Characterizing a NodeMCU-based MAC Address Detector for Enhanced Coastal Tourism Management System

Yoanda Alim Syahbana[1], Rufina Pramudita[2]
yoanda@pcr.ac.id[1], rufina@alumni.pcr.ac.id[2]

Politeknik Caltex Riau, Pekanbaru, Indonesia[1], Politeknik Caltex Riau, Pekanbaru, Indonesia[2]

Abstract. The coastal tourism management system plays a crucial role in ensuring sustainable tourism activities and integrates various elements and stakeholders of tourism. Leveraging the potential of Internet of Things (IoT) technologies, this research focuses on characterizing a NodeMCU-based MAC Address Detector. NodeMCU is a cost-effective System-on-Chip (SoC) ESP8266, constructed with open-source hardware and software. In this research, the NodeMCU is used to prototype an IoT product, MAC Address Detector. It is aimed to effectively address the issue of overcrowding in coastal tourism management systems. The proposed solution uses NodeMCU, which comes pre-equipped with a WiFi chip functioning as a sensor. The monitoring concept revolves around identifying active MAC addresses derived from smartphones or tablets used by visitors. The characterization covers experiments on detection distance, device brand, Operating System (OS), and active connection type. Results from the experiment show that device connection, OS, and distance are the most critical factors in implementing the solution. Excluding the characterizing result from airplane mode connection type, iPhone device with iOS, and GPS connection type, the proposed idea achieves a 98.36% success rate in detecting nearby MAC addresses for all distance variation. Based on this result, the implementation of NodeMCU as a sensor to detect overcrowding in coastal tourism is possible to be implemented. In the future, it will be challenging to continue the research for multiple detectors across a broad coastal area to evaluate their coverage, accuracy, response time, and energy consumption performance.

Keywords: Internet of Things, NodeMCU, MAC Address, Coastal Tourism Management System.

1 Introduction

The coastal tourism management system is essential to ensure the sustainable development of tourism activities [1]. The system integrates various tourism elements and stakeholders, i.e., environment, community, visitor, culture, and government. The system supports protecting and conserving coastal ecosystems and ensures the preservation of natural resources in coastal areas. The system also considers the community's voice and interest to ensure that coastal tourism will become sustainable with significant benefits. In line with this, it will also enhance the visitor experience for tourism activities. The system also considers the uniqueness of culture with its tradition and belief to assure appreciation for local values. In addition, when collaborating and coordinating with the government, the system can support the implementation of regulation, planning, and stakeholder management.

ABEC 2023, September 21, Bengkalis, Indonesia

DOI 10.4108/eai.21-9-2023.2342975

One of the key advantages of implementing a coastal tourism management system is to mitigate potential risks associated with coastal tourism. This research is focused on the issue regarding overcrowding in coastal tourism. Overcrowding happens when the number of visitors surpasses the capacity of coastal tourism capacity or its infrastructure [2]. Overcrowding may lead to an endangered natural environment and coastal area ecosystem. Moreover, overcrowding may also cause infrastructure strain, such as transportation, waste management, and accommodation. Overcrowding also influences safety concerns in case of emergencies. Addressing this issue is a challenging task that requires a comprehensive approach. The solution covers visitor management strategies, coastal tourism capacity assessment, effective zoning and planning.

Recent developments in dealing with overcrowding issues in coastal tourism include carrying capacity assessment, destination management plans, innovative visitor management strategies, and the use of smart technologies and data analytics [3][4]. Carrying capacity assessment has limitations of complexity and subjectivity of the assessment. The subjectivity is mainly influenced by making assumptions about visitor behavior and capacity of the infrastructure. It also requires accurate capacity data that must consider dynamic change of the environment in the coastal tourism [5]. Destination management plans and innovative visitor management strategies may deal with visitor resistance that prefer freedom in enjoying the coastal area. Failure to include all stakeholder also lead to resistance and conflicts [6]. Especially for using intelligent technologies, some examples can be explored in the case of real-time visitor management, predictive analytics for visitor management, and IoT-based intelligent monitoring and management systems.

IoT-based smart monitoring and management systems have advantages of real-time data that support decision-making and resource optimization with scalability and flexibility. This research proposed the implementation of the Internet of Things (IoT) as an alternative solution to detect overcrowding in coastal tourism. The proposed solution utilizes NodeMCU, which is already equipped with a WiFi chip used as a sensor to monitor the environment in coastal tourism. The monitoring idea is based on detected active MAC addresses that are obtained from visitors' smartphones or tablets. The unique MAC address for each device is used in this research to indicate crowdedness in nearby areas. The initial implementation of the idea is to characterize the NodeMCU as a MAC address detector. The characterization covers experiments on detection distance, device brand, Operating System (OS), and active connection type. This research aims to contribute to using intelligent technologies and data analytics to deal with overcrowding issues in coastal tourism.

This paper that explains the characterization process of a NodeMCU-based MAC address detector is arranged as follows: First section explains the overview, specific problem, current development, and proposed solution. Then, the second section explains the materials and methods that are used in this research. The third and fourth sections focus on the result of the experiment conducted in this research and its discussion. Finally, the last section is aimed to conclude the research results and to guide further research in the future.

2 Materials and Method

This section discusses materials and methods that are used in the research.

2.1 Materials

NodeMCU is a cost-effective System-on-Chip (SoC) ESP8266-based platform that is both software and hardware open-source. NodeMCU, as a 32-bit microcontroller development board, is equipped with a WiFi module that enables developers to connect IoT projects to WiFi networks. Some advantages of NodeMCU are compatibility with the Arduino development environment, flexibility as a WiFi client or access point, and enough GPIO for prototyping IoT projects. With its popularity, NodeMCU is also supported by extensive online resources, a library, and community support. Figure 1 shows the pinout of the NodeMCU V3.

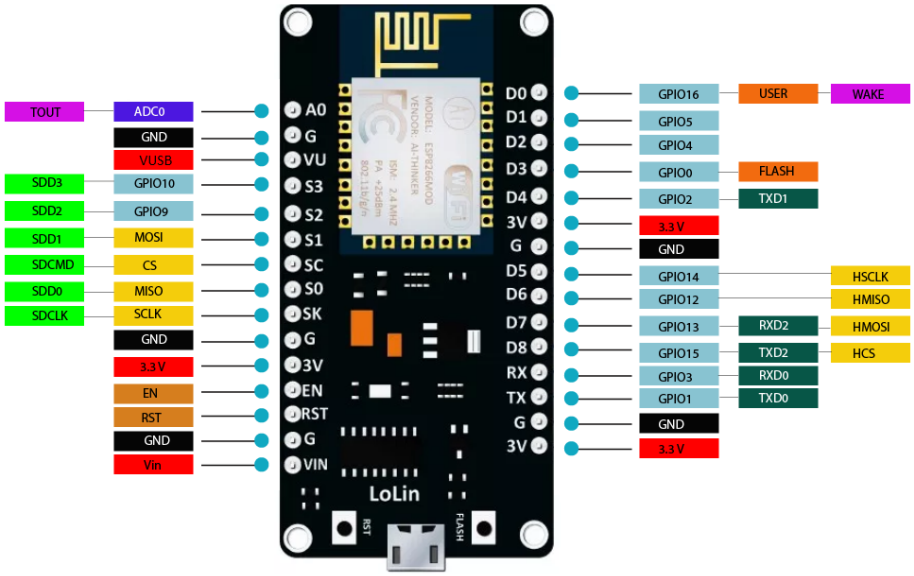


Fig. 1. NodeMCU V3 pinout

As explained in the introduction section, this research is aimed to characterize NodeMCU-based MAC address detectors as a solution to detect overcrowding in coastal tourism. The NodeMCU is used as a sensor to detect the visitor based on their device's MAC address. The devices can be smartphones and tablets that periodically try to communicate with access points or to find devices to connect by sending their MAC address. If the NodeMCU can listen to these unique communications, then the NodeMCU can count the number of nearby devices. Thus, the information may be used to estimate crowdedness in the NodeMCU coverage areas.
NodeMCU code that was used in this experiment is inspired by a project [7] that has been improved by [8] in his GitHub repository. In the previous project, the NodeMCU was designed to detect registered MAC addresses that can be used for many implementations, such as smart home and child surveillance. For example, when the owner comes home, the MAC address that his/her phone transmits can be detected by the NodeMCU in the smart home. Thus, the smart

home can prepare the air conditioning, heating, and lighting. Another example is when parents use the NodeMCU in this project and register their child devices, the nodeMCU can be used as a child surveillance system. If the registered MAC address from the child's device is not detected anymore due to being out of range, the parent can be notified immediately.
Figure 2 illustrates how the NodeMCU, as the primary material in this research, is used to detect nearby devices. Different from the previous projects, the code design in this research is simplified to detect any active MAC address in this research. Any unique and detected MAC address is aggregated to represent how crowded the nearby area of NodeMCU is.

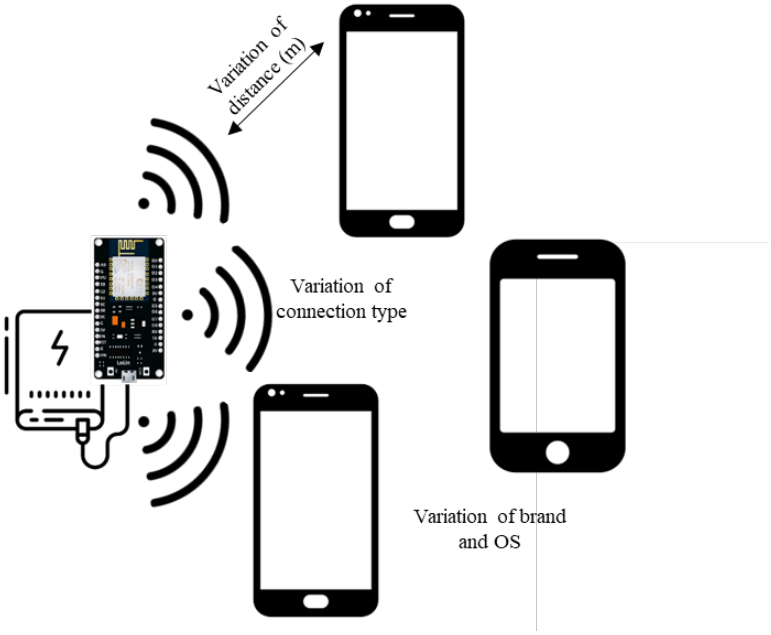


Fig. 2. Experiment diagram

2.2 Methods

Characterizing process is performed with reference to the testing plan [9]. First, the characterizing is defined as on-line testing where it is performed concurrently with the normal system operation. Second, the source of stimuli is applied by an external device that is defined as external testing. Third, the characterization is categorized as design verification testing as proof of concept of the idea. Fourth, the characterizing is performed as system-level testing. Fifth, the stimuli is applied at the normal operation speed. Therefore, it is classified as at-speed testing. Table 1 summarizes a testing plan that is designed to perform the characterizing process in this research.

Table 1. Testing plan for characterizing.

Criterion	Attribute of testing method	Terminology
When is testing performed?	Concurrently with the normal system operation	On-line testing
Where is the source of stimuli?	Applied by external device (tester)	External testing
What do we test for?	Design errors	Design verification testing
What is the physical object being tested?	System	System-level testing

This research varied four parameters as characterizing. The parameters are the device's distance to NodeMCU, brand, Operating System (OS), and type of active connection in the devices. The distances vary by 1 meter, 3 meters, 5 meters, 10 meters, and 30 meters. The characterizing process is performed for seven prominent device brands: iPhone, Samsung, Oppo, Xiaomi, Redmi, Mi, and Vivo. The OS varied by iOS, Android 13, Android 12, Android 10, and Android 9. The characterizing process is tested on three different connections, i.e. WiFi, GPS, and Airplane mode. Each parameter variation is coded to simplify the presentation in this publication. Table 2 summarizes the abbreviations that are used for each of these parameters.

Table 2. Parameter variation and abbreviation.

Parameter	Variation and abbreviation
Distance (m)	1, 3, 5, 10, 30
Brand	iPhone (B1), Samsung (B2), Oppo (B3), Xiaomi (B4), Redmi (B5), Mi (B6), Vivo (B7)
OS	iOS (OS1), Android 9 (OS2), Android 10 (OS3), Android 12 (OS4), Android 13 (OS5)
Connection type	WiFi, GPS, Airplane Mode

Characterizing process involves students of Politeknik Caltex Riau, specifically from the Computer Engineering Technology Study Program. The characterizing process is part of The Protocol Design and Troubleshoot Workshop. The student follows the predetermined test plan and uses their own devices to simulate the experiment.

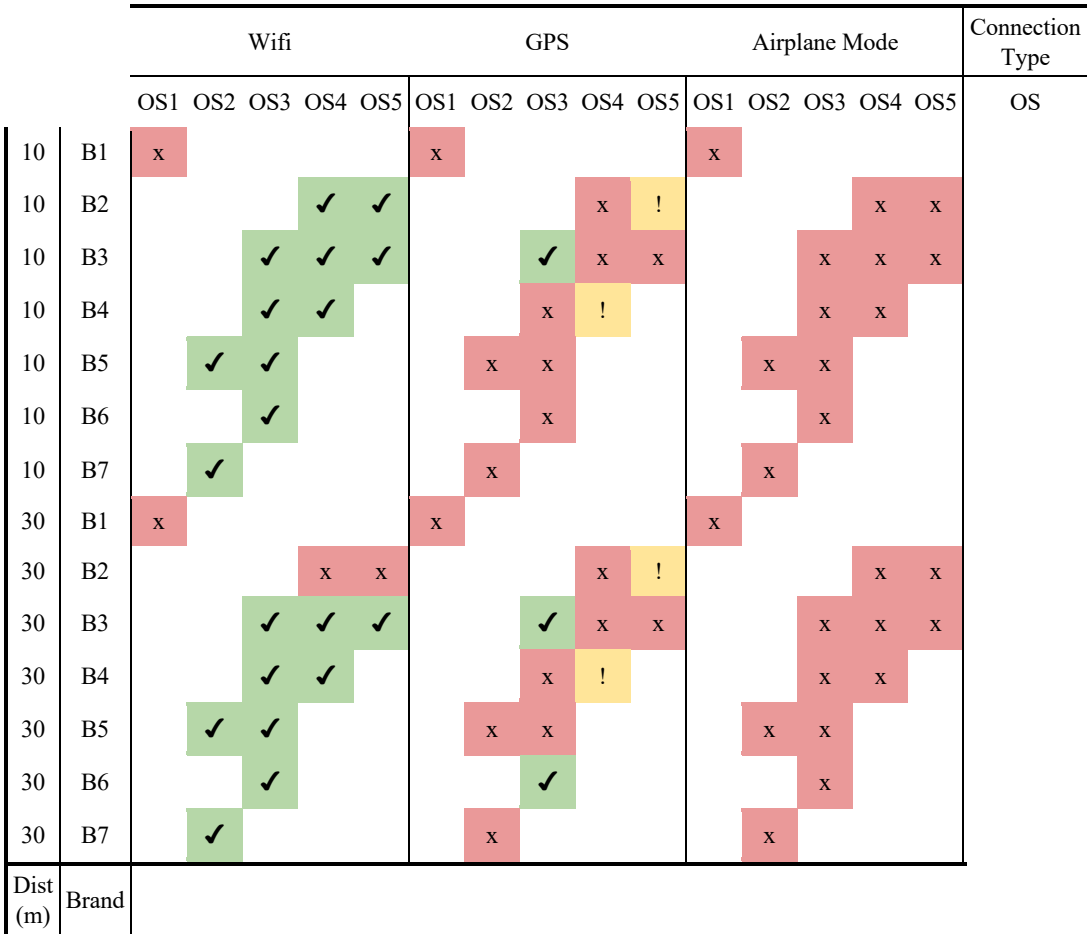
3 Result and Discussion

Table 3 summarizes the result of the characterizing process. The row represents the variation in distance and device brand. The column represents the variation of OS and connection type. The green check indicates that the NodeMCU can detect the MAC address of the devices. The red cross indicates that the NodeMCU cannot detect the MAC address of the devices. The exclamation mark indicates that sometimes the MAC address is detected, and sometimes not.
The result shows that airplane mode connection in the devices makes 98.21% of the MAC address cannot be detected by the NodeMCU. It confirms that the airplane mode is specifically purposed to suspend the device's radio frequency signal transmission. The result also shows that

iPhone with iOS performs MAC randomization that automatically spoofs the MAC address. Therefore, for three variations of connection type, 100% of MAC addresses from iPhone with iOS devices cannot be detected by the NodeMCU. The use of GPS as a connection type also influences characterizing results. Based on the experiment, GPS connection gets 72.73% undetected MAC address.

Table 3. Result of characterizing process.

		Wifi					GPS					Airplane Mode					Connection Type
		OS1	OS2	OS3	OS4	OS5	OS1	OS2	OS3	OS4	OS5	OS1	OS2	OS3	OS4	OS5	OS
1	B1	x					x					x					
1	B2				✓	✓				x	!				x	x	
1	B3			✓	✓	✓			✓	x	x			x	x	x	
1	B4			✓	✓				x	!				x	x		
1	B5		✓	✓				x	x				x	x			
1	B6			✓					✓					✓			
1	B7		✓					x					x				
3	B1	x					x					x					
3	B2				✓	✓				x	!				x	x	
3	B3			✓	✓	✓			✓	x	x			x	x	x	
3	B4			✓	✓				x	!				x	x		
3	B5		✓	✓				x	x				x	x			
3	B6			✓					x					x			
3	B7		✓					x					x				
5	B1	x					x					x					
5	B2					✓					✓					x	
5	B3			✓					✓					x			
5	B4			✓	✓				x	!				x	x		
5	B5			✓					x					x			
5	B6			✓					✓					x			
5	B7		✓					x					x				



Excluding the characterizing result from airplane mode, iPhone device with iOS, and GPS connection type, the proposed idea achieves a 98.36% success rate in detecting nearby MAC addresses for all distance variation. Table 4 shows specific results for each distance variation. Based on the characterizing process, the proposed idea cannot detect addresses if the device and the NodeMCU are separated by a thick separator such as a wall that influences line of sight between the device and the NodeMCU. In addition, some brands and OSs have the mechanism that sleeps their MAC address broadcasting to optimize its energy consumption.

Table 4. Characterizing result after excluding airplane mode, iPhone device with iOS, and GPS connection type.

Distance (m)	Detected MAC Address (%)	Undetected MAC Address (%)
1	100	0
3	100	0
5	100	0
10	100	0
30	93.3	6.67

Based on these results in Table 4, it can be understood that device connection, OS, and distance are the most critical factors in implementing the solution. As long as the device connection is not in airplane mode with distance under 4 meters, the detection is 100% confirmed. While Iphone OS has mechanism of MAC address masking [10], MAC address from Android based OS is also 100% confirmed. Therefore, this information can contribute to using intelligent technologies and data analytics to deal with overcrowding issues in coastal tourism. Figure 3 shows illustration on how the Internet of Things (IoT) can be placed in coastal tourism areas to detect overcrowding. The NodeMCU is placed in the light pole that is equipped by a public Access Point (AP). Each of the AP is distanced for 15 meters based on the characterizing result. Then, the visitor can connect their device to the public AP and the NodeMCU can use the detected MAC address as an information to estimate the overcrowding.

Fig. 3. Solution design for utilizing NodeMCU as overcrowding detection.

4 Conclusion and Future Work

The characterizing process results show that device connection, OS, and distance are the most critical factors in implementing the solution. Based on this result, the implementation of NodeMCU as a sensor to detect overcrowding in coastal tourism is feasible. Overall, this research is a valuable alternative tool for enhancing coastal tourism management systems. Compared to the previous research, understanding the results of the proposed research strengthen the advantages of IoT-based smart monitoring and management systems that are real-time data monitoring, scalability, and flexibility. In the future, it will be challenging to continue the research for multiple detectors across a broad coastal area to evaluate their coverage, accuracy, response time, and energy consumption performance. In addition, it is also important to be concerned about privacy and ethical implications while implementing the proposed solution. Idea of data anonymization and data collection transparency for the visitor are also important to gain stakeholder trust. This study also has limitations in terms of sample size. Therefore, it is also potential to broaden the testing plan.

Acknowledgements

We acknowledge the support received from the Politeknik Caltex Riau. We send special thanks to Computer Engineering Technology Study Program students G19 and G20 for data collection support.

References

[1] Developing Independent Tourist Village in Central Lombok District, Indonesia," Resources, vol. 7, no. 4, p. 69, Nov. 2018, doi: 10.3390/resources7040069.
[2] A. Capocchi, C. Vallone, M. Pierotti, and A. Amaduzzi, "Overtourism: A Literature Review to Assess Implications and Future Perspectives," Sustainability, vol. 11, no. 12. MDPI AG, p. 3303, Jun. 15, 2019. doi: 10.3390/su11123303.
[3] K. Koens, A. Postma, and B. Papp, "Is Overtourism Overused? Understanding the Impact of Tourism in a City Context," Sustainability, vol. 10, no. 12. MDPI AG, p. 4384, Nov. 23, 2018. doi: 10.3390/su10124384.
[4] A. Mandić and J. Kennell, "Smart governance for heritage tourism destinations: Contextual factors and destination management organization perspectives," Tourism Management Perspectives, vol. 39. Elsevier BV, p. 100862, Jul. 2021. doi: 10.1016/j.tmp.2021.100862.
[5] M. MacLeod and J. A. G. Cooper, "Carrying capacity in coastal areas," Encyclopedia of Coastal Science, vol. 35, p. 1211, 2005.
[6] Y. F. Leung, A. Spenceley, G. Hvenegaard, R. Buckley, and C. Groves, "Tourism and visitor management in protected areas: Guidelines for sustainability," vol. 27. Gland, Switzerland: IUCN, 2018.
[7] https://www.hackster.io/ricardooliveira/esp8266-friend-detector-12542e.
[8] https://github.com/skickar/FriendDetector.
[9] Abramovici, M., Breuer, M. A., & Friedman, A. D. (1990). Digital systems testing and testable design (Vol. 2, pp. 203-208). New York: Computer science press.
[10] K.-J. Djervbrant and A. Häggström, "A Study on Fingerprinting of Locally Assigned MAC-Addresses", Dissertation, 2019.