ACO-inspired ICN Routing Scheme with Density-Based Spatial Clustering

Jianhui Lv, Xingwei Wang, Min Huang

ACO-inspired ICN Routing Scheme with Density-based Spatial Clustering

Jianhui Lv[1], Xingwei Wang[2], and Min Huang[3]

[1]College of Computer Science and Engineering, Northeastern University, Shenyang, 110169, China
[2]College of Software, Northeastern University, Shenyang, 110169, China
[3]College of Information Science and Engineering, Northeastern University, Shenyang, 110819, China
lvjianhui2012@163.com, wangxw@mail.neu.edu.cn, mhuang@mail.neu.edu.cn

Abstract. This paper proposes a new Ant Colony Optimization (ACO)-inspired Information-Centric Networking (ICN) routing scheme with Density-based Spatial Clustering (DSC). At first, the content concentration model is established with network load to address interest routing; in particular, we investigate the failed situation of content retrieval. Then, the dot product method is used to compute similarity relation between two routers, which is considered as clustering reference attribute. In addition, DSC is exploited to detect core nodes which are used to cache contents during the data routing process. Finally, the proposed scheme is simulated over Mini-NDN, and the results show that it has better performance than the benchmark scheme.

1 Introduction

In recent years, Ant Colony Optimization (ACO)-inspired solutions have been proposed to address Information-Centric Networking (ICN) routing issue [1-3]. In ACO-inspired ICN solution which relies on Named Data Networking (NDN) architecture, Interest packet and Data packet are regarded as Interest ant (Iant) and Data ant (Dant) respectively. Lv et al. [3] considered user interest request and established similarity relation between two routers by the cached contents, i.e., ACO-inspired ICN Routing with Content concentration and Similarity relation (AIRCS). During the Iant routing process, similarity relation was regarded as a significant inspired factor to conduct the forwarding of Iant. This paper improves AIRCS based on Density-based Spatial Clustering (DSC), and the major contributions are summarized as follows. (i) The dot product method is used to compute similarity relation between two routers, in order to save more computation time. (ii) For the failed Iant, the corresponding updating strategy of content concentration (pheromone) is different from that of the successful Iant, and the updating process considers network load. (iii) DSC based on similarity relation is used to find core routers, in order to cache the multiple content during the Dant routing process.

2 The Proposed AIRD Scheme

2.1 Content concentration design

The computation of content concentration regarding the edge between routers R_i and R_j (denoted by $e_{i,j}$) is the fundamental task. Let $T_{i,j}(t, I)$ denote the total content concentration over $e_{i,j}$ at time t after $I(\in \mathbb{N})$ iterations,

$$T_{i,j}(t, I) = (1 - \rho) \cdot T_{i,j}(t, I - 1) + cc_{i,j}(t, I), \tag{1}$$

where ρ is a volatilization coefficient of content concentration, $1 - \rho$ is a residual factor of content concentration, and $0 < \rho < 1$ prevents the infinite accumulation of content concentration. In addition, $cc_{i,j}(t, I)$ denotes the content concentration over $e_{i,j}$ by some of m $(\in \mathbb{N})$ Iants after the $I - th$ iteration, and it is defined as follows.

$$cc_{i,j}(t, I) = \sum_{\lambda=1}^{m} \tau_{i,j}^{\lambda}(t, I) \cdot x_{\lambda}, \tag{2}$$

where $x_\lambda \in \{0, 1\}$. Let ia_λ denote arbitrary Iant, here $\lambda \in \{1, 2, \cdots, m\}$: if ia_λ traverses $e_{i,j}$, $x_\lambda = 1$; otherwise, $x_\lambda = 0$. In addition, $\tau_{i,j}^\lambda(t, I)$ denotes the content concentration over $e_{i,j}$ left by ia_λ after the $I - th$ iteration.

When ia_λ finds the content within one iteration, $\tau_{i,j}^\lambda(t, I)$ is defined as follows.

$$\tau_{i,j}^\lambda(t, I) = \frac{cos(1 - e^{-pcon_j})}{hop_j \cdot L_\lambda \cdot I}. \tag{3}$$

where $pcon_j$ is the times which R_j provides the content, hop_j is the hop count between interest requester and R_j, and L_λ is the total distance traversed by ia_λ within one iteration. In addition, $cos(1 - e^{-pcon_j})$ is a decreasing function regarding $pcon_j$, and its function is to reduce and balance node load with large $pcon_j$.

When ia_λ cannot find the content within one iteration, $\tau_{i,j}^\lambda(t, I)$ is defined as follows.

$$\tau_{i,j}^\lambda(t, I) = \gamma \cdot \frac{hop_j^{-\sigma}}{L_\lambda}. \tag{4}$$

Among them, σ reflects the importance of Iant to the whole network during the process of updating content concentration; γ is a regulatory factor to avoid the situation where $hop_j^{-\sigma}/L_\lambda$ becomes too large or small.

2.2 Similarity relation computation

Suppose that R_i consists of h_i types of contents and each one is regarded as a class of user interests, then

$$R_i \sim \left(int(i)_1, int(i)_2, \cdots, int(i)_{h_i} \right), \tag{5}$$

where $int(i)_l$ is one type of contents, $1 \leq l \leq h_i$.

Suppose that $int(i)_l$ is quantified as $int(i)_l'$ which equals to l, and that $int(i)_l$ corresponds to $N(i)_l$ content items, then

$$R_i := \left(N(i)_1, 2 \cdot N(i)_2, \cdots, h_i \cdot N(i)_{h_i} \right), \tag{6}$$

Assume that R_i and R_j have p same types of contents, when R_i and R_j are adjacent, then

$$r_{i,j} = \frac{1}{M} \cdot \sum_{l=1}^{max\{h_i, h_j\}} \left(l \cdot N(i)_l \cdot l \cdot N(j)_l \right) = \frac{l^2}{M} \sum_{l=1}^{p} N(i)_l \cdot N(j)_l, \tag{7}$$

$$M = \max_{\forall i,j} \left\{ \sum_{l=1}^{p} N(i)_l \cdot N(j)_l \right\}, \tag{8}$$

where $r_{i,j}$ is the similarity relation between R_i and R_j. Especially when R_i and R_j are not adjacent, $r_{i,j} = 0$.

2.3 DSC-based core detection

In this paper, the DSC-based method is used to determine core routers, and similarity relation between two routers is used as clustering reference attribute. Let $N_\varepsilon(R_i)$ denote the number of routers with a neighbourhood, and it is defined as follows.

$$N_\varepsilon(R_i) = \{R_j \in \boldsymbol{D} | mr_{i,j} \geq eps\}. \tag{9}$$

Among them, $\boldsymbol{D}$ is the ε neighbourhood regarding R_i; $mr_{i,j}$ is the clustering metric between R_i and R_j; eps is a threshold.

If R_i and R_j are adjacent, $mr_{i,j} = r_{i,j}$. If R_i is connected to R_j according to k routers, denoted by $R_{c1}, R_{c2}, \cdots, R_{ck}$ respectively, we have

$$mr_{i,j} = r_{i,c1} \cdot r_{c1,c2} \cdot \cdots \cdot r_{ck,j} = r_{i,c1} \cdot r_{ck,j} \cdot \prod_{q=1}^{k-1} r_{cq,c(q+1)}, \tag{10}$$

which indicates that $mr_{i,j}$ is computed by indirect method among $k + 2$ routers.

Then, the definition about core router is shown as follows.

Definition 1 (Core router): If $N_\varepsilon(R_i) \geq \xi$, R_i is a core router, where ξ is a positive constant.

2.4 Routing decision

The routing decision consists of two parts, i.e., ACO-based Iant routing and DSC-based Dant routing. The first one is same to that in [3] and [4]. If the number of iterations reaches the maximum, the Iant routing is finished and the Dant routing starts, as follows.

At first, content provider generates some Dants. Let Dn denote the number of Dants, and it is defined as follows.

$$Dn = \left\lceil \frac{Sc}{2^o\ B} \right\rceil, \tag{11}$$

where Sc is the content size, o is a positive integer, B is the unit of byte, and the operator of ceiling guarantees that Dn is a positive integer. Dn Dants are denoted by $\mathrm{Dant}(1), \mathrm{Dant}(2), \cdots, \mathrm{Dant}(Dn)$.

Secondly, for one of Dn Dants, it carries the corresponding content from content provider to interest requester along the path which provides the closest content copy for interest requester. When arriving at R_i, it first checks whether R_i is a core router. If yes, two cases emerge: (i) when Content Store (CS) is not full, the carried content is cached to R_i; or (ii) when CS is full, a replacement strategy is used and then the carried content is cached to R_i. Otherwise, it checks Pending Interest Table (PIT) directly to find an appropriate interface in order to forward itself.

Finally, interest requester checks whether the generated Dn Dants have been received. If yes, their carried contents are recombined in the order of $\mathrm{Dant}(1), \mathrm{Dant}(2), \cdots, \mathrm{Dant}(Dn)$. Otherwise, interest requester waits until all Dants arrive.

3 Simulation results

The proposed ACO-inspired ICN Routing scheme with DSC (AIRD) is simulated over Mini-NDN, and Deltacom topology with 97 nodes and 124 edges [4] is selected to do the performance evaluation by sending 1000 interest requests based on 100 times simulation. $\alpha = \beta = \sigma = 1$, $\eta = m = 10$, $\rho = eps = 0.5$, $\gamma = 2$, $\xi = 3$ and $o = 8$.

Fig. 1 shows routing success rate for AIRCS and AIRD. We can observe that routing success rate of AIRD mainly concentrates on 96% and 97%, whilst that of AIRCS mainly concentrates on 93% and 94%. The emergence of these abnormal results is suffered from the congestion of interest requests, which causes the instability of system. Besides, the data selection has a certain randomness. From the global perspective, AIRD has higher routing success rate than AIRCS, and two reasons are as follows. (i) AIRD handles the failed Iants besides the successful Iants and conducts them to gather around the closest content copy, which increases retrieval probability. (ii) AIRD uses DSC to find core routers to cache the content during the Dant routing process, which guarantees content retrieval, especially for the similar and even same interest requests.

Fig. 2 shows load balance degree for AIRCS and AIRD. We can observe that load balance degree of AIRD is always lower than that of AIRCS, which means that AIRD is much more stable than AIRCS. Two reasons are as follows. (i) In the Iant routing stage, AIRD considers network load when updating content concentration. (ii) in the Dant routing stage, AIRD considers the fine granularity of content to divide the content into some pieces which are

the uniform distribution; instead, AIRCS considers the content as an entire object, which leads to serious load in terms of some links.

Fig. 3 shows execution time for AIRCS and AIRD. We can observe that execution time of AIRD is always smaller than that of AIRCS, and two reasons are as follows. (i) AIRD uses dot product method to compute similarity relation, which can save much more computation time than the using of absolute subtrahend method in AIRCS. (ii) AIRD designs and updates content concentration based on discrete model rather than continuous model, which decreases much more than AIRCS.
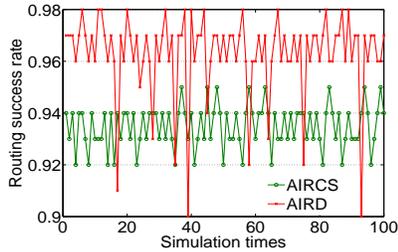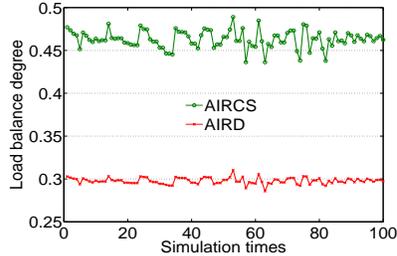

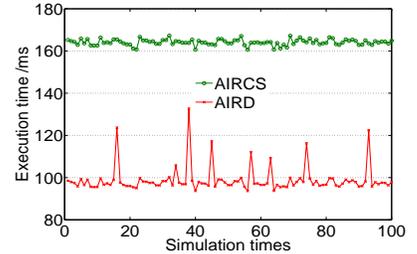**Fig. 1.** Routing success rate.   **Fig. 2.** Load balance degree.   **Fig. 3.** Execution time.

4   Conclusions

This paper extends and deepens the previous research on ACO-inspired ICN routing scheme, and the contributions are triple, i.e., (i) computing similarity relation based on dot product method, (ii) processing the successful packets and failed packets when the content concentration is updated and (iii) detecting core routers to cache the content during the Dant routing process. Then, the performance is evaluated, and the simulation results have shown that the proposed AIRD is more efficient and effective than the previous work.

Acknowledgments

References

1. Shanbhag S., Schwan N., Rimac I., Varvello M.: SoCCeR: services over content-centric routing, in Proc. ACM SIGCOMM (ICN), pp. 62-67 (2011).
2. Lv J., Wang X., Ren K., Huang M., Li K.: ACO-inspired Information-Centric Networking routing mechanism, Computer Networks, **126**, 200-217 (2017).
3. Lv J., Wang X., Huang M.: Ant colony optimization inspired ICN routing with content concentration and similarity relation, IEEE Comm. Lett., **21**(6), 1-4 (2017).
4. Lv J., Wang X., Huang M.: ACO-inspired ICN routing mechanism with mobility support, Applied Soft Computing, **58**, 427-440 (2017).