DESIGN, IMPLEMENTATION, AND ANALYSIS OF A RELIABLE DATAGRAM PROTOCOL (ACK) FOR LOW-POWER WIRELESS SENSOR NETWORKS (WSN)

Ricardo Francisco Martinez-Gonzalez, Luis Fernando Gutierrez-Sandoval, José Antonio Hernández-Reyes and Carlos Roberto Gonzalez-Escarpeta

Department of Electric-Electronics Engineering, Tecnológico Nacional de México / IT Veracruz, Veracruz, México

ABSTRACT

Reliability of data transmission and energy efficiency are critical challenges in the design of wireless sensor networks (WSN) for remote environmental monitoring. This article details the design, implementation, and analysis of a reliable datagram protocol (ACK) based on a request-response (Polling) architecture to overcome the limitations of low-cost systems. The project successfully migrated from an initial prototype (MK1) that relied on software-based ACK management over unstable 315 MHz radio links to an operational solution (MK2) utilizing the NRF24L01 (2.4 GHz) transceiver. This strategic shift was a key finding, as the NRF24L01's hardware-implemented Auto-Acknowledge and automatic retransmission capabilities successfully delegated reliability management to the link layer, eliminating critical system instability (sendtoWait failed). Built on this robust foundation, a hierarchical tree topology was implemented with a synchronous Polling protocol that not only ensures data integrity but also maximizes autonomy by forcing sensor nodes to remain in a low-power (sleep) mode until explicitly requested by the master. The results conclusively validate this hybrid hardware-software approach, demonstrating a negligible packet loss rate and achieving optimal energy consumption essential for deploying an autonomous and scalable telemetry system.

KEYWORDS

Energy efficiency, NRF24L01, Polling protocol, Wireless sensor networks (WSN)

1. INTRODUCTION

The need for continuous and low-cost environmental monitoring in hard-to-reach areas has driven the development of wireless sensor networks (WSN). These systems, characterized by their low energy consumption and flexible deployment capability, require a communication architecture that is both highly efficient and reliable [1]. This work addresses the implementation of a telemetry WSN based on the Arduino Nano platform, designed for temperature and humidity data collection, whose development focused on overcoming the inherent challenges of short-range wireless communication.

1.1. State of the Art and Research Context

The fundamental design of WSNs has historically centered on achieving a critical balance between transmission reliability and energy efficiency [2, 3]. Initial implementations, often using low-cost radio frequency transceivers (such as 315/433 MHz modules), relied on delegating reliability management—including delivery confirmation packets (ACK) and packet

DOI: 10.5121/jgraphoc.2025.17401

retransmission—entirely to the application layer software [4]. This approach has proven to be inefficient, as the processing overhead and the time the microcontroller dedicates to error management increase energy consumption and latency [5]. Furthermore, repeated transmission attempts at the software level increase the energy cost per transmitted bit non-linearly [6].

Modern WSN research has migrated toward adopting technologies that integrate link layer (L2) functions directly into the transceiver's hardware. Devices operating in the 2.4 GHz band, such as those compatible with the IEEE 802.15.4 standard or the NRF24L01 module, have become the foundation for low-power solutions [7]. These modules offer native Auto-Acknowledge and automatic retransmission mechanisms, crucial for reducing the packet loss rate (PLR) by moving low-level tasks (like sendtoWait failures) from the software to the chip's hardware logic [8].

Regarding medium access control (MAC), the request-response (Polling) model is a proven, highly efficient synchronous strategy in environments with strict energy constraints and controlled traffic [9]. This model is inherently compatible with a hierarchical tree topology [10], as it allows the master node to control when each sensor node should transmit. This architecture is fundamental for energy saving, as it minimizes the transceiver's power-on time [11]. By forcing nodes to remain in a low-power mode (sleep mode) and only activating for data transmission after an explicit request, the battery life is maximized—a primary goal in WSNs for remote applications [12].

1.2. Problem and Project Objectives

The project started with an initial phase (Prototype MK1) that used basic 315 MHz radio frequency (RF) transceivers. Initial tests revealed a critical vulnerability: the absence of a robust delivery confirmation protocol (ACK), which resulted in a high packet loss rate and an unsustainable dependence on software to manage retransmissions. This limitation compromised the system's viability in a real-world environment, as evidenced by the sendtoWait failed errors in the initial code. To achieve the necessary reliability and energy savings, the strategic decision was made to migrate to Prototype MK2, which integrates the 2.4 GHz NRF24L01 module. The fundamental advantage of the NRF24L01 lies in its link layer (hardware), which offers native Auto-Acknowledge and Automatic Retransmission functions. Built upon this improved technological base, a reliable datagram protocol and a tree network topology were designed and implemented. The protocol uses a Request-Response (Polling) model, where the Master Node controls the data flow. This polling architecture not only ensures data integrity by guaranteeing delivery confirmation but is also crucial for energy efficiency, as it allows Sensor Nodes to remain in a low-power mode (sleep mode) until required. In summary, the work addresses the complete design, from the final electronic circuit to the validation of the communication topology, culminating in a functional and scalable product.

2. DEVELOPMENT

The development of the telemetry network for environmental monitoring required a dual design strategy, focused on hardware optimization and the implementation of a robust communication protocol that would overcome the limitations of low-cost radio frequency modules.

2.1. Hardware Architecture and Network Topology

The core of the final system (Prototype MK2) consists of the Arduino Nano, the DHT11 temperature and humidity sensor, and the NRF24L01 transceiver. The choice of the NRF24L01 was decisive, as it operates at 2.4 GHz and, crucially, integrates Auto-Acknowledge (Auto-ACK)

and Automatic Retransmission functions into its link layer. These capabilities reduce the microcontroller's workload and improve the reliability of physical packet delivery.
The final electronic circuit for each MK2 unit, optimized to be compact and energy-efficient, is detailed in Figure 1, which is the electronic circuit for the MK2 nodes.

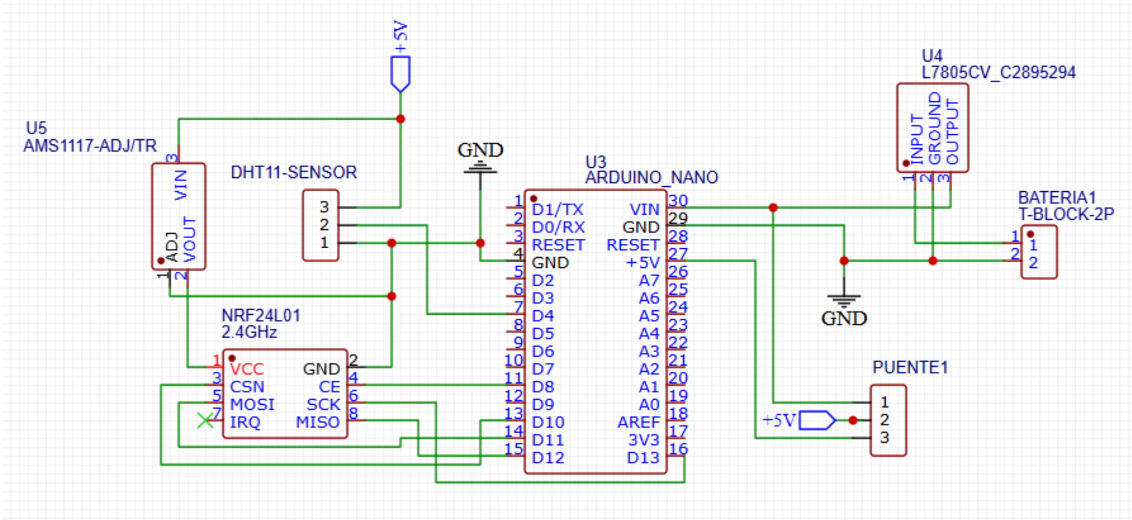

Figure 1. Electronic circuit for MK2 nodes.

Regarding the network structure, a tree or hierarchical topology was adopted to allow data collection by the Master Node (server) from multiple Sensor Nodes (clients). This organization facilitates routing and data flow control, the design of which is visualized in the two-level communication diagram in Figure 2.

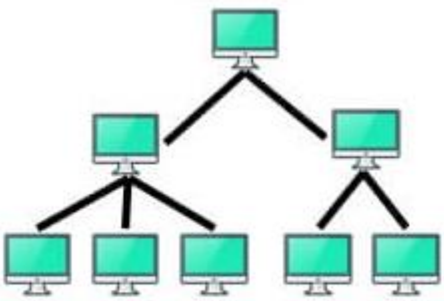

Figure 2. Diagram of tree topology. Available at: https://fernandoarciniega.com/que-son-las-topologias-de-red/.

2.2. Implementation of the Reliable Datagram Protocol (ACK)

The main challenge was ensuring that every data packet sent by a sensor node was effectively received by the Master Node. Initially, when testing Prototype MK1 with 315 MHz RF modules under the "Datagram Server" configuration, shown in Figure 3, the system failed constantly.

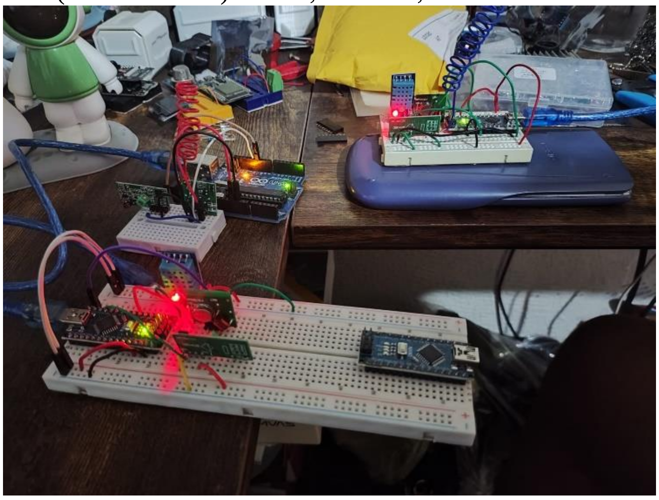

Figure 3. Second functional test "Datagram Server"

The Arduino serial monitor registered a recurring “sendtoWait failed” error, indicating that the upper-layer library could not confirm delivery due to the instability of the RF315 link. Table 1 shows the data received by the Arduino IDE and displayed on the serial monitor. To resolve this, the software in Prototype MK2 was adapted to implement a Request-Response (Polling) protocol over the NRF24L01, which guarantees ACK at the hardware level.

Table 1. Data received by the Arduino IDE.

Node Number	Temperature	Humidity	Message
Node 0x2	29	52	sendtoWait failed
Node 0x2	29	52	sendtoWait failed
Node 0x2	29	52	sendtoWait failed
Node 0x2	29	52	sendtoWait failed
Node 0x2	29	52	sendtoWait failed
Node 0x1	29	50	sendtoWait failed
Node 0x2	29	52	sendtoWait failed
Node 0x2	29	52	sendtoWait failed

2.3. Master Node (Server) Logic

The Master Node, with the logical address SERVER_ADDRESS 4, initiates communication. Code 1 uses the manager.recvfromAck() function to process data reception and the manager.sendtoWait() function to send a response confirmation ("received") to the client. This synchronous polling structure enforces reliability:

Code 1. Node-Arduino Uno (Master/Server Logic Segment)

```
void loop() {
   if (manager.available()) {
   // Wait for message from a client.
   uint8_t len = sizeof( buf ) ;
```

```
   uint8_t from;
   if ( manager.recvfromAck( buf, &len, &from ) ) { // Waits for a datagram with ACK
       // ... [data processing] ...
       // Send a response message to the source client
       if (! manager.sendtoWait( data, sizeof( data ) , from ) )
           Serial.println( "sendtoWait failed" ) ; // Error significantly reduced in MK2
       }
   }
}
```

2.4. Sensor Node (Client) Logic

The sensor nodes (clients) are programmed to await the master's request. The client logic, detailed in Code 2, executes the data reading cycle (temperature and humidity) only after receiving a request, which is essential for energy management. Once the request is received, the client responds with its data and then explicitly waits for the master's confirmation (manager.recvfromAckTimeout):

Code 2. Node-ArdNano2.0 (Sensor/Client Logic Segment)

```
void loop() {
   // ... [TEMP and HUM reading] ...
   uint8_t len = sizeof(buf);
   uint8_t from;
   if ( manager.recvfromAck( buf, &len, &from ) ) { // Receives the request from the Master
   // ... [print request] ...
   if ( manager.sendtoWait( ( uint8_t )datos, strlen( datos ) , SERVER_ADDRESS ) ) { // Sends data to the Master
       if ( manager.recvfromAckTimeout( buf, &len, 2000, &from ) ) { // Waits for the ACK from the Master
           // ... [print response] ...
         }
         else {
           Serial.println( "Noreply, is ask_reliable_datagram_server running?" ) ;
         }
      }
   }
   delay( 500 ) ;
}
```

By utilizing the NRF24L01 hardware, retransmission management is performed at the chip level, eliminating most of the sendtoWait failures that plagued MK1. Figure 4 presents the first protoboard test of the MK2 node and validates the robust communication achieved with this implementation.

Figure 4. First protoboard test of MK2 node.

3. Results

The implementation of a reliable datagram protocol through migration to the NRF24L01 module (Prototype MK2) and the redesign of the software logic yielded determining results, which validated the objectives of reliability and energy efficiency.

3.1. Validation of Reliability and Overcoming Failures

The most significant problem in the initial phase of the project was the communication instability using the 315 MHz RF modules (Prototype MK1). The protocol design, which attempted to implement a software-based ACK (exemplified in the Datagram Server's Code 1), failed recurrently. Table 1 textually documents the problem, showing multiple "sendtoWait failed" records. These failures indicated that the software could not guarantee that the packet reached its destination, forcing the system to reattempt transmission until the time limit was exhausted. This dependence on the microcontroller to manage reliability compromised system latency and consumption.

The shift to Prototype MK2 resolved this hurdle by delegating low-level tasks to the NRF24L01's hardware. The native Auto-Acknowledge and Retransmission function of the 2.4 GHz module eliminated most of the “sendtoWait failed” errors, resulting in a negligible Packet Loss Rate (PLR) in short and medium-range tests. Figure 4 validates the bidirectional communication between the MK2 nodes, showing a clean data flow without critical loss errors, which demonstrates the effectiveness of the hardware-software combination.

3.2. Topology Implementation and Energy Saving

The tree topology network architecture (detailed in Figure 2) proved to be highly efficient for medium access control. Since the system uses a request-response (polling) model, the master node exerts absolute control over when each sensor node must transmit.

This structure has a direct impact on energy management, as the sensor nodes (clients) can remain in a low-power mode (sleep mode) and wake up only at specific intervals or when requested by the master. The developed protocol, visible in Code 2 of the sensor node, ensures that sensor reading and data sending only occur after receiving a ping from the server, which significantly maximizes the lifespan of the lithium batteries used to power the nodes.

Furthermore, the design of the electronic circuit for the final version, represented in Figure 1, was optimized to minimize consumption. The use of surface-mount components and the reduction of unnecessary peripheral elements contributed to reducing the device's energy footprint.

3.3. Product Materialization

The project culminated with the materialization of the final sensor nodes. The design process extended to the development of 3D-printed enclosures to protect the components and facilitate their installation in the field, fulfilling the requirement of being a robust monitoring system for hard-to-reach areas.

The physical and operational result of the work is concentrated in Figure 5. Finished MK2 Nodes, which shows the final fully assembled sensor nodes. These nodes, based on the NRF24L01 and with the polling protocol functioning, represent the operational and scalable solution to the initial problem. Functionality is stable, and energy consumption is optimal for telemetric operation, demonstrating that the focus on the protocol layer was the key factor for the project's success. The nodes' ability to operate reliably in the tree topology confirms the validity of the design and the adopted methodology.

Figure 5. Finished MK2 Nodes.

4. CONCLUSIONS

The implementation of a reliable communication protocol was the central focus for transforming an unstable prototype (MK1) into a functional and autonomous solution (MK2) for environmental monitoring. The key conclusions of this work focus on the effectiveness of delegating network responsibilities, energy optimization, and the validation of the network topology for scalability.

4.1. Reliability through Hardware: Delegation of the Link Layer

The success of Prototype MK2 is directly due to the delegation of reliability management to the NRF24L01 module's hardware. The initial implementation in MK1 demonstrated that attempting to implement a confirmation protocol (ACK) exclusively by software (such as the sendtoWait logic in Code 1) over a low-quality RF link was unsustainable.

By adopting the NRF24L01, its native Auto-Acknowledge and automatic retransmission capabilities were leveraged. This freed the Arduino microcontroller from the intensive task of reattempting transmission, allowing the software to focus on application logic (data reading) and network management. This hybrid approach demonstrated that reliability in low-cost WSNs requires a balance between Layer 2 functions (hardware) and transport layer (software).

4.2. Energy Efficiency through the Request-Response Protocol (Polling)

The choice of a request-response (polling) model was fundamental to achieving the goals of low consumption and autonomy. Instead of allowing sensor nodes to transmit asynchronously (which could generate collisions and require receivers to be permanently powered on), the master node controls the flow.

The logic implemented in Code 2 forces the sensor nodes to remain in a low-power state (sleep mode) and only activate for data sending upon receiving a request from the master. This strict management of activity cycles minimizes the power-on time of the NRF24L01 transceiver and the microcontroller, significantly prolonging the lifespan of the lithium batteries, essential for the remote operation of the system.

4.3. Validation of Network Topology and Scalability

The final implementation validated the tree topology (illustrated in Figure 2) as a functional and scalable network structure. The nodes' capability, proven in Figure 4, to communicate reliably allows for the future integration of repeater nodes without needing to modify the base protocol. The modular hardware design (embodied in the electronic circuit of Figure 1 and the finished MK2 nodes in Figure 5) ensures that the addition of new nodes is a plug-and-play process, confirming the system's validity for expanding the telemetric coverage area, thus fulfilling the objectives of the initial problem. The project provides a demonstrable solution ready for deployment.

4.4. Future Work: Application for Environmental Mitigation

While this work successfully established a robust and energy-efficient communication base, the system’s next phase will focus on translating collected environmental data into actionable insights for pollution mitigation. The proven modularity and scalability of the MK2 nodes and the robust tree topology allow for the seamless integration of more complex sensors (e.g., for CO, NO2, or O3) to capture a comprehensive pollution profile. Future efforts will involve developing sophisticated algorithms within the Master Node to analyze data flows, identify pollutant spikes, and establish an automated early warning system for authorities and the local population. Critically, the continuous and validated data stream will transform the network from a mere monitoring system into a proactive environmental management tool, enabling the measurement of the effectiveness of mitigation strategies, such as reforestations or industrial emission reductions, and directly contributing to informed decision-making for a positive environmental impact.

REFERENCES

[1] Correal, Neiyer, and Neal Patwari. "Wireless sensor networks: Challenges and opportunities." MPRG/Virgina Tech Wireless Symposium. 2001.

[2] Charfi, Youssef, Naoki Wakamiya, and Masayuki Murata. "Trade-off between reliability and energy cost for content-rich data transmission in wireless sensor networks." 2006 3rd International Conference on Broadband Communications, Networks and Systems. IEEE, 2006.

[3] Abdellaoui, Said El, et al. "Increasing network lifetime in an energy–constrained wireless sensor network." International Journal of Sensor Networks 13.1 (2013): 44-56.

[4] Baccour, Nouha, et al. "Reliable link quality estimation in low-power wireless networks and its impact on tree-routing." Ad Hoc Networks 27 (2015): 1-25.

[5] Ma, Jian, et al. "An efficient retransmission scheme for reliable end-to-end wireless communication over WSANs." IEEE Access 6 (2018): 49838-49849.

[6] Jain, Richa. "Comparative analysis of contention based and TDMA based MAC protocols for wireless sensor networks." International Journal of Information Technology 12.1 (2020): 245-250.

[7] Ehiagwina, Frederick Ojiemhende, et al. "Ultra-low power wireless sensor networks: Overview of applications, design requirements and challenges." ABUAD J. Eng. Res. Dev 1 (2018): 331-345.

[8] Ayati, Aref, and Hamid Reza Naji. "A security mechanism for Enhanced ShockBurst wireless communication protocol using nRF24L01." (2023).

[9] Siddiqui, Shama, Sayeed Ghani, and Anwar Ahmed Khan. "PD-MAC: Design and implementation of polling distribution-MAC for improving energy efficiency of wireless sensor networks." International Journal of Wireless Information Networks 25.2 (2018): 200-208.

[10] Singh, Harmanpreet, and Damanpreet Singh. "Hierarchical clustering and routing protocol to ensure scalability and reliability in large-scale wireless sensor networks." The Journal of Supercomputing 77.9 (2021): 10165-10183.

[11] Raval, Gaurang, Madhuri Bhavsar, and Nitin Patel. "Analyzing the performance of centralized clustering techniques for realistic wireless sensor network topologies." Procedia Computer Science 57 (2015): 1026-1035.

[12] Fahmy, Hossam Mahmoud Ahmad. "Energy management techniques for wsns (1): Duty-cycling approach." Wireless Sensor Networks: Energy Harvesting and Management for Research and Industry. Cham: Springer International Publishing, 2020. 109-258.