Mini Project Report

QuickStart: Simplifying Class Room Setups

Submitted by:

Meghana M	**PES2UG21CS292**	**6E**
Sumukh Suresh	**PES2UG21CS555**	**6I**

6th Semester 'E' and 'I' Section

Prof. Animesh Giri
Associate Professor

January - May 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLEOFCONTENT

1. Introduction

This project leverages the Cooja simulator on Contiki OS to develop an IoT-based automatic teacher helper system. The implementation employs Sky Motes and simulates 7 classrooms with 3 teachers, each assigned to a specific classroom. A custom mobility model is used to direct the motes, representing the teachers' movements. The communication model follows a static sink (classroom server) and mobile sender (teacher's laptop) approach.

The primary objective of this project is to automate the connection process between teachers' laptops and classroom monitors, significantly reducing setup time. By equipping each teacher with a specific sensor and unique ID, the system ensures that when a teacher enters the transmission range of a server, their unique ID is transmitted. This triggers an actuator to initiate the connection process, seamlessly displaying the teacher's laptop and presentation content on the classroom monitor without manual intervention.

Security is a crucial aspect of this system. Robust checks are implemented to ensure that only the teacher whose ID matches the preset requirement can trigger the actuator, guaranteeing the secure and accurate display of content. This automation allows teachers to focus more on teaching and less on technical difficulties, enhancing the overall educational experience.

The significance of this innovative solution lies in its ability to streamline classroom setup processes, thus reducing technical difficulties and increasing teaching time. By ensuring a seamless transition when teachers enter the classroom, the system improves the overall efficiency of classroom activities, ultimately enhancing the educational experience for both teachers and students.

2. Background

- **Contiki OS:**

  Contiki OS is an open-source operating system designed specifically for networked, memory-constrained systems, such as those used in Internet of Things (IoT) applications. It was developed by Adam Dunkels at the Swedish Institute of Computer Science and has become a popular choice for developing IoT solutions due to its flexibility, scalability, and efficient use of resources.

  Features of Contiki OS:

  - Lightweight and Efficient: Contiki is designed to run on tiny hardware platforms with limited memory and processing power. It can operate with as little as 2 kilobytes of RAM and 40 kilobytes of ROM.
  - Networking Support: Contiki includes a full IP networking stack, including IPv6, 6LoWPAN, RPL (Routing Protocol for Low-Power and Lossy Networks), and CoAP (Constrained Application Protocol), making it ideal for IoT applications.
  - Real-Time Multitasking: The OS supports both preemptive and cooperative multitasking, allowing it to handle multiple processes efficiently.
  - Power Efficiency: Contiki includes power management features that help extend the battery life of IoT devices by minimizing power consumption.
  - Modular Design: Its modular architecture allows developers to include only the components they need, keeping the system lightweight and tailored to specific applications.
  - Simulation and Development Tools: Contiki provides tools like the Cooja simulator, which allows developers to simulate entire networks of Contiki nodes for testing and debugging.

- **Cooja Simulator:**

Overview of the Cooja Simulator:

The Cooja simulator is a network simulator specifically designed for developing and testing software for Contiki OS. It enables developers to simulate a wide range of IoT devices and network configurations, making it an invaluable tool for IoT research and development.

Relevance to the Project:

- Network Simulation: Cooja allows for the simulation of complex network topologies and communication protocols used in IoT applications. This is crucial for the project, which involves simulating a network of classrooms and teachers' laptops.
- Testing and Debugging: By simulating the entire network, developers can identify and fix issues in the software before deploying it on actual hardware. This ensures the reliability and stability of the IoT-based automatic teacher helper system.
- Custom Mobility Models: Cooja supports the creation of custom mobility models, which is essential for accurately simulating the movement of teachers and their interaction with classroom servers in the project.
- Resource Constraints: The simulator can emulate the limited resources of Sky Motes, allowing developers to optimize the code for memory and power efficiency, which is critical for real-world IoT deployments.
- Visualization and Analysis: Cooja provides tools for visualizing network traffic, node interactions, and performance metrics. This helps in understanding the behavior of the system and making necessary adjustments to improve performance and efficiency.

By using Contiki OS and the Cooja simulator, the project can effectively develop, test, and optimize an IoT-based automatic teacher helper system, ensuring a smooth and efficient setup process for teachers in a classroom environment.

3. Project Description

- **Problem Statement:**

  Teachers often face challenges when setting up their laptops and presentation slides in classrooms, leading to delays and technical difficulties. This can disrupt the flow of the lesson and reduce the time available for teaching. The manual process of connecting laptops to classroom monitors and ensuring the correct display of content is time-consuming and prone to errors, detracting from the overall educational experience.

- **Objectives:**

  - Automate Classroom Setup: Develop an IoT-based system that automatically connects teachers' laptops to classroom monitors when they enter the room.
  - Reduce Setup Time: Minimize the time teachers spend on technical setup, allowing more time for teaching.
  - Ensure Secure and Accurate Content Display: Implement robust checks to ensure that only authorized teachers can trigger the system, matching their unique ID with the server's preset requirements.
  - Improve Efficiency: Streamline the process of connecting laptops to monitors, reducing the likelihood of technical issues and disruptions.
  - Enhance Educational Experience: By automating the setup process, enable teachers to focus more on teaching and less on dealing with technical problems, thereby improving the overall educational experience for both teachers and students.

4. Implementation

Development Setup:

- Software: Contiki OS and Cooja simulator.
- Sensor Nodes: Sky Motes for simulation.
- Programming Languages: Primarily C for Contiki OS.
- Tools: Cooja for network simulation, Wireshark for network monitoring, Foren6 to visualize the network topology and standard development tools like GCC for compiling.

Mobility in Cooja:

- Custom Mobility Models: Used to represent the movement patterns of teachers within the school environment.
- Implementation: Mobility models define how sender nodes (teachers' laptops) move and interact with static sink nodes (classroom servers). These models are coded to reflect realistic movement within the classroom setup.

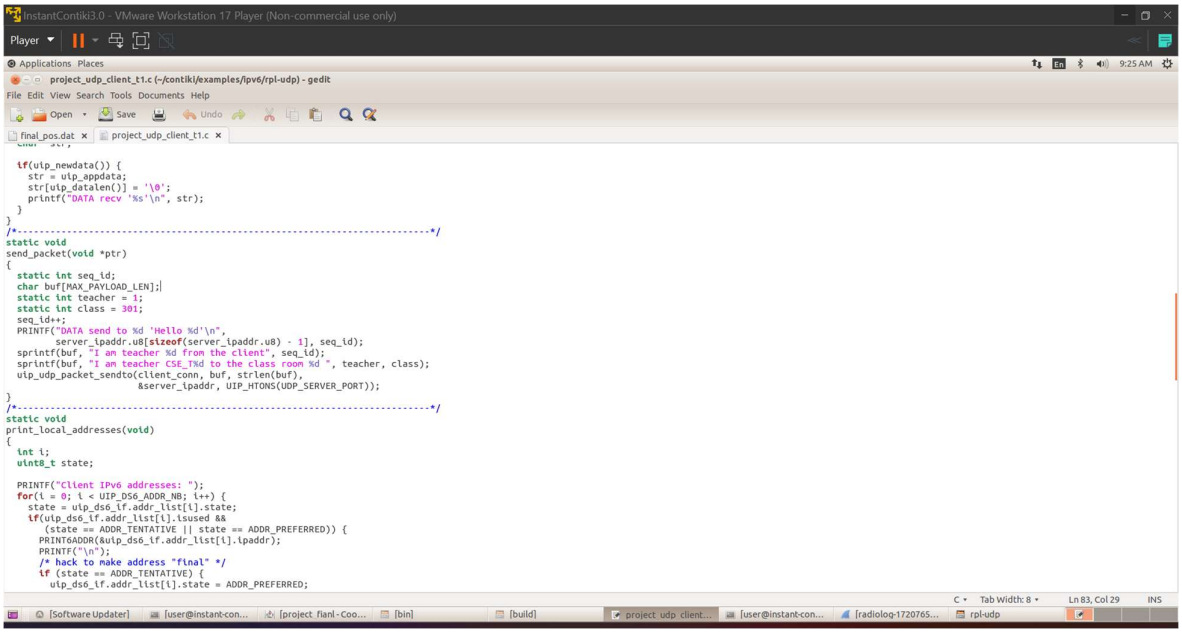
Wireshark Integration:

- Integration: Wireshark is integrated with the Cooja simulator to capture and analyze network traffic.
- Usage: Helps in monitoring packet exchanges, diagnosing network issues, and ensuring that data transmissions are secure and efficient.
- Analysis: Provides detailed insights into network performance and identifies any potential bottlenecks or issues in communication.

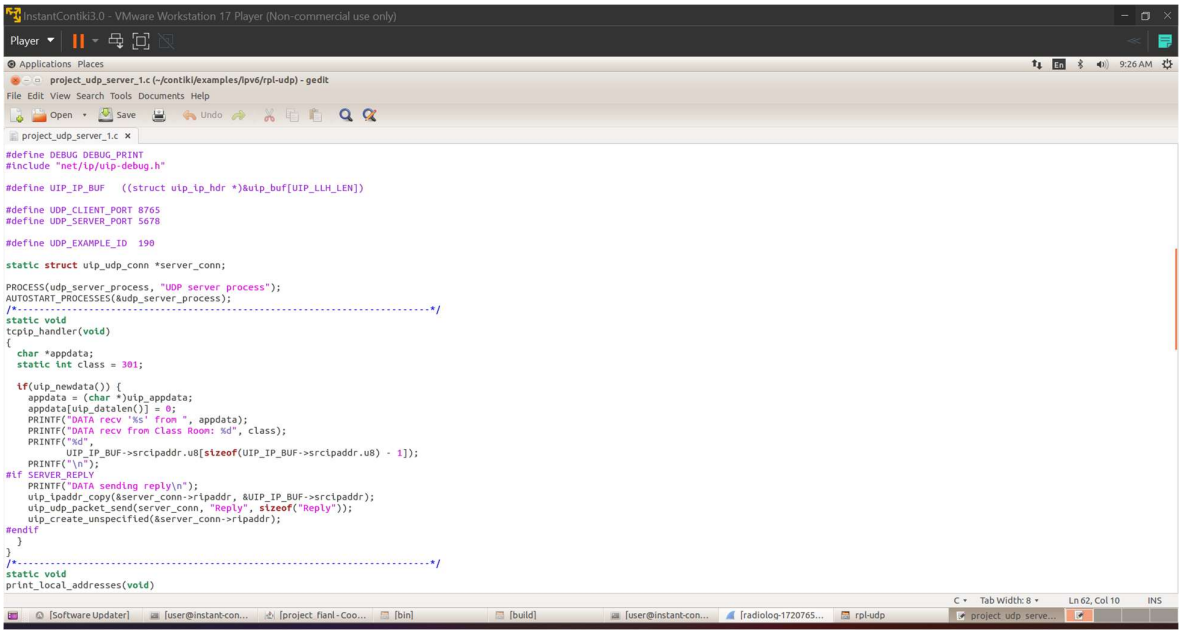
Modules and Functions:

- Sensor and Actuator Modules: Handle the detection of teacher's entry and trigger the connection process.
- Communication Protocols: Implement the RPL protocol for routing and ensuring reliable data transmission.
- Security Checks: Functions to validate teacher IDs and authorize the connection process.

Client Code : Teacher Mote : Mobile



Sever Code : Class Room Mote : Static



Challenges and Solutions:

- Simulating Realistic Mobility: Creating accurate mobility models to reflect real-world movements of teachers.
- Solution: Developed custom mobility models based on observational data and refined them through iterative testing.

- Ensuring Secure Connections: Preventing unauthorized access and ensuring only the correct teacher's content is displayed.
- Solution: Implemented robust ID validation and security checks in the communication protocol.

- Optimizing Resource Usage: Managing the limited memory and processing power of Sky Motes.
- Solution: Optimized code to minimize memory usage and improve power efficiency, using Contiki's modular design to include only necessary components.

- Network Reliability: Maintaining reliable connections in a dynamic and potentially congested network environment.
- Solution: Used the RPL protocol for efficient routing and implemented error-checking mechanisms to handle packet loss and retransmissions.

By addressing these challenges, the project ensures a reliable and efficient IoT-based automatic teacher helper system that enhances the classroom experience.

5. Results and Discussion

Project Outcomes:

- Automated Connection Process: Successfully developed and tested an IoT-based system that automatically connects teachers' laptops to classroom monitors upon entering the room.
- Reduced Setup Time: Teachers' setup time was significantly reduced, allowing more time for teaching.
- Secure and Accurate Content Display: The system reliably validated teacher IDs, ensuring secure and accurate content display.
- Efficiency Improvement: The classroom setup process was streamlined, reducing technical issues and disruptions.

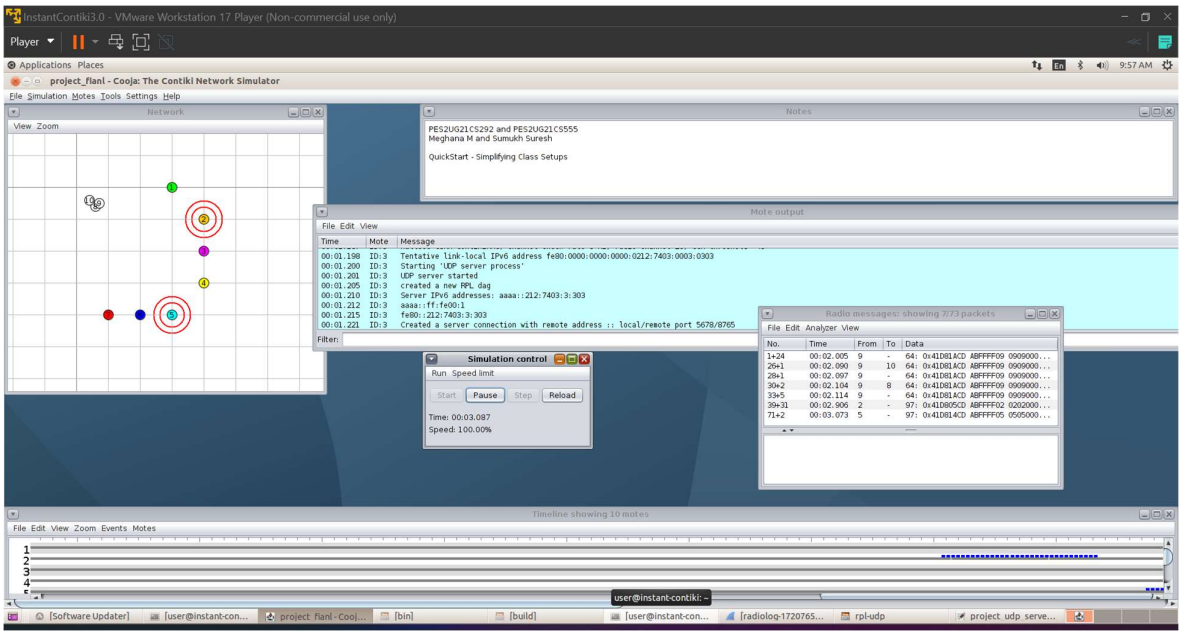
Significance of Results:

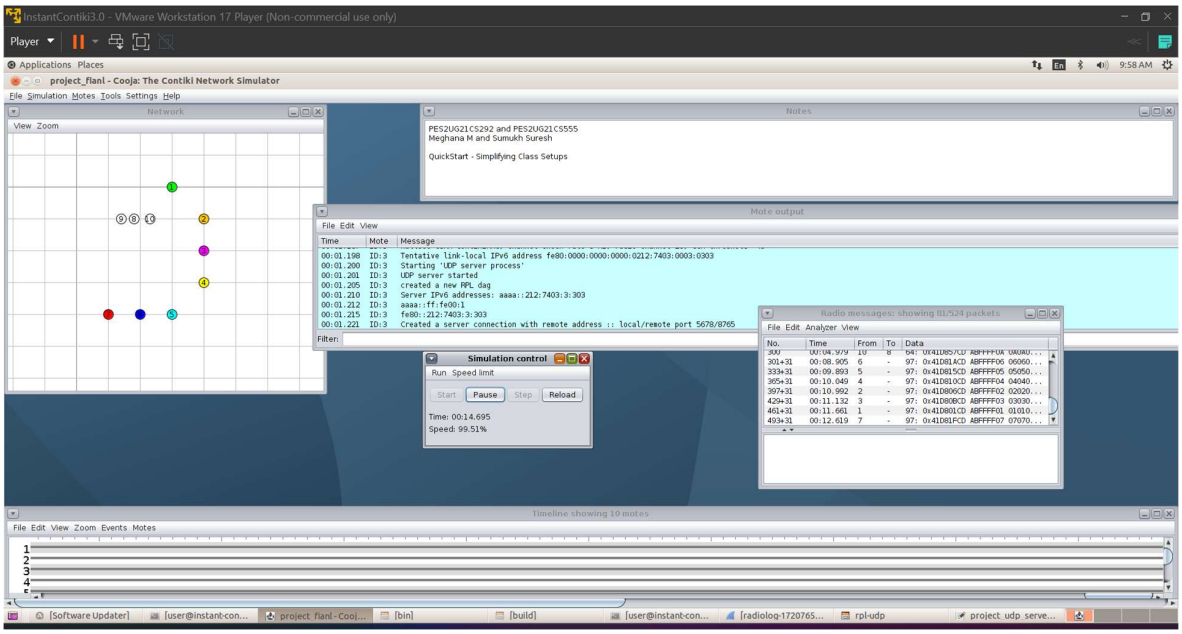
- Enhanced Teaching Efficiency: By automating the connection process, teachers can start their lessons promptly without technical delays, leading to more efficient use of class time.
- Improved Educational Experience: The seamless setup process allows teachers to focus on teaching rather than dealing with technical issues, thereby improving the overall educational experience for students.
- Practical Application of IoT: Demonstrates the practical benefits of IoT in educational settings, providing a real-world example of how technology can enhance daily operations.
- Security Assurance: The robust ID validation process ensures that only authorized teachers can trigger the connection, maintaining the integrity and security of classroom presentations.

Insights Gained:

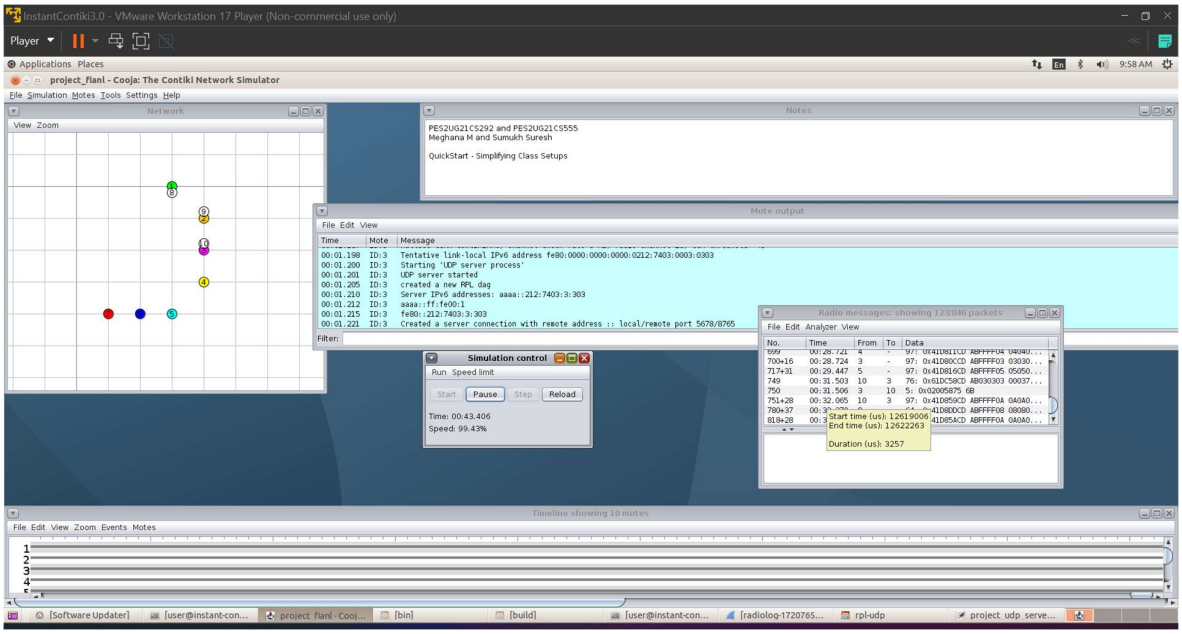
- Importance of Custom Mobility Models: Realistic mobility models are crucial for accurately simulating and addressing the movements and interactions of users in IoT applications.
- Network Optimization: Efficient routing and error-handling protocols are essential for maintaining reliable network performance in dynamic environments.
- Resource Management: Optimizing code for memory and power efficiency is vital for the successful deployment of IoT systems on constrained hardware.

Visual Aids:

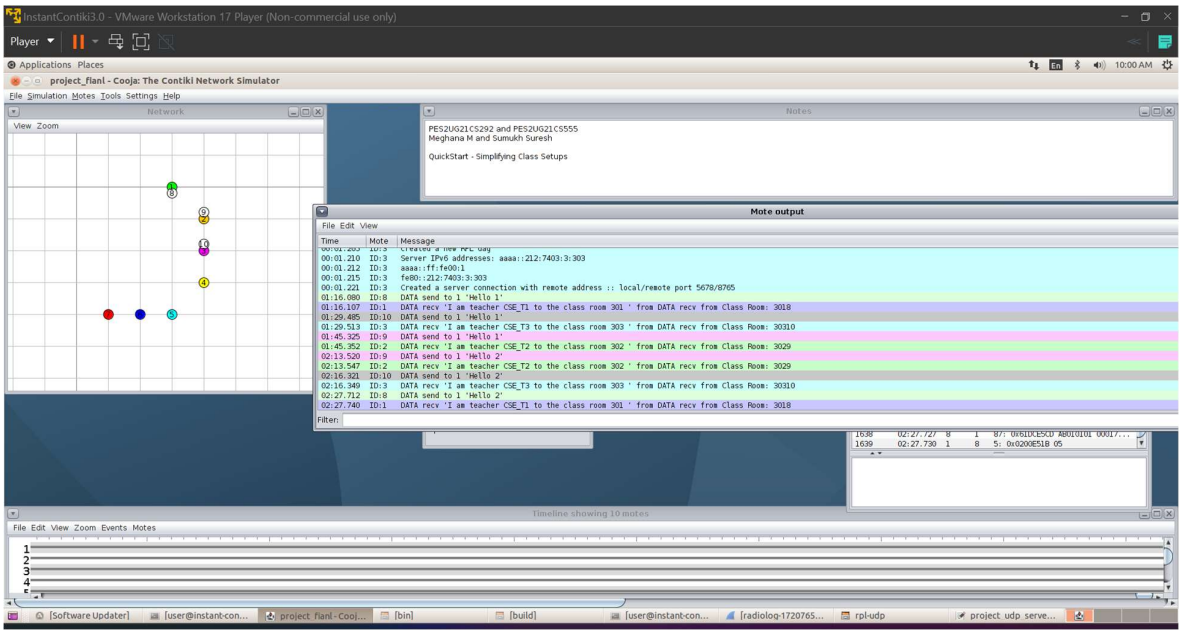


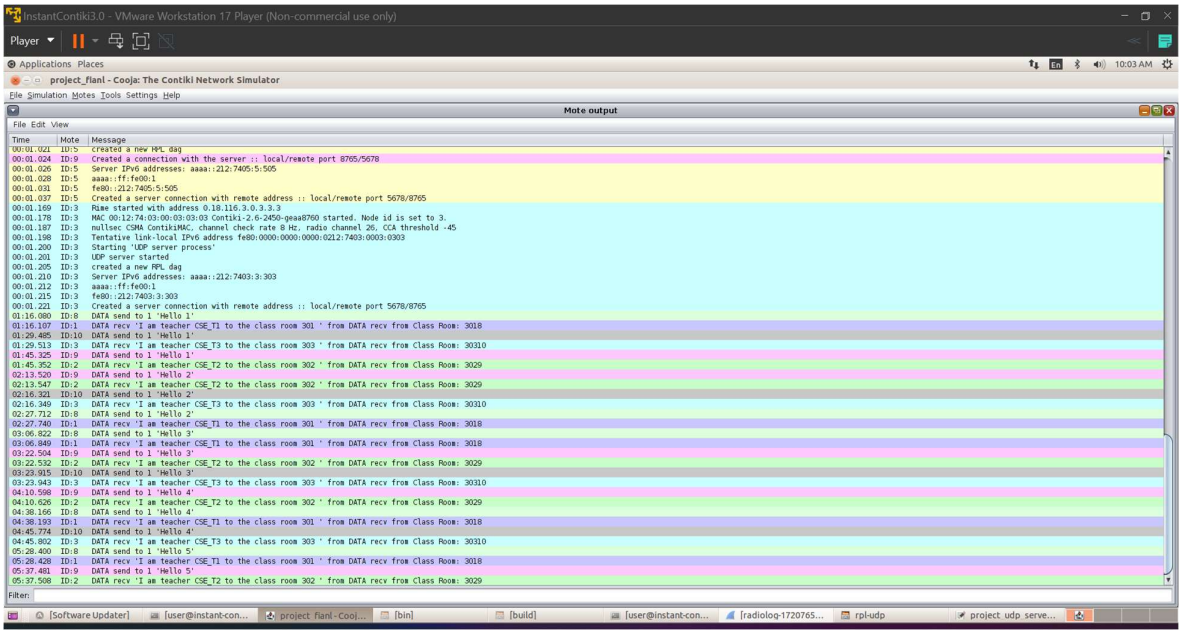


8,9 and 10 are the teacher nodes going to their class rooms

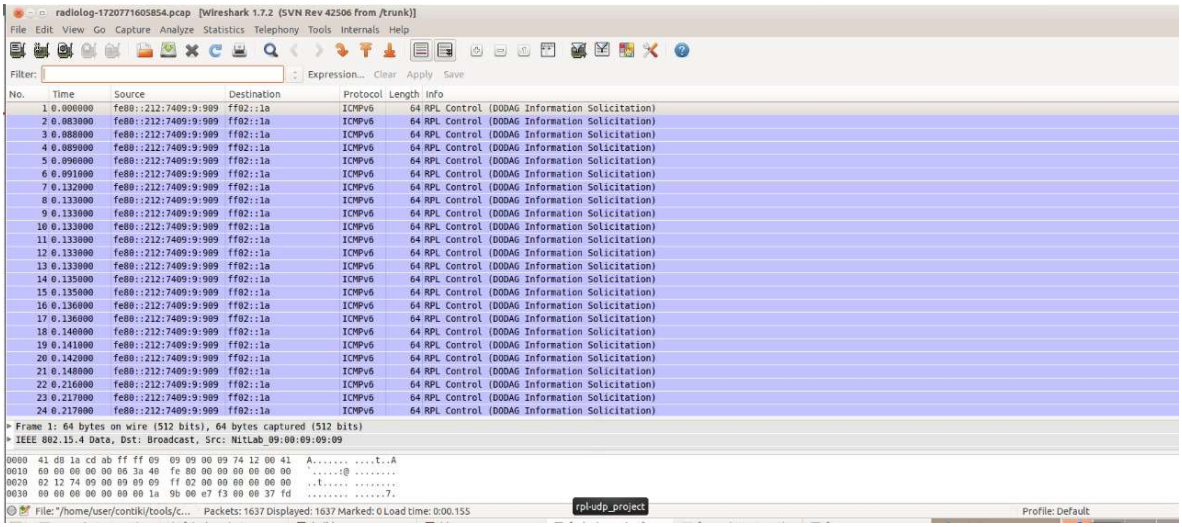


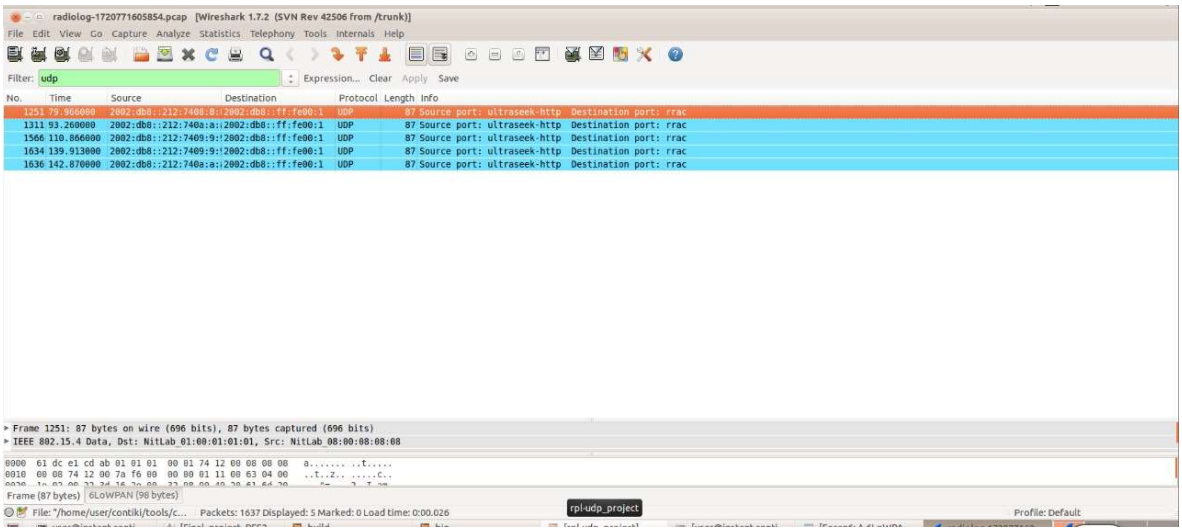
Observe Mote Output:

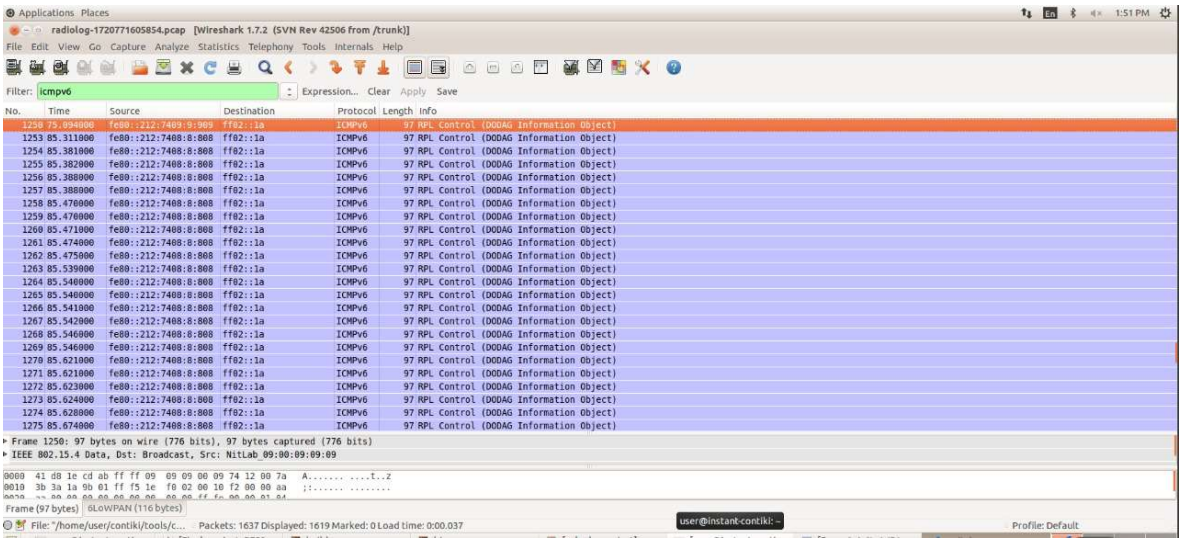


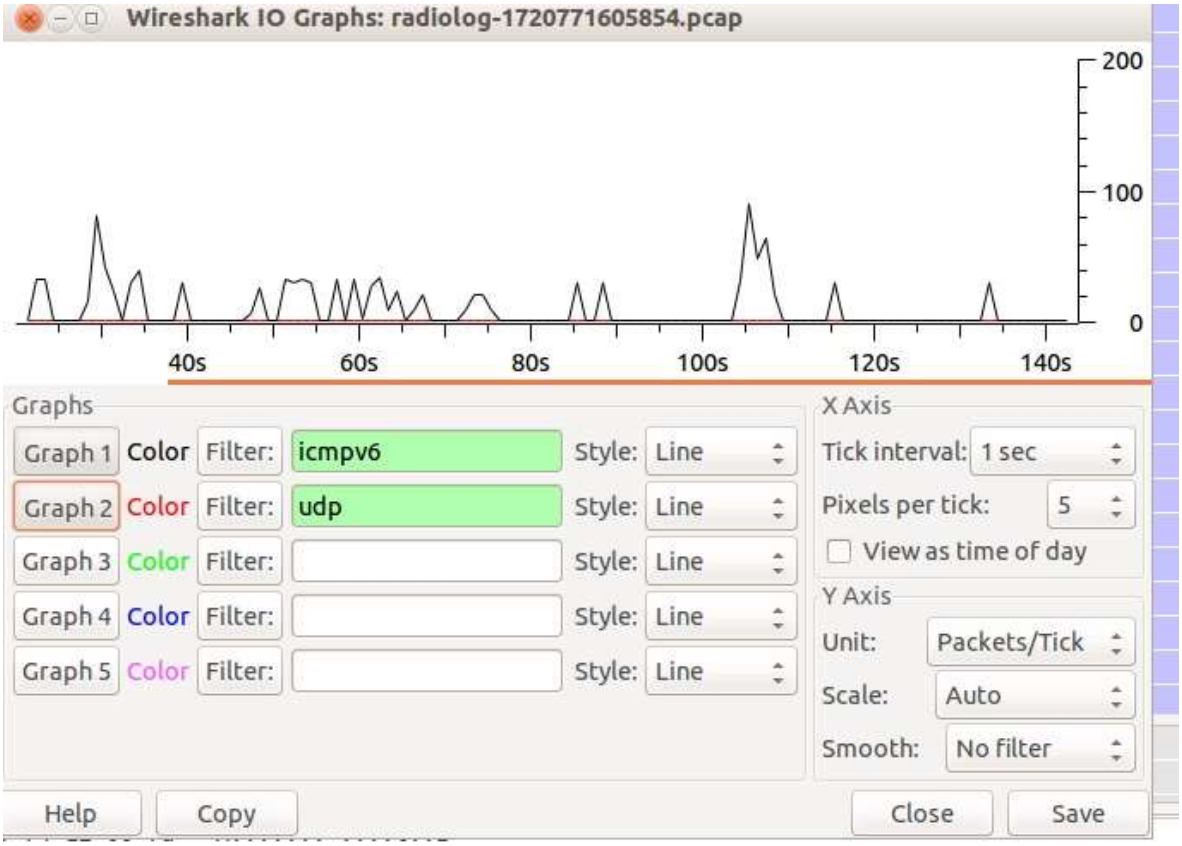


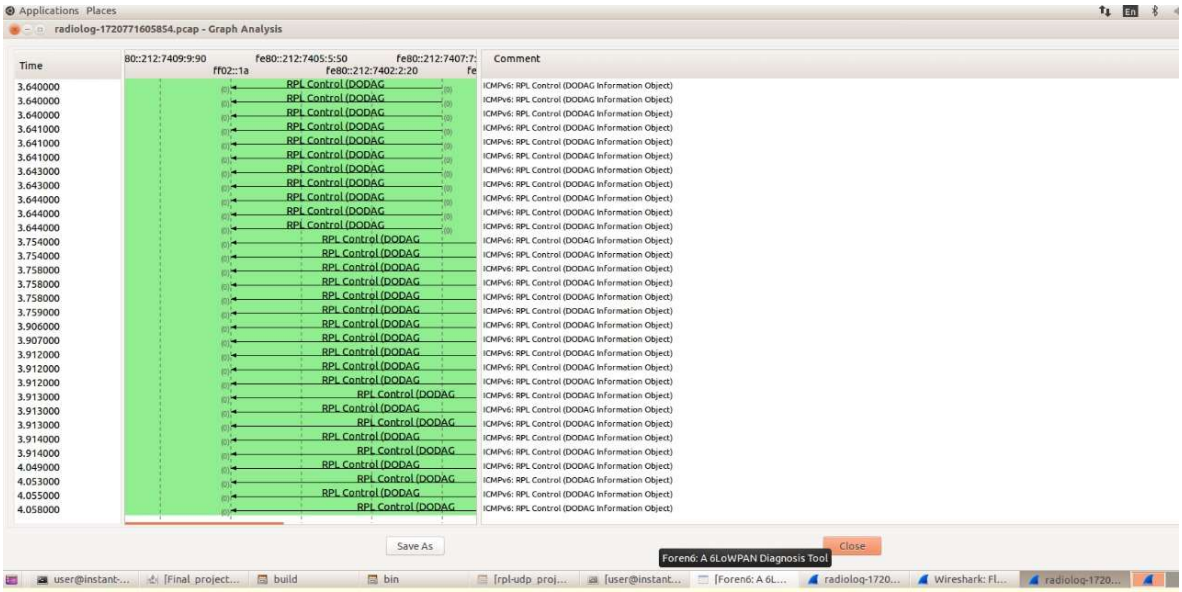
Wireshark Visualization with appropriate filters and statistics:

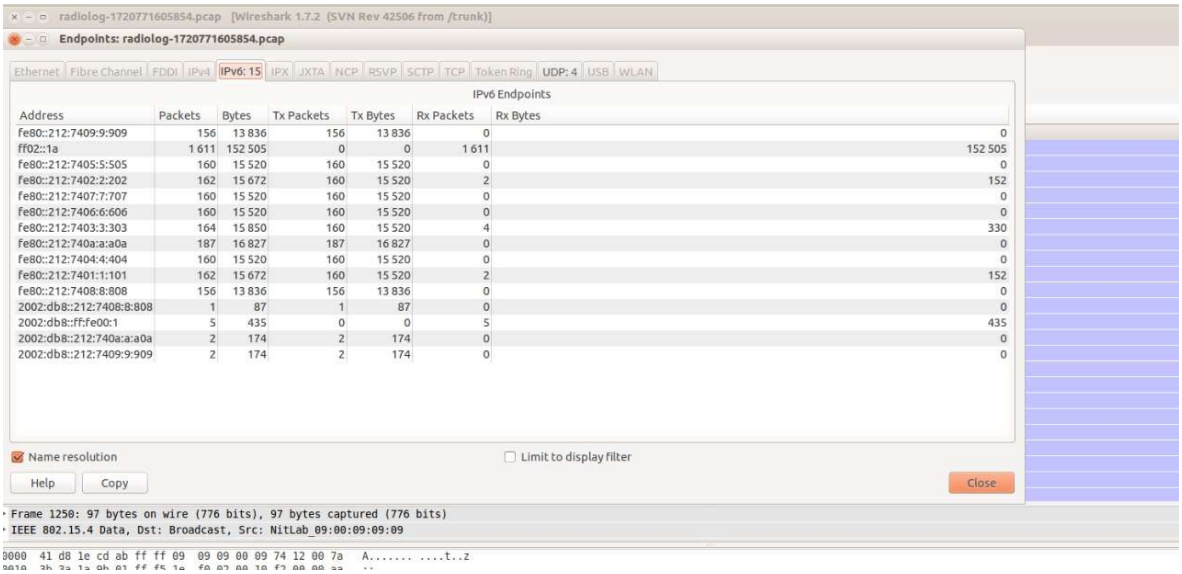




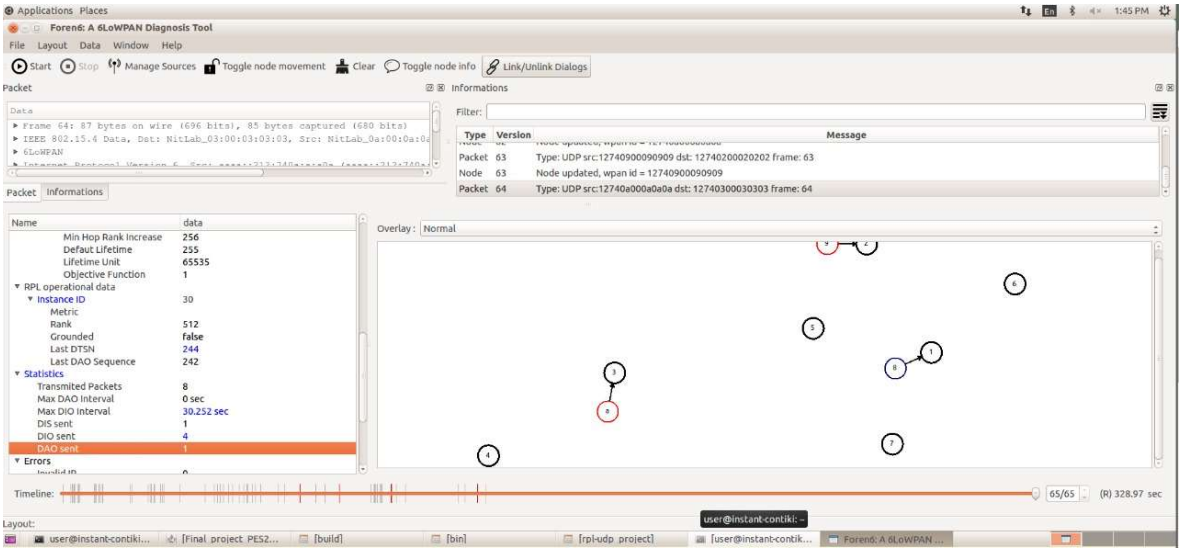

Applications Places
radiolog-1720771605854.pcap [Wireshark 1.7.2 (SVN Rev 42506 from /trunk)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: icmpv6
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1250 75.094000 fe80::212:7409:9:909 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1253 85.311000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1254 85.381000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1255 85.382000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1256 85.388000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1257 85.388000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1258 85.470000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1259 85.470000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1260 85.471000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1261 85.474000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1262 85.475000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1263 85.539000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1264 85.540000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1265 85.540000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1266 85.541000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1267 85.542000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1268 85.546000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1269 85.546000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1270 85.621000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1271 85.621000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1272 85.623000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1273 85.624000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1274 85.628000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
1275 85.674000 fe80::212:7408:8:808 ff02::1a ICMPv6 97 RPL Control (DODAG Information Object)
Frame 1250: 97 bytes on wire (776 bits), 97 bytes captured (776 bits)
IEEE 802.15.4 Data, Dst: Broadcast, Src: NitLab_09:00:09:09:09
0000 41 d8 1e cd ab ff ff 09 09 09 00 09 74 12 00 7a A....... ....t..z
0010 3b 3a 1a 9b 01 ff f5 1e f0 02 00 10 f2 00 00 aa ;:...... ....
Frame (97 bytes) 6LoWPAN (116 bytes)
File: "/home/user/contiki/tools/c... Packets: 1637 Displayed: 1619 Marked: 0 Load time: 0:00.037
user@instant-contiki: ~
Profile: Default

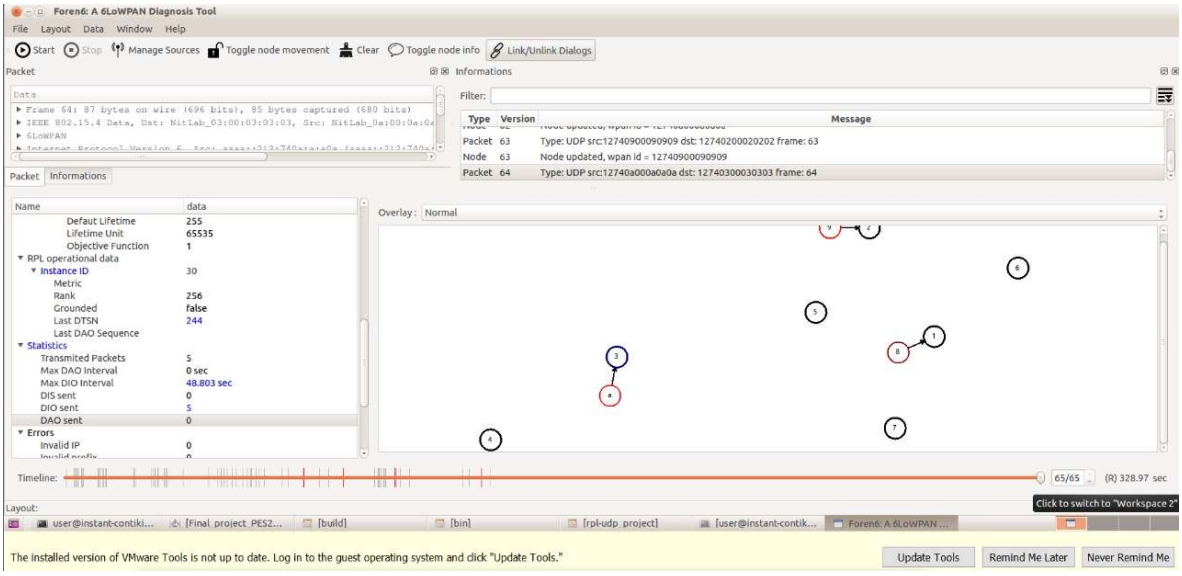

Wireshark IO Graphs: radiolog-1720771605854.pcap
200
100
0
40s
60s
80s
100s
120s
140s
Graphs
Graph 1 Color Filter: icmpv6 Style: Line
Graph 2 Color Filter: udp Style: Line
Graph 3 Color Filter: Style: Line
Graph 4 Color Filter: Style: Line
Graph 5 Color Filter: Style: Line
X Axis
Tick interval: 1 sec
Pixels per tick: 5
View as time of day
Y Axis
Unit: Packets/Tick
Scale: Auto
Smooth: No filter
Help
Copy
Close
Save


Applications Places
radiolog-1720771605854.pcap - Graph Analysis
Time
Comment
RPL Control (DODAG
ICMPv6: RPL Control (DODAG Information Object)
Save As
Close
Foren6: A 6LoWPAN Diagnosis Tool


radiolog-1720771605854.pcap [Wireshark 1.7.2 (SVN Rev 42506 from /trunk)]
Endpoints: radiolog-1720771605854.pcap
Ethernet Fibre Channel FDDI IPv4 IPv6: 15 IPX JXTA NCP RSVP SCTP TCP Token Ring UDP: 4 USB WLAN
IPv6 Endpoints
Address Packets Bytes Tx Packets Tx Bytes Rx Packets Rx Bytes
fe80::212:7409:9:909 156 13 836 156 13 836 0 0
ff02::1a 1 611 152 505 0 0 1 611 152 505
fe80::212:7405:5:505 160 15 520 160 15 520 0 0
fe80::212:7402:2:202 162 15 672 160 15 520 2 152
fe80::212:7407:7:707 160 15 520 160 15 520 0 0
fe80::212:7406:6:606 160 15 520 160 15 520 0 0
fe80::212:7403:3:303 164 15 850 160 15 520 4 330
fe80::212:740a:a:a0a 187 16 827 187 16 827 0 0
fe80::212:7404:4:404 160 15 520 160 15 520 0 0
fe80::212:7401:1:101 162 15 672 160 15 520 2 152
fe80::212:7408:8:808 156 13 836 156 13 836 0 0
2002:db8::212:7408:8:808 1 87 1 87 0 0
2002:db8::ff:fe00:1 5 435 0 0 5 435
2002:db8::212:740a:a:a0a 2 174 2 174 0 0
2002:db8::212:7409:9:909 2 174 2 174 0 0
Name resolution
Limit to display filter
Help
Copy
Close
Frame 1250: 97 bytes on wire (776 bits), 97 bytes captured (776 bits)
IEEE 802.15.4 Data, Dst: Broadcast, Src: NitLab_09:00:09:09:09

Foren6 Visualization:





By analyzing these results and insights, the project demonstrates the feasibility and benefits of an automated classroom setup system, highlighting the potential of IoT technology to enhance educational environments.

6. Conclusion

Developed an IoT-based system using Contiki OS and Cooja simulator to automate classroom setup.

Reduced teachers' setup time significantly, improving instructional efficiency.

Ensured secure and accurate content display through robust ID validation.

Demonstrated practical application of IoT technology in enhancing educational environments.

Future Work:

Scale system to support more classrooms and dynamic teacher assignments.

Develop advanced mobility models and enhance security features.

Create a user-friendly interface and integrate with other IoT devices.

Implement data analytics for performance monitoring and optimization.