3

Networked Control Systems: Emulation-based Design

Mohammad Tabbara[1], Dragan Nešić[1], and Andrew R. Teel[2]

[1] University of Melbourne, Parkville, Victoria 3052, Australia
m.tabbara@ee.unimelb.edu.au, d.nesic@ee.unimelb.edu.au
[2] University of California, Santa Barbara, CA 93106, USA
teel@ece.ucsb.edu

Abstract. A common approach to the implementation of digital systems is through the *emulation* of idealized continuous-time blocks in order to be able to leverage the rich expanse of results and design tools available in the continuous-time domain. The so-called sampled-data systems are now commonplace in practice and rely upon results that ensure that many properties of the nominal continuous-time system, including notions of stability, are preserved under sampling when certain conditions are verified. In analogy with (fast) sampled-data design, this chapter explores an emulation-based approach to the analysis and design of networked control systems (NCS). To that end, we survey a selection of emulation-type NCS results in the literature and highlight the crucial role that *scheduling* between disparate components of the control systems plays, above and beyond sampling. We detail several different properties that *scheduling protocols* need to verify together with appropriate bounds on inter-transmission times such that various notions of input–output stability of the nominal "network-free" system are preserved when deployed as an NCS.

Keywords. Nonlinear systems robust stability, scheduling, emulation-based design.

3.1 Introduction

Control of a system is to influence its behavior to achieve a desired goal, often, through the use of feedback. Diagrammatically, we are often concerned with the setup depicted in Fig. 3.1: analysis of plant P with (vector) output y and design of a controller C with a (vector) control u to achieve a desired closed-loop behavior, typically, a notion of stability.

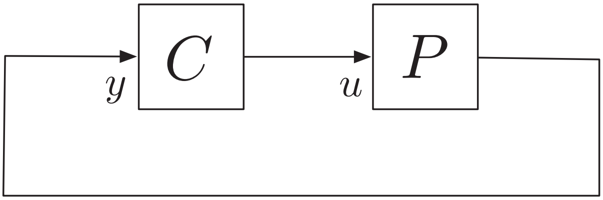

Fig. 3.1. Conceptual block diagram of feedback control

The interconnection of physical signals between controller and plant is seldom as elementary as that depicted in Fig. 3.1. Many properties of the plant including its physical size, complexity and mobile nature require the distribution of the control and observation task across multiple spatially separated *nodes*, including actuators, sensors and devices that compute the control law, connected via a *network*. For example, the system in Fig. 3.1 may potentially be implemented as in Fig. 3.2, using two output-feedback controllers C_1, C_2 and two sensors that transmit output values y_1, y_2 across a network to both controllers. Note that this implementation is suggested without specific reference to how and when and under which constraints this exchange of information takes place.

Abstractly, any set of communication channels together with a connection topology and constraints on the exchange of information across the channels that prescribe how and when information can be exchanged between nodes can be referred to as a network.

In this chapter, we restrict our attention to systems with nodes connected via a single shared communication channel or *bus* as in Fig. 3.3. The control law, plant, nodes, the bus itself and the *protocol* that describes how and when

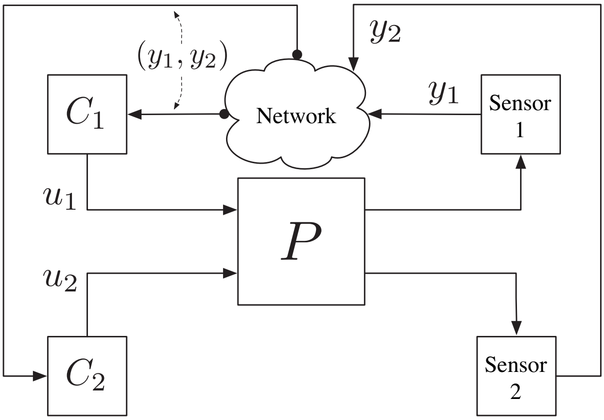

Fig. 3.2. A potential 4 node implementation of the system in Fig. 3.1 as an NCS

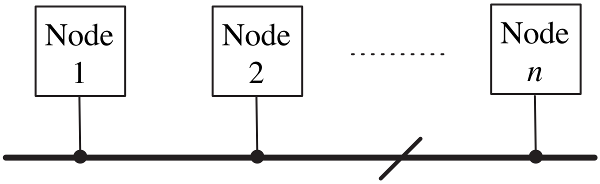

Fig. 3.3. Nodes connected with a bus network topology

information can be exchanged amongst nodes are collectively referred to as the *networked control system* (NCS).

Central to the study of NCS is the analysis and design of scheduling protocols. NCS depart from the use of dedicated point-to-point links for connectivity amongst nodes replacing some or all links with a shared network channel.

As in traditional data networks, the problem of arbitrating multiple access on the network becomes an issue, motivating the discussion of the *scheduling* of nodes and the design and analysis of scheduling protocols suitable for NCS applications. By scheduling, we mean the transmission of information across a link in the form of a discrete packet or frame.

Canonical NCS examples include so-called by-wire systems: drive-by-wire and fly-by-wire with analogies in industrial applications. Here, the *network* in NCS is thought of as in the sense of a traditional data (computer) network but the "network" may exist at a lower level of abstraction as in, for example, embedded digital control systems.

Example 3.1 (Embedded digital control systems). Transmission of controller and sensor values to and from the device executing the control law is governed by protocols of an electrical bus, e.g., a PCI bus, and typically, the scheduler of an operating system. Even if the underlying control system employs point-to-point connections from nodes to the controller, communication within the controller and its constituent components are subject to the communication constraints of various electrical buses and the operating system. □

Example 3.1 is one of the strongest motivations for studying NCS as presented in this chapter. It is perhaps taken for granted that the digital control systems designed and deployed in industry will continue to behave like their idealized continuous-time (resp., discrete-time) counterparts, besides the effects of sampling and quantization. As control systems increase in size and complexity and the levels of component integration increase, the flow of data between elements of the system is subject to constraints similar to that of a "real" network. Indeed, components of systems based on the PCI Express® architecture communicate via a switched serial network. Regardless of how controllers and sensors are connected, at least internally, every non-trivial digital control system can be thought of as an NCS.

From designs based on traditional wireless and wireline networks to the growing internal complexity of "un-networked" control systems, an increasing number of practical NCS implementations and their respective traffic scheduling protocols now exist. Standards-based component connectivity offers lower implementation costs, greater interoperability and a wide range of choices in developing control systems. The price paid for these advantages is the added complexity in the initial design and analysis of NCS. As alluded to earlier, part of this complexity comes in the form of issues of arbitration of network access amongst links, or *scheduling*, which is of fundamental importance. But above and beyond scheduling, NCS also presents the designer with the limitations of

(a) finite bandwidth of communication channels;
(b) finite precision of encoding and decoding schemes for transmitted information;
(c) pure (propagation) delays of channels;
(d) and data dropouts from unreliable channels.

These limitations are not mutually exclusive, however. As transmission rates increase, and with frame and packet sizes well in excess of machine (CPU) precision, effects of quantization and pure delay play an increasingly diminishing role in the analysis of most NCS and we forgo their consideration in this chapter. We will, however, examine models of data dropouts and unreliable channels with the Ethernet and the so-called p-persistent collision-sense multiple access (CSMA) as prime examples of such channels.

3.2 Overview of Emulation-based NCS Design

3.2.1 Principles of Emulation-based NCS Design

As stated in the introduction, scheduling and scheduling protocols are an integral part of NCS design. A survey of scheduling and various scheduling protocols is provided in [17] and stability and performance results of NCS have been examined in [8, 9, 15, 16, 17, 19]. An elementary example of a scheduling protocol, round-robin (RR), grants network access to NCS components in sequential, round-robin fashion and is used almost exclusively in practice. The aforementioned works present various alternative protocols that demonstrate a performance gain over RR in simulations and, in special cases, demonstrate the superiority of the alternative protocols analytically. The NCS design approach adopted in [8, 9, 12, 15, 16, 18], and in this chapter consists of the following steps:

(1) design a stabilizing controller ignoring the network;
(2) choose an appropriate scheduling protocol;

(3) and analyze the robustness of stability with respect to effects that scheduling within a network introduce.

The principal advantage of this approach is its simplicity – the designer of the NCS can exploit familiar tools for controller design and select an appropriate scheduling protocol and transmission rate such that the desired properties of the network-free system are preserved.

This chapter will introduce and characterize the various classes of admissible protocols for which stability results are developed but it is important to note that when the network-free system verifies a nominal stability property and an admissible protocol is chosen, stability of the resultant NCS can be achieved through sufficiently high transmission rates (or equivalently, sufficiently low inter-transmission times). Moreover, stability (robustness) properties of the NCS are actually parameterized by the transmission rate and hence, step (3) in the design process can be reinterpreted as:

(3′) choose a transmission rate (above requisite minimum) to achieve a desired degree of robust stability.

Results will be presented where this design approach is adopted with various notions of transmission rate (minimum or expected) and robust stability (uniform global exponential or asymptotic stability, L_p or L_p in-expectation or input-to-state stability).

3.2.2 Results in Perspective

Consider the following LTI control system:

$$\dot{x}_P = A_P x_P + B_P u, \qquad \dot{x}_C = A_C x_C + B_C y, \tag{3.1}$$

$$y = C_P x_P, \qquad u = C_C x_C, \tag{3.2}$$

where $x_P \in \mathbb{R}^{n_P}$, $x_C \in \mathbb{R}^{n_C}$, y, and u denote, respectively, plant state, controller state, plant output, and control, and where u has been designed ignoring the network as outlined in the previous section. In the presence of a network and an associated scheduling protocol, y and u cannot be continuously transmitted between the plant and controller. The network introduces the following limitations:

(a) transmissions occur only at specific transmission instants $\{t_i\}_{i=0}^{\infty}$; and
(b) only one logical component of the NCS is allowed to transmit (broadcast) data onto the network at a given transmission instant t_i, e.g., for a 3-output 2-input system, one component of $y = (y_1, y_2, y_3)$, $u = (u_1, u_2)$ can be transmitted.

Let $\hat{y}$ denote the "stand-in" for y available to and maintained by the device(s) that compute the control law and $\hat{u}$ denote the "stand-in" for u available to and maintained by the device(s) that actuate the plant. In effect, the NCS for the network-free system is described by

$$\dot{x}_P = A_P x_P + B_P \hat{u}, \qquad \dot{x}_C = A_C x_C + B_C \hat{y}, \tag{3.3}$$

$$y = C_P x_P, \qquad u = C_C x_C. \tag{3.4}$$

In analogy with zero-order hold sampling, $\hat{y}$ and $\hat{u}$ can be held constant between transmission instants and "reset" or updated with components of u and y as those become available and transmitted. Fig. 3.4 illustrates the situation for an NCS where two outputs are alternately transmitted in RR fashion across the network to the device(s)[3] that compute the control law and actuate the plant *at transmission instants*. RR is only one example of a scheduling protocol amongst several that we consider and one of the primary aims of the chapter will be to characterize protocol properties that capture the effects of the protocol on NCS stability.

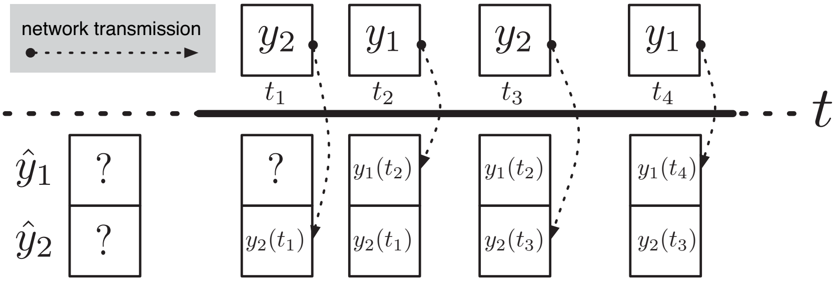

Fig. 3.4. RR scheduling of two-outputs with a zero-order hold $\hat{y}$ update policy

Although a scheduling protocol determines how the transmission of plant measurements and control values are arbitrated at transmission instants, it is useful to think of the scheduling protocol in terms of the effects on the *error* that a network induces compared to the network-free system. Indeed, if we define

$$e = \begin{pmatrix} e_y \\ e_u \end{pmatrix} = \begin{pmatrix} \hat{y} - y \\ \hat{u} - u \end{pmatrix}, \tag{3.5}$$

we can rewrite (3.3) and (3.4) between transmissions as

$$\begin{aligned} \dot{x}_P &= A_P x_P + B_P u + B_P e_u, & \dot{x}_C &= A_C x_C + B_C y + B_C e_y, \\ y &= C_P x_P, & u &= C_C x_C, \\ \dot{\hat{u}} &= 0, & \dot{\hat{y}} &= 0, \end{aligned}$$

and hence,

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}, \tag{3.6}$$

[3] Since data is presumed to be broadcast across the network, the number of controller-actuator devices that actuate the plant is immaterial so long as they adopt identical policies for updating their copies of $\hat{y}$.

where

$$A_{11} = \begin{bmatrix} A_P & B_P C_C \\ B_C C_P & A_C \end{bmatrix}, \quad A_{12} = \begin{bmatrix} B_P \; B_C \end{bmatrix}, \tag{3.7}$$

$$A_{21} = -\begin{bmatrix} C_P \; 0 \end{bmatrix} A_{11}, \quad A_{22} = -\begin{bmatrix} C_P \; 0 \end{bmatrix} A_{12}\,. \tag{3.8}$$

These equations describe how the state and NCS error evolves between transmissions and it is clear that components of e are reset or experience "jumps" *at transmissions instants.* For example, let $e_{y,j} = \hat{y}_j - y_j$. Ignoring the effects of quantization and delay, if the jth component of y is transmitted at the ith transmission instant we have

$$\hat{y}_j(t_i) \leftarrow y_j(t_i) \iff e_{y,j}(t_i) \leftarrow 0\,. \tag{3.9}$$

Hence, *the effect of the scheduling protocol is to reset components of the NCS error*[4] *at transmission instants.* An NCS model in this fashion is thus completely prescribed by:

(a) NCS continuous-time dynamics as in (3.6) and depicted conceptually in Fig. 3.5;
(b) a sequence of increasing transmission instants $\{t_i\}_{i=0}^{\infty}$; and
(c) a scheduling protocol, or *error reset map* that is described via its effect on the error, e, at transmission instants.

Regarding the NCS continuous-time dynamics as fixed, we would like to characterize the sequence of transmission instants or, equivalently, the sequence of inter-transmission intervals and the set of protocols for which we can conclude that the NCS state (x, e) is stable in an appropriate sense. The origins of emulation-based NCS design in this sense begin with the pioneering work of Walsh *et al.* in [15] and [16] where NCS models in the form of (3.6) and its nonlinear counterpart were presented, together with conditions on the *maximum allowable transmission interval* (MATI) such that the resultant NCS was uniformly globally asymptotically or exponentially stable (UGAS, UGES) when using the RR or *maximum-error-first try-once-discard* (TOD) scheduling protocols. We defer a detailed discussion of these and other protocols until Section 3.3.2 and outline results in the spirit of those presented in [15, 16, 19].

Let $e \in \mathbb{R}^{n_e}$ and $x \in \mathbb{R}^{n_x}$. The following class of nonlinear systems was considered in [15]:

$$\begin{aligned} \dot{x} &= f(t, x, e), \\ \dot{e} &= g(t, x, e), \end{aligned} \tag{3.10}$$

with the shorthand notation:

$$\dot{z} = h(t, z), \tag{3.11}$$

where $z = (x^T \; e^T)^T$.

[4] Ordinarily and as in (3.9), the result of the transmission is to reset a component of error to zero, though we stress that for many of the results outlined in the chapter, this assumption is not necessary.

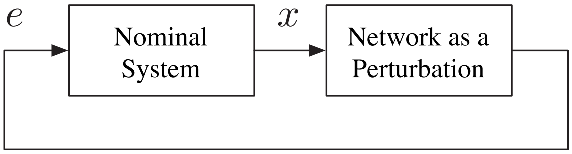

Fig. 3.5. Interconnection of signals in NCS dynamics

The Lipschitz constants for f, g and h are denoted, respectively, by k_f, k_g and k_h; that is, the right-hand side in (3.11) is assumed to be globally Lipschitz, uniformly in t. The class of linear systems (3.6) with the obvious shorthand:

$$\dot{z} = Az \tag{3.12}$$

was considered in [16, 19].

It is supposed in [15] that there exists a continuously differentiable Lyapunov function V such that the system (3.10) satisfies:

$$c_1|x|^2 \le V(t,x) \ \le \ c_2|x|^2 \qquad \text{for all } x \in \mathbb{R}^{n_x} \text{ and } t \in \mathbb{R}, \tag{3.13}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t,x,0) \le -c_3|x|^2 \qquad \text{for almost all } x \in \mathbb{R}^{n_x} \text{ and } t \in \mathbb{R}, \tag{3.14}$$

$$\left|\frac{\partial V}{\partial x}\right| \le c_4\,|x|\,, \tag{3.15}$$

where c_1, c_2, c_3, c_4 are positive constants. A similar condition was used in [16, 19] for the linear system (3.6). Indeed, it was assumed that for some positive definite and symmetric matrix Q there exists a positive definite and symmetric matrix P that solves the Lyapunov matrix equation[5]:

$$A_{11}^T P + PA_{11} = -Q\ . \tag{3.16}$$

It is obvious that (3.16) implies that (3.13)–(3.15) are satisfied for the linear system (3.6), $V(x) = x^T Px$ and

$$c_1 = \lambda_{\min}(P);\ c_2 = \lambda_{\max}(P);\ c_3 = \lambda_{\min}(Q);\ c_4 = 2\lambda_{\max}(P), \tag{3.17}$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the minimum and maximum singular value of a symmetric matrix, respectively. For linear systems, we can let

$$k_h \ = \ k_f \ = \ k_g \ = \ |A|\,. \tag{3.18}$$

A bound on MATI that guarantees the stability of the linear system (3.6) with the RR and TOD protocols was obtained in [16, 19]. We denote bounds computed in [16, 19], respectively, as τ_*^{RR} and τ_*^{TOD} for the RR and TOD

[5] The results in [16] are only presented for the special case $Q = I$. The result with general Q is presented in [19].

protocols. Similar bounds were obtained in [15] for nonlinear systems (3.10) with the RR and TOD protocols, where[6] the bounds obtained are also such that $\tau_*^{RR} = \tau_*^{TOD}$. The bounds in [15, 16, 19] can be expressed as:

$$\tau_*^{RR} = \tau_*^{TOD} = \frac{c_3}{M\ell(\ell+1)k_h k_f c_4} , \tag{3.19}$$

where the value of the constant M is different for linear and nonlinear systems and ℓ denotes the number of nodes that participate in scheduling. For nonlinear systems, we have

$$M = M_{NL} = 16 \left(\frac{c_2}{c_1}\right)^{3/2} \left(\sqrt{\frac{c_2}{c_1}} + 1\right) , \tag{3.20}$$

established in [15]. Analogously in [16, 19], the following is obtained for linear systems

$$M = M_L = 8\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} \left(\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} + 1\right) , \tag{3.21}$$

where the meaning of all constants in (3.19) is explained through (3.17) and (3.18). These MATI bounds obtained in [15, 16, 19] do not differentiate between RR and TOD; that is $\tau_*^{TOD} = \tau_*^{RR}$.

In general, intuition suggests that MATI bounds should be protocol-dependent. Significant improvements upon these MATI bounds were made in [8] by efficiently capturing protocol properties through protocol-specific Lyapunov functions and characterizing the effects of transmission errors through L_p gains. Essentially, UGES and L_p input–output stability is with a MATI of:

$$0 < \tau < \frac{1}{L} \ln \left(1 + \frac{1-\theta}{\gamma/L + \theta}\right) , \tag{3.22}$$

where $\theta \in [0, 1)$ characterizes the ability of the protocol to reduce network error at transmission instants while $L > 0$ describes the speed of the network-error dynamics, and $\gamma > 0$ captures the effect of network-error on the behavior of the ideal system through an L_p gain. In particular, τ is protocol-dependent through θ – the better the protocol is at reducing network-error at transmission instants, the larger the MATI bound is, and hence, the less frequent transmissions have to be to guarantee stability of the NCS.

3.3 Modeling Networked Control Systems and Scheduling Protocols

The premise of networked control systems (NCS) is to spatially distribute a "traditional" control system across a number of *nodes* that will exchange

[6] Note that we do not use different notation for MATI bounds for linear and nonlinear systems, although they are different in general. This is because it always will be clear from the context which bound we mean.

data subject to the constraints of a shared data channel. These nodes include sensors, actuators and units that compute various control laws and the data channel is typically a wireless or wireline computer network, many examples of which can be found in [14].

Computer networks and communication systems present rich and sophisticated models of varying degrees of complexity, within stochastic and deterministic settings, and of various underlying physical communication media. For the vast majority of computer networks described in [14], the primary constraint on the exchange of data between nodes is that the respective channels are exclusive. This means that the attempt of more than one node to transmit data at a given time will result in data loss, i.e., a collision. Collisions can be prevented by arbitration of network access through the use of scheduling protocols that decide which node(s) can transmit and at what times.

The network models presented in this chapter aim to capture the essential aspects of control over networks in the context of several important settings:

(a) locally[7] arbitrated network access without packet dropouts;
(b) arbitrated network access with and without packet dropouts; and
(c) unarbitrated network access with and without packet dropouts.

Arbitration takes place through the use of a scheduling protocol adopted by every node in the network. A protocol can be thought of as a map

$$h\colon W \to \{1, \dots, \ell\} \tag{3.23}$$

that selects the node currently being allowed to transmit and an associated dynamical system that evolves the scheduler state variable $\omega \in W$. For spatially separated nodes, this generally means that each node must maintain a copy of the state ω that is evolved identically by the node (local knowledge with globally-known inputs), or, ω is known globally and updated in a distributed fashion. Such protocols are often referred to as contentionless protocols. For example, labeling the NCS nodes $\{a_1, a_2, \dots, a_\ell\}$, round-robin scheduling would entail apportioning the channel's time, $[0, \infty)$, into slots $\{s_1 := [t_0, t_1), s_2 := [t_1, t_2), \dots, \}$ such that node a_i is permitted to transmit during slot $s_{i+k\ell}$, $k = \{0, 1, \dots\}$. Depending on the context, this scheduling protocol is also known as time-division multiplexing or token ring and relies on each node being able to count transmissions. In this case,

$$\omega = \text{number of transmissions from some initial time.}$$

For networks with a large number of nodes, mobile nodes that are spatially separated across varying distances or networks with a varying number of nodes, it may be impractical or impossible to keep ω, the state information, synchronized across all nodes.

[7] By "locally" we mean that the arbitration process takes place without the exchange of global arbitration information prior to network access, e.g., a priority field.

The alternative is unarbitrated access in the sense that there is no global policy to enforce exclusive network access for a given node at a transmission instant. In particular, collisions may occur, and have to be detected and recovered from. The number that occur can often be reduced by employing various heuristics using data available to each node *locally.* Concrete and familiar examples of this approach include the family of carrier-sense multiple access protocols (CSMA) exemplified by Ethernet, p-persistent CSMA (Bluetooth, 802.11a/b/g) and variants of ALOHA. See [14] for an overview of these protocols and their operational characteristics.

Thus far, the discussion holds true for both computer and control networks. Where computer networks and control networks differ radically is in access patterns – ideally, a continuous-time control system would have nodes constantly transmitting sensor values and constantly receiving control values, in complete contrast to the usual assumption of access in short and irregular bursts for nodes in a computer network. Stated explicitly, *we assume continuous-time controllers and plant outputs are such that there will always be data to transmit when the network channel becomes idle.*

This assumption applies to all forms of network access in NCS, the key difference being that the unarbitrated network access does not enforce a particular choice of which link to transmit when the channel becomes idle whereas global arbitration would. We present a unified approach for the analysis of NCS both for ideal channels and in the presence of random packet dropouts and random inter-transmission times – effects that are essentially attributes of non-ideal or *stochastic* network channels.

We assume that every link in the NCS contests access to the network at either predetermined time-slots or times at which the network is sensed to be idle. This results in two potential sources of randomness:

(a) At any idle time or transmission slot, either some node j transmits successfully or a collision results or the transmitted packet is dropped. Denoting the probability that a packet is dropped or a collision occurs by p_0, we will always assume that the probabilities of successful transmission of links is identically equal to $(1-p_0)/\ell$ for an ℓ-link NCS without global arbitration. While this is not strictly necessary in our analyses, there is no reason to statically (off-line) favor any one link over another during contention by adjusting transmission-success probabilities. Contentionless protocols do, however, enforce a particular choice of which link to transmit in a given slot eliminating the possibility of a collision.

(b) Sensing the network as being idle, synchronizing to transmissions time-slots or else randomly waiting for a period of time after any of these events to reduce the likelihood of collisions are common features of network protocols. These uncertainties can be faithfully modeled with a stochastic (renewal) process. For the set of protocols we discuss, it is sufficient to restrict our attention to Poisson processes with some intensity λ or a class of renewal processes where inter-transmission times are uniformly bounded, i.e., by the MATI.

3.3.1 Scheduling and a Hybrid System Model for NCS

We model the NCS as a so-called jump-continuous (hybrid) system, where jump times and the associated jump or reset maps are both potentially random but not necessarily so. Our NCS model incorporates the effects of exogenous perturbations w as first presented in [8]. As alluded to earlier, the model we present is general enough to examine several scheduling alternatives with and without packet dropouts when inter-transmission times are either uniformly bounded with a MATI or random.

Node data (controller and sensor values) are transmitted at (possibly) random transmission instants $\{t_0, t_1, \ldots, t_i\}, i \in \mathbb{N}$ and our NCS model is prescribed by the following dynamical and jump equations. In particular, for all $t \in [t_{i-1}, t_i]$:

$$\dot{x}_P = f_P(t, x_P, \hat{u}, w), \tag{3.24}$$

$$\dot{x}_C = f_C(t, x_C, \hat{y}, w), \tag{3.25}$$

$$u = g_C(t, x_C), \quad y = g_P(t, x_P), \tag{3.26}$$

$$\dot{\hat{y}} = 0, \quad \dot{\hat{u}} = 0^{8}, \quad \dot{\hat{e}} = 0, \tag{3.27}$$

and at each transmission instant t_i,

$$e(t_i^+) = Q_i(e(t_i))e(t_i)^{9}, \tag{3.28a}$$

or

$$\begin{aligned} e(t_i^+) &= Q_i(\hat{e}(t_i))e(t_i), \\ \hat{e}(t_i^+) &= \Lambda\big(i, (I - Q_i(\hat{e}(t_i)))e(t_i), \hat{e}(t_i)\big). \end{aligned} \tag{3.28b}$$

The effect of the protocol on the error is such that if the mth to nth nodes are successfully transmitted at transmission instant t_i the corresponding components of error, $e_n, \ldots, e_m$, experience a "jump". It may be the case that a single logical node (a "link") consists of several sensors or several actuators or both with the transmission of that link having the effect of setting multiple components of e to zero. It may also be the case that the network allows the transmission of more than one node at each transmission and our model allows for this extra degree of freedom. For transmission from mth node to nth node, we will always assume that $e_n(t_i^+), \ldots, e_m(t_i^+) = 0$ and hence, $Q_i(\cdot)e = [a_{kj}]e$, where $a_{kj} = 0$ for $k = j \in [n, m] \cup \{k \neq j\}$ and 1 elsewhere. We group the

[8] The assumption that $\dot{\hat{y}}$ and $\dot{\hat{u}}$ are zero simplifies the presentation and is not strictly necessary. Non-zero choices correspond to schemes that predict y and u between transmissions in an open-loop sense.

[9] Given $t \in \mathbb{R}$ and a piecewise continuous function $f\colon \mathbb{R} \to \mathbb{R}^n$, we use the notation $f(t^+) = \lim_{s \to t, s > t} f(s)$.

nodes that are transmitted together into logical links, associating a partition of size s_i, denoted by $\mathbf{e}_i = (e_{i1}, e_{i2}, \ldots, e_{is_i})$, of the error vector e such that we can write $e = (\mathbf{e}_1, \ldots, \mathbf{e}_\ell)$. We say that the NCS has ℓ links and $\sum_{i=1}^{\ell} s_i$ nodes. Note that this is purely a notational convenience and simplifies the description of scheduling protocols and the NCS itself.

The two alternative forms of the error jump-map (3.28a) and (3.28b) refer to two different situations with respect to the scheduler state ω in the abstract description of a scheduling protocol given in (3.23):

(a) $\omega \equiv (i, e)$ in (3.28a), where $Q_i(\cdot)$ may be a *random* jump map – in particular, Q_i may be the identity in the case where nothing was transmitted or a collision or dropout occurred.
(b) $\omega \equiv (i, \hat{e})$ in (3.28b), where $Q_i(\cdot)$ is an ordinary map and $\hat{e}$ is a state variable synchronously maintained and updated by all nodes.

In both cases, we refer to Q as the scheduling function and Λ as the decision-update function in (3.28b). The key difference between these two alternatives is the decision-vector $\hat{e}$. Special cases of $\hat{e}$-based scheduling were first considered in [18]. The model we introduced in [12] and described here formalizes the $\hat{e}$-based scheduling that was considered in [18] and it generalizes the NCS models considered in [8].

With respect to the available state-information, there are several alternatives as to what information the scheduler has available in making scheduling decisions prior to transmissions:

(a) (x, e, i) is known by all nodes;
(b) (e, i) is known by all nodes;
(c) i is known and any *broadcast* data becomes known *after transmissions*;[10]
(d) only i is known globally; or
(e) only local policies are adopted and no global information is used in scheduling.

These correspond to the following NCS scenarios:

(1) “Classical” control, that is, if (x, e, i) is known to all nodes prior to transmissions, transmissions would not be necessary as any of x, y, and u could be recovered.
(2) Each node can encode e into an arbitration field and participate in what is, in effect, a distributed scheduling decision, e.g., through binary countdown.
(3) Nodes only have i and local information available to make a scheduling decision, and once a transmission (broadcast) has taken place, are free to update their local information ($\hat{e}$) with the broadcast data.[11] To ensure

[10] This data can be used to evolve locally maintained state, e.g., $\hat{e}$.

[11] For reasons that shall become apparent, there is no loss of generality in assuming that the broadcast data is given by $(I - Q_i(\hat{e}(t_i)))e(t_i)$ – the component of error

that the nodes arrive at a unanimous decision, the update rule, and hence the local data is updated in the same fashion across all nodes.

(4) For situations (2)–(3) it is assumed that nodes can count the number of transmissions that have passed from some reference time and hence i is known. In this NCS scenario, no other data is known or maintained by nodes for scheduling purposes.

(5) Network access is, in effect, unarbitrated and access patterns are determined by local policy.

The maps prescribed by (3.28a) and (3.28b) are sufficiently general to capture the scenarios (2)–(5). We combine the controller and plant states into a vector $x = (x_P, x_C)$ and assuming g_P, g_C are continuous and a.e. (almost everywhere) C^1, for example, we can rewrite (3.24)–(3.28b) in a form more amenable to analysis:

$$\dot{x} = f(t, x, e, w), \quad t \in [t_{i-1}, t_i], \tag{3.29}$$

$$\dot{e} = g(t, x, e, w), \quad t \in [t_{i-1}, t_i], \tag{3.30}$$

and

$$e(t_i^+) = Q_i(e(t_i))e(t_i), \tag{3.31a}$$

or

$$\begin{aligned} e(t_i^+) &= Q_i(\hat{e}(t_i))e(t_i), \\ \hat{e}(t_i^+) &= \Lambda\big(i, (I - Q_i(\hat{e}(t_i)))e(t_i), \hat{e}(t_i)\big)\,, \end{aligned} \tag{3.31b}$$

where $x \in \mathbb{R}^{n_x}, e \in \mathbb{R}^{n_e}, w \in \mathbb{R}^{n_w}$, and $\hat{e} \in \mathbb{R}^{n_{\hat{e}}}$.

Implicit in this definition is that there are no (pure) propagation delays. Transmission at time t_i results in the instant reset of the relevant error component to zero. We appeal to the robustness properties verified by the class of systems considered to assert that the results in this chapter remain true for sufficiently small delays.

With respect to (3.24)–(3.28b) and (3.29)–(3.31b), we further assume that the sequence of (attempted) transmission times $\{t_i\}_{i\in\mathbb{N}}$ is such that $t_{i+1} - t_i$ is exponentially distributed for all i or satisfy $\epsilon < t_{j+1} - t_j \leq \tau$ for all $j \geq 0$ where $\tau > 0$ and $\epsilon > 0$.[12] The constant τ is the *maximum allowable transmission interval* (MATI).

3.3.2 NCS Scheduling Protocol Properties

We have previously described protocols in a general setting as maps that effect errors at transmission instants. We now aim to identify general protocol

that was reset to zero at the ith transmission instant and hence, appears as the only input in (3.28b).

[12] This ensures that Zeno solutions cannot occur. Zeno behavior occurs in hybrid systems when there are an infinite number of discrete transitions in a finite period of time.

properties that appropriately characterize protocol behaviors and that are able to parametrize NCS stability under appropriate conditions. Recall that by "protocol" we refer to both the maps of the form (3.31a) and (3.31b) as well as an associated sequence of transmission times $\{t_i\}_{i=0}^{\infty}$, where $t_{i+1} - t_i$ is either uniformly bounded or exponentially distributed.

We introduce several protocol properties that are phrased in terms of membership in the class of Lyapunov UGES (uniformly globally exponentially stable) protocols, the class of PE_T (persistently exciting) protocols, the class of almost surely Lyapunov UGES protocols and the class of almost surely (a.s.) covering protocols.

3.3.3 Lyapunov UGES and a.s. UGES Scheduling Protocols

Let $\mathbf{E}[\cdot]$ and $\mathbf{P}\{\cdot\}$ denote the expectation and probability operators and let $X \sim \mathrm{Exp}(\lambda)$ denote that X is an exponentially-distributed random variable with $\mathbf{E}[X] = 1/\lambda$. For purely deterministic maps and ignoring the dynamics introduced by (3.30), we can regard (3.31a) as a discrete-time system that captures the behavior of the scheduling protocol. The system is given by:

$$e^{+} = Q_i(e)e\,. \tag{3.32}$$

Maps of this form were used to capture the behavior of the protocol in [8] on an ideal network. Describing the protocol in this fashion allows one to speak of uniformly globally asymptotically and exponentially stable (UGAS and UGES) scheduling protocols whenever the associated discrete-time system (3.32) is UGAS or UGES. Beyond taxonomy, the notion of UGES and UGAS protocols and the construction of smooth Lyapunov functions for the associated UGAS and UGES discrete-time systems is central to the stability analysis approach developed in [8] and [9].

NCS employing UGES and UGAS protocols on non-ideal network channels are still subject to packet losses and varying inter-transmission times. By assigning a probability, p_0, to the event that the channel drops a packet, we model the behavior of the protocol on non-ideal channels in this section with jump maps of the form

$$\tilde{Q}_i(e)e = q_i Q_i(e) + (1 - q_i)e, \tag{3.33}$$

where q_i is an i.i.d.[13] sequence of Bernoulli random variables that model the dropout process of channel with $\mathbf{P}\{q_i = 1\} = 1 - p_0$. Depending on the specific system, the sequence of arrival times (transmission instants) $\{t_i\}_{i\in\mathbb{N}}$ are either random and defined inductively by:

$$t_0 = \tau_0,$$

where $\tau_0 \sim \mathrm{Exp}(\lambda)$ and for each $i > 0$,

[13] Independently identically distributed.

$$t_i = t_{i-1} + \tau_i,$$

$\tau_i \sim \mathrm{Exp}(\lambda)$, where the sequence $\{\tau_i\}$ is i.i.d., or inter-transmission times are uniformly (deterministically) bounded by a MATI.

As in (3.32), it becomes natural to define the associated auxiliary discrete-time system for (3.33):

$$e^+ = q_i Q_i(e)e + (1 - q_i)e, \quad i \in \mathbb{N}, \tag{3.34}$$

where the sequence $\{q_i\}$ is defined as in (3.33).

We introduce the following definition with respect to System (3.34).

Definition 3.1 (Almost surely Lyapunov UGES protocols). Let $W\colon \mathbb{N} \times \mathbb{R}^{n_e} \to \mathbb{R}_{\geq 0}$ be given and suppose that κ_i is a sequence of nonnegative i.i.d. random variables and $a_1, a_2 > 0$ such that the following conditions hold for the discrete-time system (3.34) for all $i \in \mathbb{N}$ and all $e \in \mathbb{R}^{n_e}$:

$$a_1|e| \leq W(i,e) \leq a_2|e| \tag{3.35}$$

$$W(i+1, \tilde{Q}_i(e)e) \leq \kappa_i W(i,e) \tag{3.36}$$

$$\mathbf{E}[\kappa_i] < 1\,. \tag{3.37}$$

Then we say that (3.34) (equivalently, the contentionless protocol) is *almost surely uniformly globally exponentially stable (a.s. UGES) with Lyapunov function W.* □

Before discussing implications of this definition, we present a motivating example:

Example 3.2 (Try-once-discard). The TOD protocol was introduced in [16] and can be expressed with a model of the form (3.34) where

$$Q_i(e) = (I - \Psi(e))$$

and $\Psi(e) = \mathrm{diag}\{\psi_1(e)I_{l_1}, \ldots, \psi_\ell(e)I_{l_\ell}\}$, with I_{l_j} identity matrices of dimension l_j and

$$\psi_j(e) = \begin{cases} 1, \text{ if } j = \min(\arg\max_j |\mathbf{e}_j|) \\ 0, \text{ otherwise.} \end{cases} \tag{3.38}$$

That is, TOD picks out the node with the largest magnitude of error for transmission. It was shown that TOD preserves stability properties of the network-free system in [15] (linear systems) and [8] (nonlinear systems with disturbance) for sufficiently small MATI. As in [8, Proposition 5], we set $W(i,e) = |e|$ and claim that TOD is a.s. Lyapunov UGES whenever the probability of a dropout, p_0 is such that

$$p_0 + (1 - p_0)\sqrt{\frac{\ell - 1}{\ell}} < 1. \tag{3.39}$$

□

The inequality (3.39) is a particular example of a more general condition that ensures that any Lyapunov UGES protocol in the sense of [8] is an a.s. Lyapunov UGES for sufficiently low probability of dropout. We first recall the definition of a Lyapunov UGES protocol:

Definition 3.2 (Lyapunov UGES protocols). A protocol (3.34) on an ideal channel ($p_0 = 0 \Rightarrow q_i = 1$) is said to be Lyapunov UGES in the sense of [8] if there exists $W\colon \mathbb{N} \times \mathbb{R}^{n_e} \to \mathbb{R}_{\geq 0}$, $a_1, a_2 > 0$, and $0 \leq \theta < 1$ such that for all $i \in \mathbb{N}$ and all $e \in \mathbb{R}^{n_e}$:

$$a_1|e| \leq W(i,e) \leq a_2|e| \tag{3.40}$$

$$W(i+1, Q_i(e)e) \leq \theta W(i,e)\,. \tag{3.41}$$

This definition admits the following proposition:

Proposition 3.1. Suppose that the protocol (3.34) on an ideal channel ($p_0 = 0 \Rightarrow q_i = 1$) is Lyapunov UGES. Then (3.34) is a.s. Lyapunov UGES on a non-ideal channel ($p_0 \geq 0$) if

$$p_0 + (1-p_0)\theta < 1. \tag{3.42}$$

Remark 3.1. The rationale for the introduction of the class of a.s. Lyapunov UGES protocols is to provide an analysis framework for Lyapunov UGES protocols capable of handling random packet dropouts – any Lyapunov UGES protocol is automatically an a.s. Lyapunov UGES protocol for sufficiently low p_0. In the case where inter-transmission times are uniformly bounded by a MATI and $p_0 = 0$, we recover the usual definition of Lyapunov UGES protocols as in Definition 3.2. □

Remark 3.2. The definition of Lyapunov UGAS and hence, a.s. Lyapunov UGAS protocols is analogous and we refer the reader to [9] for details and results. □

3.3.4 PE_T Scheduling Protocols

Intuition suggests that schemes such as TOD should perform better than RR, as the node with the greatest error is transmitted at each transmission instant. TOD is certainly implementable in variants of CAN[14] as the error can be encoded into an arbitration field[15] in a frame but no such arbitration is possible for wireless channels and, indeed, many wireline channels and hence, it is often unreasonable to assume knowledge of the entire error vector e in these contexts.

Several variants of TOD were introduced in [18] that "estimate" the error vector and were shown to outperform RR in simulations. Stability results

[14] Control Area Network.

[15] Specifically, through binary countdown – see [14] and [17] for details.

are also provided for linear systems that lead to conservative estimates on performance bounds. One model of NCS that accommodates these variants was proposed in [12] that is a special case of (3.29)–(3.31b).

The variants of TOD presented in [18] as well as the RR scheduling protocol satisfy the following property: *there is a fixed (finite) number of transmissions T such that all nodes of the NCS have transmitted within T transmissions.* This T is related to the notion of a node's "silent-time" in [18]. This property is the point of departure of this section, and for reasons that will become apparent, we call protocols that satisfy this property *uniformly persistently exciting scheduling protocols*, or simply, PE protocols. Whenever T is known, we say that the protocol is PE_T. Round-robin is the first example of a PE_T protocol.

Example 3.3 (Round-robin). Round-robin scheduling is employed in the token ring and token bus network protocols as well as (once) being the ubiquitous scheduling protocol of time-sharing operating systems. Each link of the network is assigned a unique index and links are "visited" in the order of index. Consider an ℓ-link NCS. In terms of NCS scheduling, the discrete-time system is a linear time-varying system where the protocol map has no dependence on state:

$$e^+ = (I - \Delta(i))e, \tag{3.43}$$

where $\Delta(i) = \mathrm{diag}\{\delta_1(i)I_{s_1}, \ldots, \delta_N(i)I_{s_\ell}\}$, and

$$\delta_k(i) = \begin{cases} 1, & \text{if } k-1 = i \bmod \ell \\ 0, & \text{otherwise.} \end{cases} \tag{3.44}$$

It was established in [8] that RR is a Lyapunov UGES protocol and that it preserves stability properties of the network-free system for high enough transmission rates. As the protocol does not depend on NCS state it makes RR easily implementable and is PE_T with $T = \ell$. □

PE in the sense we have described is verified by many network technologies. Ethernet and 802.11 are examples of CSMA/CD protocols where it is known (see [2], for instance) that for a finite number of users (links), the expected waiting time for a link is finite. We pursue a stochastic analogue of PE for such protocols in Section 3.3.5.

For a more formal characterization of the PE property, it can be shown that if we integrate the equations (3.30) and $\dot{\hat{e}} = 0$ on the interval $[t_{i-1}^+, t_i]$ and then apply the jump map (3.31b) at t_i, the NCS induces the following discrete-time system:

$$e^+ = (I - \Psi(i, \hat{e}))(e + d), \tag{3.45}$$

$$\hat{e}^+ = \Lambda(i, \Psi(i, \hat{e})(e + d), \hat{e}), \tag{3.46}$$

where d captures the inter-sample behavior of $e(\cdot)$. This idea of examining an induced discrete-time system to evaluate protocol properties was first used in [8] as outlined in Section 3.3.3 though used here with a key difference: for specific initializations $(k, e(k), \hat{e}(k))$ and specific (bounded) values of $d(j), j \geq k$ the solution of the system (3.45)–(3.46) coincides with that of (3.29)–(3.31b) at time instants $t_j^+, j \geq k$ which is not the case for (3.34). As we think of the inter-sample behavior d as a perturbation, our formal definition of PE will be stated as a property that is robust to bounded perturbations.

Definition 3.3. The protocol (3.45)–(3.46) is said to be (robustly) persistently exciting in T or PE_T if there exists $T \in (0, \infty)$ such that

$$\prod_{k=i}^{i+T-1} Q_k(\hat{e}(k))) = \mathbf{0}, \tag{3.47}$$

holds for every $k \in \mathbb{N}$ and any initial condition $e(i)$ and $\hat{e}(i)$ where we have written $\phi_{\hat{e}}(k)$ in place of $\phi_{\hat{e}}(k, i, e(i), \hat{e}(i), d_{[k,i]})$ and all $d \in \ell_\infty$, where $\phi_{\hat{e}}(i) := \phi_{\hat{e}}(i, e, \hat{e}, d_{[k,i]})$ is the $\hat{e}$ component of the solution of the system (3.45)–(3.46). That is, the T-fold product of the jump map evaluated along any set of trajectories that can be generated by (3.45)–(3.46) from any set of initial condition is the zero matrix. □

The protocols below are typical of what has been proposed in NCS literature and what is used in practice. In what follows, we will always assume an *ℓ-link* NCS with the ith linking consisting of l_i nodes and an error vector $\mathbf{e}_i$. Two PE_T protocols are presented next though we note that the simplest example of a PE protocol is RR (Example 3.3).

Example 3.4 (Hybrid RR-TOD scheduling protocol). The hybrid RR-TOD scheduling protocol enforces PE in a time-periodic manner. For a prescribed $M \in \mathbb{N}$, the protocol takes the form:

$$e^+ = (I - \Omega(i, \hat{e}))(e + d), \tag{3.48}$$

$$\hat{e}^+ = (I - \Omega(i, \hat{e}))\hat{e} + \Omega(i, \hat{e})(e + d), \tag{3.49}$$

$$\Omega(i, \hat{e}) := \begin{cases} \operatorname{diag}\{p_1(i)I_{s_1}, \ldots, p_N(i)I_{s_N}\}, & \text{if } \operatorname{mod}(i, M) = 0 \\ \operatorname{diag}\{\psi_1(\hat{e})I_{s_1}, \ldots, \psi_N(\hat{e})I_{s_N}\}, & \text{otherwise,} \end{cases}$$

where, $p_n(i) = 1$ when $\operatorname{mod}(i/M, N) = n - 1$ and $p_n(i) = 0$ otherwise with ψ_j defined in (3.38). The hybrid RR-TOD protocol is PE_T with $T = MN$. In particular, when $M = 1$, we obtain the simplest PE_T protocol: "classical" RR. □

Example 3.5 (Constant-penalty TOD). Constant-penalty TOD (CP-TOD) [18] uses the mechanism of "silent-time" to ensure that every link is eventually visited within a finite window of time: each link j has a counter r_j that is incremented at every transmission instant that it is *not* scheduled and reset to zero when it *is* scheduled. Irrespective of the underlying scheduling protocol, when a link's counter reaches a predetermined threshold, say M, it will be scheduled. This ensures that every link is scheduled within $\ell + M$ transmission instants[16].

The underlying scheduler in this example is TOD and corresponds to the constant-penalty TOD scheme in [18] with a penalty (vector) of $\varTheta$:

$$e^+ = (I - \varPhi(r,\zeta))(e + d), \tag{3.50}$$

$$\zeta^+ = (I - \varPhi(r,\zeta))(\zeta + \varTheta) + \varPhi(r,\zeta)(e + d), \tag{3.51}$$

$$r^+ = (I - \varPhi(r,\zeta))(r + \mathbf{1}), \tag{3.52}$$

where $\mathbf{1} = [1 \ \ldots \ 1]^T$, the scheduling function $\varPhi$ is given by

$$\varPhi(r,\zeta) = \mathrm{diag}\{\varphi_j(r,\zeta) I_{s_j}\}, \quad j \in [1, \ldots, \ell]$$

and

$$\varphi_n(r,\zeta) = \begin{cases} 1, & \text{if } [n = \min\{m\colon r_m \geq M\}] \vee \\ & \big[n = \min\big(\arg\max_{1 \leq j \leq N} |\boldsymbol{\zeta}_j|\big) \\ & \wedge (r_m < M, \forall\, m \in \{1, \ldots, N\})\big] \\ 0, & \text{otherwise} \end{cases} \tag{3.53}$$

$a \wedge b$ and $a \vee b$ denote the logical conjunction and logical disjunction of two conditions a and b, respectively. The role of estimating e is played by ζ and through the term $\varPhi(r,\zeta)e$, ζ is updated with $\mathbf{e}_j$ whenever the jth link is transmitted. For those links that are not transmitted, the estimated error is incremented by a fixed penalty $\varTheta$ that might capture the worst-case growth of error (in the absence of disturbance) for a given MATI. In addition to performing this *ad hoc* estimation, the scheduling protocol counts the number of transmission instants that a link has not been visited for, the link's silent time, and schedules links that have exceeded a predetermined threshold for silent-time. In this way, if ζ is degenerating into an arbitrarily bad estimate of e, all links will continue to be visited within a fixed-length, finite window of transmission instants through the mechanism of forcing a finite silent-time for each link. In a loose sense, the protocol's behavior will "often" be qualitatively similar to that of RR, a protocol that has been shown to lead to L_p stability of the NCS with appropriate conditions. □

3.3.5 a.s. Covering Protocols

By a random protocol, we mean a sequence of random transmission times together with i.i.d. random jump maps Q_i that are independent of e with

[16] The silent-time protocols described in [18] have the links measure *continuous time* as opposed to counting the number of transmission instants elapsed (discrete-time) and set the silent-time threshold in terms of an integer multiple of MATI, say $M\tau$. Since, for all $i \in \mathbb{N}$, $M\tau \geq M(t_{s_{i+1}} - t_{s_{i+1}})$, our silent-time threshold will be smaller for the same M but the protocol will behave in precisely the same manner as when using the verbatim definition of silent-time given in [18].

reference to (3.31a). That is, Q_i are i.i.d. random matrices taking values in the finite set $\mathcal{M}_{n_e} = \{M_0, M_1, \ldots, M_\ell\}$, where $M_0 = I_{n_e}$ and M_j is such that

$$\begin{aligned} M_j e &= M_j(\mathbf{e}_1, \ldots, \mathbf{e}_j, \ldots, \mathbf{e}_\ell) \\ &= (\mathbf{e}_1, \ldots, \mathbf{e}_{j-1}, \mathbf{0}, \mathbf{e}_{j+1}, \ldots, \mathbf{e}_\ell). \end{aligned}$$

We make this definition more precise shortly. The intuition behind this model is that at a transmission time t_i, either some link j will acquire the channel and have its component of e set to zero, that is,

$$\mathbf{e}_j(t_i^+) = 0, \quad \mathbf{e}_k(t_i^+) = \mathbf{e}_k(t_i), \quad k \neq j.$$

Hence $Q_i = M_j$ or else more than one node attempted to transmit resulting in a collision with e remaining unchanged ($Q_i = M_0$). Due to random "back-off" times, and wait times inserted into medium access protocols, transmission times are potentially random. Collectively, these issues are the same issues presented in multi-user access in computer and mobile voice networks though the network access patterns are somewhat different. See [14] for an overview.

Definition 3.4. For an ℓ-link NCS, abstractly, we define a random protocol as a discrete Markov chain Q_i subordinated by a renewal process[17] $N(t)$ such that

(1) $Q_i \in \mathcal{M}_{n_e}$ are i.i.d. random $n_e \times n_e$ with associated link and collision probabilities given by

$$\mathbf{P}\{Q_i = M_k\} = p_k.$$

(2) The sequence of arrival times $\{t_i\}_{i\in\mathbb{N}}$ is defined inductively by:

$$t_0 = \tau_0,$$

where $\tau_0 \sim \text{Exp}(\lambda)$ and for each $i > 0$,

$$t_i = t_{i-1} + \tau_i,$$

$\tau_i \sim \text{Exp}(\lambda)$, where the sequence $\{\tau_i\}$ is i.i.d. We set

$$N(t) = \begin{cases} 0, & t \in [0, t_0), \\ k, & t \in [t_{k-1}, t_k). \end{cases}$$

Hence, $N(t)$ is a Poisson process with intensity λ. □

Essentially, the τ_is denote the wait time after the arrival of a packet (before a new transmission begins). When not otherwise stated, we will henceforth assume that $\mathbf{P}\{Q_i = M_k\} = \mathbf{P}\{Q_i = M_j\} = (1 - p_0)/\ell$, $k, j \neq 0$, i.e., each

[17] More precisely, the process of interest is in fact a marked point process. See [10] for an exposition.

link is equally likely to transmit successfully. This assumption is not strictly necessary for our analyses, however, any other distribution of probabilities results in a *static* choice of priorities among links where one link may be favored over another during contention. There may be examples of NCS that would benefit from such an adjustment of relative link priorities offline in terms of required transmission rates or greater robustness of stability but as these choices are made offline and not in response to the evolution of the NCS state online, we believe that the scope of exploiting this degree of freedom is limited.

We pursue here a stochastic analogue of the PE_T property described in Section 3.3.4.

Definition 3.5 (Cover time). Consider a random protocol in the sense of Definition 3.4 for an ℓ-link protocol and define

$$T_0 = \min\{j \geq 1 \colon \{M_1, \ldots, M_\ell\} \subset \{Q_0, \ldots, Q_{j-1}\}\}$$

and, inductively for $i > 0$,

$$T_i = \min\left\{j \geq 0 \colon \{M_1, \ldots, M_\ell\} \subset \{Q_{T_{i-1}}, \ldots, Q_{T_{i-1}+j-1}\}\right\}.$$

We refer to T_i as the ith cover time, and collectively the cover time process. It is clear from our definition of Q_i that T_i is a stationary process. □

Definition 3.6 (Covering sequence). Let $\tau_i = t_{i+1} - t_i$, as in Definition 3.4, that is, τ_i are inter-arrival times. We say that

$$C(j,k) = \{(Q_j, \tau_j), \ldots, (Q_k, \tau_k)\}, \quad k \geq j,$$

is a covering sequence if and only if $\{M_1, \ldots, M_\ell\} \subset C_{(1)}(j,k)$.[18] It is easy to see that cover times are simply the lengths of consecutive disjoint covering sequences. □

Remark 3.3. From our definition of random protocols, the distribution of T_n is given by the solution to the (weighted) coupon collectors problem. When $p_i = p_j$, $i, j \neq 0$, we have the following closed form expression for the expectation:

$$\mathbf{E}[T] = \ell H_\ell/(1 - p_0), \tag{3.54}$$

where H_ℓ is the ℓth harmonic number and we have dropped the time index n since T_n is stationary. We also have the bound for the distribution, $\mathbf{P}\left\{T_n \geq \beta\ell \ln \ell/(1 - p_0)\right\} \leq \ell^{-(\beta-1)}/(1 - p_0)$, for any $\beta > 1$. Intuitively, $T_n = \mathbf{E}[T]$ "most of the time" and $\mathbf{P}\left\{T_n < \infty\right\} = 1$. □

Our abstract definition of a contention protocol is a model for the contention protocols discussed earlier and to that end we present two natural examples in this setting.

[18] The notation $C_{(1)}(j,k)$ refers to the covering sequence of matrices Q_i with no reference to inter-transmission times τ_i, i.e., $\{Q_j, \ldots, Q_k\}$.

Definition 3.7 (Almost surely finite cover time). We say that a protocol is *a.s. covering* or has an *a.s. finite cover time* if in Definition 3.5

$$(\forall i \in \mathbb{N}) \quad \mathbf{P}\left\{T_i < \infty\right\} = 1. \qquad \square$$

Note that from the preceding discussion, this property is verified by all contention protocols in the sense of Definition 3.4.

3.3.6 Slotted *p*-Persistent CSMA

What has been referred to as "scheduling" and the associated scheduling protocols by [13] is generally known as medium access in the communications literature. Carrier sense multiple access with collision detection (CSMA/CD) is by far the most widely used medium access protocol by virtue of the sheer volume of Ethernet and Ethernet-like networking devices shipped and manufactured each year.

CSMA/CD is a simple protocol: Links listen for transmissions on the channel. A link wanting to transmit acquires the channel when it senses that the channel is idle. When more than one link senses that the channel is idle and begins transmission, a collision occurs. At this point, all transmissions are immediately aborted. There are several variants of CSMA/CD that prescribe how transmissions are rescheduled and how links initially acquire the channel.

With slotted p-persistent CSMA, rather than have links transmit whenever the channel is idle, links are only permitted to transmit at prescribed transmission slots that occur every $t_s > 0$ seconds in slotted protocols. At the start of slot s_k, links $S = \{i, \ldots, j\}$ intending to transmit acquire the channel with a probability of p. If a collision occurs, links S^c are permitted to transmit in the next slot and links S reschedule their transmissions at slots $\{s_{k+d_i}, \ldots, s_{k+d_j}\}$.

As alluded to earlier, the primary reason that CSMA protocols and indeed, all contention protocols work in practice is that the access patterns of computer and voice networks are "bursty" in nature. The assumption is that a link will occasionally transmit a burst of information and remain otherwise idle. Transmissions are expected to eventually succeed as links are "infrequently" contending for the channel.

The situation is quite different for control networks with the implication that medium access patterns are constant rather than bursty and for slotted p-persistent CSMA, we assume that every slot will be in contention. Another key difference between computer networks and NCS is in the treatment of collisions and dropouts. NCS should not buffer failed transmissions of controller or sensor values but rather, attempt to transmit the latest values when a slot is free. As the maximum number of links contending slots is constant for every slot, there is no reason for a link to delay transmission for more than one slot after a collision.

With these assumptions, consider an ℓ-link NCS with the p-persistent CSMA protocol. The probability $\mathbf{P}\{Q_i = M_j\}$ that a *particular* link j transmits successfully during the ith slot is given by

$$\mathbf{P}\{Q_i = M_j\} = p(1-p)^{\ell-1}.$$

It is clear that $\mathbf{P}\{Q_i = M_j\}$ is maximized when $p = 1/\ell$. We will henceforth set $p = 1/\ell$ and have that

$$\mathbf{P}\{Q_i = M_j\} = \frac{1}{\ell}\left(1 - \frac{1}{\ell}\right)^{\ell-1} = \frac{(\ell-1)^{\ell-1}}{\ell^\ell}.$$

Notice that in this "optimal" case,

$$\mathbf{P}\{Q_i = M_j\} = \mathbf{P}\{Q_i = M_k\} = (\ell-1)^{\ell-1}/\ell^\ell \text{ for } j,k \neq 0$$

and the probability of a collision is given by $\mathbf{P}\{Q_i = M_0\} = 1-(\ell-1)^{\ell-1}/\ell^{\ell-1}$. Finally, we assume that slots occur every $t_s > 0$ seconds and hence, p-persistent CSMA is a contention protocol in the sense of Definition 3.4 where inter-arrival times τ_i are deterministic.

3.3.7 CSMA with Random Waits

Whereas the use of fixed slots tends to improve throughput and reduce collisions with computer networks, e.g., slotted versus pure ALOHA, the contention by every link at every slot forces transmissions to happen in lock-step with NCS network access patterns with the potential for a collision at every slot.

Suppose that instead of immediately acquiring the channel with probability p after sensing the channel to be idle or after a new slot arrives, links instead wait for a random amount of time before transmitting. In particular, if a particular link j waits for a random time $\eta_j' \sim \text{Exp}(\lambda/\ell)$, then, $\mathbf{P}\{Q_i = M_j\} = (1-p_0)/\ell, j \neq 0$. The actual wait time before any particular transmission will be given by

$$\tau = \min\{\eta_1', \ldots, \eta_\ell'\};$$

that is, the link that waits the least gets to transmit first. Hence, $\tau \sim \text{Exp}(\lambda)$. Assuming the wait times are i.i.d. for each link, this is the prototypical example of what we mean by a stochastic protocol and a stochastic channel.

In the presence of transmission errors, p_0 is generally nonzero and conceptually, p-persistent CSMA and CSMA with random waits are essentially the same apart from the fact that the transmission process is truly random with the latter. While CSMA with random waits can be thought of as a protocol in its own right when the random waits are enforced explicitly in the implementation, it can also be thought of as a model of medium access with

NCS access patterns while using a class of CSMA wireless protocols. Delays in signal detection, multi-path effects and varying processor loads mean that links are only prepared to transmit after some delay upon sensing the channel being idle and although the cumulative effects of these delays may not be exponentially distributed, the principle remains the same.

3.4 NCS Stability

The notion of *robustness* of various stability properties plays a fundamental role in practical design and implementation of control systems as evidenced by the extensive literature discussing, for example, input-to-state stability (ISS), H_2, H_∞ design and variants of robust stability. To that end, [8] and [9] have examined L_p and input-to-state stability of NCS, respectively and it was shown in [13] that persistently exciting scheduling protocols lead to L_p stable NCS when appropriate conditions are imposed on transmission rates and the nominal system and similar results were provided for UGES and UGAS protocols in [8] and [9], respectively. While the proof techniques and settings are substantially different, the novel use of various small-gain theorems is a unifying theme throughout these results and a powerful tool for quantifying robustness. See [7, Chapter 5.4] for an introduction to the notion of input/output stability gain and [6] for general ISS small-gain results.

We outline several NCS stability results in the ensuing sections and refer the reader to [8], [9] and [13] where the results are stated and proved in greater generality. Finally, while these results are ISS or input–output stability (IOS) type results, whenever exogenous perturbations are removed, UGES and UGAS can be recovered under additional mild technical assumptions. See [8, Section II-B], for instance.

We first recall the definition of L_p stability and detectability for a system Σ_z with jumps:

$$\Sigma_z \colon \dot{z} = f(t, z, w), \quad t \in [t_i, t_{i+1}], \tag{3.55}$$

output $y(t) = g(t, z)$ and with jump equation

$$z(t_i^+) = h(i, z(t_i)). \tag{3.56}$$

Let $f \colon \mathbb{R} \to \mathbb{R}^n$ be a (Lebesgue) measurable function and define $\|f\|_p := \left(\int_{\mathbb{R}} |f(s)|^p ds\right)^{1/p}$ for $1 \le p < \infty$ and define $\|f\|_\infty := \text{ess.}\sup_{t\in\mathbb{R}} |f(t)|$. We say that $f \in L_p$ for $p \in [1, \infty]$ whenever $\|f\|_p < \infty$. Let $f \colon \mathbb{R} \to \mathbb{R}^n$ and let $[a, b] \subset \mathbb{R}$. We use the notation

$$\|f[a, b]\|_p := \left(\int_{[a,b]} |f(s)|^p ds\right)^{1/p}$$

to denote the L_p norm of f when restricted to the interval $[a, b]$.

Definition 3.8. Let $p \in [1,\infty]$ and $\gamma \geq 0$ be given. We say that Σ_z is L_p stable from w to y with gain γ if $\exists\, K \geq 0$: $\|y[t_0,t]\|_p \leq K|z_0| + \gamma\|w[t_0,t]\|_p$.□

Definition 3.9. Let $p,q \in [1,\infty]$ and $\gamma \geq 0$ be given. The state z of Σ_z is said to be L_p to L_q detectable from output y with gain γ if $\exists\, K \geq 0$: $\|z[t_0,t]\|_q \leq K|z_0| + \gamma\|y[t_0,t]\|_p + \gamma\|w[t_0,t]\|_p$. □

An exposition of these ideas as they pertain to NCS can be found in [8, Section II-B].

3.4.1 L_p Stability of NCS with Lyapunov UGES Protocols

A more general version of the following result was first presented in [8] and asserts that Lyapunov UGES scheduling protocols preserve L_p stability of the network-free system under appropriate conditions and for small enough values of MATI.

Theorem 3.1. Consider NCS (3.29)–(3.31b) and suppose that:

(i) the NCS scheduling protocol (3.31a) is Lyapunov UGES with Lyapunov function W that is locally Lipschitz in e and uniformly in i, and there exists $L \geq 0$ such that:

$$\left\langle \frac{\partial W(i,e)}{\partial e}, g(t,x,e,w) \right\rangle \leq LW(i,e) + |\tilde{y}|, \tag{3.57}$$

for almost all $e \in \mathbb{R}^{n_e}$, for all $(x,w) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_w}$, for all $t \in (t_i, t_{i+1})$, for all $i \in \mathbb{N}$, where $\tilde{y}\colon \mathbb{R}^{n_e} \times \mathbb{R}^{n_w} \to \mathbb{R}$ is a continuous function of (x,w);

(ii) System (3.29) is L_p stable from (W,w) to $\tilde{y}$ with gain γ for some $p \in [1,\infty]$; (x,w) is L_p to L_p detectable from $\tilde{y}$; (e,w) is L_p to L_p detectable from W; and

(iii) MATI satisfies $\tau \in (\epsilon, \tau^*)$, $\epsilon \in (0,\tau^*)$, where

$$\tau^* = \frac{1}{L}\ln\left(\frac{L+\gamma}{\theta L + \gamma}\right), \tag{3.58}$$

and θ comes from (3.41).

Then, the NCS is L_p-stable from w to (x,e) with linear gain. □

Remark 3.4. Within the framework of hybrid systems presented in [4], results analogous to Theorem 3.1 are developed in [1] where τ^* is given by

$$\tau^* = \begin{cases} \dfrac{1}{Lr}\arctan\left(\dfrac{r(1-\theta)}{2\dfrac{\theta}{1+\theta}\left(\dfrac{\gamma}{L}-1\right)+1+\theta}\right), & \gamma > L, \\ \dfrac{1-\theta}{L(1+\theta)}, \quad \gamma = L, \\ \dfrac{1}{Lr}\operatorname{arctanh}\left(\dfrac{r(1-\theta)}{2\dfrac{\theta}{1+\theta}\left(\dfrac{\gamma}{L}-1\right)+1+\theta}\right), & \gamma < L, \end{cases} \tag{3.59}$$

where

$$r = \sqrt{\left|\left(\frac{\gamma}{L}\right)^2 - 1\right|}\,. \tag{3.60}$$

This bound is shown to improve upon (3.58) in [1] when verifying UGES. The results therein are stated for UGAS, UGES and semi-global practical ISS and can, in principle, be extended to apply to L_p IOS. □

3.4.2 L_p Stability of NCS with PE_T Protocols

The following theorem asserts that PE protocols lead to L_p stability of the NCS for sufficiently small MATI. While we do not provide a closed-form expression for MATI bounds, the bounds are readily obtained in examples by numerically solving for τ^* in (3.59). Note that we only consider stability of e and x. The decision-vector, if used in the protocol being analyzed, may fail to verify any stability properties but as $\hat{e}$ has no physical significance as a state vector whose evolution is governed by the protocol, this is generally not an issue. Let $\mathcal{A}_n$ denote the set of all $n \times n$ matrices and let $\mathcal{A}_n^+$ denote the subset of all matrices that are positive semi-definite, symmetric and have positive entries and let $\mathbb{R}_+^n$ denote the nonnegative orthant.

Theorem 3.2. Consider NCS (3.29)–(3.31b) and suppose that:

(i) the NCS scheduling protocol (3.31b) is uniformly persistently exciting in time T and there exists $A \in \mathcal{A}_{n_e}^+$ and a continuous $\tilde{y}\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}_+^{n_e}$ so that the error dynamics (3.30) satisfy[19]

$$\overline{g}(t, x, e, w) \preceq A\overline{e} + \tilde{y}(x, w) \tag{3.61}$$

for all $(x, e, w) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_e} \times \mathbb{R}^{n_w}$, for all $t \in (t_i, t_{i+1})$, for all $i \in \mathbb{N}$, where $\tilde{y} = G(x) + H(w)$;

(ii) System (3.29) is L_p stable from (e, w) to $G(x)$ with gain γ for some $p \in [1, \infty]$; (x, w) is L_p to L_p detectable from $\tilde{y}$;

(iii) and MATI satisfies $\tau \in (\epsilon, \tau^*)$, $\epsilon \in (0, \tau^*)$, where $\tau^* = \dfrac{\ln(z)}{|A|T}$ and z solves

$$z(|A| + \gamma T) - \gamma T z^{1-1/T} - 2|A| = 0, \tag{3.62}$$

where A comes from (3.61).

Then, the NCS is L_p-stable from w to (x, e) with linear gain. □

[19] Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$. The *vector* partial order $\preceq$ is given by $x \preceq y \iff (x_1 \leq y_1) \wedge \cdots \wedge (x_n \leq y_n)$ and $\overline{e}$ and $\overline{g}$ are given by $\overline{e} := (|e_1|, \ldots, |e_{n_e}|)^T$ and $t \overset{\overline{g}}{\mapsto} \overline{g(t)}$, respectively. That is, $\overline{e}$ is the vector that results from taking the absolute value of each scalar component of e and $\overline{g}$ operates analogously on the image of g.

Remark 3.5. Suppose that $g(t,x,e,w) = \overline{Bx + Ce + Dw}$ and let $A = [a_{ij}]$, where $a_{ij} = \max\{|c_{ij}|, |c_{ji}|\}$ and $\tilde{y}(x,w) = \overline{Bx + Dw}$. We immediately have that A and $\tilde{y}(x,w)$ satisfy Condition 2 of Theorem 3.2 and $\|\tilde{y}(x,w)\|_p = \|Bx + Dw\|_p \leq \|Bx\|_p + \|Dw\|_p$. Whenever g satisfies a linear growth bound of the form $|g(t,x,e,w)| \leq L(|x| + |e| + |w|)$, it is straightforward to construct an appropriate A and $\tilde{y}$. □

Remark 3.6. Suppose that the network-free system is L_p stable from w to x with gain γ and the NCS satisfies the hypotheses of Theorem 3.2. Then for any $\gamma^* > \gamma$, it is possible to show that there exists a MATI τ such that the NCS is L_p stable from w to x with gain γ^*. This corollary of Theorem 3.2 is particularly useful in the design of optimal/robust controllers. □

3.4.3 L_p Stability of NCS with Random Protocols

The following result analyzes the input–output L_p stability (IOS) of NCS (in expectation), the essence of which is that outputs (or state) of an NCS verify a robustness property with respect to exogenous disturbances. We stress that it is only the network protocol and channel that induces randomness in our models and that the exogenous disturbances are L_p signals as in [8] and [13].

Although link cover times and inter-transmission are now random, and hence, not uniform, if the network-free system is L_p stable, the NCS remains so with any contention protocol, in the sense of our definition, whenever attempted transmissions occur "fast enough." By "fast enough" we mean that there exists a choice of intensity λ of the transmission process parameterized by properties of the protocol and the NCS dynamics such that the NCS is L_p stable-in-expectation from disturbance to NCS state with a finite expected gain.

Intuitively, and despite the presence of collisions, random packet dropouts and random inter-arrival times, it seems natural to expect that the stability of the NCS (3.24)–(3.28a) for high enough "average" transmission rates and in light of the a.s. cover times of contention protocols and in analogy with persistently exciting scheduling protocols, this stability ought to be robust in an L_p sense. In fact, if we relax our notion of "L_p stability" to "L_p stability-in-expectation," we can prove a positive result in that direction. The definition of these properties is obtained, essentially, by using expected norms $\mathbf{E}\|\cdot\|$ in lieu of $\|\cdot\|$ in Definitions 3.8 and 3.9. We stress that, as developed in this chapter, these notions only apply to hybrid systems of the form (3.55) and (3.56), i.e., we insist that w is "essentially" an L_p signal and not a Lèvy process (cf. [5]) specifically because we are concerned with robustness of stability in the sense of, e.g., [3], whereas a Lèvy process characterization of disturbances may be more appropriate in modeling sensor noise and quantization phenomena.

While the following results are stated for the delay and inter-arrival processes presented in Definition 3.4, it is straightforward to extend them to a more general class of processes.

Theorem 3.3. Consider an ℓ-link NCS (3.29)–(3.31b) and suppose that:

(i) the NCS employs a contention scheduling protocol with i.i.d. cover times T_i and the inter-arrival process is Poisson with intensity λ and also suppose that the NCS error dynamics satisfy

$$\overline{g}(t, x, e, w) \preceq A\overline{e} + \tilde{y}(x, w) \tag{3.63}$$

for all $(x, e, w) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_e} \times \mathbb{R}^{n_w}$ and almost all t, where A is a nonnegative symmetric $n_e \times n_e$ matrix with nonnegative entries and $\tilde{y} = G(x) + H(w)$;

(ii) System (3.29) is L_p stable-in-expectation from (e, w) to $G(x)$ with expected gain γ for some $p \in [1, \infty]$; (3.30) is L_p to L_p detectable-in-expectation from $\tilde{y}$.

Then, there exists $\lambda < \infty$ depending on $(\ell, |A|, \gamma, \mathbf{E}[T], p_0)$ such that the NCS is L_p stable-in-expectation from w to (x, e) with a finite linear expected gain $1/(1 - \gamma\gamma^*)$. Specifically, λ solves $\gamma^*\gamma < 1$ with

$$\gamma^* = \frac{\mathbf{E}[T](1 + \rho)}{(\lambda - |A|)(1 - \rho)},$$

where,

$$\rho = (\alpha(1 - p_0))^\ell \prod_{k=1}^{\ell} \frac{\ell - (k - 1)}{\ell(1 - p_0\alpha) - (k - 1)(1 - p_0)\alpha} - 1,$$

and $\alpha = \dfrac{\lambda}{\lambda - |A|}$ and $\lambda > \dfrac{|A|}{1 - p_0}$. □

Remark 3.7. While no bounds for λ are given, the requisite intensity can be found numerically. □

3.4.4 L_p Stability of NCS with a.s. Lyapunov Protocols

The following result is a natural extension to Theorem 3.1 for channels that have a non-zero probability of packet dropout and is intended to be used in much the same way as the latter result. While [8] presents sufficient conditions for L_p stability in the presence of deterministically characterized packet dropouts for Lyapunov UGES protocols, we believe the following result is a more natural treatment of dropouts and the conditions are directly verifiable.

Theorem 3.4. Consider NCS (3.29)–(3.31b) and suppose that:

(i) the NCS scheduling protocol (3.31a) is a.e. Lyapunov U GES with Lyapunov function W that is locally Lipschitz in e and uniformly in i, and there exists $L \geq 0$ such that:

$$\left\langle \frac{\partial W(i,e)}{\partial e}, g(t,x,e,w) \right\rangle \leq LW(i,e) + |\tilde{y}|, \tag{3.64}$$

for almost all $e \in \mathbb{R}^{n_e}$, for all $(x,w) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_w}$, all $t \in (t_i, t_{i+1})$, for all $i \in \mathbb{N}$, where $\tilde{y}\colon \mathbb{R}^{n_e} \times \mathbb{R}^{n_w} \to \mathbb{R}$ is a continuous function of (x,w);

(ii) System (3.29) is L_p stable from (W,w) to $\tilde{y}$ with finite expected gain γ for some $p \in [1,\infty]$; (x,w) is L_p to L_p detectable from $\tilde{y}$ with finite expected gain; e is L_p detectable from W with finite expected gain;

(iii) the channel packet dropout probability is given by $p_0 \geq 0$ and (3.36) is satisfied with an i.i.d. sequence $\{\kappa_i\}$ such that the intensity of the inter-transmission process λ satisfies

$$\lambda > \frac{\gamma + L}{1 - \mathbf{E}[\kappa]}\,. \tag{3.65}$$

Then, the NCS is L_p-stable from w to (x,e) with finite expected linear gain. □

Remark 3.8. As the motivation for studying a.s. Lyapunov UGES comes from the use of Lyapunov UGES protocols on non-ideal channels, we can restate several of the conditions of Theorem 3.4 in light of Proposition 3.1. Let θ be as in (3.41) and let the probability of packet dropout p_0 satisfy (3.42). The requisite intensity in (3.65) becomes

$$\lambda > \frac{\gamma + L}{(1 - p_0)(1 - \theta)}\,. \tag{3.66}$$

□

Remark 3.9. As in [8] and [13], in both this and the preceding section, several generalizations and specializations of the stability results are possible. With additional technical assumptions on the NCS dynamics, one can conclude uniform global exponential stability (in expectation) and the assumptions on the various reset maps can be relaxed so as to infer ISS-like properties in lieu of L_p stability as discussed in [9]. If we forgo the detectability assumptions in the hypotheses of Theorems 3.3 and 3.4 we can only infer input-to-output stability-in-expectation. □

3.5 Case Studies and Comparisons

The aim of this section is to examine the various results presented in this chapter and compare them to results presented in the literature. For simplicity we will focus on the following linear time-invariant systems where the simplified equations for an ℓ-link NCS are given by (3.6) together with jump equations (3.31a) or (3.31b).

Example 3.6 (Batch Reactor). The linearized model of an unstable batch reactor is a two-input-two-output NCS that can be written as:

$$\dot{x}_P = A_P x_P + B_P u, \quad y = C_P x_P,$$

where $C_P = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$,

$$A_P = \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix}, \quad B_P = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}.$$

The system is controlled by a PI controller with a state-space realization prescribed by

$$\dot{x}_C = A_C x_C + B_C y, \quad u = C_C x_C + D_C y,$$

and

$$A_C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad B_C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C_C = -\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}, \quad D_C = -\begin{bmatrix} 0 & 2 \\ -5 & 0 \end{bmatrix}.$$

Assuming that only the outputs are transmitted via the network, we have a two-link NCS ($\ell = 2, l_1 = l_2 = 1$) with error and state equations prescribed by (3.6) where

$$A_{11} = \begin{bmatrix} A_P + B_P D_C C_P & B_P C_C \\ B_C C_P & A_C \end{bmatrix}, \quad A_{12} = \begin{bmatrix} B_P D_C \\ B_C \end{bmatrix},$$
$$A_{21} = -\begin{bmatrix} C_P & 0 \end{bmatrix} A_{11}, \quad A_{22} = -\begin{bmatrix} C_P & 0 \end{bmatrix} A_{12}.$$

The error equation is given by

$$\dot{e} = A_{22} e + A_{21} x. \tag{3.67}$$

□

This example is used as the benchmark in comparing the inter-transmission bounds with the stability analysis frameworks outlined in this chapter and in [16, 18, 19].

3.5.1 Comparison of Analytical Inter-transmission Bounds

Prior to making numerical comparisons with respect to the bounds obtained for Example 3.6, we provide a brief summary of the analytical bounds in Table 3.1 as they apply in general. The various constants used are defined and explained in the respective referenced sections and details can be found in the respective sources cited in the table. These are analytical bounds that

Table 3.1. Summary of inter-transmission bounds for various classes of protocols

MATI – Section 3.2.2, [16, 18, 19] (Worst Case Analysis)	
RR & TOD Protocol	$\tau^{WC}_{3.2.2} = \frac{c_3}{M\ell(\ell+1)k_h k_f c_4}$
Silent-time Protocols silent-time T	$\tau^{ST}_{3.2.2} = \min\left\{\frac{\ln(2)}{k_h T}, \frac{S}{8}, \frac{S}{16c_1\sqrt{c_1/c_2}k_h}\right\}$, where $S = [k_h\sqrt{c_1/c_2}\sum_{i=1}^{\ell}(i+T-\ell)]^{-1}$
MATI – Section 3.4.1, [8] (Lyapunov UGES Analysis)	
RR Protocol	$\tau^{RR}_{3.4.1} = \frac{1}{k_h\sqrt{\ell}}\ln\left(\frac{\sqrt{\ell}(k_h+\gamma)}{k_h\sqrt{\ell-1}+\gamma\sqrt{\ell}}\right)$
TOD Protocol	$\tau^{TOD}_{3.4.1} = \frac{1}{L}\ln\left(\frac{L\sqrt{\ell}+\gamma\sqrt{\ell}}{L\sqrt{\ell-1}+\gamma\sqrt{\ell}}\right)$
MATI – Section 3.4.2, [13] (PE_T Analysis)	
PE_T Protocols (including RR)	$\tau^{PE}_{3.4.2} = \frac{\ln(z)}{\vert A\vert T}$, where z solves $z(\vert A\vert+\gamma T) - \gamma T z^{1-1/T} - 2\vert A\vert = 0$
Reciprocal-Intensity – Section 3.4.3, [11] (a.s. Cover Time Analysis)	
Stochastic Protocols $\mathbf{P}\{\text{dropout}\} = p_0$	$\tau^{STO}_{3.4.3} < \frac{(1-p_0)}{\vert A\vert}$, solved numerically via Theorem 3.3
Reciprocal-Intensity – Section 3.4.4, [11] (a.s. Lyapunov UGES Analysis)	
RR Protocol	$\tau^{RR}_{3.4.4} = \frac{(1-p_0)(\sqrt{\ell}-\sqrt{\ell-1})}{\ell(\gamma+k_h)}$
TOD Protocol	$\tau^{TOD}_{3.4.4} = \frac{(1-p_0)(\sqrt{\ell}-\sqrt{\ell-1})}{\sqrt{\ell}(\gamma+L)}$

guarantee stability. For all bounds presented, stability is in the sense of L_p (in-expectation) except for those derived in [16, 18, 19], where UGES is the applicable notion of stability.

Table 3.2 compares a selection of these MATI bounds as they apply to TOD and RR. It is shown in [13, Section VI-C] that for LTI systems employing RR scheduling, MATI bounds obtained within the framework outlined in Section 3.4.2 are asymptotically larger by a factor of $O(\ell^{1/2})$ than the MATI obtained in [8] which are, in turn, shown to be analytically superior to the bounds in [15] for both TOD and RR. As indicated in Remark 3.4, for protocols that are Lyapunov UGES or UGAS, [1] may offer improved MATI bounds over [8] and, for the batch reactor example, these were demonstrated to be an improvement of approximately 10%.

Table 3.2. Summary of analytic comparisons for NCS without dropouts for an ℓ-link NCS with constants as in (3.13)–(3.15)

Linear Systems	
RR Protocol	$\frac{\tau^{RR}_{3.4.1}}{\tau^{WC}_{3.2.2}} \geq 2\frac{\ell+1}{\sqrt{\ell}}\sqrt{\frac{c_2}{c_1}}\left(\sqrt{\frac{c_2}{c_1}}+1\right)$ $\frac{\tau^{PE}_{3.4.2}}{\tau^{RR}_{3.4.1}} \geq O(\ell^{1/2})$ as $\ell \to \infty$
TOD Protocol	$\frac{\tau^{TOD}_{3.4.1}}{\tau^{WC}_{3.2.2}} \geq 2(\ell+1)\sqrt{\frac{c_2}{c_1}}\left(\sqrt{\frac{c_2}{c_1}}+1\right)$
Nonlinear Systems	
RR Protocol	$\frac{\tau^{RR}_{3.4.1}}{\tau^{WC}_{3.2.2}} \geq 8\frac{\ell+1}{\sqrt{\ell}}\left(\frac{c_2}{c_1}\right)^{\frac{3}{2}}\left(\sqrt{\frac{c_2}{c_1}}+1\right)$
TOD Protocol	$\frac{\tau^{TOD}_{3.4.1}}{\tau^{WC}_{3.2.2}} \geq 8(\ell+1)\left(\frac{c_2}{c_1}\right)^{\frac{3}{2}}\left(\sqrt{\frac{c_2}{c_1}}+1\right)$

3.5.2 Comparison of Numerical Inter-transmission Bounds ($p_0 = 0$)

For simplicity, and since L_p stability results are not provided in [18], we will largely restrict our discussion without exogenous disturbances and examine bounds that verify UGES and related properties. Much of the focus will be on RR scheduling as it is the only scheduling protocol that can be mutually treated by the analysis frameworks in this chapter, [8] and [18] but several other protocols will be examined as well.

We present and compare various results for the batch reactor example, Example 3.6, following [8], [13], [15]. The comparison results are summarized below:

(a) The MATI bounds are shown in Table 3.3 with the bounds computed via the PE framework larger than those obtained using the results of [18] by a factor of 10^7 and larger than the bound obtained by the results of [8] by a factor of 1.5. The bounds $\tau^{PE}_{3.4.2}$ and $\tau^{WC}_{3.2.2}$ apply to any PE_T protocol for the original two-link system. The bound $\tau^{RR}_{3.4.1}$ only applies to RR ($T = \ell = 2$).

(b) When using RR, $\tau^{PE}_{3.4.2}$ that achieves UGES is equivalent to a network throughput of 84 kbps (assuming 128 byte frames), achievable on current 802.11g and 802.11b wireless networks and $\tau^{RR}_{3.4.4}$ requires an effective network throughput of approximately 125 kbps.

(c) We formally fix the constants used to compute the respective bounds and plot $\tau^{PE}_{3.4.2}$ and $\tau^{RR}_{3.4.4}$ with $T = \ell \in [1, 1000]$ in Fig. 3.6 to examine the behavior of the bounds as the number of links grow. We also fix $\ell = 2$ and allow $T \geq 2$ to vary for $\tau^{ST}_{3.2.2}$ and $\tau^{PE}_{3.4.2}$. Despite the relatively modest improvements for the nominal two-link system using RR, the differences are significant on the $\log_{10}(T) \times \log_{10}(\tau^*)$ scale used in Fig. 3.6 when we formally increase T or, equivalently, the number of links.

Simulations and alternative techniques for calculating MATI are a key test of the practicality of the MATI bounds and stability results produced in this chapter and in the literature. For linear systems with equidistant transmission times employing RR scheduling, an actual analytic MATI bound can be computed as discussed in [8, Section VII-A]. For general protocols, however,

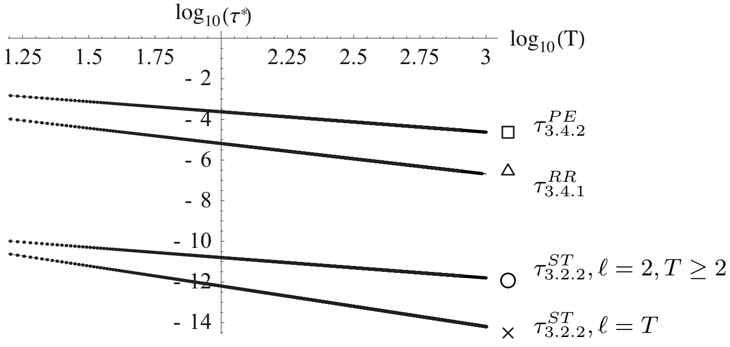

Fig. 3.6. Batch reactor MATI bounds comparison for PE_T protocols, $T \in [1, 1000]$

Table 3.3. MATI bounds achieving UGES for Example 3.6 with PE_T and Lyapunov UGES protocols

	$T = 2$	$T = 6$	$T = 50$
$\tau_{3.4.2}^{PE}$	0.0123	0.004	4.75×10^{-4}
$\tau_{3.4.4}^{RR}$	0.0082	N/A	N/A
$\tau_{3.2.2}^{WC}$	1.05×10^{-9}	2.86×10^{-10}	3.18×10^{-11}
$\tau_{3.4.4}^{TOD}$	0.01	N/A	N/A
$\tau_{3.4.2}^{PE}/\tau_{3.2.2}^{WC}$	1.18×10^{7}	1.40×10^{7}	1.49×10^{7}
$\tau_{3.4.2}^{PE}/\tau_{3.4.4}^{RR}$	1.50	N/A	N/A
$\tau_{3.4.2}^{PE}/\tau_{3.4.4}^{TOD}$	1.23	N/A	N/A

simulations are the only resort, and as such, no firm conclusions can be drawn vis-a-vis the theoretical bounds for arbitrary NCS.

3.5.3 Comparison of Numerical Inter-transmission Bounds ($p_0 > 0$)

Finally, we examine Example 3.6 for channels where $p_0 > 0$. In particular, we look at the CSMA protocol described in Section 3.3.7 and hence,

$$\mathbf{E}[T] = 2 \cdot H_2/(1 - p_0) = 3/(1 - p_0). \tag{3.68}$$

By Theorem 3.3, the batch reactor system will be L_p stable-in-expectation from w to x if

$$\frac{\mathbf{E}[T](1+\rho)}{(\lambda - |A|)(1-\rho)}\gamma < 1, \tag{3.69}$$

where γ is the L_p gain of x subsystem from the input e to an "auxiliary" output $\tilde{y} = \overline{A_{21}}x$.

By solving for λ numerically in (3.69), subject to the constraint $\lambda > |A|/(1 - p_0)$, we are able to establish expected transmission rate bounds as a function of p_0 that ensure L_p stability of the batch reactor system. The batch reactor system with the CSMA protocol was also simulated using expected transmission rates of $[1, \infty)$ transmissions per second for $p_0 \in [0.1, 0.8]$. A bisection heuristic was used to find the intensities that resulted in stability with the ensemble average of multiple simulations with fixed initial conditions to yield the simulation-derived intensity bound.

The expected transmission rate bounds and expected inter-transmission times are shown in Table 3.4 as a function of dropout/collision probability p_0 and plotted in Fig. 3.7. Simulation-derived bounds are also listed in Table 3.4.

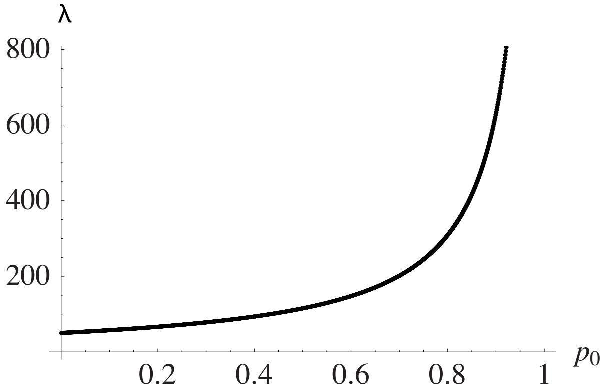

Fig. 3.7. Batch reactor expected transmission rate bounds for stochastic protocols as a function of dropout/collision probability p_0 with identical initial conditions

For the initial condition used, the bounds obtained via Theorem 3.3 are within a factor of 4 of simulation-based bounds, and for example, demonstrate that with a 50% probability of dropout/collision, the network must deliver approximately 922 kbps (116 × 8 bits) of network throughput to maintain L_p stability. This is well within the realm of ordinary Ethernet and 802.11 wireless technology.

We can also consider the example within the context of a.s. Lyapunov UGES. Suppose that the TOD scheduling is employed. From Table 3.1, the requisite intensity for the conditions of Theorem 3.4 to be verified is

Table 3.4. Transmission rate and inter-transmission time bounds λ and $\tau_{3.4.3}^{STO} = 1/\lambda$ are derived via Theorem 3.3; λ^* and $\tau_{3.4.3}^{STO*} = 1/\lambda^*$ are derived via simulation

p_0	λ	$\tau_{3.4.3}^{STO} = 1/\lambda$ (s)	λ^*	$\tau_{3.4.3}^{STO*} = 1/\lambda^*$ (s)
0	50.19	0.02	14.77	0.0677
0.1	57.46	0.017	16.05	0.0623
0.2	66.52	0.015	18.38	0.0544
0.3	78.15	0.013	21.37	0.0468
0.4	93.63	0.011	25.00	0.0400
0.5	115.27	0.0087	31.65	0.0316
0.6	147.71	0.0068	37.74	0.0265
0.7	201.74	0.0049	61.35	0.0163
0.8	309.74	0.0032	145.77	0.00686

$$\lambda > \frac{108.07}{1 - p_0}.$$

For an ideal channel ($p_0 = 0$), this corresponds to a transmission at least once every 9.25 ms compared to a MATI of 0.01 s for the deterministic results presented in [8] – a factor of 1.08 improvement in favor of the deterministic results. The notion of MATI implies that every inter-transmission time is *uniformly bounded* whereas the intensity (or reciprocal) is an “average MATI” – individual inter-transmission times can individually exceed or fall short of the average. Notably, both values fall short of the figure obtained for the CSMA protocol of 0.02 s. As the characterization of dropouts in [8] is markedly different from that presented here, we do not pursue a comparison for $p_0 > 0$. We can, however, compare CSMA and TOD in the presence of dropouts as presented in this section and we see that the trend is continued for $p_0 > 0$, e.g., the requisite intensity for $p_0 = 0.5$ is over 216 for TOD and less than 116 for CSMA.

3.6 Conclusions

This chapter presented several general frameworks for emulation-based design of a general nonlinear control systems with disturbances that rely upon properties of the network-free system and various properties of the scheduling protocol used. Our guiding philosophy in the approach is the following qualitative statement that intuition suggests: *for high enough transmission rates, a scheduling protocol that is guaranteed to reduce the network-induced error within a finite amount of time ought to preserve stability properties of the network-free system.* In particular, this is the case for (a.s.) Lyapunov UGES and UGAS protocols as well as PE_T and a.s. covering protocols.

Quantitatively, the results outlined provide the sharpest bounds for MATI and expected transmission rate currently known in the literature for the classes of systems and protocols analyzed, and in some cases, are the only known results for certain classes of systems and scheduling protocols.

We qualify this observation by noting that the various protocol properties, namely, PE_T, Lyapunov UGES and UGES and their stochastic analogs are not necessarily the finest characterization possible of any particular protocol. This is reflected in the disparity between theoretical MATI and transmission rate values and those obtained by simulations. For example, it is known that for LTI systems employing RR with equidistant inter-transmission times, analytic MATI bounds that achieve UGES can be computed and, indeed, are as sharp or sharper than those obtained by any result in this chapter. The aim of this chapter, however, was to present emulation-type results and design procedures for the largest class of systems for which results are currently known and which are useful in practice. To that end, we believe that this work serves as a useful starting point from which there is still much scope for improvement.

References

1. Carnevale D, Teel AR, Nešić D (2007) A Lyapunov proof of an improved maximum allowable transfer interval for networked control systems. IEEE Trans Automat Control 52(5):892–897
2. Coyle E, Liu B (1985) A matrix representation of CSMA/CD networks. IEEE Trans Commun 33(1):53–64
3. Dullerud GE, Paganini F (2000) A course in robust control theory. Springer, New York
4. Goebel R, Teel AR (2006) Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. Automatica 42(4):573–587
5. Hespanha JP, Teel AR (2006) Stochastic impulsive systems driven by renewal processes. In: Proc 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan
6. Jiang ZP, Teel AR, Praly L (1994) Small gain theorem for ISS systems and applications. Math Control Signals Syst 7:95–120
7. Khalil HK (2002) Nonlinear systems, 3rd edition. Prentice Hall, Englewood Cliffs, NJ
8. Nešić D, Teel AR (2004) Input–output stability properties of networked control systems. IEEE Trans Automatic Control 49:1650–1667
9. Nešić D, Teel AR (2004) Input-to-state stability of networked control system. Automatica 40:2121–2128
10. Sigman K (1995) Stationary marked point processes: an intuitive approach. Chapman & Hall, New York
11. Tabbara M, Nešić D (2007) Input–output stability of networked control systems with stochastic protocols and channels. IEEE Trans Automatic Control (to appear)
12. Tabbara M, Nešić D, Teel AR (2005) Input–output stability of wireless networked control systems. In: Proceedings of 44th IEEE Conference on Decision and Control, Seville, Spain, 209–214
13. Tabbara M, Nešić D, Teel AR (2007) Stability of wireless and wireline networked control systems. IEEE Trans Automatic Control (to appear)
14. Tanenbaum AS (2003) Computer networks, 4th edition. Prentice Hall, Englewood Cliffs, NJ
15. Walsh GC, Beldiman O, Bushnell LG (2001) Asymptotic behaviour of nonlinear networked control systems. IEEE Trans Automatic Control 46:1093–1097
16. Walsh GC, Ye H, Bushnell LG (1999) Stability analysis of networked control systems. In: Proceedings American Control Conference, San Diego, USA, 2876–2880
17. Walsh GC, Ye H (2001) Scheduling of networked control systems. IEEE Control Syst Mag 21:57–65,
18. Ye H, Walsh GC, Bushnell LG (2001) Real-time mixed-traffic wireless networks. IEEE Trans Ind Electron 48:883–890
19. Zhang W, Branicky MS, Phillips SM (2001) Stability of networked control systems. IEEE Control Syst Mag 21:84–99