AoI-OptiIoBNT: Age of Information-Driven DNA-based Internet of Bio-Nano Things Optimization

Wanli Cheng, Jinyan Fu, Qingwen Wang, Kun Yang, *Fellow, IEEE*, Yifan Chen, *Senior Member, IEEE*, and Yue Sun*, *Member, IEEE*

***Abstract*—The Internet of Bio-Nano Things (IoBNT) integrates biosensors, nanorobots, and molecular communication, significantly extending the functionality of traditional IoT systems on a nano-scale. It holds promise for targeted drug delivery and real-time health monitoring applications. However, IoBNT faces critical challenges, including high delay, low network reliability, and congestion, primarily due to biological environments' complex and dynamic nature. DNA emerges as an ideal information carrier for IoBNT due to its high information density, longevity, biocompatibility, and robustness against environmental interference. These properties make DNA uniquely suited for reliable and efficient communication within IoBNT, with additional functionalities in bio-sensing and DNA computing. This paper proposes AoI-OptiIoBNT, an innovative routing and packet forwarding strategy designed to optimize DNA-based information flow in IoBNT. AoI-OptiIoBNT combines an Age of Information (AoI)-driven approach with a Markov Decision Process (MDP)-based routing algorithm to mitigate delay and congestion. It incorporates a multi-retransmission strategy to enhance network reliability and introduces a Yin-Yang Coding (YYC) mechanism to reduce error rates and improve decoding accuracy. Simulation results demonstrate that AoI-OptiIoBNT substantially improves the efficiency, reliability, and overall performance of IoBNT networks. It offers a robust framework for addressing congestion, packet loss, and delay, making it a promising solution for advancing IoBNT applications.**

***Index Terms*—DNA-based IoBNT, MDP, network congestion, Age of Information (AoI), multi-retransmission.**

I. Introduction

The Internet of Bio-Nano Things (IoBNT) extends the traditional Internet of Things (IoT) to the nanoscale by integrating biology, nanotechnology, and information communication technologies [1]. Utilizing nanoscale sensors and nanorobots, IoBNT enables sensing, processing, and actuation tasks for applications such as targeted drug delivery, real-time health monitoring, environmental pollutant detection, and precision agriculture [2]–[4]. Its primary objective is to develop effective information transmission and network design methods in the biochemical domain through interdisciplinary technologies like molecular communication, nanosensing, and intelligent swarm technologies. Additionally, IoBNT seamlessly integrates with the electrical Internet via innovative bio-network interfaces (as shown in Fig. 1). Highly sensitive, biocompatible nanosensors ensure accurate biological signal detection and long-term safety. By providing nanoscale precision and efficiency beyond macro-scale technologies, IoBNT revolutionizes these domains and fosters the development of an interconnected Nano-Biotechnology ecosystem.

Molecular communication (MC) is critical for IoBNT, with the selection of information carriers essential for data transfer among nanorobots, in vivo networks, and broader intro-body networks. DNA is a prime candidate for high-throughput transmission due to its longevity and high information density [5]. Its ability to encode information, combined with DNA logic gates and origami for constructing complex nanostructures [6], [7], facilitates precise intra- and extra-body information integration in IoBNT. However, the effectiveness of IoBNT relies on robust transmission and routing mechanisms. Traditional molecular communication via diffusion (MCvD) suffers from inefficiencies and high delays, leading to inter-symbol and inter-link interference [8]. While molecular motors have been considered as alternatives, their unidirectional control and energy conversion limitations hinder their application [9]. In contrast, track-hopper-based MC provides externally controlled molecular directionality and adjustable step lengths. Moreover, the proposed scheme operates without the need for external energy sources and prevents off-track phenomena observed in molecular motors, presenting a promising solution for enhancing IoBNT performance [10]. Characterization of nanoparticle motion states (e.g., chemical molecules, molecular motors, molecular hoppers) in MC systems is critical for mechanism understanding and system optimization [11], [12]. Markov processes effectively model the stochastic dynamics of core transport mechanisms (diffusion/active transport) due to their memoryless property, where future states depend only on current conditions. This property also aligns inherently with IoBNT node state modeling (congestion/packet transmission), eliminating complex historical data handling and reducing

*Corresponding author: Yue Sun email:sunyuestc90@126.com
This work is supported by the National Natural Science Foundation of China (Grants No. 62301088, 62171106) The work of Yifan Chen is supported by the Municipal Government of Quzhou under Grant No. 2024D065 and No. 2023D031, and the Incubation Program for Innovative Science and Technology in the University of Electronic Science and Technology of China under Grant Y03023206100209.

Wanli Cheng, Yue Sun, and Yifan Chen are with the School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu 610000, China. Wanli Cheng, and Yue Sun are also with the School of Mechanical and Electrical Engineering, Chengdu University of Technology, Chengdu 610059, China.(email:sunyuestc90@126.com, yifan.chen@uestc.edu.cn)

Wanli Cheng, Jinyan Fu, and Qingwen Wang are with the School of Mechanical and Electrical Engineering, Chengdu University of Technology, Chengdu 610059, China. (email:cwl1716@163.com; 17308040729@163.com; wangqw818@126.com).

Kun Yang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.,(email: kunyang@essex.ac.uk).

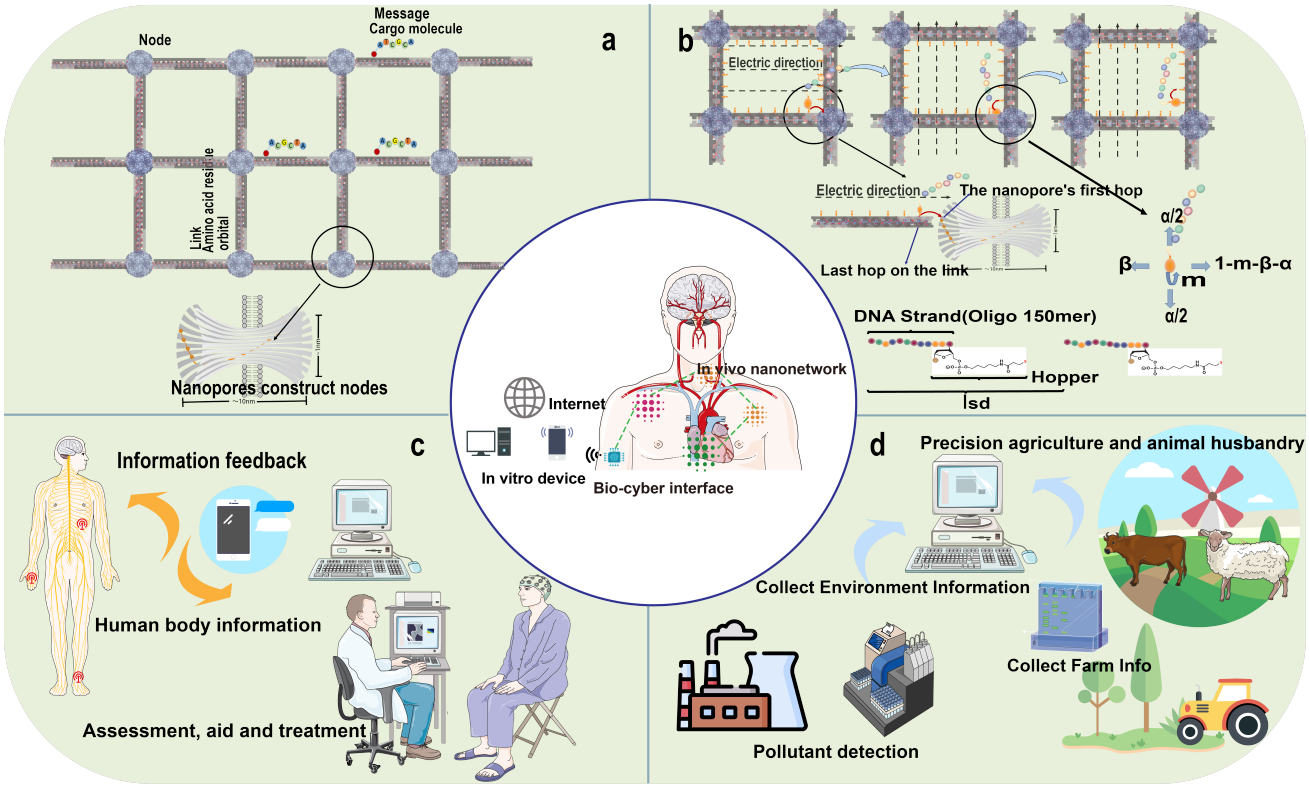


Fig. 1: The figure (a) illustrates the DNA-based IoBNT network topology; (b) demonstrates the hopper control process and the dynamic motion-state probability model for the track-hopper-based MC; (c) shows the application of IoBNT in human health monitoring and precision medicine; and (d) presents the application of IoBNT in environmental monitoring and agriculture.

computational complexity [13]–[15].

Currently, routing and forwarding strategies in IoBNT systems face persistent challenges such as high delay, congestion, and low reliability. The large-scale deployment of nanodevices often leads to extended multi-hop paths, which are prone to congestion and hinder real-time response ability. This issue is further exacerbated by the high information density of DNA molecules, imposing significant processing delays at intermediate and receiving nodes. For instance, nanopore-based sequencing technologies remain limited to approximately 400 nt/s [16], creating decoding bottlenecks. Existing approaches, including GA-based multi-hop routing [17] and adaptive molecule release strategies [18], offer improvements in energy efficiency and path optimization. However, most are designed for MCvD, where molecules propagate freely via Brownian motion. These models assume isotropic diffusion and are generally incompatible with DNA-based track-hopper systems, where transmission occurs along engineered tracks with directional relays. Moreover, these methods lack dynamic, congestion-aware control and neglect the timeliness of data, typically measured by the Age of Information (AoI). AoI has emerged as a critical metric for maintaining data freshness and supporting real-time decisions in bio-nano systems [19]–[21]. This limits their applicability in latency-sensitive scenarios such as medical monitoring or targeted drug delivery. While existing congestion models provide useful insights [22], they do not enable adaptive decision-making. Recent work emphasizes the importance of integrating congestion modeling with scheduling and policy updates to reduce delay and improve reliability [23]. However, existing routing protocols and packet scheduling mechanisms are designed primarily for larger-scale networks and do not address the characteristics of nanonetworks, resulting in inefficient information flow control and priority management. Current routing and forwarding strategies in IoBNT systems face significant challenges. This mismatch limits the broader adoption of IoBNT.

In our previous work [13], we proposed a DNA-based track-hopper MC system for IoBNT, which outlines the mesh topology network architecture [13]. However, [13] primarily focused on the mesh network topology and propagation analysis, without addressing critical issues such as network congestion, high transmission delay, and average AoI in IoBNT systems. Building on our previous work, this paper proposes AoI-OptiIoBNT, an innovative packet forwarding and routing protocol designed to reduce delay, network congestion, and average AoI, thus optimizing DNA-based information flow in IoBNT. Fig. 1(a) illustrates the core mesh network topology, where nodes are constructed by nanopores embedded within lipid bilayers. These nodes are interconnected through amino-acid-based linker channels, forming a modular and scalable framework. Message carriers, represented as encoded DNA strands, travel along these tracks, forming the basis of molecular information transfer in IoBNT. Fig. 1(b) delves into the packet forwarding process at the molecular level, demonstrating the electric field-driven hopping mechanism of the DNA-based track-hopper. The hopper undergoes multiple

translocations through nanopores, representing each hop on a communication link. The subfigure also includes the probability transition model for each hop, and the corresponding chemical structure of the DNA cargo, which ensures reliable encoding and directional control under electrophoretic forces. AoI-OptiIoBNT combines an AoI-driven approach with a Markov Decision Process (MDP)-based routing algorithm to mitigate delay and congestion [24]. It incorporates a multi-retransmission strategy to enhance network reliability and introduces the Yin-Yang Coding (YYC) mechanism [25] to reduce error rates and improve decoding accuracy. The simulation results demonstrate that AoI-OptiIoBNT substantially improves the efficiency, reliability, and overall performance of IoBNT networks. It offers a robust framework for addressing congestion, packet loss, and delay, making it a promising solution for advancing IoBNT applications. To highlight the real-world applicability of the proposed system, Fig. 1(c) presents its use in personalized health monitoring. Here, an in vivo nanonetwork continuously senses physiological parameters and transmits data through a bio-cyber interface to an external in vitro system. This configuration supports real-time health tracking, diagnostic feedback, and personalized treatment. Fig. 1(d) illustrates another application in precision agriculture and environmental monitoring, where bio-nano devices are deployed to collect ambient data, including pH, temperature, and pollutants. This information is relayed to a central system for analysis, enabling environmental protection.

The structure of this paper is as follows: Section II introduces the overall system design and the network congestion model. Section III presents the Markov state transitions, proposes an MDP-based routing algorithm for network congestion and the AoI-based packet forwarding and retransmission strategy. Section IV provides a detailed description of the YYC mechanism and the network delay and reliability models constructed to validate the performance of the proposed AoI-OptiIoBNT framework. Section V discusses our simulation setup and the empirical validation of the simulations. Section VI synthesizes our main findings and concludes the paper.

II. AoI-OptiIoBNT: An Integrated Solution for Network Congenstion

A. AoI-OptiIoBNT Framework

Network congestion is a critical issue in IoBNT systems, significantly impacting the efficiency and reliability of information transmission. In this section, we develop a comprehensive model to quantify the degree of congestion in IoBNT networks and analyze its impact on overall system performance. By understanding the root causes and dynamics of congestion, we can design effective strategies to mitigate its effects and optimize network operations.

To address these network congestion and delay challenges, we propose AoI-OptiIoBNT, an integrated routing, forwarding, and coding strategy designed to optimize DNA-based information flow in IoBNT. AoI-OptiIoBNT combines an AoI-driven approach with an MDP-based routing algorithm to mitigate delay and congestion. It incorporates a multi-retransmission strategy to enhance network reliability and introduces the YYC compression coding mechanism to reduce the number of packets, thus reducing error rates. Simulation results demonstrate that AoI-OptiIoBNT substantially improves the efficiency, reliability, and overall performance of IoBNT networks. It offers a robust framework for addressing congestion, packet loss, and delay, making it a promising solution for advancing IoBNT applications. The general information flow processing in IoBNT is illustrated in Fig. 2(a), which includes key components such as data transmission, decision-making processes, and data packet forwarding. Figs. 2(b) demonstrates the mechanisms of MDP. The MDP is a mathematical framework constructed based on elements such as the state space, action space, transition probability, and reward function. In this process, the agent selects an action according to the current state. The action changes the state through the transition probability and the agent obtains the corresponding reward. The goal is to find the optimal strategy to maximize the long-term cumulative reward. Figs. 2(c) illustrates the temporal evolution of the AoI for a single data packet. Starting from an initial AoI value of 1, each data packet has two states within a time slot: it can be successfully received or experience a packet loss. These outcomes ultimately affect the AoI of the system. Lastly, Figs. 2(d) illustrates the YYC coding mechanism for packet compression. A detailed analysis of (c) and (d) will be conducted in Section III.

At the transmission end, the source node packages the source information into packets via DNA coding. Upon data generation, the system selects the most appropriate routing path based on the current network state, modeled using MDP. The optimal data packet forwarding path is determined during the routing decision, considering network congestion and link conditions. This approach helps mitigate the impact of congestion on data transmission, a critical issue in IoBNT networks.

Once a routing decision is made, the data packet is forwarded along the selected path to the next node. The receiving node calculates and updates the AoI value of the data packet to reflect the latest transmission time. In addition, a feedback retransmission mechanism is integrated into the system to ensure network reliability. This mechanism allows the retransmission of lost or corrupted data packets, which helps maintain the integrity of the transmitted information.

The data packets are decoded at the receiving end to reconstruct the source information, completing the transmission process. Integrating the MDP-based routing protocol and AoI-driven packet forwarding strategy, AoI-OptiIoBNT ensures an optimized and reliable information flow in IoBNT systems. This strategy not only addresses the critical issue of network congestion but also enhances overall system performance by reducing delay and improving information freshness.

Although MC and IoBNT operate at the nanoscale, they share key principles with traditional networking systems, such as routing, congestion control, and information freshness. Traditional communication protocols, such as TCP/IP and probabilistic routing, rely on well-established mechanisms like error correction, retransmissions, and congestion control, all designed to ensure efficient data transfer. Similarly, in molecular communication, the use of DNA and molecular carriers

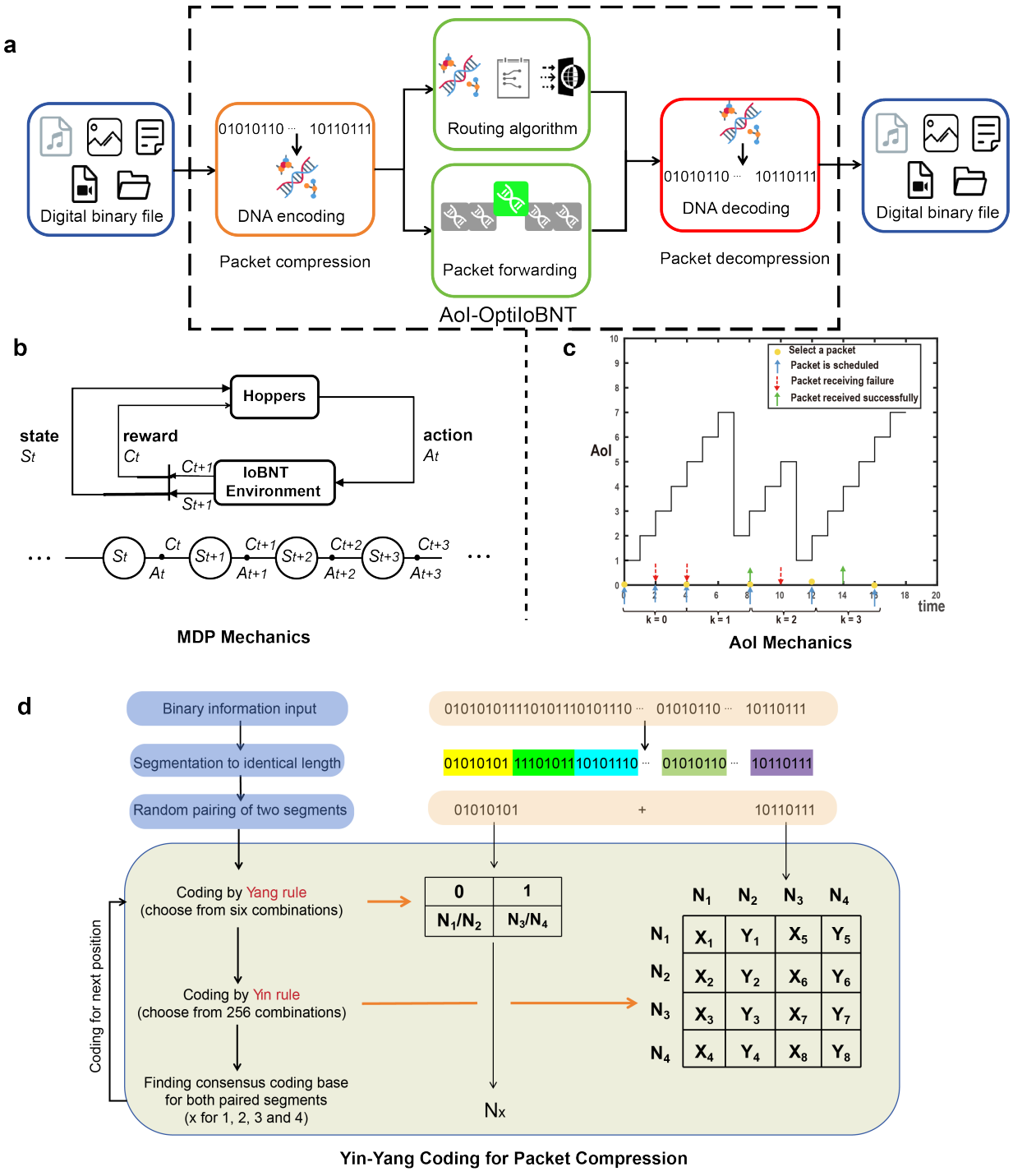


Fig. 2: AoI-OptiIoBNT Framework: Information flow processing in DNA-based IoBNT.

enables highly dense, reliable data transmission within IoBNT. However, the unique challenges of the biological environment, such as diffusion delays, interference, and limited processing speeds at the molecular level, require adaptations to these traditional models. In our approach, we bridge the gap between molecular and conventional communication by incorporating MDP for routing, akin to the probabilistic decision-making in traditional networks, and feedback-driven retransmissions to enhance reliability. The integration of AoI-driven forwarding strategies further optimizes the timeliness of data transmission, a key consideration in real-time applications such as medical monitoring and targeted drug delivery. Thus, by combining bio-inspired networking approaches with established communication principles, our AoI-OptiIoBNT framework offers a hybrid solution that enhances both the efficiency and scalability of IoBNT systems, making them more adaptable to the dynamic conditions of nanoscale environments.

B. Network Congestion Model in DNA-based Track-hopper IoBNT

Before accurately simulating the network congestion levels of the IoBNT, it is essential to clarify the status and condition of each link within the network [26]. Inspired by vehicle queuing models in road traffic, we have developed a track-hopper-based dynamic MC transmission queuing dynamics model tailored explicitly for the IoBNT architecture [27]. This innovative model employs the utilization rate of protein track as the core dynamic variable, providing a detailed description of the operational mechanisms of the orbits and the track-hopper system. By adjusting the length of the protein track, we can construct a comprehensive and unified framework to simulate the information transmission states throughout the entire network. For instance, considering a protein orbit labeled i, its utilization rate fluctuates dynamically with the number of molecular hoppers, thereby revealing the dynamic characteristics of system congestion at the molecular level. The following is a complete process of exploring the utilization rate of a single hopper on the track:

$$\alpha_i = \frac{l_{sd}}{l_i} \tag{1}$$

In Eq. 1, α_i denotes the utilization rate of an individual hopper on track i. The variable l_i specifies the length of track i, while l_{sd} represents the spacing between consecutive molecular hoppers transporting DNA sequences. This spacing accounts for a safety margin, which is defined based on the potential for hoppers to retreat during transport, considering two possible retreat events. Since the DNA sequences carried by the hoppers are fixed, l_{sd} is calculated as the sum of the DNA sequence length, the hopper length, and the safety distances required for the two retreat events.

Based on the above analysis, we extend our analysis to investigate track utilization in multiple molecular hoppers scenarios.

$$U_i\left(t\right) = \alpha_i R_i\left(t\right) \tag{2}$$

where $U_i(t)$ denotes the utilization rate of track i at the t-th time step, while $R_i(t)$ represents the number of molecular hoppers occupying track i during the same time step.

Leveraging the previously defined parameters, the precise number of hoppers transmitted along track i at a given time step t can be determined as follows:

$$O_i\left(t\right) = min\left(R_i\left(t\right), X_i\left(t\right)\right) \tag{3}$$

$$X_i\left(t\right) = \frac{T}{h_{min}} \tag{4}$$

This calculation accurately measures hopper throughput on each track, facilitating a deeper understanding of congestion dynamics within the IoBNT network. T denotes the total duration of the processing time interval.And $O_i(t)$ denotes the number of hoppers exiting track i within the time interval $[tT, (t+1)T]$. Similarly, $X_i(t)$ signifies the number of hoppers that have been processed and successfully reached the end node of track i during the time interval t. The parameter h_{min} represents the time interval required between the processing of two successive hoppers at the node. This interval is determined by summing the time needed to process the DNA sequences at the node and the time required to forward the information.

Integrating the previously discussed concepts, we can establish the relationship between the utilization rate of track i at time step $(t+1)$ and its utilization at the current time step t:

$$U_i\left(t+1\right) - U_i\left(t\right) = \alpha_i\left[I_i\left(t\right) - O_i\left(t\right)\right] \tag{5}$$

Let $I_i(t)$ denote the number of hoppers entering track i within the time interval $[tT, (t+1)T]$. The definitions and implications of the remaining parameters are detailed in Equations (1) through (4). This equation allows us to understand how the current utilization of a track influences its future state, providing valuable insights into its temporal dynamics and establishing a comprehensive framework for analyzing the system's behavior over time.

According to the framework presented in [28], assume that a molecular hopper successfully enters track i with a probability p and diverts to an incorrect track with a probability $1-p$. Over n time steps, the number of hoppers M that correctly enter track i follows a binomial distribution, denoted as $M \sim \mathcal{B}(n, p)$. When n and np is sufficiently large, the binomial distribution can be approximated by a Poisson distribution $\mathcal{P}(\lambda)$. Therefore, the probability of exactly m hoppers reaching track i within the time interval $[tT, (t+1)T]$ is expressed as:

$$P\left(I_i\left(t\right) = m\right) = \frac{\lambda_i(t)^m \cdot e^{-\lambda_i(t)}}{m!} \tag{6}$$

where $\lambda_i(t)$ denotes the anticipated arrival rate of hoppers on track i during each time interval T at time step t. This rate is derived from the network topology illustrated in Fig. 1(b). Within this figure, m represents the probability of a hopper remaining stationary, β signifies the probability of a hopper moving in the reverse direction, and $\frac{\alpha}{2}$ indicates the probability of a hopper transmitting information incorrectly. Taking into account the dual-path characteristics of the network topology, the expected arrival rate for track i at each time interval T and time step t is ultimately calculated as $2 \times (1 - m - \beta - \alpha)$.

III. MDP-Based Routing and AoI-Driven Forwarding Strategy

A. Markov State Transition Model

In this section, we develop a model to describe the transitions between different track occupancy states and derive the corresponding Markov state transition matrix. We classify the occupancy states of a track into three distinct levels:

- **State 1** No congestion.
- **State 2** Moderate congestion.
- **State 3** Severe congestion.

Based on this classification, we formulate the discretization equation for the state of track i as follows:

$$s_i(t) = \begin{cases} 1 & 0 \le U_i(t) < a \\ 2 & a \le U_i(t) < b \\ 3 & b \le U_i(t) \le 1 \end{cases} \tag{7}$$

where a and b represent the corresponding track utilization rates. For the three states, the track utilization rate $U_i(t)$

follows a discrete uniform distribution with two parameters (upper and lower bounds). This discretization maps the track utilization rate $U_i(t)$ into three distinct states corresponding to different congestion levels.

To compute the transition probabilities, we must estimate how the network state changes over time. The transition matrix $P_i(t)$ can be computed based on the network's current congestion state and the dynamics of the track utilization. The transition matrix can be described as follows:

$$P_i(t) = \begin{bmatrix} p_i^{11}(t) & p_i^{12}(t) & p_i^{13}(t) \\ p_i^{21}(t) & p_i^{22}(t) & p_i^{23}(t) \\ p_i^{31}(t) & p_i^{32}(t) & p_i^{33}(t) \end{bmatrix} \tag{8}$$

The probability of each state transition is defined as:

$$p_i^{mn}(t) = \Pr(s_i(t+1) = n \mid s_i(t) = m) \tag{9}$$

where $m, n = 1, 2, 3$. This matrix describes the likelihood of transitioning between different congestion states based on the current system parameters, such as $\lambda_i(t)$, the expected arrival rate of packets at track i, and t_p, the node processing time at time step t.

For example, the transition probability $p_i^{11}(t)$ represents the probability of staying in state 1 (no congestion) given that the system is currently in state 1. To compute this, we use the following approach:

$$p_i^{11}(t) = \sum_{n=1}^{\left\lfloor \frac{a}{\alpha_i} \right\rfloor} \frac{\lambda_i(t)^n \cdot e^{-\lambda_i(t)}}{n!} \tag{10}$$

where $\lfloor x \rfloor$ is the floor function, which rounds x down to the nearest integer. This formula models the probability of staying in state 1 based on the current track utilization and packet arrival rates.

When $\alpha_i \cdot X_i(t)$ is less than a, we can calculate the state of $U_i(t+1)$ as follows:

$$U_i\left(t+1\right) = \begin{cases} \alpha_i I_i\left(t\right), \\ 0 \le U_i\left(t\right) < \alpha_i \cdot X_i\left(t\right) < a \\ \\ U_i\left(t\right) + \alpha_i I_i\left(t\right) - \alpha_i \cdot X_i\left(t\right), \\ 0 \le \alpha_i \cdot X_i\left(t\right) < U_i\left(t\right) < a \end{cases} \tag{11}$$

The transition probability $p_i^{11}(t)$ can now be defined in terms of the total probability law:

$$\begin{aligned} p_i{}^{11}\left(t\right) = & Pr\left(s_i\left(t+1\right) \mid U_i\left(t\right) \in \left[0, \alpha_i \cdot X_i\left(t\right)\right)\right) \\ & \cdot Pr\left(U_i\left(t\right) \in \left[0, \alpha_i \cdot X_i\left(t\right)\right)\right) \\ & + Pr\left(s_i\left(t+1\right) \mid U_i\left(t\right) \in \left[\alpha_i \cdot X_i\left(t\right), a\right)\right) \\ & \cdot Pr\left(U_i\left(t\right) \in \left[\alpha_i \cdot X_i\left(t\right), a\right)\right) \end{aligned} \tag{12}$$

Finally, using the export transition probability, we can calculate according to the transfer happened in time $(t+1)$ on the track i molecular average number of hopper:

$$\overline{R}_i^{11}(t+1) = \frac{1}{\alpha_i} \sum_{n=0}^{y} U_i(t+1) \cdot \Pr(U_i(t+1) = n\alpha_i) \tag{13}$$

In similar fashion, we can calculate the whole state transition matrix $P_i(t)$ and the average number of hoppers $\bar{R}_i(t+1)$ for each probabilistic event, which is denoted as:

$$\bar{R}_i(t+1) = \begin{bmatrix} \bar{R}_{11}^i(t+1) & \bar{R}_{12}^i(t+1) & \bar{R}_{13}^i(t+1) \\ \bar{R}_{21}^i(t+1) & \bar{R}_{22}^i(t+1) & \bar{R}_{23}^i(t+1) \\ \bar{R}_{31}^i(t+1) & \bar{R}_{32}^i(t+1) & \bar{R}_i(t+1) \end{bmatrix} \tag{14}$$

The transition matrix is updated iteratively, allowing for the computation of the optimal routing policy through the integration of MDP-based decision-making.

And if the current state $s_i(t)$ is determinate, we can get the mean queue length $\bar{Q}_i^{s_i(t)}(t+1)$ for the next interval by:

$$\bar{R}_i^m(t+1) = \sum_{n=1}^{3} \bar{R}_i^{mn}(t+1) \cdot p_i^{mn}(t) \tag{15}$$

with $m, n = 1, 2, 3$.

Once $\overline{R}_i^m(t+1)$ is determined, according to Equations (3) and (4), the number of tokens arriving at the receiving node via link i within the time interval $(t+1)$ under state m can be calculated. This value corresponds to the number of output data packets under specified conditions, expressed as:

$$\overline{O}_i^m(t+1) = \min\left(\overline{R}_i^m(t+1), \frac{T}{n_{\min}}\right) \tag{16}$$

Here, $\overline{O}_i^m(t+1)$ is a function of the token count $\overline{R}_i^m(t+1)$ representing the number of output data packets. This calculation reflects the amount of information successfully transmitted through the link in the current state, which helps evaluate the efficiency of data transmission and bandwidth utilization in the network.

B. MDP-based Routing algorithm

The Markov state transition model for relay nodes is based on the states of the input and output links of each node. According to our network topology model, as seen in Fig. 1(a), each relay node has two output and two input links. Additionally, the node's control system manages the direction of the next hop for each packet.

The dimensionality of the state space for a relay node depends on the number of output links. Since the state is a combination of the states of two output links, the state of node j is characterized as follows:

$$s^j\left(t\right) = \begin{cases} (1,1) & s_1\left(t\right) = 1, s_2\left(t\right) = 1 \\ (1,2) & s_1\left(t\right) = 1, s_2\left(t\right) = 2 \\ \vdots & \vdots \\ (3,3) & s_1\left(t\right) = 3, s_2\left(t\right) = 3 \end{cases} \tag{17}$$

where s_1 and s_2 respectively represent the states of the two output links.

The action space $A_j = \{a_{j1}, a_{j2}, a_{j3}\}$, where for each state, there are three available actions: forward upward, forward rightward, and stagnation. In this context, stagnation refers to the action where no forwarding occurs, and the packet remains at the node. Therefore, the policy space is represented as follows:

$$G = \times_{s \in S} A\left(s\right), \tag{18}$$

where $A(s) = A^j, s \in s^j$, the notation $\times$ represents the Cartesian product, $G = \times_{s\in S} A(s)$ represents the selection of an action from the set of possible actions for each state. The collection of all these selections forms the policy space, which includes all possible combinations of actions taken in every possible state, encompassing all strategies.

Based on the description of the link state transitions provided, for any action a_i^j belonging to A^j, where $i = 1, 2, 3, \ldots, l$, we can compute the state transition matrices for two links, denoted as $P_1(t)$ and $P_2(t)$. Additionally, we can calculate the corresponding matrices for the average number of data packets, denoted as follows:$\overline{R}_1(t+1)$ and $\overline{R}_2(t+1)$. Subsequently, by utilizing the policy $g_i = \times_{s\in S} a_i^j$, the transition probability for the relay nodes can be expressed as follows:

$$P^{j,g_i}(t) = \begin{bmatrix} p^{j,g_i}_{11\to 11}(t) & p^{j,g_i}_{11\to 12}(t) & \cdots & p^{j,g_i}_{11\to 33}(t) \\ p^{j,g_i}_{12\to 11}(t) & p^{j,g_i}_{12\to 12}(t) & \cdots & p^{j,g_i}_{12\to 33}(t) \\ \vdots & \vdots & \ddots & \vdots \\ p^{j,g_i}_{33\to 11}(t) & p^{j,g_i}_{33\to 12}(t) & \cdots & p^{j,g_i}_{33\to 33}(t) \end{bmatrix} \tag{19}$$

where,

$$p^{j,g_i}_{m_1n_1\to m_2n_2}(t) = p_1^{m_1n_1}(t)\, p_2^{m_2n_2}(t), \tag{20}$$

with $m_1, m_2, n_1, n_2 = 1, 2, 3$.

The reward matrix $C^{j,g_i}(t)$ is defined as the penalty for congestion levels, and is expressed as follows:

$$C^{j,g_i}(t) = \left[c_{11}^{j,g_i}(t), c_{11}^{j,g_i}(t), \cdots, c_{33}^{j,g_i}(t)\right], \tag{21}$$

the specific penalty value can be calculated using the following equation:

$$\begin{aligned} c_{mn}^{j,g_i}(t) = {} & \mu_1 \left[\bar{R}_1^m(t+1) + \bar{R}_2^n(t+1)\right] \\ & + \mu_2 \left|\bar{R}_1^m(t+1) - \bar{R}_2^n(t+1)\right|, \end{aligned} \tag{22}$$

We use the number relationship of molecular hoppers on link 1 of state m and link 2 of state n flowing through node j at time step $(t+1)$, including quantity and quantity difference, as the reward function. where $m, n = 1, 2, 3$, μ_1 and μ_2 are weight coefficients that can be specifically determined for particular information flow control problems. By incorporating both the sum and the difference of quantities, the flow between link 1 and link 2 can be balanced, μ_1 can be used to encourage the maximization of total flow, while μ_2 helps reduce the quantity difference between the two links, ensuring that the system does not experience flow imbalances, thereby improving efficiency.

Considering all the actions in A^j, the basic state transition matrix $P^j(t)$ can be represented as:

$$P^j(t) = \left[P^{j,g_1}(t), P^{j,g_2}(t), \cdots, P^{j,g_l}(t)\right]^T, \tag{23}$$

the involved basic reward matrix is defined as:

$$C^j(t) = \left[C^{j,g_1}(t), C^{j,g_2}(t), \cdots, C^{j,g_l}(t)\right]^T. \tag{24}$$

For the other policy $g' \in G$ and $g' \neq g_i$, $i = 1, 2, \ldots, l$, the state transition matrix and reward matrix can be obtained from $P^j(k)$ and $C^j(k)$.

The electric field control process of relay node j can be succinctly described as follows: At the start of each cycle T_m, based on the information available from the last M cycles within the network, each with a time interval T, this set of information is used to calculate and update the Markov state transition model consisting of $P^j(t)$ and $C^j(t)$. Then, the Policy Iteration (PI) algorithm is applied to find an optimal strategy that minimizes the average penalty values $\zeta^{j,g}$, for $g \in G$, but only to be executed within the next cycle T_m. During each time interval T within the next cycle T_m, it is straightforward to select the forwarding path for each set of data packets based on real-time state information and the obtained optimal strategy. Algorithm 1 gives the pseudocode for the strategy.

C. AoI-Driven Packet Forwarding Policy

In IoBNT systems, the timely delivery of time-sensitive data, such as human health data and network status updates, is crucial. For instance, in the medical field, adjusting treatment plans based on the latest health data can significantly enhance diagnostic accuracy and treatment effectiveness.

To quantify the timeliness of data, the metric known as the AoI has been proposed [20]. AoI measures the freshness of data from the perspective of the destination node, defined as the time elapsed since the latest data received at the destination were generated. Specifically, if the latest data received at the destination at time t were generated at the source at time u, the AoI at the destination at time t is $t - u$. Compared to delay, AoI provides a more adequate representation of packet data freshness because it accounts for the transmission delay between the source and destination and the update interval between consecutive data transmissions.

In this paper, we propose an AoI-driven packet forwarding strategy implemented at network nodes. This strategy dynamically adjusts the order and action of packet forwarding based on the AoI value of each packet upon arrival at a node. Additionally, the strategy incorporates a feedback retransmission mechanism to enhance reliability and ensure timely data delivery [29]. In conventional communication systems, error control is typically achieved through automatic repeat request (ARQ) or forward error correction (FEC) [30]. These methods rely on consistent and low-delay feedback channels to trigger retransmissions or correct errors using redundant bits. However, the traditional feedback retransmission and error control methods mentioned above cannot be directly applied to IoBNT systems due to the unreliability and delay of MC feedback, the limited computing power of nanonodes, and the risk of MC congestion due to excessive redundancy in coding or retransmission [26]. To address this, we implement a feedback-based retransmission scheme tailored to the molecular domain. Upon successful decoding, the receiver emits an acknowledgment molecule; the sender retransmits only if no acknowledgment is detected within a timeout window. Unlike conventional ARQ, our mechanism imposes a strict upper limit on the number of retransmissions ($N_{\max}(t)$), thus preventing excessive delays and buffer overflow. Moreover, unlike electronic systems where retransmissions incur additional energy cost from limited power sources, our IoBNT model assumes intracellular energy supply (e.g., ATP) for powering

Algorithm 1 Optimized Routing Algorithm Based on Markov Decision Process

Initialize state transition matrices P and reward matrices C
Define packet delivery status as $delivered =$ False
function UPDATEMODEL($node$, $data$)
    Update P and C using $data$
    return updated P and C
end function
function COMPUTEOPTIMALPOLICY(P, C)
    Initialize policy g
    repeat
        Apply the Policy Iteration (PI) algorithm to compute optimal policy minimizing long-run reward ζ^g
        if policy converges to g **then**
            return g
        end if
    until convergence
end function
while $delivered ==$ False **do**
    Collect network data from the last M cycles
    for each node j **do**
        $P^j, C^j =$ UPDATEMODEL(j, collected data)
        $optimalPolicy =$ COMPUTEOPTIMALPOLICY(P^j, C^j)
    end for
    for each node j **do**
        $currentState =$ get current state of node j
        for each time interval T **do**
            $action = optimalPolicy[currentState]$
            if action is feasible **then**
                Execute $action$
                Update $currentState$
                if $j == node_{recv}$ and $currentState ==$ target state **then**
                    $delivered =$ True
                    break
                end if
            end if
        end for
        if $delivered$ **then**
            break
        end if
    end for
end while

molecular operations [31]. Therefore, retransmissions do not impose external energy burdens, and system design focuses primarily on balancing reliability and latency rather than energy efficiency.

The necessary variables for the analysis are defined as follows: the frame index is denoted by $k \in \{0, 1, 2, \ldots\}$ and the slot index within a frame by $t \in \{0, 1, 2, \ldots, T-1\}$. Together, (k, t) uniquely identifies a slot. The scheduling decision for each packet i at node j during the slot (k, t) is represented by $d_k^{j,i}(t) \in \{0, 1\}$. Let Γ denote the collection of all packets at node j. The outcome of transmitting packet i is indicated by $\zeta_k^{j,i}(t) \in \{0, 1\}$, where $\zeta_k^{j,i}(t) = 1$ signifies a successful delivery, and $\zeta_k^{j,i}(t) = 0$ denotes a failure. The event U represents the successful reception of packet i by a downstream node at the conclusion of slot (k, t). The probability of occurrence for event U is expressed as follows:

$$\begin{aligned} P(U) &= P\left(\zeta_k^{j,i}(t) = 1 \mid d_k^{j,i}(t) = 1\right) P\left(d_k^{j,i}(t) = 1\right) \\ &= P\left(d_k^{j,i}(t) = 1\right) p_i, \end{aligned} \tag{25}$$

where p_i represents the probability that each packet is successfully received at the receiver.

Then the AoI value for packet i at node j during slot (k, t) is denoted as $\Delta_k^{j,i}(t)$. Fig. 2(c) illustrates the temporal evolution of the AoI for a single packet i, beginning with an initial AoI value of 1.

Considering the possibility of packet loss in the system, if packet i is scheduled to transmit a packet during slot (k, t), two scenarios can occur:

- **Scenario 1:** If there is no packet loss, the downstream node will correctly receive the packet, and the AoI of packet i at slot $(k, t+1)$ will be updated to the number of slots elapsed since its creation (i.e., $t+1$).
- **Scenario 2:** If packet loss occurs, meaning the downstream node does not receive the packet, the AoI at slot $(k, t+1)$ will evolve to $\Delta_k^{j,i}(t) + 1$.

Additionally, if packet i is not scheduled for forwarding in slot (k, t), it will remain holding during this slot, and its AoI at slot $(k, t+1)$ will be $\Delta_k^{j,i}(t) + 1$. For ease of calculation, $\Delta_k^{j,i}(T)$ is used to denote $\Delta_{k+1}^{j,i}(0)$, indicating that the AoI of packet i is a continuous process. Thus, the dynamic evolution of packet i's AoI is as follows:

$$\Delta_k^{j,i}(t+1) = \begin{cases} \Delta_k^{j,i}(t) + 1, & d_k^{j,i}(t) = 1, \zeta_k^{j,i}(t) = 0 \\ & or\, d_k^{j,i}(t) = 0 \\ t+1, & d_k^{j,i}(t) = 1, \zeta_k^{j,i}(t) = 1. \end{cases} \tag{26}$$

Slots within each frame are categorized into scheduling slots and retransmission slots. Decisions in scheduling slots are made according to a feasible policy, while decisions in retransmission slots are based on the received feedback.

To further formalize the packet forwarding system, the system state is defined as follows:

$$s(t) = [\Delta(t), N(t)], \tag{27}$$

where, the system state is represented as $\Delta(t) = \left[\Delta_k^{j,1}(t), \cdots, \Delta_k^{j,m}(t)\right]$, illustrating the AoI of each packet within node j for every slot. Additionally, $N(t) = \left[N_k^{j,1}(t), N_k^{j,2}(t), \cdots, N_k^{j,m}(t)\right]$ details the number of retransmissions for each packet in that slot. The state spaces for decision-making and retransmissions are denoted by S_d and S_r respectively. Definitions of S_d and S_r are provided as follows:

$$S_d = \left\{s(t) \mid \forall i \in \Gamma, N_k^{j,i}(t) \in \{0, N_i\}\right\}, \tag{28}$$

$$S_r = \left\{s(t) \mid \exists i \in \Gamma, 0 < N_k^{j,i}(t) < N_i\right\}. \tag{29}$$

Thus, the complete state space S is defined as follows:

$$S = S_d + S_r, \tag{30}$$

where $N_k^{j,i}(t) \in \{0, N_i\}$ is a positive integer.

$D(t)$ denotes the complete decision space. $D_d(t)$ is the decision space for any state within S_d, and $D_r(t)$ is the retransmission space for any state within S_r. The complete decision space is described by:

$$D(t) = \{D_d(t) \mid s(t) \in S_d\} \cup \{D_r(t) \mid s(t) \in S_r\}. \tag{31}$$

For each feasible decision within the space $D(t)$, the following decision rules apply:

$$d(t) = \begin{cases} w_\pi(s(t)), & s(t) \in S_d \\ d(t-1), & s(t) \in S_r, \end{cases} \tag{32}$$

where $d(t)$ represents the scheduling decision for node j in each slot t, and $w_\pi(s(t))$ is the decision function under any feasible policy π, uniquely determining the scheduling decision for any state within S_d. On the other hand, if the system is in a retransmission state, node j will retransmit the packet scheduled in the previous slot.

Algorithm 2 AoI-driven Packet Forwarding Policy with Feedback Retransmission

```
for k = 0, 1, ..., K do
    for t = 0, 1, ..., T − 1 do
        if ρ_h(k, t) = 1 then
            Schedule the packet with the largest value of
            (Δ_k^{j,i}(0) + 2(t+1)) Δ_k^{j,i}(0) p_i that has not been scheduled;
        else
            Retransmit the current packet;
        end if
    end for
end for
```

If packet i in node j is transmitted during slot t in state $s(t)$, then let $r(s(t), d(t) = i)$ be the direct cost function. Following the description in [29], we construct this cost function using a slot-based Lyapunov drift, which is defined as:

$$r(s(t), d(t) = i) = \frac{1}{M}\left[\sum_{i\in\Gamma} \Delta_k^{j,i}(t+1)^2 - \sum_{i\in\Gamma} \Delta_k^{j,i}(t)^2\right], \tag{33}$$

where $r(s(t), d(t) = i)$ represents the one-step loss between two consecutive slots, and M represents the total number of packets in the set Γ. Further, assuming packet i is transmitted in slot t, the final form of $\mathbb{E}[r(s(t), d(t)] = i)$ is:

$$\begin{aligned} &\mathbb{E}[r(s(t), d(t)] = i) = \\ &-\frac{1}{M}\left(\Delta_k^{j,i}(0) + 2(t+1)\right)\Delta_k^{j,i}(0)\, p_i + C(t), \end{aligned} \tag{34}$$

with $C(t) = \frac{2}{M}\sum_{i\in\Gamma} \Delta_k^{j,i}(t) + 1$.

Leveraging the discussions above, we propose a packet-forwarding strategy within the AoI-OptiIoBNT framework. We employ the indicator function $\rho_h(k, t)$ to indicate the transmission state of a node at slot (k, t). If the node successfully transmitted a packet in the previous slot or has exhausted the allocated slots, then $\rho_h(k, t) = 1$; otherwise, $\rho_h(k, t) = 0$. Algorithm 2 demonstrates the execution process of the feedback-enhanced packet forwarding strategy. Subsequently, the AoI-OptiIoBNT framework is integrated with the routing algorithm in Section III to obtain Algorithm 3.

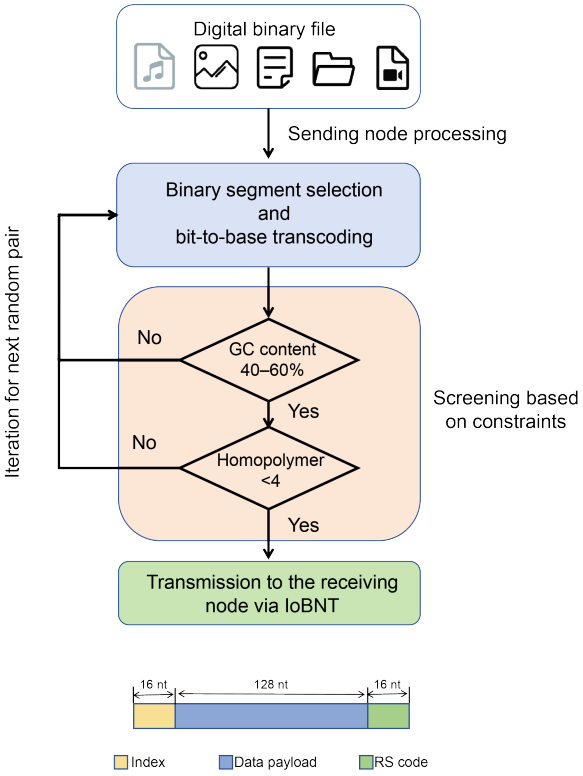


Fig. 3: The comprehensive process diagram of the IOBNT system utilizing YYC coding mechanism.

IV. DNA Coding Scheme and Network Performance Analysis

A. Yin-Yang Coding Mechanism

Over the past decade, advancements in gene synthesis and sequencing technologies have underscored the potential of DNA as an ideal messenger molecule for information transmission. Its key advantages include stability, longevity, high information density, large-scale parallelism, and biocompatibility, positioning DNA as one of the leading choices for information carriers in IoBNT systems [5], [32]. However, the current limitations of DNA synthesis technology make the de novo synthesis of long DNA sequences a significant challenge. To transmit large volumes of information in IoBNT systems, long DNA sequences that encode binary data must be fragmented into smaller pieces (150–200 bases) [33]. In the previous section, we proposed a network congestion model, which calculates the average number of hoppers transmitted by the link i under various levels of congestion. Excessive transmission of hoppers over the same link can cause congestion, potentially paralyzing local networks. As a result, a

Algorithm 3 AoI-OptiIoBNT Strategy Algorithm

Initialize state transition matrices P and reward matrices C
Define number of cycles M and time intervals T
function UPDATEMODEL($node$, $data$)
    Update P^j and C^j using $data$
    return updated P^j and C^j
end function
function COMPUTEOPTIMALPOLICY(P, C)
    Initialize policy g
    repeat
        Apply the Policy Iteration (PI) algorithm to compute optimal policy minimizing long-run reward ζ^g
        if policy converges to g **then**
            return g
        end if
    until convergence
end function
while network is running **do**
    Collect network data from the last M cycles
    for each node j **do**
        P^j, C^j = UPDATEMODEL(j, collected data)
        $optimalPolicy$ = COMPUTEOPTIMALPOLICY(P^j, C^j)
    end for
    for each node j **do**
        $currentState$ = get current state of node j
        for each time interval T **do**
            $action = optimalPolicy[currentState]$
            if action is feasible **then**
                Execute $action$
                Update $currentState$
                if packet reaches downstream node $\rho_h(k,t) = 1$ **then**
                    Schedule packet with largest value of $\left(\Delta_k^{j,i}(0) + 2(T+1)\right)\Delta_k^{j,i}(0)p_i$
                else
                    Retransmit current packet
                end if
            end if
        end for
    end for
end while

robust DNA encoding method is necessary to compress the transmitted information into fewer DNA data packets while ensuring the integrity and accuracy of the data upon decoding at the receiver.

The YYC algorithm, proposed by Ping Zhi et al. in 2022, is a coding scheme designed for DNA-based data archiving [34]. It draws inspiration from the traditional Chinese philosophy of Yin and Yang, representing two opposing but complementary and interdependent forces. Fig. 2(d) details the coding rules and procedure for YYC.

The YYC algorithm merges two independent coding rules, "yin" and "yang" into a single DNA sequence, compressing two nucleotides into one. Using N_1, N_2, N_3, and N_4 to represent the nucleic acids A, T, C, G, respectively, the Yang rule generates six coding combinations. In contrast, the yin rule maps N_1/N_2 and N_3/N_4 to independent binary numbers, resulting in 256 coding combinations. The merged yin-yang rule provides 1536 (6x256) transcoding scheme combinations for encoding binary sequences, ensuring consistent nucleotide generation and independent options for the previous nucleotide.

Table I provides a comparison of various coding schemes based on DNA. To address the unique challenges of biochemical environments, YYC differs from conventional bit-level error correction benchmarks such as Repetition [36], Reed-Solomon (RS) [37], Fountain [38] and LDPC [39]. Rather than focusing solely on redundancy-based correction, YYC provides a biochemical-aware transcoding strategy that maps binary data into DNA sequences while maintaining strict constraints on GC content (40–60%) and homopolymer length ($< 5\ nt$). These constraints effectively improve the stability of DNA sequences during synthesis and sequencing, thereby suppressing the systematic noise induced by synthesis errors at the source [34]. Moreover, YYC dynamically selects from 1536 combinations of Yin-Yang encoding rules to generate biologically robust sequences, introducing encoding diversity under fixed redundancy. This design offers strong resilience against typical molecular communication errors such as insertions, deletions, and substitutions. The information density of YYC is higher than most DNA coding schemes, with only a slight difference of $0.03\ bits\ nt^{-1}$ from DNA Fountain, which falls within an acceptable range. The information density of YYC is higher than most DNA coding schemes, enabling more effective information to be stored per nucleotide.

Considering the robust information density and error correction capabilities, we adopted the YYC coding scheme for the DNA-based IoBNT system. Fig. 3 depicts the comprehensive process diagram of the system utilizing YYC encoding schemes.

B. Network Delay and Reliability Analysis Model

The transmission delay in a network is crucial for the IoBNT system, as low delay ensures the timeliness of information transmission According to [40], we define the link delay of the network as:

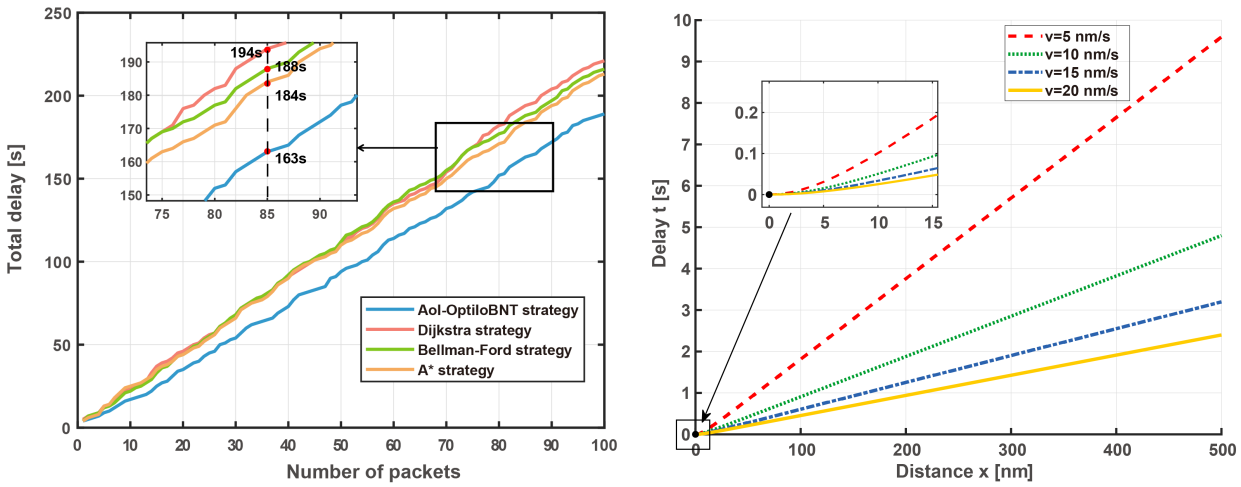
$$T(x) = \underset{t}{argmax}\left\{h(x,t)\right\} = \frac{l}{v\Lambda}\frac{\sqrt{D^2 + V^2x^2} - D}{V^2}, \quad (35)$$

where $h(x,t)$ is the impulse response function of the system, V is the drift term, D is the diffusion term, Λ is the reaction term. With $l = 1nm$ is the size of the target, that is, the scale of the receiver at the receiving end, as detailed in our previous work [12], [13].

Building on this delay model and the AoI-OptiIoBNT framework, we develop a reliability model tailored to the IoBNT network structure. This model leverages the characteristics of IoBNT discussed in the previous sections and is based on the work of Sun et al. [13], Walsh et al. [41], and Wang et al. [42].

TABLE I
COMPARISON OF DNA CODING SCHEMES

DNA Coding Schemes	**[35]**	**[36]**	**[37]**	**[38]**	**[39]**	**YYC[34]**
Error correction strategy	No	Repetition	RS	Fountain	LDPC	RS
Robustness against errors	Yes	Yes	Yes	No	Yes	Yes
Information density	1	1.58	1.78	1.98	1.24	1.95
GC content of sequences	2.5–100	22.5–82.5	12.5–100	40–60	N/A	40–60
Maximum homopolymer length	3	1	3	4	N/A	4



(a) (b)

Fig. 4: (a) Comparison of the total network delay of different routing algorithms in the case of different numbers of data packets: when the number of packets is 85, the total delay of AoI-OptiloBNT strategy is 16.3% lower than that of Dijkstra strategy, 14.1% lower than that of Bellman-Ford strategy, and 11.4% lower than that of A* strategy. (b) System delay vs. distance for different v data, where $D = 9.34\ nm^2/s$, $V = 16.49\ nm/s$, and $\Lambda = 0.05$.

We define a single link as the direct transmission from a transmitter nanomachine to a receiver without intermediate nodes. The reliability of a network link is modeled as the probability that the receiver nanomachine receives at least one data packet from the transmitter within a time frame T. This frame T is consistent with the AoI definition, where a frame consists of multiple time slots. The probability of successfully receiving a data packet in a single time slot is denoted by $H(t)$, which is the cumulative distribution function (CDF) of the inverse Fourier transform of the link response. Consequently, $(1 - H(t))$ represents the probability of failing to receive a packet in a single time slot.

Given the feedback retransmission mechanism in our system, with N_i retransmissions, the probability of all N_i retransmissions failing in a single time slot is $(1 - H(t))^{N_i}$. Therefore, the probability of successfully receiving at least one packet within a frame T is $1 - (1 - H(t))^{TN_i}$.

Thus, the reliability of the link β_{ij}, when a molecular hopper is sent from emitter node i to receiver node j can be expressed as:

$$\beta_{ij} = \left(1 - (1 - H(t))^{N_i}\right)^T. \tag{36}$$

V. SIMULATION AND RESULTS: VALIDATION OF NETWORK PERFORMANCE

TABLE II
PARAMETERS USED FOR SIMULATION

Parameters	Values
Link Length L	300 nm
Number of Nodes N	60
Interfoothold Distances Δd	0.34 nm
Hopper Speed v	35.0 μs^{-1}
Transmission Error Rate α	2 %
Backward Transmission Rate β	11.5 %
Remaining Stationary Rate m	1 %
DNA Synthesis Rate v_s	10 nts^{-1}
Target Size l	1 nm
Drift Term V	20 nm/s
Diffusion Term D	18 nm^2/s
Reaction Term Λ	0.05
Weight Coefficient μ_1	1.0
Weight Coefficient μ_2	0.5

This section of the proposed algorithm and system verification, if no specific instructions, the simulation parameters are provided by the Table II. Strategy algorithm simulation is based on Matlab R2002a, and all encoding, decoding, and error analysis experiments were run on a 8GB 7th Gen i7 central processing unit using Python 3.7.

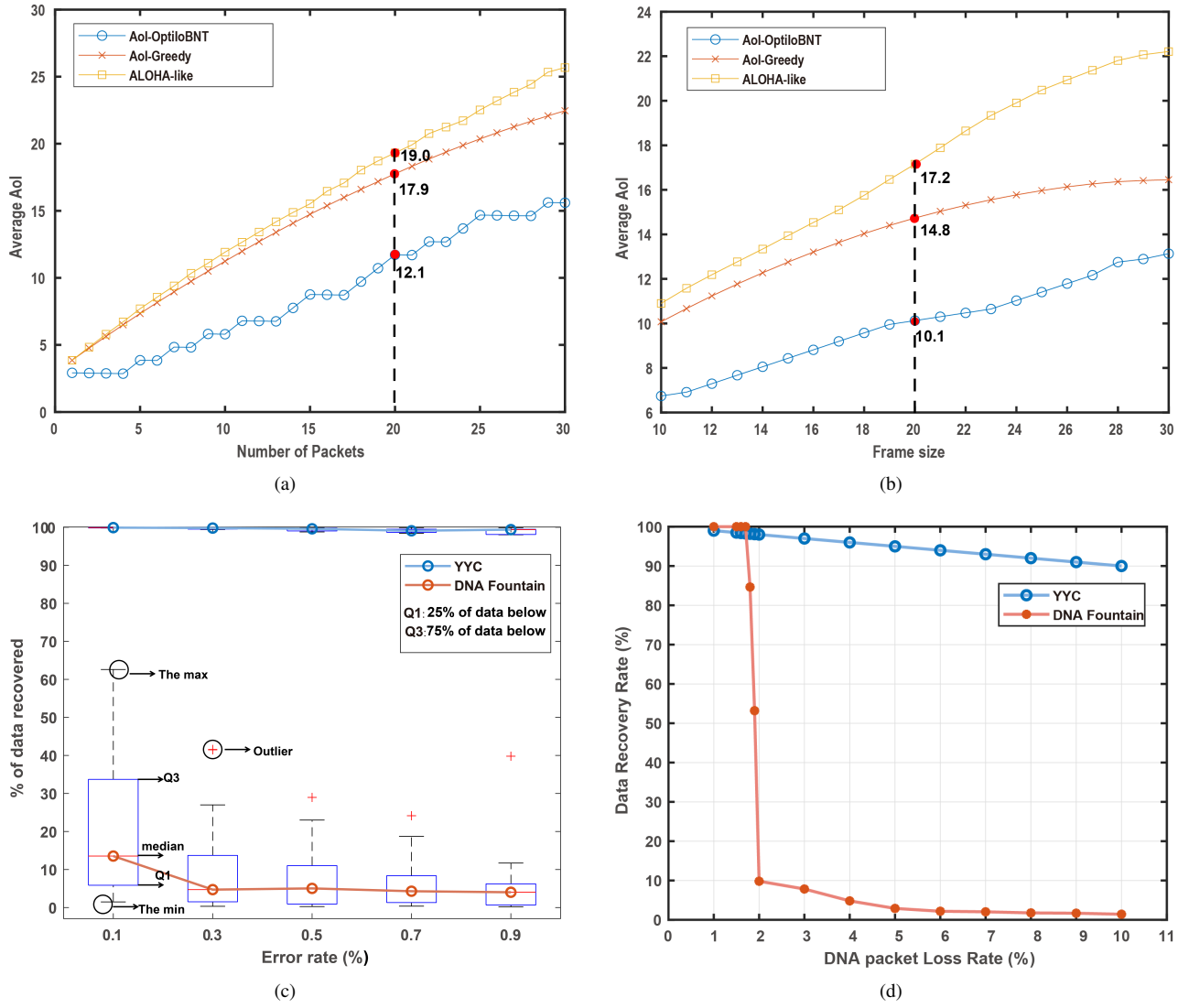
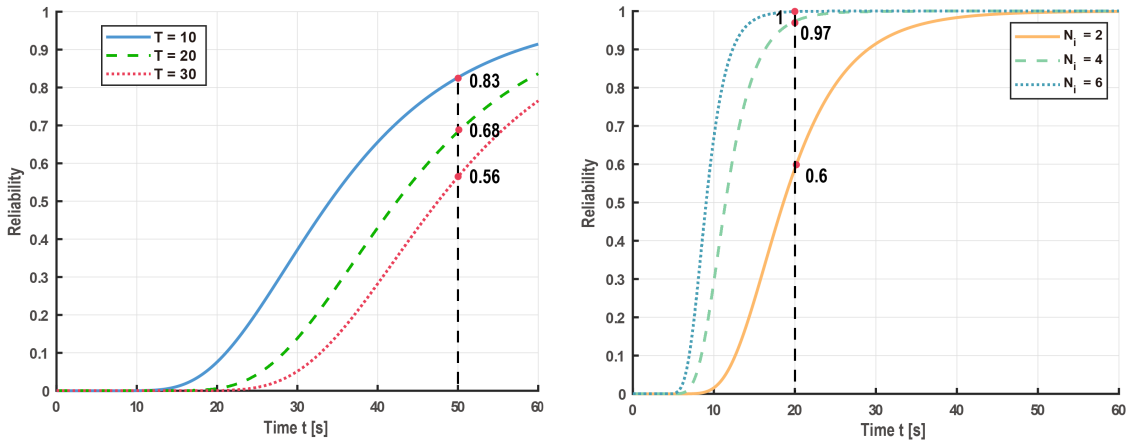


Fig. 5: (a) shows the change of the average AoI of the network with different number of packets, where $K = 1000$, $M \in \{5, 10, 15, \cdots, 30\}$, $T = 10$, $p_i = i/M$, and $N_i = 2$ for $i \in \{1, 2, 3, \cdots, M\}$. (b) shows the change of the average AoI with different frame sizes, where $K = 1000$, $T \in \{10, 12, 14, \cdots, 30\}$, $M = 10$, $p_i = i/M$, and $N_i = 2$ for $i \in \{1, 2, 3, \cdots, M\}$. (c) Robustness analysis for the YYC and DNA Fountain coding schemes. The data indicate that, at the 95% confidence level, the confidence interval for the mean of YYC is $[99.83, 99.87]$, whereas that of DNA Fountain spans $[5.12, 11.48]$. Similarly, the confidence interval for the variance is estimated as $[0.0660, 0.1090]$ for YYC and $[339.75, 560.18]$ for DNA Fountain, both at the 95% confidence level. (d) The simulation of the data recovery rate in the context of a gradient of DNA packet loss.

Fig. 4(a) illustrates a common challenge in IoBNT networks: as the number of transmitted packets increases, the total delay rises significantly. This trend is observed across all evaluated algorithms, highlighting the inherent difficulties in managing congestion in IoBNT. Notably, the proposed AoI-OptiIoBNT strategy demonstrates a marked reduction in delay compared to existing algorithms. This advantage is particularly evident at higher packet volumes, where the delay gap widens. The significant delay reduction achieved by AoI-OptiIoBNT can be attributed to its integrated congestion model and retransmission mechanism. The proposed strategy ensures efficient data transmission and minimizes delays by dynamically adjusting routing paths based on network congestion levels. This is a critical feature for real-time applications in IoBNT, such as targeted drug delivery and health monitoring. When the packet number is 85, AoI-OptiloBNT strategy of total delay is Dijkstra strategy to reduce 16.3%, than the Bellman-Ford strategy to reduce 14.1%, 11.4% less than A* strategy. This demonstrates its superior efficiency in managing network congestion and optimizing routing paths, making it a promising solution for advancing IoBNT applications.

Fig. 4(b) examines the relationship between system delay and distance in the IoBNT, revealing distinct behaviors across different scales of molecular hoppers. Specifically, two regimes are identified: small-scale (x $<$ 6 nm) and large-scale (x $>$ 6 nm). The delay exhibits a nonlinear relationship



(a) (b)

Fig. 6: (a) shows a comparison of network reliability at different frame sizes, where $D = 18\ nm^2/s$, $V = 20\ nm/s$, and $N_i = 1$. (b) shows the comparison of network reliability under different retransmission times. The parameters used in the simulation are $D = 18\ nm^2/s$, $V = 20\ nm/s$, and $T = 10$.

with distance in the small-scale regime (x $<$ 6 nm). This is attributed to the dominant influence of hopper movement velocity (v) and the characteristic length of the hopper (l) on transmission delay. At such short distances, the impact of link length on delay is minimal, resulting in relatively lower delays. However, as the distance exceeds 6 nm, the delay rapidly increases, forming a linear relationship with distance. Beyond this threshold, the delay rises rapidly, forming a linear relationship with distance. This transition at(x = 6 nm) highlights the significant impact of transmission distance on delay in the large-scale regime. These findings underscore the importance of optimizing hopper movement and link design to minimize delays, especially in applications requiring high timeliness, such as targeted drug delivery and real-time health monitoring.

Fig. 5(a) and (b) provide valuable insights into the performance of different scheduling strategies in managing the AoI within the DNA-based IoBNT system, and demonstrate the performance of AOI-Optiiobnt against benchmarks including AOI-greedy and ALOHA-like strategies. Fig. 5(a) demonstrates the impact of varying packet volumes on the average AoI, revealing that the proposed AoI-OptiIoBNT strategy significantly outperforms the AoI-Greedy and ALOHA-like strategies. When the number of data packets is 20, the average AoI of the AoI-OptiIoBNT strategy is 12.1, which is 32.4% and 36.3% lower than that of the AoI-Greedy and ALOHA-like strategies respectively. As the number of data packets increases, the average AoI for the AoI-Greedy and ALOHA-like strategies rises sharply due to reduced scheduling opportunities. In contrast, AoI-OptiIoBNT maintains a low average AoI by leveraging feedback retransmission information, which enhances scheduling efficiency.

1) **AoI-Greedy Strategy** is an AoI-based transmission strategy. It schedules the data packet with the oldest AoI at each decision time slot.
2) **ALOHA-like Strategy** is a random access strategy similar to the ALOHA protocol. It schedules data packet i with a probability of $(1/\sqrt{p_i})/\sum_{i\in\Phi(k,t)}(1/\sqrt{p_i})$.

Fig. 5(b) further evaluates the impact of frame size on AoI performance. At frame sizes of 20, the average AoI of the AOI-OptiiOBNT strategy is 10.1, which is 31.7% less than AOI-Greedy and 41.2% less than ALOHA-like. The results show that larger frame sizes negatively affect the average AoI for all strategies, as they increase the waiting time for packet updates. However, the proposed AoI-OptiIoBNT strategy consistently achieves lower AoI values across different frame sizes by jointly considering AoI and packet transmission probability p_i. This joint consideration allows for more optimized scheduling decisions than strategies relying solely on AoI or p_i. These findings highlight the importance of integrating feedback mechanisms and considering multiple factors in scheduling strategies to minimize AoI in IoBNT systems. The superior performance of the AoI-OptiIoBNT strategy underscores its potential to enhance data freshness and improve the timeliness of information in real-time applications, such as health monitoring and environmental sensing. By maintaining low AoI values, the proposed strategy ensures that the control center receives the most up-to-date information, enhancing decision-making accuracy and overall system reliability.

Fig. 5(c) and (d) offer valuable insights into the robustness and data recovery performance of two coding schemes—YYC and DNA Fountain—under varying error and packet loss conditions. Fig. 5(c) presents an analysis revealing that YYC and DNA Fountain perform well at low error rates. Each box shows the data range: the middle line is the median (typical value), the box edges mark the middle 50% of data (25th–75th percentiles), and the whiskers eQ1xtend to minimum/maximum values (excluding outliers, shown as dots). For example, YYC's 0.1% error box (99.82%–99.86% box, median 99.85%) is narrow, while DNA Fountain's 0.1% box (1.48%–8.30% box, median 3.76%) has wide whiskers and an outlier at 59.43%. However, YYC demonstrates superior

consistency and robustness as error rates increase. DNA Fountain, in contrast, shows a decline in performance with more variability and outliers, indicating less predictability and reliability. This highlights YYC's resilience under error-prone conditions.

Fig. 5(d) further evaluates the data recovery performance of both schemes under different DNA packet loss rates. The results show that YYC consistently outperforms DNA Fountain across the entire test range. While DNA Fountain's recovery rate drops rapidly with increasing packet loss, YYC maintains a relatively high recovery rate, demonstrating superior stability and reliability. This robustness is crucial for maintaining data integrity in IoBNT systems, where packet loss and errors are common. These findings underscore the importance of selecting appropriate coding schemes to ensure reliable data transmission in IoBNT applications. YYC's superior robustness and data recovery performance suggests that it is a more reliable choice for maintaining data integrity under adverse conditions.

Fig. 6(a) and (b) provide critical insights into enhancing network reliability in IoBNT systems by examining the impact of frame size and packet retransmissions. Fig. 6(a) reveals that smaller frames, which contain fewer time slots, significantly improve communication reliability. (e.g., at time $t = 50$, the network reliability of frame size $T = 10$ is 22% and 48.2% higher than that of $T = 20$ and 30, respectively.) This improvement is attributed to the reduced duration over which interference can occur, thereby minimizing transmission errors and enhancing the likelihood of successful packet delivery. This finding underscores the importance of optimizing frame size to improve the robustness of information transmission in IoBNT systems.

Fig. 6(b) further demonstrates that increasing the number of packet retransmissions significantly enhances network reliability. Additional retransmissions provide multiple opportunities for the receiver to obtain each data packet, thereby reducing the likelihood of packet loss. (e.g., at time $t = 20$, the network reliability of packet retransmissions $N_i = 6$ is 3% and 66.6% higher than that of $N_i = 4$ and 2, respectively.) This strategy is particularly effective in environments with high interference and packet loss rates, making it a valuable approach for improving the reliability of IoBNT systems. These findings suggest that combining smaller frame sizes with a robust retransmission strategy can significantly enhance the reliability of IoBNT systems.

VI. Conclusion

This study has significantly advanced the IoBNT by developing AoI-OptiIoBNT, a novel routing strategy that integrates AoI optimization with an MDP-based algorithm. Combining personalized congestion models with MDP routing, AoI-OptiIoBNT streamlines data transmission, reduces delays, and enhances network efficiency. The AoI-driven forwarding strategy ensures timely information transfer, improving system responsiveness and information freshness. Additionally, the multi-retransmission mechanism fortifies network reliability against data loss and errors. A key innovation is the implementation of YYC, which reduces BER while maintaining biocompatibility and decoding accuracy. YYC enhances the fidelity of DNA-based MC, making it more reliable for IoBNT applications. Collectively, these advancements improve IoBNT performance and expand its potential for practical applications. Beyond MC, the AoI-OptiIoBNT framework provides insights applicable to wireless sensor networks and bio-inspired IoT. AoI-driven routing and congestion-aware mechanisms can enhance network efficiency in resource-constrained environments, fostering innovation at the intersection of bio-nano and traditional networks.

Future research will focus on network coding and packet loss rate analysis to further optimize IoBNT performance. Network coding will enhance transmission efficiency and reliability among nanodevices, while detailed analysis of packet loss rates will uncover performance bottlenecks and guide the development of targeted solutions to improve network stability and reliability. These efforts will solidify IoBNT's position as a transformative technology with broad applications in healthcare, environmental monitoring, and precision agriculture.

References

[1] Akyildiz, I.F., Pierobon, M., Balasubramaniam, S., Koucheryavy, Y, "The internet of Bio-Nano things," in *IEEE Communications Magazine* **53**(3), 32–40, 2015.

[2] Y. Huang et al., "Physical-Layer Counterattack Strategies for the Internet of Bio-Nano Things with Molecular Communication," in *IEEE Internet of Things Magazine*, vol. 6, no. 2, pp. 82-87, June 2023.

[3] X. Chen, M. Wen, C. -B. Chae, L. -L. Yang, F. Ji and K. K. Igorevich, "Resource Allocation for Multiuser Molecular Communication Systems Oriented to the Internet of Medical Things," in *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15939-15952, 1 Nov.1, 2021.

[4] Malak, D., Akan, Ö.B, "Molecular communication nanonetworks inside human body. in *Nano Commun. Networks* **3**(1), 19–35, 2012.

[5] Q. Liu, K. Yang, J. Xie and Y. Sun, "DNA-Based Molecular Computing, Storage, and Communications," in *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 897–915, 2022.

[6] F. Wang et al. "Implementing digital computing with DNA-based switching circuits," in *Nature Communications*, vol. 11, no. 1, pp. 1–8, 2020.

[7] M. Dass et al. "DNA Origami-Enabled Plasmonic Sensing," in *The Journal of Physical Chemistry*, vol. 125, pp. 5969–5981, 2021.

[8] M. Ş. Kuran, H. B. Yilmaz, I. Demirkol, N. Farsad and A. Goldsmith, "A Survey on Modulation Techniques in Molecular Communication via Diffusion," in *IEEE Communications Surveys & Tutorials*, vol. 23, pp. 7–28, 2021.

[9] R. Iino et al. "Introduction: Molecular Motors," in *Chemical reviews*, vol. 120, no. 1, pp. 1–4, 2020.

[10] Y. Qing, S. A. Ionescu, G. S. Pulcu, and H. Bayley, "Directional control of a processive molecular hopper," in *Science*, vol. 361, no. 6405, pp. 908–912, 2018.

[11] Y. Chahibi, I. F. Akyildiz and I. Balasingham, "Propagation Modeling and Analysis of Molecular Motors in Molecular Communication," in *IEEE Transactions on NanoBioscience*, **15**(8), pp. 917-927, 2016.

[12] Wang, Q., Sun, Y., Cheng, W., Chen, Y., Yang, K., "Novel Interleaved Code for High-Throughput Parallel DNA-Based Molecular Communications," *IEEE Communications Letters* **27**(10), 2593–2597, 2023.

[13] Y. Sun, W. Cheng, Q. Wang, K. Yang and Y. Chen, "Advancing the Internet of Bio-Nano Things: A Novel DNA-Based Track-Hopper System for Enhanced Efficiency and Reliability," in *in IEEE Internet of Things Journal*, vol. 12, no. 4, pp. 4144-4157, 15 Feb.15, 2025.

[14] Z. Zhou, M. Liu, F. Zhang, L. Bai and W. Shen, "A data processing framework for IoT based online monitoring system," in *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Whistler, BC, Canada, 2013, pp. 686-691.

[15] P. K. D. Pramanik, A. Solanki, A. Debnath, A. Nayyar, S. El-Sappagh and K. -S. Kwak, "Advancing Modern Healthcare With Nanotechnology, Nanobiosensors, and Internet of Nano Things: Taxonomies, Applications, Architecture, and Challenges," *IEEE Access*, vol. 8, pp. 65230-65266, 2020.

[16] Goenka, S.D., Gorzynski, J.E., Shafin, K. et al, "Accelerated identification of disease-causing variants with ultra-rapid nanopore genome sequencing," in *Nat Biotechnol*, vol. 40, pp. 1035–1041, 2022.

[17] S. Ansari and K. A. Alnajjar, "Multi-Hop Genetic-Algorithm-Optimized Routing Technique in Diffusion-Based Molecular Communication," in *IEEE Access*, vol. 11, pp. 22689-22704, 2023.

[18] S. Ghasvarian, J. Abouei, M. Azadi and A. Anpalagan, "Energy Efficient Relay Selection and Routing in Diffusion-based Molecular Communication," in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, Yazd, Iran, 2019, pp. 1321-1326.

[19] İ. Kahraman, A. Köse, M. Koca and E. Anarim, "Age of Information in Internet of Things: A Survey," in *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 9896-9914, 2024.

[20] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proceedings IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 2731-2735.

[21] M. Song, H. H. Yang, H. Shan, J. Lee and T. Q. S. Quek, "Age of Information in Wireless Networks: Spatiotemporal Analysis and Locally Adaptive Power Control," *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, , pp. 3123-3136, 1 June 2023.

[22] T. Nakano, M. J. Moore, F. Wei, A. V. Vasilakos and J. Shuai, "Molecular Communication and Networking: Opportunities and Challenges," in *IEEE Transactions on NanoBioscience*, vol. 11, no. 2, pp. 135-148, 2012.

[23] S. Angerbauer, F. Enzenhofer, T. Pankratz, M. Hamidovic, A. Springer and W. Haselmayr, "Novel Nano-Scale Computing Unit for the IoBNT: Concept and Practical Considerations," in *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, vol. 10, no. 4, pp. 549-565, 2024.

[24] Pramanik, P.K., Solanki, A., Debnath, A., Nayyar, A., El-Sappagh, S., Kwak, K.S, "Advancing Modern Healthcare With Nanotechnology, Nanobiosensors, and Internet of Nano Things: Taxonomies, Applications, Architecture, and Challenges," *IEEE Access* **8**, 65230–65266, 2020.

[25] Wang, P., Mu, Z., Sun, L., Si, S., Wang, B., "Hidden Addressing Encoding for DNA Storage," *Frontiers in Bioengineering and Biotechnology* **10**, 916615, 2022.

[26] S. Angerbauer, F. Enzenhofer, T. Pankratz, M. Hamidovic, A. Springer and W. Haselmayr, "Novel Nano-Scale Computing Unit for the IoBNT: Concept and Practical Considerations," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, vol. 10, no. 4, pp. 549-565, Dec. 2024.

[27] Xu, Y., Xi, Y., Li, D., Zhou, Z., "Traffic Signal Control based on Markov Decision Process," *In: 2016 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM*, 28—30 June 2016.

[28] Hofmann, P., Cabrera, J.A., Bassoli, R., Reisslein, M., Fitzek, F.H., "Coding in Diffusion-Based Molecular Nanonetworks: A Comprehensive Survey," *IEEE Access* **11**, 16411–16465, 2023.

[29] Wang, H., Xie, X., Li, X., and Yang, J, "Scheduling Schemes for Age Optimization in IoT Systems With Limited Retransmission Times," *IEEE Internet of Things Journal* **9**(21), 21458-21468, 2022.

[30] Q. Liu, B. Zhou, W. Yu and Z. Bu, "Feedback Aggregation Based Retransmission Scheme for Multi-Hop Wireless Networks," in *2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2023, pp. 1516-1520.

[31] N. Kumar, Anuradha, N. Verma and I. Malhotra, "A technical review on application oriented comparative study of IoT, IoNT, and IoBNT," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatre, India, 2021, pp. 631-636.

[32] L. Xiang, Q. Liu, S. Chen, K. Yan, W. Wu and K. Yang, "A Tutorial on Coding Methods for DNA-Based Molecular Communications and Storage," in *IEEE Internet of Things Journal*, **11** (7), 11825-11847, 2024.

[33] Doricchi, A., et al., "Emerging Approaches to DNA Data Storage: Challenges and Prospects," in *ACS Nano* **16**(11), 17552–17571, 2022.

[34] Ping, Z., et al., "Towards practical and robust DNA-based data archiving using the yin–yang codec system," in*Nature Computational Science* **2**(4), 234–242, 2022.

[35] Church, George M et al. "Next-generation digital information storage in DNA." in *Science* **337**(6102), 1628,2012.

[36] Goldman, N., Bertone, P., Chen, S. et al. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. in *Nature* 494, 77–80, 2013.

[37] Grass, Robert N et al. "Robust chemical preservation of digital information on DNA in silica with error-correcting codes." in *Angewandte Chemie*, **54**(8), 2552-5, 2015.

[38] Erlich, Yaniv, and Dina Zielinski. "DNA Fountain enables a robust and efficient storage architecture." in *Science* **355**(6328), 950-954, 2017.

[39] Chen, Weigang et al. "An artificial chromosome for data storage." in *National science review* **8**(5), 2021.

[40] Y. Chahibi, I. Balasingham, "Channel Modeling and Analysis for Molecular Motors in Nano-scale Communications," in *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication (NANOCOM' 15)*, New York, NY, USA, 2015.

[41] F. Walsh and S. Balasubramaniam, "Reliability and Delay Analysis of Multihop Virus-Based Nanonetworks," *IEEE Transactions on Nanotechnology* **12**(5), pp. 674–684, 2013.

[42] X. Wang, Z. Wu, J. Chen, X. Hao and B. Liu, "Reliability Analysis of Molecular Communication Based on Drift Diffusion," in *2017 International Conference on Networking and Network Applications (NaNA)*, Kathmandu, Nepal, 2017, pp. 152–157.