International Journal of Pure and Applied Mathematics
Volume 93 No. 1 2014, 45-59
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: http://dx.doi.org/10.12732/ijpam.v93i1.5

ijpam.eu

USING HIGH LEVEL PETRI NETS TO MODEL THE INTER-SESSION NETWORK CODING

O. Al-Aboud[1 §], M.J. Al-Laban[2], A.A. Hanano[3]
[1,2,3]Department of Mathematics
Faculty of Science
Damascus University
Damascus, SYRIA

Abstract: In this paper, we model an acyclic communication network as a net graph, and we use this net graph to define inter session-network coding as high level Petri graph. We then model inter-session network coding as a set of streaming high level Petri net graphs. Finally, we find a particular binding for the variables sets of these streaming high level Petri graphs in such a way the sink places in the net graph which models the acyclic communication network can be marked with the requested marking.

AMS Subject Classification: 94A99
Key Words: intersession network coding, high level Petri nets, acyclic communication network

1. Introduction

The theory of network coding for a single multicast session (intra-session network coding) is well studied by Ahlswede et al. [1], Follow-up works include [2], which shows that linear network coding is sufficient for a single multicast session. An algebraic approaches to network coding have been introduced in [3]-[5]. Several other related works of intra-session network coding can be found in [6]–[8]. when we generalize the problem such that there are more than one

Received: September 27, 2013

§Correspondence author

session and receivers may demand different sets of information, finding the optimal network coding strategy is still an open question. The linear coding is shown to be insufficient for optimal coding in the multi-session case [9]. In their paper [3] Koetter and Medard introduced an algebraic point view on network coding but the solvability decision problem is shown to involve Grobner basis computation, whose complexity may prohibit practical implementations for large problems. In this paper, we model an acyclic communication network as a net graph (NG), and we use this net graph to define inter session-network coding (INC) as high level Petri graph (HLPNG) [10],[11]. We then model inter-session network coding as a set of streaming high level Petri graphs (SHLPNG). We then find a particular binding for the variables sets of these streaming high level Petri graphs in such a way sink places in the net graph which models the acyclic communication network can be marked with the requested marking. Our work is based on high level Petri nets [11], which were chosen due to their well-accepted capability of expressing concurrency, non-determinism and reactive behavior, Furthermore, the wide range of simulation and analysis tools for Petri nets can be exploited for the evaluation of models and the verification of crucial system properties [12].The rest of this article is organized as follows, we will start with a notational description of the inter-session network coding problem and the high level Petri net. We then introduce an algorithm to construct the network graph (NG) for a high level Petri net from a communication network. In Section 3, we define inter-session network coding as high level Petri net graph. In Section 4, we model the inter-session network coding (INC) as a set of streaming high level Petri net graphs. In Section 5, we find a particular binding for the variables sets of these streaming high level Petri net graphs.

2. Settings and Formulation

2.1. System Setting

We model the network by a directed acyclic graph $G = (V, E)$ where V and E are the set of all nodes and links, respectively [13], links are denoted by round brackets $(v_1, v_2) \in E$ and assumed to be directed, the head and tail of an link $e = (v', v) \in E$ are denoted by $v = head(e)$ and $v' = tail(e)$ [3], for each node $v \in V$ we define $In(v) = \{e \in E; head(e) = v\}$, $Out(v) = \{e^{'} \in E; tail(e^{'}) = v\}$,then the sets of source and sink nodes are define as follows: $S = \{v \in V : |In(v)| = 0\}$, $R = \{v \in V : |Out(v)| = 0\}$, let $S = \{s_1, s_2, ...s_w\}$ and $R = \{R_1, R_2, ...R_N\}$. Without loss of generality we assume that all source nodes produce exactly one unit of data per unit time, and assume that all sink nodes demand exactly one unit of data per unit time, so if a source node s produces more than one unit of data per unit time, we add virtual source nodes that produce exactly one unit of data per unit time, and if a sink node t demands more than one unit of data per unit time, we add virtual sink nodes that demand exactly one unit of data per unit time. Each session is denoted by a tuple (s^i, t^i) where s^i and t^i denote the source and the sink node of the session i [14].

2.2. Petri Nets

Petri nets are a graphical and mathematical modeling tool applicable to many systems, they are promising tools for describing and studying information processing systems [15].

Definition 1. [11] A net graph is a structure $NG = (P, T, F)$ where:

- P is a finite set of nodes, called Places.
- T is a finite set of nodes, called Transitions, disjoint from P.
- $F \subset (P \times T) \bigcup (T \times P)$ is a set of directed edges called arcs.

It is easy to verify that definition 1 is equivalent to definition 2.

Definition 2. A net graph is a structure $NG = (P, T; F_{in}, F_{out})$ where:

- P is a finite set of nodes, called Places.
- T is a finite set of nodes, called Transitions, disjoint from P.
- $F_{in} \subset T \times P$ is a set of of directed edges called input arcs.
- $F_{out} \subset P \times T$ is a set of of directed edges called output arcs.

Definition 3. for each $p \in P$, $t \in T$ we define the following sets:

- $Pre(p) = \{t : t \in T \wedge (t, p) \in F_{in}\}$.
- $Post(p) = \{t : t \in T \wedge (p, t) \in F_{out}\}$.
- $Pre(t) = \{p : p \in P \wedge (p, t) \in F_{out}\}$.
- $Post(t) = \{p : p \in P \wedge (t, p) \in F_{in}\}$.

- $IA(p) = \{(t,p) : t \in Pre(p)\}$.
- $OA(p) = \{(p,t) : t \in Post(p)\}$.
- $IA(t) = \{(p,t) : p \in Pre(t)\}$.
- $OA(t) = \{(t,p) : p \in Post(t)\}$.

Definition 4. [11] A HLPNG is a structure $(NG, Sig, V, H, Type, AN, M_0)$ where:

- NG is a net graph.
- $Sig = (SG, O)$ is a Boolean signature.
- V is an S-indexed set of variables, disjoint from O.
- $H = (S_H, O_H)$ is a many-sorted algebra for the signature Sig.
- $Type : P \to S_H$ is a function which assigns types to places.
- $AN = (A, TC)$ is a pair of net annotations:
  - A is a function that annotates each arc with a term that when evaluated (for any binding) results in a multiset over the associated place's type.
  - TC is a function that annotates transitions with Boolean expressions.
- M_0 is the initial marking function which associates a multiset of tokens (of the correct type) with each place.

2.3. Modeling the Acyclic Communication Network $G = (V, E)$ as a net graph $NG = (P, T; F_{in}, F_{out})$

We will introduce an algorithm to model the communication network $G = (V, E)$ as a net graph $NG = (P, T; F_{in}, F_{out})$.

2.3.1. Algorithm

Input : A directed acyclic graph $G = (V, E)$, Set of sources S, Set of sinks T.

1. $P = \phi, T = \phi, F_{in} = \phi, F_{out} = \phi$.

2. For (every node, in any upstream-to-downstream order [16])

   - If $|Out(v)| \geqslant 1$ then:
     - Let $Out(v) = \{e_{v^1}, ..., e_{v^m}\}$ where $m = Out(v)$.
     - Create a set of new places $\{p_{v^1}, ..., p_{v^m}\}$.
     - $P \leftarrow P \cup \{p_{v^1}, ..., p_{v^m}\}$.
     - For(i=1; $i \leq m$; i++)
       - If ($|In(head(e_{v^i})| \geqslant 1$ and there is a transition $t_{e_{w^j}} \in T$ where $e_{w^j} \in In(head(e_{v^i}))$)
         - Create a new arc $o_{e_{v^i}} = (p_{v^i}, t_{e_{w^j}})$.
         - $F_{out} \leftarrow F_{out} \cup \{o_{e_{w^j}}\}$.
       - Else
         - Create a new transition $t_{e_{v^i}}$.
         - $T \leftarrow T \cup \{t_{e_{v^i}}\}$.
         - Create a new arc $o_{e_{v^i}} = (p_{v^i}, t_{e_{v^i}})$.
         - $F_{out} \leftarrow F_{out} \cup \{o_{e_{v^i}}\}$.
     - For(i=1; $i \leq m$; i++)
       - If (there is a transition $t_{e_{u^k}} \in T$ where $e_{u^k} \in In(v)$)
         - Create a new arc $in_{e_{v^i}} = (t_{e_{u^k}}, p_{v^i})$.
         - $F_{in} \leftarrow F_{in} \cup \{in_{e_{v^i}}\}$.
   - else
     - Create a new places p_v.
     - $P \leftarrow P \cup \{p_v\}$.
     - If (there is a transition $t_{e_{w^j}} \in T$ where $e_{w^j} \in In(v)$)
       - Create a new arc $in_{e_v} = (t_{e_{w^j}}, p_v)$.
       - $F_{in} \leftarrow F_{in} \cup \{in_{e_v}\}$.

3. For (every place $p \in P$)

   - If $(Pre(p) = \phi)$
     - Create a new transition t_p.
     - $T \leftarrow T \cup \{t_p\}$.
     - Create a new arc $in_p = (t_p, p)$.
     - $F_{in} \leftarrow F_{in} \cup \{in_p\}$.

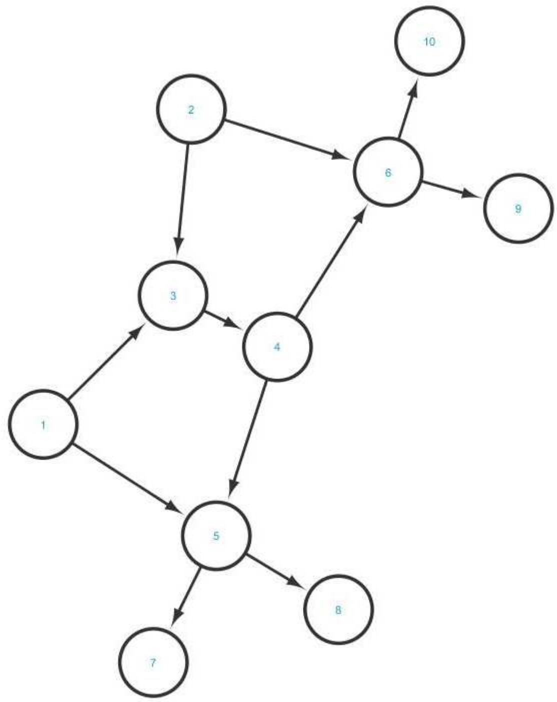

Figure 1

Output : $NG = (P, T; F_{in}, F_{out})$.

Example 5. If the input of the algorithm 2.3.1 is the network in Figure 1 then its output is the network graph in the Figure 2

We will introduce some important definitions in the net graph NG which models acyclic communication network, and define the streaming high level Petri net(SHLPN).

Definition 6. A place p is a source place if and only if

$$|IA(Pre(p)| = 0.$$

Remark 7. In the output of the algorithm 2.3.1 we denote the set of source places which represent the source node s_i, by

$$P_{s_i} = \{s_{ij} : j = 1, ..., c_i\}$$

where c_i is the number of places in NG which represent the source node s_i.

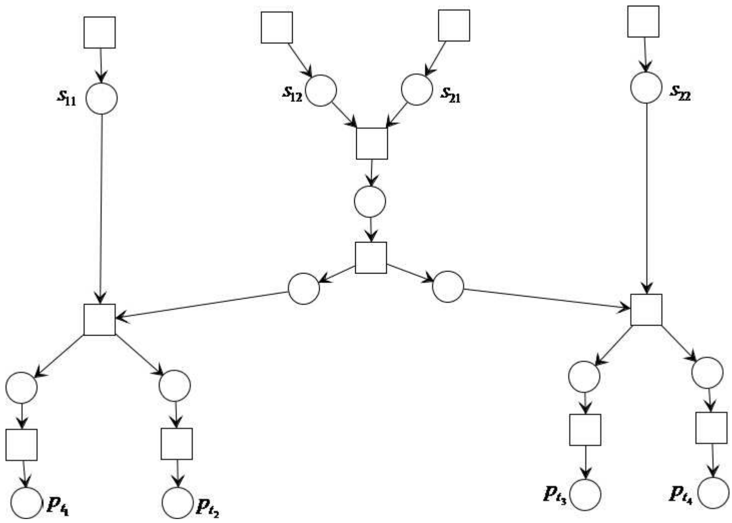

Figure 2

Definition 8. A transition t is a virtual transition if and only if $|IA(t)| = 0$.

Definition 9. A virtual input arc is an input arc from a virtual transition to a source place.

Remark 10. We denote the set of virtual transition by

$$T_s = \{t_{ij} : i = 1, ..., |s|; j = 1, ..., c_i\}$$

where c_i is the number of places in NG which represent the source node s_i.

Definition 11. A place p is a sink place if and only if $Post(p) = \phi$.

Definition 12. Streaming high level Petri net graph (SHLPNG) is a HLPNG that satisfies the following conditions:

- $|Pre(p)| \leq 1$, $|Post(p)| \leq 1$:$\forall p \in P$.
- $|Post(t)| = 1$:$\forall t \in T$.

Definition 13. The requested marking in the sink place $p_{R_i}$is the symbol which is demanded by the sink node R_i.

3. Modeling the Inter-Session Network Coding (INC) as a HLPNG

In this section we present a formal definition of inter-session network coding (INC) by using the definition of high level Petri net graph (HLPNG).

Definition 14. An INC on the communication network $G = (V, E)$ is a structure

$$INC = (NG, Sig, V, H, Type, AN, M_0)$$

where:

- $NG = (P, T; F_{in}, F_{out})$ is the net graph which represents $G = (V, E)$ according to the Algorithm 2.3.1.
- $Sig = (SG, O)$ with $SG = \{F', BOOL\}$, $O = \{\sum, true_{BOOL}\}$.
- $V = \{a_1, ..., a_n\}$ is a finite set of variables.
- $H = (\{F_q, Boolean\}, \{L, true\})$ is a many-sorted algebra for the signature Sig, where: F_q is a finite field of size q and the obvious correspondences are made :$F_H = F_q, BOOL_H = Boolean, L_H = \sum, (true_{BOOL})_H = true.$
- $Type : P \to \{F_q\}$ is a function which assigns the type F_q to all places.
- $AN = (A, TC)$ is a pair of net annotations:
  - $A : F \to F_q[V]$: $A(t, p) = \sum_{p_i \in Pre(t)} \overline{M}(p_i).a_i$ where $a_i \in V$, $\overline{M}(p_i) = A(Pre(p_i), p_i)$ and $A(t_{ij}, s_{ij}) = X_i$ for all j where t_{ij} is a virtual transition and X_i is the symbol which is produced in the source node s_i, $A(p, t) = A(Pre(p), p)$.
  - $TC : T \to \{true_{BOOL}\}$ is a function that annotates every transition with the Boolean constant true.
- $M_0 : P \to \mu F_q$: $M_0(p) = \{X_i\}$ if $p = s_{ij}$ for all j, $M_0(p) = \phi$ otherwise.

This definition can be seen to be one to one correspondence with a more simple definition given below.

Definition 15. An INC on the communication network $G = (V, E)$ is a structure

$$INC = (NG, V, A, M_0)$$

where:

- $NG = (P, T; F_{in}, F_{out})$ is the net graph which represents $G = (V, E)$ according to the Algorithm 2.3.1.

- $V = \{a_1, ..., a_n\}$ is a finite set of variables

- $A : F \rightarrow F_q[V]$: $A(t, p) = \sum_{p_i \in Pre(t)} \overline{M}(p_i).a_i$ where $a_i \in V$, $\overline{M}(p_i) = A(Pre(p_i), p_i)$ and $A(t_{ij}, s_{ij}) = X_i$ for all j where t_{ij} is a virtual transition and X_i is the symbol which is produced in the source node s_i, $A(p, t) = A(Pre(p), p)$.

- $M_0 : P \rightarrow \mu F_q : M_0(p) = \{X_i\}$ if $p = s_{ij}$ for all j, $M_0(p) = \phi$ otherwise.

This is because:

- Every transition is annotated with the Boolean constant true.

- The type of each places is the same and given by F_q hence there is no need typing places.

4. Modeling the Inter-Session Network Coding (INC) as a Set of Streaming High Level Petri Net Graphs

In this section we remodel the INC as a set of streaming high level Petri nets, by introduce couple of algorithms. The first algorithm transfers the net graph of the HLPNG (which models the INC), to a set of net graphs which satisfy the conditions of streaming high level Petri net graph. The second algorithm constructs the set of streaming high level Petri net graphs (which model the INC) from the output of the first algorithm.

4.1. Algorithm

Input: INC, P_S, $P^{'}$.

For (every $p_i \in P^{'}$)

- $P_i = \{p_i\}$, $T_i = \phi$, $F_{i_{in}} = \phi$, $F_{i_{out}} = \phi$.
- *repeat*
  - For (every $p \in P_i$, p is not signed)
    - If (there is a signed transition $t \in Pre(p)$).
      - $F_{i_{out}} \leftarrow F_{i_{out}} \setminus \{(p, Post(p))\}$.
      - $P_i \leftarrow P_i \setminus \{p\}$.
      - Sign p.
    - Else
      - $T_i \leftarrow T_i \bigcup Pre(p)$.
      - $F_{i_{in}} \leftarrow F_{i_{in}} \bigcup (Pre(p), p)$.
      - Sign p.
  - For (every $t \in T_i$, t is not signed)
    - $P_i \leftarrow P_i \bigcup Pre(t)$.
    - $F_{i_{out}} \leftarrow F_{i_{out}} \bigcup (Pre(t), t)$.
    - sign t.
- Until all p_j and t_i are signed.

Output: $NG_i = (P_i, T_i; F_{i_{in}}, F_{i_{out}}); i = 1, ..., |p^{'}|$.

Example 16. If the input of the algorithm 4.1 is the network graph in Figure 2 then it's output is the network graphs in the Figure 3.

Definition 17. $|Pre(t)|_{NG_i}$, $|Post(t)|_{NG_i}$, $|Pre(p)|_{NG_i}$, $|Post(p)|_{NG_i}$ are the cardinals of the sets $Pre(t)$, $Post(t)$, $Pre(p)$, $Post(p)$ respectively, in NG_i where $t \in T_i$, $p \in P_i$.

Definition 18. An arc $f' \in F_{i_{in}}$ connects to the arc $f \in F_{i_{in}}$ if and only if there is a transition $t \in T_i$ and a place $p \in P_i$ such that $p \in Pre(t)$, $f' \in OA(t)$ and $f \in IA(p)$.

Remark 19. In the output of the algorithm 4.1, for each $k; 1 \leq k \leq |p'|$ we denote the set of source places in NG_k which represent the source node s_i in G) by $P^k_{s_i} = \{s^k_{ij} : j = 1, ..., m_k\}$ where m_k is the number of places in NG_k which represent the source node s_i.

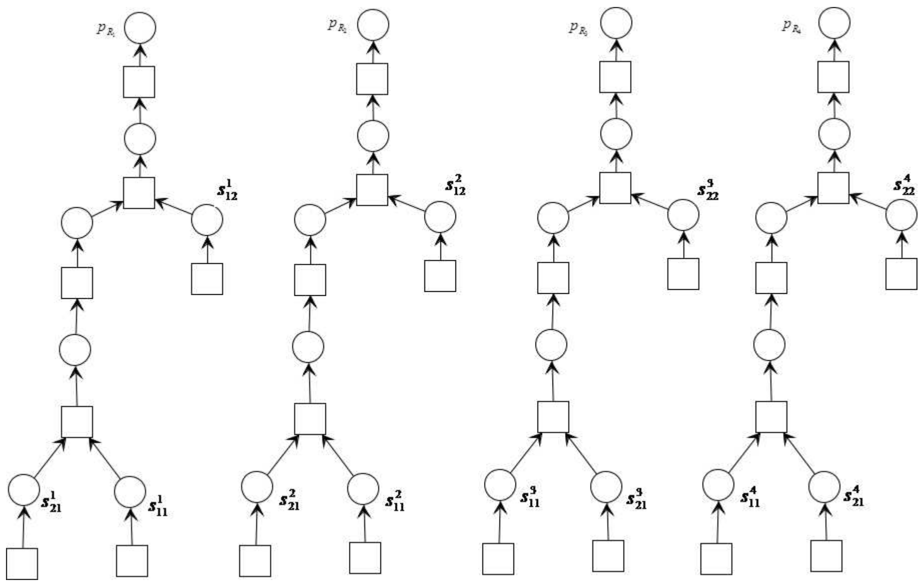

Figure 3

4.2. Algorithm

Input : $NG_i = (P_i, T_i, F_{i_{in}}, F_{i_{out}}); i = 1, ..., |P^{'}|$.

For every $NG_k; i = 1, ..., |p^{'}|$, construct $H_{k=}(NG_k, V_k, A_k, M_{k_0})$ as follows:

- For each $s^k_{ij} \in P_k$ we define a new variable a^k_{ij}.
- $M_{k_0}(p) = \{a^k_{ij} X_i\}$ if $p = s^k_{ij}$, $M_{k_0}(p) = \phi$ otherwise.
- $A_k : F_k \rightarrow F_q$;
  - $A_k(t^k_{ij}, p^k_{ij}) = a^k_{ij} X_i$ for all j where t^k_{ij} is a virtual transition.
  - If t is not a virtual transition then $A_k(t, p) = \sum_{p_i \in Pre(t)} \overline{M_k}$ where $\overline{M_k} = A_k(Pre(p_i), p_i)$.
  - $A_k(p, t) = A_k(Pre(p), p)$.

Output : $H_i = (NG_i, V_i, A_i, M_{i_0}); i = 1, ..., |P^{'}|$.

5. Solving the Inter-Session Network Coding Problem by Finding a Particular Binding for the Variables Sets $V_1, V_2, ..., V_{|P'|}$

There are two sets of equations which have to be satisfied by a binding for variables sets $V_1, V_2, ..., V_{|P'|}$ to solve the inter-session network coding [4]:

- The first one is the equations which the sink places use them to obtain the requested marking, we note that for each $p \in P'$ there is a unique Petri net H_i such that $p \in P_i$, if the requested marking in p is $M_0(p_l)$, $f = (t, p)$ is the input arc for p in H_i, $A(f) = \sum_{k=1}^{|S|} a_k^j X_k$ where $a_k^i = \sum_j a_{kj}^i$ then :

$$a_k^j = \{ \begin{matrix} 0; k \neq l \\ 1; k = l \end{matrix}$$

- the second one comes from the fact that multi-copies of the same arc in NG maybe appear in more than one net graph from $\{NG_1, ..., NG_{|P'|}\}$, assume f appears in NG_l and $NG_k$where $l, k \in \{1, 2, ..., |P'|\}$; $l \neq k$ and $A_l(f) = \sum_{i=1}^{|S|} a_i^l X_i$, $A_k(f) = \sum_{i=1}^{|S|} a_i^k X_i$ then $\frac{a_1^l}{a_1^k} = \frac{a_2^l}{a_2^k} = ... = \frac{a_{|S|}^l}{a_{|S|}^k}$, to avoid division by zero when some variable take the value zero we write $a_i^l a_j^k = a_j^l a_i^k$ for $i, j = 1, ..., |S|; i \neq j$.

we note that:

- Each equation in the first set involves mutually exclusive set of variables.
- Each equation in the second set has a maximum degree of two.

So we can use these properties to simplifying the equations.

After we find a binding for the variables sets $V_1, V_2, ..., V_{|P'|}$ which satisfies the previous equations, we use this binding to evaluate the expression for each input arc in NG as follows: Let $f \in F_{in}$ and $f_1, f_2, ..., f_{|D|}$ are the corresponding arcs in the streaming Petri nets $H_i : i = 1, ..., |P'|$ and let $A(f_i) = b_i^f A(f)$ where $A(f) = \sum_{j=1}^{|S|} a_j^f X_j$ so we can define two vectors $C^f = (a_1^f, a_2^f, ..., a_{|S|}^f)$, $B^f = (b_1^f, b_2^f, ..., b_{|D|}^f)$ for each $f \in F_{in}$, and if $t \in T$, $p \in Pre(t)$, $f' \in OA(t)$ and $f \in IA(p)$ we define sub vector $B_{f'}^f$ collecting the multiplying factors on corresponding arcs of f' that connect to corresponding arcs of f. The vectors C^f, B^f for each $f \in F_{in}$ can be computed by the following algorithm.

5.1. Algorithm

Input: $NG = (P, T, F_{in}, F_{out})$, P_S, P', $NG_i = (P_i, T_i, F_{i_{in}}, F_{i_{out}})$;$i = 1, \ldots\ldots, |P'|$ and a binding for the variables sets $V_1, V_2, \ldots\ldots, V_{|P'|}$ satisfies the two sets of equations above.

- For (every $t \in T$, in any upstream-to-downstream order)
  - If (t is a virtual transition)
    - ∗ Suppose $t = t_{ij}$ and (t^1_{ij}, s^1_{ij}),...,$(t_{ij}^{|c_{(t_{ij},s_{ij})}|}, s_{ij}^{|c_{(t_{ij},s_{ij})}|})$ are the corresponding arcs to (t_{ij}, s_{ij}) in the streaming high level Petri nets. We know that $A(t_{ij}, s_{ij}) = X_i$ so $C^{(t_{ij},s_{ij})} = (0^i 1 0^{|S|-i})$ and $B^{(t_{ij},s_{ij})}(k) = a^k_{ij}$.
  - Else
    - ∗ if (C^f, B^f have been calculated for all $f \in IA(p); \forall p \in Pre(t)$)
      - · For (every $f' \in OA(t)$)
        1. calculate the matrix $E^f_{f'} = (B^f_{f'})^T.C^f$ for all $f \in IA(p); p \in Pre(t)$ and define $E_{f'} := \sum_f E^f_{f'}$ so each row in $E_{f'}$ corresponds to $A_i(f'')$ for some $i \in \{1, \ldots.., |p'|\}$ where f'' is a corresponding arc of f.
        2. Let $C^{f'}$ be any non-zero row (say,i) of $E_{f'}$ or the zero row if $E_{f'}$ is a zero matrix.
        3. Put $B^{f'}(j)$ to equal the result of dived the j th row of $E_{f'}$ by $C^{f'}$.

Output: The vectors C^f, B^f for each $f \in F_{in}$.

The set of vectors $\{C^f : f \in F_{in}\}$ determines a binding for the variable set V, which allows us to mark the sink places with the requested marking and subsequently solve the inter-session network coding problem.

References

[1] R. Ahlswede, N. Cai, S.-Y, R. Li, and R. W. Yeung, Network information flow, *IEEE Trans. on Information Theory*, **46**, No. 4 (2000), 1204–1216, **doi:** 10.1109/18.850663.

[2] R. Li, R. W. Yeung, and N. Cai, Linear Network Coding, *IEEE Transactions on Information Theory*, **49**, NO. 2 (2003), 371-381, **doi:** 10.1109/TIT.2002.807285.

[3] R. Koetter and M. Medard, An algebraic approach to network coding, *IEEE/ACM Transactions on Networking*, **11**, No. 5 (2003), 782-795, **doi:** 10.1109/TNET.2003.818197.

[4] T. Abhay, Subramanian, Andrew Thangaraj, Path gain algebraic formulation for the scalar linear network coding problem, *IEEE Transactions on Information Theory*, **56**, No. 9 (2010), 4520-4531, **doi:** 10.1109/TIT.2010.2054270.

[5] K. R. Dinesh Kumar, A.Thangaraj, Algebraic network coding: A new perspective, *IEEE International Symposium on Information Theory*, (2009), 114-118, **doi:** 10.1109/ISIT.2009.5206017.

[6] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, Polynomial Time Algorithms For Network Information Flow, *15th ACM Symposium on Parallel Algorithms and Architectures*, (2003), 286-294, **doi:** 10.1145/777412.777464.

[7] Y. Wu, K. Jain, and S.-Y. Kung, A unification of network coding and tree-packing (routing) theorems, *IEEE Transactions on Information Theory*, **52**, No. 6 (2006), pp. 2398-2409, **doi:** 10.1109/TIT.2006.874430.

[8] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, Achieving minimum-cost multicast: A decentralized approach based on network coding,*24th Annual Joint Conference of the IEEE Computer and Communications Societies*, **3** (2005), 1607 - 1617, **doi:** 10.1109/INFCOM.2005.1498443.

[9] J. Sundararajan, M. Medard, R. Kotter, and E. Erez, A systematic approach to network coding problems using conflict graphs, *in Proc. ITA Workshop*, (2006), Invited paper.

[10] J. Billington, Many-sorted High-level Nets, *Proceedings of the Third International Workshop on Petri Nets and Performance Models*, (1989), 166-179, **doi:** 10.1109/PNPM.1989.68550.

[11] Software Engineering, ISO/IEC/JTC1/SC7, High-Level Petri Net Concepts Definitions and Graphical Notation, *ISO/IEC 15909-1, Final Committee Draft*, (2002).

[12] M. Koch, C. Rust, B. Kleinjohann, Design of intelligent mechatronical systems with high-level Petri nets, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, **1** (2003), 217-222, **doi:** 10.1109/AIM.2003.1225098.

[13] A. Khreishah, Chih-Chun Wang, N. B. Shroff, Rate Control With Pairwise Intersession Network Coding, *Transactions on Information Theory*, **18**, No. 3 (2010), 816-829, **doi:** 10.1109/TNET.2009.2032353.

[14] K. Ronasi, A. H. Mohsenian-Rad, V. W. S. Wong, S. Gopalakrishnan, R. Schober, Reliability-Based Rate Allocation In Wireless Inter-Session Network Coding Systems, *Global Telecommunications Conference*, (2009), 1-6, **doi:** 10.1109/GLOCOM.2009.5425445.

[15] T. Murata, Petri nets : Properties, analysis and application, *in Proceedings of the IEEE*, **77**, No. 4 (1989), 541-580, **doi:** 10.1109/5.24143.

[16] R.W. Yeung, Information Theory and Network Coding, *Springer Publishing Company*, US (2008), **doi:** 10.1007/978-0-387-79234-7.