Secure Communication Protocols for IoT Networks

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Aditya Singh (201394)

Under the guidance & supervision of

Dr. Pankaj Dhiman

Department of Computer Science & Engineering and
Information Technology
Jaypee University of Information Technology, Waknaghat,
Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report entitled "Secure Communication Protocols for IoT Networks" in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Aditya Singh, (201394)." during the period from January 2024 to May 2024 under the supervision of Dr. Pankaj Dhiman, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Aditya Singh
(201394)

The above statement made is correct to the best of our knowledge.

Dr. Pankaj Dhiman
Assistant Professor (SG), Dept. of CSE & IT,
Jaypee University of Information Technology,
Waknaghat, Solan, H.P., INDIA, 173234.

Candidate's Declaration

I hereby declare that the work presented in this report entitled **'Secure Communication Protocols for IoT Networks'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Pankaj Dhiman** (Assistant Professor, Department of Computer Science and Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Aditya Singh
Roll No.: 201394

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Dr Pankaj Dhiman
Designation: Assistant Professor
Department: Department of Computer Science and Engineering
Dated:

Acknowledgement

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the project work successfully.

I am grateful and wish our profound indebtedness to Supervisor Dr. Pankaj Dhiman, Assistant Professor (SG), Department of CSE, Jaypee University of Information Technology, Waknaghat, Solan. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Pankaj Dhiman, Assistant Professor (SG), Department of CSE, for his kind help to finish my project.

We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated our undertaking.

Finally, we must acknowledge with due respect the constant support and patience of my

parents.

Aditya Singh (201394)

Table of contents

Table of figures

Abstract

In an age where the Internet of Things (IoT) is prevalent and gadgets collaborate to better our lives, the security and integrity of these connections are crucial. The purpose of this research is to fortify communication protocols against dynamic cyberattacks by digging into the complicated world of Internet of Things networks. The report opens by addressing the shortcomings of traditional communication methods and emphasizing the significance of installing robust security measures as soon as possible. The paper identifies a severe vulnerability in modern IoT security by a careful evaluation of the limitations of present protocols.

The major goal of this project is to build and implement cutting-edge secure communication protocols for Internet of Things networks. By merging key management strategies, authentication procedures, and encryption tactics, we seek to construct a robust barrier against unauthorized access and data breaches. The implementation phase encompasses comprehensive testing and validation to assess the viability of the suggested solutions.

Introduction

1.1 Introduction

The Internet of Things (IoT) represents the integration of physical items, such as hardware, with advanced electronics, software, sensors, and network connectivity. This fusion allows these objects to collect and exchange data efficiently. Through the remote control of these interconnected devices, computer-based systems can be seamlessly integrated into the real world. This integration significantly enhances various aspects of daily life by improving efficiency, accuracy, and economic benefits, all while minimizing the need for human intervention.

IoT devices employ a broad spectrum of contemporary technologies to gather data. These devices operate by transmitting data packets, which travel independently between other objects in the network. For instance, smart home devices exemplify the practical applications of IoT, enabling homeowners to control lighting, heating, and security systems remotely. Wearable technology, such as the Apple Watch, extends the reach of IoT by monitoring health metrics and providing real-time updates. Smart city initiatives leverage IoT to manage urban infrastructure, including traffic lights, waste management, and public transportation, enhancing the quality of life for residents. Additionally, voice-activated assistants like Amazon Alexa exemplify how IoT can simplify daily tasks through voice commands and automation.

The scope of the Internet of Things is immense and continues to expand rapidly. One of the foundational elements in the IoT ecosystem is the use of the Internet Engineering Task Force's (IETF) 6LoWPAN protocol, which facilitates the connection of low-power devices over IP networks. This protocol is instrumental in the efficient communication between devices, ensuring that even resource-constrained gadgets can participate in the IoT network. Furthermore, the adoption of IPv6 is poised to play a critical role in enhancing the scalability of the network layer. By providing a vast number of IP addresses, IPv6 ensures that the growing number of IoT devices can be accommodated without running into address exhaustion issues.

In terms of architecture, the Internet of Things requires a highly dynamic and layered structure to function effectively. A robust IoT architecture typically consists of multiple layers, each responsible for different aspects of the system. The perception layer includes sensors and actuators that gather data from the environment. The network layer is responsible for transmitting this data to different devices and systems. The processing layer analyzes the data and makes decisions based on predefined criteria. Finally, the application layer delivers specific services to end-users based on the processed data. This layered approach ensures that IoT systems can scale efficiently, adapt to new technologies, and integrate seamlessly with existing infrastructure.

The Internet of Things (IoT) architecture comprises several distinct layers, each serving a unique purpose in the system's functionality:

Object Layering: This foundational layer, also known as the perception layer, consists of physical sensors and actuators that gather data from the environment. These sensors detect and capture various types of information, such as temperature, humidity, motion, and other physical parameters, which are then prepared for further processing.

Object Abstraction Layer: After data is collected by the sensors, it needs to be securely and efficiently transmitted to the next stage. The object abstraction layer handles this task by ensuring that the data from the perception layer is delivered to the service management layer over secure connections. Common technologies used at this stage include Bluetooth, Wi-Fi, and 4G, which facilitate reliable data transmission.

Service Management Layer: This layer acts as a bridge between the hardware and software components of IoT systems. It abstracts the complexities of the underlying hardware, allowing application developers to interact with various IoT devices without needing to understand the specifics of the hardware platform. By managing data flow and ensuring seamless communication between devices, this layer enables developers to create robust and versatile IoT applications.

Application Layer: Responsible for delivering high-level intelligent services, the application layer tailors these services to meet specific user needs. It interprets and processes the data collected and transmitted by the lower layers, addressing a wide range of applications such as smart transportation systems, smart homes, and healthcare devices. The application layer ensures that data is used effectively to provide valuable insights and services that align with user expectations.

Business Layer: At the top of the IoT architecture, the business layer oversees the entire operation of the IoT ecosystem. It focuses on managing and analyzing data to develop business strategies, models, and visual representations such as graphs and flowcharts. The insights generated by the application layer are used by the business layer to create actionable outcomes, ensuring that the final results are presented in an understandable manner. This helps businesses make informed decisions and meet customer requirements effectively once the project is completed.

1.2 IoT Protocols

The development of essential IoT protocols involves significant collaboration between prominent organizations such as the European Telecommunications Standards Institute (ETSI) and the Institute of Electrical and Electronics Engineers (IEEE). These bodies work together to create and standardize protocols that are fundamental to the efficient functioning of IoT systems. Rather than consolidating all IoT protocols at a single layer, modern architectural frameworks like the Open Systems Interconnection (OSI) model distribute these protocols across various layers, aligning them with specific organizational tiers and functionalities.

Infrastructure Layer: At the heart of IoT communication lies the infrastructure layer, which is crucial for the robust transmission of data across networks. One of the most significant protocols at this layer is IPv6 (Internet Protocol version 6). IPv6 plays a pivotal role in inter-networking by facilitating the end-to-end transmission of datagrams across diverse IP networks. This protocol supports packet switching, ensuring that data packets are efficiently routed and delivered, even in complex network environments. The vast address space provided by IPv6 is essential for accommodating the rapidly expanding number of IoT devices.

LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks): 6LoWPAN is a specialized protocol designed to enable IPv6 communication over IEEE 802.15.4 networks, which are typically used in low-power and lossy network (LLN) environments. With a data transfer rate of 250 kbps and operation within the 2.4 GHz frequency range, 6LoWPAN is optimized for energy efficiency and minimal power consumption, making it ideal for IoT applications that require long battery life and reliable performance under constrained conditions.

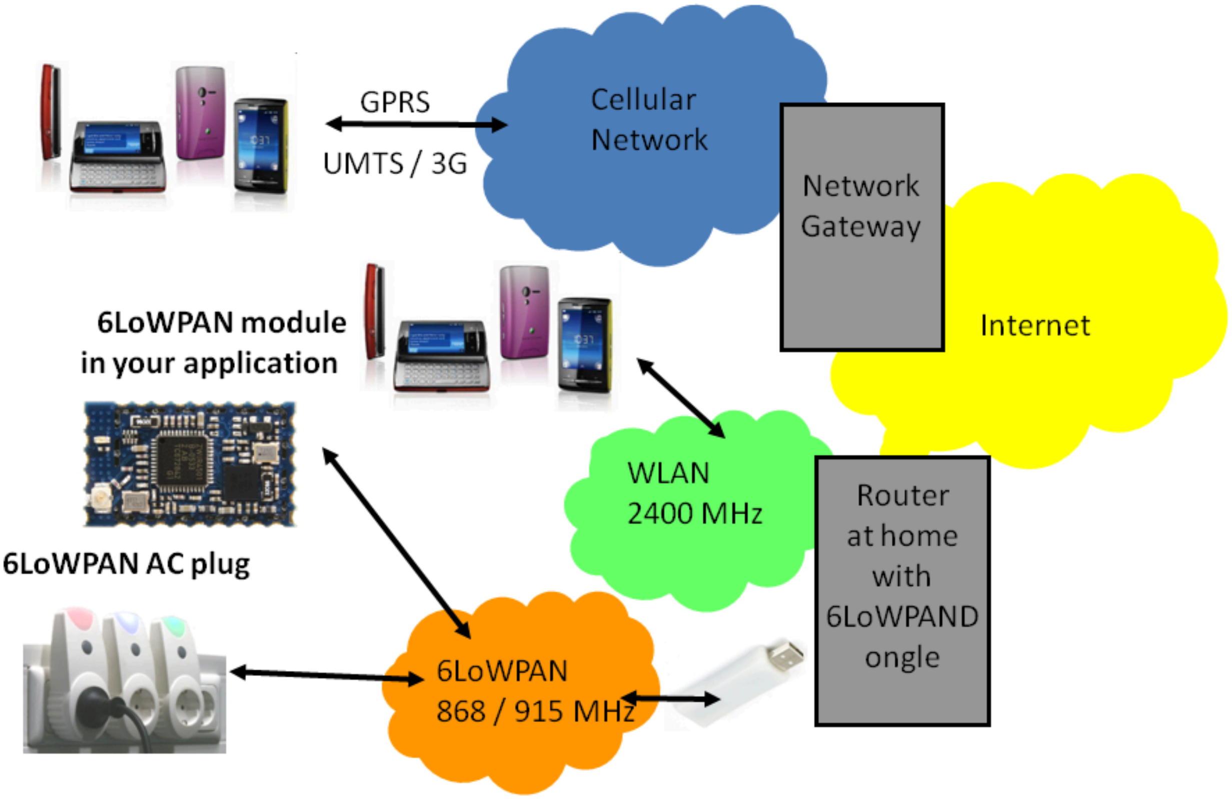


fig 1.1 LoPWAN

UDP (User Datagram Protocol): Operating at the transport layer of the OSI model, UDP is widely used in network applications due to its simplicity and efficiency. It supports various IoT protocols such as client-server IP communications, RPL (Routing Protocol for Low-Power and Lossy Networks), and MQTT (Message Queuing Telemetry Transport). Although UDP provides limited support for the publish-subscribe communication mechanism, its low overhead and brief code footprint make it highly suitable for IoT applications that demand flexibility and the capability to operate in remote or resource-constrained environments.

COAP (Constrained Application Protocol): COAP is an application layer protocol tailored for use with resource-constrained devices like wireless sensor network (WSN) nodes. Designed to be lightweight and efficient, COAP facilitates the easy translation of data to HTTP for web integration. It meets specific criteria such as multicast capabilities, minimal overhead, and simplicity. Key features of COAP include support for HTTP, URI, and various content types, which help reduce visual complexity and enhance interoperability among devices.

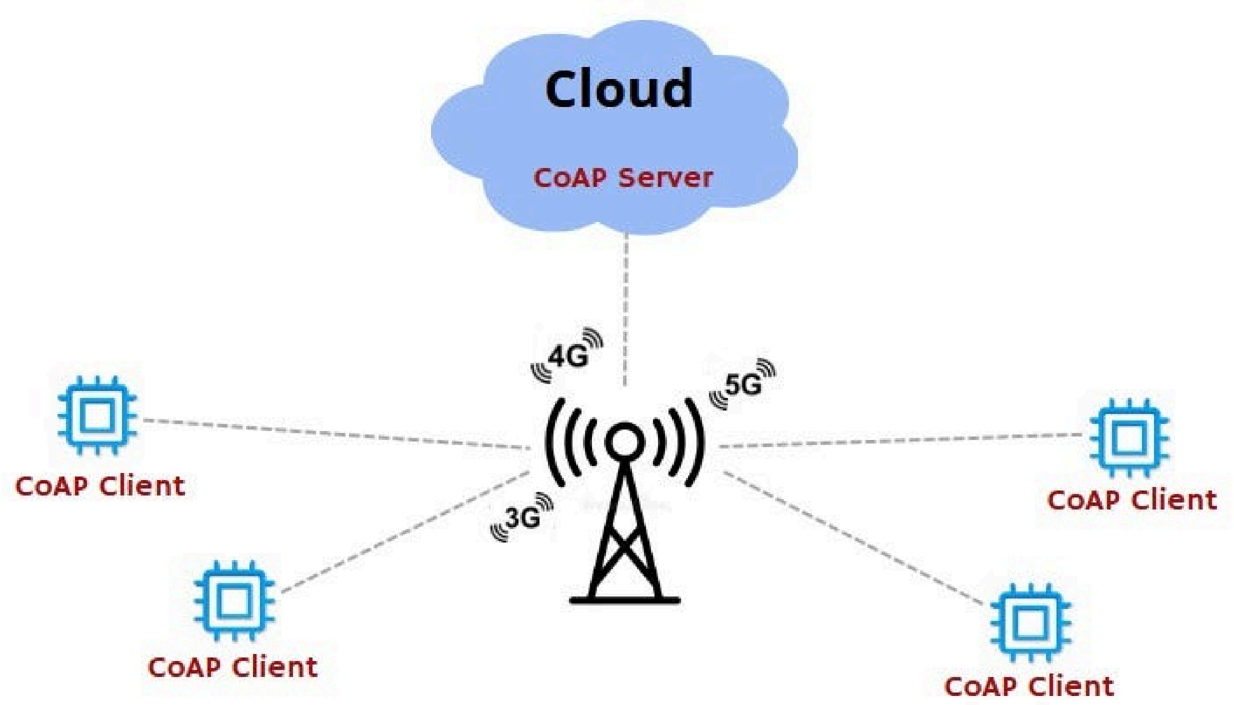


fig 1.2 CoAP Server

XMPP (Extensible Messaging and Presence Protocol): XMPP is an open-source protocol used in real-time communication systems, supporting a range of functions including messaging, presence information, phone and video calls, content syndication, lightweight middleware, and generalized XML data routing. Despite being an untrusted source, XMPP effectively manages publish-subscribe transmissions, making it suitable for real-time operating systems that require efficient and immediate data exchange.

In summary, the collaborative efforts of ETSI and IEEE have led to the development of a suite of essential IoT protocols, each designed to address specific needs within the IoT ecosystem. These protocols are strategically distributed across various layers of the OSI model, ensuring that they can support a wide range of applications and functionalities. By leveraging these protocols, IoT systems can achieve reliable, efficient, and scalable communication, paving the way for innovative applications and improved user experiences

IEEE 802.15.4: The IEEE 802.15.4 standard plays a crucial role in the development of low-rate wireless personal area networks (LR-WPANs), providing the necessary physical layer and media access control (MAC) for these networks. Overseen by the IEEE 802.15 working group, this standard forms the foundation for various upper-layer protocols and specifications, including ZigBee, ISA100.11a, WirelessHART, and MiWi. These protocols build on IEEE 802.15.4 to define additional layers and functionalities, enhancing the standard's versatility and applicability in numerous IoT scenarios. One notable application of IEEE 802.15.4 is in the construction of wireless embedded Internet systems, utilizing 6LoWPAN and conventional Internet protocols to ensure seamless communication and interoperability across diverse devices and networks.

NFC (Near Field Communication): NFC technology operates at a frequency of 13.56 MHz and facilitates short-range communication between devices. With a data transfer rate of approximately 424 kbps, NFC is well-suited for applications requiring secure, close-range interactions, such as contactless payments, access control, and data sharing. The limited range of NFC, typically a few centimeters, enhances its security by reducing the risk of unauthorized access or interception, making it ideal for sensitive transactions.

fig 1.3 NFC

Bluetooth: Bluetooth technology operates in the 2.4 GHz ISM (Industrial, Scientific, and Medical) band and employs frequency hopping techniques to minimize interference and ensure reliable communication. It supports data transmission rates up to 3 Mbps over distances of up to 100 meters, depending on the specific Bluetooth class and application. Each Bluetooth application is defined by its own profile, which specifies the protocols and procedures for various use cases, from audio streaming and file transfer to health monitoring and smart home control. The adaptability and widespread adoption of Bluetooth make it a cornerstone technology in the IoT ecosystem.

The Open Trust Protocol (OTrP): OTrP is pivotal for managing the Trusted Execution Environment (TEE), a secure area of a processor that ensures the integrity and confidentiality of code and data. This protocol is used for the installation, updating, and deletion of security software, maintaining the robustness of the TEE. By securing critical operations and preventing unauthorized access, OTrP plays a vital role in safeguarding IoT devices and their communications.

X.509: X.509 is a standard for public-key infrastructure (PKI) that defines the format of public-key certificates. It is an essential component of security protocols such as Transport Layer Security (TLS), which is used to secure Internet communications, including online transactions and email. X.509 certificates ensure the authenticity and integrity of communications by enabling encryption and digital signatures, thus protecting data from interception and tampering.

Together, these protocols and standards provide a comprehensive framework for secure, efficient, and scalable IoT communications. By addressing various layers and aspects of network interactions, from physical and MAC layers to application and security protocols, they enable the development and deployment of robust IoT systems capable of supporting a wide array of applications.

Large IoT Applications and Network Requirements: Large-scale IoT applications and networks have specific criteria to ensure their efficient and reliable operation. These criteria include:

1. **Low Energy Consumption:** IoT devices, particularly those in remote or difficult-to-access locations, must operate on minimal energy to prolong battery life and reduce maintenance needs. This requirement is critical for the sustainability and cost-effectiveness of large IoT deployments.

2. **Wireless Sensor Networks:** The ability to utilize wireless sensor networks (WSNs) is essential. WSNs consist of spatially distributed sensors that monitor and record environmental conditions, transmitting the data wirelessly. These networks are crucial for applications such as environmental monitoring, industrial automation, and smart agriculture.
3. **Ad-Hoc Network Capabilities:** IoT systems often need to support ad-hoc networking, which allows devices to communicate directly with each other without relying on a pre-existing infrastructure. This capability is vital for dynamic and flexible network configurations, particularly in scenarios where the network topology changes frequently.

4. **Point-to-Point (P2P) Communication:** Support for P2P communication is necessary for direct data exchange between devices. This type of communication is essential for applications that require low latency and high reliability, such as real-time monitoring and control systems.

5. **Mobility Support:** IoT applications must promote and accommodate mobility, allowing devices to move and still maintain connectivity and performance. Mobility is particularly important in applications like mobile health monitoring, asset tracking, and autonomous vehicles.

Protocol Evaluation: Upon comparing various IoT protocols, it becomes evident that RPL (Routing Protocol for Low-Power and Lossy Networks) and CTP (Collection Tree Protocol) are among the best choices for many IoT applications. However, these protocols also have certain vulnerabilities and limitations that need addressing.

RPL Protocol Vulnerabilities: RPL, while highly suitable for low-power and lossy networks, has several vulnerabilities:

- **VeRA (Version Number and Rank Authentication Attack):** This attack exploits the protocol's version number system to disrupt network operations.
- **Performance Issues During High Traffic:** RPL's performance can degrade under heavy network traffic conditions, leading to delays and packet loss.
- **Dynamic Traffic Handling:** RPL sometimes struggles with highly dynamic traffic patterns, which can result in inefficient routing and increased latency.
- **Mobility Constraints:** The protocol often fails to adapt to the mobility of nodes, causing frequent connection breaks and erroneous routing decisions.

Real-World Application Needs: Mobility support is critical for certain real-world IoT applications. For instance, sensor networks in agricultural fields need to measure soil moisture and other parameters to determine the land's suitability for farming. These sensors may need to be moved or redeployed as conditions change, requiring robust support for mobility.

Similarly, oceanographic monitoring systems use sensors to record conditions at various points in the sea or ocean. These sensors, which may use technologies like RADAR to detect obstacles and measure depths, need to adapt to the movement of water and change locations accordingly. This mobility necessitates IoT protocols that can handle dynamic node positions without losing connectivity or accuracy.

Attack Mitigation: To enhance the security and robustness of IoT networks, it is crucial to address vulnerabilities like the VeRA attack. This involves implementing mechanisms to authenticate version numbers and ranks within the network. For instance, ensuring that the "Version Number" in a Destination Oriented Directed Acyclic Graph (DODAG) is securely managed and updated only by authorized nodes can prevent attackers from manipulating the network topology.

By focusing on these requirements and vulnerabilities, IoT protocols can be improved to support a wider range of applications more effectively. This ensures that IoT networks remain reliable, secure, and adaptable to the evolving needs of diverse industries and use cases.

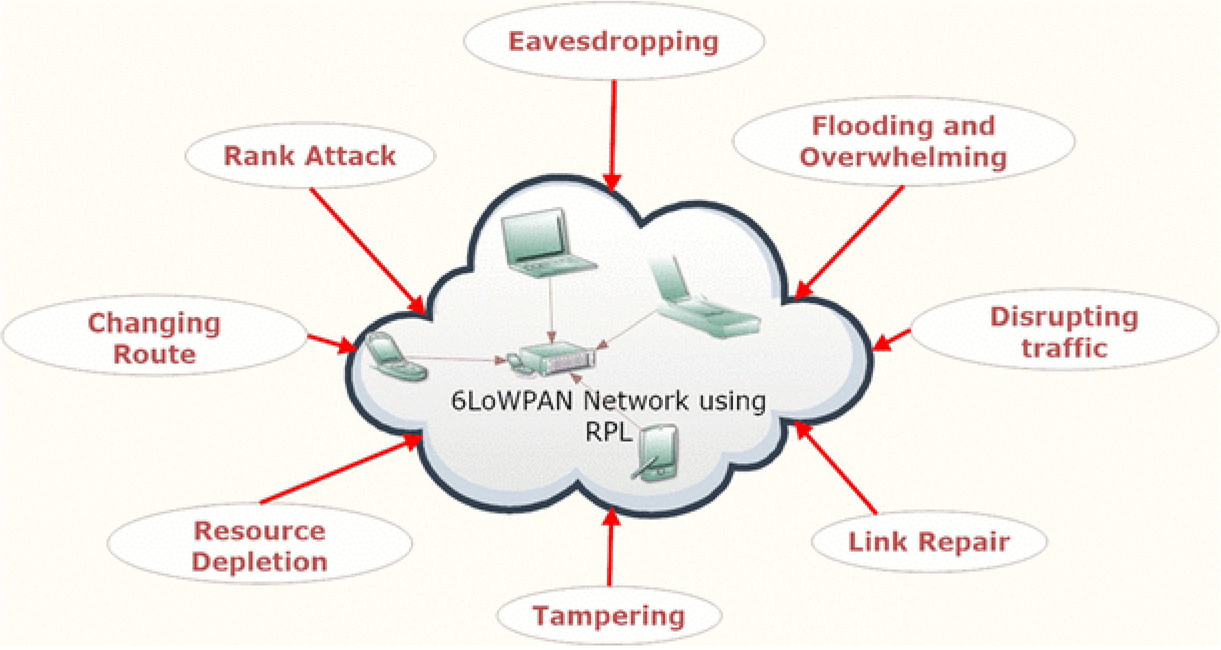


fig 1.5 LoWPAN Network using RPL

1.3 Objectives

Resolving the Difficulties Raised in RPL:

The primary goal of this report is to address and resolve the challenges associated with the RPL (Routing Protocol for Low-Power and Lossy Networks) to enhance its applicability and reliability in diverse IoT environments. Specifically, this report aims to tackle the issues of mobility and security within the RPL framework, as well as to evaluate the performance of the RPL-UDP protocol.

1. **Introducing RPL Mobility:** Mobility support is a critical aspect that current RPL implementations lack. In many IoT applications, devices are not static; they move either predictably, such as in logistics and transportation, or unpredictably, as in wildlife monitoring and personal wearables. This objective involves developing mechanisms to introduce and manage mobility within RPL networks. By incorporating mobility, we aim to ensure that RPL can maintain stable and efficient routing paths even as the positions of the nodes change. This enhancement will significantly expand the use cases for RPL, enabling its application in dynamic environments where traditional static routing protocols would fail.
2. **Concerns About RPL Security:** Security is another critical challenge for RPL, particularly given its vulnerability to various attacks, such as the Version Number and Rank Authentication (VeRA) attack. Addressing these security concerns involves implementing robust authentication and encryption mechanisms to protect the integrity and confidentiality of the data being routed through the network. This includes developing strategies to secure the version numbers and ranks within the network, preventing malicious nodes from disrupting the network topology. Enhancing the security of RPL will make it a more viable option for sensitive applications, such as smart healthcare and critical infrastructure monitoring, where data integrity and security are paramount.

To achieve these objectives, we will extensively utilize Contiki-COOJA, a widely-used network simulator for IoT, to emulate and analyze the behavior of motes (sensor nodes) in RPL networks. Contiki-COOJA allows us to create detailed simulations of IoT environments, both with and without the implementation of mobility. Through these simulations, we will be able to observe the effects of our proposed enhancements on network performance and efficiency.

Evaluation of RPL-UDP Protocol: Another aspect of our study involves evaluating the RPL-UDP (User Datagram Protocol) combination. UDP is often used in conjunction with RPL due to its low overhead and simplicity, making it suitable for resource-constrained IoT devices.

We will analyze how well RPL works with UDP, particularly under conditions of high mobility and potential security threats. This will involve testing various scenarios and configurations to identify optimal settings and potential weaknesses that need to be addressed.

Addressing VeRA and Enhancing Network Performance: A significant part of our research focuses on the VeRA problem, which currently poses a substantial threat to the reliability of RPL. VeRA attacks exploit vulnerabilities in the way RPL handles version numbers and ranks, leading to network disruptions and inefficiencies. By developing and implementing countermeasures against VeRA, we aim to enhance the overall security and robustness of RPL. Additionally, we will explore techniques to improve network performance and efficiency, ensuring that RPL can handle high traffic loads and dynamic conditions without degradation in service quality.

In summary, the objectives of this report are to introduce mobility into RPL, address security concerns, and evaluate the performance of the RPL-UDP protocol using Contiki-COOJA simulations. Through these efforts, we aim to make RPL a more versatile and secure protocol for a wide range of IoT applications, ultimately contributing to the development of more reliable and efficient IoT networks.

Chapter 2 : Literature review

DODAGs as the Cornerstone of RPL: Directed Acyclic Graphs (DODAGs) form the foundation of distance vector routing protocols, such as the Routing Protocol for Low-Power and Lossy Networks (RPL). In essence, RPL constructs a DODAG-based topology, which is crucial for routing in resource-constrained environments typically seen in IoT deployments.

Structure and Operation of SNUG RPL: In a typical RPL setup, often referred to as SNUG RPL, the protocol builds a DODAG that is anchored by a border router or another routing entity. Each node within this network usually has multiple parent nodes, although one parent is designated as the preferred parent. This preferred parent is primarily responsible for forwarding data packets towards the root node, while the remaining parent nodes serve as alternative paths, ensuring redundancy and reliability in data transmission.

Communication Patterns in RPL: RPL's architecture inherently supports multipoint-to-point communication, which is essential for efficiently transmitting data from various nodes to a single root node. This form of communication is facilitated by maintaining a minimal routing state, thereby optimizing the resource usage of the nodes. The network topology is dynamically established and maintained through control packets known as DODAG Information Objects (DIOs), which are periodically advertised by each node.

Dynamic DIO Advertisement: To ensure adaptive and responsive network behavior, each node rebroadcasts DIO packets using an adaptive method. During periods of network instability, DIOs are propagated more frequently to quickly update the network state. Conversely, when the network is stable, the rebroadcast interval gradually increases, thereby conserving energy while maintaining adequate responsiveness to any topological changes.

Destination Advertisement Objects (DAOs): For downward routing, where data needs to be sent from the root to specific nodes, every node periodically sends Destination Advertisement Object (DAO) control packets to the root. These DAOs travel upward through the preferred parent routes, establishing the necessary downward paths within the DODAG structure. This mechanism ensures that every node in the network is recognized as a reachable destination, facilitating bidirectional communication.

Storing vs. Non-Storing Modes in RPL: RPL operates in two distinct modes: storing and non-storing. In storing mode, each node maintains a routing table that includes mappings of all destinations and their corresponding next-hop nodes within its sub-DODAG. This approach allows each node to independently route data, improving scalability and reducing the load on the root node. In contrast, in non-storing mode, only the root node maintains the complete routing information, which it includes directly in the data packets. This mode simplifies the routing logic for individual nodes but increases the complexity at the root.

Impact on Point-to-Point Communication: The choice between storing and non-storing modes also affects RPL's efficiency in handling point-to-point (P2P) communication. In storing mode, nodes can quickly route P2P messages based on their local routing tables, making this mode more suitable for networks with frequent P2P communication. Non-storing mode, while simpler for individual nodes, may introduce latency and increased traffic through the root, which needs to process and route all P2P communications.

CTP System for Sensor Networks: The Collection Tree Protocol (CTP) is another prominent routing protocol designed specifically for sensor networks. CTP establishes efficient routing paths from each sensor node to a central root node. Similar to RPL, CTP focuses on minimizing energy consumption and maximizing network reliability, making it a suitable choice for environments with stringent resource constraints. CTP’s primary function is to ensure that data collected by sensor nodes is reliably transmitted to the root, where it can be processed and analyzed. This protocol is widely used in various sensor network applications, including environmental monitoring and industrial automation.

By understanding the nuances and operational details of DODAGs within RPL, as well as comparing it with protocols like CTP, we can better appreciate the complexities and requirements of routing in IoT networks. The adaptive mechanisms, control packet strategies, and operational modes of these protocols are critical for optimizing network performance and ensuring robust, scalable, and secure communication in diverse IoT applications.

Addressing the Challenges in Distance Vector Routing:

The distance vector routing protocol faces significant challenges in highly dynamic wireless networks. To address these challenges, the Collection Tree Protocol (CTP) employs three key strategies that ensure reliable and efficient routing. These strategies are critical for maintaining the integrity and performance of the network amidst rapidly changing conditions.

Link Quality Evaluation:

Wireless networks are notorious for their bursty activity, where periods of high traffic intensity are followed by brief lulls. This unpredictable behavior can severely impact the reliability of communication links. CTP tackles this issue with a four-bit link estimator, which provides accurate and timely link quality assessments. This estimator collects data from multiple layers—physical, data link, and network—allowing it to create a comprehensive picture of the link's performance. By integrating this multi-layered information, the estimator can make informed decisions about which routes to use, ensuring that data packets are sent through the most reliable paths available at any given moment.

Data Path Validation:

In a dynamic wireless environment, a routing path that is stable one moment can quickly become unreliable due to fluctuations in link quality. Such changes can lead to the formation of routing loops, where data packets are continuously circulated within the network, causing congestion and wasting energy. To prevent this, CTP employs data path validation techniques. These techniques continuously monitor the integrity of data paths by validating the routes as data packets travel through them. By identifying and correcting potential issues in real time, CTP ensures that data packets are delivered efficiently, minimizing the risk of routing loops and the associated energy loss and network congestion.

Adaptive Beaconing:

Adaptive beaconing is a crucial feature that helps CTP manage the trade-off between responsiveness and resource consumption. Routing protocols typically use beaconing to broadcast control packets at regular intervals, which helps maintain the network topology and update routing information. However, the frequency of these broadcasts can significantly impact the protocol's efficiency. A shorter interval between beacons allows the network to quickly adapt to changes, but it also consumes more bandwidth and energy. Conversely, a longer interval conserves resources but may delay the network's response to topological changes.

CTP addresses this issue with an adaptive beaconing strategy that adjusts the beaconing interval based on the current network conditions. When the network topology is unstable or experiencing rapid changes, CTP increases the frequency of beacon transmissions to quickly propagate updates and stabilize the network. Once stability is achieved, the protocol exponentially decreases the beaconing rate, conserving energy and bandwidth while maintaining adequate responsiveness. This adaptive approach allows CTP to efficiently manage control overhead and remain effective in the face of fluctuating wireless dynamics.

The Significance of CTP's Strategies:

The strategies employed by CTP—link quality evaluation, data path validation, and adaptive beaconing—are critical for maintaining robust and efficient routing in dynamic wireless networks. By providing accurate link quality assessments, CTP ensures that data packets are routed through the most reliable paths. Data path validation helps prevent the formation of routing loops, reducing congestion and energy waste. Adaptive beaconing allows the protocol to balance the need for timely updates with resource conservation, making CTP highly adaptable to changing network conditions.

CTP Versus RPL: A Comparative Analysis

In the domain of wireless sensor networks, both the Collection Tree Protocol (CTP) and the Routing Protocol for Low Power and Lossy Networks (RPL) are critical for effective data transmission. However, these protocols exhibit significant differences in their design and performance, particularly under varying network conditions.

1. **Topology Formation and Control Messaging:**

CTP, a distance-vector routing technique, constructs a tree-based topology where the root node is positioned at the network's sink. The control messages in CTP are disseminated using an adaptive beaconing technique, which dynamically adjusts the interval between beacon transmissions based on network conditions. This approach ensures rapid adaptation to changes but can lead to increased energy consumption during high activity periods.

In contrast, RPL utilizes an Objective Function (OF) to construct Destination-Oriented Directed Acyclic Graphs (DODAGs). This mechanism is more adaptable to various applications, allowing RPL to cater to a wider range of network scenarios. The flexibility in the OF makes RPL more versatile compared to the specialized link-layer technology required by CTP.

2. **Energy Consumption and Packet Reception Ratio (PRR):**

One of the primary performance metrics for routing protocols in wireless sensor networks is the Packet Reception Ratio (PRR), which indicates the reliability of data packet delivery. CTP is known for its high energy consumption and impressive PRR in smaller networks. However, as the network size increases, CTP's PRR tends to decline, particularly under high data flow conditions. This is largely due to the increased energy expenditure required to maintain adaptive beaconing and manage data path validation.

On the other hand, RPL demonstrates superior PRR with lower energy consumption in larger networks. As the network scales up to 49 nodes or more, RPL maintains a higher PRR and exhibits lower parent churn compared to CTP. This stability is attributed to the structured

DODAG construction and the efficient coordination of control messages such as DIO, DIS, and DAO.

3. **Parent Change Rate and Network Stability:**

Parent churn, or the rate at which nodes change their parent nodes, is a crucial factor in network stability. Higher churn rates can lead to increased control message overhead and energy consumption. RPL outperforms CTP in this aspect by maintaining a significantly lower parent change rate, ensuring consistent and stable network performance. This is particularly beneficial in large-scale networks where frequent parent changes can disrupt data flow and increase latency.

4. **Handling Network Size and Data Traffic:**

RPL's architecture is designed to handle large network sizes and rising data traffic efficiently. The protocol's ability to maintain high PRR and low energy consumption under these conditions makes it suitable for expansive IoT deployments. CTP, while effective in smaller networks, struggles with scalability as the network grows. The protocol's performance degrades in terms of both PRR and energy efficiency, highlighting its limitations in large-scale applications.

5. **Performance Metrics:**

Performance tests reveal that both RPL and CTP achieve PRRs exceeding 99.8%, indicating their reliability in packet delivery. However, RPL's turnover rate is higher, suggesting a more dynamic adaptation to changing network conditions. Despite this, RPL tends to take longer routes and exhibits a slightly higher per-hop Expected Transmission Count (ETX) value than CTP. This trade-off is mitigated by RPL's robust handling of diverse traffic patterns, including Point-to-Point (P2P) and Point-to-Multipoint (P2MP) communications.

6. **Traffic Pattern Flexibility:**

A significant advantage of RPL over CTP is its ability to accommodate various traffic patterns. RPL can establish direct connections with Internet nodes using IPv6 global addresses, enabling seamless integration with external networks. This capability is particularly valuable for IoT applications requiring diverse and flexible communication pathways.

7. **Security Features:**

RPL incorporates several security features to protect control messages such as DIO, DIS, DAO, and DAO-ACK. A message code bit in RPL indicates whether the message is secure, and secure versions of these control messages are available. However, due to the resource constraints in Low Power and Lossy Networks (LLNs) and the complexity of RPL, the security measures outlined in the RFC are optional and can be implemented based on specific network requirements. This flexibility allows network designers to balance security needs with resource availability.

RPL Security Modes

RPL (Routing Protocol for Low Power and Lossy Networks) offers three primary security modes designed to protect data transmission and network integrity:

1. **Unsecured Security Mode:** In this basic mode, messages are transmitted without inherent security features. However, network operators can implement additional security using Link Layer Security (LLS) or Application Layer Security (ALS) methods. This mode is suitable for less sensitive applications where the overhead of advanced security measures is not justified.
2. **Prior Installation of Security Mode:** This mode employs secure messages where a key must be pre-installed to connect RPL instances to hosts and routers. These keys ensure message secrecy, integrity, and trustworthiness during communication. This mode provides a balance between security and operational simplicity, making it suitable for many IoT applications where pre-configured devices need to communicate securely.
3. **Security Mode with Authentication:** In this most secure mode, RPL uses encrypted messaging. Pre-installed keys alone are insufficient; a second key must be obtained from a key authority before a device can be connected as a router in the network. This additional layer of security ensures message secrecy, integrity, and trustworthiness. However, due to RPL's current limitations, it does not support asymmetric algorithms, which limits the implementation of authenticated security modes with current technology.

Issues in RPL Security

Despite these security modes, RPL is not immune to various vulnerabilities and attacks, particularly when dealing with compromised internal nodes:

1. **Version Number Attack:** One of the most critical vulnerabilities involves manipulating the version number of the DODAG (Destination-Oriented Directed Acyclic Graph). An attacker can increment a sequential counter to form a newer version of the DODAG, or modify the rank value of a DODAG node, thereby altering its position within the network. This manipulation can disrupt the network's routing structure and degrade performance.
2. **Rank Authentication Attack:** In this scenario, an internal attacker publishes a lower Rank value, contrary to protocol rules, to attract more traffic and cause network inefficiencies. Such an attack can lead to resource exhaustion and degraded network performance.

Current RPL Security Measures:

To address these issues, RPL incorporates essential security options but remains susceptible to various topological attacks that facilitate resource exhaustion, interception, and black holing. To counter these vulnerabilities, RPL has introduced TRAIL, a generic topology authentication mechanism.

TRAIL (Topology Authentication for RPL):

TRAIL is designed to validate the integrity of routing paths within the RPL network. It utilizes round trip messages to confirm the authenticity of upward paths to the DODAG root. Unlike VeRA (Version and Rank Authentication), TRAIL ensures rank integrity based on an upward path that remains recursively intact, rather than relying solely on encryption chains.

- **Advantages of TRAIL:** TRAIL aims to reduce network message exchanges and node resource consumption. It maintains the viability of bit transmissions in demanding environments, ensuring that regular network operations can proceed with minimal additional effort. This approach enhances network reliability and security without imposing significant operational burdens.

VeRA (Version and Rank Authentication):

VeRA is another security mechanism proposed for RPL to prevent intruder nodes from exploiting version number and rank values. It employs a one-way hash chain of a specific length to counter these attacks. VeRA's fundamental components include hash functions (e.g., MD5, SHA), Message Authentication Codes (HMAC), and electronic signatures (e.g., RSA, DSA, ECC).

1. **Preventing Illegal Version Number Increments:** VeRA prevents nodes from pretending to be DODAG roots and sending DIO (DODAG Information Object) messages with an unlawful version number increment. This helps maintain the integrity of the network’s versioning system.
2. **Preventing Illegal Rank Advertisements:** It stops unauthorized nodes from advertising a lower rank, which could otherwise attract traffic and disrupt the network. By using cryptographic hash functions and electronic signatures, VeRA ensures that only legitimate nodes can alter their rank in the network.

Chapter 3 – SYSTEM DEVELOPMENT

3.1 COOJA – Contiki

Cooja is a powerful and flexible simulator specifically designed to emulate Contiki-related applications. Contiki-OS, known for its lightweight and adaptable nature, is an operating system optimized for resource-constrained devices in networked environments, such as those found in Wireless Sensor Networks (WSNs). The development of Cooja was spearheaded by the creators of Contiki-OS to provide a robust tool for testing and validating network protocols, configurations, and applications in a controlled and repeatable environment.

fig 3.1 contiki

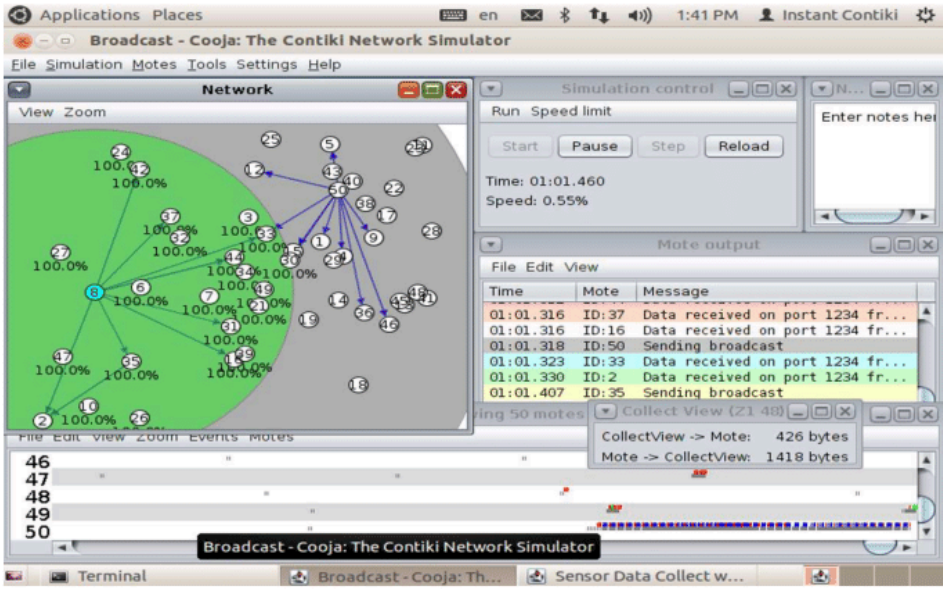
3.2 Functionality of Cooja Simulator

The primary purpose of the Cooja Simulator is to offer a detailed and customizable platform for simulating various aspects of WSNs and IoT networks. It provides a user-friendly interface and a comprehensive set of features that facilitate the design, testing, and analysis of network protocols. Here's a detailed breakdown of how the Cooja Simulator operates:

1. **Initialization of a New Simulation:**
   - To begin, a user needs to initialize a new simulation environment. This involves setting up the basic parameters and configurations that define the scope and scale of the simulation. Users can specify factors such as the network topology, simulation duration, and the types of network traffic to be generated.
2. **Creation of New Mote Types:**

- Motes are the basic units or nodes in a WSN. In Cooja, users can create different types of motes to simulate various devices and their behaviors. Each mote type can be customized with specific hardware characteristics, sensor types, and communication protocols. This allows for a realistic and detailed representation of different nodes within the network.

3. **Adding Motes to the Simulation:**
   - Once the mote types are defined, the next step is to add these motes to the simulation environment. Users can place motes at specific locations within the network topology, defining their initial positions and connectivity. This step is crucial for setting up the network structure and ensuring that the simulation accurately reflects real-world scenarios.
4. **Running the Simulation:**
   - After setting up the motes and defining the simulation parameters, the user can start the simulation. During this phase, Cooja executes the predefined network behaviors, allowing users to observe and analyze the interactions between motes. The simulator provides real-time visualization and detailed logging of network activities, which helps in understanding the performance and behavior of the network under various conditions.
5. **Saving the Simulation File:**
   - To facilitate future analysis and replication, Cooja allows users to save the simulation configuration and results. The simulation file contains all the details about the network setup, mote configurations, and simulation outcomes. This feature is essential for conducting iterative testing and for sharing simulation setups with other researchers or developers.



Time: 01:01.460
Speed: 0.55%

```
Time        Mote    Message
01:01.316   ID:37   Data received on port 1234 fr...
01:01.316   ID:16   Data received on port 1234 fr...
01:01.318   ID:50   Sending broadcast
01:01.323   ID:33   Data received on port 1234 fr...
01:01.330   ID:2    Data received on port 1234 fr...
01:01.407   ID:35   Sending broadcast
```

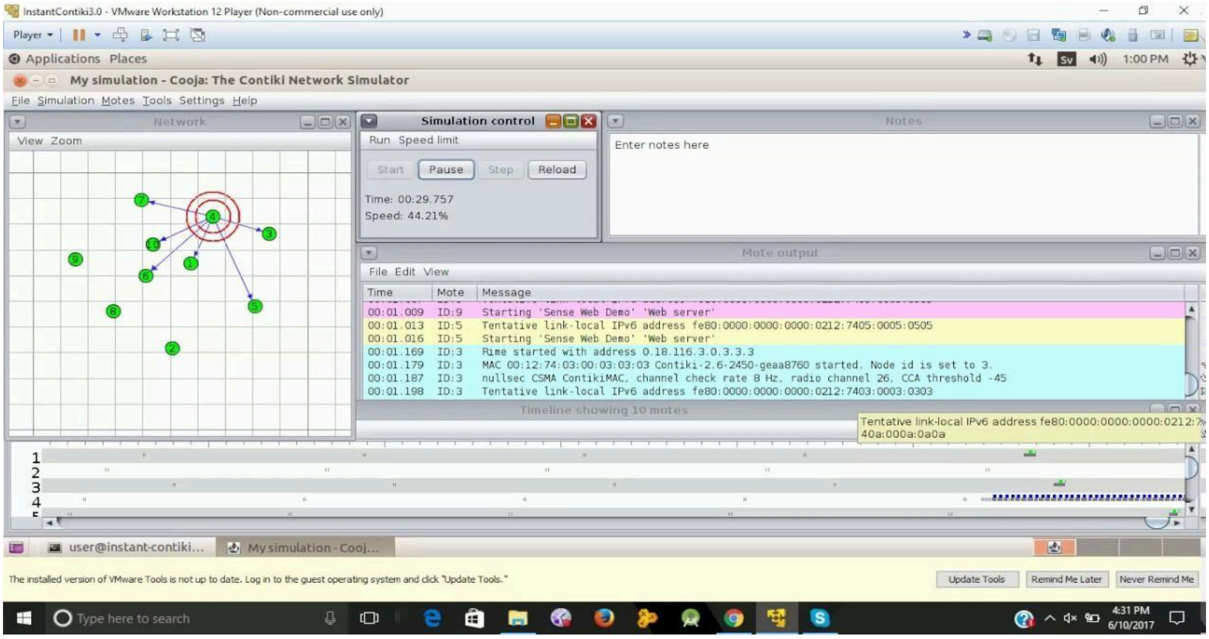
CollectView -> Mote: 426 bytes
Mote -> CollectView: 1418 bytes

fig 3.2 Cooja simulator in action

3.3 Advanced Features of Cooja Simulator

Beyond the basic functionalities, Cooja offers several advanced features that enhance its utility for IoT and WSN research:

- **Interoperability with Real Hardware:** Cooja can interface with real hardware devices, allowing for hybrid simulations where some nodes are virtual while others are actual physical devices. This feature is particularly useful for validating simulation results in real-world environments.
- **Customizable Plugins and Extensions:** Users can extend Cooja's capabilities by developing custom plugins and extensions. This allows for the addition of new functionalities and the adaptation of the simulator to specific research needs or applications.
- **Detailed Network Metrics and Analysis Tools:** Cooja provides a range of tools for monitoring and analyzing network performance. Users can track metrics such as packet delivery ratio, latency, energy consumption, and network throughput. These tools are essential for evaluating the efficiency and reliability of network protocols.
- **Support for Multiple Communication Protocols:** Cooja supports a wide variety of communication protocols used in IoT and WSNs, including IEEE 802.15.4, 6LoWPAN, RPL, and CoAP. This versatility makes it a valuable tool for testing and comparing different protocols under diverse network conditions.


InstantContiki3.0 - VMware Workstation 12 Player (Non-commercial use only)
Player
Applications Places
1:00 PM
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start Pause Step Reload
Notes
Mote output
File Edit View
Timeline showing 10 motes
user@instant-contiki...
My simulation - Cooj...
Update Tools Remind Me Later Never Remind Me
Type here to search
4:31 PM
6/10/2017

Time: 00:29.757
Speed: 44.21%

Enter notes here

```
Time       Mote   Message
00:01.009  ID:9   Starting 'Sense Web Demo' 'Web server'
00:01.013  ID:5   Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7405:0005:0505
00:01.016  ID:5   Starting 'Sense Web Demo' 'Web server'
00:01.169  ID:3   Rime started with address 0.18.116.3.0.3.3.3
00:01.179  ID:3   MAC 00:12:74:03:00:03:03:03 Contiki-2.6-2450-geaa8760 started. Node id is set to 3.
00:01.187  ID:3   nullsec CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
00:01.198  ID:3   Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7403:0003:0303
```

Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7
40a:000a:0a0a

The installed version of VMware Tools is not up to date. Log in to the guest operating system and click "Update Tools."

fig 3.3 features of Cooja

Mobility in Contiki-2.7 Cooja Simulator

The first step is to download plugins: The plugins can be downloaded from the following link:

https://github.com/vaibhav90/Mobilty_Interference_Plugin_Patch_Contiki2.7/tree/master/mobility.

A new directory is created at.

cd contiki

cd tools

cd cooja

cd apps

Give the new directory name to mobility.

After downloading the URL files, move them to the "mobility" folder.

The second step is to create the mobility plugin.

Using the command "cd contiki/tools/cooja/apps/mobility," navigate to the mobility directory. Use the sudo ant jar command to enable the plugin's construction.

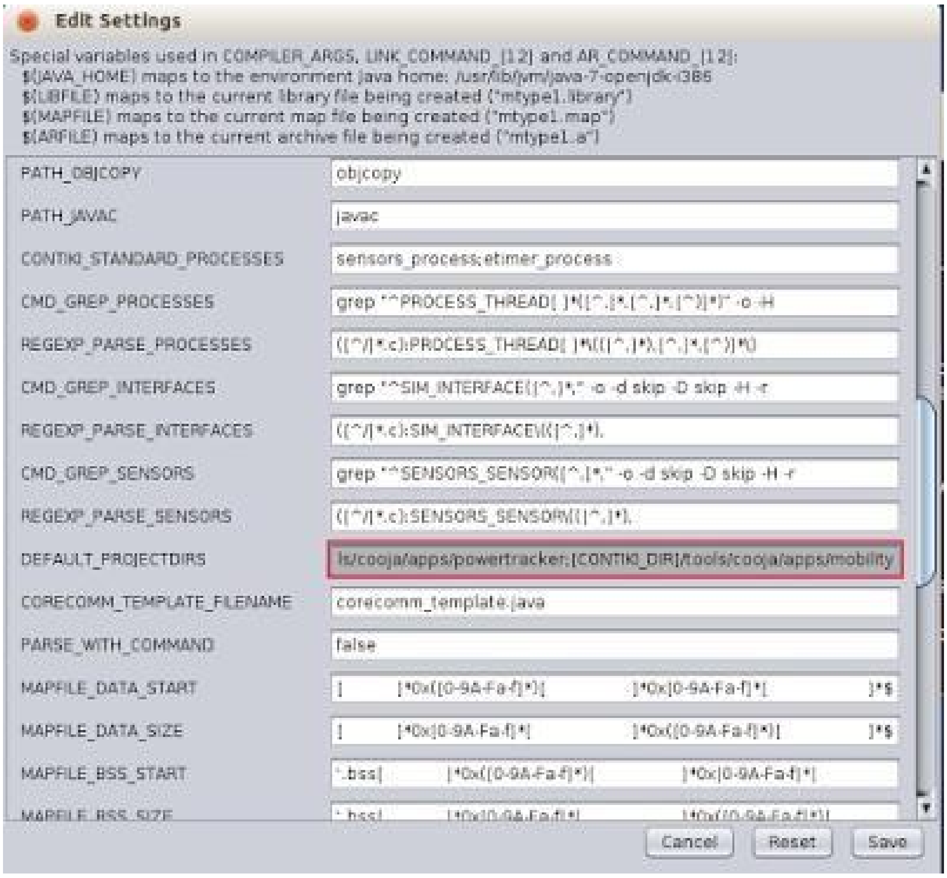
Making Mobility Possible

Cd cookja/tools/contiki sudo ant run to start the simulation

To get started with Cooja, go to Settings, then External Tool Path, and finally DEFAULT_PROJECTDIRS.

It is critical to include the entire set of Cooja Simulator plugin routes.To accomplish this, use the ';' sign to append the path of the mobility plugin to the current paths.

- This path must be added: [CONTIKI_DIR]/tools/cooja/apps/mobility.
- The changes have been saved. After completing the first two steps, restart the COOJA simulator. Navigate to Settings and select the Cooja Extension.



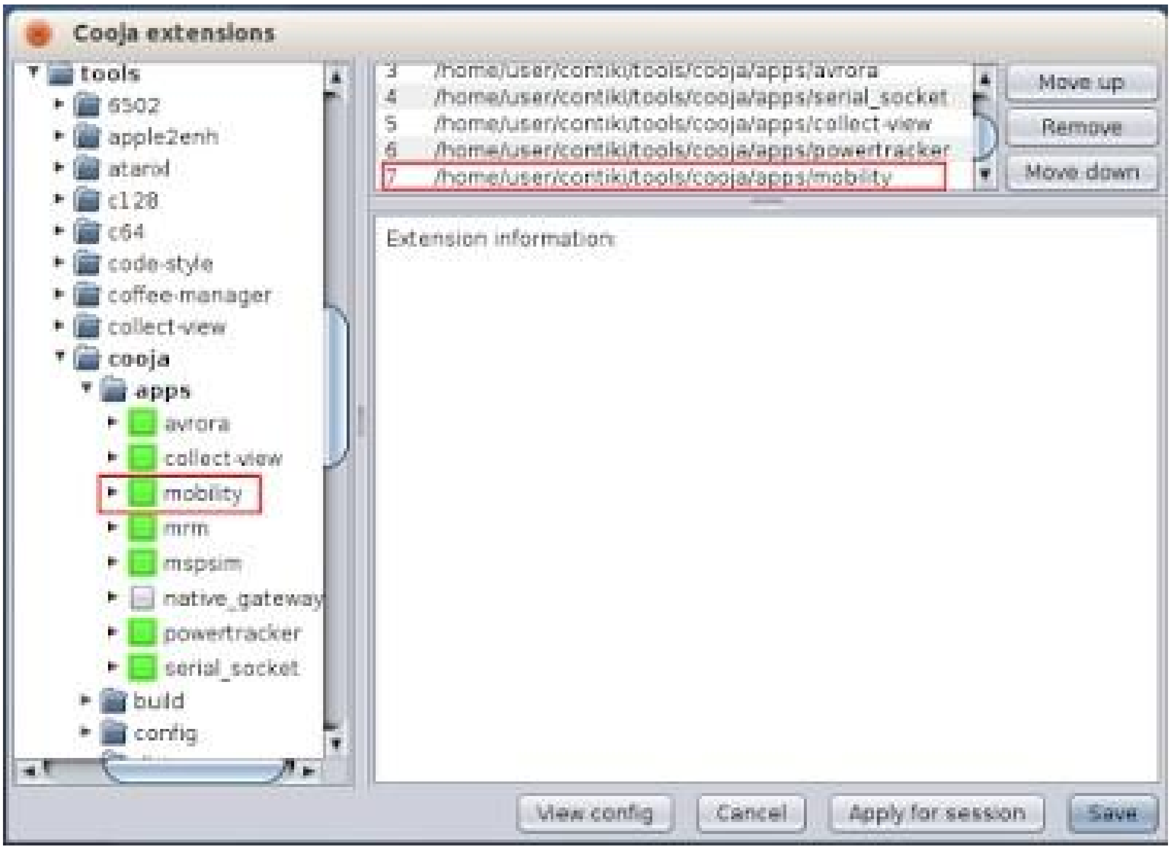
Special variables used in COMPILER_ARGS, LINK_COMMAND_[12] and AR_COMMAND_[12]:
$(JAVA_HOME) maps to the environment java home: /usr/lib/jvm/java-7-openjdk-i386
$(LIBFILE) maps to the current library file being created ("mtype1.library")
$(MAPFILE) maps to the current map file being created ("mtype1.map")
$(ARFILE) maps to the current archive file being created ("mtype1.a")

PATH_OBJCOPY	objcopy
PATH_JAVAC	javac
CONTIKI_STANDARD_PROCESSES	sensors_process;etimer_process
CMD_GREP_PROCESSES	grep "^PROCESS_THREAD[ ]*([^,]*,[^,]*,[^)]*)" -o -H
REGEXP_PARSE_PROCESSES	([^/]*.c):PROCESS_THREAD[ ]*(([^,]*),[^,]*,[^)]*)
CMD_GREP_INTERFACES	grep "^SIM_INTERFACE([^,]*," -o -d skip -D skip -H -r
REGEXP_PARSE_INTERFACES	([^/]*.c):SIM_INTERFACE\(([^,]*).
CMD_GREP_SENSORS	grep "^SENSORS_SENSOR([^,]*," -o -d skip -D skip -H -r
REGEXP_PARSE_SENSORS	([^/]*.c):SENSORS_SENSOR\(([^,]*).
DEFAULT_PROJECTDIRS	ls/cooja/apps/powertracker;[CONTIKI_DIR]/tools/cooja/apps/mobility
CORECOMM_TEMPLATE_FILENAME	corecomm_template.java
PARSE_WITH_COMMAND	false
MAPFILE_DATA_START	^.data[ ]*0x([0-9A-Fa-f]*)[ ]*0x[0-9A-Fa-f]*[ ]*$
MAPFILE_DATA_SIZE	^.data[ ]*0x[0-9A-Fa-f]*[ ]*0x([0-9A-Fa-f]*)[ ]*$
MAPFILE_BSS_START	^.bss[ ]*0x([0-9A-Fa-f]*)[ ]*0x[0-9A-Fa-f]*[
MAPFILE_BSS_SIZE	^.bss[ ]*0x[0-9A-Fa-f]*[ ]*0x([0-9A-Fa-f]*)[

fig 3.4 settings

- Cooja Sizes The window will either open or shut

- After that, select mobility once more. Fill out a session application



3 /home/user/contiki/tools/cooja/apps/avrora
4 /home/user/contiki/tools/cooja/apps/serial_socket
5 /home/user/contiki/tools/cooja/apps/collect-view
6 /home/user/contiki/tools/cooja/apps/powertracker
7 /home/user/contiki/tools/cooja/apps/mobility

fig 3.5 extenstions

- This would display the mobility plugins from the tools drop-down list.

- Under the Tools tab, a new option called Mobility

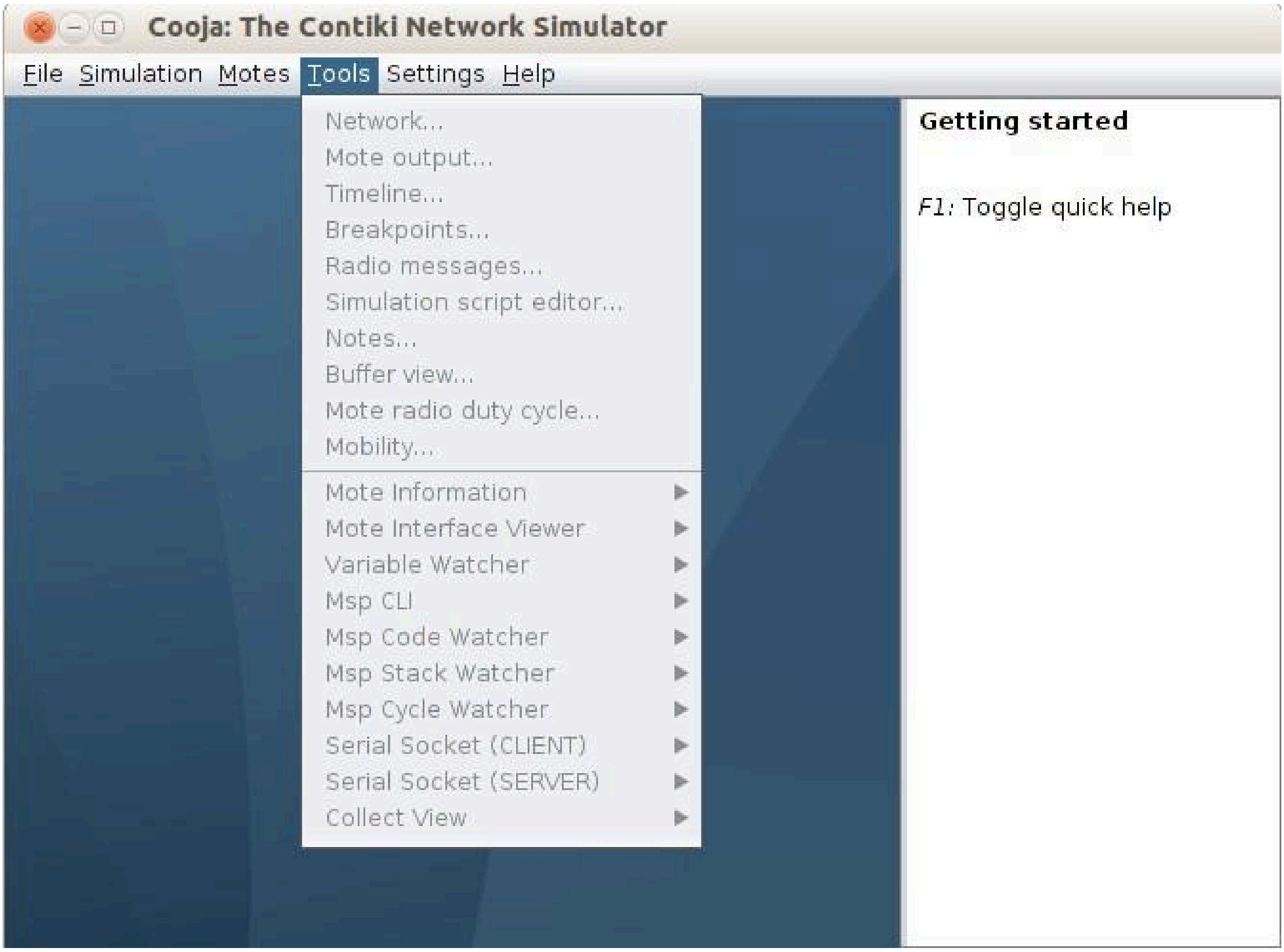


fig 3.6 mobility

Plugin is tested.

Start a new simulation.

Select a file and then click "New Simulation."This machine compiles C language code that includes hello world. This results in the formation of a mote.

Plugin testing on a particular mote

Click the Tools tab, then select Mobility from the Tools -> Mobility menu. To explore further, navigate to contiki/tools/cooja/apps/mobility/positions.dat.

We obtained the coordinate data from the positions.dat file using our mobility plugin and then clicked the Open button. A secondary window will appear, as shown below.

At the moment, select "Start Simulation." The motes will start moving in the specified directions.dat data file

3.4 Cooja Simulation

Here the DGRM model is applied.

The following are the steps to generate a replacement simulation:

Start Cooja

To start the simulation, click on the Contiki folder and travel to /tools/cooja directory and run "ant" to begin the COOJA simulation

sudo ant cooja

Open the simulation file

In the application, click on File->View

Sim->Browse Copy the repo from github

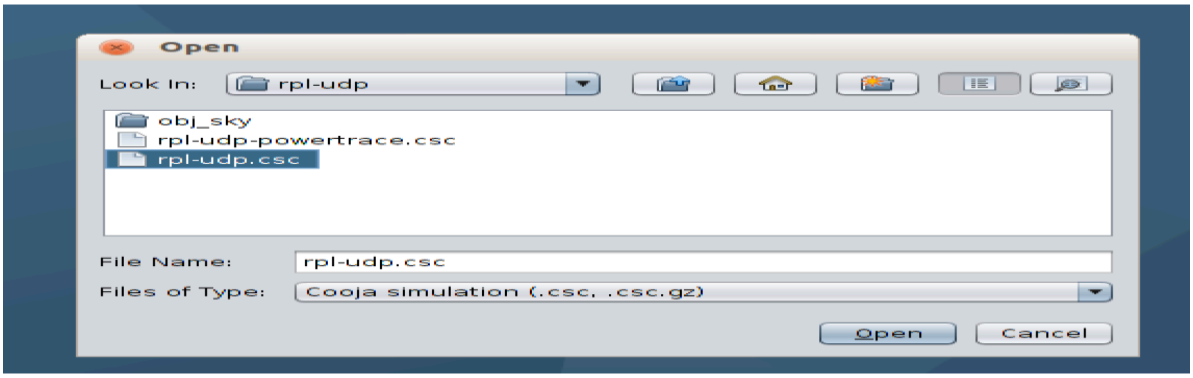


fig 3.7 implementation

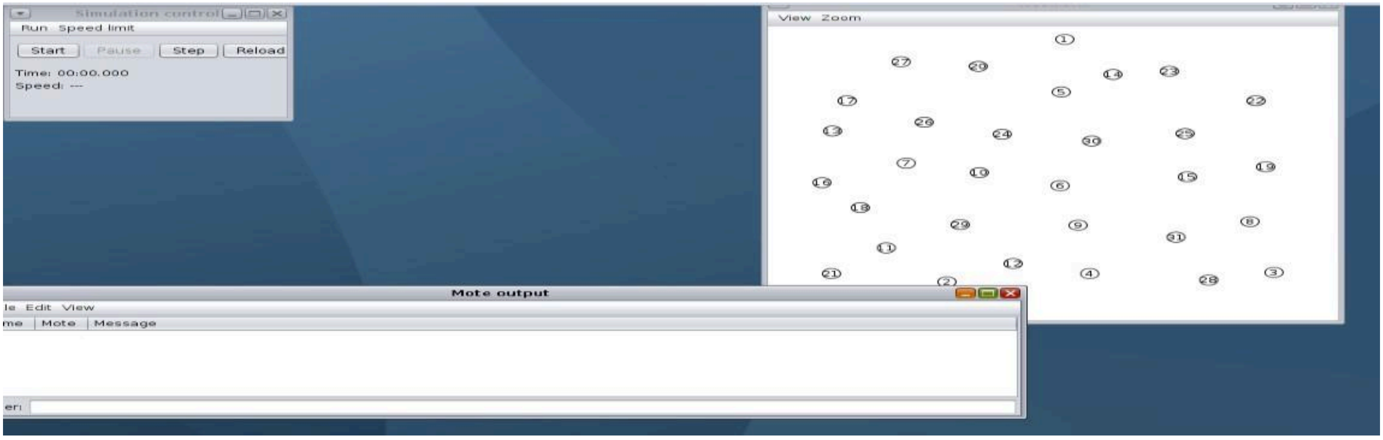


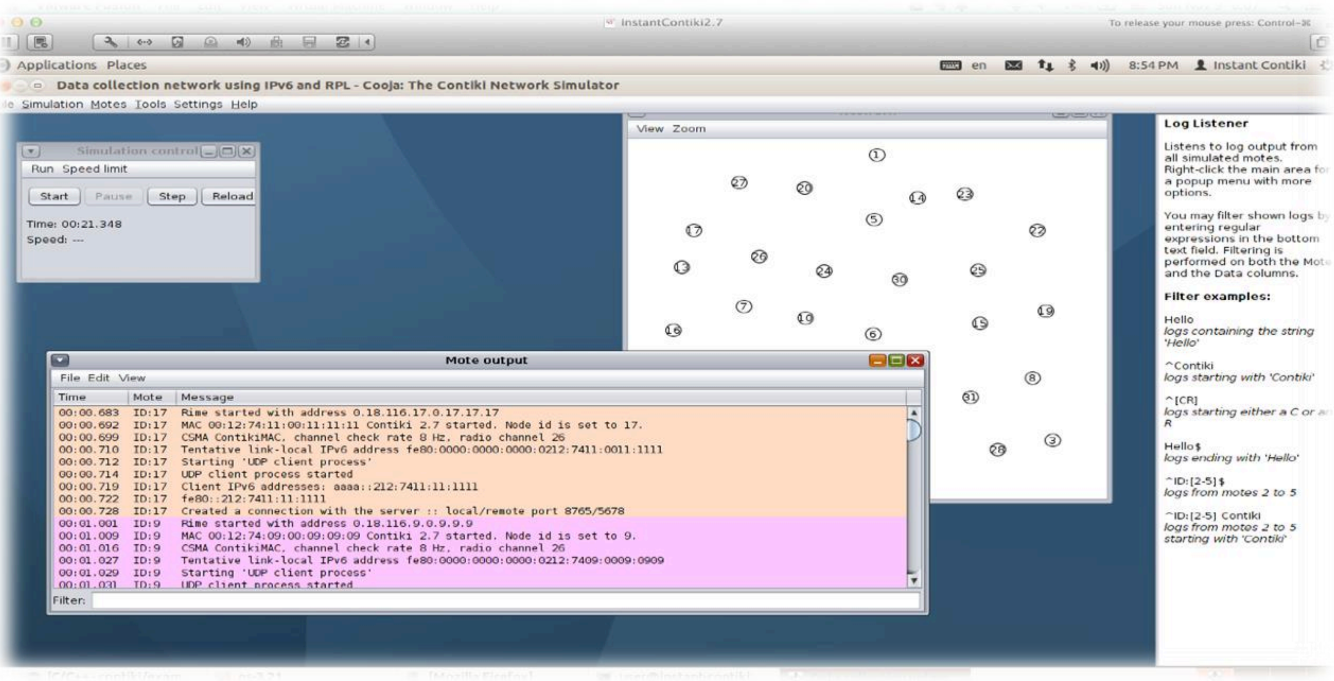
fig 3.8 implementation

Running Simulation

Start the SC-Simulation Control window by clicking its Start button. This initializes the motes and assigns each mote a new Rime address, along with any lingering startup processes.

3.5 Results

Mote Output Window



```
Time       Mote   Message
00:00.683  ID:17  Rime started with address 0.18.116.17.0.17.17.17
00:00.692  ID:17  MAC 00:12:74:11:00:11:11:11 Contiki 2.7 started. Node id is set to 17.
00:00.699  ID:17  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:00.710  ID:17  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7411:0011:1111
00:00.712  ID:17  Starting 'UDP client process'
00:00.714  ID:17  UDP client process started
00:00.719  ID:17  Client IPv6 addresses: aaaa::212:7411:11:1111
00:00.722  ID:17  fe80::212:7411:11:1111
00:00.728  ID:17  Created a connection with the server :: local/remote port 8765/5678
00:01.001  ID:9   Rime started with address 0.18.116.9.0.9.9.9
00:01.009  ID:9   MAC 00:12:74:09:00:09:09:09 Contiki 2.7 started. Node id is set to 9.
00:01.016  ID:9   CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:01.027  ID:9   Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7409:0009:0909
00:01.029  ID:9   Starting 'UDP client process'
00:01.031  ID:9   UDP client process started
```

Log Listener

Listens to log output from all simulated motes. Right-click the main area fo a popup menu with more options.

You may filter shown logs b entering regular expressions in the bottom text field. Filtering is performed on both the Mot and the Data columns.

Filter examples:

Hello
logs containing the string 'Hello'

^Contiki
logs starting with 'Contiki'

^[CR]
logs starting either a C or a R

Hello$
logs ending with 'Hello'

^ID:[2-5]$
logs from motes 2 to 5

^ID:[2-5] Contiki
logs from motes 2 to 5 starting with 'Contiki'

fig 3.9 implementation

The Mote Output window in Cooja is a vital tool for observing and analyzing the behavior of motes during a simulation. This window captures and displays output and debug messages generated by the motes, offering a comprehensive view of their activities and interactions within the network. The ability to monitor and interpret these outputs is essential for understanding the performance and functionality of the simulated network.

Key Capabilities of the Mote Output Window

1. **Filtering by Node ID:**
   - The Mote Output window includes a feature that allows users to filter messages based on node ID. This is particularly useful in simulations with a large number of motes, enabling users to focus on the output from specific nodes. By selecting a particular node ID, users can isolate and examine the messages generated by that mote, facilitating targeted debugging and analysis.
2. **File Option:**
   - The File option in the Mote Output window provides several functionalities. One of the most crucial features is the ability to save the output messages to a file. This function is essential for preserving the data generated during the simulation, allowing for post-simulation analysis and documentation. Saved outputs can be reviewed and compared with subsequent simulations, helping in iterative development and refinement of network protocols.
3. **Edit Option:**
   - The Edit option offers capabilities for manipulating the output messages. Users can copy the entire output or select specific messages to copy, which is useful for detailed analysis and reporting. Additionally, the Clear all messages option is available to clear the entire output window. This feature is beneficial when starting a new simulation or when users want to remove clutter from the previous outputs, restoring the workspace to a clean state.
4. **View Option:**
   - The View option provides functionalities to customize the display of output messages. Users can adjust the view settings to enhance readability and focus on specific aspects of the output. This customization is crucial for efficiently navigating through large volumes of data generated during complex simulations.

Utilizing Output Messages for Analysis

The messages saved in the output file play a crucial role in the analysis phase of the project. These messages provide detailed insights into the operations and performance of the motes, including error messages, status updates, and other debug information. The recorded output can be used to:

1. **Document Observations:**
   - Detailed logs of mote outputs allow researchers to document observations accurately. These logs serve as a record of the simulation, capturing the behavior and responses of the motes under different conditions. This documentation is essential for identifying patterns, diagnosing issues, and validating the effectiveness of network protocols.
2. **Construct Graphs and Visualizations:**
   - The saved output messages can be processed to create graphs and visual representations of the data. Graphs are particularly useful for illustrating trends, comparing performance metrics, and highlighting key findings. Visualizations can simplify complex data, making it easier to communicate results and support conclusions.
3. **Project Goal Alignment:**
   - Aligning the analysis with the project's goals is critical. The output messages help ensure that the simulation results are relevant to the project's objectives. By correlating the outputs with the intended outcomes, researchers can evaluate whether the network protocols meet the desired performance criteria and identify areas for improvement.

Implications for Network Performance and Protocol Development

The detailed output and debug messages provided by the Mote Output window are instrumental in evaluating the performance of IoT and WSN protocols. They offer a granular view of how motes interact, communicate, and respond to network conditions, enabling researchers to:

1. **Identify Performance Bottlenecks:**
   - Analyzing the output messages helps in pinpointing specific areas where the network may be experiencing performance issues. Identifying bottlenecks allows researchers to focus on optimizing those aspects, improving overall network efficiency.
2. **Enhance Protocol Robustness:**

   - By studying the debug messages and errors, developers can enhance the robustness of the protocols. Understanding the root causes of failures or suboptimal performance enables the refinement of algorithms and communication strategies, leading to more reliable network operations.

3. **Validate Network Scalability:**
   - The output messages provide insights into how the network behaves as it scales. This is crucial for validating the scalability of the protocols, ensuring that they can handle an increasing number of nodes and higher traffic volumes without degradation in performance.

Chapter 4 - Analysis

4.1 Overview

This chapter delves into a detailed performance analysis of scenarios with and without mobility enabled for IoT networks. The study hinges on the examination of two specific files: `testing_mobility` and `mobility_project`. The `testing_mobility` file is designed to investigate the behavior of motes in a stationary setup, where no mobility is involved. This setup allows for a baseline understanding of how motes operate and interact when they remain fixed in one location, providing a control scenario against which changes can be measured.

In contrast, the `mobility_project` file incorporates a mobility plugin, introducing dynamic movement into the network. This file utilizes a `positions.dat` file to define various Cartesian coordinates, mapping out the different locations of motes within the network graph. By simulating the movement of motes through predefined paths, this setup aims to mimic real-world conditions where devices often change positions.

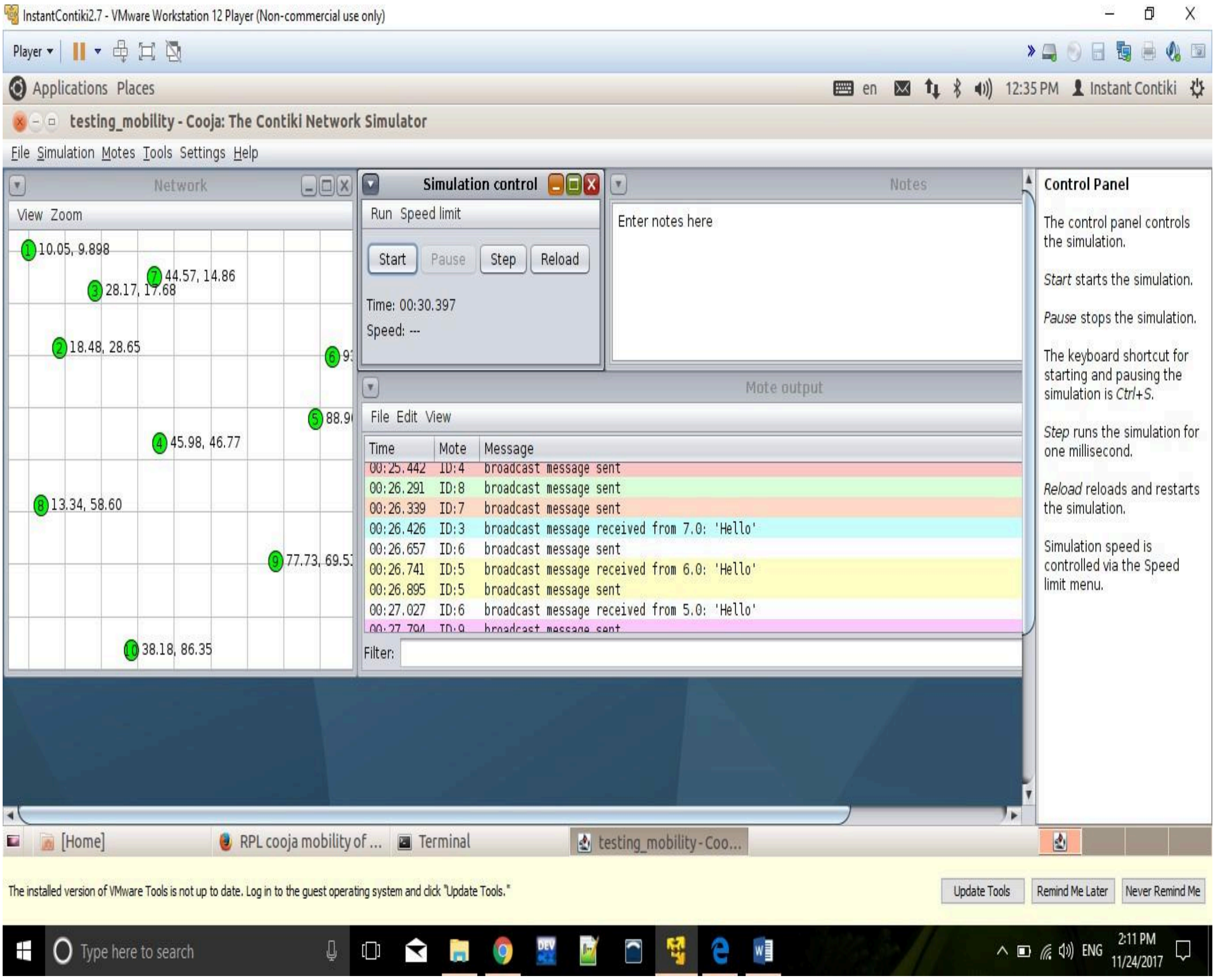
The analysis will compare the two scenarios, highlighting how mobility impacts network performance. Specifically, the study will look at how altering the `positions.dat` file to update mote locations at specific intervals affects transmission rates and overall network throughput. This dynamic positioning is crucial for understanding the adaptability and robustness of the network in mobile environments, where devices frequently move and interact in complex ways.

4.2 Comparative analysis

Through this comparative analysis, insights into the efficiency, reliability, and scalability of IoT networks under varying conditions will be gained. The findings will help in identifying potential challenges and opportunities for optimizing network protocols to better support mobile devices, ultimately enhancing the performance and resilience of IoT networks in real-world applications.

mobility.csc

1. Begin a new simulation, as explained in the last chapter, and set 10 sky motes at random locations.



Time: 00:30.397

Speed: ---

Enter notes here

```
Time       Mote  Message
00:25.442  ID:4  broadcast message sent
00:26.291  ID:8  broadcast message sent
00:26.339  ID:7  broadcast message sent
00:26.426  ID:3  broadcast message received from 7.0: 'Hello'
00:26.657  ID:6  broadcast message sent
00:26.741  ID:5  broadcast message received from 6.0: 'Hello'
00:26.895  ID:5  broadcast message sent
00:27.027  ID:6  broadcast message received from 5.0: 'Hello'
00:27.794  ID:9  broadcast message sent
```

Control Panel

The control panel controls the simulation.

Start starts the simulation.

Pause stops the simulation.

The keyboard shortcut for starting and pausing the simulation is *Ctrl+S*.

Step runs the simulation for one millisecond.

Reload reloads and restarts the simulation.

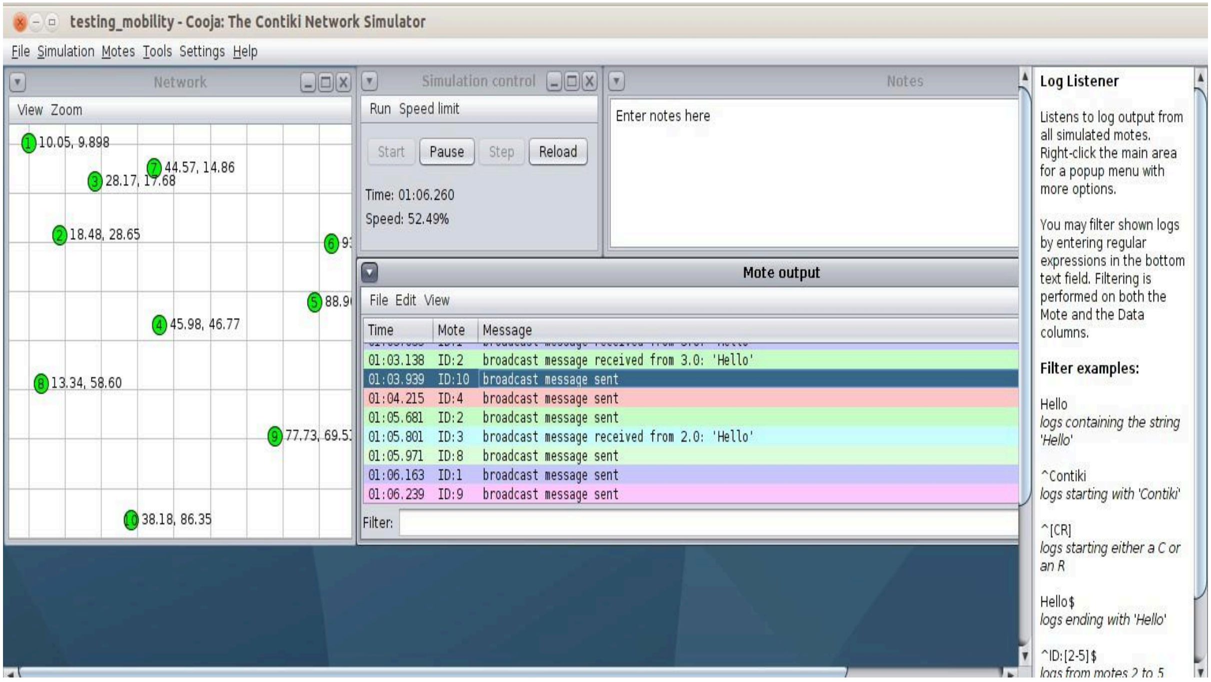
Simulation speed is controlled via the Speed limit menu.

The installed version of VMware Tools is not up to date. Log in to the guest operating system and click "Update Tools."

fig 4.1 analysis

2. Click Start in the simulation control box. Enable Radio Traffic, Moto Type, Positions, and Radio Environment on the Network view screen to diagnose the issue.When movement is disabled:

In the next screenshots, the behavior is depicted, and several result factors are investigated **without mobility.**



Time: 01:06.260
Speed: 52.49%

Enter notes here

```
Time       Mote   Message
01:03.138  ID:2   broadcast message received from 3.0: 'Hello'
01:03.939  ID:10  broadcast message sent
01:04.215  ID:4   broadcast message sent
01:05.681  ID:2   broadcast message sent
01:05.801  ID:3   broadcast message received from 2.0: 'Hello'
01:05.971  ID:8   broadcast message sent
01:06.163  ID:1   broadcast message sent
01:06.239  ID:9   broadcast message sent
```

Log Listener

Listens to log output from all simulated motes. Right-click the main area for a popup menu with more options.

You may filter shown logs by entering regular expressions in the bottom text field. Filtering is performed on both the Mote and the Data columns.

Filter examples:

Hello
logs containing the string 'Hello'

^Contiki
logs starting with 'Contiki'

^[CR]
logs starting either a C or an R

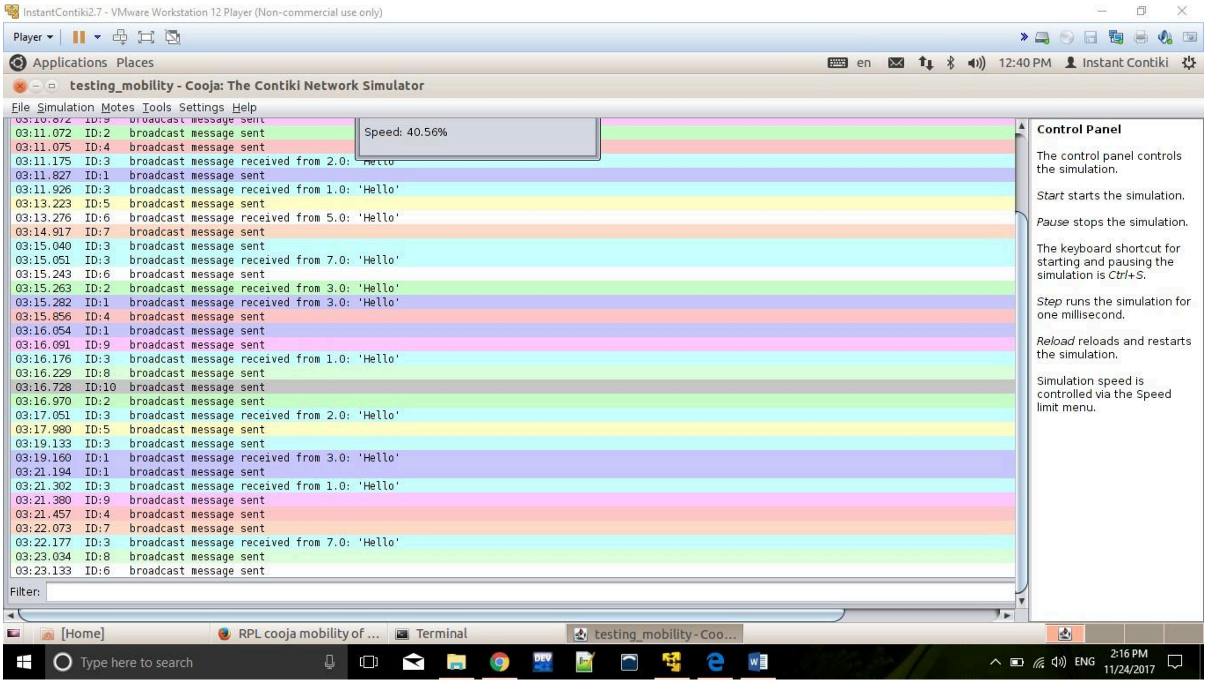
Hello$
logs ending with 'Hello'

^ID:[2-5]$
logs from motes 2 to 5

fig 4.2 analysis

Speed: 52.48% - this evaluates the transfer speed's efficiency.

Mote Output: This illustrates the development of connections between distinct motes, the messages that are transmitted between them, and the behavior of attributes such as speed over varying message transmission durations.



testing_mobility - Cooja: The Contiki Network Simulator

Speed: 40.56%

```
03:11.072   ID:2   broadcast message sent
03:11.075   ID:4   broadcast message sent
03:11.175   ID:3   broadcast message received from 2.0: 'Hello'
03:11.827   ID:1   broadcast message sent
03:11.926   ID:3   broadcast message received from 1.0: 'Hello'
03:13.223   ID:5   broadcast message sent
03:13.276   ID:6   broadcast message received from 5.0: 'Hello'
03:14.917   ID:7   broadcast message sent
03:15.040   ID:3   broadcast message sent
03:15.051   ID:3   broadcast message received from 7.0: 'Hello'
03:15.243   ID:6   broadcast message sent
03:15.263   ID:2   broadcast message received from 3.0: 'Hello'
03:15.282   ID:1   broadcast message received from 3.0: 'Hello'
03:15.856   ID:4   broadcast message sent
03:16.054   ID:1   broadcast message sent
03:16.091   ID:9   broadcast message sent
03:16.176   ID:3   broadcast message received from 1.0: 'Hello'
03:16.229   ID:8   broadcast message sent
03:16.728   ID:10  broadcast message sent
03:16.970   ID:2   broadcast message sent
03:17.051   ID:3   broadcast message received from 2.0: 'Hello'
03:17.980   ID:5   broadcast message sent
03:19.133   ID:3   broadcast message sent
03:19.160   ID:1   broadcast message received from 3.0: 'Hello'
03:21.194   ID:1   broadcast message sent
03:21.302   ID:3   broadcast message received from 1.0: 'Hello'
03:21.380   ID:9   broadcast message sent
03:21.457   ID:4   broadcast message sent
03:22.073   ID:7   broadcast message sent
03:22.177   ID:3   broadcast message received from 7.0: 'Hello'
03:23.034   ID:8   broadcast message sent
03:23.133   ID:6   broadcast message sent
```

Control Panel

The control panel controls the simulation.

Start starts the simulation.

Pause stops the simulation.

The keyboard shortcut for starting and pausing the simulation is *Ctrl+S*.

Step runs the simulation for one millisecond.

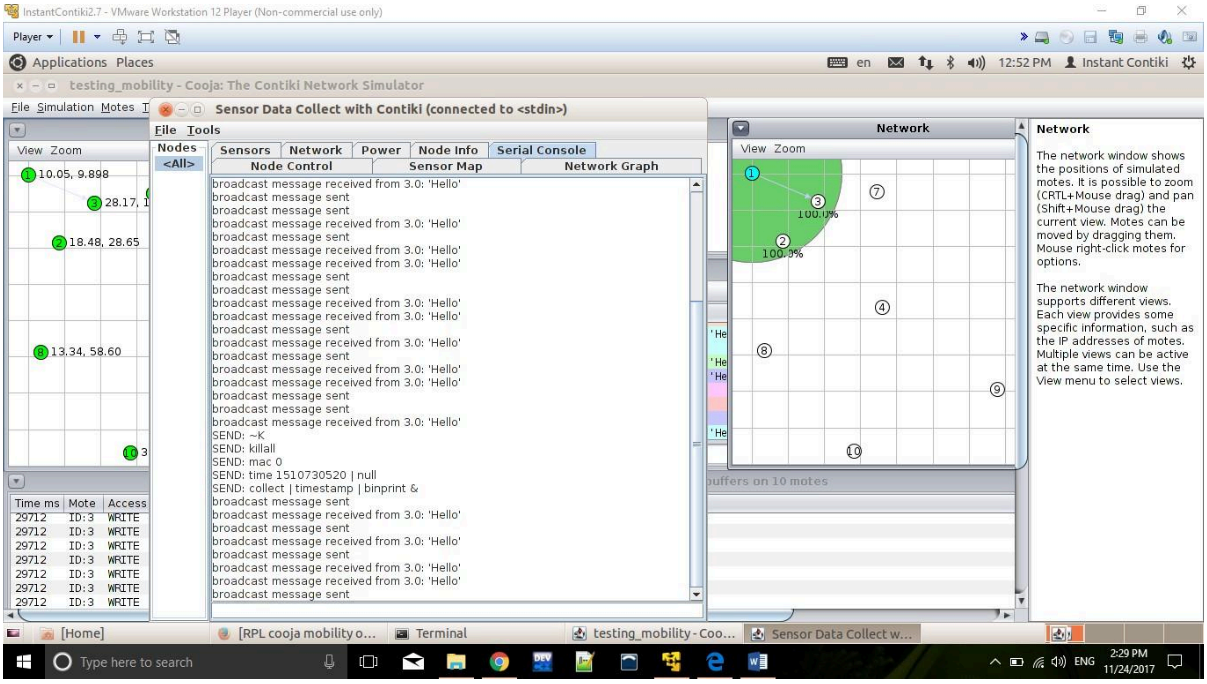
Reload reloads and restarts the simulation.

Simulation speed is controlled via the Speed limit menu.

InstantContiki2.7 - VMware Workstation 12 Player (Non-commercial use only)
Player
Applications Places
12:40 PM
Instant Contiki
File Simulation Motes Tools Settings Help
Filter:
[Home]
RPL cooja mobility of ...
Terminal
testing_mobility - Coo...
Type here to search
2:16 PM
11/24/2017

fig 4.3 analysis

Serial Console Specifications:



Sensor Data Collect with Contiki (connected to <stdin>)

```
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message sent
broadcast message received from 3.0: 'Hello'
SEND: ~K
SEND: killall
SEND: mac 0
SEND: time 1510730520 | null
SEND: collect | timestamp | binprint &
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message sent
broadcast message received from 3.0: 'Hello'
broadcast message received from 3.0: 'Hello'
broadcast message sent
```

Time ms	Mote	Access
29712	ID:3	WRITE
29712	ID:3	WRITE
29712	ID:3	WRITE
29712	ID:3	WRITE
29712	ID:3	WRITE
29712	ID:3	WRITE
29712	ID:3	WRITE

Network

The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

fig 4.4 analysis

The first file, `testing_mobility`, serves as our baseline scenario where mobility is disabled. This setup allows us to observe the behavior and performance of stationary motes. By keeping the motes fixed in place, we can analyze how the network operates when there is no movement and all nodes remain static. This provides a control environment to understand the fundamental performance characteristics of the network without the added complexity of mobility.

In contrast, the second file, `mobility_project`, introduces mobility into the simulation. Here, we enable the mobility plugin and define the positions.dat file, which specifies the Cartesian coordinates of each mote at various points in time. This setup is crucial for simulating a dynamic network environment where motes are constantly moving. By updating the positions.dat file at regular intervals, we can observe how the network adapts to changes in the motes' locations, which in turn affects transmission rates and overall network throughput.

The green zone depicted on the network graph represents the effective transmission range of the motes. This range is adjustable to suit specific network requirements. To customize the range:

- right-click on the appropriate mote
- navigate to the Transmission Range Changes section
- adjust the values in the dialog box that appears.

This allows us to tailor the communication range to match the desired network parameters.

During our simulations, we observed that messages sometimes fail to transmit successfully when the distance between the source and destination exceeds the designated transmission range of the motes. This issue is highlighted by the serial output, where certain messages are dropped due to motes being out of each other's communication range. Such failures underscore the limitations of static networks in scenarios where nodes are spread out beyond their communication capabilities.

To mitigate this problem, we incorporate mobility into our simulations. The `mobility_project` file enables motes to move according to the predefined coordinates in the positions.dat file. By updating this file at regular intervals, we ensure that the motes' positions are accurately reflected throughout the simulation. This dynamic adjustment allows motes to reposition themselves, thereby staying within effective communication range and reducing the likelihood of message drops.

Enabling mobility transforms the network's behavior significantly. Motes can dynamically adjust their positions, forming new connections and routes as needed. This ability to move and adapt enhances the network's robustness and resilience, particularly in scenarios where fixed nodes would otherwise face communication challenges. By analyzing the simulation results, we can quantify the improvements in network performance brought about by mobility, including increased transmission success rates, higher throughput, and better overall efficiency.

By comparing the stationary and mobile scenarios, we can identify key benefits and challenges associated with enabling mobility in IoT networks. This analysis will inform best practices for designing and managing networks that need to support mobile devices, ensuring optimal performance and reliability in various application contexts.

By conducting these detailed simulations and analyses, we can better understand the dynamics of IoT networks under different conditions. The insights gained from this study will be invaluable for network designers and engineers working to optimize IoT deployments for both stationary and mobile environments.

Organization of the Location.dat File

The Position.dat file provides the
information #node time(s) x y

```
0 0.0 0 20
0 1.0 10 20
```

0 2.0 12 0

0 3.0 0 10

0 4.0 0 0

Column 1 carries the node number, column 3 includes the x coordinate, column 2 contains the time stamp, and column 4 contains the y coordinate.

#node x y 000020 time(s)

It states that node 0 (i.e., mote 1) will be positioned with position coordinates (0,20) at 0.0 seconds.

Please keep in mind that in this circumstance, node 0 will reflect the Cooja's "mote 1." If we indicate it in the first column, node 1 will be for mote 2.

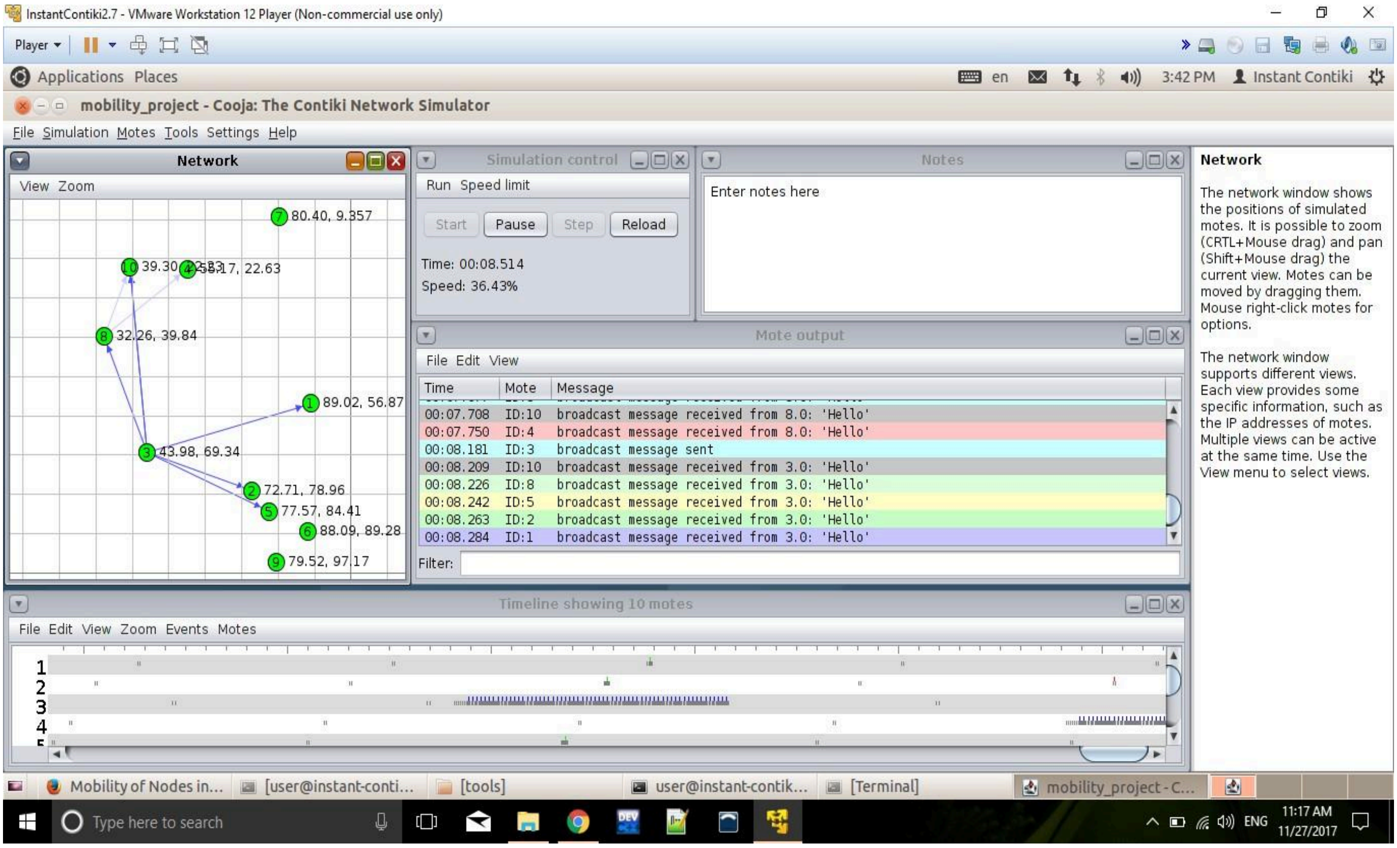
On Cooja, the position.dat node (n) is generally mote (n+1).

A change to the position.dat file is necessary. Replace the old file with the new one under **Tools>Mobility**

At this phase, add as many motes as necessary to start the simulation.

After the newly determined positions are loaded, the nodes will move in line with them.

Additional screenshots demonstrate COOJA's mobility, starting with 10 randomly positioned motes:



InstantContiki2.7 - VMware Workstation 12 Player (Non-commercial use only)

mobility_project - Cooja: The Contiki Network Simulator

Time: 00:08.514

Speed: 36.43%

Enter notes here

```
Time       Mote   Message
00:07.708  ID:10  broadcast message received from 8.0: 'Hello'
00:07.750  ID:4   broadcast message received from 8.0: 'Hello'
00:08.181  ID:3   broadcast message sent
00:08.209  ID:10  broadcast message received from 3.0: 'Hello'
00:08.226  ID:8   broadcast message received from 3.0: 'Hello'
00:08.242  ID:5   broadcast message received from 3.0: 'Hello'
00:08.263  ID:2   broadcast message received from 3.0: 'Hello'
00:08.284  ID:1   broadcast message received from 3.0: 'Hello'
```

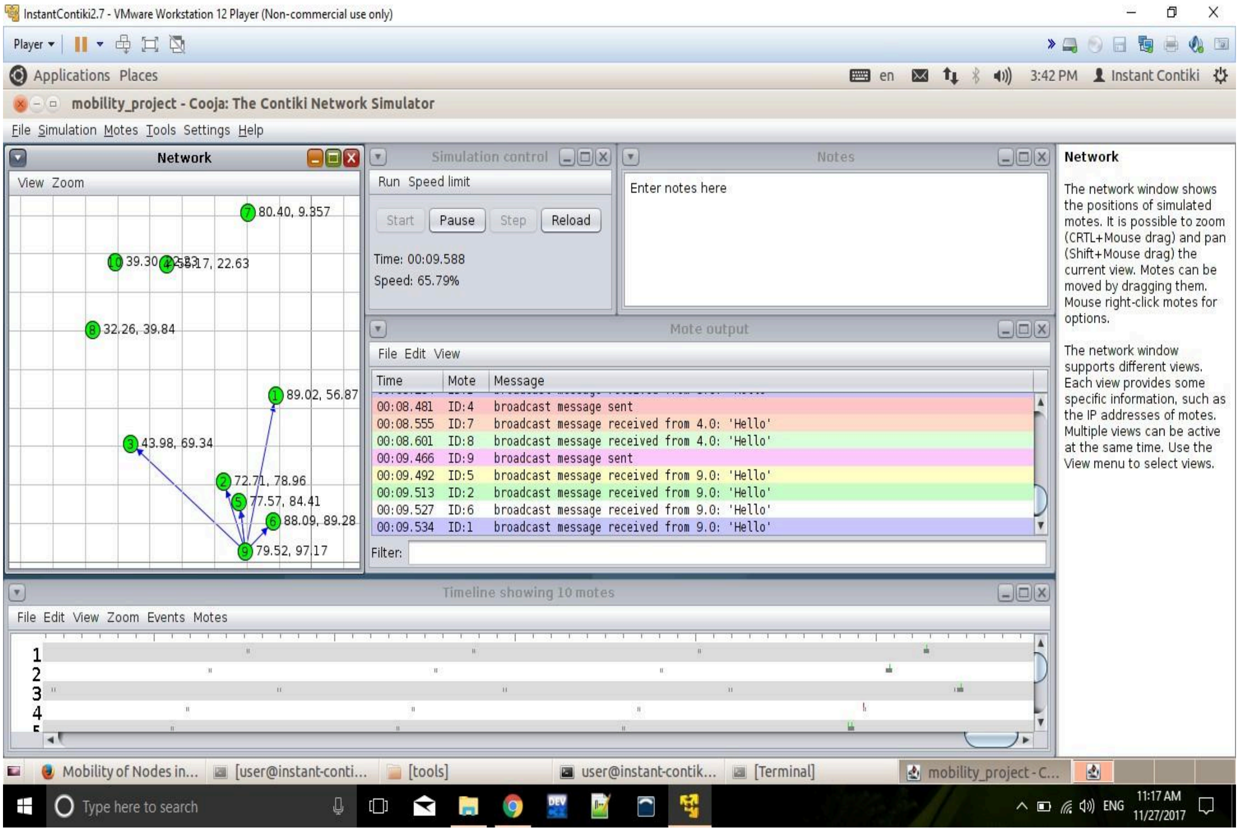
Network

The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

fig 4.5 analysis

The first is the initial position interactions of the motes.



Time: 00:09.588
Speed: 65.79%

Enter notes here

```
Time        Mote   Message
00:08.481   ID:4   broadcast message sent
00:08.555   ID:7   broadcast message received from 4.0: 'Hello'
00:08.601   ID:8   broadcast message received from 4.0: 'Hello'
00:09.466   ID:9   broadcast message sent
00:09.492   ID:5   broadcast message received from 9.0: 'Hello'
00:09.513   ID:2   broadcast message received from 9.0: 'Hello'
00:09.527   ID:6   broadcast message received from 9.0: 'Hello'
00:09.534   ID:1   broadcast message received from 9.0: 'Hello'
```

Network

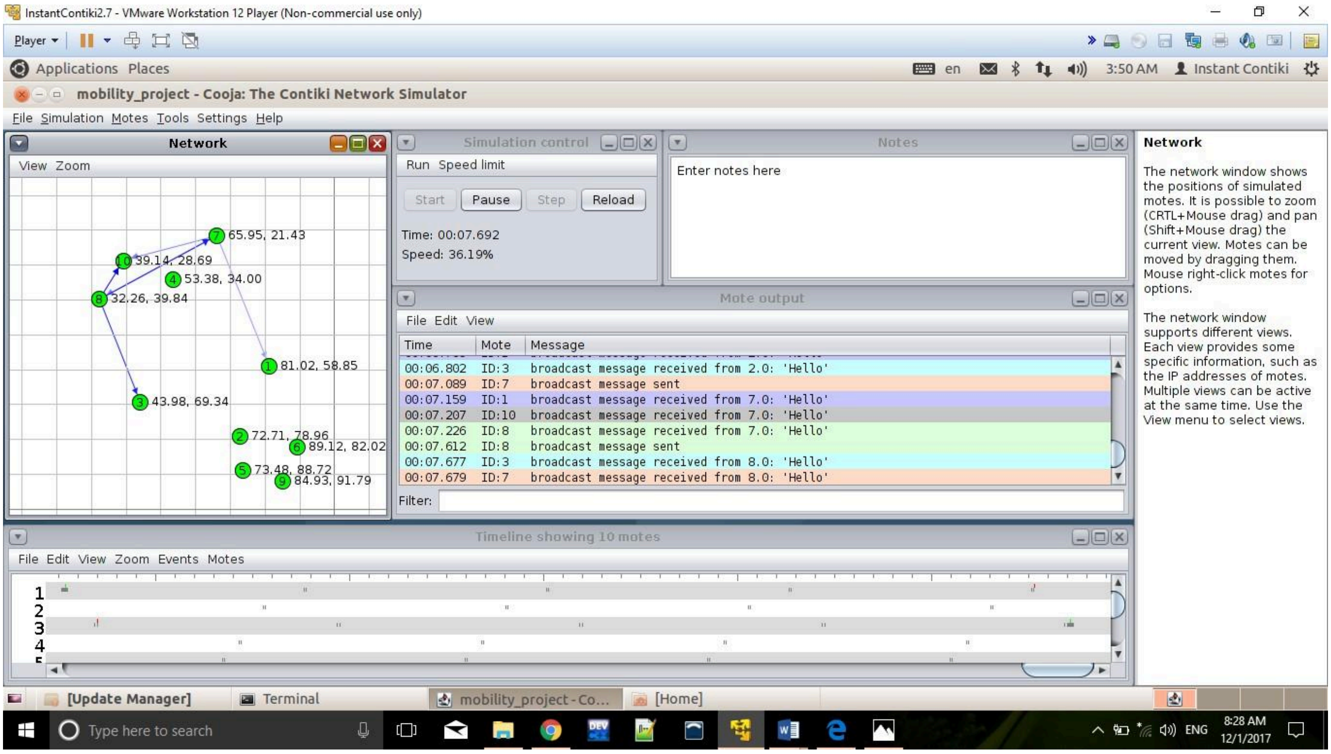
The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

fig 4.6 analysis

The mote network range restriction, as seen above, inhibits some of the motes from communicating with one another.

Additional screenshots show how positions vary when mobility is enabled.



Time: 00:07.692
Speed: 36.19%

```
Time       Mote   Message
00:06.802  ID:3   broadcast message received from 2.0: 'Hello'
00:07.089  ID:7   broadcast message sent
00:07.159  ID:1   broadcast message received from 7.0: 'Hello'
00:07.207  ID:10  broadcast message received from 7.0: 'Hello'
00:07.226  ID:8   broadcast message received from 7.0: 'Hello'
00:07.612  ID:8   broadcast message sent
00:07.677  ID:3   broadcast message received from 8.0: 'Hello'
00:07.679  ID:7   broadcast message received from 8.0: 'Hello'
Filter:
```

Network

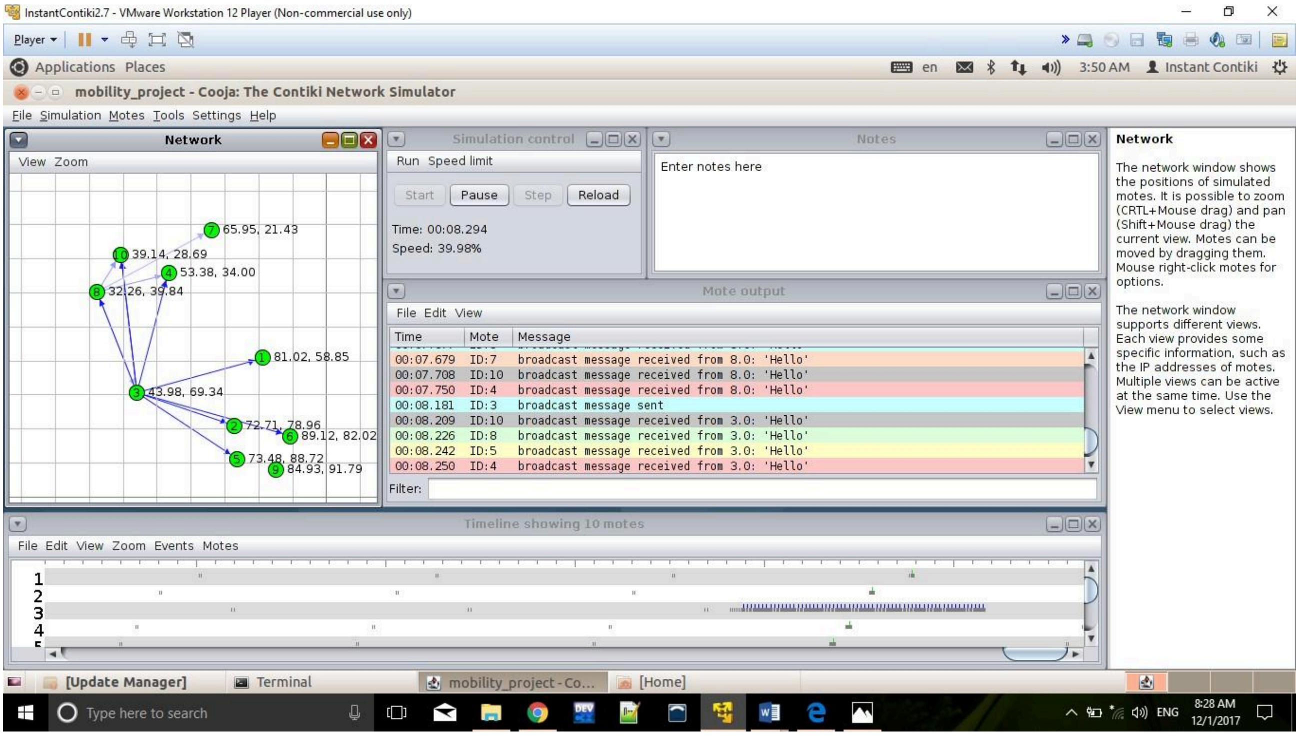
The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

fig 4.7 analysis

This displays the first stage, when the motes are barely getting started.

For the first time, motes have moved to establish contact with one another.



Time: 00:08.294
Speed: 39.98%

```
00:07.679  ID:7   broadcast message received from 8.0: 'Hello'
00:07.708  ID:10  broadcast message received from 8.0: 'Hello'
00:07.750  ID:4   broadcast message received from 8.0: 'Hello'
00:08.181  ID:3   broadcast message sent
00:08.209  ID:10  broadcast message received from 3.0: 'Hello'
00:08.226  ID:8   broadcast message received from 3.0: 'Hello'
00:08.242  ID:5   broadcast message received from 3.0: 'Hello'
00:08.250  ID:4   broadcast message received from 3.0: 'Hello'
```

Network

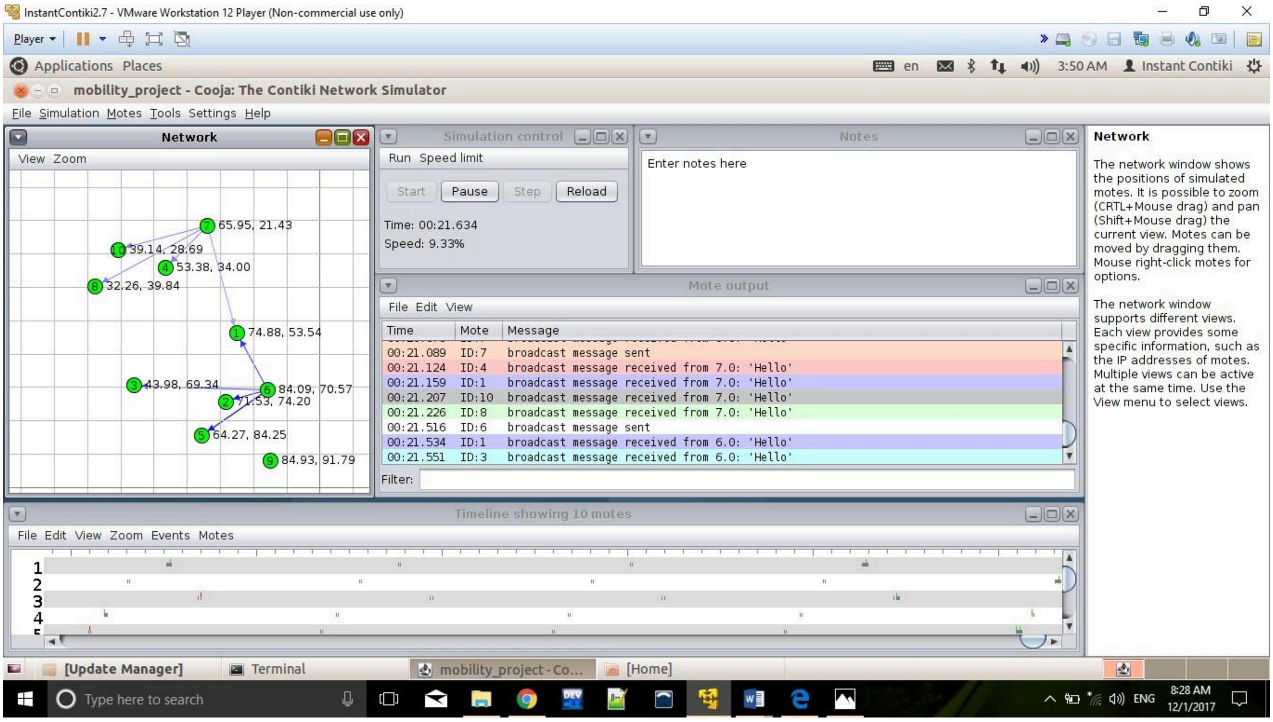
The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

fig 4.8 analysis

Motes continue to move in order to be closer to one another.

For the first time, motes have moved to establish contact with one another.



Time: 00:21.634
Speed: 9.33%

```
00:21.089  ID:7   broadcast message sent
00:21.124  ID:4   broadcast message received from 7.0: 'Hello'
00:21.159  ID:1   broadcast message received from 7.0: 'Hello'
00:21.207  ID:10  broadcast message received from 7.0: 'Hello'
00:21.226  ID:8   broadcast message received from 7.0: 'Hello'
00:21.516  ID:6   broadcast message sent
00:21.534  ID:1   broadcast message received from 6.0: 'Hello'
00:21.551  ID:3   broadcast message received from 6.0: 'Hello'
```

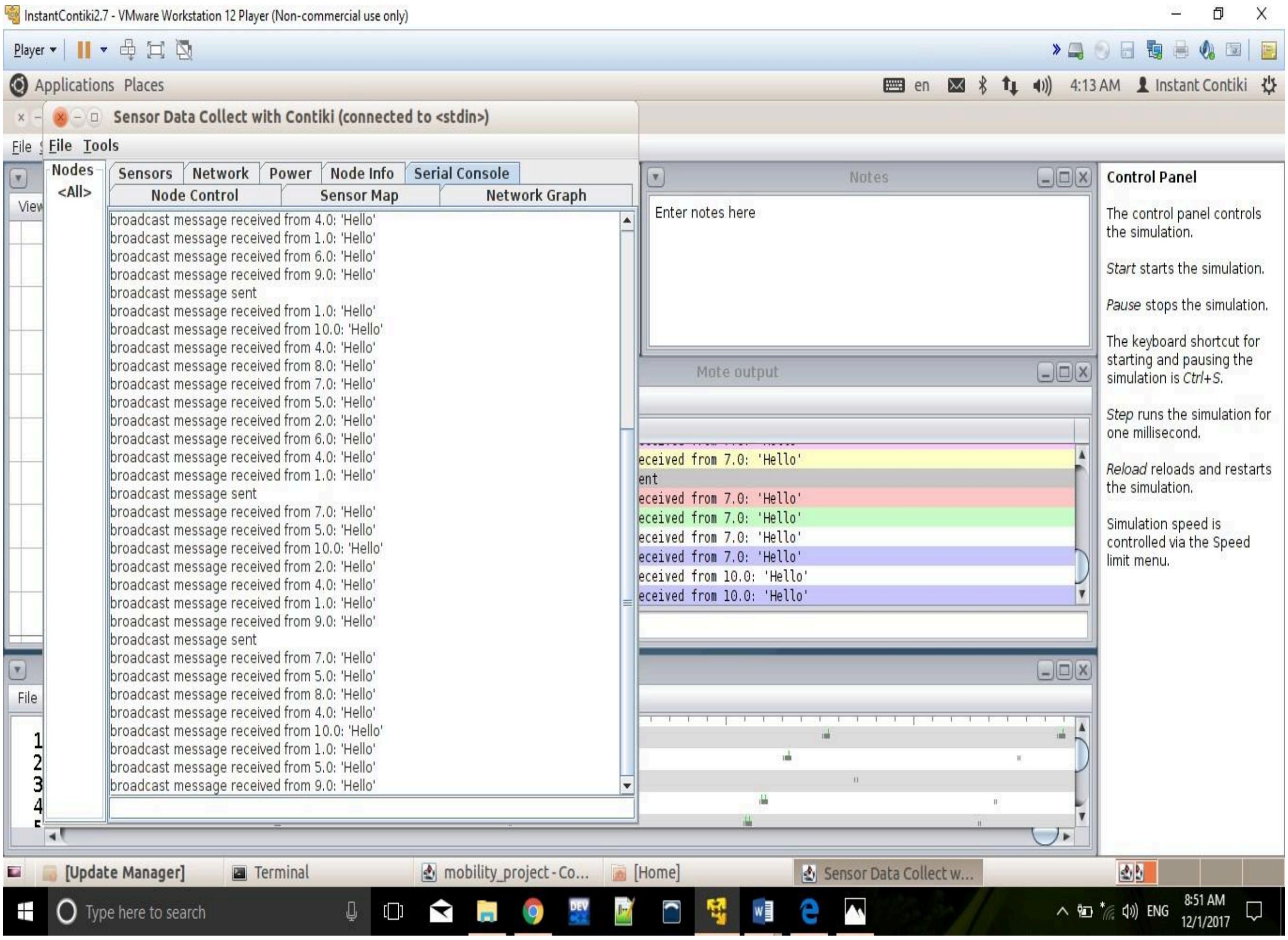
Network

The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

fig 4.9 analysis

The Sensor Data Collect with Contiki screen capture that follows allows us to deduce that increased mobility has boosted efficiency because all of the motes can now connect and deliver data.



```
broadcast message received from 4.0: 'Hello'
broadcast message received from 1.0: 'Hello'
broadcast message received from 6.0: 'Hello'
broadcast message received from 9.0: 'Hello'
broadcast message sent
broadcast message received from 1.0: 'Hello'
broadcast message received from 10.0: 'Hello'
broadcast message received from 4.0: 'Hello'
broadcast message received from 8.0: 'Hello'
broadcast message received from 7.0: 'Hello'
broadcast message received from 5.0: 'Hello'
broadcast message received from 2.0: 'Hello'
broadcast message received from 6.0: 'Hello'
broadcast message received from 4.0: 'Hello'
broadcast message received from 1.0: 'Hello'
broadcast message sent
broadcast message received from 7.0: 'Hello'
broadcast message received from 5.0: 'Hello'
broadcast message received from 10.0: 'Hello'
broadcast message received from 2.0: 'Hello'
broadcast message received from 4.0: 'Hello'
broadcast message received from 1.0: 'Hello'
broadcast message received from 9.0: 'Hello'
broadcast message sent
broadcast message received from 7.0: 'Hello'
broadcast message received from 5.0: 'Hello'
broadcast message received from 8.0: 'Hello'
broadcast message received from 4.0: 'Hello'
broadcast message received from 10.0: 'Hello'
broadcast message received from 1.0: 'Hello'
broadcast message received from 5.0: 'Hello'
broadcast message received from 9.0: 'Hello'
```

Enter notes here

```
eceived from 7.0: 'Hello'
ent
eceived from 7.0: 'Hello'
eceived from 7.0: 'Hello'
eceived from 7.0: 'Hello'
eceived from 7.0: 'Hello'
eceived from 10.0: 'Hello'
eceived from 10.0: 'Hello'
```

Control Panel

The control panel controls the simulation.

Start starts the simulation.

Pause stops the simulation.

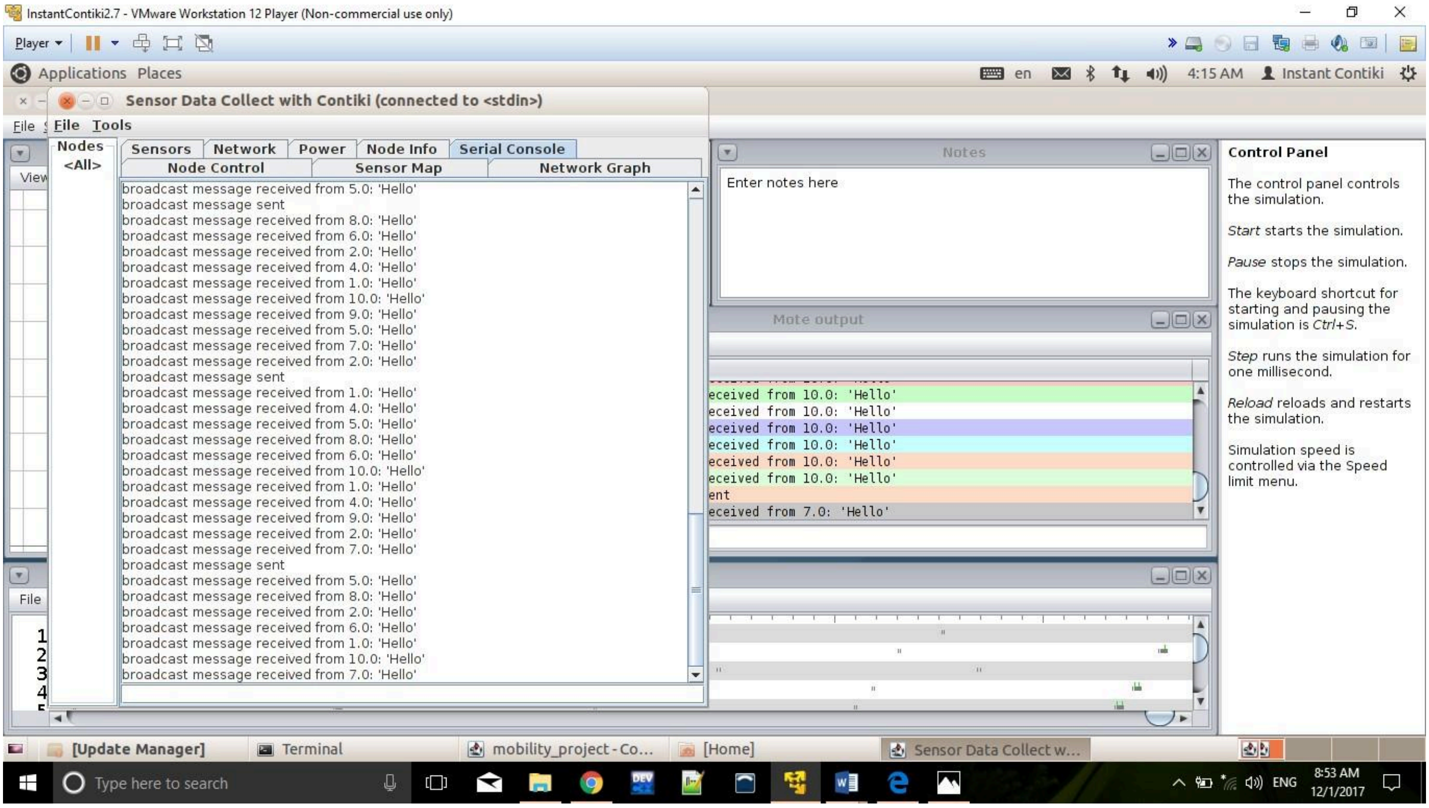
The keyboard shortcut for starting and pausing the simulation is *Ctrl+S*.

Step runs the simulation for one millisecond.

Reload reloads and restarts the simulation.

Simulation speed is controlled via the Speed limit menu.

fig 4.10 analysis



```
broadcast message received from 5.0: 'Hello'
broadcast message sent
broadcast message received from 8.0: 'Hello'
broadcast message received from 6.0: 'Hello'
broadcast message received from 2.0: 'Hello'
broadcast message received from 4.0: 'Hello'
broadcast message received from 1.0: 'Hello'
broadcast message received from 10.0: 'Hello'
broadcast message received from 9.0: 'Hello'
broadcast message received from 5.0: 'Hello'
broadcast message received from 7.0: 'Hello'
broadcast message received from 2.0: 'Hello'
broadcast message sent
broadcast message received from 1.0: 'Hello'
broadcast message received from 4.0: 'Hello'
broadcast message received from 5.0: 'Hello'
broadcast message received from 8.0: 'Hello'
broadcast message received from 6.0: 'Hello'
broadcast message received from 10.0: 'Hello'
broadcast message received from 1.0: 'Hello'
broadcast message received from 4.0: 'Hello'
broadcast message received from 9.0: 'Hello'
broadcast message received from 2.0: 'Hello'
broadcast message received from 7.0: 'Hello'
broadcast message sent
broadcast message received from 5.0: 'Hello'
broadcast message received from 8.0: 'Hello'
broadcast message received from 2.0: 'Hello'
broadcast message received from 6.0: 'Hello'
broadcast message received from 1.0: 'Hello'
broadcast message received from 10.0: 'Hello'
broadcast message received from 7.0: 'Hello'
```

Enter notes here

```
eceived from 10.0: 'Hello'
eceived from 10.0: 'Hello'
eceived from 10.0: 'Hello'
eceived from 10.0: 'Hello'
eceived from 10.0: 'Hello'
eceived from 10.0: 'Hello'
ent
eceived from 7.0: 'Hello'
```

Control Panel

The control panel controls the simulation.

Start starts the simulation.

Pause stops the simulation.

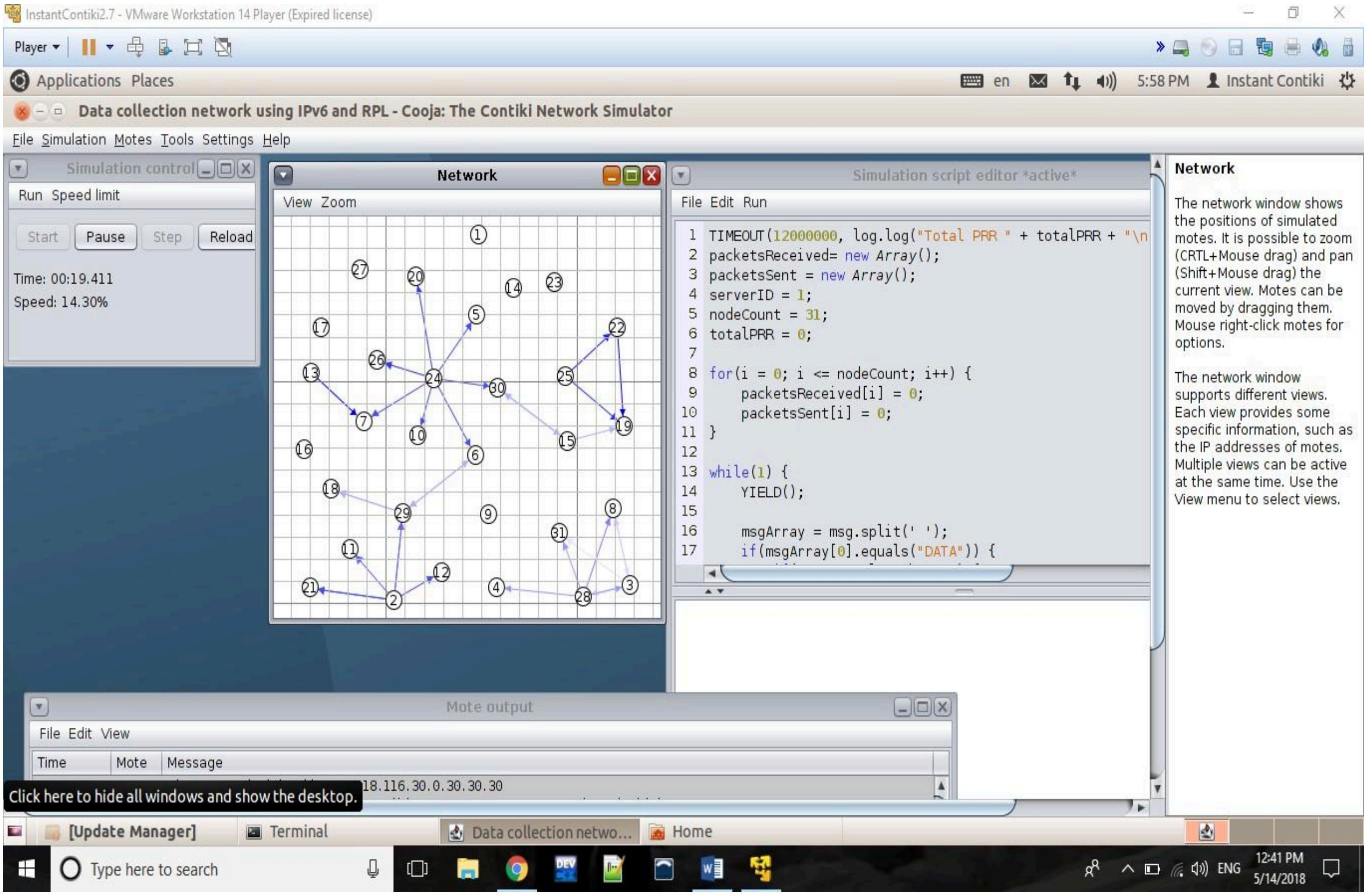
The keyboard shortcut for starting and pausing the simulation is *Ctrl+S*.

Step runs the simulation for one millisecond.

Reload reloads and restarts the simulation.

Simulation speed is controlled via the Speed limit menu.

UDP-RPL Conversion Screenshots:



Time: 00:19.411
Speed: 14.30%

```
TIMEOUT(12000000, log.log("Total PRR " + totalPRR + "\n
packetsReceived= new Array();
packetsSent = new Array();
serverID = 1;
nodeCount = 31;
totalPRR = 0;

for(i = 0; i <= nodeCount; i++) {
    packetsReceived[i] = 0;
    packetsSent[i] = 0;
}

while(1) {
    YIELD();

    msgArray = msg.split(' ');
    if(msgArray[0].equals("DATA")) {
```

```
18.116.30.0.30.30.30
```

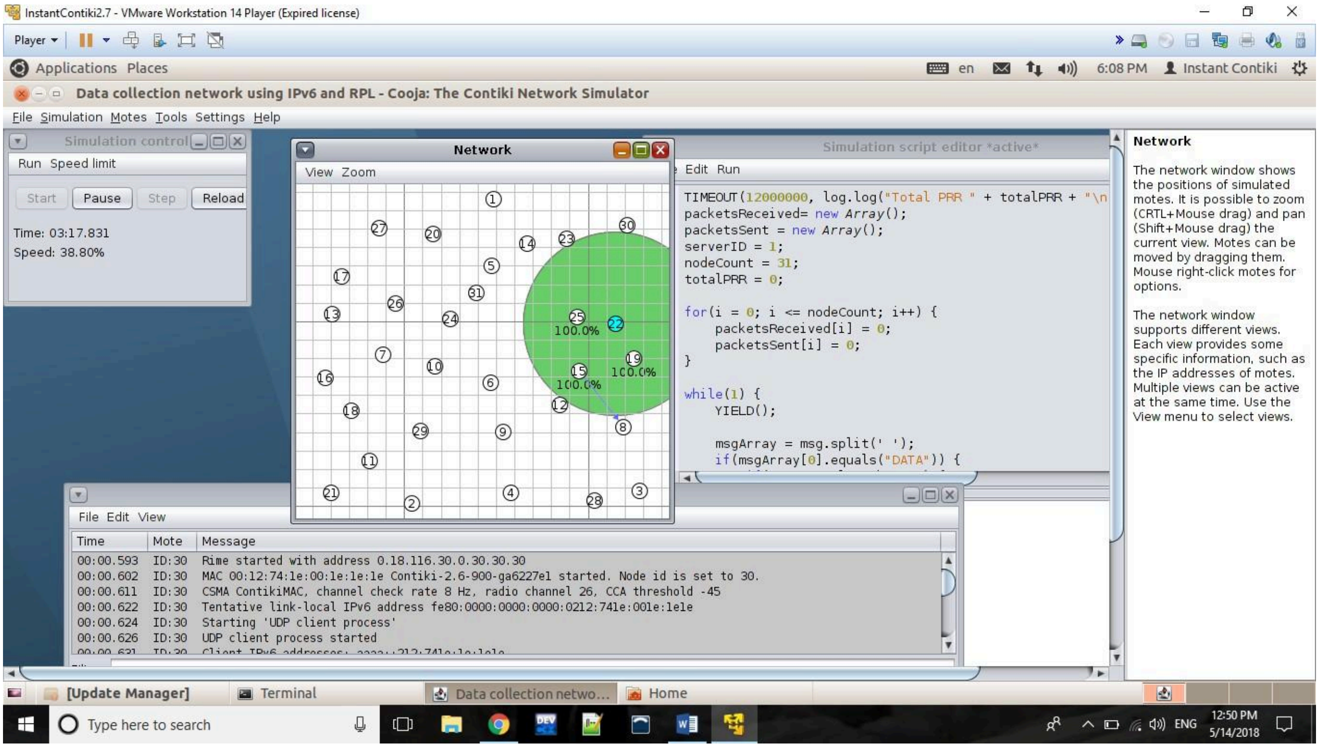
Network

The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

fig 4.11 UDP-RPL Conversion Screenshots

This displays the change made to the objective function of the RPL-UDP sink/source implementation depicts how connections are made between different motes for communication and how the sink node behaves while a packet is transported between motes.



Time: 03:17.831
Speed: 38.80%

```
TIMEOUT(12000000, log.log("Total PRR " + totalPRR + "\n
packetsReceived= new Array();
packetsSent = new Array();
serverID = 1;
nodeCount = 31;
totalPRR = 0;

for(i = 0; i <= nodeCount; i++) {
    packetsReceived[i] = 0;
    packetsSent[i] = 0;
}

while(1) {
    YIELD();

    msgArray = msg.split(' ');
    if(msgArray[0].equals("DATA")) {
```

Network

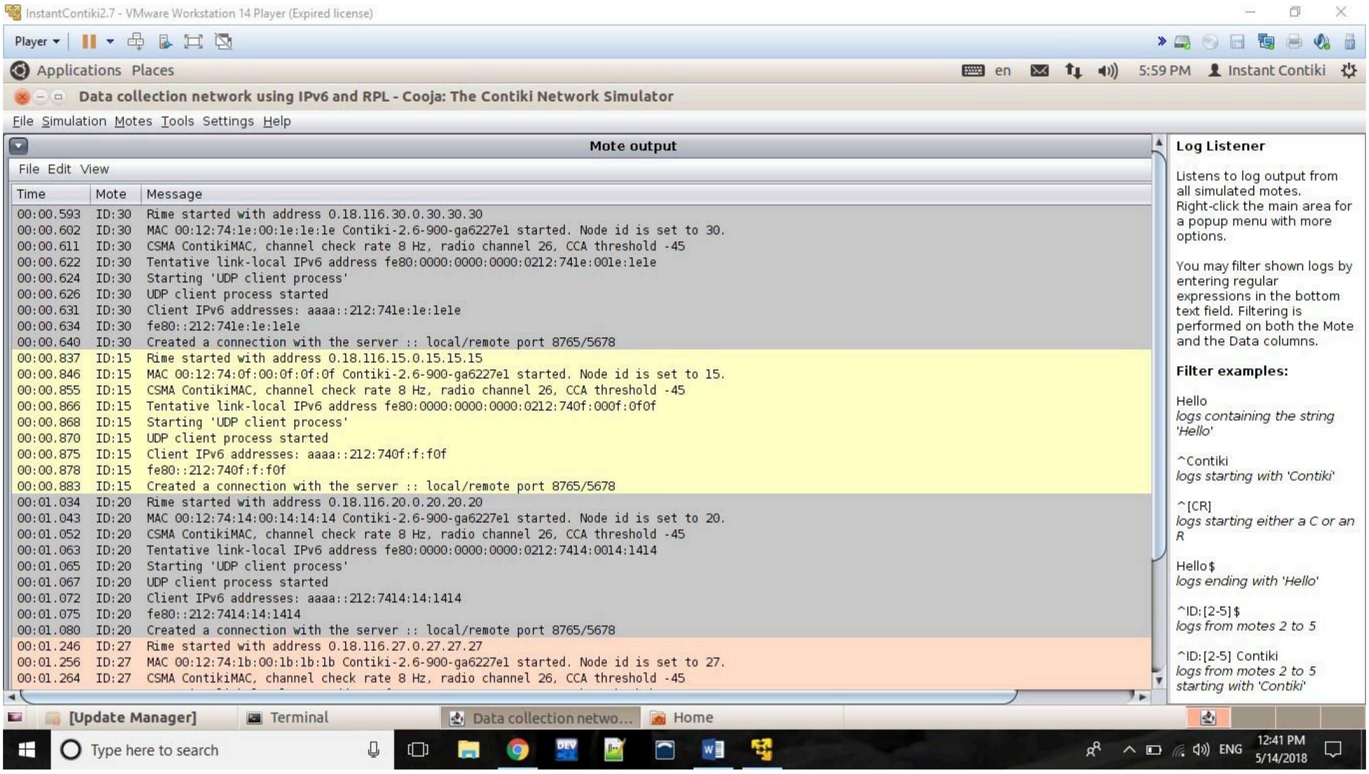
The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

```
Time       Mote   Message
00:00.593  ID:30  Rime started with address 0.18.116.30.0.30.30.30
00:00.602  ID:30  MAC 00:12:74:1e:00:1e:1e:1e Contiki-2.6-900-ga6227e1 started. Node id is set to 30.
00:00.611  ID:30  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
00:00.622  ID:30  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:741e:001e:1e1e
00:00.624  ID:30  Starting 'UDP client process'
00:00.626  ID:30  UDP client process started
00:00.631  ID:30  Client IPv6 addresses: aaaa::212:741e:1e:1e1e
```

fig 4.12 change made to the objective function

The maximum output of the built topology is presented.


InstantContiki2.7 - VMware Workstation 14 Player (Expired license)
Player
Applications Places
en
5:59 PM
Instant Contiki
Data collection network using IPv6 and RPL - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Mote output
File Edit View
[Update Manager]
Terminal
Data collection netwo...
Home
Type here to search
ENG
12:41 PM
5/14/2018

```
Time       Mote   Message
00:00.593  ID:30  Rime started with address 0.18.116.30.0.30.30.30
00:00.602  ID:30  MAC 00:12:74:1e:00:1e:1e:1e Contiki-2.6-900-ga6227e1 started. Node id is set to 30.
00:00.611  ID:30  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
00:00.622  ID:30  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:741e:001e:1e1e
00:00.624  ID:30  Starting 'UDP client process'
00:00.626  ID:30  UDP client process started
00:00.631  ID:30  Client IPv6 addresses: aaaa::212:741e:1e:1e1e
00:00.634  ID:30  fe80::212:741e:1e:1e1e
00:00.640  ID:30  Created a connection with the server :: local/remote port 8765/5678
00:00.837  ID:15  Rime started with address 0.18.116.15.0.15.15.15
00:00.846  ID:15  MAC 00:12:74:0f:00:0f:0f:0f Contiki-2.6-900-ga6227e1 started. Node id is set to 15.
00:00.855  ID:15  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
00:00.866  ID:15  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:740f:000f:0f0f
00:00.868  ID:15  Starting 'UDP client process'
00:00.870  ID:15  UDP client process started
00:00.875  ID:15  Client IPv6 addresses: aaaa::212:740f:f:f0f
00:00.878  ID:15  fe80::212:740f:f:f0f
00:00.883  ID:15  Created a connection with the server :: local/remote port 8765/5678
00:01.034  ID:20  Rime started with address 0.18.116.20.0.20.20.20
00:01.043  ID:20  MAC 00:12:74:14:00:14:14:14 Contiki-2.6-900-ga6227e1 started. Node id is set to 20.
00:01.052  ID:20  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
00:01.063  ID:20  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7414:0014:1414
00:01.065  ID:20  Starting 'UDP client process'
00:01.067  ID:20  UDP client process started
00:01.072  ID:20  Client IPv6 addresses: aaaa::212:7414:14:1414
00:01.075  ID:20  fe80::212:7414:14:1414
00:01.080  ID:20  Created a connection with the server :: local/remote port 8765/5678
00:01.246  ID:27  Rime started with address 0.18.116.27.0.27.27.27
00:01.256  ID:27  MAC 00:12:74:1b:00:1b:1b:1b Contiki-2.6-900-ga6227e1 started. Node id is set to 27.
00:01.264  ID:27  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
```

Log Listener

Listens to log output from all simulated motes. Right-click the main area for a popup menu with more options.

You may filter shown logs by entering regular expressions in the bottom text field. Filtering is performed on both the Mote and the Data columns.

Filter examples:

Hello
logs containing the string 'Hello'

^Contiki
logs starting with 'Contiki'

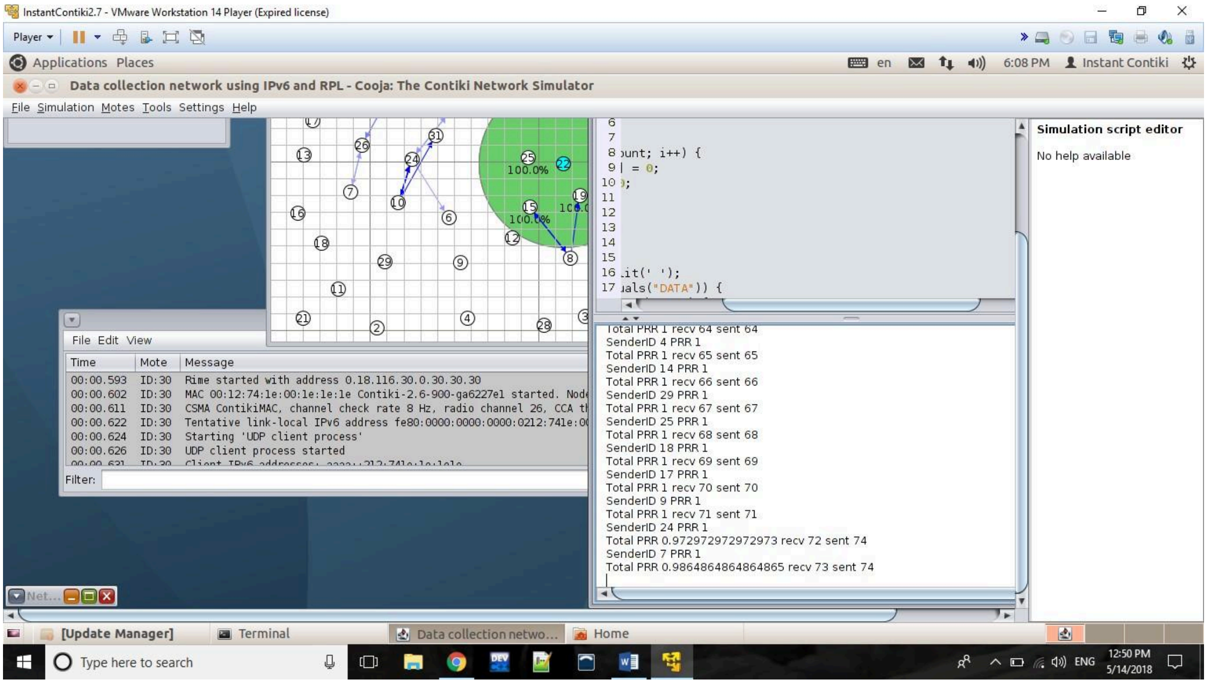
^[CR]
logs starting either a C or an R

Hello$
logs ending with 'Hello'

^ID:[2-5]$
logs from motes 2 to 5

^ID:[2-5] Contiki
logs from motes 2 to 5 starting with 'Contiki'

fig 4.13 max output of build topology



```
Time        Mote    Message
00:00.593   ID:30   Rime started with address 0.18.116.30.0.30.30.30
00:00.602   ID:30   MAC 00:12:74:1e:00:1e:1e:1e Contiki-2.6-900-ga6227e1 started. Node
00:00.611   ID:30   CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA th
00:00.622   ID:30   Tentative link-local IPv6 address fe80:0000:0000:0000:0212:741e:00
00:00.624   ID:30   Starting 'UDP client process'
00:00.626   ID:30   UDP client process started
```

```
ount; i++) {
| = 0;
);



it(' ');
ials("DATA")) {
```

```
Total PRR 1 recv 64 sent 64
SenderID 4 PRR 1
Total PRR 1 recv 65 sent 65
SenderID 14 PRR 1
Total PRR 1 recv 66 sent 66
SenderID 29 PRR 1
Total PRR 1 recv 67 sent 67
SenderID 25 PRR 1
Total PRR 1 recv 68 sent 68
SenderID 18 PRR 1
Total PRR 1 recv 69 sent 69
SenderID 17 PRR 1
Total PRR 1 recv 70 sent 70
SenderID 9 PRR 1
Total PRR 1 recv 71 sent 71
SenderID 24 PRR 1
Total PRR 0.972972972972973 recv 72 sent 74
SenderID 7 PRR 1
Total PRR 0.9864864864864865 recv 73 sent 74
```

fig 4.14 packets transmitted and receive

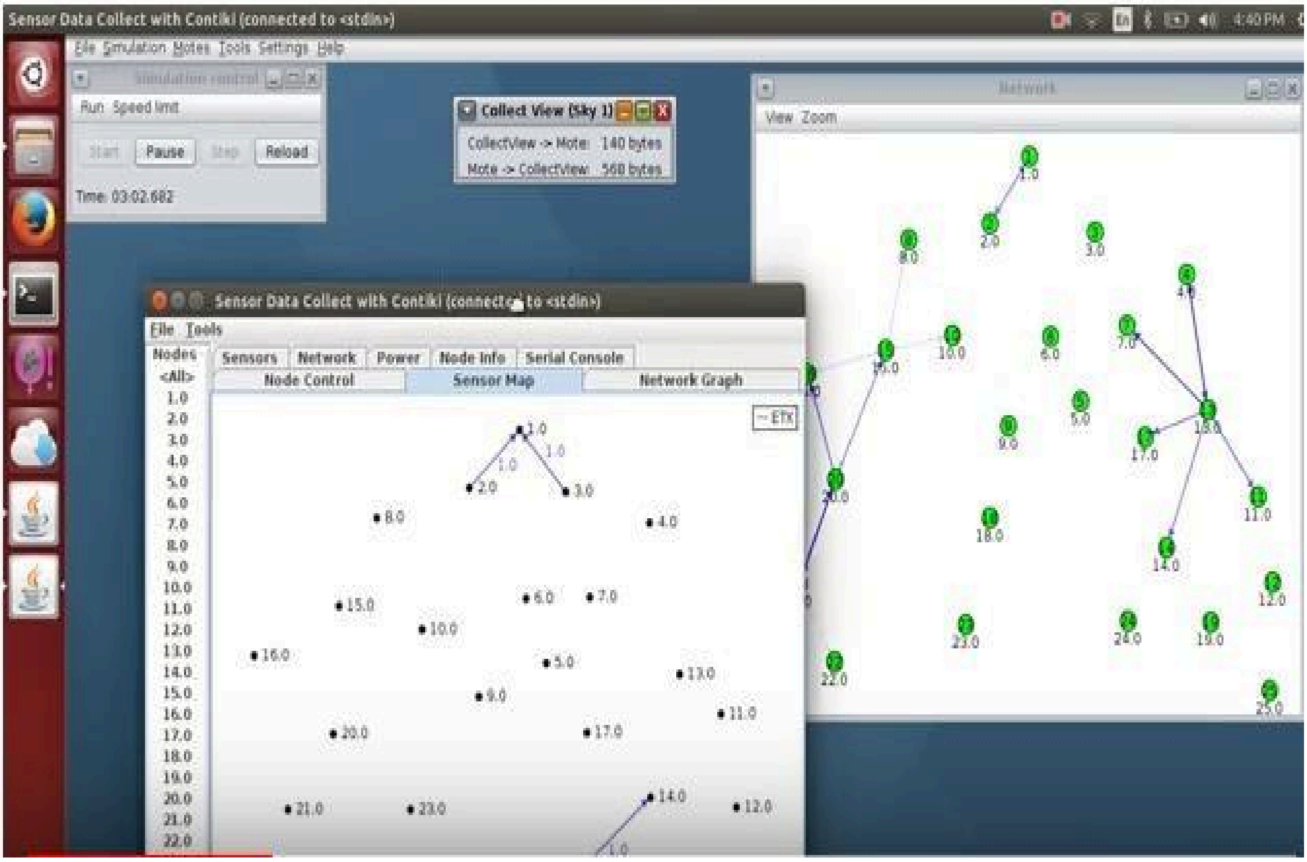
This describes the packets transmitted and received following function change.

Chapter 5 - Conclusions

> Collect View (Sky1) is used to collect sensor data (at beginning).

The Sensor Map is made up of all nodes with values ranging from 1.0 to 25.0.

The arrows from 2.0 and 3.0 are obviously heading in the direction of 1.0, and nodes are beginning to link in this fashion, signifying the map's initial step.



CollectView -> Mote: 140 bytes
Mote -> CollectView: 560 bytes

fig 5.1 collect view

> Typical Radio Duties Cycle

The Y axis is made up of the duty cycle, and the X axis is made up of the nodes.

The duty cycle measurements are continually changing. It first increases to 1.6, then drops to 1.5, then rises to 1.6 again, then continues to rise to 2.4 before dropping back to 1.5.

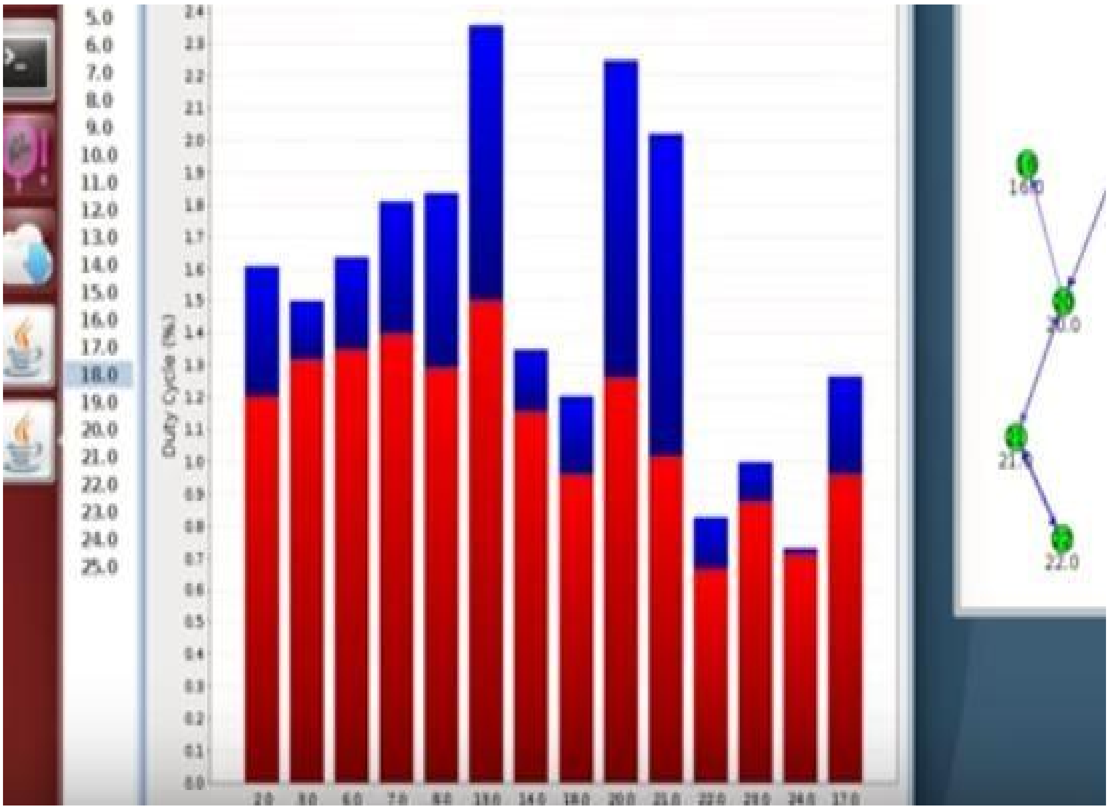


fig 5.2 Typical Radio Duties Cycle

> Instantaneous power consumption

The Y axis represents power, and the X axis represents nodes.

This graph once again exhibits fluctuations. Power usage is roughly 1.5, significantly higher up to 6.0, and then increases to 1.6. It is evident that there is no discernible pattern and that there is increasing unpredictability.

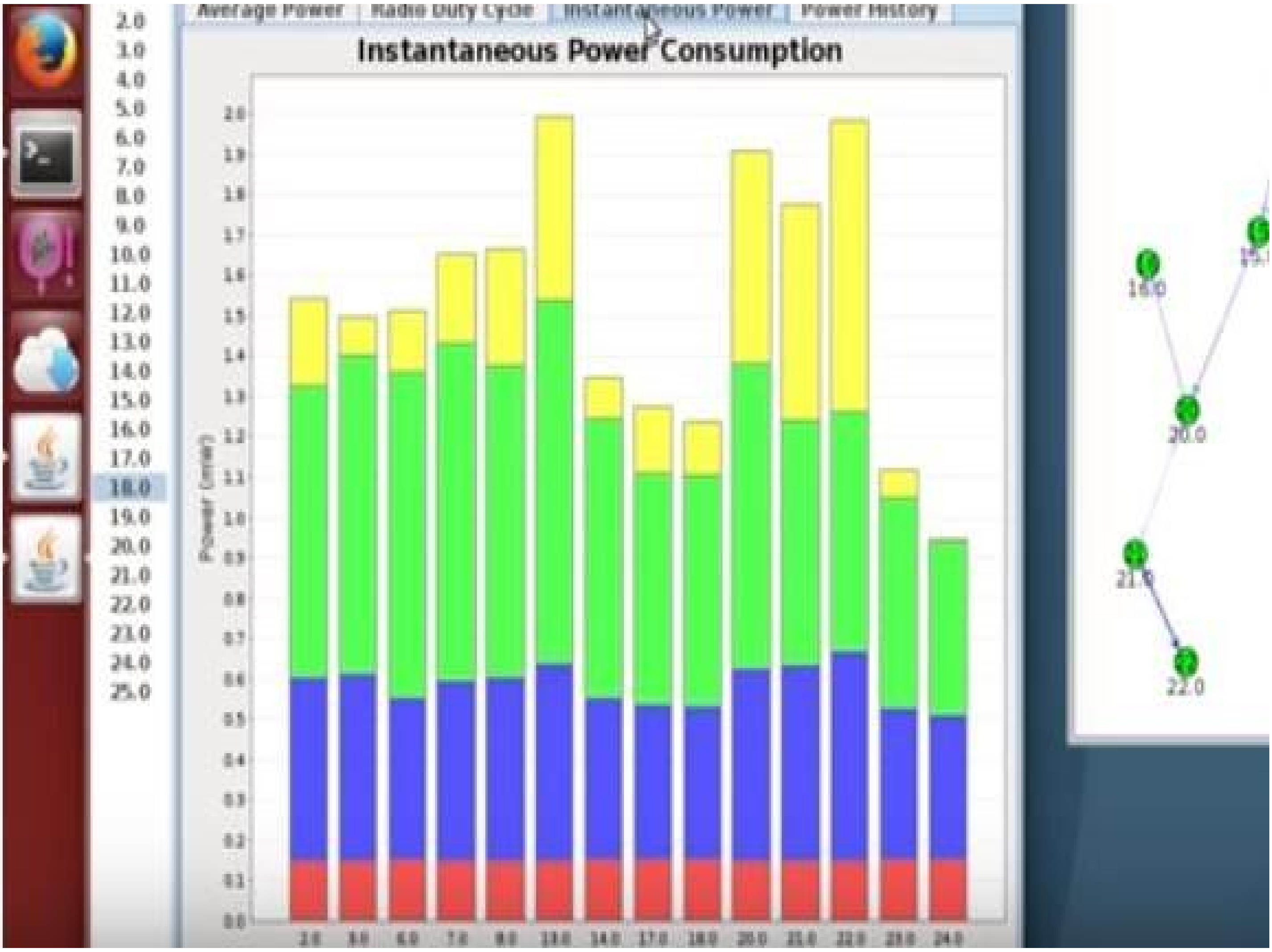


fig 5.3 Instantaneous power consumption

> Packets received by each node

The Y axis represents power, and the X axis represents nodes.

With the exception of 18.0 and 22.0, where the packet value is switched to 2, the bulk of the graph appears stable, with very slight alterations.

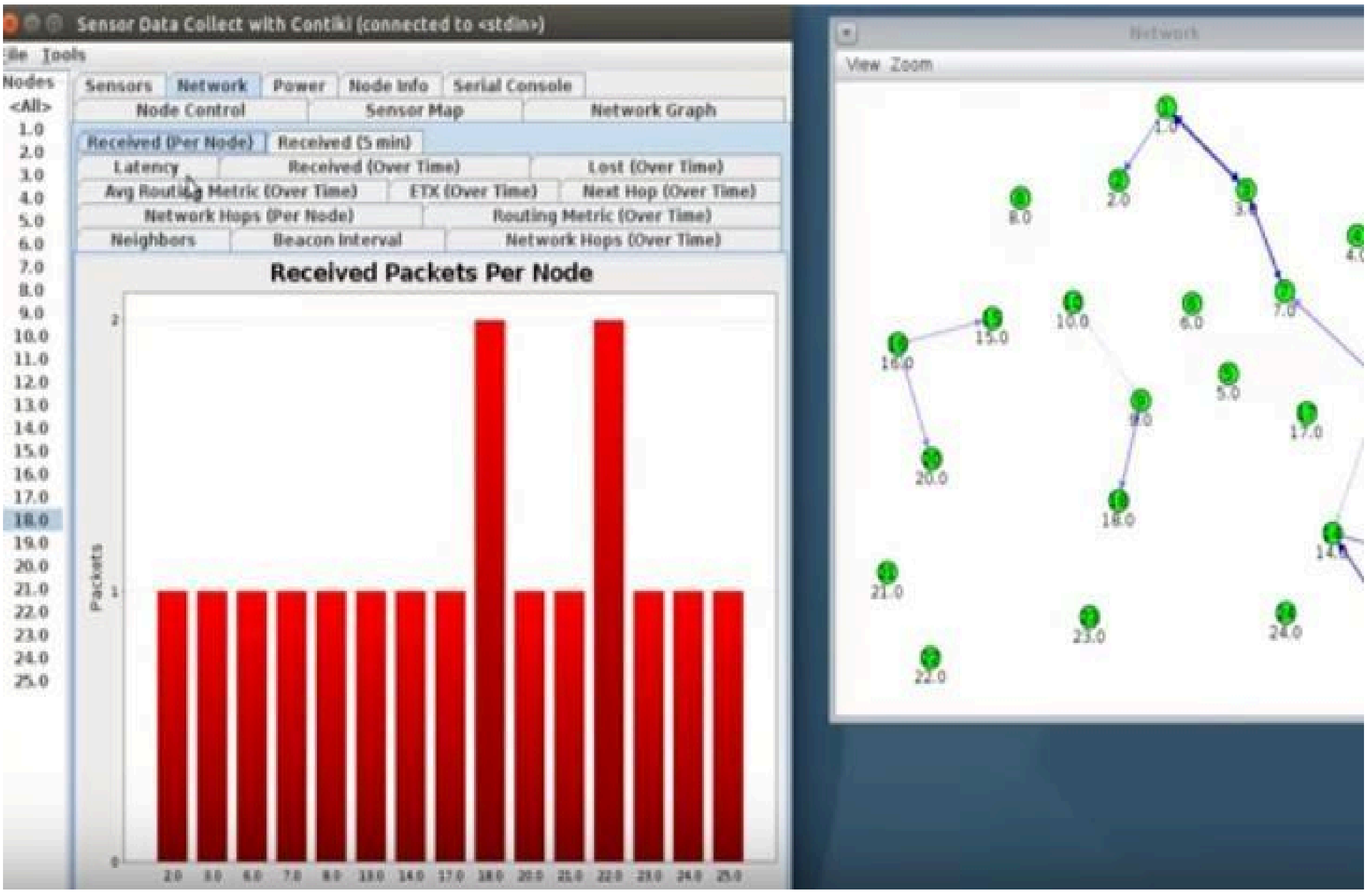


fig 5.4 Packets received by each node

> History is used by past power.

The X axis represents time, and the Y axis represents mW.

The graph generated by the X and Y axes is clearly a linear graph.

In the past, electricity usage fluctuated linearly over time.

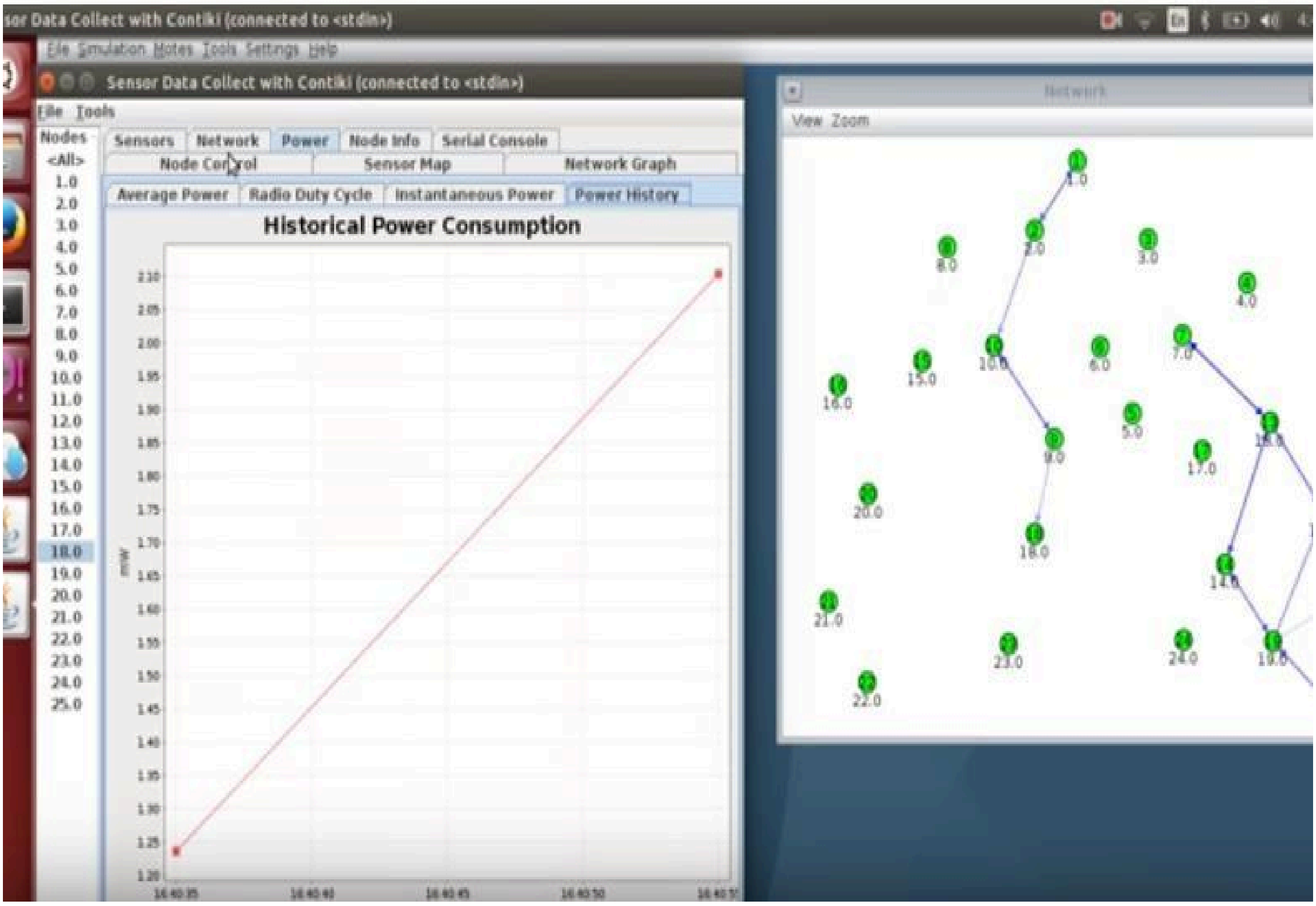


fig 5.5 History is used by past power

> Temperature

The Y axis shows temperature in degrees Celsius, and the X axis represents nodes. The graph is highly uniform, with temperatures ranging from 0 to 600 degrees Celsius for all nodes between 3.0 and 25.0.

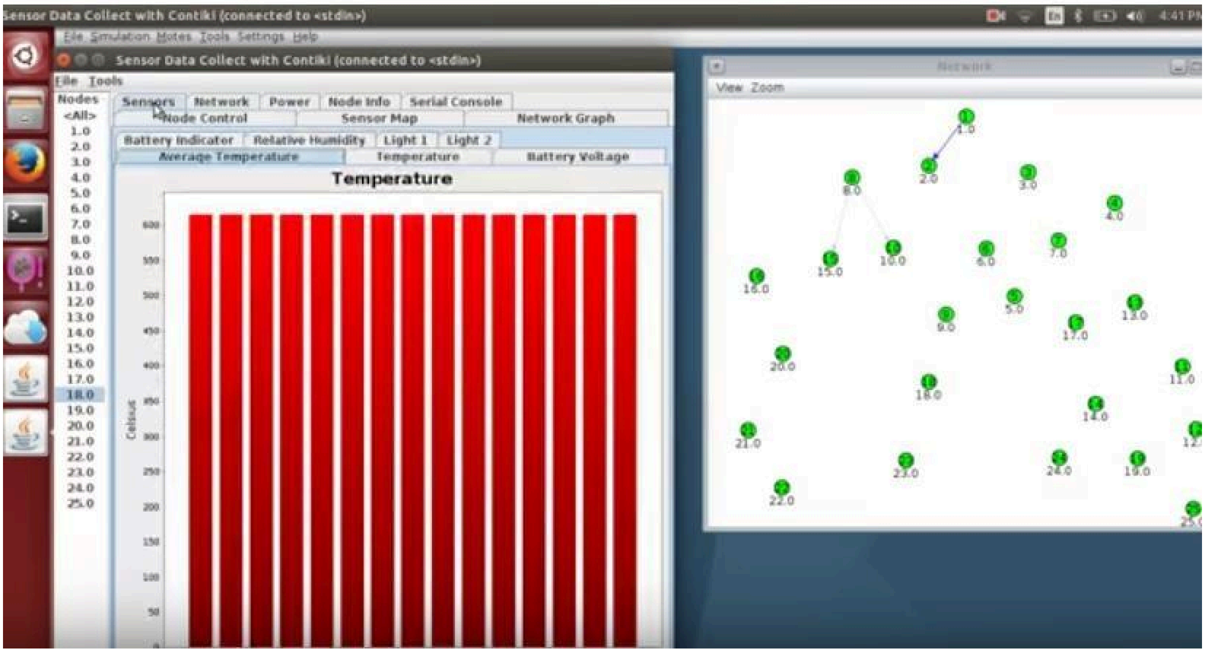


fig 5.6 Temperature

>Hops on the Network

Hops are on the Y axis, and Nodes are on the X axis.

This graph fluctuates once more, with the hop numbers for each node constantly altering. The hop value is one for nodes 2.0 and 3.0 and three for node 6.0; the graph reflects this variation.

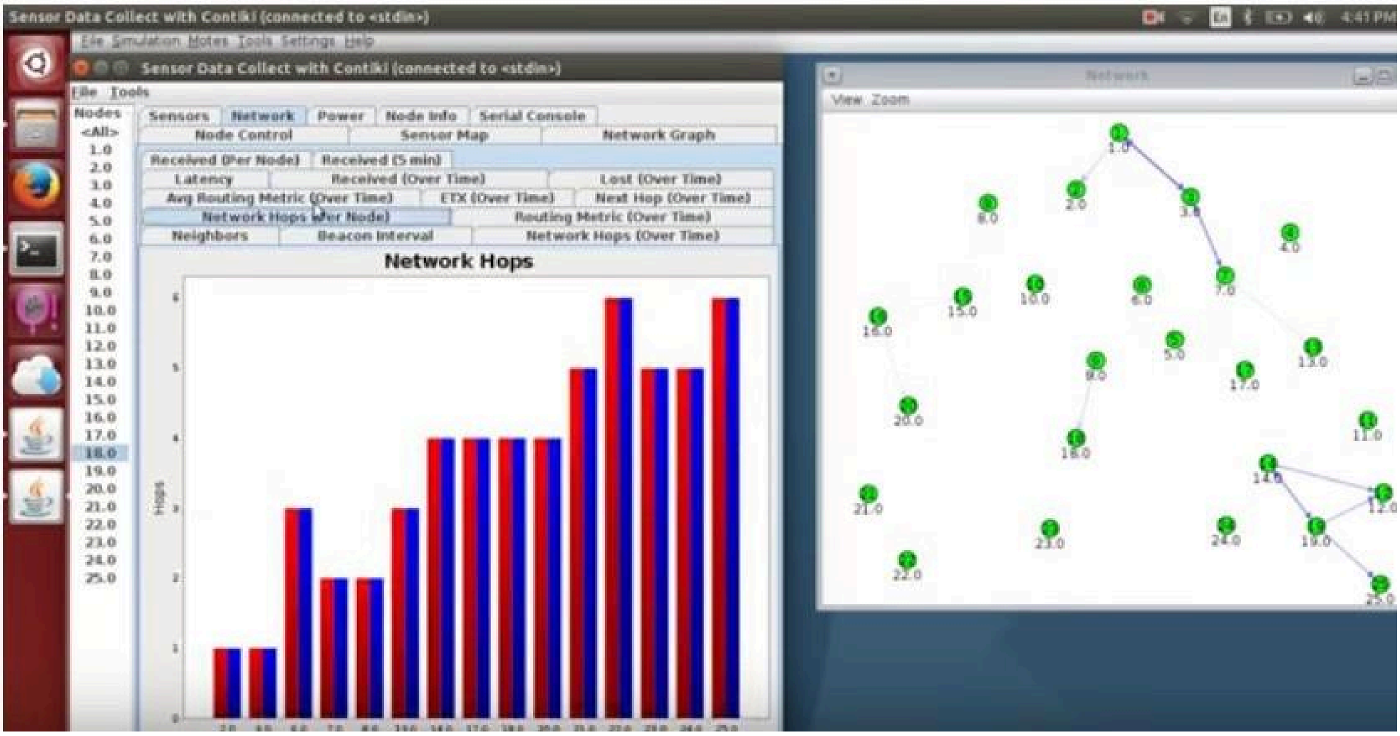


fig 5.7 Hops on the Network

> Collect View (Sky1) is used to collect final sensor data.

The Sensor Map is made up of all nodes with values ranging from 1.0 to 25.0. Every node is plainly connected to one another in some way.

Incoming arrows are clearly linked or directed at the nodes.

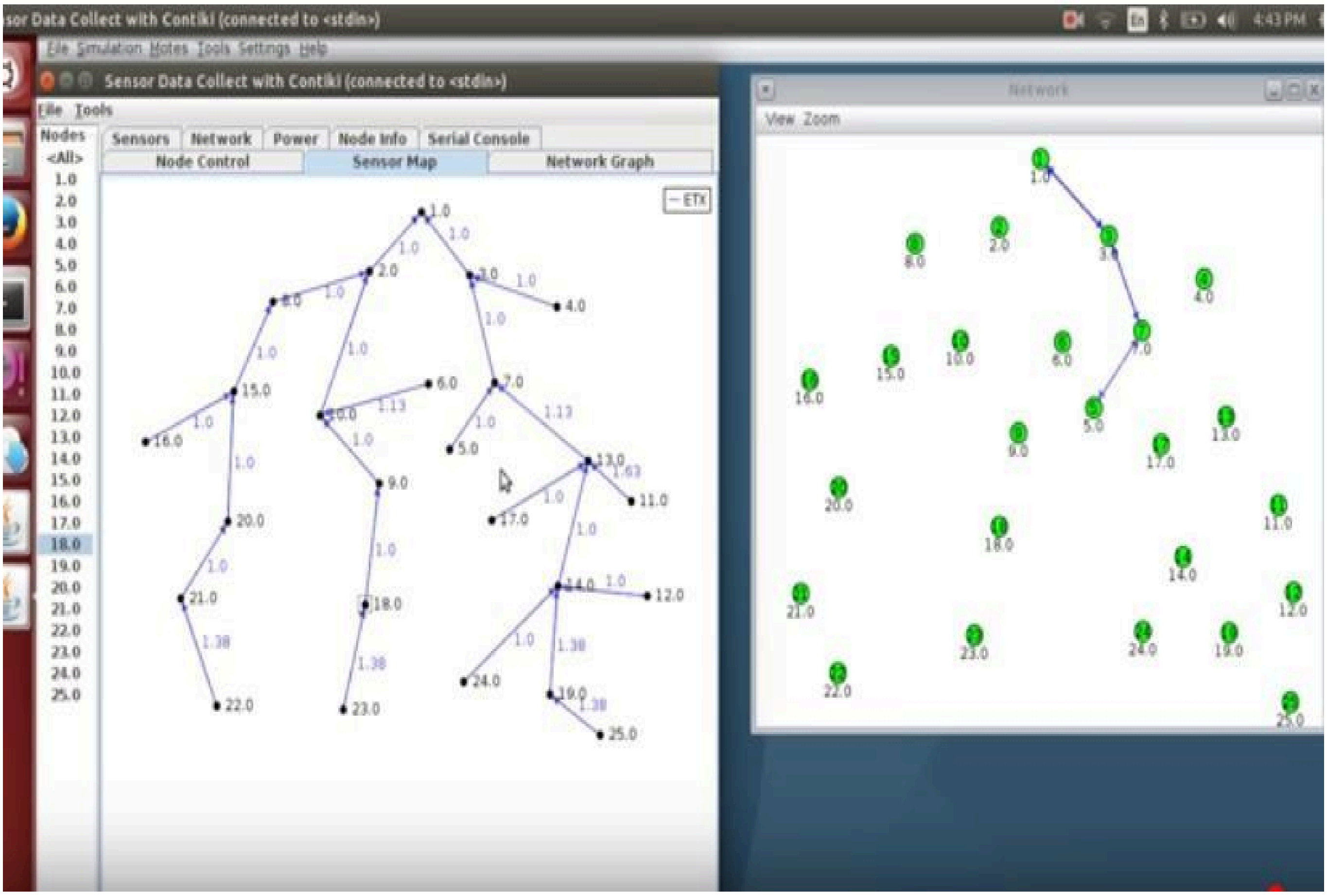


fig 5.8 Collect View

Applications:

Mobility is one of the most crucial features in RPL to allow because it broadens concepts and application fields. There are various real-world uses.

- Ocean sensors to measure depth, identify different mineral types, and do habitat studies, among other things.

Sensors in grain fields, for example, can be used to monitor soil quality and detect the level of dampness.

- In Cloud Computing and Big Data Techniques

The advent of mobility has substantially increased the Internet of Things' reach, expanding its perspective and concepts. Many applications are necessary nowadays, and these can be met if RPL mobility is accomplished.

References

1. [1] Kocakulak M, Butun I (2017) An overview of wireless sensor networks towards internet of things. In: Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual, pp 1–6
2. [2] Botta A, Donato WD, Persico V, Pescape A (2016) Integration of cloud computing and internet of things: a survey. Future Gener Comput Syst 56:684–700
3. Khalil N, Abid MR, Benhaddou D, Gerndt M (2014) Wireless sensors networks for internet of things. In: 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp 1–6
4. Kawamoto D (2017) IoT security incidents rampant and costly”, https://www.darkreading.com/vulnerabilities---threats/iot-security-incidents-rampant-and-costly/d/d-id/1329367, pp 1–9
5. Abualigah L, Diabat A (2020) A comprehensive survey of the grasshopper optimization
6. Tsamardinos I, Brown LE, Aliferis CF (2006) The max-min hill-climbing Bayesian network structure learning algorithm. Mach Learn 65:31–78
7. Gerkey BP, Thrun S, Gordon G (2005) Parallel stochastic hill climbing with small teams. In: Parker LE, Schneider FE, Schultz AC (eds) Multi-robot systems: from swarms to intelligent automata. Springer, Berlin, pp 65–77
8. Rizzo G, Fanizzi N, Amato CD (2020) Class expression induction as concept space exploration: from DL-Foilto DL-Focl. Future Gener Comput Syst 108(2020):256–272
9. Bysani LK and Turuk AK (2011) A survey on selective forwarding attack in wireless sensor networks. In: IEEE International Conference on Devices and Communications (ICDeCom). Mesra, pp 1–5
10. Gaddour O, Koubaa A (2012) RPL in a nutshell: a survey. Comput Netw 56(14):3163–3178
11. Pongle P, Chavan G (2015) A survey: attacks on RPL and 6LoWPAN in IoT. In: 2015 International Conference on Pervasive Computing (ICPC), pp 1–6
12. Adat V, Gupta BB (2018) Security in internet of things: issues, challenges, taxonomy, and architecture. Telecommun Syst 67(3):423–441
13. Javed F, Afzal MK, Sharif M, Kim BS (2018) Internet of things (IoTs) operating systems support, networking technologies, applications, and challenges: a comparative review. IEEE Commun Surv Tutor 2018:1–39
14. Liu X, Sheng Z, Yin C, Ali F, Roggen D (2017) Performance analysis of routing protocol for low power and lossy networks (RPL) in largescale networks. IEEE Internet Things J 4(6):2172–2185

15. Ghaleb B, Al-Dubai AY, Ekonomou E, Alsarhan A, Nasser Y, Mackenzie L, Boukerche A (2018) A survey of limitations and enhancements of the ipv6 routing protocol for low-power and lossy networks: a focus on core operations. IEEE Commun Surv Tutor 21(2):1607–1635
16. Wallgren L, Raza S, Voigt T (2013) Routing attacks and countermeasures in the RPL-based internet of things. Int J Distrib Sens Netw 9(8):794326
17. Sehgal A, Mayzaud A, Badonnel R, Chrisment I, Schönwälder J (2014) Addressing DODAG inconsistency attacks in RPL networks.In: Proceeding of GIIS Conference, pp 1–8
18. Mayzaud A, Sehgal A, Badonnel R, Chrisment I, Schonwalder J (2014) A study of RPL DODAG version attacks, in AIMS'14. Springer, Berlin, pp 92–104
19. Mayzaud A, Sehgal A, Badonnel R, Chrisment I, Schönwälder J (2015) Mitigation of topological inconsistency attacks in RPL-based low-power lossy networks. Int J Netw Manag 25(5):320–339
20. Mayzaud A, Badonnel R, Chrisment I (2016) A taxonomy of attacks in RPL-based internet of things. Int J Netw Secur 18(3):459–473
21. Ahmed F, Ko YB (2016) Mitigation of black hole attacks in routing protocol for low power and lossy networks. Secur Commun Netw 9(18):5143–5154
22. Aris A, Oktug SF, Yalcin SBO (2016) RPL version number attacks: in-depth study. In: 2016 IEEE/IFIP Network Operations and Management Symposium (NOMS 2016), pp 776–779
23. Johnson MO, Siddiqui A, Karami A (2017) A wormhole attack detection and prevention
24. Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for internet of things. Future Gener Comput Syst 82:761–768
25. Perazzo P, Vallati C, Varano D, Anastasi G, Dinni G (2018) Implementation of a wormhole attack against a RPL network: challenges and effects. In: 2018 14th Annual Conference on Wireless On-Demand Network Systems and Services (WONS), pp 95–102
26. Jyothisree MVR, Sreekanth S (2019) Attacks in RPL and detection technique used for internet of things. Int J Recent Technol Eng (IJRTE) 8(1):1876–1879
27. Raoof A, Matrawy A, Lung CH (2018) Routing attacks and mitigation methods for rpl-based internet of things. IEEE Commun Surv Tutor 21(2):1582–1606
28. Sahay R, Geethakumari G, Mitra B, Sahoo I (2020) Efficient framework for detection of version number attack in internet of things. In: Abraham A, Cherukuri A, Melin P, Gandhi N (eds) Intelligent systems design and applications (ISDA 2018). Advances in intelligent systems and computing (AISC 941). Springer, Berlin, pp 480–492
29. Raza S, Wallgren L, Voigt T (2013) SVELTE: real-time intrusion detection in internet of things. Ad Hoc Netw 11:2661–2674
30. Kasinathan P, Pastrone C, Spirito MA, Vinkovits M (2013) Denial-of-service detection in 6LoWPAN based internet of things, wireless and mobile computing, networking and

communications (WiMob). In: 2013 IEEE 9th International Conference on IEEE, pp 600–607

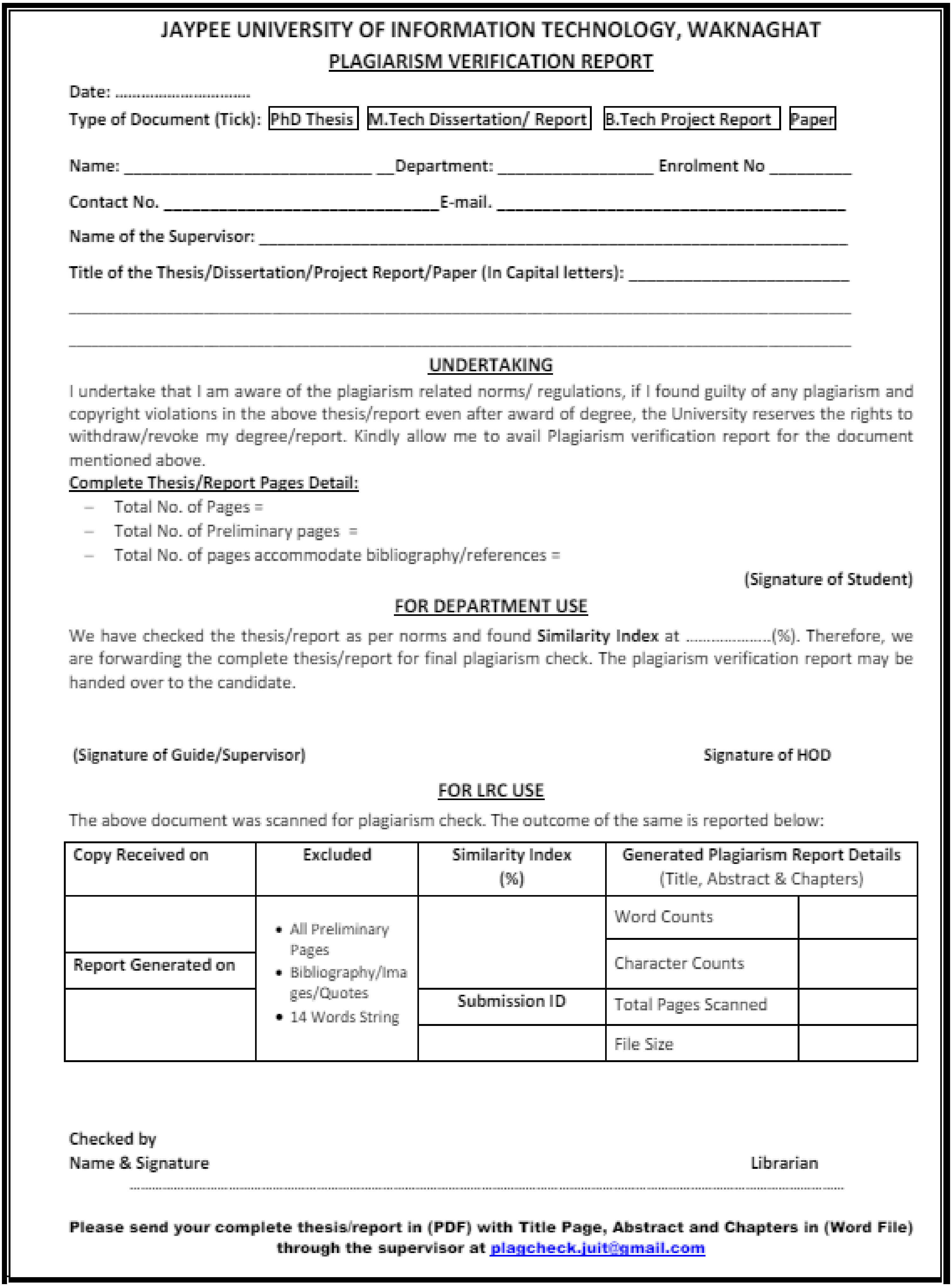
31. Rghioui A, Khannous A, Bouhorma M (2014) Denial-of-service attacks on 6LoWPAN-RPL networks: threats and an intrusion detection system proposition. J Adv Comput Sci Technol 3(2):143–153
32. Sheikhan M, Bostani H (2016) A hybrid intrusion detection architecture for internet of things. In: 8th International Symposium on Telecommunications, IEEE, pp 601–606
33. Bhosale SD, Sonavane SS (2019) A real-time intrusion detection system for wormhole attack in the RPL based internet of things. In: The 12th International Conference Inter Disciplinarity in Engineering, Procedia Manufacturing 32 (2019), pp 840–847
34. Kfoury E, Saab J, Younes P, Achkar R (2019) A self organizing map intrusion detection system for RPL protocol attacks. Int J Interdiscip Telecommun Netw 11(1):30–43
35. Sharma M, Elmiligi H, Gebali F,Verma A (2019) Simulating attacks for RPL and generating multi-class dataset for supervised machine learning. In: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp 20–26
36. Qureshi KN, Rana SS, Ahmed A, Jeon G (2020) A novel and secure attacks detection framework for smart cities industrial internet of things. Sustain Cities Soc 2020:1–33
37. Seeber S, Sehgal A, Stelte B, Rodosek GD, Schönwälder J (2013) Towards a trust computing architecture for RPL in cyber physical systems. In: IFIP/IEEE International Conference on Network and Service Management (CNSM-2013), pp 134–137
38. Fotouhi H, Moreira D, Alves M, Yomsi PM (2017) mRPL+: a mobility management framework in RPL/6LoWPAN. Comput Commun 104(2017):34–54
39. Ferreira HGC, Desousa RT (2017) Security analysis of a proposed internet of things middleware. Clust Comput 20(1):651–660
40. Khoury D, Kfoury E (2017) Generic hybrid methods for secure connections based on the integration of GBA and TLS/CA. Sens Netw Smart Emerg Technol (SENSET). https://doi.org/10.1109/SENSET.2017.8125033
41. Marques BF, Recardo MP (2014) Improving the energy efficiency of WSN by using application layer topologies to constrain RPL-defined routing trees. In: 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), pp 126–133
42. Pavkovic B, Duda A, Hwang WJ, Theoleyre F (2014) Efficient topology construction for RPL over IEEE 802.15.4 in wireless sensor networks. Ad Hoc Netw 15:25–38
43. Glissa G, Rachedi A, Meddeb A (2016) A secure routing protocol based on RPL for internet of things. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp 1–7
44. Kalyani S, Vydeki D (2018) Measurement and analysis of QoS parameters in RPL network. In: 2018 Tenth International Conference on Advanced Computing (ICoAC), pp 1–6
45. Airehrour D, Gutierrez JA, Ray SK (2018) SecTrust-RPL: a securetrust-aware RPL routing protocol for internet of things. Future Gener Comput Syst 2018:1–29

46. Tanganelli G, Virdis A, Mingozzi E (2019) Implementation of software-defined 6LoWPANs in Contiki OS. In: 2019 IEEE 20th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), pp. 1–6
47. Tutunović M, Wuttidittachotti P (2019) Discovery of suitable node number for wireless sensor networks based on energy consumption using Cooja. In: Advanced Communication Technology (ICACT) 2019 21st International Conference on, 2019, pp 168–172
48. Bhandari KS, Ra IH, Cho G (2020) Multi-topology based QoS-differentiation in RPL for internet of things applications. IEEE Access. https://doi.org/10.1109/ACCESS.2020.2995794
49. Sanila A, Mahapatra B, Turuk AK (2020) Performance evaluation of RPL protocol in a 6LoWPAN based Smart Home Environment. In: 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)
50. Roussel K, Song YQ, Zendra O (2016) Using Cooja for WSN Simulations: some new uses and limits, EWSN 2016–Next mote workshop, Austria, Feb 2016. ACM, Junction Publishing, pp 319-324
51. Uwase MP, Long NT, Tiberghien J, Steenhaut K, Dricot JM (2014) Poster abstract: outdoors range measurements with zolertia z1 motes and contiki. In: Langendoen K, Hu W, Ferrari F, Zimmerling M, Mottola L (eds) Real-world wireless sensor network. Lecture notes in electrical engineering. Springer, Berlin, pp 79–83
52. Kharche S, Pawar S (2016) Node level energy consumption analysis in 6lowpan network using real and emulated Zolertia Z1 motes. In: 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp 1–5
53. Kharche S, Pawar S (2017) Effect of radio link and network layer parameters on performance of Zolertia Z1 motes based 6LoWPAN. In: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp 1–7
54. Hendrawan INR, Arsa IGNW (2017) Zolertia Z1 energy usage simulation with Cooja simulator, informatics and computational sciences (ICICoS). In: 2017 1st International Conference on, 2017, pp 147–152
55. Bandekar A, Kotian A, Javaid AY (2017) Comparitive analysis of simulation and real-world energy consumption for battery-life estimation of low-power IoT (Internet of Things) deployment in varying environmental conditions using Zolertia Z1 motes, ICST Institute for Computer Sciences, Social Informatics and Telecommunication Engineering 2017 (LNICST 205), pp 137–148
56. Mian AN, Alvi SA, Khan R, Zulqarnain M, Iqbal W (2016) Experimental study of link quality in IEEE 802.15.4 using Z1 Motes. In: 2016 International Wireless Communications and Mobile Computing Conference (IWCMC) pp 830–835
57. Hondt AD, Bahmad H, Vanhee J (2016) RPL attacks framework, mobile and embedded computing LINGI2146 Report, pp 1–14

58. O¨sterlind F, Dunkels A, Eriksson J, Finne N, Voigt T (2006) Cross-level sensor network simulation with Cooja In: IEEE 31st Conference on Local Computer Networks, LCN ’06, IEEE Computer Society, pp 641–648

59. Dunkels A, Gronvall B, Voigt T (2004) Contiki–a lightweightand flexible operating system for tiny networked sensors. In: 29th Annual IEEE International Conference on Local Computer Networks (LCN’04), IEEE Computer Society, pp 455–462

60. Verma VK, Ntalianis K, Moreno CM, Yang CT (2019) Next-generation Internet of things and cloud security solutions. Int J Distrib Sens Netw. https://doi.org/10.1177/1550147719835098



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ____________________________ __Department: _________________ Enrolment No _________

Contact No. _______________________________E-mail. _______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

– Total No. of Pages =
– Total No. of Preliminary pages =
– Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) Signature of HOD

FOR LRC USE

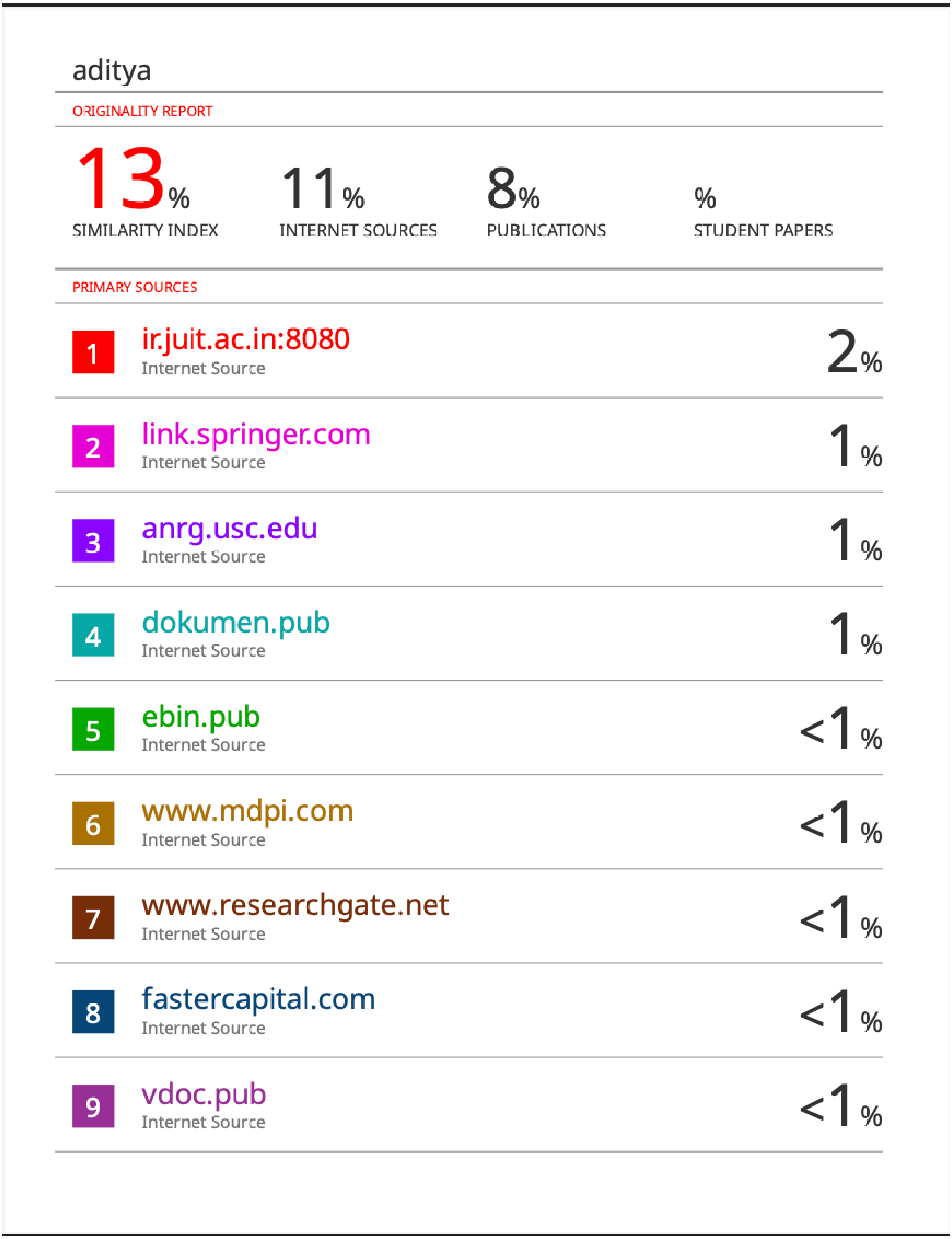
The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature Librarian

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com



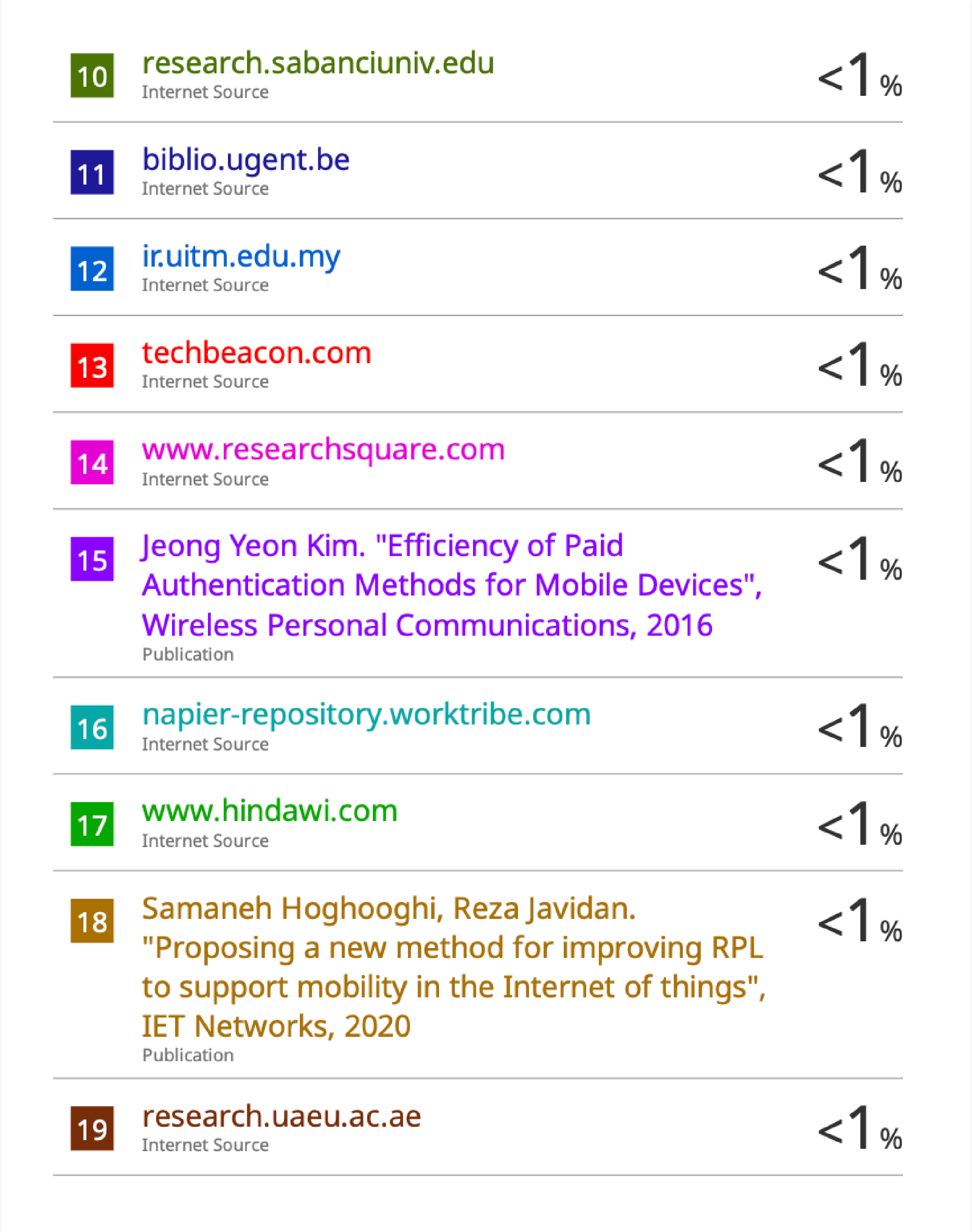
aditya

ORIGINALITY REPORT

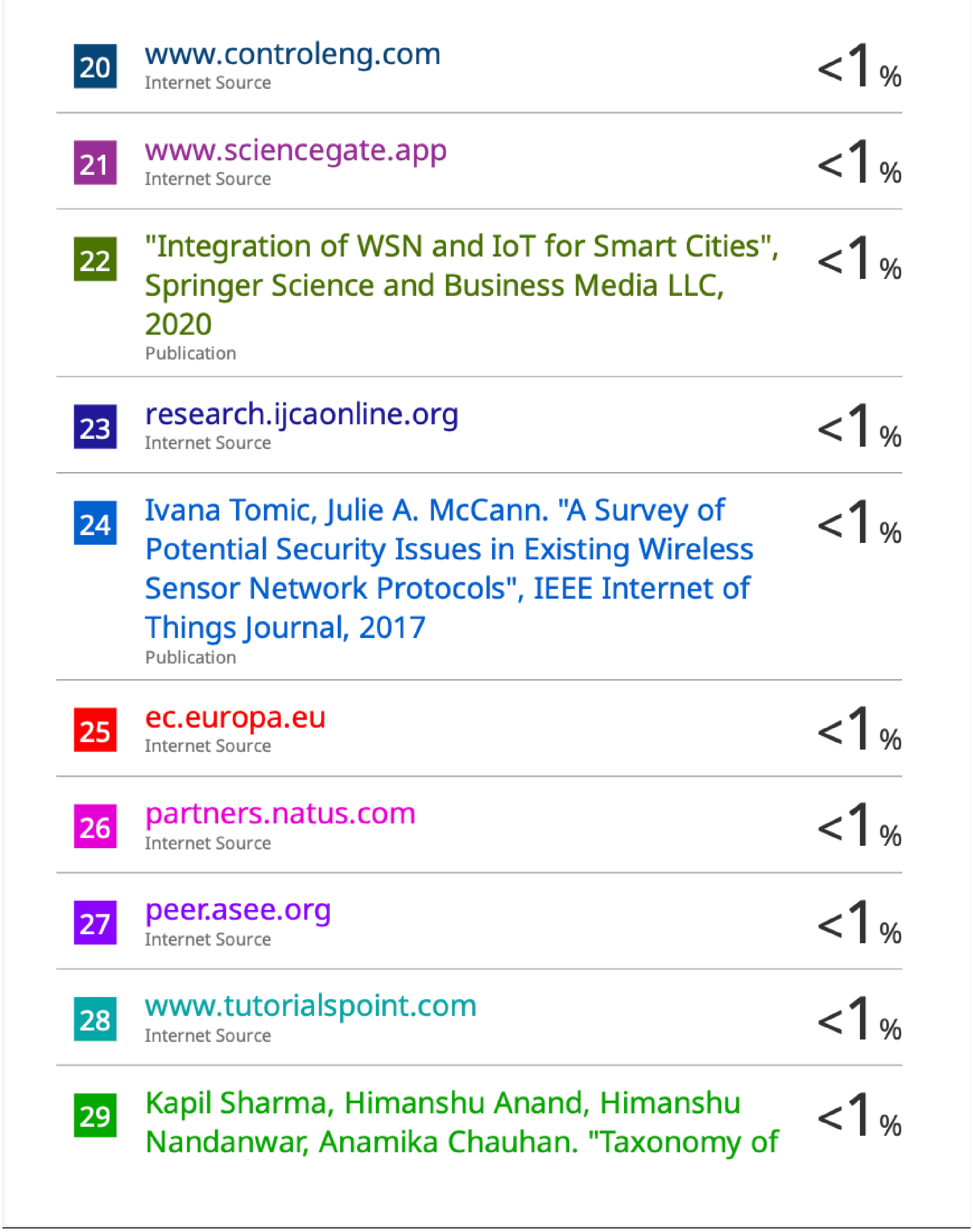
13%	11%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

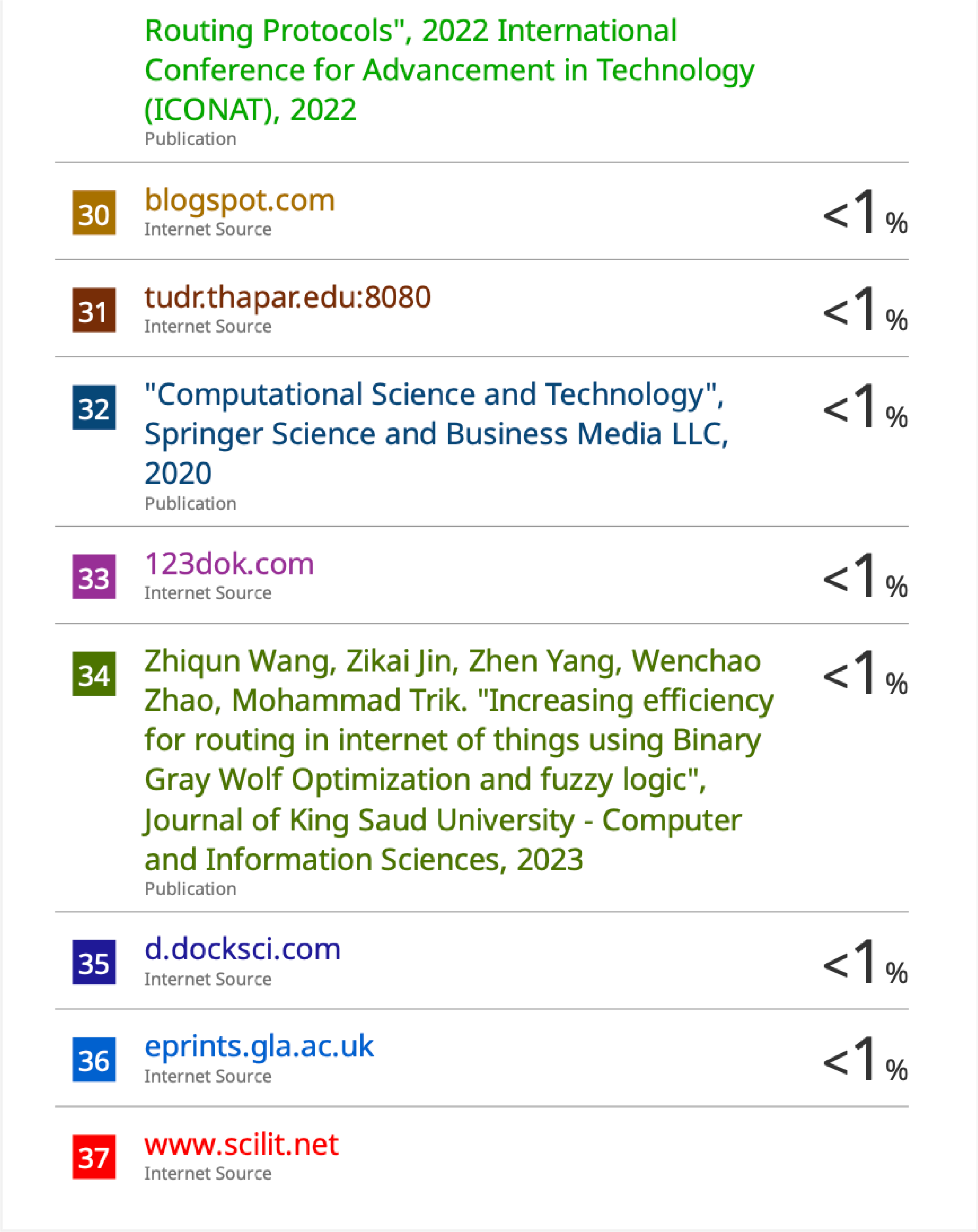
1	ir.juit.ac.in:8080 Internet Source	2%
2	link.springer.com Internet Source	1%
3	anrg.usc.edu Internet Source	1%
4	dokumen.pub Internet Source	1%
5	ebin.pub Internet Source	<1%
6	www.mdpi.com Internet Source	<1%
7	www.researchgate.net Internet Source	<1%
8	fastercapital.com Internet Source	<1%
9	vdoc.pub Internet Source	<1%



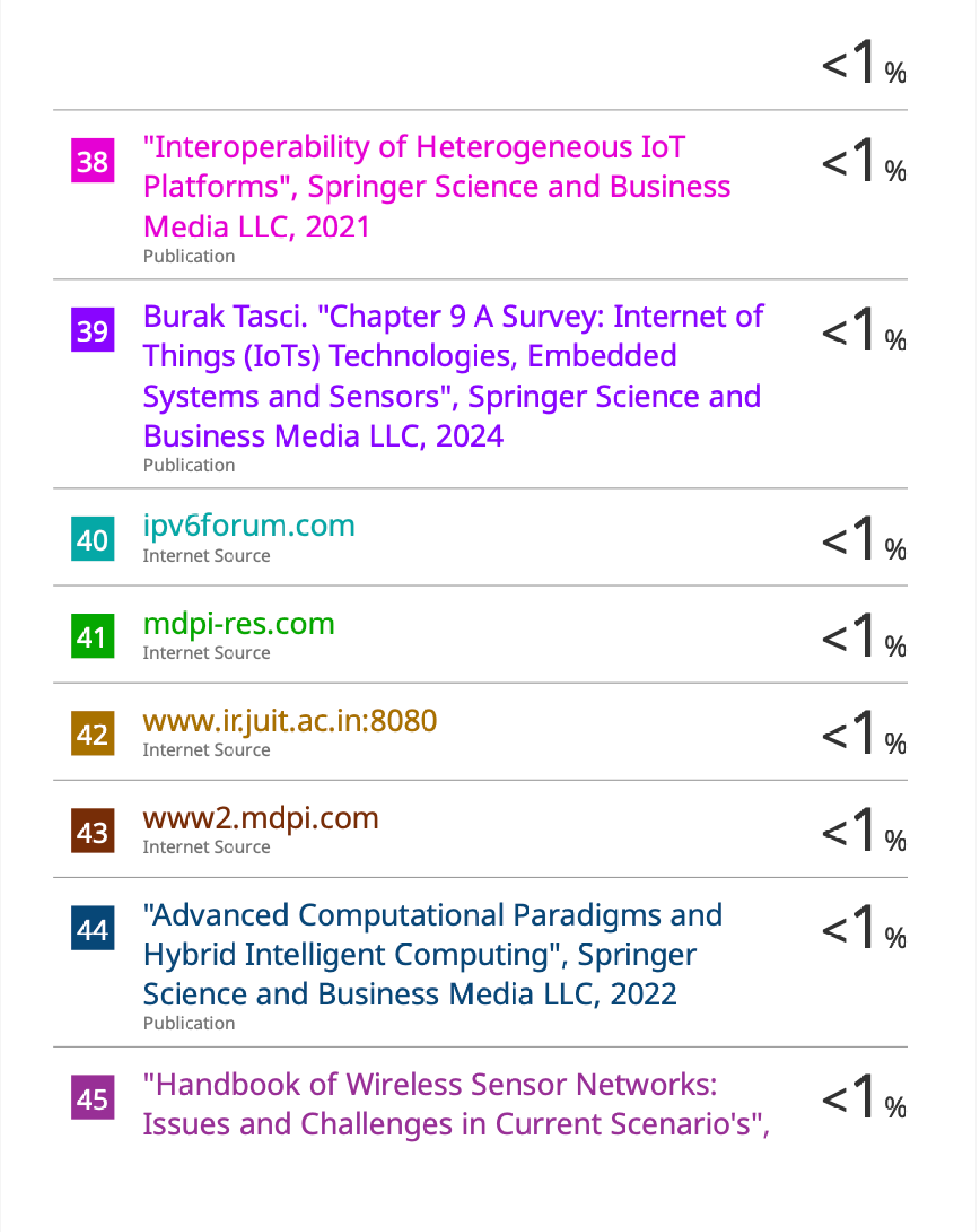
10	research.sabanciuniv.edu Internet Source	<1 %
11	biblio.ugent.be Internet Source	<1 %
12	ir.uitm.edu.my Internet Source	<1 %
13	techbeacon.com Internet Source	<1 %
14	www.researchsquare.com Internet Source	<1 %
15	Jeong Yeon Kim. "Efficiency of Paid Authentication Methods for Mobile Devices", Wireless Personal Communications, 2016 Publication	<1 %
16	napier-repository.worktribe.com Internet Source	<1 %
17	www.hindawi.com Internet Source	<1 %
18	Samaneh Hoghooghi, Reza Javidan. "Proposing a new method for improving RPL to support mobility in the Internet of things", IET Networks, 2020 Publication	<1 %
19	research.uaeu.ac.ae Internet Source	<1 %



20	www.controleng.com Internet Source	<1 %
21	www.sciencegate.app Internet Source	<1 %
22	"Integration of WSN and IoT for Smart Cities", Springer Science and Business Media LLC, 2020 Publication	<1 %
23	research.ijcaonline.org Internet Source	<1 %
24	Ivana Tomic, Julie A. McCann. "A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols", IEEE Internet of Things Journal, 2017 Publication	<1 %
25	ec.europa.eu Internet Source	<1 %
26	partners.natus.com Internet Source	<1 %
27	peer.asee.org Internet Source	<1 %
28	www.tutorialspoint.com Internet Source	<1 %
29	Kapil Sharma, Himanshu Anand, Himanshu Nandanwar, Anamika Chauhan. "Taxonomy of	<1 %



	Routing Protocols", 2022 International Conference for Advancement in Technology (ICONAT), 2022 Publication	
30	blogspot.com Internet Source	<1 %
31	tudr.thapar.edu:8080 Internet Source	<1 %
32	"Computational Science and Technology", Springer Science and Business Media LLC, 2020 Publication	<1 %
33	123dok.com Internet Source	<1 %
34	Zhiqun Wang, Zikai Jin, Zhen Yang, Wenchao Zhao, Mohammad Trik. "Increasing efficiency for routing in internet of things using Binary Gray Wolf Optimization and fuzzy logic", Journal of King Saud University - Computer and Information Sciences, 2023 Publication	<1 %
35	d.docksci.com Internet Source	<1 %
36	eprints.gla.ac.uk Internet Source	<1 %
37	www.scilit.net Internet Source	



		<1 %
38	"Interoperability of Heterogeneous IoT Platforms", Springer Science and Business Media LLC, 2021 Publication	<1 %
39	Burak Tasci. "Chapter 9 A Survey: Internet of Things (IoTs) Technologies, Embedded Systems and Sensors", Springer Science and Business Media LLC, 2024 Publication	<1 %
40	ipv6forum.com Internet Source	<1 %
41	mdpi-res.com Internet Source	<1 %
42	www.ir.juit.ac.in:8080 Internet Source	<1 %
43	www2.mdpi.com Internet Source	<1 %
44	"Advanced Computational Paradigms and Hybrid Intelligent Computing", Springer Science and Business Media LLC, 2022 Publication	<1 %
45	"Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's",	<1 %