Wireless Network Design for Distributed Control

Xiangheng Liu and Andrea Goldsmith
Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9515, USA.
`Email: {liuxh,andrea}@wsl.stanford.edu`

***Abstract*— We present a cross-layer framework for the joint design of wireless networks and distributed controllers. The design objective is to optimize the control performance. This control performance is a complex function of the network parameters, such as throughput, packet delay and packet loss probabilities. The goal of optimizing the control performance imposes implicit tradeoffs on the wireless network design as opposed to the explicit tradeoffs typical in wireless data and voice applications. Specifically, the tradeoffs between network throughput, time delay and packet loss probability are intricate and implicit in the control performance index, which complicates network optimization. We show that this optimization requires a cross-layer design framework. We first present this framework for a broad class of distributed control applications. We then illustrate this framework by a cross-layer optimization of the link layer, MAC layer, and sample period selection in an inverted pendulum system. Our results indicate that cross-layer design significantly improves the performance and stability of the controller.**

I. INTRODUCTION

The concept of networked control is universal. Information among distributed sensors, controllers and actuators needs to be exchanged over a communication network to achieve a certain control objective. For example, in Automated Highway Systems [1], vehicles within a certain vicinity need to exchange position, speed, and acceleration profiles to keep a desired distance and avoid collisions. Wireless networks have become increasingly important in such distributed control systems because they allow fully mobile operation, flexible installation, and rapid deployment.

We consider a networked control system as in Figure 1. Multiple control systems coexist and their feedback loops are closed over a shared wireless network. There are many performance tradeoffs associated with the network design. The goal here is to optimize these tradeoffs to achieve the best end-to-end control performance.

Building a distributed control system supported by a wireless network is a challenging task that requires a new design approach to both systems. Control systems and communication networks are typically designed using very different principles. Traditional control theory requires the feedback data to be accurate, timely and lossless. Conversely, random delay and packet loss are generally accepted in communication network design. Moreover, this delay and loss is much more pronounced in wireless networks than in wired networks due to limited spectrum and power, time-varying channel gains, and interference.

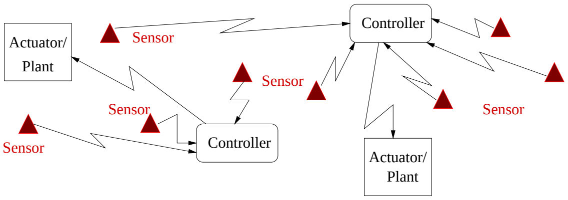


Fig. 1. Wireless Network for Distributed Control

In distributed control systems, the network design objective should be to optimize the control performance. Furthermore, there is a tradeoff between communication and controller performance. From the control perspective, the more the controller knows about the system, the better the control performance is. This can be done by increasing the number of sensors or sending sensor measurements more frequently. However, this increases the communication burden on the network and the network may become congested. The congestion results in longer delays and more packet losses, which degrade the control performance. Therefore, a joint design of the network and controller is necessary.

Joint design of control and communication is two-fold: the controller design needs to be robust and adaptive to the communication faults such as random delays and packet losses, while the network should be designed with the goal of optimizing the end-to-end control performance. Joint design of control and communication has received little attention. An example of such a joint design can be found in [2], where the controller synthesis and communication rate allocation is solved jointly with an iterative method. Our results in this paper are closely related to our previous work [3] [4], where the tradeoffs of the link and MAC layers are discussed separately. Other related work includes [5] [6] [7] [8].

We cast the joint control and communication design problem in a broader framework of cross-layer design. Cross-layer network design has recently been applied to many applications, such as video over wireless [9] and sensor networks with energy constraints [10]. Different aspects of cross-layer design in wireless ad hoc networks are considered in [11] [12] [13]. We use a cross-layer framework for the joint control and communication problem as it allows each layer of the network protocol stack to be optimized relative to the end-to-end controller performance. We will specifically investigate the interaction of the physical layer design, the MAC protocol choice, and the controller sampling period within our cross-layer design

This work is supported by ABB through the Stanford Networking Research Center and by NSF under grant 0120912.

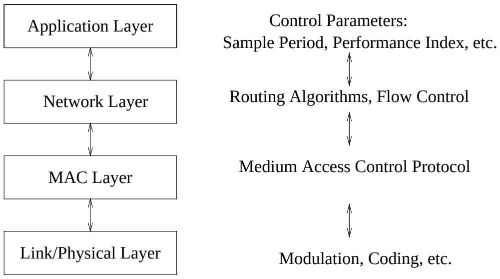


Fig. 2. Layered Structure of Wireless Network

framework.

The remainder of this paper is organized as follows. In the next section, we explain the layered structure of data networks and the cross-layer network design framework for distributed control applications. In Section III we describe our wireless network model. Our control system model and the controller design is discussed in Section IV. In Section V, we illustrate our iterative cross-layer design of the link layer, MAC layer and sample period selection with a double inverted pendulum system. Our conclusions and discussion is given in Section VI.

II. CROSS-LAYER DESIGN FRAMEWORK

The layered architecture is central to data network design. Layering provides the network design with modularity that facilitates standardization and implementation. An international standard of an OSI (Open System Interconnection) model includes seven layers from top to bottom: the Application layer, the Presentation layer, the Session Layer, the Transport layer, the Network Layer, the MAC layer, and the Physical layer. Traditionally, each layer is designed separately with control messages passing between adjacent layers. The idea of cross-layer design is to jointly design these different layers. Cross-layer design can imply a joint design across all network layers simultaneously, which is highly complex. Alternatively, it can entail choosing parameters or protocols at different network layers from existing designs in a joint fashion, which is our design approach. The goal of cross-layer design is to provide the best possible end-to-end performance of the application. Application examples include voice, web browsing, email, etc. In joint control and network design, the application is control.

In this paper, we consider a simpler four-layer architecture as shown in Fig 2. The physical layer defines a point-to-point communication link. The MAC (Medium Access Control) layer defines how the channel is shared among multiple transmitters. The network layer implements routing and flow control for the network. The application layer supports distributed control. Therefore, we consider the control system design as parameters in the application layer.

The goal of the Network, MAC, and Link layers is to optimize control performance. This performance is a complicated function of the packet delay distribution, the probability of packet loss and the data resolution associated with the network. Note that the average delay, which is often used as a performance metric in other wireless systems, is a useless metric for control applications since the closed loop system performance depends on the full delay distribution, not just on the average delay. The link design, the MAC protocol, and the routing algorithm jointly affect the delay distribution and the packet loss probability. The sample period of the control system is considered as a parameter of the application layer. The sample period determines how often new packets are generated and when the old packets are dropped. Thus the sample period affects the network traffic, which in turn affects the delay distribution and packet losses. Therefore, it is important to design the parameters of the MAC, link layers and controller jointly. It is difficult to quantify the impact of a network design on the control performance analytically for all control systems. We use a numerical example with classical inverted pendulum systems to illustrate our framework for cross-layer design and its associated performance gains.

Cross layer network design over all layers is a very challenging problem. In particular, it is difficult to simultaneously optimize all the layers, which motivated the OSI model in the first place. Thus, in this work, we study a suboptimal iterative method of cross-layer optimization over a subset of the network layers: the link layer, the MAC layer and the application layer (sample period selection only). In particular, to jointly design the MAC protocol, the link design and select the optimal sample period, we first fix a sample period and a MAC protocol and choose the best link layer design. For this link design and the sample period, we choose the best MAC protocol. The third step is to optimize the sample period for the chosen link and MAC protocol design. We then iterate the algorithm until it converges. Even though this is a suboptimal algorithm based on just a few protocol parameters, it still yields significant performance gains and insight.

III. THE WIRELESS NETWORK MODEL

A. Wireless Channel Model

We consider a discrete time channel with stationary, ergodic, slowly time-varying gain $\sqrt{g_i(k)}$ and additive white Gaussian noise (AWGN) $n_i(k)$, where the subscript i refers to the i^{th} link and k refers to the k^{th} time instant. In this paper, our analysis will be based on static channel gains. This is justified by the assumption of very slow fading where the channel coherence time (the time over which the channel remains roughly constant) is long enough so that the control system converges to steady-state within a coherence time interval. We assume that the channel power gain $g_i(k)$ is independent of the channel input and the transmission power P_i does not change as the channel gain varies.

B. Wireless Link Model

Different link layer design choices (coding, modulation, etc.) lead to different performance in terms of data rate and probability of error. We assume a simple class of communication link designs as shown in Figure 3. The figure shows the wireless link from a sensor to a controller.

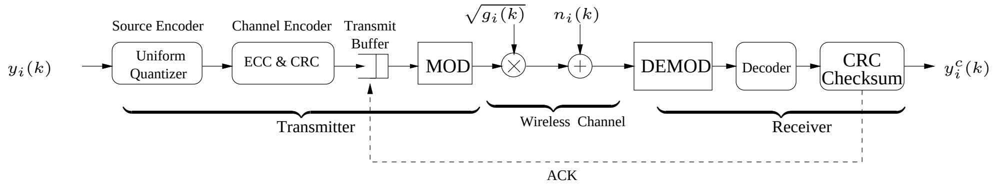


Fig. 3. Wireless Communication Link Model

We assume the same link model for all the wireless links including the links from controllers to the actuators.

Each transmitter is assigned a unique ID number and this ID number is attached to the data (sensor measurement or control command) that needs to be sent. Assume we have M transmitters, then each ID number consists of $\lceil \log_2 M \rceil$ bits. At the transmitter, the data is first quantized and converted into a binary bit stream via a uniform quantizer. The bit stream, piggy-backed by the sender ID number, goes through the channel encoder that uses BCH codes for error correction and then CRC (Cyclic Redundancy Check) for error detection. The effect of undetected errors can be disastrous in control applications since the actuator will use an erroneous control command as the correct one. We use a 16-bit CRC and the probability of undetected errors is roughly 2^{-16}, which is less than 0.01%. We thus ignore the effects of undecoded errors since the probability is negligible. We use MPSK (BPSK or QPSK in our numerical results) modulation at the transmitter. At the receiver, we assume coherent detection of the MPSK signals. The BCH decoder can correct some transmission errors depending on which codebook is used. After error correction, the receiver performs the CRC checksum for error detection. When no error is detected, the receiver may send an "ACK" back to the transmitter. This feedback channel from the receiver to the transmitter is optional and is only required by some MAC protocols. If the transmitter receives the ACK, it clears its transmit buffer and does not transmit until a new packet arrives. We assume time is slotted and we allow retransmissions if there are extra time slots. In our delay analysis, we only consider retransmission delays and assume the processing delay at the transmitter and the receiver to be negligible compared with the delay caused by retransmissions. We also assume the transmit buffer only has a capacity of one data packet. Thus a packet will be discarded[1] if it has not been successfully received by the end of the sample period. Therefore, if a packet is successfully received, the packet delay is bounded by one sample period.

From the control perspective, the relevant communication parameters are data rate, time delay and probability of packet loss. Thus we can simplify the link model as in Figure 4. This simplified model is sufficient to calculate all the communication parameters that may affect the control performance. The data rate is implicit in the covariance of the quantization noise $v_{q,i}$. Both the time delay distribution and the probability of packet loss are determined by the MAC protocols, total number of retransmissions and probability of successful transmission p_s. The probability of successful transmission p_s for each packet can be easily calculated given the link design, wireless channel gain and transmit power.

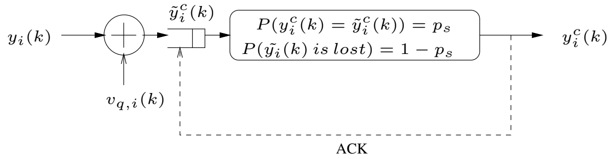


Fig. 4. Simplified Model of the Communication Link

C. MAC protocols

A common transmission scheduling protocol is TDMA (Time Division Multiple Access). TDMA is a collision-free protocol in which time slots are assigned in advance and never changed. We consider fixed TDMA and assume that time slots are divided evenly among all the transmitter/receiver pairs. Since the time slots are pre-assigned, a time slot can be wasted if the pre-assigned transmitter no longer has a packet to send.

Three contention based MAC protocols are discussed: RA (Random Access), RA with ACK and CSMA/CA. We assume no spatial reuse and any two simultaneous transmissions will collide and cause packet losses. With RA, each transmitter attempts to grab the channel independently with a probability of p at any given time slot. This random access scheme works with or without acknowledgment (ACK) from the receiver upon successful transmission. With ACK, the transmitter does not send redundant packets for the information that is already successfully decoded. Hence, the amount of traffic in the system is reduced and the probability of collision is smaller at the cost of added complexity and additional bandwidth. Another class of multiple access protocols uses carrier sensing. Basically, a user that wishes to transmit senses the channel before it attempts to send. Carrier sensing reduces the collisions by avoiding collisions with ongoing transmissions. The medium access control sub-layer of the wireless LAN 802.11 standards uses CSMA/CA (Carrier Sensing Multiple Access/Collision Avoidance). In case of collision, each transmitter will back off for a random period of time before its next attempt. In particular, we consider an exponential back-off algorithm. For simplicity, we assume in this paper that all transmitters are perfectly synchronized. Thus the CSMA/CA is essentially CA only.

---

[1] In a control system, a new measurement is always more valuable than old measurements. In a single hop network, each transmitter only needs to send the newest data available.

IV. CONTROL SYSTEM MODEL AND CONTROLLER DESIGN

A. Control System Assumptions and Performance Measure

We assume all the plants in our model are continuous-time linear time-invariant systems and we can represent the n^{th} system with the following state space equations:

$$\begin{cases} \dot{\mathbf{x}}^{<n>}(t) = & A^{<n>}\mathbf{x}^{<n>}(t) + B_1^{<n>}\mathbf{w}^{<n>}(t) \\ & +B_2^{<n>}\mathbf{u}^{<n>}(t), \\ \mathbf{y}^{<n>}(t) = & C_2^{<n>}\mathbf{x}^{<n>}(t) + \mathbf{v_s}^{<n>}(t). \end{cases}$$

Here $\mathbf{x}^{<n>}(t)$ is the system state, $\mathbf{w}^{<n>}(t)$ is the disturbance acting on the plant, $\mathbf{u}^{<n>}(t)$ is the control force, $\mathbf{y}^{<n>}(t)$ is the measured output, and $\mathbf{v_s}^{<n>}(t)$ is the measurement noise. All boldface variables are vectors.

There are many control performance measures that can be considered and the impact of imperfect communication for different measures can be different. We consider the linear quadratic cost function as our performance measure. Specifically, we want to minimize

$$J_{LQG} = \sum_{n=1}^{M} \lim_{t\to\infty} \mathrm{E}\ \mathbf{x}'^{<n>}(t)Q^{<n>}\mathbf{x}^{<n>}(t)$$

$$+\mathbf{u}'^{<n>}(t)R^{<n>}\mathbf{u}^{<n>}(t),$$

where the weight matrix $Q^{<n>} \geq 0$ and $R^{<n>} > 0$. We can tune the system performance by choosing different $Q^{<n>}$ and $R^{<n>}$. We sometimes refer to this linear quadratic cost function as the generalized H_2 norm (often abbreviated as the H_2 norm) due to their equivalence with proper transformation [14]. Since all the systems have the same state-space representations, we drop the superscript $< n >$ except when needed for clarification.

Different systems sharing the wireless network correlate with each other only through the delay distribution and the packet loss probability induced by the wireless network. If the delay distribution and packet loss probability are known, the analysis for the joint performance measure can be completely decoupled. We use the controller in [16]. This controller is LQG optimal when the delay is bounded by one sample period and there is no packet loss. Since we discard old packets after one sample period, the delay is bounded by one sample period. However, we do have packet losses, so we need to do some approximations in order to use this controller design.

The controller has two cascaded parts: the Kalman filter and the state feedback controller. The Kalman filter calculates the minimum mean square error state estimate based on received sensor measurements. When all the sensor measurements are received, the classical steady state Kalman filter is used. When none of the sensor measurements are received, we can have the Kalman filter run one step forward open loop and this also gives the optimal state estimate. When only part of the sensor measurements are received, it is possible to compute the optimal state estimate but we do not yet know if a steady state solution exists [15]. For simplicity, we treat partial observation loss as complete observation loss in this paper. The state feedback controller is a function of the total time delay in the feedback loop. Thus, it is time varying. The total time delay is from the time when measurements are taken to the time when the actuator updates with the received control command. We assume the controller calculates its control command right before its turn to transmit to the actuator. Therefore, the controller knows the time delay of the control command if the next transmission is successful. Upon receiving a control command, the actuator updates immediately if the control command is calculated based on full observation. Otherwise, the actuator holds the control command until the end of the sample period. When no control command is received at the actuator by the end of the sample period, the actuator continues to use the last received control command.

The closed-loop system is a sampled-data system since it involves both continuous-time and discrete-time dynamics. It was shown in [14], [16] that we can find an equivalent discretized system and design the optimal controller based on the discrete-time MJLS (Markovian Jump Linear System). The closed loop performance can be evaluated based on the MJLS model with a proper Markovian state space.

B. Performance Evaluation

We evaluate the system performance by choosing the right Markovian state and model the closed loop system as a MJLS. Define the augmented system state vector, $\hat{\mathbf{x}}(k) = [\tilde{\mathbf{x}}(k); \hat{\mathbf{x}}(k|k-1); \mathbf{y^c}(k-1); \mathbf{u}(k-1)]$ and the joint noise vector $\hat{\mathbf{w}}(k) = [\tilde{\mathbf{w}}(k); \mathbf{v}(k)]$, where $\mathbf{v}(k) = \mathbf{v_s}(k) + \mathbf{v_q}(k)$. Note that $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{w}}$ are the discretized state and disturbance and $\hat{\mathbf{x}}(k|k-1)$ is the Kalman filter state estimate. We choose the Markovian state $r = (D, s)$ where D is the time delay in the control command and s indicates the sensor measurement loss and/or the control command loss:

$$s = \begin{cases} 0 & \text{no packet losses,} \\ 1 & \text{sensor measurements lost, control received but} \\ & \text{updated without new sensor measurements,} \\ 2 & \text{sensor measurements received, control received} \\ & \text{but updated without new sensor measurements,} \\ 3 & \text{control lost, sensor measurements received} \\ 4 & \text{both control and sensor measurements are lost.} \end{cases}$$

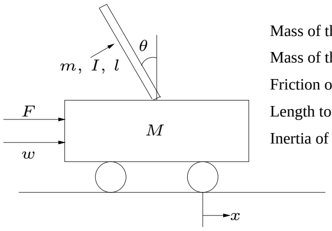
Note that for all $D < T$, where T is the sample period, we always have $s = 0$ while when $D = T$, we can have $s = 0, 1, 2, 3, 4$.[2] Therefore we have $L+4$ Markovian states and we can write the system in the form of a MJLS as $\hat{\mathbf{x}}(k+1) = F_r\hat{\mathbf{x}}(k) + G_r\hat{\mathbf{w}}(k)$ for $r = 1, 2, \cdots, L+4$. The system matrices F_r, G_r can be easily derived. Let $P(k) = \mathrm{E}\ \hat{\mathbf{x}}(k)\hat{\mathbf{x}}(k)'$, then

$$P(k+1) = \sum_{r=1}^{L+4} q_r F_r P(k) F_r' + \sum_{r=1}^{L+4} q_r G_r G_r',$$

where q_r is the probability that the MJLS is in state r. As $k \to \infty$, it can be shown [16] that a unique steady-state covariance matrix $P = \lim_{k\to\infty} P(k)$ exists when the

---

[2] The delay distribution that we use to calculate the state feedback controller is exactly the distribution of D. Note $Pr(D = T)$ sums up five different probabilities.



Mass of the cart: $M = 0.5$ kg
Mass of the pendulum $m = 0.2$ kg
Friction of the cart $b = 0.1$ N/m/sec
Length to pendulum center of mass $l = 0.3$ m
Inertia of the pendulum $I = 0.006 \text{ kg} * \text{m}^2$

Fig. 5. Inverted Pendulum and Cart

recursion is stable. We can now evaluate the linear quadratic cost function since $J_{LQG} = \text{Trace}\left(\begin{bmatrix} Q & 0 & 0 & 0 \end{bmatrix} P\right) + \text{Trace}\left(\begin{bmatrix} 0 & 0 & 0 & R \end{bmatrix} P\right)$.

V. NUMERICAL EXAMPLE

The cart with an inverted pendulum, shown in Figure 5, is controlled with a force, F, to cancel the random disturbance w and maintain the pendulum in an upright position. We use x to denote the cart position coordinate and θ as the pendulum angle from vertical.

For this example, we assume two identical inverted pendulum plants with the parameter choices as listed in Figure 5. The state of the system is chosen as $\left[x(t), \dot{x}(t), \theta(t), \dot{\theta}(t)\right]$. The system dynamics are not linear in θ. We assume the pendulum does not move more than a few degrees away from the vertical and linearize the system dynamics about $\theta = 0$. We can thus get the standard linear model for the inverted pendulum. We would like to minimize the linear quadratic cost function with $Q^{<1>}(1,1) = 1.5, Q^{<1>}(3,3) = 2.5, R^{<1>} = 1,$ and $Q^{<2>}(1,1) = 1, Q^{<2>}(3,3) = 100, R^{<2>} = 1$, where all the unspecified elements in $Q^{<i>}$ are zero. These weight matrices $Q^{<1>}, Q^{<2>}$ and $R^{<1>}, R^{<2>}$ are chosen to reflect the different priorities of different systems and different signals. In this example, the second system weighs more on θ since its main goal is to keep θ small while the first system gives roughly equal emphasis on x and θ. The measurement noise $\mathbf{v_s}(k)$ is assumed to be Gaussian with zero mean and covariance matrix $\Sigma_s = [10^{-4}, 0; 0, 10^{-6}]$.

A. Link Layer Resource Allocation and Design Tradeoffs

We first show how different link designs affect the control performance. We assume TDMA in which the transmission order repeats as sensor measurement 1, sensor measurement 2, control command for system 1 and then the same order for system 2. Since we have 6 transmitter/receiver pairs, the ID field for each packet is 3 bits. We also require a minimum of 4 bits to represent each measurement/control command and approximate the quantization noise as a Gaussian random variable. We use two modulation schemes: BPSK and QPSK. QPSK provides twice the data rate of BPSK but QPSK incurs a larger probability of bit error for a fixed transmission power and bandwidth. We consider three different frame sizes: 24 bits, 32 bits and 48 bits. Note that there is a 16 bit CRC in each frame and we also use one bit guard time between each transmission. We use BCH codes for error correction. With a 32-bit frame, we can use (15, 11) and (15, 7) codes, where the first number is the total number of coded bits in the codeword and the second is the number of information bits. The code rate is defined as the fraction of the number of message bits and the total number of bits in the code. With a 48-bit frame, we can use (31, 26), (31, 21), (31, 16) and (31, 11) codes. We also consider the cases where no error correction coding is used. We represent these cases by (7, 7), (15, 15) and (31, 31) for 24-bit, 32-bit and 48-bit frames respectively. In this example, we assume 96 Ksymbols/sec, thus the data rate is 96 Kbps for BPSK and 192 Kbps for QPSK. The transmission power is 10 mW and the noise density is $\frac{N_0}{2} = 10^{-8}\ W/Hz$. We consider a static channel gain $g = 0.2$.

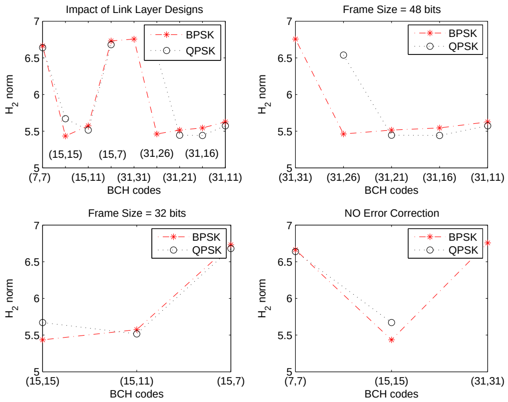


Fig. 6. TDMA with Different Link Layer Designs

In Figure 6, we plot the generalized H_2 norm against different BCH codes for both BPSK and QPSK. When a data point is missing from the plot, e.g., QPSK with (31,31), it means one or both of the pendulum systems are unstable for the particular link design. The first graph plots the performance of 18 different link designs. This is a little hard to read so we illustrate the results with 3 sub-plots. The second and third graph plot the performance under a fixed frame size of 48 bits and 32 bits respectively. For a given frame size, QPSK allows twice as many time slots as BPSK and the transmission time of each frame is only half of the BPSK transmission time due to the doubled data rate. However, QPSK incurs a higher probability of bit error, which in turn leads to higher probability of frame error. The probability of frame error can be reduced if a strong error correction code is used. As we see from the second plot, QPSK performs better when the code rate is low since more transmission errors can be corrected. Both the second and the third plot show that BPSK performs better than QPSK when the code rate is high. The last plot in the figure shows the performance comparison for different frame sizes when no error correction is used. Smaller frame sizes lead to more time slots and smaller probability of frame error when no error correction is used. Therefore, less delay and packet losses can be expected. However, the impact of data resolution kicks in. For small frame sizes, we have few bits to represent the signal. Thus the quantization noise is big. This is why (15, 15) outperforms (7,7) for BPSK and QPSK. It is surprising that the overall best performer is BPSK with 32-bit frames and no error correction coding, even though

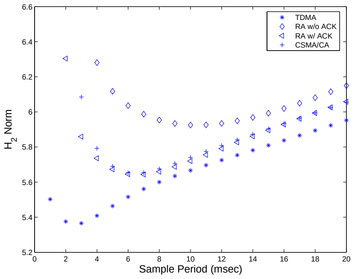


Fig. 7. Performance Comparison of Different Multiple Access Schemes

several other link designs perform only slightly worse. The reason is that this link design achieves the best tradeoff among the data resolution, time delay and probability of packet loss in our control performance measure.

B. MAC Layer Design Tradeoffs

We assume a fixed link design to compare different MAC protocols. We use QPSK with 32-bit frames. The (15, 11) BCH codes are used to correct 1-bit transmission error and a 16-bit CRC is used for error detection. In random access, we assume each transmitter attempts to grab the channel with probability $p = \frac{1}{6}$ and this maximizes the probability of successful channel access at the beginning of each sample period. For CSMA/CA, we assume the minimum collision window $CW_{min} = 3$. In the numerical example, the delay distribution and packet loss probability are obtained through simulation.

Figure 7 compares the joint linear quadratic cost function of the two identical inverted pendulum systems sharing a wireless network with different multiple access schemes: TDMA, Random Access without ACK, Random Access with ACK and CSMA/CA. TDMA is collision free while the others are random access algorithms and collisions cannot be avoided since there is no centralized control. The figure shows all three contention based access schemes lead to performance degradation compared to TDMA. Random Access without ACK has the worst performance. This is due to collisions. Both Random Access with ACK and CSMA/CA try to reduce the amount of collisions. This is why their performance is significantly better than Random Access without ACK. The performance of Random Access with ACK and CSMA/CA are comparable and the performance depends on the communication parameters chosen. For CSMA/CA, as we increase CW_{min}, which is the minimum collision window, the control performance first improves, then degrades. The performance first improves due to reduced collision. The performance degrades when CW_{min} is so large that the transmitter tends to wait too long for the second attempt after a collision occurs. This figure also shows that the control system design should depend on the wireless network design. For example, the optimal sample period selection should be different based on what multiple access scheme is used by the wireless network. Faster sampling is not necessarily better with an imperfect communication network. As a matter of fact, very small sample periods may lead to system instability due to large probability of packet loss. On the other hand, too long sample periods are always bad due to the latency.

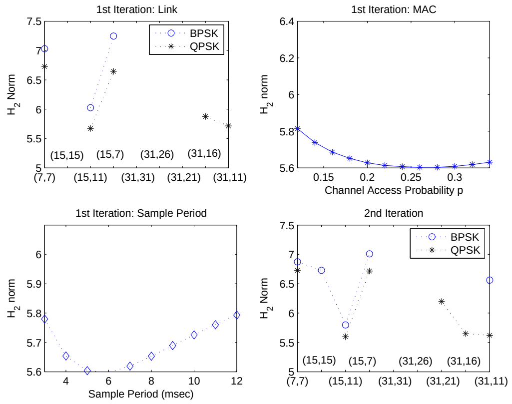


Fig. 8. Cross Layer Design: the Link, MAC and Application Layer

C. Cross Layer Design

In Figure 6, we see that for TDMA with a 6 msec sample period, the 32-bit frame with no error correction and BPSK modulation has the best performance. However, if we keep the same link design and use a contention based medium access protocol from the previous subsection, both pendulum systems go unstable. Therefore, the link layer and the MAC layer should be jointly designed.

With Figure 8, we illustrate the procedure of an iterative cross-layer design for the link layer, the MAC protocol, and the sample period. Our link design parameters are the modulation scheme, the frame size, and the error correction coding. We use RA with ACK as the MAC protocol and we vary the channel access probability p in order to optimize the performance. We assume all senders have an equal channel access probability and use the same link design for simplicity. The application layer parameter is the sample period. We start with an initial sample period of 5 msec and the channel access probability $p = \frac{1}{6}$. The first plot in Figure 8 shows the control performance as a function of different link designs. There are 18 data points on this graph but only 8 are visible. That is because the other 10 link designs lead to system instability and thus the H_2 norm is infinite. In this example, all links with a 48-bit frame size when BPSK is used make the system unstable. This is because collisions lead to a large probability of packet loss if the number of retransmissions is small, even with very reliable links. The optimal link design is QPSK and $(15, 11)$ BCH codes in 32-bit frames. We keep this link design and optimize the MAC protocol within the class of RA with ACK for the initial sample period of 5 msec. The second plot shows the H_2 norm as the function of the channel access probability in RA with ACK. The control performance first improves (H_2 norm decreases) and then degrades as the channel access probability increases. When the senders access the channel with small probabilities, the channel is mostly idle but senders are not transmitting at rates they need to clear their buffers. Thus the packet delay is long. On the other hand, large access probabilities lead to collisions, which also cause packet delay and losses. The optimal choice is $p = 0.28$. The third step is to choose the optimal sample period now that we have updated our choice of both the MAC protocol and the link design. The third plot compares the system performance as the sample period varies. The optimal sample period is $T = 6$ msec. Then we go back to the first step. The fourth plot shows the control performance versus different link designs for $T = 6$ and $p = 0.28$. Again, QPSK with (15,11) is the best link design. The algorithm converges in the next step when $p = 0.28$ is again optimal within the class of RA with ACK protocols. The performance gain from this simple example may not seem dramatic since we have a good initial guess. However, if we had designed these layers separately, we could have chosen a reliable link design with BPSK, 48-bit frames and strong error correction coding, which may lead to system instability. We also could have chosen too small a sample period for the system to stay stable when RA with ACK is used as the medium access protocol.

VI. CONCLUSIONS AND DISCUSSIONS

We propose a cross-layer framework for joint design of distributed control and wireless networks. The network design goal is to optimize the control performance, which is an implicit function of the network performance. Similarly, control design choices impact network performance, which in turn impacts controller performance. Thus a joint design of control and communication is necessary. Cross-layer design provides a broad framework where each layer of the network protocol stack, including the controller design, can be optimized relative to the end-to-end performance.

We consider an iterative cross-layer design over a subset of the network layers. We show with a numerical example that such an iterative design gives substantial performance gains while the control system may go unstable if we design the network layers separately. We also uncover some surprising insights. In particular, we show that an uncoded link design, which is often undesirable due to its unreliability, can be optimal under certain circumstances since it achieves the optimal tradeoff among data resolution, time delay and packet loss probability. Note that an iterative design is only suboptimal. A true joint design over all the network layers should give more significant performance gains.

The goal of cross-layer design is to provide the best end-to-end performance of the application. In a distributed control system, the control system is the application of the network. Thus cross-layer design also includes designing control algorithms that are adaptive and robust to the network performance. This area of control design has recently caught some momentum (e.g. [17]). Yet most studies have assumed unrealistic assumptions on the network. Authors often consider only one aspect of the network faults: random packet dropping, random delay, or data rate limitation. To jointly design control and communication, all these communication faults need to be dealt with in an integrated fashion and under more realistic scenarios. For example, bursty packet losses should be considered instead of the common assumption of independent packet losses.

The problem becomes even more difficult when we have a fast fading channel, where the performance provided by the wireless links is time varying. In a fast fading environment, the probabilistic performance provided by the network is no longer stationary. There is a lack of theory in evaluating and designing such systems. A network control system on the move, such as the Automated Highway System, needs to take the time-varying channel into account. Adaptive link layer techniques are well known. In a cross-layer design, the MAC protocol, the routing algorithm, and the controller designs must adapt to the channel states as well. This area is just beginning to be explored. An important question to ask in this adaptive cross-layer design is what parameters shall be shared among different layers of the network and how each layer can be made robust to changing network conditions.

REFERENCES

[1] P. Varaiya, Smart Cars on Smart Roads: Problems of Control, *IEEE Trans. on Autom. Control*, pp. 195-207, Feb. 1993.

[2] L. Xiao, et. al. Joint Optimization of Communication Rates and Linear Systems, *IEEE Trans. on Autom. Control*, pp. 148-153, Jan. 2003.

[3] X. Liu and A. Goldsmith, Wireless Communication tradeoffs in Distributed Control, *Proc. IEEE Conf. on Decision and Control*, pp. 688-694, Dec. 2003.

[4] X. Liu and A. Goldsmith, Wireless Medium Access Control in Networked Control Systems, *Proc. IEEE American Control Conference*, June 2004.

[5] M.S. Branicky, et. al. Scheduling and feedback co-design for networked control systems, *Proc. IEEE Conf. on Decision and Control*, pp. 1211-1217, Dec. 2002.

[6] F. Lian, et. al. Network Design Consideration for Distributed Control Systems, *IEEE Trans. on Control Systems Technology*, pp. 297-307, Mar. 2002.

[7] F. Lian, et. al. Design of Sampling and Transmission Rate for Achieving Control and Communication Performance in Networked Agent Systems, *Proc. IEEE American Control Conference*, pp. 3329-3334, June 2003.

[8] G.C. Walsh, O. Beldiman, L. Bushnell, Error Encoding Algorithms for Networked Control Systems, *Proc. IEEE Conf. on Decision and Control*, pp. 4933-4938, Dec. 1999.

[9] T. Yoo, et. al. Cross-layer Design for Video Streaming over Wireless Ad hoc Networks *To Appear, Proc. IEEE International Workshop on Multimedia Signal Processing*, Sept. 2004.

[10] S. Cui, et. al. Cross-layer Optimization in TDMA-based Sensor Networks with Energy Constraints, *To Appear, submitted to IEEE Trans. Wireless Communications*, 2004.

[11] S. Shakkottai, et. al. Cross-layer design for wireless networks, *IEEE Commun. Mag.*, pp. 74-80, Oct. 2003. 2003.

[12] T. ElBatt, A. Ephremides, Joint Scheduling and Power Control for Wireless Ad Hoc Networks, *IEEE Trans. Wireless Communications*, pp. 74-85, Jan. 2004.

[13] T. Lang, et. al. Multipacket reception in random access wireless networks: from signal processing to optimal medium access control, *IEEE Commun. Mag.*, pp. 108-112, Nov. 2001.

[14] T. Chen and B. Francis, Optimal Sampled-Data Control Systems, *Springer*, 1995.

[15] X. Liu and A. Goldsmith, Kalman Filter with Partial Observation Losses, *To Appear, Proc. IEEE Conf. on Decision and Control*, Dec. 2004.

[16] J. Nilsson, Real-Time Control Systems with Delays, *Ph.D. Thesis*, Lund Institute of Technology, 1998.

[17] P. Seiler and R. Sengupta, Analysis of Communication Losses in Vehicle Control Problems, *Proc. IEEE American Control Conference*, pp. 1491-1496, June 2001.