Hop-count Based Forwarding for Seamless Producer Mobility in NDN

Vignesh Sivaraman and Biplab Sikdar
Department of Electrical and Computer Engineering
National University of Singapore, Singapore

***Abstract*—Many future Internet architectures have been proposed to address issues like increasing traffic, mobility, efficient content dissemination and Named Data Networks (NDN) is emerging as one of the fundamental designs. With the ever-growing mobile data traffic, providing user-mobility has become a necessity. While consumer mobility is implicitly handled in NDN, producer mobility is still of one the main challenges. In this paper we propose a hop-count based forwarding strategy to support seamless producer mobility. The key idea of this strategy is: the router makes a decision based on the number of hops traveled by the interest whether to forward the interest using FIB entries or broadcast it. The intuition behind this strategy being two-fold: 1) Spatial locality of producer and 2) Data packets follow the reverse path of interests in NDN. Using simulations, we evaluate the performance of our proposed approach and compare it with Neighbor Aware Interest Forwarding. We demonstrate that our proposed strategy achieves better throughput in terms of number of interests served while reducing the overall traffic generated.**

I. INTRODUCTION

With the advancements in technology, the Internet has become essential even for performing day to day activities. CISCO Visual Networking Index forecasts that annual global IP traffic will reach 2.3 zettabytes by 2020 and video IP traffic will constitute 82% of the total traffic [1]. As the current Internet architecture is host centric, overlay solutions like content delivery networks and peer to peer networks are needed for efficient dissemination of content. There are a number of ongoing research efforts for efficient retrieval and distribution of content. Information Centric Network (ICN) is one of the many proposals in this direction. Named Data Network (NDN) is a frequently quoted ICN design which follows the hourglass model of IP. In NDN, a consumer only needs to know “what” is the content, unlike in the current Internet where the consumer is also required to know “where” is the content [2].

NDN requires every content to be uniquely named which is used by consumers for retrieving them. Consumers generate an Interest packet (IntP) with the unique name in order to request for the content. The content producers respond with the corresponding Data packet (DatP). In the rest of the paper, we will be using content and data synonymously. When an IntP arrives at a router, the router first looks for the content in its own cache called the Content Store (CS). If a hit occurs, the router replies with the data. In case of a miss, the router looks up the Pending Interest Table (PIT) which stores the information about pending interests (for which DatP is not yet received) and the corresponding incoming links (also referred as “requesting face”). If an interest is already pending for the content, then the entry is updated in the PIT to accommodate the new IntP but no new IntP is generated. If no match is found in the PIT, the router adds an entry into PIT. Then it looks up the Forwarding Information Base (FIB) which maps name prefixes to the forwarding faces and finally forwards the IntP to the corresponding face. When a DatP arrives, a router looks up the PIT for any matching entires. If found, the router sends the DatP back to the requesting face and caches the DatP in its CS. Otherwise, the DatP is considered to be unsolicited and is dropped.

We can observe the following salient features of NDN:

1) Each content is uniquely identified.
2) Content’s identity and its location are decoupled.
3) Data packet traverses the reverse path of its corresponding IntP.
4) In-network caching is implicitly present.

These features result in benefits like smaller content retrieval time and significant reduction in redundant traffic. NDN being a young architecture, also has several problems to be addressed such as user mobility, support for push based traffic etc.

CISCO Visual Networking Index reports that global mobile data traffic will increase eightfold between 2015 and 2020 [1]. With the increasing number of mobile users, support for mobility has become important in any Internet architecture and NDN is no exception. Consumer mobility is inherently supported in NDN and can be achieved by interest retransmission [3]. In contrast, producer mobility is not supported trivially by NDN because it involves routing of IntPs unlike DatPs which traverse back the path taken by IntPs using the PIT entries. When a producer moves, the routing information in FIBs corresponding to its content has to be updated. After a producer moves and is not reachable, retransmission of the IntPs will not help the consumers until all the routers along the path to the producer have updated their FIBs. Different applications would require different bounds on this handoff latency. For example, a web browser can tolerate higher handoff latency whereas delay sensitive or real time applications would require this latency to be as small as possible.

In this paper, we propose a forwarding strategy to support seamless producer mobility. Our proposed strategy makes a forwarding decision on the IntP based on the number of hops it has traveled and updates the FIB entries with the current location of a producer based on the property: DatP traverses the reverse path of its corresponding IntP.

The main contributions of this paper are as follows:

1) It presents a methodology for supporting seamless producer mobility.
2) The methodology is based on a distributed approach which mitigates the requirement of centralized resources.
3) The proposed methodology respects the principles of NDN like hierarchical naming scheme, content-location decoupling, interest aggregation, in-network caching.

We compare our proposed forwarding strategy with Neighborhood Aware Interest Forwarding (NAIF) proposed in [4] and the results show that our proposed strategy achieves better throughput in terms of number of interests served while reducing the overall traffic generated.

The rest of the paper is organized as follows. In Section II we provide an outline of the existing schemes for producer mobility in NDN. The producer mobility problem is described in Section III. Section IV discusses the proposed forwarding strategy in detail. The performed simulations and results are presented in Section V. Finally, Section VI concludes the paper.

II. RELATED WORK

Multiple solutions for addressing producer mobility have been proposed in the literature. One of the approaches is to use a *rendezvous server* which has the information regarding the current location of a producer [5]. The producer is required to update the server in the event of a handoff. Consumers are required to query the rendezvous server for the location of the producer before sending IntPs. Another solution approach presented in [6] is to use an *indirection server* which is updated with the latest prefix a mobile producer is associated with. All IntPs first reach the indirection server which either modifies or encapsulates the IntP with producer's current prefix. The producer is required to update the indirection server with its latest prefix whenever a handoff occurs. In [7], [8] the authors proposed a tunnel based approach, where a home router for the mobile producers tunnels all the IntPs to the current access router of the producer. The producer is required to keep the home router updated about is current access router. Authors in [9] proposed a solution approach where some of the routers have an additional functional component called resolvers. The producers send a new type of packet called binding update packet to inform the resolvers about their movement. A field named routing tag is added to the interest. When an interest arrives at a router with the resolver, the router updates the routing tag using the information from binding update packets. Then the router forwards the interest according to the routing tag and eventually the interest arrives at the producer. The approaches in [5]–[8] require additional centralized resources to support producer mobility and hence incurring the drawbacks of centralized systems like single point of failure, scalability etc.

Authors in [10] use the PIT entries of an IntP to track the current location of a producer. This methodology exploits the fact that the DatP traverses the reverse path of the IntP. Use of in-network caching to support producer mobility has been proposed in [11]. The authors initially predict the user traffic pattern and then solve an optimization problem to minimize the network overhead subject to constraints like bounding the distance between consumer and caching router, limiting the amount of traffic generated for supporting mobility and content store capacity limit. However, this approach cannot be used when DatPs are generated dynamically at the receipt of IntPs. In [4], authors discussed an interest forwarding strategy which takes the behavior of neighbors into consideration. It is a probabilistic flooding mechanism where each router counts different types of packets like IntPs sent, IntPs received, DatPs received and PIT entries served. These metrics are then used to calculate the flooding probability. The authors compare the performance of their approach with Listen First and Broadcast Later strategy and vanilla flooding. We compare our proposed strategy with this approach in Section V and the results show that our strategy performs better in terms of the throughput achieved and overhead incurred.

In this paper, we propose a hop-count based forwarding strategy to support seamless producer mobility. Our proposed strategy is distributed in nature and hence mitigates the disadvantages of a centralized approach. Our strategy is designed to respect the fundamental principles of NDN like hierarchical naming scheme, content-location decoupling, interest aggregation, in-network caching.

III. PROBLEM DEFINITION

In this section, we describe the producer mobility problem in detail. Let us consider the scenario depicted in Fig. 1. Here the mobile producer (MP) can move freely in one dimension (along the dashed horizontal line) and connects to the nearest WiFi access point (AP). Assume that the consumer (C) is accessing a video from the MP which is currently connected to AP_2.

- *Before handoff of MP:* When C generates IntPs for the video, every router routes it by matching its name to the FIB entries and corresponding entires are made in the PIT. Finally, the IntP reaches the MP following the path $C - R_1 - R_2 - R_3 - AP_1 - MP$. Using the PIT entries of IntP, the data traverses the reverse path $MP - AP_1 - R_3 - R_2 - R_1 - C$.
- *During handoff of MP:* Assume that the MP moves from AP_2 to AP_3. This movement of the MP is not updated in the FIB of R_2. As a result when an IntP from C is looked up in the FIB of R_2, it is still matched to R_3. Hence,

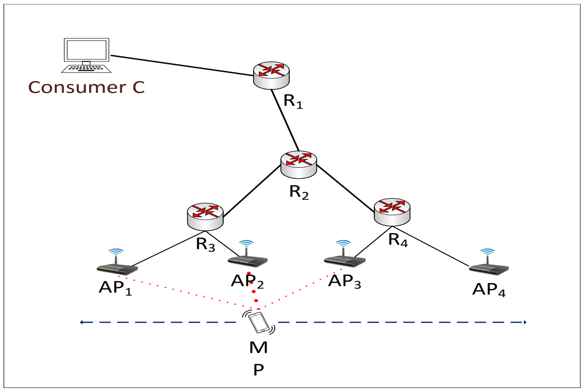


Fig. 1: Example depicting the producer mobility problem

R_2 forwards the IntP to R_3. Similarly, R_3 forwards the IntP to AP_2 where the IntP ultimately gets dropped. This leads to service interruptions for the consumer C.

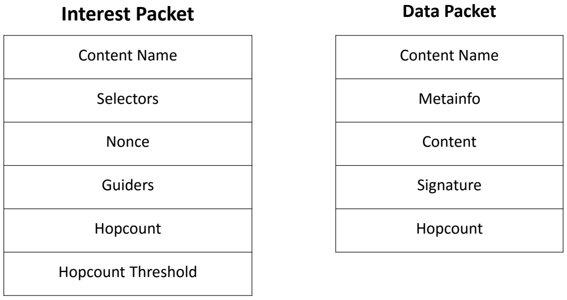
- *After handoff of MP:* Once R_2's FIB is updated with the movement of MP, the FIB lookup for IntP will match it with R_4 and the IntP is forwarded to R_4. R_4 in turn forwards it to AP_3 and subsequently it reaches the MP. Hence the IntP from C, now takes the path $C - R_1 - R_2 - R_4 - AP_3 - MP$ and the DatP traverses the reverse path $MP - AP_3 - R_4 - R_2 - R_1 - C$.

C will not receive any DatPs until the handoff occurs and the relevant FIBs are updated. This handoff latency depends on the number of FIBs to be updated, which in-turn depends on the movement of the MP and the topology of the network. Even with the optimistic assumption that FIBs are updated instantaneously, interruption due to handoff at link and physical layers are inevitable. Moreover, the worst case scenario is when the MP moves faster than the FIB updates, i.e., the amount of time the MP is connected to an AP is smaller than the time taken to update all the relevant FIBs. In this case, the IntPs will never reach the MP and C will be unable to receive its desired service.

IV. HOP-COUNT BASED FORWARDING

In this section, we describe in detail our proposed hop-count based forwarding strategy for seamless producer mobility. The proposed forwarding strategy exploits the following two properties:

1) *Spatial locality of the mobile producer:* Let us consider the example in Fig. 1. Here for simplicity, let us consider the movement of MP in one dimension. Assume that the MP is initially connected to AP_2. Then after disconnecting from AP_2, the MP would either connect to AP_1 or AP_3. It is not possible for the MP to connect to AP_4 before AP_3. Hence, given the current location of the MP, we can predict the possible locations where the MP can move. In practical scenarios, the number of possible locations is bounded. Hence for seamless producer mobility, we can multicast the IntP to these possible locations and as a result, the IntP will reach the MP even before all the relevant FIBs are updated.



Interest Packet
- Content Name
- Selectors
- Nonce
- Guiders
- Hopcount
- Hopcount Threshold

Data Packet
- Content Name
- Metainfo
- Content
- Signature
- Hopcount

Fig. 2: Interest and Data formats

2) *DatP traverses the reverse path of IntP:* As DatPs are routed using the PIT entries of IntPs, every DatP follows the reverse path of its corresponding IntP. This property can be used to decide the current location of the MP. Let us again consider the example in Fig. 1. Assume that the MP is connected to AP_2 and starts to move. Hence, we multicast the IntP to AP_1, AP_2 and AP_3. If the MP is now connected to AP_1, then a DatP generated in response to the IntP will arrive at R_3 from the face connected to AP_1. Using this property, R_3 can update its FIB to match the content served by the MP with the face connected to AP_1.

A. Interest Format

We propose to add two fields namely hop-count (α) and hop-count threshold (θ) to the IntP format mentioned in [12]. α is initially set to 0 by the consumer and is incremented at every subsequent router. The value of θ is set by the consumer depending on the previously received DatPs. Every router makes a forwarding decision for an IntP based on the values of its α and θ, as described in Section IV-B. We also define TTL (Time To Live) similar to that in the Internet Protocol, where an IntP is dropped when α exceeds TTL. The hop-count field is added to a DatP as well, since it is required by the consumer to estimate the value of θ.

B. Forwarding Strategy

1) At the consumer: To request for any data, a consumer has to generate an IntP with its corresponding name. The consumer sets the hop-count value to zero ($\alpha := 0$) which is subsequently incremented at every router. Initially as the whereabouts of the producer is not known, the first IntP has to be broadcasted. Hence the value of θ is set to 0 ($\theta := 0$). When the DatP corresponding to the first IntP arrives, say that it has a hop-count value of α_d. Then for the subsequent IntPs we set the value of θ to $\alpha_d - 1$ ($\theta := \alpha_d - 1$). The value of θ is updated on every subsequent arrival of a DatP. The value of $\alpha_d - 1$ implies that the IntP is broadcasted by the router to which the MP's access point is connected. In case of a retransmission time out, consumer decrements the value of hop-count threshold by one ($\theta := \theta - 1$) and retransmits the interest. As the producer is not reachable, reducing the value of θ increases the number of nodes where IntP is forwarded. The value of θ is reset to $\alpha_d - 1$ after successful receipt of DatP. We have summarized the updating of θ in Algorithm 1. We can intuitively observe that the value of θ depends on the movement of the MP and the topology of the network.

Algorithm 1: *Updating the value of θ*

```
$\alpha := 0$
if First IntP then
    $\theta = 0$
else if Recieved DatP then
    $\theta := \alpha_d - 1$
else if Retransmission time out then
    $\theta := \max\{\theta - 1, 0\}$
```

2) At the router: When a IntP packet arrives at a router, the router looks up for the corresponding data in its CS. If found, the router responds with the DatP. Otherwise, the router looks up the PIT for any pending interest for the same data. If an entry exists, the router updates the PIT entry to accommodate the current interest as well. When none of the above conditions are met, the router is required to forward the IntP. This behavior is the same as mentioned in [12]. In order to forward the IntP in our proposed strategy, the router compares the value of α and θ. If $\alpha < \theta$, the router looks up the FIB and forwards the IntP to the best matched face. On the other hand, when $\theta \leq \alpha <$ TTL the router broadcasts the IntP to all its neighbors and IntP is dropped by the router if $\alpha \geq$ TTL. The interest forwarding strategy at a router is summarized in Algorithm 2.

On the arrival of a DatP, the router looks for a matching PIT entry. If found, the router caches the DatP in its CS and forwards the DatP to the corresponding faces. As the DatPs traverse the reverse path of IntP, the MP must be reachable from the incoming face of a DatP. Hence the router updates it FIB indicating that the MP is reachable from the incoming face of the DatP. The data forwarding strategy is summarized in Algorithm 3.

Algorithm 2: *Interest forwarding*

```
if (Match found in CS) then
    Reply with DatP
else if Match found in PIT then
    Update the PIT entry
else if $\alpha < \theta$ then
    Forward IntP to face matched using FIB
else if $\theta \leq \alpha < TTL$ then
    Broadcast IntP to all the neighbors
else
    Drop IntP
```

Algorithm 3: *Data forwarding*

```
if (Match found in PIT) then
    Forward DatP to corresponding faces
    if FIB entry for the DatP != DatP incoming-face then
        Update the FIB entry
else
    Drop unsolicited DatP
```

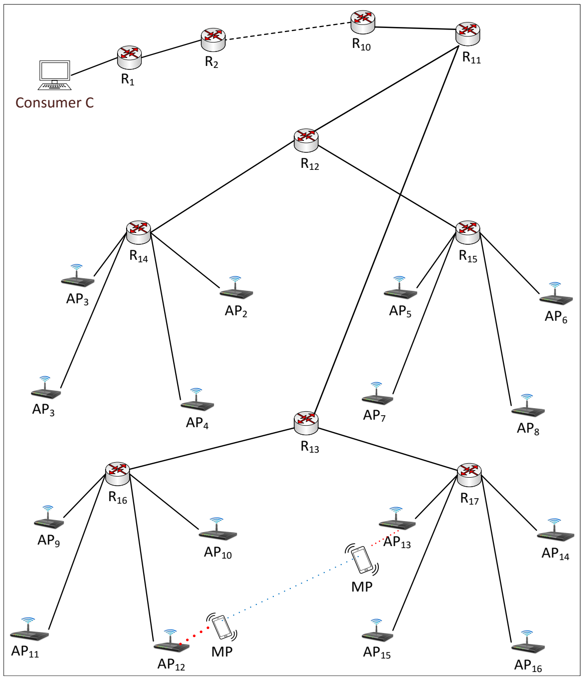


Fig. 3: Network for simulation. Note that R_2 and R_{10} are connected by a sequence of routers $R_3, R_4, \cdots, R_9$.

C. An Example

We illustrate the working of our forwarding strategy using the network depicted in Fig. 3. Consumer C generates an IntP for content served by the MP. The APs are distributed in a 2-D space with the MP capable of moving randomly in the 2-D space. Consider the following scenario: the MP is initially connected to AP_{12} and it gradually moves towards and connects to AP_{13}. Let the initial value of θ be 13 i.e., IntPs are broadcasted to all the APs connected to R_{16}.

When the MP is connected to AP_{12}, an IntP is forwarded to R_{16} by matching the FIB entries of routers along the path $R_1 - R_2 - \ldots - R_{11} - R_{13}$ as $\alpha < \theta$ for all the routers in the path. When IntP reaches R_{16}, the condition $\alpha = \theta$ is met. Therefore, R_{16} broadcasts the IntP to all the APs including AP_{12} which forwards the IntP to the MP. The MP responds to the IntP with the DatP which follows the reverse path of IntP back to C. As the incoming-face of the DatP matches the FIB entries for all the routers, there are no upates made to the FIB entries.

When the MP starts moving towards AP_{13} and disconnects from AP$_{12}$, the IntP does not reach the MP. This causes a retransmission time out at C. As a result, C will decrement the value of θ to 12 and retransmit the IntP. Meanwhile, a connection is established between the MP and AP$_{13}$ but this information is not updated in the FIB of any router. When the retransmitted IntP arrives at R$_{13}$ (following the same path as the previous IntPs), R$_{13}$ compares the values of α and θ. As $\theta = 12$, the condition $\alpha = \theta$ is satisfied and R$_{13}$ broadcasts the IntP to R$_{16}$ and R$_{17}$. Both R$_{16}$ and R$_{17}$ will broadcast the forwarded IntP to the respective APs connected to them as $\alpha > \theta$. Similarly when AP$_{13}$ broadcasts the IntP, it is received by the MP. The DatP generated by the MP traverses the reverse path of the IntP, i.e., $\mathrm{AP}_{13} - \mathrm{R}_{17} - \mathrm{R}_{13} - \mathrm{R}_{11} - \ldots - \mathrm{R}_1$. The FIB entries of AP$_{10}$, R$_{17}$ and R$_{13}$ are updated. There are no updates in FIB entries for the other routers $(\mathrm{R}_1, \mathrm{R}_2, \cdots, \mathrm{R}_{11})$. When the DatP arrives at C, it updates the value of θ to 12 and continues to generate IntPs. The subsequent IntP follows the path $\mathrm{R}_1 - \mathrm{R}_2 - \ldots - \mathrm{R}_{11} - \mathrm{R}_{13} - \mathrm{R}_{17}$ matching the FIB entries. R$_{17}$ broadcasts the IntP which then reaches MP via AP$_{13}$. IntPs follow this path until the MP moves again.

V. SIMULATION RESULTS

To evaluate the performance of our proposed hop-count based forwarding approach, we use the NS-3 based NDN simulator named NDNsim [13].

We consider the network topology shown in Fig. 3. There exist a single path from consumer C to R$_{11}$ (i.e. $\mathrm{C} - \mathrm{R}_1 - \mathrm{R}_2 - \ldots - \mathrm{R}_{10}$). R$_{11}$ is the root of a tree with R$_{12}$ and R$_{13}$ as its children. At the next level, R$_{12}$ is the parent for R$_{14}$ and R$_{15}$. Similarly R$_{13}$ has two children, R$_{16}$ and R$_{17}$. Every router at the second level is connected to 4 APs with all the APs positioned in the X-Y plane. All the links are wireline links expect the connection between an AP and the MP (which is wireless). Every wireline link has data rate of 50Mbps and each wireless link has a data rate of 25Mbps.

The mobile producer is present in the X-Y plane and moves randomly within the X-Y plane. The speed of the MP is varied from 2m/s for a human walk to 30m/s for a vehicle movement. Each link has a propagation delay of 10ms. C generates interests at a rate of 10 IntPs per second. Each DatP served by the MP is 1KB in size. The simulation parameters are summarized in Table I.

Radius of AP coverage	100m
Distance between adjacent APs	200m
MAC of APs	StaWifiMac
Data rate of wireless connection	25Mbps
Data rate of wireline connection	50Mbps
Propagation delay in wireline connection	10ms
Simulation time	1000s
Interest rate	10 IntP/sec
θ (initial value)	13
DatP size	1KB
Speed of MP (m/s)	[2,5,10,15,20,25,30]
Mobility model	RandomDirection2d

TABLE I: Simulation parameters

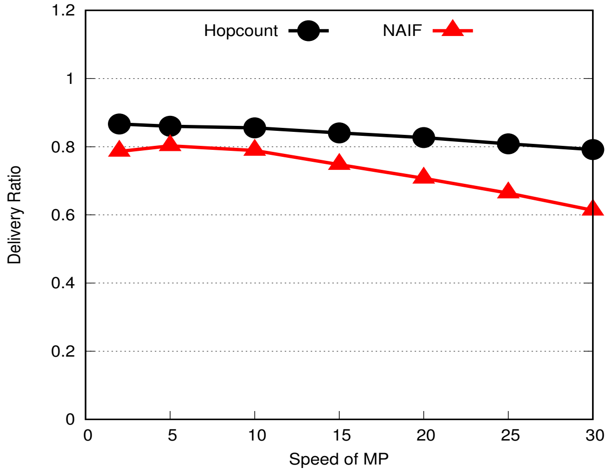


Fig. 4: Delivery Ratio

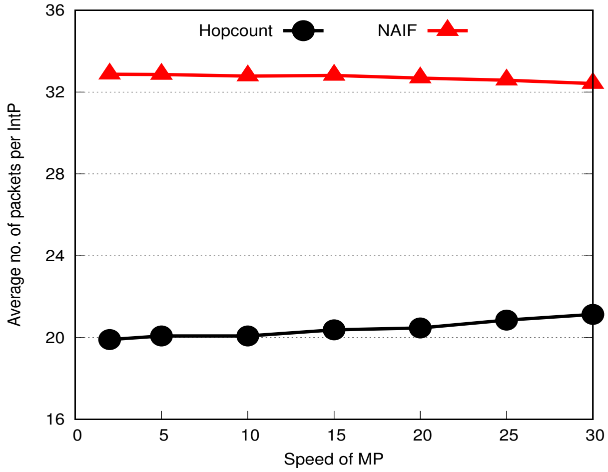


Fig. 5: Average number of packets per IntP

We compare our proposed hop-count based forwarding strategy with NAIF in [4] where IntPs are probabilistically flooded. The flooding probability is estimated by using the number of IntPs sent, IntPs received, DatPs received and PIT entries served. We consider content delivery ratio (for throughput), number of packets generated (for overhead) and consumer experienced delay as the metrics to compare the performance of both the strategies. We define the content delivery ratio ρ as the ratio of the total number of DatP received and the total number of IntP sent:

$$\rho = \frac{\text{Number of DatPs received}}{\text{Number of IntPs sent}}. \tag{1}$$

A. Content Delivery Ratio

In order to compare the content delivery ratios of the strategies, we vary the speed s of the MP, ($s = [2, 5, 10, 15, 20, 25, 30]$). The remaining parameters are set as mentioned in Table I. The obtained result is depicted in Fig. 4. We can observe that our proposed strategy performs

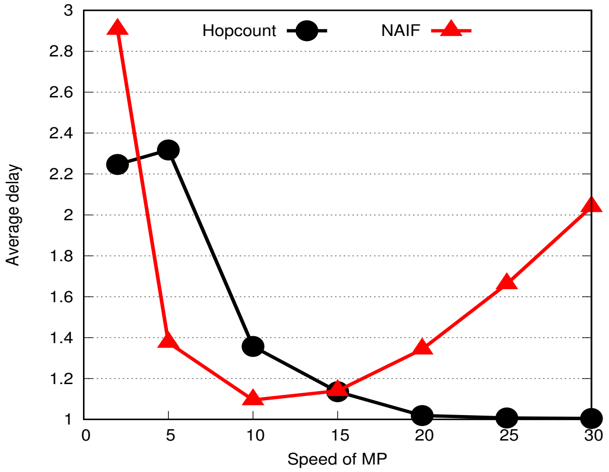


Fig. 6: Average delay

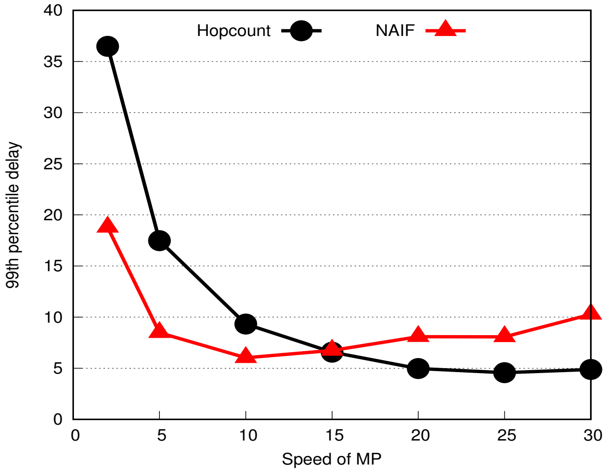


Fig. 7: 99th percentile delay

better than NAIF. As the speed of the MP increases, the content delivery ratio ρ gradually decreases from 0.86 when speed is 2m/s to 0.79 when speed is 30m/s. In case of NAIF, the content delivery ratio is 0.78 when speed is 2m/s and it reduces to 0.61 when speed is 30m/s. This behavior is intuitive as faster movement of the MP increases frequency of handoffs and as a result more IntPs are dropped.

B. *Average Number of Packets per IntP*

Next we compare the average number of packets generated per IntP under varying speed s of MP, ($s = [2, 5, 10, 15, 20, 25, 30]$). Values of other parameters are as mentioned in Table. I. Fig. 5 shows the results obtained. It can be noted that for the hop-count based strategy, the number of packets generated increase marginally as the speed increases. For example, 20 packets are generated when $s = 2$m/s which increases to 21 when $s = 30$m/s. A faster MP implies more frequent handoffs which leads to more IntP losses and hence increased number of packets. The number of packets generated by NAIF is approximately the same for varying s (33 packets). Hop-count based strategy outperforms NAIF as the proposed strategy broadcasts IntP only when the hop-count threshold is exceeded, whereas NAIF performs probabilistic flooding at all the routers.

C. *Consumer Experienced Delay*

Finally, we compare the average delay experienced by the consumer when the speed s of the MP varies, ($s = [2, 5, 10, 15, 20, 25, 30]$). The obtained results are depicted in Fig. 6 and Fig. 7. The average delay when s varies is shown in Fig. 6 whereas Fig. 7 shows the 99th percentile delay. We can observe that in case of the hop-count based strategy, the average delay reduces with increasing speed. This is counter-intuitive as increasing s will result in more frequent handoffs which in-turn should lead to increased average delay. Even though the number of handoffs is more when the MP is faster, we can observe from Fig. 7 that the delay incurred due to each handoff is significantly smaller when the MP is faster. For example when $s = 2$m/s, the 99th percentile delay is 36s as the MP takes a longer time to move within the range of a new AP after disconnecting from the current AP. In contrast when $s = 30$m/s, the 99th percentile delay is 5s as the MP the connects faster to the new AP. Therefore, the average delay decreases with increasing speed of the MP. Comparing both the strategies, we can observe that NAIF incurs smaller delay when MP's movement is slow but as s increases our proposed strategy performs better than NAIF.

VI. CONCLUSIONS

The producer mobility in Named Data Networks is an important problem being addressed by the research community. In this paper, we proposed a hop-count based interest forwarding strategy to support seamless producer mobility in Named Data Networks. The forwarding decision on an Interest packet is made by taking the number of hops it has travelled into consideration. We evaluate the performance of our strategy by comparing it with NAIF. Simulation results show that our proposed strategy achieves better delivery ratio as compared to NAIF as well as generates lesser number of packets. In terms of consumer experienced delay, our proposed approach performs better than NAIF when MP moves faster.

REFERENCES

[1] CISCO, "White paper: Cisco VNI Forecast and Methodology, 2015-2020," Tech. Rep., July 2016.

[2] 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.*, pp. 66–73, Jul. 2014.

[3] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, "A survey of mobility in information-centric networks: Challenges and research directions," in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*, 2012, pp. 1–6.

[4] Y.-T. Yu, R. B. Dilmaghani, S. Calo, M. Sanadidi, and M. Gerla, "Interest propagation in named data manets," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 1118–1122.

[5] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom, “Mobility support in content centric networks,” in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 13–18.

[6] F. Hermans, E. Ngai, and P. Gunningberg, “Mobile sources in an information-centric network with hierarchical names: An indirection approach,” *7th SNCNW*, 2011.

[7] J. Lee, S. Cho, and D. Kim, “Device mobility management in content-centric networking,” *IEEE Communications Magazine*, vol. 50, no. 12, 2012.

[8] V. Jacobson, R. L. Braynard, T. Diebert, P. Mahadevan, M. Mosko, N. H. Briggs, S. Barber, M. F. Plass, I. Solis, E. Uzun *et al.*, “Custodian-based information sharing,” *IEEE Communications Magazine*, vol. 50, no. 7, 2012.

[9] H. Nakazato, S. Zhang, Y. J. Park, A. Detti, D. Bursztynowski, Z. Kopertowski, and I. Psaras, “On-path resolver architecture for mobility support in information centric networking,” in *Globecom Workshops (GC Wkshps), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[10] Y. Zhang, H. Zhang, and L. Zhang, “Kite: A mobility support scheme for ndn,” in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 179–180.

[11] H. Farahat and H. Hassanein, “Optimal caching for producer mobility support in named data networks,” in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[12] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, “Named data networking,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[13] “ndnsim.” [Online]. Available: ”http://ndnsim.net/2.3/”