

## Diagnose de falhas resiliente a atrasos de observação em Sistemas a Eventos Discretos em Rede utilizando sequence numbers<sup>\*</sup>

Marcos V. S. Alves<sup>\*</sup> Carlos E. V. Nunes<sup>\*\*</sup> Paolo Ferrari<sup>\*\*\*</sup>  
Dennis Brandão<sup>\*\*\*</sup> Marcos V. Moreira<sup>\*\*\*\*</sup>

<sup>\*</sup> Programa de Engenharia Elétrica, Universidade Federal de Sergipe, SE (e-mail: marcosvsalves@academico.ufs.br).

<sup>\*\*</sup> Departamento de Engenharia Elétrica e Computação, Universidade Federal da Bahia, BA (e-mail: carlosevnunes@ufba.br)

<sup>\*\*\*</sup> Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Brescia, Itália (e-mail: paolo.ferrari@unibs.it; dennis.brandao@unibs.it)

<sup>\*\*\*\*</sup> Programa de Engenharia Elétrica, COPPE/UFRJ, RJ (e-mail: moreira@dee.ufrj.br)

---

**Abstract:** In order to reduce the costs of implementation and maintenance, communication networks are used in networked Discrete-Event Systems (NDES), providing an efficient way to establish communication between several devices in the plant. In many cases, multi-hop mesh networks are used, as on wireless sensor networks, with re-transmissions due to degradation, leading to delays in communication and, consequently, the observation of events in a different order from their occurrence in the system. In this paper, we address the problem of fault diagnosis of NDES, resilient to observation delays in the communication network. We consider that each event is communicated together with its sequence number, and we show how the sequence number can be used to improve the fault diagnosis capability in the presence of observation delays, without the need to reorder the sequence of events according to the sequence number.

**Resumo:** Visando reduzir os custos de implementação e manutenção, redes de comunicação são utilizadas em Sistemas a Eventos Discretos em rede (SEDR), fornecendo uma maneira eficiente de estabelecer comunicação entre vários dispositivos da planta. Em muitos casos são usadas redes *mesh* multi-saltos, como em redes de sensores sem fio, com retransmissões devido à degradação da rede, levando a atrasos na comunicação e, conseqüentemente, à observação de eventos em uma ordem diferente das suas ocorrências no sistema. Neste trabalho, o problema de diagnose de falhas de SEDR resilientes a atrasos de observação em redes de comunicação é abordado. É considerado que cada evento é comunicado com seu *sequence number*, e é mostrado como o *sequence number* pode ser usado para melhorar a capacidade de diagnose de falhas na presença de atrasos de comunicação, sem a necessidade de reordenar a seqüência de eventos de acordo com o *sequence number*.

**Keywords:** Fault diagnosis; Discrete-Event Systems; Communication networks; Automata.

**Palavras-chaves:** Diagnose de falhas; Sistemas a Eventos Discretos; Redes de Comunicação; Autômatos.

---

### 1. INTRODUÇÃO

O número de dispositivos que se comunicam é crescente nos sistemas de automação modernos. Nesses casos, a infraestrutura de comunicação pode ser complexa, cara e difícil de manter devido à grande quantidade de cabos e conectores (Huo et al., 2004). Conseqüentemente, devido à sua flexibilidade, baixo custo, mobilidade e fácil implantação, redes sem fio são adequadas no contexto da Indústria 4.0 (Li et al., 2017).

Uma característica de redes de comunicação é que, em alguns casos, a comunicação entre os dispositivos pode sofrer com instabilidades, resultando em atrasos ou interrupções no tráfego de mensagens. Tais instabilidades podem ser causadas por fatores ambientais, degradação da infraestrutura física ou devido a variações no tráfego. Conseqüentemente, dados transmitidos por redes podem estar sujeitos a atrasos ou a perda de ordenamento quando enviados por múltiplos canais a um observador (Debouk et al., 2003; Nunes et al., 2018).

Diversos trabalhos apresentados na literatura abordam diferentes problemas relacionados a Sistemas Ciber-Físicos abstraídos como Sistemas de Eventos Discretos em Rede

---

<sup>\*</sup> Este trabalho foi desenvolvido com apoio do CNPq, FAPERJ, e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

(SEDR), com canais de comunicação não confiáveis. Em Carvalho et al. (2012), é abordado o diagnóstico de falhas de sistemas sujeitos a perdas intermitentes de observação, e um diagnosticador robusto é projetado. O problema de diagnosticabilidade robusta também é abordado em Carvalho et al. (2013) e Takai (2021), considerando perdas permanentes de observação dos eventos do sistema. Em Nunes et al. (2018), a codiagnosticabilidade de SEDR é introduzida e um diagnosticador robusto a atrasos e perdas na comunicação de eventos é apresentado. Em Viana et al. (2022), uma estrutura de temporização é usada para representar os atrasos de transmissão dos eventos, e uma propriedade de codiagnosticabilidade menos conservadora em comparação com Nunes et al. (2018) é obtida. Em Alves et al. (2021a) e Liu et al. (2022), atrasos de comunicação são considerados para o projeto de supervisores, e em Alves et al. (2021b), perdas intermitentes de observação são consideradas na síntese de supervisores robustos.

Em protocolos de rede utilizam-se *sequence numbers* para marcar os pacotes de dados enviados. Um *sequence number* é um valor incremental inserido junto a cada pacote de dados, e o seu formato numérico é um número inteiro positivo com tamanho variável, dependente do protocolo. Com a adição do *sequence number*, cada pacote de dados torna-se diferente do imediatamente anterior, mesmo que os dados transportados permaneçam inalterados no tempo. O *sequence number* é, assim, utilizado em redes de comunicação para se lidar com perdas de pacotes. Por exemplo, em Tong e Ma (2022), *sequence numbers* são usados para estimar estados de SEDR na presença de perdas de pacotes.

Neste artigo, o problema de diagnose de falhas de SEDR considerando atrasos de observação de eventos é abordado. Assim como em Tong e Ma (2022), é considerado que cada evento é comunicado com seu *sequence number*, e é mostrado como este pode ser utilizado para permitir a diagnose de falhas de sistemas sujeitos a atrasos de comunicação. É importante ressaltar que, de acordo com o conhecimento dos autores, não há na literatura nenhum outro trabalho que aborde o uso de *sequence numbers* para garantir a resiliência da diagnose de falhas em sistemas sujeito a atrasos de comunicação. Note que em Nunes et al. (2018), se o sistema se torna não-diagnosticável devido a atrasos de comunicação, não há como restaurar a diagnosticabilidade. Esse problema é contornado neste trabalho.

## 2. FUNDAMENTOS TEÓRICOS

Seja  $G = (X, \Sigma, f, x_0)$  um autômato determinístico, em que  $X$  é o conjunto de estados,  $\Sigma$  é o conjunto finito de eventos,  $f : X \times \Sigma \rightarrow X$  é a função de transição, que pode ser parcialmente definida sobre o seu domínio, e  $x_0$  é o estado inicial. O domínio da função de transição  $f$  pode ser estendido para  $X \times \Sigma^*$ , em que  $\Sigma^*$  denota o fecho de Kleene de  $\Sigma$ , como a seguir:  $f(x, \varepsilon) = x$ , e  $f(x, s\sigma) = f(f(x, s), \sigma)$ , para todo  $s \in \Sigma^*$  e  $\sigma \in \Sigma$ , em que  $\varepsilon$  denota a sequência vazia. A linguagem gerada por  $G$  é definida como  $L(G) = \{s \in \Sigma^* : f(x_0, s)!\}$ , em que  $f(x_0, s)!$  denota que  $f(x_0, s)$  é definida.

Considere que o conjunto de eventos de  $G$  seja particionado como  $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$ , em que  $\Sigma_o$  e  $\Sigma_{uo}$  denotam os conjuntos

de eventos observáveis e não-observáveis, respectivamente. A operação de projeção  $P_p : \Sigma^* \rightarrow \Sigma_p^*$ , em que  $\Sigma_p \subset \Sigma$ , é definida como  $P_p(\varepsilon) = \varepsilon$ ,  $P_p(\sigma) = \sigma$ , se  $\sigma \in \Sigma_p$ , ou  $P_p(\sigma) = \varepsilon$ , se  $\sigma \in \Sigma \setminus \Sigma_p$ , em que  $\setminus$  denota a diferença de conjuntos, e  $P_p(s\sigma) = P_p(s)P_p(\sigma)$  para todo  $s \in \Sigma^*$  e  $\sigma \in \Sigma$ . A projeção pode ser aplicada à linguagem  $L$  aplicando-se  $P_p(s)$  a todas as sequências  $s \in L$ . O autômato observador (Cassandras e Lafortune, 2008) de  $G$ , denotado por  $Obs(G, \Sigma_p)$ , é construído de forma que  $L(Obs(G, \Sigma_p)) = P_p(L(G))$ .

Sejam  $G_1$  e  $G_2$  dois autômatos. Então  $G_1 || G_2$  denota a composição paralela de  $G_1$  e  $G_2$  (Cassandras e Lafortune, 2008).

O intervalo  $[n_1, n_2]$  denota o conjunto de números inteiros maiores ou iguais a  $n_1$  e menores ou iguais a  $n_2$ , ou seja,  $[n_1, n_2] = \{n \in \mathbb{Z} : n_1 \leq n \leq n_2\}$ .

Seja  $\Sigma_f \subseteq \Sigma_{uo}$  o conjunto de eventos de falha. Neste trabalho, por simplicidade, é considerado haver somente um evento de falha, *i.e.*,  $\Sigma_f = \{\sigma_f\}$ .

*Definição 1.* Uma sequência de falha é uma sequência de eventos  $s$  tal que  $\sigma_f$  é um dos eventos que formam  $s$ . Uma sequência livre de falha, por outro lado, não contém o evento  $\sigma_f$ .  $\square$

A linguagem  $L_N \subset L(G)$  denota o conjunto formado por todas as sequências livres de falha de  $L(G)$ , e o subautômato de  $G$  que gera  $L_N$  é denotado por  $G_N$ . A linguagem  $L \subseteq \Sigma^*$  é dita ser viva se para todo  $s \in L$ , existe um evento  $\sigma \in \Sigma$ , tal que  $s\sigma \in L$ .

Em Sampath et al. (1995), a seguinte definição de diagnosticabilidade é apresentada.

*Definição 2.* (Diagnosticabilidade). Seja  $L(G)$  a linguagem gerada por  $G$ . Então,  $L(G)$  é dita ser diagnosticável com relação à projeção  $P_o : \Sigma^* \rightarrow \Sigma_o^*$  e  $\Sigma_f$  se

$$(\exists z \in \mathbb{N})(\forall s \in L(G) \setminus L_N)(\forall st \in L(G) \setminus L_N) \\ (\|t\| \geq z \Rightarrow P_o(st) \notin P_o(L_N)),$$

em que  $\|\cdot\|$  denota o comprimento de uma sequência.  $\square$

O conjunto formado por todos os prefixos de uma sequência  $s \in \Sigma^*$  é denotado por  $Pre(s)$  e o prefixo de  $s$  com comprimento  $j \leq \|s\|$  é denotado por  $Pre_j(s) \in Pre(s)$ .

## 3. FORMULAÇÃO DO PROBLEMA DE PROJETO DE DIAGNOSTICADORES RESILIENTES A ATRASOS DE OBSERVAÇÃO

Neste trabalho, considera-se a estrutura de diagnose de falhas em rede ilustrada na Figura 1, a qual é composta por uma planta  $G = (X, \Sigma, f, x_0)$ , um diagnosticador e um módulo *IO* remoto (*MR*) conectado a um conjunto de sensores,  $S_i$ ,  $i = 1, \dots, n$ , responsáveis pela detecção das ocorrências dos eventos observáveis da planta. Quando uma nova ocorrência de evento é detectada, o *MR* envia essa ocorrência para o diagnosticador por meio de uma rede de comunicação. Embora a rede de comunicação seja projetada para satisfazer os requisitos do sistema, como latência e *throughput*, seu desempenho pode sofrer degradações decorrentes, por exemplo, de interferências, da mudança de topologia da rede, ou do aumento do tráfego de dados. Em muitos casos são usadas redes *mesh* multi-

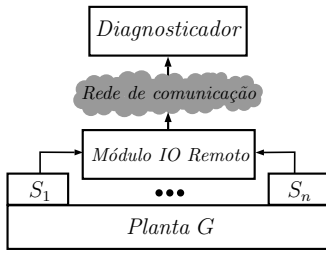


Figura 1. Estrutura de diagnóstico de falhas em rede.

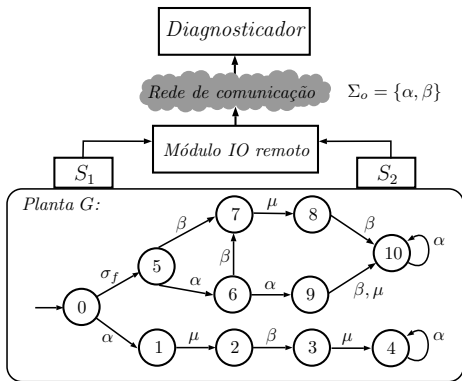


Figura 2. Exemplo de um sistema de diagnose de falhas em rede.

saltos, como em redes de sensores sem fio, com retransmissões devido à degradação da rede, levando a atrasos na comunicação e, conseqüentemente, à observação de eventos numa ordem diferente daquela em que eles foram enviados pelo *MR*. Como consequência, é importante projetar um diagnosticador que seja resiliente ao recebimento de pacotes fora de ordem, no sentido de manter a capacidade de identificação da ocorrência de um evento de falha mesmo durante uma degradação da rede.

Supõe-se que a transmissão dos pacotes de dados com ocorrências de eventos está sujeita a atrasos limitados, medidos em passos (Tripakis, 2004), em que um passo corresponde à ocorrência de um evento na planta, ou seja, o atraso na comunicação da ocorrência de um evento  $\sigma \in \Sigma_o$  é medido pelo número de eventos gerados pela planta após a ocorrência de  $\sigma$  e antes da sua observação pelo diagnosticador. Por exemplo, se  $\Sigma = \Sigma_o = \{a, b\}$  e o atraso de comunicação máximo é  $D = 2$  passos, então a sequência  $s = abbb \in \Sigma^*$  pode ser observada pelo diagnosticador em uma das formas:  $abbb$ ,  $babb$  e  $bbab$ , em decorrência de um atraso na observação de  $a$  de zero, um e dois passos, respectivamente. No exemplo a seguir, ilustram-se as consequências de atrasos de comunicação na diagnose de falhas.

*Exemplo 1.* Considere o sistema de diagnose de falhas em rede mostrado na Figura 2, cuja planta é modelada pelo autômato  $G = (X, \Sigma, f, x_0)$ , em que  $\Sigma = \{\alpha, \beta, \mu, \sigma_f\}$  é o conjunto de eventos da planta, e  $\Sigma_o = \{\alpha, \beta\}$  e  $\Sigma_f = \{\sigma_f\}$ .

Pode-se verificar que, se a comunicação entre *MR* e o diagnosticador não é sujeita a atrasos, então  $L(G)$  é diagnosticável em relação a  $P_o : \Sigma^* \rightarrow \Sigma_o^*$  e  $\Sigma_f$ , uma vez que o diagnosticador sempre detectará a ocorrência da falha  $\sigma_f$  após a observação de uma das sequências a seguir:  $P_o(\sigma_f\beta) = \beta$ ,  $P_o(\sigma_f\alpha\alpha) = \alpha\alpha$  ou  $P_o(\sigma_f\alpha\beta\mu\beta) = \alpha\beta\beta$ .

Suponha agora que a comunicação entre *MR* e o diagnosticador está sujeita a atrasos de, no máximo,  $D = 1$  passo. Nesse caso, o diagnosticador proposto em Sampath et al. (1995) não consegue diagnosticar a falha após a ocorrência da sequência de falha  $s_f = \sigma_f\alpha\beta\alpha^n$ ,  $n \in \mathbb{N}$ , uma vez que essa sequência pode ser confundida com a sequência livre de falha  $s_n = \alpha\mu\beta\mu\alpha^{n+1}$  se, em decorrência de um atraso de 1 passo, a segunda ocorrência do evento  $\alpha$  em  $s_f$  for observada após a observação de  $\beta$ , gerando a sequência observável  $\alpha\beta\alpha^{n+1}$ , que é igual a  $P_o(s_n)$ . De fato, como pode ser visto por meio da abordagem proposta em Nunes et al. (2018), a linguagem do sistema torna-se não diagnosticável (em rede) quando a transmissão das ocorrências de eventos está sujeita a atrasos de até 1 passo.  $\square$

Como ilustrado no Exemplo 1, trocas na ordem de observação dos eventos da planta, causadas por atrasos de comunicação, podem afetar a diagnosticabilidade do sistema. Neste trabalho, o problema de diagnose de falhas em rede na presença de atrasos de observação é estudado por meio de uma nova abordagem, diferente da empregada em Nunes et al. (2018), na qual é necessário que as trocas de ordens de observação não afetem a diagnosticabilidade em rede do sistema, o que não é satisfeito em muitas situações. Na abordagem proposta neste artigo, visando restaurar a diagnosticabilidade do sistema na presença de trocas de ordem de observação, utiliza-se um protocolo de comunicação que rotula cada pacote de dados, no seu envio, consoante o seu *sequence number*. Com base nesse protocolo, propõe-se um método para modelar a comunicação das ocorrências de eventos da planta para o diagnosticador e, também, um algoritmo de implementação de um diagnosticador em rede resiliente a trocas de ordem de observação.

Como em Nunes et al. (2018), supõem-se as hipóteses a seguir:

- A1.  $L(G)$  é diagnosticável em relação a  $P_o : \Sigma^* \rightarrow \Sigma_o^*$  e  $\Sigma_f$ ;
- A2. Os atrasos de comunicação são limitados, sendo o atraso de comunicação máximo  $D \in \mathbb{N}$  conhecido.

#### 4. ROTULAÇÃO DE PACOTES DE DADOS COM SEQUENCE NUMBERS

A rotulação de pacotes de dados com *sequence numbers* pode ser útil na realização da diagnose de falhas na presença de atrasos de observação, uma vez que ela permite identificar que ocorrências de eventos foram recebidas fora da ordem de ocorrência na planta. Na prática, utiliza-se um número limitado de  $b$  bits para o *sequence number*. Dessa forma, os *sequence numbers* pertencem a um subconjunto finito de números inteiros  $[0, N - 1] \subset \mathbb{Z}_+$ , em que  $N = 2^b$ . Embora, em alguns protocolos de rede,  $N$  seja grande, sistemas industriais trabalham continuamente e, como consequência, o número de pacotes enviados pode eventualmente alcançar  $N$ , fazendo com que os *sequence numbers* dos novos pacotes sejam novamente atribuídos a partir de 0. Assim, o *sequence number* atribuído ao  $i$ -ésimo pacote enviado ( $i$ -ésima ocorrência de evento observável) é igual ao resto da divisão de  $(i - 1)$  por  $N$ , denotado por  $n = (i - 1) \bmod N$ . Portanto, um pacote enviado do *MR* para o diagnosticador é denotado por um par  $(\sigma, n) \in \Sigma_o \times [0, N - 1]$ , em que  $\sigma$  é o evento transmitido nesse pacote

de dados e  $n$  é o seu *sequence number*. Vale ressaltar que, vários pacotes podem receber o mesmo *sequence number*, porém dois pacotes com mesmo *sequence number* transmitem ocorrências de eventos que estão espaçadas por, pelo menos,  $N$  passos, ou seja,  $N$  ocorrências de eventos na planta.

Em relação aos *sequence numbers*, supõe-se a hipótese a seguir:

**A3.**  $N > D$ .

Note que a hipótese **A3** é necessária para o reordenamento de pacotes que tenham o mesmo *sequence number*. Por exemplo, supondo que  $N = D = 4$ . Nesse caso, o primeiro pacote de dados pode ser recebido após o quinto pacote. Como ambos possuem o mesmo *sequence number*, igual a 0, não é possível saber qual foi enviado primeiro. Por outro lado, quando  $N > D$ , pacotes de mesmo *sequence number* nunca podem trocar de ordem entre si, possibilitando a reordenação correta dos pacotes.

Vale ressaltar que em situações reais, a Hipótese **A3** é usualmente atendida, uma vez que diversos protocolos de comunicação utilizam números com 16 ou mais *bits* para representar *sequence numbers*, tornando  $N$  relativamente grande e, por outro lado, o atraso máximo de observação  $D$  é, em geral, pequeno, uma vez que a detecção de uma falha após um grande atraso torna-se inútil na prática.

*Exemplo 2.* Considere o sistema de diagnose de falhas em rede descrito no Exemplo 1 (Figura 2). Suponha que a comunicação entre  $MR$  e o diagnosticador está sujeita a atrasos de, no máximo,  $D = 1$  passo, e considere, agora, que os pacotes de dados são rotulados em função dos seus *sequence numbers*. Por exemplo, se a sequência livre de falha  $s_n = \alpha\mu\beta\mu\alpha^{n+1}$  ocorrer na planta, então o pacote  $(\alpha, 0)$  será enviado para o diagnosticador após a detecção da primeira ocorrência de  $\alpha$ , o pacote  $(\beta, 1)$  será enviado após a detecção da ocorrência de  $\beta$ , e os pacotes  $(\alpha, i)$ ,  $i = 2, \dots, n + 2$ , serão enviados, um a um, após as detecções das demais ocorrências de  $\alpha$ . Por outro lado, se a sequência de falha  $s_f = \sigma_f\alpha\alpha\beta\alpha^n$  ocorrer, então os pacotes  $(\alpha, 0)$  e  $(\alpha, 1)$  serão enviados para o diagnosticador após a primeira e a segunda ocorrências de  $\alpha$ , o pacote  $(\beta, 2)$  será enviado após a ocorrência de  $\beta$ , e os pacotes  $(\alpha, i)$ ,  $i = 3, \dots, n + 2$  serão enviados, um a um, após as detecções das demais ocorrências de  $\alpha$ . Uma vez que os *sequence numbers* dos pacotes são enviados, a sequência de falha  $s_f$  torna-se diagnosticável, mesmo no caso em que a segunda ocorrência do evento  $\alpha$  em  $s_f$  for observada após a observação de  $\beta$ , já que o pacote  $(\beta, 2)$  (resp.  $(\beta, 1)$ ) é enviado ao diagnosticador após a ocorrência de  $\beta$  em  $s_f$  (resp.  $s_n$ ) e, conseqüentemente, o diagnosticador distingue  $s_f$  de  $s_n$  logo após receber esse pacote.

Suponha, agora, que a comunicação entre  $MR$  e o diagnosticador está sujeita a atrasos de, no máximo,  $D = 2$  passos, e considere o caso em que a planta executa a sequência de falhas  $s'_f = \sigma_f\beta\mu\beta$ . Embora o sistema não seja diagnosticável segundo a noção de diagnosticabilidade em rede proposta em Nunes et al. (2018), o diagnosticador em rede apresentado em Nunes et al. (2018) poderia identificar a ocorrência de  $\sigma_f$  após  $s'_f$ . No entanto, como a abordagem proposta em Nunes et al. (2018) não considera o envio de *sequence numbers*, seu diagnosticador

seria incapaz de diagnosticar a falha após observar apenas a primeira ocorrência do evento  $\beta$ , precisando observar também a segunda ocorrência de  $\beta$ , mesmo no caso em que a primeira ocorrência de  $\beta$  é observada sem atraso. Isso é consequência do fato da observação de apenas um  $\beta$  ser associada a uma possível ocorrência da sequência livre de falha  $s'_n = \alpha\mu\beta$ , em uma situação na qual  $\beta$  foi observado pelo diagnosticador, porém, a ocorrência de  $\alpha$  ainda está sendo transmitida devido a um atraso de observação de 2 passos. Por outro lado, considerando-se o envio dos *sequence numbers*, os pacotes  $(\beta, 0)$  e  $(\beta, 1)$  são enviados após a primeira e a segunda ocorrências do evento  $\beta$  em  $s'_f$ , respectivamente. Note que, nesse caso, ao receber o pacote  $(\beta, 0)$ , com a primeira ocorrência de  $\beta$ , um diagnosticador, adequadamente projetado, já poderia identificar a ocorrência do evento de falha, uma vez que o pacote  $(\beta, 0)$  não é enviado quando a planta executa qualquer uma das sequências livres de falha em  $L_N$ .  $\square$

O Exemplo 2 ilustra que, além de permitir a identificação de trocas de ordem de observação, os *sequence numbers* servem para rotular ocorrências de eventos em função das suas posições na sequência de eventos observáveis gerada pela planta, o que pode ser vantajoso para a diagnose de falhas em rede mesmo em situações em que trocas de ordem de observação não tenham ocorrido. Neste trabalho, propõe-se, inicialmente, um método para construção de um modelo aumentado da planta do sistema, no qual as observações de eventos bem sucedidas são rotuladas em função dos *sequence numbers* dos pacotes de dados em que foram transmitidas. Em seguida, o modelo proposto é aplicado na elaboração de um diagnosticador em rede, que se beneficia dos *sequence numbers*, para diagnosticar ocorrências de eventos de falha mesmo na presença de observações fora de ordem. Vale ressaltar que, a utilização do modelo aumentado proposto, permite obter um diagnosticador *on-line* eficiente, uma vez que esse diagnosticador não precisa reordenar pacotes recebidos, esperar o recebimento de pacotes atrasados (caso já seja possível identificar a ocorrência da falha), ou retroceder no cálculo da estimativa de estado ao receber pacotes atrasados.

## 5. MODELAGEM DE SEDR COM PACOTES DE DADOS ROTULADOS COM SEQUENCE NUMBERS

Nesta seção, apresenta-se uma forma de obter um autômato  $G_{net}$  que modela o comportamento de um SEDR, no contexto do problema de diagnose de falhas em rede apresentado na Seção 3. O autômato  $G_{net}$  representa todas as possíveis observações de eventos devido às trocas de ordem, considerando que, no envio, os pacotes de dados são rotulados com os seus respectivos *sequence numbers*.

### 5.1 Função de inserção de pacotes recebidos

Seja  $(\sigma, n) \in \Sigma_o \times [0, N - 1]$  um pacote de dados transmitido de  $MR$  para o diagnosticador, e  $(\sigma, n)_r$  o evento que representa que o pacote  $(\sigma, n)$  foi recebido com sucesso pelo diagnosticador. Dessa forma, o conjunto de eventos de recebimento de pacotes é definido como  $\Sigma_r = \{(\sigma, n)_r : (\sigma, n) \in \Sigma_o \times [0, N - 1]\}$ . Por exemplo, se  $\Sigma_o = \{\alpha, \beta\}$ , então  $\Sigma_r = \{(\alpha, 0)_r, (\beta, 0)_r, (\alpha, 1)_r, (\beta, 1)_r, \dots, (\alpha, N - 1)_r, (\beta, N - 1)_r\}$ .

Seja  $\sigma_r = (\sigma, n)_r \in \Sigma_r$ . Então,  $\sigma_r(1)$  e  $\sigma_r(2)$  denotam, respectivamente, o evento transmitido no pacote  $(\sigma, n)$  e o seu *sequence number*, ou seja,  $\sigma_r(1) = \sigma$  e  $\sigma_r(2) = n$ .

A partir dos conjuntos  $\Sigma$  e  $\Sigma_r$ , pode-se definir o conjunto de eventos aumentado como  $\Sigma_a = \Sigma \cup \Sigma_r$ . Adicionalmente, definem-se as projeções  $P_a : \Sigma_a^* \rightarrow \Sigma^*$ ,  $P_{a,r} : \Sigma_a^* \rightarrow \Sigma_r^*$  e  $P_{a,o} : \Sigma_a^* \rightarrow \Sigma_o^*$ , e a função  $\eta : \Sigma_a^* \rightarrow [0, N - 1]$ , em que  $\eta(w) = (\|P_{a,o}(w)\| - 1) \bmod N$ , ou seja,  $\eta(w)$  é igual ao *sequence number* do pacote que contém o último evento enviado de  $w$ .

*Exemplo 3.* Considere o sistema do Exemplo 1 (Figura 2), em que  $N = 4$  e  $D = 1$  passo. Assim,  $\Sigma_r = \{(\alpha, 0)_r, (\beta, 0)_r, (\alpha, 1)_r, (\beta, 1)_r, (\alpha, 2)_r, (\beta, 2)_r, (\alpha, 3)_r, (\beta, 3)_r\}$ .

Considere a sequência  $s = \sigma_f \alpha \alpha \beta \alpha \in L(G)$ . A ocorrência e observação de  $s$  pode ser representada por um conjunto de sequências aumentadas definidas sobre  $\Sigma_a = \Sigma \cup \Sigma_r$ . Por exemplo, a sequência aumentada  $w = \sigma_f \alpha (\alpha, 0)_r \alpha (\alpha, 1)_r \beta (\beta, 2)_r \alpha (\alpha, 3)_r$  representa o caso em que todas as ocorrências de eventos observáveis em  $s$  são recebidas sem atraso pelo diagnosticador. Por outro lado, a sequência aumentada  $v = \sigma_f \alpha (\alpha, 0)_r \alpha \beta (\beta, 2)_r (\alpha, 1)_r \alpha$  representa o caso em que a primeira ocorrência de  $\alpha$  e a ocorrência de  $\beta$  são observadas sem atraso, a segunda ocorrência de  $\alpha$  é observada com atraso de um passo e em ordem diferente daquela em que foi gerada na planta, e a terceira ocorrência de  $\alpha$  ainda está sendo transmitida. Note que a projeção  $P_a$  (resp.  $P_{a,o}$ ), quando aplicada em  $w$  e  $v$ , retorna  $s$  (resp.  $P_o(s)$ ), ou seja, a sequência de eventos (resp. eventos observáveis) gerada pela planta. Por sua vez, as projeções  $P_{a,r}(w) = (\alpha, 0)_r (\alpha, 1)_r (\beta, 2)_r (\alpha, 3)_r$  e  $P_{a,r}(v) = (\alpha, 0)_r (\beta, 2)_r (\alpha, 1)_r$  são as sequências de pacotes que já foram recebidos pelo diagnosticador em  $w$  e  $v$ , respectivamente.

Note que  $\eta(v) = (\|P_{a,o}(v)\| - 1) \bmod 4 = (\|\alpha \alpha \beta \alpha\| - 1) \bmod 4 = 3$ , que é igual ao *sequence number* associado ao último pacote de dados enviado, i.e., o pacote associado à terceira ocorrência de  $\alpha$  em  $w$ .  $\square$

Seja  $w \in \Sigma_a^*$ . Então,  $w^{(j)}$ ,  $j \in [1, \|w\|]$ , denota o  $j$ -ésimo evento em  $w$ . Além disso, para  $j \in [1, \|w\|]$  tal que  $w^{(j)} \in \Sigma_o$ , define-se  $\rho(w, j) = \min(\{\|w\|\} \cup I_{w^{(j)}})$  em que

$$I_{w^{(j)}} = \{i \in [j + 1, \|w\|] : (w^{(i)} \in \Sigma_r) \wedge (w^{(i)}(1) = w^{(j)}) \wedge (w^{(i)}(2) = \eta(\text{Pre}_j(w)))\},$$

ou seja,  $I_{w^{(j)}}$  é o conjunto de índices formado com as posições  $i$ ,  $j < i \leq \|w\|$ , dos eventos de recebimento de pacotes em  $w$  que transmitem um evento igual ao  $j$ -ésimo evento de  $w$ , i.e.,  $w^{(i)}(1) = w^{(j)}$ , e que possuem *sequence numbers* compatíveis com o  $j$ -ésimo evento de  $w$ , i.e.,  $w^{(i)}(2) = \eta(\text{Pre}_j(w))$ . Assim,  $\rho(w, j)$  é igual à posição do evento em  $w$  que representa a chegada, no diagnosticador, do pacote com a ocorrência de  $w^{(j)}$ , se esse pacote já foi recebido pelo diagnosticador, ou é igual a  $\|w\|$  se esse pacote ainda está sendo transmitido para o diagnosticador.

*Exemplo 4.* Para ilustrar a definição de  $\rho$ , considere a sequência  $v = \sigma_f \alpha (\alpha, 0)_r \alpha \beta (\beta, 2)_r (\alpha, 1)_r \alpha$ . Note que  $v^{(5)} = \beta$  e que o *sequence number* associado ao pacote de dados dessa ocorrência de evento é igual a  $\eta(\text{Pre}_5(v)) = \eta(\sigma_f \alpha (\alpha, 0)_r \alpha \beta) = (\|\alpha \alpha \beta\| - 1) \bmod 4 = 2$ . Dessa forma,  $I_{v^{(5)}}$  é formado com os índices  $i$ ,  $5 < i \leq 8$  tais que  $v^{(i)} = (\beta, 2)_r$ , ou seja,  $I_{v^{(5)}} = \{6\}$ . Portanto,  $\rho(v, 5) =$

$\min(\{8\} \cup \{6\}) = 6$ , que é igual à posição do evento  $(\beta, 2)_r$  em  $v$ .

Considere agora a terceira ocorrência de  $\alpha$  em  $v$ , isto é,  $v^{(8)}$ . Nesse caso, o conjunto  $I_{v^{(8)}}$  deve ser formado com os índices  $i$ ,  $8 < i \leq 8$  tais que  $v^{(i)} = (\alpha, 3)_r$  e, portanto,  $I_{v^{(8)}} = \emptyset$  uma vez que o pacote com a terceira ocorrência de  $\alpha$  ainda não foi recebido. Por fim,  $\rho(v, 8) = \min(\{\|v\|\} \cup \emptyset) = \|v\| = 8$ .  $\square$

*Definição 3.* (Função de inserção de pacotes recebidos). A função de inserção de pacotes recebidos é um mapeamento:

$$\chi : \Sigma^* \rightarrow 2^{\Sigma_a^*} \\ s \mapsto \chi(s) = \{w \in \Sigma_a^* : (w \models C1) \wedge (w \models C2) \wedge (w \models C3)\}$$

em que a notação  $w \models C_i$  representa que  $w$  satisfaz  $C_i$ , e as condições  $C1$ ,  $C2$  e  $C3$  são definidas, em função de  $N$  e  $D$ , a seguir:

- C1.**  $P_a(w) = s$ ;
- C2.** Para todo  $j \in \{1, \dots, \|w\|\}$ ,  
 $w^{(j)} \in \Sigma_o \Rightarrow \|P_a(\text{Pre}_{\rho(w,j)}(w))\| - \|P_a(\text{Pre}_j(w))\| \leq D$
- C3.** Para todo  $j \in \{1, \dots, \|w\|\}$ ,  
 $w^{(j)} \in \Sigma_r \Rightarrow (\exists i \in [1, (j - 1)], w^{(i)} = w^{(j)}(1) \wedge \eta(\text{Pre}_i(w)) = w^{(j)}(2) \wedge (\nexists l \in [i, (j - 1)], w^{(l)} = w^{(j)})$

A extensão de  $\chi$  para o domínio  $2^{\Sigma^*}$  é definida como  $\chi(L) := \bigcup_{s \in L} \chi(s)$ .  $\square$

A condição C1 da Definição 3 garante que  $w$  é obtido de  $s$  inserindo-se apenas eventos de  $\Sigma_r$ . A condição C2 garante que o atraso entre a ocorrência de um evento  $\sigma \in \Sigma_o$ , e o evento de recebimento do seu pacote  $(\sigma, n)_r$ , contado como o número de eventos gerados pela planta, é menor ou igual ao atraso de observação máximo  $D$ . Por fim, a condição C3 garante que um evento de recebimento de pacote  $(\sigma, n)_r \in \Sigma_r$  ocorre somente após um evento  $\sigma$  ainda não recebido e cuja posição em  $w$  faz o pacote com a ocorrência de  $\sigma$  ser rotulado com  $n$ . O exemplo a seguir ilustra a aplicação da função  $\chi$ .

*Exemplo 5.* Prosseguindo o Exemplo 3, considere a sequência  $s' = \sigma_f \alpha \alpha \beta \in L(G)$ . A seguir, será mostrado que, conforme a Definição 3, as seguintes sequências não pertencem a  $\chi(s')$ : (i)  $u = \alpha (\alpha, 0)_r \beta (\beta, 1)_r \alpha (\alpha, 2)_r$ ; (ii)  $w = \sigma_f \alpha \alpha (\alpha, 1)_r \beta (\alpha, 0)_r (\beta, 2)_r$ ; e (iii)  $t = \sigma_f \alpha \alpha (\alpha, 0)_r (\alpha, 1)_r \beta (\beta, 2)_r (\alpha, 1)_r$ .

(i) Note que como  $P_a(u) = \alpha \beta \alpha$  então, segundo a condição C1 da Definição 3,  $u$  não corresponde a ocorrência da sequência  $s' = \sigma_f \alpha \alpha \beta$ ;

(ii) Embora  $P_a(w) = \sigma_f \alpha \alpha \beta$ , o recebimento do pacote de dados com a primeira ocorrência de  $\alpha$ , modelado pelo evento  $(\alpha, 0)_s$  em  $w$ , indica um atraso de observação de 2 passos, enquanto se supõe um atraso máximo de 1 passo, o que é capturado pela condição C2 da Definição 3 uma vez que  $w^{(2)} \in \Sigma_o$  e  $\|P_a(\text{Pre}_{\rho(w,2)}(w))\| - \|P_a(\text{Pre}_2(w))\| = \|P_a(\text{Pre}_6(w))\| - \|P_a(\text{Pre}_2(w))\| = \|P_a(\sigma_f \alpha \alpha (\alpha, 1)_r \beta (\alpha, 0)_r)\| - \|P_a(\sigma_f \alpha)\| = \|\sigma_f \alpha \alpha \beta\| - \|\sigma_f \alpha\| = 2 > 1$ ;

(iii) A sequência  $t$  satisfaz as condições C1 e C2, mas possui uma ocorrência de  $(\alpha, 1)_r$  sem uma ocorrência

prévia do evento  $\alpha$  associado a ela, o que é capturado pela condição C3 uma vez que  $t^{(8)} \in \Sigma_r$  e  $\nexists i \in [1, 7]$  tal que  $(t^{(i)} = \alpha) \wedge (\eta(\text{Pre}_i(t)) = 1) \wedge (\nexists l \in [i, 7], t^{(l)} = (\alpha, 1)_r)$ , note que, para  $i = 3$ ,  $t^{(3)} = \alpha$  e  $\eta(\text{Pre}_3(t)) = 1$  porém, considerando-se  $l = 5$ ,  $t^{(5)} = (\alpha, 1)_r$ .  $\square$

### 5.2 Função de substituição dos pacotes recebidos

A construção de um autômato que gera a linguagem  $\chi(L(G))$  não é, em geral, computacionalmente eficiente, pois o espaço de estados desse autômato tende a crescer em função de  $N$ . Por exemplo, no caso em que são utilizados 16 bits para o *sequence number*, têm-se  $N = 2^{16}$  e, assim, um autômato que gera  $\chi(\Sigma^*)$  possuirá, no mínimo,  $2^{16}$  estados.

Para evitar a explosão de estados do modelo, nesta seção, propõe-se uma função  $F$  que substitui os eventos de  $\Sigma_r$  nas seqüências obtidas com a função  $\chi$ . Um evento  $(\sigma, n)_r \in \Sigma_r$  será substituído por um evento definido em função do número de pacotes enviados por  $MR$  antes do pacote  $(\sigma, n)$  e que ainda não foram recebidos pelo diagnosticador. Como o número de pacotes atrasados não pode ser maior que o atraso de observação máximo  $D$ , obtém-se, por consequência, uma linguagem que pode ser gerada por um autômato que cresce em função de  $D$ , o qual é, em geral, muito menor que  $N$ . Dessa forma, introduz-se o conjunto de eventos de observação bem sucedida  $\Sigma_s$  definido como

$$\Sigma_s = \{\sigma_{s,j} : \sigma \in \Sigma_o \wedge j \in [0, D]\}, \quad (1)$$

em que o evento de observação bem sucedida  $\sigma_{s,j} \in \Sigma_s$  representa a observação do evento  $\sigma \in \Sigma$  enquanto  $j$  outros eventos foram enviados antes de  $\sigma$  mas ainda não foram recebidos pelo diagnosticador.

*Exemplo 6.* Visando ilustrar como os eventos pertencentes a  $\Sigma_r$  serão substituídos por eventos em  $\Sigma_s$ , considere, mais uma vez, a seqüência aumentada  $v = \sigma_f \alpha(\alpha, 0)_r \alpha \beta(\beta, 2)_r (\alpha, 1)_r \alpha \in \chi(\sigma_f \alpha \alpha \beta \alpha)$  do Exemplo 3. O evento  $(\alpha, 0)_r$ , o qual representa o recebimento da primeira ocorrência de  $\alpha$  pelo diagnosticador, será substituído pelo evento  $\alpha_{s,0}$ , em que o índice 0 indica que nenhum evento que ocorreu antes do primeiro  $\alpha$  continua sendo transmitido no instante em que o primeiro  $\alpha$  foi recebido. Por sua vez, note que, quando o evento  $(\beta, 2)_r$  ocorre em  $v$ , o qual representa o recebimento de  $\beta$  pelo diagnosticador, há um evento (a segunda ocorrência de  $\alpha$ ) que ocorreu antes de  $\beta$  e ainda não foi recebido pelo diagnosticador. Portanto,  $(\beta, 2)_r$  será substituído pelo evento  $\beta_{s,1}$ , no qual o índice 1 indica a existência de um evento anterior ainda sendo transmitido para o diagnosticador. Por fim, o evento  $(\alpha, 1)_r$ , que representa o recebimento da segunda ocorrência de  $\alpha$  pelo diagnosticador, será substituído pelo evento  $\alpha_{s,0}$ , uma vez que nenhum evento que havia ocorrido antes da segunda ocorrência de  $\alpha$  estava sendo transmitido no instante em que ela foi recebida pelo diagnosticador. Dessa forma, a seqüência  $v$  será mapeada na seqüência  $\sigma_f \alpha \alpha_{s,0} \alpha \beta \beta_{s,1} \alpha_{s,0} \alpha$ .  $\square$

A seguir, a função  $F$  que mapeia uma seqüência de eventos definida sobre  $\Sigma_a$ , na sua correspondente seqüência de eventos definida sobre  $\Sigma_{net} = \Sigma_s \cup \Sigma$ , é apresentada. Para tanto, é necessário primeiramente introduzir algumas definições. A função  $T : (\chi(\Sigma^*) \cap \Sigma_a^* \Sigma_r) \rightarrow \mathbb{N}$ , em que  $T(w\sigma_r)$ , com  $w \in \Sigma_a^*$  e  $\sigma_r \in \Sigma_r$ , retorna a posição em  $w$

do evento  $\sigma \in \Sigma$  correspondente ao evento  $\sigma_r(1)$ , sendo formalmente definida como:

$$T(w\sigma_r) = \max(\{j \in [1, \|w\|] : w^{(j)} = \sigma_r(1) \wedge \sigma_r(2) = \eta(\text{Pre}_j(w))\}).$$

Por exemplo, se  $w = a(a, 0)_r bc(c, 2)_r$  e  $\sigma_r = (b, 2)_r$ , então  $T(w\sigma_r) = 3$ , que é igual à posição em  $w$  do evento  $b$  associado ao último pacote recebido  $\sigma_r = (b, 2)_r$ .

Seja  $w \in \chi(\Sigma^*)$ . A função  $M_w : [1, \|w\|] \rightarrow \mathbb{N}$ , em que  $M_w(l)$  é igual ao número de eventos observáveis que ocorreram antes do  $l$ -ésimo evento de  $w$  e já foram observados em  $w$ , é definida como se segue:

$$M_w(l) = |\{j \in [1, l-1] : w^{(j)} \in \Sigma_o \wedge (\exists i \in [j, \|w\|]) [w^{(i)} \in \Sigma_r \wedge w^{(i)}(1) = w^{(j)} \wedge w^{(i)}(2) = \eta(\text{Pre}_j(w))]\}|.$$

Note que, para uma seqüência  $w\sigma_r \in \chi(\Sigma^*)$ , em que  $\sigma_r = (\sigma, n)_r$ , o  $T(w\sigma_r)$ -ésimo evento em  $w$  divide os eventos pertencentes a  $\Sigma_o$  dessa seqüência em dois grupos: aqueles que ocorreram antes e aqueles que ocorreram após o envio do pacote  $(\sigma, n)$ . Então, definindo-se  $l = T(w\sigma_r)$ , tem-se que  $M_w(l)$  é igual ao número de eventos observáveis que ocorreram em  $w$  antes do envio de  $(\sigma, n)$  e já foram observados antes de  $(\sigma, n)_r$ , isto é, antes da chegada do pacote  $(\sigma, n)$  no diagnosticador. Por exemplo, se  $\Sigma_o = \{a, b, c, d\}$ ,  $D = 3$ ,  $w = a(a, 0)_r bc(c, 2)_r d$  e  $\sigma_r = (d, 3)_r$ , então  $T(w\sigma_r) = 6$ , e  $M_w(6) = 2$ , que é igual ao número de eventos observáveis (eventos  $a$  e  $c$ ) que ocorreram antes do sexto evento em  $w$  (evento  $d$ ) e já foram observados com sucesso em  $w$ . Note que, embora o evento observável  $b$  tenha ocorrido antes de  $d$  em  $w$ , ele não foi observado em  $w$ , isto é, o pacote  $(b, 1)$  ainda está sendo transmitido.

Define-se, agora, a função  $U$  que mapeia um evento  $\sigma_r = (\sigma, n)_r \in \Sigma_r$  em um evento  $\sigma_{s,j} \in \Sigma_s$ , em que  $\sigma$  é o evento observável transmitido no pacote  $(\sigma, n)$ , e  $j$  é o número de pacotes enviados antes de  $(\sigma, n)$  que ainda estão sendo transmitidos. A função  $U : (\chi(\Sigma^*) \cap \Sigma_a^* \Sigma_r) \rightarrow \Sigma_s$  é o mapeamento definido, para  $w \in \Sigma_a^*$  e  $\sigma_r \in \Sigma_r$ , como  $U(w\sigma_r) = \sigma_{s,j}$ , em que:

$$\sigma = \sigma_r(1), \\ j = \|P_{a,o}(\text{Pre}_l(w))\| - M_w(l) - 1, \text{ com } l = T(w\sigma_r).$$

*Definição 4.* A função de substituição de pacotes recebidos é o mapeamento  $F : \chi(\Sigma^*) \rightarrow \Sigma_{net}^*$ , definido recursivamente a seguir:

- (i)  $F(\varepsilon) = \varepsilon$ ;
- (ii) Para  $w \in \Sigma_a^*$  e  $\sigma \in \Sigma_a$ ,

$$F(w\sigma) = \begin{cases} F(w)\sigma, & \text{if } \sigma \in \Sigma, \\ F(w)U(w\sigma), & \text{if } \sigma \in \Sigma_r. \end{cases}$$

A extensão de  $F$  para o domínio  $2\chi(\Sigma^*)$  é definida como  $F(L_a) := \bigcup_{w \in L_a} F(w)$ .  $\square$

Note que a função  $F$  mapeia cada seqüência aumentada em  $\chi(\Sigma^*)$  numa seqüência na qual as observações de eventos pelo diagnosticador são modeladas por eventos pertencentes a  $\Sigma_s$ . Dessa forma, a projeção  $P_{net,s} : \Sigma_{net}^* \rightarrow \Sigma_s^*$  mapeia seqüências em  $\Sigma_{net}^*$  nas seqüências observadas do ponto de vista do diagnosticador.

*Exemplo 7.* Considere novamente o sistema ilustrado na Figura 2, em que o atraso de observação máximo é

$D = 1$  passo, e suponha  $N = 4$ . Para ilustrar como a função  $F$  funciona, considere a sequência aumentada  $w = \sigma_f \alpha(\alpha, 0)_r \alpha \beta(\beta, 2)_r (\alpha, 1)_r \in \chi(\sigma_f \alpha \alpha \beta)$ . Conforme a Definição 4:

$$\begin{aligned} F(w) &= \sigma_f \alpha(\alpha, 0)_r \alpha \beta(\beta, 2)_r U(\sigma_f \alpha(\alpha, 0)_r \alpha \beta(\beta, 2)_r (\alpha, 1)_r) \\ &= \sigma_f \alpha(\alpha, 0)_r \alpha \beta U(\sigma_f \alpha(\alpha, 0)_r \alpha \beta(\beta, 2)_r) \alpha_{s,0} \\ &= \sigma_f \alpha U(\sigma_f \alpha(\alpha, 0)_r) \alpha \beta \beta_{s,1} \alpha_{s,0} \\ &= \sigma_f \alpha \alpha_{s,0} \alpha \beta \beta_{s,1} \alpha_{s,0}. \end{aligned}$$

### 5.3 Autômato da rede de comunicação

O Algoritmo 1 fornece as etapas para a construção do autômato  $G_D$  que modela a rede de comunicação com atraso máximo igual a  $D$ , de forma que  $L(G_D) = F(\chi(\Sigma^*))$ . Os estados de  $G_D$  são rotulados por sequências de eventos pertencentes a  $\Sigma_{ov}^*$ , em que  $\Sigma_{ov} = \Sigma_o \cup \{\nu\}$ , sendo os eventos pertencentes a  $\Sigma_o$  representam ocorrências que estão sendo transmitidas para o diagnosticador e o novo símbolo  $\nu$  é utilizado das duas formas a seguir: (i) no estado inicial de  $G_D$ , para representar a ausência de eventos sendo transmitidos, e (ii) em estados rotulados com sequências  $q = \dots \sigma_1 \eta \sigma_2 \dots$  em que  $\eta \in \{\nu\}^*$  e  $\sigma_1, \sigma_2 \in \Sigma_o$ , para indicar que o número de eventos que ocorreram na planta entre as ocorrências de  $\sigma_1$  e  $\sigma_2$  é igual a  $\|\eta\|$ .

Duas funções sobre o conjunto  $\Sigma_{ov}^*$  são utilizadas no Algoritmo 1 (Nunes et al., 2018): a função de substituição  $rep$ , em que  $rep(q, i, \sigma)$  retorna a sequência obtida a partir da substituição do  $i$ -ésimo evento da sequência  $q$  pelo evento  $\sigma$ , e a função de corte ( $cut$ ), que retorna o maior sufixo de uma sequência  $q$  cujo primeiro evento é diferente de  $\nu$ , se  $q$  tiver pelo menos um evento diferente de  $\nu$ , ou retornar  $\nu$  caso contrário. As funções  $rep$  e  $cut$  são formalmente definidas a seguir.

*Definição 5.* (Funções  $rep$  e  $cut$ ).

- A função de substituição  $rep : \Sigma_{ov}^* \times \mathbb{N} \times \Sigma_{ov} \rightarrow \Sigma_{ov}^*$  é o mapeamento definido,  $\forall (q, i, \sigma) \in \Sigma_{ov}^* \times \mathbb{N} \times \Sigma_{ov}$  com  $q = q_1 q_2 \dots q_\ell$ , como:

$$rep(q, i, \sigma) = \begin{cases} q_1 q_2 \dots q_{i-1} \sigma q_{i+1} \dots q_\ell, & \text{if } i \leq \ell \\ \text{não definido, caso contrário.} & \end{cases}$$

- A função de corte  $cut : \Sigma_{ov}^* \rightarrow \Sigma_{ov}^*$  é o mapeamento definido, para todo  $q = q_1 q_2 \dots q_\ell \in \Sigma_{ov}^*$ , como:

$$cut(q) = \begin{cases} q_i q_{i+1} \dots q_\ell, & \text{if } (\exists i \leq \ell) [q_i \neq \nu \wedge Pre_i(q) \in \{\nu\}^*] \\ \nu, & \text{if } q_k = \nu, \forall k \in \{1, 2, \dots, \ell\}. \end{cases}$$

*Algoritmo 1.* (Construção do autômato  $G_D$ ).

*Entradas:*  $\Sigma, \Sigma_o, D$  e  $N$ .

*Saída:*  $G_D = (X_d, \Sigma_{net}, f_d, x_{0_d})$ .

1. Defina  $x_{0_d} = \nu$  e  $X_d = \emptyset$
2. Construa  $\Sigma_s$  (Eq. (1)) e  $\Sigma_{net} = \Sigma \cup \Sigma_s$
3. Defina uma fila  $Q = [x_{0_d}]$
4. Enquanto  $Q \neq \emptyset$  faça:
  - 4.1.  $q = head[Q]$
  - 4.2.  $\ell = \|q\|$

4.3. Calcule  $I_o = \{y \in [1, \ell] : q^{(y)} \in \Sigma_o\}$

4.4. Se  $\ell \leq D$ , então:

(a) Para cada  $\sigma \in \Sigma_o$ , defina  $\tilde{q} = f_d(q, \sigma) = cut(q\sigma)$ , e  $Enqueue(Q, \tilde{q})$

(b) Para cada  $\sigma \in \Sigma_{uo}$ , defina  $\tilde{q} = f_d(q, \sigma) = cut(q\nu)$ , e, se  $\tilde{q} \notin Q$ , então  $Enqueue(Q, \tilde{q})$

4.5. Para cada  $i \in I_o$ :

(a) Calcule  $j = |\{y \in I_o : y < i\}|$

(b) Faça  $\sigma = q^{(i)}$

(c) Defina  $\tilde{q} = f_d(q, \sigma_{s,j}) = cut(rep(q, i, \nu))$ , e, se  $(\tilde{q} \notin X_d) \wedge (\tilde{q} \notin Q)$ , então  $Enqueue(Q, \tilde{q})$

4.6.  $X_d \leftarrow X_d \cup \{q\}$

4.7.  $Dequeue(Q)$

*Exemplo 8.* Considere o sistema de diagnose de falhas em rede da Figura 2, em que  $\Sigma = \{\alpha, \beta, \mu\}$ ,  $\Sigma_o = \{\alpha, \beta\}$  e  $D = 1$ . O autômato  $G_D$  obtido de acordo com o Algoritmo 1 é mostrado na Figura 3.

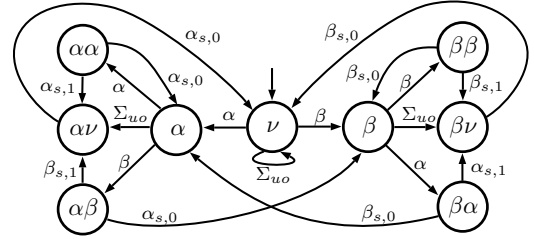


Figura 3. Autômato que modela a rede de comunicação calculado no Exemplo 8.

Pode-se observar que o estado  $q = \alpha\beta$  é alcançado quando  $\alpha$  e  $\beta$  ocorreram na planta, nesta ordem, e ambos ainda estão sendo transmitidos ao diagnosticador. Por outro lado, o estado  $q' = \alpha\nu$  corresponde ao caso em que a ocorrência de  $\alpha$  foi seguida pela ocorrência de outro evento na planta, mas apenas a ocorrência de  $\alpha$  está sendo transmitida, seja porque a outra ocorrência do evento não é observável ou porque ela já foi transmitida com sucesso.

Por fim, as transições de  $G_D$  rotuladas pelos eventos de  $\Sigma_s = \{\alpha_{s,0}, \beta_{s,0}, \alpha_{s,1}, \beta_{s,1}\}$  representam as observações bem-sucedidas de eventos da planta pelo diagnosticador. Por exemplo, a transição a partir do estado  $q = \alpha\beta$ , rotulada por  $\alpha_{s,0}$  (resp.  $\beta_{s,1}$ ), representa a observação de  $\alpha$  (resp.  $\beta$ ) enquanto nenhum (resp. um) dos eventos observáveis que ocorreram antes está sendo transmitido.  $\square$

*Lema 1.*  $L(G_D) = F(\chi(\Sigma^*))$ .

As demonstrações do Lema 1 e dos demais resultados foram omitidas em decorrência da limitação de espaço.

Seja  $P_{net} : \Sigma_{net}^* \rightarrow \Sigma^*$ . Então, os resultados a seguir podem ser apresentados.

*Teorema 1.* Seja  $L \subseteq \Sigma^*$ . Então,  $F(\chi(L)) = P_{net}^{-1}(L) \cap L(G_D)$ .

*Corolário 1.* Seja  $G_{net} = G \parallel G_D$ .  $L(G_{net}) = F(\chi(L(G)))$ .

De acordo com o Corolário 1, um autômato  $G_{net}$ , que modela um sistema de diagnose de falhas em rede (Figura 1), pode ser obtido em duas etapas: (i) construção do autômato  $G_D$  (Algoritmo 1) e (ii) cálculo de  $G_{net} = G \parallel G_D$ .

Vale ressaltar que, do ponto de vista do diagnosticador, o conjunto de eventos observáveis de  $G_{net}$  é  $\Sigma_{net,o} = \Sigma_s$ .

#### 5.4 Diagnóstico de falhas em SEDR

Uma vez que a linguagem do sistema é diagnosticável e os atrasos de observação são limitados (Hipóteses A1 e A2), pode-se inferir que, com a transmissão dos *sequence numbers*, torna-se sempre possível identificar a ocorrência de um evento de falha num número finito de passos, sendo o diagnosticador em rede projetado da forma a seguir. Constrói-se, inicialmente, um autômato rotulador  $A_\ell = (X_{A_\ell}, \{\sigma_f\}, f_{A_\ell}, \Gamma_{A_\ell}, x_{0,A_\ell})$ , em que  $X_{A_\ell} = \{N, F\}$ ,  $f_{A_\ell}(N, \sigma_f) = f_{A_\ell}(F, \sigma_f) = F$  e  $x_{0,A_\ell} = N$ . Em seguida, define-se  $G_{net,\ell} = G_{net} || A_\ell$ . Por fim, Calcula-se o autômato observador  $G_{net,s} = Obs(G_{net,\ell}, \Sigma_s)$ . Um estado de  $G_{net,s}$  é dito ser certo de falha se ele contiver apenas estados de  $G_{net,\ell}$  rotulados com  $F$ . A ocorrência de uma falha é diagnosticada assim que  $G_{net,s}$  alcançar um estado certo de falha, conforme ilustrado no exemplo a seguir.

*Exemplo 9.* Considere o sistema de diagnose de falhas em rede da Figura 2, em que  $\Sigma = \{\alpha, \beta, \mu\}$ ,  $\Sigma_o = \{\alpha, \beta\}$  e  $D = 1$ . O autômato  $G_{net,s}$  é mostrado na Figura 4.

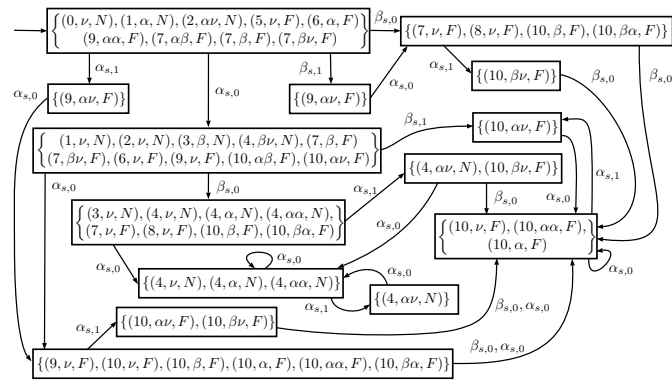


Figura 4. Autômato  $G_{net,s}$  calculado no Exemplo 9.

Considere que a planta executou a sequência de falha  $s_f = \sigma_f \alpha \alpha \beta \alpha^n$  e a segunda ocorrência de  $\alpha$  foi observada após a observação de  $\beta$ , o que é equivalente a considerar que a sequência  $w_f = \sigma_f \alpha \alpha_{s,0} \alpha \beta \beta_{s,1} \alpha_{s,0} (\alpha_{s,0})^n \in F(\chi(s_f))$  ocorreu, causando a observação de  $P_{net,s}(w_f) = \alpha_{s,0} \beta_{s,1} (\alpha_{s,0})^{n+1}$ . Pode-se observar, por meio da Figura 4, que a ocorrência da falha é detectada logo após a observação da sequência  $\alpha_{s,0} \beta_{s,1}$ , uma vez que, após essa sequência, o autômato observador  $G_{net,s}$  alcança o estado  $\{(10, \alpha \nu, F)\}$ , o qual é um estado certo de falha. Vale ressaltar que, como mostrado no Exemplo 1, a detecção da falha não seria possível nesse caso, sem a rotulação dos eventos a partir dos *sequence numbers*.  $\square$

## 6. CONCLUSÃO

Neste trabalho é proposta uma modelagem de sistemas a eventos discretos com redes de comunicação sujeitas a atrasos. Na abordagem proposta, as observações de eventos são rotuladas em função dos seus respectivos *sequence numbers*, permitindo o diagnóstico de falhas mesmo na presença de atrasos limitados de comunicação. Em trabalhos futuros será estudado o uso da modelagem

considerada neste trabalho para síntese de controladores supervisórios sujeitos a atrasos de observação.

## REFERÊNCIAS

- Alves, M.V.S., Carvalho, L.K., e Basilio, J.C. (2021a). Supervisory control of networked discrete event systems with timing structure. *IEEE Transactions on Automatic Control*, 66(5), 2206–2218.
- Alves, M.V.S., da Cunha, A.E.C., Carvalho, L.K., Moreira, M.V., e Basilio, J.C. (2021b). Robust supervisory control of discrete event systems against intermittent loss of observations. *International Journal of Control*, 94(7), 2008–2020.
- Carvalho, L.K., Basilio, J.C., e Moreira, M.V. (2012). Robust diagnosis of discrete-event systems against intermittent loss of observations. *Automatica*, 48(9), 2068–2078.
- Carvalho, L.K., Moreira, M.V., Basilio, J.C., e Lafortune, S. (2013). Robust diagnosis of discrete-event systems against permanent loss of observations. *Automatica*, 49(1), 223–231.
- Cassandras, C.G. e Lafortune, S. (2008). *Introduction to discrete event systems*. Springer, 2nd edition edition.
- Debouk, R., Lafortune, S., e Teneketzis, D. (2003). On the effect of communication delays in failure diagnosis of decentralized discrete event systems. *Discrete Event Dynamic Systems*, (13), 263–289.
- Huo, Z., Fang, H., e Ma, C. (2004). Networked control system: State of the art. In *Proceedings of the 5th World Congress on Intelligent Control and Automation*, 1319–1322. Hangzhou, P.R. China.
- Li, X., Li, D., Wan, J., Vasilakos, A.V., Lai, C.F., e Wang, S. (2017). A review of industrial wireless networks in the context of industry 4.0. *Wireless Networks*, 23–41.
- Liu, Z., Yin, X., Shu, S., Lin, F., e Li, S. (2022). Online supervisory control of networked discrete event systems with control delays. *IEEE Transactions on Automatic Control*, 67(5), 2314–2329.
- Nunes, C.E.V., Moreira, M.V., Alves, M.V.S., Carvalho, L.K., e Basilio, J.C. (2018). Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation. *Discrete Event Dynamic Systems*, 28(2), 215–246.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., e Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9), 1555–1575.
- Takai, S. (2021). A general framework for diagnosis of discrete event systems subject to sensor failures. *Automatica*, 129, 109669.
- Tong, Y. e Ma, Z. (2022). Verification of  $k$ -step and definite critical observability in discrete-event systems. *IEEE Transactions on Automatic Control*, 1–8. doi:10.1109/TAC.2022.3202983.
- Tripakis, S. (2004). Decentralized control of discrete-event systems with bounded or unbounded delay communication. *IEEE Transactions on Automatic Control*, 49(9), 1489–1501.
- Viana, G.S., Alves, M.V.S., e Basilio, J.C. (2022). Codiagnosability of networked discrete event systems with timing structure. *IEEE Transactions on Automatic Control*, 67(8), 3933–3948.