SRLG Identification from Time Series Analysis of Link State Data

Goutam Das*, Dimitri Papadimitriou[†], Bart Puype*, Didier Colle*, Mario Pickavet*, and Piet Demeester*

*Ghent University, Department of Information Technology (INTEC), Gaston Crommenlaan 8, B-9050 Gent, Belgium

Email: {goutam.das, bart.puype, didier.colle, mario.pickavet, piet.demeester}@intec.ugent.be

[†]Alcatel-Lucent Bell, Copernicuslaan 50, B-2018 Antwerpen, Belgium

Email: dimitri.papadimitriou@alcatel-lucent.com

Abstract-Failing to account for the set of links affected by a simultaneous dependent failure during the re-computation of the routing table entries leads to traffic losses until all failed links have been accounted in the re-computation of these entries. Instead, if the router learns about the existence of Shared Risk Link Groups (SRLGs) from the arriving pattern link state routing information, then decisions regarding SRLG failure can be taken promptly to avoid successive re-computations of alternate shortest paths across the updated topology. In this paper, we propose a mechanism to improve the router recovery time upon occurrence of topological link failures by detecting and identifying the existence of SRLGs from link state routing information exchanged in the routing domain. The proposed model first groups into events individual Link State Advertisements (LSAs) issued by different network nodes (routers) upon link state change; then, it combines this information to find temporal dependence among members of event groups. It further introduces a physical model interpretation derived from the application of the Weibull distribution, to determine the error on the joint probabilities of events resulting from the finite observation sample. This association allows binding the dependence of the identified groups comprising one or more events (associated to SRLG) on the corresponding estimated failure rate. Our simulation results show that the proposed technique to locally detect and identify SRLGs performs sufficiently well to trigger with enough confidence simultaneous routing table updates from the arrival of a reduced set of LSAs (ideally one).

I. INTRODUCTION

The idea of introducing a cognitive component as part of the control system of communication networks has been introduced recently to allow network elements such as routers to learn about their own behavior and environment over time, to better analyze problems, to adapt their decisions, and to tune their execution. By means of this component, nodes are expected to improve their performance and increase the total network utility without any external intervention [6], [10], [18]. Cognitive networks generally use various machine learning techniques to enable the routers to learn online about new network behavior under variable traffic patterns and topological events. The concept behind this idea is to make the network responsive enough to adapt to newer scenario (closed-loop control) instead of requiring manual intervention for the configuration of each steps implied by the observationanalysis-decision-execution sequence. Different kinds of applications that would benefit of machine learning have been

978-1-4244-8953-4/11/\$26.00 © 2011 IEEE

studied over the past few years like traffic anomaly detection and intrusion detection [12], as well as traffic-informed rerouting [16], [17].

In this paper, we propose to improve the router recovery time upon simultaneous occurrence of topological link failures by detecting and identifying the existence of shared risk groups (SRGs). The detection and identification of SRG is performed by applying machine learning technique to Link State (LS) routing information. SRGs define groups of network resources such as links or nodes that share a common source of failures and are thus likely to fail simultaneously. For example, two distinct logical links routed through the same optical channel or two physical links carried over the same duct would fail altogether in case of optical channel or duct failure, respectively. When groups of network resources correspond to links, SRGs are referred to as Shared Risk Link Groups (SRLG). Their properties can be summarized as follows: i) a link belongs at least to one SRLG, ii) a link belongs to more than one SRLG if and only if it crosses (at least) one of the resources covered by each of these SRLGs, iii) pair of links belonging to the same SRLG can belong individually to other (one or more) SRLGs. In this context, a set of one or more links fail simultaneously when their associated shared risk occurs.

A router distributes the state of its interfaces and adjacencies (link state) by flooding this information throughout the routing domain. The collected link states of all routers and networks forms the Link State DataBase (LSDB). Flooding refers to the process that distributes and synchronizes the LSDB between routers. Link State Update (LSU) packets provide the mechanism for routers to flood link states to their adjacent nodes (one hop further away from their point of origination). Each LSU packet may contain a set of distinct Link State Advertisements (LSA). When a link failure occurs, the LS routing protocol initiates advertisement of LSUs indicating link state change by the LSA(s) they contain. SRLG failures often trigger multiple topological link failures and all routers in the network take individual decision regarding each of the advertised LSA(s). Instead, if the router learns about the existence of SRLGs from the time sequence of LSAs, then decisions regarding SRLG failure can be taken promptly and simultaneously to avoid additional delay for choosing alternative shortest-paths (across the updated topology).

A. Our Contribution

In the literature, the detection and identification of SRLG is considered to be of prime importance in optical and in IP/Multi-Protocol Label Switching (MPLS) networks [5]. In these environments, SRLG information is used as part of the Constraint-based Shortest Path First (CSPF) algorithm for traffic engineered route computation (prior to data path establishment) in order to maximize link disjointness of data paths and so decrease their common failure probability [1], [3]. A similar observation can be applied to IP Fast ReRouting (FRR) as analyzed in [15]. In this case, simultaneous failures also results into unreachable address prefixes and invalidates pre-computed or post-computed recovery paths (when they do not account for simultaneous link failures).

The present paper focuses on connectionless forwarding environments where each node computes its routing table entries independently and performs forwarding decision individually for each datagram (i.e. data path does not require end-to-end provisioning before forwarding can take place). At first glance, it might seem that one can simply look at the current network configuration to derive the SRLG information. However, in reality, the IP network link topology is often constructed on top of different optical networks, i.e., IP links are carried over optical channels and links belonging to different optical network providers leasing fibers passing sometimes through the same physical duct. Moreover, optical networks are often supporting multiple IP networks. In this context, SRLG disjointness of forwarding paths¹ is a hard problem: i) two forwarding paths are disjoint with respect to an SRLG if and only if at most one of them covers this SRLG (i.e., if and only if one of these paths crosses one of the links belonging to that SRLG), and ii) two forwarding paths are disjoint with respect to a set of SRLGs if and only if the sets of SRLGs they individually cover are fully disjoint. As forwarding entries are derived from routing table entries computation, simultaneous link failures resulting from occurrence of a shared risk can trigger multiple successive routing table entries re-computation with current link state protocols. Each of these re-computations account for each link failure taken individually. Failing to prune the set of links involved by the SRLG failure during routing table entries re-computation leads to longer recovery time and thus, higher magnitude of packet losses compared to the situation where the set of links (associated to the SRLG failure) results in a single step for the re-computation of all routing tables entries affected by the failure. Indeed, failing to take into account the set of links affected by the SRLG failure leads to traffic losses until all failed links have been accounted in the re-computation of the routing table entries. Instead, if the router learns about the existence of SRLGs from the arriving pattern of LSAs, then decisions regarding SRLG failure can be taken promptly to avoid successive re-computations of alternate shortest paths across the updated topology.

For this purpose, in this paper, we develop a learning model



Fig. 1. LSA sequence for different link failures

and device a mechanism to locally detect and identify SRLGs in the network to significantly reduce the router recovery time upon failure occurrence that affects simultaneously more than one link. As depicted in Fig.1, temporal distance between successive LSA arrival patterns is used to statistically learn about the possible existence of SRLGs. By processing this information, we subsequently predict their occurrence so as to prevent successive shortest path tree re-computation upon shared-risk failure (that affects multiple links simultaneously). The result of the SRLG detection and identification phase can thus be used to trigger simultaneous routing table updates from the arrival of a reduce set of LSAs (ideally one). Evaluation of the gain in terms of recovery time resulting from the trigger of simultaneous routing table updates shows promising results (in terms of traffic losses reduction) depending on the SRLG failure scenario and topology properties.

The proposed approach relies thus on processing the incoming LS routing information to derive the set of links that show a temporal dependence. Our approach does not consist in maintaining the set of SRLGs per path and enforce path disjointness at computation time but instead maintain the set of SRLG associated to each link and prune the set of links from the LSDB before computation. The pruning operation consists in disabling from the LSDB, the set of links identified as belonging to a set of SRLGs. Failure of a given SRLG is declared once a small subset of member links belonging to that SRLG is early detected as being subject to a state change. This approach thus does not "enforce" path disjointness but mitigates effects of simultaneous link failures upon detection of a small subset of its affected members identified as belonging to the same SRLG. This identification is obtained by inferring the links belonging to SRLGs from incoming LS routing information.

The rest of this paper is organized as follows. We first outline related work in the domain addressed by this paper in Section II. Section III provides a detailed description of the proposed learning model and related algorithms. Section IV details the simulation scenarios and procedures developed to produce numerical results. These results obtained by applying our learning model are detailed in Section V that outlines the key findings of our SRLG detection and identification technique. Finally, Section VI concludes this paper.

II. RELATED WORK

To the knowledge of the authors, no method since so far has been introduced to detect SRLG from the temporal properties of the network control traffic such as link state protocol data. The SRLG information relies on manually configured local

 $^{^1\}mathrm{A}$ forwarding path is defined as an acyclic alternating sequence of IP routers and links from source to destination

entries that describes the underlying infrastructure layout [14]. These entries are then advertised to the entire network by associating an additional SRLG identifier attribute to the link state advertised by the routing protocol. This SRLG information is then used for SRLG disjoint path computation to improve the network protection and restoration mechanism. The other method commonly used to detect SRLG is based on topology and geographical location of physical nodes [19]. However, manually configured entries and geographical information are often difficult to access for legacy equipment which are already deployed in the field. In other words, none of these methods performs online detection and identification of SRLGs but instead rely on the a-priori knowledge of this information.

In this paper, we introduce a machine learning technique to process incoming LS routing information and analyze them against already available LS routing data in order to detect SRLGs. Though there has been some attempts to analyze LS routing information for SRLG identification [8] [9], they never considered the statistical information embedded in this data to detect distinct SRLGs comprising common member links (e.g. a given link part of two distinct SRLGs).

Henceforth, in [4], we have introduced a state space based Bayesian network model that enables a more complete and accurate probabilistic detection and identification of SRLGs. In terms of SRLG detection, the results obtained by means of this model are promising. However, [4] does not considered the temporal dependence among LSAs to further enhance the probabilistic prediction accuracy. In this paper, we introduce a novel algorithm to enhance the probabilistic Bayesian network model of [4] in order to detect and to identify SRLGs by considering the dependency among the inter-arrival time sequences of LSAs, in addition to their occurrence rate.

III. MODEL

In this section, we introduce the learning model for detecting and identifying SRLGs. We then describe the decision model used in association to the detection and identification phase for early prediction of SRLG failure occurrence (from the incoming routing update information). Before giving further details on the learning model used as SRLG detection and identification technique, we provide an example of network that will help us to better understand the proposed algorithm. Fig.2 shows a typical network topology with 6 nodes (routers) and 9 links (A, B, C, D, E, F, G, H, and I).

In the OSPF context, for instance, neighboring routers periodically exchange Hello messages to establish and maintain relationships (ensure two way communication) with their neighbors [11]. If a router does not receive any Hello message from its neighbor within a period of time equal to the RouterDeadInterval (that is of the order of seconds), it assumes that the link between itself and the neighbor to be down. The detecting router then originates a set of one or more new router LSA that describes the topological link state change(s). Once originated by the detecting router, this set of LSAs is reliably disseminated (flooded) by means of LSU message one hop further away from their point of origination. At each hop, a



Fig. 2. Network Topology

new LSU is generated that may contain the LSAs of several other routers. Any distant router, upon receiving new LSAs, installs them in the LSDB. Using the topological information of the LSDB as input, a router computes the shortest-path tree using itself as the root yielding a set of routes stored in the Routing Information Base (RIB). Forwarding entries are then derived from the RIB, stored in the Forwarding Information Base (FIB), and subsequently used to perform individual forwarding decision on incoming datagrams.

Multiple LSAs for the same link failure can be initiated by different routers. However, observant router can discard those repeated information and consider the first router LSA reflecting the topological change involving a particular link, and thus the LSU that comprises this LSA. With multiple link failures, the arrival time sequence defined by the series of received LSAs is depicted in Fig.1 that shows a typical LSA sequence for the same network with links D, B and H failing. To identify SRLGs, it is then required to find temporal dependence among LSA sequences. We start our algorithm by processing the sequences of arriving LSAs as time sequences; then, after determining their temporal locality, the algorithm groups LSAs into sets and declare their constituting elements (links) as part of the same SRLG. For example, links (B, A), (C, D, B), (A, H, C) and (A, B, D) forms four distinct SRLGs from Fig.1. We next form a probabilistic model to represent SRLGs in a more realistic fashion. Indeed, the temporal locality between the received LSAs is assumed to reflect the simultaneous failure of the links advertised in LSAs contained in the observed LSUs.

A. Learning Model: SRLG Detection and Identification Phase

1) Grouping Individual Events: In this phase, we construct a data structure that stores the detected groups of individual events. For each event (a received LSA), a list is maintained that determines its temporal dependence to other events that have been observed in the entire known history of events. For example, suppose that the LSA for link A has arrived at time t and we are to make a decision about which other links can be part of an SRLG containing A as a member link. We start the algorithm by grouping the event sequences as follows. We initially start with a time threshold which is referred to as the window threshold. The start time can be set to any reasonable time derived from the routing protocol constants (5 seconds in our case as determined by the MinLSInterval, the minimum time between distinct originations of any particular LSA²). This choice has on the other hand little impact on the adaptive algorithm. The window threshold is then adaptively modified as follows. We define T_{min} as the minimum value of timing threshold.

We initialize the algorithm by setting the threshold time (T_{th}) to T_{min} . When the first LSA arrives, the grouping algorithm starts; then, two cases are possible:

- Case 1: no more LSA arrives within T_{min} : the algorithm stops further grouping and keeps $T_{th} = T_{min}$.
- Case 2: $N(\geq 1)$ LSAs arrive within T_{min} : the algorithm increases the threshold time T_{th} as follows: $T_{th} = T_{min} + \left(T_{min} - \sum_{j=2}^{N} T_{inter-interval(j,j+1)}\right)$. In this formula, the term $\sum_{j=2}^{N} T_{inter,interval(j,j+1)}$ denotes the inter-arrival time between the j_{th} and the $(j + 1)_{th}$ arrival.

In the second case (i.e., when multiple arrivals of LSAs are observed), once the new threshold time T_{th} value has been computed and when the algorithm waits for further LSA arrivals during the added threshold time, two different scenarios can occur:

- Case 2a: No further LSA arrives as T_{th} elapses: If $\sum_{j=2}^{N} T_{inter_interval(j,j+1)} < T_{min}$, then T_{th} is set to T_{min} ; otherwise, $T_{th} = \sum_{j=2}^{N} T_{inter_interval(j,j+1)}$ and the algorithm stops grouping LSAs. The current T_{th} value is stored for the next phase of grouping.
- Case 2b: $N(\geq 1)$ LSA arrives within the extended T_{th} : T_{th} is further extended following the process described above for the second case (when N LSAs arrives within T_{min}) and the algorithm further waits for new LSA arrivals for this extended period.

The algorithm continues till there are no new LSA arrivals within the extended T_{th} period. Case 2a provides a mean for the algorithm to adaptively reduce the threshold time T_{th} value when there are no LSA arrivals within an extended T_{th} period. Case 2b ensures that the algorithm can adaptively increase the T_{th} period if more LSA arrivals are observed during the extended T_{th} period.

2) Time Dependence between Event Groups: In our example shown in Fig.1, links (B, A), (C, D, B), (A, H, C) and (A, B, D) forms individual event groups. To determine SRLG with link A as a group member, we next list out all the member links that share at least once the same event group as link A. Following our example, that list becomes (B, C, H, D). We next form a data structure containing all combinations possible among these links. In our example, four types of groups exist, each type is determined by the number of elements the group contains. In the first type, we consider groups of single element, i.e., (B), (C), (H), (D). The second type consists of all possible combinations with two elements

at a time, i.e, (B, C), (B, H), (B, D), (C, H), (C, D), (H, D). The third type consists of combinations with three elements, i.e., (B, C, H), (B, C, D), (B, H, D), (C, H, D) and finally the last type consists of the group containing all elements, i.e., (B, C, H, D).

The next task is to find the temporal dependence between the individual events containing A as single element and the event groups containing A as element. For the algorithm to run efficiently, we also need state variables for each of these event groups that can be updated progressively without keeping the entire history of data. The state variables associated with each event group are listed below:

- N_i: the number of events observed that include link l_i, where the integer i (1 ≤ i) indexes the concerned link. For convenience, we will identify each link (A, B, C, ...) by its link number i (1 ≤ i ≤ P), assuming that there is a total of P links in the network.
- 2) $N_{i,j,k}$: the number of events observed that include link l_i as part of an event group comprising j elements $(1 \le j \le P)$, where event groups are enumerated (per number of elements j) by the integer $k, k \ge 0$. For example, if j = 2, k can index the event group (B, C) or (B, D), and if j = 3, k can index event group (B, C, D) or (B, C, H).
- 3) $\sigma_{last,i,k}^2$: each time an event that include link l_i is observed, this value is updated as follows $\sigma_{new}^2 = \sigma_{old}^2 + (T_{current,i} - T_{last,i,k})^2$. In this formula, $T_{current,i}$ is the current time of observation of an event group that includes link l_i . $T_{last,i,k}$ is the last time at which an event group k including link l_i has been observed (as part of an LSA sequence). The absolute time of observation of a given event group k is the mean value of the arrival time of its individual component elements. For instance, the absolute time of observation of the event group (B, C, D) is the mean value of the arrival time of components B, C, and D.

Using these variables, the coefficient $\alpha_{i,j,k}$ between each individual event including link *i* and an event group *k* including *j* elements, is defined as follows:

$$\alpha_{i,j,k} = \begin{cases} \frac{N_{i,j,k}}{N_i} & \frac{\delta_t}{\sigma_{last,i,k}^2} & \text{for observed events} \\ 0 & \text{otherwise} \end{cases}$$
(1)

In Eq.(1), as the fraction $\frac{N_{i,j,k}}{N_i}$ is ≤ 1 , we can safely assume that it provides an estimation of the probability of occurrence of joint event *i* together with event group(s) *k* containing *j* elements. In this equation, the fraction $\frac{\delta_t}{\sigma_{last,i,k}^2}$ measures the temporal dependence between events including link l_i : higher the value of this fraction, higher the temporal dependence between joint events including link l_i and the event group(s) *k* containing *j* elements. The value δ_t is introduced to give to the coefficient α the dimension of a rate of observation per observation unit δ_t . As a result, the coefficients $\alpha_{i,j,k}$ are ≤ 1 , where each coefficient $\alpha_{i,j,k}$ measures the occurrence rate of dependent events including link l_i with event group(s) *k* containing *j* elements.

²Thus, two instances of the same LSA may not be originated within the time period MinLSInterval

Processing of Eq.(1) is performed as follows. To unobserved event groups k including event i, we assign the value 0 to their corresponding coefficient. Whenever a new sequence of LSAs arrives and an event group k is subsequently identified using the procedure described in Section III-A1, the $T_{last,i,k}$ value associated to the currently identified event group is updated. Upon observation of a given link l_i as part of the identified event group k, the value N_i is updated. If this link is included in the event group k, then the values $N_{i,j,k}$ and $\sigma_{last,i,k}^2$ are also updated.

3) Matrix Representation: We introduce a matrix representation of the event groups dependencies. This representation provides a structure that compacts the data storage and simplifies the search operation during the computation process. For this purpose, we define a $(P \times P)$ matrix, where P represents the number of links, as follows:

The elements of this matrix correspond to vector of weighting factors $\omega_i(j,k)$ introduced to weight the measured coefficient $\alpha_{i,j,k}$. Each factor $\omega_i(j,k)$ is defined as follows: *i* is the row number associated to link l_i (each element of a given row $i, 1 \leq i \leq P$, lists the event groups including link l_i , j is the column number where each column identifies event groups containing exactly j elements $(1 \le j \le P)$, and k indexes a given combination of j elements (links) containing the element *i*, where the total number of combinations per matrix element equals to $\binom{P-1}{j}$. In other terms, each factor $\omega_i(j,k)$ weight the observed coefficient $\alpha_{i,j,k}$ of link l_i with event groups k of exactly j elements (the number of links identified as SRLG member elements). For instance, the factor $\omega_1(2,1)$ weights the observation of link 1 (i = 1) with event groups including two SRLG member elements (j = 2) as part of the first combination k (k = 1) of two elements. All combinations of links are arranged in cyclic order and are numbered with kaccordingly once i and j are specified.

4) Weighting Factors: The weighting factors $\omega_i(j,k)$ are computed as follows. We define the weighted coefficients

$$\alpha_{i,j} = \sum_{k} \omega_i(j,k) \ \alpha_{i,j,k} \tag{2}$$

Each weighted coefficient $\alpha_{i,j}$ represents the combined value for each cell of the matrix. For normalization purposes, we impose that the weighting factors $\omega_i(j,k)$ verify

$$\sum_{k} \omega_i(j,k) = 1. \tag{3}$$

Each coefficient $\alpha_{i,j,k}$ measures the occurrence rate of event i when observing an event group k of j link elements. Therefore, each term $\omega_i(j,k) \alpha_{i,j,k}$ in Eq.(2) represents the individual contribution of the observation to the occurrence rate of joint events i together with any event group containing exactly j elements as member links of the corresponding SRLG.

As $\alpha_{i,j}$ represent combined value of dependence coefficient, we have the additional condition imposed on $\alpha_{i,j}$ values:

$$\alpha_{i,j} \le \Pi_{i,j} \tag{4}$$

$$\sum_{j} \alpha_{i,j} \le \Pi_i \tag{5}$$

$$\sum_{i} \alpha_{i,j} \le \Pi_j \tag{6}$$

Eq.(4)–(6) specifies that the sum of the normalized $\alpha_{i,j,k}$ in every element (cell), line, and column of the matrix is upper bounded by value $\Pi_{i,j}$, Π_i , Π_j respectively. The $\Pi_{i,j}$ values are derived from the assumption that any individual SRLG failure rate is determined by the failure rate of their associated two-parameters Weibull distribution [13]. Indeed, the Weibull distribution is commonly used to model and estimate the characteristics and behavior of equipment and systems such as their reliability or failure probability at specific time, and their failure rate variation over time. Each distribution is characterized by the scale parameter *b*, the shape parameter *c*, and by its Probability Density Function (PDF) that is given by

$$f(x;b,c) = \begin{cases} \frac{c}{b} \left(\frac{x}{b}\right)^{(c-1)} e^{-\left(\frac{x}{b}\right)^c} & x \ge 0\\ 0 & x < 0 \end{cases}$$
(7)

The coefficient $\alpha_{i,j,k}$ being computed from a finite sample of observation, the occurrence rate of joint events (involving *i* as part of event groups *k*) does not account for the contribution to the occurrence rate that these individual observations can not explain. This difference explains the inequality of Eq.(4).

For inequalities Eq.(5)–(6), additional boundary conditions are to be derived. The second boundary condition takes into account that, starting from j = 1, the error between event groups of j and $j + \mu$ elements is bounded by the μ_1, \ldots, μ_{th} moments of the individual Weibull distribution [13]. Thus, the error for each column j is bounded by the equation

$$\sum_{i} E_i^j = \Pi_j , \qquad (8)$$

where E_i^j is the j_{th} moment of the Weibull probability distribution for link l_i . In practice, if we use only the first moment of the individual Weibull distribution, it simply means that the error between each column j is determined by the mean distance between observations of link i in between event groups of j and j + 1 elements. The third boundary condition states that

$$\sum_{i,j} \left(\Pi_i + \Pi_j \right) = \Pi , \qquad (9)$$

This third condition means that the observed failure rate can't exceed the network failure rate. Therefore, the inequalities of Eq.(4)–(6) explains the difference between the measured and the actual occurrence rate (bounded by Π).

5) Physical Model Interpretation: To estimate this difference, referred to as the error contribution $\epsilon_{i,j}$, we introduce a physical model interpretation. This model will in turn enable to enforce the equality condition in order to solve all the factors $\omega_i(j,k)$ such that Eq.(3) is verified. To construct this model, we assume that the individual link failure probability follows a Weibull distribution [13]. For link l_i $(1 \le i \le P)$,

$$\begin{pmatrix} \omega_1(1,0) & (\omega_1(2,1),\omega_1(2,2),\dots,\omega_1(2,P)) & (\omega_1(3,1),\omega_1(3,2),\dots,\omega_1(3,C_j^{P-1})) & \dots \\ \omega_2(1,0) & (\omega_2(2,1),\omega_2(2,2),\dots,\omega_2(2,P)) & (\omega_2(3,1),\omega_2(3,2),\dots,\omega_2(3,C_j^{P-1})) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_P(1,0) & (\omega_P(2,1),\omega_P(2,2),\dots,\omega_P(2,P)) & (\omega_P(3,1),\omega_P(3,2),\dots,\omega_P(3,C_j^{P-1})) & \dots \end{cases}$$

let $F_i(x) = Pr(X_i \leq x)$ be the probability of failure up to time x, R(x) the reliability (or survival) function, and H(x) the hazard (or instantaneous failure) rate function. From the distinct observation of each link failure, we construct the individual Weibull distribution characterized by its Weibull parameters b (scale parameter) and c (shape parameter), and its representative moments. A set of links composed by jelements and belonging to the same SRLG survive as long as none of its *j* components has failed. However, as the assumption of independently failing components does not apply in the present context, i.e., failure times of the links are dependent, we need to introduce a multivariate Weibull distribution that is not simply the product of their marginal distributions. By defining the power transformations $X_i = Y_i^{c_i}, i = 1, \dots, p$, the multivariate Weibull distribution introduced in [2] has the joint survival distribution R(y) with parameters $\kappa \geq 0$ and $\nu > 0$, and individual failure rate λ_i defined as

$$\lambda_i \equiv \lambda(y_i) = \frac{c_i}{b_i} \left(\frac{y_i}{b_i}\right)^{(c_i-1)} > 0 \tag{10}$$

$$R(y) = exp\left\{\kappa^{\nu} - \left[\kappa + \sum_{i=1}^{p} \lambda_{i} y_{i}^{c_{i}}\right]^{\nu}\right\}$$
(11)

Using this physical model interpretation, the error contribution $\epsilon_{i,j}$ accounts for the difference between the observed occurrence rate that can be explained from the observation of joint events including link l_i and the rate that results from the link failure dependency that is not observable from our finite observation sample. Once these multivariate Weibull distributions are evaluated, we can estimate the errors resulting from the finite sample of observation used to derive the $\alpha_{i,j,k}$ coefficients as follows:

$$\epsilon_{i,j} = \sum_{k} \left(H_{i,j,k} - \alpha_{i,j,k} \right), \tag{12}$$

where $H_{i,j,k}$ is the instantaneous failure rate at time x associated to the joint survival function $R(x) = Pr(X_1 > x, \ldots, X_j > x)$. We can therefore modify the Eq.(4)–(6) as follows:

$$\alpha_{i,j} + \epsilon_{i,j} = \prod_{i,j} \tag{13}$$

$$\sum_{j} \left(\alpha_{i,j} + \epsilon_{i,j} \right) = \Pi_i \tag{14}$$

$$\sum_{i} \left(\alpha_{i,j} + \epsilon_{i,j} \right) = \Pi_j \tag{15}$$

$$\sum_{i,j} \left(\Pi_i + \Pi_j \right) = \Pi \tag{16}$$

The evaluation of this system of equations Eq.(13)–(16) requires the computation of the error contribution terms $\epsilon_{i,j}$ in order to solve all the factors $\omega_i(j,k)$ such that Eq.(2)–(3) are verified.

B. Decision Model: SRLG Early Identification

In this phase, we compare the value of ω with a predefined threshold ω_{th} . For a particular link *i*, we search through matrix the row of the column (associated with that link) until we hit all zero weighting factors $\omega_i(j, k)$. We then search for the highest weighting factor within the vector positioned just before the vector of zero weighting factors. We declare all the member links of the corresponding factor as members of an SRLG including link *i*.

IV. SIMULATION

Our experimental setup is based on GEANT2 network topology [7] which comprises 53 physical links. We create a logical topology of 80 links on top of the GEANT2 physical topology and accordingly define SRLG with logical links passing through the same physical link. We generate the time sequence of input LSA for such a network using two different scenarios as described in Section IV-B. We use the MATLAB numerical computing environment and programming language to generate these time sequences and then, we run our algorithm on these time sequences to validate our model.

A. Simulation Procedures

The numerical evaluation of the algorithm as described in Section III is performed as follows:

- The occurrence rate of dependent link failure events $\alpha_{i,j,k}$ are computed iteratively from the observed data using Eq.(1), where $\alpha_{i,j,k}$ is dependent on N_i , $N_{i,j,k}$ and $\sigma_{last,i,k}^2$. If a new SRLG is formed due to a change in the topology involving link l_i and the corresponding member links of k event groups, then the counter $N_{i,j,k}$ is incremented. Moreover, a small finite value of $(T_{current,i} T_{last,i,k})^2$ is added to the otherwise zero valued σ_{old}^2 which will slowly increase the values of $\alpha_{i,j,k}$ as the number of observed occurrences of such simultaneous failures increases. Thus, the algorithm is able to automatically detect and identify the presence of new SRLGs due to a change in the topology. Similarly, if an existing SRLG is seize to exist, the algorithm will automatically reduce the value of the corresponding $\alpha_{i,j,k}$ to reduce the eventual probability of its existence.
- The boundary parameters Π_{i,j} are computed using Eq.(7), assuming that any individual SRLG failure is determined

by the failure rate (see Eq.(10)) of its associated twoparameters Weibull distribution. Appropriate distribution fitting function is used to estimate the scale (b) and shape (c) parameters of each Weibull distribution. Moreover, for each j such that $1 \le j \le P$, the j_{th} moment of each Weibull distribution is computed using the moment generating function [13], which enables the computation of parameters Π_j by means of Eq.(8).

- To compute the error contributions ε_{i,j} using Eq.(12), the joint survival function R(y) is generated using Eq.(11), where b_i and c_i represent the parameters of the individual Weibull distribution associated to each link failure distribution. In Eq.(11), the value of p is set to the number of individual Weibull distributions to be considered (in the matrix representation, p is taken to be equal to the value of j for the evaluation of the weighting factors ω). Moreover, for this equation, we set ν = 1, 9 as we derived from numerical execution that this value provides for the lowest SRLG prediction error.
- The mean joint failure rate, the joint probability as well as the mean value of joint events are computed by performing discrete multi-variable integration over R(y)(see Eq.(11)) so that events corresponding to the link failures are within a predefined threshold (*th*) from each other. The result of this integration (see Eq.(17)) yields $\bar{y_p}$, the mean distance between failures for the joint event of *j* link failures within a time period delimited by the predefined threshold *th*.

$$\bar{y_p} = \int_0^\infty \int_{y_p-th}^{y_p+th} \dots \int_{y_p-th}^{y_p+th} y_p R(y) dy_1 dy_2 \dots dy_p$$
(17)

Using Eq.(17), the analytical failure rates are given by

$$H_{i,j,k} = \frac{1}{\bar{y_p}} \tag{18}$$

• The set of equations Eq.(2), Eq.(3), Eq.(13)–(16) are used to compute all the weighting factors ω . However, only those factors that are related to the observed link failures are considered. This resolves the problem of having more variables than equations when $j \geq 2$. Moreover, if any nonzero ω value exists that does not correspond to an observation within the current phase (of SRLG observation) by applying the LSA grouping algorithm, these ω values are first scaled to incorporate appropriate weighting.

B. Simulation Scenarios

As obtaining real failure data set is a difficult task (due to the fact that failure events are usually not disclosed), we create a model to generate SRLG failure data set that closely represents the real simultaneous failure scenario. For this purpose, we assume that the link failure or any SRLG failure follows a Weibull distribution (see Eq.(7)), where, b is the scale parameter and c is the shape parameter. When c > 1, the failure rate increases with time, which resemble the realistic network scenario. For our purpose, we set c = 1.5 and b to a very higher value to account for the fact that network failure are usually sparse events.

Moreover, we distinguish two cases depending on the dominant delay factor in the origination and propagation of LSAs after the occurrence of time dependent failures (resulting from the topological dependence):

- Case 1: Hello message and RouterDeadInterval detection dominates. Suppose that two distinct LSAs for two different links associated to the same SRLG, are originated by two different routers (in particular, when these LSAs are flooded as part of different LSUs). The maximum delay these LSAs can experience at the origin is due to the desynchronization of Hello messages. Indeed, these periodic messages are originated independently by each router within a time interval that ranges in the order of seconds (typical values range from 1 to 10 seconds). At the higher inter-arrival delay between two LSAs resulting from the same SRLG failure, the Hello interval desynchronization dominates the propagation or queuing delay of LSAs. So for higher inter-arrival delay, we can assume a uniform distribution for the LSA inter-arrival time because the origination of Hello messages by routers is completely random and independent between routers.
- Case 2: Queuing, Transmission, and/or Propagation dominates. When the desynchronization due to Hello messages is negligible, LSAs propagation and queuing delays dictate the cause for delay variance. Due to the accumulation of multiple mutually independent random queuing delays, we can assume without loss of generality, that the inter-arrival time between LSAs are memory less and demonstrate exponential distribution. Our simulations show that the assumption of exponential accumulated delay distribution has negligible effect on the results presented in this paper. We test long tailed distributions that have power law decay to verify that different interarrival time distributions show minimal effect on the final result. This is due to the fact that failures are rare events that are usually separated by a large amount of time. Generally, the inter-failure occurrence time is much larger than the inter-arrival time between two LSAs for the same SRLG failure to create any impact whatsoever. For the propagation delay however, we use realistic values that correspond to the GEANT2 network topology.

With this specifically generated input data file that include the time sequences of LSA arrivals, we run the LSA grouping algorithm as described in Section III-A1. Once the LSAs are grouped, we run the proposed learning model as well as the decision making algorithm to predict the existence of SRLGs. We compare our results from the prior knowledge of SRLGs from our assumed network topology and compute the amount of false positive and true negative the algorithm generates.

V. SIMULATION RESULTS

In this section, we detail the simulation results obtained using the GEANT2 network topology (except when stated otherwise). Before presenting these results, we provide here



Fig. 3. Value of weighting factor ω

below some intermediate results like the computation of the weighting factors $\omega_i(j,k)$ and the computation of the occurrence rate as measured by the α coefficients in order to get better insight regarding the execution of our algorithm and its accuracy.

Fig.3 provides the 3-dimensional plot of the computation of the $\omega_i(j,k)$ values for link 1, i.e. i = 1, forming SRLGs with two member links, i.e., j = 2. The x-axis indicates the number of two link combinations, k, with link 1 as one of its member links (for instance, (1,2), (1,3), (1,4), and (1,5)). The y-axis indicates the iteration time epoch of the algorithm which is incremented linearly with every arrival of an SRLG group indication. Thus, the z-axis provides the value of $\omega_{i=1}$ (i=2,k) with respect to the iteration time epoch of the algorithm and event group combination k when i = 1and j = 2. Here, we assume that link 1 forms 4 SRLGs (comprising two member links) with 4 distinct links; thus, the number of two link combinations k = 4. From Fig.3, we can observe that only the values of k = 1, 2, 3, 4 have a significant contribution to the weighting factors $\omega_1(2,k)$ and as the time progresses these contributions stabilize towards the theoretical values they should attain to represent their occurrence probability (i.e., 0.25). Note that the same type of figure has been generated for higher values of the number of member links *j*. However, with higher number of member links per SRLG, as a smaller number of event group combinations k show actual contribution to the ω values, it is hardly possible to observe them in such figure.

Fig.4 provides the 3-dimensional plot of α value (which measures the occurrence rate of dependent failure events) with respect to time and j, the number of member links per SRLG. As we have SRLGs with five member links (j = 5) that include link 1 (i = 1), the coefficient $\alpha_{i=1,j}$ should have non-zero values for j = 1, ..., 5. For better clarity, we ignored the values of j > 10 as all α values are equal to zero for these values of the parameter j. Again, we can observe that as time progresses the α value converges to the rate of dependent failure events (for our example, this is 0.0003 failures per day). The algorithm converges to this rate pretty fast for j = 2 and j = 3. However, as shown in Fig.4, the prediction accuracy





Fig. 5. Percentage of false positive and true negative with number of failure iterations (disjoint SRLGs)

decreases as j increases.

Fig.5 plots the percentage of false positive, the percentage of true negative, and the total prediction error as the number of failures per SRLG increases. These percentage values are computed by averaging the percentage values obtained from multiple random experiments performed after using a certain number of failure data for each SRLG to learn the state space. This experiment was carried over a set of disjoint SRLGs where the member links of each SRLG are not part of any other SRLG.

We follow the same procedure for the experiment when SRLGs share common member links. Fig.6 provides the results for multiple SRLGs having one common member link. This figure plots the results obtained when considering one common member link between two SRLGs. We can easily see that the performance degrades due to the interconnection between SRLGs. It can be observed that the algorithm produces both false positive and true negative more often when SRLGs share common member links. However, the percentage of false positive and true negative decrease as the number of failures per SRLG increases. This decrease in percentage can be explained as follows: as the algorithm gathers more statistical input regarding different SRLG failures, it progressively learns how to better predict future occurrences. We further increase the number of common member links between SRLGs from 1 to 5. From Fig.7, we observe further degradation of the SRLG prediction errors (%)



Fig. 6. Percentage of false positive and true negative with number of failure iterations (SRLGs with one common link)



Fig. 7. Percentage of SRLG prediction error with the number of common member links per SRLG

prediction error when this number increases. However, as the number of failure iterations increases, the prediction error becomes smaller than 10 percent when SRLGs have 5 member links in common.

Fig.8 shows that the variation of network delay distribution has negligible effect on our algorithm and the produced results.

We also measure the accuracy of the algorithm as the number of member links per SRLG increases. Fig.9 plots the percentage of error after 10 failure iterations for a number of links per SRLG ranging from 1 to 6. From this figure, it can be observed that the accuracy of the algorithm is very good for j = 2 or j = 3 member links per SRLG with ν parameter set to 2. However, the SRLG prediction error increases as the number of link per SRLG increases further.



Fig. 8. Effect of accumulated delay distribution on the performance of our algorithm

SRLG prediction errors (%)

Fig. 9. Percentage of SRLG prediction error with number of member links per SRLG

SRLG prediction errors (%)



Fig. 10. Percentage of false positive and true negative with number of failure iterations (disjoint SRLGs), $\nu = 3$

The percentage of error in SRLG prediction with respect to time (in terms of failure iteration) when setting the parameter ν to 3 in the multivariate Weibull distribution evaluation of Eq.(11) is shown in Fig.10. By comparing Fig.10 with Fig.5 obtained with $\nu = 2$, we conclude that i) setting $\nu = 2$ is sufficient for the algorithm to provide reasonably accurate SRLG prediction, and ii) increasing the parameter ν to 3 for instance, does not further increase the accuracy of our algorithm.

VI. CONCLUSION

In this paper, we have proposed a technique to improve the router recovery time upon occurrence of topological link failure by detecting and identifying the existence of SRLGs from link state routing information exchanged in the routing domain. The proposed model first groups into events individual LSAs originated by different network nodes (routers) upon link state change; then, it combines this information to find temporal dependence among members of event groups. It further introduces a physical model interpretation derived from the application of the Weibull distribution, to determine the error on the joint probabilities of events resulting from the finite observation sample. This association allows binding the dependence of the identified groups of one or more events (associated to SRLG) on the corresponding estimated failure rate. The result of the SRLG detection and identification phase is subsequently used to trigger simultaneous routing table

updates from the arrival of a reduced set of (ideally one) LSA.

The simulation results show that the SRLG prediction resulting from the execution of our algorithm provides enough confidence when the number of links per SRLG is relatively limited. Further improvement is expected from better adjustment of the κ parameter in the multivariate Weibull distribution to better account for time threshold effects. Nevertheless, the evaluation of the gain of the proposed SRLG detection and identification technique shows promising results with different SRLG failure scenarios and realistic network topology. To extend its applicability, future work includes execution of the algorithm with other simultaneous failure scenarios such as geographically correlated link failures, and evaluation of the model in other environments such wireless (meshed) networks.

ACKNOWLEDGMENT

This research work is (partially) funded by the European Commission (EC) through the ECODE project (INFSO-ICT-223936) part of the European Seventh Framework Programme (FP7).

REFERENCES

- [1] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*, Kluwer Academic, 1999.
- [2] M. Crowder, A Multivariate Distribution with Weibull Connections, Journal of the Royal Statistical Society, Series B (Methodological), Vol.51, No.1, pp.93–107, 1989.
- [3] D. Coudert, S. Perennes, H. Rivano, and M.-E. Voge, *Shared risk resource groups and survivability in multilayer networks*, Proceedings of IEEE/COST 293 annual conference on GRAphs and ALgorithms in communication networks, Vol.3, pp.235–238, June 2006.
- [4] G. Das, D. Papadimitriou, W. Tavernier, D. Colle, T. Dhaene, M. Pickavet, and P. Demeester, Link State Protocol data mining for shared risk link group detection, Proc. of 19th International Conference on Computer Communications and Networks (ICCCN 2010), Zurich, Switzerland, August 2010.
- [5] R. Doverspike and J. Yates, *Challenges for MPLS in optical network restoration*, IEEE Communications Magazine, Vol.39, No.2, pp.89-96, February 2001.
- [6] C. Fortuna, M. Mohorcic, *Trends in the development of communica*tion networks: Cognitive networks, Computer Networks, Vol.53, No.9, pp.1354–1376, June 2009.
- [7] GEANT2 Topology.
- Available at url:http://www.geant2.net/server/show/nav.00d007009.
- [8] R.R. Kompella, J. Yates, A. Greenberg, A.C. Snoeren, *IP Fault Localization Via Risk Modeling*, 2nd Symposium on Networked Systems Design Implementation (NSDI) 2005, Boston (MA), USA, May 2005.
- [9] R.R. Kompella, J. Yates, A. Greenberg, A.C. Snoeren, *Detection and localization of network black holes*, Proceedings of IEEE INFOCOM 2007, Anchorage (Alaska), USA, May 2007.
- [10] Q. Mahmoud, Cognitive Networks: Towards Self-Aware Networks, John Wiley and Sons, 2007, ISBN 9780470061961.
- [11] J. Moy, OSPF Version 2, RFC 2328, STD 54, Internet Engineering Task Force (IETF), April 1998.
- [12] P. Owezarski, J. Mazel, and Y. Labit, *Oday anomaly detection made possible thanks to machine learning*, 8th international conference on Wired/Wireless Internet Communications (WWIC'2010), Lulea, Sweden, June 2010.
- [13] H. Rinne, *The Weibull Distribution: A Handbook*, CRC Press, Taylor Francis Group, November 2008.
- [14] P. Sebos, J. Yates, D. Rubenstein, and A. Greenberg, *Effectiveness of shared risk link group autodiscovery in optical networks*, Proc. of IEEE/OSA Conference on Optical Fiber Communications, pp.493-495, March 2002.
- [15] M. Shand, S. Bryant, A Framework for Loop-Free Convergence, RFC 5715, ISSN: 2070-1721, Internet Engineering Task Force (IETF), January 2010.

- [16] W. Tavernier, D. Papadimitriou, D. Colle, M. Pickavet, and P. Demester, *Optimizing the IP router update process with traffic-driven updates*, Design of Reliable Communication Networks (DRCN) 2009, Washington D.C., USA, October 2009.
- [17] W. Tavernier, D. Papadimitriou, D. Colle, M. Pickavet, and P. Demeester, Using AR(I)MA-GARCH models for improving the IP routing table updates, 2nd International Worskhop on Reliable Networks Design and Modeling (RNDM) 2010, Moscow, Russia, October 2010.
- [18] R.W. Thomas, L.A. DaSilva, A.B. MacKenzie, *Cognitive networks*, Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Baltimore (MD), USA, November 2005.
- [19] G. Xiao and X. Pan, Heuristic for the Maximum Disjoint Paths Problem in Wavelength-Routed Networks with Shared Risk Link Groups, OSA Journal of Optical Networking, Vol.3, No.1, pp.38–49, January 2004.