Delay-Controlled Bidirectional Traffic Setup Scheme to Enhance the Network Coding Opportunity in Real-Time Industrial IoT Networks

Yunseong Lee, Taeyun Ha, Abdallah Khreishah, Wonjong Noh and Sungrae Cho

***Abstract*—Recently, network coding has become a promising transmission approach to support high throughput and low latency in distributed multi-hop networks. In this paper, we develop a delay-controlled distributed route establishment scheme that can provide maximal bidirectional transmission to enhance network coding gain while satisfying a time-critical route setup. The scheme is called network coding-aware delayed store and forwarding (NC-DSF). It delays the received route information packets before forwarding them according to the link status and network topology. We propose a tight delay function derived using a strict end-to-end delay bound for delay control. Subsequently, we suggest a relaxed delay function derived using realistic and practical conditions. Finally, we propose a load-weighted delay function considering the trade-off between bidirectionality and network-load balancing. The simulations confirm that the proposed scheme offers increased throughput and decreased latency in mesh and random multi-hop networks. The proposed transmission scheme, NC-DSF, can be efficiently employed in future industrial internet of things networks requiring a time-constrained route setup, high throughput, and low latency.**

***Index Terms*—Bidirectional traffic setup, delay-controlled flooding, opportunistic network coding, real-time industrial IoT.**

I. Introduction

The patterns of application services, networks, and computing change very rapidly. First, the rapid improvement of networks and end-systems has led to changes in services from simple applications to intelligent multimedia applications. Second, improved ubiquitous interoperability and convergence technologies have led to changes in networks, from cellular and Wi-Fi-based networks to heterogeneous networks including device-to-device and ad hoc networks. Finally, new software-defined radio, resource virtualization, and network security technologies have led to user-oriented computing platforms.

With these changes, various smart internet of things (IoT) services such as networked games, health care, home automation, virtual reality multimedia, artificial intelligence, and robots are becoming very important. In particular, industrial IoT (IIoT) services are receiving increasing attention. Especially in multi-hop and massive IIoT environments, IIoT applications periodically generate and transmit more packets, such as sensing data, video surveillance data, and production control data than the general IoT environment [1]. Moreover, they are susceptible to latency for work accuracy and safety, and demand time-critical transmission path establishments.

These trends can significantly increase network interference and saturate the available wireless resources. To address these issues, some researchers [2]–[4] have proposed transmission schemes based on interference avoidance. However, they exhibit inefficient resource usage and increase the computing complexity as the number of nodes increases. Therefore, some studies have proposed interference exploitation transmission approaches based on network coding schemes [5]- [6], [7]-[8] with a successive interference cancellation technique at the physical layer [9]–[12]. These transmission schemes based on network coding improve the spatial-reuse opportunity and network throughput, and the opportunity and performance gain increase as wireless IIoT networks become denser and more extensive. However, some crucial issues remain for network coding, such as network coding-aware routing, packet encoding and decoding, signal synchronization, information caching, error correction, and resource allocation. In this paper, we focus on a new network coding-aware routing control that can efficiently exploit network coding transmission opportunities while supporting the time-constrained route setup in IIoT networks.

II. Related Work

Many network coding-aware routing protocols have been proposed. Katti et al. [13] proposed the first network coding-aware routing scheme, an exclusive-OR (XOR) operation-based routing scheme called COPE. It introduces opportunistic coding and aims to improve throughput through network coding opportunities in wireless mesh networks. Wei et al. [14] proposed a network coding-aware routing protocol for lossy networks based on the ad hoc on-demand distance vector (AODV) mechanism. The protocol considers the potential coding opportunity and link delivery ratio for optimal route selection. Singh et al. [15] proposed an opportunistic network coding-based data transmission scheme. The scheme aims to determine a route with the minimum number of hops to the sink node using minimum Wiener spanning tree and artificial bee-colony optimization to reduce battery consumption. Han

This research was supported by Hallym University Research Fund 2022 (HRF-202201-002). *(Corresponding authors: Wonjong Noh and Sungrae Cho)*

Yunseong Lee, Taeyun Ha and S. Cho are with the School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea. *(e-mail: {yslee, tyha}@uclab.re.kr, srcho@cau.ac.kr)*

W. Noh is with the School of Software, Hallym University, Chuncheon 24252, Republic of Korea. *(e-mail: wonjong.noh@hallym.ac.kr)*

Abdallah Khreishah is with Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA. *(e-mail: abdallah@njit.edu)*

TABLE I
SUMMARY OF KEY FEATURES AND DIFFERENCES

Ref.	Routing	Performance metrics	Network coding	Delay constraint	Load-balancing	Features
[13]	Passive	Throughput	PLNC	✗	✗	XOR
[14]	Passive	Throughput	PLNC	✗	✗	XOR, lossy network
[15]	Passive	Latency, throughput, energy consumption	PLNC	✗	✗	XOR, ABC, MWST
[16]	Passive	Latency	RLNC	✗	✗	ACO, multipath transmission
[17]	Passive	Energy consumption, latency, throughput	RLNC	✗	✗	Shortest path with backpressure
[18]	Passive	Joint data security and availability	RLNC	✗	✗	Disjoint path routing
[19]	Passive	Energy consumption	RLNC	✗	✗	Link correlation mining
[20]	Passive	Energy consumption	PLNC/RLNC	✗	✗	Minimize the number of network coding nodes
[21]	Passive	Throughput	PLNC	✗	✗	Priority-based forwarding, XOR
[22]	Passive	Throughput	PLNC	✗	✗	Queue length as routing metric, XOR
[23], [24]	Passive	Latency, delay violation probability	RLNC	✗	✗	Multicast, Poisson cluster, flooding-based routing
[25]	Passive	Latency, energy consumption	PLNC	✗	✗	Chained secure mode
[26]	Passive	Latency	PLNC	✗	✗	Highly reliable common relay node, XOR
[27]	Active	Throughput	PLNC	✗	✗	XOR, opportunistic listening, COPE
[28]	Active	Throughput	PLNC	✗	✗	Multipath routing, XOR
Proposed	Active	Throughput, latency, and fairness	Multi-rate PLNC	✓	✓	Delay function, bidirectional flow, fully distributed

ABC = artificial bee colony optimization, ACO = ant colony optimization, MWST = minimum Wiener spanning tree, PDR = packet delivery ratio, RNC = random linear network coding, PLNC = physical-layer network coding.

et al. [16] designed a multipath data transmission scheme with network coding. The scheme maximizes the recoding opportunities and reduces routing reconstruction through ant colony optimization. Malathy et al. [17] proposed a back-pressure based routing scheme for large-scale IoT networks. The scheme balances the traffic load and optimizes fair power consumption for all nodes through network coding and clustering techniques. The scheme employs interflow, flow-oriented XOR-based network coding routing called flow oriented-network coding. Ni et al. [18] suggested a network coding-based resilient routing algorithm to achieve data security and availability jointly in software-defined networks. As a heuristic algorithm, it makes routing decisions based on the number of disjoint paths. Zhou et al. [19] proposed an opportunistic routing scheme that selects a forwarder set with a more optimal number of transmissions, for efficient energy use. The scheme exploits an intra-session network coding mechanism based on mining link correlations. Jiang et al. [20] aimed to determine a minimal number of required coding nodes for the minimum energy consumption in a multicast network. They proposed a maximum flow algorithm based on the depth-first search method to optimize the search time.

The discussed studies [13]–[20] are source routing protocols, where the source node locally determines an end-to-end path with more coding opportunities. However, all nodes have to retain the network state information; thus, a scalability problem occurs in source routing. Therefore, some studies have used hop-by-hop network coding-aware routing protocols [21]–[26], where relay nodes make a routing decision with limited network state information. Yan et al. [21] proposed an opportunistic network coding-aware routing approach protocol. The scheme determines the best packet forwarder with the awareness of potential coding opportunities, without node synchronization. Le et al. [22] put forth a distributed coding-aware routing protocol. The scheme employs a novel coding-aware routing metric, and detects coding opportunities on the entire path, eliminating the “two-hop” coding limitation in COPE. Song et al. [23], [24] suggested a flooding-based routing protocol using random network coding without routing path discovery and topology information. The protocol provides efficient paths with fewer hops and fewer generations transmitted in some hops. Raoof et al. [25] proposed a network coding based secure routing protocol. The protocol was designed for low-power and lossy networks with intra-flow network coding to mitigate attacks. Cheng et al. [26] proposed a routing scheme that exploits neighbor nodes with a high reliability and link rate as the common relay for network coding. The scheme is called the high reliability relay algorithm (HHRA). However, all of the discussed studies are based on passive network coding (i.e., they determine the routes passively through local information for network coding opportunities). Nodes passively wait for straightforward opportunities for network coding, but cannot exploit the potential, but not straightforward opportunities for network coding. Some active network coding-aware routing [27], [28] schemes have been investigated. Each node actively optimizes the routing along paths with the best coding options to reduce the number of transmissions to overcome these limitations. Jiao et al. [27] proposed an active network coding based high-throughput optimizing routing protocol. The nodes determine whether they can optimize the route to create the opportunity for network coding with already received or overheard packets. If the nodes determine an optimization possibility, they notify the related nodes to update the route. Han et al. [28] designed a coding-aware multi-path routing protocol. The protocol determines the multi-path routes considering the expected transmission count, and dynamically forwards packets over multiple paths based on the path reliability and coding opportunity (The protocol actively creates a network coding opportunity instead of passively waiting, by switching its path to maximize the switching gain).

A. Motivation, Contribution and Organization

Although routing approaches exploit the network coding opportunity, most use the first-come-first-served route information delivery method and apply only local information related to the session information preventing the network coding opportunity from being exploited efficiently. In addition, the approaches do not focus on route setup delay constraints, which are critical for IIoT services.

In this work, we propose a network coding-aware distributed and active routing that maximizes network coding opportunities while guaranteeing the end-to-end route setup delay time, which has never been studied. Table 1 lists the key features and differences compared with the existing studies. The main contributions of this study are summarized as follows.

- We propose a delay-constrained fully-distributed path establishment scheme called the network coding-aware delayed store and forwarding (NC-DSF) scheme. As a low-complexity scheme that can be easily used with the existing passive and active network coding-aware routing schemes, the NC-DSF scheme maximally exploits the network coding gain and guarantees the end-to-end route setup delay limit.
- Unlike the previous studies, considering all network traffic flows, the NC-DSF scheme establishes a maximized bidirectional route useful for network coding. As a core part of the algorithm that generates bidirectional routes, we propose a tight delay function derived using a strict end-to-end route setup delay bound, which delays the received route setup packets before forwarding them according to the link status and network topology. Subsequently, we put forward relaxed delay functions derived using substantially realistic and practical conditions. Finally, we design a load-weighted delay function considering the trade-off between bidirectionality and network load balancing. Using this function enables controlling load balancing while maximizing the network coding opportunity.
- We perform extensive simulations in mesh and random multi-hop networks, and confirm that the NC-DSF scheme increases network coding opportunities and provides enhanced throughput and latency performance.
- The NC-DSF scheme is useful for ultra-reliable and low latency communication (URLLC) services, such as real-time IIoT in massive non-infrastructure mobile sensors, ad hoc networks, and 6G networks.

This paper is organized as follows. Section II discusses the related work. Then, Section III elaborates on the network coding opportunity. Next, Sections IV and V present the proposed information forwarding algorithms. In Section VI, we present its operational procedures. The experiment results are provided in Section VII. Finally, the conclusions are presented in Section VIII.

III. Network Coding Opportunity

This section elaborates on some exemplary cases in which network coding can be accomplished.

A. Link Model

In Fig. 1, link $l(n_i, n_j)$ denotes a link that connects nodes i and j, and link $l(n_i, n_j, n_k)$ denotes a two-hop link that connects nodes i, j and k. Their directionality is denoted by $\vec{l}(n_i, n_j)$ or $\vec{l}(n_i, n_j, n_k)$, respectively. In addition, the path sequentially connecting nodes $n_i, n_j, n_k, \ldots, n_m$ is denoted as $p(n_i, n_j, n_k, \ldots, n_m)$.

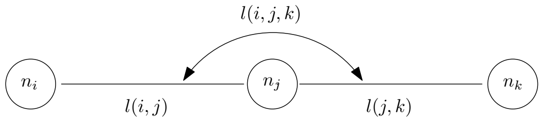


Fig. 1. Two-hop link model.

B. Network Coding Model

In some cases, network coding occurs inherently. First, in Fig. 2, we assume that nodes n_1 and n_3 send data to nodes n_3 and n_1, respectively. For example, $X_{1\to 3}$ denotes the data that node 1 sends to node 3, and $X_{1\to 3} \oplus X_{3\to 1}$ denotes the network-coded information for $X_{1\to 3}$ and $X_{3\to 1}$. If unidirectional data flow-based noncoded protocols are used, four time slots are necessary to exchange the data pair $X_{1\to 3}$ and $X_{3\to 1}$. However, if network coding is employed, it demands only two time slots for the information exchange (i.e., network coding doubles the network capacity [7]).

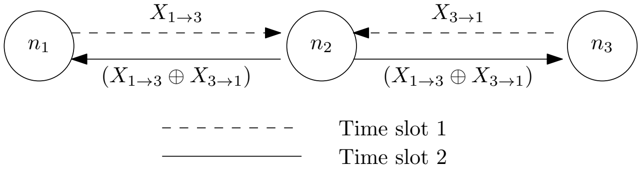
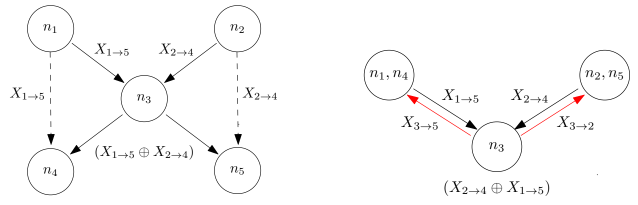

$X_{1\to 3}$
$X_{3\to 1}$
n_1
n_2
n_3
$(X_{1\to 3} \oplus X_{3\to 1})$
$(X_{1\to 3} \oplus X_{3\to 1})$
Time slot 1
Time slot 2

Fig. 2. Linear network.

The second case is an example of a cross-network, as depicted in Fig. 3. Nodes n_1 and n_2 send their data to nodes n_5 and n_4, respectively. When n_1 sends its data $X_{1\to 5}$ to n_3, it interferes with n_4 (i.e., data $X_{1\to 5}$ are also available at n_4). Accordingly, n_4 knows that the network coding information using data $X_{1\to 5}$ can also be extracted at n_4 because data $X_{1\to 5}$ are also available at n_4. In terms of network coding, n_1 and n_4 can be considered the same node regarding n_3. Therefore, n_3 can assume that a virtual data link $\vec{l}(n_4, n_3)$ exists between nodes n_3 and n_4 as a dual link $\vec{l}(n_1, n_3)$.

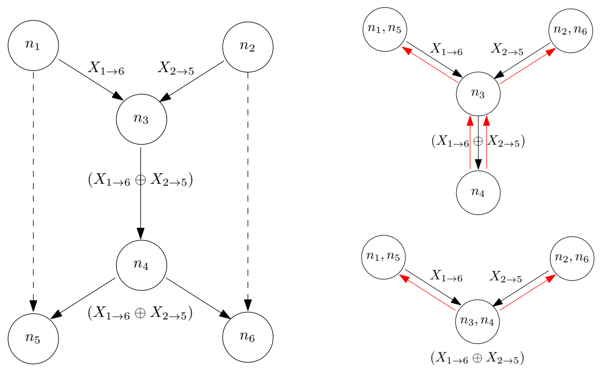


(a) Original network (b) Equivalent network

Fig. 3. Square network.

Likewise, n_2 sends its data $X_{2\to4}$ to n_3, and interferes with n_5 (i.e., data $X_{2\to4}$ are also available at n_5). Accordingly, n_5 knows that the network coding information using data $X_{2\to4}$ can be extracted at n_5 because data $X_{2\to4}$ are also available at n_5. In terms of network coding, n_2 and n_5 can be considered the same node regarding n_3. Therefore, n_3 can assume that a virtual data link $\vec{l}(n_3, n_2)$ exists between n_2 and n_3 as a dual link $\vec{l}(n_3, n_5)$. In this configuration, the conventional interference-avoidance-based transmission takes four time slots to complete the communication. However, upon employing network coding, only two time slots are required.

The other example is the butterfly network depicted in Fig. 4. We assume that nodes n_1 and n_2 send their data to nodes n_6 and n_5, respectively. From the configuration depicted in Fig. 4(a), a two-hop virtual link $\vec{l}(n_4, n_3, n_1)$ exists that connects $\vec{l}(n_4, n_3)$ and $\vec{l}(n_3, n_1)$ as a dual link of $\vec{l}(n_4, n_5)$. Additionally, a two-hop virtual link $\vec{l}(n_4, n_3, n_2)$ connects $\vec{l}(n_4, n_3)$ and $\vec{l}(n_3, n_2)$ as a dual link of $\vec{l}(n_4, n_6)$, as depicted in the upper illustration in Fig. 4(b). Accordingly, it can be reduced again as depicted in the bottom illustration in Fig. 4(b) because n_4 is a leaf node of n_3. In this configuration, the conventional interference-avoidance-based transmission takes six time slots to complete the communication. However, upon employing network coding, only three time slots are necessary.



(a) Original network (b) Equivalent network

Fig. 4. Butterfly network.

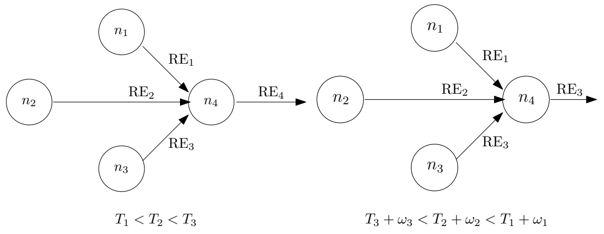
The basic elementary topology for network coding can be designed using bidirectional links from the scenarios depicted in Figs. 2–4. Therefore, we propose a bidirectional-link-based information-forwarding control scheme to enhance the network coding opportunity.

IV. Proposed Network Coding Aware Delayed Store-and-Forwarding Scheme

In this section, we explain the proposed information-forwarding scheme.

A. Proposed Information Forwarding Model

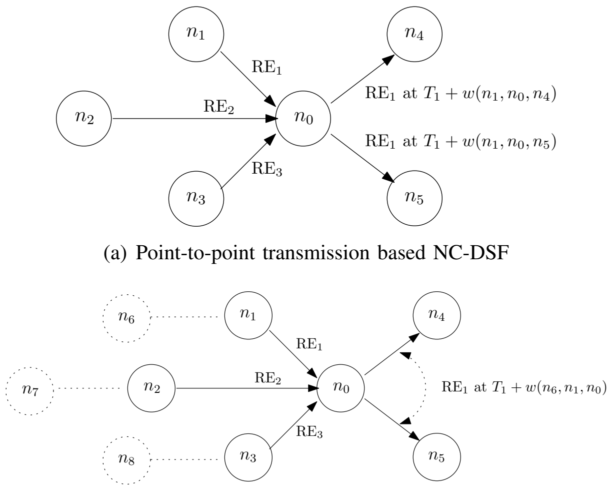
In mobile IoT networks, typical routing controls, including AODV [29] and dynamic source routing (DSR) [30], determine an end-to-end route using various route-establishment (RE) packet control schemes. In Fig. 5, packets RE_1, RE_2, and RE_3 are assumed to sequentially arrive from nodes n_1, n_2, and n_3 at time T_1, T_2, and T_3, respectively. Subsequently, the first in, first out (FIFO)-based RE control scheme in Fig. 5(a) sends the packet RE_1, which arrives earliest. The one-hop delayed forwarding scheme in Fig. 5(b) forwards packet RE_3, which has the fastest deadline after each delay ω_i for each packet RE_i, where $i = 1$, 2, and 3. However, the one-hop-based forwarding schemes in Fig. 5 do not consider the network coding opportunity.



(a) FIFO-based forwarding (b) One-hop delayed forwarding

Fig. 5. One-hop link based forwarding.

Therefore, we propose a new forwarding scheme, NC-DSF, which considers the two-hop link status to increase the network coding opportunity.


n_1
n_4
RE_1
RE_2
n_2
n_0
RE_1 at $T_1 + w(n_1, n_0, n_4)$
RE_1 at $T_1 + w(n_1, n_0, n_5)$
RE_3
n_3
n_5
n_6
n_7
n_8
RE_1 at $T_1 + w(n_6, n_1, n_0)$

(a) Point-to-point transmission based NC-DSF

(b) Flooding based NC-DSF

Fig. 6. Network coding-aware delayed store and forwarding (NC-DSF)

In Fig. 6(a), node n_0 selects the RE that arrives first. If RE_1 arrives first at T_1, node n_0 forwards the selected packet to n_4 and n_5 at $T_1 + w(n_1, n_0, n_4)$ and $T_1 + w(n_1, n_0, n_5)$, respectively, where $w(i, j, k)$ denotes the delay considering the two-hop link $l(i, j, k)$. The intermediate node n_0 determines the delay and transmits the received packet at various times for neighboring nodes. This method is referred to as point-to-point transmission-based NC-DSF. Furthermore, as shown in Fig. 6(b), node n_0 selects the RE that has the earliest deadline among $T_1 + w(n_6, n_1, n_0)$, $T_2 + w(n_7, n_2, n_0)$, and $T_3 + w(n_8, n_3, n_0)$. If $T_1 + w(n_6, n_1, n_0)$ has the earliest deadline, then the node n_0 floods RE$_1$ to n_4 and n_5 at $T_1 + w(n_6, n_1, n_0)$. This approach is referred to as flooding based NC-DSF. Between them, flooding based NC-DSF in Fig. 6(b) has lower communication overhead owing to the reduced forwarding, which is analyzed in Section VI.D. Therefore, we focus on the second approach in this work.

B. Proposed Delay Function

We propose a new delay function for the NC-DSF scheme to enhance the bidirectionality between the established routes. Before discussing the proposed network coding-aware forwarding control, we introduce new metrics as link and path bidirectionality.

Definition 1. *The link bidirectionality of a link, $l(n_i, n_j, n_k)$, denotes the number of bidirectional data flows using the link, which is measured by $\mathcal{B}(n_i, n_j, n_k)$.*

The link bidirectionality has a symmetrical value, that is $\mathcal{B}(n_i, n_j, n_k) = \mathcal{B}(n_k, n_j, n_i)$. Fig. 7 depicts examples in a linear network.

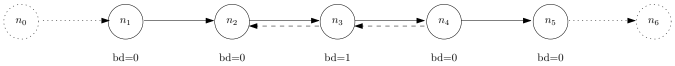


Fig. 7. Link bidirectionality in a linear network.

Definition 2. *The path-bidirectionality of a route from the source to the destination is defined as*

$$\mathcal{D} = \min_{l(n_i, n_j, n_k) \in \{\mathcal{P}(s,d)\}} \mathcal{B}(n_i, n_j, n_k), \tag{1}$$

where $\{\mathcal{P}(s, d)\}$ denotes the set of two-hop links that belong to an end-to-end path $\mathcal{P}(s, d)$ from source s to destination d.

This forwarding method uses the proposed delay function to set up a source-destination path with more bidirectional data flows. The delay function $f(\cdot)$ is determined by the value of the link bidirectionality $\mathcal{B}(n_i, n_j, n_k)$ over each link $l(n_i, n_j, n_k)$. When there is no confusion, $\mathcal{B}(n_i, n_j, n_k)$ and $\mathcal{B}$ are used interchangeably to simplify the notation. Nevertheless,the delay function should meet the following conditions (C):

(C-1) $f(\mathcal{B}) \geq f(\mathcal{B} + n)$, for $0 \leq \mathcal{B} \leq |\mathbb{U}| - 1$ and $n > 0$,

(C-2) $f(\mathcal{B}) \geq (|\mathbb{U}| - 1) \cdot f(\mathcal{B} + 1)$, for $0 \leq \mathcal{B} \leq |\mathbb{U}| - 1$,

(C-3) $(|\mathbb{U}| - 1) \cdot f(0) \leq \mathcal{E}_{limit}$,

where $|\mathbb{U}|$ denotes the total number of user nodes in the system, and $\mathcal{E}_{limit}$ denotes the maximum end-to-end route setup delay. Condition-1 indicates that the delay time in forwarding should be shorter when the bidirectionality is greater. Condition-2 states that the delay time in one link with higher bidirectionality should be less than the longest total delay by links with lower bidirectionality in the network. Finally,Condition-3 indicates that the highest end-to-end forwarding delay (i.e., $(|\mathbb{U}|-1) \cdot f(0)$) should be less than $\mathcal{E}_{limit}$. Thus, we propose a delay function satisfying these conditions, provided in Property 1.

Property 1. *The delay function is given by*

$$f(\mathcal{B}) = \frac{1}{(|\mathbb{U}| - 1)^{\mathcal{B}+1}} \cdot \mathcal{E}_{limit}. \tag{2}$$

Proof. The delay function $f(\cdot)$ can meet the first and second conditions tightly,

$$f(\mathcal{B}) = (|\mathbb{U}| - 1|) \cdot f(\mathcal{B} + 1), \ 0 \leq \mathcal{B} \leq |\mathbb{U}| - 1. \tag{3}$$

That is, it entails

$$f(0) = (|\mathbb{U}|-1) \cdot f(1) = \cdots = (|\mathbb{U}|-1)^{|\mathbb{U}|-1} \cdot f(|\mathbb{U}|-1). \tag{4}$$

However, $(|\mathbb{U}| - 1) \cdot f(0) \leq \mathcal{E}_{limit}$ according to Condition 3. Assuming that $\mathcal{E}_{limit}$ is completely applied, we find

$$f(0) = \frac{1}{(|\mathbb{U}| - 1)} \cdot \mathcal{E}_{limit} \tag{5}$$

Accordingly, we determine the delay function through (4) and (5) as follows:

$$f(\mathcal{B}) = \frac{1}{(|\mathbb{B}| - 1|)^{\mathcal{B}+1}} \cdot \mathcal{E}_{limit}. \tag{6}$$

□

C. Pseudo Algorithm

Algorithm 1 presents the pseudo code of the proposed forwarding control.

Algorithm 1 Proposed flooding-based NC-DSF

```
For any relaying node $n_k$:
while (1) do
    $\mathcal{T} \leftarrow$ current network time
    if RE$_{i,j,k}$ arrives from neighbor $n_i$ through $n_j$ then
        $\mathcal{T}_{i,j,k} \leftarrow \mathcal{T}$
        Find the link bidirectionality $\mathcal{B}(n_i, n_j, n_k)$
        Find the delay $f(\mathcal{B}(n_i, n_j, n_k))$
        Insert RE$_{i,j,k}$ into the set of collected REs, $\Psi$
    end if
    Check all RE$_k$s,
        $(i', j', k) = \arg\min_{(i,j,k) \in \Psi} \{\mathcal{T}_{i,j,k} + f(\mathcal{B}(n_i, n_j, n_k))\}$
    if $\mathcal{T}_{i',j',k} + f(\mathcal{B}(n'_i, n'_j, n_k)) \leq \mathcal{T}$ then
        Forward the earliest deadlined RE$_{i',j',k}$ packet
        Drop all remaining RE packets
        Break
    end if
end while
```

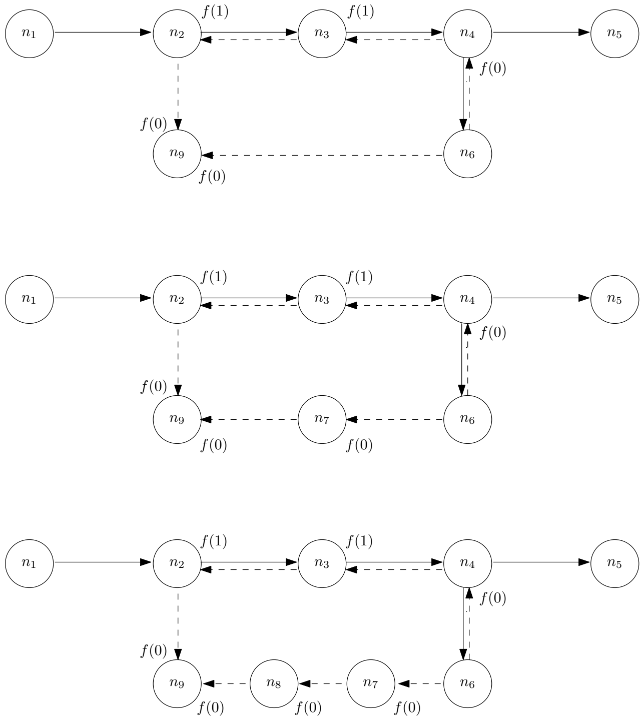


Fig. 8. Partially bidirectional path; $p(n_1, n_2, n_3, n_4, n_5)$ and $p(n_4, n_6)$ were established in advance (bold line), and n_6 attempts to send its data to n_9 (dotted line).

D. Delay for Partially Bidirectional Path

In Fig. 8, we assume that paths $p(n_1, n_2, n_3, n_4, n_5)$ and $p(n_4, n_6)$ were established, and that n_6 attempts to send its data to n_9. In Fig. 8(a), non-network coding path $p(n_6, n_9)$ and network coding path $p(n_6, n_4, n_3, n_2, n_9)$ entail the delays of $f(0)$ and $2f(0) + 2f(1)$, respectively. In this case, the non-network coding RE packet arrives earlier than the network coding packet at node n_9. In Fig. 8(b), the non-network coding path $p(n_6, n_7, n_9)$ and network coding path $p(n_6, n_4, n_3, n_2, n_9)$ entail the delays of $2f(0)$ and $2f(0) + 2f(1)$, respectively. In this case, the non-network coding RE packet arrives earlier than the network coding packet at node n_9. In Fig. 8(c), the non-network coding path $p(n_6, n_7, n_8, n_9)$ and network coding path $p(n_6, n_4, n_3, n_2, n_9)$ entail the delays of $3f(0)$ and $2f(0) + 2f(1)$, respectively. In this case, the network coding RE packet arrives earlier than the non-network coding packet at node n_9. When the route-bidirectionality is the same, the network coding RE packet could arrive early or late. If we add a delay ϵ for non-network coding RE at the last node, we can make the network coding RS RE have an earlier deadline. For example, in Fig. 8(b), if we include an additional delay $\epsilon \geq 2f(0)$ at node n_9, we can make the network coding RE have an earlier deadline than the non-network coding at node n_9 and can establish a network coding path. Additionally, ϵ can be determined considering the route-bidirectionality, link bidirectionality, and trade-off between the network coding opportunity and shortest delay.

V. Proposed Relaxed NC-DSF Control

Using the link bidirectionality $\mathcal{B}$ and delayed function $f(\mathcal{B})$, we can obtain enhanced, bidirectional route paths. However, as $\mathcal{B}$ increases, $f(B)$ becomes significantly small (i.e., a significantly fast and accurate RE packet delay control is required at the relay nodes). Due to hardware and software limitations, a shortest delay value that each node can handle exists. To address this problem, we introduce the highest bidirectionality $\bar{\mathcal{B}}$. If a link has a link bidirectionality $\mathcal{B}$ that exceeds $\bar{\mathcal{B}}$, the link has an $f(\bar{\mathcal{B}})$ forwarding delay. Second, we use γ $\left(\frac{1}{|\mathbb{U}|-1} \leq \cdot \leq 1\right)$, which relaxes the delay function as $f(B) = \gamma \cdot (|\mathbb{U}| - 1) \cdot f(B+1)$ from Condition-2.

A. Relaxed Control with $\bar{\mathcal{B}}$

We assume that Γ is the shortest forwarding delay that the system can support.

Property 2. *The relaxed delay function is given by*

$$f(\bar{\mathcal{B}}, \mathcal{B}) = \begin{cases} \frac{1}{(|\mathbb{U}|-1)^{\mathcal{B}+1}} \cdot \mathcal{E}_{limit} & , \mathcal{B} \leq \bar{\mathcal{B}}, \\ f(0) & , \mathcal{B} > \bar{\mathcal{B}}, \end{cases}$$

where

$$\bar{\mathcal{B}} = max \left\{ m \in \mathbb{Z}^* : \frac{1}{(|\mathbb{U}|-1)^{m+1}} \cdot \mathcal{E}_{limit} \geq \Gamma \right\}.$$

Proof. If m denotes the permissible bidirectionality, its delay is given by

$$f(m) = \frac{1}{(|\mathbb{U}| - 1|)^{m+1}} \cdot \mathcal{E}_{limit}. \tag{7}$$

Similar to the previous proof, the maximum permissible $\bar{\mathcal{B}}$ should be

$$\bar{\mathcal{B}} = max \left\{ m \in \mathbb{Z}^* : \frac{1}{(|\mathbb{U}|-1)^{m+1}} \cdot \mathcal{E}_{limit} \geq \Gamma \right\}. \tag{8}$$

□

B. Relaxed Control with γ

Property 3. *The relaxed delay function is given by*

$$f(\gamma, \mathcal{B}) = \frac{1}{\gamma^{\mathcal{B}}} \cdot \frac{1}{(|\mathbb{U}| - 1)^{\mathcal{B}+1}} \cdot \mathcal{E}_{limit} \tag{9}$$

Proof. In Conditions 1 and 2, the relaxed delay function $f(\cdot)$ should satisfy

$$f(\gamma, \mathcal{B}) = \gamma(|\mathbb{U}| - 1|) \cdot f(\gamma, \mathcal{B}+1), 0 \leq \mathcal{B} \leq |\mathbb{U}| - 1, \tag{10}$$

which implies that

$$\begin{aligned} f(\gamma, 0) &= \gamma(|\mathbb{U}| - 1) \cdot f(\gamma, 1) = \cdots \\ &= \gamma(|\mathbb{U}| - 1)^{|\mathbb{U}|-1} \cdot f(\gamma, |\mathbb{U}| - 1). \end{aligned} \tag{11}$$

Moreover, (9) can be determined using the following:

$$f(0) = \frac{1}{(|\mathbb{U}| - 1)} \cdot \mathcal{E}_{limit}. \tag{12}$$

□

C. Relaxed Control with $\bar{\mathcal{B}}$ and γ

Under the smallest implementable delay Γ, a jointly relaxed delay function $f(\gamma, \bar{\mathcal{B}}, \mathcal{B})$ can be designed by combining the two relaxation methods.

Property 4. *The jointly relaxed delay function is given by*

$$f(\gamma, \bar{\mathcal{B}}, \mathcal{B}) = \begin{cases} \frac{1}{\gamma^{\mathcal{B}}} \cdot \frac{1}{(|\mathbb{U}|-1)^{\mathcal{B}+1}} \cdot \mathcal{E}_{limit} & , \mathcal{B} \leq \bar{\mathcal{B}}, \\ f(\gamma, \bar{\mathcal{B}}) & , \mathcal{B} > \bar{\mathcal{B}}, \end{cases}$$

where

$$\bar{\mathcal{B}} \leq max \left\{ m \in \mathbb{Z}^* : \frac{1}{\gamma^m} \frac{1}{(|\mathbb{U}|-1)^{m+1}} \cdot \mathcal{E}_{limit} \geq \Gamma \right\}.$$

Proof. Similar to Property 1, from Conditions 1 and 2, we have

$$\begin{aligned} f(\gamma, \bar{\mathcal{B}}, 0) &= \gamma \cdot (|\mathbb{U}| - 1|) \cdot f(\gamma, \bar{\mathcal{B}}, 1) \\ &= \cdots \\ &= \gamma^{\mathcal{B}} \cdot (|\mathbb{U}| - 1|)^{\mathcal{B}} \cdot f(\gamma, \bar{\mathcal{B}}, \mathcal{B}) \\ &= \cdots \\ &= \gamma^{\bar{\mathcal{B}}} \cdot (|\mathbb{U}| - 1|)^{\bar{\mathcal{B}}} \cdot f(\gamma, \bar{\mathcal{B}}, \bar{\mathcal{B}}), \end{aligned} \tag{13}$$

where $\frac{1}{|\mathbb{U}|-1} \leq \gamma \leq 1$ because $1 \leq \gamma \cdot (|\mathbb{U}| - 1|) \leq |\mathbb{U}| - 1|$. Furthermore, from Condition 3, we have

$$f(\gamma, \bar{\mathcal{B}}, 0) = \frac{1}{(|\mathbb{U}| - 1)} \cdot \mathcal{E}_{limit}. \tag{14}$$

Accordingly, we determine the relaxed delay function $f(\gamma, \bar{\mathcal{B}}, \mathcal{B})$ using (13) and (14):

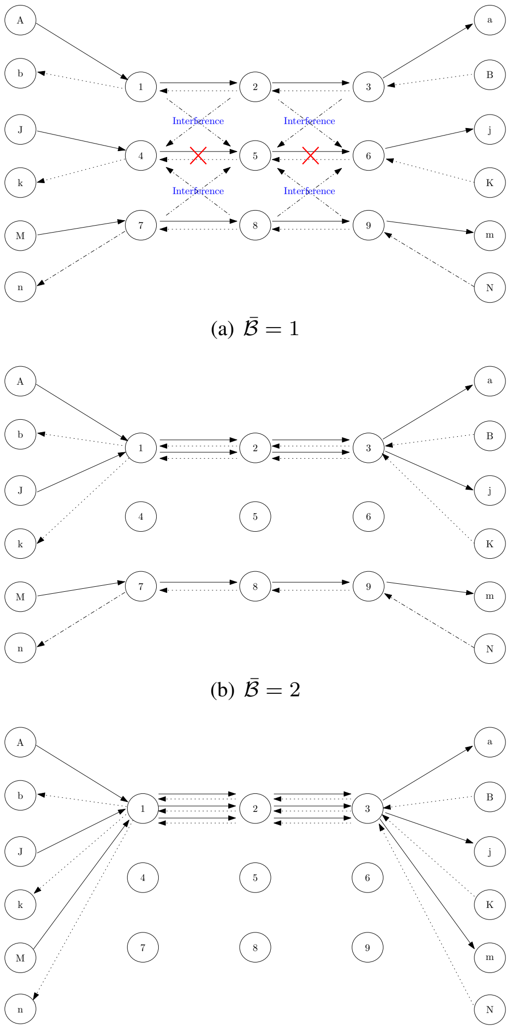
$$f(\gamma, \bar{\mathcal{B}}, \mathcal{B}) = \frac{1}{\gamma^{\mathcal{B}}} \cdot \frac{1}{(|\mathbb{U}| - 1)^{\mathcal{B}+1}} \cdot \mathcal{E}_{limit}, \text{ where } \mathcal{B} \leq \bar{\mathcal{B}}. \tag{15}$$

However, the relaxed delay should be greater than γ. That is, the highest $\bar{\mathcal{B}}$ should satisfy

$$\bar{\mathcal{B}} \leq max \left\{ m \in \mathbb{Z}^* : \frac{1}{\gamma^m} \frac{1}{(|\mathbb{U}|-1)^{m+1}} \cdot \mathcal{E}_{limit} \geq \Gamma \right\},$$

where $\mathbb{Z}^*$ denotes a set of nonnegative integers. □

There are two extreme scenarios: (i) When $\gamma = \frac{1}{|\mathbb{N}|-1}$ or $\bar{\mathcal{B}} = 0$, the proposed relaxed forwarding delay for all nodes becomes the FIFO-based control, irrespective of the link bidirectionality. (ii) When $\gamma = 1$ and $\bar{\mathcal{B}} = |\mathbb{U}| - 1$, the proposed relaxed forwarding delay becomes the strict NC-DSF control. Therefore, by controlling γ and $\bar{\mathcal{B}}$, a different relaxed forwarding delay control between FIFO-based control and the strict NC-DSF control can be designed.



(a) $\bar{\mathcal{B}} = 1$

(b) $\bar{\mathcal{B}} = 2$

(c) $\bar{\mathcal{B}} = 3$

Fig. 9. Effect of $\bar{\mathcal{B}}$.

D. Relaxed Control for Load Balancing

In Fig. 9(a), $\mathcal{B}$ is low, and it can yield low bidirectionality paths, thereby creating significant interference between links. In Fig. 9(b), if $\mathcal{B}$ increases, it starts to realize resource utilization and network coding effectively. In Fig. 9(c), if $\mathcal{B}$ is high, the traffic loads are concentrated on particular links. As $\mathcal{B}$ increases, the proposed control scheme can establish more routes with high bidirectionality. However, the scheme can induce bursty traffic on some links, deteriorating the network load balancing and resource utilization. Therefore, we designed a new bidirectionality index $\bar{B}(t)$ to address this, reflecting the resource residuals as follows:

$$\bar{B}(t) = \mathcal{B}(t)\left(1 - w\frac{\mathcal{B}(t)}{\bar{\mathcal{B}}}\right), \quad 0 \leq w \leq 1 \tag{16}$$

where w is a control weight for load balancing. The bidirectionality index $\bar{B}(t)$ is a function of time that varies according to the degree of residual resources at every instance t. Accordingly, a load-weighted relaxed delay function $f(w, \gamma, \bar{B}(t), \bar{\mathcal{B}}, \mathcal{B}(t))$ can be constructed as follows:

$$\begin{aligned} &f(w, \gamma, \bar{B}(t), \bar{\mathcal{B}}, \mathcal{B}(t)) \\ &= \begin{cases} \frac{1}{\gamma^{\bar{B}(t)}} \cdot \frac{1}{(|\mathbb{U}|-1)^{\bar{B}(t)+1}} \cdot \mathcal{E}_{limit} & , \mathcal{B}(t) \leq \bar{\mathcal{B}}, \\ f(0) & , \mathcal{B}(t) > \bar{\mathcal{B}}, \end{cases} \end{aligned}$$

The forwarding delay for the link that passes through the link with higher bidirectionality than $\bar{\mathcal{B}}$ is set to $f(0)$ to prevent the traffic from exceeding the link capacity. The proposed load-weighted relaxed delay function can provide more bidirectional sessions with balanced network-resource utilization.

VI. Operational Example

This section examines how the proposed delay functions work, and how routes are established. Moreover, it discusses how information is transmitted using three representative network models.

A. Linear Network

In Fig. 10, we assume that a path $p(n_2, n_3, n_4)$ is already established, and that n_5 attempts to send its data to n_1. Node n_5 first sends its RE message to n_4 Then, n_4 broadcasts the received RE message to its neighbor nodes n_3 and n_6. Next, n_3 and n_6 receive the packet simultaneously, but n_3 relays the received RE message to n_2 before n_6 because n_3 has higher bidirectionality than n_2(i.e., faster RE forwarding with lower delay). Eventually, at destination node n_1, the RE that passes through nodes n_4, n_3, and n_2 arrives first; thus, the path $p(n_5, n_4, n_3, n_2, n_1)$ is established and network coding can occur over $l(n_3, (n_2, n_4))$.

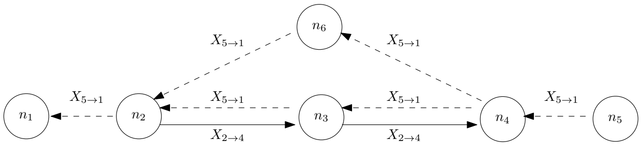


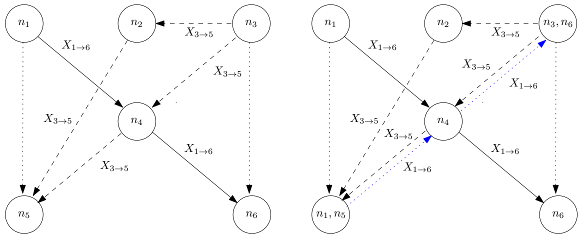
Fig. 10. Transmission in linear network.

Algorithm 2 Information exchange with the neighbor node

```
while (1) do
    Identify neighbor nodes
    Exchange neighbor node information with neighbors
    Establish virtual links
end while
```

B. Square Network

Concerning Fig. 11, we assume that a path $p(n_1, n_4, n_6)$ is already established and that n_3 attempts to send its data to n_5. Any node in the network performs the following operations. For example, n_5 identifies its neighbor nodes n_1 and n_4, and then shares the neighbor node information with n_1 and n_4. Subsequently, n_4 knows that the network coded information using data $X_{1\to6}$ can be extracted at n_5 because n_4 knows the availability of data $X_{1\to6}$ at n_5. In terms of network coding, n_1 and n_5 can be considered the same node regarding n_4. Therefore, n_4 can assume that a virtual data link $\vec{l}(n_5, n_4)$ exists between n_5 and n_4 as a dual link of $\vec{l}(n_1, n_4)$. Likewise, n_6 identifies its neighbor nodes n_3 and n_4, and shares the neighbor node information with its neighbors. Subsequently, n_4 knows that the network coded information using data $X_{3\to5}$ can be extracted at n_6 because n_4 knows the availability of data $X_{3\to5}$ at n_6. In terms of network coding, n_3 and n_6 can be considered the same node for n_4. Therefore, n_4 can assume that a virtual data link $\vec{l}(n_4, n_3)$ exists between n_4 and n_3 as a dual link of $\vec{l}(n_4, n_6)$. Therefore, if node n_3 sends an RE message to its neighbor nodes n_2, n_4 and n_6, then node n_4 forwards the received RE message before n_2 and n_6 because n_4 has higher bidirectionality than n_2 and n_6. Therefore, at node n_5, the RE message that passes through n_3 and n_4 arrives first, following which path $p(n_3, n_4, n_5)$ is established.

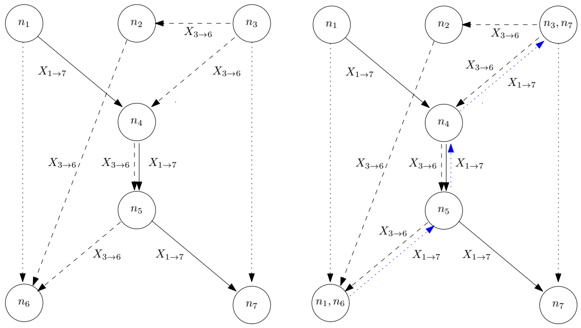


(a) Original network (b) Equivalent network

Fig. 11. Transmission in square network.

C. Butterfly Network

Concerning Fig. 12 and Table II, we assume that path $p(n_1, n_4, n_5, n_7)$ is already established, and that n_3 attempts to send its data to n_6.



(a) Original network (b) Equivalent network

Fig. 12. Transmission in butterfly network.

TABLE II
Information at Node n_5

1st-hop neighbors	2nd-hop neighbors
n_6	n_1, n_5
n_7	n_3, n_5
n_4	n_1, n_3, n_5

Using Algorithm 2, nodes identify their neighboring node information and share it with their neighbors. That is, n_5 knows that a virtual data link $\vec{l}(n_5, n_6)$ exists as a dual link of $\vec{l}(n_1, n_4, n_5)$, and that a virtual data link $\vec{l}(n_5, n_4, n_3)$ exists as a dual link of $\vec{l}(n_5, n_7)$. Therefore, when node n_3 sends an RE message to its neighbor nodes n_2, n_4 and n_7, n_4 forwards the received RE message before n_2 and n_7 because n_4 has higher bidirectionality (i.e., a lower delay using virtual links than n_2). Thus, at node n_6, the RE message that passes through nodes n_3, n_4, and n_5 arrives first, and the path $p(n_3, n_4, n_5, n_6)$ is established.

D. Complexity

1) Communication Overhead: For the analysis, the neighbor node information message, route request message, and route reply message are denoted as $\mathcal{I}$, $\mathcal{Q}$, and $\mathcal{L}$, respectively. Correspondingly, we assume that the information bits representing the neighbor node information message, route request message, and route reply message are $\log_2(\mathcal{I})$, $\log_2(\mathcal{Q})$, and $\log_2(\mathcal{L})$. For both flooding and point-to-point NC-DSF, the source node sends a route request message to neighbor nodes. If the number of neighbor nodes for the source node is $\mathcal{H}_s$, there are $\mathcal{H}_s \log_2(\mathcal{Q})$ information bits. The overheads values of the relay nodes for the two methods are different. In the point-to-point NC-DSF, the relay node forwards the message to all neighbor nodes after receiving the route request message. If the relay nodes receive a route reply message, the message is forwarded to the node that sent the previous route request message. Therefore, if the number of neighbor nodes for the relay node is $\mathcal{H}_r$, the total information bits at the relay node are $\mathcal{H}_r \log_2(\mathcal{Q}) + \log_2(\mathcal{L})$. Moreover, in the flooding NC-DSF, the relay node only forwards one route request message received from the others. Therefore, the total information bits at the relay node become $\log_2(\mathcal{Q}) + \log_2(\mathcal{L})$. Last, the destination node replies to the route request message. Therefore, the total information bits for the destination node is $\log_2(\mathcal{L})$. The communication overheads are the same for flooding and point-to-point NC-DSF. According to Algorithm 2, all nodes have to exchange their neighbor node information, which corresponds to $\mathcal{H} \log_2(\mathcal{I})$ with $\mathcal{H}$ denoting the number of neighbor nodes. The communication overheads are summarized in Table III.

TABLE III
COMMUNICATING OVERHEAD

Node	Point-to-point NC-DSF	Flooding NC-DSF
Source node	$\mathcal{H}_s \log_2(\mathcal{IQ})$	$\mathcal{H}_s \log_2(\mathcal{IQ})$
Relay node	$\mathcal{H}_r \log_2(\mathcal{IQ}) + \log_2(\mathcal{L})$	$\mathcal{H}_r \log_2(\mathcal{I}) + \log_2(\mathcal{QL})$
Destination node	$\mathcal{H}_d \log_2(\mathcal{I}) + \log_2(\mathcal{L})$	$\mathcal{H}_d \log_2(\mathcal{I}) + \log_2(\mathcal{L})$

2) Computational Complexity: We analyze the main computational complexity for the point-to-point and flooding NC-DSF. Most importantly, all nodes should compute their virtual links based on the exchanged neighbor node information. To this end, each node checks whether its neighboring links have other virtual links among the neighboring links. This method uses $\mathcal{H}^3$ comparison operations. In addition, the relay nodes calculate the route request packet forwarding delay based on bidirectionality. However, it requires constant computation $\mathcal{C}$ because the proposed function determines the delay. Considering the $\mathcal{H}$ maximum number of route request packets, it requires $\mathcal{C} \times \mathcal{H}$ computations. Therefore, overall, the computational complexities are summarized in Table IV.

TABLE IV
COMPUTATIONAL COMPLEXITY

Node	Point-to-point NC-DSF, Flooding NC-DSF
Source node	${\mathcal{H}_s}^3$
Relay node	${\mathcal{H}_r}^3 + \mathcal{C}\mathcal{H}_r$
Destination node	${\mathcal{H}_d}^3$

VII. SIMULATIONS AND EVALUATIONS

We evaluate the performance of the proposed scheme in mesh and random topologies. The source-destination route paths are established with the route setup delay constraint $\mathcal{E}_{limit} = 10$ (ms) for URLLC control latency under a 5G network. The source nodes generate packets at a constant rate of 3.7 Mbps. The simulation parameters are listed in Table V.

TABLE V
SYSTEM-LEVEL SIMULATION PARAMETERS

Parameter	Assumption
Node deployment	Mesh and random
Network size	100 m × 100 m
Node density	0.0040(sparse), 0.0068(dense) (node/m^2)
Minimum inter-node distance	5 m
Node transmit power	20 dBmW
Operating frequency	5 GHz
Channel bandwidth	80 MHz
Path-loss	$140.7 + 36.7 \log_{10}(d(\text{km}))$ (dB)
Node mobility	Static
Traffic model	Uniform distribution

In the evaluations, we compare the following schemes:

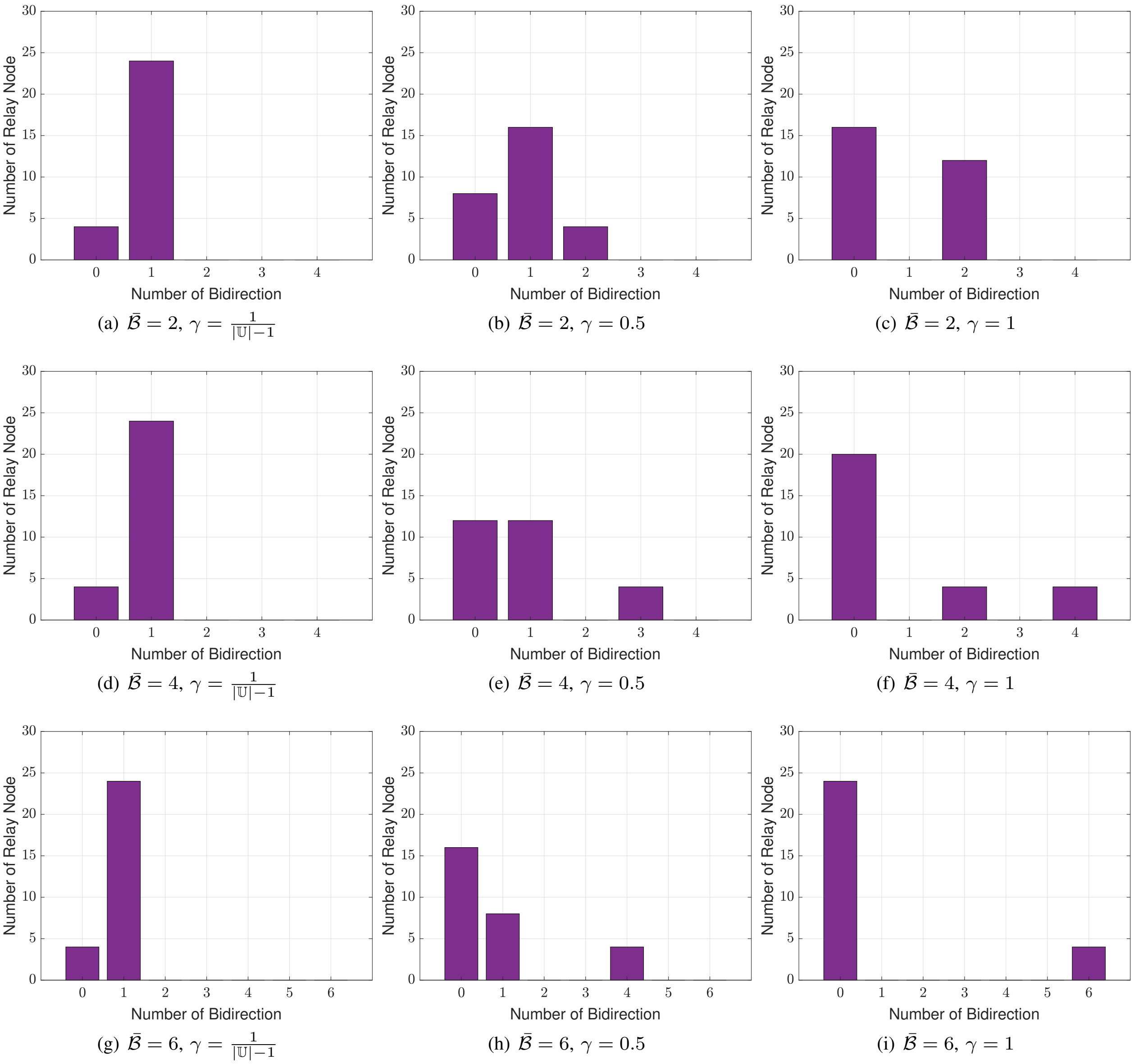
- DSR : In this well-known source routing scheme [30], the source node determines the end-to-end transmission path where the route reply arrives first from the destination.
- COPE: This is a well-known XOR network coding based transmission scheme [13]. Although network coding is not considered for routing, when a packet arrives, it is encoded and transmitted if network coding is possible.
- HRRA: This routing scheme exploits neighbor nodes with a high reliability and link rate as a typical relay for network coding [26].
- Proposed scheme: We evaluated the proposed scheme using various values of $\bar{\mathcal{B}}$ and γ.

We measure the following performance metrics.

- Throughput: This refers to the average amount of transmitted data per second for all nodes.
- Latency: This refers to the average time a packet takes to be transmitted from the source to the destination.
- Fairness: This refers to the average resource usage of the network links. We measure the metric using Jain's fairness [31], defined as

$$J(q_1, q_2, \cdots, q_n) = \frac{(\sum_{i=1}^{n} q_i)^2}{n \sum_{i=1}^{n} {q_i}^2}, \tag{17}$$

where n is the number of relay nodes and q_i is the accumulated data rate for relay node i. If the fairness index approaches 1, the network resources are evenly used.



(a) $\bar{\mathcal{B}} = 2$, $\gamma = \frac{1}{|\mathbb{U}|-1}$ (b) $\bar{\mathcal{B}} = 2$, $\gamma = 0.5$ (c) $\bar{\mathcal{B}} = 2$, $\gamma = 1$

(d) $\bar{\mathcal{B}} = 4$, $\gamma = \frac{1}{|\mathbb{U}|-1}$ (e) $\bar{\mathcal{B}} = 4$, $\gamma = 0.5$ (f) $\bar{\mathcal{B}} = 4$, $\gamma = 1$

(g) $\bar{\mathcal{B}} = 6$, $\gamma = \frac{1}{|\mathbb{U}|-1}$ (h) $\bar{\mathcal{B}} = 6$, $\gamma = 0.5$ (i) $\bar{\mathcal{B}} = 6$, $\gamma = 1$

Fig. 13. Bidirectionality histogram in the sparse mesh topology.

A. Mesh Topology

1) Sparse Source-Destination Pairs: The network consists of 12 uniformly deployed source-destination pairs and 28 relaying nodes for this simulation.

First, we evaluate how well the proposed algorithm establishes bidirectional transmission paths as planned. Fig. 13 presents a histogram of the bidirectionality of the relay nodes according to $\bar{\mathcal{B}}$ and α. In Fig. 13(a)-Fig. 13(c), Fig. 13(d)-Fig. 13(f), and Fig. 13(g)-Fig. 13(i), we set $\bar{\mathcal{B}} = 2$, 4, and 6, respectively. As γ increases, the bidirectionality of the links approaches to 2, 4, and 6, respectively.

Second, we evaluate the average throughput of all nodes, as shown in Fig. 14. At $\gamma = 0.5$, the proposed control provides throughput of 2651.25, 3411.08, and 3411.08 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme provides an enhanced throughput of 27.18%, 63.63%, and 63.63% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme provides an enhanced throughput of 2.59%, 32% and 32% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme provides an enhanced throughput of 8.73%, 39.9% and 39.9% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. When γ increases to 1, the proposed control provides throughput of 2732.1, 3610.48, and 3610.48 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme provides an enhanced throughput of 31.06%, 73.20%, and 73.20% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme provides an enhanced throughput of 5.72%, 39.71% and 39.71% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme results in an enhanced throughput of 12.05%, 48.07% and 48.07% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Higher values of $\bar{\mathcal{B}}$ and γ provide more enhanced throughput gains. The number of source-destination pairs is 12; thus, the bidirectionality $\mathcal{B}$ of relay nodes cannot be over 6. Therefore, the throughput does not change when $\bar{\mathcal{B}}$ is 6 or greater.

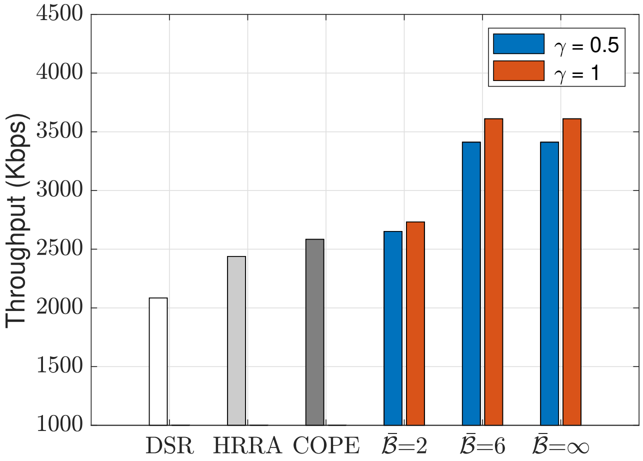


Fig. 14. Throughput in the sparse mesh topology.

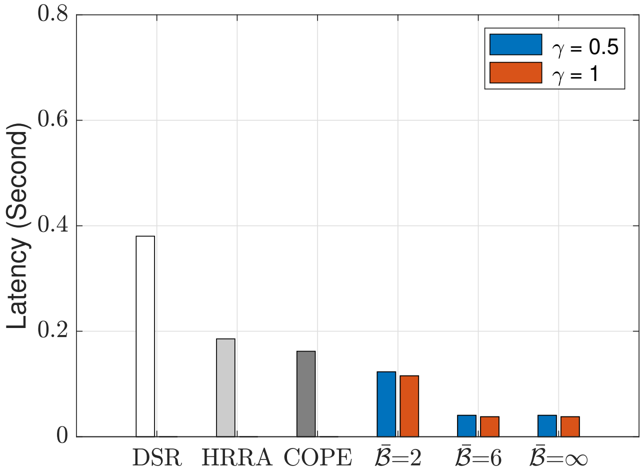


Fig. 15. Latency in the sparse mesh topology.

Third, we asses the latency, as shown in Fig. 15. At $\gamma = 0.5$, the proposed control provides latency values of 0.123, 0.0405 and 0.0405 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme has reduced latency at 67.63%, 89.32%, and 89.32% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme has reduced latency at 24.08%, 74.96%, and 74.96% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Similarly, compared with HRRA, the proposed scheme reduces latency at 33.7%, 78.13%, and 78.13% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. At $\gamma = 1$, the proposed control provides latency values of 0.1152, 0.0378, and 0.0378 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme has reduced latency at 69.69%, 90.04%, and 90.04% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Additionally, compared with COPE, the proposed scheme reduces latency at 28.89%, 76.64%, and 76.64% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme also reduces latency at 37.9%, 79.6%, and 79.6% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively.

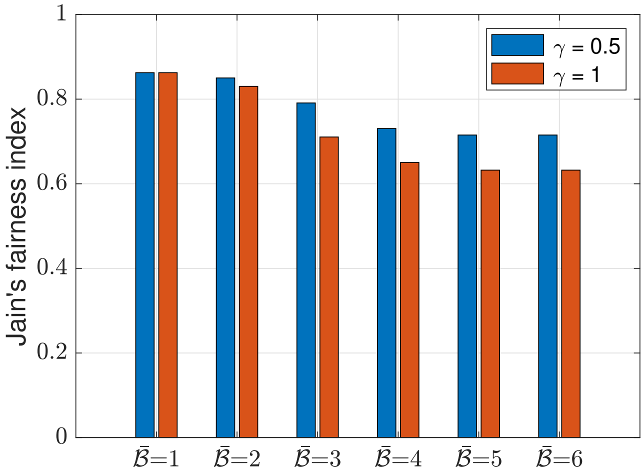


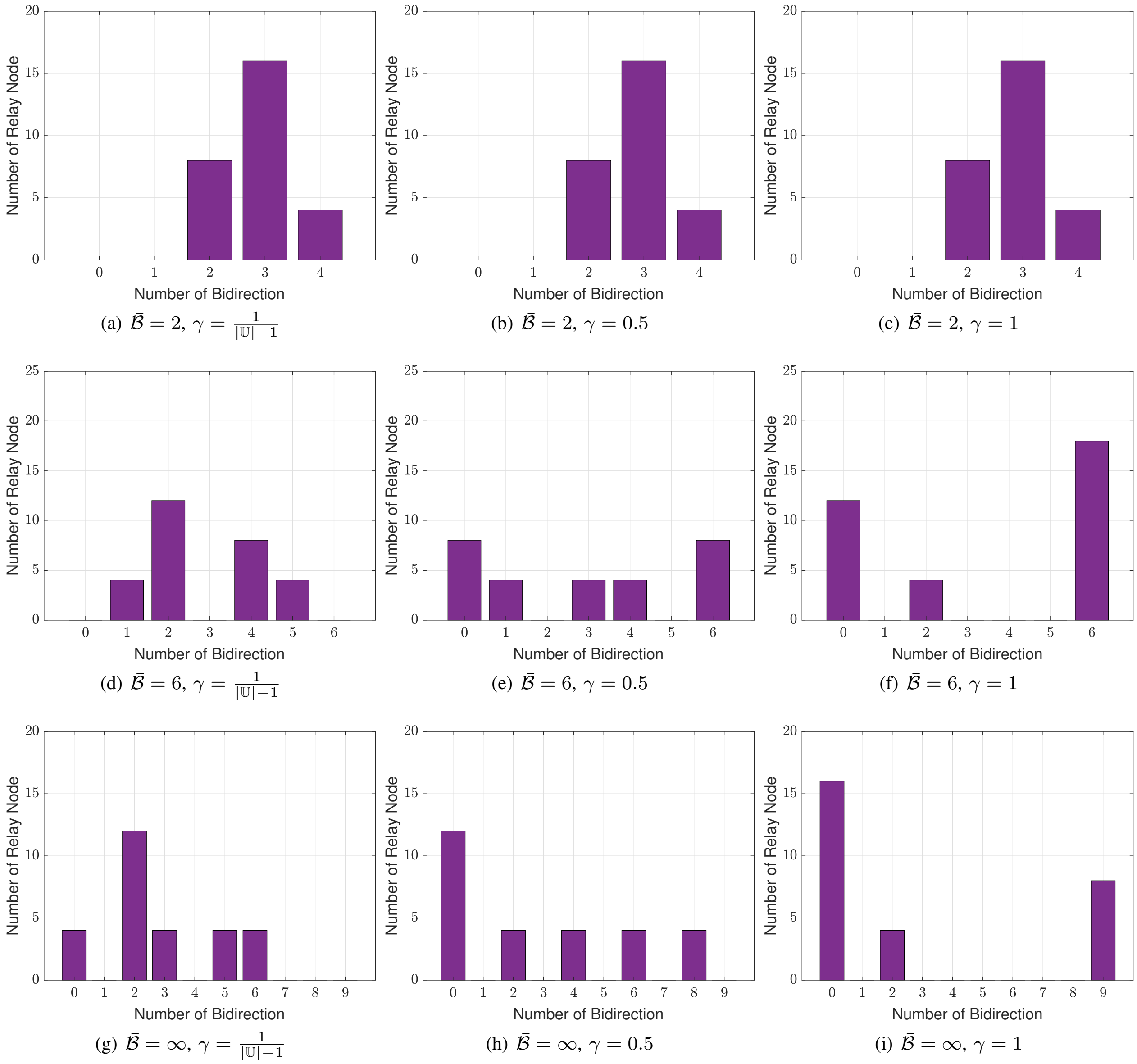
Fig. 16. Jain's fairness index in the sparse mesh topology.

Finally, we evaluate how the proposed scheme controls network resource usage, as shown in Fig. 16. We observe that Jain's fairness becomes slightly lower as $\bar{\mathcal{B}}$ and γ increase because the route paths are concentrated on a small number of relay nodes, forming more bidirectional paths. Throughput and latency are affected more by the network coding gain of the bidirectional paths than by the fair use of network resources.

2) Dense Source-Destination Pairs: The network consists of 40 uniformly deployed source-destination pairs and 28 relaying nodes for this evaluation.

First, Fig. 17 presents the histogram of the bidirectionality according to $\bar{\mathcal{B}}$ and γ. In Fig. 17(a), 17(b), and 17(c), the network has the same bidirectionality regardless of the value of γ because the number of source-destination pairs is sufficiently high and $\bar{\mathcal{B}} = 2$ is sufficiently low. In other words, regardless of γ, most paths are established following the shortest-path rule after two bidirectional paths are formed over the relay nodes. In contrast, in Fig. 17(d)-Fig. 17(f), and Fig. 17(g)-Fig. 17(i), more bidirectional paths are established as the value of $\bar{\mathcal{B}}$ or γ increases.

Second, we assess the average throughput of all nodes, as shown in Fig. 18. At $\gamma = 0.5$, the proposed control provides throughput at 2918.52, 3218.25, and 2997.18 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme provides an enhanced throughput of 47.4%, 62.53%, and 51.37% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme provides enhances the throughput at 16.27%, 28.21% and 19.4% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme has enhanced throughput of 32.16%, 45.74% and 35.72% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. At $\gamma = 1$, the proposed control provides throughput at 2918.52, 3422.17, and 2918.15 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme enhances the throughput at 47.4%, 72.83%, and 47.38% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme provides an enhanced throughput of 16.27%, 36.34% and 16.26% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme has an enhanced throughput of 32.16%, 54.97% and



(a) $\bar{\mathcal{B}} = 2$, $\gamma = \frac{1}{|\mathbb{U}|-1}$ (b) $\bar{\mathcal{B}} = 2$, $\gamma = 0.5$ (c) $\bar{\mathcal{B}} = 2$, $\gamma = 1$

(d) $\bar{\mathcal{B}} = 6$, $\gamma = \frac{1}{|\mathbb{U}|-1}$ (e) $\bar{\mathcal{B}} = 6$, $\gamma = 0.5$ (f) $\bar{\mathcal{B}} = 6$, $\gamma = 1$

(g) $\bar{\mathcal{B}} = \infty$, $\gamma = \frac{1}{|\mathbb{U}|-1}$ (h) $\bar{\mathcal{B}} = \infty$, $\gamma = 0.5$ (i) $\bar{\mathcal{B}} = \infty$, $\gamma = 1$

Fig. 17. Bidirectionality histogram in the dense mesh topology.

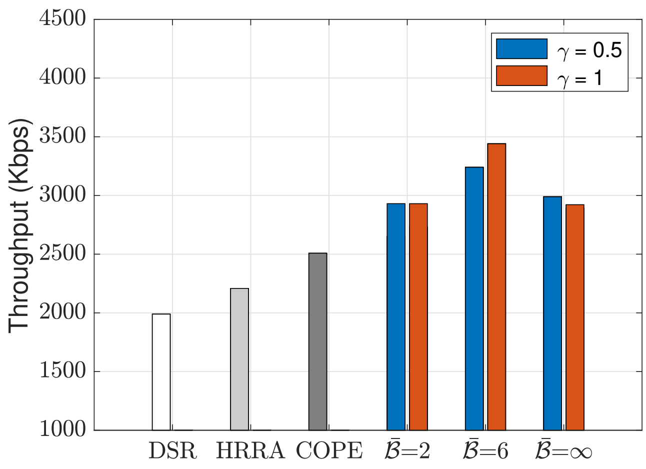


Fig. 18. Throughput in the dense mesh topology.

32.15% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. As $\bar{\mathcal{B}}$ increases, the average throughput also increases; however, when $\bar{\mathcal{B}}$ exceeds a certain level, only specific link resources are intensively used. Thus, the average throughput begins to decrease.

Third, we measure the latency, as shown in Fig. 19. At $\gamma = 0.5$, the proposed control provides a latency of 0.1158, 0.0508, and 0.0821 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme has reduced latency at 77.23%, 90.01%, and 83.85% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme reduces latency at 41.74%, 74.42%, and 58.67% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with HRRA, the proposed scheme has reduced latency at 69.83%, 86.75%, and 78.6% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. At $\gamma = 1$, the proposed control provides a latency of 0.1158, 0.0402, and 0.1085 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme reduces latency at 77.23%,

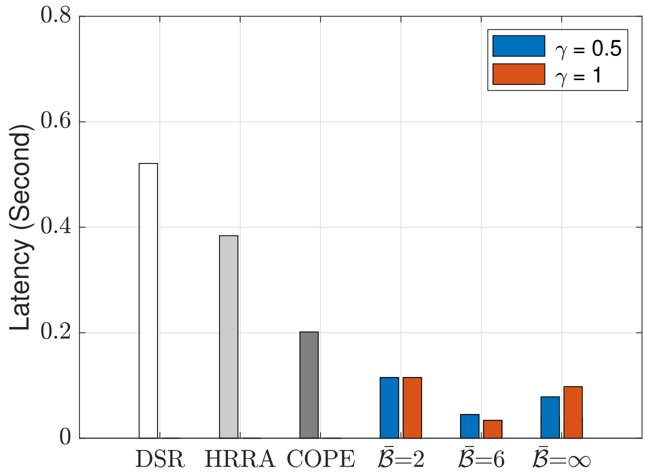


Fig. 19. Latency in the dense mesh topology.

92.08%, and 78.67% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Further, compared with COPE, the proposed scheme has reduced latency at 41.74%, 79.73%, and 45.41% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Similarly, compared with HRRA, the proposed scheme reduces latency at 69.83%, 89.5%, and 71.73% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Similar to the throughput, as $\bar{\mathcal{B}}$ increases, the average latency decreases; however when $\bar{\mathcal{B}}$ exceeds a certain level, the average latency begins to increase.

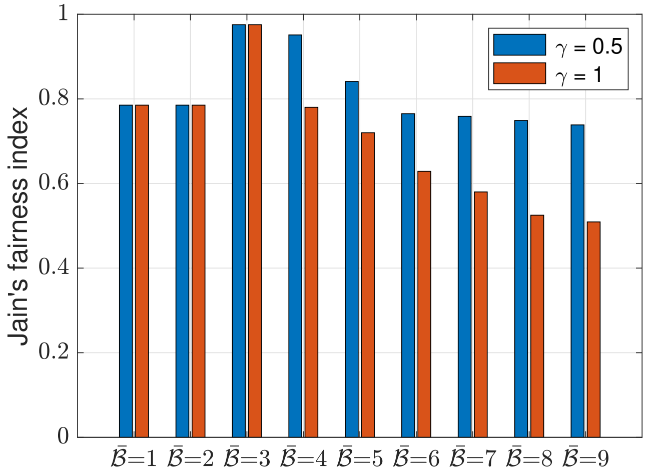


Fig. 20. Jain's fairness index in the dense mesh topology.

Finally, we assess the fairness concerning the network resource usage, as shown in Fig. 20. Jain's fairness is closest to 1 at $\bar{\mathcal{B}} = 3$. In the lower $\bar{\mathcal{B}}$, fairness depends on the source-destination distribution and network topology because more flows are established by the shortest paths rather than the bidirectionality. However, as $\bar{\mathcal{B}}$ increases, most paths are established by the proposed control such that the fairness decreases as expected, concentrated on specific relay nodes. As illustrated in Figs. 18, 19, and 20, the highest Jain's value does not indicate the highest throughput or lowest latency. Up to a specific $\bar{\mathcal{B}}$, the throughput and latency are affected more by the network coding gain of the bidirectional paths than by the fair use of the network resources. However, when $\bar{\mathcal{B}}$ exceeds a certain level, the network throughput and latency are degraded.

B. Random topology

1) Sparse Source-Destination Pairs: The network consists of 12 randomly deployed source-destination pairs and 28 relay nodes for this evaluation.

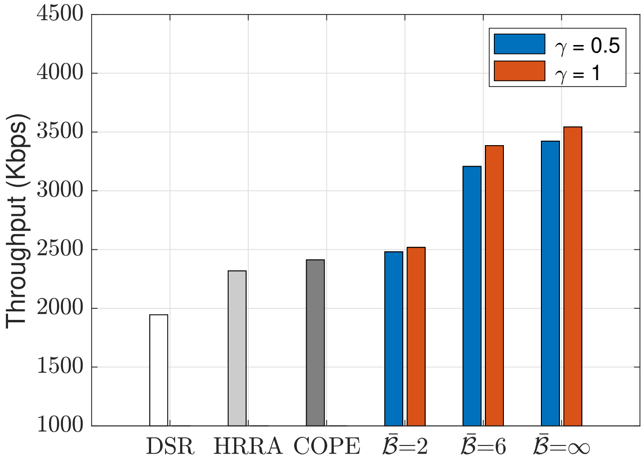


Fig. 21. Throughput in the sparse random topology.

First, we evaluate the average throughput of all nodes as shown in Fig. 21. At $\gamma = 0.5$, the proposed control outputs a throughput of 2480.74, 3207.52, and 3422.86 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme provides an enhanced throughput of 27.52%, 64.88%, and 75.95% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Further, compared with COPE, the proposed scheme enhances the throughput at 2.84%, 32.97% and 41.9% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme has an enhanced throughput of 6.97%, 38.32% and 47.6% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. When γ is increased to 1, the proposed control provides the throughput of 2518.66, 3384.18, and 3543.2 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme has an enhanced throughput of 29.47%, 73.96%, and 82.14% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Moreover, compared with COPE, the proposed scheme enhances the throughput at 4.41%, 40.3% and 46.89% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme outputs an enhanced throughput of 8.61%, 45.93% and 52.79% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Higher values of $\bar{\mathcal{B}}$ and γ provide more enhanced throughput gains.

Second, we assess the latency, as presented in Fig. 22. At $\gamma = 0.5$, the proposed control provides a latency of 0.1866, 0.0825 and 0.0482 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme reduces the latency at 70.76%, 87.07%, and 92.45% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Additionally, compared with COPE, the proposed scheme has a reduced latency at 42.46%, 74.55%, and 85.14% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Similarly, compared with HRRA, the proposed scheme reduces the latency at 46.75%, 76.45%, and 86.24% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. At $\gamma = 1$, the proposed control demonstrates a latency of

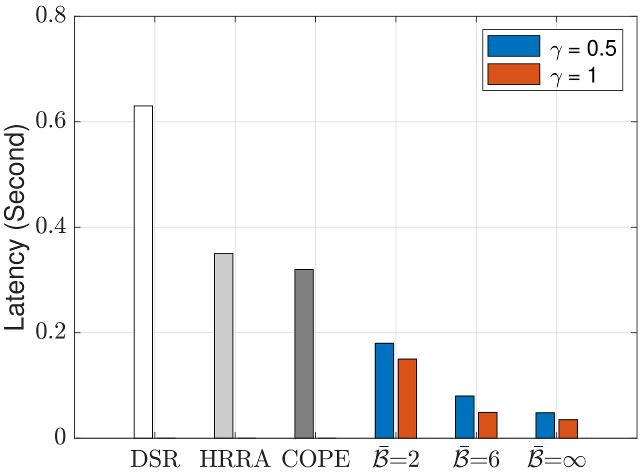


Fig. 22. Latency in the sparse random topology.

0.1553, 0.0498, and 0.035 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme has a reduced latency at 75.67%, 92.2%, and 95.51% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme reduces the latency at 52.12%, 84.64%, and 89.2% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme presents a reduced latency of 55.69%, 85.79%, and 90.01% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Higher values of $\bar{\mathcal{B}}$ and γ provide more enhanced latency gains.

2) Dense Source-Destination Pairs: The network consists of 40 randomly deployed source-destination pairs and 28 relaying nodes for this evaluation.

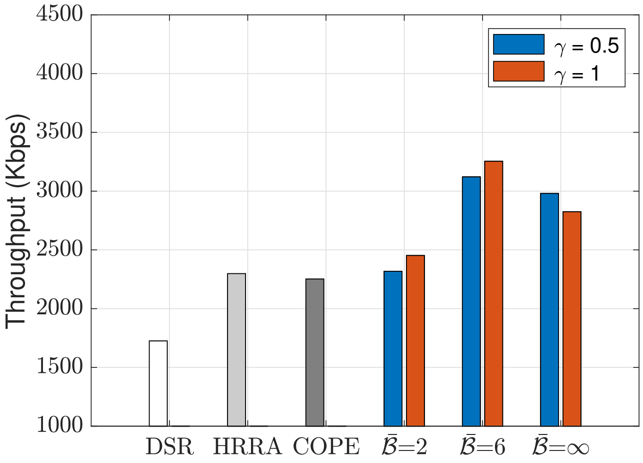


Fig. 23. Throughput in the dense random topology.

First, we evaluate the average throughput of all nodes, as shown in Fig. 23. At $\gamma = 0.5$, the proposed control provides a throughput of 2318.66, 3121.72, and 2980.61 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme enhances the throughput at 34.37%, 80.91%, and 72.73% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Further, compared with COPE, the proposed scheme enhances the throughput at 2.94%, 38.59% and 32.32% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with HRRA, the proposed scheme results in an enhanced throughput of 0.87%, 35.81% and 29.67% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. When γ is increased to 1, the proposed control provides a throughput of 2452.8, 3254.08, and 2824.5 Kbps for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme enhances the throughput at 42.17%, 88.58%, and 63.68% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with COPE, the proposed scheme has an enhanced throughput of 8.89%, 44.47% and 25.39% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Similarly, compared with HRRA, the proposed scheme enhances the throughput at 6.71%, 41.57% and 22.88% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. As $\bar{\mathcal{B}}$ increases, the average throughput also increases; however, when $\bar{\mathcal{B}}$ exceeds a particular level, only specific link resources are intensively used. Thus, the average throughput begins to decrease.

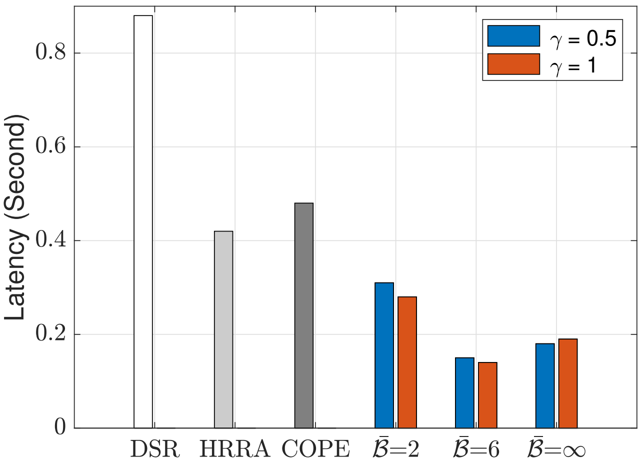


Fig. 24. Latency in the dense random topology.

Second, we measure the latency, as shown in Fig. 24. At $\gamma = 0.5$, the proposed control provides a latency of 0.315, 0.1505 and 0.1822 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme has a reduced latency of 64.29%, 82.93%, and 79.34% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme reduces the latency at 35.31%, 69.08%, and 62.58% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Likewise, compared with HRRA, the proposed scheme has a reduced latency of 26.47%, 64.86%, and 57.47% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. At $\gamma = 1$, the proposed control provides a latency of 0.2866, 0.1498, and 0.1937 s for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with DSR, the proposed scheme demonstrates a reduced latency at 67.6517%, 83.02%, and 78.04% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. In addition, compared with COPE, the proposed scheme reduces the latency at 41.14%, 69.23%, and 60.22% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Compared with HRRA, the proposed scheme has a reduced latency of 33.1%, 65.03%, and 54.79% for $\bar{\mathcal{B}} = 2, 6$, and ∞, respectively. Similar to the throughput, the average latency decreases as $\bar{\mathcal{B}}$ increases. However when $\bar{\mathcal{B}}$ exceeds a specific level, the average latency begins to increase.

VIII. Conclusion

We developed a delay-controlled distributed route establishment scheme that can provide a maximal bidirectional transmission to enhance the network coding gain while satisfying the time-constrained route setup. For route setup information forwarding at the relay nodes, we proposed a tight delay function from a strict end-to-end route setup delay bound. Second, we suggested relaxed delay functions that are more realistic and practical. Finally, we put forth a load-weighted delay function that considers the trade-off between bidirectionality and network load balancing. The simulations confirmed that the proposed scheme offers increased throughput and reduced latency for sparse/dense and mesh/random multi-hop networks compared with baseline approaches such as DSR, COPE, and HRRA. The proposed scheme can be easily combined with other conventional routing schemes without modification. Moreover, the proposed scheme can be efficiently employed for IIoT networks, in which a fast route setup, high throughput, and low latency are required under an exponentially increasing number of IoT devices and considerable interference. In a future study, we aim to design deep learning-based real-time network coding-aware control algorithms for dynamically changing network environments.

References

[1] N.-N. Dao, D.-N. Vu, Y. Lee, S. Cho, C. Cho, and H. Kim, "Pattern-identified online task scheduling in multitier edge computing for industrial iot services," *Mobile Information Systems*, vol. 2018, 2018.

[2] S. Huang, J. Cai, H. Chen, and F. Zhao, "Low-complexity priority-aware interference-avoidance scheduling for multi-user coexisting wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 112–126, 2017.

[3] B. Kang, S. Myoung, and H. Choo, "Distributed degree-based link scheduling for collision avoidance in wireless sensor networks," *IEEE Access*, vol. 4, pp. 7452–7468, 2016.

[4] G. Parissidis, M. Karaliopoulos, T. Spyropoulos, and B. Plattner, "Interference-aware routing in wireless multihop networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 716–733, 2011.

[5] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[6] R. Hou, K.-S. Lui, and J. Li, "Joint congestion control and scheduling in wireless networks with network coding," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 7, pp. 153–181, 2014.

[7] Z. Shengil, C. L. Soung, and P. L. Patrick, "Hot topic: physical-layer network coding," in *it in Proceedings of ACM MOBICOM, Los Angeles, CA*, 2006, pp. 358–365 Route–DSR.

[8] J. Goseling, M. Gastpar, and J. H. Weber, "Random access with physical-layer network coding," *IEEE Transactions on Information Theory*, vol. 61, no. 7, pp. 3670–3681, 2015.

[9] L. Qu, J. He, and C. Assi, "Congestion control, routing and scheduling in wireless networks with interference cancellation capabilities," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 7, pp. 3108–3119, 2015.

[10] R. Liu, Y. Shi, k. S. Lui, M. Sheng, Y. Wang, and Y. Li, "Bandwidth-aware high-throughput routing with successive interference cancellation in multihop wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5866–5877, 2015.

[11] L. Qu, J. He, and C. Assi, "Understanding the benefits of successive interference cancellation in multi-rate multi-hop wireless networks," *IEEE Transactions on Communications*, vol. 62, no. 7, pp. 2465–2477, 2014.

[12] C. Jiang, Y. Shi, X. Qin, a. Hou, Y. Tho, W. Lou, S. Kompella, and S. F. Midkiff, "Cross-layer optimization for multi-hop wireless networks with successive interference cancellation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5819–5831, 2016.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *IEEE/ACM Transactions on networking*, vol. 16, no. 3, pp. 497–510, 2008.

[14] X. Wei, L. Zhao, J. Xi, and Q. Wang, "Network coding aware routing protocol for lossy wireless networks," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2009, pp. 1–4.

[15] A. Singh and A. Nagaraju, "Low latency and energy efficient routing-aware network coding-based data transmission in multi-hop and multi-sink wsn," *Ad Hoc Networks*, vol. 107, p. 102182, 2020.

[16] C. Han, J. Yin, L. Ye, and Y. Yang, "Ncant: A network coding-based multipath data transmission scheme for multi-uav formation flying networks," *IEEE Communications Letters*, vol. 25, no. 3, pp. 1041–1044, 2020.

[17] S. Malathy, V. Porkodi, A. Sampathkumar, M. N. Hindia, K. Dimyati, V. Tilwari, F. Qamar, and I. S. Amiri, "An optimal network coding based backpressure routing approach for massive iot network," *Wireless Networks*, vol. 26, no. 5, pp. 3657–3674, 2020.

[18] H. Ni, Z. Guo, C. Li, S. Dou, C. Yao, and T. Baker, "Network coding-based resilient routing for maintaining data security and availability in software-defined networks," *Journal of Network and Computer Applications*, p. 103372, 2022.

[19] X. Zhou, X. Yang, J. Ma, I. Kevin, and K. Wang, "Energy efficient smart routing based on link correlation mining for wireless edge computing in iot," *IEEE Internet of Things Journal*, 2021.

[20] D. Jiang and L. Li, "Node selection algorithm for network coding in the mobile wireless network," *Symmetry*, vol. 13, no. 5, p. 842, 2021.

[21] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "Core: a coding-aware opportunistic routing mechanism for wireless mesh networks [accepted from open call]," *IEEE Wireless Communications*, vol. 17, no. 3, pp. 96–103, 2010.

[22] J. Le, J. C. Lui, and D.-M. Chiu, "Dcar: Distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 596–608, 2009.

[23] H. Song, L. Liu, S. M. Pudlewski, and E. S. Bentley, "Random network coding enabled routing protocol in unmanned aerial vehicle networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8382–8395, 2020.

[24] H. Song, L. Liu, B. Shang, S. Pudlewski, and E. S. Bentley, "Enhanced flooding-based routing protocol for swarm uav networks: Random network coding meets clustering," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[25] A. Raoof, C.-H. Lung, and A. Matrawy, "Securing rpl using network coding: The chained secure mode (csm)," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 4888–4898, 2021.

[26] X. Cheng, Q. Wang, Q. Wang, and D. Wang, "A high-reliability relay algorithm based on network coding in multi-hop wireless networks," *Wireless Networks*, vol. 25, no. 4, pp. 1557–1566, 2019.

[27] X. Jiao, X. Wang, and X. Zhou, "Active network coding based high-throughput optimizing routing for wireless ad hoc networks," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2008, pp. 1–5.

[28] S. Han, Z. Zhong, H. Li, G. Chen, E. Chan, and A. K. Mok, "Coding-aware multi-path routing in multi-hop wireless networks," in *2008 IEEE international performance, computing and communications conference*. IEEE, 2008, pp. 93–100.

[29] S.-J. Lee and M. Gerla, "Aodv-br: Backup routing in ad hoc networks," *IEEE Wireless Communications and Networking Conference*, pp. 1311–1316, 2000.

[30] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile computing*, pp. 153–181, 1996.

[31] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," in *ATM Forum contribution*, vol. 99, no. 45, 1999, pp. 1–13.

Yunseong Lee received B.S. and M.S. degrees in computer science, in 2013 and 2015, respectively, from Chung-Ang University, Seoul, South Korea, where he is currently working toward a Ph.D. degree in computer science. His current research interests include wireless communication, handover, machine learning, and cell-free network.

Taeyun Ha received a B.S. degree in computer science and engineering from Chung-Ang University, South Korea, in 2020. He is pursuing an M.S. degree in computer science and engineering at Chung-Ang University, South Korea. His research interests include deep space networking, wireless networks, and the internet of things.

Abdallah Khreishah received the B.S. degree with honors from Jordan University of Science & Technology, in 2004. He received the Ph.D. and M.S. degrees in electrical and computer engineering from Purdue University in 2010 and 2006, respectively. During the last year of his Ph.D., he worked with NEESCOM. In Fall 2012, he joined the Department of Electrical and Computer Engineering, NJIT, as an Assistant Professor and was promoted to the designation of Associate Professor in 2017. His research interests include network security, machine learning, wireless networks, visible-light communication, vehicular networks, and cloud & edge computing. His research projects are funded by the National Science Foundation of USA, New Jersey Department of Transportation, and UAE Research Foundation. He is currently an Associate Editor for several international journals including IEEE/ACM Transactions on Networking. He served as the TPC chair for WASA 2017, IEEE SNAMS 2014, IEEE SDS -2014, BDSN-2015, BSDN 2015, IOTSMS-2105. He has also served on the TPC committee of several international conferences including IEEE Infocom. He is a Senior Member of IEEE and the Chair of the IEEE EMBS North Jersey chapter.

Wonjong Noh received the B.S., M.S. and Ph.D. degrees from the Department of Electronics Engineering, Korea University, Seoul, Korea, in 1998, 2000, and 2005, respectively. From 2005 to 2007, he conducted his postdoctoral research at Purdue University, IN, USA and University of California at Irvine, CA, USA. From 2008 to 2014, he was a Principal Research Engineer with Samsung Advanced Institute of Technology, Samsung Electronics, Korea. From 2014 to 2018, he was an Assistant Professor with the Department of Electronics and Communication Engineering, Gyeonggi University of Science and Technology, Korea. He is currently an Associate Professor with the School of Software, Hallym University, Korea. His current research interests include fundamental capacity analysis in 5G/6G wireless communication and networks, federated mobile edge computing, and machine learning-based systems control. He received a government postdoctoral fellowship from the Ministry of Information and Communication, Korea, in 2005. He was also a recipient of the Samsung Best Paper Gold Award in 2010, Samsung Patent Bronze Award in 2011, and Samsung Technology Award in 2013.

Sungrae Cho is a Professor with the School of Computer Sciences and Engineering, Chung-Ang University (CAU), Seoul, Korea. Prior to joining CAU, he was an Assistant Professor with the Department of Computer Sciences, Georgia Southern University, Statesboro, GA, USA, from 2003 to 2006, and a senior member of technical staff with the Samsung Advanced Institute of Technology, Kiheung, South Korea, in 2003. From 1994 to 1996, he was a research staff member with Electronics and Telecommunications Research Institute, Daejeon, South Korea. From 2012 to 2013, he was a Visiting Professor with the National Institute of Standards and Technology, Gaithersburg, MD, USA. He received the B.S. and M.S. degrees in electronics engineering from Korea University, Seoul, South Korea, in 1992 and 1994, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2002. His current research interests include wireless networking, ubiquitous computing, and ICT convergence. He has been a Subject Editor with IET Electronics Letter since 2018, and an Editor with Ad Hoc Networks Journal (Elsevier) from 2012 to 2017. He has been an organizing committee chair for numerous international conferences, including IEEE SECON, ICOIN, ICTC, ICUFN, TridentCom, and IEEE MASS. He has also served as a program committee member for many conferences, including IEEE ICC, MobiApps, SENSORNETS, and WINSYS.