

Research Article

Control/Network Codesign Basics for IP-Based Shared Networks

**Miguel Díaz-Cacho Medina,¹ Emma Delgado Romero,²
and Antonio Barreiro Blas²**

¹ Higher Technical School of Computer Engineering, University of Vigo, 32004 Ourense, Spain

² Department of Systems Engineering and Automatism, University of Vigo, 36310 Vigo, Spain

Correspondence should be addressed to Miguel Díaz-Cacho Medina, mcacho@uvigo.es

Received 13 April 2011; Accepted 14 June 2011

Academic Editor: Blas M. Vinagre

Copyright © 2011 Miguel Díaz-Cacho Medina et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network and control relationship is an essential aspect in the design of networked control systems (NCSs). The design parameters are mainly centered in the transmission rate and in the packet structure, and some studies have been made to determine how transmission rate affects the network delay and consequently the stability of the control. In Internet, these analysis are mathematically complex due to the large number of different potential scenarios. Using empirical methods, this work deduces that the transmission scheduling problem of an NCS can be solved by designing an appropriate transport protocol, taken into account high and periodic sampling rates. The transport protocol features are determined by simulation, using a new test platform based on the NS2 network simulation suite, to develop control/network codesign solutions. Conclusions of this paper are that the transport features are packet-loss-based flow control, best effort, and fairness, supplemented by a packet priority scheme.

1. Introduction

In general, a closed-loop control system is designed without considering communication constraints due the proximity of the two extremes and the absence of sharing requirements in the control loop. If the control system has a network inserted between the controller and the plant, it is called networked control system or NCS, and communication constraints have to be taken into account, such as delays and packet losses. To maintain appropriate performance, control and sensor data transmitted through the network should have real-time requirements and are usually of small size. We call these control and sensor data, *NCS data* encapsulated in *supermedia packets*.

If the control system uses Internet as intermediate network between controller and plant, the advantages are the high availability, testability, simple installation as well as low cost. These features make the use of Internet in control systems more popular in many applications, such as traffic control, teleoperation, and mobile robotics [1]. But Internet is a shared network, with variable delays and packet losses that degrades the performance of the NCS, or it may even become unstable.

So far, the study of relationship between network and control system in an Internet based NCS or in teleoperation systems [2] is limited to adapt the controller design to several network variables (such as the delay constraint, the delay variation, and the number of consecutive losses [3, 4]). But in Internet, these network variables are unpredictable and unlimited. The network/control design techniques include some network design parameters, such as the transmission rate or the data packet structure to improve the performance of the NCS, and in Internet, this becomes mandatory. Then, a proper performance knowledge of an NCS under various network conditions allows to select proper codesign parameters that can model the response of the NCS.

Other control and network performance studies are presented by Lian et al. [5, 6] focused on the performance analysis of MIMO control systems and on the study of the interaction between control and communication mechanism. But these studies are centered in the fieldbus technologies and LAN networks, where network load is due by the multiple network agents of the NCS. In our study, similar monitor criteria and graphs are presented, but completely oriented to the IP network features, with different results than the aforementioned works.

The conclusions of Lian et al. studies [5] are that it is necessary to minimize the transmission rate to reduce the network load and not increase the delay, maintaining the control performance. This can be effective if the residual network load has no flow-control and best-effort features, such as in fieldbuses with multiagent NCSs or in IP-based networks with UDP background traffic. Further works have taken into account these results and have developed control scheduling algorithms to minimize the transmission rate. But in Internet, the main transport protocol of the residual network load is TCP, with flow-control and best-effort features, so, the aforementioned studies have no effect.

This paper analyzes the responses of the NCS in networks with UDP/TCP/IP shared traffic for a range of propagation delays and queue size of the intermediate routers. This analysis will provide specific strategies to address the problem of transmission rate for NCSs on the Internet. The paper focuses on the problem with three main contributions.

- (i) Definition of the performance basics and simulation parameters for NCSs in IP shared networks.
- (ii) The development of a simulation framework in NS2 that provides implementation of several transport schemes. It delimits the possibility of actuation in the application and transport layers of the IP protocol architecture, basically in the transmission scheduling and flow-control mechanism.
- (iii) The presentation of empirical results of the simulations in several IP shared scenarios, and codesign conclusion based on these results. Particularly, the main contribution of this paper focuses on the suitability of using a transport protocol with a packet-loss-based flow-control mechanism and best-effort feature, in addition to a packet priority scheme and fairness with other Internet data flows.

This paper is organized as follows. After the Introduction and Section 2, a normalization of network and control parameters are defined in Section 3, such as sharing

characteristics, quality of control, and error measurement. A test platform for simulation of codesign solutions is presented in Section 4. Results of diverse simulation situations in a case study are in Section 5, and this gives specific transport design lines in Section 6. Finally conclusions are presented in Section 7.

2. Related Work

Classical techniques of NCS look to the network as a passive element, trying to adapt the control to a previously defined network. Modern techniques have emerged with the generic name of codesign that look to the network as an active element of the NCS. They try to model and integrate the network in the control system. Some authors refer generically to this line of work network-based control (NBC).

2.1. Packet Based Control

This line scans the contents and structure of *supermedia packets* to optimize the network performance and efficiency [7–10]. This method allows to work on predictive control, because the controller (or the plant) can send actual, past, and future *NCS data* in the same packet. Thus, if the packet is lost (or the transport control decides not to send it), the other end has predictive information received in previous packets [11]. Some authors take in to consideration this method, as seen in [12, 13].

2.2. Codesign of Control and Transmission Rate

Transmission rate scheduling to manage the available network resources in a fairly form prioritizes dynamically the NCS that needs more bandwidth to maintain stability. This line of work studies protocols, middleware systems, and control theory. In [14], a dynamic control algorithm is proposed that allows the use of network resources by the node with the biggest error. In [15], an NCS is designed that switches between open, and closed-loop system depending on the network access availability. In [16], a study where the optimal scheduling problem under rate-monotonic constraints and NCS stability constraints is presented. In [17], a NS2-based framework is presented to make codesign experiments. In [13], predictive control with codesign techniques to use dynamically network resources has been done. An interesting area of research is the distribution of available bandwidth between different NCS on a shared network, taking into account the state of the NCS by a supervisory system using adaptive sampling [18–20].

2.3. Codesign of Control and Gain

Scheduling the transmission rate is not the only line involved of an NCS that process the network status. In [21], a middleware system that modifies control gains is studied, taking in to consideration the network status. The controller output algorithm depends on the remote plant and controller global configuration. The state of the network is known by sending probe packets.

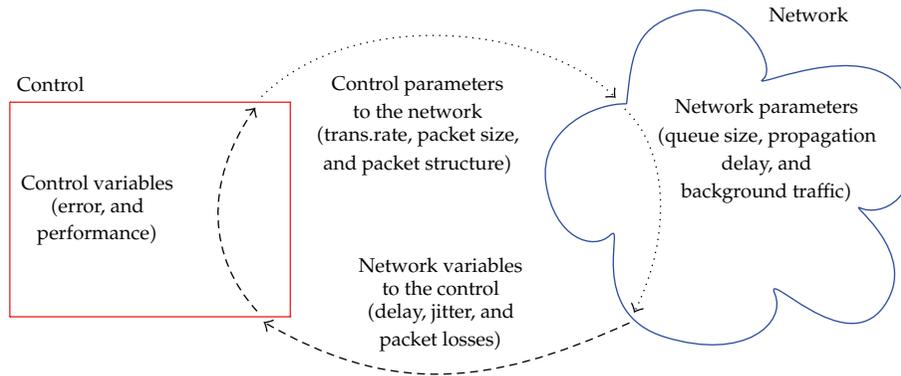


Figure 1: Control and network codesign parameters and variables.

2.4. NCS Specific Protocols

This line is completely centered in the network component of the NCS. There are some *NCS data* protocols at all the layers of the IP architecture. In layer 2, NCS protocols focus on field bus technology, CAN bus, or recently EtherCAT; in network layer, the NCS research lines are in the link-queue analysis (new RED queues) and in the multicast solutions to improve real-time requirements and network use efficiency; in the transport layer, there are some solutions around the UDP protocol, developing flow-control features, such as TFRC [22], DCCP [23], or the trinomial algorithm [24]; the application layer solutions are usually focused on developing particular or generic frameworks and *NCS data* structures.

2.5. Cosimulation

To work with codesign methods, some frameworks are developed. In [25, 26], a new agent is developed to reproduce controllers and plants in the TCL user space of the NS2 network simulation suite. Authors make tests in a distributed MIMO NCS to study relationships between control and network parameters, but protocols are not tested, and only queue sizes of the routers are varied to test the NCS responses. Shared scenarios with different transport flows are not tested.

3. Network and Control Parameters and Variables

A network/control codesign framework should take into account a lot of parameters and variables to obtain correct results in a realistic way. Parameters are design values that can be controlled by the designer and determine the NCS environment; variables are network and control responses that affect the performance of the NCS. In Figure 1, we present a flow chart for control/network parameters/variables involved in this study. From a control system point of view, the network parameters and variables cannot be modified, and the problem is how to model the control parameters that affect the network to maintain the control variables in an acceptable way.

Control performance is a variable that depends mainly on the controller design, the plant environment, and (among others) on returned network variables, such as delay,

jitter, or packet losses. All these network variables depend mainly on the characteristics of the transmission path and the amount and type of background traffic in the Internet. Transmission path is a parameter represented by the queue size of intermediate routers and propagation delay; background traffic is a parameter that has direct effect on the queue occupation and affect the network delay. Transmission rate and data packet structure are control parameters that affects directly to the control and network variables, as described below.

To normalize the empirical studies in Internet-based NCS, some initial assumptions should be made but without limiting the basic conclusions of the results.

Assumption 1. Internet (or any IP-based network) is the network involved in this study. Physical and link layer technologies are unknown and not parameterizable, but we parameterize the propagation delay and the type and amount of background shared traffic. In the IP layer, we parameterize the queue type and size of intermediate routers. The first layer where the control system can actuate is the transport layer mainly by selecting an appropriate transport protocol.

Assumption 2. All the *supermedia packets* are of fixed size $p_s|_{\text{nCS}}$ and include only one NCS sample. Thus, packet structure is fixed, and this study does not depend on it.

Assumption 3. Data sampling and packet transmission rate are equal and fixed in both extremes of the NCS. This simplifies the design without loss of generality and maintains transmission rate without dependence on the network status.

Assumption 4. The control system is well designed in closed loop with preselected fixed delays. Stability with variable delays is conditioned to situations that cannot be predicted in Internet although delays and packet losses can be monitored [4].

Assumption 5. No initial control/network codesign, no flow-control protocol and no network-dependence for the control system to know the behavior of the control system in different network situations. Network and control are completely transparent. The control system does not change its design with the network parameters/variables, and network does not change with the control parameters/variables. The base transport protocol for NCS is UDP.

Control performance, transmission rate, queue size of the intermediate routers, and network background traffic are explained below.

3.1. Performance Criteria

Multiple performance criteria can be created, based on the importance of the network and the control system. The literature uses the term quality of control (QoC) [27], but we use the term *performance index* denoted by P , adding subscripts to differentiate one index from the others. P and QoC have an opposite relationship: the higher is P , the lower is QoC and vice versa. When both values are normalized between 0 and 1, the relationship between the two terms is given by

$$QoC = 1 - P. \quad (3.1)$$

Performance criteria should have a control component and in some situations one or more network components; the control component should be an control-error based indicator. To compare the error-based performance between different NCSs, the error must be normalized. We define the normalized error E_n of an NCS as the result of dividing the instantaneous error E by the maximum permissible error E_{\max} defined as a design parameter for that NCS (3.2). The maximum permissible error depends on the design criteria of the NCS and must have a value low enough to maintain system stability and high enough to keep the NCS operating

$$E_n = \frac{E}{E_{\max}}. \quad (3.2)$$

To make the error independent of time, instead of the mainly used integral absolute error IAE indicator (3.3), we propose the use of average absolute error AAE (3.4), where T is the measuring interval

$$\text{IAE} = \int_0^T |e(\tau)| d\tau, \quad (3.3)$$

$$\text{AAE} = \frac{1}{T} \int_0^T |e(\tau)| d\tau. \quad (3.4)$$

IAE should be measured in a minimal interval of $T_{\text{ms}} = 1/v_{\text{tms}}$, where v_{tms} is the minimum transmission rate (in packets/sec) to maintain the stability of the NCS (given in (3.8)), but AAE can be measured in continuous time simulation or execution.

In shared networks, where several NCSs or distributed systems can work, performance indexes P_i must take into account the amount of control and network variables for all NCSs. It does not help optimize an NCS if that implies to reduce the benefits of others. To compute control and network responses into the performance index, two relevance and scale factors are needed: K_1 for the error component, and K_2 for the network component. So, in a network with N NCSs, we define four different performance indexes

- (i) P_0 (3.5): it measures the amount of normalized error of all NCSs in the network. This performance index takes into account only the control systems. P_0 is used to determine the stability region in Section 5.2.

$$P_0 = \sum_{i=0}^N E_{n_i}. \quad (3.5)$$

- (ii) P_1 (3.6): this performance index computes the error and normalized delay (τ_n) respect to the maximum permissible delay to maintain NCS stability (τ_{ms}) in both directions. Then, $\tau_n = \tau / \tau_{\text{ms}}$,

$$P_1 = K_1 \sum_{i=0}^N E_{n_i} + K_2 \sum_{i=0}^N \tau_{n_i}. \quad (3.6)$$

- (iii) P_2 (3.7): P_2 is used to measure the relationship between losses and benefits. The amount of packet loss should be normalized respect the amount of transmitted packets, so it matches the probability of packet loss p . If v_t and the arrival rate (or success rate) v_r in the destination are known, $p = 1 - (v_r/v_t)$,

$$P_2 = K_1 \sum_{i=0}^N E_{n_i} + K_2 \sum_{i=0}^N p_i. \quad (3.7)$$

- (iv) P_3 : finally, P_3 is used to determine the degree of involvement of the transmission rate v_t , normalized respect to the sampling rate ($v_{t_n} = v_t/v_s$, where v_s is the sampling rate). This index allows to punish excessive transmission rate to the network. In this work, using Assumption 3 $v_t = v_s$, then $v_{t_n} = 1$ and P_3 is not taken into account.

3.2. Transmission Rate

The influence of the transmission rate on the performance of an NCS is given by the relationship between the network use, the delay induced by the network, and the percentage of packet losses. Figure 3 [5] represents the influence of different types of control on their performance, as a function of the sampling/transmission rate. Performance index is usually P_0 .

In Figure 3, the performance degradation with high transmission rate is highlighted by point Z, caused mainly by increasing packet losses and autoinduced network delays. Point Y is defined as the minimum transmission rate of the NCS to maintain acceptable performance. In [5], a methodology to calculate the minimum and maximum transmission rate (points Y and Z, resp.) for unshared networks is proposed.

The lower limit for point Y is the point X, given by (3.8), where BW_c is the controller bandwidth in closed loop operation [28].

$$20 < \frac{v_t}{BW_c}. \quad (3.8)$$

Point Z has an upper limit given by (3.9), where C_l is the point to point link capacity for link l and L is the set of point to point links traversed by the NCS data flow. C_a is the available capacity of the network link

$$Z|_{\max} = C_a|_{\max} = \min_{l \in L} \{C_l\}. \quad (3.9)$$

Points Y and Z vary depending on the available capacity of the network (C_a) and the background traffic features.

3.3. Queue Size

Queue size (q_s) has direct relationship with network delay (τ) as seen in [(10) (presented in [25])] for Drop Tail queues, where l is a link index, L is the amount of links (or queues), τ_{p_l} is

the propagation delay for link l , p_s is the packet size in bytes (as in Assumption 2), q_{s_l} is the queue size for link l (in packets), and C_l is the link l capacity in bits/sec

$$\tau_{\max} = \sum_{l=1}^L \left(\tau_{p_l} + \frac{8p_s(q_{s_l} + 1)}{C_l} \right). \quad (3.10)$$

Drop Tail queues measure their size in packets, where the packet size has maximum limit given by the MTU (maximum transfer unit) parameter of the link layer technology. For shared networks with different data packet sizes, the maximum delay is calculated by substituting the p_s variable by its maximum potential value. That has direct effect on the packet efficiency of the *NCS data* in Internet, where an IP packet has a maximum value of 65.535 given by the *Total Length* field of the IPv4 packet header, and *supermedia packets* are of small size.

3.4. Background Traffic

In shared networks like the Internet, we distinguish traffic without flow control (UDP) and traffic with flow control (TCP). Available bandwidth depends on the amount of traffic of the two flow-styles. To simplify the explanation, we call *background traffic* to all data flows that are not related with NCS traffic. Thus, an unshared network has not background traffic in the host to host link. Background traffic is a network variable treated as a network parameter by the simulation environment. For IP shared networks, we can establish a basic classification for background traffic characteristics and their influence in the available capacity (represented by C_a) of the network link

- (i) UDP shared: shared network with background traffic without flow control (mainly UDP protocol). If C is the host to host link capacity and C_u is the capacity used by the UDP flows, available capacity is

$$C_a = C - C_u. \quad (3.11)$$

- (ii) TCP shared: shared network with flow-controlled background traffic (mainly TCP protocol). In these networks, a defined available capacity does not exist because of the best-effort feature of the TCP traffic. Then, in practice, $C_a = C$, but always with varying delay and packet losses. The best-effort feature is reflected in the TCP model by Padhye et al. [29], where the transmission rate increases, while the network is not saturated, that is, while there is no packet loss ($p = 0$), as seen in (3.12). In (3.12), p is the probability of packet loss and t_{RTO} is the retransmission time. The packet size of a TCP flow is usually higher than the *supermedia packet* size and is represented by $p_s|_{\text{tcp}}$

$$T = \frac{p_s|_{\text{tcp}}}{\text{RTT}\sqrt{2p/3} + t_{\text{RTO}}(3\sqrt{3p/8})p(1 + 32p^2)}. \quad (3.12)$$

- (iii) TCP/UDP shared: shared network with both types of traffic, UDP and TCP data flows. Internet is a TCP/UDP shared network.

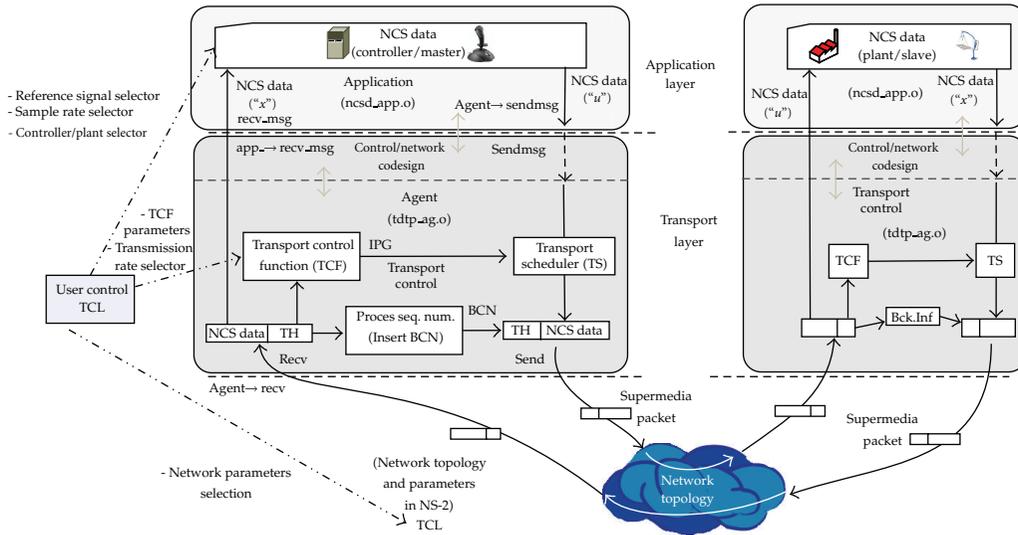


Figure 2: Test platform to develop NCS codesign solutions.

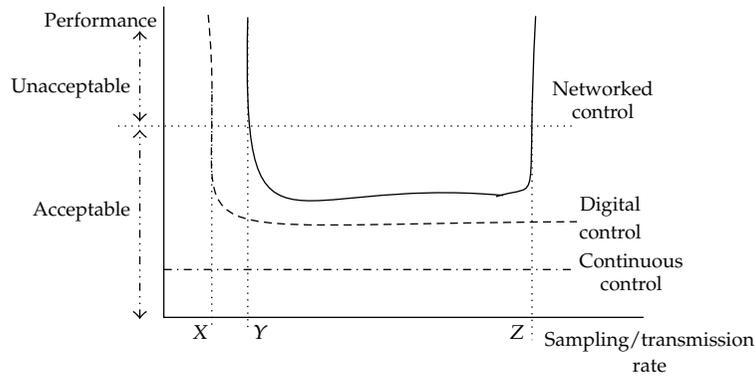


Figure 3: Relationship between performance and transmission rate. Theoretical approach.

4. NCS Test Platform

To show the influence of IP shared networks on control system performance and to test the transport and control strategies, some control and network topology experiences over a simulated test platform were performed.

The test platform is developed in the NS2 network simulation suite that allows to develop all the layers of the OSI protocol stack. NS2 has a C++ library to develop new agents and applications and a TCL environment to create nodes and network topologies. TCL environment works in user space. The scheme of the complete test platform is presented in Figure 2 and the complete test platform is available to download in [30].

The test platform has one agent (*tdtp_ag.o*) and one application (*ncsd_app.o*). The agent is focused on the transport layer and implements a bidirectional transport scheme presented in [10]. The application implements controller and plant logics. User space in NS2 allows to

select several transport and application parameters, such as the transport scheme and the NCS sampling rate, as well as the network topology and other network parameters.

The transport layer allows to test well-known transport schemes (such as TCP, UDP, trinomial algorithm, or TFRC) and to develop new transport protocols for NCS. For any solution that uses the codesign as a start point, it is necessary to define a new sublayer between the transport and application layer, where *NCS data* and network variables can interrelate. In some cases, the control system itself monitors network variables to implement a network-based control, such as the gain scheduling solutions [21].

In our experiments, UDP transport protocol is used for the *NCS data* to meet Assumptions 3 and 5. In this way, the NCS transport protocol may not affect the delay variation and packet losses of the *NCS data* flow.

4.1. Control Application Layer

Controllers and plants are simulated by using a first-order Euler discretization method, then the state-space equations are used to be implemented in the NCS application (*ncsd_app.o*). Sampling rate is fixed as a parameter, and the sample is calculated with the last received data using a software-based hold last sample strategy (HLS); initial conditions are set to 0.

At this moment, two models of plants (a mass spring and a gantry crane) and three controllers (LQR controller for each plant and a PID controller for the mass-spring) are developed in the *ncsd_app.o* module of the test platform. All the NCSs implemented are validated comparing results with Matlab simulation and are designed according to Assumption 4. In this work, we use the gantry crane model with LQR controller. Equations in state space for the gantry crane are presented in (4.1) for the plant and in (4.2) for the controller, where r is the reference signal

$$\dot{\vec{x}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -18.3 & 0.959 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -9.5 & -0.003 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 7.2 \\ 0 \\ -0.95 \end{pmatrix} u, \quad (4.1)$$

$$u = N \cdot r - K \cdot \vec{x}, \quad (4.2)$$

with $K = (12.1661, 0.6646, -22.8596, \text{ and } -3.4107)$ and $N = 12.1661$.

Figure 4 shows the simulated gantry-crane teleoperated system.

4.2. Transport Layer

This layer involves a *control/network codesign* sublayer and the *transport control* sublayer.

- (1) *Control/Network Codesign Sublayer*: this sublayer relates network parameters with control parameters and could incorporate scheduling algorithms, deadband filters, scattering transformations, or other control/network design procedures.
- (2) *Transport Control Sublayer*: the *transport control* sublayer implements several transport control functions (TCF), the transport scheduler (TS), and packet encapsulation and decapsulation using the transport header (TH).

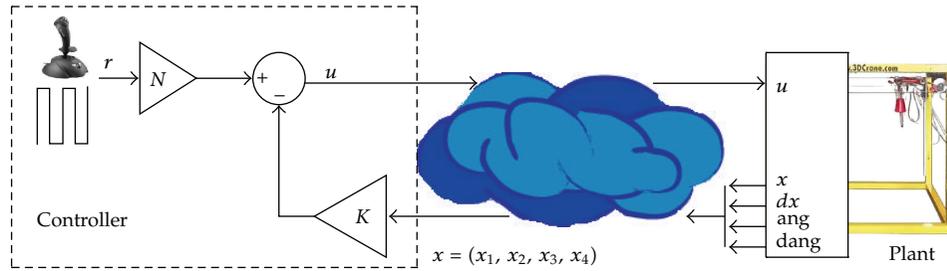


Figure 4: Gantry-crane NCS.

The transport control function (TCF) implements the transport control algorithm to determine the transmission rate in the transport scheduler (TS). Transmission rate (in packets per second) is usually given by the inverse of an interpacket gap (IPG) variable calculated by the TCF.

Transport encapsulation and decapsulation functions insert the transport header (TH) in transmission, and process the TH in reception. Transport header includes the *backward congestion notification tag* (BCN) to implement packet-loss-based flow-control by processing received sequence numbers.

Using UDP as the transport scheme, the transport scheduler is transparent, making the transmission rate equal to the sampling rate, as in Assumption 3. Then, the TCF and encapsulation/decapsulation functions have no effect on the results.

4.3. Network Topology

Figure 5 shows the created network topology to simulate a standard Internet link with a bottleneck between two routers R1 and R2 and several traffic nodes. The parameters of this link are changed to obtain performance of the NCS depending on the queue size and extreme to extreme propagation delay. The MTU parameterized in the trunk link has a value of 1500 bytes, thus the maximum packet size (p_s) in the queues has the same value to meet Assumption 2.

The simulation scenarios attempt to reproduce different network congestion situations on the basis of the classification made in Section 3.4.

- (i) Scenario e0. Scenario e0 represents an unshared network without background traffic. This scenario represents a situation with constant delays and without packet losses if the transmission speed is less than the capacity C of the trunk link between R1 and R2.
- (ii) Scenario e5. Scenario e5 represents an UDP shared network. It uses a background UDP flow with a transmission rate of $0.8 \cdot C$ in both directions and a packet size of 100 bytes. In this scenario, the delays are more variable, and there will be packet losses if the transmission speed is greater than the remaining capacity of the backbone link ($0.2 \cdot C$).
- (iii) Scenario e13. Scenario e13 represents a TCP shared network. This scenario has exclusively TCP traffic with FTP (File transfer protocol) application data using a file with infinite size and packet size limited by the MTU to 1500 bytes. There are

simulations, the stability region (in similar terms as in [31]) depending on the propagation delay is obtained.

5.1. Performance Index P_0 Depending on Queue Size and Transmission Rate

P_0 , τ ($= P_1|_{(K_1, K_2)=(0,1)}$) and p ($= P_2|_{(K_1, K_2)=(0,1)}$) are obtained as functions depending on the transmission rate (v_t) and the queue size (q_s) of the intermediate routers (5.1) for different scenarios (e_x). Measured delay τ is the average delay for success received packets along the simulation interval. The use of variable queue sizes can represent different transmission paths for the *supermedia packets*

$$P_0, \tau, p = f(v_t, q_s, e_x). \quad (5.1)$$

The results are graphically presented for scenarios e0, e5, e13, e83, and ev02 in Figures 6, 7, 8, 9, and 10 respectively, where v_t axis is in logarithmic scale for presentation purposes. Subgraphs (a) show the P_0 index and related stability region, (b) the delay, and (c) the normalized loss rate. The stability region is determined by selecting v_t and q_s , where $P_0 \leq 1$, represented graphically by the projection of $P_0 \leq 1$ in the horizontal plane in subgraphs (a).

Figure 11 represents performance indexes P_0 (Figure 11(a)), P_1 (Figure 11(b)), and P_2 (Figure 11(c)) obtained by simulation, setting $q_s = 60$. Relationship between performance P_0 and transmission rate, theoretically represented in Figure 3, is obtained in Figure 11(a) for all scenarios except e13, because in this scenario, with $q_s = 60$, the NCS is always unstable, due to a delay higher than τ_{ms} as seen in Figure 11(b). In Figures 11(a) and 11(b), points Y (minimum transmission rate) and Z (maximum transmission rate) are highlighted. Scenario ev02 is not represented in Figure 11(b), because the average delay measured among the complete simulation interval is not representative in a varying scenario.

Analyzing all the graphs, some interesting remarks are obtained, and some conclusions are given in Section 6.

Remark 5.1. The NCS stability region in all scenarios correlates with smallest delay (lower than $\tau_{ms} = 0.07$ s), as seen in Figures 6(a), 6(b), 7(a), 7(b), 8(a), 8(b), 9(a), 9(b), 10(a), and 10(b)) of Figures 6, 7, 8, 9, and 10, as theoretically expected.

Remark 5.2. In TCP/UDP shared scenarios (e13, e83, and ev02), when v_t remains constant, the delay τ (P_1 index) increases with the queue size q_s and the different packet sizes p_s involved, as seen in Figures 8(b), 9(b), and 10(b). In all scenarios, when $v_t > C_a$ and q_s increases, the queue fills and τ increases following (3.10) with $p_s = 100$ due the TCP flow-control mechanism based on packet losses. If τ is bigger than τ_{ms} , the NCS becomes unstable, as expected.

Remark 5.3. The delay does not increase if v_t increases while $v_t < C_a$ and q_s remains unchanged; transmission rate and delay are not closely related as commonly assumed. Relationship between delay and transmission rate depends on the available capacity and the type of background traffic in the network. In TCP shared scenario (e13, Figure 8), when q_s remains unchanged and v_t increases, τ decreases if $v_t < C$.

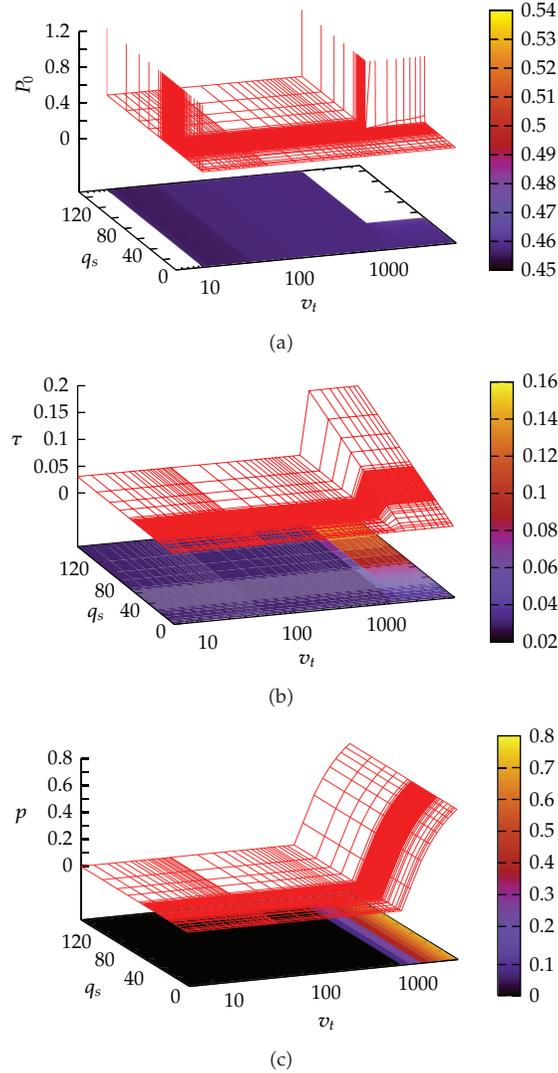


Figure 6: Scenario e0. (a) P_0 index and stability region, (b) P_1 index with $(K_1, K_2) = (0, 1)$ and $\tau_{\max} = 1$, and (c) P_2 index with $(K_1, K_2) = (0, 1)$.

Remark 5.4. The limit of the available network capacity can have two indicators: delay (P_1) or packet losses (P_2). P_1 as capacity indicator is only effective in unshared networks or UDP shared networks. P_2 capacity indicator is valid in all networks.

5.2. Stability Region Depending on Transmission Rate and Propagation Delay

Another interesting result is the stability region, meaning the results presented in [31] for fieldbuses, by fixing the queue size parameter in a very small value and changing the propagation delay of the trunk link. The stability region can be easily determined for simple

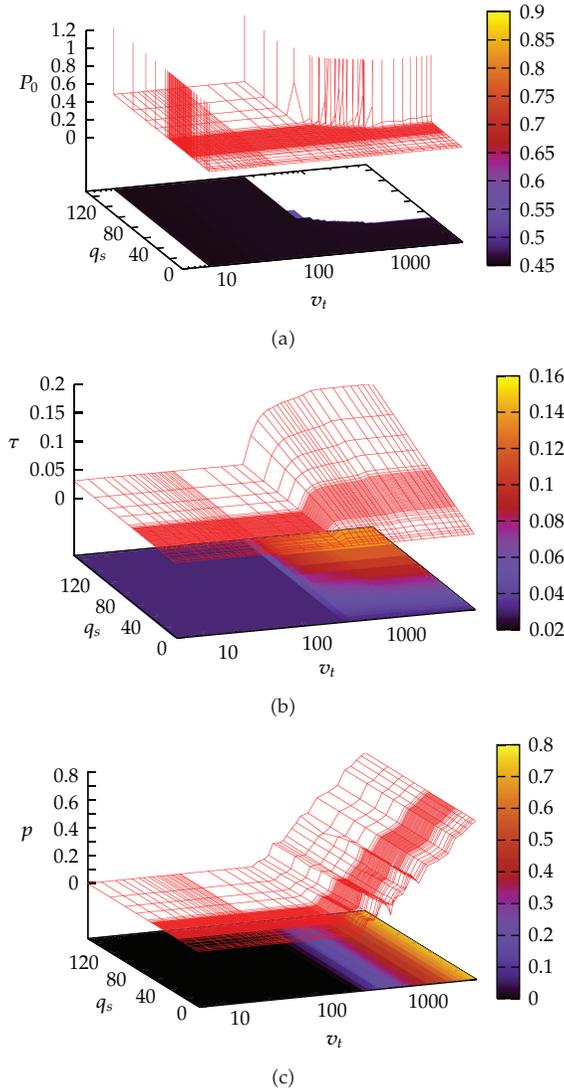


Figure 7: Scenario e5. (a) P_0 index and stability region, (b) P_1 index with $(K_1, K_2) = (0, 1)$ and $\tau_{\max} = 1$, and (c) P_2 index with $(K_1, K_2) = (0, 1)$.

scalar systems, but it may be analytically infeasible to derive the exact stability region for general systems [31]. To determine this region for NCS with network-induced delay, an analytical method using hybrid systems technique can be feasible, improving the Schurness property for the system matrices. This is more complex with NCSs in shared networks, with different traffic sources, due to the network-induced delay, autoinduced delay, and packet losses. Then, to determine the stability region a simulation methodology is needed, and the presented codesign test platform is suitable to do that.

The monitored variable is the performance index P_0 , and the stability criteria is very difficult to determine. In [25], the stability criteria is determined measuring the error in the second half interval of the simulation time and comparing it with the error in the first

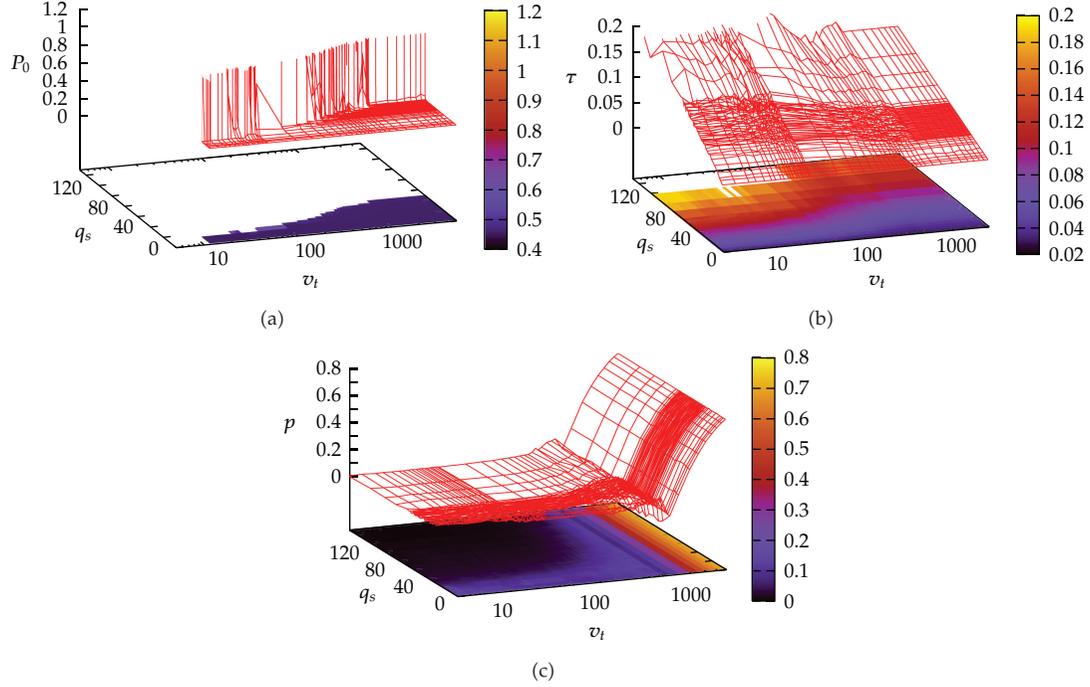


Figure 8: Scenario e13. (a) P_0 index and stability region, (b) P_1 index with $(K_1, K_2) = (0, 1)$ and $\tau_{\max} = 1$, and (c) P_2 index with $(K_1, K_2) = (0, 1)$.

half interval. If the error decreases, then the system is considered stable. Equation (5.2) summarizes this approach

$$\max_{t \in [T/2, T]} \{|\text{error}(t)|\} < \max_{t \in [0, T/2]} \{|\text{error}(t)|\}. \quad (5.2)$$

When simulations are performed on shared networks, the criterion established in (5.2) is not valid if, in the second half of the simulation interval, the background traffic is higher than in the first half or the delay increases by changing the network path. Therefore this paper presents a simple approach, measuring the error AAE at the end of the simulation and determining if $E_n = \text{AAE}/\text{AAE}_{\max}$ exceeds a threshold T , ($E_n < T$). In fact, the loss of stability of a dynamic system is not gradual and quantitative but qualitative and abrupt. However, given the complexity of the simulated scenarios (including topology, protocols, and network load), an analytical approach of stability based on accurate models would be intractable. Moreover, the proposed experimental approach detects this character of abrupt change perfectly in the loss of stability if the proposed threshold T is large enough. Using the P_0 index as E_n , and the value 1 for the threshold, the stability criteria used is (5.3), and results are presented in Figure 12 for scenarios e0, ev02, and e83

$$P_0 < 1. \quad (5.3)$$

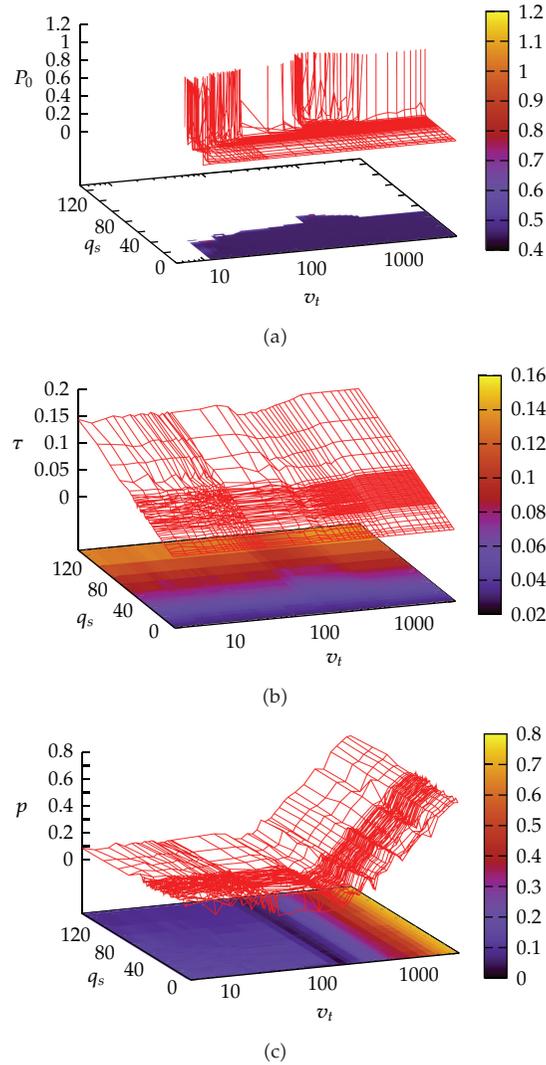


Figure 9: Scenario e83. (a) P_0 index and stability region, (b) P_1 index with $(K_1, K_2) = (0, 1)$ and $\tau_{\max} = 1$, and (c) P_2 index with $(K_1, K_2) = (0, 1)$.

6. Internet-Based NCS Considerations

From (3.11) and Remark 5.3, UDP background traffic causes that point Z decreases, and TCP background traffic causes that point Y increases. Figure 13 represents this remark. This is because when UDP traffic increases, the point Z has a lower value due to the decrease in available capacity (C_a), and when TCP traffic increases, point Y has a higher value due to packet losses and to the queue occupation caused by the best-effort feature of TCP traffic.

Remark 5.3 is full of importance due to the generally supposed assumption that increasing the transmission rate of the NCS causes delay increases and the NCS becomes unstable. Remark 5.3 is caused by the best-effort feature of TCP flows that makes that the queues are mainly occupied by TCP packets if other flows (such as NCS flows with low v_t)

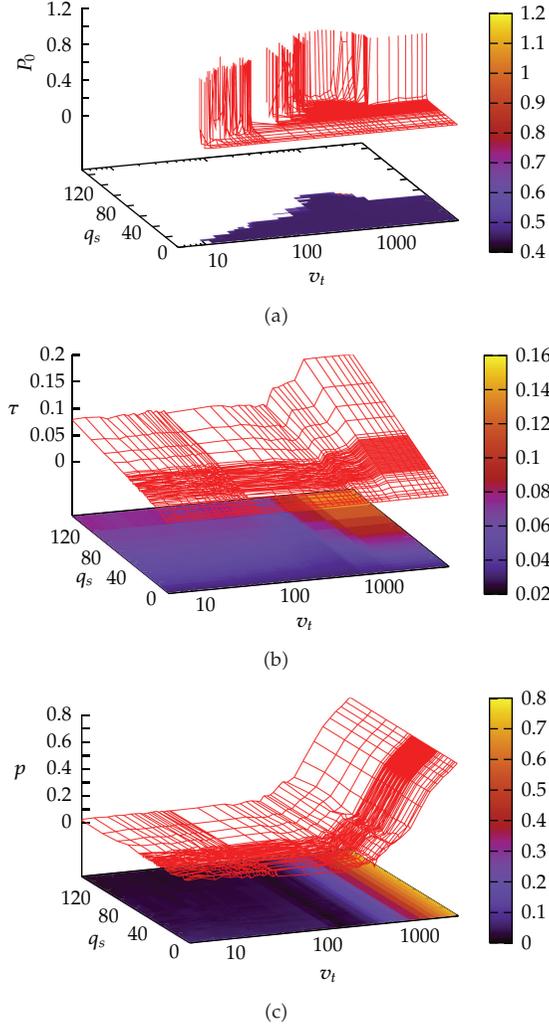
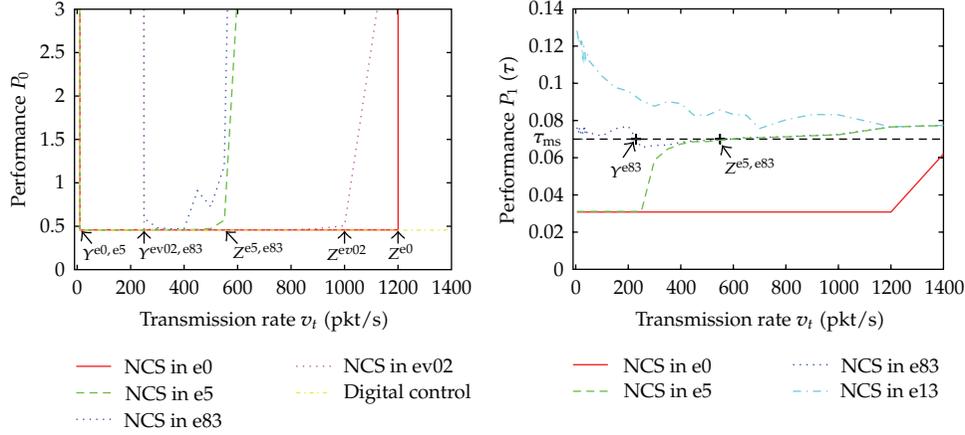


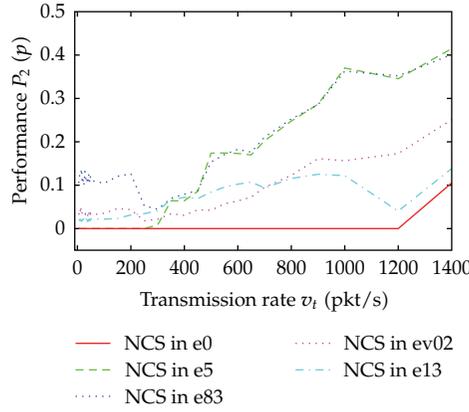
Figure 10: Scenario ev02. (a) P_0 index and stability region, (b) P_1 index with $(K_1, K_2) = (0, 1)$, and (c) P_2 index with $(K_1, K_2) = (0, 1)$.

are residual. When v_t increases, TCP flows increase their packet losses, and, due to the flow-control feature, TCP decreases its transmission rate. This causes that the queue is increasingly occupied with NCS *supermedia packets*, and, while $v_t < C$ and $p_s|_{\text{nCS}} < p_s|_{\text{tcp}}$, the delay of NCS flows decreases. This remark has two important consequences.

- (i) High v_t is more tolerant with delay variation as low v_t , as seen in the stability region of Figures 8(a), 9(a), and 10(a), and in the stability regions of Figure 12.
- (ii) In Internet, with mostly TCP background traffic, it is more important to generate high transmission rates than low transmission rates. Thus, this solution is not compatible with pure deadband solutions [32] that try to minimize transmission speed.



(a) Performance P_0 for $q_s = 60$ (NCS granty-crane LQR) (b) Performance P_1 for $q_s = 60$ (NCS granty-crane LQR)



(c) Performance P_2 for $q_s = 60$ (NCS granty-crane LQR)

Figure 11: Relationship between (a) performance, (b) delay, and (c) losses versus transmission rate for a gantry-crane NCS in different network scenarios.

From Remark 5.4, the transport protocol for NCS *data* should implement flow-control based on packet losses P_2 to detect automatically the instantaneous available capacity of the network link. Some protocols such as BTP [33] have flow control based on delay (or RTT) monitoring but may not be useful in TCP shared networks. If multiple NCS share the network, some fairness and *supermedia packet* priority features should be mandatory to maintain global performance indexes.

As global conclusion, in NCSs through Internet, v_t design in the application layer is complex and unnecessary, and may be replaced by the design of an appropriate transport protocol, with flow control, best-effort and fairness features. Flow-control allows to detect the available link capacity, best-effort feature allows taking advantage of the network resources without increasing the network delay, and the fairness and priority feature allows to solve the well-known congestion problem in TCP/IP networks and share the network with other NCSs without compromising the stability of any particular NCS.

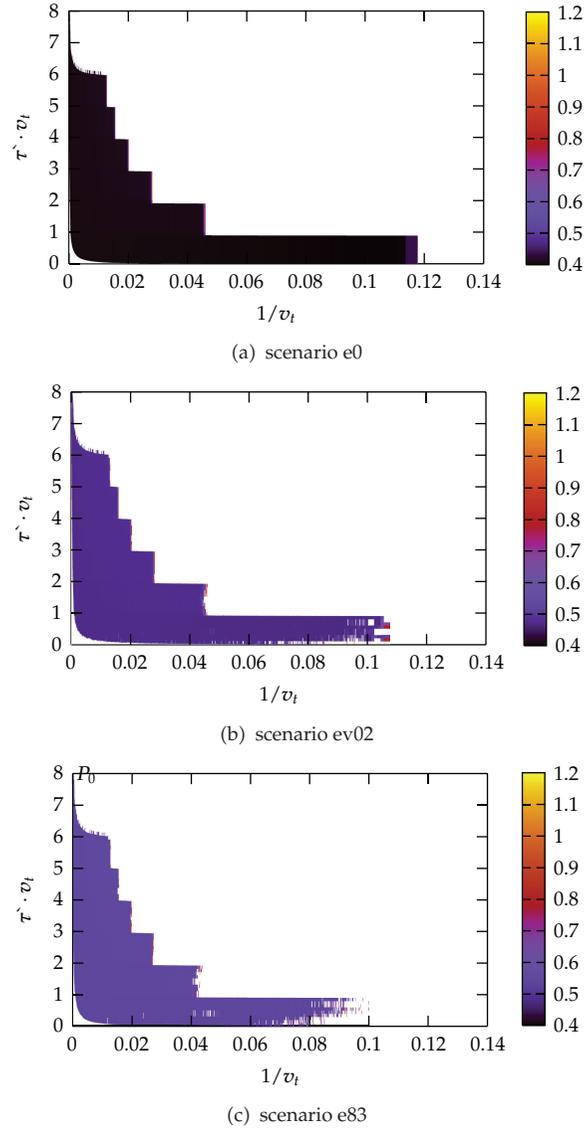


Figure 12: Stability region for scenarios e0, ev02, and e83.

Then, in Internet, the NCS sampling rate follows digitalization criteria, in place of transmission criteria, and transmission rate can be modeled by the transport protocol, according to network conditions.

7. Conclusions and Future Work

A test platform has been presented to study relationships between control and network, and to develop new codesign strategies for packet-based NCSs in shared networks. Shared network situations are normalized and have provided the basis for the realization of

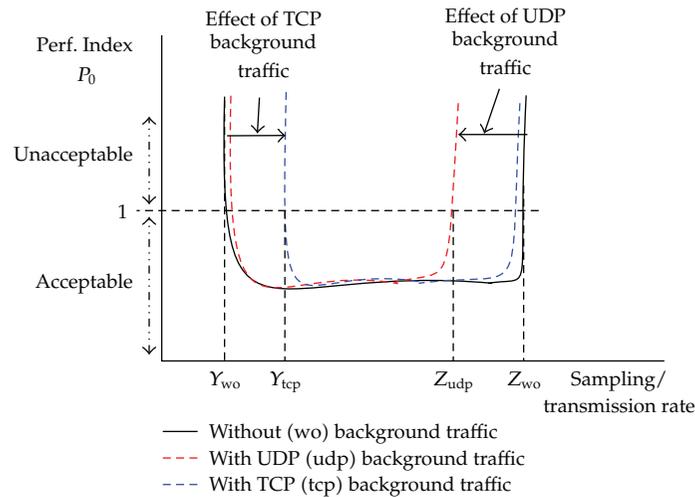


Figure 13: Relationship between performance and transmission rate with background traffic.

preliminary simulations to improve control/network codesign. Interesting remarks and considerations are obtained.

The transmission rate and packet structure are the main control parameters that can influence the network variables such as delay and packet losses. So far, the transmission rate was selected to minimize the quantization error criteria, while looking for a low transmission rate to maintain stability based on codesign studies with no best-effort background traffic. In Internet, the most used background traffic is TCP, with best-effort feature, and conclusions of this work are that in this case, it is recommended to use high transmission rates but lower than available capacity of the link.

Thus, transport protocol has a lot of importance in the stability of an Internet shared NCS, and codesign strategies are of interest to maintain performance indexes in an acceptable way. The transport protocol features are:

- (i) packet-loss-based flow-control,
- (ii) best effort and fairness to compete with TCP flows,
- (iii) a data priority schema to compete with UDP or other supermedia flows.

Transport protocols with these features can be tested in the *transport control sublayer* of the presented test platform, and data priority can be developed in the *network/control codesign sublayer*.

Future lines involve the design, development, and test of new transport schemes and codesign solutions to share the Internet or IP networks with certain guarantees of fairness and acceptable performance indexes for the control. TCP-friendly protocols for the transport scheme, and deadband solutions for the data priority problem, are a good start. Studies for packet structures are an interesting line of research, because they are involved in the packet size, packet efficiency and data priority of an NCS through Internet.

Acknowledgments

This work has been partially supported by Spanish Science and Technology Ministry in the Projects nos. CICYT DPI2010-20466-C02-01, and DPI2008-06738-C02-03 and by the Galician Industry and Research Ministry at the same country, under the Project no. 08DPI-011303PR.

References

- [1] A. Ambike, W. J. Kim, and K. Ji, "Real-time operating environment for networked control systems," in *Proceedings of the American Control Conference (ACC '05)*, pp. 2353–2358, June 2005.
- [2] V. T. Minh and F. M. Hashim, "Adaptive teleoperation system with neural network-based multiple model control," *Mathematical Problems in Engineering*, vol. 2010, Article ID 592054, 15 pages, 2010.
- [3] M. García-Rivera and A. Barreiro, "Analysis of networked control systems with drops and variable delays," *Automatica*, vol. 43, no. 12, pp. 2054–2059, 2007.
- [4] E. Delgado, A. Barreiro, and M. Diaz-Cacho, "Delay and its timederivative dependent stability of teleoperation systems," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '10)*, pp. 137–142, 2010.
- [5] F. L. Lian, J. Moyne, and D. Tilbury, "Network design consideration for distributed control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 2, pp. 297–307, 2002.
- [6] F.-L. Lian, J. K. Yook, D. M. Tilbury, and J. Moyne, "Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 1, pp. 12–24, 2006.
- [7] D. Georgiev and D. M. Tilbury, "Packet-based control," in *Proceedings of the American Control Conference (AAC '04)*, pp. 329–336, July 2004.
- [8] Y. B. Zhao, G. P. Liu, and D. Rees, "Design of a packet-based control framework for networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 859–865, 2009.
- [9] M. Diaz-Cacho, A. Barreiro, and M. Rivera, "Bidireccionalidad y eficiencia en el transporte de datos de teleoperación a través de redes ip," *Revista Iberoamericana de Automática e Informática Industrial*, vol. 7, no. 2, pp. 99–110, 2010.
- [10] M. Diaz-Cacho, A. Barreiro, and M. Rivera, "Bidirectionality in ip based teleoperation," in *Proceedings of the 4th International Scientific Conference on Physics and Control (PHYSICON '09)*, 2009.
- [11] Y. B. Zhao, G. P. Liu, and D. Rees, "Packet-based deadband control for internet-based networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1057–1067, 2010.
- [12] G.-P. Liu, Y. Xia, D. Rees, and W. Hu, "Design and stability criteria of networked predictive control systems with random network delay in the feedback channel," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 37, no. 2, pp. 173–184, 2007.
- [13] Y. B. Zhao, G. P. Liu, and D. Rees, "Integrated predictive control and scheduling co-design for networked control systems," *IET Control Theory & Applications*, vol. 2, no. 1, pp. 7–15, 2008.
- [14] G. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic behavior of networked control systems," in *Proceedings of the IEEE International Conference on Control Applications*, pp. 1448–1453, 1999.
- [15] D. Hristu-Varsakelis and P. R. Kumar, "Interrupt-based feedback control over a shared communication medium," in *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 3223–3228, December 2002.
- [16] M. S. Branicky, S. M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 1211–1217, December 2002.
- [17] M. S. Branicky, V. Liberatore, and S. M. Phillips, "Networked control system co-simulation for co-design," in *Proceedings of the American Control Conference*, pp. 3341–3346, June 2003.
- [18] Z. Li and M.-Y. Chow, "Adaptive multiple sampling rate scheduling of real-time networked supervisory control system—part 1," in *Proceedings of the IEEE 32nd Annual Conference on Industrial Electronics (IECON '06)*, pp. 4604–4609, 2006.
- [19] B. Klingenberg, U. Ojha, and M.-Y. Chow, "Predictive control of multiple ugv's in a ncs with adaptive bandwidth allocation," in *Proceedings of the 35th Annual Conference of IEEE Electronics (IECON '09)*, pp. 3027–3032, 2009.
- [20] S. L. Dai, H. Lin, and S. S. Ge, "Scheduling-and-control codesign for a collection of networked control systems with uncertain delays," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 1, pp. 66–78, 2010.

- [21] Y. Tipsuwan and M. Y. Chow, "Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation—part I: networked control," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 6, pp. 1218–1227, 2004.
- [22] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proceedings of ACM SIGCOMM Conference*, pp. 43–56, September 2000.
- [23] E. Kohler, M. Handley, and S. Floyd, "Designing dccp: congestion control without reliability," in *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '06)*, pp. 27–38, ACM, New York, NY, USA, 2006.
- [24] P. X. Liu, M. Meng, X. Ye, and J. Gu, "An UDP-based protocol for Internet robots," in *Proceedings of the 4th World Congress on Intelligent Control and Automation*, pp. 59–65, June 2002.
- [25] J. R. Hartman, *Networked control system co-simulation for co-design: theory and experiments*, Ph.D. dissertation, Western Reserve University, USA, 2004.
- [26] J. R. Hartman, M. S. Branicky, and V. Liberatore, "Time-dependent dynamics in networked sensing and control," in *Proceedings of the American Control Conference (ACC '05)*, pp. 2925–2932, June 2005.
- [27] P. Martí, J. Yépez, M. Velasco, R. Villà, and J. M. Fuertes, "Managing quality-of-control in network-based control systems by controller and message scheduling co-design," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 6, pp. 1159–1167, 2004.
- [28] G. F. Franklin, D. J. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, Prentice Hall, 3rd edition, 1997.
- [29] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," vol. 28, no. 4, pp. 303–314.
- [30] M. Diaz-Cacho Medina, "Ncs—network codesign in ns2," 2011, <http://ncs-ns2.sourceforge.net/>.
- [31] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–97, 2001.
- [32] P. G. Otañez, J. R. Moyne, and D. M. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proceedings of the American Control Conference*, vol. 4, pp. 3015–3020, May 2002.
- [33] R. Wirz, R. Marin, M. Ferre, J. Barrio, J. M. Claver, and J. Ortego, "Bidirectional transport protocol for teleoperated robots," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3772–3781, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

