Named Content Searching in Opportunistic Mobile Networks

Barun Kumar Saha and Sudip Misra, *Senior Member, IEEE*

***Abstract*—In this letter, we study the adaptation of Named Data Networking (NDN)-based named content searching in Opportunistic Mobile Networks (OMNs). Given the inherent uncertainty in OMNs, our goal is to replicate content requests to a suitable set of nodes that can help improve users' experience in terms of requests satisfaction and latency. To this end, we propose a scheme, content searching as regret minimization (CHARM), based on the technique of random regret minimization (RRM). In CHARM, content interest replication and non-replication choices are expressed in terms of several measurable attributes. Regrets associated with the two choices indicate whether or not an interest should be replicated. Moreover, to reduce overhead, CHARM uses dynamic time-to-live (TTL) adaptation, where the lifetime of a message is proactively scaled down. Results of performance evaluation based on real-life traces and synthetic mobility model indicate that CHARM can improve the number of interests satisfied and latency, respectively, by up to 1.6x and 1.4x times when compared to a contemporary scheme.**

***Index Terms*—Opportunistic Mobile Networks, Named Data Networking, Random Regret Minimization, time-to-live**

I. Introduction

The future Internet is envisaged to be content-centric. Among different initiatives taken in this regard, NDN [1] has gained much popularity. In NDN, every content is identified with a unique, hierarchical name, e.g., `example.com/song.mp3`. Any user interested in a given content sends out an *interest* message carrying the name. When another node receives such an interest via a network interface (*face*), it checks whether or not the content corresponding to the name is available in its Content Store (CS). In case it is available, a *data* message is created containing the content and name is sent back to the face via which the interest was received. Otherwise, the node searches for name of the content in its Pending Interest Table (PIT). If an entry for the name is already there in the PIT, the node adds the face to a list corresponding to the name. Otherwise, a new PIT entry is created. If any outgoing face corresponding to the name is available in the Forwarding Information Base (FIB) of the node, the interest message is forwarded to that face. On the other hand, when a node receives a data message, it checks its PIT. If no entry is found corresponding to the name, the data message is discarded. Otherwise, it is stored in its CS. It may be noted that neither interest nor data messages carry any form of node address/identifier. Moreover, name management has been scoped out of the core focus of NDN.

The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India (e-mail: {barun.kumar.saha, smisra}@sit.iitkgp.ernet.in).

OMNs, on the other hand, typically lack in end-to-end communication paths and witness intermittent connectivity among the nodes. Consequently, protocols for traditional networks fail to work in OMNs. Mobility of humans – carriers of the mobile devices – creates communication opportunities in OMNs by bringing them within the transmission circles of one another. Adapting[1] the Internet-based NDN for content searching in OMNs is, therefore, challenging. Unlike Internet routers considered in NDN, a node in an OMN typically has a single network interface, e.g., Wi-Fi. Therefore, named content routing in OMNs is infeasible with interface information. Moreover, due to largely disconnected nature of OMNs, (interest and data) messages are actually *replicated* – rather than forwarded – to enhance their delivery likelihood. Moreover, local information about content availability at the nodes in OMNs involve *uncertainty*, which directly affects the number of interests replicated and subsequently satisfied. Nevertheless, the popularity of NDN has led the following few schemes to incorporate named content searching in OMNs. PIFP [2] uses direct and transitive contact probabilities of nodes with names to decide interest replication candidates. It lacks in consideration of other aspects of network dynamics and the extent to which a node is actually able to satisfy interests. STCR [3] uses a hybrid of NDN and unicast approaches and maintains contents list indexed by nodes. It lacks in consideration that such information can get outdated while an interest is still being routed. Also, these schemes lack in exploiting the fact that a node would often have (or know about) content names of similar type/origin.

Given a set I of interests and a set N of nodes, the goal of content searching in OMNs is to find a mapping (replication strategy) $I \times N \rightarrow \{0, 1\}$, where '1' indicates that an interest is replicated. The Epidemic [4] scheme is simple, but can overwhelm network resources. Therefore, an efficient replication scheme is required to quickly spread interest messages and accelerate subsequent content delivery [5] while using moderate amount of resources. Moreover, such replication decisions should be made with limited locally available information. The focus of this work, however, is not to reduce replication redundancy or filter content [6], but to improve delivery of content searched by users.

The above scenario somewhat resembles online decision

[1] This is important since OMNs can be used to offload traffic from Wi-Fi and cellular networks, for example.

problems (ODPs)[2], where an optimal alternative is to be chosen from a given sequence of trials often based on partial information. Xu et al. summarized various approaches based on Game Theory (GT), Markov Decision Process, multi-armed bandit, and optimal stopping problem that can be used in this regard [7]. However, lack of information on statistical distribution, which is true for OMNs, is often an obstacle in these scenarios [7]. The use of GT in this scenario is difficult since a node, in general, lacks information about payoffs of other nodes. On the other hand, although a no-regret algorithm is often used with ODPs, it requires a reference (offline) strategy(ies) to compare with. Here, such a strategy would be trivial – either always replicate a message to a node or never. Moreover, GT and many decision theoretic models often make an underlying assumption of rationality. However, behavioral economics says that human decision making is often not rational [8], [9]. In particular, Regret Theory (RT) [10], explains this phenomena by suggesting that humans tend to choose an alternative that minimizes his/her *regret*. It should be remembered here that the devices in OMNs are owned and operated by humans and, therefore, users' level of *satisfaction* should also be considered alongside the overall network performance. Consequently, a node's action – whether or not to replicate an interest message to a given node – should also take into account the contemporary regret of its owner.

Motivated by this, we propose CHARM, based on the theory of RRM [11], which itself is rooted in RT. In particular, in RRM, which has been used for discrete choice modeling, two *alternatives*, say i and j, can be characterized with a set of common *attributes*, $x = (x_1, x_2, \cdots, x_M)$. Then, the regret of choosing alternative j over i, R_i, is given by $R_i = \sum_{m=1}^{M} \ln(1 + \exp[\beta_m(x_{jm} - x_{im})])$, where β_m indicates the preference (or dislike) over attribute x_m. The alternative with minimum regret (equivalently, maximum rejoice) is chosen. The use of RRM here is motivated by the fact that it has been widely used in users' travel route choice modeling[3], which closely resembles the problem of content searching in OMNs – whether or not to replicate an interest message to a node with a given cost. Moreover, RRM formulation takes into account random errors, which, however, are integrated out in the final expression. RRM and regret matching (RM) algorithms have some similarity to some extent. While RM prefers an action with maximum positive regret (reward), RRM chooses the one with the minimum regret. However, RM needs to maintain historical regret values for given actions. Since, in OMNs, a given named interest can be replicated to a given node at most once, maintaining such historical actions do not seem relevant. Finally, RRM is designed in a way so that a certain increase in one of the attributes does not necessarily affect the overall decision [11], which is useful in OMNs since available information largely has some uncertainty[4] associated with it.

The specific *contributions* of this work are as follows: 1) Proposing CHARM, an NDN-based content searching scheme for that replicates interest messages using the RRM, 2) Considering a dynamic TTL adaptation scheme to proactively scale down TTL of messages to reduce their extent of replication, and 3) Evaluating the efficiency of CHARM via simulations using real-life connection traces and synthetic mobility model.

II. Design of CHARM

We now discuss about the different components of CHARM.

A. NDN Components

A record in the PIT of a node is of the form $(\texttt{name}, \{hosts\}, ttl)$, where $\{hosts\}$ denotes the set of nodes from whom an interest message carrying `name` is received; ttl is the TTL of the concerned interest message. CS is a key-value storage of names and their corresponding data.

A node maintains names known to it and the corresponding timestamps of their last contacts[5] in its FIB. Let us consider that node i comes in contact with `name` at time instant t. Here, node i can use the information $(\texttt{name}, t)$ for its subsequent interest replication decisions, but storing distinct names would enormously increase the size of FIB. Therefore, we choose to store the *name prefixes*[6] instead. Here, "prefix" denotes the first level in the hierarchy of an NDN name. E.g., while `example.com/photo.jpg` and `example.com/music/song.mp3` are two distinct names, their common prefix is `example.com`. Thus, whenever a node comes in contact with a name, a record of the form $(\texttt{prefix}, t)$ is stored in its FIB. However, if such a record already exists, then its timestamp is set to the current time.

Let us consider a subsequent contact between nodes i and k neither of which have the data corresponding to `name` in their CS; k has no entry for it in its FIB. However, since i has some (possibly old) information about `prefix` in FIB, k adds a new record $(\texttt{prefix}, \gamma t)$ to k's FIB, where $0 < \gamma < 1$ is a damping factor. The constant γ here signifies that, since information about the name is not obtained first-hand, it could be stale or inaccurate. Without loss of generality, we assume that $\gamma = 0.99$. Moreover, if yet another node l with an entry $(\texttt{prefix}, t')$ later comes in contact with k such that $t' < t$, then l updates the corresponding record as $(\texttt{prefix}, \gamma t)$. Once again, the same reasoning for staleness is applied here.

B. Local Observations

Additionally, each node relies upon the following four locally observed metrics: 1) **Interest hits**, which occur when

[2]Generally speaking, an ODP involves computation of the form $\sum_{i=1}^{K} loss(state_i, decision_i)$ over K time instants and similar computations for other strategies. Based on these, a notion of regret is defined with the aim of minimizing it.

[3]E.g., whether to travel by flight at cost c_1 and duration d_1 or by train at cost c_2 and duration d_2; $c_1 > c_2, d_1 < d_2$

[4]It should also be noted that in OMNs, inter-contact time (ICT) among the nodes is usually high so that the decision points are often widely separated in time. So, the time spent in searching for time-averaged minimum-regret solution is significant in which the network dynamics in terms of content "association" of the nodes can largely change. In contrast, by using RRM, we minimize the contemporary (instantaneous) regret and thereby try best to utilize the information available at hand at that instant.

[5]By "coming in contact with a name" we mean that i comes in contact with another node, say j, which has data corresponding to `name` in j's CS.

[6]Somewhat similar to route aggregation in the Internet.

a node receives a name via an interest message for which it has the corresponding data available in its CS; h_i denotes the interest hits count of node i, 2) **Size of PIT**, which is the number of entries in the PIT of a node; p_i denotes the size of i's PIT, 3) **Size of FIB**, which is the number of entries in FIB; f_i denotes the size of FIB of i, and 4) **ICT**, which is the average time interval between two successive contacts for any given pair of nodes; τ_i denotes the average ICT of i. We use the exponential weighted moving average method to keep a running estimate of τ_i. In particular, if Δt be the recent ICT for a concerned pair of nodes, then the average ICT value is updated as $\tau_i = \alpha\tau_i + (1-\alpha)\Delta t$.

A higher value of interest hits indicates higher capacity of a node to satisfy incoming interests. Again, the larger the size of the FIB of a node, the more routing information is carried by it. On the other hand, larger size of PIT indicates that the concerned has received more number of interests from possibly a larger number of nodes and therefore, perhaps can route an incoming interest to those diverse nodes. Finally, ICT plays a critical role in shaping the communication pattern in OMNs and consequently, in routing. We also considered few other metrics, e.g., contact frequency, but did not find any significant effect. Moreover, except τ and λ, these parameters are already available with a node and do not incur any further overhead for storage and computation. These justify the rational behind choosing the above mentioned metrics.

C. Replication Strategy and Dynamic TTL Adaption

When a node having an entry for `name` in its PIT comes in contact with another node whose ICT is not greater than the former, it faces with two alternatives – 1) S: Send (replicate) the interest to the other node or 2) X: Do nothing, i.e., do not replicate the interest – which are characterized by the previously discussed attributes. Let $\lambda_i(\texttt{prefix})$ be the last contact time of node i with `name`. Now if i chooses alternative S, the interest is replicated to j, which then "experiences" the characteristics $(\lambda_j(\texttt{prefix}), h_j, p_j, f_j)$ pertaining to node j. However, if i chooses X, the interest remains with i with the profile $(\lambda_i(\texttt{prefix}), h_i, p_i, f_i)$. Let, $R_S \equiv$ regret of having the interest message with j rather than i and $R_X \equiv$ regret of having the message with i rather than j. Thus, following [11], the regrets associated with these choice involving interest replication are evaluated as:

$$\left.\begin{aligned} R_S &= \ln(1+\exp[-\beta_\lambda\delta(\texttt{prefix})]) + \ln(1+\exp[-\beta_h\delta_h]) \\ &\quad + \ln(1+\exp[-\beta_p\delta_p]) + \ln(1+\exp[-\beta_f\delta_f]) \\ R_X &= \ln(1+\exp[\beta_\lambda\delta(\texttt{prefix})]) + \ln(1+\exp[\beta_h\delta_h]) \\ &\quad + \ln(1+\exp[\beta_p\delta_p]) + \ln(1+\exp[\beta_f\delta_f]) \end{aligned}\right\} \tag{1}$$

where $\delta(\texttt{prefix}) = \lambda_j(\texttt{prefix}) - \lambda_i(\texttt{prefix})$, $\delta_h = h_j - h_i$, $\delta_p = p_j - p_i$, and $\delta_f = f_j - f_i$. We impose an additional constraint here so that the regret associated with replicating an interest is "historically" reduced (similar to [12]). In particular, let $r(\texttt{name})$ be the value of R_S in the previous replication action. The interest is replicated only if R_S is less than both R_X and $r(\texttt{name})$. Subsequently, $r(\texttt{name})$ is set to R_S.

It may be noted that the constants β_λ, β_h, β_p, and β_f would be positive or negative depending upon the concerned attributes. Moreover, their dimensions are inverse of the corresponding attributes. E.g., if λ is in seconds, then that of β_λ is seconds^{-1} so that $\beta_\lambda[\lambda_j(\texttt{prefix}) - \lambda_i(\texttt{prefix})]$ is dimensionless.

To mitigate issues of excessive message replication that increases overhead, CHARM dynamically adapts the TTL of interest and data messages. Specifically, let T be the TTL assigned[7] to a message during its creation. After *each replication* of the message, T is scaled down by a factor ξ, where $0 < \xi < 1$ so that after k replications, its updated TTL becomes $\xi^k T$. However, in the absence of any replication, the TTL remains unaltered. We assume that a TTL is represented as an integer, so that on scaling down, it converges to zero in a finite number of steps. This would have been difficult had TTL been a real number.

Algorithm 1 summarizes the operations of CHARM. Line numbers 3–5 indicate that if node j has a data corresponding to the name[8], then the interest message is replicated by i to j. Otherwise, a decision is made based (1). Finally, TTL of the replicated message is decayed in line 10. The time and space complexities of the Algorithm are $O(|M|)$, where M is the set of messages created in the OMN.

Algorithm 1: Interests replication by node i to j

```
1  for each msg ∈ Interest messages stored by i do
2      name = Name contained in msg
3      if name ∈ PIT of i then
4          if j has data corresponding to name then
5              Replicate msg to node j
6          else if τ_j < τ_i then
7              Compute R_S and R_X using (1)
8              if R_S < R_X and R_S < r(name) then
9                  Replicate msg to node j
10                 T = ξ_i T // T is the TTL of msg
11                 r(name) = R_S
```

III. Simulation and Results

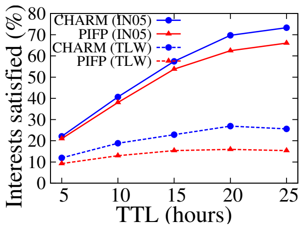
We implemented and tested the performance of CHARM using the Information Centric Opportunistic Network Environment (ICONE) simulator [2]. Interest messages were created by the nodes every 45–95 seconds; sizes of data messages were uniformly distributed between 250 and 350 KB. We used the Infocom'05 (IN05) [13] real-life connection traces collected from 40 users as well as the truncated Levy Walk (TLW) [14] mobility model with 40 nodes in our simulations. The transmission ranges of the devices were about 30 m and 100 m, respectively, in the two scenarios. We compared CHARM with PIFP [2]. Table I shows the values of different parameters used by CHARM. These values were empirically chosen based on the relative performance tradeoffs obtained in different scenarios.

[7] T indicates how long a message should remain valid since its creation before it can be dropped.

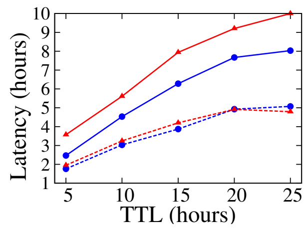
[8] It can be shared during handshaking in the communication initiation phase.

TABLE I: Values of Parameters used by CHARM

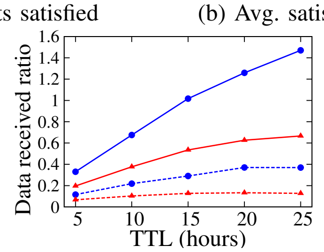
Parameter	Value	Parameter	Value
β_λ	0.10	β_h	0.60
β_p	0.40	β_f	0.60
ξ_d	0.45	ξ_i	0.50
γ	0.99	α	0.70



(a) Interests satisfied



(b) Avg. satisfaction latency



(c) Data messages received

Fig. 1: Effects of TTL on different performance metrics.

Fig. 1 (a) shows that CHARM satisfied more interests than PIFP in both the IN05 and TLW scenarios. An increasing trend in performance w.r.t. TTL can also be observed. In particular, in the TLW scenario, CHARM offered about 1.6x times improvement over PIFP when TLL = 25 hours. Fig. 1 (b) shows that CHARM reduced the interest satisfaction latency[9] w.r.t. PIFP in almost all the instances. An increasing trend in latency is observed because, while more interests were satisfied as TTL increased, the corresponding latency profiles were also diversified. Consequently, their average went up.

Fig. 1 (c) shows that the number of data messages (replicated and) received by the nodes per unit number of interests generated increased when CHARM was used as compared to PIFP. Although having lower overhead is desirable, the observations here should not be interpreted in isolation. Redundancies in transmission of data messages actually helped CHARM in achieving huge performance gains as noted earlier. Moreover, such higher overhead also indicates that the OMN has far more communication capacity available than that was used by PIFP and therefore, the performance-overhead trade-off in content searching in OMNs can be further improved.

Experimental results also revealed that compared to PIFP, the average size of FIB of the nodes using CHARM was about 6x times lower for the IN05 scenario. This is significant since both PIFP and CHARM involve nodes mutually exchanging their FIBs at every contact. In this scenario, on using dynamic TTL adaptation, the ratio of data messages received was found to be reduced by about 19–39%.

[9]The time interval between requesting a content and receiving the corresponding data averaged over all the satisfied interests.

IV. Conclusion

In this work, we presented CHARM, a scheme for named content searching in OMNs using the RRM technique, where a decision to (or not to) replicate a content request to another node is based upon the regrets obtained using different aspects related to content and the network. Moreover, to reduce overhead arising due to message replications, CHARM introduces the concept of dynamic TTL adaptation, where the TTL of a message is repeatedly scaled down. Results of performance evaluations indicate that CHARM can vastly improve the number of satisfied interests and their average satisfaction latency as compared to PIFP at the cost of additional number of message replications. In the future, it would be interesting to investigate how further we can reduce message replication redundancy while increasing the number of interests satisfied.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[2] P. Duarte, J. Macedo, A. Costa, M. Nicolau, and A. Santos, "A probabilistic interest forwarding protocol for named data delay tolerant networks," in *Ad Hoc Networks*. Springer, 2015, vol. 155, pp. 94–107.

[3] Y. Lu, X. Li, Y.-T. Yu, and M. Gerla, "Information-centric delay-tolerant mobile ad-hoc networks," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, April 2014, pp. 428–433.

[4] Q. Xu, Z. Su, K. Zhang, P. Ren, and X. S. Shen, "Epidemic information dissemination in mobile social networks with opportunistic links," *IEEE Trans. on Emerging Topics in Computing*, vol. 3, no. 3, pp. 399–409, Sept 2015.

[5] T. Han and N. Ansari, "Opportunistic content pushing via WiFi hotspots," in *3^{rd} IEEE International Conference on Network Infrastructure and Digital Content*, Sept 2012, pp. 680–684.

[6] K. Zhang, X. Liang, R. Lu, and X. Shen, "PIF: A personalized fine-grained spam filtering scheme with privacy preservation in mobile social networks," *IEEE Trans. on Computational Social Systems*, vol. 2, no. 3, pp. 41–52, 2015.

[7] Y. Xu, A. Anpalagan, Q. Wu, L. Shen, Z. Gao, and J. Wang, "Decision-theoretic distributed channel selection for opportunistic spectrum access: Strategies, challenges and solutions," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1689–1713, 2013.

[8] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, pp. 263–291, 1979.

[9] B. K. Saha, S. Misra, and S. Pal, "Utility-based exploration for performance enhancement in opportunistic mobile networks," *IEEE Trans. on Computers*, vol. 65, no. 4, pp. 1310–1322, 2016.

[10] G. Loomes and R. Sugden, "Regret theory: An alternative theory of rational choice under uncertainty," *The Economic Journal*, vol. 92, no. 368, pp. 805–824, Dec. 1982.

[11] C. G. Chorus, "A generalized random regret minimization model," *Transportation Research Part B: Methodological*, vol. 68, pp. 224–238, 2014.

[12] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proc. of ACM MobiHoc*. New York, NY, USA: ACM, 2008, pp. 251–260.

[13] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-01-31)," Downloaded from http://crawdad.org/cambridge/haggle/, Jan. 2006, [Accessed: 22 Apr. 2015].

[14] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy walk nature of human mobility: Do humans walk like monkeys?" *IEEE/ACM Trans. on Networking*, vol. 19, no. 3, pp. 630–643, Jun. 2011.