

ANÁLISE LÓGICA DO PROTOCOLO LMP EM UM AMBIENTE GMPLS

Daniela Cunha. Graça Bressan

Resumo – Alcançabilidade justa é um método de análise lógica útil na detecção de erros de progresso em protocolos de comunicação representados por MEFCs. É imprescindível a análise lógica do LMP que é um dos protocolos que constituem o GMPLS. Assim, verifica-se o comportamento deste bem como se o mesmo está ou não livre de erros de progresso.

Palavras Chave – LMP, GMPLS, análise lógica, alcançabilidade justa.

Abstract – Fair Reachability is a logical analysis method useful in detecting progress errors in communication protocols represented by CFMS. It is important the logical analysis of LMP which is one of the protocols that compose GMPLS. Thus, it is verified the protocol behavior as well as whether it is free from progress errors.

Index Terms – LMP, GMPLS, logical analysis, fair reachability.

I. INTRODUÇÃO

O GMPLS (*Generalized Multiprotocol Label Switching*), avanço lógico do MPLS para redes ópticas, é um conjunto de protocolos do plano de controle que fornece uma semântica consistente e uniforme para sinalização, roteamento e gerenciamento de enlace. O GMPLS está sendo desenvolvido com o objetivo de ser aplicável a todo tipo de tráfego de transporte e serviço. O GMPLS estende a arquitetura MPLS permitindo o interfaceamento com tecnologias que não são apenas baseadas na comutação de pacotes, mas também comutação no domínio do tempo, comprimento de onda e fibra [BERGER, 2003].

Protocolos GMPLS podem ser modelados por uma rede formada por duas máquinas de estado finitas de comunicação (MEFC) que se comunicam de forma assíncrona através do envio e recebimento de mensagens.

MEFCs são utilizadas para especificar e validar protocolos de comunicação através da análise de alcançabilidade. Esta técnica provou ser uma das mais efetivas na verificação da correção dos protocolos de comunicação baseada no modelo de transição de estados. Porém, sabe-se que a explosão de estados é a mais severa limitação da aplicabilidade desta técnica. Foram pesquisadas diferentes estratégias de reparação da análise de alcançabilidade para o problema de explosão de estados. Dentre as abordagens estudadas, a exploração de estados com progresso justo foi a abordagem escolhida para a análise do protocolo de gerenciamento de enlace (LMP - *Link Management Protocol*) devido às suas vantajosas características.

Laboratório de Arquitetura e Redes de Computadores - USP
Email: {dcunha,gbressan}@larc.usp.br

II. GMPLS – VISÃO GERAL

GMPLS é uma generalização da arquitetura MPLS, e pode se diferenciar um pouco desta arquitetura devido ao fato do GMPLS considerar também planos de dados não baseados em pacotes. O GMPLS suporta as interfaces e elementos de comutação: PSC (*Packet Switching Capable*), TSC (TDM – *Time Division Multiplexing - Switching Capable*), LSC (*Lambda Switching Capable*) e FSC (*Fiber Switching Capable*) [BERGER, 2003].

Algumas características importantes do MPLS Generalizado são [PAPADIMITRIOU, 2003]: (a) Descobrir os vizinhos de um nó; (b) Distribuir informações sobre os enlaces; (c) Fornecer topologia da rede; (d) Gerenciar caminhos; (e) Proteger/restaurar enlaces. Além disso, o GMPLS permite um controle centralizado, provisionamento automático, balanceamento de carga, largura de banda sob demanda e OVPN (*Optical Virtual Private Network*).

O GMPLS estendeu o paradigma *label switching* de tecnologias de comutação de pacotes/células/frames para tecnologias orientadas à conexão. Assim, ele inclui os dispositivos capazes de realizar: (a) Comutação de pacotes; (b) Multiplexação por divisão de tempo; (c) Comutação de lambda (ou comprimento de onda); (d) Comutação *waveband*; (e) Comutação de fibras.

III. CARACTERÍSTICAS DO LMP

O protocolo de gerenciamento de enlace (LMP - *Link Management Protocol*), está sendo desenvolvido como parte da suíte de protocolos GMPLS para gerenciar enlaces TE e permitir que elementos de rede se comuniquem de forma adequada. Este protocolo permite que os nós descubram a identidade e informação detalhada de interfaces dos nós vizinhos e mantém esta informação atualizada assim que configurações ou propriedades dos enlaces TE são modificadas [MANNIE, 2003], [ABOUL-MAGD, 2003], [LANG, 2003].

O LMP consiste de quatro funções suportadas pelos controladores de canal de controle e enlace TE; as duas primeiras funções são obrigatórias e as demais são opcionais:

- Gerenciamento de canal de controle – usado para estabelecer, configurar e manter a conectividade entre nós adjacentes através da topologia de rede de controle. Esta operação é realizada utilizando a troca de mensagens de configuração de canal de controle, seguido por uma troca de mensagens do tipo *keep-alive*.
- Correlação de propriedade de enlace – usado para agregar múltiplos enlaces de dados em um único enlace TE e sincronizar as propriedades local e remota do mesmo.

- Verificação de enlace - usada para verificar a conectividade física dos enlaces de dados e o mapeamento entre os identificadores de interface local e remota desses enlaces.
- Gerenciamento de falhas – usado para suprimir alarmes e localizar falhas de enlace em redes opacas e transparentes.

IV. ANÁLISE LÓGICA DE PROTOCOLOS DE COMUNICAÇÃO

Muitos protocolos de comunicação podem ser modelados como uma rede de duas máquinas de estado finitas de comunicação (MEFCs) que consistem em um conjunto de MEFs que trocam mensagens sobre canais FIFO (*First-in First-out*) e unidirecionais. É útil e necessário validar estes modelos de protocolos mostrando que a comunicação dos mesmos satisfaz certas propriedades desejadas e, verificando se a rede está livre de erros de progresso [GOUDA, 1984].

Existem cinco tipos de erros de progresso que um protocolo pode apresentar. Eles são definidos como:

- *Deadlock* - os canais estão vazios e nenhuma transição é possível.
- *Recepção não-especificada* - recepção que pode ser executada, mas a mesma não é especificada no projeto do protocolo.
- *Interação não-executada* - transmissão ou recepção que é especificada no projeto do protocolo, mas nunca é executada.
- *Livelock* - situação onde os processos mantêm a troca de mensagens, mas nenhum progresso efetivo é realizado.
- *Overflow* - estado de canal tal que o número de mensagens no canal não é limitado por um número inteiro positivo pré-definido.

A técnica mais direta para validar uma dada rede de duas MEFCs é chamada exploração de estados e, o método mais comum de análise utilizado por esta técnica é chamado de análise de alcançabilidade exaustiva. O resultado da aplicação deste método é a construção de um grafo de alcançabilidade formado por estados globais. Um estado global, também denominado estado alcançável, é uma tupla contendo o estado corrente de cada máquina e o conteúdo de todos os canais utilizados no sistema [GOUDA, 1984]. Se o grafo de alcançabilidade for finito, é possível determinar se a rede de MEFCs está livre dos erros de progresso através da análise de cada estado global novo que é gerado.

Este método tem a vantagem da simplicidade. No entanto, ele possui uma desvantagem denominada explosão de espaço (o número total de estados globais explorados cresce exponencialmente em função do tamanho da rede). O problema existe pois o grafo de alcançabilidade exibe o progresso de suas MEFCs permitindo todas as velocidades de progresso relativas. [RUBIN, 1982] sugeriu um método de validação de protocolos melhorada que utiliza um tipo especial de grafo de alcançabilidade chamado grafo de alcançabilidade justa. Este grafo é construído forçando as duas máquinas a avançarem na mesma velocidade.

Para redes de MEFCs com comunicação limitada, o método de alcançabilidade justa, comparado com a alcançabilidade

exaustiva convencional, pode reduzir o número total de estados globais que devem ser buscados para verificar erros de progresso. Esta é uma melhora significativa e tem sido utilizada para reparar o problema de explosão de estado.

V. REDES DE MEFCs

Considere $I = \{1, 2, \dots, n\}$, onde $n \geq 2$, é uma constante denotando o número de processos na rede de MEFCs. Uma rede de MEFCs é uma tupla $\langle P_1, P_2, \dots, P_n \rangle$, onde cada P_i é uma MEFC [GOUDA, 1986]. Uma máquina de estado finita de comunicação P_i é uma tupla $(S_i, \sum_i^\pm, \delta_i, p_{0i})$, onde [GOUDA, 1984] [BRAND, 1983]:

- S_i é o conjunto de estados locais da máquina P_i
- $p_{0i} \in S_i$ é o estado inicial da máquina P_i
- $\sum_i^\pm = \sum_{1 \leq j \leq n} \sum_{i,j} \cup \sum_{1 \leq j \leq n} \sum_{j,i}$, onde $\sum_{i,j}$, $1 \leq j, i \leq n$, é o alfabeto de mensagens que P_i pode enviar para P_j , e $\sum_{j,i}$, $1 \leq j, i \leq n$, é o alfabeto de mensagens que P_i pode receber de P_j .
- $\delta_i : S_i \times \sum_i^\pm \times I \rightarrow 2^{S_i}$ é a função de transição. $\delta_i(p, -m, j)$ é o conjunto de estados que o processo P_i pode se mover a partir do estado p depois de enviar uma mensagem m para o processo P_j , $\delta_i(p, +m, j)$ é o conjunto de estados que o processo P_i pode se mover a partir do estado p depois de receber uma mensagem m enviada pelo processo P_j .

Um estado global é uma tupla $[\bar{v}, \bar{c}]$, onde $\bar{v} = \langle p_i \rangle_{i \in I}$ são os nós dos processos (MEFCs) envolvidos na comunicação e $\bar{c} = \langle c_{i,j} \rangle_{i,j \in I}$ são os canais para troca de mensagens entre os

processos. O estado global inicial é $[\langle p_{0i} \rangle_{i \in I}, \langle c_{i,j} \rangle_{i,j \in I}]$, onde $c_{i,j} = \mathcal{E}$ ($i \neq j$). Um estado global $[\langle p'_k \rangle_{k \in I}, \langle c'_{k,l} \rangle_{k,l \in I}]$ é subsequente de um estado global $[\langle p_k \rangle_{k \in I}, \langle c_{k,l} \rangle_{k,l \in I}]$, se $\exists k, l \in I$ tal que:

- ou $p_k \xrightarrow{-(m,l)} p'_k$ está em P_k , $\forall k \in I$ e $c'_{k,l} = c_{k,l} \cdot m \quad \forall k, l \in I$
- ou $p_k \xrightarrow{+(m,l)} p'_k$ está em P_k , $\forall k \in I$ e $m \cdot c'_{l,k} = c_{l,k} \quad \forall k, l \in I$

No primeiro caso, a máquina P_k , no estado local p_k , envia uma mensagem m para a máquina P_l e então se move para o estado p'_k . Logo somente o canal c_{kl} é alterado. No segundo caso, a máquina P_k , no estado local p_k , recebe a mensagem m do canal c_{lk} , e então move para outro estado p'_k .

A tarefa de verificar uma rede de MEFCs que modela um protocolo de comunicação consiste em determinar se a rede de MEFCs possui algumas propriedades não desejáveis que refletem em alguns erros lógicos no protocolo original.

VI. ANÁLISE DE ALCANÇABILIDADE JUSTA

A análise de alcançabilidade justa, ou alcançabilidade com progresso justo [GOUDA, 1986], é um método de alcançabilidade melhorada se comparada com a alcançabilidade exaustiva convencional. Este método é

utilizado em redes de comunicação com somente dois processos (MEFCs) e, parte do princípio de que os dois processos avançam em igual velocidade.

Um estado $[\bar{v}, \bar{c}] = [t, w, c_{tw}, c_{wt}]$ da rede $[M, N]$ é justo se e somente se $lc_{w|w} = lc_{w|t}$, onde $lc_{w|w}$ é o número de mensagens no canal de mensagens c_{tw} , onde t e w são estados pertencentes as MEFCs M e N , respectivamente.

Um estado global $[\bar{v}', \bar{c}']$ é alcançável, ou subsequente, de forma justa de $[\bar{v}, \bar{c}]$ se [PENG, 1997]:

- $[\bar{v}', \bar{c}']$ é resultado de $[\bar{v}, \bar{c}]$ via duas ações de duas máquinas diferentes, P_i e P_j , tal que a ação de P_i seja uma ação ativa (de envio) para P_j , e a ação de P_j seja uma ação passiva (de recebimento) de P_i ; ou
- $[\bar{v}', \bar{c}']$ é resultado de $[\bar{v}, \bar{c}]$ via duas ações de duas máquinas diferentes, P_i e P_j , tal que a ação de P_j seja uma ação ativa, e a ação de P_i seja uma ação passiva; ou
- $[\bar{v}', \bar{c}']$ é resultado de $[\bar{v}, \bar{c}]$ via duas ações de duas máquinas diferentes, P_i e P_j , tal que as ações de P_i e de P_j sejam ativas; ou
- $[\bar{v}', \bar{c}']$ é resultado de $[\bar{v}, \bar{c}]$ via duas ações de duas máquinas diferentes, P_i e P_j , tal que as ações de P_i e de P_j sejam passivas.

A. Grafo de Alcançabilidade Justa

O resultado final da análise de alcançabilidade justa é a construção do grafo de alcançabilidade justa. O número total de estados alcançáveis de forma justa é menor ou igual a metade do número total de estados globais alcançáveis.

Os estados alcançáveis de forma justa de uma rede $[M, N]$, e a relação de ser “subsequente de forma justa” podem ser representados pelo grafo de alcançabilidade justa, rotulado G , de $[M, N]$, como segue:

- O estado inicial do grafo de alcançabilidade justa é $[\bar{v}_0, \bar{c}_0] = [t_0, w_0, \varepsilon, \varepsilon]$, onde t_0 e w_0 são os estados iniciais das MEFCs M e N , respectivamente e, ε representa uma *string* vazia indicando que ambos canais unidirecionais não transportam mensagens.
- Para cada estado $[\bar{v}, \bar{c}]$ de $[M, N]$ alcançável de forma justa a partir do estado inicial ou de outros estados já presentes em G , adiciona-se um vértice rotulado com o estado $[\bar{v}, \bar{c}]$, para G (referido como vértice $[\bar{v}, \bar{c}]$).
- Se dois estados alcançáveis de forma justa $[\bar{v}', \bar{c}']$ e $[\bar{v}, \bar{c}]$ de $[M, N]$ são de tal forma que $[\bar{v}', \bar{c}']$ é, de forma justa, subsequente de $[\bar{v}, \bar{c}]$ sobre um arco e representando duas ações de transição de duas máquinas diferentes M e N , então adicione este arco, rotulado e , do vértice $[\bar{v}, \bar{c}]$ para o vértice $[\bar{v}', \bar{c}']$ em G .

No grafo de alcançabilidade justa:

- A comunicação de $[M, N]$ é livre de *deadlock* se e somente se nenhum vértice em G for rotulado como um estado de *deadlock*;
- A comunicação de $[M, N]$ é livre de recepções não-especificadas se e somente se (i) nenhum vértice em G for

rotulado como um estado de recepção não-especificada, e (ii) nenhum vértice em G for rotulado como um estado na forma $[t, w, \varepsilon, \varepsilon]$ que é precedente de um estado de recepção não-especificado;

- A comunicação de $[M, N]$ é livre de interações não-executadas se e somente se todas as transições definidas nas MEFCs M e N estiverem presente em G ;
- A comunicação de $[M, N]$ é livre de *livelocks* se e somente se G não possuir nenhum *loop* infinito em um pequeno conjunto de estados globais;
- A comunicação de $[M, N]$ é livre de *overflow* se e somente se nenhum canal em G exceder o tamanho máximo pré-definido para os mesmos.

VII. ANÁLISE LÓGICA DO LMP

Todos os procedimentos LMP são detalhados através de suas máquinas de estado correspondentes (a seguir).

A. MEF de Canal de Controle

As MEFCs de CC (canal de controle) (Fig. 1) definem os estados e lógicas de operação de um CC LMP.

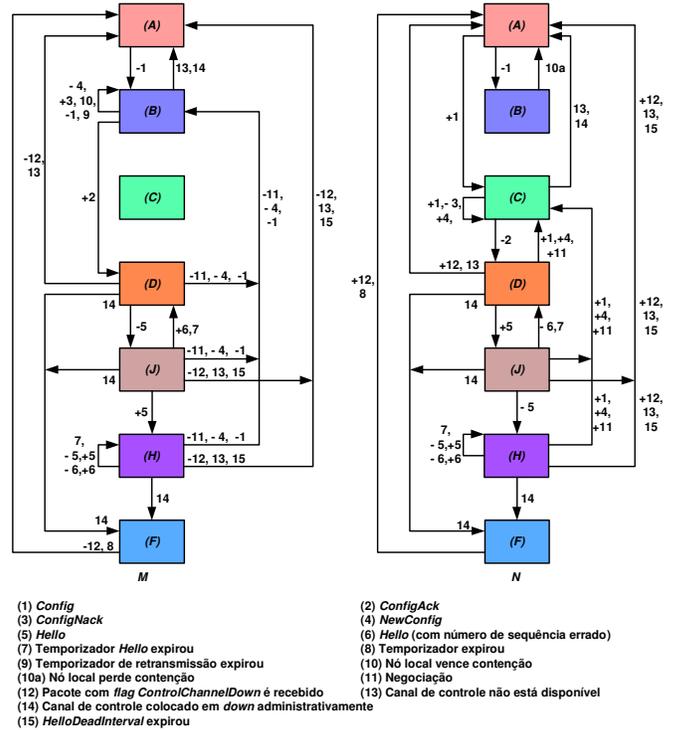


Fig. 1. Rede de MEFCs de canal de controle.

Os estados que constituem as MEFCs são definidos como:

- (A) – Estado Inicial
O CC não está em estado operacional, nenhuma mensagem é enviada e, os parâmetros do CC recebem valores iniciais.
- (B) – Estado de Negociação de Parâmetros (envio)
A mensagem *Config* é enviada periodicamente e espera-se por uma mensagem *ConfigAck* ou *ConfigNack*. Há mudança para o estado (D) somente se receber um reconhecimento positivo dos parâmetros.
- (C) – Estado de Negociação de Parâmetros (recebimento)

Nó espera por parâmetros de configuração aceitáveis oriundos do lado remoto. Assim que os parâmetros corretos são recebidos, há mudança da MEFC para o estado (D).

▪ (D) – Estado Ativo

O nó envia mensagens Hello periodicamente e muda para o estado (J).

▪ (J) – Estado de Espera

Somente depois de enviar uma mensagem Hello válida (estado D) e receber uma mensagem Hello válida (estado J) é que há mudança para o estado (H).

▪ (H) – Estado Operacional

Nó envia e recebe mensagens Hello válidas.

▪ (F) – Estado de Liberação

CC está sendo liberado. Uma ação administrativa faz com que o CC vá para este estado e o bit CCDown é marcado em todas as mensagens.

Com base na rede de MEFCs da Fig. 1, é possível aplicar o método de análise lógica e verificar se a parte do protocolo LMP relacionado com o CC possui ou não erros de progresso. O resultado desta análise é a construção do grafo de alcançabilidade justa, como mostra a Fig. 2. As mensagens e seus números correspondentes utilizados no grafo são os mesmos usados na rede de MEFCs, apresentada na Fig. 1.

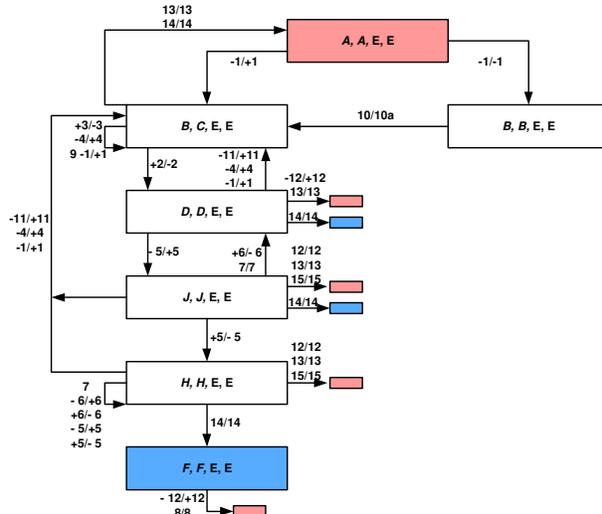


Fig. 2. Grafo de Alcançabilidade Justa resultante da rede de MEFCs do LMP relacionado ao canal de controle.

De acordo com a grafo e com base nas definições de erros de progresso, nota-se que o LMP relacionado ao CC não apresenta erros de progresso.

B. MEF de Enlace TE

A MEF de enlace TE (traffic engineering) (Fig. 3) define os estados e lógicas de operação do enlace TE. A Fig. 3 ilustra as operações de um MEF de enlace TE na forma de um diagrama de transição de estados. A listagem dos eventos utilizados na comunicação é apresentada na base da Fig. 3.

Os estados que compõem as MEFCs, são definidos como:

▪ (A) – Estado Inicial

Não existem enlaces de dados alocados ao enlace TE.

▪ (B) – Estado Alloc/SemCC

Enlaces de dados foram alocados ao enlace TE, mas não há CC operando.

▪ (C) – Estado Alloc/CC

Enlaces de dados foram alocados ao enlace TE e existe pelo menos um CC operando.

▪ (D) – Estado SemSinc

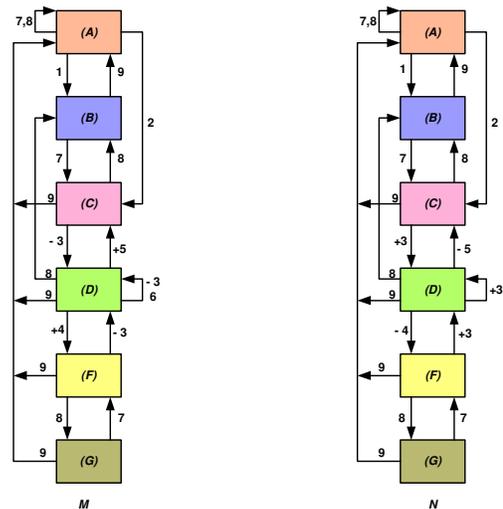
Enlaces de dados foram alocados ao enlace TE, mas a configuração ainda não foi sincronizada com os vizinhos.

▪ (F) – Estado Operacional

Configurações foram sincronizadas. Enlace está no estado operacional normal.

▪ (G) – Estado Degradado

Não há CC operando, mas o enlace TE ainda inclui enlaces de dados alocados para tráfego de usuário.



- (1) Enlace de dados atribuídos ao enlace TE, mas não há canal de controle up
- (2) Enlace de dados atribuídos ao enlace TE e existe canal de controle up
- (3) LinkSummary
- (4) LinkSummaryAck
- (5) LinkSummaryNack
- (6) Temporizador expirou
- (7) Primeiro canal de controle fica up
- (8) Último canal de controle fica down
- (9) Último enlace de dados é removido

Fig. 3. Rede de MEFCs de enlace TE formada a partir da MEF de enlace TE.

Com base na rede de MEFCs apresentada na Fig. 3, é possível aplicar o método com progresso justo. Como resultado constrói-se o grafo de alcançabilidade justa (Fig. 4).

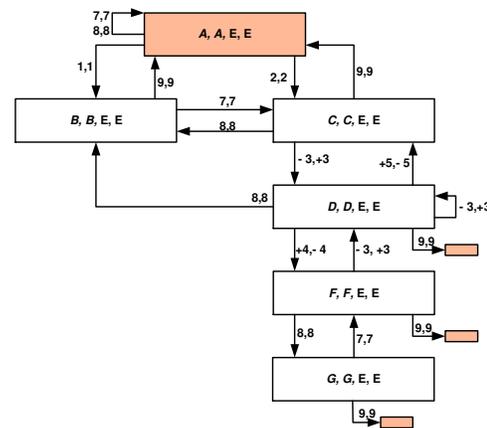


Fig. 4. Grafo de Alcançabilidade Justa resultante da rede de MEFCs do LMP relacionado ao enlace TE.

De acordo com a grafo (Fig. 4) e com base nas definições de erros de progresso, nota-se que o LMP relacionado ao enlace TE não apresenta nenhum dos cinco erros de progresso.

C. MEF de Enlace de Dados

A MEF de enlace de dados (Fig. 5) define estados e lógicas de operação de um enlace de dados.

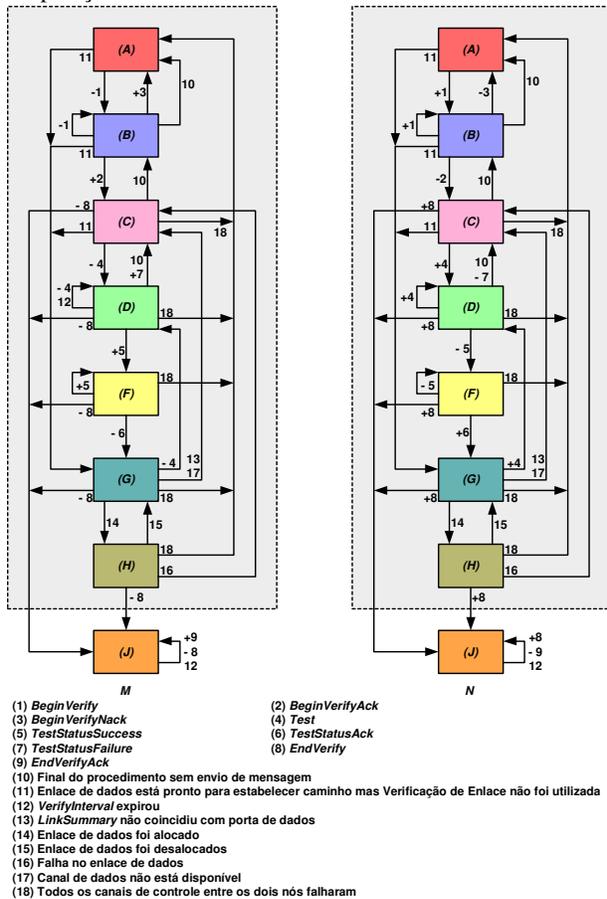


Fig. 5. Rede de MEFCs de enlace de dados.

Os estados utilizados na Fig. 5 são definidos como:

- (A) – Estado Inicial
Enlaces de dados não estão em serviço.
- (B) – Estado *SupVerif*
Confere se o procedimento de verificação é suportado ou não.
- (C) – Estado *InicVerif*
A verificação do enlace é suportada e inicializada. O enlace de dados não está operando.
- (D) – Estado de Teste
Enlace de dados está sendo testado. Uma mensagem *Test* é enviada periodicamente através do enlace.
- (F) – Estado *Teste/Local*
Enlace foi testado com sucesso, colocado em serviço pelo nó local, mas ainda não foi alocado para tráfego de dados.
- (G) – Estado *Teste/Remoto*
Enlace foi testado com sucesso, colocado em serviço pelo nó remoto, mas ainda não foi alocado para tráfego de dados.
- (H) – Estado *Alocado*

Enlace de dados está no estado de operação normal e foi alocado para tráfego de dados.

- (J) – Estado Final
Verificação dos enlaces de dados é finalizada

A região tracejada na Fig. 5 indica que o procedimento é repetido para cada enlace de dados que compõe um enlace TE. Se existe algum enlace de dados que não faz parte de nenhum enlace TE, então a região tracejada é executada somente uma vez.

É possível aplicar o método de análise com progresso justo na rede da Fig. 5 e o resultado é a construção do grafo de alcançabilidade justa, como mostra a Fig. 6.

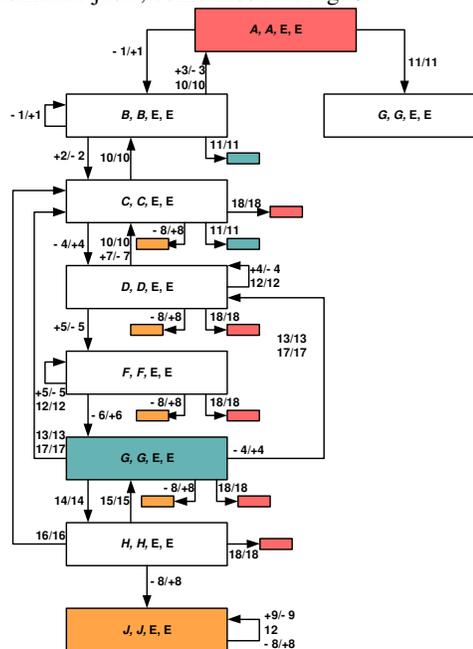


Fig. 6. Grafo de Alcançabilidade Justa resultante da rede de MEFCs do LMP relacionado ao enlace de dados.

De acordo com a grafo e com base nas definições de erros de progresso, nota-se que o LMP relacionado ao enlace de dados não apresenta nenhum dos cinco erros de progresso.

D. MEF de Gerenciamento de Falhas

A MEF de falhas (Fig. 7) define estados e lógicas de operação do procedimento de gerenciamento de falhas do LMP.

Os estados utilizados na Fig. 7 são definidos como:

- (A) – Estado de Detecção
Falha é detectada e indicada ao nó *upstream*.
- (B) – Estado de Localização
É feita a localização de falhas através do envio e recebimento de mensagens.
- (C) – Estado Pedido
Pedido para localização de falha é recebido com sucesso.
- (D) – Estado de Correlação
É realizada a correlação de falha localmente.
- (F) – Estado *Status*
O *status* do canal é requerido pelo nó vizinho.

▪ (G) – Estado de Notificação

É feita a notificação da falha ao nó que originou o gerenciamento.

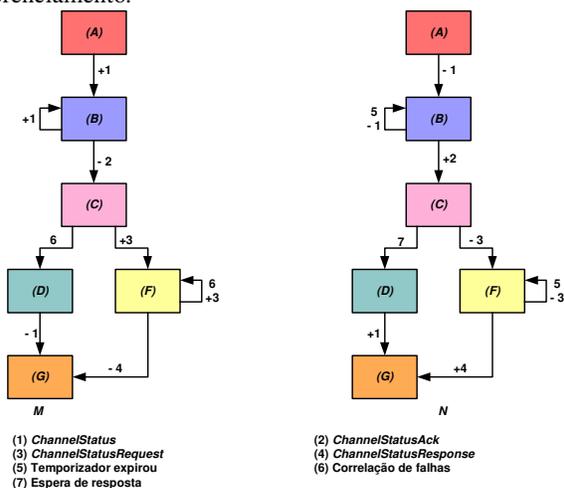


Fig. 7. Rede de MEFCs de gerenciamento de falhas.

Com base na rede de MEFCs apresentada na Fig. 7, a análise com progresso justo é aplicada resultando no grafo de alcançabilidade justa (Fig. 8). Nota-se que o LMP relacionado ao gerenciamento de falhas não apresenta nenhum dos cinco erros de progresso.

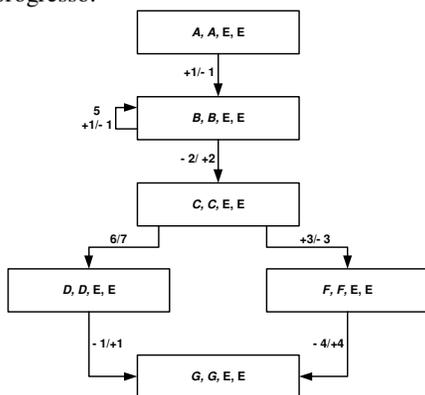


Fig. 8. Grafo de Alcançabilidade Justa resultante da rede de MEFCs do LMP relacionado ao gerenciamento de falhas.

E. ORDEM DE EXECUÇÃO DOS PROCEDIMENTOS DO LMP

O Internet Draft [draft-ietf-ccamp-lmp-10.txt] menciona que a função de correlação de propriedade de enlace deve ser, dentre alguns instantes, acionada antes do enlace TE ser estabelecido e colocado em funcionamento. Se o enlace TE ainda não foi estabelecido então o nó local tem conhecimento do *Link_ID* local (identificador de enlace TE local) mas não tem conhecimento do *Link_ID* remoto. Note que de acordo com a descrição da mensagem *LinkSummary*, esta possui o valor do *Link_ID* remoto como um dos parâmetros que a compõe, mas nenhuma citação é feita com relação ao valor que se deve atribuir ao *Link_ID* remoto caso este não for conhecido. Portanto, deve-se permitir que o *Link_ID* remoto não seja incluído na mensagem *LinkSummary* ou que seja incluído com o valor zero e o mesmo seja então aprendido dinamicamente quando o nó local receber a mensagem

LinkSummary (*Ack/Nack*) como resposta do nó remoto contendo o *Link_ID* remoto. Para que isso aconteça com sucesso, deve ser de conhecimento do nó local todos os *Interface-IDs* local/remoto dos enlaces de dados que compõe o enlace TE para que não haja erros quanto ao enlace TE sendo utilizado. O conhecimento dos *Interface_IDs* somente é possível através de uma configuração manual ou, dinâmica através da verificação de conectividade de enlace que deve portanto, ser obrigatoriamente realizada antes da correlação de propriedade de enlace e assim, deixa de ser um procedimento opcional.

Com base nas especificações de cada procedimento LMP, bem como em observações como a apresentada à cima, propõe-se uma ordem de execução dos procedimentos:

- (i) Gerenciamento de CC - (a) Ativação do CC; (b) Manutenção do CC
- (ii) Verificação de conectividade de enlace – (a) Pelo menos um CC deve estar ativo; (b) Descobrimto dinâmico das associações entre *Interface_IDs* local e remoto de um mesmo enlace de dados; (c) Verificação de conectividade de enlaces de dados; (d) Descobrimto dinâmico de associações de enlace TE
- (iii) Correlação de Propriedade de enlace – (a) Mapeamento de *Interface_IDs* deve ser um conhecimento já adquirido pelo nó local; (b) Agregação de múltiplos enlaces de dados a um enlace TE; (c) Sincronização de propriedades de enlace
- (iv) Gerenciamento de Falhas – (a) Notificação do *status* de enlaces; (b) Localização de falhas.

Esta ordem deve ser seguida quando a rede for colocada em funcionamento, pois nesse caso, a maior parte das informações será gerada e descoberta dinamicamente. No decorrer do funcionamento da rede, essa ordem de execução não necessariamente precisa ser seguida na maioria dos possíveis cenários.

VIII. CONCLUSÃO

Verificou-se que o LMP está livre dos erros de progresso (*deadlocks*, recepções não-especificadas, interações não-executadas, *livelocks* e *overflow*) e, com base na análise de alcançabilidade justa e nas definições de cada procedimento do LMP, foi proposta uma ordem de execução destes procedimentos.

REFERÊNCIAS

- ABOUL-MAGD, O., et al., "A Framework for Generalized Multiprotocol Label Switching", Internet Draft, draft-many-ccamp-gmpls-framework-00.txt, Março 2001.
- BERGER, L., "Generalized Multiprotocol Label Switching (GMPLS) Signaling Functional Description", RFC3471, Janeiro 2003.
- BRAND, D., ZAFIROPOULO, P., "On Communicating Finite-State Machines", Journal of the Association for Computing Machinery, vol. 30, no. 2, Abril 1983, pp. 323-342.
- GOUDA, M. G., Chang, C. K., "A Technique for Proving Liveness of Communicating Finite State Machines With Examples", ACM 1/84/008/0038, 1984, pp/ 38 - 49.
- LANG, J., "Link Management Protocol (LMP)", Internet Draft, draft-ietf-ccamp-lmp-06.txt, Setembro 2002, data de expiração Março 2003.
- MANNIE, E., "Generalized Multiprotocol Label Switching Architecture", Internet Draft, draft-ietf-ccamp-gmpls-architecture-07.txt, Maio 2003, expiração Novembro 2003.
- PAPADIMITRIOU, D. Rousseau, B., "Demystifying GMPLS - A Technical Perspective", Technology White Paper, Alcatel, 2003.
- PENG, W., "Deadlock Detection in Communicating Finite State Machines by Even Reachability Analysis", Mobile Networks and Applications 2, pp 251-257, 1997.
- RUBIN, J., West, C. H., "An Improved Protocol Validation Technique", Computer Networks, 6:65-73, 1982.