The Role of Monitoring Frequency in Optimizing Digital Twin Networks

Jorg Wieme
imec - IDLab
Ghent University
Ghent, Belgium
jorg.wieme@ugent.be

Mathias Baert
imec - IDLab
Ghent University
Ghent, Belgium
mathias.baert@ugent.be

Jeroen Hoebeke
imec - IDLab
Ghent University
Ghent, Belgium
jeroen.hoebeke@ugent.be

***Abstract*—Digital Twin Networks have garnered increasing interest in both industry and academia for managing, monitoring, and controlling networks, irrespective of the underlying technology. However, in networks characterized by low frequency and throughput, the monitoring data is often inadequate to provide sufficient insights into network behavior on its own. In essence, establishing bi-directional communication between the Digital Twin Network and the physical network is crucial for gaining insights into the network and enabling informed reconfiguration based on those insights. This is ideally achieved without relying on out-of-band communication. Therefore, alternative paradigms must be considered. This study investigates the impact of monitoring frequency on the reliability and accuracy of a Digital Twin Network within a Bluetooth Mesh environment. Multiple frequency settings were tested across two experiments to evaluate their effect on network performance. The comparison between real network paths and Digital Twin Network simulations provides insights into its behavior. Results indicate that higher update frequencies generally improve Digital Twin Network performance by enhancing path prediction accuracy. However, this improvement is counterbalanced by a reduction in network reliability at higher frequencies. Furthermore, it was noted that while higher frequencies tend to result in more accurate paths predicted by the Digital Twin Network, they also adversely affect the accuracy of end-to-end latency predictions. These findings highlight the need for a strategic balance between requirements in Digital Twin Network deployment.**

***Index Terms*—Digital Twin Network, DTN, Monitoring, Bluetooth Mesh, Network Management, Network Reliability, Network Digital Twin**

I. Introduction

In recent years, communication protocols and technologies have rapidly evolved to meet growing digital demands. From Wi-Fi's advancement with versions like Wi-Fi 6 and upcoming Wi-Fi 7 [1]–[3], to Bluetooth's transformation with BLE for IoT devices and scalable solutions like Bluetooth Mesh and LE Audio [4]–[6], these technologies have significantly expanded their capabilities. As networks become more complex, the need for efficient management has driven the development of software-managed networking solutions, incorporating AI [7] and Digital Twin Network (DTN) [8], [9]. Moving forward, with the rise of 5G, Wi-Fi 6, and future iterations, these advanced management approaches will play a key role in ensuring seamless operation and scalability.

A Digital Twin Network or Network Digital Twin [10] is essentially a specialized type of digital twin that specifically represents and manages entire networks. Unlike traditional digital twins that focus on simulating and optimizing single entities like machines or processes, DTNs replicate and manage the collective behavior, performance, and interactions of interconnected systems within a network.

In our earlier research [11], we introduced and implemented a DTN for Bluetooth Mesh (BM). In that work, we aim to improve performance and reliability by monitoring and managing the network based on predictive what-if scenarios. This approach allows us to anticipate network behavior under different conditions, enabling proactive adjustments to optimize operations.

Typically, bi-directional communication exists between the DTN and the physical network—updating network state and configuring functions. In-band telemetry is a common method, but due to packet size limitations in BM networks, our previous work [12] proposed a self-sufficient custom monitoring paradigm.

In our previous work [12], we established that the data is statistically sufficient to accurately represent the network within the DTN. However, this method does not adequately capture the dynamic nature of BM networks, which undergo continuous changes over time. Therefore, the focus of this study:

- Investigates how the frequency of monitoring data impacts overall network quality.
- Explores the effects of transporting monitoring data as additional traffic within the same network.
- Analyzes the potential for excessive monitoring traffic to induce network congestion and lead to unintended denial of service scenarios.
- Evaluates how varying monitoring data frequencies influence the fidelity and precision of DTNs internal representations.

The paper is organized as follows: Section II provides an overview of related work. Following this, Section III offers a brief introduction to BM and a description of the DTN. Section IV presents a overview of the physical testbed used and outlines the methodology of the measurement process. Next, in Section V, we analyze the impact of monitoring frequency. Finally, Section VI concludes the paper, summarizing the findings and discussing future directions.

II. Related Works

While DTN research is still in its early stages, several studies have contributed significantly to its advancement [8], [9]. These investigations have explored various aspects such as architecture, functionality, and practical applications across different domains. Researchers have focused on enhancing operational efficiency, predictive maintenance capabilities, and decision-making processes through DTNs.

The authors in [13] introduce B5GEMINI, proposing it as a modular and scalable DTN system designed for 5G beyond networks. This system is envisioned as a platform capable of supporting the development and deployment of intelligent-driven DTN applications. The DTN framework aligns with the guidelines set forth in the IETF Draft [10], ensuring compatibility and adherence to standardized protocols and practices within the network architecture.

In the study [14], the researchers aim to develop a DTN for a Software-defined Vehicular Network (SDVN), serving as a simulation environment to evaluate the quality and reliability of Machine Learning (ML) and Deep Learning (DL) models before deployment in the physical network. Referred to as the Intelligent Digital Twin (IDT), this simulation environment enables assessment of model robustness against the dynamic changes inherent in Vehicular Networks (VNs). Additionally, the IDT facilitates model retraining as necessary based on real-time data feedback, ensuring continuous optimization and adaptation to evolving network conditions.

The paper [15] proposes a Network Digital Twin for the Industrial Internet of Things (IIoT), facilitating services like predictive maintenance and energy optimization through intelligent network management, validated via a prototype implementation.

In [16], the author introduce TwinNet, a Graph Neural Network-based Digital Twin designed to accurately estimate Quality-of-Service (QoS) metrics such as delay and jitter in complex network environments. TwinNet models the intricate interplay between queueing policies, network topology, routing configurations, and traffic matrices, enabling robust performance across diverse and unseen scenarios.

Numerous studies on In-band telemetry have been conducted across various technologies. Most of this research has concentrated on IP-based and 6TiSCH networks [17]–[19] exploring methods to embed telemetry data within standard network traffic. Despite the extensive research on DTNs, most studies, including those cited, do not address the specific constraints of BM networks, particularly the limitation of small packet sizes. However, due to the limited Maximum Transmission Unit (MTU) of BM networks, no one, to the authors' knowledge, has proposed an In-band telemetry system specifically for BM as of this writing.

III. Background on Core Technologies

A. Bluetooth Mesh

BM is a network protocol built on Bluetooth Low Energy (BLE), designed to enable many-to-many communication in

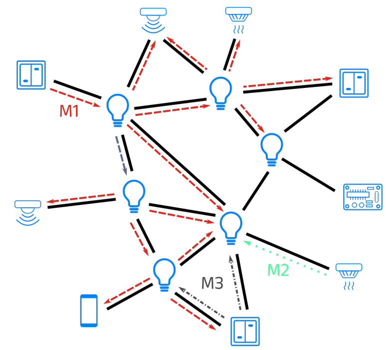


Fig. 1: Illustration of a Bluetooth Mesh network demonstrating three scenarios of the relaying process used to achieve the flooding based communication: M1) A message is relayed through multiple nodes, M2) A message terminates at a non-relaying node, and M3) A message successfully reaches its intended destination in one hop.

large-scale device networks. Operating in the 2.4 GHz ISM band, BLE divides the spectrum into 40 channels spaced 2 MHz apart, primarily using channels 37, 38, and 39 for advertising. These channels facilitate the fundamental communication process of BM, where devices known as advertisers broadcast data at regular intervals, and scanners listen on these channels during specified windows to capture the broadcast data.

The BM network follows a publish-subscribe model, where devices, or nodes, either publish data to or subscribe to specific addresses, which can be unicast, group, or virtual. Each node is assigned one or more unicast addresses upon joining the network, allowing for precise identification and targeted communication.

BM uses a flooding-based communication method, visualized in Figure 1, where nodes relay messages to ensure network-wide coverage. To address potential congestion and scalability issues, the protocol includes features like relay configuration and message caching. The Time-To-Live (TTL) parameter further limits message relays.

B. Digital Twin Network

The DTN, proposed by [11], [12], is an advanced approach to managing wireless networks, combining multiple traditional methodologies into a single, integrated system. As illustrated in Figure 2, the DTN connects theoretical models with simulations to create a dynamic, multifaceted representation of the physical network. This virtual counterpart is continuously synchronized with the physical network through real-time monitoring mechanisms embedded in each network node.

A key feature of the DTN is its selective simulation capability, which utilizes the Click modular framework [20] to emulate BM network components. Each BM node is modeled as a collection of Click elements, and the overall network topology is represented as a Click element as well. This setup allows for detailed simulation of packet exchanges and network interactions.

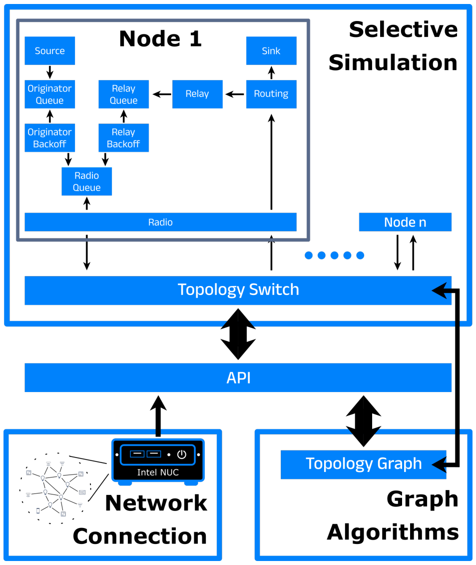


Fig. 2: Visual representation of the multi-faceted Digital Twin Network for Bluetooth Mesh

In addition to simulations, the DTN incorporates a graph-based theoretical model to represent the network structure and dynamics. This model uses algorithms to analyze and optimize the network, providing valuable insights into potential improvements.

The connection between the DTN and the physical network is facilitated through two specialized models within the BM framework, as illustrated in Figure 3. The first, the Neighbor Control Model, Figure 3a, maintains a real-time communication link by sending periodic notifications to neighboring nodes, which are used to assess and model the link quality based on received data gaps. The second, the Monitoring Model, Figure 3b, consolidates critical network information and transmits it to a central gateway. This gateway acts as a conduit between the physical network and the DTN, which can reside on the same device, in the cloud, or even on the network's provisioner. These models ensure that the DTN accurately mirrors the physical network's state and provides the necessary data for effective network management and optimization. The frequency of the DTN Monitoring Model's data collection is directly tied to the frequency of the Neighbor Control Model.

In this paper, we focus on the impact of the frequency of ”neighbor notifications”, a combination of the DTN Neighbor Control Model and the DTN Monitoring Model, on network performance and reliability. These notifications are crucial for ensuring the DTN accurately mirrors the physical network and provides essential data for management and optimization.

IV. Testbed and Measurement Setup

For our testbed setup, we utilized the IDLab OfficeLab [21], a sophisticated environment comprising 110 nodes across multiple floors designed to emulate an office setting. This testbed provides a realistic backdrop with constant Wi-Fi activity and BLE signals from personal devices. Our experiments focused on a single floor, using 15 nodes configured as regular Bluetooth devices, and one as gateway, depicted in Figure 4.

Each of these nodes is based on Intel NUC D54250WYKH systems, featuring Intel Core i5-4250U processors, 8 GB of RAM, and running Ubuntu 18.04.2 LTS. These nodes are interconnected via UTP cables and are equipped with either Nordic Semiconductor nRF52840 DK or nRF52832 DK modules running a customized implementation of the Nordic Bluetooth Mesh Stack [22]. We employed the standard configuration for the BM network, where each node is configured to relay messages once within the network.

Additionally, we employed a cloud-based server running Ubuntu 18.04.1 LTS with mosquitto version 1.4.15 as our Message Queuing Telemetry Transport (MQTT) broker. This server enables seamless communication between the network and the DTN. The DTN instances are hosted on a local Windows 11 Pro laptop, equipped with an 11th Gen Intel Core i7-1185G7 processor and 16 GB of RAM.

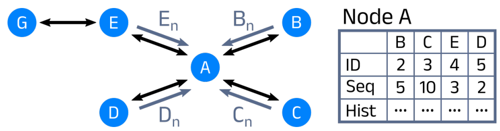
To evaluate the impact of monitoring frequency on both the reliability of the network and the accuracy of the DTN, we designed a series of experiments. Each frequency of monitoring data was tested through two distinct scenarios, each running for one hour and thirty minutes.

In the first experiment, a single application message was sent from node S1 to node D1 every 30 seconds, resulting in a total of 180 messages. For each message sent in the physical network, the DTN simultaneously performed 100 simulations, which will be used for validation against the actual network state.

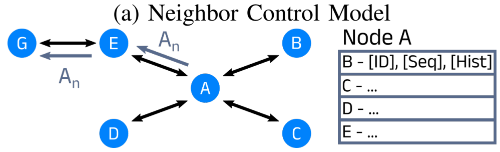
In the second experiment, the complexity increased by involving three separate applications. Messages were sent concurrently from nodes S1, S2, and S3 to nodes D1, D2, and D3, respectively. This setup also resulted in 180 messages per application, making it 540 messages in total. Given the time constraints of the simulator, each of these applications was simulated 50 times every 30 seconds in the DTN.

We tested various frequencies for monitoring data submission to the DTN: 30 seconds, 1 minute, 2 minutes, 4 minutes, 5 minutes, 10 minutes, 15 minutes, 30 minutes, 45 minutes, 1 hour, and 2 hours. Where 2 hours represent only one update at the experiment's start. Each node in the network submits its Monitoring Models to the DTN via the gateway at regular intervals. Meanwhile, Neighbor Control Model updates are evenly spaced within the interval to ensure fairness. For instance, with a 5-minute interval, a notification is sent to all neighbors every 18 seconds, aiming to send 16 messages before each report.

Several important considerations influenced the selection of parameters for the experiments. Frequencies were chosen to range from low to high, with the lowest values reflecting applications in smaller networks and higher, arbitrarily selected values aimed at evaluating performance in larger networks. While smaller incremental steps could have been tested, the experiments were conducted on a physical testbed during working hours to simulate real-life traffic, which limited the number of



(a) Neighbor Control Model



(b) Monitoring Model

Fig. 3: Representation of the physical network and the DTN through two Bluetooth Mesh models, with node G acting as gateway.

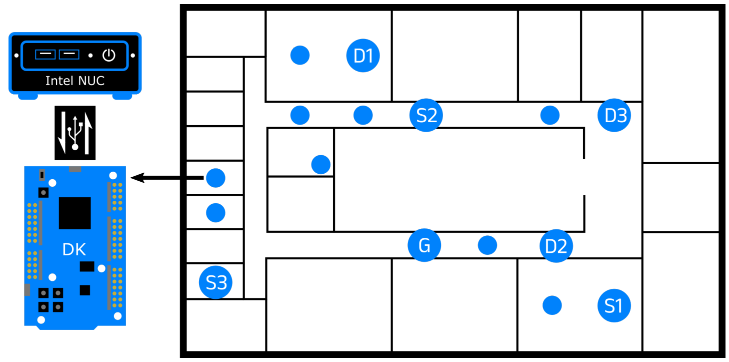


Fig. 4: Graphical representation of the physical network in an active office environment. The G node serves as the gateway, while S1 to D3 represent source and destination nodes in the experiments. The left side depicts the typical structure of each node in the network.

parameter configurations we could evaluate. Additionally, the number of simulations is set to the maximum feasible value without causing delays at the lowest frequencies. This choice is dictated by a bottleneck in the DTN implementation, where the simulator operates slow. To ensure fairness across experiments, the same maximum number of simulations was used for all intervals within the same experiment.

V. Results

In this section, we analyze our findings on network reliability and DTN accuracy. The first part, where we use the default of all nodes relaying, focuses on real network data to assess reliability metrics like message delivery and duration. The second part compares these real-world results with DTN simulations to evaluate the DTNs accuracy and predictive capability under various monitoring frequencies.

A. Impact Frequency on Reliability

To evaluate network reliability, we examined the success rate of application message delivery across various monitoring frequencies in our experiment. Table I summarizes the measured data for the first experiment for each frequency, including average message duration and standard deviation, average number of hops per message, failure occurrences, run counts indicating consecutive success or failure streaks, and unique paths utilized.

Notably, we observed a trend where lower frequencies correlated with reduced failure rates, as well as lower average message durations and standard deviations. The number of hops per message also showed similar trends, suggesting more efficient routing with less frequent monitoring intervals. Due to the increased likelihood of packet loss during relaying, the packets may reach their destination via longer paths, effectively functioning as a backup system.

Interestingly, we found no significant correlation between monitoring frequency and the number of unique paths utilized. Regardless of frequency, the network consistently utilized a comparable number of distinct paths to transmit messages, indicating a robust flooding principle.

TABLE I: Network reliability metrics for one application at various monitoring frequency.

Frequency	Duration		Hops		Failure	Run	Unique
	μ (ms)	σ	μ	σ			
30 s	82.35	47.09	2.63	0.72	54	74	40
1 m	71.15	40.26	2.58	0.67	55	78	33
2 m	62.17	31.04	2.66	0.73	17	29	39
4 m	62.69	31.30	2.48	0.63	9	15	32
5 m	56.18	28.01	2.40	0.58	4	7	34
10 m	54.43	19.81	2.49	0.58	2	3	31
15 m	53.57	21.25	2.30	0.48	3	5	26
30 m	56.79	20.66	2.58	0.64	3	5	38
45 m	54.84	18.02	2.46	0.53	0	1	23
1 h	57.33	22.69	2.44	0.59	0	1	31
2 h	51.14	21.02	2.39	0.53	0	1	31

Specifically, monitoring intervals of 30 seconds and 1 minute demonstrated the poorest performance across most metrics, with frequent alternation between success and failure noted in the run counts. The number of failures goes up to 30%. In contrast, intervals of 2 minutes exhibited notably improved reliability in terms of reduced failures and more stable message delivery patterns. In Figure 5, a similar trend is evident, where the duration of messages becomes more centralized with fewer outliers as the monitoring frequency decreases.

At the 5-minute interval, a slight impact on reliability was observed, albeit still within acceptable limits compared to lower intervals. This suggests that while longer monitoring intervals generally enhance network reliability, overly infrequent monitoring may not necessarily yield significant additional benefits beyond a certain threshold.

The results for network reliability with 3 applications reveal consistent trends observed in the single-application scenario, as listed in Table II. Application 1 demonstrated performance levels similar to its standalone operation in the previous experiment, albeit with slightly poorer outcomes. In contrast, applications 2 and 3 exhibited significantly worse overall results. Application 2 experienced up to 11 failures in optimal

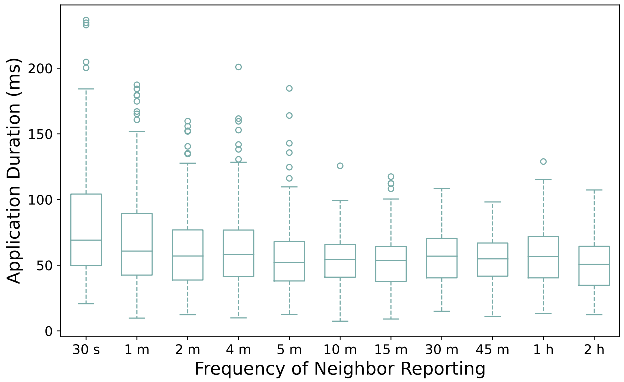


Fig. 5: Duration distribution per frequency in first experiment.

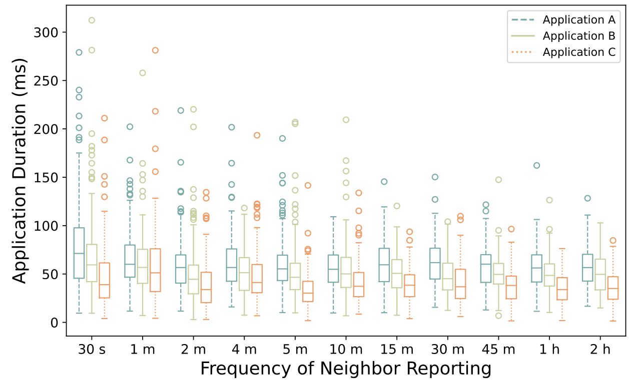


Fig. 6: Duration distribution per frequency in second experiment.

TABLE II: Network reliability metrics for three applications at various monitoring frequencies.

Application		Duration		Hops		Failure
Frequency	ID	μ (ms)	σ	μ	σ	
30 s	1	82.42	49.82	2.59	0.69	57
	2	70.13	49.17	2.56	0.61	66
	3	48.99	36.70	2.41	0.57	80
1 m	1	65.81	31.19	2.56	0.71	39
	2	61.25	33.00	2.46	0.67	41
	3	58.20	39.51	2.47	0.63	43
2 m	1	58.77	28.57	2.44	0.58	20
	2	48.92	30.73	2.36	0.56	19
	3	39.47	25.25	2.35	0.57	54
4 m	1	61.36	28.26	2.62	0.69	15
	2	51.35	22.89	2.41	0.52	22
	3	48.14	27.90	2.52	0.63	60
5 m	1	58.96	26.91	2.48	0.57	12
	2	51.55	29.81	2.30	0.54	24
	3	34.04	19.30	2.35	0.49	50
10 m	1	56.34	20.82	2.57	0.65	4
	2	53.35	27.98	2.32	0.58	9
	3	41.29	21.45	2.45	0.56	51
1 h	1	57.11	21.99	2.48	0.64	2
	2	49.64	18.14	2.36	0.59	18
	3	35.31	16.24	2.34	0.52	33

conditions, while application 3 saw a best-case scenario with 33 failures, equating to an 18% failure rate. These findings indicate potential issues such as insufficient reliable neighbors surrounding certain nodes, likely stemming from isolation of the source node.

Similar to the single-application case, message duration exhibited comparable trends across different frequencies. Notably, both Figure 6 and Table II illustrate that message durations for applications 2 and 3 showed greater volatility with changing frequencies. Moreover, there was a notable increase in variability between the 1-minute and 30-second frequencies compared to the previous experiment. These observations are attributed to the higher failure rates observed, reaching up to 44% under the most unfavorable conditions, making the results more susceptible to outliers

Based on all the measurements gathered from both experiments, it is evident that higher frequencies exert a significant impact on the frequency of failures. Moreover, they introduce variability in message duration, thereby reducing its predictability. However, as we transition to frequencies of 4 to 5 minutes, we observe a notable improvement in network reliability. This observation holds true despite the network being entirely unconfigured, highlighting promising results at both the application and monitoring model levels.

B. *Impact Frequency on DTN Accuracy*

1) Path accuracy: For DTN accuracy, we assessed how often the most frequently simulated paths in our experiments matched the actual paths used by the network. In the first experiment, we conducted 100 simulations simultaneously with sending an application message in the network, and in the second experiment, we did 50 simulations per application. Figure 7 illustrates the proportion of the top 3 most simulated paths and other paths utilized by the network, including instances where a path was used that wasn't simulated at all.

The data reveals that the top 3 paths became more dominant as the monitoring frequency decreased. This trend corresponds with our earlier observation: as reliability decreases with lower frequencies, the DTN becomes less aware of dynamic changes in the network. Interestingly, the occurrence of unsimulated paths was lower with higher frequencies, suggesting that more frequent updates lead to a more accurate reflection of the network's state. Therefore, in the current version, it is intriguing to explore the optimal balance between two extremes. However, by adapting the DTN, we could achieve a blend of both approaches rather than settling for a middle ground.

In summary, higher monitoring frequencies allow the DTN to better capture network dynamics, reducing the chances of the network using unsimulated paths. However, as frequency decreases, the DTNs ability to adapt to short-term disturbances diminishes, suggesting a balance is needed between update frequency and network reliability to maintain accurate and efficient monitoring. Another approach to reduce traffic towards the DTN could involve implementing an event-based system. In this upgraded version, nodes would possess enhanced intelligence to detect specific observations. Instead of

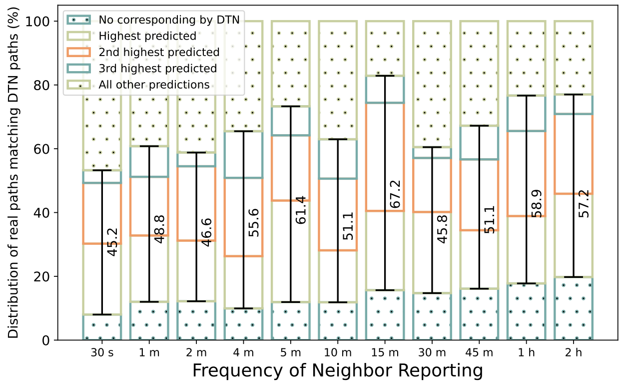


Fig. 7: Path popularity distribution across different monitoring frequencies for all experiments combined. The black interval represents the paths utilized by the network.

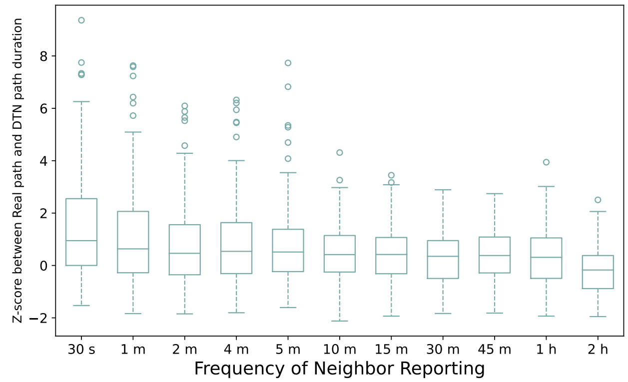


Fig. 8: Z-score distribution of path durations, between the DTN prediction and the real path, first experiment.

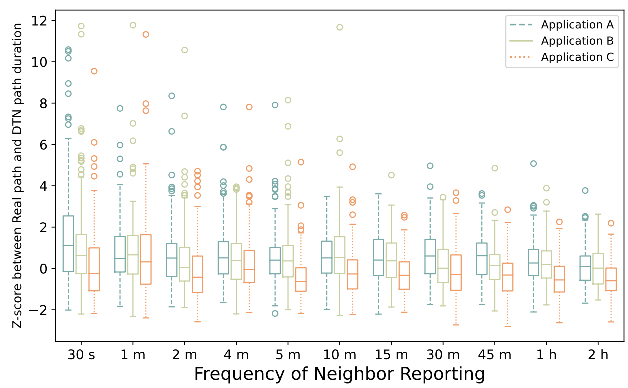


Fig. 9: Z-score distribution of path durations, between the DTN prediction and the real path, second experiment.

continuously transmitting data, they would selectively send relevant events to the DTN, ensuring that only significant information is communicated.

2) Duration accuracy: we analyzed the z-score distribution of path durations between the real network and DTN simulations for both experiments, shown in Figure 9. This figure was achieved by comparing the actual durations of successful paths with the DTN predicted durations. The z-score indicates how much the actual duration deviates from the mean duration predicted by the DTN for the same path.

Our observations, for both experiments, reveal a trend consistent with the reliability results in Figures 5 and 6, indicating that while the DTN accurately predicts the paths used by the network, as previously observed, it often misestimates their durations at higher monitoring frequencies. This inaccuracy is due to increased network traffic at higher frequencies, leading to larger queuing overhead. Bluetooth Mesh networks, when using Nordic SDk, utilize a random backoff for relays, causing additional delays that the DTN doesn't currently account for, as it is not aware of the extra traffic.

To improve accuracy, the DTN could incorporate the neighboring process and traffic dynamics into its simulations. This enhancement would allow the DTN to better predict path durations by considering the additional delays and randomness from network traffic and backoff times.

VI. Conclusions

It can be observed that higher update frequencies enhance the DTNs ability to predict network paths accurately. However, they also introduce greater network congestion, leading to longer message relay times and increased delivery failures.

Interestingly, while more frequent updates improve path accuracy, they result in less precise duration predictions. This discrepancy arises because the added network traffic from frequent updates and its associated queueing delays are not incorporated by the DTN, causing it to predict shorter durations than observed in reality.

Our findings suggest that the optimal frequency of updates depends on the network's goals and constraints. In low-power, stable environments, less frequent updates can still offer reliable predictions and preserve network reliability. In contrast, environments requiring real-time precision and high accuracy may need more frequent updates, despite the associated reliability challenges.

In future work, our focus lies on exploring two complementary approaches that, when combined, promise to yield optimal results for managing and enhancing the functionality of the DTN. We propose refining the DTN reporting strategy by using an adaptable frequency scheme that adjusts based on reliability issues. An event-based approach allows nodes to autonomously report significant changes using rule-based or advanced methods. For load monitoring, we can extract queue information from nodes or use transmission frequency to approximate conditions. A hybrid approach, triggered by critical events, optimizes resource use while maintaining essential data. Finally, monitoring data can be acquired using Bluetooth mesh's always-on radios, leveraging application flow traffic. In low-traffic networks, nodes may revert to TTL 1 messages when needed.

REFERENCES

[1] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of things (iot) communication protocols: Review," in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685–690.

[2] G. Baldini, S. Karanasios, D. Allen, and F. Vergari, "Survey of wireless communication technologies for public safety," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 619–641, 2014.

[3] P. P. Parikh, M. G. Kanabar, and T. S. Sidhu, "Opportunities and challenges of wireless communication technologies for smart grid applications," in *IEEE PES General Meeting*, 2010, pp. 1–7.

[4] Bluetooth SIG, "Bluetooth specification version 4.0," https://www.bluetooth.com/specifications/specs/core-specification-4-0/, 2010.

[5] ——, "Bluetooth specification version 5.0," https://www.bluetooth.com/specifications/specs/core-specification-5-0/, 2016.

[6] ——, "Mesh profile specification: 1.0.1," https://www.bluetooth.com/specifications/specs/mesh-profile-1-0-1/, 2019.

[7] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[8] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[9] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong, V. Ram, S. Xiao, X. Shi, X. Cheng, A. Cabellos-Aparicio, and P. Barlet-Ros, "Digital twin network: Opportunities and challenges," 2022.

[10] C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, and C. Jacquenet, "Digital twin network: Concepts and reference architecture³," https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/, 2024.

[11] J. Wieme, M. Baert, and J. Hoebeke, "Managing a qos-enabled bluetooth mesh network using a digital twin network: An experimental evaluation," *Internet of Things*, vol. 25, p. 101023, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660523003463

[12] M. Baert, E. De Poorter, and J. Hoebeke, "A digital communication twin for performance prediction and management of bluetooth mesh networks," in *Proceedings of the 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, ser. Q2SWinet '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1–10. [Online]. Available: https://doi.org/10.1145/3479242.3487327

[13] A. Mozo, A. Karamchandani, M. Sanz, J. I. Moreno, and A. Pastor, "B5gemini: Digital twin network for 5g and beyond," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, pp. 1–6.

[14] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Network*, vol. 34, no. 5, pp. 178–184, 2020.

[15] M. Kherbache, M. Maimour, and E. Rondeau, "Network digital twin for the industrial internet of things," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2022, pp. 573–578.

[16] M. Ferriol-Galmés, J. Suárez-Varela, J. Paillissé, X. Shi, S. Xiao, X. Cheng, P. Barlet-Ros, and A. Cabellos-Aparicio, "Building a digital twin for network optimization using graph neural networks," *Computer Networks*, vol. 217, p. 109329, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622003681

[17] L. Tan, W. Su, W. Zhang, J. Lv, Z. Zhang, J. Miao, X. Liu, and N. Li, "In-band network telemetry: A survey," *Computer Networks*, vol. 186, p. 107763, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620313396

[18] R. Ben Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, "Pint: Probabilistic in-band network telemetry," in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 662–680. [Online]. Available: https://doi.org/10.1145/3387514.3405894

[19] A. Karaagac, E. De Poorter, and J. Hoebeke, "In-band network telemetry in industrial wireless sensor networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 517–531, 2020.

[20] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, p. 263–297, aug 2000. [Online]. Available: https://doi.org/10.1145/354871.354874

[21] imec IDLab UGent, "Testbed officelab," https://boss.wilab1.ilabt.iminds.be/inventory/?viewMode=inventory, 2024.

[22] Nordic Semiconductors, "nrf5 sdk for mesh," https://www.nordicsemi.com/Products/Development-software/nrf5-sdk-for-mesh, 2024.