A Thesis

On

Design and Implementation of Delay Tolerant Techniques in Internet of Things

Submitted in fulfilment of the requirements for the award of the degree of

Doctor of Philosophy
by
ANAMIKA CHAUHAN
(2K14/PhD/IT/05)

Under the supervision of

Prof. Kapil Sharma

Department of IT, DTU, Delhi

Department of Information Technology

Delhi Technological University

Delhi, India

2024

CANDIDATE DECLARATION

I hereby declare that the thesis entitled “Design and Implementation of Delay Tolerant Techniques in Internet of Things” submitted to Delhi Technological University, Delhi, in the partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy in the Department of Information Technology, is an original work and has been done by myself under the supervision of Prof. Kapil Sharma (Supervisor), Department of Information Technology, Delhi Technological University, Delhi, India.

The interpretations presented are based on my study and understanding of the original texts. The work reported here has not been submitted to any other institute for the award of any other degree.

Anamika Chauhan
Roll No. 2K14/PhD/IT/05
Department of Information Technology
Delhi Technological University
Delhi-110042, India

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
(Govt. of National Capital Territory of Delhi)
Shahbad Daulatpur, Main Bawana Road,
Delhi-110042, India

Date:_______________

CERTIFICATE

This is to certify that the work incorporated in the thesis entitled “Design and Implementation of Delay Tolerant Techniques in Internet of Things” submitted by Ms. Anamika Chauhan (Roll No. 2K14/PhD/IT/05) in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy, to the Delhi Technological University, Delhi, India is carried out by the candidate under my supervision and guidance at the Department of Information Technology, Delhi Technological University, Delhi, India.

The results embodied in this thesis have not been presented to any other University or Institute for the award of any degree or diploma.

Prof. Kapil Sharma
Department of Information Technology
Delhi Technological University
Delhi-110042, India

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my supervisor Prof. Kapil Sharma, for his invaluable guidance, support, and encouragement throughout my doctoral studies. His expertise in the field and dedication to teaching has been instrumental in shaping my research and fostering my growth as a scholar. I am deeply grateful for his insightful feedback and contributions to my research. His thoughtful critiques and suggestions have challenged me to think more deeply and critically about my work.

I would also like to express my sincere thanks to. Prof. Dinesh Kumar Vishwakarma, HoD, Information Technology. It is indeed my privilege to submit this thesis during his headship.

I would like to thank my colleagues with whom I have had the pleasure of working with throughout my time at DTU. Their support, encouragement, and camaraderie have made my experience in the doctoral program truly memorable.

I am indebted to my family and friends for their unwavering love and support, even during the most challenging times. My parents and brother have always been my pillars of strength and it is their constant support that has helped me to reach this stage in life. I am grateful to have two lovely daughters who have given all their love to me and motivated me to complete what I started. Without the collective support of these individuals, this research would not have been possible.

Finally, above all I thank the Almighty in giving me the strength to achieve this milestone.

ABSTRACT

As the internet of things (IoT) continues to pervade its way into all areas of real-life applications a simultaneous requirement for supporting infrastructure also appears. The newer and more complex applications, that continue to be designed, require a network and communication system that is consistent and dependable. Another factor to be considered is, the pace of digitisation spread is much faster than what growth in physical infrastructure could keep up with. This manifests differently but simultaneously in both sparsely populated rural as well as densely populated urban areas. There is a requirement for alternative solutions that would help the consistent spread of IoT-based services even in areas that might lack full-fledged technical infrastructure support. Delay Tolerant Networking (DTN) can fill this gap by providing alternative as well as hybrid solutions for achieving this goal. Delay-tolerant solutions are designed for challenged or infrastructure-lacking environments. Thus, they can aid in the expansion of IoT services in unstable environments.

In most IoT applications data acquisition is performed via sensors that are resource constrained, so the major challenge is processing a large amount of real-time, even multimedia, data from different types of sensors and maintaining reliable communication. The main requirement is that the network makes optimal usage of network resources as, well as has an assured Quality of Service. The major concerns in these sensor-based IoT networks are that permanent connections cannot be set up with the narrow spectrum available; also, the limited processing ability and memory cannot maintain consistent state information per connection. The mobility of nodes adds a further layer of complexity. Research shows that Delay Tolerant solutions can improve overall network performance Furthermore; IoT applications that require multicast services would have even more suitable solutions as DTNs have excellent performance for multicast data dissemination to large groups of heterogeneous nodes.

Most solutions for DTN-enabled IoT centre on decentralising the routing process and replacing continuous connectivity with "Opportunistic" connectivity. Opportunistic behaviour means, that the neighbours and time of data transmission be decided in an

opportune way i.e. when a chance meeting occurs. The success of such decision algorithms depends on the availability and accuracy of apriori knowledge (predictions), which is sometimes not available but research has shown that considering the resources and stability required for absolute optimum requires an unrealistic amount of computation, thus hybrid algorithms are proposed that are much more practical. The most prevalent algorithms fall under the class of the store-carry-forward mechanism. The nodes (stationary or mobile) are divided into different clusters and carry and forward messages across the network to deliver them as destined.

For achieving the mentioned objectives, this study utilizes modifying DTN protocols for handling intermittent connectivity in networks where there is no immediate path from source to destination. Optimizing transmission strategies and reducing redundant transmissions, to minimize the energy consumption of IoT devices. Leverage buffering techniques to store data at intermediate nodes until a forwarding opportunity arises. These approaches have tremendous applicability to solving the problems of actual value. The following strategies are used to achieve the targeted objectives:

For achieving the first objective, an extensive investigation is performed to provide a comprehensive survey of DTN routing solutions tailored for IoT applications. The goal is to identify, analyse, and categorize these protocols based on their design principles, performance metrics, and applicability to IoT use cases.

In the second objective, a DTN routing protocols, Spray and Wait routing protocol, is explored and it presents a viable solution for ensuring reliable communication by utilizing a store-carry-forward mechanism. Spray and Wait is an efficient routing protocol designed to address the challenges posed by delay and disruption in networks where continuous paths between source and destination are unavailable.

For the third objective, after performing a thorough literature survey and attacks of DTN and IoT it was observed Both DTNs and IoT environments face critical security challenges, particularly in various intrusion and attacks, such as denial of service, routing attacks, and unauthorized access. This objective explores Anomaly detection mechanism based on energy aware routing which is effective in DTNs, and implements an Intrusion detection engine for Denial of service, version and rand attacks

For the final objective, after performing the extensive survey it was observed that Intrusion detection in Delay Tolerant Networks and IoT presents unique challenges .Traditional IDS approaches must be adapted or redesigned to handle these environments effectively. By leveraging hybrid detection systems, machine learning, and trust-based mechanisms, intrusion detection can be made more robust in these challenging environments.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Abbreviations

PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
ONE	Opportunistic Network Environment
MQTT	Message Queuing Telemetry Transport:
6LowPAN	IPv6 over Low-power Wireless Personal Area Networks
CoAP	Constrained Application Protocol
ON	Opportunistic Networks
ICNs	Intermittently Connected Networks
MANETs	Mobile Ad-Hoc networks
RF	radio frequency
UWB	ultra-wideband
UAVs	Unmanned Aerial Vehicles
V-IoT	Vehicular IoT
SCF	store-carry-forward
AODV	Ad Hoc On-Demand Distance Vector,
BDP	Bandwidth-Delay Product
RPL	Routing Protocol for Low-Power and Lossy Networks
DIS	DODAG Information Solicitation
DoS	Denial of Service
DODAG	Destination Oriented Directed Acyclic Graph

1 INTRODUCTION

This chapter introduces the concept that features of DTN mechanisms and solutions that make them well-suited and adaptable to address the dynamic nature of IoT communications. The objectives of the research work are highlighted. The chapter concludes with chapter wise summary of the thesis.

1.1 Background of the study

As the Internet of Things (IoT) continues to infiltrate various real-life applications, the need for a supporting infrastructure becomes increasingly apparent. The emergence of more intricate and advanced applications necessitates a network and communication system that is both reliable and consistent. Compounding this challenge is the rapid spread of digitization, which often outpaces the growth of physical infrastructure. This issue presents itself in both sparsely populated rural areas and densely populated urban centres. To enable the widespread adoption of IoT-based services, particularly in areas lacking comprehensive technical infrastructure, alternative solutions are required. Delay Tolerant Networking (DTN) can bridge this gap by offering alternative and hybrid solutions to address this challenge. DTN solutions are purpose-built for environments with limited or unreliable infrastructure, making them valuable for extending IoT services in such conditions.

In most IoT applications, data is collected through resource-constrained sensors, posing a significant challenge in processing a vast amount of real-time, and often multimedia, data from diverse sensor types while maintaining dependable communication. The critical requirement is that the network optimally utilizes its resources while ensuring a consistent Quality of Service. Sensor-based IoT networks face substantial concerns, such as the inability to establish permanent connections due to limited available bandwidth and constraints in processing power and memory, making it challenging to maintain continuous state information for each connection. The mobility of nodes adds an additional layer of complexity. Research indicates that Delay Tolerant solutions can enhance the overall network performance. Moreover, IoT applications that demand multicast services could benefit from DTNs, as they demonstrate excellent performance in disseminating multicast data to large groups of heterogeneous nodes.

Delay-Tolerant Network (architecture is designed to operate in environments where a reliable end-to-end network connection is unavailable. With the proliferation of the Internet of Things, DTN principles have been increasingly applied to support IoT deployments in challenging network conditions. DTNs are particularly suited for IoT applications where connectivity is intermittent, latency is high, or network infrastructure is sparse. The integration of DTNs into IoT allows data to be "stored, carried, and forwarded" when connectivity becomes available, making them ideal for remote monitoring, urban IoT systems, and emergency communication scenarios.

One of the primary areas of DTN application in IoT is remote environmental monitoring, where IoT devices are deployed in remote areas with limited connectivity. For example, in wildlife tracking and environmental sensor networks, DTNs allow devices to collect and temporarily store data until connectivity is restored, ensuring data integrity despite network disruptions. In smart city applications, DTNs manage the flow of data across densely populated areas with varying network loads, reducing data loss during temporary connectivity outages. Similarly, in emergency and disaster management, DTNs enable IoT devices to maintain communication with emergency responders even when conventional network infrastructure is unavailable, playing a critical role in situations where rapid data dissemination is essential.

Various DTN protocols have been adapted to address the unique needs of IoT applications. For instance, the Bundle Protocol (BP) leverages the DTN store-carry-forward model, holding data packets in intermediate nodes until they can be forwarded, which is particularly beneficial for high-latency IoT scenarios. However, the integration of DTNs into IoT also presents several challenges, as highlighted in the literature. Scalability is a significant concern, particularly in dense IoT environments where devices are numerous and network congestion can occur. Additionally, the DTN reliance on data replication and storage can lead to increased energy consumption, an issue that IoT devices—often operating on limited power—must overcome. Security is another major consideration, as DTNs in IoT applications may be susceptible to data breaches or tampering. Research into secure DTN protocols has focused on authentication and encryption techniques to ensure data integrity during storage and transfer. Lastly, maintaining Quality of Service (QoS) in DTNs is challenging due to the variable delays associated with the store-carry-forward approach, which can impact critical IoT applications like healthcare or disaster response.

Recent advancements aim to address these challenges. Machine learning has been introduced to predict node encounters and optimize routing; improving the efficiency of DTN data transfer in IoT networks. Additionally, block-chain technology is being explored for its potential to secure data exchanges within DTNs, providing decentralized authentication and secure logging capabilities. Energy-aware DTN protocols are also being developed, allowing IoT devices to adjust transmission power and data replication strategies based on energy availability, contributing to sustainable DTN-based IoT deployments.

The use of DTNs in IoT shows considerable potential, particularly for applications in remote monitoring, smart cities, and emergency response. However, to fully realize this potential, on-going research is addressing scalability, security, and energy efficiency challenges. Emerging solutions, such as adaptive protocols, machine learning, and blockchain integration, could further enhance the robustness of DTN-based IoT networks, making them an increasingly viable solution for resilient IoT communication in challenging network environments.

DTN routing plays a significant role in the Internet of Things, particularly in scenarios where traditional networking methods may not be feasible or efficient. DTN routing can primarily be applied in IoT and its following potential benefits:

- **Integration with Emerging Technologies:** Combining DTN with technologies like edge computing and 5G to enhance connectivity and reduce latency in IoT applications.
- **Advanced Routing Protocols:** Developing new routing protocols that can better handle the specific challenges of IoT networks while maintaining the benefits of DTN.

In summary, DTN routing offers significant advantages for IoT applications, particularly in challenging or dynamic environments. By leveraging DTN’s flexibility and resilience, IoT systems can achieve more reliable data collection and transmission, even in the face of intermittent connectivity and other obstacles.

1.2 Research objectives

Upon performing a comprehensive review of DTN routing solutions tailored for IoT applications. And after analysis, and categorization of existing protocols based on their design principles, performance metrics, and applicability to IoT use cases, the following research questions were identified:

RO1: Investigate and propose a comprehensive survey of DTN routing solutions for IoT applications, their similarities and areas of convergence.
RO2: To design and implement a routing protocol based on the combination of the benefits of the DTN routing strategies.
RO3: To design and implement optimisations solution used for identification and mitigation of attacks on IoT based networks.
RO4: To perform a comparison of proposed protocols with existing DTN routing solutions for IoT.

To answer the aforementioned research questions and to understand, design and implement the solutions with convergence of DTN and IoT, so that trade-off between energy and delay, fairness, security and reduced overhead is achieved following research objectives were chalked out:

1.3 Contribution of research work

Research Contribution 1: In this objective, an extensive investigation is performed to provide a comprehensive survey of DTN routing solutions tailored for IoT applications. The goal is to identify, analyse, and categorize these protocols based on their design principles, performance metrics, and applicability to IoT use cases.

Research Contribution 2: In this objective, a DTN routing protocols, Spray and Wait routing protocol, is explored and it presents a viable solution for ensuring reliable communication by utilizing a store-carry-forward mechanism. Spray and Wait is an efficient routing protocol designed to address the challenges posed by delay and disruption in networks where continuous paths between source and destination are unavailable.

Research Contribution 3: After performing a thorough literature survey and attacks of DTN and IoT it was observed Both DTNs and IoT environments face critical security challenges, particularly in various intrusion and attacks, such as denial of service, routing attacks, and unauthorized access. This objective explores Anomaly detection mechanism based on energy aware routing which is effective in DTNs, and implements an Intrusion detection engine for Denial of service, version and rank attacks

Research Contribution 4: After performing the extensive survey it was observed that Intrusion detection in Delay Tolerant Networks and IoT presents unique challenges .Traditional Intrusion Detection System (IDS) approaches must be adapted or redesigned to handle these environments effectively. By leveraging hybrid detection systems, machine learning, and trust-based mechanisms, intrusion detection can be made more robust in these challenging environments.

1.4 Outline of the thesis

The thesis comprises of six chapters and a detailed summary of all the chapters have been summarised below:

Chapter 2: In this chapter, an exhaustive literature review has been performed to provide a comprehensive overview of existing DTN routing solutions designed for IoT applications, highlighting their similarities, differences, and areas of convergence.

Chapter 3: This chapter is dedicated to identification of similarities, differences and areas of convergence and overlap of DTN and IoT applications. This research further includes their challenges, protocols, and security aspects. Furthermore, identifying open research questions and future directions, focusing on how DTN can enhance the reliability and efficiency of IoT communications in a range of applications.

Chapter 4: This chapter discusses many routing protocols in DTN and how they differ significantly in their approach to message replication, prioritization, and handling of network disruptions. Future research should focus on addressing scalability, security, and QoS requirements to further optimize DTN routing for the diverse and growing IoT ecosystem.

Chapter 5: This chapter continues with the discussion of IoT and their integration with DTNs, ensuring secure communication while maintaining energy. Security must be adapted to the constrained environments of DTNs and IoT, where battery life and limited computational resources are key concerns. The chapter details comprehensively energy-aware intrusion and attacks in DTNs and IoT, focusing on existing approaches, challenges, and future directions for developing lightweight and energy-efficient security mechanisms.

Chapter 6: This chapter discusses the Spray and Wait (SNW) routing protocol in the context of IoT environments, emphasizing how it can support IoT applications requiring communication in intermittently connected or highly mobile scenarios. The two-phase approach of SNW significantly reduces the overhead associated with message duplication. An optimised buffer protocol OSNW is proposed and implemented to ensure a higher delivery probability. The protocol is especially suited for networks where node mobility can vary widely, and energy efficiency is critical.

Chapter 7: This chapter discusses how to ensure security in data delivery since many DTN nodes (e.g., sensors, mobile devices) are decentralised in nature. As a result, integrating security awareness into DTN routing protocols is crucial. A Residual Energy-Based DTN routing approach focuses on improving security and energy efficiency by considering the residual energy (i.e., remaining battery power) of nodes when making routing decisions is proposed. The chapter concludes with a proof of concept for improved performance of the proposed EA-RPL protocol.

Chapter 8: This chapter presents a comprehensive summary of the research work done. It includes the research summary of the work done, and the limitations of the research study identified. The chapter also presents the future aspects of the research work performed and how the study can help the future researchers in the said domain.

2 LITERATURE REVIEW

This chapter presents an overview of recently published research on explore DTN solutions in IoT applications, evaluating their effectiveness in addressing connectivity challenges while examining the opportunities and constraints associated with each technology. This aims to highlight the possibility of solutions designed for enabling delay-tolerant IoT solutions. A taxonomy of these DTN-based approaches in IoT and discuss the advantages and limitations associated with each study is also given.

2.1 Introduction

The systematic literature review is performed in two distinct parts, part one considers the routing solution using DTN based solutions for IoT, the second part explores the approaches used DTN security and their applications to IoT security. Various parameters were considered for identifying the relevant research papers to be included in the study and they were broadly identified into two areas; first being efficient routing to improve performance, the second area being secure routing. In both cases, the overlap between DTNs and IoT being at the centre of consideration. In this section, the research methodology adopted in this study has been explored. Broadly, the research methods have been based upon several attributes, depending upon the purpose of performing the study.

The Internet of Things (IoT) is a rapidly growing technology involving a network of physical devices that communicate with each other and share data over the internet. However, in IoT scenarios, especially in remote, harsh, or highly mobile environments, reliable end-to-end communication cannot always be guaranteed due to network disruptions and intermittent connectivity. This makes traditional networking protocols inadequate for IoT systems deployed in such environments. DTN offers a promising solution by leveraging a store-carry-forward mechanism to handle intermittent connectivity and ensure data delivery over delayed or disrupted links (1).

This thesis explores the integration of DTN into IoT environments, providing a detailed review of DTN solutions tailored for IoT applications, including their challenges, protocols, and security aspects. Furthermore, we identify open research questions and future directions, focusing on how DTN can enhance the reliability and efficiency of IoT communications in a range of applications such as smart agriculture, environmental monitoring, disaster recovery, and vehicular networks (2).

2.2 Reviews on the basis of identified categories

Numerous survey papers have tackled the challenges and opportunities within DTN, offering both classifications of routing protocols and discussions on design issues. This section highlights several prominent surveys that address DTN routing protocols. Tabular data provides a timeline of these surveys, describing their content in detail, with a chronological review following this timeline. Most solutions for DTN-enabled IoT centre on decentralising the routing process and replacing continuous connectivity with "Opportunistic" connectivity. Opportunistic behaviour means, that the neighbours and time of data transmission be decided in an opportune way i.e. when a chance meeting occurs. The success of such decision algorithms depends on the availability and accuracy of apriori knowledge (predictions), which is sometimes not available but research has shown that considering the resources and stability required for absolute optimum requires an unrealistic amount of computation, thus hybrid algorithms are proposed that are much more practical. The most prevalent algorithms fall under the class of the store-carry-forward mechanism. The nodes (stationary or mobile) are divided into different clusters and carry and forward messages across the network to deliver them as destined (3) (4).

The most prevalent hybrid approaches embed a DTN-based layer also known as the Bundle layer in the existing TCP/IP protocol architecture. The approaches can be further classified based on the features and placement of the bundle layer. These adaptations do not completely cover the drawbacks of IoT applications but can offer a reasonable trade-off between performance and resource utilization. The given research explores all these topics and offers a novel alternative.

The primary classes of DTN solutions are based on several classifications

2.2.1 Bundle based & Routing based Classification

Bundle protocols transports packets in form of messages or bundles together, not as individual packets. Bundling allows multiple packages to be sent in a message-oriented way on top of a disruptive network. This acts as an overlay over the existing network. This approach was derived from existing solutions for Wireless Sensor Networks (WSN). Such networks have disruptive mobility and data can be stored locally at each node till the next hop is available. The key change is to have a convergence layer on top of any transport layer protocol. This is very significant as uninterrupted interoperability is achieved, without change to the existing Physical and MAC layer and security concerns are also easily addressed. We must emphasize the efficiency of the Bundle Protocol (BP) in IoT applications and WSN). BP is specifically for DTNs, implementing a store-carry-mechanism where messages are stored locally at each node until a connection is available for them to be carried and forwarded to their destination. Common BP implementations include IBR-DTN (5), µDTN (6) , and Nano-DTN (6) A network acts as overlay to bridge the bundle layer and underlying protocols.

Routing based category of protocols addresses three major criteria for obtaining solutions, route selection, degree of replication, and also the selection of good next-hop or relay nodes so that a minimal amount of forwarding is required for routing. Another critical factor to be kept in mind is buffer management such that the overhead packet loss rate and the delay in delivery are minimized. Routing protocols typically focus on route selection and controlling message replication within the network. However, in IoT, selecting suitable relay nodes to forward messages can significantly impact routing performance. Furthermore, buffer management algorithms play a crucial role in reducing loss related delays, thereby enhancing overall routing efficiency.

2.2.2 Deterministic & Stochastic classification

Shen et al. (7) categorized Delay Tolerant Network routing protocols into two main families based on how they determine the destination: Flooding and Forwarding. In Flooding, routing is achieved through message replication, whereas Forwarding relies on knowledge of the network for targeted delivery. Expanding on the same, Khabbaz et al. (3), provided a more detailed breakdown of Forwarding-based methods.

In their classification, D'Souza et al. (8) distinguished routing protocols by the type of information used. The first category use flooding information, nodes lack network knowledge and distribute packets broadly across the network. In contrast, History-based routing leverages encounter histories between nodes to inform routing decisions. Special Devices-based routing introduces additional devices, either stationary or mobile, to facilitate communication.

These classifications generally align DTN routing into two primary categories: Deterministic approaches based on Forwarding or History and Stochastic approaches based on Flooding or Replication. Abraham et al. (9) explored unicast and broadcast routing, while Cao et al. (10) extended the taxonomy by identifying Unicasting, Multicasting, and Any-casting approaches. Within Unicasting, they further divided routing strategies into Naive Replication, Utility Forwarding, and Hybrid approaches, with a focus on Opportunistic Routing that prioritizes relay nodes based on mobility and contact probability.

Zhu et al. (11) analysed Social-based DTN routing, in this classification the positive and negative social attributes form the basis of comparison, these attributes may include community structures and centrality, which influence routing decisions. Their survey highlighted both Social and Opportunistic routing, studied in detail by the authors of (12).

DTN routing in Underwater Wireless Sensor Networks (UWSNs) was addressed in a survey by the authors of (13) , who divided protocols based on DTN concepts into Contact frequency, the three classes being Scheduled, Opportunistic and Predicted. In the first method, base stations act as schedulers and relay nodes, facilitating sensor communication. Opportunistic Contact uses relays and repeaters, while Predicted Contact relies on forecasted communication patterns.

2.2.3 Routing & Replication strategy classification

The work in (2) defined a different classification criterion for categorization. They defined categories on the basis of the topology of the network, the strategy for routing and replication semantics. Network Topology encompassed Replication-based and Forwarding-based approaches, Routing Strategy included

Social and Opportunistic routing techniques. Message replication and semantics covered Unicast, Multicast, and Any-cast routing.

To enhance the classification framework, an additional category—Routing Technique—is introduced, distinguishing protocols based on the use of auxiliary devices for packet routing. This encompasses Routing with Assistance (utilizing stationary or mobile devices for node communication) and Routing without Assistance (where nodes rely on prediction and replication without auxiliary devices).

2.2.4 Mobility, Heterogeneity & Scalability Classification

D'zousa & Jose (8) expand on previous classifications by introducing a third category, special-device-based schemes, which utilize additional devices (stationary or mobile) to facilitate communication. Their classification covers DTN schemes published through 2009 under three categories: flooding, forwarding, and special-device schemes. Abraham (9)in their survey categorize routing strategies into forwarding and replication families, discussing the benefits and drawbacks of several solutions published from 2000 to 2011. Khabbaz et al. (3) provided an extensive analysis of DTN design issues and categorize forwarding protocols from 2007 to 2010 into various opportunistic approaches including other approaches based on probability, vector, load, encounter, resource, network coding schemes. Their work also introduced co-operative routing which reduced node selfishness.

Zhang (14) categorizes uni-cast routing algorithms on the basis of mobility models. Deterministic approaches which are included are, predictable network topology, tree/space based routing and modified shortest path. Comparisons of stochastic schemes which have uncertain topology are also discussed. Uncertainty makes pre-scheduling unfeasible; therefore the approaches are based on history, model, controlled movement, coding, and epidemic/randomized flooding. Similarly, Shen et al. (7)categorised DTN routing protocols into two categories based on forwarding or flooding. In forwarding schemes, some network knowledge informs path selection, while in flooding schemes, nodes transmit multiple copies of a message to a set of recipients. This survey does not include a detailed taxonomy but does compare both approaches, highlighting each family's pros and cons.

2.2.5 Resource Conservation based Classification

The taxonomical separation of these categories actually shows a comparison between the primary constraints in the IoT environment like heterogeneity, scalability, mobility and resources constraints. It is obvious from the comparative analysis that we can make such a comparison because of multiple areas of overlap that each forms a basis for convergence.

Cao and Sun (15) survey DTN protocols from 2006 to 2010, presenting a taxonomy that classifies protocols by unicast, multicast, and anycast transmission methods, further distinguishing between naive replication, utility forwarding, and hybrid schemes. Naive replication guarantees delivery by sending multiple message copies, while utility-based approaches rely on a single copy, with relay nodes selected based on utility metrics. Zhu et al. (11) concentrated on the social properties of DTN routing, positive factors include community and centrality. Centrality may be based on closeness or degree and negative factors like selfishness are also considered. They classify and compare routing protocols based on these social characteristics. Finally, Sobin et al. (16) provide a survey that presented a classification of DTN routing protocols and data dissemination schemes. The criterion considered is message replication and network primitives with the combination of social and opportunistic characteristic with message replication and delivery delay semantics covering literature from 2012 to 2023.

It can also be observed that most work and research performed is oriented towards the purpose of optimizing delivery probability and minimizing resource consumption. Some of the solutions that are derived are also based on wireless sensor network based environment because heterogeneity and scale ability is common to all the three different types of network.

2.3 Solution Environment & Communication Technology

Several Simulation Environments and Tools are used for designing; modelling and testing of DTN based solutions for IoT network. The primary parameter for selection is the communication technology and also the type of protocol being worked upon. The list of most prevalent software tools and test-beds is as follows

- **The ONE** (Opportunistic Network Environment): A widely-used simulator for DTN scenarios, it supports routing protocol testing, buffer management, and mobility modelling. It’s customizable, with support for various routing schemes like PRoPHET and Epidemic Routing, making it ideal for DTN and IoT simulations.
- **OMNeT++** with INET Framework: OMNeT++ is a versatile network simulator that supports DTN and IoT through the INET framework and custom modules. It’s highly extensible and allows for simulations with specific IoT protocols, delay-tolerant architectures, and mixed mobility patterns.
- **ns-3** (Network Simulator 3): ns-3 is a discrete-event network simulator that supports DTN and IoT modules for testing communication protocols in IoT environments. DTN routing protocols, such as Epidemic and Spray-and-Wait, can be implemented and tested here.
- **Castalia**: Built on OMNeT++, Castalia focuses on Wireless Sensor Networks (WSN) but is also applied to IoT and DTN scenarios. It’s ideal for energy-efficient protocol simulations and environmental data collection, often essential for IoT.
- **Matlab and Simulink**: Matlab offers customizable simulations for IoT environments and DTN protocols, especially useful for algorithm testing and mathematical modelling in IoT.
- **QualNet:** A high-fidelity network simulator supporting DTN and IoT scenarios. It’s designed for complex wireless and mobile environments, including energy and mobility management, making it suitable for extensive DTN-IoT networks.
- **EmuLab and PlanetLab**: Both are large-scale network testbeds that support real-world DTN and IoT deployment tests. EmuLab offers emulated network scenarios, while PlanetLab is used to test real-world DTN-IoT applications on a distributed global scale.

- **Cooja** (part of Contiki OS): Specialized for IoT and Wireless Sensor Networks (WSN), Cooja can be configured for DTN-based scenarios and provides tools to simulate IoT deployments with various network conditions.

2.4 Summary of various Classification schemes

As discussed in the previous sections DTN based solutions for IoT can be categorized into three categories, a number of solutions based on these are proposed in the literature. Table 1 and table 2 present a summary of these solutions. Table 1 consists of a taxonomical separation of these categories and shows a comparison between the primary constraints in the IoT environment like heterogeneity, scalability, mobility and resources constraints. Table 2 presents a comparison of the hardware, communication technologies and the environment that are used for simulating and comparing DTN based IoT routing protocols. Table 3 presents the adaptation of DTN protocols for IoT. It is obvious from the comparative analysis that we can make such a comparison because of multiple areas of overlap that each forms a basis for convergence.

Table 1 Summary of DTN based solutions for IoT.

S.NO	**Title**	**Year**	**Mobility**	**Heterogeneity**	**Scalabilty**	**Resources**		
BP- Based						**Energy**	**Memory**	**Throughput**
1	(17)	2012	Yes	Yes	No	Yes	No	No
2	(18)	2014	No	No	Yes	Yes	No	Yes
3	. (19)	2015	No	No	No	No	No	No
4	. (20)	2015	Yes	Yes	No	No	yes	No
5	(21)	2016	No	No	No	No	No	No
6	(22)	2017	No	No	No	No	No	No
Routing-based								
7	(23)	2013	Yes	No	No	No	No	No
8	(24)	2013	Yes	No	No	Yes	No	No
9	(25)	2015	No	No	No	No	Yes	No
10	(26)	2017	Yes	No	No	No	No	Yes
11	(27)	2018	No	No	No	No	Yes	Yes
12	(28)	2019	No	No	Yes	No	Yes	Yes
X-DTN								
13	(12)	2013	No	No	No	No	No	No

14	(29)	2013	No	Yes	No	No	No	No
15	(30)	2014	Yes	No	No	Yes	No	No
16	(31)	2017	No	No	No	No	No	Yes

Table 2 Summary of DTN classification based of Simulation Environmnet and Communication Technology

S.NO	**Reference**	**Year**	**Solution Environment**	**Communication Technology**	**Implementation**
BP-based					
1	(17)	2012	WSN	BP, IBR-DTN, IEEE 802,15.4	Contiki OS, iMOte2
2	(18)	2014	IoT	IEEE 802.11, BLE	Andrioid, iOS, Raspberry Pi Arduino
3	(19)	2015	IoT	CoAP , BP, IBR-DTN	-
4	(20)	2015	Sensor Network with IoT	BP, □DTN	Contiki OS, C, Cooka Simulator
5	(21)	2016	IoT	IBR-DTN , MQTT	-
6	(22)	2017	IoT	IBR-DTN , MQTT, 6LowPAN	Raspberry PiZoleratia Re-Mote
Routing-based					
7	(23)	2013	IoT	Integer Linear Programming	RFID
8	(12)	2013	WSN	Greedy Algo	-
9	(32)	2015	WSN	-	SimPy
10	(26)	2017	IoT	IBR-DTN	Raspberry Pi
11	(27)	2018	WSN	Binary SnW	SUMO, OMNet++
12	(33)	2019	Smart Cities	PropHET	ONE simulator
X-DTN					
13	(29)	2013	IoT	BP, IP, PUB/SUB	-
14	(30)	2014	Smart Cities	HTTP, CoAP REST model	JSON, Open MTC
15	(31)	2017	IoT	MQTT, PUB/SUB	Raspberry Pi, Mosquito

Table 3 Evolution of DTN solutions adapted for IoT

Reference	**DTN routing protocol**	**Replication Strategy**	**Routing Strategy**	**Year**	**Adapted for IoT**
(34)	Epidemic	Unlimited	Flooding	2000	-

(14)	Prophet	Controlled	Forwarding	2003	-
(14)	Spray and Wait	Controlled	Flooding	2005	-
(7)	Maxprop	Controlled	Forwarding	2006	-
(3)	Spray and focus	Controlled	Flooding	2007	-
(3)	RAPID	Controlled	Flooding	2010	-
(35)	Prophetv2	Controlled	Forwarding	2011	-
(13)	IoB-DTN	Unlimited	Flooding	2018	Yes
(2)	Hybrid type dtn routing protocol considering storage capacity	Hybrid	Hybrid	2019	Yes
(36)	Scheduling-PROPHET	Controlled	Forwarding	2019	Yes
(19)	Multi-objective based deployment of throwboxes in delay tolerant networks for the Internet of Things environment	Hybrid	Hybrid	2020	Yes
(37)	Energy efficient emergency rescue scheme in wireless sensor networks	Controlled	Forwarding	2021	Yes
(38)	A novel communication framework between MANET and WSN in IoT based smart environment.	Controlled	Forwarding	2021	Yes
(39)	IoT enabled smart dustbin with messaging alert system.	Hybrid	Hybrid	2022	Yes
(40)	Agent driven resource scheduling in wireless sensor networks: fuzzy approach	Hybrid	Hybrid	2022	Yes
(41)	A novel scheduling algorithm development and analysis for heterogeneous IoT protocol control system to achieve SCADA optimization	Hybrid	Hybrid	2023	Yes

3 DTN & IOT INTERDEPENDENCY

This chapter presents the integration of DTN into IoT environments, providing a detail of similarities, differences and areas of convergence and overlap of DTN and IoT applications. This research further includes their challenges, protocols, and security aspects. Furthermore, identifying open research questions and future directions, focusing on how DTN can enhance the reliability and efficiency of IoT communications in a range of applications

3.1 Introduction

During the same period as IoT another class of networks the Delay/Disruption Tolerant Networks (DTN), sometimes also referred to as Opportunistic Networks (ON) was developed for providing routing in a challenged network where no stable end-to-end path is available (42). A challenged network can be defined as a network with no stable and direct end-to-end path from source to destination. This is caused by such networks being infrastructure less (43). It has frequent network disruptions and a lack of resources. The nodes are highly mobile and dynamic. DTN uses this very property of mobility of nodes to form paths opportunistically and deliver messages from one node to another node. The mobile nodes move in different clusters and carry and forward messages across the network to deliver them as destined. These networks initially had ad-hoc applications in areas such as tracking wildlife in difficult terrain using sensor networks, military, and underwater purposes, satellite networks, etc.

The reason for DTN successfully working in this application is that it overcomes the difficulty of accessing the network continuously, even in remote or dynamic environments where there is no guarantee of the availability of a complete and stable path from source to destination. The traditional infrastructure-based routing protocols quite naturally due to the nature of their design fail to deliver in such a challenging environment.

Literature analysis shows several similarities between the design issues, node/ traffic behaviour and resource constraints, and performance metrics of DTN and IoT. This has led to an array of solutions being designed with hybrid mechanisms. The DTN-enabled IoT network solutions enable smart objects to effectively communicate with more efficiency even in the presence of frequent disruptions. This also addresses the much larger issue of lifetime constraints. Recent studies have shown that DTN within the IoT framework provides the most suitable and satisfactory results. The interdependency of DTN and IoT is depicted in Fig. 1.

The Internet of Things (IoT) is a growing technology involving a network of cyber physical devices communicating and sharing data over the internet. However, in IoT scenarios, especially in remote, harsh, or highly mobile environments, reliable end-to-end communication cannot always be guaranteed due to network disruptions and intermittent connectivity. This makes traditional networking protocols inadequate for IoT systems deployed in such environments. DTN offers a promising solution by leveraging a store-carry-forward mechanism to handle intermittent connectivity and ensure data delivery over delayed or disrupted links (14) (21), such as smart agriculture, environmental monitoring, disaster recovery, and vehicular networks (38).

3.2 Delay Tolerant Networks (DTN)

Delay (or disruption) tolerant networking (DTN) offers an alternative solution for emerging wireless applications and architectures that face the limitations of the transport and routing layers in the traditional TCP/IP model. The Internet model is based on assumptions of low error rates, minimal delays, and reliable end-to-end connections between nodes. However, growing classes of "challenged networks" defies these assumptions and remain underserved by TCP/IP. These networks, often called Intermittently Connected Networks (ICNs), experience frequent, temporary disconnections, particularly in rural or remote areas lacking infrastructure.

DTNs are a class of wireless systems designed for environments with frequent and prolonged network partitions. They address scenarios characterized by intermittent connectivity, heterogeneous standards, high delays, and elevated error rates, while also coping with resource constraints like limited CPU power, memory, and bandwidth. DTN protocols aim to ensure eventual connectivity for complex applications (26) (44), such as:

- Wireless sensor networks (WSNs) for wildlife tracking, monitoring volcanic areas or underwater sensing (45).

- Mobile Ad-Hoc networks MANETs connecting remote devices using Global Positioning System.
- Exotic Media Networks (EMNs) such as satellites communication systems or Inter-Planetary Networks (IPNs) (46) .

DTNs employ a range of wireless technologies, including radio frequency (RF), ultra-wideband (UWB), free-space optical communication, and acoustic methods like sonar. The concept of DTNs traces back to NASA's Inter-Planetary Network (IPN) project in 1998, which first applied these principles to overcome communication barriers by using storage, replication, parallel forwarding, and other techniques.

Each DTN application operates under extreme conditions, often in environments unsuitable for traditional wireless networks. The DTN architecture extends network reach by enabling communication between disparate systems that use inconsistent standards or operate without a stable infrastructure. Its primary goal is to ensure message delivery in challenging network scenarios.

Challenged networks share several characteristics:

- Intermittent connectivity: With no consistent end-to-end path, TCP/IP protocols fail, requiring alternative approaches.
- Asymmetric data rates: When data asymmetry increases beyond typical Internet support, traditional protocols like TCP become inefficient.
- High error rates: High error rates require more bandwidth for correction or packet retransmission, increasing traffic.
- Unpredictable mobility patterns: Unlike fixed routes, the movement of nodes in DTNs is often random yet recurrent.
- Long or variable delays: Prolonged or fluctuating delays exceed acceptable limits for traditional Internet protocols, causing issues for applications that depend on quick acknowledgments.

Among these, delay is a critical challenge for DTNs, as intermittent connectivity severely impacts performance. In the TCP/IP model, the handshake process to establish a connection depends on low delay, with typical Internet delays measured in milliseconds. As delays approach a TCP timeout threshold, establishing or maintaining a connection becomes increasingly difficult.

3.3 DTN & IOT similarities and convergence

Delay Tolerant Networks (DTNs) and the Internet of Things (IoT) share several similarities, particularly in their approach to handling connectivity challenges and their underlying communication principles. Here’s a comparison highlighting their similarities:

3.3.1 Handling Intermittent Connectivity

- **DTN:** Designed to operate in environments with intermittent connectivity. DTNs use a Store-Carry-Forward approach, where data is temporarily stored at intermediate nodes until a suitable path to the destination becomes available.
- **IoT:** Often involves devices deployed in environments where connectivity can be unreliable or sporadic, such as remote areas, industrial sites, or disaster zones. IoT networks may also benefit from a similar approach, where data is buffered and forwarded when connectivity allows.

3.3.2 Adaptive Communication

- **DTN:** Adapts to varying network conditions by using different routing strategies based on the available knowledge of node mobility and network partitioning.
- **IoT:** Requires adaptive communication protocols to handle diverse and changing network topologies, as well as varying connectivity patterns. IoT systems need to adapt to dynamic conditions and may use techniques similar to those in DTNs for efficient data transfer.

3.3.3 Buffering and Storage

- **DTN:** Relies on buffering data at intermediate nodes to cope with delays and intermittent connectivity. Nodes store data until they can forward it to the next node or the destination.
- **IoT:** IoT devices often need to store data locally when connectivity is not available or when communication with a central server is delayed. Buffering and local storage are crucial for ensuring data integrity and timely transmission once connectivity is restored.

3.3.4 Data Forwarding

- **DTN:** Uses various routing schemes to forward data through the network, including deterministic, stochastic, and hybrid methods, depending on the knowledge of network conditions.
- **IoT:** Requires efficient data forwarding strategies to manage the transmission of data between devices and central systems. Depending on the IoT network's characteristics, forwarding strategies may need to handle unpredictable connectivity and data delivery challenges.

3.3.5 Addressing Challenges in Harsh Environments

- **DTN:** Particularly useful in harsh environments with challenging conditions, such as space missions or remote areas, where traditional network solutions are insufficient.
- **IoT:** Often deployed in similar harsh environments, including remote locations, industrial sites, and disaster areas. DTN-like strategies can be applied to ensure reliable communication in these settings.

3.3.6 Utilization of Opportunistic Communication

- **DTN:** Utilizes opportunistic communication by taking advantage of transient connections between nodes to deliver data. This is crucial in scenarios where end-to-end paths are not always available.
- **IoT:** Can benefit from opportunistic communication strategies, especially in networks where devices may come into contact intermittently. This approach helps in collecting and transmitting data when possible, even if direct communication paths are not always present.

3.3.7 Network Partitioning and Connectivity Issues

- **DTN:** Designed to handle network partitioning by storing data until connectivity is restored, allowing data to traverse disconnected segments of the network.
- **IoT:** IoT networks often face similar partitioning issues, especially in large-scale deployments or in areas with limited connectivity. Techniques used in DTNs for managing partitioned networks can be adapted for IoT scenarios.

3.3.8 Energy Efficiency Considerations

- **DTN:** Aims to manage energy consumption by optimizing data storage and forwarding strategies, especially in environments with limited power resources.

- **IoT:** IoT devices are typically battery-powered and have limited energy resources. Efficient communication strategies, including those used in DTNs, are essential to minimize energy consumption and extend the operational lifetime of IoT devices.

3.3.9 Remote and Rural Areas

- **Use Case:** IoT devices deployed in remote or rural areas, where network connectivity is intermittent or non-existent.
- **Benefit:** DTN routing can enable data collection and transmission by allowing devices to store and forward data when they encounter other devices or communication gateways.

3.3.10 Addressing Challenges in Harsh Environments

- **Use Case:** IoT sensors deployed in disaster-stricken areas (e.g., earthquake zones, flood areas) where network infrastructure might be damaged or unreliable.
- **Benefit:** DTN can facilitate the collection and forwarding of critical data about environmental conditions or damage assessment, even when direct connectivity is not available.

3.3.11 Wildlife Monitoring

- **Use Case:** Tracking wildlife with IoT sensors placed on animals or in their habitats, where direct communication might be challenging due to mobility or environmental factors.

- **Benefit:** DTN routing allows data collected by mobile or stationary sensors to be forwarded when they come into contact with other nodes or data collectors.

3.3.12 Industrial IoT

- **Use Case:** Sensors in industrial environments where wireless communication might be disrupted by physical obstructions or interference.

- **Benefit:** DTN routing helps in buffering and forwarding critical data, ensuring that information is eventually transmitted to monitoring systems even in challenging conditions.

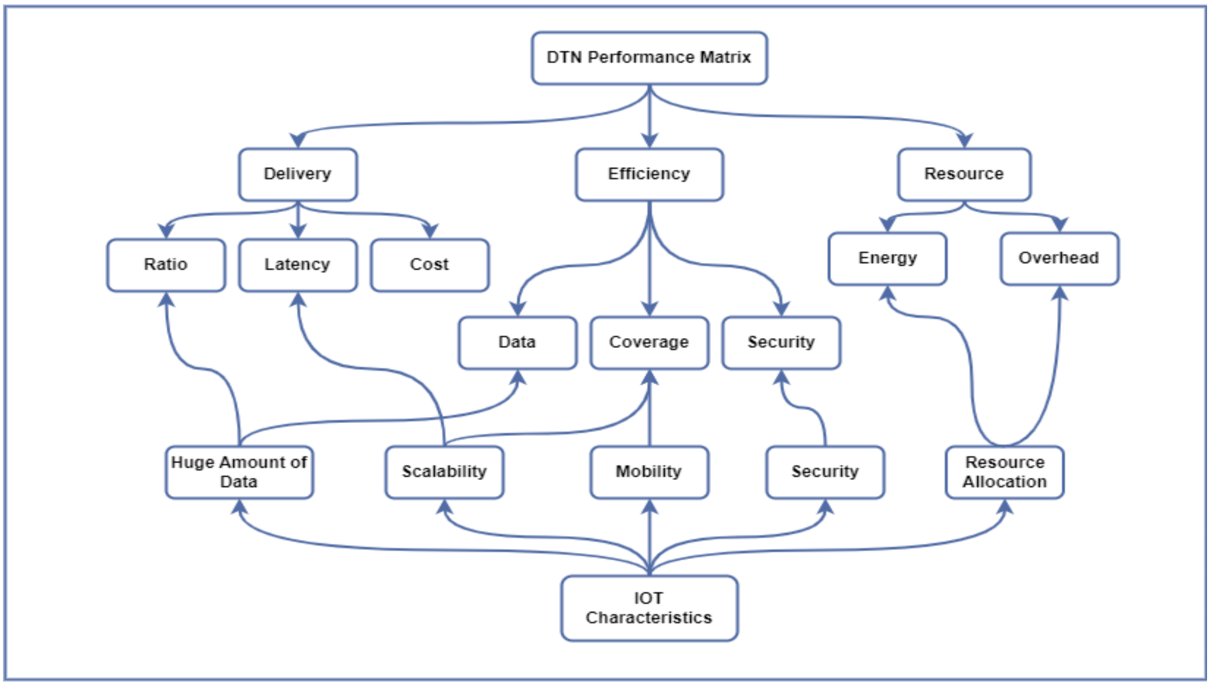


Figure 1 DTN & IoT Interdependency

3.4 Advantages of DTN Routing in IoT

3.4.1 Handling Intermittent Connectivity

Benefit: DTN's Store-Carry-Forward approach is well-suited for environments where connectivity is sporadic. It allows data to be collected and stored locally until a suitable transmission opportunity arises.

3.4.2 Industrial IoT

Flexibility in Network Topologies

Benefit: DTN is adaptable to various network topologies, including ad hoc and dynamic networks. This is useful for IoT networks with changing node locations and connectivity patterns (17).

3.4.3 Enhanced Data Delivery

Benefit: DTN routing can improve data delivery rates in environments with high partitioning or long delays. It ensures that data is eventually delivered, even if not in real-time.

3.5 Challenges and Considerations

3.5.1 Resource Constraints

- **Challenge:** IoT devices often have limited resources (e.g., battery life, storage capacity). DTN routing must be optimized to handle these constraints effectively (27).

- **Solution:** Implement energy-efficient protocols and buffer management techniques that align with the resource limitations of IoT devices.

3.5.2 Latency and Delay Management

- **Challenge:** While DTN can handle delays, excessive latency might not be acceptable in all IoT applications.

- **Solution:** Use adaptive routing schemes that balance between delay tolerance and timely data delivery based on application requirements.

3.5.3 Data Security and Privacy

- **Challenge:** Ensuring data security and privacy in DTN-based IoT networks.
- **Solution:** Implement encryption and secure data handling practices to protect sensitive information as it is buffered and transmitted.

3.6 IoT Challenges and DTN Solutions

IoT systems often face in the following challenges Remote and Harsh Environments:

- **Intermittent Connectivity**: IoT devices deployed in remote areas may experience disconnections from the network for extended periods.
- **Resource Constraints**: Many IoT devices are resource-limited in terms of processing power, memory, and battery life.
- **Scalability**: As the number of IoT devices grows, managing communications efficiently across a massive network becomes challenging.

3.7 DTN as a Solution for IoT:

DTN protocols provide solutions by:

- **Handling Intermittent Connectivity**: DTN protocols such as **Epidemic** or **Spray and Wait** route data even in networks where there is no immediate path from source to destination.
- **Energy Efficiency**: By optimizing transmission strategies and reducing redundant transmissions, DTN can minimize the energy consumption of IoT devices.
- **Buffer Management**: DTN leverages buffering techniques to store data at intermediate nodes until a forwarding opportunity arises

3.8 DTN Applications in IoT

3.8.1 Smart Agriculture:

In rural areas where connectivity is sparse, IoT devices used for monitoring environmental conditions (e.g., temperature, humidity, soil moisture) can use DTN to ensure reliable data collection and forwarding through mobile nodes like tractors or UAVs (Unmanned Aerial Vehicles) (47).

3.8.2 Environmental Monitoring:

Sensor nodes deployed in remote forests or oceans to monitor wildlife or pollution levels can employ DTN protocols to store data locally until a passing node (e.g., satellite or drone) can forward the data to a central server (35).

3.8.3 Disaster Recovery:

In post-disaster scenarios where infrastructure is damaged, DTN can facilitate communication between rescue teams by enabling mobile devices and drones to serve as data carriers, delivering critical information between isolated nodes and command centers.

3.8.4 Vehicular Networks:

DTN is a natural fit for Vehicular IoT (V-IoT) where vehicles act as mobile nodes. DTN protocols can optimize data delivery even in the presence of high mobility and intermittent connectivity between vehicles and roadside units.

3.9 Security in DTN for IoT

IoT devices, especially in DTN, are vulnerable to several security threats such as blackhole attacks, flooding attacks, and message tampering. The lack of continuous connectivity complicates the ability to verify and secure data transmissions.

Some proposed security measures include:

- **Identity-based cryptography** to authenticate nodes in disconnected settings.
- **Buffer management strategies** to prioritize messages from trusted nodes.
- **Anomaly detection systems** to identify and mitigate malicious activities by analyzing traffic patterns and node behavior.

3.10 Open research challenges

Despite the progress in integrating DTN into IoT, there are still open research challenges:

- **Energy Optimization**: Developing energy-efficient DTN routing protocols tailored for ultra-low-power IoT devices.
- **Hybrid DTN-IoT Solutions**: Combining DTN with other networking paradigms (e.g., Low-Power Wide-Area Networks, or LPWAN) to enhance communication in different IoT environments (46).
- **Security Frameworks**: Further research is required to design lightweight, robust security frameworks that can operate efficiently in resource-constrained IoT-DTN systems.

- **Real-World Deployments**: Conducting real-world experiments and developing practical deployment strategies for DTN in IoT use cases (39).

4. DTN ROUTING PROTOCOLS FOR IOT

This chapter explores the various aspects of routing in DTN routing, primarily the store and forward routing. This includes packet delivery, adapting to topological instability, loops avoidance, congestion, and minimizing overhead as inherent challenges in DTN environments. This chapter also discusses the suitability of the these routing mechanism for IoT.

4.1. Introduction

Delay Tolerant Network based routing offer a viable solution to address the challenges of intermittent connectivity in IoT deployments. By using store-carry-forward mechanisms, DTN protocols can ensure reliable data transmission in environments where traditional networks fail. The application of DTN to IoT opens up opportunities for innovation in diverse fields such as agriculture, environmental monitoring, and disaster recovery. However, more research is needed to improve energy efficiency, scalability, and security, which are essential for the widespread adoption of DTN in IoT ecosystems (2) (42).

Within the context of routing, particularly DTNs, the terms routing and forwarding are often used interchangeably. But though related they are different processes, forwarding is performed locally by a router, based on the decision to select the best candidate for next hop among intermediate nodes, while routing is performed at the overall network level foe determination of an end-to-end path between a sender node and a destination node

In this context the term routing may not be most suitable per se for DTNs, as end to end connectivity is unstable, but the term is still used as packet delivery still involves all aspects including adapting to topology and traffic changes, loop avoidance and minimizing resource consumption overhead. However, due to the nature of DTN environments, as aforementioned, many of these tasks are not feasible. Additionally, since there’s no guarantee that a sent bundle will be delivered successfully to the destination or that the forwarding opportunity being used is optimal, DTN "routing" is better described as an **opportunistic forwarding algorithm**. This approach relies on selecting the next hop based on certain rules to maximize the likelihood of eventual delivery to the destination (48).

The assumption is that forwarding a bundle increases the chances of delivery, with the process repeating at subsequent hops until the message reaches its destination. The core challenge in DTN forwarding lies in selecting the best next-hop candidate from neighbouring nodes and determining the right moment to forward the bundle. These decisions depend on the next hop’s likelihood of successful delivery and its contribution to optimizing network performance based on a predefined metric.

However, poor forwarding decisions can result in indefinite delays. Therefore, buffer management becomes crucial, as it helps mitigate this risk. Various buffer management schemes for DTNs will be discussed in later sections.

To understand routing in DTN the two aspects that must be addressed are queuing and forwarding issues, as these two outline the main policies and strategies used in DTN buffer management. Further sections explain the need for specialized DTN routing schemes, comparing them with traditional IP approaches, explaining the requirement of mobility knowledge in process of selection of suitable schemes from major routing schemes, further a comparative analysis is presented based on various performance metrics like delivery ratio, delay, and nodal mobility (13).

4.1. Queuing Policies and Forwarding Strategies

Most DTN studies included in the research survey emphasize the significance of selecting the most suitable buffer management policy and forwarding strategy, further combining them with a optimal routing method. The goal is to enhance message delivery, minimize overheads, and reducing end-to-end delay.

As discussed in previous section, the store-carry-forward (SCF) approach is a modern and optimised variation of store-and-forward routing. In SCF, a node may not have an immediate next-hop available, so it buffers data until forwarding opportunity arises. As a result DTN protocols are designed with the assumption that each node shall maintain a buffering queue (49).

These buffering queues are designed to survive in spite of the challenging DTN environment. Consequently very buffer sizes with small queue are not suitable for DTN. Most protocols require DTN nodes to be equipped with large buffers storage to keep data bundles indefinitely until they can be forwarded according to the SCF scheme.

But buffering alone cannot be a panacea for all network issues. Messages are to be forwarded efficiently toward the goal of delivery, and in any case, for preventing buffer overflow-this is a major problem. Because of the uncertainty of link states, nodes often distribute multiple copies of messages to neighbouring nodes by a technique called flooding. Flooding increases the probability of delivery, but it soon leads to rapid buffer overflows increasing drop rates. Many DTN routing schemes are designed with unlimited buffer spaces; however, this is only theoretical assumption and actual implementation has finite buffer. Therefore buffer space is a limited resource which is critical to protocol success and should be managed judiciously.

Buffer management determines which messages to remove from the queue and when to do so, often in coordination with forwarding decisions. In cases of congestion, it also decides which messages to drop (50) (51).

Though queuing policies in DTNs are mot the primary focus of this research but they form the basis of any improvement to routing in DTN and subsequent adaptation to IoT. Also queue or buffer space is one of the most critical resources that affect overall performance. Interestingly, less emphasis has been placed on buffer management strategies and forwarding schemes improvement for routing in DTN literature (34). Nonetheless, the following sections provide an exhaustive overview of key DTN queuing policies and forwarding strategies.

Local queuing policies at the nodal level in DTNs can establish rules to manage bundles at two key levels:

- **Forwarding Level**: When a node becomes overloaded or congested, bundles should not be forwarded to it temporarily. Bundles traffic should also be reduced from a congested node to avoid loading nearby nodes until the congestion subsides.
- **Nodal Buffer Level**: Here, the focus is on identifying which bundles should be dropped to free up buffer space. This may include bundles currently being received.

Reference introduces several queue management policies that determine which message to drop when the buffer is full, and a new message needs to be accommodated:

- **FIFO (First In, First Out)**: The messages are dropped from queue in the same order as inserted, so the first message is the first to be dropped.
- **MOFO (Evict Most Forwarded First)**: in this strategy the number of times a message is forwarded, and then the same order is followed, when required dropping the first. This gives messages that have been forwarded fewer times a better chance of delivery.
- **MOPR (Evict Most Favourably Forwarded First)**: This policy is a weighted version of MOFO, where the forwarding count is adjusted based on delivery predictability (P) for each message. Each node maintains a forwarding predictability (FP) value for each message, which is updated as:

$$FP = FP_{old} + P \quad \textbf{(1)}$$

The message with the highest FP value is dropped first.

- **SHLI (Evict Shortest Lifetime First)**: In DTN, each message has a timeout value, after which it becomes useless. This policy drops the message with the shortest remaining lifetime first.
- **LEPR (Evict Least Probable First)**: The message with the lowest delivery predictability (P-value) is dropped, as the node is less likely to deliver it.

Queue managers can combine multiple queuing policies in an ordered set, where the primary policy is applied first, and subsequent policies are used to resolve ties. For example, a queuing policy might use the order {MOFO; SHLI; FIFO}.

4.2. Forwarding Strategies

In environments where bandwidth is limited, and connections are prone to interruptions, nodes may not always be able to transmit all the messages they wish to forward. Therefore, the order of message transmission becomes most crucial. Forwarding strategies are generally based on delivery predictability, which refers to the likelihood that a node will successfully deliver a message to its destination.

For example, consider node A forwarding a bundle M to node B, the node B is only an intermediary the destination for the bundle is node D. The delivery predictability, denoted as P(A, B), reflects how likely node A thinks node B is to deliver bundle M to node D.

It may be noted, that if the node being encountered is the destination of any messages, those messages must be delivered instantly, here the destination takes priority over forwarding strategy. Additionally, nodes typically retain messages after forwarding them (if buffer space allows), as they may encounter a more optimal node or the final destination later.

Below are some common forwarding strategies. In each, A and B are the nodes that meet, D is the destination, and P(X, Y) denotes the delivery predictability of node X for destination Y:

- **GRTR (Greater Than)**: Forward the message only if P(B, D) > P(A, D). When two nodes meet, a message is forwarded to the other node if its delivery predictability for the message's destination is higher than the current node’s.
- **GRTRSort**: Forward messages in descending order of P(B, D) – P(A, D). This strategy processes the message queue differently than GRTR. Instead of scanning the queue linearly, it ranks messages based on the difference in delivery predictabilities between the two nodes and forwards the message with the largest improvement first.
- **GRTRMax**: Forward messages in descending order of P(B, D), regardless of P(A, D). This strategy prioritizes messages for which the encountered node (B) has the highest delivery predictability, without focusing on the difference between nodes A and B.
- **COIN**: A message is forwarded based on a random variable X drawn from the uniform distribution U(0, 1). Forward the message only if X > 0.5. This strategy mimics Epidemic Routing but uses a "coin toss" to reduce the number of message transfers. Delivery predictability is not considered.

4.2..1. Forwarding and Queuing Interactions

The relationship between forwarding and queuing strategies, and how they tie into specific routing schemes, will be discussed in Section 3. For now, it is worth noting that various routing strategies have been proposed for DTNs, which draw inspiration from cache replacement policies (52) (53).

Several algorithms have been proposed that resemble common cache management techniques. Each node maintains a list of neighbouring nodes, sorted based on a "cache replacement policy," and broadcasts this information over the network. These caching policies serve as a criterion or basis for prioritising, the ranking inversely also serves as the routing cost. The result is used to assign weights to the network edges.

Some of these strategies include:

- **Most Recently Seen (MRS)**: Similar to the Least Recently Used (LRU) policy, nodes sort their neighbours by the time they were last encountered. The value of weight of any edge between nodes i and j at time t is given by:

$$w(e_{ij\,,}\, t) = t - lastseen_{ij} \qquad \textbf{(2)}$$

○ This strategy focuses on recent encounters, though it may lead to out-dated information as the network topology changes.

- **Most Frequently Seen (MFS)**: Analogous to the Least Frequently Used (LFU) policy, these strategy increments a counter for each neighbour after every encounter. The edge weight is calculated as the inverse of the counter value:

$$w(e_{ij\,,}\, t) = \frac{1}{counter_{ij}\ (t)} \qquad \textbf{(3)}$$

○ MFS tends to reflect more stable, recurring encounters, resulting in lower average delays compared to MRS.

- **Weighted Storage and Frequency (WSF)**: This strategy modifies the edge weight by considering both the buffer size of the node and the frequency of encounters:

$$w(e_{ij\,,}\, t) = \frac{B_j}{counter_{ij}\ (t)} \qquad \textbf{(4)}$$

○ Here B_j represents the buffer size of node j. This strategy balances storage constraints with delay minimization.

- **Aging**: To account for node mobility, an aging factor is applied to reduce the effect of out-dated encounters on current routing tables. This helps the network adapt more quickly to dynamic changes.

4.3. Routing Protocols in DTNs

At first glance, routing in Delay Tolerant Networks (DTNs) might seem like a standard dynamic routing problem, similar to those in **Mobile Ad Hoc Networks (MANETs)**, with the added complexity of prolonged link failures. However, this is not the case. In MANETs, a variety of routing protocols—such as **OLSR**, **AODV**, **LAR** and **STAR** (54)—have been developed to accommodate their dynamic topologies. These protocols can be broadly categorized into reactive and proactive approaches.

- Reactive protocols, like AODV and DSR, do not attempt to find a route until a packet needs to be delivered.
- Proactive protocols, such as OLSR, DSDV, and STAR, periodically exchange control messages to maintain up-to-date routing information, providing immediate routes at the expense of consuming bandwidth for periodic topology updates.

Though there is intermittent connectivity but all these protocols assume that the network is connected, which means that, there does exist an end-to-end path with respect to time and space between all source and destination pairs. In traditional dynamic routing problems, the network is considered continuously connected, with only brief intervals of disconnection. The goal is to find the best available path at any given moment to transfer traffic.

These assumptions, however, do not hold in DTNs. In DTNs, nodes typically lack consistent network state information, such as the location or status of other nodes, or the current network topology (33). Therefore, traditional routing protocols like AODV and OLSR fail to function effectively. In these protocols, when a packet arrives and no end-to-end path is available, the packet is simply dropped.

4.4. AODV Example and Limitations in DTNs

The AODV protocol fails in an intermittently connected network. When a source needs to send data to a destination, f any node along this path becomes unavailable, the protocol fails. Specifically, the entire path is broken. Even though node may have an alternative route, AODV will not use that route because it had previously determined that the next hop. As a result, packets are lost, and the protocol is rendered ineffective.

In a DTN context, intermediate could go into sleep mode to conserve energy. While a node is active, AODV establishes a route through it, which is then stored in the source nodes routing table. If the intermediate node is inactive, packets continue to be sent along this non-functional path, wasting resources. This issue becomes even more severe if multiple such nodes are the only ones in neighbourhood of the gateway to the Internet. If all these nodes go into sleep mode simultaneously, the entire MANET may become disconnected.

The limitations of AODV in intermittently connected networks apply to many other Internet routing protocols as well. In DTNs, end-to-end paths are only intermittently available. As a result, routing and forwarding must be carried out over time to ensure eventual delivery. This is achieved using the **store-carry-forward** mechanism, where intermediate nodes store messages until they can be forwarded.

Thus, point-to-point forwarding becomes a key component of any DTN routing strategy. Given the unique challenges posed by intermittent connectivity, it is also essential to discuss forwarding and buffer management techniques in DTNs as part of the overall routing framework. These techniques are crucial for managing the dynamic nature of DTNs and ensuring efficient message delivery across disconnected segments of the network.

4.5. Types & Classification of DTN Routing Protocols

Routing in Delay Tolerant Networks (DTNs**)** can be conceptualised as an optimization problem, where network connections may become unavailable for some duration, and each node has limited storage capacity. This makes DTN routing a more complex and challenging problem compared to traditional networks. To address these challenges, significant efforts have been made to develop protocols tailored specifically for DTNs (10).

In DTNs, a source node has no advance knowledge about existence of an end-to-end path. As a result, routing involves opportunistic forwarding using the store-carry-forward (SCF) approach, where a message is moved closer to the destination one hop at a time. Knowledge of the mobility patterns of nodes within a DTN partition becomes crucial.

DTN routing protocols may be classified on basis of several factors, with one common method based on the mobility behaviour of nodes, which can be either **deterministic** or **stochastic**.

- **Deterministic Mobility:** The mobility behaviour of a node is termed deterministic if it can be known or predictable, the messages can then be scheduled in advance to achieve optimal results.
- **Stochastic Mobility:** When the future topology of the network is unpredictable, nodes must roam randomly, carrying data in anticipation of encountering a suitable forwarding opportunity.

DTN routing protocols can be divided into two categories:

- **Flooding Protocols**: An example of the stochastic method is Flooding, also known as random routing, relying completely on the absence of mobility relation data. In scenarios where a node has no knowledge of the network's state, it randomly forwards packets to neighbouring nodes, hoping that one will eventually deliver the packet to its destination**.**
- **Forwarding Protocols:** An example of the deterministic method also termed as history or estimation based routing are Forwarding protocols. They improve upon random flooding by allowing nodes to estimate the forwarding probability of their neighbours. This enables more informed forwarding decisions.

DTN routing protocols can also be classified into **source routing** and **per-hop routing** (1) (48).

- **Source Routing:** In this approach, the complete path is determined at the start by source and encoded within the message. The route remains unchanged as the message traverses the network. While it may seem counterintuitive in a DTN, where source nodes cannot predict end-to-end paths, source routing can be useful in specific DTN applications that rely on some level of mobility knowledge.
- **Per-Hop Routing:** In, per-hop routing each next hop for a message is determined by the next hop at every forwarding step. This method takes advantage of **local information** about available contacts and queues at each hop, which would be unavailable to the source node at the time of transmission. Per-hop routing is more flexible and adaptive to the intermittent connectivity of DTNs.

When mobility patterns can be used to estimate forwarding probabilities, model-based forwarding protocols provide a more efficient way of routing messages. These protocols utilize mobility models to make more informed decisions about which node is likely to bring the message closer to its destination.

Finally, routing schemes in DTNs can be further categorized into proactive and reactive routing protocols.

- **Proactive Routing:** In proactive routing, node movements are controlled, and routes are computed in advance, independent of traffic demands. This allows for the creation of efficient routes before any traffic needs to be sent.
- **Reactive Routing:** In reactive routing, routes are discovered on-demand as node movements are not predictable, this type of routing is used the destination is unknown. This is more adaptive to unpredictable environments where nodes move freely.

In summary, DTN routing is a complex and dynamic problem, with numerous approaches depending on node mobility, available information, and network goals. Understanding these different routing strategies is key to addressing the unique challenges posed by intermittently connected networks.

4.5..1. Mobility Knowledge and levels of Mobility Knowledge

Understanding nodal inter-arrival times—when nodes come into transmission range of each other—is crucial for efficient routing in Delay Tolerant Networks (DTNs). This information is measured in terms of link-state knowledge about the behaviour of node in the network. Depending on the application, a DTN might have full, partial, or no knowledge of the network topology, and this degree of knowledge directly influences the routing approach best suited for that scenario (15).

The level of mobility knowledge available in a DTN largely determines the appropriate routing strategy. It is therefore possible to match specific routing protocols to particular applications based on the application’s level of awareness of nodal mobility and other performance metrics, such as delivery rate and delay tolerance. This report will later introduce a mapping between DTN routing schemes and applications based on such metrics.

- In some DTN applications, future node behaviour is almost fully predictable.

Examples include:

- **Public transportation systems (e.g., bus routes)**: The buses follow pre-defined and scheduled paths.
- **Space missions (e.g., planetary trajectories)**: Satellites move along pre-determined orbits.

In such cases, **full mobility knowledge** allows precise scheduling of message transmissions to optimize delivery.

However, in many other DTN applications, mobility patterns are less predictable. One might expect that wild animals, military personnel, or civilians in rural areas move in a random manner. Yet, studies suggest that while mobility may appear random, it often follows recurrent patterns (12) (11). For instance, individuals, vehicles, or animals tend to revisit specific locations over time, forming recognizable patterns.

Recurrence is a notable feature in many DTN mobility models:

- **Humans** frequently revisit the same places, such as workplaces, grocery stores, and recreational areas.
- **Workers** repeat specific tasks, like meeting the same clients or completing regular duties.
- **Vehicles** and **animals** tend to return to the same destination.

Recurrence and temporal locality are two different phenomena ,in that it doesn't rely on strict timing for events to repeat. While temporal locality focuses on the exact time an event happens, recurrence makes no assumptions about when a location or interaction will occur again. As a result, **Least Recently Used (LRU)** cache policies, which prioritize the most recent data, are less effective in DTNs. Instead, **Least Frequently Used (LFU)** policies, which emphasize frequency of visits or interactions, are better suited since they account for the recurrent nature of node mobility.

The recurrence of mobility patterns plays a critical role in choosing appropriate queue management and routing strategies. While the details of queue management policies in

DTNs are beyond the scope of this document, the importance of recurrence over temporal factors influences how messages are stored, carried, and forwarded in a DTN.

In the following sections, we will examine the main DTN routing schemes in detail, focusing on their performance and how they address the challenges of varying mobility knowledge.

4.6. DTN Routing Schemes

Random routing (or simple flooding) is utilized when there is no knowledge of the network's topology. In this case, the node carrying a data bundle sends the message to every node it encounters, resulting in redundancy and buffer space depletion as multiple copies of the same message circulate. To reduce redundancy, Epidemic Routing (ER) checks if a neighbour already has a copy of the message before forwarding it. If the neighbour has not received the message, it is forwarded. This method mimics the spread of an epidemic by "infecting" new nodes with unseen messages (7).

Though ER ensures nearly all messages are delivered in some studies, it assumes unlimited buffer space, which is unrealistic. Spray and Wait was introduced to control flooding by spreading a limited number of message copies (L) to distinct relay nodes. If the destination is not found, the nodes enter a "wait" phase until the destination is within reach. Spray and Wait requires sufficient mobility to be effective (55) (56).

Direct Delivery is an alternative extreme, where the source holds onto the message and delivers it directly when the destination comes within range. This method has minimal overhead but potentially high delay, particularly in networks with no knowledge of topology. The approach works well in cases where mobility patterns are highly predictable, such as trains on a fixed schedule.

- **Probabilistic/History-Based Routing**

When nodes follow recurrent patterns, **Probabilistic Routing** offers an efficient alternative to Epidemic Routing. A prominent protocol in this category is **PRoPHET** (Probabilistic Routing Protocol using History of Encounters and Transitivity). PRoPHET uses a delivery predictability metric, P(a,b), which indicates the likelihood of node a delivering a message to node b. As nodes encounter each other, they exchange summary vectors containing delivery predictability information. Nodes that are encountered frequently increase their delivery predictability values.

The PRoPHET metric is affected by time (aging) and transitivity. For example, if node A frequently meets node B and node B frequently meets node C, then node C is likely a good relay for messages destined for node A. However, probabilistic schemes can suffer from high message latency or message loss if a bundle is held by a single node that gets destroyed.

- **Model-Based Routing**

Model-Based Routing (MBR) leverages predefined world models (e.g., roadmaps) and user mobility profiles to make better decisions on message relaying. These models help predict the movements of nodes in structured environments like highways or cities. For instance, nodes traveling on highways or satellites in orbit follow strict paths, which MBR can exploit to reduce flooding and improve delivery.

Mobility models often assist in evaluating routing protocols, including probabilistic approaches, by simulating real-world mobility traces.

- **Node Movement Control-Based Routing**

In Node Movement Control-Based Routing, nodes are designed to adjust their mobility patterns to improve network performance, such as reducing transmission delay. Message Ferrying is a popular technique, where special mobile nodes called Message Ferries (MFs) move between disconnected nodes to carry data bundles. There are two main variants of this scheme:

1. **Node-Initiated Message Ferrying (NIMF)**: Nodes move toward predefined ferry routes to exchange data.
2. **Ferry-Initiated Message Ferrying (FIMF)**: Ferries adjust their paths to meet nodes based on service requests.

Alternatively, Data MULES are mobile nodes with random mobility patterns that collect data from static sensor nodes and deliver it to access points. This approach saves power for sensor nodes but introduces higher latency.

- **Coding-Based Routing**

Erasure Coding and **Network Coding** techniques are applied to DTNs to reduce latency and improve delivery rates. In Erasure Coding, an original message is divided into multiple blocks, and only a subset of these blocks is needed to reconstruct the message. This technique is useful for applications with strict time constraints. Network Coding further improves throughput by combining multiple packets into a single transmission, enabling efficient data exchange between nodes.

- **Vector-Based Routing Schemes**

Vector-Based Routing (VBR) is an enhancement to both flooding and History-based routing approaches. In **Flooding-based Vector Routing (FVR)**, nodes compute movement vectors based on their coordinates and velocities, and forward messages to nodes moving in more favourable directions. **History-based Vector Routing (HVR)** extends this idea by keeping a record of neighbour vectors, improving forwarding efficiency. However, rapid vector exchanges may burden buffer space in dense networks.

Other routing schemes include:

- **Multicast Routing**: This scheme supports group-based communication, such as in disaster recovery scenarios where information must be distributed to rescue teams.
- **Inter-Region Routing**: Nodes form clusters with defined boundaries for communication between different regions, addressing challenges like protocol translation and route selection.
- **Delegation Forwarding (DF)**: This scheme addresses Epidemic Routing's overhead by assigning quality levels to nodes. Forwarding occurs only when the receiving node has a higher quality level, reducing redundant message replication.

These schemes aim to optimize DTN performance based on the performace metrics mentioned earlier.

4.6..1. Queuing Policies and Routing Schemes

Queue management and forwarding policies, while related; serve distinct functions in Delay-Tolerant Networks (DTNs). These processes complement each other in scenarios like when a node holding a data bundle encounters the final destination for that bundle. In such cases, the bundle is forwarded and deleted from the queue since it is no longer needed. However, if the destination is not reached, a single copy of the bundle may be forwarded, based on the specific strategy being used also remaining copy in the node's queue may be discarded.

The key consideration in such cases is the global goal of optimizing routing performance across the network, which depends on the available knowledge of the network state. Achieving this "global knowledge" is highly unlikely in most DTN instances. As a result, comparing routing schemes and queuing policies becomes complex,

In the context of probabilistic routing, queuing policies like FIFO, MOFO, MORP, SHLI, and LERP were proposed through simulations. The study shows that combining probabilistic routing with thoughtful buffer management and forwarding strategies significantly improves DTN performance in terms of the overall delay, overhead and delivery rate. This improvement is expected, as probabilistic routing involves learning through past encounters to improve delivery predictability. However, probabilistic routing is not universally ideal and has certain limitations (57) (43) (54).

For Epidemic Routing (ER), it is argued that common buffer management policies are not suitable given the overhead constraints in DTNs. Instead, they propose a theoretical Global knowledge-Based Scheduling and Drop (GBSD) policy to optimize delivery rate and delay. However, GBSD's reliance on global network information makes it impractical for real-world DTNs.

Another work proposes the N-Drop (ND) policy, which aims to control congestion in networks using ER. In ND, a node's buffer is scanned for bundles that have been forwarded a certain number of times (≥ N) and those bundles are discarded to make room for new ones. If no such bundles are found, the oldest bundle in the queue is dropped. Although this policy reduces disruptions and delays, moving bundles between buffer and storage introduces significant processing overhead, particularly in extreme environments.

Some authors explore buffer allocation fairness in Message Ferrying DTNs (MF-DTNs), where the contention is for buffer space on Message Ferries (MFs) rather than for wireless channels. MFs use information like node sequence, session details, and expected contact time to construct forwarding tables and manage buffer space. Based on performance metrics such as Path Metric, Ferry Transportation Cost, Fair Buffer Allocation Scheme etc. are proposed to optimize forwarding decisions and buffer allocation.

Similarly buffer Management Policy for Mars Intelligent Proximity Networks is proposed for prioritizing image data based on significance. Bundles with higher significance are forwarded first, while lower-priority bundles are dropped when the buffer is full. This ensures that the most important data is delivered promptly. While designed for IPNs, this policy could also apply to direct delivery or model-based routing schemes.

Table 4 Relations between Buffer Management Policies and Routing Schemes

Buffer Management Policy	**Routing Scheme**	**Metric(s)**
N Drop	History-Based	Delivery rate
DT, DF, etc.	Epidemic	None suitable for DTNs
GBSD	Epidemic	Delivery rate, delay
ND	Epidemic	Disruption, delay
MF-DTN	MF-DTN	Path Metric, Ferry Transportation Cost
BMP-MIPN	Direct Delivery, Model-Based	Delivery of higher-priority data

4.6..1.1. Epidemic Routing:

Epidemic routing involves flooding the network with multiple copies of the message, ensuring that at least one copy reaches the destination. This is particularly useful in networks with unpredictable mobility, but it comes with high overhead, which may be unsuitable for resource-constrained IoT devices.

4.6..1.2. Spray and Wait Protocol:

Spray and Wait is a more resource-efficient approach compared to Epidemic routing. It involves spraying a limited number of copies of the message across the network, and then each node waits to forward the message until it encounters the destination or a node closer to it. This approach strikes a balance between delivery ratio and energy efficiency, making it a good candidate for IoT scenarios with intermittent connectivity (56).

4.6..1.3. PRoPHET Routing:

The **Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET)** leverages the history of node encounters to estimate the probability of successful message delivery. This protocol reduces unnecessary transmissions, making it more energy-efficient, which is essential for IoT networks with constrained devices.

4.6..1.4. MaxProp:

MaxProp uses a priority mechanism for message forwarding based on a cost metric (such as message hop count). It works well in highly dynamic networks, such as vehicular IoT systems, where node mobility

Here is a table summarizing different Delay Tolerant Routing (DTR) protocols used for the Internet of Things (IoT):

Table 5 Delay Tolerant Routing (DTR) protocols used for the Internet of Things (IoT)

Protocol	Key Features	Strengths	Weaknesses	Suitable Applications
Probabilistic Routing (PRoPHET)	Utilizes encounter history to predict future node contacts and make routing decisions.	Adaptive to dynamic network conditions, good for mobile IoT.	High overhead due to historical data tracking.	Mobile IoT, social IoT networks.
Spray and Wait	Distributes a limited number of message copies and waits for the destination to be reached.	Efficient in energy and resource-limited environments.	Can cause significant delays due to "wait" phase.	Resource-constrained IoT devices.
MaxProp	Prioritizes messages based on estimated delivery likelihood and deletes least likely ones during storage.	Good for storage-limited nodes, efficient prioritization.	High overhead for message prioritization.	Vehicular IoT, smart cities.
Epidemic Routing	Floods the	High delivery ratio	High	IoT networks

	network by replicating messages to all nodes in contact, ensuring maximum delivery probability.	in highly dynamic networks.	overhead, network congestion, and excessive energy use.	with intermittent connectivity.
Delay Tolerant Networking (DTN)	Relies on store-carry-forward mechanism to deliver messages across intermittent or long-delay connections.	Suitable for environments with long delays and disruption.	High latency due to store-carry-forward process.	Remote sensing IoT, satellite IoT.
RAPID	Prioritizes messages based on delivery deadlines, ensuring that time-sensitive messages are delivered first.	Ensures low delay for high-priority messages.	Complexity in prioritization and decision-making processes.	Time-sensitive IoT applications, e.g., healthcare IoT.
Context-Aware Routing	Uses contextual information like location, energy levels, and movement patterns to make routing decisions.	Context-aware routing improves efficiency in dynamic networks.	May not be effective in highly unpredictable environments.	Smart homes, IoT-based industrial monitoring.
CAR (Context-aware Adaptive Routing)	Combines mobility prediction and encounter probability with energy constraints to optimize routing.	Optimizes routing in heterogeneous IoT environments.	High computational cost for context analysis.	IoT environments with dynamic node behaviour.
Spray and Focus	Combines spraying limited message copies with forwarding based on encounter probability.	Balances between message replication and delivery probability.	May still cause high delays in sparse networks.	IoT applications with semi-predictable mobility patterns.

This table covers popular protocols with varying strategies for delay-tolerant routing in IoT environments, balancing between delivery reliability, overhead, and energy efficiency.

4.7. Classification of DTN Applications

The document proposes a classification of DTN applications based on characteristics such as node mobility, buffering, transmission challenges, and delay tolerance. Table 6 provides a mapping of applications to these characteristics:

Table 6 Mapping of applications to DTN characteristics

Application	Stationary Sensing Nodes	Random Mobility Sensors	Physically Challenging Medium	Higher Level of Partitioning	Need for MFs/Couriers	Delay is Significant	Buffer Constraint	Energy Constraint
Forestry	✓		✓	✓		✓	✓	✓
Wildlife Monitoring	✓		✓	✓		✓	✓	✓
Natural Disasters Sensing	✓		✓	✓	✓	✓	✓	✓
Underwater Sensing	✓		✓	✓	✓	✓	✓	✓
Inter-Planetary Networks	✓		✓	✓				
Village NWs	✓		✓		✓			
Bus Networks	✓		✓		✓			
Personnel Monitoring	✓		✓					
Accident/Disaster Recovery	✓		✓	✓	✓	✓	✓	✓
Industrial Sensing	✓		✓					

4.8. Analysis of Routing Schemes

Table 7 provides a summary of DTN routing schemes, focusing on their properties and weaknesses:

Table 7 Summary of DTN routing schemes

Routing Scheme	Properties	Downsides
Direct Delivery	Guaranteed delivery if the destination node is encountered	Data may be delayed indefinitely if destination is never met
Deterministic	Used when future topology is known	Limited to scenarios with known mobility; often not applicable
Random/Epidemic	Almost guaranteed delivery; useful without mobility knowledge	High overhead; requires large buffer sizes; broadcast storms
Probabilistic/History-based	Less overhead than random routing; popular in smaller networks	Long learning curve; fewer message copies; higher risk of data loss

Model-based	Effective with life-model of the topology	Requires knowledge of motion patterns; limited to predictable scenarios
Movement Control-based/DataMULES	Suitable when node movement can be controlled; uses special mobile nodes	Requires special nodes; limited delay reduction
Coding-based	Higher delivery rates in specific settings; efficient in certain cases	Additional energy consumption; requires coding operations

DTN routing schemes must be selected based on a comprehensive understanding of the specific application characteristics and requirements. The mapping provided in this document offers a framework for aligning DTN applications with appropriate routing strategies, emphasizing the need for application-specific solutions to effectively address diverse challenges in delay-tolerant networking.

Delay Tolerant Networks represent a crucial framework for enabling communication in scenarios where traditional Internet protocols fall short. DTNs are designed to address the challenges posed by intermittent connectivity, long-lasting network partitions, and extreme environmental conditions. Such challenges are commonly encountered in environments like forests, deserts, industrial plants, underwater habitats, disaster zones, battlefields, and inter-planetary networks (1) (36) (32) (28) (3).

5. ADRESSING SECURITY CHALLENGES IN DTN & IOT

This chapter explores the intricacies of security challenges and attacks in DTNs and IoT, highlighting key challenges and potential solutions for detecting malicious activities while ensuring network resilience in the face of evolving threats.

5.1. Introduction

The rise of the Internet of Things (IoT) has significantly increased the number of connected devices in networks, often operating in environments with limited connectivity or high latency, such as Delay Tolerant Networks (DTNs). DTNs are characterized by their ability to operate in challenging network conditions, where there may not be a continuous end-to-end connection. The store-carry-and-forward mechanism allows messages to be carried by intermediary nodes until a suitable forwarding opportunity is available.

Both DTNs and IoT environments face critical security challenges, particularly in intrusion detection**.** Intrusion Detection Systems (IDS) are vital in protecting these networks from various attacks, such as data tampering, denial of service, routing attacks, and unauthorized access. However, due to the intermittent connectivity, resource constraints, and decentralized nature of these networks, designing efficient IDS for DTNs and IoT becomes highly complex (58) (59) .

This thesis explores the intricacies of intrusion detection in DTNs and IoT, highlighting key challenges and potential solutions for detecting malicious activities while ensuring network resilience in the face of evolving threats**.**

5.2. Security Challenges

Security in Delay Tolerant Networks (DTN) and Internet of Things (IoT) involves addressing different challenges due to the unique characteristics of each type of network. Here's a detailed comparison of security considerations for DTNs and IoT networks:

DTN Security:

- **Intermittent Connectivity:** The lack of continuous connectivity can make it difficult to establish secure channels and verify the authenticity of communication.
- **Buffering and Storage:** Data stored in intermediate nodes may be vulnerable to unauthorized access or tampering. Ensuring the integrity and confidentiality of buffered data is a challenge.
- **Node Mobility:** The dynamic nature of node movement can complicate the implementation of security protocols and key management.
- **Limited Resources:** Some DTN nodes may have limited computational power and storage, affecting the feasibility of complex security algorithms.

IoT Security:

- **Large Scale:** IoT networks often consist of a vast number of devices, making it challenging to manage and secure all endpoints (60).
- **Resource Constraints:** Many IoT devices have limited processing power, memory, and battery life, which can restrict the use of resource-intensive security mechanisms.
- **Diverse Devices:** The heterogeneity of IoT devices can lead to varied security requirements and vulnerabilities.
- **Data Privacy:** IoT devices often handle sensitive data, raising concerns about data privacy and secure communication.

5.3. DTN & IoT Security Objectives

DTN Security Objectives:

- **Authentication:** Ensuring that nodes are legitimate and can be trusted to forward data.
- **Data Integrity:** Verifying that data has not been altered during transit or while stored in intermediate nodes.
- **Confidentiality:** Protecting data from unauthorized access or exposure, especially while stored in intermediate nodes.
- **Availability:** Ensuring that data and services are accessible despite the intermittent connectivity and dynamic nature of the network.

IoT Security Objectives:

- **Authentication:** Validating the identity of devices and users to prevent unauthorized access.
- **Data Integrity:** Ensuring that data collected and transmitted by IoT devices remains unaltered.
- **Confidentiality:** Protecting sensitive data from eavesdropping or unauthorized access.
- **Access Control:** Implementing policies to control who can access or modify data and network resources.
- **Privacy:** Safeguarding the personal information collected by IoT devices.

5.4. Security Mechanisms

DTN Security Mechanisms:

- **Cryptographic Techniques:** Encryption and digital signatures can protect data confidentiality and integrity. Public Key Infrastructure (PKI) can be used for authentication and key management.
- **Secure Data Storage:** Techniques such as data encryption and access controls can help protect data stored in intermediate nodes.
- **Reputation Systems:** To address trust issues in environments where nodes are frequently disconnected, reputation-based systems can assess and validate node trustworthiness.
- **End-to-End Security:** While end-to-end security is challenging due to intermittent connectivity, hybrid approaches that combine end-to-end encryption with secure storage in intermediate nodes can be employed.

IoT Security Mechanisms:

- **Device Authentication:** Use of secure authentication protocols such as certificates or tokens to validate devices and users.
- **Data Encryption:** Ensuring that data is encrypted during transmission and at rest to maintain confidentiality and integrity.

- **Access Control:** Implementing role-based access control (RBAC) and access control lists (ACLs) to regulate access to data and resources.
- **Network Security:** Using firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) to protect the network from unauthorized access and attacks.
- **Firmware Updates:** Regularly updating firmware to patch vulnerabilities and enhance security.

5.5. Threats and Vulnerabilities

DTN Threats:

- **Data Tampering:** Intermediate nodes may alter or inject malicious data if not properly secured.
- **Node Compromise:** If a node is compromised, it may disrupt the network or tamper with data.
- **Denial of Service (DoS):** Attacks that target the network's ability to deliver data by flooding it with traffic.

IoT Threats:

- **Botnets:** Compromised IoT devices can be used to form botnets for large-scale attacks.
- **Data Breaches:** Unauthorized access to sensitive data due to inadequate security measures.
- **Physical Attacks:** Devices can be physically tampered with or stolen to gain unauthorized access or disrupt operations.

5.6. Best Practices

DTN Best Practices:

- **Implement Robust Authentication:** Use strong cryptographic methods for node authentication and data integrity.
- **Encrypt Data:** Apply encryption to data at rest and in transit to protect confidentiality.

- **Regularly Update Security Protocols:** Adapt security mechanisms to evolving threats and network conditions.
- **Monitor and Respond:** Continuously monitor network activities and respond to suspicious behaviours.

IoT Best Practices:

- **Secure Device Deployment:** Ensure secure setup and configuration of devices, including changing default credentials.
- **Use Strong Encryption:** Encrypt communications and stored data to safeguard against unauthorized access.
- **Regular Updates:** Keep device firmware and software up-to-date to protect against known vulnerabilities.
- **Implement Access Controls:** Apply granular access controls to manage data and network resource access.

5.7. Challenges of Intrusion Detection in DTNs and IoT

- **Intermittent Connectivity**

- **DTN Aspect**: In DTNs, nodes experience intermittent connections due to mobility or harsh environments, making it difficult for IDS to detect and respond to intrusions in real-time.
- **IoT Aspect**: In IoT, devices may have limited or irregular connections to the Internet, meaning that security systems relying on continuous connectivity may not be effective.

- **Resource Constraints**

- **DTN Aspect**: Nodes in DTNs are often battery-powered and have limited storage and processing capabilities, restricting the complexity of security algorithms.
- **IoT Aspect**: Many IoT devices, such as sensors and actuators, have minimal computational resources, further limiting the feasibility of traditional intrusion detection mechanisms.

- **Decentralized Control**

- **DTN Aspect**: The absence of a central authority or consistent communication paths in DTNs makes coordination of IDS across nodes difficult.
- **IoT Aspect**: IoT networks often consist of heterogeneous devices with decentralized control, requiring IDS that can operate in distributed and autonomous environments.

- **High Latency**

- **DTN Aspect**: The delay-tolerant nature of DTNs introduces high latency, which can delay the detection and mitigation of intrusions.
- **IoT Aspect**: IoT networks deployed in remote areas or resource-constrained environments may experience similar latency, delaying real-time detection of security threats.

5.8. Intrusion Detection Techniques for DTNs and IoT

- **Signature-Based Detection**

- **Principle**: This method relies on predefined patterns or signatures of known attacks. Each packet or message is scanned and compared against these known signatures.
- **Application in DTNs**: Given the lack of continuous communication, signature-based IDS in DTNs may only detect attacks when nodes eventually connect to others and share their observations.
- **Application in IoT**: In IoT, signature-based systems can be effective for detecting known attacks, but the frequent update of signatures may be difficult due to resource constraints.

- **Anomaly-Based Detection**

- **Principle**: This method identifies unusual behaviours or deviations from normal network activity, flagging them as potential intrusions.
- **Application in DTNs**: Anomaly detection can be effective in DTNs, especially since normal behaviour patterns are often predictable. However, nodes must store abnormal behaviour patterns locally until a connection is available for wider dissemination.

- **Application in IoT**: In IoT networks, anomaly detection can be useful for identifying novel attacks, but resource constraints limit the complexity of anomaly-detection algorithms that can be run on edge devices.

- **Hybrid Detection**

- **Principle**: Hybrid detection combines both signature-based and anomaly-based methods, leveraging the advantages of each to improve detection accuracy.
- **Application in DTNs**: Hybrid methods can be used to detect known attacks using signatures and identify new or unknown threats through anomaly detection. The challenge is optimizing these systems to operate in environments with limited resources and delayed communication.
- **Application in IoT**: In IoT, hybrid systems can balance real-time detection using lightweight signatures and more resource-intensive anomaly detection for deeper threat analysis.

5.9. Intrusion Detection Algorithms for DTNs and IoT

- **Watchdog Mechanism for DTNs**

- **How It Works**: Nodes in the DTN monitor the forwarding behaviour of their neighbours. If a node fails to forward packets as expected, it may be flagged as misbehaving or malicious.
- **Challenges**: In highly partitioned networks, watchdog systems may suffer from false positives due to legitimate delays in forwarding, rather than malicious behaviour.

- **Trust-Based Intrusion Detection**

- **How It Works**: Trust-based systems evaluate the behaviour of nodes over time. Each node is assigned a trust value based on its past interactions, and nodes with low trust scores are excluded from participating in routing decisions.
- **Challenges**: Trust evaluation can be difficult in DTNs and IoT due to the intermittent connectivity and dynamic nature of these networks. Additionally, malicious nodes can engage in "trust attacks" by initially behaving well to build trust, then launching an attack after gaining high trust scores.

- **Energy-Aware IDS**

- **How It Works**: This system monitors the energy consumption patterns of nodes. Anomalies in energy usage, such as a node consuming too much energy due to unnecessary packet forwarding, may indicate malicious behaviour.
- **Challenges**: In both DTNs and IoT, energy-aware IDS must operate with minimal overhead to avoid depleting the limited resources of the devices they are protecting.

5.10. Specific Attacks in DTNs and IoT

- **Blackhole Attack**

- **Description**: In this attack, a malicious node falsely advertises itself as having the best route to the destination, then drops all packets it receives.
- **Defense**: Trust-based routing protocols or collaborative IDS can help mitigate blackhole attacks by monitoring and flagging nodes that consistently fail to deliver packets.

- **Wormhole Attack**

- **Description**: A wormhole attack involves two colluding nodes that establish a low-latency link between each other and use it to selectively drop or modify messages.
- **Defense**: Secure neighbour discovery protocols and time-based detection mechanisms can help identify wormhole attacks by measuring the actual distance between nodes.

- **Sybil Attack**

- **Description**: In a Sybil attack, a single node pretends to be multiple identities, potentially overwhelming the network with false information.
- **Defense**: IDS using identity verification and trust management can detect Sybil attacks by tracking inconsistencies in node behaviour and verifying the uniqueness of node identities.

5.11. Recent Research Trends

Several solutions including recent ML based techniques and Blockchain technology have been proposed by researchers, but are computationally extensive and thus not suitable for the LLN.

- Machine Learning-Based IDS

- **Principle**: Machine learning (ML) techniques can be used to detect intrusions by analyzing vast amounts of network data and identifying patterns associated with malicious behaviour.
- **Application in DTNs and IoT**: ML-based IDS can be trained to detect a wide range of attacks. However, applying ML in DTNs and IoT is challenging due to limited computational resources and the difficulty of obtaining real-time data in delay-prone environments.

- Blockchain-Based Security

- **Principle**: Blockchain technology can be used to create a tamper-proof, decentralized ledger for securing communications in DTNs and IoT.
- **Application in DTNs and IoT**: Blockchain can help secure trust-based routing and intrusion detection by ensuring that trust values and intrusion reports are immutable. However, the high energy and bandwidth requirements of blockchain present challenges in resource-constrained environments.

Intrusion detection in Delay Tolerant Networks and IoT presents unique challenges due to intermittent connectivity, resource limitations, and decentralized network structures. Traditional IDS approaches must be adapted or redesigned to handle these environments effectively. By leveraging hybrid detection systems, machine learning, and trust-based mechanisms, intrusion detection can be made more robust in these challenging environments (19) (61) (62).

Further research is needed to develop lightweight, efficient IDS that balance security, energy consumption, and network performance, ensuring that DTNs and IoT systems remain.

6. OSNW: OPTIMIZED BUFFER MANAGEMENT POLICY FOR TAILORING DTN ROUTING PROTOCOLS TO IOT

This chapter explains the proposed concept of Optimised Spray and Wait tailored to the specific needs of IoT applications, with consideration of factors like energy efficiency, network overhead, and latency enhances the protocol's adaptability in dynamic environments, optimizing message spraying techniques, and integrating buffer management strategies to further improve its performance in resource-constrained settings.

6.1. Introduction

The Spray and Wait (SNW) protocol is a well-known routing strategy specifically designed for Delay-Tolerant Networks (DTNs). It is used to manage the challenges associated with intermittent connectivity and long delay paths in DTNs. The protocol balances the trade-offs between resource utilization and delivery efficiency, aiming to achieve reliable data delivery even in the absence of continuous end-to-end paths.

This chapter discusses the Spray and Wait routing protocol in the context of IoT environments, emphasizing how it can support IoT applications requiring communication in intermittently connected or highly mobile scenarios. This two-phase approach significantly reduces the overhead associated with message duplication compared to Epidemic Routing, while still ensuring a high delivery probability in delay-tolerant environments. The protocol is especially suited for networks where node mobility can vary widely, and energy efficiency is critical (56) (63).

6.2. The Spray and Wait Protocol

It consists of two main phases:

A. **Spray Phase**: A limited number of message copies (L) are "sprayed" or distributed to nodes within the network. Instead of flooding the entire network (as in **Epidemic Routing**), a controlled number of message copies are disseminated.

B. **Wait Phase**: Each node carrying a message copy enters a "wait" phase. The nodes carrying these message copies will only forward the message to the destination when they directly encounter it.

This two-phase approach significantly reduces the overhead associated with message duplication compared to Epidemic Routing, while still ensuring a high delivery probability in delay-tolerant environments. The protocol is especially suited for networks where node mobility can vary widely, and energy efficiency is critical.

- **Spray and Wait in IoT Applications**

IoT environments present unique challenges due to **resource constraints**, **mobility**, and **intermittent connectivity**. The Spray and Wait protocol can address these challenges in several IoT scenarios:

- Smart Agriculture

In smart agriculture, IoT devices like sensors are often deployed across vast, rural areas. Stable network connectivity is not guaranteed, and nodes may be dispersed far from communication infrastructure. Mobile nodes, such as drones or tractors, can act as carriers for data transmission between isolated sensors and base stations.

Spray and Wait can be applied by having these mobile nodes "spray" a limited number of message copies to nearby IoT devices. As mobile carriers move throughout the farm, the stored data can be delivered when they encounter nodes with better connectivity.

- Environmental Monitoring

Environmental monitoring in remote forests or oceans involves collecting data from sensor nodes deployed in disconnected or harsh environments. These sensors may only encounter connectivity at infrequent intervals, such as when a satellite or drone passes overhead.

In this case, the Spray and Wait protocol can help by distributing data among mobile nodes (e.g., boats, drones) and waiting until one of the mobile nodes encounters a satellite or an access point to forward the data to the central server.

- Vehicular IoT (V-IoT)

Vehicular IoT involves communication between vehicles and roadside units (RSUs) or other mobile nodes in scenarios like traffic management or autonomous driving. These networks often experience intermittent connectivity due to the high mobility of vehicles.

Spray and Wait is ideal for such applications, as vehicles can carry data packets and deliver them when they come into contact with an RSU or another vehicle closer to the destination.

- Disaster Recovery

During natural disasters, network infrastructure is often damaged, and IoT devices used for monitoring and recovery efforts are deployed in highly mobile and disconnected environments.

Spray and Wait ensures that critical information from sensors or drones in disaster zones can be forwarded to rescue teams or command centers once an opportunity for forwarding arises, even if no immediate path is available at the time of data generation.

6.3. Advantages of Spray and Wait for IoT

- **Energy Efficiency**: IoT devices are often battery-powered and resource-constrained. Spray and Wait minimizes excessive transmissions, reducing energy consumption compared to other DTN protocols like Epidemic Routing.
- **Low Network Overhead**: By limiting the number of message copies sprayed into the network, Spray and Wait prevents the network from being flooded with redundant data, reducing bandwidth consumption and memory use on IoT devices.
- **Scalability**: Spray and Wait is well-suited for large-scale IoT deployments where managing network traffic is crucial. The protocol's ability to scale efficiently makes it an attractive option for IoT systems involving thousands of devices.
- **High Delivery Probability**: Even in highly disconnected environments, Spray and Wait ensures that at least a limited number of message copies reach their destination through opportunistic forwarding.

6.4. Detailed working of Spray and Wait Protocol

As mentioned in section 1 The Spray and Wait protocol is characterized by its combination of two main phases:

Spray Phase: In this phase, each node that has a message (bundle) spreads or "sprays" a limited number of copies of the message to other nodes it encounters. The goal is to increase the probability that the message will eventually reach the destination node by distributing copies throughout the network.

Wait Phase: After the spraying phase, the node that holds the message transitions into the wait phase. During this phase, it attempts to deliver the message directly to the destination node when it encounters it. The node that has the message does not spray any more copies but waits for the destination node to come into contact.Detailed Operation of each phase is given in the following sections.

6.4..1. Spray Phase

- **Initialization:** When a source node generates a message, it creates a fixed number of copies to be spread throughout the network.
- **Spraying Copies:** Each node that receives a message during the spray phase will distribute a portion of the remaining copies to other nodes it encounters. The number of copies to be sprayed is determined based on predefined rules or thresholds.
- **Spray Limit:** The spraying phase is constrained by the total number of copies. Once the total number of sprayed copies reaches the specified limit, the node stops spraying and moves to the wait phase.

6.4..2. Wait Phase

- **Direct Delivery Attempts:** After the spraying phase is complete, the node holding the message will attempt to deliver it directly to the destination node when it encounters it. If the destination node is not reachable, the node continues to wait for further opportunities.
- **Message Holding:** During the wait phase, only the nodes with remaining copies of the message can attempt to forward it. The protocol assumes that the destination node is eventually encountered, and the message is delivered upon such an encounter.

- **Limitations and Challenges**

- **Message Duplication:** There is a risk of unnecessary message duplication if nodes with copies encounter each other multiple times.
- **Delay:** The wait phase introduces additional delay, as messages are held until the destination node is encountered.
- **Performance Dependence:** The performance of the Spray and Wait protocol is sensitive to the network's mobility patterns and the number of copies used. Too few copies may result in low delivery probability, while too many copies may lead to excessive resource consumption.

- **Variants and Extensions**

Several variants and extensions of the Spray and Wait protocol have been proposed to address its limitations and adapt it to specific DTN scenarios:

- **Adaptive Spray and Wait:** Adapts the number of copies and spraying strategy based on network conditions or node density.
- **Probabilistic Spray and Wait:** Uses probabilistic methods to determine when and how many copies to spray, aiming to balance between delivery probability and resource usage.
- **Hybrid Approaches:** Combines Spray and Wait with other routing techniques, such as direct delivery or epidemic-based approaches, to enhance performance in various network scenarios.

- **Bandwidth-Delay Product**

The measures of how effectively a protocol uses available bandwidth in the presence of delay are referred to as **Bandwidth-Delay Product (BDP)**. It is calculated as:

$$BDP = Round\ Trip\ Time\ (RTT) * Connection\ Bandwidth \qquad \textbf{(5)}$$

As delays increase, TCP becomes less efficient, wasting bandwidth, especially in networks subject to frequent disconnections like ICNs, where repeated failed handshakes are likely.

In this work, "delay" refers to the end-to-end transmission time required for data delivery, which is influenced by three key factors:

- **Characteristics nature of the medium**: The media and its characteristics play a very significant role on which frequencies may travel through and thus determine rate and delay.
- **System geometry**: Factors like long distances between nodes.
- **Temporary packet storage**: Packets may be held in nodes' buffers while in transit from the source to the destination.

In some DTN contexts, the term “disruption” and “delay” are often used interchangeably. Disruptions are defined as frequent and erratic connection failures caused by momentary changes in system characteristics. These may include power failure, positioning errors, temporary disturbance, and changes in network topology etc. The following section focuses specifically on the impact of mobility.

- **Mobility and Storage**

In most DTN applications, movement of nodes is either randomly or deterministically across a given area. This feature has both benefits and drawbacks. On one hand, mobility aids message delivery by bringing nodes closer together, allowing them to forward and/or receive data packets. In DTNs, these data packets or "bundles," which encapsulate routing and delivery related information. Bundling ensures that semantically related data is transmitted as one complete message rather than split into smaller packets, avoiding issues if one packet is lost or delayed (64).

However, mobility also creates constant shifts in network topology, with links between nodes frequently appearing and disappearing. This mobility also imposes energy constraints. Consequently, bundles may need to be forwarded immediately but would require to be buffered for some duration by intermediate nodes, adding to the overall delay. Intermediate nodes must exploit their mobility by exchanging packets as they encounter other nodes to move the data closer to its destination.

In general, routers require substantial storage for several reasons intermittent connectivity for the next hop.

- **Asymmetry**: One node may transmit data faster or more reliably than its counterpart.
- **Re-transmission needs** due to errors or recipient failures.

Buffering may last for extended period depending on the specific application. However, buffer overload is a significant issue as it drastically increases packet drop rates. As a result, buffer management schemes have become a key area of DTN research, which will be discussed in a later section.

Additionally, understanding node mobility patterns is essential for optimizing performance and routing in DTNs. Several mobility models are prevalent and are used to simulate the movement of mobile nodes. However, mobility modelling is beyond the scope of this work.

6.5. Proposed approach

The proposed approach OSNW modifies the buffer capacity usage. The resource efficiency introduced would make this suitably useful for the IoT environment, especially low-power/memory sensor-based scenarios. Two new variables represented by **'Eligibility'** of transfer & **'Priority'** in the buffer queue are introduced. The queue is being managed as per 'Priority'. Therefore, the proposed approach would be referred onwards as ”Optimized Spray And Wait” or abbreviated to ”OSNW”. The same is presented in (65).

The OSNW routing protocol has modifications to both the Spray and Wait phases. In the first phase, the message is 'Sprayed' thereby being replicated into a limited number of copies (as SNW is a limited-replications method) and then forwarded to several neighbour nodes. These nodes then in turn also engage in further spraying of the message, in a tree-like fashion. Different variations use different types of spraying and replication mechanisms.

After spraying the Wait phase begins. In this phase, the nodes wait till the message is delivered to its destination. If it fails to be delivered to the destination, the protocols switch into a direct delivery routing approach and the message is delivered directly. So, this protocol combines the features of having higher speed and simplicity as it combines epidemic routing and the direct delivery routing protocol.

The performance assessment and scope of improvement/modification of any routing protocol are dependent on the delay or latency, that it takes to deliver packets correctly and the number of copies required. As per this the work can be divided into two parts that can be modified separately or together:

A. **Scheduling part:** The strategy or way of sending messages to the next node(s)

B. **Buffer Queue Management part:** The strategy or way of deciding which message to delete from the messages in the buffer queue.

The changes in either of these strategies shall optimise the performance of the protocol in significant ways. A variety of scheduling methods are used, ProPHET uses the history of encounters (54) with other nodes, a statistical property of two parameters delivery predictability and transitivity. And the standard buffer management method is First-In-First-Out.

In this thesis, the proposed algorithm suggests a modification to both of these two strategies to implement a buffer-adapted variation. The proposed modifications are as follows:

- **Scheduling Strategy:** The proposed algorithm attempts reduction of flooding of the network by application of conditional new parameters. If the destination node satisfies these conditions, then the source node will forward the message further. In case the condition is not met the message will not be forwarded to neighbour node. The primary condition is that the receiving node is not the destination node, as destination consumes the message and does not need to forward or schedule messages. The first condition being met, the intermediary node must have at least two connections to forward the message further. The second condition considers the distance from the sender node should be minimum or the buffer load be minimum.. This ensures that a local minima is achieved and minimal replication takes place, thereby reducing flooding. As it is mathematically proven that shortest distance is inversely proportional to the delivery probability. Likewise, the buffer load is also inversely proportional to the probability of the message being dropped.

$$ED_{best} \propto \frac{1}{distance} \tag{6}$$

$$ED_{best} \propto \frac{1}{buffer\ load} \tag{7}$$

$$ED_{best} \propto \frac{1}{distance} * \frac{1}{buffer\ load+1} \quad \textbf{(8)}$$

Consequently the sender will calculate the 'ED_{best} or Best Delivery Eligibility' for each intermediate node in the path and shall forward the message to only the two optimal neighbours. And following the same principle these node would pass the message onto two most suitable connections with highest eligibility value. The algorithm for calculation of ED_{best} is as follows:

Algorithm 1: Scheduling Process for calculation of ED_{best}

```
Input: $ED_{initial}$, initialisation of Eligibility for a node in the Neighbour circle
S = $\{s_i \mid 1<i< n\}$, set of all nodes in the network
$E_i = \{s_k \mid n_{ik} \neq 0, 1 \leq k \leq n\}$, set of nodes encountered by node $s_i$,
Initialize $ED_{initial} = Null$, for node $S_i$;
1: if $s_i$, Encountered $s_j$, and $s_j \notin E_i$, then
2:     $E_i = E_i \cup \{s_j\}$
3 if $s_j \notin ED_{best}$ then
4:         $D_{(i,j)} = | E_i \cap E_j |$
5      if $D_{(i,j)} > ED_{threshold}$ then
6:             $ED_{best} = ED_{initial} \cup \{s_j\}$
7.         End if
8:     End if
9: End if
```

- **Queue Management Strategy:** Most routing protocols in DTN follow the FIFO mechanism. This is to say that any message in the buffer queue would be processed in the FIFO order for scheduling to the next eligible node. Once the buffer is full any newly arriving messages will be stored and the oldest message in the buffer queue would be deleted and discarded regardless.

The proposed work applies three new strategies to utilise the space in the buffer queue adaptively based on a new variable '$P_{Transfer}$ or Transfer Priority' message and compares them with 3 of the most applied protocols.

$$P_{transfer}(x,y) = P_{transfer}(x,y)_{old} + \left(1 - P_{transfer}(x,y)_{old}\right) \times P_{transfer_{init}} \qquad \textbf{(9)}$$

$$P_{transfer}(x,y) = P_{transfer}(x,y)_{old} \times \gamma^{k} \qquad \textbf{(10)}$$

Where γ is an ageing constant (54)and $\gamma \in [0, 1]$ where an ageing factor k is introduces ro account for the time elapsed from the last delivery to that destination.

The rationale behind the proposed strategy is to identify the optimal path among the various possible paths. This traversal problem can be modelled as a resource, in this case, buffer, allocation model in which an incoming or existing packet will be discarded or assigned buffer space, to maximize the probability of the packet reaching the destination node. This is a combinatorial optimization, similar to a knapsack problem with a single limitation, which is an NP-hard problem. And hence an optimal solution can be obtained from a large set of possible solutions.

Due to the mobility, low energy and space requirement of sensors and devices in IoT, network routes are unpredictable and unstable and therefore delay tolerance is required. Therefore, instead of solving the end-to-end problem, the routing is converted into a set of simpler sub-problems. The optimal substructure ensures an overall optimal solution. The pseudo-code for the proposed strategy is given below as algorithm 2 followed by a detailed explanation of the same:

Algorithm 2: The Queue Management Strategy for calculation of $P_{transfer}$

Input: S = {s_i | 1<i< n}, set of all nodes in the network

$R_i(m_k)$, the replicas of message m_k carried by node s_i,

NC_d, the Neighbour circle of the destination

nNodes, the number of neighbour nodes in S_i's transmission range

SM_i, set of messages in queue carried by s_i,

SM_j, set of messages in queue carried by s_j,

Initialize $P_{transfer}$ = *Null*

```
if $s_i$ encounters $s_j$ and $R_i(m_k))$ > 1 and $s_j$ $NC_d$ then
    update $P_{transfer(i,j)}$ and $NC_i$
    $SM = SM_i \cap SM_j$,
    for each message $m_k$ in SM do
        $s_d = m_k$'s destination
        if $s_d == s_j$ then
            $s_i$ directly forwards $m_k$ to $s_j$
        else if $P_{transfer(i,d)} < P_{transfer(j,d)}$ or nodes < 2 then
            $R_j(m_k) = \lfloor R_i(m_k)/2 \rfloor$
            add $L_j(m_k)$ copies of $m_k$ to $P_{transfer}$
        end if
    end for
    if $P_{transfer}$ != *Null* then
        sort $P_{transfer}$ in ascending order of TTL
        $s_i$ forward $P_{transfer}$ to $s_j$
    end if
end if
```

The first strategy involves removing deleting messages with recently encountered destinations. This refers to the message whose destination node was most recently encountered. This approach addresses the issue of redundant flooding, where nodes repeatedly transmit the same messages stored in their buffers. By deleting the message for the most recently encountered destination, the algorithm ensures that the node does not continuously forward identical messages to the same or other nodes, optimizing buffer utilization.

In the second strategy, the algorithm prioritizes deleting messages with least recently encountered destinations. This means that messages whose destination nodes have been encountered the least recently would be deleted. The rationale is that these messages are less likely to be delivered promptly, as the node has not interacted with the intended recipient for a considerable amount of time. If multiple messages have destinations with the same "least recently encountered" status, the protocol employs a First In, First Out

(FIFO) mechanism, removing the oldest message in the buffer. This ensures that buffer space is efficiently managed without prioritizing messages with equal encounter histories.

The third strategy involves deleting messages with farthest destinations. This requires deleting messages whose destination nodes are geographically farthest from the current node. Since the likelihood of successful delivery decreases as the distance between the source and destination increases, this approach maximizes the utility of limited resources. By focusing on messages with a higher probability of successful delivery, the algorithm not only reduces overhead but also improves overall delivery efficiency. This strategy is particularly valuable in Delay Tolerant Networks (DTNs), where resource constraints like buffer space and energy are critical factors.

These strategies collectively aim to enhance buffer management, reduce redundancy, and increase message delivery probability in resource-constrained DTN environments.

Delivery probability table are maintained which show the probability of a message getting delivered from sender to receiver in the last wait cycle. The nodes exchange their delivery predictability table upon encounter with each other and update their delivery probability table. The law of transitivity is also followed, a node A frequently encounters node B which frequently encounters node C then, it can be concluded that node C is a good relay to deliver the message coming from node A. As each node calculates the delivery predictability for all known destination nodes where, $P(x, y) \in [0, 1]$. To calculate delivery predictability where a node encounters another node:

This assumption the protocols makes is that bandwidth available is unlimited, and the total time taken to deliver messages is not taken into consideration. The transitivity property also reduces the rate and probability of messages being dropped, this also helps in reducing the wait time and length of message queue of each node. This also ensures reduction in load and pressure on individual nodes.

6.6. Simulation Environment

This Simulation of DTN routing protocol cannot be efficiently performed by tools used for traditional networks. The DTN routing protocol requires node and route characteristics that deal with mobility, intermittent connectivity and resource constraints.

One of the most prevalent tools is the Opportunistic Network Environment (ONE) simulator. This software facilitates the modelling of different scenarios for existing and new DTN routing protocols. It is a powerful tool that allows recreating and testing of Epidemic, Spray and Wait, MaxProp, Rapid and ProPHET very easily. The ONE simulator is specifically designed for the investigation, comparison and evaluation of various DTN routing protocols.

- **Advantages and evaluation parameters of ONE Simulator**

The ONE (Opportunistic Network Environment) simulator stands out as an ideal tool for evaluating DTN routing protocols due to its specialized design for simulating delay-tolerant network scenarios. This thesis leverages the ONE simulator to implement and compare new and existing DTN routing protocols effectively.

The ONE simulator is tailored to assess DTN routing protocols, taking into account critical factors like mobile node movement, node density, and the distance between sender and receiver. These factors influence key performance metrics, such as latency, delivery probability, and overhead ratio, providing a comprehensive evaluation environment.

The simulator's primary goal is to identify suitable routing and forwarding approaches while aligning them with real-time mobility scenarios. It allows researchers to test how protocols perform under dynamic and realistic network conditions.

The ONE simulator is open-source, providing flexibility for researchers to edit, execute, and develop programs. It supports result visualization and detailed analysis, making it a valuable tool for academic and professional research. The simulator requires Eclipse IDE for Java Developers (v.2020-06) for effective execution. It supports compilation on both Windows and Eclipse IDE environments, offering versatility in deployment.

The ONE simulator has advanced simulation features, like generation of detailed mobility traces to simulate node movement patterns. It also supports running DTN messages and simulating their behaviour across the network. The simulator offers a graphical visualization of simulations, along with comprehensive logs for analyzing execution results.

In summary, the ONE simulator's robust feature set, adaptability, and focus on DTN protocol evaluation make it an indispensable tool for researching and testing innovative routing techniques in delay-tolerant networks. (66) (67)

- **Metrics of Performance & Simulation parameters**

Several factors are commonly utilized for the assessment of the performance of DTN routing protocols: overhead ratio, packet delivery ratio, average latency, and average hop count. These metrics are described as follows:

- **Overhead Ratio:** One of the most important metrics used to assess the performance of DTN routing protocol Overhead Ratio. It can be defined as the number of duplicate packets that are required to be transmitted to ensure successful delivery. The overhead ratio provides a measure of the network congestion status, which is used to determine the bandwidth required and the number of successful replications required for the packet delivery. It is given by Eq. (11):

$$OR = (R - D)/D \tag{11}$$

Where R is the number of successful transmissions and D refers to the number of messages delivered to the destination.

- **Delivery Probability:** The delivery probability is yet another important metric to assess the performance of any DTN routing protocol. It is a measure of the ratio of the actual number of packets delivered to the destination and the actual number of packets sent from the source node. A high packet delivery ratio signifies less loss and thus better performance of the network It is calculated as is given by Eq. (12):

$$PDR = DM/CM \qquad (12)$$

Where DM is the number of successfully delivered messages, and CM is the number of created messages.

- **Average Latency:** Average latency is defined as the time elapsed from the time the message is sent from the source node to the time it is delivered at the destination node. That is to say, it is the average time taken by the message to be created by the source node till the time it is received by the destination node .
- **Number of Hops:** The hop account is defined as the number of nodes that a message has been sent to thus far. If the message is created at a node, the hop count is calculated as zero. That is to say, the hop account indicates the number of hops that the message makes between the source node and the destination node

The simulation parameters considered for analysis in the DTN routing protocols are summarized in Table 2

Table 8 List of simulation parameters

Parameters	**Values**
Simulation Area	4500*3400
Simulation Time	43200
Mobility Model	Shortest Path Map-Based Movement
TTL	300
Buffer Size	5MB
Transmission Range	10
No. of Nodes	126
Bundle creation rate	25 to 35 seconds
Bundle size	500kB - 1MB

6.7. Results and Discussion

The results of the ONE simulation have been generated through reports that are created by report modules during the run time of the simulation. The Simulation engine provides the data to the report modules for the run-time events, reports are then created based on these received results.

As suggested above, the routing protocols reports contain measures and values for different factors such as lateness, packet delivery ratio and bandwidth consumption.

There are two ways to visualize the simulation results in the ONE simulator: By generating images from the information gathered during the simulation, and via an interactive GUI. The simulator has a graphical user interface (GUI) that is launched with Java. It is possible to zoom in and out and to change the speed by using the GUI update icon in this playfield graphics in the ONE simulator. The playfield graphics has various buttons and icons to play the simulation step forward internally, enable and disable fast forward, and play the simulation for a specific time.

Consequently, This thesis has conducted a comparative analysis by pitting our implemented routing protocols against Epidemic and ProPHET. The results demonstrate that the proposed protocol surpasses Epidemic, ProPHET, and Stop-and-Wait routing protocols in several key performance metrics. The assessment was based on parameters such as delivery probability, overhead, the number of hops, and latency.

The simulation results and subsequent discussion present a set of comparative graphs showcasing the performance of the three existing protocols in comparison to the proposed 'OSNW' approach.

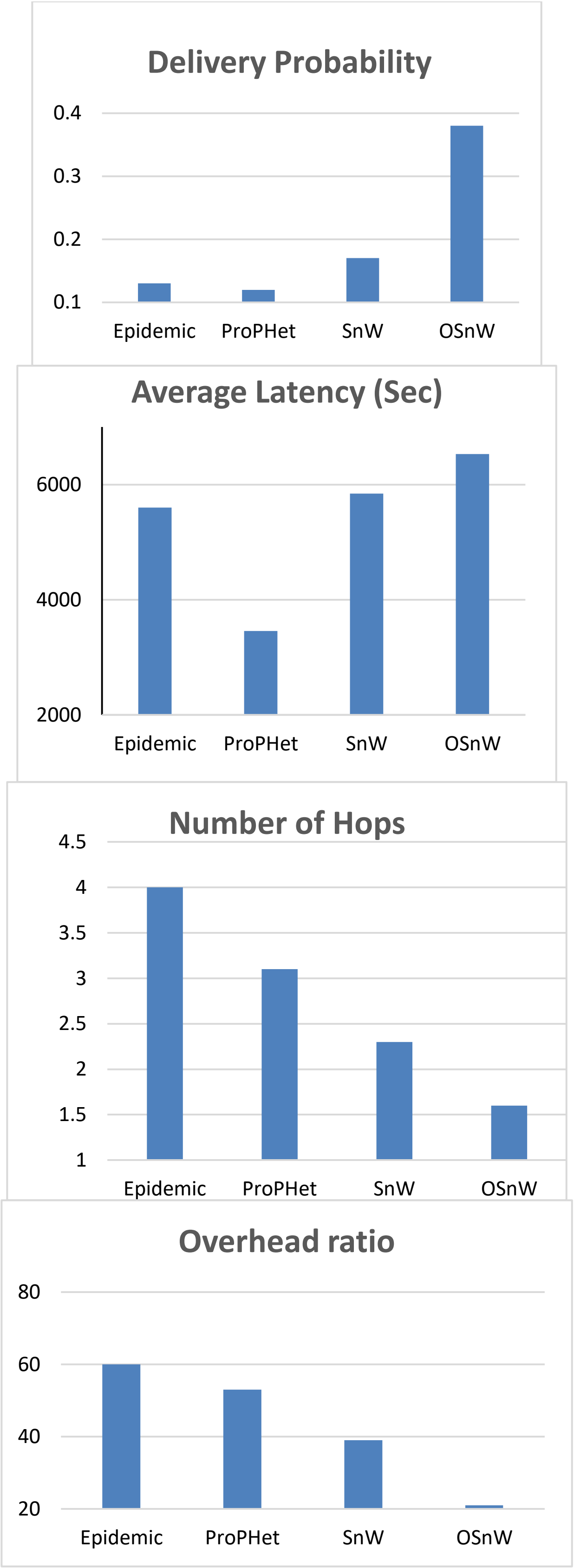


Figure 2 Comparative graphs for Epidemic, ProPHet and SNW protocols with the proposed approach 'OSNW'

As evident from the graphs for all 4-performance metrics, the proposed approach outperforms the existing protocols.   It can be noted that for most parameters Epidemic and Spray and Wait protocols offer quite similar results, which are considered the optimum compared with the other protocols. In contrast, the PRoPHET behaves differently and represents the latency due to its history requirement.

The number of hops also consistently improves among the four routing protocols for the same set of simulation parameters. Thereby, reflecting the overall superiority of the proposed approach. Spray and Wait tailored to the specific needs of IoT applications, with careful consideration of factors like energy efficiency, network overhead, and latency enhances the protocol’s adaptability in dynamic environments, optimizing message spraying techniques, and integrating buffer management strategies to further improve its performance in resource-constrained settings.

7. EA-RPL: A DELAY-DISRUPTION TOLERANT APPROACH FOR RPL-BASED IOT NETWORKS AGAINST INIMICAL ATTACKS

This chapter explains the proposed method for mitigating the security risks posed by DIS flooding and Version number attacks in IoT using a mechanism used by DTN based networks. Particularly an Anomaly based detection engine with statistical outlier detection based on energy consumption patterns. This will lead to potential theoretical solutions or, where feasible, practical implementations to mitigate attacks.

7.1. Introduction

Routing Protocols for Low Power and Lossy Networks (RPL) was conceived with the primary objective of bridging low-power sensor nodes with IoT networks. RPL's design philosophy centers around simplicity and flexibility, making it compatible with a wide range of resource-constrained devices. Consequently, this empowers a plethora of applications spanning across multi-hop mesh networks, encompassing industrial, urban, and domestic settings. RPL optimizes the utilization of smart device energy, establishes adaptable network topologies, and ensures efficient data routing.

7.2. Attacks on RPL (Routing Protocol for Low-Power and Lossy Networks)

Despite its numerous advantages, RPL faces vulnerability to a variety of attacks, primarily classified into three categories: attacks on resources, traffic, and network topology. Safeguarding the security of RPL-based sensor IoT networks presents a formidable challenge. This research seeks to delve into the existing body of literature pertaining to these attacks and identify gaps in research, particularly in the realm of mitigating the security risks posed by DIS flooding and Version number attacks in IoT applications (60).

RPL is a routing protocol designed for low-power and lossy networks, such as those found in IoT and sensor networks. Given its deployment in environments where resources are constrained and network conditions are challenging, RPL is vulnerable to various attacks. These attacks can undermine the integrity, availability, and confidentiality of the network. Here’s an overview of common attacks on RPL:

7.2..1. Wormhole Attack

- **Description:** An attacker creates a tunnel between two distant points in the network, misleading nodes into believing that they are closer to the destination than they actually are.
- **Impact:** Disrupts routing by causing packets to be routed through the wormhole, leading to increased latency and potential data loss.

7.2..2. Sybil Attack

- **Description:** An attacker creates multiple fake identities (nodes) in the network to manipulate routing decisions or inject malicious traffic.
- **Impact:** Can lead to network congestion, incorrect routing information, and loss of network resources. It also compromises the network's integrity.

7.2..3. Black Hole Attack

- **Description:** An attacker nodes claims to have the best route to the destination and then discards all incoming packets.
- **Impact:** Causes data loss as packets sent to the attacker’s node are dropped.

7.2..4. DoS (Denial of Service) Attack

- **Description:** An attacker floods the network with excessive traffic or disrupts routing operations to prevent legitimate communication.
- **Impact:** Leads to network congestion, increased packet loss, and degraded network performance.

7.2..5. Replay Attack

- **Description:** An attacker captures and replays valid routing messages or data packets to deceive the network or disrupt its operation.
- **Impact:** Can cause confusion in routing tables, data duplication, or the re-routing of packets to unintended destinations.

7.2..6. Selective Forwarding Attack

- **Description:** An attacker selectively forwards or drops packets to disrupt communication or degrade network performance.
- **Impact:** Results in data loss and reduced network reliability.

7.2..7. RPL-Specific Attacks

- **Version Number Attack:** Exploits the version numbers in RPL control messages to create inconsistencies and cause nodes to incorrectly update their routing tables.
- **Rank Attack:** Manipulates the rank of a node to affect routing decisions, causing traffic to be misrouted or leading to network instability.

7.3. Impact on RPL Networks

- **Integrity Compromise:** Attacks can corrupt or manipulate routing information, leading to incorrect routing paths and data loss.
- **Availability Issues:** Denial of service and selective forwarding attacks can degrade network performance, making it difficult for legitimate nodes to communicate.
- **Confidentiality Threats:** Certain attacks, such as replay attacks, can expose sensitive information by reusing valid data packets.

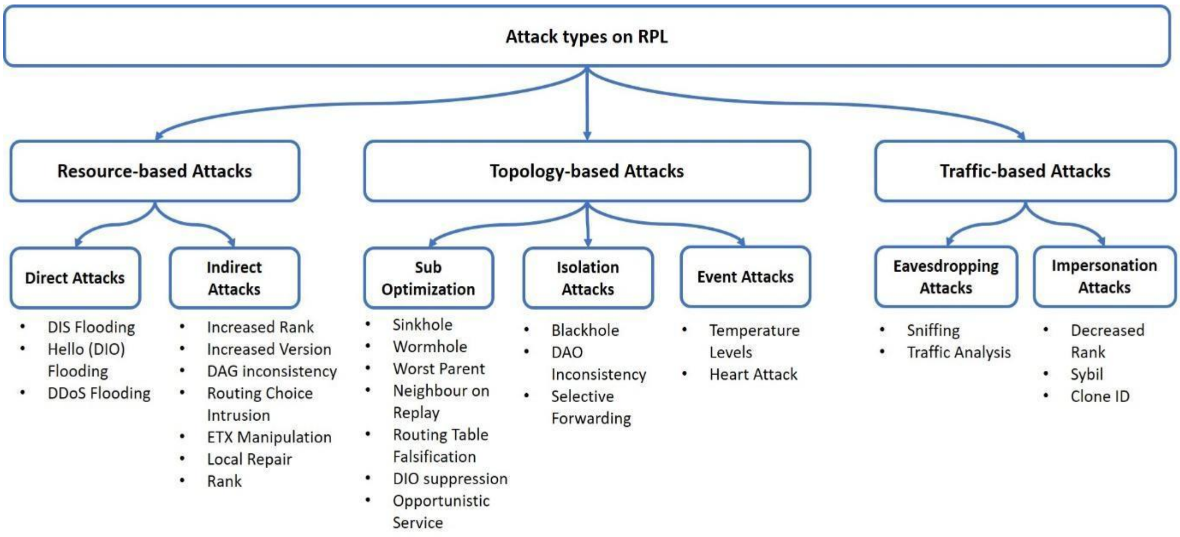


Figure 3 Classification of attacks on RPL

7.4. Mitigation Strategies

- **Cryptographic Techniques**

- **Encryption:** Protects the confidentiality of data transmitted across the network.
- **Authentication:** Ensures that routing messages come from legitimate sources and are not tampered with.

- **Secure Routing Protocols**

- **Secure RPL Extensions:** Implementing extensions to RPL that include security features to protect against specific attacks.

- **Monitoring and Detection**

- **Anomaly Detection:** Identifies unusual patterns or behaviours in the network that may indicate an attack.
- **Intrusion Detection Systems (IDS):** Monitors network traffic for signs of malicious activity.

- **Redundancy and Robustness**

- **Path Diversity:** Using multiple routing paths to reduce the impact of a single attack.
- **Redundant Nodes:** Implementing redundancy to ensure network resilience and recovery in case of node compromise.

- **Protocol Enhancements**

- **Rank Validation:** Ensuring that the rank values in RPL messages are valid and not manipulated.
- **Version Control:** Implementing mechanisms to validate the version numbers and prevent version number attacks.

RPL networks are susceptible to a range of attacks that can affect their performance, security, and reliability. Understanding these attacks and implementing appropriate mitigation strategies is crucial for maintaining the integrity and functionality of low-power and lossy networks. By incorporating secure routing protocols, cryptographic techniques, and monitoring systems, the impact of these attacks can be significantly reduced.

7.5. Anomaly Detection Engine Based on Outlier

Anomaly detection is a crucial aspect of securing networks, including Delay Tolerant Networks (DTNs) and IoT systems. Anomaly detection engines based on outlier analysis focus on identifying patterns or data points that deviate significantly from the norm, which may indicate potential security threats or system malfunctions.

- **Overview of Outlier Analysis**

Outlier analysis involves identifying data points that differ markedly from the majority of the data in a given dataset. These outliers, or anomalies, may represent errors, unusual behaviour, or potential security threats. In the context of network security, outlier analysis can help in detecting attacks, malfunctions, or unauthorized activities.

- **Key Concepts**

- **Outlier:** A data point that significantly deviates from the expected range of values. In network data, this might be unusually high or low traffic, unexpected routing behaviours, or unusual patterns of node activity.
- **Anomaly Detection:** The process of identifying outliers or deviations from expected patterns in data.

7.5..1. Types of Anomalies

- **Point Anomalies:** Individual data points that are significantly different from the rest. For instance, a node transmitting an unusually high volume of data.
- **Contextual Anomalies:** Data points that are anomalous in a specific context but may be normal in a broader context. For example, a spike in traffic might be normal during certain times but anomalous during off-peak hours.
- **Collective Anomalies:** A group of data points that together form an anomaly, even if individual points may not be unusual on their own. For instance, a sudden shift in the traffic pattern of multiple nodes.

7.5..2. Techniques for Outlier Analysis

7.5..2.1.Statistical Methods

- **Z-Score:** Measures how many standard deviations a data point is from the mean. Data points with a high z-score are considered outliers.
- **Box Plot Analysis:** Uses quartiles and interquartile ranges to identify outliers. Data points falling outside the whiskers of a box plot are considered anomalies.

*7.5..2.2.*Machine Learning Methods

- **Clustering-Based Methods:** Techniques like K-Means or DBSCAN group data points and identify those that do not fit well into any cluster as outliers.
- **Classification-Based Methods:** Supervised learning algorithms like Support Vector Machines (SVM) or Neural Networks can be trained to distinguish between normal and anomalous data based on labelled training data.
- **Ensemble Methods:** Combine multiple anomaly detection algorithms to improve robustness and accuracy. Examples include Isolation Forest and Random Cut Forest.

7.5..2.3.Distance-Based Methods

- **k-Nearest Neighbours (k-NN):** Measures the distance between data points. Data points with large distances from their neighbours are considered anomalies.
- **Local Outlier Factor (LOF):** Evaluates the density of data points relative to their neighbours. Points with significantly lower density are identified as outliers.

7.5..2.4.Model-Based Methods

- **Autoencoders:** Use neural network architectures to learn a compact representation of the data. Reconstruction errors are used to detect anomalies.
- **Hidden Markov Models (HMMs):** Model temporal sequences and identify deviations from the expected sequence patterns.

7.6. Application in Network Security

a) ***Monitoring Network Traffic***: Detect unusual traffic patterns, such as spikes in data volume or unexpected routing changes, which may indicate attacks like DoS or unauthorized access.

b) ***Identifying Malicious Nodes:*** Spot nodes that exhibit anomalous behaviour, such as sending or receiving abnormal amounts of data or deviating from typical communication patterns.

c) ***Anomaly Detection in Routing:*** Detect routing anomalies such as unexpected changes in path metrics or node rankings that may indicate routing attacks or misconfigurations.

7.7. Challenges and Considerations

a) **High Dimensionality:** Many network datasets are high-dimensional, making it challenging to detect outliers effectively. Dimensionality reduction techniques like PCA (Principal Component Analysis) can help address this issue.

b) **Adaptive Behaviour:** Attackers may adapt their strategies to avoid detection. Anomaly detection systems need to continuously learn and update their models to stay effective.

c) **False Positives and Negatives :** Balancing sensitivity and specificity is crucial to avoid false alarms and missed threats. Anomaly detection systems must be fine-tuned to achieve an optimal balance.

d) **Scalability:** In large-scale networks, the anomaly detection system must efficiently handle vast amounts of data and perform real-time analysis

7.8. Proposed framework

The proposed EA-RPL approach consists of two conceptual modules,:

●**Part A** for the calculation of rate of drain of energy by residual energy measurement which is DTN inspired Dead Node identification method and then
●**Part B** for anomaly detection based on outlier analysis to identify intrusion for mitigation of attack. The features of EA-RPL specifications are used to check for the power consumption pattern and detect anomaly.

- **Energy efficient Dead node identification in DTN**

DTN are resource constrained and messages must be transmitted to nodes having high encountering ability, for maximising efficiency and lifetime. Hence, any message transmission to a destination which is about to be dead, causes wastage of network resources because any messages destined to or passing via dead/nearly dead node will never reach destinations. Also any Node with high battery drainage will die early. The delay tolerant network transmits multiple copies of each message that lead to higher resource consumption.

The objective function for DTN is optimised and controlled by forwarding the messages via nodes with high encountering ability. This will ensure intermediate nodes lose energy only in transmission and reception. This requires that for Energy Efficiency messages must avoid inactive and dead nodes. Dead node will not be able to forward any messages to their destinations.

Authors propose an anomaly detection mechanism based on a routing protocol for Delay Tolerant Network based on Optimising the Objective Function based on Destination to Dead node, which is a function of connectivity and delay in encounter time. The same principle may be applied to server nodes under attack the delay in processing is the Objective function of CPU power and Transmission Power,

- **Anomaly detection engine based Outlier analysis**

The methodology introduced comprises several key elements, including hypothesis formation, testing, observation, and drawing conclusions. The initial hypothesis posits that in a network employing RPL, nodes may exhibit abnormal power consumption patterns when subjected to attacks. Such nodes will display a distinctive power drainage behaviour that significantly deviates from typical observations, raising suspicions of a distinct underlying cause. These exceptional observations are commonly referred to as outliers, representing data points that appear inconsistent with the rest of the dataset.

Depending on the information scope considered for outlier detection, outliers can be categorized as either global or local. In the experimental results that follow, the anomalies are revealed to be local outliers. Detecting and eliminating local outliers help reduce the communication overhead, subsequently leading to a reduction in overall energy consumption.

It's worth noting that outliers can stem from various sources, including noise, errors, and other events, not just malicious attacks. However, this research primarily focuses on outliers induced by malicious attacks, specifically concerning network security.

Statistical-based methods offer a straightforward approach to handle outlier detection. These techniques are model-based and prove particularly effective in sensor-based networks and Low-Power and Lossy Networks (LLN) due to their minimal overhead. In this context, the Mean and Standard Deviation Method serves as the predictor. The detection of DIS Flooding attacks and Version Attacks is accomplished using statistical outlier analysis. This process involves the simulation of the EA-RPL Protocol, the collection of power-related data, and the computation of Power Consumption and Radio Duty Cycle percentages for different nodes. The calculations for both parameters adhere to the following formulas:

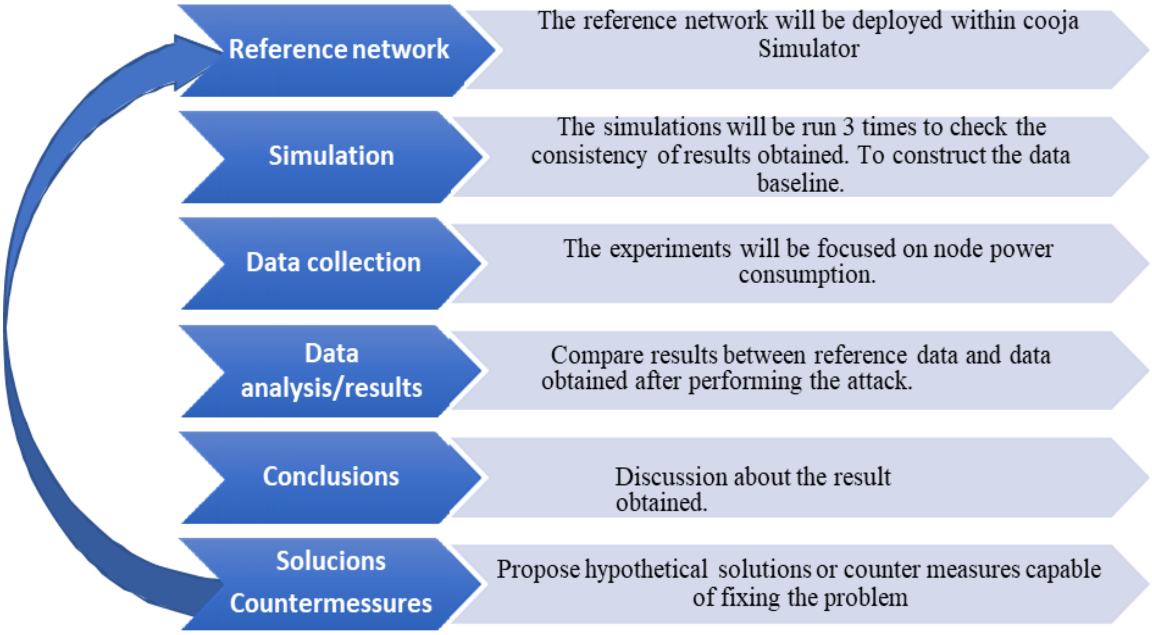
Energy consumption (Power in mW) represented as E may be defined as given in equation 1

$$E = \frac{Energest\_Value \times Current \times Voltage}{RTIMER\_SECOND \times Runtime} \tag{13}$$

Radio Duty Cycle% of node represented as R is defined as

$$R = \frac{Energest\_TX + Energest\_RX}{Energest\_CPU + Energest\_LPM} \tag{14}$$

. Architectural flow for proposed method is given below.



The reference network will be deployed within cooja Simulator

The simulations will be run 3 times to check the consistency of results obtained. To construct the data baseline.

The experiments will be focused on node power consumption.

Compare results between reference data and data obtained after performing the attack.

Discussion about the result obtained.

Propose hypothetical solutions or counter measures capable of fixing the problem

Figure 4 Architectural flow for proposed method

To simulate the impact of an attack on the lower three layers of the UDP/IP/RPL protocol stack, TCP implementation is carried out using the Contiki-Cooja pair. These layers are particularly vulnerable in resource-constrained IoT nodes. Contiki OS, in conjunction with the Cooja simulator, is chosen due to its minimalist feature set that is well-suited for a complete TCP/UDP/IP/RPL stack [1].

Contiki, an operating system designed for memory-constrained and networked environments, is tailored for low-power wireless devices. The Collect View application, an integrated power profiler in Contiki OS, is employed for power parameter measurements. The required parameters include transmission time, receiving time, LPM power, and CPU power. Utilizing relevant formulas, the Power Consumption and Radio Duty Cycle percentages are calculated, and graphical representations are generated. Subsequently, an Outlier Analysis is conducted to identify both attackers and victim nodes.

The IoT scenario simulation utilizes RPL as the routing protocol for various network topologies. Power Consumption and Radio Duty Cycle percentages are computed for all nodes. The implementation methodology involves a comparison between the network under attack and a baseline reference network. Hence, it begins with the deployment of a reference network, followed by the deployment of one or more malicious nodes for simulation of given attacks on the network. This requires modification of codes in the RPL configuration files are in order to achieve simulation of actual attacks , these changes ensure that nodes mimic attacker behaviour.

The values of the reference network are measured through the Cooja GUI, and the power consumption data from the nodes is collected. Each attack is then deployed to demonstrate the changes in power consumption normal pattern of each nodes versus the power consumption pattern in the presence of an attack. Two types of motes are created in the reference network a sink node which generates traffic that mimics low power lossy Network Border Router (LBR) and DODAG router, and leaf motes which mimic wireless sensor collectors. To maintain a realistic simulation environment, a 100x100-meter area with randomly distributed motes is used. This ensures that the simulation closely mimics real-world conditions.

The objective of simulation of attacks is to modify the behaviour of a mote or motes only, these now act as attackers while any alteration in the normal behaviour of the remaining motes of the network. This allows the assessment of network reactions to irregular patterns or situations.

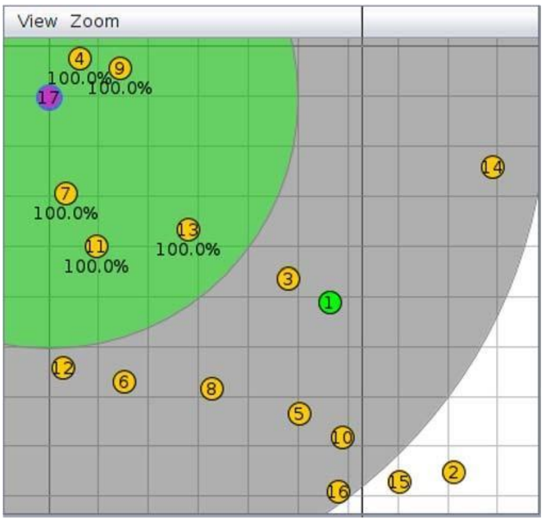


Figure 5 Reference Network with a Malicious Mote

The method used to achieve that approach can be accomplished by following the steps below:

• Create a duplicate folder to duplicate Contiki O.S. instance.

• Code modification in the corresponding files based on each attack being simulated.

•Create and add new malicious mote(s), followed by compilation in firmware to include them within the duplicate Contiki instance in the reference network.

• Run the network

The simulation parameters for testing EA-RPL are given in Table 9

Table 9 Simulation Parameters

Simulation Parameters	**Value**
Simulation tool	Contiki 3.0 & Cooja simulator
Mote type	Sky mote
Simulation run time	600 seconds

Number of leaf nodes	16
Total number of modes with malicious nodes	17
Radio Channel	UDGM6: Distance Loss
Range of Transmission	50m
Range of Interference	100 meters
Mote start delay	1,000 msec
Random seed	654321
Positioning	Random positioning
Sink node	1
Legitimate nodes	15
Malicious nodes	1

Table 10 Parameters obtained under DIS Flooding Attack.

Node	CPU Power	LPM Power	Listen Power	Transmitting Power	Total Power
2	0.429436921	0.047469827	1.064172012	1.116901283	2.957980043
3	0.419455896	0.04777203	1.242622192	0.704243921	2.714094039
4	1.76286332	0.007096638	20.52036859	1.987472321	24.57780087
5	0.458203595	0.046598836	1.108506478	0.967168285	2.880477194
6	0.569932865	0.043215922	1.841257247	1.55360238	4.308008415
7	1.955413487	0.001266647	20.98135045	2.553331238	25.79136183
8	0.558032383	0.043576242	1.704332116	1.507205219	4.11314596
9	2.149477821	-0.00460919	24.74559236	2.239792252	29.43025325
10	0.436049017	0.047269627	1.219687584	1.083416375	3.086422603
11	1.84191986	0.004702982	21.58966294	1.478212204	25.21449799
12	1.811479121	0.00562466	22.06001002	1.739849311	25.91696312
13	1.981403993	0.000479712	22.50831345	0.982049479	25.77224663
14	0.301297404	0.051349606	0.528188563	0.552141249	1.732976823
15	0.33814213	0.05023403	0.877077521	0.528442475	2.093896156
16	0.22433355	0.053679901	1.211768531	0.351706762	2.141488743

Table 11 Parameters obtained under Version Attack

Node	CPU Power	LPM Power	Listen Power	Transmitting Power	Total Power
2	0.502729	0.045251	1.043023	1.193798	3.0848
3	0.626128	0.041514	1.565977	1.25384	3.78746
4	0.674497	0.04005	1.798859	2.35655	5.169957
5	0.593996	0.042487	1.428769	1.160165	3.525417
6	0.639039	0.041124	1.829971	1.279761	4.089893
7	0.586793	0.042705	1.344965	1.431812	3.706276
8	0.655925	0.040612	1.832812	1.204027	4.033376
9	0.612355	0.041931	1.536302	2.012633	4.50322

10	0.555797	0.043644	1.301083	1.132513	3.333037
11	0.641452	0.04105	1.703722	1.599925	4.286149
12	0.610229	0.041996	1.434094	1.863687	4.250006
13	0.747708	0.037833	2.16769	1.701375	4.954605
14	0.565057	0.043364	1.228748	2.027283	4.164451
15	0.534679	0.044283	1.200595	1.092799	3.172357
16	0.502729	0.045251	1.043023	1.193798	3.0848

7.9. Results and analysis

A DIS attack is a direct assault targeting network resources, characterized as a flooding attack designed to overwhelm nodes and links by generating a significant volume of traffic. The objective of simulating this attack is to assess the effect on change, if any, on pattern and rate of power consumption experienced by leaf nodes when an attacker node initiates a DIS attack in a network. Subsequently, the examination of result values will lead to the exploration of potential theoretical solutions or, if feasible, the implementation of countermeasures against this specific attack.

Conversely, a version attack is an indirect offensive tactic in which a malicious node introduces a higher version number for a DODAG tree. Nodes receiving DIO messages with the new version number begin to establish a new DODAG tree. This attack disrupts the network by introducing inconsistencies and inefficiencies into its topology. The simulation aims to elucidate the consequences of leaf nodes repeatedly receiving DIO messages with higher versions and their impact on mote energy consumption. Furthermore, this investigation will explore potential theoretical solutions or, where feasible, practical implementations to mitigate this particular attack.

The graphical representation in Figures 6 and 7 illustrates the disparity between the average parameters of unaffected nodes and attacker nodes. The data clearly reveals that the radio duty cycle percentage for nodes under attack or acting as attackers is significantly higher than that of regular nodes. This outcome aligns with the initial theoretical hypothesis upon which the experimentation was based.

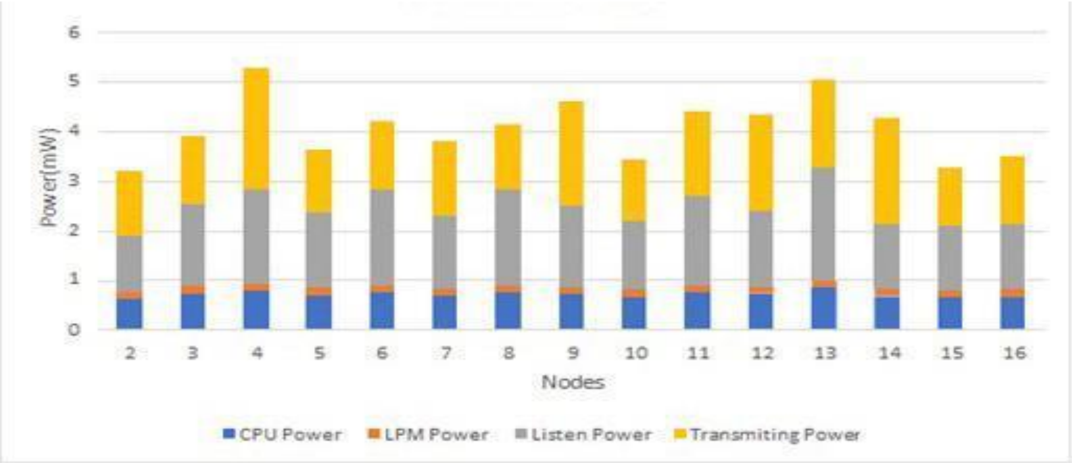


Figure 6 Nodes Consumption by CPU, LPM, Listen and Transmitting Power –DIS Flooding Attack

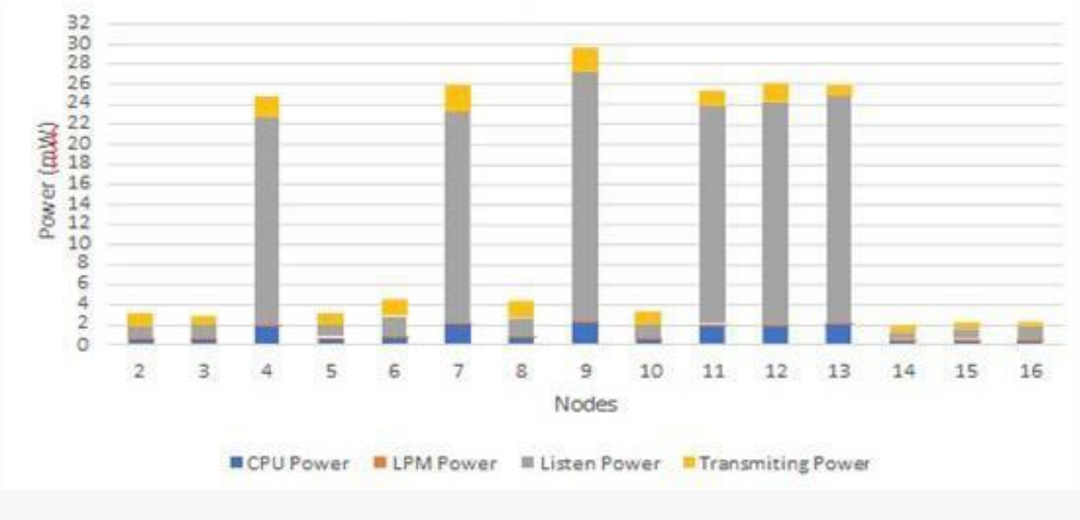


Figure 7 Nodes Consumption by CPU, LPM, Listen and Transmit Power –Version Attack

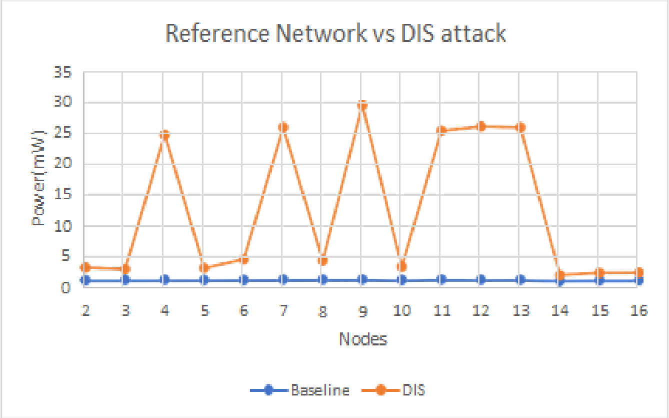


Figure 8 Comparison of Reference network vs DIS attack

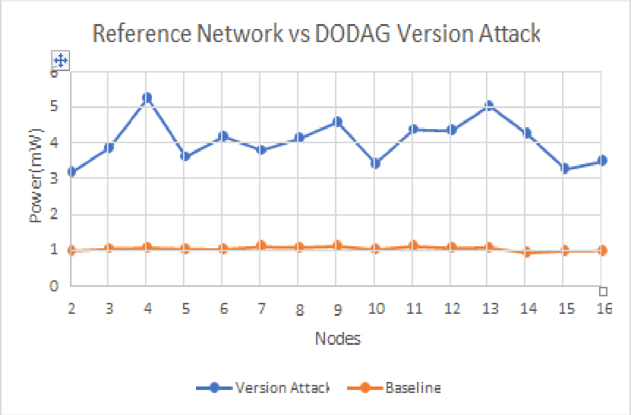


Figure 9 Comparision of Reference network vs Version attack

As can be seen in the fig 8 when an attacker is present or the server is under attack its duty cycle % is approximately 100%. This is an indication that it is active all the time during the simulation. The case for nodes far away is visibly different. The other node's duty cycle are falling in a much lower range. It is Firstly affecting the nodes in its range then all other nodes deployed in the region. This is due to request packets being generated in high number causing the flooding attack. It can also be concluded confidently from the above Fig 6 that the power consumption measured in Mw is also comparatively higher than the other nodes excluding server. So the rate of battery drainage rate for attacker is also much higher that rate of battery drainage for other nodes.

8. CONCLUSION

This chapter presents a comprehensive summary of the research work done. It includes the research summary of the work done, and the limitations of the research study identified. The chapter also presents the future aspects of the research work performed and how the study can help the future researchers in the said domain.

8.1. Introduction

Numerous recent surveys and research endeavours have delved into the application of Delay-Tolerant Network (DTN) routing protocols in traditional networks. However, only a limited number of these efforts have specifically tackled the challenge of enabling delay-tolerant Internet of Things (IoT). The proposition of protocols and solutions dedicated to DTN within the realm of IoT is a relatively recent development, offering substantial scope for further exploration and practical applications.

This thesis investigates and affirms the feasibility of adapting existing DTN architectures to suit IoT applications, while considering resource constraints and other related limitations. This adaptability is underscored by the superior performance of the proposed routing scheme.

The convergence of DTN and IoT solutions holds the potential to usher in the development of new and enhanced solutions catering to a wide range of existing and emerging IoT applications. To bridge this existing gap by presenting the design of an adaptive buffer DTN-based routing protocol, tailored to effectively facilitate a delay-tolerant IoT architecture and related applications.

The conclusion can be drawn in two parts for each of the approaches towards delay tolerant IoT.

8.2. Advantages of Optimized Spray and Wait

- **Reduced Overhead:** By limiting the number of copies and distributing them across the network, OSNW reduces the overhead compared to fully flooding the network with message copies.
- **Controlled Resource Utilization:** The protocol manages the trade-off between resource utilization (e.g., buffer space and energy consumption) and delivery probability by controlling the number of copies.
- **Improved Delivery Probability:** The spraying phase increases the likelihood of message delivery by distributing copies across different nodes, even if end-to-end connectivity is not available.

8.3. Applications and Use Cases

The OSNW protocol is suitable for applications where intermittent connectivity and large delays are expected. It is often used in:

- Mobile Networks: Scenarios involving mobile nodes with sporadic connectivity.
- Remote Sensor Networks: Networks with sensor nodes deployed in challenging environments.
- Interplanetary Networks: Space communications where end-to-end paths are not always feasible.
- Considerations while using OSNW

8.4. Routing scheme suitability

- The suitability of routing schemes varies widely based on application needs. For instance, direct delivery is less effective in harsh environments due to long delays.
- Random/Epidemic flooding is generally avoided except in extreme cases like military applications where all schemes are potentially used due to the critical nature of the data.
- Energy consumption and resource management are crucial factors. Flooding-based protocols consume significantly more power compared to history-based methods.
- The choice of routing scheme should be tailored to the specific application, considering factors like node mobility patterns, environmental conditions, and delay requirements. For example, life-monitoring applications may benefit from history-based or model-based routing schemes, while underwater and inter-planetary networks might require specialized approaches due to their unique challenges.

8.5. Advantages of EA-RPL

Based on the results derived from the implementation of the proposed approach, it is conclusively established that the initially assumed hypothesis is indeed valid. The simulations conducted within the Contiki-OS Cooja simulator confirm the alignment of deductions with the original hypothesis. This approach serves as a proof-of-concept identification method and offers a computationally efficient alternative for analyzing network parameters to detect and prevent various threats to the network. The significance of deriving this solution from delay-tolerant routing, despite its resource constraints, introduces several notable advantages:

a. Implementation does not necessitate the introduction of new parameters or fields. It relies on predetermined traits, such as the operation modality of RPL or anomaly-based actuators (alerting when power consumption exceeds a specific threshold or when certain message types exceed normal behaviour).

b. In terms of scalability and energy efficiency, the implementation is straightforward.

c. It can function as a Network-based Intrusion Detection System, allowing for the seamless addition of new nodes to the network without the need to install separately a host-based IDS individually on hosts.

d. This approach also mitigates potential incompatibilities (e.g., operating systems, firmware) since it requires no hardware modifications.

Moreover, this solution doesn't impact the performance or power consumption of nodes, eliminating the need for constant power supplies and avoiding added complexity due to encryption or authentication mechanisms, making it a lightweight and practical solution

8.6. Future Scope and challenge areas

While the Spray and Wait protocol offers several benefits for IoT systems, there are also some challenges to consider:

a) **Delay Tolerance**: IoT applications requiring real-time data (e.g., healthcare monitoring, autonomous vehicles) may not benefit from Spray and Wait due to the inherent delays in the **"wait"** phase. For applications with strict latency requirements, alternative protocols or hybrid solutions may be more suitable.

b) **Optimal Number of Copies (L)**: Selecting the optimal number of message copies to spray into the network is critical for balancing delivery probability and resource consumption. In dynamic IoT environments, determining this number can be challenging and may require adaptive mechanisms.

c) **Buffer Management**: IoT devices with limited memory may struggle to buffer multiple copies of messages, particularly when carrying messages for extended periods in the wait phase. Efficient buffer management techniques are essential to mitigate this issue.

The Spray and Wait routing protocol provides a robust solution for IoT networks deployed in environments characterized by intermittent connectivity and high mobility. Its controlled message replication mechanism helps ensure that data is delivered even when immediate connectivity is unavailable, making it particularly useful for applications such as smart agriculture, environmental monitoring, vehicular IoT, and disaster recovery. Further research should focus on optimizing routing schemes based on real-world deployment scenarios and resource constraints, potentially incorporating hybrid approaches that balance the strengths and weaknesses of various schemes.

The thesis systematically elucidates the points of convergence and intersection between infrastructure-based IoT and the seemingly infrastructure-less DTN. Both areas are comprehensively examined, taking into account their advantages and limitations. In conclusion, it can be confidently asserted that this represents a promising and intriguing area of exploration for research communities.

9 References

1. *A delay-tolerant network architecture for challenged internets.* **Fall, K.** 2003, pp. 27–34.

2. *Using delay tolerant network for the Internet of Things: Opportunities and challenges.* **F.Z. Benhamida, A. Bouabdellah, Y. Challal.** 2017, pp. 252–257.

3. *Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges.* **M.J. Khabbaz, C.M. Assi, W.F. Fawaz.** 2012, IEEE Communications Surveys and Tutorials, pp. 607–640.

4. *How to enable delay tolerant network solutions for internet of things: from taxonomy to open challenges.* **Bounsiar, Selma, Fatima Zohra Benhamida, Abderrazak Henni, Diego López de Ipiña, and Diego Casado Mansilla.** s.l. : In Proceedings, vol. 31, no. 1, p. 24. M.

5. *IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation.* **S. Schildt, J. Morgenroth, W.B. Pöttner, L. Wolf.** 2011, Electronic Communications of the EASST.

6. *An overview of μDTN: Unifying DTNs and WSNs.* **G. Von Zengen, F. Büsching, W.B. Pöttner, L. Wolf.** 2012.

7. *Routing protocols in delay tolerant networks: A comparative survey.* **J. Shen, S. Moh, I. Chung.** 2008, pp. 6-9.

8. *Routing approaches in delay tolerant networks: A survey.* **R. D'souza, J. Jose.** 2010, International Journal of Computer Applications.

9. *Routing strategies in delay tolerant networks: A survey.* **A. Abraham, S. Jebapriya.** 2012, International Journal of Computer Applications, pp. 44-48.

10. *Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges.* **Y. Cao, Z. Sun.** 2013, IEEE Communications Surveys and Tutorials, pp. 654–677.

11. *A survey of social-based routing in delay tolerant networks: Positive and negative social effects.* **Y. Zhu, B. Xu, X. Shi, Y. Wang.** 2013, IEEE Communications Surveys and Tutorials, pp. 387–401.

12. *Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things.* **B. Guo, D. Zhang, Z. Wang, Z. Yu, X. Zhou.** 2013, Journal of Network and Computer Applications, pp. 1531–1539.

13. *Survey on underwater delay/disruption tolerant wireless sensor network routing .* **H.H. Cho, C.Y. Chen, T.K. Shih, H.C. Chao.** 2014, IET Wireless Sensor Systems, pp. 112-121.

14. *Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges.* **Zhang, Z.** 2006, IEEE Communications Surveys and Tutorials, pp. 24–37.

15. *Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges.* **Yue Cao, Zhili Sun.** 2013. pp. 654-677.

16. *.A survey of routing and data dissemination in delaytolerant networks. .* **Sobin, C., et al., et al.** 128–146., s.l. : J. Netw. Comput. Appl. , 2016, Vol. 67.

17. *Data elevators: Applying the bundle protocol in delay tolerant wireless sensor networks.* **W.B. Pöttner, F. Büsching, G. Von Zengen, L. Wolf.** 2012, pp. 218–226.

18. *Opportunistic interaction in the challenged internet of things.* **H. Wirtz, J. Rüth, M. Serror, J.Á. Bitsch Link, K. Wehrle.** 2014, pp. 7–12.

19. *CoAP over BP for a delay-tolerant Internet of Things.* **M. Auzias, Y. Mahéo, F. Raimbault.** August 2015, pp. 118–123.

20. *Experiments and results on DTN for IOT III Urbanet collaboration.* **P. Raveneau, H. Rivano.** 2015.

21. *Internet of Hybrid Opportunistic Things: A novel framework for interconnecting IoTs and DTNs.* **Y. Xu, V. Mahendran, S. Radhakrishnan.** 2016, pp. 1067–1068.

22. *"A disruption tolerant architecture based on MQTT for IoT applications,"* . **J. E. Luzuriaga, M. Zennaro, J. C. Cano, C. Calafate and P. Manzoni,.** Las Vegas, NV, USA : 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC),, 2017.

23. *A delay-tolerant framework for integrated RSNs in IoT.* **F.M. Al-Turjman, A.E. Al-Fagih, W.M. Alsalih, H.S. Hassanein.** 2013, Computer Communications, pp. 998–1010.

24. *Generic prediction assisted single-copy routing in underwater delay tolerant sensor networks.* **Z. Guo, B. Wang, J.H. Cui.** 2013, Ad hoc Networks, pp. 1136–1149.

25. *"A comparative analysis of buffer management algorithms for delay tolerant wireless sensor networks.* **Söderman, Pehr, Karl-Johan Grinnemo, Markus Hidell, and Peter Sjödin.** 9612-9619., s.l. : " IEEE Sensors Journal 21, , (2021):, Vols. no. 7 ,.

26. *Reliable multicast disruption tolerant networking: Conceptual implementation using message ferry.* **K.S. Wong, T.C. Wan.** 2017, pp. 1817–1822.

27. *IoB-DTN: A lightweight DTN protocol for mobile IoT applications to smart bike sharing systems.* **Y. Zguira, H. Rivano, A. Meddeb.** 2018 Wireless Days (WD), pp. 131–136.

28. *An optimized probabilistic delay-tolerant network (DTN) routing protocol based on scheduling mechanism for Internet of Things (IoT) Sensors.* **Mao Y., Zhou C., Ling Y., Lloret J.** 2019, p. 24.

29. *Information Centric Delay Tolerant Networking: An Internet Architecture for the Challenged.* **A. Sathiaseelan, D. Trossen, I. Komnios, J. Ott, J. Crowcroft.** 2013.

30. *Interconnecting standard M2M platforms to delay tolerant networks.* **A. Elmangoush, A. Corici, M. Catalan, R. Steinke, T. Magedanz, J. Oller.** 2014, pp. 258–263.

31. *A proposal for a publish/subscribe, disruption tolerant content island for fog computing.* **P. Manzoni, E. Hernández-Orallo, C.T. Calafate, J.C. Cano.** 2017, pp. 47–52.

32. *Mind the smartgap: A buffer management algorithm for delay tolerant wireless sensor networks.* **P. Söderman, K.J. Grinnemo, M. Hidell, P. Sjödin.** 2015, pp. 104–119.

33. *An Optimized Probabilistic Delay Tolerant Network (DTN) Routing Protocol Based on Scheduling Mechanism for Internet of Things (IoT).* **Y. Mao, C. Zhou, Y. Ling, J. Lloret.** 2019, Sensors, p. 243.

34. *Delay-Tolerant Networking Architecture.* **V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss.** 2007, RFC (Request for Comments) 4838.

35. *A survey of routing and data dissemination in delay tolerant networks.* **C. Sobin, V. Raychoudhury, G. Marfia, A. Singla.** 2016, Journal of Network and Computer Applications, pp. 128–146.

36. *A review on Internet of Things solutions for intelligent energy control in buildings for smart city applications.* **I. Khajenasiri, A. Estebsari, M. Verhelst, G. Gielen.** 2017 , Energy Procedia, pp. 770–779.

37. *Energy efficient emergency rescue scheme in wireless sensor networks.* **Rishiwal, V., Singh, O.** 2021, International Journal of Information Technology., pp. 1951–1958.

38. *A novel communication framework between MANET and WSN in IoT-based smart environment.* **Tripathy, B.K., Jena, S.K., Reddy, V., et al.** 2021, International Journal of Information Technology, pp. 921–931.

39. *IoT-enabled smart dustbin with messaging alert system.* **Yadav, D., Pandey, A., Mishra, D., et al.** 2022, International Journal of Information Technology, pp. 3601–3609.

40. *Agent-driven resource scheduling in wireless sensor networks: fuzzy approach.* **Kori, G.S., Kakkasageri, M.S.** 2022, International Journal of Information Technology, pp. 345–358.

41. *A novel scheduling algorithm development and analysis for heterogeneous IoT protocol control system to achieve SCADA optimization: a next-generation post-COVID solution.* **Deshpande, S.N., Jogdand, R.M.** 2023, International Journal of Information Technology, pp. 2123–2131.

42. *How to Enable Delay Tolerant Network Solutions for Internet of Things: From Taxonomy to Open Challenges.* **Selma Bounsiar, Fatima Benhamida, Abderrazak Henni, Diego Ipiña, Diego Casado Mansilla.** s.l. : MDPI (Multidisciplinary Digital Publishing Institute), 2019. Proceedings. p. 24.

43. *DTN Routing Hierarchical Topology for the Internet of Things.* **El Arbi Abdellaoui Alaoui, Stéphane Cédric KOumetio Tekouabou, Antoine Gallais, Said Agoujil.** 2020. pp. 490-497.

44. *Current State of Multicast Routing Protocols for Disruption Tolerant Networks: Survey and Open Issues.* **K.S. Wong, T.C. Wan.** 2019, Electronics, p. 162.

45. *Wireless sensor networks: a survey. Computer Networks, .* **Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E.** (2002). , , Vols. 38(4), 393–422. . https://doi.org/10.1016/S1389-1286(01)00302-4.

46. *Delay-tolerant networking: An approach to interplanetary internet. .* **Burleigh, S., et al., et al.** 2003, IEEE Commun. Mag., pp. 128–136.

47. *"IOT based Smart Environment Monitoring Systems: A Key To Smart and Clean Urban Living Spaces."* . **Nandanwar, Himanshu, and Anamika Chauhan.** s.l. : IEEE, 2021. Asian Conference on Innovation in Technology (ASIANCON), . pp. pp. 1-9.

48. *Routing in a Delay Tolerant Network; ACM: .* **Jain, S., Fall, K. and Patra, R.** 2004, p. 34.

49. *Routing approaches in delay tolerant networks: A survey.* **R. D'souza, J. Jose.** 2010, International Journal of Computer Applications, pp. 8-14.

50. *Probabilistic routing in intermittently connected networks.* **Anders Lindgren, Avri Doria, Olov Schelén.** 2003. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc).

51. *Epidemic routing for partially connected ad hoc networks.* **Amin Vahdat, David Becker.** 2000.

52. *Quality of service in delay tolerant networks: A survey.* **A. Roy, T. Acharya, S. DasBit.** 2018. pp. 121-133.

53. *An overview of queuing delay and various delay based algorithms in networks.* **Roy, Arnab, Joseph Lalnunfela Pachuau, and Anish Kumar Saha.** no. 10 (2021): 2361-2399, s.l. : Computing 103,, 2021.

54. *Age matters: efficient route discovery in mobile ad hoc networks using encounter ages.* **Dubois-Ferriere, Henri, Matthias Grossglauser, and Martin Vetterli.** 2003., Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing., pp. 257-266.

55. *Single-copy routing in intermittently connected mobile networks.* **Thrasyvoulos Spyropoulos, Konstantinos Psounis, Cauligi S. Raghavendra.** s.l. : IEEE, 2004. 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004. pp. 235-244.

56. *Spray and wait: an efficient routing scheme for intermittently connected mobile networks.* **T. Spyropoulos, K. Psounis, C.S. Raghavendra.** Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, pp. 252–259.

57. *Practical routing in delay-tolerant networks.* **Evan PC Jones, Lily Li, Jakub K. Schmidtke, Paul AS Ward.** 2007. pp. 943-959.

58. *Bundle security protocol specification.* **S. Symington, S. Farrell, H. Weiss, P. Lovell.** 2011.

59. *Request for Comments 5050: Bundle Protocol Specification.* **Scott, K.** 2007.

60. *A survey on the internet of things security. .* **Zhao, K. and Ge, L.** 2013, pp. 663-667.

61. *Internet of Things (IoT): A literature review.* **Madakam, S., Ramaswamy, R. and Tripathi, S.** 2015. p. 164.

62. *"An efficient intrusion detection scheme for mitigating nodes using data aggregation in delay tolerant network." .* **Navaz, AS Syed, J. Antony Daniel Rex, and P. Anjala Mary.** September–2015, . International Journal of Scientific & Engineering Research, Vol No-6.

63. *Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility.* **T. Spyropoulos, K. Psounis, C.S. Raghavendra.** Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), pp. 79–85.

64. *Multiple-copy routing in intermittently connected mobile networks.* **Konstantinos Psounis, Cauligi S. Raghavendra.** 2004.

65. *Optimized Buffer Management Policy for Tailoring DTN Routing Protocols to IoT.* **Anamika Chauhan, Kapil Sharma.** 138–147, s.l. : International Journal of Intelligent Systems and Applications in Engineering, , 2024, Vols. 12(4),.

66. *Delay-tolerant network protocol testing and evaluation.* **Y. Li, P. Hui, D. Jin, S. Chen.** 2015. pp. 258-266.

67. *Simulating Mobility and DTNs with the ONE.* **Ari Keränen, Teemu Kärkkäinen, Jörg Ott.** 2010. pp. 92-105.

68. *Performance and security analyses of onion-based anonymous routing for delay tolerant networks.* **K. Sakai, M.-T. Sun, W.-S. Ku, J. Wu, F.S. Alanazi.** 2017. pp. 3473-3487.