Dynamic Risk-Aware Routing for OSPF Networks

Bruno Vidalenc
Alcatel-Lucent Bell Labs, Nozay, France
Bruno.Vidalenc@alcatel-lucent.com

Ludovic Noirie
Alcatel-Lucent Bell Labs, Nozay, France
Ludovic.Noirie@alcatel-lucent.com

Laurent Ciavaglia
Alcatel-Lucent Bell Labs, Nozay, France
Laurent.Ciavaglia@alcatel-lucent.com

Eric Renault
Telecom SudParis, Evry, France
Eric.Renault@it-sudparis.eu

***Abstract*—Carrier networks are designed to provide high availability of communication services. Unfortunately, in case of failure, recovery mechanisms are getting involved only after a failure occurrence which cannot prevent a certain impact on traffic flows. However, there are often forewarning signs that a network device will stop working properly. Based on an embedded and real-time risk-level assessment, a proactive fault-management can be performed to isolate failing routers out of the routed topology, and thus totally avoid detrimental impact on the service availability. Our novel approach enables routers to preventively steer traffic away from risky paths by temporally tuning OSPF link cost. The consequences in terms of stability and availability are estimated based on an analytical model and then simulated to measure the expected benefits of the proposed proactive self-healing function. Finally, the functionality has been implemented in an experimental prototype in order to validate the proof of concept.**

I. Introduction

The sustained demand for reliable services pushes network operators to provide always higher Quality of Service (QoS) and availability to remain competitive. Since availability is a strict commitment, network operators must deal with failures using protection and restoration mechanisms. Protection can recover a path extremely quickly, but is resource-consuming due to the needed spare capacity; while restoration can be much more inexpensive, but generates a longer outage. Moreover, even if performances are different, all these mechanisms are reactive and therefore cannot completely mask the effects of a failure on end user traffic. To improve this situation, our paper focuses on an extension of the IP restoration mechanism provided by the Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) and Intermediate system to intermediate system (IS-IS). The time between the occurrence of a failure and the end of the OSPF convergence implies packet loss, temporary routing loops or routing black holes that disrupt the operator's routing topology. Indeed, the recovery mechanism is only involved after the failure occurrence which only allows limiting the failure impact on traffic, but not removing it completely. We propose a complementary approach that removes this drawback by anticipating failures with a Risk Assessment Module (RAM), in order to override the slowness of recovery mechanism. This allows completely eliminating the failure incidence on traffic by performing reconfiguration actions few seconds before the failure. The failure prediction functionality is exploited to provide a Risk-Aware Routing (RAR) that dynamically adjust the link metrics to steer the traffic away from the risky network elements and therefore isolate failures before they cause damage on traffic. Many works have been done on the failure prediction task but none of them interest in how properly exploit such information. We differ by proving a realistic way to take advantage of a failure prediction, and evaluating the expected benefit for the operators, in order to justify the interest of implementing failure prediction functions into the network equipments.

The paper begins by first introducing IP restoration and its related works in Sec. II, then by describing the self-healing functionality in Sec. III, and by explaining its integration with the OSPF protocol in Sec. IV. The second part is dedicated to the evaluation with Sec. V that defines an analytical model to measure the effect of our mechanism on the network availability and the routing stability. Afterwards, the instantiation of this model on network examples and comparison with simulation experiments are analyzed in Sec. VI. To finish the Sec. VII describes an experimental prototype designed to test and verify the feasibility of the concept in a real network environment and the Sec. VIII ends the paper with the concluding remarks.

II. Context and related work

Current fault-management systems, especially in telecommunication infrastructures, handle outages in a reactive way, since it is the easiest way to address the problem. But a reactive approach involves acting only after the failure occurrence; hence it never allows removing all the failure effects. Indeed, current IP restoration inevitably leads to packet loss as long as the OSPF convergence process has not ended. This process is composed of 4 main phases [1]: a failure detection with t_D as the failure detection time, the Link State

978-3-901882-50-0

Advertisement (LSA) flooding with t_F as LSA flooding time, the Shortest Path First (SPF) computation with t_{SP} as SPF computation time, and the routing table and forwarding table update with t_U as table update time. The resulting convergence time t_C during which the OSPF routing topology is not consistent is calculated as follows: $t_C = t_D + t_F + t_{SP} + t_U$. As it stands, the downtime of flows affected by a failure due to the convergence process is of the order of several seconds, which is not acceptable for QoS traffic.

Previous works were focusing on reducing the duration of these phases. In particular, a well-known problem concerns failure detection time [2] where short timer settings lead to network routing instability and long timers increase the loss of traffic. The stability constraint imposes the operators to set reasonable timers of more than one second at the expense of fast recovery. Although subsecond failure detection had been considered as resolved by using link layer detection or hardware implemented BFD protocol in the data plane, these mechanisms are not always available and, most of all, do not allow to detect software failures present in the control plane. Unfortunately, software failures have become so important [3] that disregarding such failure is not possible anymore. In the other side, the IP Fast ReRoute (IP FRR) technique [4] provides a way to bypass the $(t_F + t_S + t_U)$ delays by pre-calculating certain backup paths, however the detection time t_D remains the most impacting and challenging factor.

We are studying a new, complementary approach, by adding a preventive mechanism to the current technique. This proactive approach evaluates the risk of failure in real time and creates a time window wherein preventive actions can be taken, for instance adapting the routing behaviour to avoid the detrimental impact of the forthcoming failure. Past research works have already proposed routing functions taking into account the risk of failure. In [5] long-term link failure statistics are used to choose the best path capable of satisfying a specific availability. We differ from these works by building a dynamic estimation of network elements risk of failure computed in real-time. After having explored the applicability to GMPLS recovery in a previous paper [6], this study concern pure IP routing that is a much more uncontrollable environment. Observing the equipment health enables to adapt the routing scheme, in conjunction with the evolution of the risk of failure.

III. A self-healing mechanism using risk-assessment

Despite all the efforts to improve the reliability of network infrastructure, their inherent complexity makes failures unavoidable. The origin of the failures has evolved [3], but some network monitoring techniques make some of them predictable: failure tracking, symptom monitoring, errors reporting or undetected errors auditing [7]. These techniques rely on the monitoring of multiple parameters of network devices that are already available in current network managers.

The origin of a network failure can be diverse and can be detected by appropriate means. First, one can cite hardware sensors such as the temperature of the processing unit, the power supply voltage information available via the Advanced Configuration and Power Interface (ACPI) or the hard drives status indicated by the Self-Monitoring, Analysis, and Reporting Technology (S.M.A.R.T.). We also have network performance indicators such as the bit error rate or packet loss rate measured in network interfaces. And the software malfunctions, that have become predominant [3], can be detected by their various symptoms like abnormal log entries, system errors, unreleased file locks, file descriptor leaking, data corruption, or abnormal usage of CPU, memory and I/O. Finally, communications with external systems, like Intrusion Detection System (IDS) or Network Management System (NMS) are also envisioned.

Based on the monitored data, the risk assessment module proposed in the Generic Autonomic Network Architecture (GANA) [8] use machine learning techniques [7] like Bayesian networks, time series analysis, Support Vector Machines or Semi-Markov to make predictions as accurate as possible. This risk information is then disseminated in order to create a time window preceding the failure to preventively prompt traffic to avoid risky links, which will prevent the end users traffic flows from the failure consequences. To get proactive rerouting of traffic flows and avoid the risky links, one can increase the OSPF cost of these links in order to make safer links preferable. Since networks dispose of spare capacities and are dimensioned to support failures [9], this will not cause congestion issue but only anticipate what will happen few minutes later if the failure occurs.

Any wrong detection will trigger useless rerouting and thus will create network instability. Therefore, the failure prediction must be reliable enough to avoid such behaviour. This paper does not focus on the evaluation of the failure prediction by itself, since many studies have concentrated their efforts on this aspect and good prediction can be achieved. What we need is to identify the characteristics of the failure prediction at the different time periods involved in the online failure prediction process (see Fig. 1). T is the time when the failure risk is detected. Δt_d is the data window time while the failure predictor conserves the data used to predict the upcoming failures. Δt_l is the lead time defined as the minimal period between the prediction and the failure event. The warning time Δt_w is the time required to set up proactive self-healing actions, which is always smaller than Δt_l. The prediction period Δt_p is the period during which the prediction is considered to be valid. Since the best failure prediction methods of the literature consider values of at least few minutes [10], we prefer to not be restricted with regards to the failure prediction method and to take a margin by considering a

Δt_p of one hour.

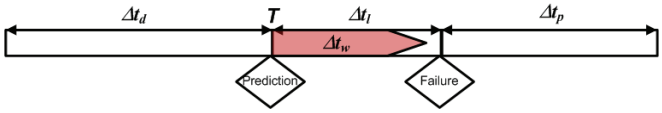


Figure 1. Time relation in online failure prediction.

The performance of the failure predictors is characterized by their unpredicted failures (FN) and their wrong predictions (FP). *Precision* and *Recall* are the usual metrics that are considered in the paper, which are related to the previous parameters:

$$Recall = \frac{TP}{TP+FN}\ ,\ Precision = \frac{TP}{TP+FP}. \quad (1)$$

Recall is the ratio of correctly predicted failures (TP) to the total number of actual failures. *Precision* is the ratio of correctly identified failures to the number of all predictions. Advances in the failure prediction field allow more than 90% for *Recall* and 80% for *Precision* [10]. As a consequence, 80% has been chosen as the most competitive value for both metrics.

Precision and *Recall*, as well as the prediction period Δt_p, are input parameters in Sec. V and VI that are used to analyze the failure prediction performances required to enable a useful Risk-Aware Routing. Indeed one must consider the uncertainty of the failure prediction, what can be done with *Recall* and *Precision*. But before evaluating our mechanism, we need to detail its functioning.

IV. Dynamic Risk-Aware OSPF link cost Assignment

What is proposed, is a risk-aware routing module that relies on a continuous computation and assessment of the failure risk to exploit the time window preceding the actual failure to preventively prompt traffic to avoid risky links, and in that way mitigating failure consequence on end user traffic flows. In OSPF networks, one can perform such proactive rerouting by tuning OSPF link costs in such a way that risky links will be avoided by the traffic flows. Many studies concentrate their efforts to put forward failure prediction mechanism but neglect that how exploiting this information is just as important. In consequence, we choose to repair this deficiency by proposing a pragmatic and extremely fast to activate proactive mechanism to exploit failure prediction and by evaluating the expected gains, as well as the constraints with regards to the failure prediction task.

The advantage of our method is twofold. First, it does not require any standardization process and it is fully compatible with currently deployed protocols by locally modifying links cost and exploiting the flooding capability of OSPF (potentially all link state protocols) to propagate the new risk-augmented scheme to the entire network and its traffic flows. Secondly, it is crucial to not change the current practices used by network operators to engineer their networks, by configuring links cost as previously, in order to optimize bandwidth, delay, load balancing, etc. The preventive intervention is kept restricted to a short period of time where an important risk of failure is detected, and only concerns the weight value of the set of risky links. Our approach allows combining operator-defined metrics most of the time and risk-level metrics only in case of failure anticipation, which trade-off and priority can be managed via operator-defined policies. The method consists in keeping the cost of the reliable links low (i.e. preferred route), and increasing significantly the cost of the risky links in order to steer traffic away from them towards more reliable low cost paths. For that purpose, costs of risky links must be assigned under many restraints that have been previously detailed in [11]. This risky value is equal to $MaxPossibleCost - MaxInitialCost + InitialCost(link_i)$ with $MaxPossibleCost$ the highest metric that is possible with OSPF (*i.e.* $2^{16}-2$), $MaxInitialCost$ the highest link metric of the network topology and $InitialCost(link_i)$ the initial metric of the risky link i configured by the operator. Lastly, once the risky cost values have been computed, the process in charge of switching the OSPF link cost from the initial configuration to the risky value shall do it in a smooth and iterative way as described in [12] in order to avoid routing loops during the OSPF convergence process. Next, we develop an analytical model describing the mechanism characteristics.

V. Analytical modelling

A. Notations

The network is modelled as a directed graph $G=(N,E)$ where N is the set of nodes (routers) and E the set of directed edges (links). We define the set F of traffic flows to be transported by the network G, where each traffic flow $f \in F$ is defined by its ingress node $In(f) \in N$, its destination node $Out(f) \in N$, and its throughput $\mu(f)$. For each traffic flow $f \in F$, the routing protocol (OSPF or IS-IS) defines the shortest path by the subset $sp(f) \subseteq N$ which contains the ordered list of transit routers. With the RAR mechanism, after a failure prediction on the node n, new metrics are configured, updating some shortest paths $sp(f)$ also noted $sp_{RAR}(f)$ with $f \in F$ such that $n \notin sp_{RAR}(f)$. This study does not consider the impact of the failure of ingress or egress nodes, as there is no way to protect or restore the traffic flow in such a case. In real networks, such a case is handled by multi-homing. Each node $n \in N$ is characterized by its Mean-Time-Between-Failure $MTBF(n)$ and the Mean-Time-To-Repair $MTTR(n) << MTBF(n)$. For risk awareness modelling, like in Sec. III, $Recall(n)$ and $Precision(n)$ values for each node n are defined. Changes in time of these parameters are not considered. For a stationary ergodic process, the probability that a node is in a failure state is:

$$P_{node}(n) = \frac{MTTR(n)}{MTBF(n)+MTTR(n)} << 1. \quad (2)$$

When the case of identical routers is considered with regards to failure probability, the dependency in n in the notations for all these parameters can be omitted. For each flow $f \in F$, when a transit node $n \in sp(f)$ is failing, the flow is only restored after the convergence process which duration t_C, as mentioned in Sec. II, is define by the following formula:

$$t_C = t_D + t_F + t_{SP} + t_U. \quad (3)$$

t_F and t_U are constant and are less than a hundred milliseconds, with 0.03 seconds for t_F [13] and 0.2 seconds for t_U [14]. The computation time of the shortest paths is also a very short delay but depends on the number of nodes of the network which, according to [14], is equal to $t_{SP}(N) = 2.47.10^{-6} * |N|^2 + 9.78.10^{-3}$ where N is the set of routers of the network (G). For clarity reason, the dependence on N by using t_{SP} in the following formula has been omitted. The real impacting step is the failure detection that takes several seconds by using the *Hello* protocol. Indeed even if in some cases, the lower level protocol allows detecting failure more quickly, this mechanism is not always available and does not detect some faults in the router such as failure of the controller or software failure. The *Hello* protocol periodically send keep-alive messages at the *Hello Interval* period (noted t_{HI}) and wait the *Router Dead Interval* (t_{RDI}) timers expiry to declare a failure. When considering that failures occur uniformly between two *Hello* message, and that t_{RDI} is set to $4*t_{HI}$, the average detection time is $t_D = (3 * t_{HI}) + (t_{HI}/2)$. In consequence, with the use of a *Hello Interval* of one second in the evaluations (which is the smallest value accepted by the current OSPF protocol), the average failure detection time is about 3.5 seconds.

B. Unavailability computing

Considering the functioning of OSPF convergence process is conform with the Eq. (3), the average unavailability of the flow $f \in F$ due to the failure of the router $n \in N$ is expressed by the following formula by using Eq. (2):

$$\begin{aligned} U_{OSPF}(f,n) &= t_C/(MTBF(n) + MTTR(n)) \\ &= P_{node}(n).t_C/MTTR(n). \end{aligned} \quad (4)$$

For the RAR case, the conditional probability part can be split in the sum of the two disjoint probabilities of successful risk detection (*Recall*) or unsuccessful risk detection ($1 - Recall$):

- if the failure is not anticipated ($1 - Recall$), traffic flows rely on IP restoration with unavailability defined by the Eq. (4);
- if the failure is predicted (*Recall*), the RAR mechanism update the shortest path $sp_{RAR}(f)$ to the flow f, such as $n \notin sp_{RAR}(f)$, removing the unavailability due to the convergence delay.

Following these two points, unavailability of a flow f in case of failure in a router $n \in sp(f)$ is:

$$U_{RAR}(f,n) = (1 - Recall(n)).U_{OSPF}(f,n). \quad (5)$$

The Eq. (5) highlights the importance of *Recall* on the availability provided by the RAR mechanism, and where the worst case ($Recall(n) = 0$) leads to a behaviour similar to standard OSPF protocol defined by Eq. (4) and the best case ($Recall(n) = 1$) provides a total availability.

By considering that router failures are independent events, with $X = RAR$ or $OSPF$, the network unavailability for flow $f \in F$ is:

$$U_X(f) = 1 - \left(\prod_{n \in sp(f)} (1 - U_X(f,n)) \right) \approx \sum_{n \in sp(f)} U_X(f,n). \quad (6)$$

The approximation is valid because of Eq. (2), which means that the case of simultaneous failures in the network has been neglected. Then one can give a weight for each flow f, e.g., their throughput $\mu(f)$, in order to define the average network unavailability:

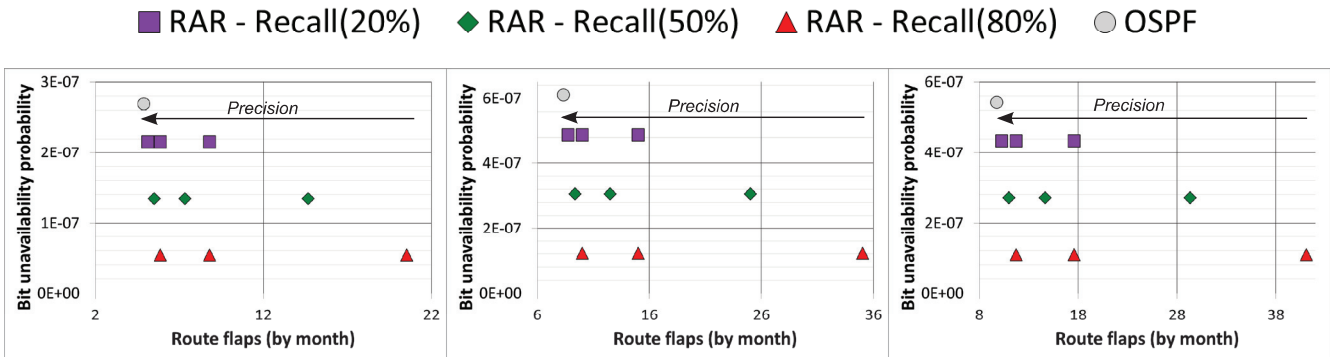
$$U_X(G,F) = \frac{\sum_{f \in F} \mu(f).U_X(G,f)}{\sum_{f \in F} \mu(f)}. \quad (7)$$

1) Routing instability quantification: The advantage of the RAR mechanism is a greater availability, but in return, the false predictions lead to useless changes in routing. These modifications result in a temporarily suboptimal routing, but most of all, if they are too frequent, they create an instability that is detrimental for network performance. Operators are sensitive to this stability, it is then important to measure the number of routing flaps of the RAR mechanism and to compare it with the standard OSPF behaviour. With a standard IGP protocol, routing changes only append around the failures, one time when the failure is detected, and another time when the failure is repaired. In consequence the number of routing oscillation is:

$$RF_{OSPF}(G,F) = \sum_{n \in N} \frac{2}{MTTR(n) + MTBF(n)}. \quad (8)$$

But with the RAR mechanism, it is necessary to consider true positive (TP) and false positive (FP) failure prediction:

- with a false prediction, a routing metric change triggers a first routing oscillation and at the end of Δt_p the prediction expire, resetting the metrics to their initial values, creating a second modification of routing. For this case, we have a formula similar to Eq. (8), but it happens with probability FP instead of $TP + FN$. We need to add a correction factor $FP/(TP + FN)$ which, with Eq. (1), is equal to $Recall * (1/Precision - 1)$.
- when a failure is predicted, the metrics are increased to push the traffic away from the risky nodes, which creates a first modification of routing. At the effective failure, the removal of failed node creates a negligible change in routing. The convergence process only



(a) (b) (c)

Figure 2. Routing flaps and unavailability estimation ($MTTR = 5000h$, $MTTR = 5h$, $\Delta t_p = 1h$) for the 3 networks.

removes the paths having the failed node at a end, not creating an extra routing oscillation. The second oscillation is only triggered after the repair period like unpredicted failures.

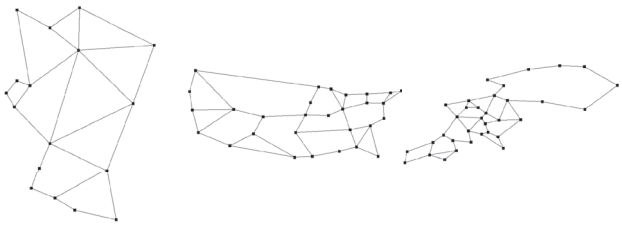
Because the failure probability of each node is low (see Eq (2)), we neglect the cases of multi-failures as well as the cases of multi false failure predictions. From this observation, the definition of *Recall* and *Precision* at Eq. (1), the routing flap rate of the RAR mechanism is:

$$RF_{RAR}(G,F) = RF_{OSPF}(G,F) + \sum_{n \in N} \left(\frac{1}{Precision(n)} - 1 \right) \cdot \left(\frac{2}{MTTR(n) + MTBF(n)} \right). \quad (9)$$

The next section is dedicated to the RAR mechanism evaluation on concrete examples.

VI. Case study analysis

A. Network topologies and traffic matrices



(a) (b) (c)

Figure 3. Network topologies.

G	$\|N\|$	$\|E\|$	$\bar{d}$	Density	$d(G)$	$\|F\|$	Traffic (Gbit/s)	#IF
a	17	26	3.06	0.19	8	242	1363	626
b	29	44	3.03	0.11	9	812	485	440
c	34	49	2.88	0.09	14	1122	1554	1244

Table I
Topology Characteristics.

To evaluate the aforementioned models, three realistic core network topologies, as illustrated in Fig. 3, have been chosen: a) a national network in Europe, b) a US network and c) a Pan-European network . The characteristics of these networks are detailed in Tab. I with $|N|$ representing the number of routers, $|E|$ the number of links, $\bar{d}$ the average degree, and $d(G)$ the diameter of the network. Moreover, the IGP metrics used for shortest path calculation are based on distance between each node (i.e., city). Concerning the traffic matrices, the flows distribution is proportionally balanced by the city populations and the aggregated information such as the number of flows ($|F|$) or the sum of these flows (Traffic) can be found in Tab. I.

B. Theoretical comparison between RAR and OSPF

The purpose of this section is to compare the performance of Risk-Aware augmented OSPF versus classical OSPF. In the analysis, the convergence time uses the aforementioned values given in Sec. II and V with Mean Time Between Failure (MTBF), Mean Time To Repair (MTTR) and Δt_p respectively assigned to 5000 hours, 5 hours and 1 hour as reference values. Nevertheless, for exhaustivity reason, we mind to vary the MTBF from 1000 to 10000 hours, the MTTR from 1 to 10 hours and Δt_p from 5 minutes to 10 hours on the three topologies but for space reason only the most representative results are shown. Since the main benefit of the RAR solution is a higher availability probability than OSPF while containing the routing flap rate, we plot for each network x (x = a, b and c) on Fig. 2 what gives OSPF and RAR cases for the routing flap rate (X-axis) and the bit unavailability probability (Y-axis). Moreover, for RAR, three different levels of performance are considered for both *Recall* and *Precision* of the online failure predictor: low level at 20%, medium level at 50% and a high level at 80%, giving in total nine points for the different RAR cases.

Thanks to the RAR mechanism one can observe, independently from the topology (see Fig. 2), a gain in availability of more than 5 for the high *Recall* value, almost 2 for the medium *Recall* value and 1.2 for the low *Recall* value. When analyzing the routing flaps, Fig. 2 shows that

the RAR mechanism generates a reasonable number of oscillations that is not impacting the network, even with poor *Precision* performance. Except for the b topology where two cases with poor *Precision* performance reach about one oscillation by days, all the others cases keep under 20 oscillations by months that is very acceptable for any network operator. While *Recall* values have an important impact on the availability performance of the RAR scheme, *Precision* value impact is almost negligible since routing flap keeps relatively contained. Beyond these

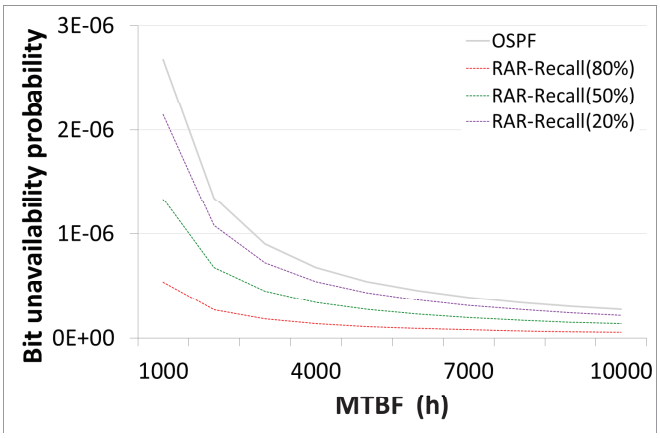


Figure 4. MTBF impact on network c availability.

reference conditions results, it is interesting to look into the RAR behaviour on one topology (c) when the failure rate evolves and MTBF is the most impacting parameter. The Fig. 4 shows how MTBF influences the unavailability of each strategy, but although the differences of performance increase with failure rate, the ratio between each mechanism remains constant. Only *Recall* has incidence on availability, and it is why the RAR mechanism is represented by only three curves. The Fig. 5 is rather

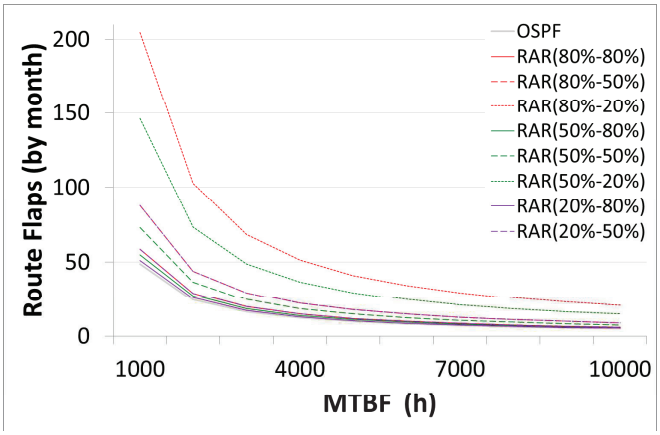


Figure 5. MTBF impact on network c routing flaps.

focused on the routing flap rate evolution resulting from the variation of the failure rate. In all the configurations, the routing flap increases a lot when the MTBF decreases below 3000 hours. It is also useful to notice that the failure prediction performance plays an important role on the routing flap quantity but it is mostly the quantity of prediction (TP and FP) which is responsible of routing flap variations. This figure allows to start identifying the restricted mode. Indeed the usage of poor *precision* with high failure rate implies a routing flap rate that operators are not ready to accept. Nevertheless, even if networks are dimensioned to handle failures, a high number of failure predictions can generate congestion that are not quantified by our model and will also restrict the utilization mode of the RAR mechanism. As a consequence, only simulation-based assessment can figure out this issue.

C. Simulation experiments

In order to verify our analytical hypothesis, the RAR mechanism has been implemented in the NS3 discrete-event simulator [15]. The general reliability theory [16] has been applied to generate failure events using of exponential distribution ($mean = MTBF$) for time between failures and lognormal distribution $\ln N(\mu, \sigma^2)$ with $\mu = \log(MTTR) - ((0.5) * \log(1 + ((0.6 * MTTR)^2/MTTR^2)))$ and $\sigma = \sqrt{\log(1 + ((0.6 * MTTR)^2/MTTR^2))}$ for time to repair. Concerning failure prediction, anticipated failures have been uniformly chosen following *Recall* ratio and the false predictions have been uniformly generated during the simulated time to reach targeted *Precision*. Using the same parameters as the analytical evaluation for network topology and traffic matrix, the influence of MTBF, MTTR, *Recall*, *Precision* and Δt_p has been analyzed using 7 trials with a simulated time equals to $5 * (MTBF + MTTR)$. Using the same configurations than the analytical experiments of Fig. 2, the simulation results with a confidence interval of 99% are presented in Fig. 6. These simulation results are in adequacy with the analytical results, which supports the validity of the analytical model we proposed.

But the real purpose of the simulation experiment is to identify when congestion appears and in which conditions. What is observed, for the three topologies, is that the MTBF decreases towards 1000 hours always imply congestion. As the three topologies have similar behaviours, only topology c is shown on Fig. 7 where one can see the congestion issue for a MTBF of 1000 hours with the three configurations having the lowest *Precision* performance (20%).

Congestion appears when multiple rerouting overlap and exceed the network capability, which happens when the failure rate becomes high or when the prediction period becomes too long. The Fig. 8 illustrates the problem with the network b that is the most impacted topology. Congestion is firstly present for poor *Precision* performance but also for the other cases, yet the most important information of this figure is that usage of Δt_p should be restricted to one hour at maximum in order to preserve a

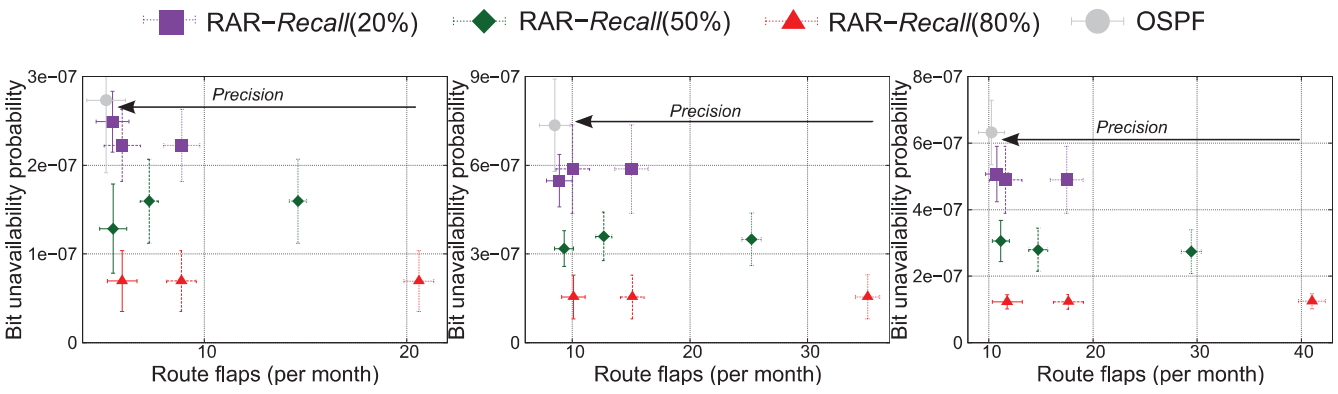


Figure 6. Routing flap and unavailability simulation ($MTTR = 5000h$, $MTTR = 5h$, $\Delta t_p = 1h$) for the 3 networks.

congestion-free network.

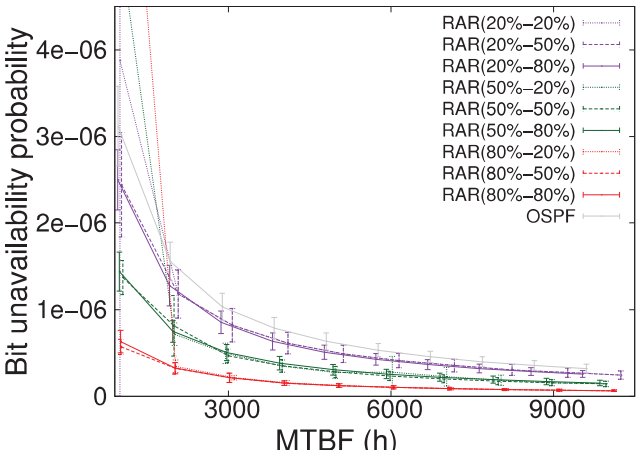


Figure 7. MTBF impact on network c availability.

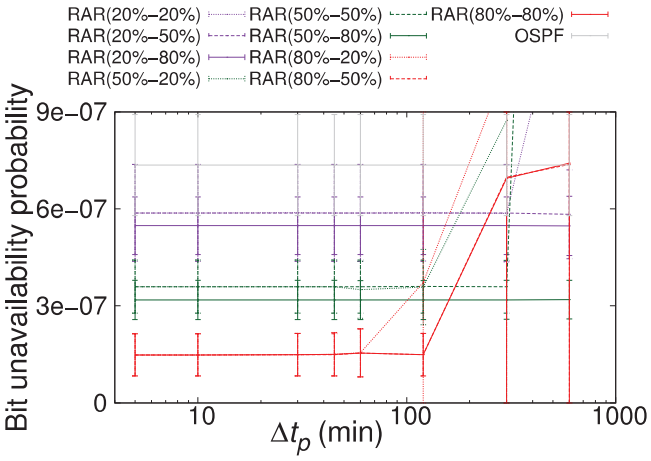


Figure 8. Δt_p impact on network b availability.

VII. Experimental prototype

We developed a proof of concept prototype in which our mechanism has been implemented, in order to demonstrate its feasibility with the real world assumptions. The main purpose of this prototype is to show that, within a non-simulated environment, dynamic assignment of link cost triggered by failure risk detection actually preserves traffic flows from service interruption, transparently from the users' point of view, maintaining the required QoS.

To validate the assumptions of the previous section with a real network carrying real traffic, node failures were emulated in our prototype. Our prototype implementation is made of two main functions, one for the risk level assessment and another for the link weight assignment. First, the node risk level assessment exploits local information retrieved by a local standalone process accessing to sensors information such as the Linux sensors information recorded in the */proc/acpi/* directory. We use the knowledge plane implementation of Ginkgo Networks [17] to implement the communication of this risk-awareness data between the risk assessment module and the link weight assignment engine. The link weight assignment engine requires a communication with the OSPF routing engine to modify the link cost values. For that purpose, the Quagga [18] open source implementation of OSPF was used and the *Command Line Interface* (CLI) facilities was exploited which are fast enough to trigger the OSPF Link State Advertisements in a timely manner.

The global experimental platform is made of three computers (bi-processor Intel Xeon 2 GHz, 4 GBytes memory, with four 100 Mbps Ethernet interface), on which a topology of eight routers is implemented: Each OSPF router is configured on top of a VMware virtual machine which includes the Linux operating system, the Quagga software and our risk-aware agent.

With an OSPF configuration similar to what is described in the previous section (*Hello Interval* of 1 second and *Router Dead Interval* equals to $4 * HelloInterval$), the impact of failure on a constant traffic of 1 Mbit/s

generated by Iperf was studied. The Fig. 9 shows the different results, with and without the RAR mechanism. While OSPF generates a service interruption of about the t_D delay, the RAR mechanisms allow to completely remove the effect of failure from the user point of view.

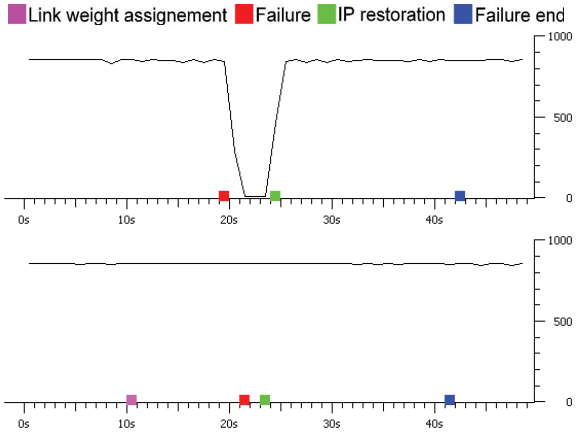


Figure 9. Failure consequence on constant traffic without and with RAR.

We also run experiments on HD video traffic. Without the risk-aware routing mechanism, one can observe an interruption in the video stream, while the activation of the mechanism allows watching the complete video sequence without interruption. The Fig. 10 is the result of one these experiments. Like previously highlighted, the failure is clearly visible with only OSPF while RAR activation allows to be undisturbed by the failure.

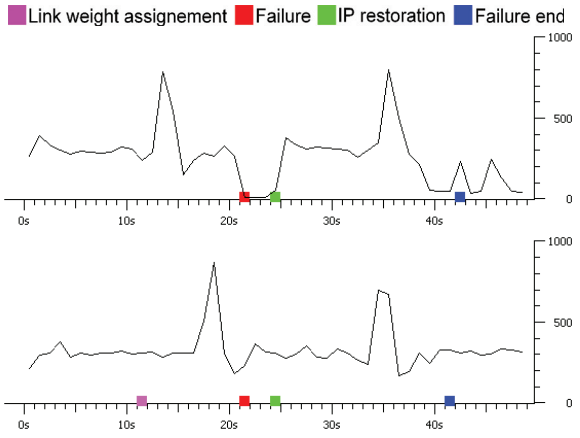


Figure 10. Failure consequence on HD video traffic without and with RAR.

VIII. Conclusion

This paper differs from common proactive self-healing contribution by focusing on the resilience action and its evaluation instead of the failure prediction aspect. The Risk-Aware Routing allows creating a small window of time before the failure that has been anticipated to prompt traffic flow to avoid the failure. Routing is preventively reconfigured to achieve a situation close to what will become effective after the failure occurrence by assigning a risk metric to future down links which is high enough to prevent this link to be used by traffic. This will completely remove the service interruption caused by the breakdown when the predictive failure is confirmed.

The principle of this mechanism relies on the fact that networks are already dimensioned to handle failures. Such a rerouting is then easily supported by the network when the quantity of failure predictions is reasonable. Unfortunately, routing changes resulting from failures prediction can introduce instability when they are too frequent. It was therefore necessary to take this into account in the evaluation of our mechanism. For this purpose, an analytical model is proposed to quantify the theoretical gains of the RAR mechanism in terms of both unavailability and routing oscillations. An implementation in a discrete event simulator was used to check the validity our modelling on the one hand, and observe congestion incidence of our mechanism on the other. Finally, an experimental prototype was implemented to test the feasibility of the functionality. The findings of this study shows a real gain, proportional to the number of failures predicted, while containing the routing number of oscillations in proportions ensuring stability for the network.

The number of oscillations could be problematic for networks requiring very high stability, if the failure detection *Precision* is low (*i.e.* 20%) and the failure rate is high (*i.e.* MTBF <3000 hours). In such a bad case, the main drawback mostly comes from congestion issue that outweighs all the benefits of the mechanism by generating more losses than the standard IP restoration. This phenomenon is present when the failure rate is very high and the number of false predictions is very important and/or with a prediction period Δt_p of several hours. For this reason, our mechanism should be implemented when the failure prediction does not generate too many false predictions and with a prediction period less than two hours. When this is the case, our risk-aware routing mechanism is really beneficial for the network availability.

References

[1] S. Poretsky, B. Imhoff, and K. Michielsen, "Terminology for benchmarking Link-State IGP data plane route convergence."

[2] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks," *SIGCOMMComput. Commun. Rev.*, vol. 35, no. 3, pp. 35–44, 2005.

[3] A. Medem, R. Teixeira, N. Feamster, and M. Meulle, "Joint analysis of network incidents and intradomain routing changes," in *Proceedings of CNSM*, 2010, pp. 198–205.

[4] M. Shand and S. Bryant, "IP Fast Reroute Framework," RFC 5714, Internet Engineering Task Force, Jan. 2010.

[5] M. Xia, M. Tornatore, C. U. Martel, and B. Mukherjee, "Risk-Aware provisioning for optical WDM mesh networks," *IEEE/ACM Transactions on Networking*, 2010.

[6] B. Vidalenc, L. Noirie, and L. Ciavaglia, "GMPLS adaptive level of recovery," in *Proceedings of the IEEE International Conference on Communications, ICC*, Ottawa, Canada, Jun. 2012, pp. 2735–2740.

[7] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Computer Survey*, vol. 42, no. 3, pp. 1–42, 2010.
[8] N. Tcholtchev, M. Grajzer, and B. Vidalenc, "Towards a unified architecture for resilience, survivability and autonomic fault-management for self-managing networks," in *Proceedings of the 2009 international conference on Service-oriented computing*, Stockholm, Sweden, 2009, p. 335–344.
[9] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 756–767, 2002.
[10] F. Salfner, M. Schieschke, and M. Malek, "Predicting failures of computer systems: a case study for a telecommunication system," in *International Parallel and Distributed Processing Symposium*, 2006.
[11] B. Vidalenc and L. Ciavaglia, "Proactive fault management based on risk-augmented routing," in *IEEE GLOBECOM Workshops*, Miami, USA, Dec. 2010, pp. 481–485.
[12] P. Francois and O. Bonaventure, "Avoiding transient loops during the convergence of link-state routing protocols," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1280–1292, 2007.
[13] A. Shaikh and A. Greenberg, "Experience in black-box OSPF measurement," in *ACM SIGCOMM Workshop on Internet Measurement*, San Francisco, California, USA, 2001, pp. 113–125.
[14] M. Goyal, K. Ramakrishnan, and W. chi Feng, "Achieving faster failure detection in OSPF networks," in *IEEE International Conference on Communications*, vol. 1, 2003, pp. 296–300.
[15] www.nsnam.org.
[16] M. Ohring and J. R. Lloyd, *Reliability and Failure of Electronic Materials and Devices*. Academic Press, 2009.
[17] T. Bullot, H. Zimmermann, D. Gaiti, L. Merghem-Boulahia, and G. Pujolle, "Autonomous agents for autonomic networks," *Annales des télécommunications*, pp. 1017–1045, 2006.
[18] "Quagga software routing suite," www.quagga.net.