ICN 中改进的拥塞控制和请求转发策略研究

赵巾帼，罗庆云，任长安
（湖南工学院 计算机与信息科学学院，湖南 衡阳 421002）

摘　要： 针对信息中心网络(ICN)中终端用户体验不佳，网络资源利用率低等不足，对 ICN 中联合多径拥塞控制和请求转发问题展开研究。将该问题看成是以用户吞吐量最大化和总体网络成本最小化为目标的全局优化问题。通过分解策略解决这一全局优化问题，为接收机确定一组最优拥塞控制策略，为网络节点确定一组分布式动态请求转发算法。为了评估其性能，使用 CCNx 部署该协议，并设置测试床用于大规模实验仿真。不同网络场景下的实验评估证明了该方案的效率和全局公平性，达到最优方案的预期效果，即使面对路径内缓存和网络拥塞导致的延时不确定性也具有稳定的性能。

关键词： 信息中心网络；资源利用率；拥塞控制；请求转发；测试床
中图分类号： TP393　　**文献标志码：** A　　**文章编号：** 1001-3695(2016)02-0575-05
doi:10.3969/j.issn.1001-3695.2016.02.059

Research on improved congestion control and request forwarding strategies in information-centric networks

Zhao Jinguo, Luo Qingyun, Ren Changan
(*School of Computer & Information Science*, *Hunan Institute of Technology*, *Hengyang Hunan* 421002, *China*)

Abstract: In order to resolve the lacking of end-user experience and the low utilization of network resources in information centeic network (ICN), this paper carried out a research of joint multipath congestion control and request forwarding in ICN. It formulated the question as a global optimization problem with the twofold objective of maximizing user throughput and minimizing overall network cost, decomposed the problem for the purpose of fixing it. It proposed a group of optimal congestion control strategies for the receiver and a set of distributed algorithms for dynamic request forwarding at network nodes. To assess the performance of the design, deployed the proposed protocols in CCNx and a set up testbed for large scale experimentation. Experimental evaluations under different network scenarios confirm the efficiency and fairness of the overall scheme, the desired effect of the optimal solution is achieved. Even in the face of the delay uncertainty caused by the in-path caching and network congestion the scheme has a stable performance.

Key words: information centric networks; resource utilization; congestion control; request forwarding; testbed

0　引言

过去几十年间，互联网的架构一直以 ICP/IP 协议为核心，并且在上层和下层架构拥有很多创新。然而，IP 寻址和基于名称的位置无关内容寻址之间的不匹配现象，以及当前的点对点通信模式，都严重阻碍了信息驱动型网络技术的发展。Web 服务、云技术和社交网络的普及，推动互联网向信息传递方向发展，并诞生了一种以信息交换、缓存和实时处理为基础的新的通信模型：信息中心网络[1]（ICN）。在 ICN 中，信息传递解决方案大多采取覆盖策略部署于当前 IP 网络基础设施的顶端，同时采用了 CDNs 和透明缓存等技术。然而，学界已经指出覆盖策略在性能稳定性和资源利用率方面存在不足[2,3]，其可持续性亦存在问题。

ICN 的传输模式之所以能显著提升灵活性、终端用户性能以及网络资源利用率，主要归功于如下三个因素间的耦合性：a)发送方未知的无连接通信方式；b)接收方的唯一端点；c)由网络节点以逐段转接方式为实现内容临时拷贝本地化而确定的动态请求路由。

大量文献研究了 TCP/IP 网络的联合多径路由和拥塞控制问题。其中，文献[4]提出了一种新的面向服务和信息的网络体系结构 SIONA。它采用了 IP 地址加信息内容的 hash 值作为信息名字，支持缓存加多源组播的路由模式，阐述了路由器反馈接收端驱动的多窗口的拥塞控制协议，解决了 ICN 的可扩展和部署难问题。文献[5]提出了一种更好的支持网络可感知、应用可区分、策略可管理的数据中心网络新型传输控制模型。文中对 IB 传输控制核心的 1B 拥塞控制算法及其参数进行了深入的分析，证明了在流级通过区别化的拥塞控制参数配置能够实现成比例的带宽划分。最后，在真实系统中的实验验证了通过参数的动态配置能够保持租户级的公平带宽共享，更好地支持云数据中心的带宽隔离需求。文献[6]研究了多径控制器的稳定性，获得了可以与 TCP 兼容的多种最优控制器。文献[7]中的多径 TCP 从高效的网络使用率和 RTT 公共补偿方面对先前的设计进行了改进。然而这些方法都不适合于 ICN，因为 ICN 的数据源事先未知且会随着时间变化，同

收稿日期： 2014-10-04；**修回日期：** 2014-11-27　**基金项目：** 湖南省科技计划项目(2013FJ3095)
作者简介： 赵巾帼(1965-)，女，湖南邵阳人，副教授，硕士，主要研究方向为无线传感网、网络算法；罗庆云(1965-)，男，教授，主要研究方向为无线传感网、云计算(48489502581@qq.com)；任长安(1980-)，男，山东菏泽人，讲师，硕士，主要研究方向为进化算法、云计算.

时还受到网络内动态缓存和网络节点在线请求转发的约束。文献[8]对 ICN 的自适应转发机制作了介绍。文中明确了拥塞控制的定义,并通过返回给接收端的 Interests NACK(否定确认)作了假设。然而这些明确的拥塞控制方法可能会要求对信息进行优先级排序,导致额外开销。为了弥补以上方法的不足,本文研究了 ICN 中的联合多径拥塞控制和请求转发问题。

1 网络模型

在 ICN 中,信息目标被分为多个数据报文,用唯一的名称标志,且永久性地存储于一个或多个仓库中。报文请求(本文又称为 Interests)通过基于名称的路由沿着一条或多条路径向一个或多个仓库传输。由终端用户进行速率和拥塞控制。中间节点跟踪数据结构中称为 PIT(待定兴趣表)的待定 Interests,以便通过反向路径把被请求的数据返回给接收机。每个 PIT 项均关联了一个计时器,只要首次检索没有解决,在该时间间隔内申请相同数据的所有请求都不会向上游转发。此外,节点在称为内容仓库的本地缓存中临时存储数据报文。当从输入接口接收到 Interests 后,中间节点执行如下操作:a)内容存储查询,以确定被请求的数据是否在本地存储,如果存在,则通过使用发送兴趣报文的接口将数据传输出去;b)否则,PIT 查询,以确认存在针对相同内容名称的 PIT 项,此时,丢弃 Interests,因为已经有一个待处理请求还未处理完毕,如果没有该 PIT 项,则创建新的 PIT 项;c)通过最长前缀匹配方法进行 FIB 查询[9],返回用于转发 Interests 的接口(从潜在接口中选择)。

1.1 问题定义

网络建模为带有双向弧的有向图 $\zeta=(V,A)$:如果流 n 的 Interests 通过链路(i,j),则对应的数据根据 ICN 的路由对称原则通过链路(j,i)传输。链路的容量有限,$C_{ij}>0,\forall(i,j)\in A$。内容检索(也表示为流)$n\in N$ 明确关联于 $U\subset V$ 中的用户节点以及一个或多个仓库。变量 $x_{ij}^n(\tilde{x}_{ji}^n)$ 表示流 n 在链路$(i,j)(j,i)$上的数据率,单位为报文/s。对每个节点 $i\in V$,定义 $\Gamma^+(i)$ 和 $\Gamma^-(i)$ 分别表示出口节点集合和入口节点集合。每个网络节点 $i\in V$ 有一个容量有限的缓存以存储正在传输的数据。设 $h^n(i)\in\{0,1\}$ 表示二进制函数,当节点 i 可以满足流 n 的数据时,函数值为 1,否则为 0。于是,用 $S(n)=\{i:h^n(i)=1\}$ 表示流 n 的可用源集合。相关符号的含义如表 1 所示。

表 1 相关标记法

参数	定义
V,A,U	顶点,圆弧,用户集
N	流(内容检索)集
$L(n)=\{(i,j)\in A:x_{ij}^n>0\}$	流 n 使用的链路集
C_{ij}	$(i,j)\in A$ 链路的容量
$h^n(i)\in\{0,1\}$	二进制缓存函数(流 n,节点 i)
$S(n)=\{i:h^n(i)=1\}$	流 n 的源节点集合
$x_{ij}^n(\tilde{x}_{ji}^n)$	链路数据率
$y^n(\tilde{y}^n),y_i^n$	总数据率
ρ_{ij}	$(i,j)\in A$ 的链路负载
$\Gamma^+(i)=\{j\in V:(i,j)\in A\}$	$i\in V$ 的出口节点集合
$\Gamma^-(i)=\{\ell\in V:(\ell,i)\in A\}$	$i\in V$ 的入口节点集合

本文研究的优化问题的全局目标是用户性能最大化以及网络成本最小化。考虑用户的凸效用函数 U_n 及链路成本凹函数 C_{ij},本文将全局目标定义为用户总体效用和网络总成本间的差值如式(1)。问题被阐述为带有线性约束和凸目标的多商品流问题:

$$\max_y\sum_n U_n(y_i^n)-\sum_{i,jA}C_{ij}(\rho_{ij}) \tag{1}$$

$$\sum_{n\in N}x_{ij}^n=\rho_{ij},\forall(i,j)\in A \tag{2}$$

$$\sum_{\ell\in\Gamma^-(i)}x^n_{\ell i}=y_i^n,\forall i,n \tag{3}$$

$$\sum_{j\in\Gamma^+(i)}x_{ij}^n(1-h^n(i))=y_i^n,\forall i\notin U,n \tag{4}$$

$$\rho_{ij}\leqslant C_{ij},\forall(i,j)\in A \tag{5}$$

$$x_{ij}^n\geqslant 0,\forall i,j,n \tag{6}$$

其中:利用 ρ_{ij},将链路负载表示为通过链路(i,j)的总流率;而 y_i^n 表示流 n 到达节点 i 的到达率,且定义为入口率 $x^n_{\ell i},\ell\in\Gamma^-(i)$之和(式(3));$y_i^n$ 也等于出口率 $x_{ij}^n,j\in\Gamma^+(i)$之和,除非 $h^n(i)=1$(因此,Interests 在本地得到满足,无须向上游转发)。这对除了流 n 用户节点之外的所有其他节点均成立,且 $x_{ij}^n=0,\forall j$。数据率受到式(5)的链路容量约束,且不为负(式(6))。

1.2 问题分解

根据上述两个目标对问题进行分解:终端用户吞吐量的效用最大化(接收方),以及网络成本最小化(路径节点上)。

a)通过固定辅助变量 y_i^n 可以得到如下的子问题 1(终端用户吞吐量的效用最大化问题):

$$\begin{cases}\max\limits_y\sum\limits_n U_n(y^n) & (7)\\ \sum\limits_{n:(i,j)\in L(n)}y^n\leqslant C_{ij},\forall(i,j)\in A & (8)\\ y^n\geqslant 0,\forall n\in N & (9)\end{cases}$$

其中:$L(n)=\{(i,j)\in A:x_{ij}^n>0\}$ 表示流 n 使用的所有链路集合。根据文献[10]可知,式(7)中的效用最大化表达式表示分布式拥塞控制。第 3 章将给出如何在接收端通过 Interests 的传输控制获得 ICN 的最优拥塞控制器。

b)以网络成本最小化为优化目标,可以得到如下的子问题 2 :

$$\begin{cases}\min\limits_x\sum\limits_{i,j\in A}C_{ij}(\sum\limits_{n\in N}x_{ij}^n) & (10)\\ \sum\limits_{\ell\in\Gamma^-(i)}x^n_{\ell i}=y_i^n,\forall i,n & (11)\\ \sum\limits_{j\in\Gamma^+(i)}x_{ij}^n(1-h^n(i))=y_i^n,\forall i\notin U,n & (12)\\ x_{ij}^n\geqslant 0,\forall(i,j)\in A,\forall n\in N & (13)\end{cases}$$

为了求解以上两个子问题,分别计算拉格朗日算子 L_U、L_C:

$$\begin{aligned}L_U(y,\lambda)&=\sum_n U_n(y^n)-\sum_{ij}\lambda_{ij}(\sum_{n:(i,j)\in L(n)}y^n-C_{ij})=\\&\sum_n U_n(y^n)-\sum_n\sum_{(i,j)\in L(n)}\lambda_{ij}y^n+\sum_{ij}\lambda_{ij}C_{ij}=\\&\sum_n(U_n(y^n)-\lambda^n y^n)+\sum_{(i,j)}\lambda_{ij}C_{ij}\\L_C(x,\mu)&=\sum_{i,j}C_{ij}(\sum_{n\in N}x_{ij}^n)-\sum_n\sum_i\mu_i^n(\sum_{l\in\Gamma^-(i)}x_{li}^n-\sum_{j\in\Gamma^+(i)}x_{ij}^n)=\\&\sum_{i,j}C_{ij}(\rho_{ij})-\sum_n\sum_{i,l}\mu_i^n x_{li}^n+\sum_n\sum_{i,j}\mu_i^n x_{ij}^n=\\&\sum_i\sum_{l\in\Gamma^-(i)}[C_{li}(\rho_{li})-\sum_n(\mu_i^n-\mu_l^n)x_{li}^n]\end{aligned}$$

其中,$\lambda^n\equiv\sum_{(i,j)\in L(n)}\lambda_{ij}$。每个用户通过使本地拉格朗日算子 L_U^n 最大化来实现终端用户吞吐量的效用最大化:

$$L_U^n=U_n(y^n)-\lambda^n y^n \tag{14}$$

每个节点 i 通过减小本地拉格朗日算子 L_{C_i} 来实现网络成本最小化:

$$L_{C_i}=\sum_{l\in\Gamma^-(i)}[C_{li}(\rho_{li})-\sum_n(\mu_i^n-\mu_l^n)x_{li}^n] \tag{15}$$

为了求解式(14)和(15),下面分别提出了 Interests 的传输控制协议以及确定将在网络节点处执行的 Interests 接口选择协议。

2　接收端驱动的拥塞控制

在 ICN 中,通过调整接收端的 Interests 传输速率 $\tilde{y}_t^n$ 来实现最优数据传递率 y_t^n。为了获得一组最优控制器,使用文献[11]中的梯度算法来求解式(14)中的效用最大化子问题,有

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_{ij}(t)=k_{ij}(t)\Big(\sum_{n;(i,j)\in l(n)}y^n(t)-C_{ij}\Big) \tag{16}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}y^n(t)=\gamma^n(t)(U'_n(y^n(t))-\lambda^n(t)) \tag{17}$$

其中:$k_{ij}(t)$是链路(i,j)的函数,$\lambda_{ij}(t)$是(i,j)的链路延时,$\lambda^n(t)$是流 n 经历的网络总体延时。将 $\lambda^n(t)$分解为路由延时 $\lambda^n(r,t)$之和,其中路由 $r\in R^n$ 可以确定从任一数据源 $s\in S(n)$ 到接收端节点间节点的唯一序列,且对流 n 有 $\lambda^n(t)=\sum_{r\in R^n}\lambda^n(r,t)\phi(r,t)$,其中 $\phi(r,t)$表示在时间 t 通过路由 r 传输的流的比例(由请求转发算法采取分布式策略确定)。同时,有

$$\lambda^n(r,t)=\sum_{(i,j)\in r;r\in R^n}\lambda_{ij}(t)$$

显然,接收端没有为已知数据报文提供服务的缓存的相关知识。同时,还需要确定可用的快速和慢速路由,因为这一信息在本地使用时可以更好地调节 Interests 传输策略。通过明确的标志可以区分不同路由,如文献[10]所示。式(17)变为

$$\frac{\mathrm{d}}{\mathrm{d}t}y^n(t)=\gamma^n(t)\Big(U'_n(\tilde{y}^n(t))-\sum_{r\in R^n(s)}\lambda^n(r,t)\phi(r,t)\Big) \tag{18}$$

请注意,如果速率的降低量与路由延时的组合成正比,则每个流的速率增长量都具有唯一性,因为它取决于总体流速率的最大值。根据文献[10],通过设置 $\gamma^n(t)=K_1(t)(y^n(t))^2$ 及 $U(y)=K_2(t)/(\lambda^n(t)y^n(t))$,可以选择 AIMD(加性增量乘性减量)控制器。于是,对所有的流,定义如下 Interests 传输率控制方程:

$$\frac{\mathrm{d}}{\mathrm{d}t}y^n(t)=\frac{K_1K_2}{\lambda^n(t)^2}-K_1y^n(t)\sum_{r\in R}\lambda^n(r,t)\phi(r,t)y^n(t) \tag{19}$$

进一步地,根据式(19)定义了一种接收端驱动型拥塞控制机制:根据基于窗口的 AIMD 机制,按照应用确定的次序,通过 Interests 来请求数据。流 $n\in N$ 的$\tilde{y}^n(t)$由 Interests 拥塞窗口 $W^n(t)$来控制,且该窗口保存于接收端,可以确定接收端允许发送的待处理 Interests 最大数量。通过假设 $W^n(t)=\tilde{y}^n(t)\cdot R^n(t)=y^n(t)\cdot R^n(t)$,便可以根据式(19)确定拥塞窗口的动态性,其中 $R^n(t)\equiv\lambda^n(t)$表示对其所有路由 $r\in R^n$ 取均值后流 n 经历的总体双程延时。

3　动态请求转发

在 ICN 中,无论是接收端还是网内节点通过对转发接口的选择问题作出在线决策,便可以进行报文间动态请求调度(网络内请求调度)。根据第 1 章对本文问题的定义,通过结合路径节点对 L_c 进行分解,实现分布式请求转发,每个节点需要实现最小化的本地目标在式(15)中给出。为了获得一组最优分布式转发算法,首先证明如下引理。

引理 1　每个网络节点 $i\in\nu$ 式(15)的最小化等价于对所有的 $n\in N$ 使 μ_i^n 最小。

证明:首先对式(15)施加 Karush-Kuhn-Tucker 条件,即对所有的入口接口 $j\in\Gamma^-(i)$,有

$$\frac{\partial C_{ji}}{\partial x_{ji}^n}=\frac{\partial C_{ji}}{\partial\rho_{ji}}\frac{\partial\rho_{ji}}{\partial x_{ji}^n}=\frac{\partial C_{ji}}{\partial\rho_{ji}}=\mu_i^n-\mu_j^n \tag{20}$$

于是有 $x_{ji}^n>0$ 且

$$\frac{\partial C_{ji}}{\partial x_{ji}^n}\leqslant\frac{\partial C_{li}}{\partial x_{li}^n},\forall l\in\Gamma^-(i) \tag{21}$$

式(21)表明,选择在节点 i 进行 Interests 转发的接口都是那些实现了成本导数最小化的接口。这表明,可能有一或多个接口具有相等的最小成本导数。对每一个这样的接口,再次重复推理过程,对所有后续传输中继段 $j,j+1,j+2,\cdots,r$ 运行式(20)直到有一个数据源 $s\in S(n)$。在最后一个中继段,成本导数的最小值直接对应于 μ_r^n 的最小值,在先前的中继段上,被最小化的变量为 $k=j+1,\cdots r$ 时的差值为 $\mu_k^n-\mu_{k+1}^n$,且第 1 段时为 $\mu_i^n-\mu_j^n$。很显然,从数据源开始,只要 μ_r^n 最小化且表示为 $\mu_r^{n,*}$,则$(\mu_{r-1}^n-\mu_r^{n,*})$的最小化就等价于$\mu_{r-1}^n$的最小化,依此类推,直到 μ_i^n 为止。

因此,可以得出结论,通过对每个节点 i 及 $\forall n\in N$ 使 μ_i^n 最小化,便可获得一组用于转发请求的分布式最优算法。为了计算 μ_i^n,对拉格朗日算子 $L_C(x,\mu)$使用文献[9]中的梯度算法,于是有

$$\frac{\mathrm{d}\mu_i^n}{\mathrm{d}t}=\eta_i^n\Big(\sum_{j\in\Gamma^+(i)}x_{ij}^n-\sum_{l\in\Gamma^-(i)}x_{li}^n\Big) \tag{22}$$

式(22)表明,最优策略在于能够实现节点 i 待处理 Interests 总数量最小化。为了确定一种最优接口选择算法,将每个输出接口的 Interests 区分开来,利用利特尔等价法则将 Interests 的数量和相应的响应时间结合起来。省略索引 n 和 i,因为本文重点关注流 n 和节点 i。$\mu=\sum_{j\in\Gamma^-(i)}\mu_j=\sum_{j\in\Gamma^-(i)}x_j\mathrm{VRTT}_j(x_j)$,其中 $\mathrm{VRTT}_j(x_j)$表示输出接口 j 的平均响应时间。因此,在每个输出接口需要解决如下问题:$\max\sum_{j\in\Gamma^-(i)}x_j\mathrm{VRTT}_j(x_j)$,约束为$\sum_{j\in\Gamma^-(i)}x_j=y,x_j\geqslant0$。假设:a) $\mathrm{VRTT}_j(x_j)$随着 x_j 单调上升;b) $\mathrm{VRTT}_j(0)=\mathrm{VRTT}_{\min}$;c) $\mathrm{VRTT}_j(x_j)$可微,于是通过施加 KKT 条件便可证明最优分配策略,即 $x_j=\mu_j'^{-1}(\theta)$,θ 是等式$\sum_j\mu_j'^{-1}(\theta)=y$的唯一解。

为了保证连续的接口监测可以实现最优接口选择,需要对节点 i 的每个输出接口的目标作适当修改,即使与负载最高的输出接口相关联的待处理 Interests 数量最小化:$\min\max_j\mu_j(x_j)$,约束为$\sum_jx_j=y,x_j\geqslant0$。目标的实现方法非常简单,因为可以发现最优分配策略确定后对所有的接口 $j\in\Gamma^+(i)$有 $\mu_j=\mu$。假设时间间隔 T 内,如果对部分时间 ΔT_j 有 $T=\sum_j\Delta T_j$,则每个接口 j 已经探查和选择完毕,即 $x_j(t)=y(t)$,否则为 0。因此:

$$\mu=\lim_{T\to\infty}\frac{1}{T}\int_0^T\mu_j(t)\,\mathrm{d}t=\lim_{T\to\infty}\frac{\int_0^Tx_j(t)\,\mathrm{VRTT}_j(x_j(t))\,\mathrm{d}t}{\sum_l\Delta T_l}=\frac{\Delta\overline{T}_j}{\sum_l\Delta\overline{T}_l}y\mathrm{VRTT}_j(y)=\varphi_j\overline{\mu}_j \tag{23}$$

其中:$\overline{\mu}_j$ 表示接口 j 待处理的 Interests 的平均测量值,$\varphi_j=\mu/\overline{\mu}_j$ 表示选择接口 j 的概率,$1/\mu=\sum_j\mu_j$ 表示归一化常数。该算法不需要 VRTT_j 的明确知识,且可探查慢速接口,同时速率 $x_j=\mu_j/\mathrm{VRTT}_j$。算法只需要本地测量使用过的每个接口的 Interests 平均数量,而这一数量是 CCN 网络的可用信息。

请求转发决策基于 FIB 中选择的最长前缀匹配接口。具体步骤见算法 1 所示。每次接收到数据报文时,与已知 FIB 元

素(名称前缀)以及具体的输出接口相关联的待处理 Interests(PI)数量需要进行更新(FIB_PI_Incr/FIB_PI_Decr)。根据 PI 的瞬时值,滑动均值(参数 0.9)定期更新,并用于计算接口的权重(FIB_Weight_Update)。使用随机加权算法为每个入向 Interests(RND_Weight)选择输出接口。对于给定的前缀/输出接口,权重需要相对于与该前缀可用接口相关联的所有权重之和进行正规化处理。在开始时,每个接口的权重相同,均等于 1,可用输出接口的随机转发过程是统一的。

算法 1　请求转发算法

```
At Data_Packet_Reception (name; face_in)
if (! PIT_miss) then
   Forward_Data(face_out);
   FIB_PI_Decr(face_in, prefix);
   FIB_Weight_Update(face_in, prefix);
else Drop_Data;
end if
At Interests_Packet_Reception (name; face_in)
if (CACHE_miss && PIT_miss) then
   (prefix,weight_list) = FIB_Lookup(name);
   f* = RND_Weight(weight_list);
   FWD_I(f*);
   PIT_I_Update(name,f*);
   FIB_PI_Incr(f*, prefix);
   FIB_Weight_Update(f*, prefix);
else CCN_standard_processing;
end if
At PIT_Timeout(name; prefix; face)
   FIB_PI_Decr(face, prefix);
   FIB_Weight_Update(face, prefix);
procedure FIB_WEIGHT_UPDATE((face, prefix))
avg_ PI(face, prefix)←α. avg_PI + (1 − α)I(face,prefix);
w(face, prefix)←1/avg_PI(face,prefix);
end procedure
```

4　实验评估

4.1　在 CCNx 原型上的部署

为 ccnx-0.6.1 版本开发专门的模块来部署上述协议,开发的模块适用于 ccnx-0.6.1-ccnx-0.7.1 间的所有版本。该模块包括一个单机应用,icp_multipath_ccn.{c,h}与拥塞控制功能运行于用户端;ccnd 守护程序中的路径标志机制;请求转发策略在 strategy_callout()中部署。下面对部署情况作简要描述,具体细节不在本文讨论范围内。

1)拥塞控制　根据第 2 章给出的协议与 Interests 调度器的用户端部署特性,Interests 调度器可以确定哪些 Interests 将要插入传输窗口。通过下游数据报文收集并且经过专门的 XML 报文区段 path_label 散列处理过的唯一性节点标志符序列,已经部署了路径标志功能。

2)请求转发　部署 CCNx 要求一种新的数据结构 ccn_pin_record 来跟踪已知命名前缀时待处理 Interests 数量的变化情况。在 CCNx 中,前缀元素存储于 nameprefix_entry(名称组件特里结构(trie))中,默认情况下每个 FIB 元素均实例化一个 ccn_pin_record。

4.2　实验配置

本文实验评估环境是由通用型服务器构成的大规模网格 Grid'5000[12]。基础设置支持启动定制型 Linux 内核镜像(本文测试为 Debian Wheezy 3.2.35-2 x86 64 GNU Linux),包括 4.1 节描述的 CCNx 软件。服务器预留配置和远程内核启动服务支持大规模实验管理。本文测试基于以下四大板块:

a)网络拓扑。使用基于 Tap IP 隧道的虚拟接口作为链路抽象层,使用内核信令桶(kernel token)成形器来创建本文网格中带有特殊链路速率的底层覆盖网络拓扑。

b)CCNx 网络。在 IP 覆盖拓扑顶层构建 CCNx 覆盖网络,且根据事先计算好的 FIB 来配置基于命名的路由,以便客户端可以访问到内容。

c)实验启动。利用可在远程服务器发送 ssh exec 无密码指令来筹划测试过程的集中式应用程序来配置并监控本文实验。通过启动并行客户端的工作负载来运行测试。

d)动态负载。测试中使用的动态负载来自于移动网络 GGSN 采集的数据轨迹。从该测量点提取了某一地区高峰时间的 HTTP 流量,该地区服务于 50 000 名已知用户,同一时刻有 200 名用户处于活跃状态。从该样本中,可以获得被下载的内容量、其受欢迎度及可共享数据量。在 1 h 内测量了 60 GB 的数据,35% 可共享(被请求次数超过一次的字节比例)。在下文中均采用该工作负载模型,鉴于 CCNx 吞吐量约束,采用动态工作负载,降低可用内容量和活跃用户量。

4.3　场景 1:多径流控制的效率

首先讨论简单的多径网络场景(图 1),在该图上,用户通过 4 个非分离路径与 4 个仓库相连。具有唯一性的访问节点 6 中的用户可以检索具有公共前缀的 20 种不同的被命名对象。分开考虑两组链路容量,一组小范围数值,一组大范围数值(情况 1 和 2),以评估不同延时情况下各路由的远程延时控制效率。表 2 给出了两种情况下的平均链路利用率与预期最优利用率间的比较情况。可以看到,链路(0;4)、(1;4)、(5;6)成为瓶颈,并且实现了全面利用。相反,链路(2;5)、(3;5)没有成为瓶颈,请求转发算法几乎以 1:1 的比例均衡了到达节点 5 的请求负载。图 2 给出了在节点 4、5、6 分别相对节点 1、2、4 的分流比 Φ(剩余流量流向节点 0、3、5)。如预期一致,分流比趋近于最优值。然而,客户端节点 6 的收敛时间较晚,原因是与路径节点相比,响应时间较高,降低了待处理 Interests 的更新速度。

表 2　场景 1:平均链路利用率

链路	情况 1 $x_{i,j}$/Mbps 实验值/最优值	情况 2 $x_{i,j}$/Mbps 实验值/最优值
(0,4)	4.74/5	1.79/2
(1,4)	9.18/10	18.35/20
(2,5)	2.44/2.5	2.53/2.5
(3,5)	2.40/2.5	2.46/2.5
(4,6)	13.91/15	20.12/22
(5,6)	4.82/5	4.85/5

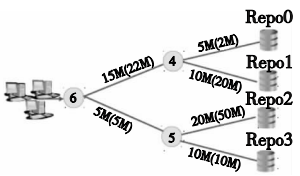

图1　场景1的拓扑结构1(情况2)

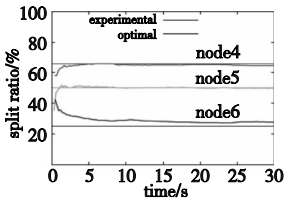

图2　分流率收敛(场景1)

4.4　场景 2:网络内缓存的影响

下面考虑一种典型的 CND 网络场景以评估网络内缓存对本文拥塞控制和转发机制的影响。拓扑结构由 15 个节点和 11 条路由构成(图 3)。每个仓库存储整个内容目录,且目录

由 1 000 个对象构成,分配到尺寸为 4 KB 的 1 000 个报文中。来自叶节点(用户)的对象请求到达规律服从速度为 0.1 对象/s 的泊松过程以及外观(shape)为 0.35、比例(scale)为 1 000 的 Weibull 受欢迎度分布模型。叶节点(即 7 ~ 10 节点)的缓存尺寸固定为 100 MB,而 2 ~ 6 号节点在不同测试下的缓存尺寸为 200 MB ~ 2.8 GB。每个实验持续 2 h(每个实验约 3 000 个对象请求),运行 5 次取均值作为输出。表 3 给出了不同缓存尺寸时的分流率和无缓存时的预期最优值。图 4 给出了增加缓存尺寸时,链路(2,5)和(3,5)的服务器负载下降情况及总体带宽节约情况。带宽节约量定义为 $\frac{\sum_{i,j} x_{i,j}(S=0) - x_{i,j}(S)}{\sum_{i,j} C_{i,j}}$。其中 S 表示系统的总体缓存尺寸,单位为 pkts;$x_{i,j}(S=0)$,$x_{i,j}(S)$分别表示没有缓存和存在缓存时的链路率。当缓存尺寸上升时,带宽节约量上升,但是仓库的负载相应下降。尽管如此,10 GB 缓存就能处理 50% 的请求数据。

表 3　分流比

链路	最优值		实验值	
	0 缓存	0	6.4 GB	14.4 GB
(4,2)	33.33	36.13	37.76	38.11
(5,2)	66.66	63.21	59.58	57.65
(6,2)	50	49.25	49.63	49.53
(7,4)	50	51.93	55.25	56.99
(8,4)	50	47.61	47.79	47.74
(9,5)	50	50.51	56.30	57.56
(10,5)	66.6	65.61	71.32	73.87

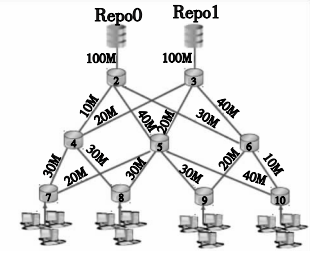

图3　场景2的拓扑结构

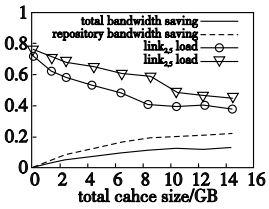

图4　带宽/存储性能折中(场景2)

4.5　场景 3:请求转发的伸缩性

本节在骨干网络拓扑结构(图 5 中的 Abilene 结构)下研究本文协议的性能。在节点 12、16、19 处有三个仓库,每个仓库在给定名称前缀下存储内容(ccnx:/amazon. com, ccnx:/google. com,ccnx:/warner. com)。每个种类的特点与前一种场景相同。用户请求规律服从不同速率的泊松过程:#11,#13,#14,#15,#17,#18,#20,#21 节点的速率分别为每 100 s 有 45,48,36,42,30,24,30,33 个对象。为了保持网络数据传输次数稳定,选择速率时要保证总体负载供应量低于网络服务容量。

1)单径和多径的比较　因为使用多径总是可以提升总体性能,所以给出不同名称前缀下的用户性能。图 6 显示了单径和多径两种情况下,每个名称前缀和每次用户访问节点时的平均传递时间。多径方案显著降低了传递时间,但是节点 14 和 15 的延时较长,原因是流数量变大,而且利用了可以公平共享可用带宽的链路(3,4)和(4,5)。同时,基于多径的总体增益很大。

2)对象名称 VS 前缀请求转发　采用单名称前缀(ccnx:/google),单仓库(16 号仓库,存储 1 000 个尺寸为 4 KB 1 000 pkts 的对象),单访问节点(3 号节点),双向链路容量如下:链路(3,4),(4,5) = 25 Mbps,(5,7) = 5 Mbps,(7,8),(8,3) = 50 Mbps。在节点 7 和 8 处插入两个缓存,尺寸为 100 MB(25 个对象)。节点 7 的请求到达规律服从速率为 1 个对象/s 的泊松过程和参数为 $\alpha = 1.3$ 的 zipf 受欢迎度模型。在该设置下,最受欢迎的对象(排名较小)可从节点 8 的缓存中获取,获取速度是到达仓库(#3,#4,#5)的另一条路径的两倍。运行两组实验:一组是每个前缀转发,另一组是每个对象专门带有含 30 个元素的 FIB 状态。如图 7 所示,最受欢迎对象的性能略高,更重要的是,它们为大多数不受欢迎的对象释放了上游带宽,显著提高了不受欢迎对象的性能。通过采取混合式策略,最受欢迎的项目(通过 PIT 内聚合 Interests 的数量加以识别)在 FIB 内单独监测和处理,同时保存其他前缀下的其他项目,即可实现很好的性能折中。

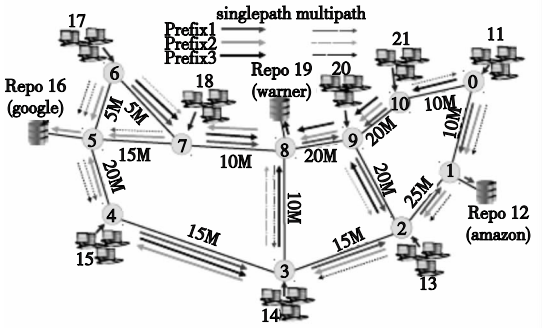

图5　场景3的拓扑结构

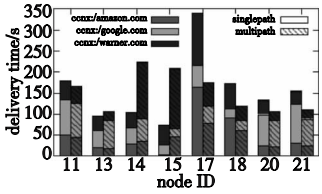

图6　场景3中每个节点的平均传递时间

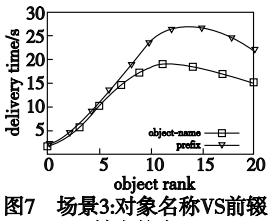

图7　场景3:对象名称VS前缀转发粒度

5　结束语

本文研究了 ICN 下联合多径拥塞控制和请求转发问题,设计了可伸缩、完全分布式互通机制,以实现高效的单点对多点内容传递。最后的仿真实验也验证了本文研究的有效性。下一步将针对高速 ICN 数据层,研究基于名称的转发引擎。同时,对 PIT/FIB 的动态性展开理论分析,提出可行的预测模型以预测它们的动态性。

参考文献:

[1]　夏春明,徐明伟. 信息中心网络研究综述[J]. 计算机科学与探索,2013,7(6):601-605.

[2]　Koponen T, Shenker S, Balakrishnan H, *et al*. Architecting for innovation[J]. ACM SIGCOMM Computer Communication Review, 2011,41(3):24-36.

[3]　Nordström E, Shue D, Gopalan P, *et al*. Serval: an end-host stack for service-centric networking[[C]//Proc of the 9th USENIX Conference on Networked Systems Design and Implementation. Berkeley : USENIX Association,2012:85-98.

[4]　夏春梅,徐明伟. 面向服务和信息的网络体系结构——SIONA[J]. 计算机学报,2014,37(2):289-300.

[5]　张子文,吕高锋,孙志刚,等. 面向 InfiniBand 数据中心的区别化传输层带宽划分机制[J]. 计算机学报,2013,35(12):2505-2514.

(下转第　　页)

(上接第　　页)

[6] Han H, Shakkottai S, Hollot C V, *et al*. Multi-path TCP: a joint congestion control and routing scheme to exploit path diversity in the Internet[J]. IEEE/ACM Trans on Networking, 2006, 14(6): 1260-1271.

[7] Wischik D, Raiciu C, Greenhalgh A, *et al*. Design, implementation and evaluation of congestion control for multipath TCP[C]//Proc of the 8th USENIX Conference on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2011: 99-112.

[8] Yi Cheng, Afanasyev A, Wang Lan, *et al*. Adaptive forwarding in named data networking[J]. ACM SIGCOMM Computer Communication Review, 2012, 42(3): 62-67.

[9] Jacobson V, Smetters D K, Thornton J D, *et al*. Networking named content[C]//Proc of the 5th International Conference on Emerging Networking Experiments and Technologies. [S. l.]: ACM Press, 2009: 1-12.

[10] De Cicco L, Mascolo S. A mathematical model of the Skype VoIP congestion control algorithm[J]. IEEE Trans on Automatic Control, 2010, 55(3): 790-795.

[11] Asanovic K, Bodik R, Demmel J, *et al*. A view of the parallel computing landscape[J]. Communications of the ACM, 2009, 52(10): 56-67.

[12] Carofiglio G, Gallo M, Muscariello L, *et al*. Multipath congestion control in content-centric networks[C]//Proc of IEEE NOMEN. 2013: 13-21.