Dynamic Spectrum Access and Autonomous Channel Allocation in Wireless Sensor Networks

Swati Rathore[1], Aravind T[2], Avinash Dubey[3], Aryan[4], Chirayu N Hudar[5]
[1]Assistant Professor Electronics and Communication Atria Institute of Technology Bengaluru, India
[2,3,4,5] Electronics and Communication Atria Institute of Technology (VTU) Bengaluru, India

***Abstract*—The 2.4 GHz ISM band has become critically saturated. The 2.4 GHz range is subject to severe overcrowding issues. With billions of Internet of Things (IoT) devices sending signals at once, the spectrum space becomes prone to collisions constantly. Because of this, static frequency assignment fails. The initial CR and DSA models were developed under the assumptions about large macro-cellular network configurations with servers located centrally. Forcing such an approach on the edge nodes with energy constraints proves to be inefficient, leading to overheating and high processing lags. In this research, we propose another solution by moving from simple thresholding to distributed machine learning on the edge. Specifically, we will look into what is possible when training and utilizing lightweight classification algorithms, such as Random Forest, on energy-constrained microcontrollers. This paper intends to reduce the higher than desirable number of false alarms generated by affordable sensors. Moreover, we deal with MAC-layer latency issues. Avoiding Wi-Fi association overheads, we showcase how intelligent secondary nodes can switch channels in microseconds without the need for any connections**

***Index Terms*—Cognitive Radio, Dynamic Spectrum Access, Edge Computing, Machine Learning, Media Access Control, Wireless Sensor Networks.**

I. INTRODUCTION

The 2.4 GHz Industrial, Scientific, and Medical (ISM) band is currently exhibiting heavy spectral saturation. Billions of independent IoT sensors compete with traditional Wi-Fi and Bluetooth transmitters for channel capacity simultaneously . In case of mission-critical WSNs used in automation and urban surveillance systems, high traffic load in the channel causes considerable latency jumps. Given the extent of the problem, the frequency allocation problem for small areas becomes impossible to solve mathematically.

Here come CR technology and Dynamic Spectrum Access. This technology allows for detecting unused “white spaces” in the RF environment and using them opportunistically by secondary users. Thus, they greatly relieve the pressure on the ISM frequency band. However, the initial DSA algorithm design assumed the presence of wide cell networks with powerful base stations. Therefore, their mathematical model assumes unlimited energy, very sensitive wideband antennas, and considerable computational power on the backend side.

Adapting macro-technologies for edge-level networks brings many constraints. Microcontrollers in WSNs operate within strict energy limitations. Our sensor nodes have no dedicated control channels. When a collision hits they guess blindly until they find each other again there is no negotiation. Routing decisions through a backend server or waiting on TCP/IP adds hundreds of milliseconds of delay, which is completely unworkable for real-time frequency evasion. The only practical solution is connectionless MAC-layer coordination running below the network stack, keeping phase-lock times in the microsecond range.

There is another threat emerging aside from network congestion. Congestion is only part of the problem. The ISM band is actively hunted by eavesdroppers, packet forgers, and jammers and the predictable transmission schedules of standard IoT protocols make their job straightforward. Defending the network means producing channel hops that look random from the outside, sequences no fixed prediction model can reliably track. This paper traces the shift from centralised threshold-based CR designs toward ML inference running directly on edge hardware.The hardware limits power budgets and connectionless protocol decisions that shape anti jamming design are all covered below.

II. ARCHITECTURE

The current edge-centric paradigm is characterized by the physical disjunction of sensing, computing, and actuation functions. Centralised CRN designs fail on latency backhaul round trips add overhead that compounds fast in a congested band. Microsecond-range frequency evasion is simply not achievable if every decision requires a server trip. Working through the existing DSA literature makes this clear, and consistently points toward a three-tier decentralised hardware structure as the workable solution for IoT-scale edge deployments.

At the bottom layer, multiple distributed nodes handle spectrum sensing. Using low power microcontrollers, such as the ESP32-based ones, in tandem with narrowband transceivers, these slave devices constantly monitor the ISM spectrum. Each node tracks RSSI values and ambient noise floor nothing more.Collecting full I/Q samples was deliberately ruled out: at 80 MHz with no cooling, processing I/Q streams causes immediate ADC saturation and thermal problems. These nodes stay passive by design.

Raw telemetry from the sensor layer feeds up to the second tier the Localised Computation Core.Specialized microprocessors, namely the Raspberry Pi devices, consume the received analog signals. The computation of deep neural networks leads to heavy thermal throttling, insufficient memory resources, and long inference times. The computation core runs lightweight supervised ML instead. In particular, ensemble methods including decision trees, e.g., the Random Forest algorithm, classify the ambient noise from deliberate jamming. The ESP32 sensor nodes never touch the heavy computation. It all stays at the gateway.

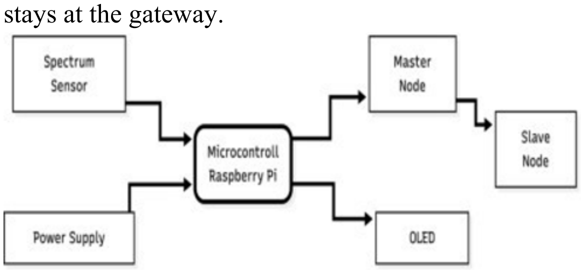

Fig.-1 Block diagram of Dynamic Spectrum Access and Autonomous Channel Allocation in Wireless Sensor Networks

Finally, control over actuation is provided by the third layer. It makes no sense identifying a free-frequency slot unless there are means to coordinate the action of a distributed system of nodes. Regular IEEE 802.11 association protocols prove unreliable while performing dynamic spectrum access as TCP/ IP connections incur a significant delay and total packet loss. The current architecture avoids this bottleneck. Following mathematical proof of the presence of the white space, the computational core commands slaves to hop to the new channel using a connectionless MAC protocol. It means that all sensors switch frequencies at once below the network level with predefined MAC addresses. This protocol architecture allows designing asynchronous localized IoT networks with fully autonomous Dynamic Spectrum Access.

III. CHARACTERISTICS

the edge-cognitive DSA framework possesses several key attributes suited to operations in the overcrowded 2.4 GHz environment.it shows outstanding speed thanks to utilizing connectionless MAC-layer protocols like ESP-NOW, effectively cutting out network-layer latency associated with establishing an association. Such a capability is vital for maintaining microseconds-long phase-locks of frequencies while jammed and allowing secondary nodes to avoid jammers without waiting for a TCP/IP handshake. The system is also designed for satisfying strict thermal and energy requirements. Namely, it specifically constrains a computational engine - a Raspberry Pi 4 gateway to using light-weight Random Forest decision trees instead of heavy deep neural networks.

It helps in reducing heat in processors and makes it possible for real-time machine learning to take place on edge devices that are not cooled down. the proposed mathematical model is designed in such a way that it takes into account affordable and deteriorating edge hardware. Instead of purchasing expensive hardware, we use algorithms that boost signal strength and eliminate phase noise, I/Q imbalance, and Nakagami-m fading caused by sub-$5 CC2500 radios.

the designed system is completely decentralized in order to ensure that there is room for scalability. There is no need to send large amounts of data to a central server to facilitate training; rather, we use the Fed-MADRL mechanism within local nodes in order to compress data and cope with changes in spectrum.

IV. APPLICATION

By resolving these edge bottlenecks, our autonomous DSA setup provides significant improvements to the usefulness of the system for systems that cannot tolerate losses of packets in the congested 2.4GHz band.

A. IIoT:

Manufacturing settings usually suffer from heavy Wi-Fi interference. This system guarantees uninterrupted functioning of the machines in factories by changing to other, less polluted frequencies autonomously and thus keeps the telemetric data of machines intact.

B. Smart Grid Architecture

City-based power distribution infrastructures are unable to rely on static frequencies due to high spatial congestion in this environment. This architecture supports uninterrupted connection of sensors on the grid despite heavy interference, which helps to avoid a loss of important data before blackout.

C. Telemetry in Hospitals

In hospitals, the number of users and spectrum usage is high. This setup prevents potential losses of telemetric data from patient monitors. Upon saturating of the channel, the medics' devices change the operating frequencies to avoid losses of patient data.

D. UAVs and Drones

Unmanned aerial vehicles operate in conditions of constant signal degradation caused either by noise or jamming attempts. The solution protects communication channels of the drones and video stream with the help of changing frequency of operation in microseconds.

E. Home Automation Network

Home automation networks have a lot of sensors located inside densely packed Wi-Fi areas. Using a connectionless MAC layer protocol allows the devices to bypass network congestion actively, reducing cases of sensor disconnections when routers become overloaded.

V. CHALLENGES

A. Processor Heating and Latency:

Running Random Forest continuously on live 2.4 GHz telemetry pushes the Raspberry Pi 4 Model B much harder than most expect from a lightweight classifier. Without a cooling fan the board hits thermal limits fast and throttles its own clock speed to protect itself. That slowdown cascades directly into prediction latency when the evasion window is microseconds wide, even a 200ms delay means the hop opportunity is already gone.

B. Cost-Effective Hardware Interference and Distortion:

While theoretical sensing models work well, real-world hardware limitations could easily break them.Sub-$5 transceivers like the CC2500 produce consistent phase noise, thermal drift, and I/Q imbalance that are simply part of the hardware at that price point. Our classifier has to tell those distortion signatures apart from real interference. If it gets confused, every hardware artifact triggers a false hop and the network destabilises from its own noise rather than actual jamming.

C. Nodes Synchronization Without A Shared Clock

ESP-NOW removes Wi-Fi association delays cleanly, but keeping multiple ESP32 boards locked to the same timing without a master clock is genuinely difficult. One dropped hop command during a fast channel switch leaves that node stranded on the old frequency while the rest of the network moves on. Detecting the missing node, locating it, and pulling it back in without disrupting the running network required a dedicated recovery protocol

D. Adaptive Smart Jamming

The 2.4 GHz band now has adversarial devices running their own ML models to map a network's hopping pattern and predict the next channel before the switch happens. Staying ahead of this means our hop sequences have to be genuinely statistically chaotic not just pseudo-random, but hard to extract patterns from even with minutes of observation data. Producing that level of unpredictability within a Raspberry Pi's power and compute budget is a real design constraint, not a theoretical one

E. Energy Scavenging Deficits

Continuous spectrum sweeping drains battery-dependent IoT nodes exponentially faster than standard transmission cycles. While scavenging ambient radio frequency (RF) energy to recharge edge devices is theoretically viable, current RF-to-DC energy conversion efficiencies remain insufficient to sustain the active computational load required for continuous machine learning inference.

VI. RELATED WORK

1. S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications." In this seminal work, the author formally introduced the idea of cognitive radio, providing the conceptual foundation for dynamic spectrum access (DSA). The article proves how secondary transceivers can perform computational processing of their RF environment to take advantage of the available bandwidth gaps between primary transmitters and receivers.
2. T. Yücek and H. Arslan, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications." This review paper outlines all traditional sensing algorithms, including energy and cyclostationary feature detections. The authors provide the mathematical proof of why raw thresholding is insufficient in low signal-to-noise ratios (SNR), emphasizing the need for enhanced algorithms in modern cognitive radios.
3. A. Khan, A. Z. Shaikh, S. Naqvi, and T. Altaf, "Implementation of Cooperative Spectrum Sensing Algorithm using Raspberry Pi." This practical study proves the viability of DSA on an edge-computing device, demonstrating how microprocessors like Raspberry Pi may perform real-time ingestion of telemetry to enable cooperative sensing. Simultaneously, the authors identify the heat generation and latency problems of localized processing.
4. S. Zareei, A. K. M. N. Islam, C. Vargas-Rosales, and M. Mansoor, "MAC Protocols for Cognitive Radio Sensor Networks: A Comprehensive Survey." The work provides a detailed analysis of the latency issues related to CSMA in cognitive systems. In the conclusion, the authors claim that master-slave connectionless MAC protocols are vital for frequency hop coordination by constrained cognitive sensor nodes.
5. W. Ejaz and M. Ibnkahla, "Multiband Spectrum Sensing and Resource Allocation for IoT in Cognitive 5G Networks." In response to Internet of Things (IoT) network congestion problems, the authors develop a mechanism for resource allocation by balancing the energy consumed by spectrum sensing and data transmission in congested environments.
6. Wang, S.; Liu, H.; Gomes, P. H.; Krishnamachari, B. Deep Reinforcement Learning Dynamic Spectrum Access in Cognitive Radio Networks. This is an analysis of self-learning based channel allocation methods. The work uses a markov decision process to demonstrate that cognitive secondary users can use learning capabilities to discover the most effective channel selection rules.
7. Erpek, T.; Sagduyu, Y. E.; Shi, Y. Deep Learning for Launching and Mitigating Wireless Jamming Attacks. In this work, the adversarial processes in the 2.4 GHz wireless environment are explained. The authors describe how deep learning can be used by jammers for predicting the channel hopping process in secondary users and how cognitive radios need nonlinear escape strategies for mitigating the advanced attacks in the physical layer.
8. Eddin, A. S.; Arabi, M.; Obeid, A. Machine-learning-based spectrum sensing enhancement for software-defined radio applications. This research addresses hardware limitations in non-optimal transceivers. The researchers show that machine learning techniques applied to Software Defined Radios can reduce the impact of phase noise and IQ imbalance in wideband frequency spectrum sensing operations.
9. Guimarães, D. A.; et al. Design Guidelines for Database-Driven Internet of Things-Enabled Dynamic Spectrum Access. This paper is a discussion of the shift in architecture toward IoT spectrum management decentralization. Stringent requirements for edge-driven spectrum management are given and it is shown that telemetry data stored in local DBs are able to react faster to any interference than cloud data stores.
10. Singh, R. K.; Singh, A. K. Optimizing Spectrum Sensing in Cognitive Radio Networks Using Bayesian-Optimized Random Forest Classifier. This new research proves the effectiveness of simple ensemble methods in spectrum intelligence. It is shown that if properly optimized, the Random Forest classification algorithm ensures high accuracy with low inference latency and, hence, outperforms complex neural networks in thermally limited edge processors.

VII. SYSTEM WORKFLOWAND TESTING PLAN

Step A: Spectrum Scanning (Observation)

Firstly, the system should be able to determine the state of the electromagnetic environment by scanning the frequencies. HackRF One or CC2500 can be used to monitor the 2.4 GHz Wi-Fi spectrum. At this step, the sensors will function only as observers, gathering information related to such parameters as RSSI and ambient noise. When the channel is considered empty, the process continues at the same point.

Step B:Signal Classification(Decision)

sensors send their findings to the Raspberry Pi. To reduce the load from running heavy neural networks, a relatively fast decision-making algorithm called Random Forest written in Python (Scikit-Learn) may be applied. It is supposed to be capable of separating normal hardware noises from real jamming attacks. In case there is an interference, the process goes to the next stage.

Step C: Channel Hopping (Action)

After recognizing congestion or jamming on the current channel via the Raspberry Pi, the latter chooses an unused channel for the whole network. After that, the signal instructing the ESP32 microcontrollers to switch is sent by the Raspberry Pi. In order to increase speed, the ESP-NOW communication protocol is used instead of regular Wi-Fi, allowing ESP32 to instantly jump to the new frequency without any disconnection from each other. When that is accomplished, the process comes back to Step A.

VIII. CONCLUSION

At this time, congestion of the ISM band is such that static assignment of frequencies has become unreliable for use. With the proliferation of IoT devices, Wi-Fi technology struggles with interference and susceptibility to deliberate jamming attacks. This paper details an innovative, decentralized dynamic spectrum access approach aimed specifically at addressing this serious problem.

Firstly, using a Raspberry Pi 4 as the local processing device prevents the major latency involved in sending spectrum information to cloud servers. Second, using a lightweight Random Forest machine learning model allows the network to determine whether there is normal ambient interference or if active jamming is taking place, without creating thermal issues for the microcontroller. Finally, replacing standard Wi-Fi association protocols with ESP-NOW MAC layer connections allows sensor modules like the ESP32 to synchronize and change frequencies within microseconds.

Ultimately, combining software-defined spectrum scanning with local computation and MAC layer coordination showcases the capabilities of hardware-aided autonomy for spectrum allocation. This novel DSA architecture creates a robust, scalable communications network that will be able to resist wireless interference, showing strong potential application in future Smart Grid and IIoT deployments.

REFERENCES

[1] Suriya, M. "Overview of Spectrum Sharing and Dynamic Spectrum Allocation schemes in Cognitive Radio Networks". 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT). 2022, pp. 1-6.

[2] Jasim, D. K. "Cognitive Radio Network: Security and Reliability trade-off - Status, Challenges, and Future trend". 2021 International Conference on Advanced Computer Applications (ICACA). 2021, pp. 112-118.

[3] Ahmad, W. S. H. M. W., et al. "5G Technology: Towards Dynamic Spectrum Sharing Using Cognitive Radio Networks". IEEE Access. Vol. 8, pp. 14460-14488. 2020.

[4] Abdi, Y., and T. Ristaniemi. "Random Interruptions in Cooperation for Spectrum Sensing in Cognitive Radio Networks". 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). Montreal, QC, Canada, 2017,pp. 1-7.

[5] Elkashlan, M., et al. "On the security of cognitive radio networks". IEEE Transactions on Vehicular Technology. Vol. 64, Issue 8, pp. 3790-3795. 2014.Communications. Vol. 23, Issue 2, pp. 201-220, Feb. 2005.

[6] Yücek, T., and H. Arslan. "A survey of spectrum sensing algorithms for cognitive radio

applications". IEEE Communications Surveys & Tutorials. Vol. 11, Issue 1, pp. 116-130, First Quarter

[7] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," Computer Networks, vol. 50, no. 13, pp. 2127-2159, 2006.

[8] J. M. Peha, "Sharing Spectrum Through Spectrum Policy Reform and Cognitive Radio," Proceedings of the IEEE, vol. 97, no. 4, pp. 708-719, April 2009.

[9] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking Spectrum Gridlock With Cognitive Radios: An Information Theoretic Perspective," Proceedings of the IEEE, vol. 97, no. 5, pp. 894-914, May 2009.

[10] A. Ghasemi and E. S. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," IEEE Communications Magazine, vol. 46, no. 4, pp. 32-39, April 2008.

[11] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004, pp. 772-776.

[12] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," IEEE Signal Processing Magazine, vol. 24, no. 3, pp. 79-89, May 2007.

[13] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," IEEE Personal Communications, vol. 6, no. 4, pp. 13-18, Aug. 1999.

[14] 14.Y.-C. Liang, Y. Zeng, E. C. Y. Peh, and A. T. Hoang, "Sensing-Throughput Tradeoff for Cognitive Radio Networks," IEEE Transactions on Wireless Communications, vol. 7, no. 4,pp. 1326-1337, April 2008.

[15] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum Sensing for Cognitive Radio: State-of-the-Art and Recent Advances," IEEE Signal Processing Magazine, vol. 29, no. 3, pp. 101-116, May 2012.

[16] Y. Zeng and Y.-C. Liang, "Eigenvalue-based spectrum sensingalgorithms for cognitive radio," IEEE Transactions on Communications, vol. 57, no. 6, pp. 1784-1793, June 2009.R. K. Singh and A. K. Singh, "Optimizing Spectrum Sensing in Cognitive Radio Networks Using Bayesian-Optimized Random Forest Classifier," International Journal of Intelligent Engineering and Systems, vol. 16, no. 6, 2023.

[17] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks," IEEE Journal on Selected Areas in Communications, vol. 31, no. 11, pp. 2209-2221, Nov. 2013.

[18] K. B. Letaief and W. Zhang, "Cooperative Communications for Cognitive Radio Networks," Proceedings of the IEEE, vol. 97, no. 5, pp. 878-893, May 2009.

[19] A. Ali and W. Hamouda, "Advances on Spectrum Sensing for Cognitive Radio Networks: Theory and Applications," IEEE Communications Surveys & Tutorials, vol. 19, no. 2,pp. 1277-1304, Second Quarter 2017.

[20] L. Lu, X. Zhou, U. Onunkwo, and G. Y. Li, "Ten years of research in spectrum sensing and sharing in cognitive radio," EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 28, 2012.

[21] B. Wang and K. J. R. Liu, "Advances in cognitive radio networks: A survey," IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 1, pp. 5-23, Feb. 2011.

[22] A. Khan, A. Z. Shaikh, S. Naqvi, and T. Altaf, "Implementation of cooperative spectrum sensing algorithm using Raspberry Pi," 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), Karachi, Pakistan, 2017, pp. 1-5.

[23] S. Zareei, A. K. M. N. Islam, C. Vargas-Rosales, M. Mansoor, S. Barakabitze, and L. Azpilicueta, "MAC Protocols for Cognitive Radio Sensor Networks: A Comprehensive Survey," Sensors, vol. 17, no. 9, p. 2136, 2017.

[24] C. Cormio and K. R. Chowdhury, "A survey on MAC protocols for cognitive radio networks," Ad Hoc Networks, vol. 7, no. 7, pp. 1315-1329, 2009.

[25] D. A. Guimarães et al., "Design Guidelines for Database-Driven Internet of Things-Enabled Dynamic Spectrum Access," IEEE Access, vol. 8,

pp. 147587-147602, 2020.