**文章编号**:1007-5321(2017)03-0097-07 **DOI**:10.13190/j.jbupt.2017.03.014

# 新型命名数据网络校验机制设计

朱 轶, 康浩浩, 黄茹辉, 曹清华
(江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

**摘要**: 提出了一种基于流行度的概率存入校验机制(PCS-CP),根据接收内容的流行度,概率抽取内容校验,并只存入校验通过内容,确保节点的有限计算资源尽可能服务于用户关注内容,无须校验网内命中内容,降低了校验开销. 由于 PCS-CP 机制只有与特定缓存策略配合才能最大化效用,进而提出了一种基于缓存更新时间的网内缓存策略,对网内副本冗余进行优化控制,有效提升了 PCS-CP 机制的校验效果. 数值结果表明,与命中校验机制相比,PCS-CP 可有效降低网内校验次数,有效防御内容污染攻击.

**关 键 词**: 命名数据网络; 内容污染; 内容校验; 校验开销
**中图分类号**: TN915 **文献标志码**: A

# Content Checking Mechanism Design in Named Data Networking

ZHU Yi, KANG Hao-hao, HUANG Ru-hui, CAO Qing-hua
(School of Computer Science and Communication Engineering, Jiangsu University, Jiangsu Zhenjiang 212013, China)

**Abstract**: This article proposed a checking mechanism, named as probabilistic checking before storing based on content popularity (PCS-CP). The design of PCS-CP includes two aspects, one is the node which should check the received content randomly according to its popularity, and the other is the node only stores the legitimate content. On the one hand, PCS-CP can ensure the limited computing resource of node to serve the high popularity content as much as possible. On the other hand, PCS-CP guarantees the authenticity of cached content, and then reduces the computation overhead of node. Because only co-operating with certain caching policy, PCS-CP can maximize its effectiveness. The in-network caching strategy based on cache update time (ICS-CUT) was further proposed to optimize the copy redundancy in network. It is shown that, comparing with checking on hit mechanism, PCS-CP can effectively reduce the average amount of checking in network and well defense the content pollution attack.

**Key words**: name data networking; content pollution; content checking; checking overhead

命名数据网络(NDN, named data networking)[1]是未来互联网架构的典型代表. NDN 通过发布者对内容签名,用户(或节点)校验签名的方式保证内容完整性和安全性,提供一种基于内容本身的安全机制. 理论上说,这种安全机制能有效保证网内数据安全,但实际部署中,由于节点计算资源有限,无法对所有接收内容进行签名验证,所以节点不强制内容签名的验证,因此无法保证网内缓存内容的安全性,导致内容污染成为 NDN 的严重安全隐患[2-4]. 因此如何有效进行内容校验、降低内容污染对 NDN 性能的影响,成为当前 NDN 研究的关键问题之一.

**收稿日期**: 2016-09-30
**基金项目**: 国家自然科学基金项目(41474095)
**作者简介**: 朱 轶(1977—), 男, 副教授, 硕士生导师, E-mail: zhuyi@ujs.edu.cn.

目前,对 NDN 内容校验机制的改进主要有 2 种方案:1)概率校验[5],通过降低缓存概率来降低内容校验计算量. 这一思路虽然可取,但 Bianchi 等[5]未作具体设计,未深入考虑概率存入的内容校验量与节点处理能力关系. 2)命中校验(COH,check on hit)[6-7],内容存入缓存时不校验,仅当出现缓存命中事件时才对命中内容校验. 该方案虽然降低了计算消耗,但是内容存入缓存时未经过校验,缓存中存在污染内容.

合理选择转发路由是抑制内容污染攻击的另一种思路,DiBenedetto 等[8]提出通过检测可疑内容来源,尽量把兴趣包转发至真实内容来源. 不过该方案仍需与有效的校验机制联合实施,才保证网内传输内容的安全性.

鉴于有效的内容校验是保证 NDN 可靠的关键所在,设计了一种基于流行度的概率存入校验机制(PCS-CP,probabilistic checking before storing based on content popularity),通过按流行度概率抽取内容校验并只存入校验通过内容,构建可信网络,在保障网络可靠性的同时,降低校验开销. 考虑到 PCS-CP 机制只有与特定缓存策略配合,才能最大化工作效率,从控制网内副本冗余的角度[9],进而建议了一种基于缓存更新时间的网内缓存策略(ICS-CUT,in-network caching strategy based on cache update time),最大化 NDN 的网络存储效率,有效配合 PCS-CP 机制的工作.

## 1 NDN 内容签名与校验原理

NDN 中的内容签名与校验原理如图 1 所示. 内容提供者接收到请求兴趣包后,将对所请求内容以 Hash 函数计算出内容的哈希值,进而以 RSA(Rivest-Shamir-Adleman)加密算法对生成的信息哈希值进行加密运算,计算出内容签名值,完成内容的签名过程. 当带有签名的数据包返回到请求节点时,节点首先向内容提供者请求公钥,这里公钥获取过程也等同于一次内容请求,之后用 RSA 算法对数据包中的签名值进行解密,还原出信息哈希值,并与本地用哈希函数对数据包内容计算出的哈希值进行对比,完成签名验证过程. 如果验证通过,则进一步执行缓存存入以及转发操作;如果验证失败,则直接丢弃数据包.

目前 RSA 算法是 NDN 中常用的内容签名与验证算法,这种非对称的加密算法对内容的保护能力

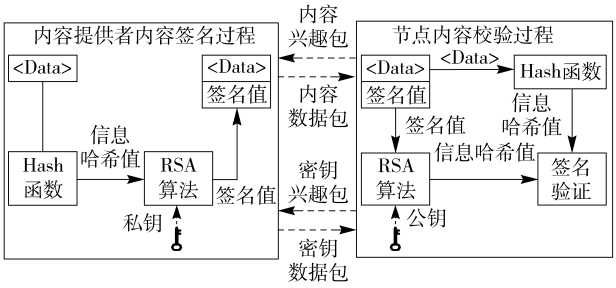



图 1 NDN 内容签名与校验原理图

依赖密钥长度,密钥越长则安全性越强,但带来的负面效应就是签名验证时需要消耗大量的计算资源. 由于路由节点的计算资源有限,当接收数据流量较大时,节点无法完成所有接收数据包的校验,导致现有 NDN 签名校验机制形同虚设,因此对校验机制进行改善,设计一种可行的 NDN 校验机制迫在眉睫.

## 2 PCS-CP 校验机制

### 2.1 条件设定

后续机制描述以及理论分析将基于以下条件设定:1)参考文献[10]的网络拓扑,笔者采用 $N$ 层级联 NDN 网络;2)节点的极限校验计算处理能力为 $C_P$(数据包/秒);3)源服务器提供 $M$ 个不同的内容,根据内容流行度均匀划分为 $K$ 个不同类别,即每一类包含 $m = M/K$ 个内容文件,每个内容的大小相同;4)用户请求以泊松分布到达[10],$\lambda_i$ 为第 $i$ 层节点的请求到达率,$\lambda_k^i$ 为第 $i$ 层节点第 $k$ 类的请求到达率(第 $i-1$ 层第 $k$ 类未命中的请求),即 $\lambda_k^i = \lambda_k^1 \prod_{j=1}^{i-1} [1 - P^j(k)]$,$P^j(k)$ 为第 $j$ 层节点第 $k$ 类的命中率;5)第 1 层接入的第 $k$ 类内容请求概率为 $q_k^1$,服从 Zipf 分布,$q_k^1 = c/k^\alpha$,$c > 0$,这里 $\alpha$ 代表了流行度分布的集中程度,$\alpha$ 越大,内容请求越集中于 $k$ 较小的内容;6)$\tau_i$ 为第 $i$ 层节点的缓存更新时间;7)节点缓存大小相同,均为 $C$ 个内容文件;8)源服务器存在内容污染,$r_T$ 为源服务器真实内容比例,虚假内容均匀分布于每一类别;9)分析中忽略节点间的传输时延.

### 2.2 校验机制描述

结合 Bianchi 等[5,7]所给出的现有校验机制,NDN 校验问题中同样存在有效性与可靠性的矛盾,校验量与网络污染内容比例(校验结果)之间互相矛盾,而校验机制的设计目标正是致力于约束网络污染内容的同时,降低校验量. 目前 NDN 节点校验

开销过高，其根本原因为网内节点之间的不信任，从而导致节点之间出现重复校验，浪费了大量的计算资源. 如果能构建一个可信 NDN 网络，并尽可能最大化网络的存储效率，必然可以显著提升校验机制的运行效果，这就是本研究设计出发点. 所提出 PCS-CP 机制的具体设计如下：对于网内第 $i$ 层节点，当接收到来自源服务器的数据包时，根据该内容所属流行度进行概率抽取校验，即对于接收到的来自源服务器的第 $k$ 类内容，以概率 $P_{\text{verf}}(k,i)$ 进行校验.

$$P_{\text{verf}}(k,i)=\frac{c}{k^{\alpha}}\times\min\left\{\frac{C_{\text{p}}}{\sum_{k=1}^{K}\lambda_k^i\left[1-P_{\text{PCS-CP}}^i(k)\right]},1\right\} \tag{1}$$

其中：$C_{\text{p}}$ 为节点的最大校验能力，$P_{\text{PCS-CP}}^i(k)$ 为第 $i$ 层节点第 $k$ 类真实内容命中率，显然 $\lambda_k^i[1-P_{\text{PCS-CP}}^i(k)]$ 代表第 $i$ 层节点第 $k$ 类内容最大可能出现的返回数据流量. 这一校验概率的设置确保节点将以大概率对于重要内容（高流行度内容）进行校验，放弃校验大部分用户关注度低的内容. 为了构建节点之间的可信关系，本机制实施存入校验规则：仅有校验通过的数据包才能存入节点的内容存储（CS，content store），而未被校验的内容将不予存储，同时校验失败的数据包将被丢弃. 这一规则确保了网内每个节点的缓存中存储的都是可信内容，因而节点之间可以相互信任，这就意味着对于网内命中的内容，节点无须校验；而仅有从源服务器命中的内容，节点才需要校验.

### 2.3　配合校验机制的缓存策略设计

结合可信网络的设计，为了进一步降低节点校验压力，必须有效减少网内存储副本的冗余，尽可能扩大 NDN 的网内存储效率，提高网内命中率. 因此，PCS-CP 机制只有配合特定的网内缓存（Innetwork caching）策略，才能发挥应有的作用. 从降低最小化网内副本冗余的角度出发，建议了 ICS-CUT，该策略优点在于不仅可约束网内副本，同时平衡每个节点的数据包存入压力（数据包校验压力）.

ICS-CUT 策略要求网内每个节点定期统计自身的缓存更新时间，这里缓存更新时间定义为：内容在缓存队列中的平均驻留时间. 以最近最少使用策略（LRU，least recently used）为例，该策略下的缓存更新时间即为某一内容从缓存队列首部移动到缓存队列尾部所需的平均时间. 由 Wang 等[11]的分析可知：在实际网络中，节点到达的请求流量越多，需要存入缓存的数据也越多，导致缓存中的内容更替频繁，节点缓存更新时间变快. 因此，缓存更新时间越长，代表该节点存入的数据流量压力越小. 在此基础上，用户发出的请求兴趣包新增一个字段，用于记录经过节点的缓存更新时间. 当出现不命中事件时，节点就在兴趣包的该字段中添加自身的缓存更新时间，再转发出去. 基于传输路径上所有缓存更新时间的记录信息，若兴趣包在源服务器命中，回传的数据包将存入反向路径上具有最大缓存更新时间的节点；若兴趣包在网内某一节点命中，回传的数据包将移动至反向路径上具有最大缓存更新时间的节点（从命中节点删除）.

ICS-CUT 策略保证了任意内容，在网内只会存在一个副本，从而最大化了 NDN 的网络存储效率；此外，由于缓存更新时间代表了缓存的数据包存入压力，因此动态选择具有最大缓存更新时间的节点作为存入目标，有助于平衡回传路径上各个节点的存入数据量，便于充分利用每个节点的计算能力进行数据校验.

需要说明的是，设计的 ICS-CUT 策略，用于配合 PCS-CP 校验机制，提升校验工作的有效性. 但是配合 PCS-CP 校验机制的缓存策略设计，不是唯一的，存在多种设计方案，只要能有效控制网内副本冗余、平衡缓存存入压力的缓存策略，都可以配合 PCS-CP 机制实施.

## 3　校验机制性能分析

本节针对 $N$ 层级联的 NDN 网络，选择网络真实内容命中率和网内平均校验次数作为校验机制性能评估指标，网络拓扑如图 2 所示. 为了对比说明 PCS-CP 机制的性能，选择基于 LRU 策略的 COH 机制作为对比策略，对 2 个校验机制同时给出理论分析结果，这里定义 $T_{\text{PCS-CP}}$ 为 PCS-CP 机制下网内平均校验次数，$T_{\text{COH}}$ 为 COH 机制下网内平均校验次数，$P_{\text{PCS-CP}}^i(k)$ 为 PCS-CP 机制下第 $i$ 层节点第 $k$ 类真实内容命中率，$P_{\text{COH-T}}^i(k)$ 为 COH 机制下第 $i$ 层节点第 $k$ 类真实内容命中率.

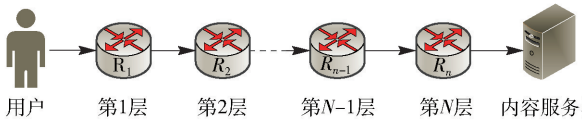



图 2　$N$ 层 NDN 级联拓扑图

### 3.1 真实命中率分析

3.1.1 PCS-CP 机制命中率

由 PCS-CP 及 ICS-CUT 策略设计可见,该联合设计等效于概率缓存机制. 考虑到当网络处于稳态时,由 ICS-CUT 策略,该级联路径上每个节点将近似等概被选择;进而选中节点按概率 $P_{\text{verf}}(k,i)$ 选择第 $k$ 类内容进行校验,并将校验结果为真的内容存入缓存. 若设 $\beta_k^i$ 为第 $k$ 类内容在第 $i$ 层节点的缓存存入概率,则 $\beta_k^i$ 至少应满足

$$\beta_k^i \geqslant \frac{r_{\text{T}} P_{\text{verf}}(k,i)}{N} \tag{2}$$

其中 $r_{\text{T}}$ 为源服务器真实内容比例. 式(2)表明,由于 PCS-CP 机制下,网内节点均可信,仅校验在源服务器命中的内容,因此源服务器的真实内容比例即为网内节点接收数据流中最小可能的真实内容比例(经过上游节点概率校验后,转发至下游节点的真实内容比例会提高),则存入概率至少为节点选择概率、校验概率、源服务器真实内容比例三者的乘积.

由文献[12]可知,对于概率缓存机制,以相继出现的 2 次请求为分析对象,命中事件的出现有 2 种可能:

1) 目标内容已存在于缓存中,第 1 次请求命中,2 次请求的间隔小于缓存更新时间 $\tau_i$,则第 2 次请求命中;

2) 目标内容不存在于缓存中,第 1 次请求不命中,内容以概率 $\beta_k^i$ 存入缓存,2 次请求的间隔小于缓存更新时间 $\tau_i$,则第 2 次请求命中.

由于节点处于稳态时,相继出现请求的命中概率视为相同,均为 $P_{\text{PCS-CP}}^i(k)$,且设 $\tau_k^i$ 为第 $i$ 层节点第 $k$ 类内容的请求间隔

$$\begin{aligned} &P_{\text{PCS-CP}}^i(k) P\{\tau_k^i \leqslant \tau_i\} + \\ &(1 - P_{\text{PCS-CP}}^i(k)) \beta_i^k P\{\tau_k^i \leqslant \tau_i\} = \\ &P_{\text{PCS-CP}}^i(k) \end{aligned} \tag{3}$$

这里,$\lambda_k^i$ 为第 $i$ 层节点第 $k$ 类的请求到达率,服从泊松分布,则式(3)可化简为

$$P_{\text{PCS-CP}}^i(k) = \frac{\beta_k^i (1 - \mathrm{e}^{\frac{\lambda_k^i}{m}\tau_i})}{1 + (\beta_k^i - 1)(1 - \mathrm{e}^{\frac{\lambda_k^i}{m}\tau_i})} \tag{4}$$

PCS-CP 机制下,网内命中内容均为真实内容,因此 $P_{\text{PCS-CP}}^i(k)$ 即为 PCS-CP 机制下的网内真实内容的命中率. 在上述命中率分析中,缓存更新时间 $\tau_i$ 是一个重要指标,参考文献[12],$\tau_i$ 应表示为

$$\begin{aligned} &m\sigma \sum_{k=1}^{K} [P\{\tau_k^i \leqslant \tau_i\} (\beta_k^i - \beta_k^i P_{\text{PCS-CP}}^i(k) + \\ &P_{\text{PCS-CP}}^i(k))] = C \end{aligned} \tag{5}$$

其中:$m$ 为每一类的文件个数,$\sigma$ 为每个文件的大小. 式(5)表明,在 $\tau_i$ 时间内,若插入缓存队列首部的所有类别内容数量总和,恰好等于缓存队列长度 $C$,则原本处于缓存队列首部的内容将会移至队列尾部,因此 $\tau_i$ 即为任意内容在缓存中的平均驻留时间,即缓存更新时间.

3.1.2 COH 真实内容命中率

COH 机制未对缓存策略做任何调整,接收到的内容直接存入缓存,仅在命中时加以校验,若选择 LRU 策略作为置换策略,由文献[7],第 $i$ 层节点的第 $k$ 类内容命中率应如式(6).

$$P_{\text{COH}}^i(k) = P\{\tau_k^i \leqslant \tau_i\} = \sum_{i=k}^{K} (1 - \mathrm{e}^{\frac{\lambda_k^i}{m}\tau_i}) \tag{6}$$

由于 COH 校验机制下,网内节点缓存的内容不完全可靠,则真实命中率 $P_{\text{COH-T}}^i(k)$ 应近似等于实际内容命中率 $P_{\text{COH}}^i(k)$ 与源服务器真实内容比例 $r_{\text{T}}$ 的乘积.

$$P_{\text{COH}}^i(k) = P_{\text{COH}}^i(k) r_{\text{T}} \tag{7}$$

### 3.2 网内平均校验次数分析

3.2.1 PCS-CP 平均校验次数

PCS-CP 机制下网内节点存储内容均为可靠内容,网内命中无须校验,只有源服务器命中的内容需要校验,因此 PCS-CP 机制下第 $k$ 类内容的网内总校验量仅取决于该类别内容的源服务器的内容命中率、单位时间内网络接收到的该类别内容到达请求次数,即边缘接入的请求到达率以及节点的校验概率. 现定义 $T_{\text{pcs-cp}}(k)$ 为 PCS-CP 机制下第 $k$ 类内容的网内平均校验次数,则

$$\begin{aligned} &T_{\text{PCS-CP}}(k) = \\ &\lambda_k^1 \left[1 - \sum_{i=1}^{N} P_{\text{PCS-CP}}^i(k)\right] \frac{1}{N} \sum_{i=1}^{N} P_{\text{verf}}(k,i) \end{aligned} \tag{8}$$

其中:$1 - \sum_{i=1}^{N} P_{\text{PCS-CP}}^i(k)$ 为第 $k$ 类内容的源服务器命中率,$\sum_{i=1}^{N} P_{\text{verf}}(k,i)/N$ 为第 $k$ 类内容的网内平均校验概率(由提出机制,源服务器命中内容在网内仅存储一个副本,仅需校验一次).

显然,PCS-CP 机制下所有类别的网内平均校验次数应为

$$T_{\text{PCS-CP}} = \sum_{k=1}^{K} \left[\frac{c}{k^{\alpha}} T_{\text{PCS-CP}}(k)\right] \tag{9}$$

其中 $c$ 为 Zipf 分布的约束参数，$c=\sum_{k=1}^{K} k^{-\alpha}$.

3.2.2　COH 平均校验次数

对于 COH 校验机制，节点仅在命中事件发生时进行校验，全网校验次数即为各节点校验次数之和. 对于第 $i$ 层节点，其校验次数受路由自身的处理能力影响：当命中次数小于或等于处理能力时，命中内容将被全部校验；当命中次数大于处理能力时，节点尽力校验，校验量最大为处理能力 $C_{\mathrm{P}}$. 因此，COH 机制下所有类别的网内平均校验次数 $T_{\mathrm{COH}}$ 应为

$$T_{\mathrm{COH}}=\sum_{i=1}^{N}\left\{\min\left[C_{\mathrm{P}},\sum_{k=1}^{K}\lambda_k^i P_{\mathrm{COH}}^i(k)\right]\right\} \quad (10)$$

其中 $\sum_{k=1}^{K}\lambda_k^i P_{\mathrm{COH}}^i(k)$ 为所有类别内容请求在第 $i$ 层节点的平均命中次数.

## 4　数值计算与分析

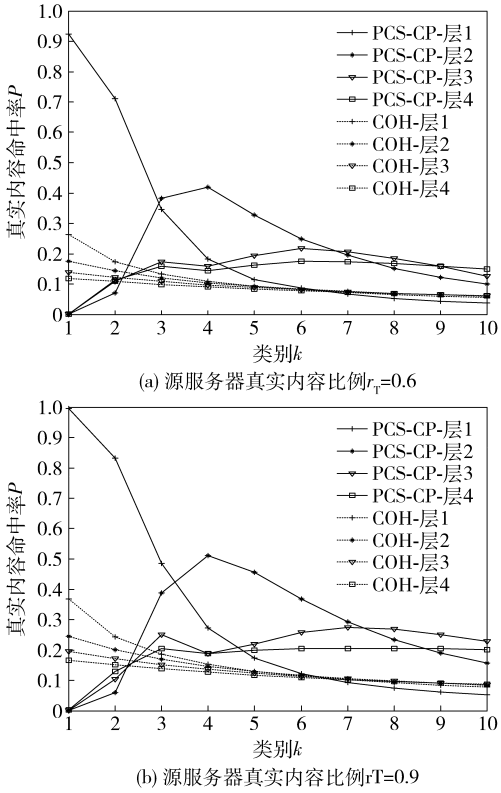
利用 Matlab 对 PCS-CP 与 COH 校验机制下的真实内容命中率和网内平均校验次数分别进行了数值计算与分析，数值计算结果根据式(4)、式(7)、式(9)、式(10)得出，分析中设定源服务器存在一定比例的内容污染.

参考文献[10]，分析参数设置如下：网络拓扑采用 $N=4$ 层级联的结构，网络提供内容根据流行度分为 $K=40$ 个类别，每一类包含 $m=200$ 个内容文件，网络接入内容请求服从参数为 $\lambda_1=10^5$ 文件/$s$ 的泊松分布，选取 Zipf 分布参数值 $\alpha=1.2$，节点每秒处理能力 $C_{\mathrm{P}}=1.25\times10^4$ 个数据包(路由器条件[5]：Intel Core 2 Duo 2.53 GHz 处理器，每个文件大小为 1.5 KB、RSA 加密算法).

### 4.1　源服务器污染程度影响

现分析 2 种校验机制在不同源服务器真实内容比例 $r_{\mathrm{T}}$ 下的性能，计算中设定缓存大小与网络内容总量的比值($C/M$)固定为 0.1，真实内容比例 $r_{\mathrm{T}}$ 改变，图 3、图 4 分别给出了 $r_{\mathrm{T}}$ 对节点真实内容命中率和平均校验次数的影响.

如图 3 所示，当源服务器真实内容比例 $r_{\mathrm{T}}$ 从 0.9 降低至 0.6，PCS-CP 与 COH 机制下的命中率均有所下降，但 PCS-CP 机制的网内总体命中率较高. PCS-CP 机制下，各层节点的真实内容命中率明显优于 COH 机制. 即使源服务器存在 40% 的虚假内容，PCS-CP 机制下第 10 类内容的 4 层真实内容命中率也分别为 0.06、0.15、0.17、0.20，累计 58%，而 COH



(a) 源服务器真实内容比例$r_{\mathrm{T}}$=0.6

(b) 源服务器真实内容比例rT=0.9

图 3　源污染对真实内容命中率的影响($C/M=0.1$)

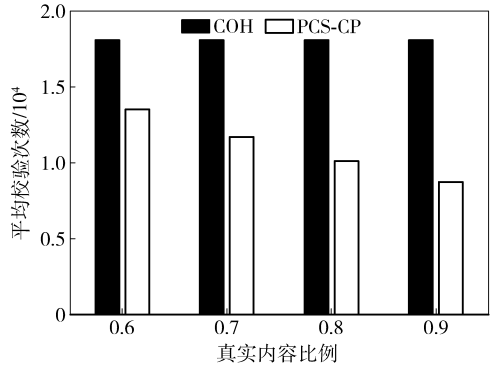


图 4　源污染平均校验次数的影响($C/M=0.1$)

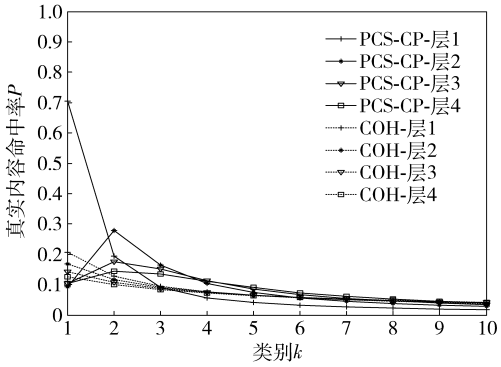
机制的第 10 类内容网内真实内容命中率不到 40%. PCS-CP 机制对内容污染的防御优势源自于可信网络的设计，由于网内节点全部无污染，网内命中内容均为真实；此外，通过采用基于流行度的概率校验与 ICS-CUT 缓存策略，网内不存在副本冗余且大概率存储高流行度内容，这些设计也保障了高流行度内容的网内缓存比例与命中率. 而 COH 机制下节点缓存存在较严重内容污染，仅通过命中后再校验出虚假内容，对内容污染基本没有效防御.

图 4 给出了源服务器污染下，PCS-CP 机制与 COH 机制的校验负载. 由图可见，当源服务器污染

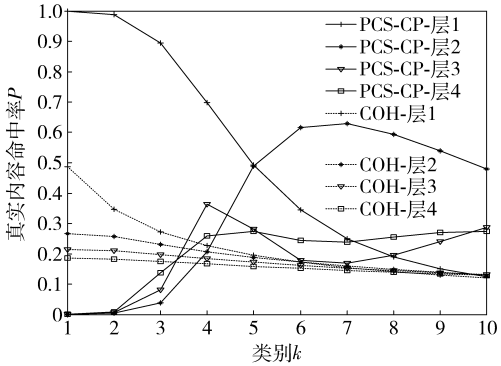
程度提升,COH 机制平均校验次数基本不变,这是因为 COH 机制的校验次数仅取决于命中率(不是真实内容命中率),而命中率并不受虚假内容影响. PCS-CP 的平均校验次数随着内容污染程度加深有所增加,但明显低于 COH 机制. 这一结果也表明,相比较 COH 机制,PCS-CP 机制有效控制了校验计算量,更易于实施,尤其在网络相对良好时,仅需要较小的校验计算开销就可以维持优质的网络内容安全状态. 如 $r_T = 0.9$ 时,PCS-CP 机制的计算开销仅为 COH 机制的一半,但是命中率超过 COH 机制的一倍.

### 4.2 *C/M* 的影响

*C/M* 本质上指向缓存大小,若固定网络内容总量不变,则 *C/M* 增大(减小)代表缓存容量增大(减小),而由于缓存容量直接影响节点命中率,因此对校验计算量及校验效果产生重要影响. 计算中固定源服务器真实内容比例 $r_T$ 为 0.7. 图 5 给出了缓存大小变化对命中率的影响,如图计算结果,当 *C/M* 从 0.15 降至 0.05,2 种校验机制下真实内容的命中率均明显下降,这是由于缓存空间减小,NDN 网内可存储副本降低. 但 PCS-CP 机制相比较 COH 机制依然维持明显优势. 图 6 给出了缓存大小变化对校验负载的影响,随着 *C/M* 从 0.05 增至 0.2,COH 机制由于网内命中率提高,其校验计算开销越来越大;而 PCS-CP 机制由于网内可容纳的副本越来越多,源服务器命中率降低,需要校验的内容越来越少,平均校验次数变得更低,这一趋势也充分说明了 PCS-CP 机制对校验开销的有效控制.




(a) 缓存大小与网络内容总量之比*C/M*=0.05




(b) 缓存大小与网络内容总量之比*C/M*=0.15

图 5 缓存大小对真实内容命中率的影响($r_T = 0.7$)

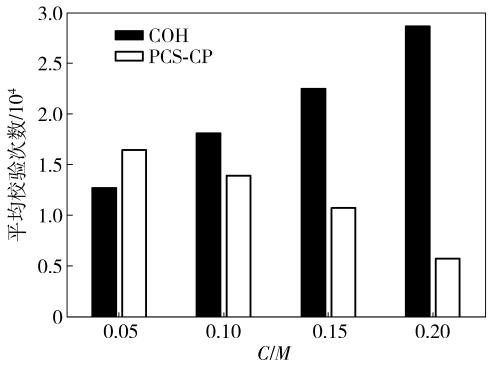



图 6 缓存大小对平均校验次数的影响($r_T = 0.7$)

## 5 结束语

如何设计有效的校验机制,解决计算开销问题,降低内容污染对 NDN 网络性能影响,保证网络内容的完整性与安全性,已经成为当前 NDN 研究的重点. 未来的研究中,将进一步探讨如何改进校验算法,从算法设计本身去提高内容校验的可行性,同时关注如何改进 NDN 节点间的信任关系,设计更为健壮的可信网络,从信任角度抑制内容污染的影响.

**参考文献:**

[1] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content[C]// ACM Conference on Emerging Networking Experiments and Technology, CONEXT 2009. Rome: ACM, 2012: 117-124.

[2] Lauinger T. Security & scalability of content-centric networking [D]. Darmstadt: TU Darmstadt, 2010.

[3] Ribeiro I, Rocha A, Albuquerque C, et al. On the possibility of mitigating content pollution in content-centric networking[C]// 2014 IEEE 39th Conference on Local Computer Networks. Edmonton: IEEE, 2014: 498-501.

[4] Gasti P, Tsudik G, Uzun E, et al. DoS and DDoS in named data networking[C]// 2013 22nd International Conference on Computer Communication and Networks (ICCCN). Nassau: IEEE, 2013: 1-7.

[5] Bianchi G, Detti A, Caponi A, et al. Check before storing: what is the performance price of content integrity verification in LRU caching [J]. ACM SIGCOMM Com-

puter Communication Review, 2013, 43(3): 59-67.
[6] Nam S W, Kim D, Yeom I. Content verification in named data networking[C]//2015International Conference on Information Networking. Cambodia: IEEE, 2015: 414-415.
[7] Kim D, Nam S, Bi J, et al. Efficient content verification in named data networking[C]//Proceedings of the 2nd ACM Conference on Information-Centric Networking. New York: ACM, 2015: 109-116.
[8] DiBenedetto S, Papadopoulos C. Mitigating poisoned content with forwarding strategy[C]//Computer Communications Workshops (INFOCOM WKSHPS). San Francisco: IEEE, 2016: 164-169.
[9] Li Yun, Zhao Ling, Liu Zhanjun, et al. N-Drop: congestion control strategy under epidemic routing in DTN [C]//International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. Leipzig: ACM, 2009: 457-460.
[10] Wang Guoqing, Huang Tao, Liu Jiang, et al. Modeling in-network caching and bandwidth sharing performance in information-centric networking[J]. The Journal of China Universities of Posts and Telecommunications, 2013, 20(2): 99-105.
[11] 崔现东, 刘江, 黄韬, 等. 基于节点介数和替换率的内容中心网络网内缓存策略[J]. 电子与信息学报, 2014, 36(1): 1-7.
[12] Zhu Yi, Mi Zhengkun, Wang Wennai. A probability caching decision policy with evicted copy up in content centric networking [J]. Journal of Internet Technology, 2017, 18(1): 33-43.

(上接第 96 页)

[10] Park Y, Bahn H. Management of virtual memory systems under high performance PCM-based swap devices [C]//COMPSAC 2015. Taichung: IEEE Press, 2015: 764-772.
[11] Anirudh B, Vivek S P. SSDAlloc: hybrid SSD/RAM memory management made easy[C]//2011 Conference on Networked Systems Design and Implementation. Boston: ACM Press, 2011: 211-224.
[12] Anirudh B, Vivek S P. Better flash access via shapeshifting virtual memory pages [C]//SIGOPS 2013. Farmington: ACM Press, 2013: 1-14.
[13] Jian Xu, Steben S. NOVA: a log-structured file system for hybrid volatile/non-volatile main memories[C]//2016 USENIX Conference on File and Storage Technologies. Santa Clara: USENIX Press, 2016: 323-338.
[14] Mendel R, John K O. The design and implementation of a log-structured file system[J]. ACM Transactions on Computer Systems, 1992, 10(1): 26-52.