# Avoiding transient loops during the convergence of link-state routing protocols

Pierre Francois and Olivier Bonaventure
Université catholique de Louvain

***Abstract*— When using link-state protocols such as OSPF or IS-IS, forwarding loops can occur transiently when the routers adapt their forwarding tables as a response to a topological change. In this paper[1], we present a mechanism that lets the network converge to its optimal forwarding state without risking any transient loops and the related packet loss. The mechanism is based on an ordering of the updates of the forwarding tables of the routers. Our solution can be used in the case of a planned change in the state of a set of links and in the case of unpredictable changes when combined with a local protection scheme. The supported topology changes are link transitions from up to down, down to up, and updates of link metrics. Finally, we show by simulations that sub-second loop free convergence is possible on a large Tier-1 ISP network.**

## I. Introduction

The link-state intradomain routing protocols that are used in IP networks [2], [3] were designed when IP networks were research networks carrying best-effort packets. The same protocols are now used in large commercial ISPs with stringent Service Level Agreements (SLA). Furthermore, for most Internet Service Providers, fast convergence in case of failures is a key problem that must be solved [4], [5]. Today, customers are requiring 99.99% reliability or better and providers try to avoid all packet losses.

Vendors are actively working on improving their implementations to achieve faster convergence [6], [5]. Solving the fast convergence problem is complex as it involves detecting the failure on the attached router, producing a new Link State Packet (LSP) describing the failure, flooding this new LSP and finally updating the Forwarding Information Base (FIB) in all the routers using the failed resources in the network. Sub-second convergence has been made possible, but the sub-50 msec target can only be achieved by the means of a local restoration scheme. Achieving very fast convergence in an IP network will thus require temporary tunnels to quickly reroute traffic around failures, as in MPLS networks [7]. Several solutions to establish such local protections have been proposed in the literature [8], [9], [10], [11], [12]. Unfortunately, in an IP network, using a protection tunnel to locally reroute the traffic around the failed link is not sufficient as transient loops may occur during the update of the FIBs of the other routers in the network.

To understand this problem, let us consider the Internet2/Abilene backbone[2]. Figure 1 shows the IGP topology of this network. Assume that the link between IP and KC fails but was protected by an MPLS tunnel between IP and KC via AT and HS. When AT receives a packet with destination DN, it forwards it to IP, which forwards it back to AT, but inside the protection tunnel, so that KC will decapsulate the packet, and forward it to its destination, DN.

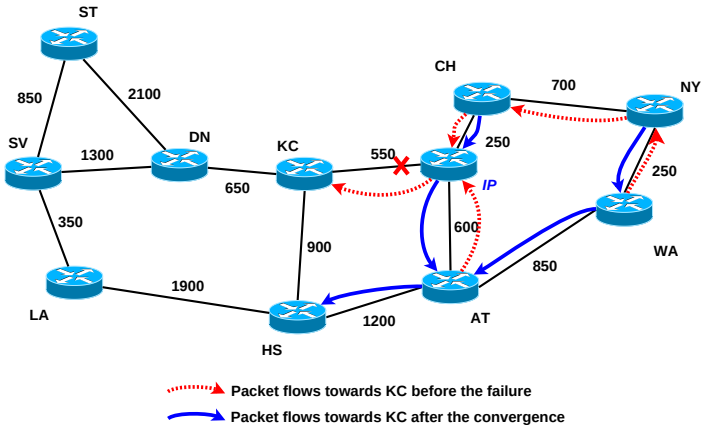


Fig. 1: Internet2 topology with IGP costs

This suboptimal routing should not last long, and thus after a while the routers must converge, i.e., adapt to the new shortest paths inside the network, and remove the tunnel. As the link is protected, the reachability of the destinations is still ensured and thus the adaptation to the topological change should be done by avoiding transient loops rather than by urging the updates on each router. The new LSP generated by IP indicates that IP is now only connected to CH and AT. Before the failure, the shortest path from WA to KC, DN, ST and SV was via NY, CH and IP. After the failure, NY will send its packets to KC, DN, ST and SV via WA, AT and HS. During the IGP convergence following the failure of link KC-IP, transient loops may occur between NY and WA depending on the order of the forwarding table updates performed by the routers. If NY updates its FIB before WA, the packets sent by NY to KC via WA will loop on the WA-NY link. To avoid causing a transient loop between WA and NY, WA should update its

[1]A preliminary version of this paper was presented at INFOCOM 2005 [1].

[2]This network is much smaller than large ISP backbones, but it is one of the few networks whose detailed topology is publicly available. We verified that similar transient loops could occur in larger ISP backbones, but the size of those backbones prevented us from using them as an example in this paper. Note that the IGP metrics have been rounded off to facilitate the understanding of the topology. The round off does not influence the routing tables of the network.

FIB before NY for this particular failure. A detailed analysis of the Internet2 topology shows that transient routing loops may occur during the failure of most links, except ST-DN and ST-SV. The duration of each loop will depend on how and when the FIB of each router is updated. Measurements on commercial routers have shown that updating the FIB may require several hundred of milliseconds [5]. Transient routing loops of hundred milliseconds or more are thus possible and have been measured in real networks [13].

As shown with the simple example above, the transient routing loops depend on the ordering of the updates of the FIBs. In the remainder of this paper, we first discuss in section II other types of changes to the topology of an IP network that must be handled without causing transient routing loops. In section III, we prove that the updates of the FIB can be ordered to avoid transient loops after a topology change affecting a set of links. This proof is constructive as we give an algorithm that routers can apply to compute the ranks that let them respect the proposed ordering. Next, in section V, we propose to use ”completion messages” to bypass the ranks computed by the routers, so that the loopfree convergence process can complete faster. In section VI, we evaluate by simulations the time required by our modified link-state protocol to converge. In Section VII, we present an optimization that lets routers find out when they can reroute without respecting their rank while ensuring that no loop will occur. Finally, in section VIII, we review the other mechanisms that have been proposed to enhance the convergence of the IGP.

## II. Topology changes in IP Networks

Several types of changes can occur inside the topology of an IP network. The most common type of change is the failure of a link [14]. A network typically contains point-to-point links and LANs. Point-to-point links are typically used between Points of Presence (POPs) while LANs are mainly used inside POPs. We focus on point-to-point links in this paper as there are special techniques to protect LANs [15] used in ISP networks.

When a point-to-point link fails, two cases are possible. If the link is not locally protected, the IGP should converge as quickly as possible. If the link is protected with a special tunnel or another technique [16], [9], the IGP should converge without causing transient loops as the traffic passes through the tunnel during the IGP convergence. We will call such events *link down* events in this paper.

It should be noted that *link down* events are often caused by manual operations and thus can be considered as planned events. Surveys conducted by a large ISP [4] revealed that, over a five month period, 45 % of the failure events occurred during maintenance hours. Another ISP [17] indicates that over one month, 75 % of the IS-IS events were caused by maintenance operations. Another study [14] mentions that 20 % of *all link down events* were planned. Those planned events should not cause transient forwarding loops [17]. In the case of a maintenance of a link, some operators set the metric of the link to MAX_METRIC in order to let packets be forwarded on the link during the convergence [18]. However, doing this is not sufficient as transient loops can still occur .

It is also important to consider the increasing integration between the IP network and the underlying optical network [19]. As the integration with the optical layer increases, the topology of IP networks will change more frequently than today. For example, [20] proposed to allow routers to dynamically establish optical links to handle traffic spikes. Similar approaches have been proposed with MPLS tunnels. Once a new optical link or MPLS tunnels becomes active, an IGP adjacency will be established between the attached router and the link will be advertised in the IGP [21]. Unfortunately, the addition and removal of each of those tunnels can cause transient loops in the network.

Another source of changes in IP networks are the IGP metrics. Today, network operators often change IGP metrics manually to reroute some traffic in case of sudden traffic increase [18]. Furthermore, several algorithms have also been proposed to automate this tuning of the IGP metrics for traffic engineering purposes [22]. Today, those algorithms are mainly implemented in network planning and management tools [23], [24]. However, ISPs are still reluctant to use such tools to frequently change their IGP metrics as each change may create transient forwarding loops in their network.

A second type of important events are those that affect routers. Routers can fail abruptly, but often routers need to be rebooted for software upgrades. For example, figure 6 of [14] shows that during September and October 2002, many links of the Sprint network “failed” once per week during maintenance hours. Those failures are probably due to planned software upgrades of all routers in the network.

When an IS-IS[3] router needs to stop forwarding IP packets, IS-IS can flood a new LSP indicating the router as overloaded [3]. Some ISPs have even defined operational procedures [17] to bring routers down by changing link metrics and setting the `overload bit`, but those procedures are not sufficient to ensure that transient loops will not occur during the IGP convergence. The graceful restart extensions [25], [26], [27] could be used when a router is rebooting. However, those extensions cannot be used for the maintenance operations affecting the forwarding plane of the router. As shown by the above discussion, there are many different types of changes in IP networks that should be handled without risking to create transient routing loops in the network.

## III. An ordering for the FIB updates

To avoid transient loops during the convergence of link-state protocols, we propose to force the routers to update their FIB by respecting an ordering that will ensure the consistency of the FIB of the routers during the whole convergence phase of the network.

In the context of a predictable maintenance operation, the ressources undergoing the maintenance will be kept up until the routers have updated their FIB and no longer use the links to forward packets. In the case of a sudden failure of a link that is protected with a Fast Reroute technique, the proposed ordering ensures that a packet entering the network will either

[3]We limit our discussions to IS-IS in this paper, but a similar reasoning is valid for OSPF as well

follow a consistent path to its destination by avoiding the failed component or reach the router adjacent to the failure and will be deviated by the Fast Reroute technique to a node that is not affected by the failure, so that it will finally reach its destination.

In this section, we briefly review the orderings in the case of single link events (link down or metric increase, link up or metric decrease), that we proposed in [1]. Then, we extend the solution to events affecting Shared Risk Link Groups. Finally, we discuss router and line card events, which are particular SRLG cases.

As those orderings are applied in the case of **predictable changes** and in the case of sudden changes where a local protection is provided, avoiding transient loops will permit to avoid all the packet losses during the IGP convergence inside the network.

Note that the proposed orderings are valid when asymmetrical link metrics are used in the topology, i.e., when there exists links $X \leftrightarrow Y$ such that the metric of $X \rightarrow Y$ is not equal to the metric of $Y \rightarrow X$.

Also, the solution takes into account the case where multiple equal cost paths from one router to another are used before and/or after the event. In the following sections, we use the terms of Shortest Path Trees, and reverse Shortest Path Trees to respectivley denote the set of shortest paths from a router to the other routers of the network and the set of shortest paths from all the routers to a given router. When Equal Cost MultiPath (ECMP) is used, the union of these paths form an acyclic graph, not a tree. We will explain how routers deal with this when it could lead to ambiguous results in the provided proofs and algorithms.

### A. Single Link Events

*1) Link down or metric increase:* In the case of a link down or metric increase event for a link $X \rightarrow Y$, a router $R$ must update its FIB **after all the routers that used** $R$ to reach $Y$ **before** the event.

To respect this ordering, $R$ computes $rSPT_{old}(X \rightarrow Y)$, the part of the reverse Shortest Path Tree (rSPT) of $Y$ in the old topology that is affected by the change. The rSPT of a node is the set of shortest paths to this node. The part of interest in this rSPT is the set of shortest paths *to* $Y$ that are affected by the failure of $X \rightarrow Y$. Within this part, the branch of the tree that is under $R$ in $rSPT_{old}(X \rightarrow Y)$ contains all the paths to $R$ that were used to reach at least one destination via $R$ and link $X \rightarrow Y$ before the event.

The rank of $R$ is equal to the depth of this part of the tree, $depth(R, rSPT_{old}(X \rightarrow Y))$. In the case of ECMP, the rank of $R$ is the maximum number of hops among the equal cost shortest paths to $R$ inside the graph. This value can be easily obtained by computing $rSPT_{old}(Y)$, the set of shortest paths to $Y$.

The time at which $R$ will be allowed to update its FIB is equal to the obtained rank multiplied by a configurable worst-case FIB update time ($MAX\_FIB\_TIME$), that depends on the number of prefixes that are advertised in the network. Theorem 1 states that the proposed rank will let a router $R$ update its FIB before the routers that $R$ used to reach the failing link.

**Theorem 1** *By using the proposed rank computation for a single link down or metric increase event affecting link $X \rightarrow Y$, a router $R$ that has not yet updated its FIB for a destination $d$, that it reached via $X \rightarrow Y$, will forward packets to $d$ along routers that have not updated their FIB yet.*

*Proof:*

1) Let us assume that a router $R$ was using a neighbor $N$ to reach $Y$ via $X$
2) $R$ is below $N$ in $rSPT_{old}(X \rightarrow Y)$
3) From 2, we have

$$Rank(N) = depth(N, rSPT_{old}(X \rightarrow Y)) \geq depth(R, rSPT_{old}(X \rightarrow Y)) + 1$$

4) The same property can be verified hop by hop along the paths from $R$ to $X$

■

Theorem 1 implies that the routers along those paths will not have updated their FIB when $R$ has not updated its FIB yet. The packets forwarded by $R$ will thus arrive in $Y$ and be forwarded on non affected paths from $Y$ to $d$. It is sure that the paths from $Y$ to $d$ are not affected by the event. Indeed, if one router was using $X \rightarrow Y$ to reach $d$, then $Y$ could not use $X \leftrightarrow Y$ to reach $d$. The contrary would imply an intra domain forwarding loop while the network was stable.

As an example, let us consider the shutdown of link $IP \leftrightarrow KC$ in figure 1. According to the ordering, the rank of $IP$ is 3, as longest branch under $IP$ in $rSPT_{old}(IP \rightarrow KC)$ is $IP - CH - NY - WA$. $AT$ has a rank of 0, because it is a leaf in $rSPT_{old}(IP \rightarrow KC)$. So, $IP$ will reroute after $AT$ and no loop will occur along $IP \leftrightarrow AT$. Similarly, the rank of $NY$ is one because the deepest branch under $NY$ in $rSPT_{old}(IP \rightarrow KC)$ is $NY - WA$. $WA$ has a rank of 0, as it is a leaf in $rSPT_{old}(IP \rightarrow KC)$. So, $WA$ will update its FIB before $NY$ and no loop will occur along $WA \leftrightarrow NY$.

*2) Link up or metric decrease:* When a link $X \rightarrow Y$ is brought up in the network, or its metric is decreased, the required ordering is such that a router $R$ updates its FIB **before the routers that will use** $R$ to reach $Y$ via $X$. To apply this ordering, $R$ computes $PathLength(R, X)$, the number of hops of its path from $R$ to $X$. Note that in the case of ECMP, the considered number of hops is the largest one among the multiple equal cost paths. This value, that we call the rank of $R$, is easily obtained by $R$ when it computes its new SPT to update its FIB. The time at which $R$ will be allowed to update its FIB is equal to its rank multiplied by the worst-case FIB update time.

We state in Theorem 2 that each router $N$ being on the new paths from $R$ to $X$ will update its FIB before $R$.

**Theorem 2** *By using the proposed rank computation for a single link up or metric decrease event affecting link $X \rightarrow Y$,*

*a router $R$, that has already updated its FIB for a destination $d$ that it will reach via $X \rightarrow Y$, will forward packets towards $d$ along routers that have already updated their FIB.*

*Proof:* For each router $N$ on the path from $R$ to $X$ :

1)
$$Rank(R) = PathLength(R, X)$$
$$\geq$$
$$PathLength(R, N) + PathLength(N, X)$$

2)
$$Rank(N) = PathLength(N, X)$$

3) $PathLength(R, N) > 0$
4) From 2 and 3, we have
$$PathLength(R, X) = Rank(R)$$
$$>$$
$$PathLength(N, X) = Rank(N)$$

5) From 4, $N$ updates its FIB for destination $d$ before $R$

■

Theorem 2 implies that packets rerouted by $R$ towards $X \rightarrow Y$ will be forwarded by routers with updated FIBs, so that the packets deviated by $R$ will reach $X \rightarrow Y$ and finally reach their destination.

As an example, let us consider the re-activation of link $KC \leftrightarrow IP$ in the topology depicted in figure 1. There could be a forwarding loop in that case if $WA$ updates its FIB with regard to this event before $NY$, as $WA$ would forward packets destined to $KC$ along $WA \rightarrow NY$, although $NY$ was forwarding such packets along $NY \rightarrow WA$ before the link up event. Also, a forwarding loop could take place along $AT \leftrightarrow IP$ if $AT$ updates its FIB before $IP$. However, this second forwarding loop should not happen in practice because $IP$ will be the first to be aware of the link up event. According to the proposed ranking, $IP$ updates its FIB directly because $PathLength(IP, IP) = 0$. $AT$, will update its FIB after one worst-case FIB update time, as $PathLength(AT, IP) = 1$. Similarly, $WA$ will update its FIB after $NY$ because $PathLength(NY, IP) = 2$ and $PathLength(WA, IP) = 3$, so that the potential loop between $NY$ and $WA$ could not occur if the ranking is applied.

### *B. Shared Risk Link Group events*

In this section, we extend the idea underlying the scheme for single link cases to predictable events affecting a set of links in the network.

One could argue that when an operator wants to shut a set of links down, he could consecutively shut down each link of the set and let IS-IS apply the solution for single link events.

This technique has some disadvantages. Firstly, this technique can produce a large number of end-to-end paths shifts, as routers may, as a response to the shutdown of a link, reroute packets on alternate paths via other links to be shut down. The techniques proposed in this section let routers use their post-convergence outgoing interfaces towards a given destination upon the first and unique update of their FIB for this destination. Secondly, predictable events affecting multiple links can be caused for example by the installation or the shutdown of an optical switch supporting a set of links in the network. As the optical layer and the IP network tends to be more and more integrated, an optical switch undergoing a shutdown could notify the IS-IS routers to which it is connected of its upcoming failure. In this case, the event is not under the control of the operator of the IP network so that it would not be possible for the operator to schedule a sequence of single link shut down operations.

These two issues motivated the generalization of our techniques to the events affecting a set of links.

Currently, IS-IS does not allow to perform a shutdown or installation of a set of links, using a single command issued in one router, or by flooding one single routing message. Indeed, to describe the failure of an SRLG, it is required that at least one router adjacent to each of the links of the SRLG floods a link-state packet describing the failure of this link. The only cases where this is possible is for the particular SRLG cases being the set of links connected to one router. But this does not cover the case of a shutdown or installation of an optical switch connected to a set of routers. We thus need to introduce the possibility to send IS-IS or OSPF messages stating that a given SRLG is going to be shut down or brought up in the network as a result of the event occuring at the optical level. This could be achieved by assigning SRLG IDs to the links of the network and let each router describe the ”shared state” of the SRLG to which its links belong. In order to consider a given SRLG as being up, all the advertised shared states associated with this SRLG must be set to up by the routers that are adjacent to one member of this SRLG. To manually shut down a set of links, an operator could then issue a command in one router adjacent to the members of the SRLG, so that the router will flood its Link-State Packet by setting the state of this SRLG to down.

Note that we do not cover, in this paper, the case where a set of unrelated sudden link failures occur concurrently in the network. When routers face this situation they should, as described in [28], fall back to the regular, fast convergence process.

In the remainder of this section we describe how routers can adapt to the manual shut down of a set of links by avoiding transient loops. Next, we present the solution when a set of links comes back up in the network. Finally, we consider the operational case of an SRLG whose links are connected to one common node. These specific cases cover router shut down and installation, as well as line card shutdown and installation.

*1) SRLG Shutdown or SRLG metric increase:* In this section, we propose an ordering of the FIB updates that preserves the transient forwarding consistency among the routers of the network, in the case of a metric increase (or shutdown) of a set of links. We firstly give a property of the transient forwarding states that allows a loop-free convergence, and then we present an ordering that permits to respect this property. As we present the solution in the context of a predictable topology change, we can assume that the links affected by the shut down operation

remain up until the routers adjacent to these no longer forward packets along those links, i.e., the routers will keep the link up until they have updated their FIB.

The idea underlying the scheme is the same as for the single link case. We want to ensure that, during the whole convergence phase, if a packet with destination $d$ arrives at a rerouting router $R$ that has not yet updated its FIB for $d$, then all the routers along the paths from $R$ to $d$ have not yet updated their FIB for $d$ either. This implies that *once a packet reaches a rerouting router with an outdated FIB for its destination, it will follow an outdated but consistent path towards it*.

If this property is always verified, no transient loop can occur, as each packet entering the network will first follow a path that contains a sequence of routers with an updated FIB. Then, either it reaches its destination or it reaches a router with an outdated FIB. In the later case, we know from the preceding paragraph that the packet will reach its destination. Thus, we know that each packet entering the network follows a loop-free path towards its destination if the proposed ordering is respected.

To ensure the respect of this ordering using a rank, the ranking must be such that if a router $R$ updates its FIB for a destination $d$ with a rank $r$, then all the routers lying on the initial paths from $R$ to $d$ that must update their FIB for destination $d$, must do so with a rank that is strictly greater than $r$.

Considering the shut down of a set of links $\{l_1, l_2, ..., l_j\}$, this property is verified when each router $R$ reroutes for a destination $d$ with a rank equal to $min\{depth(R, rSPT_{old}(l_k) \mid l_k \in Paths(R,d)\}$. $Paths(R,d)$ is the set of paths that are used by $R$ to reach $d$ before the event. In other words, a router computes the rank associated with each individual link being shut down that it is currently using. For each destination for which it has to perform a FIB update, it applies a rank being the minimum among the ranks associated with the links that it uses to reach this particular destination.

$rSPT_{old}(l_k)$ is the acyclic graph containing all the shortest paths towards the tail-end of link $l_k$ on the topology before the event. $depth(R, rSPT_{old}(l_k))$, is the maximum hop distance among the paths to $R$ in this acyclic graph. This depth can be easily computed on the fly of a reverse SPT computation with the tail-end of $l_k$ as a root.

Theorem 3 states that the proposed rank computation lets router apply an ordering such that a router $R$ updates its FIB after the routers that were using it to reach the considered destination.

**Theorem 3** *By using the proposed rank computation for a set of link down event, a router $R$ will update its FIB for a given destination $d$ before the routers that were lying on the initial paths from $R$ to $d$.*

*Proof:* Let us consider that a router $R$ updates its FIB for a destination $d$ with a rank ($Rank(R,d)$).

We have to prove that, for a router $N$ lying on the initial paths from $R$ to $d$ we have $Rank(R,d) < Rank(N,d)$.

Let us denote the affected links on the paths between $N$ and $d$ by $\{l_1, l_2, ..., l_s\}$.

According to the definition of an SPT, we can see that all the affected links on the paths between $N$ and $d$ are also on the paths between $R$ and $d$, as $R$ has $N$ on its shortest paths towards $d$. Note that $R$ can also have other affected links on its paths towards $d$. These are the affected links used by $R$ to reach $N$, and the affected links that are on other equal cost paths to $d$ then the ones via $N$. We denote the links that are used by $R$ and not by $N$ to reach $d$ by $\{l_{s+1}, l_{s+2}, ..., l_{s+t}\}$.

1) From the definition of a rank we have

$$Rank(N,d) = \min_{1 \le i \le s} (depth(N, rSPT(l_i))),$$

and

$$Rank(R,d) = \min_{1 \le i \le s+t} (depth(R, rSPT(l_i))).$$

2) As, before the event, $R$ uses $N$ to reach $d$, and $N$ uses $l_{1...s}$ to reach $d$, we have that $R$ uses $N$ to reach $l_{1...s}$, so that $R$ is below $N$ in $rSPT_{old}(l_i), with\, 1 \le i \le s$, and thus

$$\forall i : 1 \le i \le s : \\ depth(R, rSPT_{old}(l_i)) < depth(N, rSPT_{old}(l_i))$$

So that we have $Rank(R,d) < Rank(N,d)$.

■

Theorem 3 implies that *the rank to reroute for destination $d$ in a router $R$, according to the failure (or the metric increase) of a set of links $l_1, l_2, ..., l_s$ is* $min\{depth(R, rSPT(l_v)) \mid l_v \in Paths(R,d)\}$.

Note that each destination is associated with a rank whose value belongs to the set of ranks computed for each failing link, so that in the worst-case, the FIB updates will be split in as many parts as there are links being shut down.

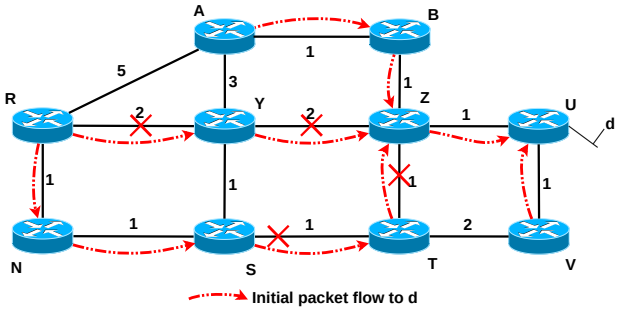


Fig. 2: Illustration of the SRLG down case

Let us illustrate with figure 2 the various properties that lead to a loop free convergence when the proposed ranking is respected. In this figure, the links $R \leftrightarrow Y$, $Y \leftrightarrow Z$, $S \leftrightarrow T$, and $T \leftrightarrow Z$ are being shut down. Initially, $R$ is using $N$ to reach destination $d$, so that to apply the ordering, $R$ should have a rank strictly lower than the rank of $N$ w.r.t. destination $d$.

All the affected links that $N$ uses to reach $d$, i.e., $S \rightarrow T$ and $T \rightarrow Z$, are used by $R$ to reach $d$, because $R$ uses $N$ to

reach $d$. $R$ also has other affected links in its paths towards, $d$; $R \rightarrow Y$ and $Y \rightarrow Z$. $N$ will consider its rank as being the minimum between the depths of the two branches under $N$ in $rSPT(S \rightarrow T)$ and $rSPT(T \rightarrow Z)$. $R$ will consider its rank between the depths of the four branches under $R$ in $rSPT(S \rightarrow T)$, $rSPT(T \rightarrow Z)$, $rSPT(R \rightarrow Y)$ and $rSPT(Y \rightarrow Z)$. $R$ is below $N$ in $rSPT(S \rightarrow T)$, so that the rank associated by $R$ to this link is strictly lower than the one associated by $N$ to the same link. The same reasonning can be applied for link $T \rightarrow Z$. So, $R$ could not have a rank larger or equal to the rank of $N$ w.r.t. destination $d$, as $R$ will uses as its rank the minimum depth among those of the branches under itself in these two rSPTs and also in the branches below $R$ in $rSPT(R \rightarrow Y)$ and $rSPT(Y \rightarrow Z)$.

*2) SRLG up event or metric decrease:* When a set of links is brought up in the network, or when the metrics of a set of links are decreased, routers can also apply a rerouting scheme that ensures the transient forwarding consistency during the whole convergence phase that follows the event.

The proposed scheme allows a rerouting router $R$ to update its FIB for a destination $d$ once all the routers along the paths from $R$ to $d$ have updated their FIB for $d$.

If this property is always verified, no transient loop can occur, as each forwarded packet for a given destination $d$ will first follow a path composed of a set of routers whose FIBs have not been updated yet for $d$. Then, either it reaches $d$, or it reaches a router $R$ that has already updated its FIB for $d$. In the later case, we know that all the routers on the path from $R$ to $d$ have updated their FIB for $d$, so that the packet will be consistently forwarded to $d$.

Now, we show how routers can apply an ordering that respects this property.

In the case of a single link $X \rightarrow Y$ being brought up, a rerouting router $R$ updates its FIB by respecting a rank equal to the length (in hops) of its new shortest path to $X$.

In the multiple link case, a router can have a new SPT such that the shortest paths towards a destination $d$ can contain several of the affected links. However, $R$ will still compute the ranks associated with each link being brought up individually. Then, for each destination $d$, it will apply a rank equal to the maximum of the ranks among those associated with the affected links that it will use to reach $d$.

Theorem 4 states that the proposed ranking computation will let a router $R$ update its FIB for destination $d$ after the routers that are on the new paths from $R$ to $d$.

**Theorem 4** *By using the proposed rank computation for a set of link up event, a router $R$ will update its FIB for a given destination $d$ after the routers that will lie on the paths from $R$ to $d$ after the convergence.*

*Proof:* Let us consider that a router $R$ updates its FIB for a destination $d$. We have to prove that for a neighbor $N$ of $R$ lying on the new paths from $R$ to $d$, we have $Rank(R, d) > Rank(N, d)$.

According to the definition of a SPT, we can see that all the links of the considered SRLG that are on the new paths from $N$ to $d$ are also on the new paths from $R$ to $d$, as $R$ will use $N$ to reach $d$. We will denote those links by $\{l_1, l_2, \ldots, l_s\}$. $R$ can also have other links of this SRLG in its new paths towards $d$. It could be, for example, $R \rightarrow N$, or links on another equal cost path towards $d$. We will denote them by $\{l_{s+1}, l_{s+2}, \ldots, l_{s+t}\}$.

As $R$ will use $N$ to reach $d$, and $N$ will use $l_{1\ldots s}$ to reach $d$, we have that $R$ will use $N$ to reach $l_{1\ldots s}$, so that the rank that $R$ associates with $l_i$ is at least equal to $PathLength(R, N) + PathLength(N, head\_end(l_i))$, i.e., the maximum hop length among the shortest paths from $R$ to $N$ plus the rank that $N$ associates with $l_i$, which is the maximum hop length among the shortest paths from $N$ to the head end of the link $l_i$, i.e, $X$ if $l_i = X \rightarrow Y$. This gives the maximum hop length among the shortest paths (considering the IGP metrics) from $R$ to the head end of $l_i$ via $N$.

From the following properties,

1) $Rank(N, d) = \max_{1 \leq i \leq s} (PathLength(N, head\_end(l_i)))$
2) $Rank(R, d) = \max_{1 \leq i \leq s+t} (PathLength(R, head\_end(l_i)))$,
3) $\forall i : 1 \leq i \leq s$ :

$$PathLength(R, head\_end(l_i)) > PathLength(N, head\_end(l_i))$$

So that we have $Rank(R, d) > Rank(N, d)$

The same property can be recursively discovered between $N$ and its nexthops towards $d$, so that we prove that the rank applied by $R$ for $d$ will be greater then the rank applied by each router on new paths from $R$ to $d$. ■

Theorem 4 implies that loop-free properties described above are respected by the proposed ordering.

As the rank that a router applies for a destination $d$ belongs to the set of ranks that the router computed for each affected link, the number of distinct ranks that can be applied by a router is bounded by the number affected links.

### C. Router and Linecard events

Among the events concerning sets of links, we can find particular predictable events for sets of links connected to a single router. This is the case for router shut down and setup events, and for line card removal or installation. These kind of events are easy to identify as a set in IS-IS if, upon the shutdown of the router, the IS-IS overload bit is set and a link-state packet is flooded by the concerned router. In the case of a router or line card up, the event can be easily identified as a set if the router sends a link-state packet describing all the links being enabled.

In such specific SRLG cases, the first possible behavior of the routers is to consider the event as any other set of link events, and apply the mechanism proposed for the general SRLG cases. However, a simpler behavior is applicable, which will let each router compute one single rank and perform its FIB update in one shot.

When a router or a line card of $X$ is shut down, the behavior is similar to a link down event. The rank computed by a router $R$ is equal to the depth of the tree below $R$ in $rSPT_{old}(X)$.

When a router $X$ or a line card of $X$ is brought up in the network, the behavior is similar to a link up event. The rank computed by a router $R$ is equal to the maximum length (in hops) of the new paths from $R$ to $X$. The proofs are very similar to the ones provided for the single link events. We omit them for the sake of brievety.

## IV. Analysis of the rank based ordering in ISP topologies

If the ordering of the FIB updates is ensured by the means of a timer whose value is set according to a rank and a worst-case FIB update time, the delaying of the FIB updates can be long if the topology is such that large rank values could be computed by the routers for some events.

To analyze this, we computed the ranks that routers would apply in the case of single link failures. For each link shutdown, we looked at the rank applied by the router being the head-end of the link being shutdown. This router is the one with the largest rank for the considered event. The rank that is applied by this router is equal to the worst-case rank that would be applied when the link is brought back up in the topology, so that the figure for the link up cases is the same.

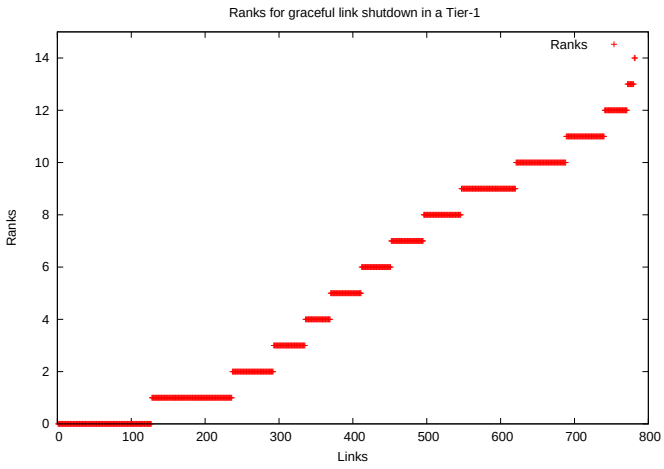


Fig. 3: Ranks for the shutdown of the links in a Tier-1 ISP

In Figure 3, we present the ranks associated with the links of a Tier-1 ISP, containing about 800 (directed) links and about 200 nodes. Note that among those links, the IGP metrics are such that some links are not used and a few others are used only in one direction. The ranks associated with those unused links are equal to 0 in the figure. Note that some links have a rank of 0 even if they are used. This is typically the case of a link from an access router to a core router that is only used by the access router itself. From this figure, we can see that some paths are 14 hops long. Moreover, a large number of prefixes are advertised in this network, so that the worst case FIB update time could be set quite long in order to be conservative. If the worst case FIB update time were set to 1 second, the maintenance of a link in this network could last up to 14 seconds. This could be considered too long by operators, as other events could occur within such a time window.

However, in the case of a maintenance of a link terminating those 14 hops paths, very few routers using the link are rerouting routers. This means that the FIB update time allocated to them is a waste of time, as routers will not perform FIB updates during those periods. The effect is the same in the case of a link up event.

We performed the same analysis on Geant, a network containing 72 (directed) links and 22 nodes [29]. We learned from this analysis that 20 of the 72 directed links were only used by the head-end of the link, so that the obtained rank was 0. No delaying would be applied if those links were shut down, and the link could be effectively shut down just after the FIB update performed by the head-end of the link. The worst-case rank is 4, and was obtained for 7 links. So, even with a very conservative worst-case FIB update time of 1 second and no completion messages, the maintenance of a link in Geant would cause a transiently loop free convergence time of 4 seconds.

This long convergence time motivated the introduction of completion messages to shortcut the delaying allocated to the routers as soon as possible [1].

## V. Completion Messages to speed up the convergence phase

One issue of the rank based ordering scheme is that it assumes a worst-case FIB update time in each router taking part in the process. However, in many cases, routers only have to perform a FIB update for a subset of the reachable destinations, if any. Moreover, the performances of the routers in a network can differ, so that the assumed worst-case FIB update time could be artificially long. In summary, the timer-based ordering works, but it tends to unnecessarily delay the FIB updates in the routers.

To solve that issue, we introduce completion messages [1]. These messages can be placed inside IS-IS Hello PDUs [30]. They are sent by routers to their neighbors to announce that they have performed their FIB update by respecting the ordering. When computing its rank, a router implicitly computes the set of neighbors from which a completion message should be received before it can update its own FIB. Routers will retain this set in a ”Waiting List”.

In this section, we explain how such lists can be built, and when routers are allowed to send completion messages to their neighbors, by still ensuring the proposed loop free ordering of the FIBs.

We firstly present the scheme for single link events, and then we generalize the solution to events affecting sets of links.

### A. Single Link Events

*1) Link down or metric increase:* In the case of a link $X \rightarrow Y$ down or metric increase event, a router $R$ computes $rSPT_{old}(X \rightarrow Y)$ to obtain its rank. By doing this, it also computes the set of its neighbors that were using it to reach $Y$. This set of neighbors will compose the waiting list of $R$. When this waiting list empty, i.e., when $Rank(R) = 0$, $R$ can update its FIB directly. When a router has updated its FIB, it sends a completion message to the neighbors that it was using to reach

$X \rightarrow Y$. When a router $R$ receives a completion message from one neighbor, it removes the sender from its waiting list. When the waiting list of $R$ becomes empty, it is allowed to update its FIB and send its own completion message.

When a router receives a completion message from a neighbor, it knows that the sender has updated its FIB by respecting the ordering. Indeed, the sender could only send the completion message because the computed delay for its FIB update obtained by the ranking has elapsed or because its Waiting List has been emptied. In other words, when the Waiting List of a router $R$ becomes empty, all the routers that were using $R$ to reach $X \rightarrow Y$ have sent their completion message, so that all of them have updated their FIB.

*2) Link up or metric decrease:* In the case of a link $X \rightarrow Y$ up or metric decrease event, a router $R$ recomputes $SPT(R)$ to determine the FIB updates that are required and its rank. If $X \rightarrow Y$ is in its new SPT, $R$ will have to reroute after its nexthops for $X$. Those nexthops will compose its waiting list for the event. When a router updates its FIB, it will send a completion message to its neighbors. When a router receives a completion message from one neighbor, it removes the sender from its waiting list. When the Waiting List becomes empty, it is allowed to update its FIB and send its own completion message.

The ordering is still respected as if the Waiting List of a router $R$ is empty, all the routers on the paths from $R$ to $X \rightarrow Y$ have sent their completion message, so that all of them have updated their FIB.

### B. Shared Risk Link Group events

*1) SRLG down or SRLG metric increase:* Each router will maintain one waiting list associated with each link being shut down during the rSPT computations. A rerouting router R will update its FIB for a destination $d$ (which means that its paths to $d$ contain one or more links of the SRLG) once it has received the completion messages that unlock the FIB update in $R$ for one of the links being shut down. When updating its FIB, $R$ selects the outgoing interfaces for destination $d$ according to the new topology, i.e., by considering the removal or the metric increase of all the affected links.

The meaning of a completion message concerning a link $l$ sent by a router $R$ is that $R$ has updated its FIB for all the destinations that it was reaching via $l$ before the event.

Let us now show that if a packet with destination $d$ reaches a rerouting router $R$ that has not performed its FIB update for destination $d$, then all the routers on its paths to $d$ cannot have performed a FIB update for $d$.

If $R$ has not updated its FIB for destination $d$, it cannot have sent a completion message for any of the failing links $l$ that it uses to reach $d$. The failing links that a router $N$ on $Paths_{old}(R, d)$ uses to reach $d$ are used by $R$ to reach $d$, so that $N$ cannot have received all the necessary completion messages for any of those links. In other words, $R$ did not send a completion message for the links that it uses to reach $d$. Thus $R$ locks the FIB update for those links along its paths towards them.

In Figure 4, we provide the pseudocode that implements the ordering with completion messages. To process the metric increase (or shutdown) of a set of link $S$, a router $R$ will compute the reverse SPT rooted on each link $l$ belonging to $S$, that it uses in its current, outdated SPT. During this computation, it will obtain the rank associated with $l$. It will then record the nexthops that it uses to reach $l$ in a list $I(l)$. These are the neighbors to which it will send a completion message concerning link $l$. If the rank associated with a link is equal to zero, then $R$ updates its FIB directly for the destinations that it reaches via this link, and it sends a completion message to the corresponding nexthops. In the other cases, $R$ builds the waiting list associated with $l$, containing the neighbors that are using $R$ to reach $l$, and it starts the timer considering the rank associated with this link.

Once a waiting list for a link $l$ becomes empty or its associated timer elapses, $R$ can update its FIB for all the destinations that it reached via this link and send its own completion message $CM(l)$ towards the neighbors that it used to reach the link.

*2) SRLG up or SRLG metric decrease:* In the case of a set of link up or link metric decrease events, each router will maintain a Waiting List associated with each link being brought up in the network. For each affected link, its associated Waiting List is the same as for the single link case.

A router $R$ is allowed to reroute packets for a destination $d$ to a new nexthop $N$ when it has received the completion messages from $N$ associated with all the affected links of at least one of the equal cost paths between $N$ and $d$ in the new SPT of $R$.

A router $R$ will send completion messages for a link $X \rightarrow Y$ to its neighbors once it has updated its FIB for the destinations that it reaches via $X \rightarrow Y$ and the affected links for which it already sent a completion message. Note that if there are some destinations that $R$ now reaches via $X \rightarrow Y$ and some other upcoming links, the fact that $R$ sent a completion message for the link $X \rightarrow Y$ does not mean that $R$ has updated its FIB for this destination. It means that $R$ has updated its FIB for the destinations that are only reached via the new upcoming link $X \rightarrow Y$. When a router has sent completion messages for a set of upcoming links $S$, it means that it has updated its FIB for all the destinations that it reaches via any subset of $S$.

When there are equal cost paths between $N$ and $d$, $R$ has the choice to deviate packets destined to $d$ towards $N$ when $N$ has sent the completion messages associated with all the upcoming links on all those paths, or when $N$ has sent the completion messages associated with all the upcoming links belonging to at least one of those equal cost paths.

In Figure 5, we present the pseudocode that implements the ordering with completion messages. We only present the one which allows a FIB update for a destination $d$ in a router $R$, towards a new neighbor $N$, as soon as $N$ uses one of its post-convergence equal cost paths towards $d$.

To process the metric decrease (or the installation) of a set of links $S$, a router $R$ will compute $SPT_{new}$ to obtain the FIB updates that must be performed. Then, the router initializes a set ($Rerouted$) containing the destinations for which an update has already been sent to the line cards, and a set ($CMSent$), containing the set of upcoming links for

```
Metric increase event for a set of Link S processed by router R:
//Computation of the rSPTs of the affected links used by R
foreach Link X → Y ∈ S do
    if X → Y ∈ SPT_old(R) then
        //Computation of the rSPT
        LinkRSPT = rSPT(X → Y);
        //Computation of the rank
        LinkRank = depth(R, LinkRSPT);
        //Computation of the set of neighbors to which a
        //completion message concerning this link will be sent
        I(X → Y)=Nexthops(R, X → Y);
        if LinkRank ==0 then
            //R is a leaf in rSPT(X → Y),
            //it can update its FIB directly
            foreach d : X → Y ∈ Path_old(R, d) do
                UpdateFIB(d);
            end
            //R can send its completion message for this link.
            foreach N ∈ I(X → Y) do
                send(N, CM(X → Y));
            end
        end
        else
            //R is not a leaf in rSPT(X → Y),
            //Computation of the waiting list.
            WaitingList(X → Y)= Childs(R,LinkRSPT);
            //Start the timer associated with this link.
            StartTimer(X → Y, LinkRank * MAXFIBTIME);
        end
    end
end

Upon reception of CM(X → Y) from Neighbor N :
WaitingList(X → Y).remove(N);

Upon (WaitingList(X → Y).becomesEmpty() ||
    Timer(X → Y).hasExpired()) :
//All the necessary completion messages have been received for
//the link or the timer associated with this link has expired
//Update the FIB for each destination that was reached
//via this link.
foreach d : X → Y ∈ Path(R, d) do
    UpdateFIB(d);
end
//Send the completion messages to the neighbors that were
//used to reach this link.
foreach N ∈ I(X → Y) do
    send(N, CM(X → Y));
end
```

**Fig. 4**: Processing of a set of link metric increase events

which it has already sent a completion message. The first set is useful if more than one new outgoing interfaces will be used for some destinations. The second set will permit to avoid sending duplicates of completion messages.

$R$ must then build the waiting lists associated with each of the affected links that it will use. When $R$ receives a completion message for a link $X \rightarrow Y$, it applies the procedure $followNewSPT$. This procedure will perform the FIB updates that are unlocked by the reception of the completion message. The reception of $CM(X \rightarrow Y)$ from $N$ means that $N$ is using at least one post-convergence path for the destinations that are below $X \rightarrow Y$ in $SPT(N)$. It also means that $N$ does not use any outdated path towards those destinations. $R$ can thus follow its own $SPT$ and deviate to $N$ the packets towards the destinations that it will reach via $N$ and $X \rightarrow Y$. The SPT will be followed from $X \rightarrow Y$ until $R$ reaches another upcoming link within this part of its SPT. At that time, if a completion message concerning this link had already been received from $N$, then $R$ is allowed to follow its SPT further on and perform the unlocked FIB updates.

The first time a new nexthop for a destination $d$ is installed in the FIB of a router, all the nexthops that will no longer be used to reach $d$ are removed from its FIB. If an additional (equal cost) nexthop is discovered later for $d$, it will simply be added because $d$ will belong to $Rerouted$ at that time.

The first time an upcoming link is followed by the $followNewSPT$ procedure, and the corresponding updates are performed, the router will send a completion message for this link. If the link is followed again, because the router has multiple paths towards this link, no additional completion message will be sent because the link will belong to $CMSent$ at that time.

### C. Router and Line card events

*1) Router and Line card down events:* Let us consider that a line card of a router $X$ is to be removed, or that $X$ is to be shut down.

The waiting list of a router $R$ for such an event contains the neighbors of $R$ that are below $R$ in $rSPT_{old}(X)$. These are the neighbors of $R$ that were using $R$ to reach $X$. If $R$ is a leaf in $rSPT_{old}(X)$, it is allowed to update its FIB directly, and send a completion message to its nexthops for $X$. If $R$ is not a leaf, then it waits for completion messages from its neighbors. When a router $R$ receives a completion message specifying the router or line card down event in $X$, it removes the sender from its Waiting List. When this Waiting List becomes empty, $R$ is allowed to perform its FIB update and then send its own completion messages to its nexthops to $X$.

When $X$ has received the completion messages from all its neighbors, it is allowed to actually shut itself down or shut the line card down. During the whole convergence phase, when a packet reaches a router $R$ that has not updated its FIB for this destination, its nexthops for this destination did not receive a completion message from $R$, so that they also have outdated FIB. This property can be verified hop by hop along the path from $R$ to $X$, so that the packet will reach $X$ and be forwarded to a neighbor of $X$ whose paths towards the destination is not affected by the event.

*2) Router and Line card up events:* When a router $X$ or a line card of $X$ is brought up in the network, the Waiting List of a router $R$ contains the neighbors of $R$ that $R$ will use to reach $X$. $X$ will be the first router to update its FIB, and will send a completion message to all its neighbors. When a router $R$ receives a completion messages specifying the router or line card up event in $X$, it removes the sender from its Waiting List. When this Waiting List becomes empty, $R$ is allowed to perform its FIB update and send its own completion messages to all its neighbors.

During the whole convergence phase, when a packet reaches a router $R$ that has updated its FIB, it is sure that the nexthop for its destination has sent a completion message to $R$, so that this nexthop has also updated its FIB. This property can be verified hop by hop along the path from $R$ to $X$, so that

```
Metric decrease event for a set of Link S processed by router R:
SPT_new = recomputeSPT();
//Compute the set of updates that will be performed on the FIB
nexthopsUpdates = getNexthopUpdates(SPT_new);
//Initialize the set of Link in S for which a completion message
//has been sent.
CMSent = {};
foreach Link X → Y ∈ S : X → Y ∈ SPT_new do
    //Get the nexthops used to reach the upcoming links.
    //The new nexthops are used if these have changed
    WaitingList(X → Y) = getNexthops(X);
end

Upon reception of CM(X → Y) from neighbor N :
WaitingList(X → Y).remove(N);
//Perform the updates that are unlocked by this
//completion message;
if X → Y ∈ SPT_new and X reached via N then
    followNewSPT(Y,N);
    if not CMSent.contains(X → Y) then
        SendToNeighbors(CM(X → Y));
        CMSent.add(X → Y);
    end
end

followNewSPT(Y,N):
//Explore the graph and perform the necessary FIB updates
if nexthopUpdates.contains(destination Y, nexthop N) then
    //Add nexthop N for destination Y.
    //First call to SendFIBUpdateToLC(Y, .) will remove
    //the nexthops that are not used anymore
    //to reach Y from the FIB in the LineCards
    SendFIBUpdateToLC(Y,N);
end
//FIB updated for destination Y if needed,
//Update the FIB for the destinations behind Y in the new SPT.
foreach Link Y → T ∈ SPT_new do
    if Y → T ∈ S then
        if not WaitingList(Y → T).contains(N) then
            //N already sent a CM for this upcoming link
            followNewSPT(T,N);
            if not CMSent.contains(Y → T) then
                SendToNeighbors(CM(Y → T));
                CMSent.add(X → Y);
            end
        end
        else
            //Do nothing, this part of the SPT will be followed
            //when N sends the necessary completion message.
        end
    end
    else
        // This link is not an upcoming link, N sent the
        //necessary completion messages to continue the update
        //of the destinations behind this link.
        followNewSPT(T,N);
    end
end
```

**Fig. 5**: Processing of a set of link metric decrease events

the packet will reach $X$ and will then be forwarded on a path containing routers whose paths towards the destination are not affected by the event.

## VI. Convergence time in ISP networks

In this section, we analyze by simulations the convergence time of the proposed technique, in the case of a link down event. The results obtained for link up events are very similar. Indeed, the updates that are performed in the FIB of each router for the shutdown of a link impact the same prefixes for the linkup of the link. The only difference in the case of a link up is that the routers do not need to compute a reverse Shortest Path Tree.

As no packets are lost during the convergence process, we cannot define the convergence time as the time required to bring the network back to a consistent forwarding state, as it would always be equal to zero. What is interesting to evaluate here is the time required by the mechanism to update the FIB of all the routers by respecting the ordering. A short convergence time is desired because other events occurring in the network during the ordered convergence process will force the routers to fall back to a fast, non loopfree, convergence, and we want to make this as rare as possible.

To perform this analysis, we took the measurements of [5] that presented the time to perform a SPT computation and a FIB update on current high-end routers. The ordering of the FIB update requires to compute the new Shortest Path Tree, and the computation of a reverse Shortest Path Tree in the case of a link down event. The Waiting List can be computed on the fly of the SPT computation, so that we only introduced a fixed amount of time to consider the computation of those lists.

We also added a fixed Hold Down before the process starts, in order to ensure that all the routers have received the link state packet describing the topology change before the scheme begins. We set the hold time before completion messages are being sent to 200 msec. This is a very large value compared to the time required to perform a SPT computation and a rSPT computation on the topologies under study. So, in our simulations, routers were ready to perform their FIB updates and send their completion messages when this hold time elapses.

Note that a router will start this Hold Down Timer as soon as it receives the Link State Packet describing the topology change. Thus, the time at which the Hold Down Timer expires on each router depends on the flooding time of link-state packets in the network. We also took the measurements of [5] to obtain the delay that is required to flood a link state packet from the router where the shutdown is performed towards the other routers in the network.

We assume that the time required to parse and process a Completion Message is similar to the time required to parse a Link-State Packet and insert it in the link-state database, i.e., a value between 2 msec and 4 msec [5]. When a router sends a completion message to a neighbor, it is thus removed from the neighbor's waiting list after the delay of the link on which the message is sent plus the time required to process a link-state

packet. The time required to perform the FIB update in each rerouting router is obtained by computing their new FIB and multiplying the number of prefixes to update by the time to perform a prefix update that we obtained in the measurements (i.e., 100 $\mu$sec per prefix). The number of prefixes associated with each router is obtained from an IS-IS trace. A summary of the parameters of the simulation is presented in Table I.

TABLE I
SIMULATION PARAMETERS

| | |
|---|---|
| lsp_process_delay | [2,4]ms |
| update_hold_down | 200ms |
| (r)spf_computation_time | [20,30]ms in Tier-1 ISP<br>[2,4]ms in GEANT |
| fib_prefix_update_delay | 100 $\mu s$/prefix |
| completion_message_process_delay | [2,4]ms |
| completion_message_sending_delay | [2,4]ms |

Our simulations work as follows. Upon an event, the link-state packet is flooded through the network. Upon reception of the link-state packet, each router starts its Hold Down Timer and computes its SPT, rSPT, and its Waiting List. When their Hold Down Timer expires, the routers that have an empty Waiting List perform their FIB update, and send their completion messages. When a router has finished the computation of its SPT and rSPT, it considers the completion messages that it has received. When a router has a non empty Waiting List, it waits for it to become empty, and then it performs its FIB update and sends its own completion message. For each link down event under study (link-id on the x-axis), we plot the time at which all the routers have updated their FIB, so that all the operations implied by the scheme have been performed. We sorted the link-ids according to the obtained convergence times.

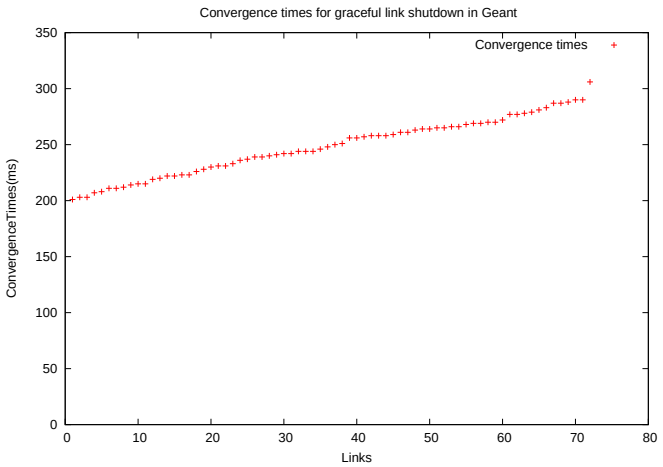


Fig. 6: Convergence times in Geant

Figure 6 shows the convergence times considering the removal of each directed link of Geant, an European research network containing 22 nodes and 72 (directed) links. We can see that, even if FIB updates are delayed, the convergence time remains short and the main component of the convergence is the fixed 200 msec hold time. The worst-case convergence time with the solution is 50 msec longer than the convergence time presented on the same topology in [5], when the same hold time is used.

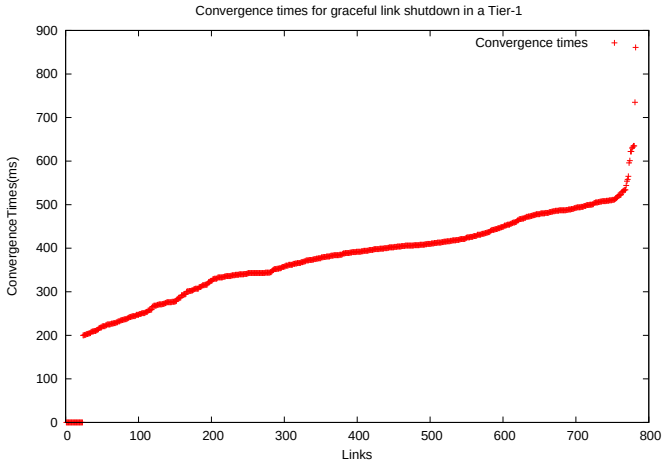


Fig. 7: Convergence times in a Tier-1 ISP

Figure 7 shows the convergence times considering the removal of each directed link of a Tier-1 ISP. The values of 0 correspond to the shutdown of 23 directed links that did not carry packets due to their large IGP metric. This number is odd, which can be explained by the fact that some links have asymmetrical metrics, so that one direction of the link is used while the other not. The worst loop-free convergence time was 861 msec. This can be explained by the fact that the rSPT of this link contained a branch of 4 routers that had to perform a FIB update that lasted approximately 120 msec. The other components of the convergence are the 200 msec to compute the SPT and rSPT, and the delays of the links on which the completion messages were sent. Compared to [5] the convergence time is in the worst-case 400 msec longer than the convergence time when loops are not avoided.

To conclude, this analysis shows that a sub-second convergence is feasible even if a loop avoidance mechanism is used. The increase in the convergence time compared to the convergence time without the loop avoidance mechanism is small. With the solution operators could shut down links in their topology without loosing packets, by letting the network adapt to the change and stop using the link within one second, so that the use of the mechanism would not be a constraint for the operators.

In order to reduce the delaying of the FIB Updates as much as possible, we combined the proposed solution with a technique that lets a router find if its new nexthop for a destination already provides a loop-free path. So that, in some cases, routers can safely update their FIB for the destination without respecting the ordering. In the next section, we will briefly explain this technique, and we will evaluate the provided gain in the convergence time.

## VII. Ranking Shortcuts

As explained in the previous section, the motivation for shortcuts is to reduce as much as possible the delaying of the FIB updates, which is the interval between the moment at which a router is ready to update its FIB for a destination by using the nexthops corresponding to the new shortest paths through the network, and the moment at which the router actually does it.

In this section, we will show that a router applying the proposed ordering scheme will implicitely compute a sufficient information to decide wether it can shortcut the scheme and perform its FIB update directly, while preserving the transient forwarding consistency accross the network.

The decision to use this optimization is local to the router, i.e., each router can independently decide to apply the shortcut or not.

In the case of a link $X \rightarrow Y$ down or metric increase event, a router $R$ computes $rSPT(X \rightarrow Y)$. From this tree, $R$ obtains the set of routers that are using $R$ to reach $Y$ via $X \rightarrow Y$.

By doing this, $R$ also computes the set of unaffected routers, i.e., the routers that do not use the link $X \rightarrow Y$ at all. These are the routers in $rSPT(Y)$ that do not have a path towards $Y$ that contains $X \rightarrow Y$. Routers that are below $X \rightarrow Y$ can be marked during the computation of the rSPT, so that, at the end of the computation, a router $N$ that is not marked is known to be an unaffected router, so that $X \rightarrow Y \notin SPT(N)$.

The shortest paths from this router to the destinations that $R$ will have to reroute will not change, so that if the new nexthops of $R$ for one destination belong to this set of unaffected routers, $R$ is allowed to directly reroute the destination towards these new nexthops by disregarding its rank or the state of its Waiting List.

Several implementations of this shortcut are possible. Firstly, one router can decide to perform a full FIB update by shortcutting its rank if all the new nexthops to which it will reroute packets are unaffected routers. Secondly, a router can decide, destination per destination, if the set of new nexthops for one destination only contains unaffected routers. When this is done, the router is allowed to update its FIB for those destinations directly, and perform a second FIB update with the remaining destinations by respecting its rank or when its Waiting List becomes empty.

The first solution is the simplest, and preserves the property that routers update their FIB in one shot in the case of a single link event. The second solution is more complex, but this shortcut will be applicable more often.

To evaluate the gain of such shortcuts, we performed the same analysis as presented in Section VI, by considering the first shortcut solution. More precisely, when the Hold Down Timer expires in a router which is allowed to apply the shortcut, the router performs its FIB update directly. Note that this router will not send its completion message before its Waiting List is empty, in order not to change the meaning of a completion message. But, when a router has already performed its FIB update when its Waiting List becomes empty, it is allowed to send its own completion message directly.

In Geant, the gain was negligible. This can be explained by the fact that a small amount of prefixes are advertised in Geant, so that the FIB update time component is negligible compared to the Hold Down time, and the sending of completion messages through the network.

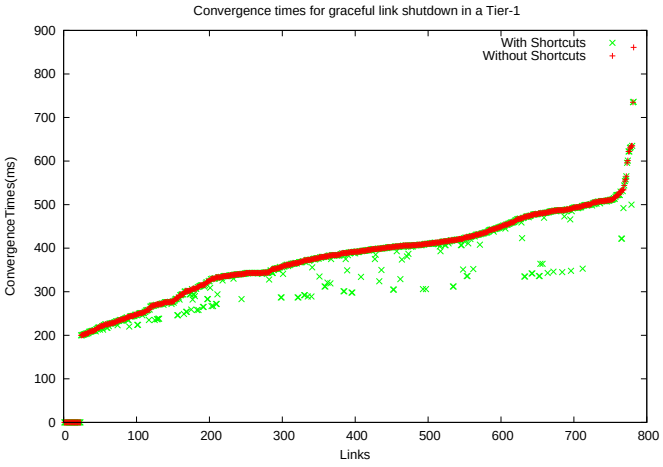


Fig. 8: Convergence times in a Tier-1 ISP

In the tier-1 ISP, the gain of the shortcut is more perceptible, because many prefixes are advertised in the network, and in many link maintenance cases, the rerouting routers were allowed to do the shortcut. For example, in the worst-case convergence time of 861 msec without shortcuts, the convergence time with shortcuts is 736 msec. In fact, some of the routers that were contributing to this long convergence time could safely perform their FIB updates in parallel.

We analyzed the coverage of both shortcut mechanisms, and found out that in the Tier-1 ISP, 54 % of the FIB updates that had to be performed by routers during the analysis could be shortcut with the first solution. With the second shortcut solution, 69 % of the FIB updates could be shortcut for at least one prefix. The second shortcut solution does not provide a significative gain in coverage. As the goal of the scheme was to permit an ordered convergence where, in the case of a single link event, all the prefixes can be updated in one shot, we think that the first solution is to be preferred over the second. As the application of any shortcut solution can be decided independently by each router of the network, the choice of applying one method or another or not applying a shortcut at all can be made according to the software design and performance of each router of the network.

## VIII. Related Work

The problem of avoiding transient loops during IGP convergence has rarely been studied in the literature although many authors have proposed solutions to provide loop-free routing. An existing approach to loop-free rerouting in a link-state IGP [31] requires that the rerouting routers take care of routing consistency for each of their compromised destinations, separately. In fact, those mechanisms were inspired by distance-vector protocols providing a transiently loop-free

convergence [32]. With this kind of approach, a router should ask and wait clearance from its neighbors for each destination for which it has to reroute. This implies a potentially large number of message exchanged between routers, when many destinations are impacted by the failure. Every time a router receives clearance from its neighbors for a given destination, it can only update forwarding information for this particular one. This solution would not fit well in a Tier-1 ISP topology where many destinations can be impacted by a single topological change. Indeed, in such networks, it is common to have a few thousands of prefixes advertised in the IGP [5]. Note that those solutions do not consider the problem of traffic loss in the case of a planned link shutdown.

In [33], a new type of routing protocol allowing to improve the resilience of IP networks was proposed. This solution imposes some restrictions on the network topology and expensive computations on the routers. Moreover, they do not address the transient issues that occur during the convergence of their routing protocol. In [34], extensions to link-state routing protocols are proposed to distribute link state packets to a subset of the routers after a failure. This fastens the IGP convergence, but does not solve the transient routing problems and may cause suboptimal routing.

In [10], transient loops are avoided when possible by using distinct FIB states in each interface of the routers. Upon a link failure, the network does not converge to the shortest paths based on the new topology. Indeed, the failure is not reported. Instead, the routers adjacent to the failed link forward packets along alternate links, and other routers are prepared to forward packets arriving from an unusual interface in a consistent fashion towards the destination. As such, the solution is a Fast Reroute technique. Our solution is orthogonal to [10] as our goal is to let the network actually converge to its optimal forwarding state by avoiding transient forwarding loops when a Fast Reroute mechanism has been activated, or when the failure is planned.

In [11], transient loops are avoided by selectively discarding the packets that are caught in a loop, during a fast convergence phase following an unplanned event. The idea is to also to use distinct FIB states in each interface of the routers, and let routers drop packets when they would be caught in a loop. Care has been taken to avoid dropping a packet arriving from an unusual interface if the router cannot ensure that the packet is actually caught in a loop. Once again, our goals differ as we focus on transient loops occuring during the convergence from an initial forwarding state to the optimal forwarding state based on the new topology.

In [35], we propose an alternative approach to avoid transient loops in the case of maintenance operations. The technique uses progressive reconfigurations of the metric of the link whose state is modified, ensuring that each step of the process provides a loopfree convergence. The advantage of this technique is that it does not require modifications to IS-IS or OSPF in order to be deployed, as modifying the metric of a link is already doable. On the other hand, if the number of intermediate metrics required to achieve a loopfree convergence is large, the convergence time can become long compared to the technique proposed in this paper.

## IX. Conclusion

In this paper, we have first described the various types of topology changes that can occur in large IP networks. Recent measurements indicate that many of those changes are non-urgent. When such a non-urgent change occurs, the FIB of all routers must be updated. Unfortunately, those updates may cause transient routing loops and each loop may cause packet losses or delays. Large ISPs require solutions to avoid transient loops after those non-urgent events.

The first important contribution of this paper is that we have proved that it is possible to define an ordering on the updates of the FIBs that protects the network from transient loops. We have proposed an ordering applicable for the failures of protected links and the increase of a link metric and another ordering for the establishment of a new link or the decrease of a link metric. We also proposed orderings that are applicable in the case of a non-urgent router down or up event, as well as line card events. Then, we generalized the scheme to events affecting any kind of sets of links in the network. Next, we presented optimizations to the scheme that allow routers to update their FIB by disregarding the proposed ordering when it is proved not to lead to forwarding loops.

Finally, we have shown by simulations that our loop-free extension to currently deployed link-state protocols can achieve sub-second convergence in a large Tier-1 ISP.

## Acknowledgments

This work was supported by Cisco Systems within the ICI project. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of Cisco Systems.

We would like to thank Stewart Bryant, Mike Shand, Clarence Filsfils and Stefano Previdi for their suggestions and comments on this work. We would like to thank Ariel Orda and the anonymous reviewers of this paper for their suggestions and comments.

## References

[1] P. Francois and O. Bonaventure, “Avoiding transient loops during IGP Convergence in IP Networks,” in *Proc. IEEE INFOCOM*, March 2005.

[2] J. Moy, “OSPF version 2,” Internet Engineering Task Force, Request for Comments 1247, July 1991.

[3] ISO, “Intermediate system to intermediate system routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (iso 8473),” ISO/IEC, Tech. Rep. 10589:2002, April 2002.

[4] G. Iannaccone, C. Chuah, S. Bhattacharyya, and C. Diot, “Feasibility of IP restoration in a tier-1 backbone,” *IEEE Network Magazine*, January-February 2004.

[5] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, “Achieving sub-second IGP convergence in large IP networks,” *Computer Communication Review*, vol. 35, no. 3, pp. 35–44, 2005.

[6] C. Alaettinoglu, V. Jacobson, and H. Yu, “Towards millisecond IGP congergence,” November 2000, internet draft, draft-alaettinoglu-ISIS-convergence-00.ps, work in progress.

[7] P. Pan, G. Swallow, and A. Atlas, “Fast Reroute Extensions to RSVP-TE for LSP Tunnels,” May 2005, Internet RFC 4090.

[8] M. Shand and S. Bryant, “IP Fast Reroute Framework,” October 2006, internet draft, draft-ietf-rtgwg-ipfrr-framework-06.txt.

[9] S. Bryant, C. Filsfils, S. Previdi, and M. Shand, “IP Fast Reroute using tunnels,” November 2005, internet draft, draft-bryant-ipfrr-tunnels-03.txt, work in progress.

[10] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, and C.-N. Chuah, "Proactive vs. reactive approaches to failure resilient routing," in *Proc. IEEE INFOCOM"*, March 2004.

[11] Z. Zhong, R. Keralapura, S. Nelakuditi, Y. Yu, J. Wang, C.-N. Chuah, and S. Lee, "Avoiding Transient Loops Through Interface-Specific Forwarding," in *IWQoS*, 2005, pp. 219–232.

[12] A. Atlas and A. Zinin, "Basic Specification for IP Fast-Reroute: Loop-free Alternates," February 2006, internet draft, draft-ietf-rtgwg-ipfrr-spec-base-05.

[13] U. Hengartner, S. Moon, R. Mortier, and C. Diot, "Detection and analysis of routing loops in packet traces," in *Proceedings of the second ACM SIGCOMM Workshop on Internet measurment*. ACM Press, 2002, pp. 107–112.

[14] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *IEEE Infocom2004*, Hong Kong, March 2004.

[15] S. Spadaro, J. Solé-Pareta, D. Careglio, K. Wajda, and A. Szymanski, "Positioning of the RPR standard in contemporary operator environments," *IEEE Network Magazine*, vol. 18, March-April 2004.

[16] A. Atlas, R. Torvi, G. Choudhury, C. Martin, B. Imhoff, and D. Fedyk, "IP/LDP Local Protection," February 2004, internet draft, draft-atlas-ip-local-protect-00.txt, work in progress.

[17] N. Dubois, B. Fondeviole, and N. Michel, "Fast convergence project," January 2004, presented at RIPE47, http://www.ripe.net/ripe/meetings/ripe-47/presentations/ripe47-routing-fcp.pdf.

[18] R. Teixeira and J. Rexford, "Managing routing disruptions in Internet Service Provider networks," *IEEE Communications Magazine*, March 2006.

[19] G. Bernstein, B. Rajagopalan, and D. Saha, *Optical Network Control: Architecture, Protocols, and Standards*. Addison-Wesley Professional, 2003.

[20] P. Pongpaibool, R. Doverspike, M. Roughan, and J. Gottlieb, "Handling IP Traffic Surges via Optical Layer Reconfiguration," *Optical Fiber Communication*, 2002.

[21] N. Shen and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering tunnels," October 2004, Internet RFC 3906.

[22] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communications Magazine*, October 2002.

[23] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "NetScope: Traffic Engineering for IP Networks," *IEEE Network Magazine*, March 2000.

[24] J. Lepropre, S. Balon, and G. Leduc, "Totem: A toolbox for traffic engineering methods," Poster and Demo Session of INFOCOM'06, April 2006. [Online]. Available: ftp://ftp.run.montefiore.ulg.ac.be/pub/RUN-PP06-08.pdf

[25] A. Shaikh, R. Dube, and A. Varma, "Avoiding Instability during Graceful Shutdown of OSPF," in *Proc. IEEE INFOCOM*, June 2002.

[26] M. Shand and L. Ginsberg, "Restart signaling for IS-IS," July 2004, Internet RFC 3847.

[27] J. Moy, P. Pillay-Esnault, and A. Lindem, "Graceful OSPF Restart," November 2003, Internet RFC 3623.

[28] P. Francois, O. Bonaventure, M. Shand, S. Bryant, and S. Previdi, "Loop-free convergence using oFIB," December 2006, internet draft, draft-ietf-rtgwg-ordered-fib-00, work in progress.

[29] http://www.geant.net/.

[30] O. Bonaventure, P. Francois, M. Shand, and S. Previdi, "ISIS extensions for ordered FIB updates," February 2006, internet draft, draft-bonaventure-isis-ordered-00.txt, work in progress.

[31] J. J. Garcia-Luna-Aceves, "A unified approach to loop-free routing using distance vectors or link states," *SIGCOMM Comput. Commun. Rev.*, vol. 19, no. 4, pp. 212–223, 1989.

[32] J. M. Jaffe and F. H. Moss, "A Responsive Distributed Routing Algorithm for Computer Networks," *IEEE Trans. Commun.*, pp. 30:1758–1762, July 1982.

[33] G. Schollmeier, J. Charzinski, A. Kirstdter, C. Reichert, K. Schrodi, Y. Glickman, and C. Winkler, "Improving the Resilience in IP Networks," in *High performance switching and routing (HPSR'03)*, Torino, June 2003.

[34] P. Narvez, K.-Y. Siu, and H.-Y. Tzeng, "Local restoration algorithms for link-state routing protocols," in *Proceedings of the 1999 IEEE International Conference on Computer Communications and Networks*, 1999.

[35] P. Francois, M. Shand, and O. Bonaventure, "Disruption free topology reconfiguration in OSPF Networks," in *Proc. IEEE INFOCOM*, Anchorage, USA, May 2007.



**Pierre Francois** is currently finishing his Ph. D. at the Université catholique de Louvain (UCL), Belgium. He obtained his MS degree in computer science from the University of Namur, Belgium. He received the Infocom best paper award in 2007. His research interests include intra- and interdomain routing. (http://inl.info.ucl.ac.be/) e-mail: pierre.francois@uclouvain.be



**Olivier Bonaventure** is currently a professor in the Department of Computing Science and Engineering at Université catholique de Louvain (UCL), Belgium. From 1998 to 2002 he was a professor at the Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium. Before that, he received a Ph.D. degree from the University of Liège and spent one year at the Alcatel Corporate Research Center in Antwerp. He is on the editorial board of IEEE Network Magazine and IEEE/ACM Transactions on Networking. He received the Wernaers and Alcatel prizes awarded by the Belgian National Fund for Scientific Research in 2001. His current research interests include intra- and interdomain routing, traffic engineering, and network security. (http://inl.info.ucl.ac.be) e-mail : olivier.bonaventure@uclouvain.be