Behavior Modeling of Coded Networks in Dynamic Networks

Farzad Amirjavid, Abdenour Bouzouane, Bruno Bouchard

Abstract— **Predicting, considering and designing all of the possible states of real world problems that are justified in artificial intelligence domain is relatively difficult or rather impossible for the experts. One reason is that the real world problems are highly complicated and they depend on a lot of variables. Furthermore, they do not practice their behavior as similar as their past comportments and it is rare that we find a linear behavior from such mentioned problems. We propose to apply data-driven data mining approaches to learn non-linear systems' behaviors (rather than expert's knowledge driven ones), so we could define delicate fuzzy states to indicate the system behavior. Intelligence may be a reason of non-linearly behavior of real world problems. The more intelligence is with a system, more variables are included in its behavior and a more non-linear behavior is expected. Observing a lot of variables and features concerning to the behavior of intelligent systems, would lead to a relatively great data warehouse. Analyzing this data we would model the behavior of non-linear systems and finally we would propose an optimization approach for non-linear systems to achieve their goals. As a case-study, a network coding problem is explained to illustrate well the proposed approach.**

***Keywords*— Network coding, artificial intelligence, temporal data mining, fuzzy logic, modelling.**

I. Introduction

Intelligent systems actuate the real material world by realizing scenarios and activities. Sensors observe these actuations and they would indicate features and aspects of the activities or in fact, the intentions of the intelligent system. We believe that intelligence within the intelligent systems aims to actuate the world (or its environment) to transit the world to a desired state. The more environmental aspects and features of an intelligent system are observed, the more we know and in the consequence we can reason more precisely in the intentions of the intelligent system. This idea leaded to make ambient environments such as smart homes[1] to survey the intelligence and intentions of intelligent humans and/or intelligent systems, which act non-linearly. In the consequence by performing this research we could provide assistance and provide optimization facilities for the intelligent systems and/or humans. The problem with such these problems is that a lot of variables with a lot of observations should be studied and by applying traditional quantitative data mining approaches such as Hidden Markov Model (HMM) [14], [15] or Naïve Bayesian Networks (NBN) [16] we would face a very high process complexity with very low efficiency to model the activities. Moreover, the gathered data contains high imprecision[2] and uncertainty [3][17].

To solve the mentioned problems, we apply fuzzy logic [1] to infer the fuzzy states of intelligent system. To handle the imprecision problem, the observations from a sensor are compared to themselves, so instead of consideration of absolute values, their quotas regarding to themselves are considered. To handle the uncertainty problem, values of all variables are compared together and fuzzy classes are made. Possibility theory is applied to calculate the similarity degrees of learned states with new observations, in contrast to what happens in traditional quantitative data mining approaches which calculate the probability of the transition between system states. To illustrate and describe our approach, we propose to observe the behavior of a system that its optimum behavior is already known, and then a bigger and more complicated system behavior[4] is considered. Next, we would survey if we can improve a big non-linear system functionality by finding contexts in which previously known optimum system can be applied to help the bigger system. In this way; naturally, the functionality of the more complicated system would be improved. Butterfly network in network coding problem is surveyed as a case study in our proposed approach.

A. Introduction to network coding

Networks provide connections between nodes and they can be modeled by graphs in which each edge represents a link and each node represents a clients or networking device. Connecting the nodes is a basic goal to create and apply networks. Traditional networks rely on physical components such as routers and switches to be structured and modeled. Today the world is more connected than ever. At one hand,

Manuscript received March 05, 2012; revised April 19, 2012. Basic concepts of this work such as fuzzy temporal data mining techniques, modeling and prediction techniques are already researched and supported by LIARA [13].

Farzad Amirjavid is PhD candidate in Computer Science in the department of mathematics and computer science of the University of Quebec at Chicoutimi (UQAC).

Abdenour Bouzouane is professor of Computer Science in the department of mathematics and computer science of the University of Quebec at Chicoutimi (UQAC).

Bruno Bouchard is professor of Computer Science in the department of mathematics and computer science of the University of Quebec at Chicoutimi (UQAC).

[1] LIARA [13] is an instance of smart homes that surrounds realization of activities.

[2] Imprecision is caused when sensors do not measure the activities features precisely.

[3] Uncertainty is caused when we face the lack of knowledge. The lack of knowledge is caused when some variables are not considered or we have a partial observation.

[4] It can act probably in a more non-linear manner.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

different types of connections for each node can be provided and at the other hand today's technology let each node be a data transmission device while being a user of a network. Moreover, links get down or are added, users get online and offline frequently; so, it can be inferred that today's networks are more dynamic. In this context, network coding is proposed to improve the efficiency of networks by improving the information flow. Minimizing the time of data transfer between two nodes is the final goal of the network coding [10]. In this context; butterfly network, is known already that acts optimally from different aspects such as network throughput, and data transfer speed.

We propose to consider networks as dynamic non-linear systems that can adapt dynamically to new environmental conditions. The key point is that we should consider the network structure completely abstracted from its physical entity. To do that the knowledge about the network's behavior is considered as network structure instead of its physical architect. According to the knowledge, components of the network can be arranged to provide desired functionality. Temporal data observed from network is mined and the behavior of the network is learned and modeled. Fuzzy logic [1] and fuzzy data clustering are applied to learn fuzzy states of the model. The advantage of the proposed solution is firstly that we can predict the behavior of the network components by observing the uncompleted information and secondly the delay of data transfer is considered. Distinguishing the normal world state from abnormal world state and recognition of correct data transmission are other goals of performing this research[5].

B. Introduction to butterfly network

Butterfly network is a special case of network coding. Butterfly network represents a better functionality and throughput rather than a router-based version in multicasting, especially for multimedia contents [11]. In physical layer, it allows a high rate of data transmission [3], [4], [5]. Furthermore, in wireless networks [7] and satellite communications [10], it has a famous functionality. The butterfly network is surveyed as a case study during the parts of this paper.

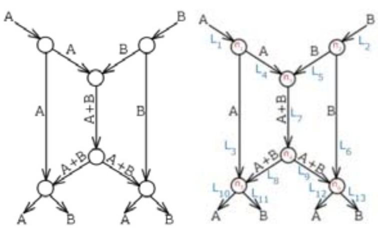

Fig 1. Butterfly network: the links and nodes of Butterfly network are names at the right side image

II. MODELING THE NETWORK BEHAVIOR

Perception from a network depends on the model that represents the behavior of network conceptual structures. The models are made according to the knowledge about network behavior. By models, we would represent our knowledge from the networks. For example, table1 represents linear model of butterfly network in its four possible contexts[6].

TABLE 1
Four possible contexts of butterfly network

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}	L_{13}
A=1, B=1	1	1	1	1	1	1	0	0	0	1	1	1	1
A=1, B=0	1	0	1	1	0	0	1	1	1	1	0	1	0
A=0, B=1	0	1	0	0	1	1	1	1	1	0	1	0	1
A=0, B=0	0	0	0	0	0	0	0	0	0	0	0	0	0

A definitive system with definitive states is considerable for this network and a simple linear model can be imagined for it. To do that according to each context the possible values of each link are calculated. The problem with a linear model is that it presumes the data is transmitted instantaneously, it does not represent the transitions from possible a step to another step, and delay in transmission of data may lead to conflictions to the linear model. Furthermore, to reason about normality of network behavior, the linear modeling of butterfly network with consideration of possible transitions between nodes, may lead to 2^{13} possible states. The networks that are more complex rather than the mentioned network would have even worse complexity. As the conclusion, it can be said that linear modeling of networks would not be easily practical for real world problems and instead we propose non-linear modeling of networks that consider uncertainties of time, transitions and changes in the network physical structure.

The knowledge can be learned in two ways; which are data-driven and the expert's knowledge-driven approach. The first approach models the knowledge applying data analysis and its learned knowledge is applied in the same environment that data was captured. The knowledge-driven approach inserts the knowledge directly from the expert. Considering that butterfly network could have 2^{13} possible states, so data-driven approaches seem to be more practical. We propose to apply data-driven approaches to discover the knowledge, because it lets the intelligent system to reason and to infer about runtime observations autonomously and expert's supervision would not be necessary.

By applying Fuzzy Inference System (FIS) in our research, we could learn the models of behaviors of network directly from data and we could provide high-level knowledge (fuzzy rules) for the expert to edit the knowledge if needed. To discover knowledge about network behavior (in the frame of fuzzy rules) we would mine the temporal data observed from network links. Finally, the result would be a model in conceptual graph frame that represents transitions from fuzzy states of the network. It should be mentioned that the concept of network model is different from the concept of network *topology* because models represent the network behavior; however, topologies represent the physical arrangement of network physical components. Here, we would introduce an approach to model the network behavior.

A. Observation

As it was mentioned earlier, we consider a network as a dynamic system and the behavior of this system depends on

[5] It includes validity of links and nodes functionality and correct transmission of data.

[6] These contexts are made based input values which are: {(A=1, B=1), (A=1, B=0), (A=0, B=0), (A=0, B=1)}

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the inputs of this system. The first step to model a network is to observe[7] the network's behavior. The observation of network links provides the primary data for analyze of network behavior. Temporal data is the result of frequently observation of network links and huge amount of data records may be created.

For instance, considering each link (edge) of butterfly network is observed frequently, we would have a dataset that per each link of the network has a data field and per each observation, a new record of data is inserted. A partial observation from butterfly network is like table2. Each record of temporal dataset represents a flash view to the network. In the consequence, we apply a temporal data mining technique to learn the knowledge.

TABLE 2
Temporal data captured from observation of butterfly network

ID	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}	L_{13}
1	1	1	1	1	1	1	0	0	0	1	1	1	1
2	1	1	1	1	1	1	0	0	0	1	1	1	1
3	1	1	1	1	1	1	0	0	0	1	1	1	1
4	0	1	1	1	1	1	0	0	0	1	1	1	1
5	0	1	0	1	1	1	0	0	0	1	1	1	1
6	0	1	0	0	1	1	0	0	0	1	1	1	1
7	0	1	0	0	1	1	0	0	0	1	1	1	1
8	0	1	0	0	1	1	1	0	0	1	1	1	1
9	0	1	0	0	1	1	1	1	0	1	1	1	1
10	0	1	0	0	1	1	1	1	1	1	1	1	1
11	0	1	0	0	1	1	1	1	1	0	1	1	1
12	0	1	0	0	1	1	1	1	1	0	1	0	1
13	0	1	0	0	1	1	1	1	1	0	1	0	1
14	0	1	0	0	1	1	1	1	1	0	1	0	1

Next step is to analyze the temporal data and mine it. Mining temporal data is different from non-temporal data. In non-temporal databases each record of represents a concept; however, each dataset (set of records) of temporal databases represent a system behavior. For more information about temporal data please refer to [2]. To mine the temporal data from networks we propose a fuzzy temporal data mining technique that creates fuzzy states for network. Transition from fuzzy states of the network creates fuzzy scenarios about network behavior and these models represent the normal behavior of network.

B. *Knowledge discovery*

In the proposed approach, we define the knowledge as couple of two sets. These sets are the *contexts* and the *series of fuzzy events* that possibly could occur in the concerning context. Contexts are the clues to recognize the possible initial states of the system. After recognition of the current world's context, we can verify if the occurring events are normal or not. In the network-coding problem, contexts are the possible combinations of *inputs*. For example the butterfly network has four possible contexts which are [(A=1,B=1),(A=1,B=0),(A=0,B=1),(A=0,B=0)].

Recognition of contexts is vital to recognize the normality of world state and to recognize the correct realization of scenarios[8], because contexts give us the clues to recognize the possible initial states of the system. After recognition of contexts (initial states), we can trace the occurring events to recognize if the scenarios are realized correctly[9]. In the next sections, we would discuss how to infer the scenarios by a data-driven approach.

C. *Temporal data mining and inference of fuzzy events*

Fuzzy subtractive clustering[10] is applied to compare the observed data of a network. Data points that have noticeable concentration of data around are recognized as fuzzy states of the network. Fuzzy states are the non-hardware-based states of a network that exist in the imagination of a reasoning system and these states may be different from states of nodes of network[11]. Subtractive clustering makes the fuzzy states of the network for us. For instance applying subtractive clustering algorithm on the presented data in table2 and comparing L_1 and L_4 we could have three possible states for that relation, which are {(0,0), (0,1), (1,1)}[12]. All the 13 existing links can be compared and represent in a 13-D environment. To illustrate the fuzzy states on a piece of paper - which lets us illustrate 2D environment - we can draw the results of each pair comparison. Figure2 illustrates the comparison results of L_1 and L_4.

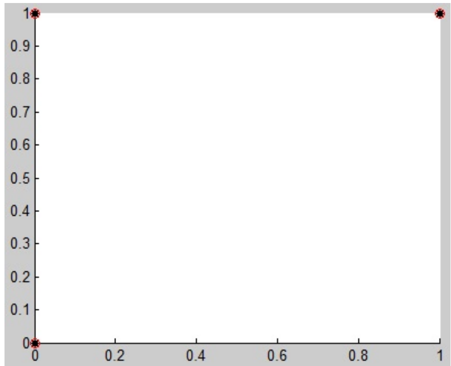

Fig 2. Three possible states of L_1 and L_4

Transition from a fuzzy state to another one is significant for the reasoning system and it represents a fuzzy event in the reasoning system. Therefore, the core of knowledge in the proposed system is the inferred fuzzy rules or fuzzy classes. For the butterfly network, nine fuzzy states are discovered by fuzzy inference system (figure 3). These nine fuzzy states form the knowledge core of the proposed system.

D. *Discovery of fuzzy states and fuzzy rules applying Fuzzy Inference System (FIS)*

Fuzzy Inference System [12] is a tool that is applied to learn the fuzzy states of the network. For instance, the butterfly network makes a FIS with thirteen inputs and one output. The inputs of this system observe the link values of the butterfly network and the output is set to '1' for the learning phase. Here, the assigned value to output ('1') represents normal world for the reasoning system. Expert can edit the

[7] The observation is performable frequently (for example every one microsecond) from each network link.

[8] Scenarios are the series of events. In other words, they are the sets of events that could consequently occur in the world.

[9] Here it means that if data is transferred correctly.

[10] The reason that we propose to apply subtractive clustering is that we have no idea about possible states of a network; furthermore, as it is a one-pass algorithm, it could be efficient.

[11] Fuzzy states are different from hardware states. Fuzzy states are made by comparing the data of hardware devices.

[12] The mentioned states are made according to the available training data; however, having more or less training data different fuzzy states could be resulted.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

knowledgebase – which is in fact the fuzzy rule base - to manage better the system. Further explanation is that expert can train the system with abnormal scenarios and select "0" for those scenarios. The inference system regards the world between normal state (it is outputted with '1') and abnormal state (it is outputted with '0') and reasons how much the world is normal. With the training data of table 2, nine fuzzy states are formed. At the next step, we would draw the conceptual graph of the butterfly network. Transition between fuzzy states of the network is in fact the *fuzzy event*.

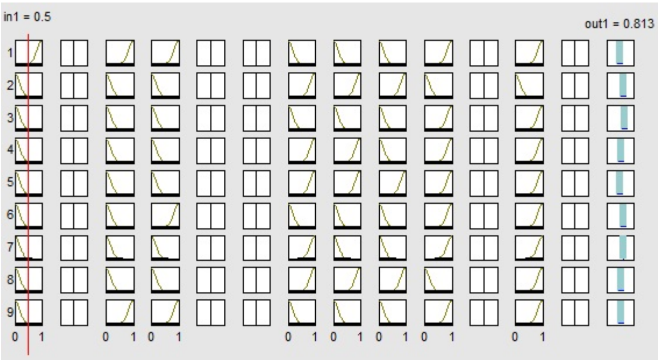

Fig 3. Six inferred rules of butterfly network behavior by FIS

E. Conceptual graph of network behavior

Conceptual structure is formalism for knowledge representation. Conceptual graphs illustrate graphically the conceptual structure behind the scenarios. In this paper, we propose to apply the inferred fuzzy states of the network as nodes of conceptual graph and the possibility to transit from a fuzzy state to another one (fuzzy event) is illustrated as edge between the fuzzy states of the network too.

As it is mentioned in the previous section, nine fuzzy states of the butterfly network are inferred. The concerning conceptual graph is proposed in figure 4. Based on the fuzzy rule base fuzzy state 1 is made when $L_1 = L_2 = L_3 = \ldots = L_{13} = 1$. The "fuzzy event 1" is a transition between fuzzy state1 to fuzzy state2. The physical evidence for this fuzzy event could be a transition from L_1=1 to L_1=0.

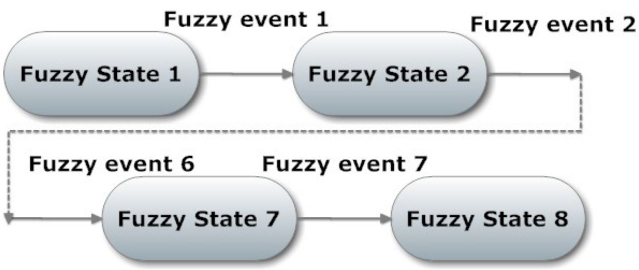

Fig 4. Conceptual graph of butterfly network.

The conceptual structure in figure 4, illustrates a normal scenario that could happen in the context of (A=1, B=1). The discovered knowledge has three applications. Firstly, if one of the mentioned fuzzy states during the testing time is recognized it means that we could find an explanation for the world state and the world state would be recognized as the *normal*. Secondly, if such sequence of fuzzy events and fuzzy states are observed then it can be inferred that a normal *scenario* for the network is happened. Thirdly, we can *predict* the network's behavior if primary states of the scenario are recognized, because the possible sequence of events that could occur in future are already known.

The learned fuzzy rule base is the heart of the proposed approach. The more it learns the more it would be precise in judgment to reason about the normality of world state. As it can be seen, the discovered knowledge from a network is completely abstracted from physical entity of the network and only the behavior of network is mentioned.

According to the learned knowledge, the world can be actuated. Recovery of network, extension of network and improvement of network functionality are some ways to actuate the world. The knowledge discovered from temporal data mining provides the basic criteria to make decision to actuate the world. The mentioned phrases represent specifications of a dynamic network that can automatically reform its structure to adapt well to the new situations. The goal of a dynamic network is to achieve the higher normality of the world. As it has already learned the normal states of the world through a data-driven approach, then it can be said that it is able to reform itself without interference of expert. In the next sections, we would explain two ways to actuate the world (reform a network structure) based on the learned knowledge[13]. In fact, we would intend to explain how to interpret the knowledge to actuate the world.

F. Network anomaly detection applying non-linear behavior factor

Links of networks carry values that are usually in a statistical relationship to other links. For example, a link may always carry the same value of another link while it is in contract to the values of another link. This is calculated by summarizing the correlation and dependence of each link and we call it non-linear behavior factor. For each link this comparison can be done and if a link is permanently in an abnormal non-linear behavior then it can be inferred that it there is a problem with it. Comparing all the links of the butterfly network to each other, we can make a fuzzy controller to take care about each link. According to the available knowledge in table 2, we can have the non-linear behavior factor for thirteen links of the butterfly network existing in table 3.

TABLE 3
Correlation of the butterfly links

	var1	var2	var3	var4	var5	var6	var7	var8	var9	var10	var11	var12	var13
var1	1.0												
var2	0.0	0.0											
var3	0.82572	0.0	1.0										
var4	0.70065	0.0	0.84853	1.0									
var5	0.0	0.0	0.0	0.0	0.0								
var6	0.0	0.0	0.0	0.0	0.0	0.0							
var7	-0.52223	0.0	-0.63246	-0.74536	0.0	0.0	1.0						
var8	-0.45227	0.0	-0.54772	-0.6455	0.0	0.0	0.86603	1.0					
var9	-0.38925	0.0	-0.4714	-0.55556	0.0	0.0	0.74536	0.86066	1.0				
var10	0.33029	0.0	0.4	0.4714	0.0	0.0	-0.63246	-0.7303	-0.84853	1.0			
var11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
var12	0.27273	0.0	0.33029	0.38925	0.0	0.0	-0.52223	-0.60302	-0.70065	0.82572	0.0	1.0	
var13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

In the consequence, we represent the non-linear behavior values of each link in the table 4.

[13] If a network reforms its structures automatically, then it can be inferred that it acts as a dynamic system.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE 4
The non-linear behavior factor values of each link

L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}	L_{13}
4.47	0	5.03	5.32	0	0	5.66	5.70	5.56	3.23	0	4.64	0

By fuzzifying these values by fuzzy subtractive clustering method, we would make a fuzzy controller to control the expected non-linear behavior of each link. The fuzzy classes in figure 5 represent the expected non-linear factor of the links for the fuzzy controller.

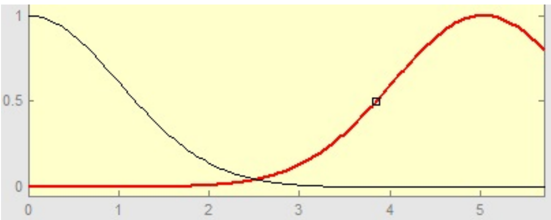

Fig 5. Fuzzy Classes representing the non-linear behavior factor

According to the available information in figure 5, it can be inferred that there are two noticeable groups of links that act in the network from the linear action viewpoint. If a link that is originally a member of the red class such as L_8 moves to the lower values then it can be inferred that this link does not act normally.

III. NETWORK IMPROVEMENT

We proposed to consider the knowledge as the structure of the network. In this way according to the available knowledge, the network tries to rearrange its physical structure to achieve the desired behavior, which can lead to improve the network throughput.

"Network recovery", "network extension" and "network improvement" can be some applications of network modeling. Here, in this paper we would introduce network functionality improvement as an application of our approach. Although network improvement is expected by the proposed approach, it may not necessarily lead to network optimization. To explain conceptually and briefly this application we use graphical shapes.

Considering that we know a basic but efficient network behavior such as a linear butterfly network, we can look at the more complicated networks behaviors to find where and how applying a collection of linear networks can improve the functionality of the complicated networks. To do that the rules of each simple model are drawn like figure 6.

The figure 6 represents the behavior of two especial links of a high performance network (butterfly) and we know already that it is an efficient network. At the other hand we can model a non-linear and complicated behavior. Figure 7, illustrates the behavior of two links of a non-linear network[14].

[14] The networks that apply memory-based and CPU- based devices such as routers to route the data packets.

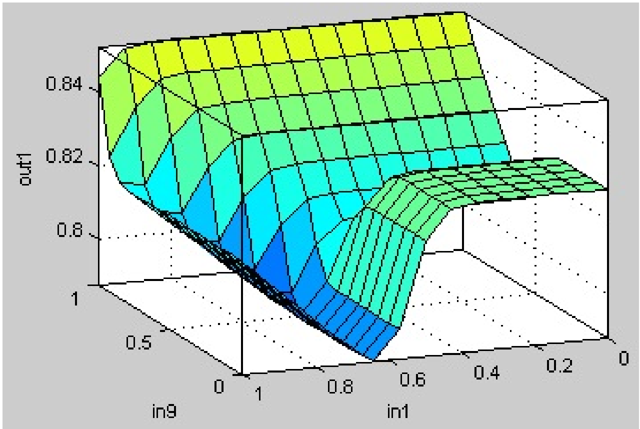

Fig 6. Surface of the rule resulted from comparison of Link 1 to Link 9 in butterfly network

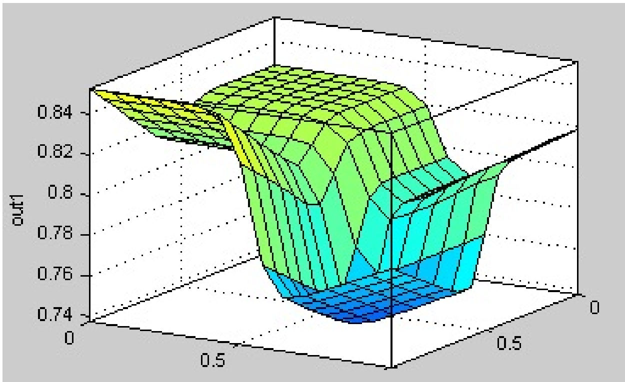

Fig 7. Surface of the rule resulted from comparison of two links of a non-linear network

Comparing two surfaces of links of two networks, we could see that the surface which is concerning to the butterfly network, can include the surface of the non-linear network. In other words, each point in figure 6 with ($X_6 = x_6$, $Y_6 = y_6$, $Z_6 = z_6$) has the *equal value* or *smaller value* in the figure 7 with ($X_7 = x_6$, $Y_7 = y_6$, $Z_7 \geq z_6$). In this way if we could find thirteen links from the non-linear network that their concerning rules are included with the butterfly network then, it can be inferred that the butterfly network can act the same context and probably it would do the same job in a more efficient manner.

As additional information, it should be said that if the data of table 2 be inputted to a router-based network instead of an XOR-based network, the *similar surfaces* such as the one in figure 6 would be created. Figure 7 illustrates the behavior of a router-based network, when the sources target more destinations. Anyway, the butterfly network behavior shouldn't be recognized as abnormal or erroneous in the non-linear network, if so (if we find some points that the router-based network does not recognize the butterfly network behavior as normal behavior) it is not the desired case of butterfly this network. If necessary conditions of the butterfly network are provided (described in the past paragraph), then we can define some contexts that a temporary substitution with butterfly network would be beneficial for the network. This will arise a bigger question which is concerning to dynamic network or dynamic structure and we would explain it in our future paper.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. CONCLUSION

In the proposed approach, the behavior of a coded network as a sample of a dynamic system is modeled and predicted. The knowledge to reason in normality of the world state (network functionality) is provided and provided some criteria to detect anomalies. The discovered knowledge provides information to actuate the world in the case of anomaly. Applications of the proposed approach are explained briefly. Network improvement is presented as one result of this research, in a way that we explained how to find (temporary) contexts for temporary substitution or temporary application of a more efficient sub-system.

V. FUTURE RESEARCHES

In this paper, we proposed how to model the behavior of networks, so according to the proposed research the network behaviors can be observed and learned. Considering that behavior of different simple linear networks is already discovered, we can propose replacement of linear methods for especial contexts of the non-linear networks according (This is explained already in part IV). In our future researches we would explain how to dynamically replace non-linear networks with linear operations based on the primitive observed contexts.

REFERENCES

[1] L. A zadeh, probability measures of fuzzy events, Journal Math. Anal. Appl., vol 23, pp. 421-427, 1968.

[2] T. Mitta, temporal data mining, CRC press, 2010.

[3] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. M´edard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, July 2006, pp. 1758–1762.

[4] R. W. Yeung, "Multilevel diversity coding with distortion," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 412–422, Mar. 1995.

[5] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[6] R. Koetter and M. M´edard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, ct. 2003.

[7] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. M´edard, and N. Ratnakar, "Network coding for wireless applications: A brief utorial," in *Proc. International Workshop on Wireless Ad-hoc Networks (IWWAN) 2005*, May 2005, invited paper.

[8] R. Koetter, "Network coding: a transportation vs. computation model", *lecture in Technische universitaet Muenchen*, information available in http://www.future-internet.eu/fileadmin/documents/consultation_meeting_31_Jan_08/Koetter_talk_Brussels_31_1_08.pdf

[9] A. Eryilmaz and D. S. Lun, "Control for inter-session network coding," in *Proc. of Workshop on Network Coding, Theory, & aplications (NetCod), San Diego*, Jan 2007.

[10] R. Yeung, S. Li, N. Cai, Z. Zhang, " network coding theory ",*foundations and trends in communications and information theory*, Now Publishers Inc (June 16, 2006).

[11] D. Chiu, "Network coding and multimedia content delivery", *information available in: http://home.ie.cuhk.edu.hk/~dmchiu/E-Letter_feb10.pdf*

[12] Information available in: http://www.mathworks.com/help/toolbox/fuzzy/fp351dup8.html

[13] Information available in: *http://liara.uqac.ca/index.htm*

[14] L. Baum, T. Petri Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains", *The Annals of Mathematical Statistics* **37** (6): 1554–1563:10.1214/aoms/1177699147, 1996.

[15] Z. Ghahramani, M. Jordan , "Factorial Hidden Markov Models". *Machine Learning* **29** (2/3): 245–273:10.1023/A:1007425814087,1997.

[16] D. Heckerman, "Tutorial on Learning with Bayesian Networks". In Jordan, Michael Irwin. *Learning in Graphical Models*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: MIT Press. 1998. pp. 301–354. ISBN 0-262-60032-3, 1997.

[17] J. Ross Quinlan and Joydeep Ghosh, Top 10 Algorithms in Data Mining, IEEE international Conference on Data Mining (ICDM) in December 2006.