

命名数据网络中基于 Dec-POMDP 的缓存策略

姚进发

(锐捷网络股份有限公司 锐捷研究院, 福建 福州 350002)

摘要: 针对命名数据网络(Named Data Networks, NDN)中数据缓存的问题,提出了一种基于分布式部分可观测马尔科夫决策过程(Dec-POMDP)的缓存优化模型。在该模型框架下,NDN 网络节点间以分布式协作的方式逐渐收敛至最优联合缓存策略,从而使 NDN 中的有限缓存资源得到充分利用。仿真实验表明,与现有 NDN 缓存方法相比,该方法能有效提升 NDN 网络的缓存效率,实现较高的缓存命中率及较小的请求平均跳数。

关键词: 命名数据网络;分布式部分可观测马尔科夫决策过程;缓存策略

中图分类号: TP393.0

文献标识码: A

DOI: 10.19358/j.issn.2096-5133.2020.08.012

引用格式: 姚进发. 命名数据网络中基于 Dec-POMDP 的缓存策略[J]. 信息技术与网络安全, 2020, 39(9): 61-67, 73.

A Dec-POMDP-based cache policy in named data networks

Yao Jinfa

(Research Institute of Ruijie, Ruijie Networks Co., Ltd., Fuzhou 350002, China)

Abstract: To address the caching problem of Named Data Networks(NDN), this paper proposes a caching optimization model based on Decentralized Partially Observable Markov Decision Process(Dec-POMDP). Under the framework of Dec-POMDP model, NDN nodes gradually converge to the optimal joint caching strategy in the way of distributed cooperation, resulting in efficient utilization of the limited NDN in-network caching resources. Simulation results demonstrate that the proposed scheme outperforms the existing NDN caching strategies by achieving a higher cache hit rate and a smaller average response hops.

Key words: named data networks; Decentralized Partially Observable Markov Decision Process(Dec-POMDP); cache policy

0 引言

随着网络技术的发展以及互联网用户的快速增加,网络应用的主体正逐步向内容获取和信息服务演进。早期为解决端到端通信问题而设计的基于 TCP/IP 的体系架构对计算机网络性能的限制使得传统互联网难以满足海量的网络数据处理需求,这激发了人们对未来网络架构设计的重新思考与研究。信息中心网络(Information-Centric Networking, ICN)^[1]作为一种“革命性”体系架构,其以内容为中心的特点无缝迎合了未来网络的发展趋势,因而受到研究学者的广泛关注。在 ICN 体系的诸多部署方案中,命名数据网络(Named Data Networks, NDN)因其先进的设计理念、灵活的路由转发机制以及分布式的网内缓存方式等良好特性已经成为 ICN 中的研究热点。

为了满足高效的内容分发与获取的需求,NDN 在设计时通过引入网内缓存(in-network caching)机制来减少不必要的网络数据传输,从而提高数据传输效率,增强网络的可扩展性。在 NDN 中,每个网络节点都具有一个内容存储库(Content Store, CS),用于缓存经过本地节点的数据,从而为后续与数据对应的相关请求提供路径缓存服务。然而,与海量的数据相比,网络节点中 CS 的容量相当有限,因此如何合理地进行内容放置和缓存决策,是影响 NDN 性能的关键因素。

NDN 在设计之初默认采用处处缓存(Cache Everything Everywhere, CEE)策略^[2],但该方法会导致节点缓存内容趋于同质化,故无法充分发挥网内缓存效率。近年来,学术界围绕 NDN 缓存技术的研究已经取得了不少成果。文献[3]针对 CEE 策略的缓

存冗余问题,提出只在请求命中节点的直接下一跳缓存数据(Leave Copy Down, LCD),一定程度上提高了网络缓存的利用率,但流行度高的内容需要被访问多次才能缓存到边缘节点上。文献[4]提出了一种基于内容流行度的协作缓存策略(WAVE),它根据内容请求次数以指数方式逐步增加沿途节点上所缓存的数据包个数,从而实现数据在空间存储位置上的差异化,但该方案并没有考虑内容请求序列的相关性。文献[5]通过估算路径的剩余存储能力来计算同一路径上的不同数据流在沿途各节点上的缓存概率,从而提出了一种兼顾不同数据流间存储公平性的概率缓存策略(ProbCache)。文献[6]提出了一种分布式沿途缓存策略,即最大增益网内缓存(MAGIC)。网络节点基于内容流行度和路由跳数来计算内容的缓存增益,并在数据传输路径上选择具有最大缓存增益的节点进行内容缓存,从而达到减少网络带宽消耗的目的。但该方案在进行缓存决策时需要重新计算各内容的流行度,因此计算量大,执行复杂度高。文献[7]提出了一种主动缓存策略,其主要思想是利用熵来衡量移动性预测的不确定性,并定位最佳的预取节点,从而降低服务器负载,并减少缓存冗余。

针对 NDN 的网络架构特性,本文提出了一种基于 Dec-POMDP 的 NDN 缓存策略。首先利用 Dec-POMDP 理论框架对 NDN 网络的缓存问题进行建模,该模型考虑了缓存节点间的相互协作,以实现降低缓存内容冗余度和内容优化存储的目的。在此基础上,通过限制节点的协作域的方法避免引入过量的额外通信开销,进而降低模型求解的复杂度。最后,本文给出了一种基于强化学习的局部近似最优缓存策略的求解算法。仿真结果表明,该方法能够有效增加缓存内容的多样性,提升缓存命中率,进而减小用户请求内容的总跳数。

1 NDN 工作原理

NDN 网络中的数据传输采用基于用户驱动的“拉”模式数据获取方式。NDN 中存在两种数据包类型:兴趣包(Interest Packet)和数据包(Data Packet)。用户通过发送包含内容名称的兴趣包来向网络请求数据,该请求兴趣包将被路由到存有对应数据的邻近节点或服务器节点;然后,携带该请求内容的数据包沿着兴趣包的反向路径被逐跳传送给数据请求者。

如图 1 所示,当请求兴趣包到达时,网络节点根据所请求的内容名字依次在内容存储库(Content Store, CS)、待定兴趣表(Pending Interest Table, PIT)和转发信息库(Forwarding Information Base, FIB)中进行匹配查询。若 CS 中存在与该请求对应的数据包则直接将数据传回给请求者;否则,如果 PIT 中含有与该请求对应的条目,则将该请求的进入接口添加到对应条目的接口列表中,然后将该请求抛弃;如果 PIT 也匹配失败,则新建一个 PIT 条目并查询节点的 FIB,将请求转发到下一跳节点。

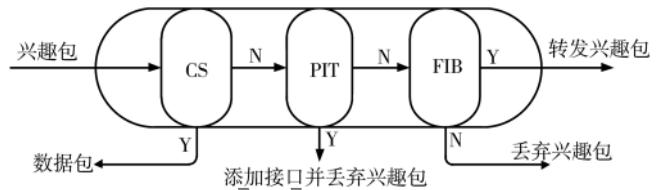


图 1 NDN 中请求包转发过程

当节点收到响应数据包时,节点根据对应的 PIT 表项中所记录的请求接口列表进行数据转发,并根据相应的存储策略将其存储在节点的 CS 中,如图 2 所示。

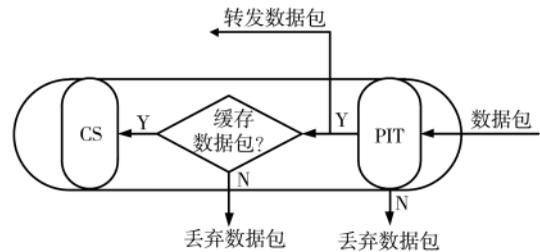


图 2 NDN 中数据包转发过程

2 基于 Dec-POMDP 的 NDN 缓存策略

考虑一个离散时间的 Dec-POMDP 动态系统。在每一个决策时刻,每个智能体首先从系统中获取在当前状态下各自的观测信息,然后做出决策并执行动作,系统根据状态转移函数转移到下一个状态,并获得一个即时回报,整个过程周而复始。因此,一个 Dec-POMDP 模型通常可以由一个 7 元组定义^[8]:

$$\{N, S, A, \Omega, P, O, R, H\}$$

其中, $N = \{1, \dots, N\}$ 表示系统中所有智能体的集合。 $S = \{S_i\}$ 表示一个有限的系统状态集合,其中 S_i 表示智能体 i 的内部状态集。 $A = \{A_i\}$ 表示所有智能体的联合动作集,其中 A_i 表示智能体 i 的可用动作集。

$\Omega = \{\Omega_i\}$ 表示系统联合观测的有限集, 其中 Ω_i 为智能体 i 的观测集。 $P: S \times A \rightarrow [0, 1]$ 表示系统的状态转移函数, $p(s' | s, a)$ 表示系统在状态 s 下采取联合行动 a 后转移到新状态 s' 的概率。 $O: S \times A \times S \rightarrow [0, 1]$ 表示系统的观测函数。 $O(o | s, a, s')$ 表示状态 s 中采取联合行动 a 后转移到新状态 s' 时获得联合观测 $o = \{o_1, \dots, o_N\} \in \Omega$ 的条件概率。 $R: S \times A \rightarrow Z$ 表示系统的效用函数。 $R(s, a)$ 表示智能体集合在状态 s 中采取联合行动 a 后所获得的回报。 H 表示整个决策过程的总周期数, 称为步数 (Horizon)。下面给出关于 NDN 缓存问题的 Dec-POMDP 模型中各元组定义。

2.1 网络模型及相关假设

NDN 网络模型如图 3 所示。网络由源服务器节点、路由节点以及用户节点构成。源服务器节点是内容数据的发布者, 维护内容数据的一个永久副本。路由节点具备存储功能, 可以缓存经过节点的部分数据。用户节点通过边缘路由节点向网络请求数据。用户请求被逐跳转发至拥有该请求内容的中间路由节点或源服务器节点, 并触发相应数据包沿着请求兴趣包的反向路径传送给请求者。

在 NDN 中每个内容对象被划分成若干相同大小的数据块, 并以数据块为单位进行存储。假设网络中共有 M 种不同的数据块对象, 记为 M , 且都具有一个固定的数据源。考虑具有 N 个路由节点的网络, 节点 n 的缓存大小记为 C_n , 且 $C_n \ll M$; 节点接收来自本地用户或相邻下游节点所转发的内容请求。节点 n 上的转发接口个数记为 F_n 。假设网络中的用户请求服从泊松分布, 根据随机服务系统原理, 服从泊松分布的输入流的输出也服从泊松分布, 因此邻节点上未命中的请求序列也服从泊松分布。

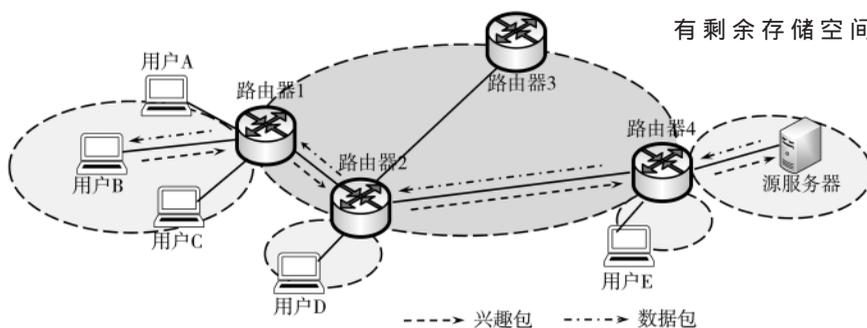


图 3 NDN 网络模型

2.2 状态和状态空间

网络中的所有节点构成 Dec-POMDP 模型的智能体集合 N 。记网络节点 n 的状态为 $s_n = [x_n, y_n]$, 其中 $x_n = [x_{n1}, \dots, x_{nM}, \dots, x_{nf1}, \dots, x_{nfM}]$ 表示当前时刻节点 n 中各转发接口上待响应的内容请求的情况, $x_{f,m}$ 表示节点 n 中第 f_n 个接口上关于数据块 m 的待响应请求数量; $y_n = [y_{n1}, \dots, y_{nM}]$ 表示当前时刻节点 n 上 CS 中的缓存情况, $y_{nm} \in \{-1, 0, 1\}$ 表示数据块 m 的缓存状态。 $y_{nm} = -1$ 表示节点 n 上不存在数据块 m 的副本; $y_{nm} = 0$ 表示节点 n 上拥有数据块 m 的临时副本, 但数据块 m 不被存储于节点的 CS 中而是仅用于响应当前请求, 且当数据转发完成后就会被立即删除; $y_{nm} = 1$ 表示数据块 m 被存储于节点 n 的 CS 中。各网络节点的状态构成当前决策时刻的网络状态, 记为 $s = [s_1, \dots, s_N]$ 。

2.3 决策时刻

本文中考虑基于事件驱动的离散时间 Dec-POMDP 模型。定义 NDN 中的如下三种网络事件: 令 e_{nm}^D 表示节点 n 接收到关于数据块 m 的请求包; e_{nm}^I 表示节点 n 接收到数据块 m 响应包; e_{nm}^Q 表示节点 n 服务完成一个关于数据块 m 的请求。那么, 网络中的事件集合 E 可以表示为:

$$E = \{e_{nm}^w \mid n \in N, m \in M, w \in [I, D, Q]\}$$

2.4 行动和行动空间

由 NDN 的工作原理可知, 当收到响应数据包时, 节点根据其 CS 中的存储剩余空间以及该内容被访问的情况来决定是否缓存该数据包的副本。因此, 本文定义 NDN 节点接收到响应数据包的时刻为有效决策时刻。具体地, 当事件 e_{nm}^I 或事件 e_{nm}^Q 发生时节点不进行任何存储决策, 此时节点 n 的行动记为 $a_n = d_{-1}^n$ 。当事件 e_{nm}^D 发生时, (1) 若节点 n 的 CS 具有剩余存储空间则直接缓存该数据包副本而不需要任何决策, 也即采取行动 $a_n = d_{-1}^n$; (2) 若节点 n 的 CS 中无足够存储空间且节点需要存储该数据包, 那么节点根据存储策略选择当前 CS 中的某个数据包 $k (k \in \{1, \dots, M\})$ 进行替换, 记为 $a_n = d_k^n$; (3) 此外, 令 $a_n = d_0^n$ 表

示节点 n 采取“不存储该数据副本”的行动。综上，网络节点 n 的行动空间 A_n 可以表示为： $A_n = \{d_k^n | k = \{-1, 0, 1, \dots, M\}\}$ 。进而可得到所有网络节点的联合行动为 $\mathbf{a} = [a_1, \dots, a_N]$ ，以及网络节点的联合动作集 $A = \bigcup_{i \in N} A_i$ 。

2.5 观测空间

在每个决策时刻，网络节点先获取当前网络状态的一个观测然后再进行存储决策。假设同一时刻只有一个事件 $e \in E$ 发生。定义节点的观测表示当前决策时刻下该节点上 CS 中缓存内容的变化情况。具体地，当事件 $e_{nm}^w \in E (\forall n, \forall m, w \in [I, Q])$ 发生，即节点接收到数据请求或完成数据请求服务时，定义节点 n 的观测为 $\mathbf{o}_n = \mathbf{y}_n$ 。当事件 $e_{nm}^D \in E$ 发生，即节点 n 接收到新响应数据包时，定义节点 n 的观测为 $\mathbf{o}_n = \mathbf{y}_n + 2\mathbf{B}_m$ ，其中 $\mathbf{B}_i = (0, \dots, 1, \dots, 0)$ 表示第 i 个元素为 1 的单位向量。根据上述定义可知，当 $\sum \mathbf{o}_n \leq C_n$ 时，节点不需要进行决策，当 $\sum \mathbf{o}_n = C_n + 1$ 时，节点需要根据存储策略删除 CS 中的某个数据包。进而各网络节点的本地观测 \mathbf{o}_n 构成网络的一个联合观测 $\mathbf{o} = \{\mathbf{o}_1, \dots, \mathbf{o}_N\}$ 。

2.6 效用函数

网络性能分析是网络优化的基础，网络性能评价指标是判断网络优化效果的依据。常用的评价指标包括带宽、时延、吞吐量等。本文考虑以缓存命中率以及网络平均跳数作为联合策略的衡量标准。定义节点 n 的效用函数为：

$$R_n(s, \mathbf{a}) = \sum_{f_n} \sum_m (x_{f_n m} (1 - I_1(y_{nm})) + h_{f_n m} x_{f_n m}^h) \quad (1)$$

式中， $\sum_{f_n} \sum_m (x_{f_n m} (1 - I_1(y_{nm}))$ 表示未命中的请求数， $h_{f_n m}$ 表示关于数据包 m 的请求在节点 n 命中时所经过的跳数， $x_{f_n m}^h$ 表示关于数据包 m 的请求在节点 n 命中的数量。由此可得到网络的整体效用函数为 $R(s, \mathbf{a}) = \sum_n R_n(s, \mathbf{a})$ 。此外，假设系统的初始状态为 s^0 ，那么一个联合策略 $\omega(t)$ 在时间 T 内平均效用可以表示为：

$$J(\omega(t), s^0) = \lim_{H \rightarrow \infty} \mathbf{E}_{s^0}^\omega \left\{ \sum_{h=0}^H \frac{R(s^h, \mathbf{a}^h)}{t^{h+1} - t^h} \right\} \quad (2)$$

如果对于任意联合策略向量 $\omega(t)$ ，任意初始状态 s^0 ，有 $J(\omega^*(t), s^0) \geq J(\omega(t), s^0)$ ，那么策略 $\omega^*(t)$ 为最优联合策略，其相应的最优性能值记为 J^* 。

2.7 状态转移函数

记 $\lambda_{nf_n m}^I$ 表示节点 n 中第 f_n 个转发接口上数据包 m 的请求到达率， $u_{nf_n m} = r_{nf_n m} / L$ 表示节点 n 对数据包 m 的请求服务率，其中 $r_{nf_n m}$ 表示节点 n 中第 f_n 个转发接口上数据包 m 的传输速率， L 表示数据包的大小。记 $\lambda_{nf_n m}^D$ 表示节点 n 中第 f_n 个接口上响应数据包 m 的到达率。由定义可知， $\lambda_{nf_n m}^D = I_1(y_{nm}) I_2(x_{vf_n m}) u_{vf_n m}$ ，其中 f_v 表示与接口 f_n 相连的邻居节点 v 上的对端接口。函数 $I_1(x)$ 和 $I_2(x)$ 为示性函数，其定义分别为：

$$I_1(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{其他} \end{cases} \quad (3)$$

$$I_2(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{其他} \end{cases} \quad (4)$$

为方便表述，令：

$$\mathbf{x}_{f_n m}^+ = [x_{11}, \dots, x_{f_n m} + 1, \dots, x_{F_n M}] ,$$

$$\mathbf{x}_{f_n m}^- = [x_{11}, \dots, x_{f_n m} - 1, \dots, x_{F_n M}] ,$$

$$\mathbf{y}_{nm}^+ = [y_{n1}, \dots, y_{nm} + 1, \dots, y_{nM}] ,$$

$$\mathbf{y}_{nm}^- = [y_{n1}, \dots, y_{nm} - 1, \dots, y_{nM}] ,$$

$$\hat{\mathbf{y}}_n^{\alpha\beta} = [y_{n1}, \dots, -y_{n\alpha}, \dots, -y_{n\beta}, \dots, y_{nM}] ,$$

$$\mathbf{M}_1 = \left\{ m_1 \left[\begin{array}{l} y_{nm_1} = 1, x_{f_n m_1} > 0 \text{ or} \\ y_{nm_1} = 0, x_{f_n m_1} > 0, \sum_{f_n} x_{f_n m_1} \geq 2 \end{array} \right. \right\} ,$$

$$\mathbf{M}_2 = \left\{ m_2 \left[\begin{array}{l} y_{nm_2} = 0, x_{f_n m_2} = 1, \sum_{f_n} x_{f_n m_2} = 1 \end{array} \right. \right\}$$

由于当 $\sum \mathbf{y}_n \leq C_n$ 时节点 n 采用贪婪存储策略，故不需要进行存储决策，仅考虑在 $\sum \mathbf{y}_n = C_n + 1$ 的情况下，节点 n 在内部状态 $s_n = [\mathbf{x}_n, \mathbf{y}_n]$ 上采取行动 a_n 后转移到下一个状态 s'_n 的概率。

(1) 当 $a_n = d_0^n$ 时，

$$p(s'_n | s_n, a_n) = \begin{cases} \frac{\sum_{f_n} \lambda_{nf_n m}^D}{\lambda_1^D}, & s'_n = [\mathbf{x}_n, \mathbf{y}_{nm}^+] \\ 0, & \text{其他} \end{cases} \quad (5)$$

(2) 当 $a_n = d_\beta^n$ 时，

$$p(s'_n | s_n, a_n) = \begin{cases} \frac{\sum_{f_n} \lambda_{nf_n, \alpha}^D}{\lambda_1^D}, & s'_n = [x_n, \hat{y}_n^{\alpha\beta}], y_{n\beta} = 1 \\ 0, & \text{其他} \end{cases} \quad (6)$$

(3) 当 $a_n = d_{-1}^n$ 时,

$$p(s'_n | s_n, a_n) = \begin{cases} \frac{\lambda_{nf_n, m}^I}{\lambda^I + \lambda_2^D + u^T}, & s'_n = [x_{f_n, m}^+, y_n] \\ \frac{u_{nf_n, m_1}}{\lambda^I + \lambda_2^D + u^T}, & s'_n = [x_{f_n, m_1}^-, y_n], m_1 \in M_1 \\ \frac{u_{nf_n, m_2}}{\lambda^I + \lambda_2^D + u^T}, & s'_n = [x_{f_n, m_2}^-, y_{nm_2}^-], m_2 \in M_2 \\ \frac{\lambda_2^D}{\lambda^I + \lambda_2^D + u^T}, & s'_n = [x_n, y_n] \\ 0, & \text{其他} \end{cases} \quad (7)$$

式中, $\lambda^I = \sum_{n, f_n, m} \lambda_{nf_n, m}^I$, $\lambda_1^D = \sum_{f_n, m} \lambda_{nf_n, m}^D$, $\lambda_2^D = \sum_{i(i \neq n), f_i, m} \lambda_{if_i, m}^D$,

$$u^T = \sum_{n, f_n, m} I_1(y_{nm}) I_2(x_{nf_n, m}) u_{nf_n, m} \circ$$

2.8 观测概率

定义 $M_3 = \{m | y_{nm} = -1\}$ 表示不存在于节点 n 上的内容集合。根据节点观测状态的定义, 可得到如下节点观测概率函数:

$$O(o_n | s_n) = \begin{cases} \frac{\sum_{f_n} \sum_m \lambda_{nf_n, m}^I + \sum_{f_n} \sum_m I_1(y_{nm}) I_2(x_{nf_n, m}) u_{nf_n, m}}{\lambda^I + \lambda_3^D + u^T}, & o_n = y_n \\ \frac{\sum_{f_n} \lambda_{nf_n, m}^D}{\lambda^I + \lambda_3^D + u^T}, & o_n = y_n + 2B_m, \text{ for } m \in M_3 \\ 0, & \text{其他} \end{cases} \quad (8)$$

式中, $\lambda_3^D = \sum_{n, f_n, m} \lambda_{nf_n, m}^D$ 。

3 基于 Dec-POMDP 的分布式缓存协作策略搜索算法

Dec-POMDP 模型最优策略的求解是一个 NEXP-complete 问题, 即使是小规模的问题也往往需要在巨大策略空间上进行最优策略搜索, 从而限制了其在较大规模实际问题中的运用。鉴于 Dec-POMDP

模型精确求解所要求的时间和空间复杂度高, 因此不少学者致力于模型近似解的研究, 通过利用动态规划或者启发式算法来搜索最优策略, 以缓解模型求解过程的维度灾难和计算复杂度的问题。而 Q 学习算法作为经典的强化学习算法, 近年来被广泛应用于多智能体系统体系结构模型的求解中^[9-13]。此外, 理想情况下, 节点通过感知网络全局状态信息, 并根据其他节点所采取的缓存策略进行缓存决策, 从而实现网内缓存资源配置的全局最优化。然而, 这种全域节点间的相互协作会带来大量的额外通信开销, 因此实际中一般只考虑一定跳数范围内的节点协作方式。这里只考虑一跳的情况, 即网络节点通过与相邻节点进行缓存协作逐渐达到整个网络的最优联合策略。基于上述考虑, 本文采用一种近似的 Q 学习方法来求解局部最优策略。

定义节点 n 的邻居节点集合为 N_n , 根据文献[14], 假设网络状态满足状态转移独立性, 则可以得到近似局部联合观测概率为:

$$\begin{aligned} O(o_n^t, o_{N_n}^t | s^{t-1}, a^{t-1}) &\approx \\ O(o_n^t, o_{N_n}^t | s_n^{t-1}, s_{N_n}^{t-1}, a_n^{t-1}, a_{N_n}^{t-1}) & \\ = \prod_{i \in \{n, N_n\}} O(o_i^t | s_i^{t-1}, a_i^{t-1}) & \end{aligned} \quad (9)$$

进而有节点 n 的局部 Q_n 函数为:

$$\begin{aligned} Q_n(o_n^t, o_{N_n}^t, a_n^t, a_{N_n}^t) &= R_n(s_n^t, s_{N_n}^t, a_n^t, a_{N_n}^t) + \\ &\sum_{o_n^{t+1}, o_{N_n}^{t+1}} O(o_n^{t+1}, o_{N_n}^{t+1} | s^t, a^t) Q_n(o_n^{t+1}, o_{N_n}^{t+1}, a_n^{t+1}, a_{N_n}^{t+1}) \end{aligned} \quad (10)$$

从而整个网络的 Q 函数可以近似为所有节点局部 Q_n 函数的和, 即 $Q(o, a) = \sum_n Q_n(o_n, o_{N_n}, a_n, a_{N_n})$ 。此外, 观测动作值函数 $Q_n(o_n, o_{N_n}, a_n, a_{N_n})$ 的迭代公式为:

$$\begin{aligned} Q_n^{t+1}(o_n, o_{N_n}, a_n, a_{N_n}) &= (1 - \alpha^t) Q_n^t(o_n, o_{N_n}, a_n, a_{N_n}) + \\ &\alpha^t [R_n(o_n, o_{N_n}, a_n, a_{N_n}) + \max_{a_n} Q_n(o_n, o_{N_n}, a_n, a_{N_n})] \end{aligned} \quad (11)$$

理论上, $O(o_n^t | s_n^{t-1}, a_n^{t-1})$ 可通过计算得到, 但由于实际中其计算复杂度往往相对较大, 因此, 实际中可通过蒙特卡罗法得到相应的估计值, 即:

$$O(o_n^t | s_n^{t-1}, a_n^{t-1}) = \frac{f(s_i^{t-1}, a_i^{t-1}, o_i^t)}{\sum_{o_i^t} f(s_i^{t-1}, a_i^{t-1}, o_i^t)} \quad (12)$$

4 数值仿真

为了评估本方案的性能,本文采用基于 NS-3 的 ndnSIM^[15]进行仿真。仿真拓扑共有 100 个节点,其中 1 个源服务器节点和 20 个客户端节点,其余为路由节点。源服务器节点中存储 5 000 个不同的数据,每个数据包的大小均为 1 024 KB,数据包的流行度服从 Zipf 分布,实验中参数 α 的取值范围为 0.7~1.3。除特殊说明外,实验中路由节点 CS 的总容量设定为数据总数的 10%,且初始为空。客户端通过边缘路由节点向网络请求数据,且请求到达率服从泊松分布。仿真时间为 2 000 s,其余参数均为固定默认值。下面给出本文所提出的基于 Dec-POMDP 缓存算法与 CEE、LCD 缓存策略的实验结果对比分析。首先定义如下性能指标:

(1)平均缓存命中率:定义为由网内路由节点服务的请求数与总的用户请求数之比,即:

$$HIT_{avg} = \frac{\sum_n Req_hit_n}{\sum_n Req_user_n} \quad (13)$$

式中,Req_hit_n表示在路由节点上命中的请求数量;Req_user_n表示路由节点 n 上接收到的本地用户请求个数。

(2)平均跳数减少率:反映由于缓存系统的加入使用户请求内容跳数减少的比率,定义为:

$$HR_{avg} = 1 - \frac{\sum_i^{R_count} hop_cs_i}{\sum_i hop_ser_i} \quad (14)$$

式中,hop_cs_i表示缓存命中节点到请求用户间的跳数;hop_ser_i表示请求用户到源服务器节点间的跳数;R_count表示总的用户请求数。

图 4 为不同缓存策略下,用户请求在网内的平均缓存命中率随 Zipf 分布参数 α 变化的曲线。如图所示,相比于另外两种策略,基于 Dec-POMDP 的缓存策略能够显著提高用户请求的缓存命中率。这是由于 Dec-POMDP 模型中节点在进行缓存决策时考虑了邻居节点间的缓存协作,因此能够有效地减少缓存内容的冗余度,提升缓存利用率。此外,随着参

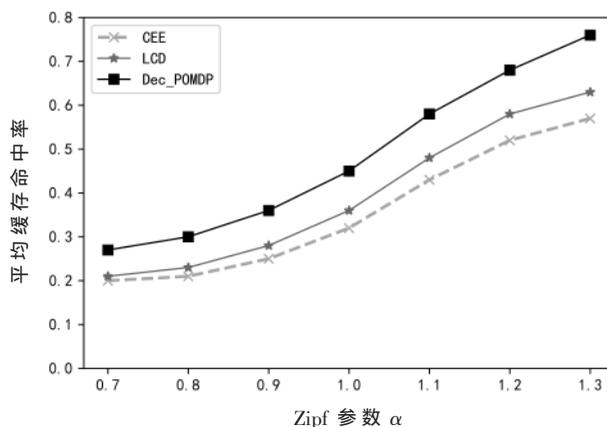


图 4 平均缓存命中率随参数 α 变化曲线

数 α 的增大,三种策略的缓存命中率不断增大;当参数 α 大于 1.1 时,命中率增大的速率有所减缓,这主要是受限于网络路由节点的缓存容量。

图 5 为不同缓存策略下,用户请求的平均跳数减少率随 Zipf 分布参数 α 变化的曲线。由于 Dec-POMDP 模型能够更好地刻画网络状态的不确定性和用户请求的随机性,因此 Dec-POMDP 策略能够更准确地预测请求内容的流行度并有选择性地缓存更有价值的内容,从而有效减少用户请求所需的跳数。

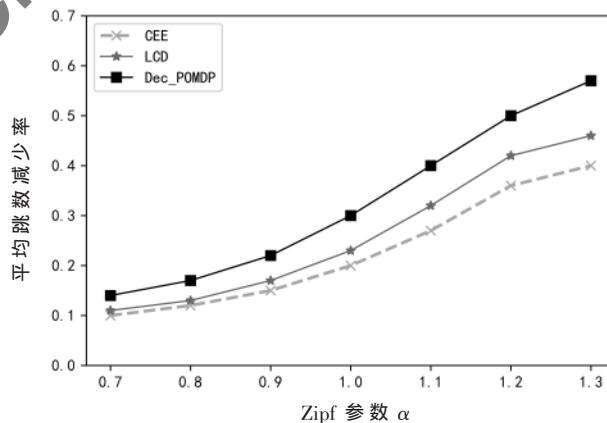


图 5 平均跳数减少率随参数 α 变化曲线

图 6 为不同缓存策略下,用户请求在网内的平均缓存命中率随网内存储总容量变化的曲线。由图可知,在不同网内存储容量配置下,基于 Dec-POMDP 的缓存策略的性能均优于 CEE 和 LCD,说明本文提出的缓存策略能更有效地利用缓存资源。此外,对于 CEE 和 LCD 两种缓存策略,当网内存储容量小于 15% 时,网内存储容量的增加对平均缓存命中率的提升效果并不明显;当网内存储容量继续增加直至达到 35% 时,网内请求的平均缓存命中率有显著提升;此后,网

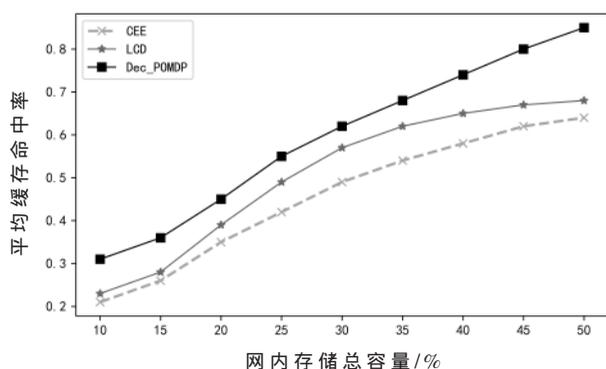


图6 平均缓存命中率随网内存储容量变化曲线

内存容量继续增长,平均缓存命中率的增长速度逐渐变缓。这是因为这两种策略存在缓存冗余,因此无法充分利用网内缓存资源。而基于 Dec-POMDP 的缓存策略由于能够处理请求的动态变化,因此在不同网内存储容量下均能更有效地利用缓存资源。

图7为不同缓存策略下,用户请求的平均跳数减少率随网内存储总容量变化的曲线。从图中可看出,在不同网内存储容量条件下,三种缓存策略的平均跳数减少率性能曲线与图6中对应的曲线趋势相符。由于基于 Dec-POMDP 的缓存策略能够更有效地利用缓存资源,因此在不同网内存储容量下,基于 Dec-POMDP 的缓存策略相对于 CEE 和 LCD 能够更有效减少用户请求所需的跳数;且随着网内存储容量的增加,基于 Dec-POMDP 的缓存策略与另两种策略间的性能差异越明显。

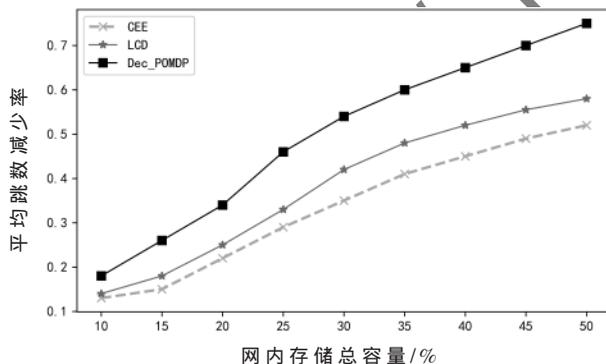


图7 平均跳数减少率随网内存储容量变化曲线

5 结论

NDN 作为一个新型的网络体系架构,广泛存在的网内缓存是其最重要的特征之一。然而,必须合理地利用 NDN 中有限的网内缓存资源才能充分发挥 NDN 网络的性能。因此,本文将 NDN 中的网内缓存问题抽象为一个多智能体分布式协作的

Dec-POMDP 模型,并将其建模为一个多目标优化问题。基于缓存性能与网络通信开销的权衡,本文仅考虑有限跳数范围内的网内节点缓存协作并在该模型的基础上提出了一种近似 Q 学习的强化学习算法以求解局部最优缓存策略。仿真实验结果表明,基于 Dec-POMDP 缓存策略能够根据用户请求的动态变化更好地分配缓存资源,从而提高了用户请求的网内缓存命中率,有效地缓解了服务器压力。

参考文献

- [1] GHODSI A, SHENKER S, KOPONEN T, et al. Information-centric networking: seeing the forest for the trees[C]. Proceedings of the 10th ACM Workshop on Hot Topics in Networks, 2011: 1.
- [2] JACOBSON V, SMELTERS D K, THORNTON J D, et al. Networking named content[J]. Conext, 2009, 55(1): 1-12.
- [3] LAOUTARIS N, CHE H, STAVRAKAKIS I. The LCD interconnection of LRU caches and its analysis[J]. Performance Evaluation, 2006, 63(7): 609-634.
- [4] CHO K, LEE M, PARK K, et al. Wave: popularity-based and collaborative in-network caching for content-oriented networks[C]. Proceedings IEEE 31st International Conference on Computer Communications, Orlando, FL, USA, 2012: 316-321.
- [5] PSARAS I, CHAI W K, PAVLOU G. Probabilistic in-network caching for information-centric networks[C]. Proceedings of the 2nd ACM SIGCOMM Workshop, ICN, Helsinki, Finland, 2012: 55-60.
- [6] REN J, QI W, WESTPHAL C, et al. Magic: A distributed max-gain in-network caching strategy in information-centric networks[C]. Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2014: 470-475.
- [7] ABANI N, BRAUN T, GERLA M. Proactive caching with mobility prediction under uncertainty in information-centric networks[C]. Proceedings of the 4th ACM Conference on Information-Centric Networking. ACM, 2017: 88-97.
- [8] ARAS R, DUTECH A. An investigation into mathematical programming for finite horizon decentralized POMDPs[J]. Journal of Artificial Intelligence Research, 2010, 37: 329-396.

(下转第 73 页)

件加窗处理。另一方面,由于数据量较大,本文采取先缓存后处理方式,整体消耗时间相对偏长,接下来将尝试采用多路并行 FFT 方案进一步提升速度。

4 结论

本文设计了一种基于 FPGA 的功率谱分析系统,通过硬件平台和逻辑算法的结合,灵活实现了数据采集、触发、缓存、功率谱估计、结果上传等系列功能。测试结果表明,两种 FFT 处理方案均达到较高的精度和速度。在上位机软件的控制下,系统能够实现实时功率谱估计,并通过分段多次累加改善功率谱估计的性能,提高了激光雷达对微弱信号的探测处理能力。

参考文献

- [1] 宋宁,关华.经典功率谱估计及其仿真[J].现代电子技术,2008,31(11):159-161.
- [2] 邓泽怀,刘波波,李彦良.常见的功率谱估计方法及其 Matlab 仿真[J].电子科技,2014,27(2):50-52.
- [3] 王仕奎.随机信号分析理论与实践[M].南京:东南大学出版社,2016.
- [4] 贾晓东.1.55 μm 相干测风激光雷达样机的研制[D].合肥:中国科学技术大学,2015.
- [5] 徐文波,田耘.Xilinx FPGA 开发实用教程(第二版)[M].北京:清华大学出版社,2012.
- [6] 张晓光.基于激光相位的高速量子随机数发生器[D].合肥:中国科学技术大学,2017.
- [7] 段艳丽,王敏,林永照,等.数字信号处理[M].北京:电子工业出版社,2015.
- [8] 苏斌,刘畅,潘志刚.基于 FPGA 的高速浮点 FFT/IFFT 处理器设计与实现[J].中国科学院大学学报,2015,32(2):259-263.
- [9] 高亚军.基于 FPGA 的数字信号处理[M].北京:电子工业出版社,2012.
- [10] HE S S, TORKELESON M. Design and implementation of a 1024-point pipeline FFT processor[C]. Proceedings of the IEEE 1998 Custom Integrated Circuits Conference. IEEE, 1998: 131-134.
- [11] 付友涛.相干测风激光雷达实时数据采集处理系统研究[D].青岛:中国海洋大学,2013.
- [12] 漆洋.小型激光测风雷达信号处理器研制[D].成都:成都信息工程大学,2018.
- [13] 董佳.基于 FPGA 的相干测风激光雷达实时信号预处理实验研究[D].哈尔滨:哈尔滨工业大学,2017.

(收稿日期:2020-04-29)

作者简介:

付恩琪(1996-),女,硕士研究生,主要研究方向:物理电子学。

孙秀男(1992-),男,博士研究生,主要研究方向:物理电子学。

梁昊(1970-),通信作者,男,博士,副教授,主要研究方向:核电子学、信号采集与处理等。E-mail: simonlh@ustc.edu.cn。

(上接第 67 页)

- [9] OLIEHOEK F A, SPAAN M T J, VLASSIS N. Optimal and approximate Q-value functions for decentralized POMDPs[J]. Journal of Artificial Intelligence Research, 2008, 32: 289-353.
- [10] BANERJEE B, LYLE J, KRAEMER L, et al. Solving finite horizon decentralized POMDPs by distributed reinforcement learning[C]. Proceedings of the AAMAS Workshop on Multiagent Sequential Decision Making Under Uncertainty (MSDM-12), 2012: 9-16.
- [11] FOERSTER J N, ASSAEL Y M, FREITAS N D, et al. Learning to communicate to solve riddles with deep distributed recurrent Q-Networks[J]. arXiv:1602.02672, 2016.
- [12] NISIOTI E, THOMOS N. Fast Q-learning for improved finite length performance of irregular repetition slotted ALOHA[J]. IEEE Transactions on Cognitive Communications and Networking. DOI:10.1109/TCCN.2019.2957224.
- [13] ZHANG W X, MA L, LI X N. Multi-agent reinforcement learning based on local communication[J]. Cluster Computing, 2019, 22: 15357-15366.
- [14] BECKER R, ZILBERSTEIN S, LESSER V, et al. Solving transition independent decentralized markov decision processes[J]. Journal of Artificial Intelligence Research, 2004, 22: 423-455.
- [15] AFANASYEV A, MOISEENKO I, ZHANG L X. ndnSIM: NDN simulator for NS-3[D]. Los Angeles: University of California, 2012.

(收稿日期:2020-04-27)

作者简介:

姚进发(1990-),男,博士,研究员,主要研究方向:未来网络、强化学习。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《信息技术与网络安全》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、JST日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《信息技术与网络安全》编辑部
中国电子信息产业集团有限公司第六研究所