RESEARCH STATEMENT

Md Ashikul Haque

My areas of interest and expertise include the Internet of Things (IoT), wireless networked systems, real-time and embedded systems, cyber-physical systems (CPS), and their integration with artificial intelligence (AI). My research addresses the challenges of security, reliability, scalability, timeliness, and energy efficiency in these areas through theory and system design, with applications in autonomous and connected vehicles, smart cities, industrial control, and disaster response. Defense against jamming and coexistence with other networks are very challenging in low-power wide-area networks (LPWANs). Solutions designed for Wi-Fi, cellular, and other wireless networks cannot be adopted directly, because LPWAN devices have limited computation, low transmission power, and small batteries. My major research initiatives and future plans are summarized as follows.

1 Anti-Jamming for Low-Power Wide-Area Networks (LPWANs)

LPWANs such as LoRa let battery-powered devices send data over kilometers for years, but their weak signals are easy to jam. A jammer is a transmitter that deliberately corrupts legitimate packets. I first modeled jamming as a Stackelberg game in which the nodes of Sensor Network Over White spaces (SNOW), our laboratory's LPWAN, lead and a reactive, energy-limited jammer follows (EWSN 2023).[1] The nodes counter the jammer by hopping across frequencies and adjusting their transmission power. The base station learns this strategy with multi-agent reinforcement learning (RL) on behalf of the nodes, which do none of the computation. This defense improved the packet reception rate (PRR) by up to 31.83× and the energy efficiency by up to 34× in simulations with 50 to 240 nodes, and the joint game converges in 10 to 15 minutes, while power control alone takes over an hour and frequency hopping alone did not converge in two hours. It achieved a PRR of 94% on an indoor testbed of Universal Software Radio Peripheral (USRP) B200 radios and Texas Instruments CC1310 nodes, against 5.4% for SNOW without it.

Multi-Gateway Decoding. A packet and a jamming signal reach each gateway with a different relative delay, because the gateways sit at different distances from the node and from the jammer. Exploiting this variation, I have designed an anti-jamming technique for LoRa that decodes packets at signal-to-noise ratios below −35 dB, where seven existing LoRa decoders have a 100% symbol error rate (IoTDI 2024). Three gateways in a line with equal spacing leave a clean fragment for almost every jammer position. The exceptions are a jammer very close to the node or at the node's mirror position, and a fourth gateway off the line removes the mirror case. The LoRa Network Server (LNS) aligns the copies of a packet from the gateways and stitches their jammer-free fragments into one packet. It finds clean fragments by a threshold on signal power, and a convolutional neural network classifies the fragments near the threshold. It tolerates a stealthy jammer that mimics the LoRa preamble, varies its packet length, and moves between bursts. The technique is link-layer-agnostic and imposes no overhead on the nodes. I evaluated it outdoors with USRP gateways and commercial off-the-shelf LoRa nodes. Against a 40 dBm jammer, it improved PRR by up to 83.91× and cut the energy per delivered packet by up to 115.65×.

Single-Gateway Decoding. Placing several gateways around every node is not always possible, so I have also developed decoders for a single gateway. The first handles multiple jammers that transmit LoRa packets (MobiHoc 2025). After recovering every symbol boundary, the decoder estimates the legitimate signal power from the preamble, which usually reaches the gateway before the jamming signals. It then zeroes every frequency bin that exceeds this power by a margin. The decoder adds about 3 ms per 32-byte packet, a small fraction of the packet's airtime. Against

[1] Each linked tag (e.g., SenSys 2026) cites one of my peer-reviewed papers. My CV lists all of them in full.

jammers with 1,000 times the node's transmit power, it improved PRR from 0.32% to 34.14%, a gain of 106.56× over the best prior LoRa defense, which collapses beyond one jammer.

The second decoder handles up to four collaborative jammers that need not transmit chirps (EWSN 2026). A special node transmits a known signal that stays orthogonal to the LoRa signal but collides with the jamming signals. It listens for uplinks with the channel activity detection feature of LoRa chips, which Long Range Wide Area Network (LoRaWAN) end devices do not use for channel access, and transmits the known signal as soon as it detects activity. The gateway uses the known signal to disentangle the combined jamming signal and cancel it. A lightweight generative model at the gateway maps the jamming energy from one spreading factor to another when the known signal and the packet use different spreading factors. With three jammers at 50 m, PRR rose from 0.4% to 40.7%. The decoder kept 40 to 70% PRR under channel-switching, proactive, and random jammers. Decoding takes about 12 ms per 32-byte packet, against 11 ms for a standard receiver.

To our knowledge, my game-theoretic defense (EWSN 2023) is **the first anti-jamming work for LPWANs**, and my multi-gateway technique (IoTDI 2024) is the first to mitigate jamming in LoRa. My decoders run at the gateway or network server, so they need no hardware change to the 150 million deployed LoRaWAN devices.

2 Coexistence and Cross-Technology Communication

Learning-Based Coexistence. As LPWANs grow, several LoRaWAN operators in a city share the same unlicensed channels, and their nodes transmit without carrier sensing. Their frames collide, and no operator sees the others' traffic. Existing learning-based schemes run tabular Q-learning at each node, so they spend scarce energy and see only a local view. I have moved the learning to the LNS, which sees the packets and acknowledgments of all its gateways (INFOCOM 2025). I formulated parameter selection as a Markov decision process over each node's channel, spreading factor, transmission time, and power, with a reward that weighs delivery against energy. A deep Q-network learns this policy for the uplink, and a second agent learns when and how to acknowledge, so nodes stop retransmitting packets that already arrived. The LNS infers a lost acknowledgment when a delivered packet is retransmitted, so the second agent learns without extra messages. The LNS sends these parameters through the existing downlink. I evaluated this approach through physical experiments and NS-3 simulations. Over the decentralized state of the art, it raised PRR by up to 70.97% and reduced energy per packet by up to 62.91%.

When neighboring operators each run such an agent, the agents can converge on the same choices and collide. I have developed a coordination service that shares only model-level information, never traffic (ICLR 2026 workshop). A hypernetwork maps this information to an update for each agent, so that the agents make complementary choices. Over independent agents, coordination raised PRR by at least 20% with two networks and 10% with four.

Cross-Technology Communication. Long Range Frequency Hopping Spread Spectrum (LR-FHSS) is a LoRa physical layer for long-range and direct-to-satellite uplinks. However, LR-FHSS devices receive only LoRa, so the downlink to an LR-FHSS device must use LoRa. Because LoRa has a shorter range, a gateway can hear a far LR-FHSS device but cannot reply to it. A LoRa receiver cannot decode an LR-FHSS transmission, because LoRa requires chirps with linear frequency modulation, whereas LR-FHSS sends narrowband hops. I enabled the gateway's LR-FHSS transmitter to produce a waveform that the device's LoRa receiver can decode. It needs baseband access at the gateway and in-phase and quadrature (I/Q) access at the device, but no change to the radio frequency (RF) front end (SenSys 2026). By controlling its hop sequence, the transmitter synthesizes a pseudo-chirp from narrowband hops. A Chirp Reconstruction Module at the receiver aligns the fragments, corrects carrier and timing offsets, and passes a LoRa-compatible chirp to the standard demodulator. With Aakriti Jain and Prof. Venkata Prashant Modekurthy, I implemented this

cross-technology communication (CTC) link on USRP B200 with GNU Radio and deployed it with one gateway and five nodes in a suburban area. It improved long-range reliability by up to 79% over LoRa. Acknowledgments sent over the link raised end-to-end LR-FHSS PRR from 79.04% to 94.57%.

To our knowledge, my CTC technique (SenSys 2026) is **the first to enable communication from LR-FHSS to LoRa**. It lets a gateway acknowledge a far device, change its reporting rate, or trigger an emergency transmission. My network-server learning (INFOCOM 2025) is cited by a 2026 paper in IEEE Transactions on Green Communications and Networking that schedules satellite LoRa uplinks with deep RL. I developed both coexistence designs with Prof. Haibo Zhang (now at the University of New South Wales), and we still collaborate.

3 Real-Time Guarantees for Wireless Networks and AI Agents

Real-Time Wireless Networks. A forest fire makes every LoRa node near it report at once, and under LoRaWAN's random access many reports collide when they matter most. *Burst-MAC*, which Aakriti Jain led with my help on the protocol design and the physical experiments, switches the nodes to time slots only during a burst (RTSS 2024). Each node derives its slots from a hash function, so the gateway broadcasts no schedule. Gateway acknowledgments resolve slot collisions and bound the latency of delivered packets. On LoRa devices and in NS-3, Burst-MAC improved PRR by 4.5×, cut energy use by up to 6×, and reduced latency by up to 2.5×.

Industrial wireless sensor-actuator networks (WSANs) run control loops of different criticality over one network. In a plant emergency, high-criticality (HC) loops need more airtime, but low-criticality (LC) loops must not be starved. I have designed *ReMix*, which keeps a stable base schedule and a contingency pool of preinstalled cells, which are reserved pairs of time slot and channel (ICCCN 2026). A short broadcast bitmap selects which cells of the pool to activate, and a duplicated-payload frame layout lets the receiver rebuild a corrupted packet in software on stock radios. In HC mode, ReMix reclaims redundancy from LC traffic, while a service budget bounds each LC loop's service loss. On a 15-node Tmote Sky testbed, ReMix raised HC deadline satisfaction from 32.9% to 85.9% under the heaviest stress, and LC delivery stayed at 63.4% in HC mode.

With Prof. Dali Ismail, I have also studied scalable real-time control over WSANs (SDN-CPS 2024). This work proposes to co-design model predictive control with the transmission schedule online, with an anytime local search that would find a feasible schedule first and then improve it.

Real-Time AI Systems. An AI agent that drives an actuator has a deadline on its first useful action, not on its last token. Existing large language model (LLM) serving systems improve throughput and tail latency, but they do not reserve capacity for the part of a response that makes an action usable. I have developed *Deadline-Aware Token Scheduling*, which splits each request into three parts: prefill, which processes the prompt; a mandatory prefix, the shortest output that a validator accepts as a usable action; and an optional suffix (RTSS 2026). It admits a critical request only when its mandatory work passes a deadline guard and its key-value cache reservation fits in memory. It dispatches admitted mandatory work in earliest-deadline-first order and runs optional tokens only when slack remains. I have proved a sufficient schedulability condition for the mandatory mode under bounded service and blocking, and the scheduler held zero admitted-critical misses under arrival jitter and service noise. Without the mandatory protection, 68.4% of admitted critical requests missed their deadlines. A multi-mode variant gives each critical request three certified service modes, so that it can fall back to a compact or safe response when its full response is infeasible. On a vLLM stress test with no per-workload tuning, it cut the share of critical requests rejected or late from 92.5% under first-come first-served scheduling to 23.8%.

The artifact of Deadline-Aware Token Scheduling passed the RTSS 2026 Artifact Evaluation. Burst-MAC is cited by LoRaHART at ECRTS 2025. My workshop paper on real-time control in industrial CPS is cited at IEEE RTAS 2025 and RTSS 2025.

4 Future Research Direction

My goal is to establish a leading research group that makes IoT and physical AI dependable by combining AI with next-generation wireless and real-time systems. Physical AI is AI that senses the physical world and acts on it through machines. My doctoral research asked how a wireless network stays reliable and on time when jammers, neighboring networks, and competing deadlines contend for its resources. Physical AI and IoT beyond cellular coverage raise this question again with shared edge servers, short satellite passes, and moving vehicles. I will address it in multiple new projects.

4.1 Real-Time Physical AI

A warehouse robot may need a stop command from an AI model within 100 to 250 ms. Serving systems for robot fleets raise the share of on-time actions on a shared graphics processing unit (GPU), but none bounds the probability that an action reaches its robot late.

This project treats each action as a real-time task that spans a shared GPU and a wireless link, with ground robots as the testbed. I will schedule inference and transmission together to bound the probability that an action misses its end-to-end deadline. Agents that call tools pause, and the server must keep or rebuild their GPU memory. I will treat these pauses as self-suspensions and decide which memory to keep through a response-time analysis for chains of calls. Because timing assumptions can fail, I will build monitors that switch a machine to a verified safe controller.

4.2 Secure Direct-to-Satellite IoT for Remote Areas

Direct-to-satellite (DtS) IoT lets remote LoRa sensors reach a satellite in low Earth orbit. These sensors can warn of floods, wildfires, and other hazards, which leave little time to act. In July 2025, Texas's Guadalupe River rose 32.5 feet in 1.5 hours. Jamming raises challenges that are new or harder with satellites. A satellite at 550 km passes over in about 10 minutes, so a lost warning may wait hours. One ground jammer can disrupt reception across a satellite's whole coverage area. Because satellites are hard to update in orbit, a defense should need no change on the satellite. Acknowledgments are rare, so a device cannot tell jamming from collisions with unseen networks. Prior designs handle intermittent contact or jamming, but none certifies deadlines under both.

Each device will derive its transmission times and channels from its LoRaWAN session key and frame counter. Devices already hold these keys, so this needs only a firmware update, not new key management. The network server will know when and on which channel each packet starts, but a jammer will not, so it cannot aim at a packet in advance. Given the predicted passes and a jammer with limited power, airtime, and reaction speed that knows everything except the key, I will certify which messages meet their deadlines and decline the rest. I will also develop AI models that estimate, from signal measurements across passes, whether missing packets were jammed, collided, or faded. The models will only choose among certified schedules, so a wrong estimate cannot break a certificate. I will test on a campus ground station and a drone gateway.

4.3 Beyond-Line-of-Sight Perception for Connected Autonomous Vehicles

Any obstacle can hide an oncoming car or a crossing child from a vehicle's cameras and lidars. Intersections account for about one-quarter of U.S. traffic deaths. Cooperative perception lets vehicles share detections, but only timely ones help. Standards set per-message latency and reliability targets, but no system bounds the probability that shared data arrives too late for a maneuver.

Links can change within a single maneuver, so I will bound this probability over whole maneuvers and have a vehicle yield when shared data may be late. A vehicle should wait out congestion but rely on its onboard sensors under jamming, so I will develop models that tell the two apart from signal measurements at the receiver. When a jammer blocks reception at a vehicle, I will have roadside units that heard the shared detections forward them over the cellular network. Prototypes will run on software-defined radios and small vehicles.