Stability and Capacity through Evacuation Times

Leonidas Georgiadis*, Georgios S. Paschos†, Leandros Tassiulas†‡, Lavy Libman§
*AUTH, †ITI-CERTH, ‡UTH, Greece, and §University of NSW, Australia

***Abstract*—We consider a system where jobs (packets) arrive for processing using one of the policies in a given class. We study the connection between the minimal evacuation times and the stability region of the system under the given class of policies. The result is used to establish the equality of information theoretic capacity region and system stability region for the multiuser broadcast erasure channel with feedback.**

I. Introduction

In this work we consider a time slotted system where packets arrive to one of n different input queues - there may be other system queues to which packets are placed during their processing. The packets are processed by a policy from an admissible class. We study the connection between system stability and minimal *evacuation time*, i.e. the time it takes to complete processing a number of packets placed at the input queues at time 0, provided that no further arrivals occur afterwards. Under certain general assumptions on admissible policies and system statistics, it is shown that the stability region of the system is completely characterized by the asymptotic growth rate of minimal evacuation time. We make very few assumptions on the system structure and hence the result is applicable to a large number of applications in communications as well as more general control systems. However, we point out that the result, while intuitive, has to be applied with caution since there are systems for which its application leads to wrong conclusions. As an application to our methodology, we consider the N-user broadcast erasure channel with feedback. In this setup, we compare the information theoretic capacity region with the stability region and show that they are equal.

Concepts akeen to evacuation time and their relation to stability has been investigated in earlier works. Baccelli and Foss [1] consider a system fed by a marked point process and operating under a given policy. The concept of *dater* is used to describe the time of last activity in the system, if the system is fed only by the mth to nth , $m \leq n$ of the points of the marked process. Assuming that the dater is a deterministic function of the arrival times and the marks of the point process, and under additional assumption on dater sample paths, they show that stability under the specified policy is characterized by the asymptotic behavior of daters. These results are extended to continuous time input processes by Altman [2]. In our setup, the system evolution may depend on random factors as well as the characteristics of the arrival process. Moreover, we do not make sample path assumptions on specific policies. We rather specify features that admissible policies may have, and based on these we characterize the stability region of the class of admissible policies by the asymptotic growth rate of minimal (over all admissible policies) evacuation times.

A different, yet related, methodology is developed in [3] by Meyn; the *workload* $w(t)$ is defined as the time the server must work to clear all of the inventory of the system at time t when operating in the fluid limit. This basic concept is elaborated and used to derive significant results and obtain intuition for good control policies in specific complex networks. The concept of workload is closely related to the evacuation time, however we make minimal assumptions on system structure and the derived results are applicable to more general systems.

Regarding the relation between the information theoretic capacity and queueing theoretic stability regions, the equality of these has been shown recently in [4] for systems without feedback. The system studied in this work uses feedback, and as will be seen it can be derived in a simple manner based on stability characterization through evacuation time.

Due to space limitations, we refer the interested reader to [5] for proofs, examples and explanatory discussions.

II. System Model and Admissible Policies

We consider a time-slotted system where slot $t = 0, 1, ...$ corresponds to the time interval $[t,\ t+1)$. The system has n input queues of infinite length where jobs, or packets in case of a communication network, arrive; packets may move to other system queues during their processing. Packets arriving at each input may have certain properties, e.g., service times, routing options etc. There may be additional queues in the system, where packets may be placed during its operation. At the beginning of time slot t, i.e., at time t, $A_i(t)$ packets arrive at input i. We assume that the arrival processes satisfy the ergodicity conditions

$$\lim_{t\to\infty} \frac{\sum_{\tau=0}^{t} A_i(\tau)}{t} = \lambda_i > 0, \quad i = 1, 2, ..., n \qquad (1)$$

$$\lim_{t\to\infty} \frac{\mathbb{E}\left[\sum_{\tau=1}^{t} A_i(t)\right]}{t} = \lambda_i, \qquad i = 1, 2, ..., n. \qquad (2)$$

The operation of the system is characterized by a finite set of states $\mathcal{S}$, and control sets $\mathcal{G}_s$ for each $s \in \mathcal{S}$: if at the beginning of a slot the system state is $s \in \mathcal{S}$, one of the available controls $g \in \mathcal{G}_s$ is applied. There may be randomness in the behavior of the system, that is, given s and g at the beginning of a slot, the system state and the results at the end of a slot (e.g. packet losses) may be random.

Arriving packets are processed by the system following a policy π belonging to a class of admissible policies Π. At time t, when the system state is s, an admissible policy specifies:

1) The control $g \in \mathcal{G}_s$ to be chosen.
2) An action α among a set of available actions $\mathcal{A}_g$ when control g is chosen. An action specifies how packets are handled within the system.

The choice of controls and actions depends on the information available at time t, $\mathcal{H}_t$, which includes arrival and departure instants, states, controls, actions and results up to, and including time t. System states and controls are used to set the operational characteristics of the system and are distinct from actions taken once the system characteristics are set.

Departures. There are well-defined times when each arriving packet is considered to depart from the system. In some systems, it is natural to consider departure time as teh time at which a packet is delivered to a destination node. On the other hand, if the packet must be multicasted to a subset of the nodes, the departure time of the packet can be defined as the first time at which all nodes in the subset receive the packet. Thus, several definitions of departure times may make sense, and the particular choice depends on the performance measures of interest. At any time between the arrival and departure times of a packet p, we say that p *is in the system.*

There may be several restrictions on the policies in Π. The following properties are used by some admissible policies.

Properties of the admissible set of policies Π

1) At time t, the history of the system up to t, $\mathcal{H}_t$ is known.
2) At any time t at which there are packets only at the inputs of the system, it is permissible to take controls and actions taking into account only the packet at the inputs at time t, and proceed without taking into account the rest of $\mathcal{H}_{t}$.
3) If at time t there are $\boldsymbol{k}$ packets at the inputs of the system, it is permissible to pick any $\boldsymbol{m} \leq \boldsymbol{k}$ packets and continue processing the $\boldsymbol{m}$ packets, along with other packets that may be in the system, *without taking into consideration* the remaining $\boldsymbol{k}-\boldsymbol{m}$ packets in the input queues.

At the beginning of slot 0 let the system state be s and let there be $k_i \geq 0,\ i=1,...,n$ packets at input i and no arrivals afterward, i.e., $A_i(0)=k_i, A_i(t)=0, t=2,3,....$ Let $T_s^{\pi}(\boldsymbol{k}) \geq 0$, $\boldsymbol{k} \neq \boldsymbol{0}$ be the time it takes until all of these packets depart from the system under policy π. We call $T_s^{\pi}(\boldsymbol{k})$ *evacuation time* under policy π when the system starts in state s with $\boldsymbol{k}$ packets at the inputs, and denote its average value, $\bar{T}_s^{\pi}(\boldsymbol{k}) = \mathbb{E}\left[T_s^{\pi}(\boldsymbol{k})\right],\ \boldsymbol{k} \neq \boldsymbol{0}$. We define

$$\bar{T}_s^{*}(\boldsymbol{k}) \triangleq \inf_{\pi\in\Pi} \bar{T}_s^{\pi}(\boldsymbol{k}), \qquad \bar{T}^{*}(\boldsymbol{k}) \triangleq \max_{s\in\mathcal{S}} \bar{T}_s^{*}(\boldsymbol{k}). \tag{3}$$

We call $\bar{T}^{*}(\boldsymbol{k})$ *critical evacuation time function*. It will be seen that under certain statistical assumptions, this function determines the stability region of the policies under consideration.

Next, we present statistical assumptions regarding the system under consideration.

Statistical Assumptions

1) System and arrival statistics are known to a policy.
2) Markings (such as service times, permissible routing paths etc) associated with packets arriving to a given input are independent and statistically identical.
3) The system may start in any of the states in $\mathcal{S}$, i.e., any probability distribution of the initial state s_0 is possible.
4) If at the beginning of a slot t the system state is $s_t \in \mathcal{S}$ and control $g_t \in \mathcal{G}_{s_t}$ is taken, the results at time $t+1$ are independent of the system history before t. However, the system state s_{t+1} and the results at time $t+1$ may depend on both s_t and g_t. Hence the system states may be affected by the controls taken by a policy. Formally, if W_t is the (random) outcome at the end of a slot, we have for all t,
$$\Pr\left(W_{t+1}, S_{t+1}\left|s_t, g_t, \mathcal{H}_t\right.\right) = \Pr\left(W_{t+1}, S_{t+1}\left|s_t, g_t\right.\right).$$
5) At time $t=0,1,2,..$ let there be $\boldsymbol{k}$ packets in the system, not necessarily at the inputs. There is a policy π_h which can process all these packets until they all depart from the system by time $t+F^{\pi_h}(\boldsymbol{k})$, ($F^{\pi_h}(\boldsymbol{k})$ may be random) such that
$$\mathbb{E}\left[F^{\pi_h}(\boldsymbol{k})\right] \leq C_1 \sum_{i=1}^{n} k_i + C_0, \tag{4}$$
where the finite constants C_1, C_0 may depend on system statistics but not on the policy used.
6) Let $\boldsymbol{e}_i$ be the unit $n-$dimensional vector whose coordinates are all zero except the ith one which is one. It holds for all $i=1,..,n$, $\boldsymbol{k}$ and s,
$$\bar{T}_s^{*}(\boldsymbol{k}) - \bar{T}_s^{*}(\boldsymbol{k}+\boldsymbol{e}_i) \leq D_0 < \infty. \tag{5}$$

Although the above assumptions are satisfied by many practical systems, there exist systems (like the ALOHA protocol) for which they do not hold and as a consequence the corresponding result, namely Theorem 4, does not hold. See [5] for a detailed discussion.

A. Properties of Critical Evacuation Time Function

The following property of Critical Evacuation Time Function will play a key role in the following.

Lemma 1. *[Subadditivity] The Critical Evacuation Time Function is subadditive, i.e., the following holds for* $\boldsymbol{m} \geq \boldsymbol{0}$*,* $\boldsymbol{k} \geq \boldsymbol{0}$
$$\bar{T}^{*}(\boldsymbol{k}+\boldsymbol{m}) \leq \bar{T}^{*}(\boldsymbol{k}) + \bar{T}^{*}(\boldsymbol{m}) \tag{6}$$

Let $\mathbb{R}_0$ be the set of nonnegative real numbers. Based on the subadditivity of the Critical Evacuation Time, the following can be shown.

Theorem 2. *[Asymptotic Growth Rate] For any* $\boldsymbol{r} \in \mathbb{R}_0^n$*, the limit function* $\hat{T}(\boldsymbol{r}) = \lim_{t\to\infty} \frac{\bar{T}^{*}(\lceil t\boldsymbol{r}\rceil)}{t}$*,exists and is finite, positively homogeneous, convex and Lipschitz continuous, i.e., it holds* $\left|\hat{T}(\boldsymbol{r}) - \hat{T}(\boldsymbol{s})\right| \leq D\sum_{i=1}^{n}|r_i - s_i|$*. Moreover, for any sequence* $\boldsymbol{r}_t \in \mathbb{R}_0^n$ *such that* $\lim_{t\to\infty}\boldsymbol{r}_t = \boldsymbol{\lambda} < \boldsymbol{\infty}$*, it holds*
$$\lim_{t\to\infty} \frac{\bar{T}^{*}(\lceil t\boldsymbol{r}_t\rceil)}{t} = \hat{T}(\boldsymbol{\lambda}). \tag{7}$$

III. Stability

Let $D_{s,i}^{\pi}(t)$, $t \geq 1$, be the number of packets arrivals at input i that depart from the system by time t (end of time slot $t-1$) under policy $\pi \in \Pi$ when the system starts in state s. Define also, $D_{s,i}^{\pi}(0)=0$. In the following we will use the notation $\tilde{A}_i(t) = \sum_{\tau=0}^{t} A_i(\tau)$, $\tilde{D}_{s,i}^{\pi}(t) = \sum_{\tau=0}^{t} D_{s,i}^{\pi}(\tau)$, to denote the cumulative number or arrivals and departures respectively up to time t. Hence the number of packet arrivals at input i that are still in the system at time t (queue size) is, $Q_{s,i}^{\pi}(t) = \tilde{A}_i(t) - \tilde{D}_{s,i}^{\pi}(t)$. We denote by $\boldsymbol{Q}_s^{\pi}(t) = \left(Q_{s,i}^{\pi}(t)\right)_{i=1}^{n}$ and the total queue size by, $Q_s^{\pi}(t) = \sum_{i=1}^{n} Q_{s,i}^{\pi}(t)$.Let $\mathcal{V}$ be a set of assumptions about the statistics of the arrival processes, e.g., stationarity, independence, satisfying only (1) etc. Let also, $\mathcal{K}_0 = \{\boldsymbol{k},\ \Pr(\boldsymbol{A}(0)=\boldsymbol{k}) > 0\}$ be the set of all possible arrival vectors at time 0.

Definition 3. [System Stability] Under statistical assumptions $\mathcal{V}$, a policy $\pi \in \Pi$ is called stable for an arrival rate vector $\boldsymbol{\lambda} \geq \mathbf{0}$, if under any initial system state s, and under any $\boldsymbol{A}(0) = \boldsymbol{k} \in \mathcal{K}_0$, it holds,

$$\lim_{q\to\infty} \limsup_{t\to\infty} \Pr\left(Q_s^{\pi}(t) > q\right) = 0. \quad (8)$$

The stability region $\mathcal{R}^{\pi}$ of a policy π (under $\mathcal{V}$) is the closure of the set of the arrival vectors for which the policy is stable. The stability region $\mathcal{R}$ of the system is the closure of $\cup_{\pi\in\Pi}\mathcal{R}^{\pi}$.

Theorem 4. *a) If (1) and (2) hold, then*

$$\mathcal{R} \subseteq \left\{\boldsymbol{r} \geq \mathbf{0} : \hat{T}(\boldsymbol{r}) \leq 1\right\}.$$

b) If the arrival process vectors are independent and identically distributed (for a given time, the components of the vector may be dependent), it holds,

$$\mathcal{R} = \left\{\boldsymbol{r} \geq \mathbf{0} : \hat{T}(\boldsymbol{r}) \leq 1\right\}$$

Proof: (sketch) For part a) Assuming an arrival vector $\boldsymbol{\lambda}$ for which there is a stabilizing policy π_0, for any time t we generate $\boldsymbol{A}(\tau), \tau = 0, 1, ..., t$ packets according to the statistics of the arrival process (and the statistics of possible markings) and place them in the corresponding inputs at time 0. Next, we construct an evacuation policy which mimics π_0 for the first t slots and then evacuates any packets that may still be in the system at time t using policy π_h (see Statistical Assumption 5). Based on this construction and using the fact that due to the stability of π_0, for large t the number of packets processed by π_h are small, it can be shown that $\hat{T}(\boldsymbol{r}) \leq 1$.

For the sufficiency part, assuming $\hat{T}(\boldsymbol{r}) < 1$ we define an *epoch-based policy*, which works as follows. The policy operates in epochs of random duration, and during each epoch it evacuates the packets existing in the system at the beginning of the epoch. Any packets arriving during the current epoch, remain at the inputs waiting for the next epoch. Using the assumption of i.i.d. arrivals, the process of epoch lengths is shown to be Markovian and ergodic from which we derive the system stability. The complete proof can be found in [5]. ■

IV. Capacity and Stability region of the Broadcast Erasure Channel with Feedback

Consider a communication system consisting of a single transmitter and a set $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ of receivers/users. The transmitter has n infinite queues where packets destined to each of the receivers are stored. Packets consist of L bits and are transmitted within one slot. The channel is modeled as memoryless broadcast erasure (BE), so that each broadcast packet is either received unaltered by a user or is *erased* (i.e. the user does not receive the packet, but knows that a packet was sent). The latter case is equivalent to considering that the user receives the special symbol E, which is distinct from any other possible transmitted symbol and does not map to a physical packet (since it models an erasure). To complete the description of the system we also need to specify the outputs when no packet is sent by the transmitter, i.e., the slot is empty: in this case we assume that all receivers realize that the slot is empty. An empty slot will be denoted by S. Equivalently, we may view *no transmission* as transmission of a special symbol S.

In information-theoretic terms, the broadcast erasure channel under consideration is described by the tuple $(\mathcal{X}, (\mathcal{Y}_i \in \mathcal{N}), p(\boldsymbol{Y}_l|X_l))$, where $\mathcal{X}$ is the input symbol alphabet, $\mathcal{Y}_i = \mathcal{Y} = \mathcal{X} \cup \{E\}$ is the output symbol alphabet for user i, and $p(\boldsymbol{Y}_l|X_l))$ is the probability of having, at slot l, output $\boldsymbol{Y}_l = (Y_{i,l}, i \in \mathcal{N})$ for a broadcast input symbol X_l. The memoryless property implies that $p(\boldsymbol{Y}_l|X_l))$ is independent of l, so that it is simply written as $p(\boldsymbol{Y}|X))$. We denote by $\epsilon_{\mathcal{S}}, \mathcal{S} \subseteq \mathcal{N}$, the (common) probability that a transmitted packet (i.e. a symbol in $\mathcal{X} - \{S\}$) is erased by all users in the set $\mathcal{S}$. To avoid unnecessary complications we assume in the following that $\epsilon_{\{i\}} < 1$ for all i. Note that for the empty slot (symbol S) we have $\Pr(\boldsymbol{Y} = (S, \ldots, S)\,|X = S)$=1.

We consider that there is feedback from the users to the transmitter, so that at the end of each slot l, all users inform the transmitter whether the transmitted symbol was received or not (essentially, a simple ACK/NACK - of course no feedback is required if no transmission occurs, i.e., the slot is empty) through an error-free zero-delay control channel.

We define two regions for this channel, the information theoretic capacity region, and the stability region. The information theoretic capacity region describes transmission rates under which it is possible to transmit sets of messages (one for each user) placed at the transmitter by using proper encoding, so that all users receive the messages destined to them with arbitrarily small probability of error. For the stability region, Definition 3 is used, with proper definition of system queue size, under the assumption that packets arrive randomly to the system and are transmitted using proper encoding. We also place the requirement that packets are decoded by the receivers with *zero* probability of error. We show in the following that the two regions are identical.

Information theoretic capacity region

A channel code, denoted as $c_t = (M_1, \ldots, M_n, t)$, for the broadcast channel with feedback has the following components (this is an extension of the capacity definition of [6] to n users):

- **Message sets** $\mathcal{W}_i$ of size $|\mathcal{W}_i| = M_i$ for each user $i \in \mathcal{N}$, where $|\cdot|$ denotes set cardinality. Denote the message that needs to be communicated as $\boldsymbol{W} \triangleq (W_i, i \in \mathcal{N}) \in \mathcal{W}$, where $\mathcal{W} = \mathcal{W}_1 \times \ldots \times \mathcal{W}_n$. For our purposes it is helpful to interpret the message set $\mathcal{W}_i$ as follows: assume that user i needs to decode a given set $\mathcal{K}_i$ of L-bit packets. Then, $\mathcal{W}_i$ is the set of all possible $|\mathcal{K}_i|\, L$ bit sequences, so that it holds $|\mathcal{W}_i| = M_i = 2^{|\mathcal{K}_i|L}$. Henceforth we will assume this relation.
- **An encoder** that transmits, at slot l, a symbol $X_l = f_l(\boldsymbol{W}, \boldsymbol{Y}^{l-1}, X^{l-1})$, based on the value of $\boldsymbol{W}$ and all previously gathered feedback $\boldsymbol{Y}^{l-1} \triangleq (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{l-1})$, $\boldsymbol{Y}_k = (Y_{1,k}, ..., Y_{n,k})$ and channel input $X^{l-1} \triangleq (X_1, \ldots, X_{l-1})$. X_1 is a function of $\boldsymbol{W}$ only. A total of t symbols are transmitted for message $\boldsymbol{W}$.
- n **decoders**, one for each user $i \in \mathcal{N}$, represented by the decoding functions $g_i : \mathcal{Y}^t \to \mathcal{W}_i$ that map Y_i^t, where $Y_i^t \triangleq (Y_{i,1}, \ldots, Y_{i,t})$ is the sequence of symbols received by user i during the t slots, to a message in $\mathcal{W}_i$.

In the following we write $(M_1, \ldots, M_n, t)$ to denote the code c_t, with the understanding that the full specification requires all the components described above. The probability of erroneous decoding is defined as $q_t^e = \Pr(\cup_{i\in\mathcal{N}}\{g_i(Y_i^t) \neq W_i\})$, where it is assumed that the messages are selected according to the uniform distribution from $\mathcal{W}$. The rate $\boldsymbol{R}$ for this code, measured in information bits per transmitted symbols, is now defined as the vector $\boldsymbol{R} = (R_i : i \in \mathcal{N})$ with $R_i = (\log_2 M_i)/t$. Hence, it holds $R_i = |\mathcal{K}_i|\, L/t = r_i L$, where $r_i = |\mathcal{K}_i|\,/t$ is the rate of the code in packets per slot, and the bits of each packet are uniformly distributed and independent of the bits of the other packets. For our purposes, it will be convenient to define the capacity region of the system in terms of the rate vector $\boldsymbol{r}$.

A vector rate (in packets per slot) $\boldsymbol{r} = (r_1, \ldots, r_n)$ is achievable if there exists a sequence $\{c_t\}_{t=1}^{\infty}$ of codes $(2^{\lceil tr_1 \rceil L}, \ldots, 2^{\lceil tr_n \rceil L}, t)$ such that $q_t^e \to 0$ as $t \to \infty$. The capacity region $\mathcal{C}$ of the system is the closure of the set of achievable rates.

Stochastic Arrivals: Definitions of admissible policies

As in Section II, we assume that packets arrive randomly to the system according to the stochastic process $\boldsymbol{A}(t)$ and are stored in infinite buffers at the transmitter. We denote by $\boldsymbol{\mathcal{A}}(t)$ the content of these messages, i.e., $\boldsymbol{\mathcal{A}}(t) = (\boldsymbol{\mathcal{A}}_1(t), ..., \boldsymbol{\mathcal{A}}_n(t))$ where $\boldsymbol{\mathcal{A}}_i(t) = \left(p_{i,1}(t), ..., p_{i,A_i(t)}(t)\right)$, and $p_{i,j}(t)$ denotes the sequences of bits corresponding to the jth packet with destination node i that arrived at the transmitter at time t - if no packets arrive we consider that $\boldsymbol{\mathcal{A}}_i(t)$ is the empty set. We assume that $p_{i,j}(t)$ are uniformly distributed and independent of each other. We use again the standard notation, $\boldsymbol{\mathcal{A}}^l = (\boldsymbol{\mathcal{A}}(0), ..., \boldsymbol{\mathcal{A}}(l))$. An admissible policy consists of

- An encoder that transmits, at slot l, a symbol $X_l = f_l(\boldsymbol{Y}^{l-1}, X^{l-1}, \boldsymbol{\mathcal{A}}^l)$, based on all previously gathered feedback, $\boldsymbol{Y}^{l-1} \triangleq (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{l-1})$, channel inputs, $X^{l-1} \triangleq (X_1, \ldots, X_{l-1})$, and the contents of packet arrivals up to time l, $\boldsymbol{\mathcal{A}}^l$.
- n decoders, one for each user $i \in \mathcal{N}$, represented by the decoding set-valued functions $g_{i,t}(Y_i^t)$ that at time t maps Y_i^t to a subset $\mathcal{S}_t$ of the packets that have arrived up to time $t-1$ with destination node i, i.e.,

$$\mathcal{D}_t \subseteq \{p_{i,j}(\tau) : \tau \leq t-1,\ 1 \leq j \leq A_i(t)\}.$$

A packet is decoded the first time it is included in $\mathcal{D}_t$. We set the requirement that packet decoding is correct with probability one. Note that there is at least one policy that satisfies this requirement: this is the time-sharing policy where packets destined to destination i are (re)transmitted until successful reception First-Come-First-Served, in slots specifically assigned to i, say slots $kn + i - 1$, $k = 0, 1, ...$ We call such a policy One By One (OBO) policy, π_O. For this policy it can be easily seen that

$$\bar{T}_{\pi_O}(\boldsymbol{k}) \leq \sum_{i=1}^{n} C_i k_i + C_0,$$

where C_i depends on the erasure probabilities, but not on $\boldsymbol{k}$. Hence, π_O satisfies (4).

In order to apply the stability definition 3 to the class of policies specified above, we must define the time instant at which a packet leaves the system. There are two ways to define this instant. According to the first, a packet is considered to leave the system when it is correctly decoded by the destination receiver. While this definition make sense if one is interested in packet delivery times, it does not capture the fact that a decoded packet may still be needed for further encoding and decoding, in which case the packet will keep occupying buffer space even after its correct decoding. Also, the feedback information may need to be stored in the buffers of the transmitter if needed for further encoding. To capture buffer requirements we assume that each of the receivers has infinite buffers where received packets are stored. We next introduce a second definition of queue size, where we take into account the following.

1) Each transmission results in storing at most n packets, one at each receiver. Each of these packets are functions of *native* packets that have arrived exogenously at the transmitter (as well as functions of previously transmitted packets and the received feedback at the transmitter). Hence, in this case packets may be generated internally to the system during its operation.
2) A packet stored at a receiver buffer departs when it is not needed for further decoding.
3) A feedback packet is stored at the transmitter until it is not used for further encoding.
4) A native packet departs from the receiver if a) it has been decoded by the receiver to which it is destined and b) it is not used for further encoding.

If $Q_D^{\pi}(t)$ and $Q_B^{\pi}(t)$ respectively are the sum of queue sizes under π according to the previous two definitions of packet departure time (Delay, Buffer), it holds, $Q_D^{\pi}(t) \leq Q_B^{\pi}(t)$. Hence, if $\mathcal{S}_D$ and $\mathcal{S}_B$ are respectively the stability regions according to the two definitions, it holds

$$\mathcal{S}_B \subseteq \mathcal{S}_D \tag{9}$$

Relation between Capacity and Stability Regions.

The distributed nature of the channel introduces some new issues that must be addressed in order to apply the results of the previous sections. Specifically, while the transmitter has full knowledge of the system through the channel feedback, this is not the case for the receivers. Transferring appropriate information to the receivers takes extra slots which must be accounted for.

Note first that there are some differences in the information available at the receivers in the definition of the two regions given above. Specifically, in the capacity region definition, it is assumed that the receivers know the number of packets at the transmitter when the algorithm starts. On the other hand, when arrivals are stochastic, this information cannot be assumed *a priori* and if needed it must be communicated to the receivers. Also, in the capacity definition, all receivers under any admissible coding know implicitly the instant t at which the decoding process stops. For the stochastic arrival model, however, under a general evacuation policy, this may not be the case. Note that the One-By-One policy π_O does not need the information regarding the number of packets at the transmitter when the system starts. Also, an evacuation policy that is based on π_O can be easily modified to inform the receivers about the end of the decoding process: when all packets to destination i are transmitted, an empty slot is transmitted in the next slot allocated to i, informing all receivers of this event. Hence if the last packet is delivered to the appropriate destination at time t, all receivers will know at time $t+1$ that all packets are evacuated. Note that (4) still holds under this modification. We denote this modified policy again π_O^e

Since it can be agreed *a priori* which evacuation policy to employ when a given number of packets $\boldsymbol{k}$ is initially at the transmitter, once the number of packets initially at the transmitters is known by all receivers, the employed evacuation policy is also known by the receivers.

We proceed by assuming the following two conditions-these conditions will be removed later.

- When an evacuation policy starts, the number of packets at the transmitter is known to the receivers.
- An evacuation policy ensures that all receivers realize the end of the evacuation process at some time t, which is defined as the end of the evacuation process.

Under these conditions, it can be seen that Lemma 1 and Theorem 2 are still valid. Moreover, we have

Lemma 5. *It holds,*

$$\mathcal{C} = \mathcal{R} \triangleq \left\{ \boldsymbol{r} \geq \boldsymbol{0} : \hat{T}(\boldsymbol{r}) \leq 1 \right\}.$$

Proof: (sketch) We first show that $\mathcal{R} \subseteq \mathcal{C}$. For this, it suffices to show that if for some $\boldsymbol{r}$ it holds $\hat{T}(\boldsymbol{r}) < 1$, then there is a sequence of codes $c_t = (2^{\lceil tr_1 \rceil L}, \ldots, 2^{\lceil tr_n \rceil L}, t)$ with $\lim_{t\to\infty} q_t^e = 0$. The sequence c_t is constructed as follows. Based of the fact that $\hat{T}(\boldsymbol{r}) < 1$, for large enough t we can find an evacuation policy π_{t_0} such that

$$\frac{\bar{T}_{\pi_{t_0}}(\lceil t_0 \boldsymbol{r} \rceil)}{t_0} \leq \frac{\bar{T}^*(\lceil t_0 \boldsymbol{r} \rceil)}{t_0} + \delta < 1, \ \delta > 0.$$

Next, writing for any t, $t = l_t t_0 + v_t$, $0 \leq v_t < t_0$ we construct the code c_t as follows.

1) Use π_{t_0} to transmit successively $l_t + 1$ butches of $\lceil t_0 \boldsymbol{r} \rceil$ packets (the last butches may contain dummy packets) until they are decoded by all receivers. Let $T_{\pi_{t_0}}^m(\lceil t_0 \boldsymbol{r} \rceil)$ be the (random) time it takes to transmit the mth bunch, and
$$\tilde{T}_{\pi_{t_0}}^t(\lceil t_0 \boldsymbol{r} \rceil) = \sum_{m=1}^{l_t+1} T_{\pi_{t_0}}^m(\lceil t_0 \boldsymbol{r} \rceil).$$
2) If
$$\tilde{T}_{\pi_{t_0}}^t(\lceil t_0 \boldsymbol{r} \rceil) \leq t$$
all packets are correctly decoded. Else declare error.

Based on the fact that $T_{\pi_{t_0}}^m(\lceil t_0 \boldsymbol{r} \rceil)$ are i.i.d, an application of the Law of Large Numbers shows that the error probability of this sequence of codes goes to zero.

Next we show that $\mathcal{C} \subseteq \mathcal{R}$. Assume that there is a sequence of coding algorithms c_t with rate $\boldsymbol{r}$ whose error probability goes to zero as n goes to infinity. We then construct an evacuation policy π_t for evacuating $\lceil t\boldsymbol{r} \rceil$ packets as follows.

1) For $\epsilon > 0$, select t so that $q_t^e < \epsilon$.
2) Follow the steps of c_t for the first t slots.
3) If all receivers decoded correctly, leave slot $t+1$ empty, thus signaling to all receivers the end of the decoding process.
4) Else (i.e., if any of the receivers makes an error), send a dummy packet in slot $t+1$ (thus informing the receivers that decoding continues) and resend all the $\lceil t\boldsymbol{r} \rceil$ packets using the one-by-one policy π_O^e.

Note that the transmitter knows through channel feedback whether a receiver makes an error and hence the third step above is implementable. Then, using this evacuation policy to bound $\bar{T}^*(\lceil t\boldsymbol{r} \rceil)$, it can be shown that $\hat{T}(\boldsymbol{r}) \leq 1$. ■

It remains to relate $\mathcal{R}$ to $\mathcal{S}_D$ and $\mathcal{S}_B$ under the current model. Revisit the proof of Theorem 4, and use a policy $\pi_0 \in \mathcal{S}_D$ for the first t slots. If all packets are decoded correctly by t, leave slot $t+1$ empty, thus informing all receivers of successful decoding. Else send a dummy packet in slot $t+1$ and afterward the One-By-One policy π_O^e as policy π_h in the proof to evacuate the remaining packets. With these modifications, the proof can be used to show that

$$\mathcal{S}_D \subseteq \mathcal{R} \tag{10}$$

We now consider the implementation of the Epoch Based policy π_ϵ under the current model. This policy selects a particular evacuation policy for each epoch, which is a function of the number of packets $\boldsymbol{k}$ at the beginning of the epoch. In order to implement π_ϵ in the current model, the receivers must generally know $\boldsymbol{k}$ at the beginning of an epoch. The transfer of information about the number $\boldsymbol{k}$ is done by transmitting $O\left(\sum_{i=1}^n \log(k_i+1)\right)$ packets using π_O^e and hence the average number of slots to achieve this transfer is $O\left(\sum_{i=1}^n \log(k_i+1)\right)$. This increases the length of the evacuation period but since the increase is logarithmic in the number of packets, it does not affect the stability arguments. Note also that once an epoch ends, all $\boldsymbol{k}$ packets, as well as the feedback information and the packets stored at the receivers can be discarded since they are not used for further decoding by π_ϵ. Hence we conclude that

$$\mathcal{R} \subseteq \mathcal{S}_B \tag{11}$$

Taking into account (9), (10), (11) we finally conclude,

Theorem 6. *It holds,*

$$\mathcal{C} = \mathcal{R} = \mathcal{S}_B = \mathcal{S}_D.$$

ACKNOWLEDGMENT

This work was supported by the EC-FP7 projects STAMINA 265496 and CONNECT 257616.

REFERENCES

[1] F. Baccelli and S. Foss, "On the Saturation Rule for the Stability of Queues," *Journal of Applied Probability*, vol. 32, pp. 494–507, Jun. 1995.

[2] E. Altman, "The Baccelli-Foss Saturation Rule for stability for continuous time input processes," in *Allerton*, Sep. 1994, pp. 396–403.

[3] S. Meyn, *Control Techniques for Complex Networks*. Cambridge University Press, Jan. 2008.

[4] H. Yao, T. Ho, and M. Effros, "On the equivalence of Shannon capacity and stable capacity in networks with memoryless channels," in *Proceedings of IEEE ISIT*, Aug. 2011, pp. 503–507.

[5] L. Georgiadis, G. S. Paschos, L. Tassiulas, and L. Libman, "Minimal Evacuation Times and Stability," Tech. Report, https://sites.google.com/site/gpasxos/report.pdf, 2012.

[6] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.