

基于订阅推送的 NDN 实时通信机制^①

靳鹏飞^{②***} 李俊^{③*} 吴海博* 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 研究了代表未来互联网体系结构发展方向的命名数据网络(NDN)的实时通信机制。研究发现,NDN 用户通过发送请求包(Interest)获得对应数据(Data),而对于实时通信类应用(如即时聊天、物联网监控等),由于数据源的数据产生存在突发性和随机性,而且存在数据接收时延,传统 NDN 用户驱动的“拉”模式数据获取方式,难以实现用户请求速率与数据产生速率的同步,无法满足实时通信应用需求。从减小数据接收延迟出发,在不改变原 NDN 路由缓存机制的基础上提出了一种基于订阅推送的 NDN 实时通信机制,该机制通过提前在路由器中添加订阅信息,实现实时数据源的即时推送。相关理论分析和仿真实验表明,该机制在实时数据接收延迟上明显优于传统 NDN 方法。

关键词 命名数据网络(NDN), 实时通信, 订阅, 推送

0 引言

随着互联网的发展,现有 TCP/IP 通信模式面临诸多问题^[1],以命名数据网络(named data networking, NDN)^[2]为代表的未来网络体系结构已得到学术界的广泛关注和研究^[3-5]。NDN 以命名数据为中心,用户只需关注信息(内容)本身,而不必关注信息的来源,从而使网络通信模式与应用需求相适应。对于实时通信类应用,由于数据源的数据产生存在突发性和随机性,现有 NDN 方案难以很好地满足实时通信需求。NDN 即时通信研究^[6,7]均讨论了 NDN 下如何实现即时通信,最终 NDN 项目应用团队设计了同步通信组件^[8]。通信多方通过该组件周期性交换本地消息摘要实现最新消息的同步,当发现有消息更新后发送对应的请求包获取相应实时数据。该方式可以保证消息传输的可靠性,但是增加了数据接收延时。对于物联网传感器数据传输、

紧急报警通知等应用,其对网络延迟敏感,基于请求者“拉”模式的通信无法满足应用的需求。Carzaniga 等人强调网络中存在两种应用通信模式^[9],即内容分发类应用和消息通知类应用,并提出了一种结合内容中心网络(content-centric networking, CCN)^[10](content-centric networking,与 NDN 机制相同但所属研究机构不同)和 IP 的解决方案,该方案在一定程度上解决了 CCN 中消息推送的问题,但在一定程度上增加了路由、安全等方面设计和管理的复杂性,此外该方案对于实时通信应用没有很好利用命名数据网络(NDN)设计思想中天然多播和网内缓存的优势。François 等人^[11]提出了一种消息推送机制,它将推送数据放在请求包中,直接查找转发信息库(forwarding information base, FIB)将消息推送出去。这种设计需要对 CCN 转发模块进行修改,对于推送数据量大的消息,内容封装在请求包中将变得不再可取。

本文在原 NDN 框架基础上,提出了一种基于订

① 973 计划(2012CB315803),CNIC 所级专项(CNIC_PY-1401)和中国科学院知识创新工程青年人才领域(CNIC_QN_1508)资助项目。

② 男,1991 年生,硕士生;研究方向:命名数据网络;E-mail: jinpengfei@cstnet.cn

③ 通讯作者,E-mail: lijun@cnic.cn

(收稿日期:2015-12-01)

阅推送的实时通信传输机制,该机制通过扩展 NDN 包格式,利用 Interest 包携带订阅信息,在不影响原有通信模式的前提下,在路由器中添加订阅记录表记录订阅信息,支持实时数据源数据的推送,减小数据接收延迟。本方案具有实现简单,传输高效,易扩展等优点。通过 ndnSIM^[12] 仿真平台验证,该方案对于实时通信应用、数据接收延迟明显优于基于原 NDN 机制的设计。

1 NDN 传输原理简介

NDN 传输的核心是命名数据对象,数据的标识采用特定的名字。NDN 中有两种类型的包:请求 (Interest) 包和数据 (Data) 包。数据请求者通过发送 Interest 包向数据产生 (或拥有) 者请求特定名字的数据。而数据产生 (或拥有) 者回复带有相应名字的数据包。

NDN 路由器为实现包的转发,需要维护 3 张表^[2,13]:转发信息库 (FIB)、待定 Interest 表 (pending-interest table, PIT)、缓存 (content store, CS)。Interest 包的转发需要查找 FIB 表, FIB 条目的维护由 NDN 路由协议^[14,15] 沟通实现。在路由器转发 Interest 包前,需将 Interest 包的名字和入端口号记入 PIT 中,以便路由器收到对应 Data 包时,查找 PIT 并从相应端口转发。在执行完 Data 包转发后,路由器会将该 PIT 表项删除。因此,如果请求者没有发送 Interest 包,数据源推送的数据因在 PIT 中无法找到匹配表项而被丢弃,从而防止恶意数据的推送。当多个用户请求相同的内容时, PIT 会在同一条目下记录多个端口,在 Data 包转发时,以此实现类似多播的效果。CS 用于有策略的缓存路由器收到的 Data 包,当其它用户再次请求该数据时,路由器可以直接对其进行回复,从而减轻内容源以及网络传输的负担。

2 方案设计

对于实时通信而言,数据的传输是由数据源驱动的。当数据源有数据产生时应该主动将数据推送

给对应用户。采用订阅的方式具有两个好处:(1) 可以标识数据源应该将数据推送给哪些用户;(2) 如果用户没有进行订阅,数据源不能主动将消息推送出去,即推送数据的传输与否由用户决定。

该方案在兼容原有通信模式的前提下,通过扩展 NDN 包格式,利用 Interest 包携带订阅信息,在路由器中添加订阅记录表记录用户订阅信息,当实时数据源有新消息产生时,查找订阅记录表,将消息推送给订阅用户,减小实时应用数据接收延迟。

2.1 包格式扩展

如 Carzaniga 等人^[9] 所述,网络通信中既有内容分发为主的应用,又有实时通信类应用的存在,为支持这两种通信模式的并存,方案设计通过扩展包格式增加类型字段对两种应用进行区分。其中 Interest 包类型为三种,即原有普通 Interest 包、订阅 Interest 包及取消订阅 Interest 包。Data 包也有三种,即普通类型、对订阅 Interest 包的回复及推送数据。此外由于实时应用数据具有时序性,Interest 包和 Data 包均增加序列号字段作为标记,增加字段及含义见表 1、表 2。

表 1 Interest 包增加字段及含义

| 字段 | 含义 | | |
|-----|----|--------------|------|
| 类型 | 普通 | 订阅 | 取消订阅 |
| 序列号 | - | 请求者已收到的最新序列号 | - |
| 窗口值 | - | 最多允许推送的数据包数 | - |

表 2 Data 包增加字段及含义

| 字段 | 含义 | | |
|-----|----|-------|-------------|
| 类型 | 普通 | 推送 | 订阅回复 |
| 序列号 | - | 数据序列号 | 数据源产生的最新序列号 |

订阅 Interest 包同样包含生存期字段,为防止链路出现异常,用户需要周期性发送订阅 Interest 包。

同时用户每发送一个订阅 Interest 包,数据源需要对其进行回复,以表示收到订阅信息,该回复中同时包含数据源目前产生的最新序列号信息,用于实

现数据同步。

此外,为方便对推送方数据进行流量控制,Interest包增加窗口值字段。

2.2 路由器添加订阅记录表

在NDN路由器中增加订阅记录表(subscribe pending interest table, SPIT)用于记录用户订阅信息。SPIT的数据结构和原NDN路由器中特定Interest表(PIT)相似,当路由器收到用户订阅信息后在SPIT中添加表项,记录订阅内容名字和Interest包入端口,当路由器收到数据源推送内容后,查找SPIT,对推送内容进行转发。

SPIT和原PIT的不同在于:(1)原有机制中PIT在收到Data包后会对表项进行删除,以此实现一个Interest包对应一个Data包,阻碍了后继数据的推送,而本文设计机制在收到Data包后SPIT表项不做删除,允许数据包的持续推送;(2)SPIT表项中增加序列号标记,每当路由器收到一个推送Data包后,判断序列号是否大于该标记,并进行更新,防止数据的重复推送以及意外导致的推送环路的发生。

在路由器中,SPIT与原PIT并存,路由器在收到Data包后根据Data类型查找不同的表项。对于推送数据发生网络丢包的情况,用户可通过发送正常请求获取丢失数据,其会在PIT中建立表项。

2.3 方案命名设计

NDN中内容的命名由应用决定^[16]。对于实时通信应用,由于内容具有时序性,因此内容命名通常为数据源标识加序列号,如:/cstnet/chat/usrA/001;对于本文设计机制,订阅Interest包的名字为用户要订阅的数据源标识,如:/cstnet/chat/usrA,路由器收到订阅信息后将其记入SPIT中,当数据源推送数据时,路由器根据内容名采用前缀匹配方式查找SPIT,将内容从记录端口转发。

2.4 转发流程设计

本文设计的NDN路由器包处理流程如图1和图2所示,转发流程算法如表3和表4所示。

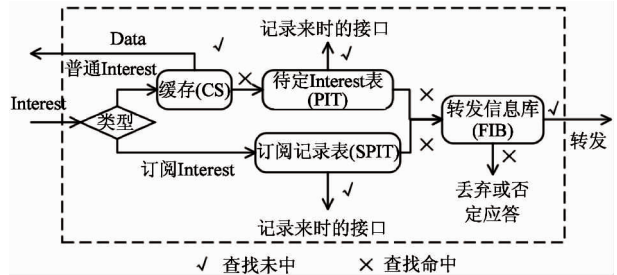


图1 Interest包转发流程

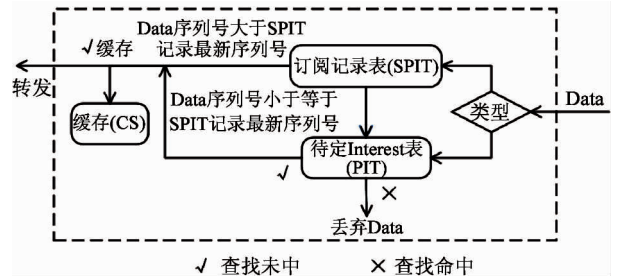


图2 Data包转发流程

表3 Interest包处理算法

```

1:/* Interest类型判断 */
   IF Interest类型 = 普通 THEN
       Goto 3
2:/* 订阅Interest包执行操作 */
   /* 更新SPIT */
   IF Interest类型 = 取消订阅 THEN
       查找SPIT,删除对应表项
       Exit
   ELSE
       /* Interest类型 = 订阅 */
       查找SPIT:
       IF 表项不存在 THEN
           添加SPIT表项记录入端口号
       ELSE
           SPIT表项添加入端口号
           表项保留计时器 = Interest包生存期
       Goto 4
3:原有Interest包处理流程
4:查找FIB表,找到Interest包转发端口
5:根据转发策略对Interest包进行处理
6:转发Interest包
    
```

表 4 Data 包处理算法

```

1: /* Data 类型判断 */
   IF Data 类型 = 普通 THEN
       Goto 3
2: /* 实时应用数据处理 */
   查找 SPIT
   IF 查找命中 THEN
       IF Data 类型 = 订阅回复 THEN
           Goto 5
       IF Data 序列号 <= SPIT 记录 THEN
           /* 对已推送数据的请求 */
           Goto 3
   ELSE
       更新 SPIT 表项记录序列号
       记录命中条目转发端口
       Goto 4
   ELSE
       丢弃非法 Data 包
3: 原有 Data 包处理流程
4: 按照缓存策略缓存 Data 包
5: 将 Data 包从记录端口转发出去

```

与文献[9,11]的方案相比,本文方案设计的优点在于没有更改原 NDN 路由器中路由和缓存模块, Interest 包路由方式和原有方式相同,与 NDN 原有机制相比,该方案不仅保留原有请求者“拉”模式的数据获取,还增加了对实时应用数据推送的支持。对于实时通信应用,请求者向数据源发送订阅信息后,数据源有新的数据便可直接推送出去,大大减小了数据接收延迟和网络传输代价。

2.5 网络丢包处理

某些物联网实时监控应用,少量数据丢失不需要进行重复请求^[9,11]。但是对于某些需要网络可靠性保证的应用,数据丢失需要用户重新请求。

针对网络传输带来的丢包问题,用户端可以通过两种方式发现丢失数据:(1)根据收到的最新推送数据中的序列号判断之前丢失序列号内容;(2)根据数据源对用户端周期性订阅 Interest 包的回复判断丢失序列号内容,因为该回复中包含数据源最新产生数据的序列号。

用户端在发现丢失数据后采用原有“拉”方式获取丢失序列号的数据。而 NDN 网内缓存将对数

据的再次获取带来加速的效果。

3 设计分析

本节主要分析原 NDN 传输机制与本文设计机制在实时数据传输中的数据接收延迟。对比方案包括 3 个:(1)同步方式^[8],即用户端周期性发送同步用 Interest 包,当发现数据源有新的消息产生时再分别请求对应的 Data 包;(2)滑动窗口数据请求机制^[17],即用户端发送固定窗口值的 Interest 包,当收到对应 Data 包后,发送窗口向后滑动,请求后继 Data 包,如果超时没有收到则重新请求;(3)本文设计订阅推送机制,用户端周期性发送订阅 Interest 包,路由器 SPIT 记录回送路径,数据源有数据产生便直接推送。

3.1 数据接收延迟分析

对于同步传输方式,令同步用 Interest 包的发送时间间隔为 $1/\lambda$, 期间数据源产生数据的个数为 μ/λ , 则用户端数据接收平均延迟为

$$AD_1 = \frac{\sum_{i=1}^{\frac{n\lambda}{\mu}} \sum_{i=1}^{\frac{\mu}{\lambda}} (\alpha_i + 3 \times \sum_{j=1}^k t_j)}{n}, \quad \alpha_i \in [0, 1/\lambda] \quad (1)$$

λ ——Interest 包平均发送速率

μ ——Data 包平均产生速率

α_i ——第 i 个数据产生到数据源收到下一个最近 Interest 包的时间间隔

t_j ——第 j 段链路的传输时延

k ——从数据源到用户的链路条数

n ——数据源一段时间内产生实时数据的个数

AD ——一段时间内平均数据接收延迟

对于滑动窗口传输方式,用户端以 λ 的速率发送 Interest 包,当用户端收到序列号 i 的 Data 包后,请求序列号向后滑动为 $i+1$ 。其和同步方式不同点在于,如果 $\lambda < \mu$, 用户端接收到的数据将长期滞后于数据源的数据产生,数据接收延迟将逐渐增大,因此该方式如果要及时获取最新的实时数据,需保证 $\lambda \geq \mu$ 。该方式用户端数据接收平均延迟为

$$AD_2 = \frac{\sum_{i=1}^n (\alpha_i + \sum_{j=1}^k t_j)}{n}, \quad \alpha_i \in [0, 1/\lambda] \quad (2)$$

对于基于 NDN 原有通信模式的方案设计,增加 Interest 发送速率即 λ , 可以减小 α_i , 从而减小平均接收延迟。

对于本文设计的订阅推送通信模式, 订阅保活用 Interest 包发送频率为, 当数据源收到用户端发送的订阅请求后, 有实时数据产生便直接推送, 用户端数据接收平均延迟为

$$AD_3 = \frac{\sum_{i=1}^n \sum_{j=1}^k t_j}{n} \quad (3)$$

其中, 由公式(1)、(2)、(3)可见, 在 λ 相同的情况下,

$$AD_1 - AD_2 \approx 2 \times \sum_{j=1}^k t_j \quad (4)$$

$$AD_2 - AD_3 = \frac{\sum_{i=1}^n \alpha_i}{n}, \alpha_i \in [0, 1/\lambda] \quad (5)$$

由式(4)和式(5)可见, 本文设计的方案在实时数据接收延迟上比 NDN 原有通信模式要低。

3.2 缓存和网络丢包的影响

与内容分发类应用不同, 实时通信类应用中数据源和最新产生的实时数据具有唯一性。

对于 NDN 原有通信方式, 单用户在无网络丢包的情况下, 网内缓存对实时数据传输没有影响, 因为网络中没有缓存用户所需的数据。而在多用户无网络丢包的情况下, 由于不同用户 Interest 包的请求时刻不同, 缓存对实时数据的获取有一定的加速作用。同样, 对于网络出现丢包的现象, 对于用户的重复请求, 网内缓存对于实时数据的获取也会带有一定的加速作用。

对于本文设计的通信方式, 在网络无丢包的情况下, 网内缓存对实时数据的推送没有影响, 但在网络出现不同情况的丢包时, 中间路由节点如果缓存有推送的数据, 缓存对用户的重复请求将起到一定的加速作用。

本文设计机制没有涉及原有 NDN 路由器缓存模块的变动, 因此不影响现有缓存模块的扩展性需求。由于缓存对实时数据传输没有带来决定性影响, 而具体缓存策略及大小设置因涉及因素较多, 非本文机制探讨的重点, 有待后继开展相关研究。

3.3 方案性能分析

在本文设计机制中, 对于原有普通类型 Interest

包和 Data 包, 路由器仅多一步类型判断, 其余处理流程仍按原有机制进行。对于订阅 Interest 包和推送 Data 包, 本文机制和原机制不同在于需要查找 SPIT, 执行后继操作, 而 SPIT 的组织方法和查找算法与原机制中 PIT 完全相同, 因此, 本文机制在路由器处理时延上没有带来过多影响。

4 实验结果

本文采用 ndnSIM^[12] 进行相关实验, 实验对比为第 3 节分析的基于原有 NDN 通信模式的同步方式和滑动窗口机制。为便于分析, 设置数据源数据产生平均速率为 1 个/秒, 对应设置滑动窗口机制的窗口值为 1。实验验证结果表明: (1) 在无缓存、无丢包情况下, 本文设计的机制在数据接收时延上优于原有传输方案设计; (2) 网络丢包和网内缓存对数据接收延迟带有影响, 在该影响下本文设计的机制仍优于原有传输方案; (3) 用户数量对 NDN 数据接收延迟带有一定影响。

4.1 单用户数据接收延迟

实验拓扑如图 3 所示, 相关参数如表 5 所示。



图 3 单用户实验拓扑

表 5 实验主要参数设置

| 参数 | 取值 |
|-----------|---------------------|
| 每条链路带宽 | 100Mbps |
| 每条链路延迟 | 50ms |
| 数据源数据产生速率 | 均值 1 个/秒泊松分布 |
| 数据产生个数 | 1000 |
| 数据包大小 | 1024Byte |
| 缓存替换策略 | LRU ^[12] |

在链路无丢包, 并且不设置网内缓存的情况下, 三种方式用户端数据接收延迟随用户端 Interest 包发送频率变化如图 4 所示。对于订阅推送机制, 数据源收到用户订阅 Interest 包后, 有实时数据产生便

直接推送,故数据接收延迟基本等于单向网络传输时延,且不受发送速率影响。而基于原有 NDN 传输机制的同步方式和滑动窗口方式,数据接收延迟在一定范围内随用户端 Interest 包发送频率增加而减小,但平均速率高于本文设计订阅推送方式。图 5 为三种方式在 Interest 发送速率为 5 个/秒时时延的累积概率密度曲线,其中本文设计订阅推送机制基

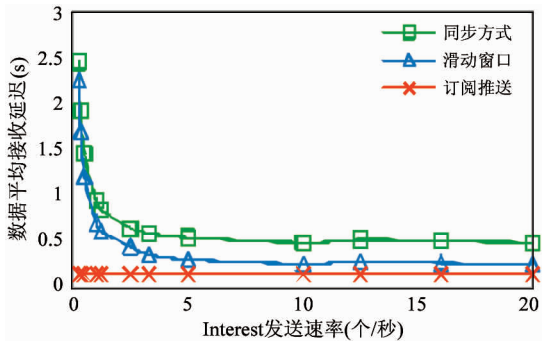


图 4 数据接收延迟随 Interest 发送速率变化

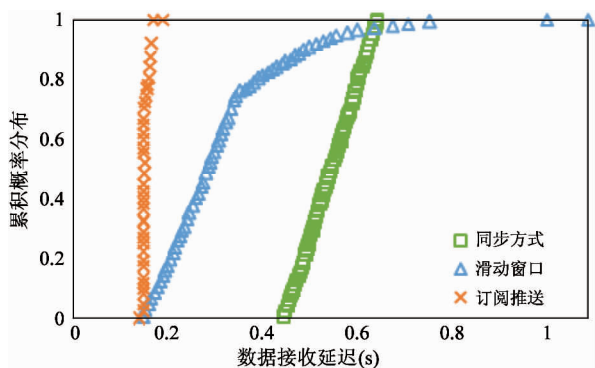


图 5 数据接收延迟累积概率分布

本所有数据的接收延迟都在 0.2 秒以内。(由于实验数据量较大,图 5 及下文涉及概率密度曲线的图片实验数据均做按比例采样展示)

链路丢包对实时数据接收延迟带来一定负面影响,图 6 为用户端 Interest 发送速率为 5 个/秒时,链路 L1 不同丢包率下三种方式数据接收平均延迟。其中随着链路丢包率的上升,三种方式数据接收延迟增大。同步机制和本文设计机制均可通过周期性发送的同步 Interest 包获取数据源目前最新产生的内容名字,对丢失数据及时请求获取,而滑动窗口机制因网络丢包未获取最新数据而延缓了窗口的向后

滑动,增加了数据接收延迟,故随链路丢包率的提高,数据接收延迟增大幅度比另两种快。

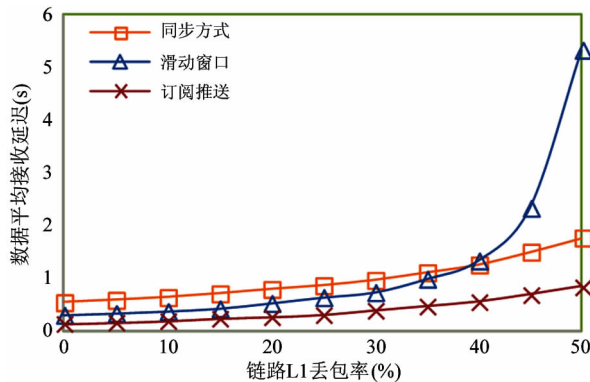


图 6 数据接收延迟随 L1 丢包率变化

由于实时通信类应用中数据源和最新产生的实时数据具有唯一性,因此在单用户无丢包的情况下,NDN 节点缓存对实时数据接收没有影响。然而在网络出现丢包时,缓存对于丢失数据的重复请求带有一定的加速作用。图 7 为用户端 Interest 发送速率为 5 个/秒,链路 L1 丢包率为 20% 时,三种传输方式在节点有无缓存情况下数据接收延迟累积概率分布。其中对于订阅推送方式,有 80% 左右的数据包通过推送方式到达用户,所以延迟在 0.15 秒左右的数据占 80% 左右,其不受缓存大小的影响。有 20% 左右的推送数据出现了丢包,通过用户端重新请求获得,由于丢包数据需要用户端发现并重新请求,用户接收时延明显增大,因此图 7 中曲线出现了断点,而该部分数据由于缓存的存在,数据接收延迟得到普遍降低,订阅推送方式数据平均接收延迟比不带缓存的降低了 9.5%。

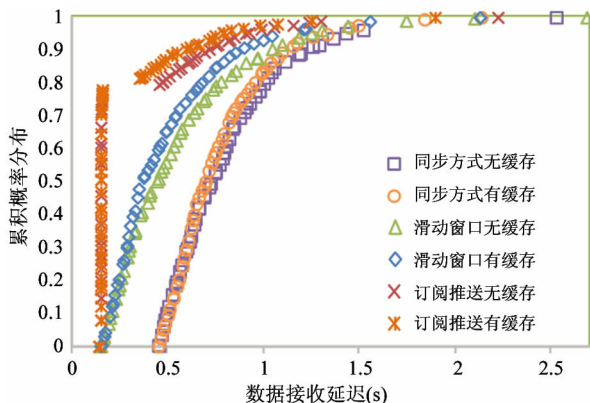


图 7 缓存及丢包影响下接收延迟累积概率分布

由此可见,在单用户情况下,本文设计机制在实时数据接收延迟上优于原有 NDN 传输模式,在出现网络丢包的情况下,NDN 网内缓存对数据获取带有一定的加速作用。

4.2 多用户影响下数据接收延迟

和现有 TCP/IP 相比,NDN 的优势之一在于天然多播和网内缓存,在原有机制中,由于不同用户对最新实时数据的请求时刻不一定相同,存在有用用户先请求,数据被中间节点缓存,使得其它用户数据请求获得加速。因此针对单用户实验设计多用户实验拓扑,如图 8 所示,其中用户 2、3 作为实验影响因子,对用户 1 数据接收进行统计分析,相关实验参数同表 5。

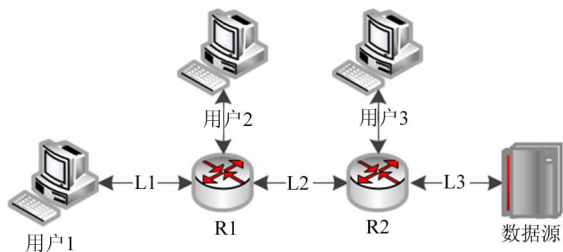


图 8 多用户实验拓扑

作为对比实验,图 9 展示了在链路无丢包且不设节点缓存时的三种方式下,用户 1 数据接收延迟随 Interest 发送频率变化的情况,图 10 展示在无网内缓存,Interest 发送速率为 5 个/秒时,链路 L1 不同丢包率下,用户 1 在三种方式下的数据平均接收延迟。图 11 展示 Interest 发送速率为 5 个每秒,链路 L1 丢包率为 20%,用户 1 数据接收延迟随节点

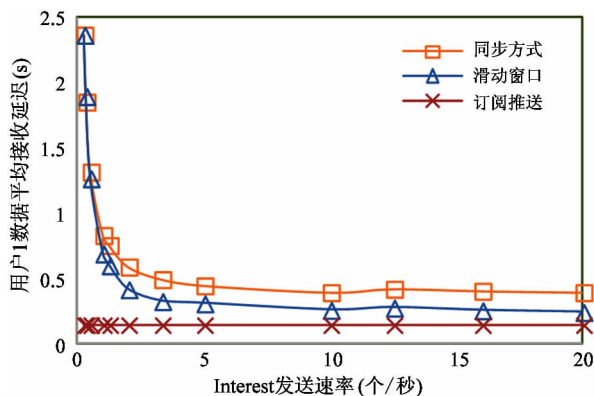


图 9 数据接收延迟随 Interest 发送速率变化

缓存大小变化的情况。其中本文设计机制和基于原有 NDN 机制的设计方案相比,数据平均接收延迟降低 41% ~ 62%。

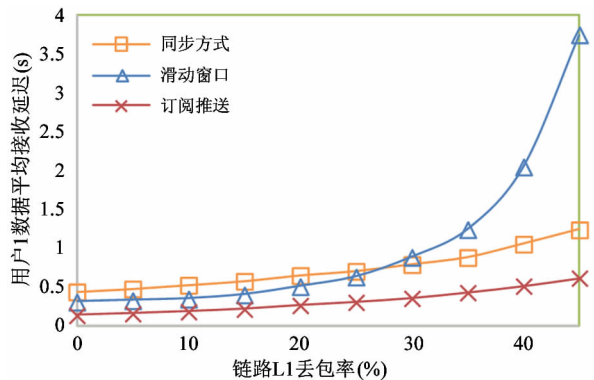


图 10 数据接收延迟随 L1 丢包率变化

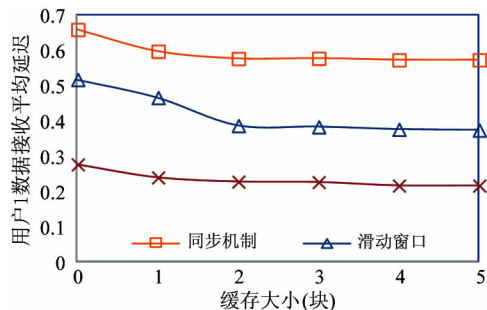


图 11 缓存大小对接收延迟影响

图 12 展示订阅推送机制,Interest 发送速率为 5 个/秒,链路 L1 丢包率为 20%,用户 1 数据接收延迟累积概率分布受多用户、缓存的影响情况。由于 80% 未丢包数据不受缓存及用户数量影响,图中主要展示 20% 丢包部分的数据传输延迟。其中节点

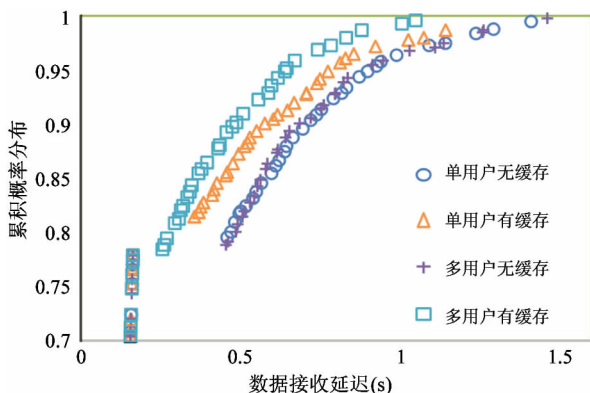


图 12 多用户及缓存影响下延迟累积分布

没有缓存时,单用户和多用户影响下的订阅推送机制基本没有差异。而有缓存的情况下,多用户影响的数据平均接收延迟比单用户的数据接收平均延迟降低7.3%,因为之前丢包的数据可能被其它用户获取过而被中间节点缓存,当用户请求该丢包数据时,延迟将被缩短。

5 相关讨论

作为网络通信传输机制,本文设计在路由选择、链路异常、流量控制方面均有所考虑。

5.1 路由选择

NDN 中路由一方面通过路由协议如命名数据链路状态路由 (named-data Link State Routing, NLSR) 协议^[14]学习得到并记入 FIB,另一方面通过基于端口状态标记的自适应路由方式^[15]。对于第一种方式,本文设计机制并不影响正常的路由学习,并且可以利用学习到的 FIB 信息转发用户端发送的订阅 Interest 包。对于第二种方式,原有机制根据接口状态(连接、断开)以及发送的 Interest 包在正常往返时间内是否收到对应 Data 包来标记端口状态,本文设计中,通过用户端发送订阅 Interest 包,数据源回复带有数据源最新序列号的回复数据包也可实现端口状态的鉴别,如果中间路由器收到订阅 Interest 包但由于接口断开等原因无法转发,同样可以回复 NACK 包^[15]告知前一跳此路不通,同时不添加 SPIT 表项。

因此,本文设计机制完全兼容原有 NDN 路由设计及实现,且不会在性能上造成额外影响。

5.2 链路异常

当链路发生偶然的异常变动,用户周期性发送的订阅 Interest 包将按路由器路由规则和相应转发策略选择新的端口进行自适应转发,因此旧路径上的路由器 SPIT 表项超时会被删除,新路径上的路由器 SPIT 表项会有相应记录,数据源推送数据将沿新的路径传输,旧的路径上将不会再收到推送的数据。

5.3 流量控制

首先,在网络带宽及网络节点处理能力充足的情况下不需要进行流量控制,但是如果网络资源在

短时间内匮乏或者数据源受到攻击者攻击,这时数据源大量推送的数据应当受到抑制,以防加重网络负担。

本文通过在订阅 Interest 包中增加窗口值字段,限定上游数据源推送数据速率。在转发订阅 Interest 包时,每跳节点可以根据网络当前状态更新窗口字段值,内容为一段时间内允许上游推送数据包个数,上游节点在收到传来的订阅 Interest 包后会在 SPIT 中进行标记,限定向对应端口 Data 包的推送速率,以此实现推送流量控制。

6 结论

实时通信类应用数据产生具有一定突发性和随机性,且时延敏感,传统命名数据网络(NDN)用户驱动的“拉”模式数据获取方式无法满足实时通信应用需求。本文设计了基于订阅推送的 NDN 实时通信机制,该机制首先在 NDN 路由器增加订阅记录,随后实时应用数据便可直接推送给订阅用户,从而大大减小数据的接收延迟。同时该机制可以和原 NDN 传输机制并存,借用网内缓存减小因网络丢包造成的接收延迟。相关实验及分析表明,该机制在数据接收延迟上明显优于基于“拉”方式的传统 NDN 传输方案。

参考文献

- [1] 谢高岗, 张玉军, 李振宇等. 未来互联网体系结构研究综述. 计算机学报, 2012, 35(6): 1109-1119
- [2] Zhang L, Afanasyev A, Burke J, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 2014, 44(3): 66-73
- [3] Ghodsi A, Shenker S, Koponen T, et al. Information-centric networking: seeing the forest for the trees. In: *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, Cambridge, USA, 2011. 1
- [4] Ahlgren B, Dannewitz C, Imbrenda C, et al. A survey of information-centric networking. *IEEE Communications Magazine*, 2012, 50(7): 26-36
- [5] Bari M F, Chowdhury S, Ahmed R, et al. A survey of naming and routing in information-centric networks. *IEEE Communications Magazine*, 2012, 50(12): 44-53

- [6] Wang J, Peng C, Li C, et al. Implementing instant messaging using named data. In: Proceedings of the 6th Asian Internet Engineering Conference, Bangkok, Thailand, 2010. 40-47
- [7] Zhu Z, Bian C, Afanasyev A, et al. Chronos: serverless multi-user chat over NDN, Technical Report NDN-0008. Los Angeles: University of California, 2012
- [8] Zhu Z, Afanasyev A. Let's chronoSync: decentralized dataset state synchronization in named data networking. In: Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP), Göttingen, Germany, 2013. 1-10
- [9] Carzaniga A, Papalini M, Wolf A L. Content-based publish/subscribe networking and information-centric networking. In: Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, Toronto, Canada, 2011. 56-61
- [10] Jacobson V, Mosko M, Smetters D, et al. Content-centric networking. *Whitepaper, Palo Alto Research Center*, 2007; 2-4
- [11] François J, Cholez T, Engel T. CCN traffic optimization for IoT. In: Proceedings of the 4th International Conference on Network of the Future, Pohang, Korea, 2013
- [12] Mastorakis S, Afanasyev A, Moiseenko I, et al. ndnSIM 2.0: a new version of the NDN simulator for NS-3. Technical Report NDN-0028. Los Angeles: University of California, 2015
- [13] Afanasyev A, Shi J, Zhang B, et al. NFD developer's guide. Technical Report NDN-0021. Los Angeles: University of California, 2014
- [14] Hoque A K M, Amin S O, Alyyan A, et al. Nlsr: Named-data link state routing protocol. In: Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking, Hong Kong, China, 2013. 15-20
- [15] Yi C, Abraham J, Afanasyev A, et al. On the role of routing in Named Data Networking. In: Proceedings of the 1st International Conference on Information-centric Networking, Paris, France, 2014. 27-36
- [16] Yu Y, Afanasyev A, Zhu Z, et al. Ndn technical memo: naming conventions. Technical Report NDN-0022. Los Angeles: University of California, 2014
- [17] Jacobson V, Smetters D K, Briggs N H, et al. VoCCN: voice-over content-centric networks. In: Proceedings of the 2009 Workshop on Re-architecting the Internet, Rome, Italy, 2009. 1-6

Subscription push based real-time communication mechanism for NDN

Jin Pengfei^{***}, Li Jun^{*}, Wu Haibo^{*}, Zhi Jiang^{***}

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

(** University of Chinese Academy of Sciences, Beijing 100049)

Abstract

The study focused on the real-time communication mechanism of the promising Internet architecture of named data networking (NDN), and pointed out that NDN users obtain contents by sending out request messages, while for real-time communications such as instant chatting or monitoring Internet of things, the synchronization of the request rate of users and the data production rate is difficult to realize and the needs of real-time applications are unable to meet when using the conventional user-driven pull mode for data requiring because of data generation's abruptness and randomness as well as the time delay of data receiving. For the purpose of reducing the data response delay, the study presented a subscription-push-based real-time communication mechanism on the basis of unmodifying conventional NDN routing and caching mechanisms. Under the new mechanism, the subscription information is added into NDN routers in advance, and later the real-time producer could push data to the subscriber immediately. The experimental results show that this new mechanism can greatly reduce the reception delay compared with the original NDN design.

Key words: named data networking (NDN), real-time communication, subscribe, push