Leveraging Content Connectivity and Location Awareness for Adaptive Forwarding in NDN-based Mobile Ad Hoc Networks

Muktadir Chowdhury
University of Memphis
mrchwdhr@memphis.edu

Junaid Ahmed Khan
West Washington University
junaid.khan@wwu.edu

Lan Wang
University of Memphis
lanwang@memphis.edu

ABSTRACT

Communication in Mobile Ad-hoc Networks (MANETs) is challenging due to their highly dynamic topology, intermittent connectivity, and low data rate. Named Data Networking (NDN) offers a data-centric approach to communication with an adaptive forwarding plane and in-network data caching, which can be leveraged to address these challenges. In this work, we propose a forwarding strategy called Content Connectivity and Location-Aware Forwarding (CCLF) for NDN-based MANETs. CCLF broadcasts NDN packets and lets each node make independent decisions on whether to forward packets based on per-prefix performance measurements and any available geo-location information. In addition, it employs a density-aware suppression mechanism to reduce unnecessary packet transmissions. Moreover, we have developed a link adaptation layer for ad-hoc links to bridge the gap between CCLF and the capabilities of the underlying link. Our evaluation shows that CCLF not only reduces packet overhead significantly compared to flooding, but also has a data fetching performance close to that achieved by flooding. It also outperforms three other forwarding strategies proposed for information-centric vehicular networks.

CCS CONCEPTS

• **Networks → Routing protocols**; **Mobile ad hoc networks**.

KEYWORDS

Named Data Networking, MANET, routing, forwarding

ACM Reference Format:
Muktadir Chowdhury, Junaid Ahmed Khan, and Lan Wang. 2020. Leveraging Content Connectivity and Location Awareness for Adaptive Forwarding in NDN-based Mobile Ad Hoc Networks. In *7th ACM Conference on Information-Centric Networking (ICN '20), September 29-October 1, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3405656.3418713

1 INTRODUCTION

Mobile Ad-hoc Networks (MANETs) are desirable in many environments that lack fixed infrastructures, such as emergency response, vehicle-to-vehicle communication, and military missions. However, the intermittent connectivity in MANETs makes it difficult to retrieve data successfully over multiple hops, which means an effective routing/forwarding solution has to make use of short-lived routes and be highly adaptive. Unfortunately, many MANET routing solutions use control messages to acquire topological information and use unicast to forward packets to next hops. The topology acquisition process is both costly and ineffective in a highly dynamic network. Moreover, unicast to specific neighbors is not robust when connectivity changes frequently. Other solutions use geographic forwarding to reach mobile nodes, but they must handle local minima when nodes geographically close to the destination do not have physical connectivity to it. In addition, they must resort to flooding when location information is unavailable or the destination moves. The root problems for these solutions are two-fold: (1) they cannot handle packet loops effectively due to the lack of forwarding state at the network layer, so they have to rely on unicast and loop-free paths for multi-hop communication, and (2) they cannot easily detect or react to forwarding issues due to a lack of feedback loop in the data plane at the network layer.

Named Data Networking (NDN) [28] follows a data-centric approach to communication – every piece of data is named and retrieved using its name. Moreover, it maintains fine-grained forwarding state and Interest/Data feedback loops at the network layer, which enables broadcast-based forwarding strategies to handle connectivity changes effectively and efficiently. In addition, NDN nodes offer in-network data caching for improved data availability. All of these features can be leveraged to address the above problems in current solutions.

In this paper, we propose an adaptive forwarding strategy for NDN-based MANETs – ***Content Connectivity and Location-Aware Forwarding (CCLF)***. CCLF broadcasts NDN packets and allows each node to make independent decisions on whether to forward packets based on *Content Connectivity*, i.e., the success of a node in fetching data for a particular name prefix, and any available geo-location information. More specifically, each node that receives an Interest sets a forwarding timer and forwards the Interest when the timer expires. The timer is set such that those nodes with better content connectivity for the Interest and shorter distance to the data location will have a higher chance of forwarding the Interest. If a node hears another node's transmission of the Interest before its timer expires, it suppresses its own forwarding with a probability proportional to the number of neighbors in its transmission range. This density-aware suppression mechanism, in conjunction with the smart forwarding timer, significantly reduces unnecessary packet transmissions without negatively impacting the success of data fetching.

Furthermore, we have developed the Ad-hoc Link Adaptation Layer (A-LAL) to bridge the gap between CCLF and the capabilities of the underlying ad-hoc link. A-LAL attaches location

---

ICN '20, September 29-October 1, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-8040-9/20/09…$15.00
https://doi.org/10.1145/3405656.3418713

information to Interest/Data packets when such information is available. It also provides neighbor information to CCLF for calculating suppression probabilities. Finally, it buffers packets when a node does not have any neighbors and sends them out when a neighbor is detected.

We have implemented CCLF and A-LAL in the NDN Forwarding Daemon (NFD) [2] and evaluated them in ndnSIM [1]. Our evaluation shows that CCLF not only reduces packet overhead significantly compared to flooding, it also has a data fetching performance close to that achieved by flooding. It works well with and without geo-location information. Moreover, it outperforms three other forwarding strategies, VNDN [8], Navigo [9], and STRIVE [13], all proposed for information-centric vehicular networks.

The remainder of the paper is organized as follows: the next section discusses background and related work. Sections 3 and 4 present the design of CCLF and A-LAL, respectively, followed by implementation details in Section 5 and evaluation results in Section 6. Section 7 concludes the paper.

2 BACKGROUND AND RELATED WORK

In this section, we present basic information about NDN and discuss prior research on MANET routing/forwarding.

2.1 Named Data Networking (NDN)

By and large today's Internet applications request data by names, but the data names must be first translated to Internet addresses of *specific* end hosts for the applications to fetch data from those hosts. Named Data Networking (NDN) [28] is an Internet architecture developed to support applications to fetch data directly by name. NDN's requests and responses work at a network packet granularity – each request, carried in an NDN *Interest* packet, contains the name of the requested data and fetches one NDN *Data* packet back. Each NDN forwarder uses the name in an Interest to determine to which interface or interfaces this Interest should be forwarded. As shown in Figure 1, each NDN node's forwarding module contains the following basic components: a *Content Store (CS)*, a *Pending Interest Table (PIT)*, a *Forwarding Strategy module*, and a *Forwarding Information Base (FIB)* populated by routing protocols. The CS caches previously received data packets, while the PIT stores previously forwarded Interests that have not fetched the corresponding data yet. If an Interest does not match the CS or PIT, the forwarding strategy will forward the Interest to one or more interfaces using information in the FIB and any performance measurements of the interfaces based on previous Interest/Data exchanges.

2.2 MANET Routing/Forwarding in Host-Centric Networking

A plethora of routing protocols have been developed for MANETs in a host-centric network model such as IP [3]. Most ad-hoc routing protocols select specific neighbors as next hops based on topological information. Some routing protocols, such as OLSR [5], DSDV [21], WRP [19], FSR [20], and STAR [6], proactively build and update a routing table by periodically requesting information from other nodes. Others protocols, such as DSR [11] and AODV [22], do it reactively, i.e., they find the routes on demand by sending route probing messages. Protocols that use control messages to build the routing table incur high overhead in a mobile environment. Our proposed forwarding strategy, CCLF, does not collect topology information or specify which neighbors should forward a packet. Rather, each CCLF node decides independently whether it can be a forwarder for a packet.

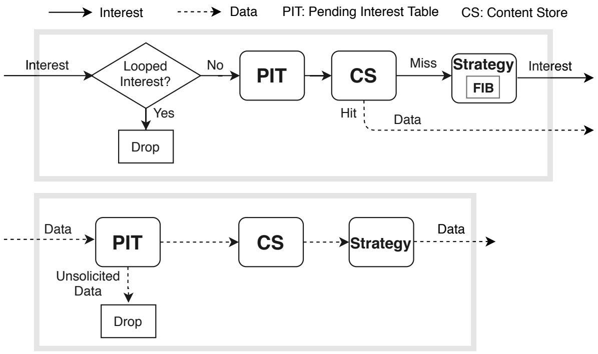


Figure 1: NDN Packet Forwarding Pipeline

Another class of MANET routing protocols consider other information, such as speed, location, or direction of the mobile node, in making forwarding decisions, For example, Directional Greedy Routing (DGR) [7] considers the moving direction of vehicles. The scheme proposed by Lin et. al. in [16] uses vehicular density, vehicular transmission range, and velocity variance information in forwarding decisions. Geographic location is by far the most popular non-topological information used in MANET forwarding [24]. In GPSR (Greedy Perimeter Stateless Routing Protocol) [12], each node chooses a neighbor closest in geographic distance to the destination as the next hop to forward a packet. CCLF can also take advantage of geo-location information in its forwarding, but it does not rely solely on such information.

A common characteristic among many IP-based MANET solutions is that their routing protocol establishes a single path between each sender and receiver and the forwarding plane uses unicast to forward packets between neighbors on the path. Such designs do not work well under high mobility, but since IP's stateless forwarding plane cannot handle packet loops effectively, it has to rely on unicast and loop-free paths to ensure reacheability.

Recognizing that IP's host-centric model and stateless forwarding plane are problematic for MANETs, Meisel et. al. [18] proposed using application data names instead of IP addresses for nodes to request data, and they developed the forwarding protocol LFBL (Listen First, Broadcast Later) that uses exclusively broadcast communication for all packets and lets receivers decide whether to forward a packet based on their distance to the source and destination. LFBL was shown to significantly outperform AODV [18], which suggests using data names instead of IP addresses and broadcasting packets instead of unicasting to a specific neighbor are steps in the right direction. However, LFBL still uses some design elements from host-centric communication. For example, it tries to select paths between specific sources and destinations, and in order to reduce the number of forwarding paths, it lets sources choose specific responders, which may not be immediate neighbors, to relay their packets after the initial flooding.

2.3 MANET Routing/Forwarding over NDN

NDN is a data-centric architecture that facilitates the communication of mobile nodes [17, 30]. NDN's in-network state supports forwarding packets intelligently without a routing table. Moreover, MANETs can benefit from NDN's hierarchical naming [25], built-in security [4], and in-network caching [8]. Given the inherent benefits of NDN in mobile communication, there have been a number of studies on the design of forwarding strategies in NDN-based MANETs.

STRIVE [13] is a forwarding strategy for information-centric vehicular environments in which each node selects the next hop for an Interest based on the neighbors' centrality score (i.e., Interest satisfaction ratio). This approach requires the neighbors to exchange their centrality information. Moreover, it uses unicast to forward Interest/Data packets between neighbors, which is not robust under high mobility as we discussed earlier.

Other approaches take advantage of the wireless broadcast medium to forward NDN packets, and use controlled flooding to reduce bandwidth usage. Yu et. al. proposed NAIF [27], a cooperative forwarding scheme in which nodes forward packets probabilistically based on estimates of their data retrieval rate so that the nodes in one neighborhood will each forward a fraction of the packets. This scheme outperforms LFBL in multi-consumer scenarios [27]. In VNDN [8] and Navigo [9], nodes decide whether to forward packets based on geographic location. VNDN [8] calculates a timer based on the distance to the previous hop, whereas Navigo [9] makes forwarding decisions based on the location of the data producer. In [15], Kuai et. al. proposed to calculate the forwarding timer based on the location of the previous hop and that of the producer. The over-reliance on geo-location makes these protocols infeasible when such information is unavailable.

Our evaluation demonstrates that CCLF outperforms VNDN, Navigo, and STRIVE with or without the help of geo-location information (Section 6).

3 CCLF DESIGN

In this section, we first provide a design overview and then present the individual components of CCLF design.

3.1 Design Overview

The dynamic topology of MANET makes unicast-based forwarding ineffective. Therefore, we leverage the broadcast nature of wireless medium in the design of the CCLF forwarding strategy. In CCLF, each node forwards Interests to all the nodes in its broadcast transmission range. Each receiving node decides *independently* whether or not to forward a packet based on its ***Content Connectivity***, a measure of the node's past performance in fetching data under the same name prefix. If the ***geographic location*** of the requested data is known, then both content connectivity and location information are considered when making forwarding decisions. The design of CCLF is unique in several aspects. First, its use of a fine-grained data-centric connectivity metric helps each node make precise Interest forwarding decisions for individual name prefixes. As such, CCLF can achieve better performance than forwarding schemes that use a coarse-grained connectivity metric. Second, while CCLF can use geo-location information in forwarding Interests, it does not depend on such information which may be unavailable in indoor settings or privacy-sensitive applications, making it applicable to a wide range of environments. Third, as we will show in Section 6, CCLF outperforms pure geographic forwarding schemes as it can use content connectivity information to avoid poorly-connected nodes that are geographically close to the data.

Figure 2 illustrates the operations of CCLF at a high level. Initially, nodes far from the data producer P, e.g., 1 and 2, may not have any information about that producer's data. When the consumer C sends an Interest for the data, the nodes hearing the Interest enter a competition based on a random timer and the winners will further broadcast the Interest. In this case, both 1 and 2 forward the Interest after their timer expires, because they are not within each other's transmission range (so one's transmission did not cancel the other's). Node 1's Interest unfortunately experiences a packet loss and does not reach Node 6, but Node 2's Interest is received by 3 and 5 which set a timer to forward the Interest. Node 3's timer expires first due to its better content connectivity and shorter distance to the producer, which triggers the forwarding of the Interest. This Interest is received by Node 4 and 5. Node 5 detects that the received Interest matches one that is already scheduled to be forwarded so it cancels the scheduled forwarding based on a probability function defined in Section 3.5. Node 4 forwards the Interest to the producer P which returns the matching data. As the data traverses back to the consumer, the intermediate nodes 2, 3, and 4, as well as the consumer, update their content connectivity information and possibly the data location for the corresponding name prefix. Such information is used to calculate a forwarding timer for subsequent Interests matching the same name prefix. Those nodes that have previously fetched data for the name prefix and/or are closer to the data location will have a shorter timeout so they will forward the Interest first, essentially suppressing the other nodes' forwarding. The end result is that the Interests will be forwarded by a sequence of self-selected nodes that have the best chance of fetching the matching data. Since mobile adhoc networks have highly dynamic topology, this set of self-selected nodes is not static – it constantly changes based on the latest content connectivity and data location information.

3.2 Content Connectivity

Content connectivity reflects how successful a node has been in fetching the data under a certain name prefix. In general, nodes with high content connectivity for a name prefix have good network connectivity leading to the corresponding data producer or repository, and they have previously fetched data under the name prefix for themselves or other consumers. While previous work by Khan and Ghamri-Doudane [13] quantifies the connectivity (or centrality) of a node based on its overall Interest satisfaction ratio, our proposed scheme takes advantage of the hierarchical naming of NDN and the availability of application data names at the network layer to calculate a *Content Connectivity Score (CCS) for* ***each name prefix***, which means a node can have different scores for different name prefixes. *This fine-grained measure of forwarding performance helps the network layer make more informed decisions when forwarding Interests to retrieve data from different producers.*

Each node calculates the CCS of a name prefix based on the percentage of satisfied Interests for the name prefix and its descendant

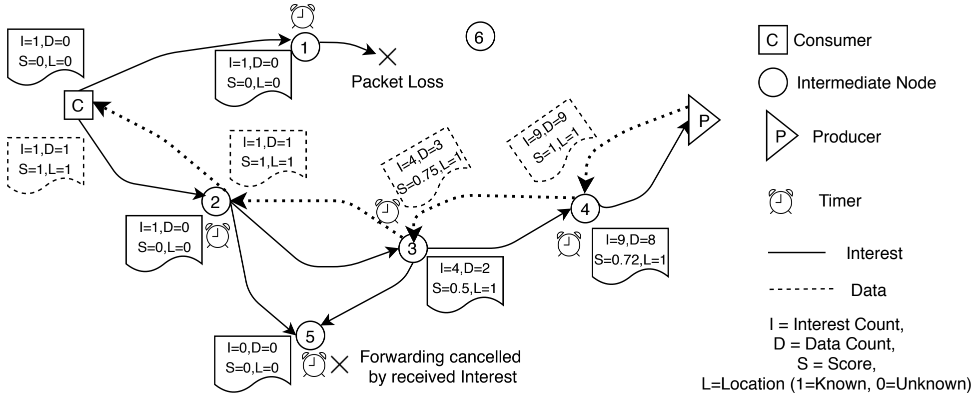


Figure 2: A Simplified Example of CCLF Operations

prefixes as follows:

$$CCS_j = \frac{D_j + \sum_{i \in Desc(j)} D_i}{I_j + \sum_{i \in Desc(j)} I_i}, \tag{1}$$

where D_j is the Data count for Prefix j, I_j is the Interest count for Prefix j, and $Desc(j)$ is the set containing the descendants of Prefix j. This calculation requires knowing the name prefix that a data producer owns for publishing its data. In our scheme, when a data producer responds to an Interest, it attaches the name prefix to the packet in the NDN Link Protocol (NDNLP) header [23] (see Section 5). When a node receives a data packet, it updates the Data count for the corresponding name prefix.

The above CCS definition (in Equation (1)) includes the Interest and Data count of a name prefix's descendants in the calculation of that prefix's CCS. In other words, the CCS of a name prefix represents the average satisfaction ratio of all the known name prefixes under that namespace. This approach helps us avoid forwarding an Interest purely randomly when the corresponding prefix is not in the FIB – the Interest will match an ancestor of the actual prefix and the forwarding timer will be set using the CCS information from the ancestor prefix. Of course it is possible that in some cases the average satisfaction ratio of the ancestor prefix may not be a good predictor of how well that specific Interest will be served by this node. Fortunately, our approach will correct itself – when the Data packet comes back, the node will learn the actual name prefix and start measuring its CCS more accurately, which will help forwarding future Interests under that prefix more effectively. We plan to further refine the CCS definition in our future research.

CCS is calculated periodically based on the Interest and data count in each period, but we use the exponentially weighted moving average of CCS, which considers both past and current performance (Equation (2)), when making forwarding decisions.

$$\widehat{CCS}_{i,N} = \alpha \cdot CCS_{i,N} + (1 - \alpha) \cdot \widehat{CCS}_{i,N-1}, \tag{2}$$

where i is a name prefix, N is the current period, and α is the weight for the weighted average. By default, each time period is 6 seconds long and α is 0.125 in our implementation. The moving average is desirable in a dynamic environment. If a node moves near a data producer, its $\widehat{CCS}$ will gradually increase for that producer's data. On the other hand, if the node moves away and fails to fetch the producer's data, the moving average $\widehat{CCS}$ will decay over time.

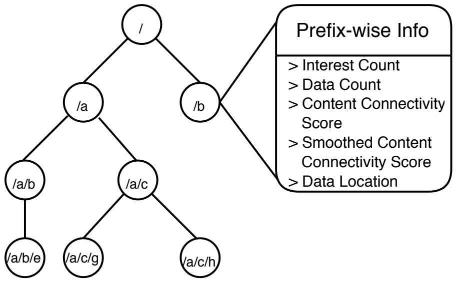


Figure 3: An example C-L Tree: Interest Count is updated when an Interest is forwarded. Data Count and Data Location are updated when a Data packet is received. Content Connectivity Score (CCS) and Smoothed Content Connectivity Score ($\widehat{CCS}$) are updated periodically.

We use a data structure called Connectivity-Location (C-L) Tree (Figure 3) to store the content connectivity score and location information (discussed in 3.3) of each name prefix. The tree structure enables us to store and retrieve information efficiently for each prefix. Figure 3 shows an example C-L tree with the the root prefix / and its descendants. When a node first starts, its C-L tree only has the root prefix with all the associated values set to 0 or NULL. The initial Interests will match the root prefix and be forwarded in a random walk-like manner due to a $\widehat{CCS}$ of 0 and a lack of other information (see Section 3.4). However, the tree will be populated with more name prefixes and statistics over time as more data is received.

3.3 Geographic Location

Data consumers may be interested in data associated with a specific geographic area, e.g., accident information at a traffic intersection. In our scheme, these consumers can attach the data location to their Interests (using an NDNLP header) and nodes geographically closer to the data location are given priority in forwarding the Interests. More specifically, each node receiving an Interest associated with a data location calculates a Location Score (LS) using the location of the previous hop (in the NDNLP header), its own location, and the data location (also in the NDNLP header) as follows:

$$LS = 1 - \frac{Dist(n,d)}{max(Dist(n,d), Dist(p,d))} \tag{3}$$

where LS is the Location Score, $Dist(n,d)$ is the distance from this node n to the data location d, and $Dist(p,d)$ is the distance from the previous hop p to the data location d. This equation gives an LS of 0 if a node is farther from the data location compared to the previous hop, which will make it unlikely for this node to forward the Interest. On the other hand, those nodes closest to the data location (among all the nodes receiving this Interest) will have the highest LS, and they will have a better chance in forwarding the Interest (see Section 3.4).

Note that data location information is optional in CCLF – not every data producer can obtain or is willing to share its geo-location. CCLF can use content connectivity information to forward Interests when location information is absent. In fact, a node's priority in forwarding an Interest is determined by a combination of Content Connectivity Score and Location Score (see Section 3.4). *This is a major difference between our scheme and pure geographic forwarding schemes*, and our evaluation shows that CCLF is more effective in retrieving data than VNDN and Navigo which use only location information in Interest forwarding.

Our design can take advantage of data location information even when an Interest does not carry a data location and when a data producer is moving. This is because CCLF enables data producers to attach their geo-location (e.g. longitude and latitude) to their data in an NDNLP header whenever they send a data packet. If a node receives a data packet, say /a/b/1 with the name prefix /a/b and location *(long, lat)*, then it will remember this mapping in the C-L tree. Subsequent Interests for data under that prefix, e.g., /a/b/2, will be forwarded toward that location even if they do not contain data location information (CCLF looks up the C-L tree to retrieve it). If the data producer moves, its data will carry the new location and the corresponding field in the C-L tree will be updated. As long as the producer does not move too fast, Interests forwarded based on the old location can still fetch the desired data, especially with the help of content connectivity information. If the producer moves too far from the previous data location, the Interest may not reach the producer and the consumer needs to retransmit the Interest. Note that the data location information has an expiration time specified by the producer based on its speed, so stale location information will not affect Interest forwarding. How to handle inaccurate or out-of-date location information is one of our future research areas.

3.4 Forwarding Timer Calculation

After an Interest is sent by a node over its broadcast link, other nodes within the transmission range decide whether and when to forward the Interest in a decentralized manner. More specifically, each node receiving the Interest first finds the longest matching prefix of the Interest name in its C-L tree. Afterwards, it retrieves the content connectivity and location information of the name prefix (if available). It then calculates the location score LS using Equation 3 and a weighted average w using Equation 4.

$$w = \beta \cdot \widehat{CCS} + (1-\beta) \cdot LS, \tag{4}$$

where $0 \leq \beta \leq 1$. Here β controls the relative importance of the two factors, $\widehat{CCS}$ and LS, in influencing Interest forwarding. We set β to 0.5 by default in our implementation to balance the two influences. Next, w is used to calculate a value t based on Equation 5.

$$t = \begin{cases} min(\frac{1}{w}, T), & \text{if } w > 0 \\ T, & \text{if } w = 0 \end{cases} \tag{5}$$

Here T is an upper bound on t (we set T to 10,000 by default). The forwarding timer is set to a random value uniformly distributed between $0.5t$ μs and $1.5t$ μs, and the Interest will be forwarded upon the timer's expiration (the randomization on the timer value prevents multiple nodes with the same t value from forwarding the Interest at the same time). If the node receives the same Interest during this time period, which means some other node nearby has forwarded the Interest, it cancels its own forwarding of the Interest with a probability based on how many neighbors it has (Section 3.5). Nodes with higher values of $\widehat{CCS}$ and LS for the Interest will have a larger w and consequently a smaller t, so their timer will expire sooner, giving them a higher priority in forwarding the Interest.

3.5 Density-Aware Forwarding Suppression

We have developed the Ad-hoc Link Adaptation Layer (Section 4) to periodically inform the CCLF strategy the number of neighbors in a node's vicinity. We consider any node whose transmissions this node can hear a neighbor. If the node receives an Interest before its forwarding timer of the same Interest expires, it uses a suppression probability proportional to its neighbor count to decide whether to cancel its own forwarding of the Interest. The suppression probability p is calculated as follows.

$$p = min(K \cdot n, 1), \tag{6}$$

where K is a suppression constant, and n is the number of neighbors. In general, nodes with more neighbors will have a higher probability to suppress their own forwarding when they overhear another node forward the same Interest. On the other hand, nodes with fewer neighbors are less likely to cancel their scheduled forwarding in order to maximize the propagation of the Interest. *The same probabilistic approach is used to suppress Data packets.* We have performed experiments to select an appropriate value for the suppression constant K (Section 6).

3.6 CCLF Strategy Algorithms

We summarize the CCLF strategy for forwarding Interests and Data packets in Algorithm 1 and 2, respectively. Since the previous sections have presented the overall design and individual steps in depth, we do not explain the algorithms in detail again. However, we would like to point out the following. **First**, an Interest may match a shorter name prefix in a node's C-L tree than the intended data producer's actual name prefix. This happens when the node has not received any data from the producer yet. If data for the shorter name prefix is roughly in the same direction, the Interest will propagate just fine. Otherwise, the Interest may experience a detour until it reaches a node with the correct name prefix in its C-L tree. However, as soon as the data comes back, the correct name prefix and its information will be installed in the intermediate nodes and subsequent Interests will have better forwarding paths.

Algorithm 1: Interest Forwarding Algorithm

```
1  Algorithm ProcessInterest(Interest $I$)
2      $I_{name}$ ← GetName($I$);
3      if Interest with $I_{name}$ is scheduled to be forwarded then
4          $p$ ← CalcSuppressProb($N_{nbrs}$);
5          CancelForwarding($I_{name}$, $p$);
6          return;
7      end
8      if $I$ is looped then
9          return;
10     end
       // Do not forward if an Interest with
       // the same name was recently forwarded.
       // $T_D$ is a constant on the order of RTT.
11     $I_{time}$ ← GetLastForwardTime($I_{name}$);
12     if ($I_{time}$ > 0) and (now - $I_{time}$ <= $T_D$) then
13         return;
14     end
15     $L_{data}$, $L_{prev}$ ← ExtractLocation($I$);
16     $CCS$ ← LookUpCCS($I$);
17     if $L_{data}$ = null then
18         $L_{data}$ ← LookUpLocation($I_{name}$);
19     end
20     $t$ ← CalculateTimer($CCS$, $L_{data}$, $L_{prev}$);
21     if $t$ + now >= PIT entry's expiration time then
22         return;
23     end
24     ScheduleInterest($I$, $t$, ForwardInterest());
25     Function ForwardInterest ($I$):
26         SendInterest($I$);
27         UpdateCLTree($I_{name}$);
28     End Function
```

Algorithm 2: Data Forwarding Algorithm

```
1  Algorithm ProcessData(Data $D$)
2      if $D$ is scheduled to be forwarded then
3          $p$ ← CalcSuppressProb($N_{nbrs}$);
4          CancelForwarding($D$, $p$);
5          DeletePitEntry();
6          return;
7      end
8      $D_{name}$ ← GetName($D$);
9      $L_{data}$ ← ExtractLocation($D$);
10     $t$ ← $T$;
11     ScheduleData($D$, $t$, ForwardData());
12     Function ForwardData ($D$):
13         SendData($D$);
14         UpdateCLTree($D_{name}$, $L_{data}$);
15         DeletePitEntry();
16     End Function
```

Second, CCLF drops a received Interest in any of the following cases (see Algorithm 1): (1) there is a scheduled Interest with the same name (line 8-12); (2) the Interest is looped, i.e., an Interest with the same name and nonce has been forwarded as recorded in the PIT or dead nonce list (line 13-15); (3) an Interest with the same name and a different nonce has been forwarded recently, i.e., within a short period of time on the order of an RTT (line 16-19); and (4) if the PIT entry will have expired when the scheduled forwarding happens (line 27-29).

Third, CCLF tries to limit the number of duplicate Interests using a forwarding timer and density-aware suppression, but it does not intend to eliminate all the duplicates. In fact, since a wireless network is prone to packet losses and link failures, it is beneficial to have a certain degree of redundancy. Therefore, Interests in CCLF may be forwarded along several paths to reach the data, but still CCLF has a much lower message overhead than pure flooding (Section 6).

Fourth, when a Data packet is forwarded by a node, it may be received by multiple nodes that have previously forwarded the corresponding Interest and they may forward the Data packet at the same time. Therefore, ***we also need to suppress duplicate Data packets***. The difference from Interest forwarding is that all of these nodes are equally good as a forwarder for the Data packet, so they will use the same value of t with a default of 10ms as the average for their forwarding timer (the actual timer value is a random number between $0.5t$ and $1.5t$).

3.7 CCLF Security

CCLF relies on the prefix announcements and location information attached to Data packets to build the C-L tree and make forwarding decisions. Therefore, it is subject to attacks where a rogue node injects false prefix/location information to mislead network traffic. CCLF utilizes NDN's built-in data-centric security to handle such attacks [29] – a producer node needs to sign the prefix announcement or location information attached to its Data packet with its private key. The receiving nodes will verify the information using the producer's public key, the signature, and a pre-defined trust model [26]. The trust model depends on the specific MANET type and application scenario. For example, in case of VANET, we can use the trust model proposed in [4].

4 AD-HOC LINK-ADAPTATION LAYER

We have developed the ***Ad-hoc Link Adaptation Layer (A-LAL)*** to supply CCLF with important information for its operations and to augment the reliability of packet forwarding in a mobile ad-hoc environment (Figure 4). It has three functions. **First**, A-LAL adds NDNLP headers to Interests as needed, i.e., *previous hop location* and *data location* in Interests (Figure 5). Note that the Data Location header in the Interest can also be added by the consumer application, while the Prefix Announcement header and Data Location header in the Data packets are added by the producer application.

Second, A-LAL leverages information from the MAC layer to keep track of the number of neighbor nodes. Whenever it receives a packet from the MAC layer, it adds the sender's MAC address to the neighbor list if it is new. If a neighbor has not been heard from

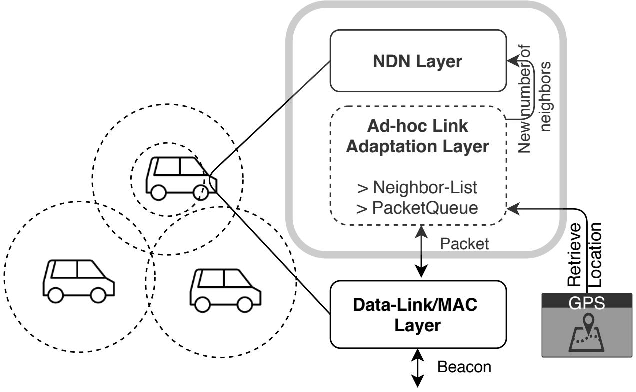
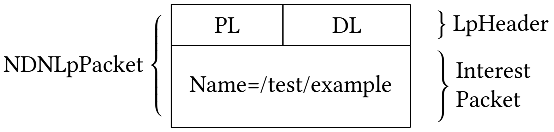
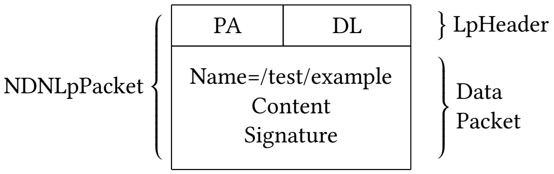


Figure 4: Ad-hoc Link Adaptation Layer's Roles



(a) Interest Packet



(b) Data Packet

Figure 5: NDNLP Headers in Interest and Data (PL = Previous Hop Location, DL = Data Location, PA = Prefix Announcement)

for a while, it will be removed from the list.[1] A-LAL also generates beacons if there are no packets sent for some time.[2] This way nodes in a neighborhood can detect the presence of each other, even if there is no traffic from the application layer. A-LAL informs CCLF whenever the number of neighbors changes so that CCLF can use this information to calculate packet suppression probability (Section 3.5).

Third, in a sparse network, there may be times when there are no neighbors around a node. In this case, sending a packet is a waste of the node's resources, and the application will have to resend the packet to ensure reliability. The neighborhood-awareness of A-LAL is useful here. A-LAL maintains a queue to store packets when the neighbor list is empty, and it starts sending packets from the queue when it detects a neighbor (Algorithm 3). This mechanism improves the chance that packets will propagate to at least one neighbor node.

Algorithm 3: A-LAL Packet Forwarding Algorithm

```
Algorithm ForwardPacket(Packet P)
    if P is not a received beacon then
        if NeighborList is not empty then
            Send P and packets from PacketQueue;
        else
            Put P in PacketQueue;
        end
    else
        // Received a beacon from a neighbor,
        // send queued packets
        Send packets from PacketQueue;
    end
```

5 IMPLEMENTATION

We implemented CCLF as a new *Strategy Module* in the NDN Forwarding Daemon (NFD) [2]. To implement the C-L tree, we utilized the existing NameTree structure in NFD by adding content connectivity and location information to the tree nodes. In addition, we made a few modifications to NFD's forwarding pipeline logic as it was not designed for an ad-hoc networking environment. **First**, in the existing NFD implementation, if an incoming Interest packet with the same nonce is received again by a node, it is deemed as a duplicate and dropped by NFD before CCLF can inspect it. However, CCLF relies on receiving such packets to infer that an Interest is already forwarded by another node in the network so it can cancel its own scheduled forwarding of the same Interest. Therefore, we have modified NFD to allow CCLF to receive such duplicate Interests.

Second, we have a similar problem for Data packets where the first incoming Data packet will remove the corresponding PIT entry and the duplicate one will be dropped by NFD without being forwarded to CCLF. We modified NFD to not delete the PIT entry in the forwarding pipeline so that the duplicate Data packet will be forwarded to CCLF, which will decide whether to cancel its scheduled forwarding. The PIT entry will be removed by CCLF when it cancels the scheduled forwarding or sends out the scheduled Data packet.

Lastly, while a Data packet is scheduled to be forwarded after a delay, if an Interest for the same data hits the Content Store, then the data will be sent immediately by the forwarding pipeline (outside of CCLF). We modified NFD to inform CCLF of such cases so that CCLF can cancel the scheduled packet forwarding.

We implemented A-LAL as a new *Link Service*, which is a submodule of the Face data structure in NFD that provides various features to overcome the limitations of the underlying link. A-LAL encodes and decodes location information in NDNLP headers, and attaches the headers to Interest/Data packets.

6 EVALUATION

We use a Vehicular Ad-hoc Network (VANET) as a case study to evaluate the performance of CCLF. VANET is a specialized MANET where nodes (cars) move with high velocity. Many VANET applications are time-sensitive, but vehicular connectivity is typically short-lived, which makes it difficult to fetch data on time. Running

---

[1] If the MAC protocol does not provide the sender's MAC address, A-LAL can add an NDNLP header with the sender's name to each outgoing packet.

[2] The MAC protocol may have its own beacons that can serve this purpose, but this information needs to be exposed to A-LAL.

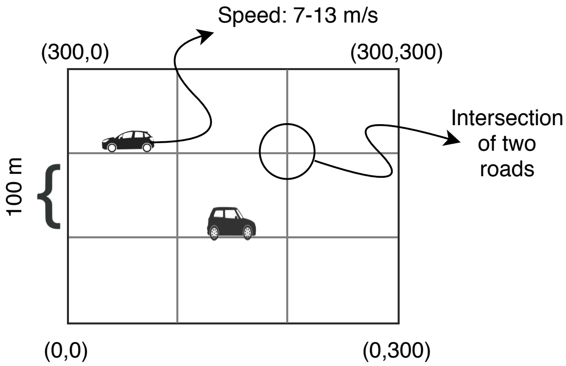


Figure 6: 3x3 Manhattan Grid Topology

experiments in the VANET environment gives us important insight into CCLF's performance.

6.1 Simulation Setup

We used ndnSIM [1], a widely used open-source NDN simulator based on ns-3 [10], to conduct our experiments. Below are the simulation settings.

MAC and Physical Layer Parameters We use the MAC layer protocol 802.11p for ad-hoc communication between vehicles. In the ad-hoc mode, there is no RTS/CTS signaling to avoid collisions. At the physical layer, we use OFDM (orthogonal frequency-division multiplexing) with a 6Mbps data rate and a 10MHz channel width, the range propagation loss model, and constant speed propagation delay model. The transmission range is set to 100 meters.

Road Network and Mobility Model We use SUMO (Simulation of Urban MObility) [14], a widely used microscopic road traffic simulation package, to generate the road network and vehicular traffic trace. We use two road networks - a two-lane road that is 1600m long and 8m wide, and a 3x3 Manhattan grid with 6 two-lane roads covering an area of 300x300 m^2. The two-lane and grid road networks are used to mimic suburban and urban settings, respectively. The vehicles move along the roads at a speed of 7–13 m/s. In the grid road network, they turn at each intersection with a pre-defined probability (we use the default probability setting of 25% left, 25% right, and 50% straight).

Application Traffic In each experiment run, we randomly select multiple nodes, some as consumers and the other as producers. Each consumer generates 1 Interest per second. The producers respond with the corresponding data.

CCLF Parameters We use $\alpha = 0.125$ in Equation 2 to calculate the smoothed content connectivity score, and $\beta = 0.5$ in Equation 4 to balance the effect of content connectivity and location score in determining the forwarding timer. In some of the experiments, we set $\beta = 1$ to exclude the location score from the calculation of the forwarding timer. The constant T in Equation 5 is set to 10,000 (i.e., 10ms), and T_D in Algorithm 1 is set to 1 second.

6.2 Experiments

We have run four sets of experiments. The first set focuses on choosing a suitable suppression constant (K in Equation 6). A large suppression constant means more packets will be suppressed, which lowers overhead, but can also result in fewer Interests being satisfied when the node density is low.

The second set of experiments evaluates CCLF's performance with geo-location information for one pair of randomly chosen consumer and producer. We compare CCLF with VNDN [8] and Navigo [9], two location-based forwarding strategy designed for VANETs in NDN. In VNDN, the most distant node from the previous hop gets the highest priority in forwarding a packet and the other nodes overhearing that node's transmission will drop the packet. In Navigo, each node makes its forwarding decision based on the location of the data.

The third set of experiments evaluates CCLF's performance without geo-location information for one pair of randomly chosen consumer and producer. We compare CCLF with STRIVE [13], a centrality-based forwarding strategy for information-centric vehicular environments, to evaluate how well CCLF performs without using geo-location information. In STRIVE, instead of having each receiver make independent forwarding decisions, the sender selects a specific neighbor with high centrality (i.e., node-level satisfaction ratio) as the next hop for forwarding.

In the last set of experiments, we randomly choose 10 consumers and 5 producers (2 consumers per producer) in each experiment, and compare the performance of all the strategies. We use flooding as a baseline strategy for comparison in all the above experiments.

In each set of experiments, we vary the total number of vehicles to evaluate how CCLF performs under different vehicular traffic densities. For each setting of the vehicle count, we run ten experiments with different consumer-producer pairs. The simulation time is 600 seconds in each run except that we reduce it to 180 seconds in the last set of experiments due to their long running time.

6.3 Performance Metrics

We use the following metrics in our evaluation.

- **Protocol Overhead** is the total number of packets sent by all the nodes in the network. We further normalize it by the protocol overhead incurred by flooding.
- **Satisfaction Ratio** is the percentage of Interests from a consumer that are satisfied by Data packets received by the consumer.
- **Delay Stretch** is the ratio between two delays: (a) the delay from sending an Interest to receiving the matching Data packet in a non-flooding forwarding strategy, and (b) the delay in the flooding strategy.

We report the median value for each metric unless otherwise noted.

6.4 Selecting a Suitable Suppression Constant

To get a rough idea of what value the suppression constant K should have, we ran experiments with different vehicular traffic conditions in the two-lane and grid road networks, and measured the number of neighbors around every vehicle. Table 1 shows that the median number of neighbors is 1 - 4 (two-lane) and 1 - 11 (grid) in sparse vehicular traffic, 5 - 10 (two-lane) and 12 - 26 (grid) in medium vehicular traffic, and 10 - 15 (two-lane) and 26 - 34 (grid) in dense

Table 1: Number of Neighbors and Duplicate Suppression Probability in Various Traffic Conditions

Traffic Condition	Number of Vehicles/ Mile/ Lane	Median Number of Neighbors (two-Lane)	Median Number of Neighbors (Grid)	Duplicate Suppression Probability with $K = 0.04$ (two-Lane)	Duplicate Suppression Probability with $K = 0.04$ (Grid)
Sparse	1 - 12	1 - 4	1 - 11	4%-16%	4%-44%
Medium	13 - 30	5 - 10	12 - 26	20%-40%	48%-100%
Dense	31 - 40	10 - 15	26 - 34	40%-60%	100%

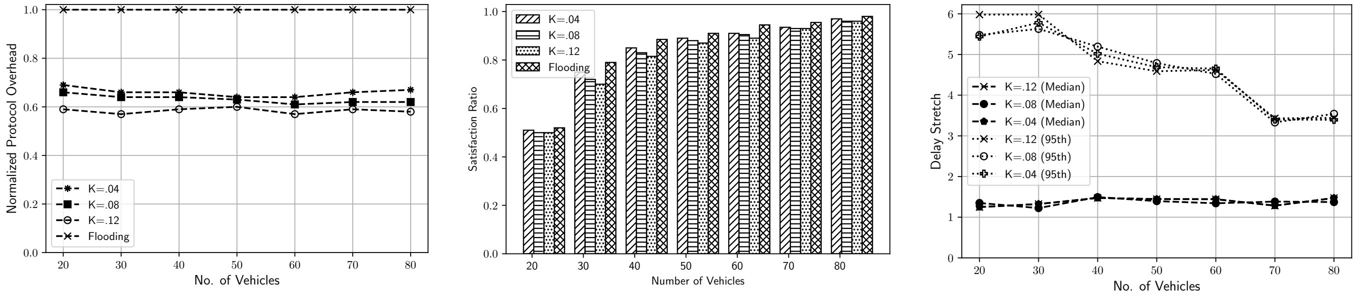


Figure 7: CCLF Performance Analysis for Various Suppression Constants

vehicular traffic. If K is 0.04, the suppression probability will be between 4% and 100% depending on the traffic density.

Figure 7 shows the effect of setting the suppression constant to be 0.04, 0.08 and 0.12 on the performance of CCLF in the grid road network. We varied the number of vehicles between 20 and 80 which correspond to a vehicular traffic density between 6.7 and 26.8 vehicles/mile/lane. In other words, *our experiment setting covers sparse, medium, and dense vehicular traffic conditions*. We can observe that, when the number of nodes is 20, $K = 0.04$ results in 15% more protocol overhead than $K = 0.12$. However, the difference in satisfaction ratio is small, i.e., 56.5% for $K = 0.04$ and 55% for $K = 0.12$. As we increase the number of vehicles, the difference in satisfaction ratio remains very small while $K = 0.12$ stays the lowest in protocol overhead. In addition, the median delay stretches for all the three K values are very similar. While $K = 0.12$ incurs a slightly higher 95th-percentile delay stretch when the number of vehicles is 20 and 30, the difference becomes negligible across various values of K when the number of vehicles is higher.

Based on the above results, we decided to use $K = 0.12$ for the remaining experiments, as it gives a satisfaction ratio as good as that of flooding and a reasonable delay stretch, but has the lowest protocol overhead among all the cases (about 41.2% less than flooding).

6.5 CCLF with Geo-location Information

Figure 8 shows the performance comparison among CCLF, VNDN, Navigo, and flooding, when geo-location information is available. We use the grid road network in this set of experiments. We can see that VNDN, Navigo, and CCLF have 34%, 36.7%, and 41.2% lower protocol overhead than flooding on average. Moreover, CCLF has higher satisfaction ratio than VNDN and Navigo. CCLF's median delay stretch is close to 1 (similar to that of VNDN and Navigo), which means the packets in CCLF typically follow paths that are as short as those found in flooding. Furthermore, CCLF's 95th-percentile delay stretch is lower than that of VNDN and Navigo.

Unlike VNDN and Navigo, CCLF's forwarding decision is not purely location-based. The content-connectivity based forwarding in conjunction with density-aware packet suppression enables CCLF to retrieve more data with lower delay and cost. Moreover, as the next experiment will show, CCLF works well when there is no geo-location information, which is not possible for VNDN and Navigo.

6.6 CCLF without Geo-location Information

Figure 9 shows the performance comparison among CCLF, STRIVE, and flooding, when geo-location information is not available. We use the grid road network in this set of experiments. As STRIVE forwards Interests to specific selected neighbors over unicast, it has much less protocol overhead than CCLF. Nevertheless, CCLF's protocol overhead is still 23.9% less than that of flooding on average.

Moreover, CCLF's satisfaction ratio is very close to that of flooding and it is 93.4% higher than that of STRIVE on average. Even when the node density is high in the 80-node case, STRIVE's satisfaction ratio is only 42%, while CCLF has a 95% satisfaction ratio. Furthermore, STRIVE has a much higher 95th-percentile delay stretch, more than than 11 for all the cases, while CCLF achieves a 95th-percentile delay stretch of about 5 with 20 nodes and close to 2 with 80 nodes. Although STRIVE has a slightly better median delay stretch than CCLF (both are less than 2), we note that 56.3% of the Interests in STRIVE are not satisfied on average. In other words, STRIVE cannot fetch the data more than 50% of the time. When STRIVE can fetch the data, it is mostly because the data producer is within one hop.

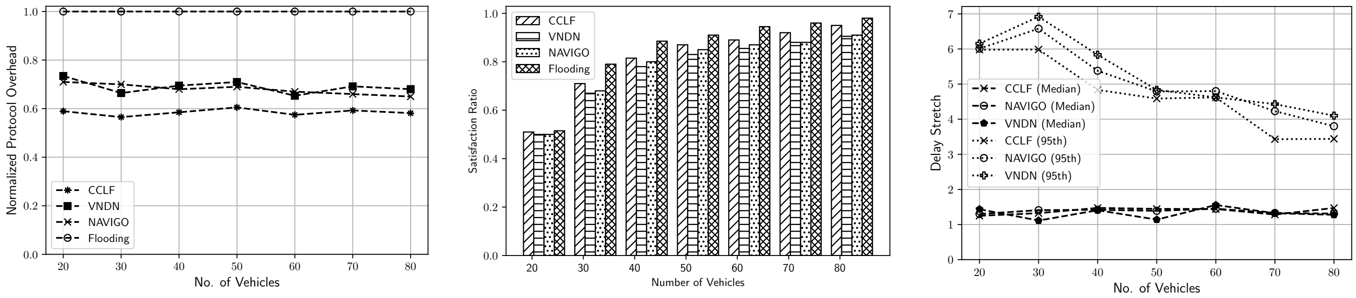


Figure 8: Performance of CCLF, VNDN, Navigo, and Flooding with Geo-location Information

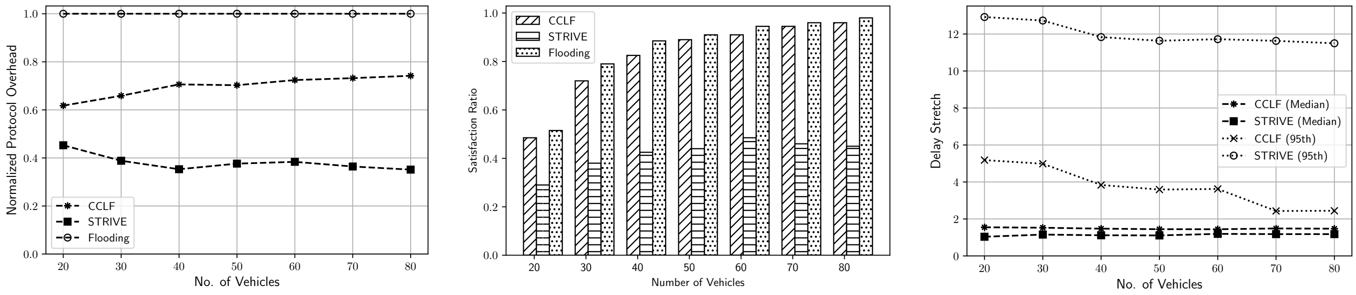


Figure 9: Performance of CCLF, STRIVE, and Flooding without Geo-location Information (one consumer and one producer)

6.7 Multiple Consumers and Multiple Producers

We run experiments with 5 producers and 10 consumers in the grid road network. Each producer serves data under one of five prefixes (/a/b/c, /a/c/d, /b/c/d, /b/c/e, and /c/d/e) to two consumers. We summarize the results in Table 2, 3, and 4. Table 2 shows that VNDN, Navigo, and CCLF have 33%, 36%, and 49% lower protocol overhead than flooding on average, which suggests that CCLF reduces bandwidth usage more effectively than VNDN and Navigo. Moreover, CCLF achieves better satisfaction ratio than VNDN and Navigo (Table 3). Finally, CCLF's delay stretch is similar to or lower than that of VNDN and Navigo (Table 4).

We also compared CCLF and STRIVE by disabling geo-location (see the results in Table 2, 3, and 4). Although STRIVE has lower protocol overhead (0.45) than CCLF (0.54), its satisfaction ratio (0.41) is much lower than that of (0.74). The low median stretch of STRIVE (1.11) is due to the fact that majority of the time it can only retrieve data one hop away. Note that it has a much higher 95th-percentile delay stretch (12.08) than that of CCLF (4.65). In summary, CCLF's broadcast scheme combined with prefix-wise centrality score and density-aware suppression makes it much more effective in data retrieval than STRIVE.

7 CONCLUSION

In this paper, we propose CCLF, an adaptive forwarding scheme for MANETs in NDN. CCLF enables nodes to make forwarding decisions based on per-prefix content connectivity and geo-location information, and suppress Interest/Data packets probablistically based on the number of neighbors in their communication range. Moreover, we developed the link adaption layer A-LAL for ad-hoc links. CCLF and A-LAL have been implemented in NFD and evaluated in ndnSIM. Our results show that it achieves a satisfaction ratio and median delay close to flooding, while its overhead is up to 49% less than flooding. Our future work includes a more in-depth study of different ways to calculate the forwarding timer and suppression probability, and a more extensive evaluation of CCLF, such as varying the content store size of NFD and the speed of the producers, and retrieving data from multiple producers serving the same content from different locations. We will also fine-tune the parameters in our algorithms and offer guidelines on how to set the parameters.

ACKNOWLEDGMENT

We are grateful for the invaluable suggestions made by our Shepherd, Marc Mosko, and the comments from the anonymous reviewers. This work was supported by NSF Award CNS-1629769.

REFERENCES

[1] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. 2012 (revised October 2012). *ndnSIM: NDN simulator for NS-3.* Technical Report NDN-0005. NDN Project.

[2] Alexander Afanasyev, Junxiao Shi, Beichuan Zhang, Lixia Zhang, Ilya Moiseenko, Yingdi Yu, Wentao Shang, Yi Huang, Jerald Paul Abraham, Steve DiBenedetto, et al. 2016. NFD developer's guide. *Dept. Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0021, Revision 7* (2016).

[3] Elizabeth M Belding-Royer, S Basagni, M Conti, S Giordano, and I Stojmenovic. 2004. Routing approaches in mobile ad hoc networks. *Mobile ad hoc networking* 1, 1 (2004), 275–300.

Table 2: Multiple Producers and Multiple Consumers: Normalized Protocol Overhead

No. Nodes	CCLF	CCLF (w/o location)	Navigo	VNDN	STRIVE
20	0.54	0.56	0.67	0.72	0.46
40	0.51	0.55	0.65	0.67	0.45
60	0.49	0.50	0.61	0.62	0.44
average	0.51	0.54	0.64	0.67	0.45

Table 3: Multiple Producers and Multiple Consumers: Satisfaction Ratio

No. Nodes	CCLF	CCLF (w/o location)	Navigo	VNDN	STRIVE
20	0.53	0.51	0.52	0.53	0.30
40	0.86	0.84	0.85	0.83	0.44
60	0.89	0.88	0.86	0.84	0.50
average	0.76	0.74	0.74	0.73	0.41

Table 4: Multiple Producers and Multiple Consumers: Delay Stretch (Median, 95th-Percentile)

No. Nodes	CCLF	CCLF (w/o location)	Navigo	VNDN	STRIVE
20	1.25, 5.98	1.19, 5.18	1.30, 6.01	1.43, 6.15	1.03, 12.90
40	1.47, 4.83	1.43, 4.47	1.42, 5.38	1.41, 5.83	1.12, 11.83
60	1.44, 4.62	1.41, 4.29	1.45, 4.86	1.56, 4.62	1.19, 11.52
average	1.39, 5.14	1.34, 4.65	1.39, 5.42	1.47, 5.53	1.11, 12.08

[4] Muktadir Chowdhury, Ashlesh Gawande, and Lan Wang. 2017. Secure information sharing among autonomous vehicles in NDN. In *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 15–26.

[5] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. 2003. Optimized link state routing protocol (OLSR). (2003).

[6] Jose Joaquin Garcia-Luna-Aceves and Marcelo Spohn. 1999. Source-tree routing in wireless networks. In *Proceedings. Seventh International Conference on Network Protocols*. IEEE, 273–282.

[7] Jiayu Gong, Cheng-Zhong Xu, and James Holle. 2007. Predictive directional greedy routing in vehicular ad hoc networks. In *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on*. IEEE, 2–2.

[8] Giulio Grassi, Davide Pesavento, Giovanni Pau, Rama Vuyyuru, Ryuji Wakikawa, and Lixia Zhang. 2014. VANET via named data networking. In *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 410–415.

[9] Giulio Grassi, Davide Pesavento, Giovanni Pau, Lixia Zhang, and Serge Fdida. 2015. Navigo: Interest forwarding by geolocations in vehicular named data networking. In *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 1–10.

[10] Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. 2008. Network simulations with the ns-3 simulator. *SIGCOMM demonstration* 14, 14 (2008), 527.

[11] David B Johnson and David A Maltz. 1996. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*. Springer, 153–181.

[12] Brad Karp and Hsiang-Tsung Kung. 2000. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 243–254.

[13] Junaid Ahmed Khan and Yacine Ghamri-Doudane. 2016. Strive: Socially-aware three-tier routing in information-centric vehicular environment. In *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–7.

[14] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. 2002. SUMO (Simulation of Urban MObility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*. 183–187.

[15] Meng Kuai, Xiaoyan Hong, and Qiangyuan Yu. 2016. Density-aware delay-tolerant interest forwarding in vehicular named data networking. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE, 1–5.

[16] Zhihua Lin, Meng Kuai, and Xiaoyan Hong. 2016. Reliable forwarding strategy in vehicular networks using NDN. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE, 1–5.

[17] Michael Meisel, Vasileios Pappas, and Lixia Zhang. 2010. Ad hoc networking via named data. In *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*. 3–8.

[18] Michael Meisel, Vasileios Pappas, and Lixia Zhang. 2010. Listen first, broadcast later: Topology-agnostic forwarding under high dynamics. In *Annual conference of international technology alliance in network and information science*. 8.

[19] Shree Murthy and Jose Joaquin Garcia-Luna-Aceves. 1996. An efficient routing protocol for wireless networks. *Mobile Networks and applications* 1, 2 (1996), 183–197.

[20] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. 2000. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record*, Vol. 1. IEEE, 70–74.

[21] Charles E Perkins and Pravin Bhagwat. 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM computer communication review*, Vol. 24. ACM, 234–244.

[22] Charles E Perkins and Elizabeth M Royer. 1999. Ad-hoc on-demand distance vector routing. In *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 90–100.

[23] Junxiao Shi and Beichuan Zhang. 2012. NDNLP: A link protocol for NDN. *NDN, NDN Technical Report NDN-0006* (2012).

[24] Ivan Stojmenovic. 2002. Position-based routing in ad hoc networks. *IEEE communications magazine* 40, 7 (2002), 128–134.

[25] Lucas Wang, Ryuji Wakikawa, Romain Kuntz, Rama Vuyyuru, and Lixia Zhang. 2012. Data naming in vehicle-to-vehicle communications. In *2012 Proceedings IEEE INFOCOM Workshops*. IEEE, 328–333.

[26] Yingdi Yu, Alexander Afanasyev, David Clark, Van Jacobson, Lixia Zhang, et al. 2015. Schematizing Trust in Named Data Networking. In *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 177–186.

[27] Yu-Ting Yu, Raheleh B Dilmaghani, Seraphin Calo, MY Sanadidi, and Mario Gerla. 2013. Interest propagation in named data manets. In *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 1118–1122.

[28] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. 2014. Named Data Networking. *ACM SIGCOMM Computer Communication Review (CCR)* 44, 3 (Jul 2014), 66–73.

[29] Zhiyi Zhang, Yingdi Yu, Haitao Zhang, Eric Newberry, Spyridon Mastorakis, Yanbiao Li, Alexander Afanasyev, and Lixia Zhang. 2018. An overview of security support in Named Data Networking. *IEEE Communications Magazine* 56, 11 (2018), 62–68.

[30] Zhenkai Zhu, Alexander Afanasyev, and Lixia Zhang. 2013. *A New Perspective on Mobility Support*. Technical Report NDN-0013. NDN.