WiLEAD: Bounded-Latency Physical-Layer Leader Election

Bingwu Fang, Jonathan Oostvogels, Sam Michiels, Danny Hughes
DistriNet, KU Leuven, Leuven, Belgium
Email: {bingwu.fang, jonathan.oostvogels, sam.michiels, danny.hughes}@kuleuven.be

***Abstract*—Emerging cyber–physical systems such as drone swarms and vehicle platoons require leader election within milliseconds to maintain closed-loop stability under failures and mobility. Today, leader election is mostly implemented in the application layer, necessitating multiple message exchanges with their attendant contention resolution, acknowledgments and retransmissions, all of which yields tens to hundreds of milliseconds of latency in wireless deployments. To bridge this gap, we present WiLEAD, a physical-layer (PHY) over-the-air leader election primitive that optimizes the different failure modes of a binary countdown mechanism to achieve fast and reliable leader election. WiLEAD is implemented on an ADALM-Pluto Software Defined Radio (SDR), achieving reliable elections within $5ms$ using only $100kHz$ of bandwidth; an order-of-magnitude improvement over the state of the art. Moreover, WiLEAD guarantees with high probability that completion times that grow sublinearly with the network size N, showing $\mathcal{O}(logN)$ completion times at the 90th percentile latency bound, compared to $\mathcal{O}(N)$ for conventional designs.**

***Index Terms*—real-time distributed systems, wireless networks, leader election, low latency**

I. Introduction

Real-time cyber-physical networks such as unmanned aerial vehicle (UAV) swarms, vehicular networks enabling cooperative driving (e.g., platooning), and other ad-hoc wireless deployments often require leader election to designate a coordinator for time-critical coordination primitives, including cooperative sensing and data aggregation [1] as well as group synchronization and schedule alignment [2]. The timeliness and scalability of leader election are critical to ensure seamless operation. For instance, in UAV swarms deployed in ad-hoc wireless environments (e.g., disaster response), highly dynamic channel conditions can render the current leader temporarily unreachable, necessitating rapid in-network re-election to preserve operational continuity and safety margins [3].

In current wireless systems, leader election is usually implemented in the application layer, building on a stack of transport, network and link layer primitives. Progress therefore depends on frame exchanges, random backoff, and network wide synchronization [4]–[7], which yields tens to hundreds of milliseconds of latency in wireless deployments. Even when only a small amount of information is needed to break symmetry, the election time is dominated by frame-level procedures such as preambles, inter-frame spacing (IFS), carrier sensing, and contention backoff, whose delays accumulate and fluctuate with contention and channel conditions. For example, the Arbitrating Value Transfer Protocol (AVTP) requires network-wide synchronization, which takes 4.2 ms to converge before leader election can proceed [4]. While theoretical analyses demonstrate the feasibility of leader election schemes that scale logarithmically with the number of participating nodes N [8], [9], such asymptotic results do not account for the synchronization and frame-level overheads of practical radios. Consequently, even protocols with efficient election logic can still incur end-to-end election latencies on the order of tens to hundreds of milliseconds in wireless deployments [4].

This paper presents WiLEAD, a physical-layer primitive that achieves millisecond-level leader election by implementing binary countdown within the wireless physical layer through symbol-level interaction. On a Software Defined Radio platform, the WiLEAD prototype completes reliable elections within $5\,\mathrm{ms}$ using only $100\,\mathrm{kHz}$ of bandwidth, enabling rapid re-election under noise while keeping spectrum use low. While legacy designs rely on frame-level interactions, which incur the overhead of preambles and Inter-Frame Spacing (IFS), WiLEAD embeds the election logic directly into the PHY layer. We demonstrate that leader election can thus with high probability be achieved within tight latency bounds. We argue that this model is appropriate for infrastructure-less cyber-physical systems such as UAV swarms, where missing a fail-over deadline can immediately degrade safety margins [3].

Our approach departs from prior work via three architectural innovations: (i) WiLEAD shifts the fundamental unit of leader election from frame exchanges to PHY symbol-level interactions, eliminating accumulated overheads such as preambles, headers, and inter-frame spacing. (ii) WiLEAD enables rapid leader election while eliminating additional synchronization overhead, removing the need for explicit network-wide synchronization phases during re-election. (iii) WiLEAD provides reliable network-wide agreement on a unique leader via wireless physical layer symbol-level interaction. The scientific contributions of this paper are as follows.

- We design and implement a low-latency PHY symbol-level leader election protocol that operates on half-duplex radios without higher-layer frame exchanges or external synchronization.
- We provide a configuration framework that translates a hard deadline and target reliability into PHY parameters for binary-countdown-style leader election on Software

ISBN 978-3-903176-82-9

Defined Radio (SDR), delivering low bandwidth consumption and interference robustness.
- To promote research in this area, we provide a validated SDR stack with adaptive channel sensing and thresholding. Open-source artifacts[1] have been made available online to ensure reproducibility.

The rest of this paper is organized as follows. Section 2 provides background and related works on leader election. Section 3 presents our core designs for WiLEAD. Section 4 describes the prototype implementation, including bare-metal signal processing and the system architecture. Section 5 details the experimental evaluation. Section 6 offers a discussion. Section 7 concludes the paper.

II. Background and Related Work

Leader election establishes a single coordinator to manage shared resources and maintain time-critical consistency. In real-time cyber-physical networks, such as UAV swarms [3] and vehicular platoons [10], coordinator failure be resolved within a bounded duration, typically on the order of milliseconds, to preserve safety margins and seamless operation. These strict latency constraints necessitate election mechanisms that provide a bounded completion time together with a high-probability guarantee of electing a unique leader.

In wireless systems, reliable message exchanges – upon which conventional election protocols build – are expensive: completions time are considerably affected by MAC/PHY effects such as preambles, inter-frame spacing, carrier sensing, contention backoff, and retransmissions. Consequently, the classical round/RTT message-exchange abstraction [11]–[14] is not predictive of end-to-end election latency in real-world wireless environments, where interference and time-varying connectivity further amplify delay variability and can push election latency into the tens to hundreds of milliseconds. Therefore, providing millisecond-scale leader election guarantees remains difficult without additional support services, e.g., accurate network-wide time synchronization [3]. For instance, Raft recommends conservative election timeouts of 150–$300\,\mathrm{ms}$ [14].

A natural way to avoid the stochastic latency and overheads of reliable messages exchanges is to adopt dominance arbitration, exemplified by the wired CAN bus [15], where dominant bits overwrite recessive ones and the system resolves contention without destroying the transmission. However, prior wireless adaptations largely map arbitration bits to data-link or network layer time slots or entire frames. For instance, WiDom [6] and AVTP [4] implement binary countdown by assigning priorities to time slots, but this necessitates network-wide slot synchronization enforced by a MAC schedule. BitMAC [5] relies on an OR-channel abstraction that demands precise start-of-slot alignment via hardware timers, making correctness dependent on tight platform timing. WiFi-BA [7] modifies 802.11 drivers to perform arbitration on OFDM subcarriers, but remains tied to specific hardware constraints and frame structures. These approaches effectively operate at frame granularity, inheriting per-frame overheads (e.g., preambles/headers), synchronization requirements, and transceiver turnaround times, which lead to tens to hundreds of milliseconds of latency for reliable leader election.

WiLEAD moves leader election to the PHY symbol stream and eliminates frame exchanges and reliance on network-wide time services found in the state of the art. The resulting election time scales sub-linearly with network size, consistent with $O(\log N)$ behavior within milliseconds.

III. System Design

A. Principle

The goal of WiLEAD is to elect a unique leader within a predictable deadline of at most several milliseconds. WiLEAD performs leader election directly at the physical layer using a pulse–silence comparison applied bit by bit over a short sequence.

Core idea. As shown in Fig. 1, to elect a leader, nodes concurrently transmit a per-election *random key* encoded as a fixed-length bit string from the most significant bit to the least significant bit. Each node chooses this key independently and uniformly at the start of the election. From any node's perspective, time is divided into fixed-length symbol windows. We use two symbols: a *dominant* symbol means the node transmits a narrowband pulse in that window, and a *recessive* symbol means the node remains silent. With envelope detection, if any contender sends a pulse in a window, listening nodes register an above-threshold envelope. Any node whose bit for that window is recessive and yet detects a pulse immediately drops out, while nodes with a dominant bit continue to the next window. This bit-by-bit elimination continues across the configured number of windows and, under independently randomized keys, with high probability leaves a single survivor as the leader. Random keys avoid bias toward particular identifiers and exercise the worst-case behavior of the leader election procedure. The chance of a tie becomes negligible when the key is long enough, and prior work on randomized leader election in wireless and radio-network models reports fast convergence with high probability [16], [17], consistent with our design.

Timing behavior. During normal operation, the coordinator emits a narrowband pilot tone that serves as an on-air heartbeat. Nodes that can reliably detect this pilot infer that the leader is alive and do not start an election. When the coordinator fails, the pilot tone ceases, for example due to a crash or power loss. Each node hence monitors the envelope on the pilot subband and applies the CFAR-based thresholding rule in Sec. III-D to detect *qualified falling edges*. When such an edge is detected, the node enters the guard-time phase (P2 in Fig. 1) and then starts bitwise arbitration (P3 in Fig. 1). This minimizes failure-detection latency, at the cost of a 100% transmit duty cycle at the incumbent leader. When the leader fails, the tone ceases and all nodes start a new round to elect a new leader. A short guard time before the first window absorbs skew in detecting leader failure so nodes align locally without

[1] https://gitlab.com/Etherly/WiLEAD

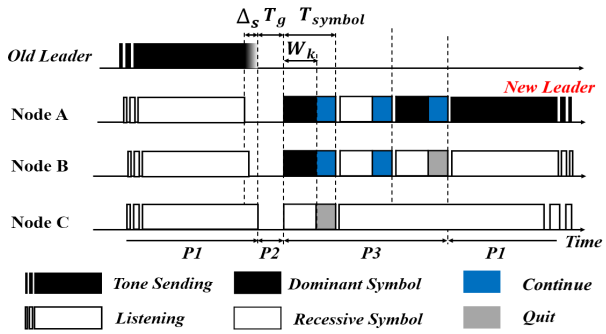

Fig. 1. The WiLEAD procedure: symbol-level arbitration for leader election.

external time services. Concretely, the pre-round guard is sized to cover the worst-case entry skew, caused by propagation delay spread across the network, latency spread due to analog and digital filtering, as well as quantization effects due to sampling and batched memory transfers. In our prototype, the guard is dominated almost entirely by implementation-level effects, while over-the-air propagation delays are negligible.

The election process advances through a pre-configured number of windows, each separated by a small guard time Radios operate in half duplex, which means a node either transmits or listens within a window. In band full duplex would allow a node to listen while sending on the same frequency [18], but COTS RF transceivers generally do not provide this capability and practical designs require strong self interference cancellation [19]. WiLEAD therefore achieves leader election using observations made by nodes that are silent in the current window. In energy-constrained deployments, the continuous-tone heartbeat can be replaced by a pulsed heartbeat with pulse width T_p and inter-pulse interval T_b, which reduces the leader duty cycle to T_p/T_b. Nodes then use the previously observed heartbeat cadence to predict the next expected heartbeat and trigger arbitration when no valid heartbeat is detected. This changes the entry skew timing analysis relative to the continuous-tone case. More precisely, the missed-heartbeat decision is now made over a finite slot rather than from an instantaneous falling edge, which means guard and symbol duration must now absorb additional uncertainty. We do not study this effect any further: for our prototype, hardware constraints impose a minimum symbol duration of $38\,\mu$s, where we do not yet observe the reliability effects of such additional uncertainty.

B. PHY Encoding and Decoding

WiLEAD operates within a narrow bandwidth of $100\,$kHz. We partition this band into a small pilot subband for the leader's continuous heartbeat tone and one subband to support the election process. Each node isolates the pilot subband with a narrow bandpass filter and applies a square-law detector on that band. A matched bandstop filter suppresses the pilot on the election subband so the pilot carries no payload and does not affect the signal seen by data detectors.

We now detail the signaling and detection path. During the dominant bit, a node transmit a constant-amplitude narrow-band tone, while during the recessive bit, it stays silent and keep listening. A silent node runs an envelope detector and compare the envelope's magnitude with an adaptive Constant False Alarm Rate (CFAR) threshold. Envelope detection is phase agnostic and works with un-synchronized carriers. Concretely, for the complex base-band $r[n] = I[n] + \mathrm{j}Q[n]$ signal, the decoder computes $A^2[n] = I[n]^2 + Q[n]^2$ and compares the result with an adaptive threshold η, described in Section III-D. This design requires no carrier or phase recovery. We disable Automatic Gain Control (AGC) and use manual gain control instead to eliminate gain-loop jitter and amplitude variance.

C. Bitwise Leader Election

To elect a new leader, each node first draws a per-election *random value* and encodes it as a fixed-length sequence of arbitration bits, wherein 1-bits indicate windows in which a node transmits a pulse, and 0-bits correspond to windows in which a node remains silent. When multiple contenders transmit a pulse in the same window, their transmissions superimpose and the envelope at a listening node remains detectable. Every silent node computes windowed envelope statistic and compares it with the adaptive threshold η from Sec. III-D. If a node holds a recessive bit and still detects a pulse it drops out, while nodes with a dominant bit advance to the next window. Bitwise elimination proceeds across the configured number of windows and with independently randomized keys, yielding a unique survivor with high probability. WiLead's round duration equals the number of arbitration windows multiplied by the per-window time, where the per-window time is the symbol duration plus guards. WiLEAD can allocate more bits by shortening the symbol window, or use fewer windows with a longer window to improve detection margins. Section V describes how we select the pair of parameters that meets the deadline while maximizing the probability of a unique winner. To maintain a target reliability as the number of contenders grows, the required deadline increases sublinearly, which enables rapid and stable re-election under mobility and failures.

D. Channel Sensing and Thresholding

Reliable triggering and decoding need a detector whose decisions stay stable as noise and interference change, without manual tuning. We use a per-symbol window Constant False Alarm Rate (CFAR) rule [20]. Silent nodes compute a windowed envelope statistic from the I/Q samples and compares it with an adaptive threshold η. We consider two failure modes in a symbol window. A *false alarm* occurs when the channel is idle yet the detector declares a pulse. In binary countdown this is harmful because a recessive node that falsely detects will drop out permanently, and different nodes can drop out in different windows, which can fragment survivors and create multiple tentative leaders. A *missed pulse* occurs when a dominant transmission is present yet the detector stays below the threshold. This is often recoverable as elimination can happen at a more significant bit later in the sequence or in a rerun, at the cost of extra latency. We therefore configure

a small per-window false alarm target α and accept a modest increase in miss probability at low SNR.

The threshold η is updated online in real time so that decisions remain stable across operating conditions and computation time per window remains constant. During brief intervals when the medium is idle, the receiver collects noise-only baseband samples to calibrate and updates the threshold. This calibration occurs during idle gaps when the pilot tone is absent. In particular, we use the maximum of the squared envelope over a block of samples as the statistic

$$m_t \triangleq \max_{n \in B_t} \{I[n]^2 + Q[n]^2\}, \qquad t = 1, \ldots, T \tag{1}$$

where B_t indexes samples in the t-th block and $I[n]$, $Q[n]$ are the in phase and quadrature samples. Working with the squared envelope is consistent with envelope detection because it is a monotone transform of amplitude and it keeps fixed point arithmetic simple. Processing block maxima rather than storing entire blocks reduces state to one scalar per block and still tracks slow drift in the noise floor while remaining sensitive to occasional spikes.

We instantiate CFAR as follows, building on the $\{m_t\}$ noise-only statistic. We set the decision threshold to a high quantile $q \in \{0.95, 0.99\}$ of $\{m_t\}$, which corresponds to a target false-alarm probability $P_{\mathrm{FA}} = 1 - q \in \{0.01,\, 0.05\}$. and update it online within constant memory using the P^2 estimator [21]. In parallel, we track a leaky envelope

$$M_t = \max\big\{\, m_t,\ \max(M_{t-1} - d, 0)\,\big\}, \qquad M_0 = 0, \tag{2}$$

where d controls per block decay. The envelope forgets isolated spikes after roughly $1/d$ blocks and follows gradual rises in the background. The detection threshold is a guarded quantile;

$$\eta = \min\Big\{\ \max\{\widehat{\xi}_q,\, \eta_{\min}\},\ k\, M_t\Big\}, \tag{3}$$

where $\widehat{\xi}_q$ is the online estimate of the q quantile of $\{m_t\}$, $\eta_{\min} > 0$ prevents collapse under deep fades, and k limits growth relative to the local envelope. We fix $k = 1.10$. The inner maximum maintains CFAR behavior and avoids underestimating the threshold during transient dips. The outer minimum caps overestimation when the quantile is inflated by short blocks or heavy tails, which reduces missed detections while keeping false alarms controlled. Under stationary noise the nominal per window false alarm probability is $P_{\mathrm{FA}} \approx 1-q$. We then choose q to keep P_{FA} near a small target α.

With this design, the detector performs a single integrate and compare per-symbol window and the CFAR estimator updates once per block with constant time and constant memory. This preserves predictable per-window computation and provides robustness in non Gaussian and non stationary environments [22].

IV. Implementation

A. Bare-metal signal processing

All timing-critical logic runs as bare-metal firmware on the Pluto SDR Zynq 7000 SoC with an ARM Cortex A9 processing system. The RF front end is an AD936x, and the baseband sample clock is the only time base. DMA streams interleaved I and Q samples into contiguous buffers in SDR memory. Under bare metal firmware, the control program runs directly on the processor without a general purpose operating system, thereby avoiding operating system scheduling jitter and yielding deterministic timing. Host side software does not participate in the timing loop.

All arithmetic is fixed point with saturation. Every time duration is expressed in sample counts. A hardware timer driven by the sample clock advances the window index so the scheduler and detector share the same clock domain. After processing a window, it writes the Boolean decision and the local timestamp to a small buffer, then increments the window index. This makes the per-window outcome explicit and prevents the next window from starting before the current decision is recorded.

Pulse generation. Dominant symbols are produced by gating the AD936x test-tone numerically controlled oscillator at baseband. Frequency and amplitude are programmed once; a digital enable toggles within the symbol window. Gating the numerically controlled oscillator (NCO) avoids phase lock loop (PLL) retunes and generates reproducible narrowband pulses, which keeps guard intervals short.

Signal detector. The detector operates directly on the interleaved I/Q buffers. For each window it computes a block-maximum of $A^2[n] = I[n]^2 + Q[n]^2$ and compares it to the adaptive CFAR threshold defined in Sec. III-D. The implementation uses vectorized fixed-point ARM NEON kernels. This keeps the timing loop predictable so the system can run in real time and process the data faster. Optional tracing can record per-window maxima for debugging without affecting the timing loop.

B. Experimental Infrastructure

The SDR alone executes symbol-level arbitration, and timestamps events in sample time. A companion BeagleBone Black handles configuration and log retrieval over UART. UART carries commands and logs only and is never on the critical path.

As shown in Fig. 2, a site manager on a Raspberry Pi coordinates experiments. Control uses request–reply with barriers for parameter changes. Measurements are published to an MQTT broker and optionally forwarded to a cloud database. All orchestration stays off the radio channel.

At the start, one SDR is instructed to emit a pilot tone as the initial leader. Others monitor the tone, detect the falling edge, and run symbol-level leader election. Each SDR records wins or losses and timestamps in device's local buffers, and the BeagleBone ships these records to the Cloud and the local computers asynchronously.

A round's latency depends only on SDR-side parameters such as window duration, sequence length, detector cost. Host or network delays affect when configuration takes effect and when logs appear, not the duration of the election process. All timestamps originate on the SDR in sample time, making them

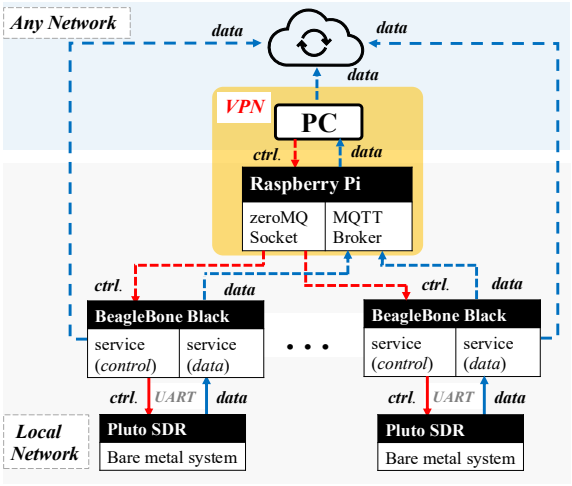

Fig. 2. WiLEAD system architecture.

insensitive to host load and transport jitter. The architecture scales because control traffic is small and logs are compact counters and events.

V. Evaluation

WiLEAD's design relies on two keys assumptions. The first is that contending nodes form a single collision domain: when a surviving contender transmits a dominant symbol, every recessive contender must in principle be able to observe that symbol within the same arbitration window. Second, the system must operate in a detection regime in which dominant symbols can be driven reliably above the local noise floor, while the CFAR detector must keep the false alarm rate sufficiently low for reliable bitwise elimination. The following experiments quantify the extent to which these assumptions hold using a center frequency of 433 MHz within the EU SRD/ISM sub-band, which is used by industrial remote-control links, EU433 LoRa/LoRaWAN, and Wireless M-Bus. We selected this band as a representative sub-GHz setting for evaluating WiLEAD under practical interference. In particular, Sec. V-A quantifies the distance regime in which dominant symbols remain reliably observable and CFAR thresholding continues to separate false alarms and true detections. Table I summarizes the radio configuration used throughout the evaluation. All experiments were conducted in a meeting room with interference patterns typical of an office environment, as shown in Fig. 3. The arbitration procedure itself is not specific to 433 MHz. Moving to another band does not require changes to the protocol logic. It requires re-establishing the same operating regime under the propagation, interference, and regulatory conditions of that band by retuning the runtime RF parameters and re-evaluating the achievable range and detection margin.

Evaluation Metrics. We evaluate WiLEAD using metrics that capture election correctness, time-to-recovery, and spectral footprint—primary constraints in infrastructure-less cyber-physical networks such as UAV swarms and vehicular systems, where coordinator loss must be recovered within a bounded time and control signaling need to coexist with application traffic.

TABLE I
Configuration

Symbol	Value	Meaning
f_c	433 MHz	Carrier frequency
f_s	61.44 MSPS	Sampling rate
BW	10 MHz	Filter bandwidth
P_t	-0.5 dBm	Transmission power
G_{MGC}	50 dB	Manual gain setting
$CFAR$	$\{1\%, 5\%\}$	Constant False Alarm Rate target

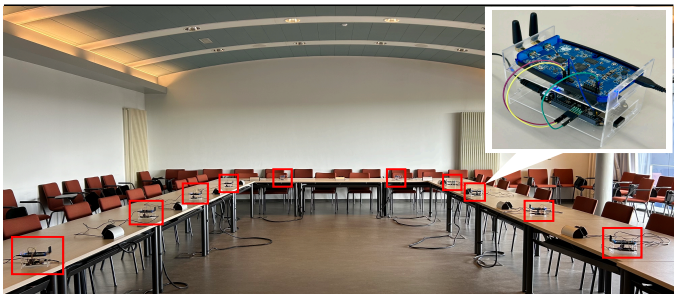

Fig. 3. Indoor deployment of 10 nodes with quasi-grid topology

- *Reliability*: the fraction of trials in which exactly one contender survives through the final arbitration bit, which means a unique leader is elected, capturing correctness under RF impairments and interference.
- *Latency*: the time from starting leader election to the moment a unique winner is decided and all others stop transmitting (P2–P3 in Fig. 1), capturing time-to-recovery under tight reconfiguration budgets.
- *95% Occupied Bandwidth*: the minimal contiguous frequency span containing 95% of the signal's total mean power [23], capturing spectral compactness and thus the feasibility of control–data coexistence in spectrum-constrained deployments.

A. Range Performance

This subsection measures how the CFAR threshold governs the range performance of WiLEAD, examining which choice of threshold yields sufficiently reliable operation. We use three nodes. One node acts as the incumbent leader and is placed at the center of the room. The other two nodes are contenders that compete to become the new leader. We vary the distance between these two contenders while sweeping the CFAR threshold, and we record the leader election reliability as a function of this contender separation. Overall, the results identify the range within which CFAR tuning is effective.

As shown in Figure 4, reliability decreases with distance and is subject to small fluctuations arising from multi-path effects. Across most of the measured range, lowering the CFAR design target from 5% to 1% improves reliability, indicating that WiLEAD is predominantly limited by spurious detections rather than by insufficiently strong dominant symbols in this regime. With fixed gain, WiLEAD maintains at least 90% reliability at up to ≈ 6 m with a 5% P_{FA} threshold and up

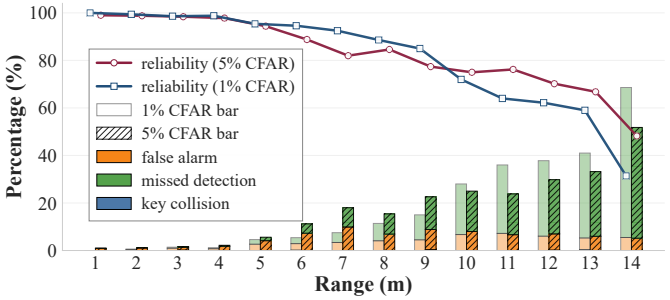

Fig. 4. Range performance with 5% and 1% CFAR thresholds. The bars at the bottom of the graph break down the total probability of failure into the the prevalence of the three possible failure modalities.

to $\approx 7\,\mathrm{m}$ with a 1% P_{FA} threshold. Likewise, the region with at least 80% success reaches $\approx 9\,\mathrm{m}$ for the 5% threshold and $\approx 10\,\mathrm{m}$ for the 1% threshold. A higher threshold hence extends WiLEAD range, and the separation between the curves shows that, over short and medium ranges, the dominant reliability loss is due to false alarms rather than misses. Only at the very edge of the system's useful range, the benefit of a higher threshold starts to disappear because the channel delivers fewer strong symbols. Curves associated with different P_{FA} values then move closer together. This pattern reveals a transition from a false-alarm-limited regime at short and medium ranges to a miss-limited regime near the coverage edge, where missed dominant symbols become the main contributor to reliability loss; equivalently, the single-window miss probability implied by Figure 4 becomes the relevant performance limiter. CFAR tuning therefore primarily improves reliability in the former regime, while offering only limited benefit in the latter.

These observations have two practical implications. When the target operating range lies in the regime where the CFAR curves separate, configuring a low per window false alarm probability, for example on the order of 10^{-2}, effectively suppresses spurious detections and extends the usable range. When the target range approaches the coverage edge, further lowering the false alarm probability provides little benefit, because reliability is then limited by missed symbols rather than false alarms. In that regime, improving reliability relies on additional signal-to-noise ratio, such as antenna gain or power control, rather than more aggressive CFAR settings. Taken together, this suggests that CFAR tuning is the primary mechanism to harden WiLEAD against spurious detections at short and medium range, while link budget and propagation conditions dominate near the range limit.

B. Latency Budget

This subsection studies how an application-imposed latency budget affects WiLEAD's ability to elect a unique leader, and how to map that budget into PHY parameters such as symbol duration and detector thresholding. This is motivated by infrastructure-less cyber-physical networks (e.g., UAV swarms and vehicular systems) where a coordinator can become unreachable and coordination must be restored within a bounded time. Our key conclusion is that WiLEAD exhibits a clear reliability threshold in the few-millisecond regime: at that point, the probability of purely combinatorial key collision becomes negligible, and the remaining failures are governed primarily by physical-layer detection effects that can be controlled through CFAR-based thresholding.

In our experiments. we fix a latency budget and vary the symbol duration, which implicitly trades per-symbol error rates against the number of arbitration bits that fit within the deadline. For each configuration, we run > 500 trials across network sizes and CFAR settings. Figure 5 shows a sharp transition: at a $1\,\mathrm{ms}$ budget, the best success remains below 90%, whereas budgets of $\geq 3\,\mathrm{ms}$ achieve $\geq 97\%$ success in the best operating region. Across settings, a moderate symbol duration around ≈ 40–$60\,\mu\mathrm{s}$ consistently performs best: very long symbols reduce success because too few arbitration bits fit within the budget,while very short symbols degrade detection to the extent that additional arbitration opportunities no longer compensate for it. This interpretation is consistent with the key-collision bound $P_{\mathrm{kc}}(N,B) \leq \binom{N}{2}2^{-B}$. In particular, at the representative $3\,\mathrm{ms}$, $43\,\mu\mathrm{s}$ operating point, the election contains $B = 36$ arbitration symbols, yielding $P_{\mathrm{kc}} \leq 1.46 \times 10^{-10}$ for $N = 5$ and $P_{\mathrm{kc}} \leq 6.55 \times 10^{-10}$ for $N = 10$. The corresponding values observed in our experiments are 0.97×10^{-10} and 4.32×10^{-10}, respectively. Thus, once the latency budget reaches the few-millisecond range, uniqueness is no longer materially limited by arbitration-key collisions. As in the previous section, tightening the threshold from 5% to 1% CFAR improves reliability, especially under tight budgets.Network scale has only a limited impact in the operating region.

These results yield a practical guideline for deadline-provisioned PHY leader election. Under extremely tight budgets (e.g., $\leq 1\,\mathrm{ms}$), unique-winner election is fundamentally challenging on commodity half-duplex radios. In this regime, the deadline simultaneously restricts detection quality and arbitration-key length, and stricter thresholding and conservative symbol choices improve robustness. The key payoff is that once the budget reaches the few-millisecond range (e.g., 3–$4\,\mathrm{ms}$ in our setting), WiLEAD can be provisioned with a simple configuration to achieve consistently high success, by combining moderate symbol durations with enough arbitration symbols to make key collisions negligible. This bounded-latency primitive operates well below the tens-to-hundreds-of-milliseconds recovery commonly observed in (i) higher-layer binary-countdown variants that depend on MAC-level synchronization or platform-specific capabilities [4], [6], [7], and (ii) message-exchange-based leader election protocols whose latency is inherently sensitive to RTT, contention, and loss [13], [14]. In this regime, the residual risk shifts from combinatorial non-uniqueness to physical-layer detection errors, which are more directly governed by thresholding and link conditions and can, if needed, be complemented by lightweight recovery above the PHY.

C. Distribution and Scaling

This subsection quantifies the scalability of WiLEAD under increasing contender counts (N) within a fixed deadline. For

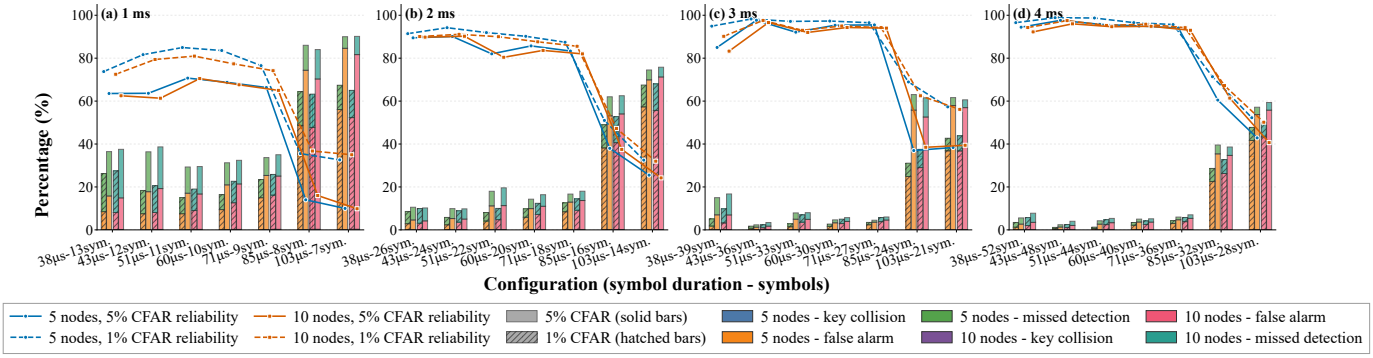

Fig. 5. Leader election reliability and failure mode decomposition under different latency budgets and symbol configurations for 5-node and 10-node systems with 5% and 1% CFAR thresholding. Lines denote reliability, and stacked bars show the contributions of key collision, false alarm, and missed detection.

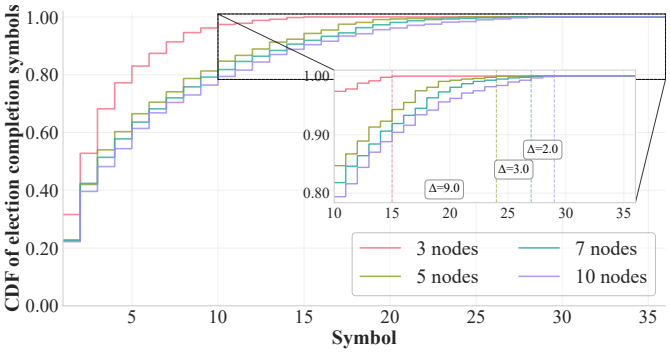

Fig. 6. CDF of election completion symbols under 3 ms latency budget with 1% CFAR and 43 μs symbol duration for different numbers of contenders $N \in \{3, 5, 7, 10\}$.

dynamic systems such as UAV swarms, the protocol must ensure that the completion time does not explode as the network scales. Our key conclusion is that both the typical and near worst-case completion times grow slowly with N, consistent with the expected $\log_2 N$ behavior of symbol-level arbitration, and remain well within a single 3 ms election round.

We fix a 3 ms budget, a 43 μs symbol duration, and a 1% CFAR detector, and vary the number of contenders $N \in \{3, 5, 7, 10\}$. For each trial we record the first bit index at which the election yields a *unique* leader. Repeating 500 trials per N produces an empirical CDF of the finishing bit, shown in Fig. 6. The success-rate metric used earlier corresponds to evaluating this CDF at the last configured bit (36 bits). We also annotate the smallest bit index at which each CDF reaches 1.0; these "finish bits" capture the longest completion observed within one election round. Across all N, the CDFs rise steeply within the first few bits and approach one well before the 36th bit, indicating that most elections terminate quickly and that the 3 ms budget provides ample headroom. As N increases, the distributions shift slightly to the right, but the separation remains small: doubling the number of contenders from 5 to 10 increases the completion bit by about one bit over most of the distribution. The annotated finish bits move by only a few bits across the tested range, showing that even the tail grows slowly with group size. Overall, the observed behavior is consistent with a cost that tracks $\log_2 N$ for typical and near worst-case elections, rather than increasing linearly with N.

These results indicate WiLEAD's favorable scaling: the number of arbitration bits required to reach a unique winner grows slowly with network size and is consistent with $\log_2 N$ behavior in our measurements. This makes WiLEAD well-suited for large and dynamic deployments, where the active set can fluctuate (e.g., UAV swarms or vehicular groups) yet re-election must remain within a fixed deadline. In contrast, message-exchange-based leader election typically incurs $\mathcal{O}(N)$ to $\mathcal{O}(N^2)$ control messages in the worst case, and its completion time becomes increasingly sensitive to RTT, contention, and loss as N grows.

D. Comparison

TABLE II
COMPARISON OF SOLUTIONS

Solutions	Frequency / Bandwidth / Protocol / Platform	Layer	Latency
Raft [14]	• Runs over TCP/IP; • No specific wireless protocol/hardware	Application	• 150-300ms
WiDom [6]	• 2.4 GHz; •∼2MHz Bandwidth; • IEEE 802.15.4; • MicaZ + CC2420	Data Link	• 28.011ms (10 nodes)
WiLEAD	• 433 MHz; • ∼100kHz Bandwidth; • No specific protocols • ADALM-Pluto SDR	Physical	• <5ms (10 nodes)

We compare WiLEAD with representative leader-election solutions implemented at different layers of the protocol stack. As shown in Table II, the comparison highlights a fundamental design trade-off: as leader election is pushed downward from the application layer to the data-link layer and further to the physical layer, the overhead associated with message exchanges, frame structures, and synchronization can be substantially reduced, leading to much lower failover latency. In this context, Raft [14] represents a general-purpose application-layer solution, WiDom [6] brings arbitration closer

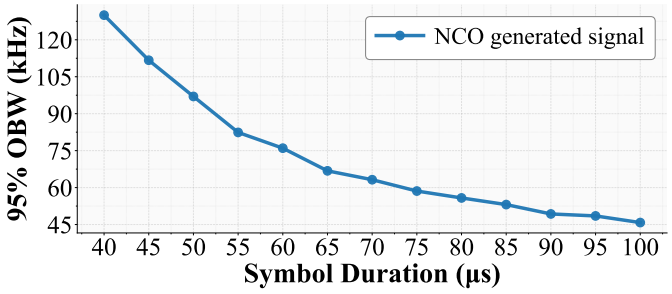

Fig. 7. Symbol duration versus 95% occupied bandwidth (OBW) of NCO generated signal.

to the medium through frame-level dominance, and WiLEAD further moves the election primitive into symbol-level PHY interaction. As a result, WiLEAD achieves the smallest latency and the lowest bandwidth consumption among the compared solutions, which makes it particularly attractive for deadline-driven cyber-physical systems.

Concretely, Raft operates over TCP/IP and typically relies on conservative election timeouts of 150–300 ms, which are appropriate for robust consensus-oriented coordination but are too large for millisecond-scale recovery. WiDom reduces part of this overhead by performing dominance-based arbitration at the data-link layer, which incurs synchronization-related costs, resulting in a latency of 28.011 ms for 10 devices. In contrast, WiLEAD performs leader election directly through PHY symbol-level bitwise elimination at 433 MHz, requires only about 100 kHz bandwidth, and completes election within 5 ms for 10 devices. This advantage follows from eliminating higher-layer frame exchanges and avoiding explicit network-wide synchronization. Moreover, our evaluation shows that, once the latency budget reaches the few-millisecond regime, WiLEAD can achieve consistently high success rates, while its completion time grows sublinearly with network size.

These results suggest that WiLEAD should be understood as a bounded-latency leader-election primitive rather than as a direct replacement for higher-layer consensus protocols. For infrastructure-less mobile systems such as UAV swarms and vehicular groups, where coordinator re-establishment must occur within a strict deadline and control signaling must coexist with application traffic under limited spectrum, WiLEAD provides a more suitable operating point than application-layer or frame-level alternatives. At the same time, this comparison should be interpreted carefully: Raft provides stronger safety semantics under richer failure models, whereas WiLEAD prioritizes fast recovery and spectral efficiency. Therefore, a practical design direction is to use WiLEAD as a fast PHY-level failover mechanism, and, when stronger coordination guarantees are required, complement it with lightweight higher-layer confirmation or recovery logic.

E. Bandwidth Consumption

We quantify WiLEAD's spectral footprint and identify how the symbol duration of the gated NCO tone controls the 95% occupied bandwidth ($\mathrm{OBW}_{95\%}$) on our AD936x-based transmitter. This is critical for infrastructure-less cyber-physical networks such as UAV swarms and vehicular systems, where control signaling must coexist with application traffic under constrained spectrum without causing mutual interference. Our key conclusion is that WiLEAD is inherently narrowband in our setting ($\mathrm{OBW}_{95\%} < 130\,\mathrm{kHz}$), and that modestly increasing symbol duration significantly tightens $\mathrm{OBW}_{95\%}$ with diminishing returns beyond $\approx 100\,\mu\mathrm{s}$.

We vary only the symbol duration of the NCO-generated pulses used for dominant bits while keeping all other parameters fixed, sweeping from $40\,\mu\mathrm{s}$ to $100\,\mu\mathrm{s}$ and holding each setting long enough for a stable spectrum estimate. Figure 7 shows a monotonic contraction of $\mathrm{OBW}_{95\%}$ from $\approx 130\,\mathrm{kHz}$ at $40\,\mu\mathrm{s}$ to $\approx 53\,\mathrm{kHz}$ at $100\,\mu\mathrm{s}$. This inverse relationship follows standard time–frequency shaping: shorter gating yields sharper edges and stronger high-frequency components, whereas longer symbols smooth transitions and confine energy closer to the main lobe. The AD936x NCO maintains phase continuity and gates in digital baseband, reducing edge jitter that would otherwise widen the spectrum.

WiLEAD maintains a compact spectral footprint that is directly tunable via symbol duration. Concretely, the measurement lies within $130\,\mathrm{kHz}$ $\mathrm{OBW}_{95\%}$ across the evaluated configurations , which is only a small fraction of widely used channels, for example $\approx 6.5\%$ of a $2\,\mathrm{MHz}$ IEEE 802.15.4 channel, $\approx 1.3\%$ of a $10\,\mathrm{MHz}$ vehicular channel specified in IEEE 802.11p/DSRC. Increasing symbol duration toward $\approx 100\,\mu\mathrm{s}$ further tightens $\mathrm{OBW}_{95\%}$ with limited impact on millisecond-scale election budgets, whereas shorter symbols reduce election time at the cost of a modestly wider footprint.

VI. Discussion

Control Plane and Symbol Granularity. Our prototype drives the AD936x RF chain via SPI register writes. While this bare-metal approach ensures deterministic timing, the overhead of command serialization dictates a minimum symbol duration in the tens of microseconds, effectively limiting the number of arbitration bits that fit within a hard latency constraint. Although commercial stacks typically abstract such low-level access, we identify two practical integration paths. First, as evidenced by WiFi-BA [7], arbitration logic can be injected via firmware updates on modern chipsets using programmable DSPs. As WiLEAD relies on on-off keying modulation, which is low complexity and hence suitable for lightweight microcode implementation. Second, for strictly standardized environments, the protocol can be offloaded to a low-cost, narrowband auxiliary radio. Since the signal occupies less than $100\,\mathrm{kHz}$, such a dedicated circuit adds low hardware cost. More importantly, this out-of-band approach allows the primary high-bandwidth radio to operate without modification.

Energy Efficiency and Duty Cycling. The current implementation employs a continuous pilot tone. The receiver utilizes the falling edge of this tone as an immediate trigger for leader election. While this design minimizes failure detection latency, it imposes a 100% transmit duty cycle which challenges power-constrained platforms. To address strict power

budgets, the system can transition to a pulsed heartbeat model. By broadcasting short, periodic beacons, the duty cycle is reduced to the ratio of the pulse duration to the beacon interval. In this mode, the trigger signal shifts from a physical falling edge to a logical watchdog timeout referenced to the last successfully received heartbeat. This introduces clock drift between the last heartbeat and the timeout. However, standard crystal oscillators typically have a frequency error of around $20\,\mathrm{ppm}$, which translates to only a few microseconds of drift over a typical $100\,\mathrm{ms}$ interval. Since WiLEAD employs a pre-round guard interval in the tens of microseconds (Sec. III-B), the system can absorb this residual skew, ensuring that the core physical-layer arbitration remains robust.

Relationship to off-the-shelf PHYs. WiLEAD is orthogonal to existing PHYs. Its logic does not depend on a particular packet structure or higher-layer frame exchange, but only on direct fine-grained control of radio hardware. This paper hence calls for the realisation of leader election protocol as a first-class citizen next to, rather than on top of, link-layer communication mechanisms provided by conventional technologies such as Bluetooth or WiFi.

VII. Conclusion and Future Work

This paper presented WiLEAD, a physical-layer primitive that shifts leader election from frame-level exchanges to symbol-level arbitration. By eliminating reliance on MAC backoff and network-wide synchronization, WiLEAD with high probability achieves latency bounds that scale logarithmically with the number of contenders. Our SDR implementation demonstrates reliable leader election within $5\,\mathrm{ms}$ using only $100\,\mathrm{kHz}$ of bandwidth, proving that physical-layer logic can deliver predictable control for mobile cyber-physical systems such as UAV swarms and vehicle systems with a minimal spectral footprint.

Our future work will focus on addressing the deployment and efficiency challenges identified in Section VI. To overcome the symbol duration floor imposed by the SDR control plane, we plan to migrate the arbitration logic to firmware-defined implementations on commercial chipsets or FPGA-based platforms. Simultaneously, to address energy constraints in power-critical applications, we will implement the pulsed heartbeat model, which allows system designers to balance failure discovery latency against battery life without altering the core election primitive.

Acknowledgment

This work is partially funded by the Research Fund, KU Leuven (C14/24/093; SwarmNet) and J. Oostvogels' Research Foundation–Flanders Fellowship (FWO; 1224325N).

References

[1] X. Zhang, L. You, and G. Hu, "An efficient and robust multidimensional data aggregation scheme for smart grid based on blockchain," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 3949–3959, 2022.

[2] R. Gusella and S. Zatti, "An election algorithm for a distributed clock synchronization program," University of California, Tech. Rep., 1985.

[3] A. Franchi, H. H. Bülthoff, and P. R. Giordano, "Distributed online leader selection in the bilateral teleoperation of multiple uavs," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 3559–3565.

[4] D. Christmann, R. Gotzhein, and S. Rohr, "The arbitrating value transfer protocol (avtp) - deterministic binary countdown in wireless multi-hop networks," in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, 2012, pp. 1–9.

[5] M. Ringwald and K. Römer, "Bitmac: a deterministic, collision-free, and robust mac protocol for sensor networks." in *EWSN*, 2005, pp. 57–69.

[6] N. Pereira, B. Andersson, and E. Tovar, "Widom: A dominance protocol for wireless medium access," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 120–130, 2007.

[7] P. Huang, X. Yang, and L. Xiao, "Wifi-ba: Choosing arbitration over backoff in high speed multicarrier wireless networks," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1375–1383.

[8] R. Vacus and I. Ziccardi, "Minimalist leader election under weak communication," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 2025, pp. 406–416.

[9] Y. Ashkenazi, R. Gelles, and A. Leshem, "Brief announcement: Noisy beeping networks," in *Proceedings of the 39th Symposium on Principles of Distributed Computing*, 2020, pp. 458–460.

[10] Z. Ying, M. Ma, Z. Zhao, X. Liu, and J. Ma, "A reputation-based leader election scheme for opportunistic autonomous vehicle platoon," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3519–3532, 2021.

[11] H. Garcia-Molina, "Elections in a distributed computing system," *IEEE transactions on Computers*, vol. 31, no. 01, pp. 48–59, 1982.

[12] E. Chang and R. Roberts, "An improved algorithm for decentralized extrema-finding in circular configurations of processes," *Communications of the ACM*, vol. 22, no. 5, pp. 281–283, 1979.

[13] I. Moraru, D. G. Andersen, and M. Kaminsky, "There is more consensus in egalitarian parliaments," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, 2013, pp. 358–372.

[14] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 USENIX annual technical conference (USENIX ATC 14)*, 2014, pp. 305–319.

[15] Tindell, Hansson, and Wellings, "Analysing real-time communications: controller area network (can)," in *1994 Proceedings Real-Time Systems Symposium*, 1994, pp. 259–263.

[16] A. Itai and M. Rodeh, "Symmetry breaking in distributive networks," in *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*. IEEE, 1981, pp. 150–158.

[17] B. S. Chlebus, D. R. Kowalski, and A. Pelc, "Electing a leader in multi-hop radio networks," in *International Conference On Principles Of Distributed Systems*. Springer, 2012, pp. 106–120.

[18] D. Bharadia and S. Katti, "Full duplex {MIMO} radios," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 359–372.

[19] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 375–386.

[20] S. Haykin and X. B. Li, "Detection of signals in chaos," *Proceedings of the IEEE*, vol. 83, no. 1, pp. 95–122, 1995.

[21] R. Jain and I. Chlamtac, "The p2 algorithm for dynamic calculation of quantiles and histograms without storing observations," *Communications of the ACM*, vol. 28, no. 10, pp. 1076–1085, 1985.

[22] H. Rohling, "Radar cfar thresholding in clutter and multiple target situations," *IEEE transactions on aerospace and electronic systems*, no. 4, pp. 608–621, 2007.

[23] M. Höyhtyä, A. Mämmelä, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson, "Spectrum occupancy measurements: A survey and use of interference maps," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2386–2414, 2016.