Wireless Model Based Predictive Networked Control System ⋆

Alphan Ulusoy * **Ahmet Onat** * **Ozgur Gurbuz** *

* *Sabanci University, Faculty of Engineering and Natural Sciences, Istanbul 34956, Turkey, (e-mail: alphan@su.sabanciuniv.edu, onat@sabanciuniv.edu, ogurbuz@sabanciuniv.edu).*

Abstract: Owing to their distributed architecture, networked control systems are proven to be feasible in scenarios where a spatially distributed control system is required. Traditionally, such networked control systems operate over real-time wired networks over which sensors, controllers and actuators interact with each other. Recently, in order to achieve the utmost flexibility, scalability, ease of deployment and maintainability, wireless networks such as IEEE 802.11 LANs are being preferred over dedicated wired networks. However, basic networked control systems cannot operate over general purpose wireless networks since the stability of the system is compromised due to unbounded delays and unpredictable packet losses that are typical in the wireless medium. This work proposes a novel wireless networked control system that can achieve decent control even under unbounded delay, bursts of packet loss and ambient wireless traffic. Ambient wireless traffic is handled with modified 802.11b medium access control parameters providing the proposed system with a greater medium access priority. Packet deadlines defined at each node of the system reduce unbounded packet delay to packet loss. Performance degradation due to packet loss is kept at a minimum using the predicted plant states and control signals.
The proposed system is implemented and thoroughly evaluated on a dedicated test platform under numerous scenarios. Results of the experiments show that the proposed system can work under bursts of packet loss and ambient wireless traffic levels which are intolerable for basic networked control systems while not being hindered by restraining assumptions of existing methods.

Keywords: wireless networked control systems, model predictive control, computer controlled systems.

1. INTRODUCTION

Networked Control Systems (NCS) where the components of a distributed feedback control system communicate over a network are composed of three types of computer nodes carrying out specific tasks. In an NCS, sensor nodes are responsible for periodically measuring plant outputs and communicating this data to controller nodes over the network. Controller nodes use plant outputs to calculate the control signals and communicate them to actuator nodes. Ultimately, actuator nodes apply the control signals to the plant.

NCS's are very appealing solutions to scenarios where the nodes of a control system have to be distributed spatially by design. However, to ensure the proper operation of an NCS, all latency components that make up its total latency must be bounded. Without losing generality, end to end latency of an NCS can be broken down to 5 main components: internal latencies of the sensor, controller, actuator nodes, sensor to controller communication latency and controller to actuator communication latency. Out of these delay components, those that refer to internal latencies of the nodes include all computational and functional latencies that a particular node introduces and are bounded for a hard real-time system. Remaining delay components refer to the latencies induced by the communication medium of choice and are not bounded for regular networks where medium access is based on contention and random back-off times. Thus, basic NCS's (b-NCS) require dedicated real-time networks since they can not perform satisfactorily over a regular network due to the deteriorating effect of unbounded delays and packet losses. However, the overhead of installing a dedicated network often hinders the commissioning of the control system and discourages its use.

In order to overcome the problem of unpredictable delays and loss that the data packets of an NCS are subject to when operating over a regular network, Wen et al. (2007) analyze the effects of the network on the control system and propose to take the characteristics of the network into consideration during the design of the control system. Similarly Kato and Ohnishi (2005) propose to compensate the disturbance force from transmission time delay with the disturbance observer included in the model of the control system. Despite the successes reported in these works, making the communication medium an integral part of the actual controller being designed may not be a good design practice since the underlying network and the control system operating on it are two distinct entities

⋆ This work was supported in part by the Scientific and Technological Research Council of Turkey, project code 106E155.

and the traffic load and the delay of the communication medium can change during operation of the NCS. On the other hand, model predictive controllers are used in similar scenarios as given by Liu et al. (2004) and Rawlings (2000) but they either do not take the synchronization between the nodes into account or are not set up to be networked control systems due to the fact that they rely on a direct-link between between the sensor and controller and a transmission failure would inhibit future predictions.

As a remedy to addressed problems, Model Based Predictive Networked Control System (MBPNCS) proposed by Onat et al. (2008) improves the performance of a basic NCS under variable time delays and packet losses by assuming standard NCS architecture with no requirement of direct links and a priori knowledge of the reference signal. MBPNCS, which operates over an ethernet LAN, employs a model based predictive controller which utilizes a model of the plant to predict control signals into the future and is shown to provide resilience against packet losses. However, the level of immunity MBPNCS provides against packet losses is only tested with a uniform packet loss model where probability of losing a packet during transmission is determined by a pseudo-random number generator with a uniform distribution which is not representative of true channel characteristics since packet losses are generally correlated and largely occur in bursts. Additionally, no experiments have been performed regarding the extent to which the traffic generated by other nodes on the network degrades the performance of the system.

Nevertheless, a truly flexible NCS requires wireless communication since it may well be the case that the nodes of the NCS have to be placed such that dedicated cabling for communication is not preferred or simply is not an option. However, when the nodes of the control system are distributed and the communication medium is air, transmission failures and delays owing to re-transmission attempts are no longer inconsequential as they are in the case of a reliable field-bus or some other dedicated and guarded wired communication medium. Thus, packet losses and unbounded delays occurring during wireless communication have to be rigorously evaluated and various counter-measures have to be taken during the design phase of a wireless NCS (W-NCS). Additionally, a desirable W-NCS implementation would be the one which achieves satisfactory performance without deteriorating the architectural robustness of the overall system by introducing unnecessary and undesired coupling between functionally independent blocks of the system, namely the communication block and the control block. Such an approach would make the transmission medium a part of the control system, which is not desirable, and would severely undermine the flexibility and adaptability of the overall system.

In this work, a novel wireless MBPNCS (W-MBPNCS) is presented which overcomes the limitations of traditional cabled NCS's such as lack of mobility and need for dedicated infrastructure by operating over an 802.11b wireless ad-hoc network formed between its nodes. High medium access latencies induced by ambient wireless traffic is handled with a modification of 802.11b medium access control (MAC) parameters, giving W-MBPNCS a greater priority in the presence of other nodes contending to access the medium. Unbounded packet latency is reduced to packet loss by introducing relative packet deadlines at each node of the system after which packets are assumed to be lost. The effect of packet loss is minimized by the estimations of the model based predictive controller. The implemented W-MBPNCS is throughly tested in demanding realistic scenarios and is shown to be resilient against packet losses, delays and ambient wireless traffic.

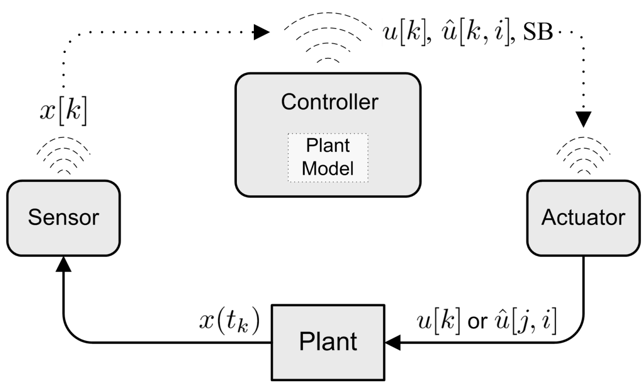

Fig. 1. Overall architecture of W-MBPNCS.

2. SYSTEM ARCHITECTURE AND BACKGROUND

In this section, an overview of the overall architecture is presented followed by the details of wireless access, the channel model and the control algorithm used to evaluate the performance of W-MBPNCS.

W-MBPNCS is made up of 4 components as given in Fig. 1: the sensor node, the controller node which also contains the model $\hat{P}$ of the plant, the actuator node and the actual plant P. During the operation of W-MBPNCS, the sensor periodically reads plant outputs and communicates this data to the controller over the ad-hoc wireless network. In addition to calculating the control signal, the controller also predicts an additional number of control signals into the future using $\hat{P}$. Upon retrieval of controller packets, the actuator applies appropriate control signals to the plant. The functionalities of the controller and the actuator of W-MBPNCS are discussed in greater detail in Section 3.

2.1 Wireless Access

IEEE 802.11b uses a contention based medium access mechanism called *distributed coordination function* (DCF). DCF is responsible for avoiding collisions and resolving them when they occur as multiple wireless stations (STA) try to transmit simultaneously. Fig.2 by IEEE 802.11 Working Group (2007) illustrates the basics of DCF.

Functionality of DCF primarily depends on 4 key parameters: DCF interfame space ($DIFS$), contention window (CW) and CW's minimum and maximum bounds (CW_{min} and CW_{max}). In a nutshell, DCF works as follows: Each STA has a backoff timer which is loaded according to (1) whenever the medium is found to be busy. `Random()` is a pseudo-random integer from a uniform distribution over the interval $[0, CW]$. Initially CW equals

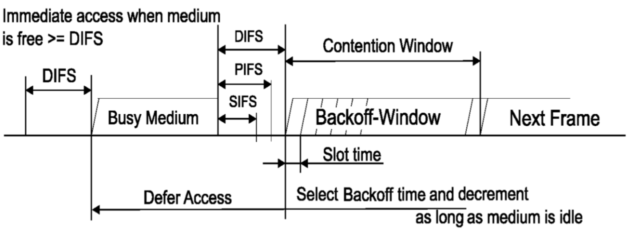

Fig. 2. DCF Basic Access Method in IEEE 802.11b (IEEE 802.11 Working Group, 2007)

CW_{min}. After each retry due to transmission failure CW is updated according to (2) until it reaches CW_{max}.

$$Backoff\_Time = Random() \times slot\_time \quad (1)$$

$$CW = 2^{retries} - 1 \quad (2)$$

Using DCF, a STA attempts to transmit only if it thinks the channel is clear. At the beginning of each transmission attempt, each STA senses the channel for an interval determined by the DIFS parameter. If the medium is not free throughout DIFS, the STA defers transmission until the next time the channel is free. Otherwise, the STA waits for an additional amount of time determined by its backoff timer. If the channel remains free until its backoff timer expires, the STA begins transmission. If not, the STA defers transmission until the next time the channel is free. Each successful transmission is concluded with an acknowledgment (ACK), thus absence of ACK indicates that a collision might have occurred. Each STA whose packet is lost due to collision (contention) updates its CW parameter and reloads its backoff timer according to (1) and (2) before trying to transmit again. This scheme is repeated until all contentions are resolved and all STAs receive their ACK messages.

2.2 Channel Model

Wireless communication is unreliable when compared to wired communication and suffers from mainly three issues as given by Pahlavan and Krishnamurthy (2002): multipath fading, shadow fading and fast fading. *Multipath fading* is caused by the interference between the *out-of-phase* arriving copies of the same signal at the receiver resulting in irreducible error rates. *Shadow fading* occurs due to obstacles which block signals from arriving a wireless station and causes variations in the received signal strength. *Fast fading* occurs when channel characteristics change faster than the delay constraint of the channel. Since these issues are closely coupled to the surroundings of a wireless STA, characteristics of the wireless channel remain correlated for some time after a change. Thus, transmission errors and packet losses on the wireless channel occur in bursts followed by practically error-free periods rather than occurring completely randomly.

In order to model the bursty packet loss characteristics of the wireless channel, this work uses a Gilbert/Elliot model (Elliot, 1963). The model is composed of two states, a *good* state and a *bad* state as shown in Fig.3 which determine the characteristics of the channel at a given time. In *good* and *bad* states of the channel, packets are lost according to packet loss probabilities P^g_{loss} and P^b_{loss} respectively. The next state of the channel is determined

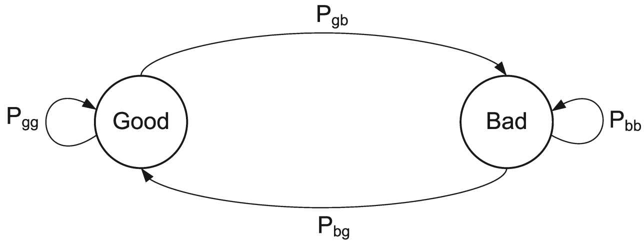

Fig. 3. Gilbert/Elliot Model

Table 1. Gilbert/Elliot Model Parameters

P_{gb}	P_{bg}	P^g_{loss}	P^b_{loss}
0.0196	0.282	0	1

according to state transition probabilities P_{gb} and P_{bg} after each packet. Since state transition probabilities are typically small, the channel state remains unchanged for some time after a transition is taken imitating bursts of packet loss when the model is in the *bad* state and periods of almost error free transmission when the model is in the *good* state. The parameters of the employed model (Tb.1) are derived from the results presented by Willig et al. (2002) which were determined from realistic test-cases in an industrial setting. Since this work uses the same mode of transmission (2 Mbps with QPSK) measurement results presented there can be directly used here.

As a means for comparison, a much simpler *uniform packet loss model* is also implemented where packets are lost with a predetermined probability with no correlation. A packet's fate is determined by comparing a random number to a predetermined threshold value. If the random number is greater, the packet is sent, if not it is dropped deliberately.

2.3 The Plant and The Control Algorithm

There is a myriad of plants and control approaches that can be used to evaluate the performance of a discrete time control system. However, in order to ease the design task of the controller a linear approximation of the plant can be used as given in (3) to employ a state feedback approach as given in (4) where k is the sample index, u is the control signal, K is the controller gain matrix, r is the reference, G_r is the gain of reference and x is the plant state.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (3)$$

$$u[k] = G_r r[k] - Kx[k] \quad (4)$$

Additionally, if some of the state variables can not be measured directly, an observer can be used to estimate them as given in (5) where $\bar{A}$, $\bar{B}$, $\bar{C}$, $\bar{D}$ are discretized versions of A, B, C, D matrices, $\hat{x}$ is the estimated plant state and K_o is the observer gain matrix.

$$\begin{aligned} \hat{x}[k] = &\bar{A}\hat{x}[k-1] + \bar{B}u[k-1] \\ &+ K_o(y[k] - \bar{C}(\bar{A}\hat{x}[k-1] + \bar{B}u[k-1])) \end{aligned} \quad (5)$$

To evaluate the performance of W-MBPNCS, this work aims position control of a Maxon RE-35 DC-motor with the following linear approximation:

$$
\begin{aligned}
A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -b/J & K_t/J \\ 0 & -K_v/L & -R/L \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} \\
C &= [1\ 0\ 0], D = [0] \\
x &= \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix}
\end{aligned} \tag{6}
$$

where b is the friction coefficient, J is the rotor moment of inertia, K_t is the torque constant, K_v is the speed constant, L is the terminal inductance, R is the terminal resistance, θ is the position, $\dot{\theta}$ is the speed and i is the current of the motor. Once the relevant parameters of the plant are obtained and a discrete-time model is prepared for 100Hz sampling rate, a state-feedback controller for position control is implemented as given in (4).

In order to estimate the speed and the current of the motor an observer is used as given in (5). By selecting the observer gain K_o such that $\bar{C}K_o = I$ as given by Astrom and Wittenmark (1997), a reduced order Luenberger observer for the plant can be obtained as given in (7)

$$
\begin{aligned}
x_1[k] &= y[k] \\
\hat{x}_2[k] &= \bar{A}_{21}x_1[k-1] + \bar{A}_{22}\hat{x}_2[k-1] + \bar{A}_{23}\hat{x}_3[k-1] \\
&\quad + \bar{B}_2 u[k-1] \\
\hat{x}_3[k] &= \bar{A}_{31}x_1[k-1] + \bar{A}_{32}\hat{x}_2[k-1] + \bar{A}_{33}\hat{x}_3[k-1] \\
&\quad + \bar{B}_3 u[k-1]
\end{aligned} \tag{7}
$$

where x_1 is the measured motor position (θ), x_2 is the estimated motor speed($\hat{\dot{\theta}}$) and $\hat{x}_3$ is the estimated motor current($\hat{i}$).

3. WIRELESS MODEL BASED PREDICTIVE NETWORKED CONTROL SYSTEM

In this section an in-depth analysis of the implementation of W-MBPNCS is provided, discussing the resilience W-MBPNCS provides against unbounded packet delays, packet losses and ambient wireless traffic.

3.1 Resilience Against Unbounded Packet Delays and Packet Losses

Per-node Relative Packet Deadlines The sensor samples, appends a time stamp to and transmits the plant outputs at a period of $T = 1/f_s$ to the controller. The controller starts operating a certain amount of time after the sensor, $T/10$ in this case, introducing a relative deadline for sensor packets. This is achieved as follows: the controller listens for the first n sensor packets, 10 in this case, and finds the latency of the 1^{st} packet by evaluating the arrival times of the following $n-1$ packets. Then, this information is used to accurately calculate the time the controller must wait before beginning execution. This way, the controller can initialize with the correct relative packet deadline even if the 1^{st} packet arrives late. The number of packets to listen for prior to initialization and the relative packet deadline is chosen based on the quality of the wireless link. Following initialization, the controller node checks the time stamps of the incoming packets and ignores any packets that fail to meet their deadlines, effectively reducing the unbounded packet latency to packet loss. The initialization and the time stamp mechanisms also work in the same way between the controller and the actuator.

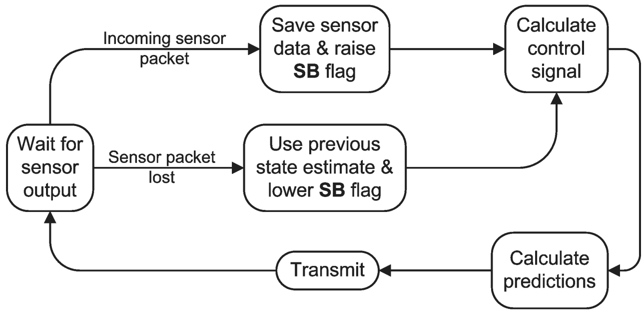

Fig. 4. Operation of the controller node.

Model Based Predictive Controller As given by Onat et al. (2008), in addition to calculating the control signal for the current time step, $u[k]$, the controller node also predicts n future control signal estimates ($\hat{u}[k,i], \{i : [1,n]\}$) using the model of the plant $\hat{P}$ iteratively which predicts future plant output estimates ($\hat{y}[k,i], \{i : [1,n]\}$) and state estimates ($\hat{x}[k,i], \{i : [1,n]\}$). The number of predictions n, which is 50 in this case, is chosen based on factors such as available bandwidth, characteristics of the wireless channel, characteristics of the plant, accuracy of the plant model and available processing power. Control signal estimate $\hat{u}[k,i]$ calculated at time step k is applied to the plant in case of a communication failure between the controller and the actuator i time steps later (at $k+i$). The source of the control packet produced at time step k can be either the incoming sensor data, $x[k]$, if it arrives on time for time step k, or the first state estimation produced at the previous time step, $\hat{x}[k-1,1]$. In the latter case, the control packet is valid only under the assumption that the previous control signal $u[k-1]$ which is applied to $\hat{P}$ at time $k-1$ is also applied to P implying that the controller packet sent at time step $k-1$ is not lost. Needless to say, this assumption is very unlikely to hold and thus a flag named *SB*, which stands for sensor based, is also stored in the controller packet indicating whether the control signals of a particular packet are based on sensor data or previous state estimations. The actuator node uses this *SB* flag to decide which control signal must be applied to the plant at a given time. A summary of the above discussion is illustrated in Fig. 4.

Actuator State Machine Since the controller node calculates the control signal estimates of the plant model with the assumption that all of its packets are received by the actuator and all the control signals it calculates are applied to the plant, a scenario where:

(1) At sampling time $[k]$ controller sends a sensor or model based packet to the actuator.
(2) At sampling time $[k]$ actuator receives the packet and applies the control signal, $u[k]$.

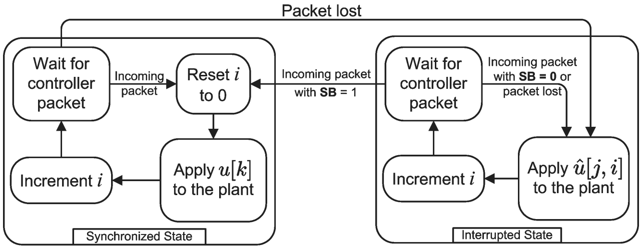

Fig. 5. Operation of the actuator node.

(3) At sampling time $[k+1]$ controller sends a *packet with* $r[k+1] \neq r[k]$ *or* $y[k+1] \neq \hat{y}[k,1]$ to the actuator.
(4) At sampling time $[k+1]$ the packet is lost. Actuator applies the 1^{st} control signal estimate of the previous packet, $\hat{u}[k,1]$.

breaks this assumption since at step 4 $\hat{P}$ receives $u[k+1]$ calculated at step 3 whereas P receives the model based prediction $\hat{u}[k,1]$ calculated at step 1. Since $u[k+1] \neq \hat{u}[k,1]$ the states of the actual plant P and the model that runs inside controller $\hat{P}$ are no longer expected to be equivalent.

In summary, whenever a controller packet is lost, ignoring the exceptional cases where $r[k+1] = r[k]$ *and* $y[k] = \hat{y}[k,1]$, control signals sent by the controller become obsolete until the next time the controller is *synchronized* with the plant by receiving a sensor packet and successfully sending its *sensor based* calculations to the actuator. In order to cope with this problem, the actuator node embodies a state machine with two states similar to the actuator given by Onat et al. (2008): the *synchronized state* corresponding to the instants when states of $\hat{P}$ and P are synchronized and the *interrupted state* corresponding to the instants when states of $\hat{P}$ and P are out of synchronization.

Upon retrieval of a controller packet, the actuator node checks its current state and the *SB* flag of the incoming packet. If the actuator state machine is in the *synchronized state*, the control signal ($u[k]$) of each packet is applied to the plant regardless of the condition of the *SB* flag until a controller packet is lost and actuator state machine makes a transition to the *interrupted state*. In the *interrupted state*, incoming controller packets are ignored and predictions of the last packet received in the *synchronized state* are applied to the plant in a consecutive manner ($\hat{u}[j,i]$, { i : [1,n] } assuming that the transition to the *interrupted state* occurred at $j+1$) until a sensor based control packet is received indicated by a high *SB* flag. When such a packet is received, the actuator state machine makes a transition back to the *synchronized state*. If the actuator node runs out of predictions in the *interrupted state*, if i reaches n, it keeps applying the last control signal estimate $\hat{u}[j,n]$ to P until the actuator node makes a transition to the *synchronized state*. An overview of the functionality of the actuator is illustrated in Fig. 5.

3.2 Resilience Against Ambient Wireless Traffic

Wireless channel is of broadcast nature and thus, performance of a W-NCS can be seriously affected by the wireless traffic created by neighboring STAs. When multiple STAs try to transmit data simultaneously, their transmissions interfere with each other causing their packets to be lost. Since STAs typically keep retrying until their packets are successfully sent, each collision increases the latency of a packet. However, since DCF is stochastic in nature with no upper bound on medium access latency, a packet can be delayed so long that by the time it is received by its destination the information it carries is no longer relevant. Thus, ambient wireless traffic effectively increases the packet loss rate of the channel since the control algorithm must receive sampled states of the plant at exact intervals and a packet received after its expected time is equivalent to a lost packet.

As a remedy to this problem, similar to the ideas behind 802.11e given by IEEE 802.11 Working Group (2007), CW_{min} and CW_{max} values can be decreased in order to control the maximum delay spread in case of collisions. Following a collision, a STA with smaller CW_{min} and CW_{max} values clearly attempts to access the channel before any neighboring STAs with default values, increasing its chances for successful transmission since the neighboring STAs will defer their transmissions once they hear its transmission. Furthermore, a decreased $DIFS$ value also gives a higher priority to a STA in channel accesses as the STA with the smaller $DIFS$ value waits shorter and forces other STAs to defer their transmissions by starting its transmission sooner.

W-MBPNCS employs both of the suggested modifications to increase its nodes' medium access priority and to decrease the delay their packets are subject to, ultimately improving the overall performance of the system under ambient wireless traffic. When deployed on a factory floor or some other facility open to wireless traffic generated by other sources such as mobile devices and workstations; W-MBPNCS can operate with no major loss in performance owing to its modified 802.11b parameters.

4. EXPERIMENTS

4.1 Test Setup

Hardware Platform The platform on which W-MBPNCS is realized and tested is, for the most part, made up of 4 hardware components: Advantech PCM-9584 single board computer, CNET CWP-854 802.11b/g wireless NIC, Mesa 4i30 quadrature counter daughter board and a Kontron 104-ADIO12-8 ADC/DAC daughter board.

Software Platform In order to realize a discrete-time control system, a real-time operating system is required. To meet this end, above mentioned hardware runs a Debian Linux distribution patched with Xenomai, a real-time development framework which provides hard real-time support to user-space applications. Additionally, all nodes of the system operate on the same disk image which is loaded to RAM during boot. Running the entire system solely on RAM minimizes disk access latencies which promotes the timeliness of the system. Furthermore, since changes are never written back to the original disk image and this image is re-loaded every time a node boots, it is guaranteed that all nodes are always operating on the same set of software despite any possible failures.

Notes on Implementation In order to reduce the number of components required to realize the W-MBPNCS, sensor and actuator nodes of the system are co-located in the same computer. However, since the sensor and the actuator never directly interacts with each other, this beneficial simplification does not alter the behavior of the W-MBPNCS in any way.

For the experiments with external traffic, a traffic generator which is identical to the nodes of the W-MBPNCS is placed strategically in the middle of two other nodes. A laptop is placed 1,5 m away to capture this traffic and measure its rate.

4.2 Results

Results presented next are obtained from 4 tests each consisting of 40 experiments. Each test is repeated 10 times resulting in a total of 1600 experiments each 30 seconds long. An automated test method is used during test runs in order to eliminate human error: a particular node (the traffic generator since it is not a part of the W-MBPNCS) controls all of the nodes in the testbed by sending relevant commands that prepare the overall system for the next experiment, initiates the experiment and collects the results in an organized fashion when the experiment ends.

In the experiments conducted, mainly the performance of the W-MBPNCS is compared with the performance of a b-WNCS that behaves in the following way in case of packet loss: if a sensor packet is dropped on its way to the controller, the controller does not produce any control signals and if the actuator does not receive any packets from the controller, it does not alter the last input applied to the plant. The nodes of the b-WNCS have the same protection against late coming packets (time stamps) as the nodes of the W-MBPNCS do.

Another focus of the experiments is the effect of packet loss on the performances of the compared W-NCS's. In order to be able to evaluate the effect of packet loss in a controlled and reproducible way, both packet loss models are implemented in the nodes of the W-NCS's. Before transmission of each data packet, each node runs its own channel model and either transmits the packet or drops it deliberately according to the state of the model.

The effect of ambient wireless traffic and the performance improvement provided by modified MAC parameters are other points of interest. In order to evaluate the performance of the W-NCS's in a real-life scenario where they would be expected to operate in the presence of wireless traffic created by other 802.11b nodes, tests both with and without constant bit-rate (CBR) traffic (750 UDP packets/s with a payload of 50 bytes) using both stock and modified MAC parameters(Tb. 2) are conducted.

Table 2. Modified 802.11b Parameters

Profile	DIFS	CW_{min}	CW_{max}
Stock	50	31	1023
Modified	30	0	3

MAC parameters of the traffic generator are left at their stock settings throughout the tests in order to observe the effects of the modified MAC parameters on the performances of the nodes of W-NCS's. Furthermore, no packet loss model is employed in the traffic generator to account for the case where the traffic source would be located very close to a particular W-NCS node and all of its traffic would make its way to the W-NCS node causing maximum interference.

The actual effect of the MAC modification is given in Fig. 6 which is the delay distributions of 3000 packets sent by the nodes of the W-MBPNCS under different conditions. Under ambient wireless traffic, the delay spread of the packets transmitted by the nodes with modified MAC parameters is shrunk to a great extent when compared with the delay spread of the packets transmitted by the nodes with unmodified MAC parameters. With a relative packet deadline of 1 ms, almost half of the packets miss their deadlines and are discarded under ambient wireless traffic when W-NCS's operate with stock MAC parameters whereas modified MAC parameters provide an almost 100% improvement.

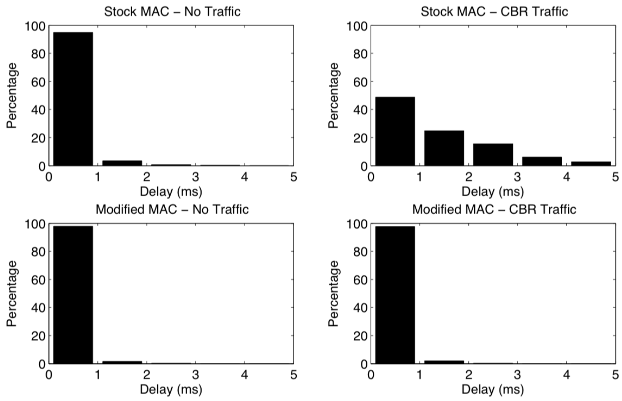

Fig. 6. Delay distribution of W-MBPNCS packets under different conditions

In the following, performances of both W-NCS's are compared with respect to their percentage root mean square of error (eRMS) averages taken over 10 identical test-runs and results of the tests are presented as percentage eRMS versus packet loss rate plots. For the Gilbert/Elliot model packet loss rate is the weighted average of P_{loss}^{g} and P_{loss}^{b} with respect to steady state probabilities of the model being in a given state and for the uniform packet loss model packet loss rate equals P_{loss}. The same reference signal is used in all experiments which is a 0.5 Hz step reference with an amplitude of *2 radians* except for the additional experiment where a sawtooth reference with a slope of *4 radians/s* is used.

In the first test (Fig. 7), the performance of W-NCS's are evaluated using stock MAC parameters under bursts of packet loss created by the Gilbert/Elliot model both with and without ambient wireless traffic. Good state packet loss probability P_{loss}^{g} of the model is swept from 0% to 45% at 5% increments to imitate worse than ideal channel characteristics in the good state. For the case with no traffic, the percentage $eRMS$ of W-MBPNCS is 54 at 7% packet loss and remains under 75 up to 39% packet loss whereas b-WNCS becomes immediately unstable under bursty packet loss. Since the percentage eRMS of b-WNCS exceeds 800 even at 7% packet loss rate, its curve is not included in the plot. Under ambient wireless traffic performance of W-MBPNCS degrades by at least 15%, nevertheless it still remains stable and outperforms b-WNCS owing to its model based predictive controller. On the other hand, b-WNCS becomes unstable under ambient wireless traffic with a percentage eRMS exceeding 160 even at 0% packet loss.

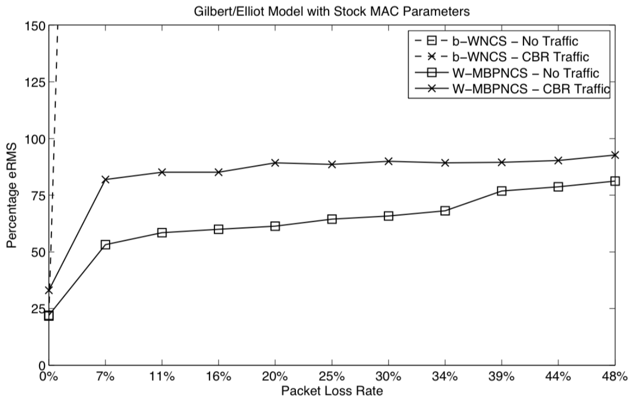

Fig. 7. Test 1: Gilbert/Elliot model with stock MAC parameters

In order to observe the performance gains introduced by the modified MAC parameters in the second test (Fig. 8), the scenario of the first test is repeated using modified MAC parameters. As expected, performances of W-NCS's are not affected when there is no ambient traffic. However, under ambient wireless traffic the performance degradation of W-MBPNCS is reduced by up to 100% and b-WNCS becomes functional again at 0% packet loss rate owing to more deterministic channel access provided by modified MAC values. These results indicate that modified MAC parameters indeed reduce the latency that W-NCS packets are subject to under ambient wireless traffic, enabling them to perform better at a given packet loss rate.

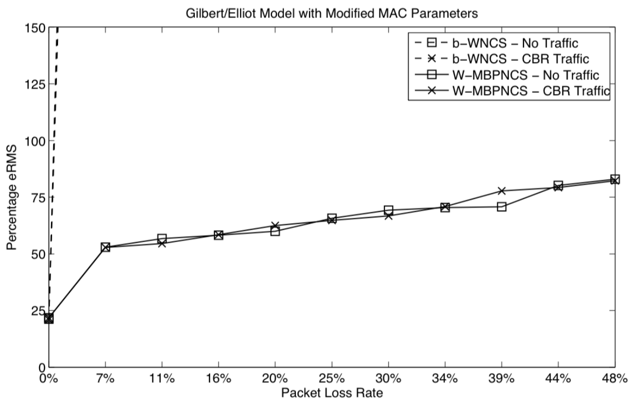

Fig. 8. Test 2: Gilbert/Elliot model with modified MAC parameters

As a means for comparison between loss models, a third test (Fig. 9) is conducted using the uniform packet loss model by sweeping the packet loss probability (P_{loss}) between 0% and 90% with 10% increments. In the first test, W-MBPNCS suffers from relatively long periods of insensitivity to the reference due to bursts of packet loss coinciding with the transitions of the reference and b-WNCS becomes abruptly unstable under bursty packet loss. In contrast, both W-NCS's perform better in this test which shows that uniform loss model results in misleading optimistic results since it does not take the correlation between packet losses into account. Nevertheless, when there is no ambient traffic W-MBPNCS's percentage eRMS remains below 50 up to 70% packet loss whereas b-WNCS's percentage eRMS exceeds 50 around 35% packet loss. Under ambient wireless traffic percentage eRMS of W-MBPNCS remains below 50 up to 40% packet loss while b-WNCS becomes unstable with a percentage eRMS exceeding 160 at 0% packet loss.

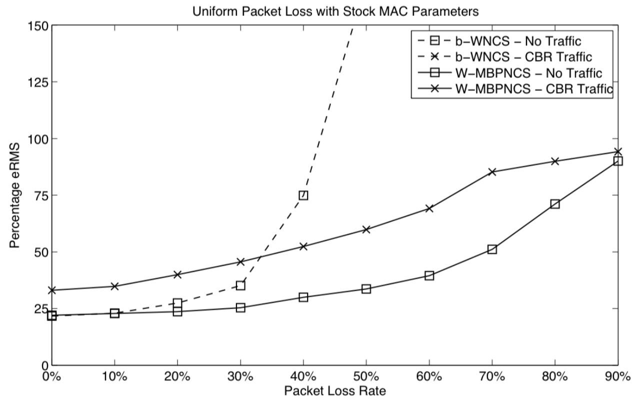

Fig. 9. Test 3: Uniform packet loss model with stock MAC parameters

Similar to the second test, in the fourth test (Fig. 10) the scenario of the third test is repeated with modified MAC parameters. As expected, effect of ambient wireless traffic on W-MBPNCS's performance is reduced by up to 100% and b-WNCS becomes functional again under ambient wireless traffic.

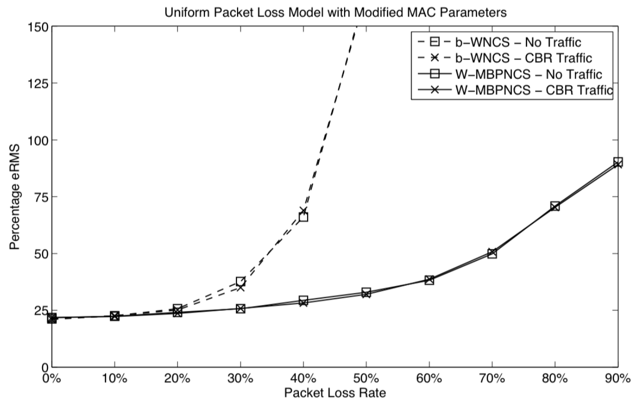

Fig. 10. Test 4: Uniform packet loss model with modified MAC parameters

Finally, in order to provide a better insight into the responses of both W-NCS implementations to different reference inputs, two time plots of plant output (motor position) versus reference input obtained using Gilbert/Elliot loss model with a packet loss rate of 7% with no ambient wireless traffic are given in Figs. 11,12. In Fig. 11 W-MBPNCS closely follows the reference owing to predictions produced by its controller, whereas b-WNCS is unstable. However, in Fig. 12, where a sawtooth reference with a slope of 4 radians/s is applied, W-MBPNCS shows some insensitivity to the tracking of the reference input due to loss of communication between the controller and actuator. Percentage eRMS values of b-WNCS, W-MBPNCS for the test presented in Fig. 12 are 726% and 77% respectively.

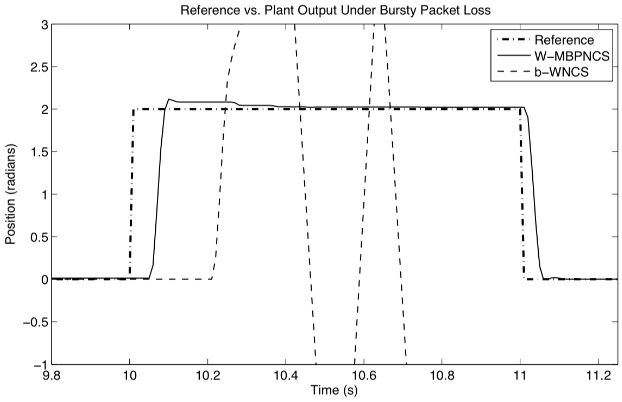

Fig. 11. Step reference time plot

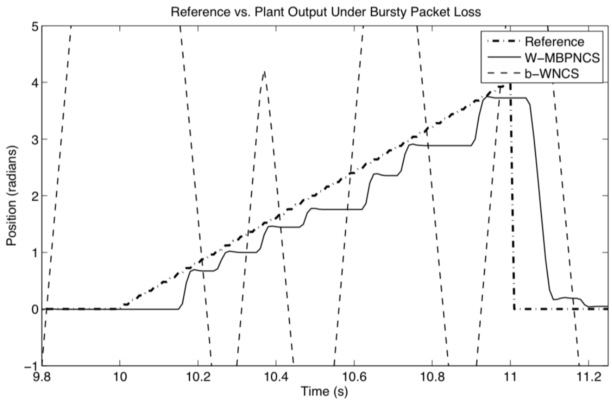

Fig. 12. Sawtooth reference time plot

5. CONCLUSION

In this work, the Wireless Model Based Predictive Networked Control System (W-MBPNCS) is presented and its performance is evaluated through extensive experiments on a test platform. W-MBPNCS is a robust wireless networked control system (W-NCS) which operates over a 802.11b wireless ad-hoc network formed between its nodes. In order to minimize packet delays and losses due to collisions caused by ambient wireless traffic, W-MBPNCS takes advantage of modified medium access control parameters for higher priority medium access. Relative packet deadlines defined on each node of the system introduce an upper bound on packet latency by discarding late arriving packets, even though the network does not provide such a bound. As a means to tolerate intermittent packet losses, the controller of the W-MBPNCS employs a model of the plant to be used in prediction of future control signals which are applied to the plant by the actuator. Thus, unbounded packet latency is reduced to tolerable packet loss and the negative effect of the packet loss is minimized by the model based predictive controller.

The performance of the proposed W-MBPNCS is experimentally evaluated in comparison with a basic W-NCS (b-WNCS) on a testbed aiming position control of a DC motor. In order to produce realistic scenarios during experiments, a Gilbert/Elliot loss model is employed to imitate the bursts of packet loss in the wireless channel and a traffic generator is used to generate wireless traffic to disrupt the communication between the nodes. W-MBPNCS outperforms b-WNCS in all test cases and its percentage eRMS is shown to remain below 60% under ambient wireless traffic and bursts of packet loss with a packet loss rate of 16% owing to its modified MAC parameters, imposed relative packet deadlines and model based predictive controller while b-WNCS is inoperative under such conditions.

The proposed W-MBPNCS is applicable to the industry since all components of the system are readily available, granted that its performance satisfies relevant requirements and the operation environment permits its use.

REFERENCES

Astrom, K.J. and Wittenmark, B. (1997). *Computer Controlled Systems*. Prentice Hall, 3rd edition.

Elliot, E.O. (1963). Estimates of error rates for codes on burst-noise channels. *Bell Systems Technical Journal*, 42, 1977–1997.

IEEE 802.11 Working Group (2007). *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE.

Kato, A. and Ohnishi, K. (2005). RTLinux based bilateral control system with time delay over network. In *Proceedings of the IEEE International Symposium on Industrial Electronics*.

Liu, G.P., Mu, J.X., and Rees, D. (2004). Networked predictive control of systems with random communication delay. In *UKACC Intl. Conf. on Control*.

Onat, A., Naskali, A.T., and Parlakay, E. (2008). Model based predictive networked control systems. In *IFAC08 World Congress*.

Pahlavan, K. and Krishnamurthy, P. (2002). *Principles of Wireless Networks: A Unified Approach*. Prentice Hall.

Rawlings, J.B. (2000). Tutorial overview of model predictive control.

Wen, P., Cao, J., and Li, Y. (2007). Design of high-performance networked real-time control systems. *IET Control Theory and Applications*, 1(5), 1329–1335.

Willig, A., Kubisch, M., Hoene, C., and Wolisz, A. (2002). Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6).