Optimal Caching for Producer Mobility Support in Named Data Networks

Hesham Farahat
ECE Department, Queen's University
Kingston, ON, Canada
h.farahat@queensu.ca

Hossam Hassanein
School of Computing, Queen's University
Kingston, ON, Canada
hossam@cs.queensu.ca

***Abstract*—Named Data Networks (NDNs) offer a promising paradigm for the future Internet to cope with the growing demand for data. One of the main challenges in NDNs is how to support a seamless operation during mobility. In this paper, we investigate optimal caching for Producer mobility support and propose a scheme (named *OpCacheMob*) that exploits location predictors and data requests' patterns to cache the data proactively before handover occurs. In essence, *OpCacheMob* adopts the predicted future Interests, that will be sent to the mobile producers, and caches their data contents ahead. Thus, avoids Interest retransmission or redirection that increase the consumer's delay and decreases the network efficiency during producer's mobility. We provide a mathematical formulation for such caching problem that bounds both the cache update cost and the consumer delay while minimizing the total network overhead due to the change of content availability. *OpCacheMob* is then implemented in ndnSIM and evaluated against mainstream NDN mobility solutions. We demonstrate how the scheme can be used as a benchmark to measure the performance of other mobility schemes. In addition, a sensitivity analysis is presented to measure the impact of errors on the prediction gain of such solution.**

I. Introduction

The evolution of content, devices and applications caused a shift in the way the Internet operates. With an increasing use of the Internet, CISCO's Virtual Networking Index (VNI) [1] predicts that the global traffic will reach 164 Exabyte per month by 2019, nearly a threefold increase over 2014 statistics. Since the Internet was built on a host-to-host model, patching it up is the current solution to maintain scalable operations. Overlays such as Content Distribution Network (CDN) and Peer-to-peer (P2P) are examples of such patching protocols, which are bound to fail under the projected future traffic [2].

Recent research efforts have been directed at designing a content-based paradigm for the new Internet, namely Information-centric Network (ICN). ICN is proposed to address the growing traffic challenges with content-oriented application [3]. One of the main features of ICN is the in-network caching, where content can be stored near the users for shorter access time and less congestion [4], [5]. Several potential ICN architectures have been proposed while considering the strength and weaknesses of the current design. Named Data Network (NDN) [2] is one of the pioneering ICN designs, that was proposed by PARC. The main goal of NDN is to evolve the role of the IP architecture such that the packets can name content not hosts.

A core feature of any future Internet design is supporting seamless mobility as a network primitive. This is a requirement to cope with the projected growth in mobile nodes and content, where wireless and mobile devices will account of 66% of IP traffic by 2019 [1]. The challenge of mobility in NDN is in how to find and trace content instead of hosts as in the current Internet. Additionally, supporting seamless operations, where users are able to move in the network without impacting applications' performance, is essential in NDN. However, the implicit assumption that mobility is supported intrinsically is impractical. Therefore, a design of a mobility management scheme is inevitable to successful NDN.

Several efforts have addressed the producer mobility challenge in ICN generally, and NDN specifically. Whereas most of the proposed schemes use reactive techniques to recover after a mobile event, in this paper, we propose an optimal caching scheme to support seamless producer mobility. The scheme, called *OpCacheMob*, exploits position prediction techniques in addition to predicting users' access patterns to proactively store potential data on in-network caches. Thus, Interests generated during a producer mobility event can be satisfied by the intentionally placed data items. Finding the best placement that guarantees no Interests drops and with bounded overhead has the potential to provide an optimal proactive solution to the producer mobility problem. We implement the scheme in ndnSIM and integrate it with Gurobi [6] to provide the optimal solution. We use the assessment framework in [7] to evaluate the effectiveness of the proposed scheme. This is in addition to measuring the gap in performance between the optimal caching approach and other mobility schemes.

The remainder of the paper is organized as follows. In Section II, an overview of the mainstream NDN mobility management schemes is introduced. The formulation of the optimal caching is in Section III. The simulation model is explained in detail in Section IV. Our simulation experiments are discussed in Section V. We conclude our findings in Section VI and present insights into future directions in NDN mobility support.

II. Related Work

NDN is one of the proposed architectures for the future ICN, where communication is receiver-driven. Since there are

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

no IP addresses in NDN, data is requested by sending Interest packets with its hierarchical and unique name. We refer to the requester as Consumer and the data owner as Producer. Every node in NDN has two structures to serve forwarding two types of packets, Interests and Data. The Forwarding Information Base (FIB) is used to forward Interests to Producers, which has a similar functionality of the IP table in the IP network. The second structure is the Pending Interest Table (PIT), which keeps track of unsatisfied Interests and route back Data to Consumers. To support in-network caching in NDN, a structure called Content Store (CS) is available in NDN nodes.

Given that the mobility events are inevitable, seamless mobility support is a requirement in the future Internet. However, there exists no scheme to support mobility in NDN. It relies on Interests' retransmission to recover from such events. Specifically, in the case of a mobile consumer, the requester will retransmit unsatisfied Interests again after connecting to the new PoA. On the other hand, Producer mobility requires routing state updates to be able to reach the producer in its new location. However, this solution is not scalable and will result in Interest retransmissions during the convergence time. Therefore, a Producer mobility scheme is needed to replace the current proposed design in NDN.

Multiple mobility management schemes have been proposed in the literature focusing on Producer mobility since it is more critical than Consumer mobility. There are two mainstream solutions: Mobility Anchor and Location Resolution approaches. Detailed discussion and performance comparison of the different approaches were presented in [7]. Mobility anchor schemes [8], [9] are based on MobileIP protocol [10] used in the current Internet. It uses special nodes in every home network called anchors that forward the Interests to the Producer even if the latter is moving to another network (roaming). In this case, the Interests will not be dropped but it may take longer paths to reach the producer. The approach in Location Resolution schemes [11], [12] is similar to the one used in Domain Name Systems (DNSs), where the Consumer queries the location of the Producer before sending Interests.

A class of Consumer mobility management schemes is worth mentioning, since it considers proactive methods. The technique used utilizes the caching feature in ICNs to store the data requested by a moving Consumer in its 1-hop neighbor. This approach [13], [14] has the limitations of caching only at access routers, having many retransmissions and being designed for consumer mobility only. The scheme proposed by T. Woo in [15] is designed for Producer mobility where it pushes the data to the best 1-hop neighbor of the producer. However, its performance is suboptimal when mobile devices switch networks (i.e., vertical handover). Moreover, considering neighbors only will limit the resource utilization to successfully manage producers mobility.

III. The Optimal Solution *OpCacheMob*

The problem in Producer mobility is that during the convergence time, the network is not updated with the new location of the producer. Hence, any Interest that is directed to this producer's old Point of Attachment (PoA) will be dropped, unless it finds the data in one of the caches along the path from the Consumer to the Producer. The basic idea of *OpCacheMob* is to proactively place the data that will be requested during the convergence time in the caches before the handover occurs. In order to make such decision, two pieces of information are required prior to that. The first is the time of handover of a moving Producer which can be predicted using location estimators [16]–[18]. The location predictor will estimate the new position of the node at a specific time. If the change in location results in a change in the PoA (e.g., node is closer to another AP), a handover is detected.

The second required information is to know what data will be requested from the mobile producer during the convergence time. Request's pattern predictors such as in [19]–[21] can be used to find the time, data and Consumer of a future request.We can then find the potential Interests to be dropped after a producer mobility event, and take the necessary action in advance to cache the data. We assume such information to be available and correct (i.e., perfect knowledge of the future), to find an optimal solution to the data caching problem.

Assuming the handover event takes τ seconds to finish, which includes the time needed by the Producer to connect to the new AP and the convergence time needed by the network to update the routing state. Hence, the Interests directed to a mobile producer and requested between $t_{HO}-T$ and $t_{HO}+\tau$, where t_{HO} is the time of handover and T is the average time to reach the producer in its old PoA, are predicted to be dropped and will be retransmitted. Given this set of requests, we can define the problem as finding the optimal placement of data in network caches such that no retransmissions of Interests are required and with bounded overhead on the network.

A. *System Model*

Let U be the set of users in the network. Every user $u \in U$ can consume or produce data $d \in D$, where D is the list of all data items that can be produced by all producers. The two binary indicator variables i_u^d, o_u^d define the role of user u for data d. Specifically, i_u^d is 1 if the user u requested the data d or 0 otherwise. Similarly, o_u^d is 1 if user u produces data d.

Let the set of routers be R, every $r \in R$ has a maximum cache capacity of c_r^{max} and current capacity denoted by c_r. The network topology is represented by a graph $G = (V, E)$, where nodes in V are either users, APs or routers. The path taken by a packet (Interest or data) from node x to node y is denoted by a sequence of nodes $P_{x \to y}$. Accordingly, the cardinality $|P_{x \to y}|$ represents the number of hops traversed to reach the destination y from the source x.

Each Data d has a content value ϵ_d which can represent content's popularity, Producer's quality, application type or user's priority. In this work, ϵ_d is assumed to be the popularity of an item d (i.e., probability that d will be requested next).

B. *Problem Formulation*

As mentioned earlier, caching the data required in advance will allow the Interests to be satisfied on the first attempt.

However, this comes with an overhead which can be of three types:

1) Path updates: Data can be placed on any cache in the network. Caching on a router that is not on the path from a Consumer to a Producer will require a route to be created temporarily from the Consumer to the router chosen. Hence, an overhead of updating the FIBs of the new path is needed.
2) Cache cost: Adding data to a full cache will replace some other items, based on the replacement policy in place. Both the total number and value of data removed should be bounded to avoid affecting other users.
3) Producer traffic: The mobile producer will be required to handle the proactive requests before the handover in addition to its regular load. This traffic is an overhead on the Producer and should be bounded.

Finding the best possible placement of data with bounded overhead can be formulated as an optimization problem as follows:

$$\min_{\delta_r^d,\rho_r^m} \sum_{d\in D}\sum_{u\in U} i_u^d \sum_{u'\in U} o_{u'}^d \sum_{r\in R}(|P_{u\to r}| - |P_{u\to u'} \cap P_{u\to r}|)\delta_r^d$$

S.t.

$$\sum_{r\in R} \delta_r^d = 1 \quad \forall d \in D \tag{C1}$$

$$\sum_{d\in D}\sum_{u'\in U}\sum_{r\in R} o_{u'}^d |P_{u'\to r}|\delta_r^d < \zeta \tag{C2}$$

$$\sum_{r\in R} i_u^d |P_{u\to r}|\delta_r^d < \gamma \quad \forall d \in D, \forall u \in U \tag{C3}$$

$$\sum_{m=0}^{c_r} \rho_r^m = 1 \quad \forall r \in R \tag{C4}$$

$$c_r + \sum_{d\in D} \delta_r^d - \sum_{m=0}^{c_r} m\rho_r^m \leq c_r^{max} \quad \forall r \in R \tag{C5}$$

$$\sum_{d\in D} \delta_r^d - \sum_{m=0}^{c_r} m\rho_r^m \geq 0 \quad \forall r \in R \tag{C6}$$

$$\sum_{m=0}^{c_r} m\rho_r^m < \alpha \quad \forall r \in R \tag{C7}$$

$$(\sum_{m=0}^{c_r} \rho_r^m k_r^m - \sum_{d\in D} \epsilon_d.\delta_r^d)/K_r < \beta \quad \forall r \in R \tag{C8}$$

$$c_r^{Sp} + \sum_{d\in D} \delta_r^d \leq c_r^{SpMax} \quad \forall r \in R \tag{C9}$$

The decision variable δ_r^d is 1 if the solution decides to cache data d on router r. The second decision variable is ρ_r^m, which decides on how many items should be removed from the cache in order to cache the new data. In particular, ρ_r^m is 1 of m items from router r should be removed before the action of placement is to be made.

The objective function is to minimize the number of path updates since it is the most challenging overhead. The number of path updates for each data d is the number of non-common edges between $P_{u\to u'}$ (which is the path from the consumer u to producer u') and $P_{u\to r}$ (which is the new path from the consumer u to the chosen router r).

Constraint (C1) ensures that all data will be cached once in any of the routers. Constraint (C2) bounds the amount of traffic generated to support mobility (the third type of overhead). Specifically, the summation of the number of hops the data required to be cached has to be less than a threshold ζ. Constraint (C3) bounds the distance between consumers to the chosen routers. In particular, the number of hops between the consumer u and the chosen router r has to be less than γ. This is added to ensure that the delay to reach the data in the new location is less than the delay requirements specified by the Consumer's application.

Constraints (C4) to (C9) are related to caching and data replacement. To ensure that the number of items in the cache does not exceed the max capacity, (C5) is used. Constraint (C6) ensures that no replacements are needed more than the amount placed. To bound the number of replacement per router, (C7) is used with a threshold α. Since some content will be replaced, the total content value of the router will change. The reduction of this value is bounded by a threshold β in Constraint (C8). The value of variable k_r^m is the total content value of m items to be replaced from router r, and K_r is the total content value of all the items in r.

The objective function and all the constraints are linear. Moreover, the value of the decision variables are either 0s or 1s. Hence, the optimization problem is a 0-1 Integer Linear Program (0-1 ILP).

C. *Special Cache Design*

The data will be cached in general content stores, which may get replaced with other content. Hence, the data must be treated differently so that it is not replaced by other content before the consumer makes the request. Therefore, a split cache design is proposed to serve this purpose. The split cache is a regular cache with a reserved space for mobile requests. The special part can be expanded to store more special data as long as it does not exceed the maximum capacity allocated to it (C_r^{SPMax}). Since the data in the special cache should not be replaced, the amount of data to be stored in a router should not exceed the number of empty slots in the special cache. Constraint (C9) enforces this condition, where C_r^{SP} is the number of items in r's special cache.

IV. Simulation Model

A. *Benchmark Tool*

To implement the optimal solution and evaluate its performance, we use our assessment framework proposed in [7]. The benchmarking tool is designed to evaluate Producer mobility management schemes in various scenarios using multiple performance metrics. The tool consists of four major blocks; Mobility, Topology and User Planes and the NDN network simulator which is based on ndnSIM [22].

TABLE I
SIMULATION PARAMETERS

	Parameter	Value
General	Simulation Duration	1000s
	Transit Period	80s
	Map size	$1400m \times 1400m$
	Number of Blocks	7×7
	Number of Users	100
	Producers	50
	Consumers	50
Application	Interest Rate	50-80 *I*/s
	Zipf α	0.2
	Content per Producer	$1000 \times 1KB$
Topology	APs	49
	AP Range	200m
	Number of Routers	40
	Core router's links	10Mbps
	Access router's links	5Mbps
	Propagation delay	10ms
Mobility	Model	Manhattan
	Handover delay	0.5s
	Speed	70 km/h
NDN	Forwarding Scheme	BestRoute
	Cache replacement	LRU
	Cache Size	1000 objects

B. Implementation of the Optimal Solution

The *OpCacheMob* is implemented into the simulator as a new Mobility Management Module. The new module will add the following to support the optimal solution: 1) New content store to handle the Split cache design proposed earlier. 2) Mechanism to add and update FIB entries to reach off-path routers. 3) Mechanism to read the current state of the network and create the optimization model in Gurobi [6] based on the formulation in the previous section.

The time horizon will be split into small time slots. At the beginning of every slot, *OpCacheMob* will run to support producer mobility during the complete slot. Since perfect knowledge is assumed for finding the optimal solution, the new module will use the information provided from the Mobility and User planes to find the set of Interests that will be dropped.

Given the set of Interests and the current network status, the Gurobi model is created and the solver runs to find the best placement of the data corresponding to the Interests input. For every $\rho_r^m = 1$, m items will be removed from router r, and for every $\sigma_r^d = 1$, data d will be cached in router r. Moreover, if r is not on the path from the Consumer to the Producer (i.e., $r \notin P_{u \to u'}$), FIB entries are added to every non-common router in $P_{u \to r}$.

V. RESULTS AND DISCUSSION

A. Experiment Setup

The simulation parameters of the experiments are summarized in Table I. There are 50 Consumers and 50 Producers moving in 7x7 street grid plan (such as Manhattan). The Consumers send requests with a rate of 50-80 request/s following a popularity distribution Zipf with $\alpha = 0.2$. The topology used is hierarchal with 40 core routers distributed in 5 domains. Every core router has a cache with a capacity of 1000 items and uses Least Recently Used (LRU) replacement policy.

For every experiment one factor is varied to test the impact of it on the performance of the scheme. *OpCacheMob* is controlled by the optimization window (set to 2 seconds) and four thresholds to bound the overhead. After tuning the parameters for best results, the thresholds are set to the following:

1) ζ: controls how far the data will travel from the producer side. The value is calculated using the current state of the network and it is based on the average number of hops needed to reach the Producer from the Consumer. Hence, the chosen router will not be far more than the average hop count.
2) δ: controls how far the data will be placed from the Consumer side. Similar to the previous threshold, it is based on the average number of hops to reach the producer. Thus, the resulted delay will not be longer than the one to reach the Producer.
3) α: the maximum number of replacements per router is set to 10% of the cache size.
4) β: the total content value of a router should not decrease more than 10% of its total value.

We execute each experiment for 10 runs with different random seeds, then the averages of the metrics were calculated. The 95% confidence interval has a maximum deviation of 5% from the reported average values.

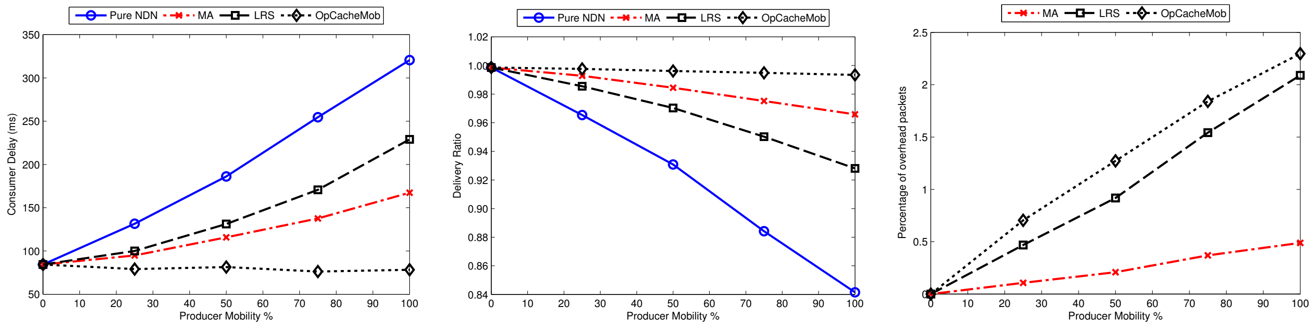
B. Schemes and Evaluation Metrics

In the evaluation, we evaluate *OpCacheMob* and compare it with three mobility management approaches. Namely, NDN with no scheme (Pure-NDN), Mobility Anchor (*MA*) represented by [8] and Location Resolution Scheme (*LRS*) represented by [11]. Additionally, We consider the following three main metrics that reflect the satisfaction of both the Consumer and the network :

1) Consumer Delay: Calculated as the time difference between the first attempt of sending the Interest and successfully receiving the Data at the consumer. This includes the total time out period and the delay of both the retransmitted Interests and Data packets.
2) Delivery Ratio: is the proportion of successful Data packets received by the Consumer to the total number of Interests sent. This metric is a measure of how successful is the scheme in avoiding both Interest and Data drops.
3) Overhead: Calculated as the percentage of total number of control packets generated by the scheme to the total number of Interests.

C. Producer Mobility

The first scenario evaluates all the mobility management schemes under different percentages of mobile producers in the network. The results in Fig. 1a and Fig. 1b demonstrate the ability of the introduced *OpCacheMob* to proactively cache the future content of the mobile producer. Therefore, avoids retransmission of Interests to the new location and attained the maximum delivery ratio irrespective of mobile producers percentage. Moreover, the optimal placement of the content near the consumer's current location resulted in the minimal delay that is also stable over the percentage of mobility events as shown in Fig. 1a. These optimality conditions required an extra overhead, shown in Fig. 1c, for delivering the mobile producer's content to the selected cache and also send path update packets in the network. Accordingly, *OpCacheMob*



(a) Consumer Delay (b) Delivery Ratio (c) Overhead Percentage

Fig. 1. Comparing the performance of *OpCacheMob* with Pure-NDN, *MA* and *LRS* by varying the percentage of Producer mobility

provides lower and upper bounds for Consumer's delay and delivery ratio, respectively.

As such, the results revealed significant deterioration of the total delay which increased by more than 100%, in case of the pure-NDN, when only 50% of the Producers change their location Fig. 1a. This is attributed to the simplicity of NDN that depends on Interest retransmission to recover from Producer Mobility and thus suffers from a poor delivery ratio shown in Fig. 1b especially at higher percentages of mobile Producers. The existing non-predictive *MA* and *LRS* attain acceptable delays at low producer's mobility (i.e., 25%), redirecting the Interests to the new locations is suboptimal at more dynamic scenarios. Thus, when 75% of Producers are mobile, consumers suffered from 90% and 65% increase in the delay compared to the lower bound, *OpCacheMob*, using *LRS* and *MA*, respectively as shown in Fig. 1a. This delay is also associated with an increased number of Interest retransmissions when the original Interests become outdated for the consumers (i.e., after the time out). Thus, a drop in the delivery ratio is also experienced by the network under these non-predictive techniques as shown in Fig. 1b.

While *OpCacheMob* evaluated the pure-NDN, *MA* and *LRS*, the overhead gap between the *OpCacheMob* and each of these schemes provides a room for enhancing their performance. In particular, these non-predictive mobility schemes can utilize the available network capacity (i.e., increase overhead) in order to decrease the delay and delivery ratio.

D. Sensitivity Analysis

The above performance of the proposed *OpCacheMob* was evaluated under perfect knowledge of both producers' mobility and consumers' requests. We therefore perform sensitivity analysis to evaluate the effect of prediction errors on the aforementioned gains.

1) Imperfect Producer Mobility Prediction: For this type of error, a uniform distributed random number is added to the producer's location record (i.e., node, time and PoA), thus the Producer's PoA might be falsely predicted and one of the neighboring PoAs is selected instead.

For a mobile Producer, the predictor will estimate that such Producer will either remain in current PoA (i.e., handover

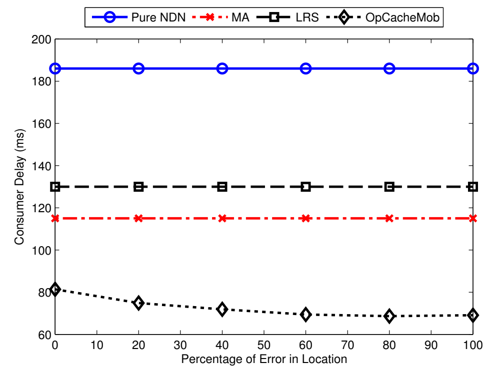


Fig. 2. Delay in 50% mobility and errors in location predictor

missed) or moved to one of the neighbouring PoAs (i.e., handover performed). Given the methodology of the *OpCacheMob*, the former case is the only situation that impacts the performance since some Interests will not be considered in the next optimization window (i.e., no caching for the Interests of undetected mobile users). The probability of this event is calculated as $(p/N)^L$ where p is the probability of producer's location error, N is the average number of neighbours for each PoA and L is the length of the prediction window. In the second case (i.e. handover performed), there is no effect on the scheme even when the new PoA of the mobile producer is incorrectly estimated. This is because *OpCacheMob* caches the content of mobile producers based on their previous locations (to avoid overhead) irrespective of their exact future locations. On the other hand, for a non-mobile Producer, the predictor will either correctly estimate that such Producer is static or moving to a neighboring PoA (i.e., false handover). The only impact of the second case is the extra overhead generated by an unnecessary optimization.

The results in Fig. 2 and 3 demonstrate that the mobility prediction errors have no significant effect on Consumer delay and Delivery ratio since the probability of handover missed event is low (i.e., $(p/3)^2$ in our case). However, an increase in the overhead is observed due to the false handover case discussed above.

2) Imperfect Consumer Request Prediction: This scenario is simulated by adding a uniform distributed random error to the Consumer's predicted Interest. Thus, a new wrong Interest

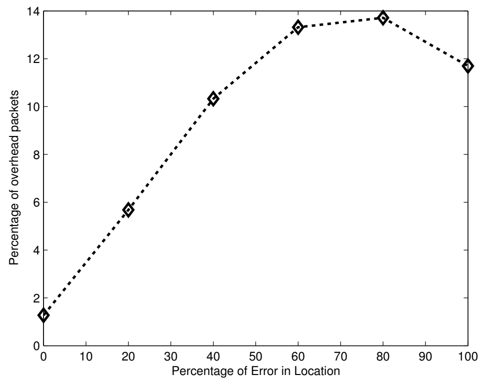


Fig. 3. Overhead in 50% mobility and errors in location predictor

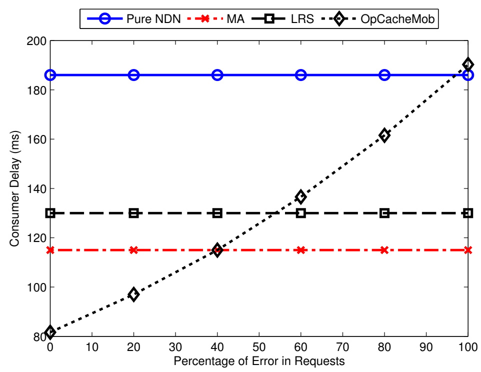


Fig. 4. Delay in 50% mobility and errors in requests' predictor

will be predicted and used in the optimization, while the actual Interest is ignored. The results in Fig. 4 demonstrate the ability of the *OpCacheMob* to provide the lowest consumer delay up to 40% of wrong content prediction. A gradual deterioration is then observed when larger percentages are of wrong contents are predicted. As such the *OpCacheMob* will fail to proactively cache the content of the mobile producers and will rely on Interest retransmissions instead.

VI. Conclusions and Future Work

Supporting seamless mobility in NDN is essential for reliable network operation under both Consumer's and Producer's mobility. This paper designed and implemented an optimal producer mobility management scheme *OpCacheMob* that exploits the available in-network caching resources, and both location and data predictions. Under perfect knowledge, the results demonstrated the ability of *OpCacheMob* to proactively cache the future content of the anticipated mobile producers. While compromising the network overhead, both the Consumer delay and delivery ratio remained almost constant, regardless of the number of mobility events as long as there is enough space in the content store. Such results provide a novel benchmark for the main mobility solutions in the literature. This is in addition to provisioning insights on the trade-off between the prediction effort and overhead, on one hand, and the consumer delay and delivery ratio on the other hand. Moreover, sensitivity analysis proved the robustness of the introduced *OpCacheMob* to imperfect mobility predictions and suboptimal traffic forecasts.

Our future work will consider the applicability of the predictive mobility management scheme in practice. This includes studying the computational complexity and scalability of the *OpCacheMob* and introduce guided heuristic techniques to provide real-time near-optimal caching decisions.

References

[1] "Cisco visual networking index: Forecast and methodology, 2014–2019," White Paper, Cisco, May 2019.

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

[3] G. M. Brito, P. B. Velloso, and I. M. Moraes, *Information Centric Networks: A New Paradigm for the Internet*. Wiley, 2013.

[4] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[5] S. Abdelhamid, H. Hassanein, G. Takahara, and H. Farahat, "Caching-assisted access for vehicular resources," in *IEEE Conf. Local Comput. Networks (LCN)*, 2014, pp. 28–36.

[6] Gurobi, "Gurobi optimizer reference manual," http://www.gurobi.com, 2015, accessed on February 28, 2016.

[7] H. Farahat and H. Hassanein, "On the design and evaluation of producer mobility management schemes in named data networks," in *ACM Int. Conf. Modeling, Anal. and Simulation Wireless and Mobile Systems(MSWIM)*, Cancun, Mexico, 2015, pp. 171–178.

[8] J. Lee, S. Cho, and D. Kim, "Device mobility management in content-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 28–34, 2012.

[9] D. Han, M. Lee, K. Cho, T. Kwon, and Y. Choi, "Publisher mobility support in content centric networks," in *Int. Conf. Inform. Networking (ICOIN)*, Phuket, Thailand, Feb 2014, pp. 214–219.

[10] E. C. Perkins, "IP mobility support for IPv4, revised," RFC 5944, 2010.

[11] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom, "End-to-end mobility support in content centric networks," *Int. J. Commun. Syst.*, vol. 28, no. 6, pp. 1151–1167, 2015.

[12] A. Azgin, R. Ravindran, and G. Wang, "A scalable mobility-centric architecture for named data networking," *CoRR*, vol. abs/1406.7049, 2014. [Online]. Available: http://arxiv.org/abs/1406.7049

[13] X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis, "Proactive selective neighbor caching for enhancing mobility support in information-centric networks," in *Proc. ICN Workshop Information-centric networking*, Helsinki, Finland, 2012, pp. 61–66.

[14] V. Siris, X. Vasilakos, and G. Polyzos, "Efficient proactive caching for supporting seamless mobility," in *IEEE Int. Symp. WORLD WIRELESS, MOBILE AND MULTIMEDIA NETWORKS (WoWMoM)*, Lucca, Italy, Jun. 2014, pp. 1–6.

[15] T. Woo, H. Park, S. Jung, and T. Kwon, "Proactive neighbor pushing for enhancing provider mobility support in content-centric networking," in *Int. Conf. Ubiquitous and Future Networks (ICUFN)*, Shanghai, China, Jul. 2014, pp. 158–163.

[16] N. Samaan and A. Karmouch, "A user centric mobility prediction approach based on spatial conceptual maps," in *IEEE Int. Conf. Commun. (ICC)*, vol. 2, Seoul, Korea, May 2005, pp. 1413–1417.

[17] R. Atawia, H. Abou-zeid, H. Hassanein, and A. Noureldin, "Chance-constrained QoS satisfaction for predictive video streaming," in *IEEE Conf. Local Comput. Networks (LCN)*, 2015, pp. 253–260.

[18] H. Abu-Ghazaleh and A. Alfa, "Application of mobility prediction in wireless networks using markov renewal theory," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 788–802, Feb. 2010.

[19] X. Chen and X. Zhang, "A popularity-based prediction model for web prefetching," *IEEE Computer*, vol. 36, no. 3, pp. 63–70, Mar 2003.

[20] R. Atawia, H. Abou-zeid, H. Hassanein, and A. Noureldin, "Joint chance constrained predictive resource allocation for energy-efficient video streaming," *IEEE J. Sel. Areas Commun.*, 2015, in Press.

[21] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: Discovery and applications of usage patterns from web data," *SIGKDD Explorations Newslett.*, vol. 1, no. 2, pp. 12–23, Jan. 2000.

[22] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, Oct. 2012.