

On Data-centric Forwarding in Mobile Ad-hoc Networks: Baseline Design and Simulation Analysis

Md Ashiqur Rahman
The University of Arizona
marahman@cs.arizona.edu

Beichuan Zhang
The University of Arizona
bzhang@cs.arizona.edu

Abstract—IP networking deals with end-to-end communication where the network layer routing protocols maintain the reachability from one address to another. However, challenging environments, such as mobile ad-hoc networks or MANETs, lead to frequent path failures and changes between the sender and receiver, incurring higher packet loss. The obligatory route setup and maintenance of a device-to-device stable path in MANETs incur significant *data retrieval delay and transmission overhead*. Such overhead exaggerates the packet loss manifold.

Named Data Networking (NDN) can avoid such delays and overhead and significantly improve the overall network performance. It does so with direct application-controlled named-data retrieval from any node in a network instead of reaching a specific IP address with protocol message exchange. However, existing works lack any explicit or systematic analysis to justify such claims. Our work analyzes the core NDN and IP architectures in a MANET at a baseline level. The extensive simulations show that NDN, when applied correctly, yields much lower data retrieval latency than IP and can lower the network transmission overhead in most cases. As a result, NDN's *stateful forwarder* can significantly increase the retrieval rate, offering a better trade-off at the network layer. Such performance comes from its caching, built-in multicast, and request aggregation without requiring an IP-like separate routing control plane.

Index Terms—network architecture, mobile ad-hoc networks, data-centric forwarding.

I. INTRODUCTION

The use of broadcast-based self-learning in an IP-based network is a reasonably common technique, broadcasting the first packet and waiting for a response. The returning response packet allows the network to learn a path between a sender-receiver pair for future packet exchange. However, unlike in a wired network, wireless mobile ad-hoc networks (MANET) present significant communication challenges. Not only do they have issues like wireless channel contention and interference, but they also suffer from frequent link breakage due to mobility. As a result, [1] proposed the AODV routing protocol for MANETs, designed on top of the broadcast-based self-learning philosophy. It was also a reactive routing protocol that minimizes the frequent routing message exchange in proactive routing protocols (e.g., OLSR [2]). Although proactive protocols exchange these messages to compute and maintain forwarding paths, they cause significant collision and contention in the wireless channel.

AODV reduces the proactive routing overhead by only sending control messages when there is traffic to transmit. However, path discovery and maintenance still lead to added

latency *before* actual data exchange can occur. Thus, latency also plays a crucial role in the overall MANET performance because wireless links can break between path discovery and data exchange. Higher latency and transmission overhead from discovery and maintenance lead to higher packet loss. While there has been lots of research in this area, most of the proposed solutions are still built on top of the traditional IP architecture, which requires the discovery and maintenance of a working path between the sending and receiving hosts. Maintaining such working paths is inherently challenging in a MANET environment.

Named Data Networking (NDN) [3] is an emerging Internet architecture that moves away from IP's address-based point-to-point communication model to a data-centric communication model. NDN packets do not carry any address; instead, they carry the name of the content that consumer nodes want to retrieve. A consumer sends an Interest (i.e., a request packet) to the network, and the network returns a Data (i.e., a response packet) whose name matches what was requested by the Interest. Data can come from any node in the network, including data producers and caches. NDN's broadcast-based self-learning can be a better fit for MANET than IP because (1) data discovery and link breakage detection can be achieved using the Interest/Data exchange without extra control messages to reduce collision, contention, and most importantly, retrieval latency; (2) since data can be cached and retrieved from any node, the Interest does not have to reach the original producer; thus, the impact of link/path breakage can be significantly mitigated; (3) multicast is achieved in NDN without any extra mechanism, which is a tremendous advantage since multicast in IP MANET is still a challenging problem.

The self-learning forwarding by [4] is also a broadcast-based discovery technique employing NDN as the underlying architecture. Their work shows how NDN can be deployed in a wired local network and suggests a promising MANET design. There have also been some research proposing NDN-based MANET routing and forwarding protocols [5]–[8]. However, the impact of the architecture on MANET forwarding performance has not been systematically investigated nor fully understood. In this paper, we conducted comprehensive simulations to compare the packet delivery performance of NDN and IP in a MANET at the network layer and bridge the gap between *promises* and *simulated results* for NDN in MANET. We use AODV [1] in IP and our proposed Data-

centric Ad-hoc Forwarding (DAF) in NDN as the network-layer protocol for routing/forwarding. These two protocols use similar mechanisms to discover paths or contents, and we do not apply any optimization techniques beyond the basic protocol. Therefore our results provide a baseline comparison between NDN and IP, reveal their architectural differences, and point to further optimizations.

Our simulations show DAF outperforming AODV in this aspect, especially under mobility. The main reason is that DAF does away with routing protocol by relying on Interest/Data exchange to discover data and detect failures. It significantly reduces the latency and, in most cases, the transmission overhead as well, which are two of the most challenging factors in an IP-based MANET routing. As a result, DAF vastly improves the data retrieval success rate by properly utilizing NDN as its underlying architecture. The benefits of NDN are even more prominent for many-to-one and many-to-many communication due to its built-in caching and multicast.

The evaluation of NDN and IP's performance in the MANET environment illustrates their architectural differences. It demonstrates that NDN's data-centric design is a better fit for MANET, where maintaining paths between two moving nodes is inherently challenging and sometimes unnecessary. With a new underlying architecture, many potential techniques can be considered to optimize NDN-based MANET performance in the future.

II. RELATED WORKS

Routing in MANETs has been one of the most challenging and well-studied topics in computer networks. [9] presents a comprehensive classification of ad hoc routing protocols, e.g., reactive, proactive, hybrid, geolocation aware, to name a few. DSDV [10], OLSR [2], GSR [11] are some examples from the proactive class. They exchange periodic routing updates to calculate address-based routes quickly. However, it incurs significant bandwidth usage, collision, and contention and is unsuitable under moderate to high mobility.

Reactive routing protocols such as DSR [12] and AODV [1] reduce the proactive routing overhead by exchanging routing information only before data transfer. However, they suffer from route setup and maintenance delay, which gets worse with increasing mobility. Hybrid approaches like ZRP [13], DST [14] either suffer from high computational overhead or single point of failure, respectively. GPSR [15] has quick route maintenance but high computational overhead from frequent neighbor updates and requires geolocation information. REEF [16] selects the next hop in a probabilistic manner but still needs to reach a specific destination address.

Another major challenge in IP networking is multicast support to reduce network overhead in group communication. Deploying multicast over IP in a regular wired network is already tricky enough [17]. The multicast tree needs to keep track of the endpoints or IP addresses and maintain a path towards them. Re-adjusting the tree when members join or leave introduces additional overhead, which is more frequent in MANETs, leading to further complexities [18]–[20]. [21]

dissects multicast over MANETs, pointing towards specialized protocols for specific environments making it even harder to find common ground. All these protocols try to minimize the routing overhead, but with the IP's address-based end-to-end path setup and maintenance, they all require substantial modifications and end up with intricate designs. Without a specialized multicast routing, IP requires dedicated flow per sender-receiver pair for the same content, leading to high transmission overhead and high collision and contention.

On the contrary, NDN's *data first* scheme supports multicast by design and enables caching to reduce network load and increase retrievability. [7] shows a comparative analysis between NDN and OLSR in MANETs but lacks any forwarding improvement and motivation of why and how NDN can be a better solution than IP. LFBL [22] uses wireless devices' broadcast nature as an advantage for their producer-independent data retrieval. However, it considers specific consumer and producer id-based communication and does not include caching and aggregation, two fundamental components of NDN's design. Implementation of LFBL [23] also shows a very high hop-by-hop forwarding delay, terrible for mobile networks. Performing pure broadcast in a wireless network also incurs the potential overhead of high CPU processing. Self-learning in [4] promises possible transmission event reduction. However, it yields very high-Interest broadcast and packet loss under mobility as it only performs Interest broadcast from a consumer upon Interest timeout. Thus many unicast Interests do not reach the next hop from mobility, and the consumer ends up sending a high number of broadcast Interests for discovery.

Forwarding or retransmission decision based on RTT measurement is present in existing NDN literature. The hyperbolic routing [24] uses an adaptive smoothed RTT-based forwarding (ASF) as a sub-module to minimize the sub-optimal path selection. However, overhead from periodically probing individual interfaces and high convergence time on link failure is unsuitable for mobile networks. DAF avoids such complexities and employs a quicker reaction mechanism for a better baseline analysis of NDN and IP architectures at the network level.

III. SYSTEM MODEL

Our MANET's architectural analysis employs the general system model following the random geometric graphs [25]. The only difference is that at the beginning of a simulation, instead of placing nodes at random points in the 2-dimensional plane, we place the stationary nodes randomly in a grid topology with either 10x5 or 10x10 setup for 50 and 100 nodes network, respectively. We use this setup to analyze the stationary wireless network behavior when a client or consumer can always reach a server or a producer. Each node has a fixed transmission range of 250 meters in diameter. When stationary (speed=0 m/s), we place the nodes 100 meters apart in X and Y coordinates to communicate only with their immediate neighbors at the top, bottom, left, and right (maximum four). For 100 nodes, we placed them in a 900m x 900m 2D grid (10x10) within an area of 1500m x 1500m.

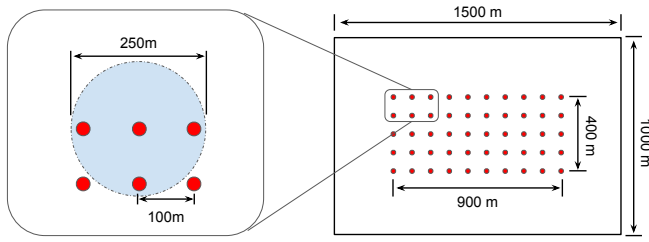


Fig. 1. Initial system setup for 50 nodes when all nodes are stationary. The right side shows the grid topology setup, and the left side shows inter-node distance and transmission range (colored circular area).

For 50 nodes, we placed them in a 900m x 400m grid (10x5) within an area of 1500m x 1000m. One example setup of the system model for 50 nodes topology is in Fig. 1.

In simulations with mobility, nodes start to move around at time $t > 0$, and their speed is also set to > 0 m/s. We use the random walk 2d mobility model, with 5 seconds of mobility, and without pause before setting the next random point. This basic mobility helps us to establish a baseline analysis. Scenario specific mobility model (e.g. manhattan or circular mobility) may impact the forwarding performance which we leave for future work. We also consider nodes' mobility in various simulations to range between 0 m/s to 8 m/s. Each node has only one wireless interface communicating over a single channel of IEEE 802.11b at 11Mbps. The system consists of 10 consumers or clients and either 10 or 2 producers or servers with the remaining nodes as packet forwarders. Thus we conduct experiments with 1-to-1, many-to-many, and many-to-1 communication. In IP, they are trivial as one client IP address can communicate with only one server address. However, in NDN, any Data node (producer or cache) can serve a consumer following its address and location-agnostic design.

IV. DESIGN BACKGROUNDS

While a network architecture provides the backbone model for communication, a routing protocol or forwarding strategy utilizes the backbone to make packet forwarding decisions. Thus, we go over the background and understand the general idea of AODV routing, the general NDN architecture, and some existing forwarding designs.

IP-AODV Routing: Like most IP routing for MANETs, AODV [1] also exchanges protocol messages to ensure an end-to-end path or route between source-destination addresses before sending data packets. These protocol messages ensure a *loop-free, shortest-path* packet forwarding between the source and destination addresses as the IP forwarding plane cannot detect any loop by itself. AODV follows a broadcast-based discovery mechanism to learn the shortest path. When a source node's application wants to initiate data exchange, AODV buffers the packet and broadcasts route-request (RREQ) to the network. On an RREQ, the destination address, or any network with an existing route to the destination, send back a route-reply (RREP) as unicast. The intermediate nodes and eventually, the source node, learn the shortest path to the destination IP address from multiple RREP coming from

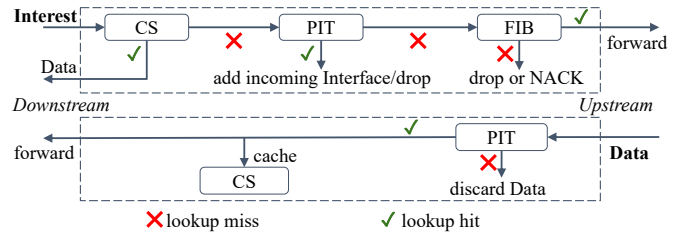


Fig. 2. Stateful forwarding pipeline in an NDN node.

possible multiple neighboring nodes. AODV then forwards the buffered packet over the learned path.

AODV ensures end-to-end path availability by exchanging periodic HELLO messages between nodes on an active route. In the case of a path-break from HELLO packet loss, network nodes issue route-error (RRER) packets back to the source node to trigger RREQ broadcast. A node never forwards data in the absence of a valid path. Thus, route setup and maintenance induce delay before and during data exchange.

Given that AODV offers a straightforward routing over the IP model without any further added complexities such as multicast or using geolocation information, we consider it as a suitable baseline.

Named Data Networking: Fig. 2 shows the general architecture of an NDN node. A consumer sends an Interest with a name, e.g., /news/image.png. Upon receiving, a node looks at its Content Store (CS) and replies with data on a name match. Otherwise, the node passes the Interest to its Pending Interest Table (PIT), and if there is a previous entry, it suppresses the Interest with necessary lifetime updates (e.g., 2s lifetime). Otherwise, the node creates a new PIT entry with the incoming interface and passes the Interest to its Forwarding Information Base (FIB). There, it performs a longest common prefix (LCP) match to existing FIB prefix entries. On a FIB match, e.g., /news, a node forwards the Interest through the associated interface(s) to the next-hop node(s). Otherwise, it either drops the Interest or sends a NACK (negative acknowledgment) back to the downstream. A NACK informs the downstream node that a next-hop in the upstream is not available and that it may try to forward the Interest to the next available FIB entry.

PIT also helps with loop-free Interest forwarding from a packet's NONCE. When a node receives data, it first checks the PIT, and on a match, passes the data to the CS; otherwise drops. After updating CS, the node forwards data to the associated downstream(s) and clears the PIT entry.

NDN flooding: An NDN flooding strategy follows the above design but broadcasts all Interests regardless of a FIB hit or miss. Thus, flooding can benefit from multiple paths between a consumer and data node(s), increasing data retrieval possibility. However, a data broadcast may satisfy multiple nodes' PITs resulting in a broadcast storm. It leads to significantly high collision and channel contention in the network. There is also no NACK involved as every packet transmission is broadcast. Flooding data in IP without a routing protocol is impossible as it lacks built-in loop detection.

NDN self-learning [4]: It can minimize the broadcast effect of flooding and is very close to IP-AODV in terms of learning and then performing unicast. A consumer broadcasts (discovery) Interests like RREQ until it *learns* a next-hop node towards data. Data travels as unicast towards the consumer like RREP in a hop-by-hop manner populating downstream nodes' FIBs with prefix announcement. A node then unicasts Interests to a next-hop on the shortest path. In a wired network, self-learning can send NACK back to a consumer, similar to RRER in AODV. It is one of the reasons we consider AODV as an IP-baseline in our network layer evaluation.

However, sending NACK in a MANET by self-learning is expensive for three reasons, a) a node does not know which upstream generated it, b) NACK can also get lost from mobility, causing Interest timeout, and c) it can add transmission overhead similar to protocol message exchange. Moreover, unlike AODV's in-network RREQ broadcast, self-learning drops unicast Interest on a FIB mismatch. It may also unicast an Interest to a stale FIB entry. As only the consumer makes discovery decisions, it suffers from many application-level timeouts, and the reaction to mobility gets delayed, which is undesirable.

To mitigate the challenges and limitations of flooding and self-learning, we propose the DAF strategy as an NDN-baseline to reduce excessive network transmission and offer better mobility support. In our simulations, DAF outperforms AODV with a better application success rate by avoiding separate routing protocol and thus lowering latency and, in most cases, the overall network transmission events. Such improvements in a MANET come from NDN's architectural advantages over IP, using data names for communication instead of IP addresses.

V. THE DATA-CENTRIC AD-HOC FORWARDING (DAF)

We present a data-centric ad-hoc forwarding (DAF) strategy that finds a balance between flooding and self-learning to establish an NDN baseline in MANETs. It is similar to the well-known broadcast-based discovery. Moreover, DAF only *manipulates* the NDN architecture innovatively to make it suitable for MANETs, just like AODV does the same in IP-based MANETs. Both AODV and DAF try to learn a next-hop over the shortest path to an IP-address and data name, respectively. However, AODV learns a route before sending or requesting data, whereas DAF uses NDN's Interest packet to broadcast a request to the network. DAF learns next-hop(s) towards the data node(s) like the self-learning strategy. However, instead of a consumer timeout-based broadcast, DAF relies on an in-network *feedback-reaction* mechanism and opportunistically reacts to possible failures in a decentralized manner.

Positive feedback: Receiving a Data packet denotes positive feedback. It means that a neighbor can serve a name prefix, thus send future Interest as a unicast to that next-hop. Positive feedback creates a new FIB entry or updates an existing one.

Negative feedback: When a learned next-hop fails to reply with data within a specific time, it acts as negative or implicit

feedback. It indicates a collision, contention, or link failure at any hop between the current and the data node. Negative feedback removes the expected next-hop from FIB and helps to avoid protocol messages to minimize packet collisions.

A. In-network RTT measurement and FIB-next-hop lifetime

Given that our DAF strategy does not use protocol messages for neighborhood maintenance, we need a way to detect path failure between consumer and data, potentially resulting from mobility. It allows any node to remove a next-hop neighbor towards the data from its FIB entry and fall-back into discovery mode for Interest forwarding when no other neighbor is present. Thus we set a timeout value or lifetime for each FIB next-hop entry. To calculate the lifetime of a FIB next-hop entry, we propose a decentralized in-network Round-Trip Time (RTT) measurement-based technique that alleviates NDN's concept of using the name as content identifier. It requires minimal changes to the existing NDN forwarding tables for storing the associated timestamps with negligible processing overhead. Any node in the network can perform calculations independent of the actual consumer, network diameter, and data nodes, which are some of NDN's data-centric design's primary goals. As this is a software-based technique, it applies to any wireless communication technology. A node calculates the RTT of the i -th data satisfying the i -th Interest locally as,

$$RTT_i = T(D_i) - T(R_i). \quad (1)$$

Here, $T(R_i)$ and $T(D_i)$ are the Interest transmission (request) and satisfaction (on Data) timestamps, respectively. Upon receiving D_i for R_i , a node initializes or updates the smoothed RTT ($SRTT(nh_p)$) and RTT variation ($RTTV(nh_p)$) of the FIB-next-hop entry, respectively, similar to [26],

$$SRTT(nh_p) = \begin{cases} RTT_i, & \text{if new entry} \\ \alpha \cdot SRTT(nh_p) + (1 - \alpha) \cdot RTT_i, & \text{otherwise.} \end{cases} \quad (2)$$

$$RTTV(nh_p) = \begin{cases} \frac{RTT_i}{2}, & \text{if new entry} \\ \beta \cdot RTTV(nh_p) + (1 - \beta) \cdot |RTT_i - SRTT(nh_p)|, & \text{otherwise.} \end{cases} \quad (3)$$

Here, nh_p is the FIB-next-hop which replied with data containing prefix $/p$, indicating $/p$ is reachable through that nh . We also set $\alpha = 7/8$ and $\beta = 3/4$. Performing such in-network RTT measurement in IP is a much complex task and requires substantial modifications. It is because IP routers are only aware of addresses and not the actual content names.

When a node unicasts an Interest (e.g., $/p/a$) to a FIB-next-hop nh_p , it starts a timer,

$$Timeout(nh_p) = SRTT(nh_p) + 4 \times RTTV(nh_p) \quad (4)$$

Here, $Timeout(nh_p)$ is the lifetime of nh_p in the FIB. The added variance in Equation 4 helps to minimize the RTT fluctuations. This fluctuation can occur very often in an NDN-based MANET as any node in the network with data can reply. If the unicast Interest forwarder does not get data back through that nh_p within $Timeout(nh_p)$, it treats it as negative feedback and removes the related FIB-next-hop entry.

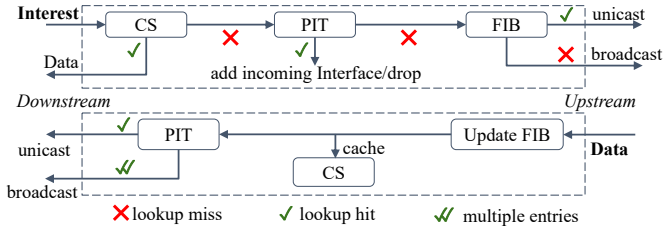


Fig. 3. Forwarding pipeline of the DAF strategy for MANETs.

B. Packet Forwarding

We show our DAF forwarding pipeline in Fig. 3 and discuss the modifications made over Fig. 2.

1) **Interest:** It is similar to self-learning [4]. However, instead of dropping the Interest on a FIB mismatch, DAF sends it out as broadcast, similar to in-network RREQ broadcast in AODV. When a node receives an Interest with name $/p/a$, it creates a PIT entry with tuple $\langle NONCE, ep, T(R_i) \rangle$. The ep is a unique identifier (e.g., MAC address) of the Interest forwarder (previous hop).

2) **Data:** A Data node replies upon receiving an Interest. On a broadcast Interest, it adds the producer prefix $/p$ with data. When a node receives a data, it performs a FIB lookup, measures the RTT_i for D_i on a match using Equation 1, and computes $SRTT(nh_p)$ and $RTTV(nh_p)$ using Equations 2 and 3 respectively. Afterwards, it creates or updates a FIB entry for $/p$ with tuple,

$$\langle nh_p, hc, T(D_i), RTT_i, SRTT(nh_p), RTTV(nh_p) \rangle.$$

Here, hc is the hop count between data and current node and is used to select the closet nh to $/p$ for future Interest(s). A node then updates the CS *before* PIT lookup for opportunistic caching. Afterward, it unicasts data if one ep in the PIT entry for $/p/a$ or broadcasts for more than one and ignores otherwise.

Note that with opportunistic caching on unsolicited data (no PIT entry), a FIB-next-hop entry may become stale by not being used for future unicast Interest. A periodic checker with a configurable timer (e.g., 1s) removes such stale entries using $T(D_i)$ as the starting time.

VI. PERFORMANCE ANALYSIS

This section analyzes *how* and *why* the NDN architecture can perform better than IP in MANETs at the network layer and compare our DAF strategy against NDN flooding, self-learning, and IP-AODV schemes. We exclude any name-based forwarding outside the core NDN design like [22]. We consider a client or consumer as a *requester* and a server or data-node as a *responder*. There are four performance metrics in our analysis,

a) **Expected success rate (ESR):** percentage of data packets retrieved by requester applications with respect to unique requests sent.

b) **Average latency:** average round-trip time (RTT) at the requester applications. We measure RTT with respect to the first request packet when retransmission is enabled.

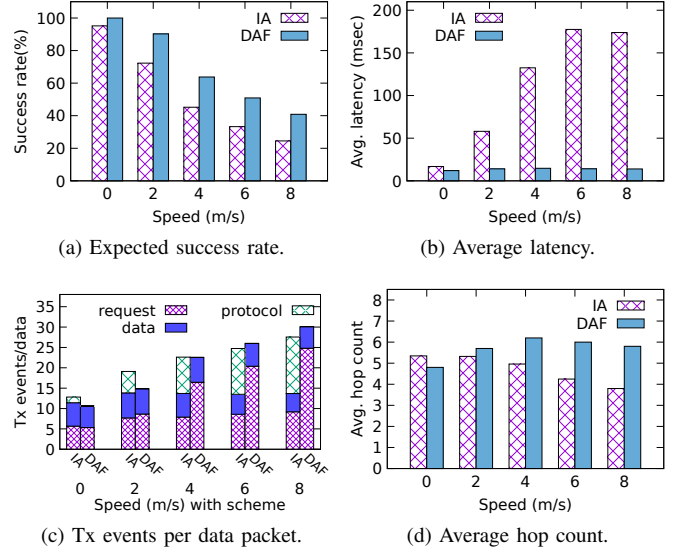


Fig. 4. Effect of mobility in IP-AODV (IA) and NDN-DAF (DAF) in 50 nodes network with 1-to-1 requester-responder communication at five requests per second. Retransmission is disabled and CS=200 packets per node for DAF.

c) **Network Tx-events:** the ratio of the total packets transmitted (Tx) by all nodes in a network to the total data packets retrieved by the requester applications; and,

d) **Average hop count:** average number of hops between requester-responder.

A forwarding strategy or routing protocol should yield high ESR with low latency, Tx-events, and average hop counts. Note that, for retrieving a single data packet at the application level, the number of data packet transmissions can be > 1 because of multi-hop traversal. We simulated the four schemes using ndnSIM [27]. Each requester expects 500 different data packets of 512 bytes using a constant bit-rate (CBR) *request-response* application at five requests per second. A requester may re-transmit ($RTx=0$ or max. 2) a request upon timeout. [22] explains the request-response application in IP-MANET. They argue that such an application is a proper fit for a MANET evaluation because sending data in a single direction with UDP CBR is indistinguishable from DDoS attacks. An IP request has twelve bytes, four for an increasing sequence number and eight for a timestamp.

A requester application starts randomly between 1.0 to 3.0 seconds and runs until it receives all 500 data packets or the final Interest times-out. RTS/CTS is disabled as a proper solution is not available for broadcast Interest and Data. In NDN, each node has a CS capacity of 0, 5, or 200 (default) packets. We also collect an average of 10 runs per simulation.

A. Effect of node mobility or speed in AODV and DAF

We start by analyzing the effect of mobility in IP-AODV and NDN-DAF in a 50 node network in Fig. 4 with a 1-to-1 communication (ten requesters and ten responders) and retransmissions disabled. In IP, this setup is apparent, with one requester address talking to a specific responder address. In NDN, each consumer requests the unique producer prefix's content, and each producer serves only one prefix content.

The analysis also shows the lower-bound forwarding/routing performance of DAF against AODV as there is no caching or Interest aggregation effect for multicast. We also see the latency effect in a MANET and how lower latency helps DAF outperform AODV in terms of expected success rate (ESR). The increasing speed increases the possibility of link breakage, and thus the ESR gradually goes down for both AODV and DAF in Fig. 4a. On average, DAF shows 38.1% more ESR than AODV. It is because, on average, DAF has 75.5% less latency in Fig. 4b and 5.08% fewer Tx-events per data retrieved in Fig. 4c than AODV.

At speed=0 m/s, AODV shows lower ESR than DAF because IP requires route set up *before* data exchange, which incurs added delay (Fig. 4b). Even the small number of protocol messages (Fig. 4c) is good enough to induce delay and collisions/contentions during the initial route setup, causing packet loss at the beginning of application startup, and thus AODV's ESR is about 95%. On the contrary, DAF can *learn* a next-hop towards data only with Interest-Data packets without any protocol message exchange and achieves 100% ESR.

AODV performance drops further under mobility because frequent link breakage increases protocol messages (Fig. 4c). It also adds more delay from RREQ exponential back-off during route setup/maintenance, as shown in Fig. 4b (maximum 25x than DAF). Higher latency in AODV also leads to nodes moving away before retrieving data. As a result, AODV retrieves most data when a requester is close to a responder, i.e., lower average hop-counts in Fig. 4d). On the other hand, DAF maintains its lower latency with nodes' increasing moving speed as it learns path towards data with Interest packets. Thus, DAF can retrieve data over a longer path, contributing to its higher transmission events than AODV in Fig. 4c under high mobility. We consider it a fair trade-off for obtaining higher ESR.

AODV packet loss at startup: We verify AODV's packet loss during the initial route setup by testing with a higher request sending rate in Fig. 5. We consider the cases where the nodes are either stationary or moving at 4 m/s. Fig. 5a shows that when nodes are stationary, AODV's ESR drops to around 90% at ten requests per second compared to 95% at five requests per second. Higher request rates cause more packet loss due to higher collision and contention, causing further route setup delay. The added latency in Fig. 5b at ten requests/s compared to 5 requests/s proves our claim.

Under mobility, however, AODV shows better ESR at a higher request rate. The protocol message exchange for *route maintenance* goes down (Fig. 5c) as request-response exchange updates the routes' states more frequently. Moreover, the network sparsity goes higher under mobility in our simulation model, leading to less chance of collision. Thus, the latency goes down slightly (Fig. 5b), helping to quickly fetch data even from longer paths as seen in (Fig. 5d).

A similar effect is also visible in DAF, where a higher Interest rate fetches data more frequently, updating FIB entries in the process under fewer collision possibilities. However, the latency goes up by a small margin (Fig. 5b) in 10 Interests/s

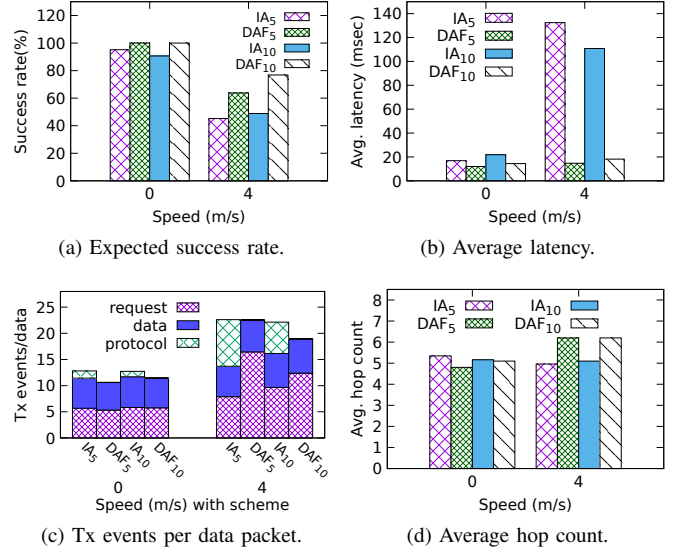


Fig. 5. Validating mobility effect in IP-AODV (IA) and NDN-DAF (DAF) with different request rates in 50 nodes scenario with 1-to-1 requester-responder communication. Subscript shows requests per second. Retransmission is disabled and CS=200 packets per node for DAF.

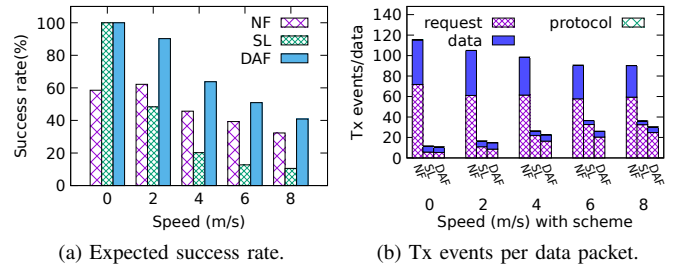


Fig. 6. Effect of mobility in DAF, Flooding (NF), and Self-learning (SL) in 50 nodes scenario with 1-to-1 communication at five requests per second. Retransmission is disabled and CS=200 packets per node.

compared to 5 Interests/s as the opportunistic forwarding can take longer paths (Fig. 5d) slightly.

To summarize the design differences, AODV forwards data packets only over an active route between the source and destination IP addresses mandated by the IP design. On the other hand, NDN's topology-agnostic design allows DAF to opportunistically explore and repair the path between consumer and data node with in-network Interest broadcast. It also enables DAF to avoid any explicit route setup before data exchange, as seen in IP-AODV, offering a better performance trade-off in MANETs.

B. NDN baseline for MANET

Here, we analyze our DAF baseline forwarding performance in MANET and why we need it by comparing it against NDN-flooding and self-learning.

1) Effect of mobility: Fig. 6 shows the limitations of flooding and self-learning and DAF's advantages in MANETs. The average latency and hop counts in 1-to-1 communication are very similar for all three schemes and thus not included in the figure. Fig. 6a shows that when nodes are stationary, the apparent effect of flooding leads to very high collision/contention

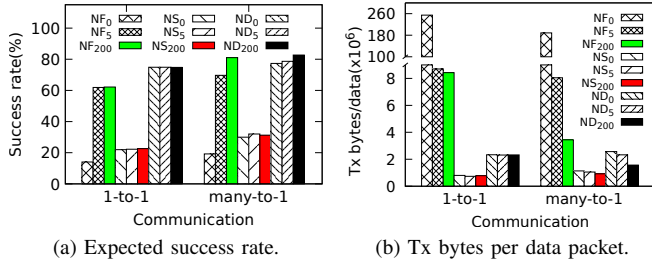


Fig. 7. Effect of content store (CS) in 1-to-1 and many-to-1 communication in 50 nodes scenario at speed=4 m/s with five Interests per second and application retransmissions (RTx) enabled (NDN-flooding: NF; NDN-self-learning: SL and NDN-DAF: DAF); Subscript shows CS capacity in packets.

resulting in only 59.8% ESR. Fig. 6b verifies the high packet transmission. Self-learning at speed=0 m/s performs similar to DAF, given it also learns the next-hop to data only once.

However, under mobility, self-learning's ESR degrades rapidly because only a consumer can broadcast Interest upon timeout at the application level. Flooding's ESR stays higher than self-learning because the network gets sparse under mobility in our simulation model, leading to less collision and better multi-path exploration. However, such performance comes at the cost of very high network transmission events.

DAF offers the right balance between flooding and self-learning with its RTT-based in-network unicast-broadcast interchange. It allows it to re-learn next-hops quickly and reduce overall network transmission cost. On average, DAF offers 42.28% and 177.93% more ESR than flooding and self-learning, respectively.

2) *Effect of Content Store (CS)*: To further strengthen our claim that DAF properly utilizes NDN's stateful forwarding plane, we analyze the effect of content-store (or CS) capacity in Fig. 7 at speed=10 m/s using a least-recently-used (LRU) eviction policy. Besides 1-to-1 communication, we also include the simulation scenario with many-to-1 communication where we have only two responders, one serving prefix /A, and another serving /B. Half of the requesters ask for data with prefix /A, and the rest ask for data with prefix /B. The goal is to observe the network behavior with caching and data multicast through Interest aggregation in NDN.

1-to-1 communication: Flooding at CS=0 has a 77.4% less success rate than CS=200 (Fig. 7a). Without CS, all Interests (including RTx) will always head for the actual producer leading to a broadcast storm. Fig. 7b shows very high Tx-bytes/data in flooding, causing higher collision/contention. However, caching significantly lowers flooding Tx-bytes and improves ESR. Note that at CS=5, flooding has lower Tx-bytes than CS=200. It is because large CS (late eviction) creates a higher number of disjoint paths than small CS (early eviction). Caching shows a minimal effect on self-learning and DAF as all the data packets are unicast in a 1-to-1 setup. Thus, only the nodes along a data-requester path can cache, and a retransmitted Interest hardly hits a CS under mobility.

Many-to-1 communication: Here, flooding ESR in all three CS capacities is higher than their respective 1-to-1 bars (Fig. 7a). The effect of Interest aggregation and multicast is

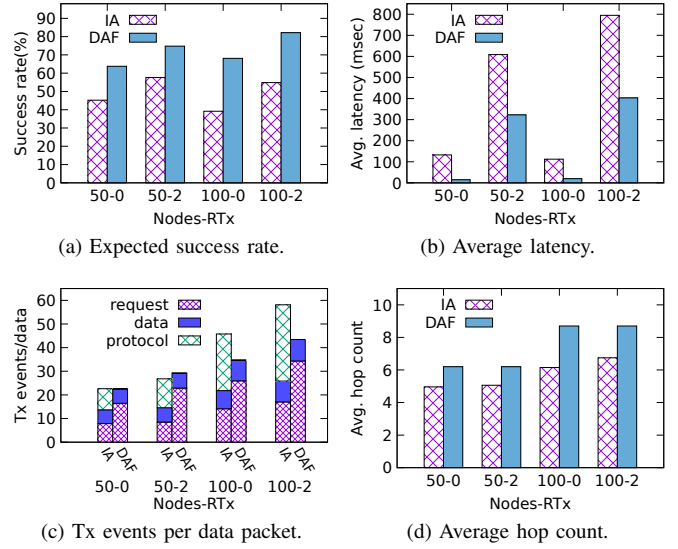


Fig. 8. Effect of network size on AODV (IA) and DAF with 1-to-1 communication at speed=4 m/s with five requests per second and application retransmission (RTx) enabled and disabled; CS=200 packets per node in DAF.

evident here, which significantly lowers the Tx-bytes (Fig. 7b). Self-learning again does not gain much success rate with caching as most Interests are lost or dropped. Thus Tx-bytes per data is also the lowest as a consumer retrieves most data packets when it is closer to the data node. Our DAF strategy appropriately reacts to link failure with in-network Interest broadcast on a FIB mismatch. The opportunistic caching from data broadcast with NDN multicast helps DAF improve ESR with increasing cache capacity, mitigating flooding's broadcast storms and self-learning's higher Interest loss.

The analysis above validates our proposed NDN baseline forwarding in a mobile network. In the following sections and subsections, we will only consider DAF and AODV for further comparative analysis between NDN and IP in MANETs.

C. Effect of network size or number of nodes

Fig. 8 shows the effect of network size (or nodes) and provides the worst-case comparison between the DAF and AODV under mobility. A cache hit is impossible without RTx unless a broadcast Interest finds cached data. A tuple X-Y in the x-axis shows X = the number of nodes and Y = RTx.

At similar RTx, the latency (Fig. 8b) and Tx-events (Fig. 8c) of both schemes go up when the network size increases. A requester is further away from a responder on a more extensive network (Fig. 8d). At RTx=2, both DAF and AODV get better ESR at the cost of extra Tx. The latency also increases as our application-level timeout is 2s, and we measure RTT with respect to the first request when RTx is enabled.

A more extensive network at the same RTx worsens AODV's ESR in Fig. 8a. Protocol messages also increase, introducing longer delays. However, DAF's Interest-Data-only feedback-reaction mechanism reduces the latency by an average of 69.7% compared to AODV. It also has fewer request packets than AODV's protocol and requests messages combined (Fig. 8c). With lower latency and Tx-events, DAF

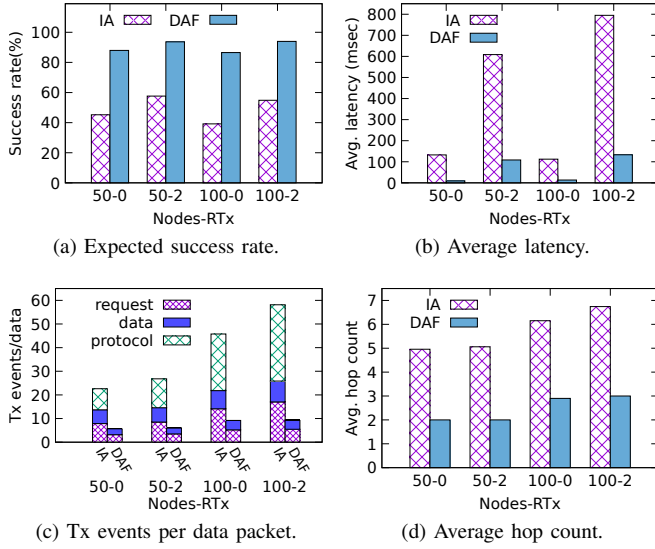


Fig. 9. Effect of many-to-many requester-responder communication on AODV (IA) and DAF at speed=4 m/s with five requests per second and application retransmission (RTx) enabled and disabled; CS=200 packets per node in DAF.

achieves an average of 45.46% more ESR (Fig. 8a) than AODV. Note that DAF shows an average of 27.03% higher hop count than AODV (Fig. 8d) as the lower latency enables it to retrieve data quicker under mobility. It leads to an average of 6.54% more transmitted-bytes per data than AODV (not shown in plots). However, we consider it a fair trade-off, given DAF reduces Tx-events and improves the success rate. Such performance comes from better utilizing the underlying NDN design.

D. Effect of many-to-many communication

Next, we consider that 5 out of 10 responders serve data with the same prefix (e.g., /A), while the other half serve another prefix (e.g., /B). Similarly, half of the requesters ask for data with prefix /A, and the rest for prefix /B. Even though various responders can serve the same data in IP, one requester address can always communicate with only one responder IP address. It provides a best-case example for NDN forwarding against IP routing in MANET. The results are in Fig. 9.

We can see that DAF outperforms AODV with higher ESR under all setups in Fig. 9a. The direct effect of NDN's location-agnostic design, built-in caching and multicast is now visible. Given the network's responsibility to retrieve data rather than reach a specific IP address, DAF's feedback-reaction mechanism learns the opportunistic shortest path to an eligible data node, including cached data. When a path to such a data node breaks (or a cache miss occurs), especially under mobility, DAF can quickly react to it and switch to another data node without any protocol message using its in-network RTT measurement. As a result, DAF achieves an average of 88.09% less latency compared to AODV. Opportunistic caching with data broadcast through NDN multicast helps DAF lower the hop count by 57.58% in Fig. 9d, which then helps lower the Tx-events by an average of 80.14% compared to AODV in Fig. 9c. AODV cannot use the multicast

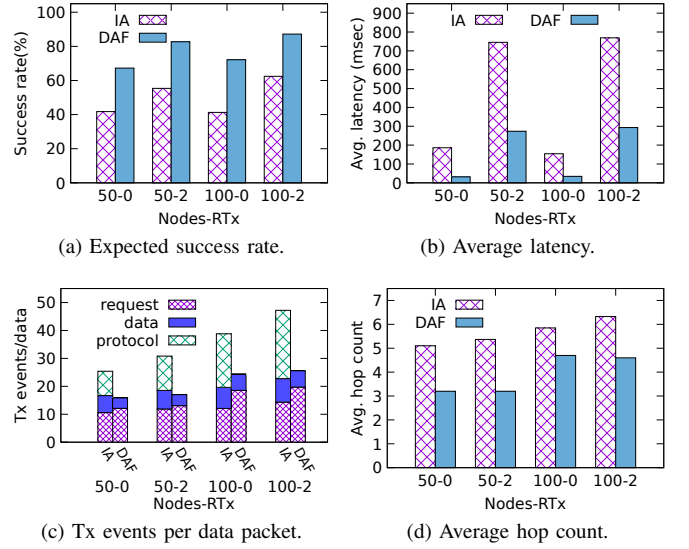


Fig. 10. Effect of many-to-1 requester-responder communication on AODV (IA) and DAF at speed=4 m/s with five requests per second and application retransmission (RTx) enabled and disabled; CS=200 packets per node in DAF.

and caching features without a highly-modified protocol. It provides the same results as 1-to-1 communication because the underlying address-based IP architecture lacks any application-level knowledge.

Combining the effects mentioned above, DAF achieves, on average, 82.98% more success rate (Fig. 9a) than AODV. Although DAF shows lower average hops than AODV, the possible sub-optimal longer path formation still results in higher hops than flooding and self-learning (not shown in plots). Note that flooding can achieve an average of 85% more ESR than AODV in the many-to-many scenario (not shown in plots) as opportunistic data caching can lead hop counts to be zero for many data packets. However, the broadcast storm still leads to very high Tx-events in the network, which DAF mitigates with its in-network unicast-broadcast interchanging.

E. Effect of many-to-1 communication

Here, we have only two responders, one serving prefix /A, and the other /B. Ten requesters ask for content in a similar way to Section. VI-D. The goal is to observe the network behavior with a higher traffic load at and near the actual data responder or "hot zones." Fig. 10 shows the simulation results.

DAF shows an average of 50.35% more success rate than AODV in Fig. 10a. Data broadcast from PIT aggregation reduces dedicated per downstream Tx and helps with opportunistic caching in overhearing neighbors. Thus, DAF reduces the traffic load at and near a producer with NDN's built-in multicast when multiple consumers ask for the same content. As a result, DAF's TX-events per data, latency, and hop count goes down by an average of 41.66% (Fig. 10c), 61.85% (Fig. 10b), and 27.31% (Fig. 10d), respectively, compared to AODV. DAF also has 29.98% lower Tx-bytes (not shown). Compared to DAF's underlying NDN architecture, the underlying IP model of AODV requires per requester dedicated flow for the same responder even though all requesters ask for the same content.

Thus we can say that NDN offers a better baseline forwarding than IP in a MANET.

VII. CHALLENGES AND FUTURE WORK

To maintain a baseline forwarder design, we do not consider a collision avoidance (random delay timer or back-off) mechanism in DAF. We believe that a better trade-off between success rate and latency is possible with proper modifications under mobility. Our baseline may also require further optimization, e.g., using directionality in vehicular networks, better reliability in disaster, or mission-critical networks. Actual device implementation of DAF, along with testing, including multiple channels and NICs, should provide valuable insights into NDN's real-world performance.

We also conducted experiments with in-network retransmissions in DAF in a 50 node 1-to-1 communication network to improve the success rate. However, it yielded feeble ESR gain ($< 1\%$) over baseline DAF and increased packet transmission per data packet by over 25%. This overhead occurs when multiple nodes get a timeout on their FIB-next-hops at the same time. Thus, each node ends up sending out in-network retransmission as broadcast. This effect produces a skyrocketing number of Interests, duplicated data packets from the producer(s) and cached node(s), leading to increased collision and contention. Thus we did not consider this approach in our analysis.

VIII. CONCLUSION

This work provides a baseline comparison between NDN and IP in the MANET environment. IP-based protocols need to maintain working paths between the sending and receiving mobile hosts, which is inherently challenging in MANET, causing significant latency and transmission overhead. By moving away from the point-to-point model to a data retrieval model, NDN no longer requires path maintenance between two specific endpoints; it allows retrieving requested data from anywhere in the network, thus enables caching, multicast, and stateful forwarding. Our DAF strategy is a baseline NDN-based MANET forwarding mechanism. It reduces the network latency and, in most cases, the transmission overhead, thus improving application packet retrieval over IP-AODV. While our work demonstrates NDN architecture's advantages for MANET, further improvements and optimizations to the baseline design are possible.

REFERENCES

- [1] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, ser. WMCSA '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 90–.
- [2] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, 2001, pp. 62–68.
- [3] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM Computer Communication Reviews*, Jun. 2014.
- [4] J. Shi, E. Newberry, and B. Zhang, "On broadcast-based self-learning in named data networking," in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, June 2017, pp. 1–9.
- [5] B. Etefia and L. Zhang, "Named data networking for military communication systems," in *2012 IEEE Aerospace Conference*. IEEE, 2012, pp. 1–7.
- [6] A. Azgin, R. Ravindran, and G. Wang, "Mobility study for named data networking in wireless access networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 3252–3257.
- [7] T. Teixeira, C. Wang, and M. Zink, "Can ndn perform better than olsr in wireless ad hoc networks?" *arXiv preprint arXiv:1609.06270*, 2016.
- [8] H. Khelifi, S. Luo, B. Nour, H. Mounghla, Y. Faheem, R. Hussain, and A. Ksentini, "Named data networking in vehicular ad hoc networks: State-of-the-art and challenges," *IEEE Communications Surveys & Tutorials*, 2019.
- [9] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in ad hoc networks: A survey," *Computer networks*, vol. 55, no. 13, pp. 3032–3080, 2011.
- [10] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, Oct. 1994.
- [11] T.-W. Chen and M. Gerla, "Global state routing: a new routing scheme for ad-hoc wireless networks," in *ICC '98. 1998 IEEE International Conference on Communications. Conference Record. Affiliated with SUPERCOMM'98 (Cat. No.98CH36220)*, vol. 1, June 1998, pp. 171–175 vol.1.
- [12] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Boston, MA: Springer US, 1996, pp. 153–181.
- [13] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Transactions on networking*, vol. 9, no. 4, pp. 427–438, 2001.
- [14] S. Radhakrishnan, G. Racherla, C. N. Sekharan, N. S. Rao, and S. G. Batsell, "Dst-a routing protocol for ad hoc networks using distributed spanning trees," in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, vol. 3. IEEE, 1999, pp. 1543–1547.
- [15] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: Association for Computing Machinery, 2000, pp. 243–254.
- [16] M. Conti, E. Gregori, and G. Maselli, "Reliable and efficient forwarding in ad hoc networks," *Ad Hoc Networks*, vol. 4, no. 3, pp. 398–415, 2006.
- [17] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service and architecture," *IEEE network*, vol. 14, no. 1, pp. 78–88, 2000.
- [18] S. Ziane and A. Mellouk, "Amdr: A reinforcement adaptive mean delay routing algorithm for manet," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, Nov 2007, pp. 726–730.
- [19] Y.-B. Ko and N. H. Vaidya, "Geotora: a protocol for geocasting in mobile ad hoc networks," in *Proceedings 2000 International Conference on Network Protocols*, Nov 2000, pp. 240–250.
- [20] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999, pp. 207–218.
- [21] C. de Morais Cordeiro, H. Gossain, and D. P. Agrawal, "Multicast over wireless mobile ad hoc networks: present and future directions," *IEEE network*, vol. 17, no. 1, pp. 52–59, 2003.
- [22] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," in *Proceedings of the Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture*, ser. MobiArch '10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 3–8.
- [23] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *Journal of Network and Computer Applications*, vol. 50, pp. 148–158, 2015.
- [24] V. Lehman, A. Gawande, B. Zhang, L. Zhang, R. Aldecoa, D. Krioukov, and L. Wang, "An experimental investigation of hyperbolic routing with a smart forwarding plane in ndn," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*. IEEE, 2016, pp. 1–10.
- [25] J. Dall and M. Christensen, "Random geometric graphs," *Physical review E*, vol. 66, no. 1, p. 016121, 2002.
- [26] V. Paxson, M. Allman, H. J. Chu, and M. Sargent, "Computing tcp's retransmission timer," 2011. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6298.txt>
- [27] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnSIM: an open-source simulator for NDN experimentation," *ACM Computer Communication Review*, Jul. 2017.