Design of a Low-Latency, High-Reliability Wireless Communication System for Control Applications

Matthew Weiner, Milos Jorgovanovic, Anant Sahai, and Borivoje Nikolić

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley

Abstract—**High-performance industrial control systems with tens to hundreds of sensors and actuators use wired connections between all of their components because they require low-latency, high-reliability links to maintain stability; however, the wires cause many mechanical problems that moving to wireless links would solve. No existing or proposed wireless system can achieve the latency and reliability required by the control algorithms because they are designed for either high-throughput or low-power communication between a pair or a small number of terminals. A preliminary wireless system architecture is proposed that focuses on low-latency operation through the use of reliable broadcasting, semi-fixed resource allocation, and low-rate coding. For an industrial printer application with 30 nodes in the control loop and a moderate information throughput of 4.8Mb/s, the system can achieve latencies under 2ms for SNRs above 7dB.**

Index Terms—**Wireless control, industrial control, low-latency, high-reliability, bounded latency, M2M, Internet of Things, cyber-physical systems, wireless sensor and actor networks**

I. Introduction

The explosion in the number and capability of mobile devices has fueled an insatiable demand for higher data rates. To increase throughput and deal with limits on available spectrum, the goal has been to maximize the spectral efficiency of wireless systems using information theoretic tools. These gains have come at the cost of secondary system parameters, such as latency, that do not fit directly into information theory's framework. As mobile devices move toward ubiquity, new and important applications are emerging beyond delivering high-speed data to individual users. In the vision of the Internet of Things, a huge number of ubiquitously distributed, mobile embedded systems and access devices will communicate both with each other and with the cloud. This opens the door for truly immersive computing paradigms where wireless devices move beyond only sensing the environment; they will also be wirelessly connected to actuators that can manipulate the surrounding environment. In many instances, the sensors and actuators will operate in control loops with varying degrees of latency requirements (Table I) [1].

TABLE I
Control system requirements

Application	Latency	Error Rate	# Nodes	Throughput
VoIP	10ms	10^{-2}	1-10	500kb/s
Smart grid/M2M	> 1s	10^{-5}	10-1000	1-100kb/s
Industrial control	1-2ms	10^{-8}	10-100	5Mb/s

In recent years, researchers have looked at the problem of wireless control from two angles. On the theoretical side, they examine how to change control algorithms to cope with the latency introduced by communication systems, ranging from using a modified form of optimal control to using non-uniform or event-triggered sampling [2]–[5]. On the implementation side, there has been interest in determining the performance of control systems using existing wireless standards [6]–[9] and modifying those standards to increase performance for applications such as the smart grid, VoIP, and M2M type communication [10]–[13]. Additionally, the wireless sensor and actor network (WSAN) community has developed numerous protocols that have guaranteed latency bounds and acceptable reliability [14].

Despite this work, industrial control systems do not have a wireless solution because their latency and reliability specifications are too stringent. However, these systems would greatly benefit from wireless links because wired connections cause many mechanical issues. In industrial and medical robots, wires are the primary cause of failure because the wiring from the controller to the sensors and actuators suffers from stress and fatigue. In automobiles and airplanes, wires are some of the most heavy and costly components and are difficult to route. Focusing primarily on latency and reliability requires a different approach to the design of wireless systems. This paper quantifies the needs of wireless control systems, provides a method to guarantee that communication is reliable and that the latency constraint is not violated up to a tolerable probability of error for a given channel model, and presents an example wireless system design tailored to industrial control. The system has redesigned PHY and MAC layers that can meet the tight latency and reliability specifications in a slow fading environment. It does this in part by having a fixed initial transmission schedule, budgeting enough time for the worst-case number of retransmissions, and by using very low-rate codes to optimally balance the number of retransmissions with the coding overhead.

The paper is organized as follows: Section II defines a metric for the requirements of wireless control systems, Section III analyzes the problems with current wireless systems, Section IV proposes a preliminary low-latency, high-reliability wireless architecture, and Section V evaluates the architecture for an industrial printer application.

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

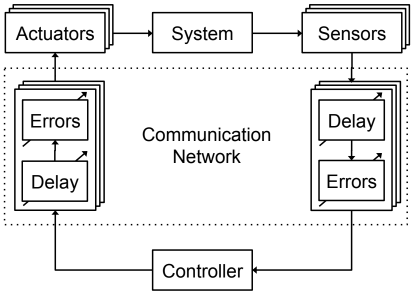

Fig. 1. Block diagram of a centralized control system.

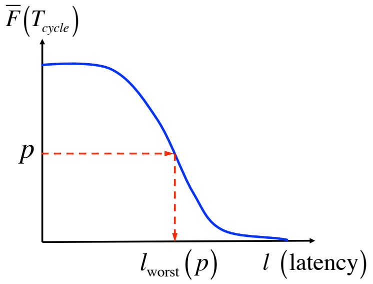

Fig. 2. Graphical interpretation of the worst-case latency $l_{\text{worst}}(p)$.

II. Wireless Control Systems

A. Requirements of Wireless Control Systems

Control systems have three basic elements: sensors, actuators, and controllers. The sensors measure the state of the system, the actuators manipulate the system, and the controllers give instructions to the actuators based on the sensors' observations and information from other controllers. Many industrial control systems are centralized or can be broken down into several centralized subsystems, and the topology is fixed ahead of time. In a centralized control system, a single controller issues instructions to all of the actuators (broadcast traffic), then receives updated state information from all of the sensors (convergecast traffic). The time it takes to complete this is called the cycle time, or T_{cycle}. Ideally, T_{cycle} is negligible and no errors occur during data transmission between the nodes.

When the control system is implemented, a communication system links the controller to the actuators and sensors in a star or daisy chain network topology with the controller at the center. These links might have random, unbounded delays associated with them, and they can inject errors into the transmitted data (Fig. 1). Communication delay degrades the performance of the control system. If the delay is larger than 20-60% of the time constant of the closed loop system, here called the critical delay l_{crit}, the controller cannot respond to changes in the system quickly enough, and the control system fails [15]. Any errors in the transmitted data can cause the system to fall out of specification or become unstable. Since subsequent measurements are correlated, the system's state may be estimated if a sensor measurement is lost or has errors [16], but this results in suboptimal performance at best. Errors in instructions transmitted to the actuators cannot be corrected in the same manner since the actuators simply execute the received instructions. Therefore, this situation is best avoided.

B. Metrics for Evaluating Wireless Control Systems

A communication system used for control cannot have delays larger than the target control algorithm's l_{crit} or allow any errors in transmitted data. However, communication systems can never guarantee error-free operation due to channel impairments, such as noise and fading, so a tolerable probability of failure, p, must be defined. This should be selected small enough such that errors are not expected to occur during the system's lifetime and is analogous to an allowable bit-error rate in a traditional communication system. Assuming that failing to detect errors in the received data has probability much smaller than p (which is possible to accomplish with FEC and CRC), errors in the received data can be avoided by retransmitting the data until it is received correctly. If the channel is poor, this would require a potentially unbounded amount of time, and might cause the information to miss the deadline. This implies that the only way that the system can fail practically is if its latency is larger than l_{crit}. Therefore, the metric of interest for wireless control systems is its worst-case latency $l_{\text{worst}}(p)$ for a given value of p. Formally, for a given p, the worst-case latency of the system $l_{\text{worst}}(p)$ is here defined as

$$l_{\text{worst}}(p) = \min l \text{ s.t. } \Pr[T_{\text{cycle}} \geq l] < p \tag{1}$$

Fig. 2 shows a graphical interpretation of $l_{\text{worst}}(p)$ using the complementary cumulative distribution function of T_{cycle}, $\overline{F}(T_{\text{cycle}})$, as a lookup table for $l_{\text{worst}}(p)$.

If $l_{\text{worst}}(p)$ is smaller than l_{crit}, then the communication system can be used in the control system. This gives a probabilistic guarantee on the performance of the system, and it allows the achievable latency of the system to scale with the reliability requirement. Wired systems have a small $l_{\text{worst}}(p)$ since they have good channels and require few retransmissions. Current wireless systems have a large $l_{\text{worst}}(p)$, which limits them to being used in control systems with an l_{crit} of seconds. To implement a control system with an l_{crit} on the order of milliseconds, a new wireless architecture must be created and validated. To do this, the shortcomings of current wireless systems that lead to a large $l_{\text{worst}}(p)$ in industrial control environments are analyzed.

III. Shortcomings of Current Wireless Systems

Previous designs of a wireless communication system for control have either modified existing wireless standards or developed protocols based off those in wireless sensor networks.

A. Standards-Based Systems

Wireless standards fall into two broad categories: high-performance and low-power. High-performance standards have been designed with the mindset of rapidly sending large amounts of data between a pair of users, one of which is

TABLE II
MAC AND PHY LAYERS OF CURRENT WIRELESS STANDARDS

	IEEE 802.11ac [17], [18]	**LTE** [19], [20]	**ZigBee/ W-HART** [21]–[23]
Network structure	Star	Star	Mesh
Medium access	CSMA/CA	Scheduled	CSMA/CA
Retransmissions	ARQ	HARQ	ARQ
Signaling	OFDM	DL: OFDMA, UL: SC-FDMA	DSSS
FFT Sizes	64-512	128-2048	-
Bandwidth (MHz)	20-160	1.25-20	5
Reference signals	Preamble	Continuous	Preamble
Code types	Convolutional, LDPC	Turbo	No FEC
Code rates	1/2, 2/3, 3/4, 5/6	Punctured 1/3	1
Modulations	BPSK-64QAM	QPSK-64QAM	OQPSK
Peak data rate (Mb/s)	6930	DL: 326, UL: 86	0.25
Max. antennas	8	DL: 4, UL: 1	2
Multi-user MIMO	Yes	Yes	No
Multi-hop	No	No	Yes
Diversity Sources	Frequency, Time, Beamform or Space-time BC	Frequency, Time, Beamform or Space-time BC	Time, Multi-user

usually a human that can tolerate moderate latencies. Low-power standards attempt to send data efficiently between a sensor and central node, usually via short hops between other nodes in the network. Each node transmits data infrequently, and latency is often sacrificed to increase efficiency. In contrast to the design targets for the high-performance and low-power standards, control systems periodically send small amounts of data to many different users, all of which are machines interacting with a system that has a strict delay tolerance.

IEEE 802.11ac and LTE are the best examples of high-performance wireless standards, and ZigBee and WirelessHART are the most widely adopted low-power standards and are used in lower-performance wireless control systems. Table II summarizes the media access control (MAC) and physical (PHY) layers for these standards. Modifying these standards for use in high-performance control systems has not been successful because they have large deterministic or random latency that does not scale well with the number of nodes in the network.

Contention-based MACs and packet-based networks have a large deterministic and random latency overhead due to using a preamble, interframe spacings, and random backoffs. Systems with a central node scheduling medium access and that periodically broadcast reference signals have comparatively less overhead and can achieve tighter synchronization, but they have difficulty informing nodes of assigned retransmission slots quickly over poor channels. Also, they must reserve resources for the reference signals and for distributing the schedule, which decreases the useful data rate or limits the maximum number of connected users.

None of the standards primarily focus on minimizing their block error rates because that is not optimal for throughput or efficiency [24]. They use code rates no lower than 1/3 (but usually select significantly higher rates) and rely on retransmissions to correct any errors that occur. In 802.11ac, retransmissions have a moderate overhead due to recontending for the medium and retransmitting the preamble. On top of the overhead of 802.11ac, ZigBee and WirelessHART have additional overhead since the recontention and transmission occurs at each hop. In LTE retransmissions have a large overhead of at least 4-8ms due to the network architecture.

Most of the standards allow a combination of time diversity from interleaving codewords in time, frequency diversity from interleaving codewords across subcarriers, and spatial diversity through multiple antenna techniques. Time diversity cannot be used since the cycle time is shorter than the coherence time. LTE schedules resources for all users based on current channel conditions, but this may not be possible in control systems because the low latency constraint prevents the central node from learning all of the channels. ZigBee and WirelessHART terminals gain multi-user diversity by sending data in multiple hops over high SNR links, but this requires the slow process of finding another good path when nodes move too far. Therefore, only the frequency and spatial diversity techniques are useful for control systems.

Finally, current standards have many layers above the PHY and MAC layers. These add tens of bytes of overhead to the desired information, and they are often implemented in software, which increases the latency greatly.

B. Wireless Sensor and Actor Network (WSAN) Protocols

Wireless sensor network (WSN) protocols primarily focus on energy-efficiency, so their latency and reliability performance is strictly best-effort. In recent years, WSNs have been augmented with actuators to form WSANs that can be used for control applications. New protocols were developed to ensure timely and reliable data transport in WSANs, among which Burst and GinMAC are the best examples [14]. They use offline dimensioning and preallocated transmission time slots to obtain a guaranteed latency bound.

However, these protocols rely on time and multi-hop diversity to achieve reliability, which is not available or practical when the cycle time is smaller than the coherence time of the channel. They also rely on multiple rounds of ARQ to achieve reliability. Waiting for ACKs has a high latency overhead, and, if the channel is in a slow fade, it is difficult to recover using only retransmissions and not modifying other system parameters. Finally, these solutions target the MAC layer, but there are additional optimizations that can be made at the PHY layer that affect the latency significantly, such as modifying the code rate.

IV. PRELIMINARY LOW-LATENCY, HIGH-RELIABILITY WIRELESS SYSTEM ARCHITECTURE

Since the network only needs a very basic interaction with other wireless control systems to coordinate resource usage,

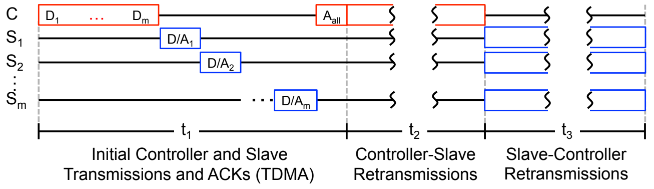
C
S1
S2
Sm
D1 ... Dm
Aall
D/A1
D/A2
D/Am
t1
t2
t3
Initial Controller and Slave Transmissions and ACKs (TDMA)
Controller-Slave Retransmissions
Slave-Controller Retransmissions

Fig. 3. Timing diagram of the proposed architecture with no frequency multiplexing

the network layer can be greatly simplified or combined with the MAC layer and the other higher level layers can be eliminated. This reduces the complexity of the protocol, keeps as much of the protocol in hardware as possible, and reduces the amount of overhead in the transmitted data because headers from the higher layers are eliminated. This leaves only the MAC and PHY layers to discuss, which are co-optimized and have the following key differences from current systems:

- Fixed resource schedule for initial transmissions
- Controller broadcasts initial data and ACK
- Sensors/actuators combine their initial data and ACK
- Fixed retransmission durations
- Very low-rate coding

The specific system described below is just one of many possible implementations that embody the above points.

A. MAC Layer

The proposed MAC assumes the system has a star topology that is known and fixed, which covers a broad range of control systems. The controller (C) is the central node and the sensors and actuators are the slave (S) nodes. This discussion assumes that all slaves have both sensors and actuators on them since it occurs often in real systems.

The operation of the MAC is defined by its operation over one cycle because the data transmission of the control system is periodic. As shown in Fig. 3, the protocol has three phases in every cycle: (1) initial C and S data and ACK transmissions, (2) C to S retransmissions, and (3) S to C retransmissions. Each phase starts at a fixed time within the cycle, so synchronization is necessary for all nodes to have a global sense of time. This consumes extra communication resources, but it is preferable to distributing a schedule each cycle since a node in a deep fade cannot receive its schedule.

In the first phase, each node has an assigned time set during system initialization to send this cycle's instructions or observations. The controller starts by broadcasting all of the instructions to the slaves in one codeword. This increases the blocklength of the slaves' data, which decreases the probability of decoding error. It also packs the data into the minimum number of OFDM symbols, which minimizes the overhead from the cyclic prefixes. Next, in a predetermined order each slave sends its observations as well as an acknowledgment indicating whether it received the controller's data. If a slave does not have an actuator, it can send a fixed value for the ACK; if it does not have a sensor, it can send only the ACK. The controller then broadcasts a block ACK to all of the slaves indicating which slaves need to retransmit. The length of the first phase, t_1, is set by the total time needed to transmit the data and ACKs once from each node.

During the second phase, the controller retransmits data to any slaves that responded with a NAK or whose transmission the controller could not decode. Since only the controller is transmitting, it does not have to worry about collisions. It can simply repeat a single slave's data in one codeword a given number of times based on the SNR to that slave. The slave can use an ARQ or HARQ approach to decoding. Note that there are no ACKs since it would waste time waiting for an ACK that would most likely not be received due to the poor channel (the channel is poor since the first transmission was not received). The length of the second phase, t_2, is fixed during system initialization and is equal to the time needed for the minimum number of controller to slave retransmissions to guarantee the specified reliability (see Section V for more details). This allows the third phase to have a fixed start time.

In the third phase, the sensors retransmit data to the controller if they received a NAK in the block ACK or could not decode it. Since possibly multiple slaves may be retransmitting, they can be preallocated time-frequency resources. Alternatively, they can use CDMA, which allows the controller increase the effective SNR of the received signal since the data is essentially repeated many times. Again, there are no ACKs since they would waste resources. The length of the third phase, t_3, is fixed during system initialization and can be calculated in the same way as for t_2. Therefore, the entire cycle has a deterministic length. Note that if t_2 and t_3 were calculated together, the total cycle time would be smaller. However, this would make the border between the second and third phases random, which would be difficult to implement.

B. PHY Layer

This architecture uses OFDMA in the initial transmission phase to allow multiple slaves to be scheduled at the same time over different frequency subchannels. Alternatively, OFDM can be used for a simpler time-multiplexed implementation, which is shown in Fig. 3. Either OFDMA or CDMA can be used in the retransmission phase depending on the implementation chosen. Note that during retransmissions the SNR to the remaining nodes is small, so it is best to use a scheme that allows maximum ratio combining of retransmissions to boost the effective SNR. Similar to LTE, the controller broadcasts reference and synchronization signals to all slave nodes. The synchronization and channel estimation will need to be more accurate than in LTE because of the fixed schedule, so longer sequences are needed. This requires additional time-frequency resources, which lowers the data rate. This overhead does not scale with the number of nodes since it is broadcast.

Coding is one of the most powerful tools to reduce the number of retransmissions, but it does this at the expense of adding deterministic overhead. Current wireless systems rely on retransmissions to clean up any errors that occur due to using too high rate of a code and too dense of a modulation. Control systems cannot tolerate the number of retransmissions required for this, so they must be able to use lower rate codes with low-order modulations. The proposed architecture uses the optimal code rates for the broadcast frame and individual frames to balance the deterministic overhead and the number of retransmissions. Note that this affects the choice of retransmission scheme chosen in the MAC layer, so the code rate and retransmission scheme should be chosen jointly. The code rate for the individual frame is usually much lower than 1/3. This optimization differs from that in [24] and [25] because the channel realization for the original transmission and for the retransmission are the same since the latency constraint is smaller than the coherence time.

The code rates are fixed during system initialization to avoid the communication needed to adapt the code rate to the channel and because having different code rates would make having a fixed schedule impossible during the initial transmission phase. Having fixed code rates works well if the average SNRs of the nodes are equal, which can be accomplished through CDMA-like power control feedback appended to the data. Since the same system may be used in different conditions, the hardware needs to be flexible so that any rate can be chosen during initialization. Since the information length is constant, two options that are promising are rateless codes and low-rate punctured codes.

Diversity must be extracted in any way possible because reliability is of the utmost importance. Since the latencies are on the order of milliseconds and the coherence time for carrier frequencies and velocities of interest are on the same order or larger, time diversity, achieved through interleaving in time, cannot be used. However, both frequency diversity from sending data across subcarriers separated by more than the coherence bandwidth and spatial diversity from multiple antennas are available to the system. Because the problem for control systems is meeting the reliability requirement, antennas should be used for diversity over multiplexing.

V. Evaluation of Proposed Architecture

A. Industrial Printer Specifications

To evaluate the proposed architecture, a representative application is needed. An industrial printer provides a good model of an ultimate immersive or automotive environment, and it comes with the practically-deployed wired control protocol SERCOSIII [26] that the proposed architecture can be measured against. The printer has 30 moving printing heads that move at speeds up to 3m/s over distances up to 10m. The heads have sensors on board to measure velocity and other state variables, and they have actuators that move them in 3-D space. Every cycle, each sensor transmits 20 bytes to the controller, and each actuator receives 20 bytes from the controller. For these specifications, the SERCOSIII protocol supports the printer's required cycle time of 2ms with a packet error rate (PER) smaller than 10^{-8}. Note that this system does not use the same definition of latency as proposed in Section II, so the cycle time at a small enough PER is given instead. To ease the analysis, the calculation assumes a time-division multiple access approach for sharing channel resources, and the only sources of diversity are from frequency diversity and spatial diversity from multiple antennas.

B. Worst-Case Latency Calculation Methodology

Following is the general procedure to find $l_{\text{worst}}(p)$ with extra details on how it can be done for the industrial printer with the proposed wireless architecture:

1) Define operating conditions and system parameters: This step includes defining the SNR, channel type, the number of sensors and actuators in the system, the number of antennas on each node, the amount of data each node needs to send and receive, the code rates, and the available bandwidth. All of these parameters, except the SNR and coding, are part of the system specifications for a given application and are known ahead of time. Here, the channel is assumed to be Rayleigh, p is 10^{-8}, there are 30 sensors and 30 actuators that each send/receive 20 bytes, each node has 2 antennas that are all used for spatial diversity, and the system has 20MHz bandwidth available. Based on 802.11ac and LTE data rates with QPSK modulation and assuming a 30% overhead for additional reference signals, the raw data rate is set to 24Mb/s. Assuming that the channel gives a frequency diversity of 2 (due to the relatively large bandwidth of 20MHz) and spatial diversity of 4 (due to 2x2 MIMO), the maximum diversity that can be extracted is 8. Due to channel impairments and using broadcast on the downlink, the full diversity usually cannot be achieved, so it is set to 4. For this analysis, the SNR and code rates are fixed. The analysis can be repeated for other combinations of parameters in order to explore the design space or to optimize the system.

2) Define the protocol layers of the system architecture: In the case of the proposed architecture, only the PHY and MAC layers are used, and the details are given in Section IV. Essentially, all details that relate to the timing and duration of transmissions and idle periods must be well-defined. For this example, only time-division duplexing will be used. Also, a TDMA-based preallocated transmission slot scheme is used for the controller to slave and slave to controller retransmission phases. This does not use HARQ techniques to increase the SNR, but instead takes an ARQ approach, which has higher latency but is much simpler to implement.

3) Derive an architecture-specific cycle time equation: Based on Fig. 3 and using a preallocated TDMA retransmission scheme, the cycle time equation for the architecture, is:

$$\begin{aligned}T_{\text{cycle}} = {} & t_{data} \cdot ((m \cdot d_s + d_{CRC})/R_c + t_{CP}) \\ & + t_{data} \cdot ((d_s + d_{CRC} + 1)/R_s + t_{CP}) \cdot m \\ & + t_{data} \cdot ((d_a + d_{CRC})/R_s + t_{CP}) \\ & + t_{data} \cdot ((d_s + d_{CRC})/R_s + t_{CP}) \cdot (N_c + N_s)\end{aligned} \tag{2}$$

where d_s is the number of bits of information each node transmits, d_{CRC} is the number of CRC bits, d_a is the number of bits in the block ACK, R_b is the broadcast code rate, R_i is the individual packet code rate, t_{CP} is the cyclic prefix length, N_c is the number of controller retransmissions, and N_s is the total number of slave retransmissions. The first line corresponds to the controller sending the broadcast frame, the second corresponds to the m sensor nodes sending their data plus ACKs, the third corresponds to the controller sending the block ACK, and the fourth corresponds to the controller and sensor TDMA retransmissions phases.

4) Calculate distributions for the random terms: The distributions for the random terms in (2) must be modeled, which in this case are N_c and N_s. This can be done analytically if possible or by simulation, which was used here. The simulation performed iterations of generating channel realizations for each controller-slave pair based on the parameters from step 1, modeling the links as switches where the probability of the switch being closed is the PER, and then sending codewords across each link until it succeeds. In each iteration, the value of N_c and N_s is recorded, and those values are used at the end of the simulation to calculate an empirical joint probability mass function (PMF) for N_c and N_s.

A key step in the simulation is calculating the PER. Since the codes have shorter blocklengths, the effects of non-ideal codes must be considered. This can be simulated for every code under consideration, but this can be prohibitively slow if the behavior at low error rates is required and many codes are under consideration. Fortunately, there exist bounds on the performance of codes with finite blocklengths [27], and those bounds can yield the PER over an arbitrary channel when rearranged into the following form:

$$\epsilon = Q\left(\left(C - R + \frac{1}{2}\frac{\log(k/R)}{(k/R)}\right)\sqrt{\frac{k}{RV}}\right) \qquad (3)$$

where ϵ is the PER, Q is the tail probability of the standard normal distribution, C is the capacity of the channel, R is the code rate (either R_i or R_b), k is the information length, and V is the dispersion of the channel. Both C and V are known for K parallel AWGN channels and are only a function of SNR [28], and the PER can be found for fading channels by averaging over the fading statistics for a given noise variance.

5) Calculating $l_{worst}(p)$: Using the distributions of the random variables, the PDF or PMF of T_{cycle} can be calculated. Then, $\overline{F}(T_{\text{cycle}})$ can be calculated empirically from the PDF of T_{cycle}, and then it is used as a lookup table to find $l_{\text{worst}}(p)$ using p as the lookup argument. Alternatively, the same procedure can be performed on the controller and slave retransmission phases separately to get their individual worst-case times. This allows each phase to have its own fixed duration, although the cycle time will be longer than if the overall PMF of T_{cycle} is used.

6) Rerun procedure to optimize free variables: This procedure can be rerun for different system parameters, such as R_b and R_i, to find the minimum possible value of $l_{\text{worst}}(p)$ for the fixed parameters, which results in the optimal architecture.

C. Latency in the Industrial Printer Example

The methodology for evaluating the proposed architecture was implemented using Matlab. The finite blocklength PER bound given by (3) was used because the main concern was validating and optimizing the architecture. The bounds provided an effective tool to explore the effects of different code rates on the performance of the system. In fact, the simulation was run for a set of parameters with many different pairs of broadcast and individual packet code rates to find the minimum worst-case latency of the system under those parameters.

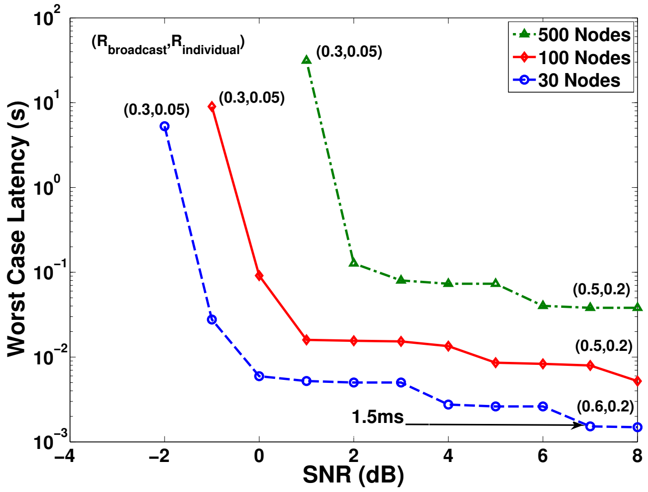

Fig. 4. The proposed architecture's minimum worst-case latency versus SNR for an industrial printer with 30, 100, and 500 sensor/actuator nodes.

After running the methodology, the resulting minimum worst-case latency as a function of SNR for the printer is shown in Fig. 4 along with the optimal code rates for the lowest and highest SNRs. The minimum worst-case latency point is fairly insensitive to the exact value of the code rate pairs, so the simulation grid of the pairs is somewhat coarse. This causes the steps observed in the worst-case latency curve. The latency specifications of the 30 node printer are met at SNRs above 7dB with a code rate of 0.6 for the broadcast codeword and 0.2 for the individual codewords. At lower SNRs, the required code rate decreases in order to reduce the number of retransmissions. Since (3) was used to model the codes, the actual minimum SNR will be several dB larger than 7dB, but this is still within a practical SNR range. The minimum latency is limited by the time to transmit the controller data, the sensor data and ACKs, and the controller block ACK in the initial phase. The latency increases sharply for SNRs below 0dB due to one node being stuck in a deep fade, which occurs because the probability of failure being considered at is 10^{-8}, which is extremely small. Fig. 4 also shows the worst-case latency curves for a printer system with 100 and 500 sensors and actuators. Their worst-case latency at high SNR is larger because the deterministic amount of data to transmit increases linearly with the number of nodes. The worst-case latency goes to infinity sooner for a larger number of nodes because the probability of one node being in a deep fade increases. Fig. 5 provides another view of the system's performance as the number of nodes increases at low, medium, and high SNR.

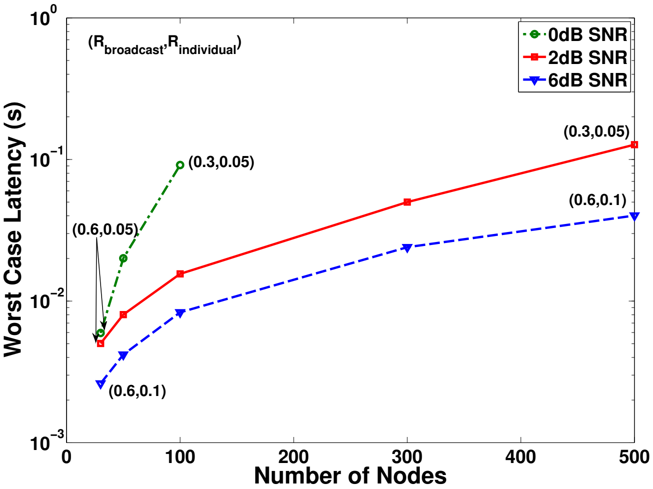

Fig. 5. The proposed architecture's minimum worst-case latency versus the number of nodes at 0dB, 2dB, and 6dB SNR.

VI. Conclusions

Wireless control systems must be judged by their worst-case latency under the constraint of having error-free communication during the lifetime of the system. Current wireless standards do not have worst-case latencies that can support high-performance control systems with latency constraints on the order of milliseconds, such as robotics and computer interfaces. For this reason, a preliminary architecture with a focus on reducing deterministic and random overhead and reducing the number of retransmissions is proposed. It achieves these constraints by having a reliable broadcast for controller to slave data and ACKs, a semi-fixed schedule, and the optimal (low) rate codes.

The main changes to a traditional PHY is using very low to low-rate coding. In the future, very low-rate coding could be a block added to a standard's chipset to support control. The MAC layer differs significantly from other standards, but once designed it can be reused for many different control specifications.

Moving forward, several design choices and optimizations have to be made, such as the coding scheme, diversity mechanisms, and retransmission policy. A practical coding scheme must be chosen that is scalable and implements low-rate codes well. Due to the fixed information size and the need for variable code rates, rateless codes look promising, but their performance relative to using many fixed rate codes needs to be characterized [29]. Another important subject is increasing diversity since deep fades limit the minimum operating SNR of the system. An interesting option to gain diversity is to have the slaves cooperate and thereby exploit multiuser diversity [30]. They can either use a form of relaying or use network coding since they can all listen to each other's transmissions and have a common goal. Finally, a retransmission policy is needed that optimally uses the reserved retransmission resources given the remaining nodes and their channel qualities.

Acknowledgment

This work is supported by the NSF under Grant Numbers CNS-1321155 and ECCS-1343398. The authors thank the BWRC sponsors, Lakshmi Venkatraman, Vivek Jain, Jan Rabaey, Paul Wright, Gireeja Ranade, and Vinayak Nagpal.

References

[1] G. Fettweis and S. Alamouti, "5G: Personal mobile internet beyond what cellular did to telephony," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 140–145, February 2014.
[2] J. Nilsson, "Real-time control systems with delays," Ph.D. dissertation, Lund Institute of Technology, 1998.
[3] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Trans. on Automat. Control*, vol. 55, no. 9, pp. 2030–2042, 2010.
[4] G. Walsh *et al.*, "Stability analysis of networked control systems," *IEEE Trans. on Control Syst. Technology*, vol. 10, no. 3, pp. 438–446, 2002.
[5] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Trans. on Automat. Control*, vol. 56, no. 10, pp. 2456–2461, Oct 2011.
[6] N. Ploplys *et al.*, "Closed-loop control over wireless networks," *IEEE Control Syst.*, vol. 24, no. 3, pp. 58–71, 2004.
[7] J. Song *et al.*, "WirelessHART: Applying wireless technology in real-time industrial process control," in *IEEE Real-Time and Embedded Technology and Applicat. Symp.*, 2008, pp. 377–386.
[8] N. Nikaein and S. Krea, "Latency for real-time machine-to-machine communication in LTE-based system architecture," in *European Wireless Conference*, 2011, pp. 1–6.
[9] V. Gungor *et al.*, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. on Ind. Informatics*, vol. 7, no. 4, pp. 529–539, 2011.
[10] P. Kumar, "New technological vistas for systems and control: The example of wireless networks," *IEEE Control Syst.*, vol. 21, no. 1, pp. 24–37, 2001.
[11] D. Jiang *et al.*, "Principle and performance of semi-persistent scheduling for VoIP in LTE system," in *Int. Conference on Networking and Mobile Computing Wireless Commun.*, 2007, pp. 2861–2864.
[12] K. Zhou *et al.*, "Contention based access for machine-type communications over LTE," in *IEEE Veh. Technology Conference*, 2012, pp. 1–5.
[13] G. Wu *et al.*, "M2M: From mobile to embedded internet," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 36–43, 2011.
[14] P. Suriyachai *et al.*, "A survey of MAC protocols for mission-critical applications in wireless sensor networks," *IEEE. Commun. Surveys and Tutorials*, vol. 14, no. 2, pp. 240–264, 2011.
[15] K. Åström and B. Wittenmark, *Computer Controlled Systems: Theory and Design*. Prentice Hall, 1997.
[16] F. Xia *et al.*, "Cyber-physical control over wireless sensor and actuator networks with packet loss," in *Wireless Networking Based Control*. Springer, 2011, pp. 85–102.
[17] *Wireless LAN MAC and PHY*, IEEE Std. 802.11, 2012.
[18] *Wireless LAN MAC and PHY Amendment 4*, IEEE Draft Standard 802.11ac, Rev. 6.0, 2013.
[19] *E-UTRA; MAC Protocol Specification*, 3GPP TS 36.322, 2013.
[20] *E-UTRA; LTE PHY; General Description*, 3GPP TS 36.201, 2013.
[21] *Low-Rate WPANs Amendment 4*, IEEE Std. 802.15.4, 2013.
[22] *ZigBee 2012*, ZigBee Alliance , 2012.
[23] *WirelessHART*, HART Communications Foundation , 2007.
[24] P. Wu and N. Jindal, "Coding versus ARQ in fading channels: How reliable should the PHY be?" *IEEE Trans. on Commun.*, vol. 59, no. 12, pp. 3363–3374, 2011.
[25] D. Baron *et al.*, "Coding vs. packet retransmission over noisy channels," in *Conference on Inform. Sci. and Syst.*, 2006, pp. 537–541.
[26] (2013, August) Introduction to Sercos III with industrial ethernet. [Online]. Available: http://www.sercos.com/technology/sercos3.htm
[27] Y. Polyanskiy *et al.*, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.
[28] ——, "Dispersion of Gaussian channels," 2009, pp. 2204–2208.
[29] M. Luby, "LT codes," in *IEEE Symp. on Found. of Comput. Sci.*, 2002, pp. 271–280.
[30] J. Laneman, "Cooperative diversity in wireless networks: Algorithms and architectures," Ph.D. dissertation, MIT, 2002.