SDN in IoT

SUBMITTED BY
Aman Sirohi [CB.EN.U4AIE21003]
Vikhyat Bansal [CB.EN.U4AIE21076]
R Sriviswa [CB.EN.U4AIE21046]
Rakhil M L [CB.EN.U4AIE21048]

SUPERVISED BY
prof. Jaisooraj

AMRITA VISHWA VIDYAPEETHAM
CENTRE FOR EXCELLENCE IN COMPUTATIONAL ENGINEERING AND NETWORKING
Amrita School of Engineering, Coimbatore

December, 2023

INTRODUCTION

In the rapidly evolving landscape of technology, the fusion of Software-Defined Networking (SDN) and the Internet of Things (IoT) emerges as a promising avenue for reshaping the way our interconnected devices function. SDN serves as the orchestrator, akin to a conductor guiding a symphony, ensuring seamless communication and coordination among smart devices. By leveraging a combination of theoretical exploration, simulation studies, and real-world experiments, the project endeavours to bridge the theoretical and practical realms, ensuring that proposed solutions not only sound promising on paper but also demonstrate efficacy in real-world IoT scenarios. A successful integration of SDN in IoT could pave the way for a future where smart devices seamlessly communicate, offering enhanced efficiency and connectivity. As we embark on this journey at the crossroads of SDN and IoT, the possibilities unveiled may redefine how our interconnected world operates, bringing forth a new era of intelligent and responsive technological ecosystems.

MOTIVATION

Lack of Cryptographic Algorithms that are efficient and feasible to be used for secure communication between Devices in SDN-based IoT Network. Proliferation of IoT devices, Emergence of Edge Computing and Advancement in Artificial Intelligence. providing tailor-made network solutions to align with customer expectations for personalised and efficient services. Efficient resource allocation and low-latency communication are paramount for applications ranging from smart cities to healthcare. By enhancing dynamic prioritization and adaptability, the proposed solution aims to elevate the Quality of Service (QoS) parameters.

1.1 Problem statement
To improve security in SDN-enabled IoT Networks with respect to Confidentiality protection and Integrity verification.

1.2 Proposed Methodology
To address the issue of lack of encryption algorithms which could be feasibly used in SDN-based IoT Networks due to Resource Constraint, I have proposed a Lightweight Cryptographic Algorithm which utilizes the high computation power of SDN for computation intensive Key Generation tasks while the Encryption-Decryption steps that are performed in End Systems are lightweight enough to cause no issues with the lack of resources.

1.3 Tools identified
- **CrypTool 2 (CT2):** It includes not only the encryption and cryptanalysis of ciphers, but also their basics and the whole spectrum of modern cryptography with over 200 ready-to-use templates with workflows. We can also easily combine and execute cryptographic functions to create workflows in CT2 by ourselves (visual programming)
- **Cryptol (The Language of Cryptography):** Cryptol is a domain-specific language

for specifying cryptographic algorithms. A Cryptol implementation of an algorithm resembles its mathematical specification more closely than an implementation in a general-purpose language.
- **GNS3:** Graphical Network Simulator for simulating network topologies and protocols. Allows integrating custom programs for encryption simulation.

1.4 Conclusion Future works
- It isn't just lightweight and fast, but provides a great deal of security as well, since backtracking in ECC is still one of the most complex tasks with a very low success rate. Additionally, I have used ECDSA to ensure integrity is maintained.
- I have implemented an algorithm that offers even greater security and ensures integrity in SDN-based IoT Networks. However, I haven't presented it here due to the absence of benchmarks on the computation power of End Devices. I had doubts about whether my algorithm would work for almost all IoT Constrained Devices.
- For future scope, I aim to incorporate a methodology to induce more nonlinearity in my key generation and plan to use Hashing Functions as well. My goal is to extend this work to a Conference Paper.

2.1 Problem Statement
To improve QoS parameter such as latency, load balancing, bandwidth and real time decision in networks based on SDN-IoT. It is crucial to improve above QoS parameters like latency as it can drastically improve the real time result and decision making based on those results. Practical application of the same is Tesla's Self Braking system.

2.2 Proposed Methodology
To address the current situation of QoS parameter, we propose the utilization of Deep Reinforcement Learning integrated in edge computing based SDN-IoT network where

an optimized algorithm will be implemented in every edge node on reducing computational complexity on those edge nodes while maintaining Quality of Service (QoS).

Benefits:

• **Reduced Latency:**

– Edge Processing: By performing deep learning tasks directly on the edge computing nodes, data processing occurs closer to the source of data generation. This significantly reduces the round-trip time to a centralized data center, resulting in lower latency and faster response times.

• **Bandwidth Optimization:**

– Local Data Processing: Edge computing enables local processing of data, reducing the need to transmit large volumes of raw data to a central server. Only relevant information or processed results may be sent, leading to efficient bandwidth usage and reduced network congestion.

• **Real-time Decision-Making:**

– Immediate Responses: Deep learning models running on edge devices can make real-time decisions without relying on cloud-based processing. This is critical for applications requiring instant responses, such as autonomous vehicles, industrial automation, or augmented reality.

2.3 Tools Identified

ONF(Open Network Operating System), MiniNet

2.4 Conclusion

In conclusion, the proposed solution of utilizing reinforcement learning on edge nodes can be a promising approach as it can drastically improve the QoS depending upon the requirement of the data that is being going to and from IoT devices. An optimized algorithm can help in reducing traffic congestion and deciding that what kind of data is required to be computed on the edge nodes and what kind of data can be transmitted to cloud for computation purpose which reduces congestion and latency.

2.5 Future Work

The proposed solution should be implemented and evaluated in practical applications to assess its performance under realistic conditions. Field trials would provide valuable insights into the effectiveness of the solution. Keep on looking for methods other than reinforcement learning and choosing from them if they are better than current implementation. Rather than a trade-off between traditional optimization methods and new machine learning algorithms, we can look for a way where both can be used at the same time as a hybrid solution where strengths of both can be leveraged.

3.1 Problem Statement

To improve data transmission in SDN-IoT networks with respect to latency and bandwidth via Network Slicing

3.2 Proposed Methodology

Collect network data for predictive analysis, process it with Python libraries, and create SDN slices based on predictions. Monitor real-time performance and automate resource allocation. Enforce security and document configurations. Utilize predictive insights to

drive QoS policies, traffic engineering, and routing optimization, ensuring efficient resource usage, customized service levels, and adaptive network responsiveness.

3.3 Tools Identified

Data Collection: **PRTG** Network Monitor capture and analyse network traffic to extract flow data for predictive analytics.
Data Processing: **Pandas and NumPy** for data manipulation, **Scikit-learn** and **TensorFlow** for building predictive models.
SDN Controllers: **OpenDaylight** provide APIs and interfaces to manage SDN networks.
Real-time Monitoring Solutions: **Grafana** offers real-time monitoring capabilities, enabling adjustments based on live network conditions.
Simulation Tools: **Mininet** helps simulate network environments for testing and validation before deployment.

3.4 Conclusion and Future Work

Implementing network slicing with predictive algorithms empowers adaptable, efficient, and responsive networks. By leveraging predictive insights, SDN slices are tailored for optimal resource allocation, QoS fulfillment, and traffic management. This approach fosters enhanced performance, cost-effectiveness, and scalability, catering to diverse service requirements while ensuring security and adaptability. Ultimately, network slicing driven by predictive algorithms optimizes operations, ensuring tailored, high-performing services aligned with evolving demands and fostering a robust, future-ready network infrastructure.

4.1 Problem Statement

To improve the performance of flow space distribution in SDN-enabled fog computing for enhanced IoT application performance.

4.2 Proposed Methodology

Introduced a Dynamic Prioritized Flow Space Allocation (DPFSA) system leveraging SDN and fog computing. Implementing dynamic prioritization, adaptive allocation, and cross-layer coordination to optimize flow space distribution. Utilized protocols like OpenFlow, MQTT, and custom protocols for efficient communication.

4.3 Tools Identified

Simulation Tools : **NS-3, Mininet, or OMNeT++** for discrete-event simulatoion framework for networking
Machine Learning Tools : **TensorFlow or PyTorch** for building and training machine learning models and deep learning framework with a dynamic computational graph.
Security Tools : **Wireshark** for monitoring network traffic.
Network Monitoring : **Nagios or Icinga** for continuous monitoring of network performance and resource usage.
Visualization: **Gephi or Matplotlib** for visualizing simulation results, network topology.
Containerization: **Docker** for developing, shipping, and running applications in containers.
Orchestration : **Kubernetes** for automating the deployment, scaling, and management of containerized applications.
Real-time Metrics Monitoring: **Prometheus and Grafana** for analytics and monitoring.

4.4 Conclusion and Future work

The proposed Dynamic Prioritized Flow Space Allocation (DPFSA) system demonstrates significant advancements in optimizing flow space distribution for enhanced IoT application performance. The integration of dynamic prioritization, adaptive allocation, and cross-layer coordination addresses existing challenges, fostering improved Quality of Service (QoS) parameters. Implementing this theoretical approach to testing with the tools and technologies identified, we can test how well it works with bigger networks and in different situations.

1.1 problem statement -1 [Aman Sirohi [CB.EN.U4AIE21003]]
2.1 problem statement -2 [Vikhyat Bansal [CB.EN.U4AIE21076]
3.1 problem statement -3 [R Sriviswa [CB.EN.U4AIE21046]
4.1 problem statement -4 [Rakhil M L [CB.EN.U4AIE21048]