或使用该算法所需的参数设置,以此控制 Interest 包的发送速率,从而缓解网络拥塞问题。

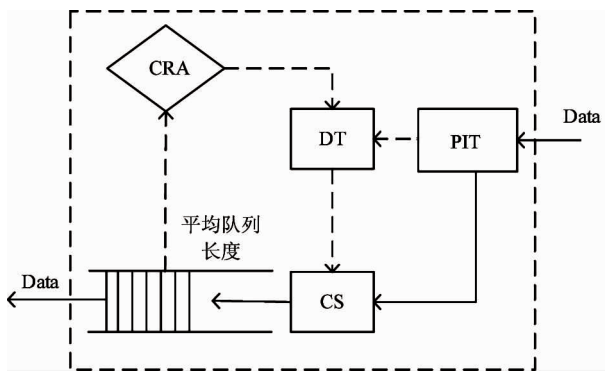


图 1 拥塞控制模型

1.3 基于缓存交互的 Data 包转发控制算法

利用缓存控制 Data 包的转发是基于缓存交互的网络拥塞控制算法的核心部分。本节将着重介绍如何进行实际的缓冲队列扩充,即如何通过缓存交互控制 Data 包的转发。

NDN 中,Data 包远大于 Interest 包大小,因此扩

充缓冲队列实质是扩充 Data 包的输出队列长度。同时 one-Interest-one-Data 的传输特点^[4]使得可以通过调整下游节点 Interest 包的发送速率控制 Data 包的返回速率,因此 Data 队列增大,实际上是 Interest 队列增大。我们将结合图 2 给出更详细的描述。

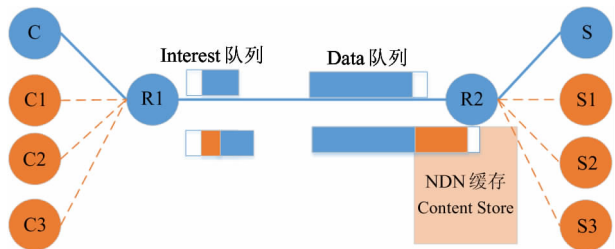


图 2 动态扩充缓冲队列示例

图 2 中,C-R1-R2-S 为工作流量,假设 Interest 和 Data 队列长度已达到饱和且稳定(此时网络不拥塞),在某一时刻由于突发流 C1, C2, … 的加入,使得 R1 的 Interest 队列瞬间增大。如果使用缓存交互模块,则进行如下操作,R1 接受新增流 C1、C2、C_n 发来的 Interest 请求包,这样就达到了放行突发流请求的效果。同时,R2 对 Data 队列检测,如果新增流量持续较长,为防止流量后压,需向下游节点 R1 通过显式或隐式方式反馈拥塞信息使其控制 Interest 包的速率,从而从根本上缓解网络拥塞。

算法中增加 DT 模块,DT 为 FIFO 类型,DT 中一个条目记录未被转发的 Data 的控制信息,用于控制这些 Data 的转发,即控制 Data 包进入输出缓冲队列的速率。当 Data 包被正常转发后,删除 DT 中相应的条目。

Data 从到达离开路由器的过程伪代码:

BEGIN

输入: Data 包

PIT match; // 查询 PIT 表

If (PIT 表中存在匹配条目)

{

CS insert;

If ($\bar{q} \geq q_{\max}$)

{

If ($\bar{q} \geq p_{\max}$)

拥塞通知请求端降低

Interest 发送速率;

```

}
else
{
    在 Data Table 中增加条目,
    表示 Data 包需要调整速率发送;
    CRA( Data );//发送 Data;
}
}
else
{
    将 Data 包原速率转发到对应的端口;
}
else( PIT 表中不存在匹配条目)
{
    丢弃 Data 包;
}
END

```

其中,在 DT 中增加条目是本算法的重点。 $q_{\max}$ 表示不使用缓存交互时输出队列的阈值。当 $\bar{q} \geq q_{\max}$ 时,表示网络可能会出现拥塞,新到达的 Data 经过 PIT 模块之后,并不是直接将其转发到端口输出队列,而是在 DT 中创建一个条目,然后再利用 CRA 算法控制其转发,CRA 可选择原速率发送 Data 包,也可用窗口值调整 Data 发送。 $p_{\max}$ 表示使用缓存交互算法时 DT 的阈值,当 DT 的条目超过一定 $p_{\max}$ 时,需结合 Interest 速率整形算法,显式或隐式通知下游节点减小 Interest 包的发送速率。 $\bar{q}$ 计算公式见 1.4 节。

1.4 Data 缓冲队列检测方法

如图 3 所示,R2 使用缓存扩充 Data 包的输出缓冲队列长度,只限于短暂突发流量的场景,而非永久扩充缓冲队列,如果缓冲队列一直过长,则会增加大量包的排队时延,从而影响网络服务质量。因此,当检测到突发流量时间过长或流量过大时,则需要向下游节点发送拥塞信息通知其减小 Interest 包的发送速率。这里,我们通过检测 R2 的 Data 包的输出队列长度,如果队列长度饱和超过一定时间 Δt (Δt 的值与带宽有关),并且 DT 中仍有未完成发送的 Data 条目,表示突发流量时间过长或流量过大,此时 R2 通知 R1 进行端的 Interest 发送速率控制,R2 检测 Data 包的输出队列长度的方法如下:

路由器的端口 i 每发出一个 Data 就计算一次 Data 的输出队列占用率,记为 $q(t)$,再根据 $q(t)$ 计算移动平均值 $\bar{q} = (1 - \mu)\bar{q} + \mu q(t)$,其中 μ 表示瞬时队列占用率 $q(t)$ 的权重值。 $\bar{q}$ 即为检测到的 R2 的 Data 包的输出队列长度。

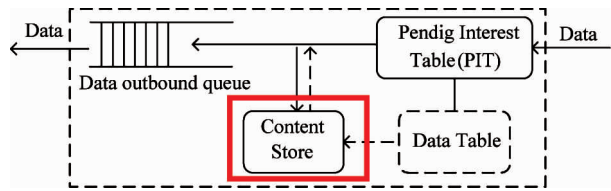


图 3 Data 包转发过程

2 性能评价

2.1 实验环境与设置

本实验在 ndnSIM^[16] 仿真平台上进行。ndnSIM 基于 NS-3^[17],它将 NDN 网络协议模块化,模拟实现了 NDN 通信模型,是目前最为流行的 NDN 开源仿真工具。

实验中在接收端采用 NDN 中一种典型的 Interest 请求速率控制算法——ICP 算法^[5] (也可以使用其他现有的 NDN 拥塞控制算法),在 ICP 基础上,针对突发流量情况比较使用缓存模块和不使用缓存模块的性能,即对 ICP + CS-based 模块与 ICP + non-CS-based 模块的性能进行分析比较。

实验拓扑如图 4、图 5 所示,实验参数设置如表 1 所示,接收端 Client1、Client2 分别向发送端 Server1、Server2 请求不同的数据。Client1 和 Server1 之间的流量为工作流量,记作 flow1。Client2 和 Server2 之间的流量为背景流量,特点是具有突发性,记作 flow2。瓶颈链路带宽为 10Mbps,其他链路带宽为 100Mbps,链路时延为 10ms。图 4 所示为单瓶颈链路,瓶颈链路为 R2-R3,图 5 所示为双瓶颈链路,瓶颈链路为 R1-R2 和 R3-R4。Interest 包和 Data 包大小分别是 40B 和 1040B (后续实验均采用此设置)。随着突发流量 flow2 的变化,比较两种方案的性能。

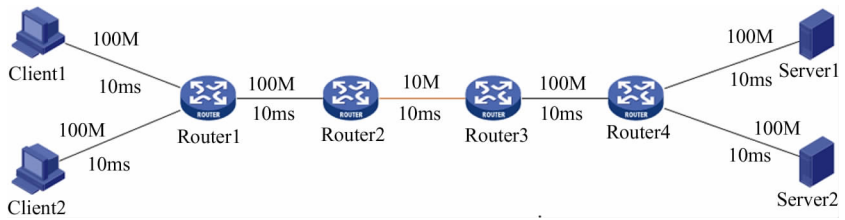


图 4 单瓶颈链路拓扑

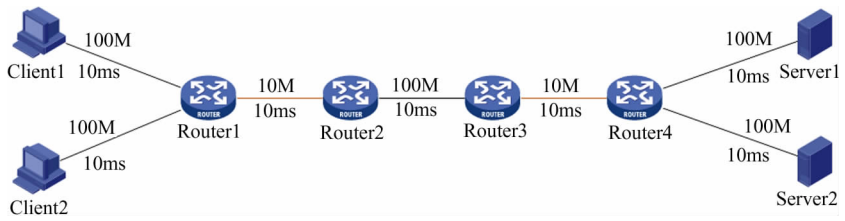


图 5 双瓶颈链路拓扑

表 1 实验参数设置

工作流量	Client1—Server1
突发流量	Client2—Server2
Interest 包大小	40B
Data 包大小	1040B
链路带宽	100Mbps
瓶颈链路带宽	10Mbps
链路时延	10ms

2.2 实验场景与结果分析

2.2.1 单瓶颈-1 次突发 CBR-突发流量持续时间 1s

该场景使用图 4 所示拓扑:当 $t = 0s$ 时,实验开始运行,Client1 向 Server1 请求 80M 的 Data 包。当 $t = 20s$ 时,加入突发流量 flow2,Client2 向 Server2 发送 Interest 请求 Data 包。

flow2 使用恒定比特率 (CBR) 控制请求 Interest 的速率,参数 Frequency 控制突发流大小,参数 Maxseq 控制突发流量持续时间,例如 $Frequency = 400$, $Maxseq = 400$,表示突发流大小为 400Interests/s,持续时间为 1s。该场景改变 flow2 的大小,持续时间为 1s。

图 6 为突发流 flow2 的 Frequency 大小为 600 时 Client1 的吞吐量。实验开始时,只有 flow1,因此 Client1 的吞吐量即收到 Data 包的速率不断增加,在 $t = 16.5s$ 时达到最大,此时瓶颈链路 R2-R3 的容量达到饱和,R2、R3 的 Interest 和 Data 队列达到最大值,在 $t = 23.5s$ 时,Client2 请求的突发流量到来。由

于增加了新的流量,为避免网络发生拥塞即 R2-R3 发生拥塞,CS-based 和 non-CS-based 方案都使 Client1 收到 Data 包的速率降低,但原因不同。在 CS-based 方案中,由于突发流量很短暂,Client1 并不需要减低发送 Interest 的速率,同时放行 Client2 的 Interest 请求,Server1、Server2 收到 Interest 请求后返回 Data,由于利用缓存在逻辑上增加了 R2 和 R3 的输出缓冲队列长度,并且控制 Data 的速率,所以 Client1 收到 Data 的速率降低,同时由于 Client1 不降低 Interest 请求速率,所以 Client2 突发流量消失后,可以很快恢复饱和状态。在 non-CS-based 方案中,由于突发流量的到来,R2、R3 的 Interest/Data 队列满,因此丢包触发 Client1 降低 Interest 发送速率,使

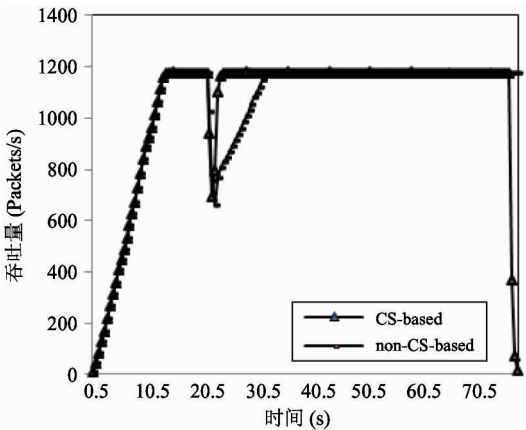


图 6 Client1 吞吐量 600Interest/s

其收到 Data 的速率降低,突发流量消失后,Client1 进入拥塞避免阶段,需要一段时间恢复到饱和状态。因此从实验结果可以看出,CS-based 在 $t = 23.5\text{s}$ 时达到最大速率,在 $t = 76.5\text{s}$ 时传输完成,non-CS-based 在 $t = 39.5\text{s}$ 时达到最大速率,在 $t = 83.5\text{s}$ 时传输完成。与 non-CS-based 相比,CS-based 的收敛性提高了 $(39.5 - 23.5)/23.5 = 68\%$,带宽利用率提高了 $(1179.6 - 864.4)/1179.6 = 26.7\%$,传输时间减少了 $(83.5 - 76.5)/83.5 = 8.38\%$ 。

如图 7 所示,当突发流量持续时间不变时,Client1 的传输完成时间受突发流量大小的影响。其中,突发流量从 400Interests/s 增大到 1000Interests/s 时,使用 CS-based 的传输完成时间递增的非常缓慢且几乎不发生改变,而使用 non-CS-based 的传输完成时间出现了较大幅度递增且波动的现象,并且 CS-based 比 non-CS-based 的完成时间更短。因此 CS-based 能更有效地吸收突发流量,具有更好的传输性能。

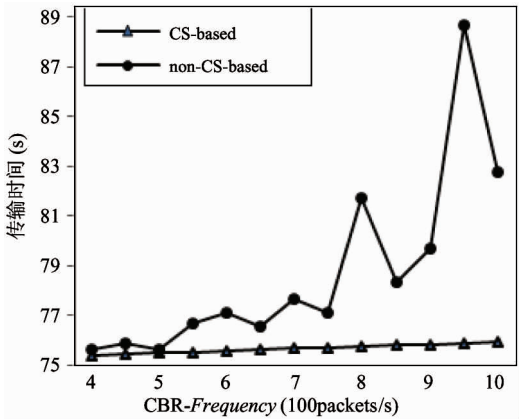


图 7 Client1 完成时间

2.2.2 单瓶颈-多次突发 CBR-突发流量持续时间 1s

该场景使用图 4 所示拓扑,突发流量 flow2 使用 CBR 控制请求 Interest 的速率。改变突发流量 flow2 的大小,比较突发流量次数不同时,两种机制的传输性能。

如图 8 所示,为 Client1 的传输完成所需时间随突发流量大小的改变。CS-based 比 non-CS-based 的完成时间更短。且突发流次数增加对 CS-based 几

乎无影响,而 non-CS-based 完成时间随突发流次数增加而增加。因此 CS-based 能有效地平滑多次突发流量,具有更好的传输性能。

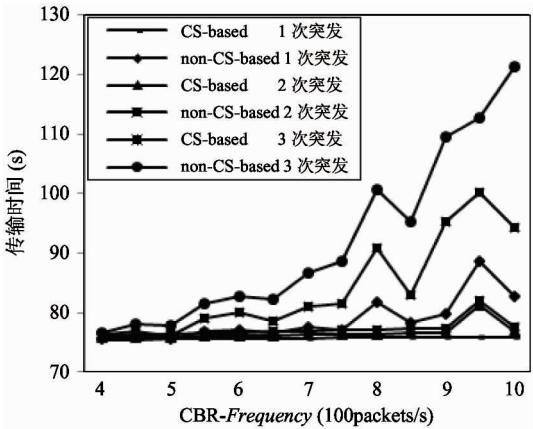


图 8 Client1 完成时间

2.2.3 单瓶颈-1 次突发 CBR-突发流量持续时间 5s

该场景使用图 4 所示拓扑,突发流量 flow2 使用 CBR 控制请求 Interest 的速率。改变突发流量 flow2 的持续时间,比较两种机制的传输性能。

如图 9 所示,随着突发流量持续时间的增大,两种方案下,Client1 完成时间都增大,但 CS-based 比 non-CS-based 的完成时间更短。因此 CS-based 具有更高的传输效率。

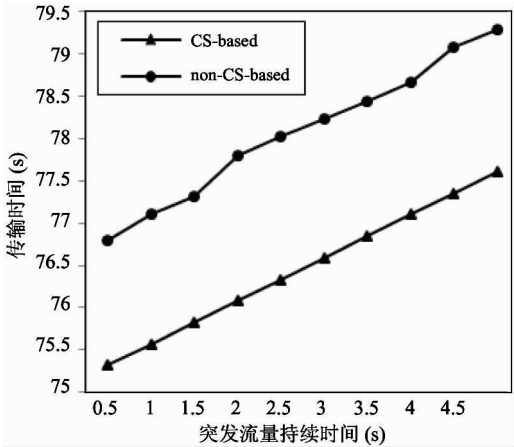


图 9 Client1 完成时间

2.2.4 单瓶颈-1 次突发-Zipf 突发流量持续时间 1s

该场景使用图 4 所示拓扑,突发流量 flow2 使用 Zipf 控制请求 Interest 的速率。突发流量的 flow2 的

大小不变,为 800Interests/s,改变 Zipf 参数 α 。分析两种机制下的传输性能。

如图 10 所示,在 Client1 端,CS-based 比 non-CS-based 的完成时间更短,且分布指数增加对 CS-based 几乎无影响,而使用 non-CS-based 时传输完成时间随 Zipf 参数 α 的增大而增大。因此 CS-based 能有效地平滑突发流量,具有更好的传输性能。

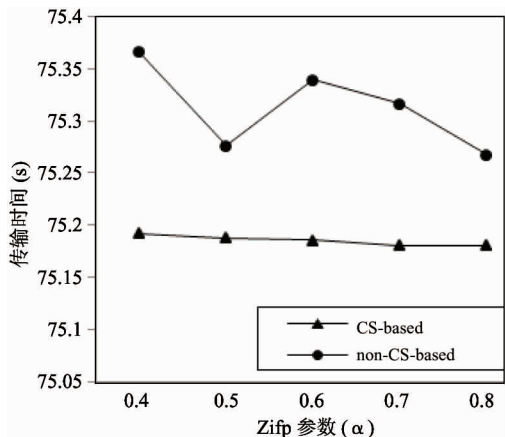


图 10 Client1 完成时间

2.2.5 多瓶颈-突发流量持续时间 1s

该场景使用图 5 所示的双瓶颈链路拓扑,突发流量 flow2 使用 CBR 控制请求 Interest 的速率。改变突发流量 flow2 大小,同时与单瓶颈链路场景进行对比。

如图 11 所示,在 Client1 端,CS-based 比 non-CS-based 的完成时间更短,且瓶颈链路数对 CS-based 无影响,但在 non-CS-based 方案中,双瓶颈比

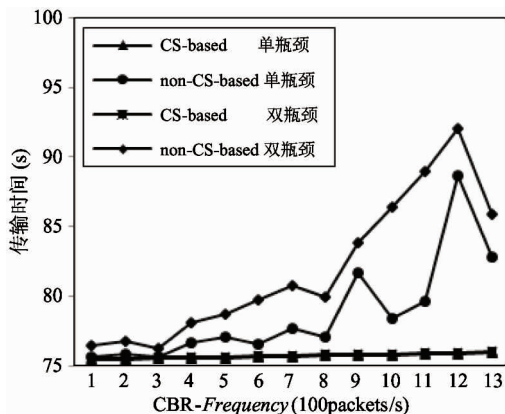


图 11 Client1 完成时间

单瓶颈需要更长的时间。因此 CS-based 能有效地平滑突发流量,且适应多个瓶颈链路的情况,具有更好的传输性能。

3 结论

本文从 NDN 内部署缓存这一特点出发,结合缓冲区大小与拥塞控制的相互影响,简单论述了动态扩充缓冲队列的合理性,提出了一种基于 NDN 缓存交互的拥塞控制算法。该算法使用 NDN 缓存在逻辑上扩充缓冲队列长度,并与已有的 NDN 拥塞控制算法结合,动态调整 Interest 发送速率阈值,目的在于充分利用 NDN 缓存的特点和优势,提高 NDN 资源利用率、网络传输性能,尤其是针对突发流量的情形,减小突发流量给网络带来的不稳定性。该算法可作为独立的组件与已有的 NDN 拥塞控制算法相结合。基于 ndnSIM 仿真实验结果表明,该拥塞控制算法可以更好地平滑突发流量以及减小丢包率,与不使用缓存模块的拥塞控制算法相比具有更高的传输效率、吞吐量以及资源利用率。

在下一步工作中,将进一步通过实验和理论计算,确定 NDN 路由器缓冲队列长度的最优值,以及动态扩充的缓冲队列长度的最优值,从而进一步提高传输效率和资源利用率。同时,需要对算法的公平性进行评估,进而在考虑公平性的基础上优化算法,使本算法在保证高效传输的同时兼顾公平性。

参考文献

- [1] Jacobson V, Smetters D, Thornton J, et al. Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 2009. 1-12
- [2] Zhang L, Estrin D, Burke J, et al. Named Data Networking (NDN) Project: [Technology Report]. Xerox Palo Alto Research Center-PARC, 2010
- [3] Ahlgren B, Dannewitz C, Imbrenda C, et al. A survey of information-centric networking. *IEEE Communications Magazine*, 2012, 50(7): 26-36
- [4] Yi C, Afanasyev A, Wang L, et al. Adaptive forwarding in named data networking. *ACM SIGCOMM Computer Communication Review*, 2012, 42(3): 62-67
- [5] Carofiglio G, Gallo M, Muscariello L. ICP: Design and

- evaluation of an interest control protocol for content-centric networking. In: Proceedings of Computer Communications Workshops (INFOCOM WKSHPS), Orlando, USA, 2012. 304-309
- [6] Zhang L X. Why TCP timers don't work well. *ACM SIGCOMM Computer Communication Review*, 1986, 16(3): 397-405
- [7] Zhang F, Zhang Y, Reznik A, et al. A transport protocol for content-centric networking with explicit congestion control. In: Proceedings of the 23rd International Conference on Computer Communication and Networks, Shanghai, China, 2014. 1-8
- [8] Ren Y M, Li J, Shi S S, et al. An interest control protocol for named data networking based on explicit feedback. In: Proceedings of the 11th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, Oakland, USA, 2015. 199-200
- [9] Zhou J M, Wu Q H, Li Z Y, et al. A proactive transport mechanism with explicit congestion notification for NDN. In: Proceedings of IEEE International Conference on Communications (ICC), London, UK, 2015. 5242-5247
- [10] Shan D F, Jiang W C, Ren F Y. Absorbing micro-burst traffic by enhancing dynamic threshold policy of data center switches. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 2015. 118-126
- [11] Alizadeh M, Greenberg A, Maltz D A, et al. Data center TCP (DCTCP). *ACM SIGCOMM Computer Communication Review*, 2010, 40(4): 63-74
- [12] Choudhury A, Hahne E. Dynamic queue length thresholds for shared-memory packet switches. *IEEE/ACM Transactions on Networking*, 1998, 6(2): 130-140
- [13] Appenzeller G, Keslassy I, McKeown N. Sizing router buffers. *ACM SIGCOMM Computer Communication Review*, 2004, 34(4): 281-292
- [14] Wischik D. Buffer sizing theory for bursty TCP flows. In: Proceedings of the International Zurich Seminar on Communications, Zurich, Switzerland, 2006. 98-101
- [15] Nichols K, Jacobson V. Controlling queue delay. *ACM Communications*, 2012, 55(7): 42-50
- [16] Alexander A, Moiseenko I, Zhang L X. ndnSIM: NDN simulator for NS-3. Named Data Networking (NDN) Project: [Technology Report]. Los Angeles: University of California, 2012
- [17] Henderson T R, Roy S, Floyd S, et al. ns-3 project goals. In: Proceedings of the 2006 Workshop on ns-2: the IP Network Simulator, Pisa, Italy, 2006

A content store-based congestion control algorithm for named data networking

Shi Shanshan^{***}, Ren Yongmao^{*}, Li Jun^{*}, Li Lingling^{*}, Zhi Jiang^{***}

(^{*} Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190)

(^{**} University of Chinese Academy of Sciences, Beijing 100049)

Abstract

The congestion control of named data networking (NDN) was studied. To solve the bursty traffic problem, increase the throughput, and improve the network resource utilization, a content store-based congestion control algorithm for NDN was proposed under the considerations of the mutual influence between the buffer size of the router and the congestion control mechanism, as well as the in-network caching scheme of NDN. This algorithm logically performs the dynamical extension of the buffer size and controls the forwarding rate of Data packets by utilizing the router content store of NDN, and dynamically adjusts the sending rate threshold of Interest packets to smooth the bursty traffic and alleviate the network congestion by the combination with the existing NDN congestion control algorithms. The results of the experiment conducted based on ndnSIM indicate that this algorithm can effectively improve NDN's transmission efficiency, throughput and the network resource utilization.

Key words: named data network (NDN), congestion control, content store, buffer size, dynamic threshold (DT), bursty traffic