DFOR: A Hybrid Forwarding Framework with Opportunistic Recovery for Flying Ad Hoc Network

Liyou Deng
School of Computer Science and Engineering
Beihang University
Beijing, China
by2306002@buaa.edu.cn

Zhiyuan Wang
School of Cyber Science and Technology
Beihang University
Beijing, China
zhiyuanwang@buaa.edu.cn

Xiaohan Qiu
School of Computer Science and Engineering
Beihang University
Beijing, China
qiu.xh@buaa.edu.cn

Zichuan Zheng
School of Computer Science and Engineering
Beihang University
Beijing, China
zhengzichuan@buaa.edu.cn

Shan Zhang
School of Cyber Science and Technology
Beihang University
Beijing, China
zhangshan18@buaa.edu.cn

Mingsheng Tang
School of Cyber Science and Technology
Beihang University
Beijing, China
tangmingsheng@buaa.edu.cn

Fusang Zhang
School of Cyber Science and Technology
Beihang University
Beijing, China
zhangfusang@buaa.edu.cn

Hongbin Luo
School of Cyber Science and Technology
Beihang University
Beijing, China
luohb@buaa.edu.cn

Hao Song
Zhongguancun Laboratory
Beijing, China
songhao@zgclab.edu.cn

Binbin Ge
Zhongguancun Laboratory
Beijing, China
gebinbin@zgclab.edu.cn

***Abstract*—Unmanned Aerial Vehicle (UAV) swarms provide connectivity in dynamic environments, where frequent topology changes and unstable wireless links make reliable data transmission challenging. Existing forwarding schemes generally follow deterministic or opportunistic principles, with the former being vulnerable to link failures and the latter potentially introducing excessive overhead. To address this problem, this paper proposes Deterministic Forwarding with Opportunistic Recovery (DFOR), a hybrid forwarding framework that uses deterministic forwarding by default and activates local opportunistic recovery when forwarding over a link fails. Specifically, DFOR enables only neighboring nodes with link-specific forwarding information to participate in opportunistic recovery. DFOR then coordinates recovery nodes through ACK-based confirmation and timers, keeping opportunistic forwarding controlled while suppressing redundant transmissions. Moreover, DFOR is not restricted to a specific routing paradigm and can be instantiated under different routing paradigms by establishing forwarding information via native packet exchanges. Packet-level simulations evaluate DFOR against representative deterministic and opportunistic forwarding schemes under both host-centric and content-centric routing paradigms. The results show that DFOR improves packet delivery ratio by up to 27.08% over deterministic forwarding and 22.04% over opportunistic forwarding, while reducing unnecessary forwarding overhead and improving transmission effectiveness.**

***Index Terms*—Flying ad hoc network (FANET); Deterministic forwarding; Opportunistic forwarding; Routing paradigm**

I. Introduction

A. Background and Motivation

Flying ad hoc network (FANET) enables UAV nodes to establish flexible wireless connectivity without fixed infrastructure. It is widely considered for scenarios such as disaster rescue [1], agricultural monitoring [2], and emergency communication [3]. In these scenarios, UAV mobility and wireless channel variations cause frequent topology and link-quality changes. FANET routing mechanisms establish forwarding information for data delivery in such dynamic environments. Packet delivery then proceeds through local forwarding decisions governed by the corresponding forwarding principle.

Existing FANET routing mechanisms generally follow two forwarding principles. **First, most routing mechanisms adopt deterministic forwarding as their basic forwarding principle** [4]–[7]. This principle specifies a designated next-hop receiver for each packet transmission, thereby constraining packet delivery to a single designated forwarding path. This design simplifies forwarding decisions and limits redundant transmissions. However, packet delivery becomes vulnerable to failures along the designated forwarding path. When the designated receiver fails to receive a packet, forwarding progress is interrupted because no other receiver is allowed to continue forwarding immediately. **Second, some studies adopt opportunistic forwarding to exploit the broadcast nature of the wireless medium** [8]–[10]. A packet transmission can be received by multiple neighboring nodes rather than only one designated receiver. These neighboring receptions create additional forwarding opportunities and allow multiple receivers to serve as potential forwarders. Accordingly, packet delivery may proceed along multiple potential forwarding paths. This design improves robustness under unreliable links. However, it can introduce redundant transmissions because coordination among potential forwarders cannot consistently

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

suppress duplicate forwarding. Such redundancy can accumulate during subsequent forwarding, leading to path dispersion and excessive forwarding overhead.

The two forwarding principles reveal a tradeoff between forwarding efficiency and robustness in FANET. **Accordingly, this paper argues for the integration of deterministic and opportunistic forwarding to leverage their complementary strengths.** This integration is motivated by two considerations:

- In many packet transmissions, the established forwarding path remains usable. In such cases, deterministic forwarding is preferable because it avoids unnecessary forwarder coordination and reduces redundant transmissions.
- When packet forwarding over a local link is interrupted, neighboring nodes that have received the packet may provide immediate recovery opportunities. In such cases, opportunistic forwarding can exploit these receptions for local recovery.

To this end, this paper aims to develop a general hybrid forwarding framework that preserves deterministic forwarding for regular packet delivery and confines opportunistic forwarding to local recovery in FANET. To achieve this goal, it is essential to address the following key questions:

Question 1. *How can regular deterministic forwarding be coordinated with local opportunistic recovery?*

To address Question 1, potential recovery nodes need to be associated with specific links along the deterministic forwarding path. Based on this association, recovery nodes remain inactive when packet forwarding over the corresponding link succeeds and participate only when it fails. Such coordination relies on forwarding information established before data forwarding, raising the next key question:

Question 2. *How can the hybrid forwarding framework remain independent of a specific routing paradigm?*

In different routing paradigms, routing mechanisms establish forwarding information through paradigm-specific packet exchanges. Therefore, to address Question 2, it is necessary to identify a common packet role shared by these exchanges. Based on this common role, the framework allows forwarding information to be established in a paradigm-specific way while keeping the subsequent forwarding process unified.

B. Main Results and Key Contributions

The key contributions are summarized as follows.

- *A Hybrid Forwarding Framework for FANET:* This paper proposes Deterministic Forwarding with Opportunistic Recovery (DFOR) for data transmission in FANET. DFOR uses deterministic forwarding by default and activates local opportunistic recovery only when forwarding over a local link fails. In this way, DFOR preserves the efficiency of deterministic forwarding while exploiting opportunistic forwarding to recover from failures.
- *Link-Specific Opportunistic Recovery:* This paper introduces a link-specific recovery design that associates potential recovery nodes with individual forwarding links. Only neighboring nodes with corresponding forwarding information participate when forwarding over the associated link fails. DFOR further coordinates recovery nodes via ACK-based confirmation and timers, thereby suppressing redundant transmissions and keeping opportunistic forwarding under control.
- *Paradigm-Independent Instantiation:* This paper abstracts preceding packets as a common packet role for establishing forwarding information. Based on this abstraction, DFOR can be instantiated under different routing paradigms by reusing native packet exchanges to establish forwarding entries. This enables DFOR to support both host-centric and content-centric routing while keeping the subsequent DATA forwarding process unified.
- *Packet-Level Evaluation:* This paper implements DFOR under both host-centric and content-centric routing paradigms and evaluates it through packet-level simulations in OMNeT++. The results show that DFOR improves packet delivery ratio by up to 27.08% over deterministic forwarding and 22.04% over opportunistic forwarding, while reducing unnecessary forwarding overhead and improving transmission effectiveness.

II. Related Work

This section reviews the principles of deterministic and opportunistic forwarding in FANET routing mechanisms.

A. Deterministic Forwarding

Deterministic forwarding specifies a designated next-hop receiver for each packet transmission, making only that receiver eligible to forward the packet. As a data-plane property, it is widely adopted in FANET routing schemes under both host-centric and content-centric paradigms.

The host-centric routing paradigm organizes forwarding based on host identifiers, such as IP addresses or node identifiers, with deterministic forwarding governed by next-hop information in routing tables. Such information can be maintained proactively, as in OLSR [4], or established reactively on demand, as in AODV [5]. Link-quality metrics, such as expected transmission count (ETX) [11] and expected transmission time (ETT) [12], can further refine path selection.

The content-centric routing paradigm organizes forwarding based on content names. In this paradigm, request packets establish downstream forwarding state that governs deterministic forwarding for returning the content. Representative examples include NDNF [6], [13] and AIRDROP [7].

In summary, deterministic forwarding simplifies forwarding along a single designated path but remains sensitive to path reliability. Existing mechanisms typically rely on retransmission or route re-establishment to handle failures [5], [14], [15], which cannot immediately exploit neighboring nodes that have already received the packet for local recovery.

B. Opportunistic Forwarding

Opportunistic forwarding allows multiple receivers to serve as potential forwarders, rather than relying on a unique next-

TABLE I
COMPARISON BETWEEN DETERMINISTIC AND OPPORTUNISTIC FORWARDING IN FANET

Aspect	**Deterministic forwarding**	**Opportunistic forwarding**
Forwarding participants	Designated next-hop receiver	Eligible forwarder set
Forwarding path	Single designated path	Multiple potential forwarding paths
Failure handling	Retransmission or route re-establishment	Exploiting neighboring receptions for continued forwarding
Adaptability	Efficient but sensitive to path unreliability	Robust but may introduce redundant transmissions

hop receiver.[1] After receiving a packet, each node determines its forwarding eligibility based on routing or forwarding information. This flexibility creates multiple forwarding opportunities from a single transmission and improves robustness under unreliable links. Opportunistic forwarding schemes can also be classified into host-centric and content-centric paradigms.

In host-centric routing, opportunistic forwarding typically relies on forwarding eligibility specified by the upstream forwarder. The upstream forwarder conveys this through packet-carried candidate sets and priorities. ExOR [8] is a representative scheme in which the source encodes a prioritized candidate list ordered by estimated cost to the destination. Only listed nodes can relay the packet according to their priorities. SOAR [9] extends this design by allowing each forwarder to update the candidate list per hop and coordinating listed nodes with priority timers. PRIOR [16] introduces a prioritized forwarder that relays without backoff, while other eligible nodes coordinate around it. In general, host-centric opportunistic forwarding schemes require timely routing information to align forwarding eligibility with topology changes. In highly dynamic FANET, such information is more likely to become stale, thereby impairing forwarding coordination.

In content-centric routing, opportunistic forwarding typically relies on forwarding eligibility inferred at the receiver side. Such eligibility is derived from the local forwarding state associated with the requested content. LFBL [10] is a representative scheme that requires no explicit neighbor or next-hop knowledge. A receiver is eligible only if it is closer to the consumer than the upstream forwarder, and eligible receivers wait for a listening period to let a more suitable relay forward first. BREB [17] adapts routing by preferring the shortest-hop route under normal conditions and adjusting it when retransmissions indicate path degradation.

In summary, opportunistic forwarding improves robustness by exploiting overheard packets and link diversity. Most existing mechanisms rely on end-to-end progress information, such as hop count or cost to the destination, to prioritize forwarding decisions. However, redundancy at one hop may accumulate across subsequent forwarding, leading to path dispersion and increased forwarding overhead. A comparison between deterministic and opportunistic forwarding [18] suggests that the performance gains of opportunistic forwarding may not justify its coordination overhead under mobility. Table I summarizes the key differences between the two forwarding principles.

[1]Due to the broadcast nature of wireless communication, a single transmission may be successfully received by multiple neighboring nodes.

III. PROPOSED FORWARDING FRAMEWORK

This section presents the DFOR framework, including its packet fields, forwarding processes, and key components.

A. Packet Format

The DFOR framework involves three packet types: preceding packets, DATA packets, and ACK packets. This subsection defines the fields carried by these packets for establishing forwarding information and for DATA forwarding. All these packets contain *CID*, which associates packets and forwarding entries with the same forwarding instance. The composition of *CID* depends on the routing paradigm.

1) Preceding Packet: The preceding packet is disseminated before DATA forwarding to establish forwarding information. It propagates from the data consumer toward the data provider, opposite to the direction of subsequent DATA forwarding. Its concrete form depends on the underlying routing paradigm. For example, it can be instantiated as an RREP packet in AODV [5] or an Interest packet in NDN [13]. Section IV presents the detailed instantiation. During preceding-packet propagation, each relaying node updates the following fields:

$$\langle CID, prevId, curId, prio \rangle.$$

The fields *prevId* and *curId* identify the current propagation link, where *curId* denotes the current transmitter and *prevId* denotes its immediately preceding transmitter. The field *prio* records the forwarding priority of link $(prevId, curId)$.

2) DATA Packet: The DATA packet carries the data content and propagates from the data provider toward the data consumer. During DATA forwarding, each forwarder updates the following fields:

$$\langle CID, upId, downId, fwdId, prio \rangle.$$

The fields *upId* and *downId* identify the intended forwarding link, where *upId* denotes the upstream node and *downId* denotes the downstream node. The field *fwdId* identifies the actual transmitter of the DATA packet. The field *prio* records the forwarding priority of link $(fwdId, downId)$.

3) ACK Packet: The ACK packet is a short control packet used for forwarding confirmation and coordination. It carries the following fields:

$$\langle CID, upId, downId, ackId, prio \rangle.$$

The fields *upId* and *downId* identify the forwarding link to which the ACK refers. The field *ackId* identifies the ACK generator. The field *prio* carries the forwarding priority used by *ackId* for forwarding coordination when needed.

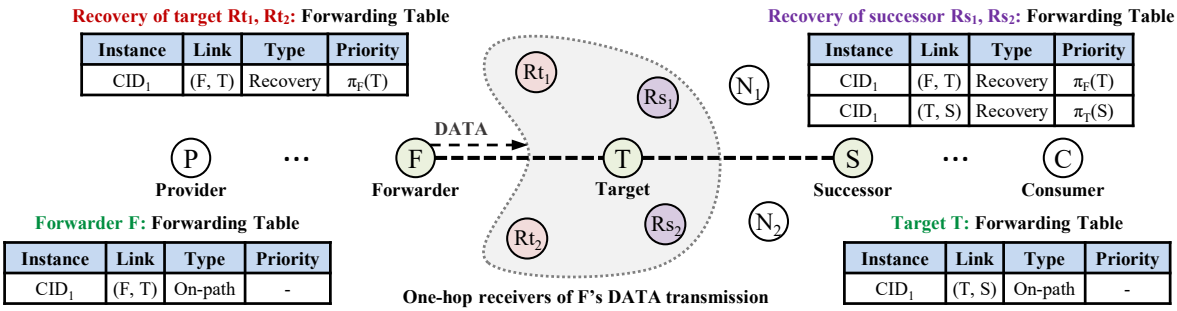


Fig. 1. Common DATA forwarding scenario under the DFOR framework

B. Deterministic Forwarding with Opportunistic Recovery

Fig. 1 shows a common DATA forwarding scenario over an end-to-end forwarding path from the data provider P to the data consumer C. For a local two-hop segment, nodes involved in the current DATA transmission are classified as follows:

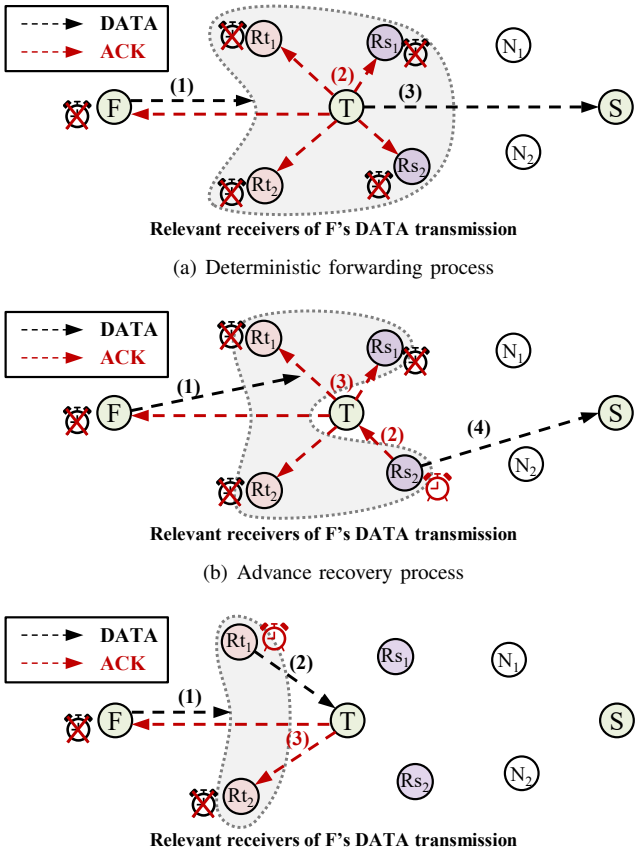
- **Forwarder** (F): the node that transmits the current DATA packet.
- **Target** (T): the intended next hop of the forwarder F.
- **Successor** (S): the intended next hop of the target T.
- **Recovery of successor** (Rs): a neighboring node that can assist recovery by forwarding the DATA packet toward S when forwarding over link (F, T) is not confirmed.
- **Recovery of target** (Rt): a neighboring node that can assist recovery by retransmitting the DATA packet toward T when forwarding over link (F, T) is not confirmed.

In Fig. 1, the dotted region denotes the relevant receivers of the DATA transmission from F, while the tables summarize forwarding information for the corresponding nodes.

Based on this scenario, the DFOR framework consists of three forwarding processes: deterministic forwarding, advance recovery, and retransmission recovery. In DFOR, forwarding over a link is considered failed when it remains unconfirmed within the corresponding confirmation interval. Deterministic forwarding serves as the default process, while advance recovery and retransmission recovery provide opportunistic alternatives when forwarding over link (F, T) is not confirmed.

1) Deterministic Forwarding: Fig. 2(a) illustrates the deterministic forwarding process. In Step (1), the forwarder F transmits the DATA packet to the target T. Afterward, F waits for forwarding confirmation, while neighboring recovery nodes $\mathrm{Rt}_1, \mathrm{Rt}_2, \mathrm{Rs}_1, \mathrm{Rs}_2$ that overhear the DATA packet start their recovery timers. In Step (2), after receiving the DATA packet, T broadcasts an ACK packet to confirm forwarding over link (F, T). This ACK suppresses the pending timers associated with this link. In Step (3), T forwards the DATA packet toward the successor S. The forwarding process then advances from link (F, T) to link (T, S).

2) Advance Recovery: Fig. 2(b) illustrates the advance recovery process. In Step (1), the forwarder F transmits the DATA packet, but the target T does not receive it. Neighboring recovery nodes that overhear the DATA packet start their recovery timers. Without an ACK from T, forwarding over



(a) Deterministic forwarding process

(b) Advance recovery process

(c) Retransmission recovery process

Fig. 2. DATA forwarding processes under the DFOR framework

link (F, T) remains unconfirmed. In Step (2), the recovery-of-successor node Rs_2 obtains the recovery opportunity and broadcasts an ACK packet to announce its recovery action. In Step (3), after receiving this ACK, T broadcasts a confirmation ACK for link (F, T). In Step (4), Rs_2 forwards the DATA packet toward the successor S. Thus, advance recovery bypasses T and advances DATA forwarding to link (T, S).

3) Retransmission Recovery: Fig. 2(c) illustrates the retransmission recovery process. Retransmission recovery is used when forwarding over link (F, T) remains unconfirmed and no recovery-of-successor node takes over the DATA

packet. In Step (1), the forwarder F transmits the DATA packet, but the target T does not receive it. Recovery-of-target nodes that overhear the DATA packet start their recovery timers. In Step (2), the recovery-of-target node Rt_1 obtains the recovery opportunity and retransmits the DATA packet toward T. In Step (3), after receiving the retransmitted DATA packet, T broadcasts an ACK packet to confirm forwarding over link (F, T). This ACK suppresses the remaining pending timers associated with link (F, T). The forwarding process is then restored over link (F, T).

These processes describe how DATA forwarding proceeds under the DFOR framework. Their execution relies on forwarding information, ACK-based confirmation, and recovery timers. The following subsections detail these components.

C. Information in Forwarding Table

Each node maintains a forwarding table FST that records the forwarding information required by the DFOR framework. As shown in Fig. 1, each entry in FST contains four attributes:

$$\langle \textit{Instance}, \textit{Link}, \textit{Type}, \textit{Priority} \rangle.$$

The attribute *Instance* denotes the forwarding instance associated with *CID* carried by packets. The attribute *Link* denotes the forwarding link associated with this entry. The attribute *Type* indicates how the entry is used in DATA forwarding, including on-path and recovery entries. The attribute *Priority* records the forwarding priority of the corresponding link. It can be derived from locally observable link-quality indicators, such as PRR, RSSI, or CSI [19], [20]. Each FST entry is also associated with an expiration time and is removed when the corresponding forwarding instance expires.

On-path entry. An on-path entry records the deterministic forwarding information for the current forwarding instance. It allows a node to identify the next hop for deterministic DATA forwarding. For example, the forwarding table of the forwarder F contains an on-path entry for link (F, T), indicating that F forwards the DATA packet to the target T. Similarly, the forwarding table of T contains an on-path entry for link (T, S), indicating the subsequent forwarding link.

Recovery entry. A recovery entry records the deterministic forwarding link that a neighboring node can assist with recovery. Instead of recording only the downstream endpoint, the entry identifies the link by both endpoints. This link-level representation associates recovery eligibility with a specific forwarding link. For example, recovery-of-target nodes Rt_1 and Rt_2 store a recovery entry for link (F, T), which enables them to retransmit the DATA packet toward the target T. Recovery-of-successor nodes Rs_1 and Rs_2 store recovery entries for both links (F, T) and (T, S), which enables them to advance DATA forwarding toward the successor S.

In addition to FST, each node maintains a link table LST that records current priority estimates of links to its neighbors. During recovery, the priority recorded in FST serves as the reference priority of the corresponding forwarding link. A recovery node compares this reference priority with the corresponding link priority in LST to determine whether it can provide a better forwarding opportunity. The establishment and update of these tables are presented in Section IV.

D. Explicit Confirmation Based on ACK Packet

During DATA forwarding, several opportunistic forwarding schemes exploit overhearing to coordinate forwarders and suppress duplicate transmissions [8]–[10]. In such designs, subsequent DATA transmissions can serve as implicit indications of forwarding progress, avoiding additional control packets. However, this DATA-based implicit confirmation is not sufficient for DFOR. A subsequent DATA transmission is directed toward the next forwarding link and may not be overheard by all nodes with pending recovery timers for the current link. For example, when a recovery-of-successor node advances the DATA packet toward the successor S, recovery-of-target nodes waiting for confirmation of link (F, T) may not overhear this DATA transmission. Their timers may therefore remain active, triggering unnecessary recovery attempts. Moreover, an ACK packet is shorter than a DATA packet, making explicit confirmation less vulnerable to transmission errors in wireless communication.

Therefore, the DFOR framework adopts ACK-based explicit confirmation to decouple forwarding confirmation from subsequent DATA forwarding. This explicit confirmation is associated with a specific forwarding link. For link (F, T), ACK packets are used in two forms:

- **Confirmation ACK.** The target T broadcasts this ACK to confirm link (F, T) when the link has been completed or validly bypassed.
- **Announcement ACK.** In advance recovery, a recovery-of-successor node broadcasts this ACK to inform T that it has taken over DATA forwarding toward the successor S. After receiving this announcement, T can broadcast a confirmation ACK for link (F, T).

Both ACK forms suppress pending timers associated with link (F, T). The link is completed when T receives the DATA packet in deterministic forwarding or retransmission recovery, and is validly bypassed when T receives an announcement ACK in advance recovery.

E. Recovery Timers

The DFOR framework uses recovery timers to coordinate opportunistic recovery when forwarding over a link remains unconfirmed. The timer design serves two purposes. First, it separates forwarding confirmation from recovery contention, so that recovery nodes act only after the expected confirmation is absent. Second, it coordinates competing recovery nodes by assigning earlier recovery opportunities to higher-priority nodes. In addition, advance recovery is given precedence over retransmission recovery because it can directly advance DATA forwarding toward the successor.

During recovery, a node starts a recovery timer only when its current link priority p recorded in LST is higher than the reference priority p_{ref} recorded in the corresponding recovery entry of FST. Let t_0 denote the time when the forwarder F transmits the DATA packet over the forwarding link (F, T).

For advance recovery, let τ_A^{conf} denote the advance confirmation interval and W_A^{cont} denote the advance contention window. A recovery-of-successor node first waits for confirmation of link (F, T) during interval $[t_0, t_0 + \tau_A^{\text{conf}}]$. If no such confirmation is observed, the node enters the advance contention window. For a recovery-of-successor node with priority p, the advance recovery time is

$$t_A(p) = t_0 + \tau_A^{\text{conf}} + \Delta_A(p),$$

where $\Delta_A(p) \in [0, W_A^{\text{cont}}]$ is the contention delay. For two recovery-of-successor nodes i and j with priorities p_i and p_j, respectively, the contention delays satisfy

$$p_i > p_j \Rightarrow \Delta_A(p_i) < \Delta_A(p_j).$$

For retransmission recovery, let τ_R^{conf} denote the retransmission confirmation interval and W_R^{cont} denote the retransmission contention window. To preserve the precedence of advance recovery, the retransmission confirmation interval satisfies

$$\tau_R^{\text{conf}} > \tau_A^{\text{conf}} + W_A^{\text{cont}}.$$

Thus, retransmission recovery is considered only after the advance recovery opportunity has elapsed. For a recovery-of-target node with priority p, the retransmission recovery time is

$$t_R(p) = t_0 + \tau_R^{\text{conf}} + \Delta_R(p),$$

where $\Delta_R(p) \in [0, W_R^{\text{cont}}]$ is the contention delay. For two recovery-of-target nodes i and j with priorities p_i and p_j, respectively, the contention delays satisfy

$$p_i > p_j \Rightarrow \Delta_R(p_i) < \Delta_R(p_j).$$

These constraints define the relative timing order required by DFOR. The confirmation intervals can be configured based on the expected response time, similar to ACK timeout settings in link-layer acknowledgment mechanisms [21]. The contention delays can be assigned within the contention windows, provided that the priority order is preserved.

F. Packet Processing Procedure

This subsection summarizes DATA/ACK processing and recovery-timer expiration handling in the DFOR framework.

DATA packet processing. Algorithm 1 summarizes DATA packet processing, where the current forwarding link is denoted as (F, T), with $\mathrm{F} = \textit{upId}$ and $\mathrm{T} = \textit{downId}$. Here, $\mathsf{FST}_{\text{on}}^{\text{nh}}[\textit{CID}]$ denotes the next hop recorded in the on-path entry, while $\mathsf{FST}_{\text{rec}}^{\text{prio}}[\textit{CID}, u, d]$ denotes the priority recorded in the recovery entry for link (u, d). The symbol $\perp$ denotes an unused priority field. Upon receiving or overhearing a DATA packet, the node updates link information for the DATA transmitter *fwdId*. If the node is the target, it confirms (F, T) by broadcasting an ACK packet. When an on-path entry exists, the node forwards the DATA packet toward the successor S and starts a retransmission timer for the new forwarding link. A non-target node participates in recovery only if it has a recovery entry for link (F, T). If the node has recovery information for the subsequent link (T, S) and the current priority of link (x, S) is higher than the recorded priority of link (T, S), it starts an advance timer. Otherwise, it starts a retransmission timer when the current priority of link (x, T) is higher than the recorded priority of link (F, T).

Algorithm 1: Processing DATA Packet

Input: Node x; DATA packet P^D carrying $\langle \textit{CID}, \textit{upId}, \textit{downId}, \textit{fwdId}, \textit{prio} \rangle$

$\mathrm{F} \leftarrow \textit{upId}$, $\mathrm{T} \leftarrow \textit{downId}$;
Update $\mathsf{LST}[\textit{fwdId}]$ with the link observation;
`// Target node`
if $x = \mathrm{T}$ **then**
  Generate P^A with $\langle \textit{CID}, \mathrm{F}, \mathrm{T}, x, \perp \rangle$;
  Broadcast P^A to confirm link (F, T);
  if $\mathsf{FST}_{\text{on}}^{\text{nh}}[\textit{CID}]$ exists **then**
    $\mathrm{S} \leftarrow \mathsf{FST}_{\text{on}}^{\text{nh}}[\textit{CID}]$;
    Update P^D with $\langle \textit{CID}, x, \mathrm{S}, x, \mathsf{LST}[\mathrm{S}] \rangle$;
    Forward P^D toward S;
    Start retransmission timer for P^D over (x, S);
`// Neighboring recovery node`
else if $\mathsf{FST}_{\text{rec}}^{\text{prio}}[\textit{CID}, \mathrm{F}, \mathrm{T}]$ exists **then**
  $p_{\text{ref}} \leftarrow \mathsf{FST}_{\text{rec}}^{\text{prio}}[\textit{CID}, \mathrm{F}, \mathrm{T}]$;
  if $\exists \mathrm{S} : \mathsf{FST}_{\text{rec}}^{\text{prio}}[\textit{CID}, \mathrm{T}, \mathrm{S}]$ exists and $\mathsf{LST}[\mathrm{S}] > \mathsf{FST}_{\text{rec}}^{\text{prio}}[\textit{CID}, \mathrm{T}, \mathrm{S}]$ **then**
    Start advance timer for P^D over (F, T) with S;
  else if $\mathsf{LST}[\mathrm{T}] > p_{\text{ref}}$ **then**
    Start retransmission timer for P^D over (F, T);

ACK packet processing. Algorithm 2 summarizes ACK packet processing. Upon receiving or overhearing an ACK packet, the node updates link information for the ACK generator *ackId*. If the node is the target T and the ACK packet is generated by another node, the ACK is treated as an announcement ACK from a recovery-of-successor node. The target then broadcasts a confirmation ACK for link (F, T). Regardless of its form, an ACK packet suppresses any pending timer associated with the same *CID* and forwarding link.

Algorithm 2: Processing ACK Packet

Input: Node x; ACK packet P^A carrying $\langle \textit{CID}, \textit{upId}, \textit{downId}, \textit{ackId}, \textit{prio} \rangle$

$\mathrm{F} \leftarrow \textit{upId}$, $\mathrm{T} \leftarrow \textit{downId}$;
Update $\mathsf{LST}[\textit{ackId}]$ with the link observation;
`// Recovery announcement`
if $x = \mathrm{T}$ and $\textit{ackId} \neq \mathrm{T}$ **then**
  Generate $P^{A'}$ with $\langle \textit{CID}, \mathrm{F}, \mathrm{T}, x, \perp \rangle$;
  Broadcast $P^{A'}$ to confirm link (F, T);
`// Timer suppression`
if a pending timer for (F, T) under *CID* exists **then**
  Suppress the pending timer;

Recovery-timer expiration. Algorithm 3 summarizes the procedure triggered by recovery-timer expiration. When an advance timer expires, the node broadcasts an ACK packet to announce its recovery action. After an ACK response interval, the node forwards the DATA packet toward the successor S and starts a retransmission timer for the new forwarding link (T, S). When a retransmission timer expires, the node retransmits the DATA packet toward T unless the retransmission limit has been reached.

Algorithm 3: Processing Recovery-Timer Expiration

Input: Node x; pending DATA packet P^D; expired recovery timer for *CID* and (F, T)

```
// Advance recovery
if advance timer expires then
    S ← successor associated with the advance timer;
    Generate P^A with ⟨CID, F, T, x, LST[S]⟩;
    Broadcast P^A to announce the recovery action;
    Wait for an ACK response interval;
    Update P^D with ⟨CID, T, S, x, LST[S]⟩;
    Forward P^D toward S;
    Start retransmission timer for P^D over (T, S);
// Retransmission recovery
if retransmission timer expires then
    if the retransmission limit is not reached then
        Update P^D with ⟨CID, F, T, x, LST[T]⟩;
        Retransmit P^D toward T;
        Start retransmission timer for P^D over (F, T);
```

IV. Framework Instantiation

This section introduces how DFOR is instantiated under different routing paradigms. As described in Section III, DFOR operates on the forwarding information recorded in FST and LST, while its DATA forwarding process remains independent of how such information is established. Therefore, each routing paradigm only needs to provide native packet exchanges for establishing the required forwarding entries. Once these entries are established, DATA forwarding proceeds in the same way.

A. Instantiation under Host-centric Routing

In host-centric routing, forwarding information guides packet delivery toward a specific destination and can be maintained proactively or established reactively. In proactive routing [4], [22], nodes periodically exchange control packets to update routing information. In reactive routing [5], [23], a source initiates route discovery on demand by disseminating a route request. The destination then returns a route reply along the reverse path, allowing intermediate nodes to establish forwarding information. In highly dynamic FANET, proactive routing may incur considerable update overhead, while reactive routing establishes routes only when needed. Therefore, this subsection instantiates DFOR under reactive routing.

In this instantiation, the route reply packet serves as the preceding packet. Here, *CID* can be instantiated by the source identifier, destination identifier, and sequence number. Specifically, as the route reply propagates back to the source, nodes that receive or overhear the packet establish forwarding information based on the carried fields and link observations. Algorithm 4 summarizes this process. For each preceding packet, the node updates LST for the current transmitter *curId* and records its neighbor state under *CID*. A node participating in preceding-packet propagation establishes an on-path entry for link (x, curId). If neighbor states for both *prevId* and *curId* exist under the same *CID*, the node establishes a recovery entry for link $(\mathit{curId}, \mathit{prevId})$ with the carried priority. When forwarding the preceding packet, the node updates the packet fields. DATA forwarding then follows the common DFOR procedure described in Section III-F.

Algorithm 4: Establishing Forwarding Information

Input: Node x; preceding packet P^P carrying $\langle \mathit{CID}, \mathit{prevId}, \mathit{curId}, \mathit{prio} \rangle$

```
Update LST[curId] with the link observation;
Record neighbor state for curId under CID;
// On-path entry
if x participates in preceding-packet propagation then
    Set on-path entry for link (x, curId) under CID;
// Recovery entry
if neighbor states for prevId and curId under CID exist then
    Set recovery entry for link (curId, prevId) under
      CID with priority prio;
// Packet update
if x forwards P^P then
    Update P^P with ⟨CID, curId, x, LST[curId]⟩;
```

B. Instantiation under Content-centric Routing

Content-centric routing retrieves data according to content names rather than host identifiers. Specifically, a consumer disseminates a request packet, also known as the Interest packet in NDN [13], to discover and retrieve the desired content from any available node in the network. This process generally consists of two stages:

- *Content Discovery*: The data consumer disseminates a request packet to discover the requested content. During request propagation, intermediate nodes record pending request state for subsequent content delivery.
- *Content Delivery*: The data provider returns the requested content toward the consumer along the recorded state.

Thus, content-centric routing does not require dedicated routing protocols [24], but relies on stateful forwarding strategies.

In the content-centric instantiation, the request packet serves as the preceding packet. Here, *CID* can be instantiated by the consumer identifier, content name, and sequence number. During request propagation, nodes establish forwarding information following Algorithm 4. Unlike host-centric routing, the request packet is not forwarded along a pre-established path.

Instead, its propagation creates reverse forwarding state toward the consumer. When a data provider returns the requested content, the DATA forwarding path is determined by the recorded state. Recovery entries are established in the same way, based on consecutive request-packet transmissions overheard under the same *CID*. The returned DATA packet is then processed according to the procedure described in Section III-F.

V. Packet-Level Simulation

This section evaluates DFOR through packet-level simulations in OMNeT++ [25]. We compare DFOR with representative deterministic and opportunistic forwarding schemes under both host-centric and content-centric routing paradigms.

A. Simulation Setup

Simulation settings. We simulate a 64-node UAV swarm in an $800\mathrm{m} \times 800\mathrm{m}$ area. Nodes follow the Random Waypoint model at speeds of 20m/s and 80m/s, representing moderate and high mobility conditions, respectively. We randomly select eight source-destination pairs with send intervals from 200ms to 800ms. The size of each DATA packet is 1000 bytes. The wireless interface uses IEEE 802.11ac on the 5GHz band with 12mW transmit power. Each simulation run lasts 900s, and we average the results over 60 runs.

Baselines. In the host-centric paradigm, we use AODV [5] as the underlying route discovery mechanism, whose basic data forwarding process is deterministic. Based on the routes established by AODV, we further implement a SOAR-like opportunistic forwarding scheme [9] and DFOR. These host-centric schemes share the same route discovery process and link-priority estimation but differ in their data forwarding logic. In the content-centric paradigm, we use NDNF [6], [13] and LFBL [10] as the deterministic and opportunistic forwarding schemes, respectively. All content-centric schemes share the same content discovery process while adopting different content delivery strategies. In both paradigms, we also include DFOR w/o recovery as an ablation where opportunistic recovery is disabled.

Forwarding settings. In the deterministic baseline schemes, we enable MAC-layer retransmission with the retransmission limit set to 7. In DFOR, the advance and retransmission confirmation intervals are 1ms and 6ms, respectively, and both contention windows are 4ms.

We evaluate the compared schemes in terms of packet delivery ratio (PDR), forwarding overhead, effective forwarding ratio, and per-hop delay.

B. Performance under Host-centric Routing Paradigm

Fig. 3 shows the performance of the compared forwarding schemes under the host-centric routing paradigm.

Packet delivery ratio. Fig. 3(a) and Fig. 3(b) show the PDR under different send intervals at speeds of 20m/s and 80m/s. DFOR consistently achieves the highest PDR at both speeds, while the gap between DFOR and DFOR w/o recovery confirms the necessity of local opportunistic recovery. Compared with deterministic AODV, DFOR improves PDR by up to 21.96% at 20m/s and 27.08% at 80m/s. This is because deterministic forwarding in AODV relies on a fixed next hop and is vulnerable when the selected forwarding link becomes unreliable. Compared with opportunistic SOAR, DFOR improves PDR by up to 11.90% at 20m/s and 11.16% at 80m/s. This shows that pure opportunistic forwarding does not always achieve high reliability, because redundant forwarding competition and path dispersion may reduce delivery effectiveness. We also observe that the PDR of AODV increases as the send interval decreases, because more frequent DATA transmissions improve route reuse before routes become stale.

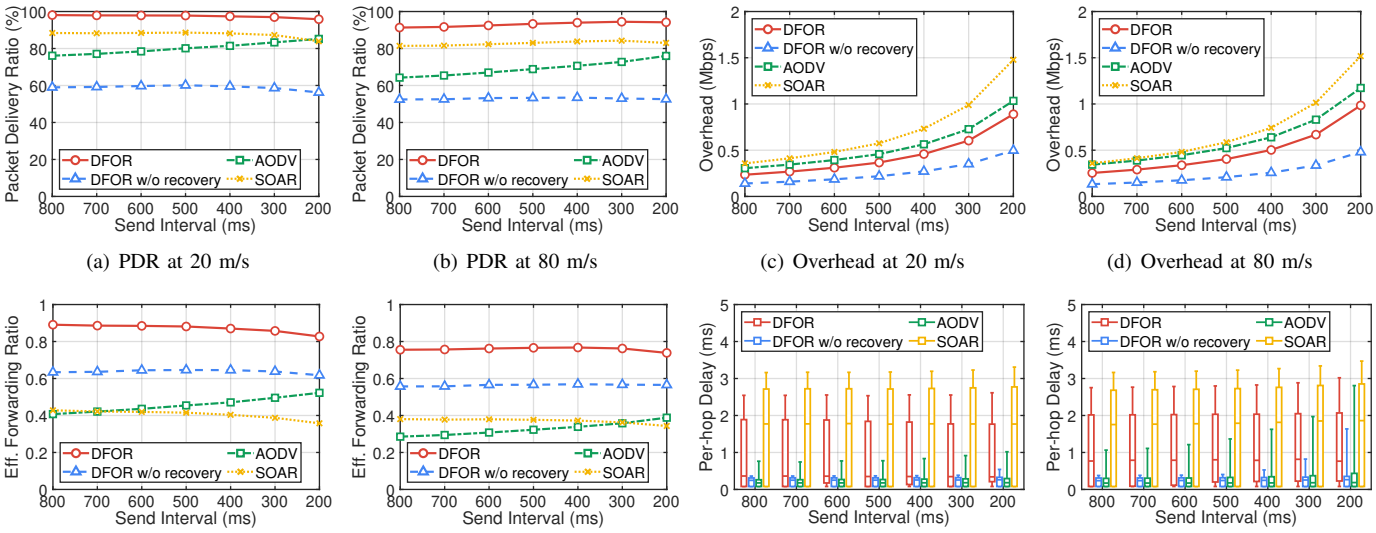
Forwarding overhead. Fig. 3(c) and Fig. 3(d) show the forwarding overhead under different send intervals. The forwarding overhead includes both DATA and ACK transmissions generated during DATA forwarding. The overhead of all schemes increases as the send interval decreases. DFOR incurs higher overhead than DFOR w/o recovery due to additional recovery transmissions, but this extra cost is limited relative to its PDR gain. Compared with deterministic AODV, DFOR reduces forwarding overhead by up to 0.14Mbps at 20m/s and 0.19Mbps at 80m/s. Compared with opportunistic SOAR, DFOR reduces forwarding overhead by up to 0.59Mbps at 20m/s and 0.53Mbps at 80m/s. This shows that DFOR improves delivery reliability without relying on aggressive redundant forwarding.

Effective forwarding ratio. Fig. 3(e) and Fig. 3(f) show the effective forwarding ratio under different send intervals. The effective forwarding ratio is defined as the fraction of DATA transmissions that belong to packets eventually delivered to the destination or consumer. DFOR consistently achieves the highest effective forwarding ratio at both speeds. Compared with deterministic AODV, DFOR improves the effective forwarding ratio by up to 0.48 at 20m/s and 0.47 at 80m/s. Compared with opportunistic SOAR, DFOR improves the effective forwarding ratio by up to 0.47 at 20m/s and 0.40 at 80m/s. This indicates that a larger fraction of DATA transmissions in DFOR directly contributes to successful delivery. In contrast, AODV suffers from failures along deterministic paths, while SOAR consumes more transmissions in redundant opportunistic forwarding.

Per-hop delay. Fig. 3(g) and Fig. 3(h) show the per-hop delay under different send intervals, with whiskers denoting the 10th and 90th percentiles. DFOR incurs additional delay to coordinate opportunistic recovery with staged recovery timers. Compared with deterministic AODV, DFOR introduces about 0.16–0.19ms additional median delay at 20m/s and about 0.58–0.63ms at 80m/s. This additional delay lets DFOR coordinate local recovery decisions, improving PDR while reducing unnecessary forwarding overhead. Compared with opportunistic SOAR, DFOR reduces the median delay by about 1.40–1.43ms at 20m/s and 0.98–1.10ms at 80m/s. Overall, DFOR introduces a controlled delay cost to improve delivery reliability and reduce forwarding overhead.

C. Performance under Content-centric Routing Paradigm

Fig. 4 shows the performance of the compared forwarding schemes under the content-centric routing paradigm.



(a) PDR at 20 m/s (b) PDR at 80 m/s (c) Overhead at 20 m/s (d) Overhead at 80 m/s

(e) Eff. forwarding ratio at 20 m/s (f) Eff. forwarding ratio at 80 m/s (g) Per-hop delay at 20 m/s (h) Per-hop delay at 80 m/s

Fig. 3. Performance under host-centric routing with different send intervals

Packet delivery ratio. Fig. 4(a) and Fig. 4(b) show the PDR under different send intervals at speeds of 20m/s and 80m/s. Consistent with the host-centric results, DFOR consistently achieves the highest PDR. Compared with deterministic NDNF, DFOR improves PDR by up to 22.06% at 20m/s and 22.12% at 80m/s. Compared with opportunistic LFBL, DFOR improves PDR by up to 22.01% at 20m/s and 22.04% at 80m/s. The results at the two speeds are close because content discovery refreshes forwarding state for each request, reducing the impact of route staleness. These results show that DFOR also provides reliable content delivery by combining deterministic forwarding with controlled opportunistic recovery.

Forwarding overhead. Fig. 4(c) and Fig. 4(d) show the forwarding overhead under different send intervals at speeds of 20m/s and 80m/s. DFOR incurs higher overhead than DFOR w/o recovery due to additional recovery transmissions. However, DFOR still maintains lower overhead than both NDNF and LFBL. Compared with deterministic NDNF, DFOR reduces forwarding overhead by up to 0.61Mbps at 20m/s and 0.60Mbps at 80m/s. The reduction is greater against opportunistic LFBL, reaching up to 4.44Mbps at both speeds. This indicates that DFOR improves content delivery reliability while avoiding the excessive DATA transmissions caused by pure opportunistic forwarding.

Effective forwarding ratio. Fig. 4(e) and Fig. 4(f) show the effective forwarding ratio under different send intervals. DFOR consistently achieves the highest effective forwarding ratio at both speeds. Compared with deterministic NDNF, DFOR improves the effective forwarding ratio by up to 0.51 at both speeds. Compared with opportunistic LFBL, DFOR improves the effective forwarding ratio by up to 0.70 at both speeds. This indicates that a larger fraction of DATA transmissions in DFOR contributes to successful content delivery. In contrast, LFBL has a much lower effective forwarding ratio due to redundant opportunistic forwarding.

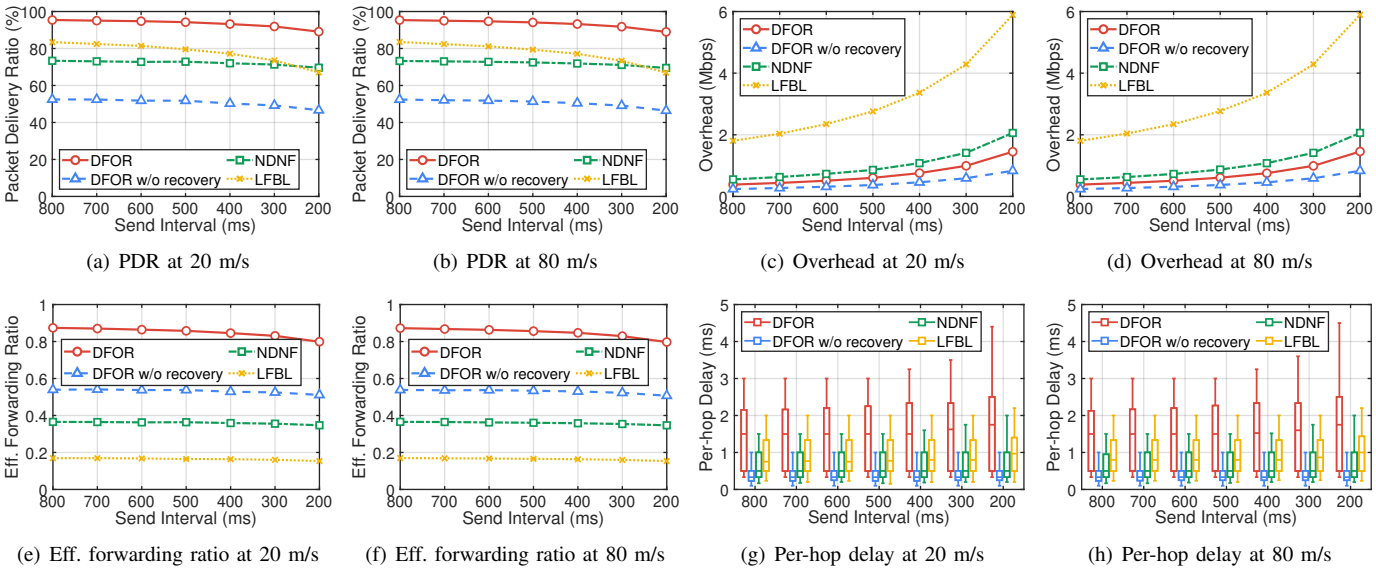
Per-hop delay. Fig. 4(g) and Fig. 4(h) show the per-hop delay under different send intervals. The median per-hop delay of DFOR remains within 1.50–1.75ms at both speeds. Compared with deterministic NDNF, DFOR introduces about 1.00–1.25ms additional delay at both speeds. Compared with opportunistic LFBL, DFOR introduces about 0.70–0.83ms additional delay at 20m/s and about 0.70–0.75ms at 80m/s. Similar to the host-centric results, this delay comes from staged recovery coordination. Together with the above performance results, this shows that DFOR trades a controlled delay cost for more reliable and efficient content delivery.

VI. Conclusion

This paper proposed DFOR, a hybrid forwarding framework for data transmission in highly dynamic UAV swarm networks. DFOR uses deterministic forwarding by default and activates local opportunistic recovery when forwarding over a link fails. To keep opportunistic forwarding controlled, DFOR associates recovery eligibility with link-specific forwarding information and coordinates recovery nodes through ACK-based confirmation and timers. By abstracting preceding packets as a common packet role, DFOR can be instantiated under both host-centric and content-centric routing paradigms while keeping DATA forwarding unified. Packet-level simulations show that DFOR improves packet delivery ratio by up to 27.08% over deterministic forwarding and 22.04% over opportunistic forwarding, while reducing unnecessary forwarding overhead and improving transmission effectiveness.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants U24B20128, 62225201, 62422201, and 62271019.



(a) PDR at 20 m/s (b) PDR at 80 m/s (c) Overhead at 20 m/s (d) Overhead at 80 m/s

(e) Eff. forwarding ratio at 20 m/s (f) Eff. forwarding ratio at 80 m/s (g) Per-hop delay at 20 m/s (h) Per-hop delay at 80 m/s

Fig. 4. Performance under content-centric routing with different send intervals

REFERENCES

[1] Y. Liu, J. Xie, C. Xing, S. Xie, and B. Ni, "Construction of FANETs for user coverage and information transmission in disaster rescue scenarios," *Computer Communications*, vol. 207, pp. 164–174, 2023.

[2] G. A. Kakamoukas, P. G. Sarigiannidis, and A. A. Economides, "FANETs in Agriculture-A routing protocol survey," *Internet of Things*, vol. 18, p. 100183, 2022.

[3] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8958–8976, 2019.

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings of IEEE International Multi Topic Conference (INMIC)*. IEEE, 2001, pp. 62–68.

[5] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, 1999, pp. 90–100.

[6] M. A. Rahman and B. Zhang, "On data-centric forwarding in mobile ad-hoc networks: Baseline design and simulation analysis," in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–9.

[7] D. Kim, J.-h. Kim, C. Moon, J. Choi, and I. Yeom, "Efficient content delivery in mobile ad-hoc networks using CCN," *Ad Hoc Networks*, vol. 36, pp. 81–99, 2016.

[8] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2005, pp. 133–144.

[9] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu, "SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 12, pp. 1622–1635, 2009.

[10] M. Meisel, V. Pappas, and L. Zhang, "Listen first, broadcast later: Topology-agnostic forwarding under high dynamics," in *Annual Conference of International Technology Alliance in Network and Information Science*, 2010, p. 8.

[11] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, 2003, pp. 134–146.

[12] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, 2004, pp. 114–128.

[13] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, 2009, pp. 1–12.

[14] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[15] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[16] T. Yamazaki, R. Yamamoto, T. Miyoshi, T. Asaka, and Y. Tanaka, "PRIOR: prioritized forwarding for opportunistic routing," *IEICE Transactions on Communications*, vol. 100, no. 1, pp. 28–41, 2017.

[17] H. Han, M. Wu, Q. Hu, and N. Wang, "Best route, error broadcast: a content-centric forwarding protocol for MANETs," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. IEEE, 2014, pp. 1–5.

[18] J. Kim and S. Bohacek, "A comparison of opportunistic and deterministic forwarding in mobile multihop wireless networks," in *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, 2007, pp. 9–16.

[19] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11 b mesh network," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2004, pp. 121–132.

[20] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 159–170, 2010.

[21] I. Tinnirello, G. Bianchi, and Y. Xiao, "Refinements on IEEE 802.11 distributed coordination function modeling approaches," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1055–1067, 2009.

[22] J. Chroboczek and D. Schinazi, "The Babel Routing Protocol," RFC Editor, RFC 8966, Jan. 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc8966

[23] C. E. Perkins, J. Dowdell, L. Steenbrink, and V. Pritchard, "Ad Hoc On-demand Distance Vector Version 2 (AODVv2) Routing," Internet Engineering Task Force, Internet-Draft draft-perkins-manet-aodvv2-06, Jun. 2025, Work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-perkins-manet-aodvv2/06/

[24] M. Farkhana, A. A. Hanan, H. Suhaidi, K. A. Tajudin, and Z. K. Zuhairi, "Energy conservation of content routing through wireless broadcast control in NDN based MANET: A review," *Journal of Network and Computer Applications*, vol. 131, pp. 109–132, 2019.

[25] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*. Springer, 2010, pp. 35–59.