This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.

Disruption free topology reconfiguration in OSPF networks

Pierre Francois
Dept CSE
Université catholique de Louvain (UCL)
Belgium
francois@info.ucl.ac.be

Mike Shand
Cisco Systems
mshand@cisco.com

Olivier Bonaventure
Dept CSE
Université catholique de Louvain (UCL)
Belgium
Bonaventure@info.ucl.ac.be

***Abstract*— A few modifications to software and/or hardware of routers have been proposed recently to avoid the transient micro loops that can occur during the convergence of link-state interior gateway protocols like IS-IS and OSPF. We[1] propose in this paper a technique that does not require modifications to IS-IS and OSPF, and that can be applied now by ISPs. Roughly, in the case of a manual modification of the state of a link, we progressively change the metric associated with this link to reach the required modification by ensuring that each step of the progression will be loop-free. The number of changes that are applied to a link to reach the targeted state by ensuring the transient consistency of the forwarding inside the network is minimized. Analysis performed on real regional and tier-1 ISP topologies show that the number of required transient changes is small. The solution can be applied in the case of link metric updates, manual set up, and shut down of links.**

I. Introduction

Internet Service Providers have to cope with more and more stringent Service Level Agreements (SLA), that are justified by the increasing use of their networks to transport voice, video, and TV broadcast traffic accross their networks. Such SLA generally define uper bounds on the packet loss ratio and on the duration of losses of connectivity.

These losses of connectivity are generally caused by network topology changes, which are common events in large IP networks. In a study of the Sprint IP backbone, Markopoulos et al. report in [1] that 20% of the failures were caused by maintenance operations. Several other studies reveal that maintenance operations occur frequently [2]. Most operators perform those maintenance operations during nightly maintenance windows to reduce their impact on the traffic. However, this increases the cost of operating the network and network operators must work during night shifts. Some ISPs have even defined procedures [3] where the network operators must first set the IGP metric of a link to MAX_METRIC - 1 to "gracefully" reroute the traffic before actually shutting down the link. In IP over optical networks, the optical network, and thus the topology, can be regularly reconfigured according to the needs of the operators [4]. Other topology changes include the modification of IGP weights for traffic engineering purposes. Many techniques to tune the IGP weights have been proposed [5] and some are implemented in network optimisation tools that are used by ISPs. Several networks using MPLS also change the IGP weight associated to MPLS tunnels [6].

All those changes to the network topology affect the traffic passing through the network. Besides the long-term effect of moving traffic, each topology change forces the routers to recompute and update their Forwarding Information Bases. During these updates, transient routing loops can occur. Measurement studies performed in the Sprint network notably have shown that such transient loops were real and could last for more than several seconds [7].

Each of these transient loops will cause packet losses and may prohibit the provider from meeting the promised connectivity recovery time and packet loss ratio specified in their service level agreements. This is unfortunate when the topological change is predictable or implied by a manual operation. That is, ISPs require solutions to avoid packet loss upon predictable topology changes. Protocol, Software and Hardware modifications have been proposed at the IETF to tackle this problem. However, all of these are works in progress and it will take years before extensions to IS-IS and OSPF are standardized and implementations reach the market.

The goal of this paper is to present a solution relying on progressive reconfigurations of a link metric such that the desired updated state of the link can be reached by never putting the routers of the network in an inconsistent forwarding state during the convergence process. In essence, the solution does not require modifications to the routing protocols or router software, as changing a link metric has always been a feature of Link-State Interior Gateway Protocols.

This paper is organized as follows. We firstly illustrate the problem and the solution with a small example. In Section II we introduce a few notations and the basic properties on which the proposed solution relies, and we prove that there always exists a sequence of metrics that permits to reach the desired link-state without introducing transient forwarding loops. In Section III, we present how to compute short metric sequences that can be used to adapt to a metric increase or the removal of a link by avoiding transient forwarding loops. In Section IV,

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

0743-166X/07/$25.00 ©2007 IEEE

we present the solution for the case of a link metric decrease and a link reactivation. In Section V, we present the results of an analysis performed on ISP topologies, showing that the Merged Reroute Metric Sequences are short in practice. In Section VI, we present the related work, and we conclude the paper in section VII.

II. Loop free convergence using metric increments

To understand the transient routing loops mentioned in the previous section, let us consider the simple example shown in Figure 1. In this network composed of five routers and six links, all links have an IGP metric of 1 except the link between routers A and B whose IGP metric is set to 5. Let us consider what happens when link $B-C$ needs to be shutdown for maintenance reasons. This link can be shutdown in one step, by removing it from the link state database or in two steps as proposed in [3] by first setting its IGP metric to $MAX\_METRIC - 1$ and later removing it from the link state database. In both cases, after the first step all routers must update their FIB. Before the topology change, router B sent the packets towards A via C. After the topology change, it will send the packets via D. Unfortunately, before the topology change, router D was sending the packets towards A via routers B and E. This implies that if router B updates its FIB before router D, a likely event as router B will learn the topology change before router D, then packets destined to A will loop on the $B-D$ link until router D has updated its FIB.

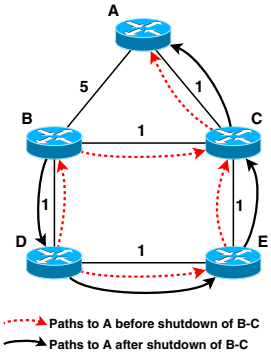

Fig. 1: Simple network

Let us reconsider the example above, we will see that there exists a sequence of metrics for link $B-C$ that permits to shut down the link without causing packet loops and losses. Next, we will show that, in any possible network topology, there always exists a sequence of metric increments that will allow a loopfree convergence for the metric update of a link $A \rightarrow B$ from one value m to another $m' > m$.

Let us assume that the IGP metric of link $B-C$ changes from 1 to 2 in the topology of Figure 1. Before the change, the FIB of all routers is as shown in table I. When the metric of link $A-B$ is set to 2 (in both directions), routers B, C, D and E update their FIB. At router B, the consequence of

Router	A	B	C	D	E
A	-	C	C	C	C
B	C	-	C	D	C and D
C	A	B	-	B and E	E
D	E and B	B	E and B	-	E
E	C	C and D	C	D	-

TABLE I
FIB of all routers when $B-C=1$

Router	A	B	C	D	E
A	-	C	C	C	C
B	C	-	C	D	~~C and~~ D
C	A	B	-	~~B and~~ E	E
D	E ~~and B~~	B	E ~~and B~~	-	E
E	C	~~C and~~ D	C	D	-

TABLE II
FIB of all routers when $B-C=2$

Router	A	B	C	D	E
A	-	C	C	C	C
B	~~C~~ D	-	~~C~~ D	D	D
C	A	~~B~~ E	-	E	E
D	E	B	E	-	E
E	C	D	C	D	-

TABLE III
FIB of all routers when $B-C=4$

the metric change is that it will stop using router C to reach destination E. C will stop using B to reach E and D will stop using B to reach C and A. Thus, the metric change has reduced the number of equal cost paths used by some routers to reach several destinations. It is interesting to note that no transient loops occur during this metric change.

Let us look at what happens when the metric of link $B-C$ changes from 2 to 4. The new FIB of all routers is shown in table III. This change caused routers B and C to update their FIB. Routers B and C no longer use link $B-C$ to reach any destination. As in the previous step, there are no transient loops during this update and with this metric value, link $B-C$ does not carry packets anymore. It can thus be safely shut down by the operator.

Now, let us show that metric sequences allowing a loopfree convergence always exist. We firstly introduce a few notations. $SPT_{A \underset{m}{\rightarrow} B}(X)$ is the shortest path tree of X based on the initial topology where the metric of the link $A \rightarrow B$ is set to m[2]. $Paths(X,Y,S)$ is the set of equal cost paths from X to Y in the shortest path tree S. $Dist(X,Y,S)$ is the IGP distance from X to Y according to the shortest path tree S. When a change in a link metric is performed, we use $Dist(X,Y)$ to denote the distance from X to Y before the change, and

[2]Although the use of Equal Cost Multi Path makes this "tree" actually be an acyclic graph, with potentially more than one shortest path from a source to a destination, we use the term "tree" to respect the IS-IS and OSPF terminology.

$Dist'(X,Y)$ to denote the distance from X to Y after the change. $rSPT(X)$ is the reverse Shortest Path Tree of X. This is a tree containing all the shortest paths from the nodes of the network graph towards X. Note that when Equal Cost Paths are used, this graph is actually an acyclic graph. When a change in a link metric is performed, we respectively denote the rSPT of X before and after the change with $rSPT(X)$ and $rSPT'(X)$.

We say that a change is loopfree for a destination D if transient forwarding loops during the routing convergence cannot occur. A change is loopfree for destination D if the merging of $rSPT(D)$ with $rSPT'(D)$ does not contain a cycle. If it contains a cycle, then there exists an ordering of the FIB updates performed by the routers for destination D that transiently puts the network in an inconsistent forwarding state such that packets can loop. We say that a change is loopfree if it is loopfree for all the nodes of the network.

To prove the existence of a sequence of metric increments that allows a loopfree convergence when updating the metric of a link, we will show that incrementing the metric of the link by 1 never causes transient loops, so that progressively incrementing the metric of a link can be performed to avoid loops.

Theorem II.1 *In a stable network, incrementing the metric of a link $A \longrightarrow B$ by one leads to a loop-free convergence process.*

We can prove this theorem by contradiction. Let us show that it is absurd to have a transient loop in the network when the metric of link $A \rightarrow B$ is increased by one. There can be a loop for a destination D while the routers adapt to the metric change if there exists two distinct nodes X and Y such that X was in the paths from Y to D before the change, and Y will be in the paths from X to D after the change. In other words, there can be a transient loop for packets destined to D if the merging of the rSPT of D before and its rSPT after the change contains a cycle.

$$X \in Paths(Y, D, SPT_{A \underset{m}{\longrightarrow} B}(Y)) \quad (1)$$

$$Y \in Paths(X, D, SPT_{A \underset{m+1}{\longrightarrow} B}(X)) \quad (2)$$

If X was in the paths from Y to D before the change, X was not using Y to reach D before the change, so that if (2) is true, then the new SPT of X is such that one of the shortest paths from X to D contains Y and its length is the length of its initial shortest path to D plus 1 :

$$\begin{aligned} &Dist'(X,Y) + Dist'(Y,D) \\ &= \quad Dist(X,D) + 1 \end{aligned} \quad (3)$$

If Y was using X to reach D before the change, then

$$Dist(Y,D) = Dist(Y,X) + Dist(X,D) \quad (4)$$

In a **first case**, when $Dist(Y,D) = Dist'(Y,D)$, by replacing $Dist'(Y,D)$ in (3) by the value of $Dist(Y,D)$ in (4), we obtain

$$\begin{aligned} &Dist'(X,Y) + Dist(Y,X) + Dist(X,D) \\ &\qquad = Dist(X,D) + 1 \end{aligned} \quad (5)$$

Thus,

$$Dist'(X,Y) + Dist(Y,X) = 1 \quad (6)$$

Which is impossible as X and Y are two distinct nodes and the sum of two path lengths must at least be equal to 2.

In the **other cases**, Dist'(Y,D) is equal to Dist(Y,D)+1, as only one metric of a link has been updated by incrementing it by 1. By replacing $Dist'(Y,D)$ in (3) by the value of $Dist(Y,D)$ in (4) plus one, we obtain

$$\begin{aligned} &Dist'(X,Y) + Dist(Y,X) + Dist(X,D) + 1 \\ &\qquad = Dist(X,D) + 1 \end{aligned} \quad (7)$$

From 7, we obtain

$$Dist'(X,Y) + Dist(Y,X) = 0 \quad (8)$$

Which is impossible as X and Y are two distinct nodes. Thus, it is impossible to increment a link metric by one and verify both (1) and (2), which are necessary for a transient forwarding loop to happen. ∎

We have thus proved that we can always change the metric of a link to a larger metric, by progressively incrementing the metric of the link by one, until the target metric is reached. When the link must be shut down, the metric can be incremented until it becomes so large that the link does not carry packets anymore. When this metric has been reached, the link can be safely shut down.

III. Loop free convergence using Key metric increments

The technique described above is **inefficient** as a large number of increments could have to be used when a link with a low metric must be shut down. To solve this problem, we propose to perform larger increments of the metrics when they are known to provide a loopfree convergence. As the metric space of links is wide in IS-IS and OSPF, it is not realistic to totally explore the metric space and try to find a possible loop free increment sequence for a given link metric transition. Indeed, many operators take advantage of the whole width of the metric space. For example, in the European Geant Research Network [8], there exists a link with a metric of 1 and a link with a metric of $20,000$. Such variety of link metrics is also present in the tier-1 ISP topologies that we analyse in Section V.

Let us consider the topology of Figure 2. If we were to set the metric of the link $B-C$ to 40 with the previous technique, we would have to perform 30 metric changes. In real networks, the utilization of wide metrics is frequent, which would lead to a large amount of increments to be performed.

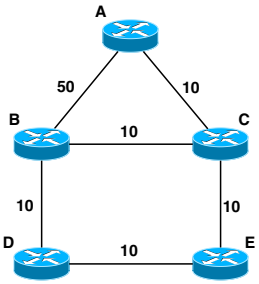

Fig. 2: Simple network with large metrics

However, we can see that the direct update of the metric for link $B-C$, from 11 to 40, could not cause a forwarding loop, so that $\{10, 11, 40\}$ is a valid metric sequence to change the metric of the link without loosing packets.

Now, we identify several key aspects of the transition from one link metric to another, that we will use to reduce the set of metric increments used to perform a progressive loopfree convergence.

A. Reroute Metric Sequences

Let us consider the set of equal cost shortest paths from a source S towards a destination D, such that some of these paths contain a link $A \rightarrow B$. We can identify three different cases when the metric of this link is incremented by 1.

The first case is when the metric increase **does not change the forwarding path from S to D**; except that the new distance from S to D is increased by one, the set of paths from S to D does not change. This implies that all the paths used by S to reach D before the change contained the link $A \rightarrow B$. Indeed, if this was not the case only the paths that do not contain this link would be used after the change, as their length is not affected. Note that in this first case $Dist'(S,D) = Dist(S,D)+1$. For example, when the metric of link $B \rightarrow C$ in Figure 2 is changed from 10 to 11, the paths from B to C do not change, except that the distance between B and C is increased by 1.

The second case is when the metric change **increases** the number of equal cost paths from S to D. This is the case when the paths via the link $A \rightarrow B$ are still among the shortest paths towards D after the change, and other paths to D not via $A \rightarrow B$ now become shortest paths. Note that in this case, $Dist'(S,D) = Dist(S,D)+1$. For example, when the metric of link $B \rightarrow C$ in Figure 2 is changed from 29 to 30, the previous paths from B to C are still used, and another path via D and E is used.

The third case is when the metric change **decreases** the number of equal cost paths from S to D. This is the case when equal cost paths to D, not via $A \rightarrow B$, existed before the change, and are the sole paths being used by S after the change. In this case, $Dist'(S,D) = Dist(S,D)$. For example, when the metric of link $B \rightarrow C$ in Figure 2 is changed from 30 to 31, only the path $B \rightarrow D \rightarrow E \rightarrow C$ is used by B to reach C.

Keeping this in mind, let us focus on a particular ordered sequence of metrics for a link $A \rightarrow B$, considering an initial metric m_1, a target metric m_t, and a destination D initially reached via this link by some routers. This sequence, called "Key Metric Sequence" (KMS), contains m_1, m_t, and all the metrics within $[m_1, m_t]$ for the link $A \rightarrow B$ that will force at least one router R to use an additional equal cost path towards D that does not contain $A \rightarrow B$. We will call m the "Key Metric" for destination D at R if R uses an additional path not via $A \rightarrow B$ when the link metric is set to m.

In Figure 2, the Key Metric Sequence for link $B \rightarrow C$, considering an initial metric of 10, a target metric of 40, and destination A is $\{10, 30, 40\}$. 30 is the Key Metric for destination A at node B since B will start using path $B \rightarrow D \rightarrow E \rightarrow C$ to reach A when the metric is set to 30. 10 is the initial metric, and it is also the Key Metric for destination A at node D since D uses both paths via and not via $B \rightarrow C$ to reach A when the metric of the link is 10.

Computing the KMS of a destination D, considering a link $A \rightarrow B$, its initial metric m_i, and a target metric m_t for this link is simple. We compute the rSPT of D with both initial and target metric for $A \rightarrow B$. When the distance from a node N to D differs in those rSPTs, $m_i + Dist'(N,D) - Dist(N,D)$ is inserted in the sequence. This metric is the one that will let N use paths via as well as not via $A \rightarrow B$ to reach D, so that this value is the Key Metric of N.

Let us consider one KMS $\{m_1, m_2, \ldots, m_i, \ldots, m_t\}$ for a destination D. Let us now insert, between each pair of elements (m_i, m_{i+1}), an intermediate value m'_i equal to $m_i + 1$.

We will show in Theorem III.1 that such a sequence, that we call a Reroute Metric Sequence (RMS) for destination D, is such that the progressive setting of each metric contained in the sequence provides a loop free convergence for D, for each successive metrics in the sequence, until the target metric is reached.

In Figure 1, the Reroute Metric Sequence for link $B \rightarrow C$, considering an inital metric of 10, a target metric of 40, and destination A, is $\{10, 11, 30, 31, 40\}$. If the metric of the link is progressively set to those values, then no transient forwarding loop could occur for destination A.

Theorem III.1 *Given a link $A \rightarrow B$, progressively setting the metric of the link with the metrics of a Reroute Metric Sequence for D will provide a loop free convergence for destination D.*

Let us consider a RMS for a link $A \rightarrow B$ and a destination D, $\{m_1, m_1+1, m_2, m_2+1, \ldots, m_i, m_i+1, \ldots, m_t\}$.

For each i, a transition from m_i to $m_i + 1$ is loopfree according to Theorem II.1.

For each i, a transition from $m_i + 1$ to m_{i+1} is loopfree. In a first case, if $m_{i+1} = m_i + 1$ there is no metric increment to perform. Otherwise, if the metric of $A \rightarrow B$ is $m_i + 1$, there is no router that will update its FIB for destination D if the metric of the link is set to a value within $[m_i + 1, m_{i+1}[$. The contrary would mean the there is a rerouting router whose Key

Metric is not present in the RMS. So, increasing the metric of the link from $m_i + 1$ to m_{i+1} is equivalent to changing the metric of the link from $m_i + 1$ to $m_{i+1} - 1$, which does not change anything in the paths used by the routers to reach D, and then incrementing the metric of the link from $m_{i+1} - 1$ to m_{i+1}. Doing this cannot cause forwarding loops according to Theorem II.1. ∎

We showed in the beginning of this section that, in the topology depicted in Figure 2, the Metric Sequence $\{10, 11, 40\}$ was sufficient to provide a loopfree convergence for destination A when setting the link metric of $B - C$ to 40, even if the RMS computed for this link would have been equal to $\{10, 11, 30, 31, 40\}$ for A.

In fact, most of the metrics of a RMS are actually not necessary to provide a loopfree convergence for a given destination D. But these are the key metrics that cause FIB Updates for destination D on the routers of the network. So, we will try to remove the unnecessary increments from the RMS. We will call the obtained sequences Reduced Reroute Metric Sequences (RRMS). When the size of a RRMS for a destination D is minimal, i.e. when there does not exist a shorter metric sequence ensuring a loop-free convergence, we call the sequence an Optimal Reroute Metric Sequence (ORMS).

B. Reduced and Optimal Reroute Metric Sequences.

Here, we will explain our technique to reduce an RMS to an RRMS, considering a destination D, a link $A \rightarrow B$, with its intial metric m_1 and a target metric $m_t > m_1$. Next, we will prove that our technique provides Optimal Reroute Metric Sequences.

To reduce a RMS for a destination D to an RRMS, we propose to start from the initial metric and perform the largest possible metric increment that does not lead to forwarding loops. We do that at each step until the target metric is reached. We call this technique the "Largest Increase First" technique (LIF).

For example, given a Reroute Metric Sequence $\{m_1, m'_1, m_2, m'_2, \ldots, m_i, m'_i, \ldots, m_t\}$, we find the largest metric M in that sequence, such that setting the metric of $A \rightarrow B$ to M will not lead to forwarding loops. To do that, we compute the rSPT of D considering the largest metric for the link in the sequence. Then, we merge the initial rSPT of D with its rSPT after the change, and we detect cycles within the obtained graph. When a cycle is detected, we try it again with smaller metrics until we find one metric M such that the merging of the rSPTs is cycle free. Then we reapply the technique, starting from M, and we do that repeatedly until we reach the target metric m_t.

When computing the largest metric increment, we chose to try the largest metric first and decrease it when cycles are detected to be able to reuse the rSPTs computed with large metrics during the remainder of the RMS reduction. Also, very few metrics are generally necessary to reach the target metric even if the initial RMS is long. Thus starting by the end of the sequence reduces the number of rSPTs to compute during the RMS reduction.

Now, let us show why the proposed reduction technique provides Optimal Reroute Metric Sequences. The reasoning is based on lemma III.2.

Lemma III.2 *If a metric transition for a link $A \rightarrow B$ from m to n, with $m < n$, is not loopfree, then*

1) *A metric transition from k to n for this link, with $k < m$, is not loopfree*
2) *A metric transition from m to o for this link, with $n < o$, is not loopfree*

Let us prove this lemma. If the transition from metric m to n is not loopfree for a destination D, then there is a cycle in the merging of $rSPT(D)$ and $rSPT'(D)$, being respectively the rSPT of X when the metric of $A \rightarrow B$ is set to m and n.

Let us denote the rSPT of D when the metric of the link is set to o with $rSPT''(D)$. The second proposition is true if there is a cycle in the merging of $rSPT(D)$ and $rSPT''(D)$. When setting the link metric from m to n, the shortest path of a set of nodes towards D were no longer via link $A \rightarrow B$, which led to the possibility of a loop. Let us denote this set of nodes by $\mathcal{N}$. If the link metric was set to o, instead of being set to m, each node in N would also use their shortest paths to D not via $A \rightarrow B$. Basically, these are the same as the ones they use when the metric is set to n. So, the path from each node in $\mathcal{N}$ to D in $rSPT'(D)$ is the same path as in $rSPT''(D)$. So, when merging $rSPT''(D)$ with $rSPT(D)$, we obtain at least the same cycles as when merging $rSPT(D)$ with $rSPT'(D)$.

The same reasoning can be applied to prove the first proposition. ∎

From this lemma, we can prove that our reduction technique provides Optimal Reroute Metric Sequences.

Let us consider a RRMS obtained with our technique, $\{m_1, \ldots, m, m'', m''', \ldots, m_t\}$.

Due to the definition of the LIF technique, we know that

1) A transition from m'' to the metric of the initial RMS following m''', say m^{loopy} is not loopfree.
2) From 1) and Lemma III.2, we know that a transition from a metric $x < m''$ to m^{loopy} is not loopfree.

If the LIF technique does not always provide an ORMS, this implies that another technique could provide a shorter valid sequence by not always selecting as next metric to a given m the largest possible metric increment that ensures a loopfree convergence. Starting from metric m, the better technique would thus select as next metric in its resulting sequence a metric $m' < m''$.

3) In order to spare a metric increment in comparison with LIF, it would have to select as the next metric after m', a metric $m^{better} > m'''$, so that $m^{better} \geq m^{loopy}$.

So, the better technique would have the subsequence $\{m, m', m^{better}\}$ in its Reroute Metric Sequence.

4) Knowing that $m' < m''$ and $m^{better} \geq m^{loopy\_2}$, we obtain from 2) and Lemma III.2 that this transition is not loopfree, so that this better technique does not exist. ∎

In Figure 2, the RMS for destination A, considering the metric change of link $B \rightarrow C$ from 10 to 40, is $\{10, 11, 30, 31, 40\}$. When applying the LIF technique the obtained ORMS for A is $\{10, 11, 40\}$. Indeed, a direct change from 10 to 30 would cause a loop between B and D, so that the metric 11 is mandatory, and a direct change from 11 to 40 is loopfree for destination A, so that the intermediate metrics are skipped by the technique.

C. Merged Reroute Metric Sequences.

In practice, routers react to the the update of a link metric by updating their FIB for all the destinations towards which their shortest paths have changed. So, knowing the ORMS for a destination D, according to a metric transition for a link, is not sufficient to provide a working solution.

In this part, we show that the merging of the ORMS obtained for each destination gives a valid, loopfree Reroute Metric Sequence for all the destinations affected by the change.

Let us consider an ORMS for link $A \rightarrow B$ and a destination D, $\{m_i, \ldots, m_j, m_k, m_l, \ldots, m_t\}$, We need to prove that inserting values in that sequence also gives a loopfree Metric Sequence for destination D.

Let us consider the sequence $\{m_i, \ldots, m_j, m_s, m_k, m_l, \ldots, m_t\}$, with $m_s \in]m_j, m_k[$. Let us denote the rSPT of D when the metric of link $A \rightarrow B$ is set to m_x by $rSPT_x(D)$.

As m_j and m_k are consecutive metrics in the initial ORMS, we know that the merging of $rSPT_j(D)$ and $rSPT_k(D)$ does not contain a cycle. The set of source-destination paths that differs between those rSPTs forms a superset of the paths that differ between $rSPT_j(D)$ and $rSPT_s(D)$. Indeed, every path not via $A \rightarrow B$ that becomes used to reach D when the metric of the link is set to m_s also becomes used when the metric of the link is set to a larger value. Also, every path via $A \rightarrow B$ that is still used to reach D when the metric of the link is set to m_k is also still used when the metric is set to $m_s < m_k$. This implies that the merging of $rSPT_j(D)$ and $rSPT_s(D)$ is the merging of $rSPT_j(D)$ and a subgraph of $rSPT_k(D)$, so that this merging does not contain a cycle. The same reasoning can be used to show that the merging of $rSPT_s(D)$ and $rSPT_k(D)$ is cycle free, so that the metric sequence $\{m_j, m_s, m_k\}$ is loopfree for destination D.

As the same reasoning can be applied when inserting a metric between m_s and m_k in the new sequence, we have proved that the insertion of an arbitrary number of metrics within an ORMS still gives a loopfree metric sequence for its destination. ∎

D. Optimization of Merged Reroute Metric Sequences.

The merging of two Optimal Reroute Metric Sequences S_a and S_b associated with two destinations a and b might be such that there exists a shorter sequence providing a loopfree convergence for both destination a and b.

Firstly, an Intermediate Metric in a Reroute Metric Sequence for S_a becomes unnecessary in the merged sequence if a Key Metric of S_b can play the role of the Intermediate Metric in S_a.

Let us for example assume that $S_a = \{3, 4, 8\}$, and $S_b = \{5, 8\}$, with 3, 8, and 5 being Key Metrics. The metric 4 in S_a is an Intermediate Metric introduced when the Reroute Metric Sequence is computed for a. This means that the only reason to transiently set the metric of the link to 4 is to force a router R to stop using its equal cost paths to a that contain $A \rightarrow B$, as 4 is not a Key Metric and the next Key Metric is 8. An intermediate value of 5 would have the same effect and would also be loopfree.

This implies that, $S'_a = \{3, 5, 8\}$ is also a valid Reroute Metric Sequence for destination a. So, we can replace the initial Merged Reroute Metric Sequence $\{3, 4, 5, 8\}$ by $\{3, 5, 8\}$, still ensuring that no transient forwarding loop will occur during the convergence.

Secondly, a Key Metric in a Reroute Metric Sequence for S_a becomes unnecessary in the merged sequence if another Metric present in S_b can play the role of this Key Metric. Let us for example assume that $S_a = \{3, 4, 8\}$, and $S_b = \{5, 8\}$, with 3,4,5,8 being Key Metrics. It is possibe that the Key Metric 5 for S_b, obtained with the LIF technique, would be also valid if 5 is replaced by 4, so that $\{3, 4, 8\}$ would still ensure that no transient forwarding loops occur during the convergence.

Due to space limitations, we cannot present a detailed description of the technique performing such optimizations. Roughly, it is achieved by applying a technique similar to LIF on the obtained Merged Reroute Metric Sequences.

In Figure 3, we present the pseudo-code for the computation of a Merged Reroute Metric Sequence considering a metric increase to m_t for a link $A \rightarrow B$.

The algorithm firstly explores the SPT of A to obtain the set of destinations that are reached via $A \rightarrow B$. Then, it computes the Optimal Reroute Metric Sequence for each destination D reached via this link. To do that, it computes the set of Key Metrics for D, by analysing the reverse Shortest Path Trees of D with the initial and target metric set to $A \rightarrow B$, and it inserts the Intermediate Metrics to give the Reroute Metric Sequence.

Then, it optimizes the Sequences by applying the LIF technique. In the implementation, we stop the merging of the rSPTs performed by the LIF technique as soon as a length-2 cycle is detected, so that the cycle detection performed on the merged rSPTs is not necessary in those cases.

Finally, we merge the obtained optimal Reroute Metric Sequences, and we prune Intermediate and Key Metrics that become unnecessary due to the merging. Note that the computed rSPTs are put in a cache along the computation of an optimized reroute metric sequences, so that the number of rSPT computations is in the worst case equal to the length of the initial Reroute Metric Sequence for each destination.

The algorithm has been implemented in Java as a Proof of Concept and will be integrated in the next version of the Totem toolbox [9].

```
Metric increase to m_t for Link A → B:
//Computation of the affected Destinations
AffectedDest = follow(A → B, SPT_init(A));
//Computation of the ORMS
ORMSSet = {};
foreach Destination D ∈ AffectedDest do
    RMS = GetRMS(D,A → B, m_t);
    ORMS = OptimizeRMS(D, RMS, A → B, l.metric, m_t);
    ORMSSet.add(ORMS);
end
MergedRMS = MergeSequences(ORMSSet);
MergedRMS = PruneUnecessaryMetrics(MergedRMS);
return MergedRMS;

MetricSequence GetRMS(Destination dest, Link L, Metric
target_metric):
RMS = {L.metric, target_metric};
//Compute the rSPT of D with the initial metric of L
initialRSPT = computeRSPT(dest,L,L.metric);
//Compute the rSPT of D with the target metric of L
targetRSPT = computeRSPT(dest,L,target_metric);
foreach Node S | PathLength(S, D, initialRSPT) ≠
PathLength(S, D, targetRSPT) do
    KeyMetric = L.metric + PathLength(S,D,targetRSPT) -
    PathLength(S,D,initialRSPT);
    RMS.add(KeyMetric);
    //Introduce Intermediate Metric
    if KeyMetric ≠ target_metric then
        RMS.add(KeyMetric+1);
    end
end
return RMS;

MetricSequence OptimizeRMS(Destination D,MetricSequence
RMS,Link L,Metric StartMetric,Metric TargetMetric):
tempORMS = {StartMetric};
currentMetric = StartMetric;
while (!(currentMetric==TargetMetric)) do
    //Find the largest Metric M in RMS such that transition from
    //currentMetric to M is loopfree for destination D
    M = TargetMetric;
    bool loopfree=false;
    while (! loopfree) do
        MergedrSPT =
        merge(rSPT(D,L,currentMetric),rSPT(D,L,M)):
        if MergedrSPT.containsCycle() then
            M = Metric Before M in RMS;
        end
        else
            loopfree = true;
        end
    end
    tempORMS.add(M);
    CurrentMetric = M;
end
return tempORMS;

ShortestPathTree rSPT(Destination Dest, Link L, metric m):
if (rSPTCache.contains(Dest,L,m)) then
    return getrSPTCache(Dest,L,m)
end
else
    rSPT = Compute rSPT of D with the metric of L set to m;
    putInCache(Dest,L,m,rSPT);
end
```

Fig. 3: Algorithm to compute Merged Reroute Metric Sequences

IV. Loop free convergence using metric decrements

What has been presented in the previous section holds for the cases where a link is shut down or its metric is increased. We based the correctness of the provided metric update sequences on the fact that, at each step, for each destination affected by the change, the merging of its rSPT before and after the event is cycle free.

That is, when we consider the transition between a metric m_i towards a metric m_t smaller than m_i, we know that reversing a valid Reroute Metric Sequence for the transition of the link metric from m_t to m_i will provide transitions such that the merging of the rSPTs of the affected destinations are cycle free, at each step of a transition from m_i to m_t.

So, it is not necessary to provide an algorithm that specifically solves the metric decrease problem as soon as an algorithm is provided for the metric increase problem.

Note that when a link is being brought up in the network, we first set the metric of the link to a value such that the link will not be used. Then, we apply the same technique as for a metric decrease event.

V. ISP Topologies Analysis

To evaluate the performance of our rerouting scheme, we use three real ISP topologies. The first one is GEANT, the pan-European Research Network [8]. We use the GEANT topology as it was in 2005. GEANT connected all the National Research networks in Europe and had interconnections with research networks in other continents. GEANT was composed of 22 routers, 21 in Europe and one in New-York, USA. The network topology was highly meshed in the core (Germany, Switzerland, France, UK, Netherlands) and there was fewer redundancy in the other parts of the network. Each POP was composed of a single router.

The second studied network contains all the routers of a Tier-1 ISP with presence in Europe, America and Asia. This network is composed of about 110 routers and 170 links.

The third studied network contains the backbone nodes of a large Tier-1 ISP. The backbone of this network has about 200 routers and 400 links in Europe, America and Asia. For both Tier-1 ISPs, each POP is usually composed of two core routers as well as several aggregation and access routers.

We applied the technique on all the directed links of those ISPs. We did not try to write optimized Java code in our proof of concept. However, the time required to compute the reroute metric sequences for Geant was negligible. For the two Tier-1 ISPs, a few seconds was required in the worst case to compute a reroute metric sequence. As we will see in the results, around 50% of the links shutdown could lead to a forwarding loop in the studied topologies. So, directly setting the metric of a link to MAX_METRIC as described in [3] is not sufficient to gracefully shut down links.

We considered the worst-case scenario where the considered link must be shutdown, so that the target metric of the link is MAX_METRIC.

In Figure 4, we can see that among the 72 directed links of Geant, the length of the MRMS is 1 for 39 links. In fact, these are the links that can be shut down without causing forwarding loops, so that the reroute sequence only contains MAX_METRIC. Forwarding loops can occur during the shutdown of 33 links. For 30 of them, less then 3 metrics including MAX_METRIC are required. 4 metric changes are necessary for 2 links, and 6 metric changes are necessary for one link. This last link is connecting the Eastern Europe routers to one router in Germany. Eastern Europe routers form a ring, which favours the occurence of forwarding loops, so that many destinations reached via this link have a non-empty Optimized Reroute Metric Sequence.

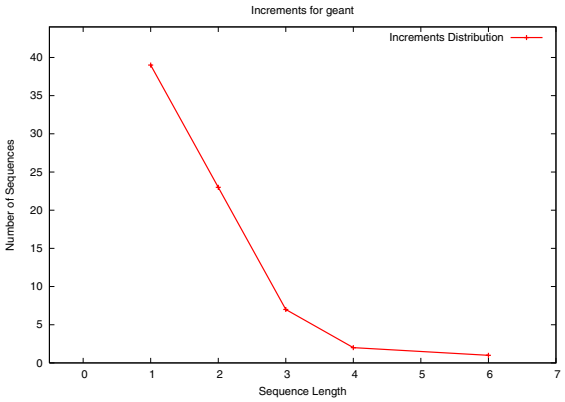

Fig. 4: Reroute Metric Sequence length distribution for Geant

For the second topology (Figure 5), we see that all the obtained reroute metric sequences have a length shorter than 12. 94.1% of them are shorter then 5 and 98.8% shorter than 10. We can see that a small percentage of the reroute metric sequences have a length of 0. These are the sequences for links that are unused in the topology, so that it is not necessary to change their metric before shutting them down.

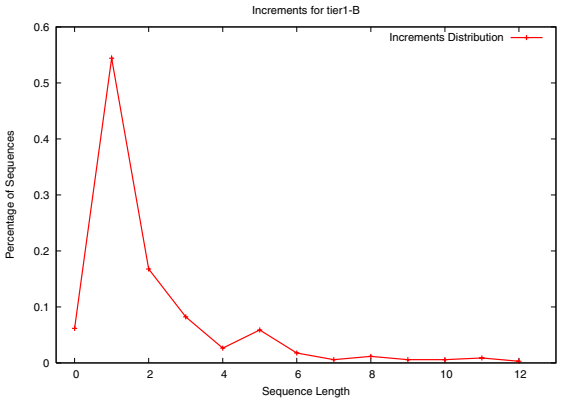

Fig. 5: Reroute Metric Sequence length distribution for the first Tier-1 ISP

For the third topology (Figure 6), 50% of the links cannot be shutdown directly without causing forwarding loops. Though,

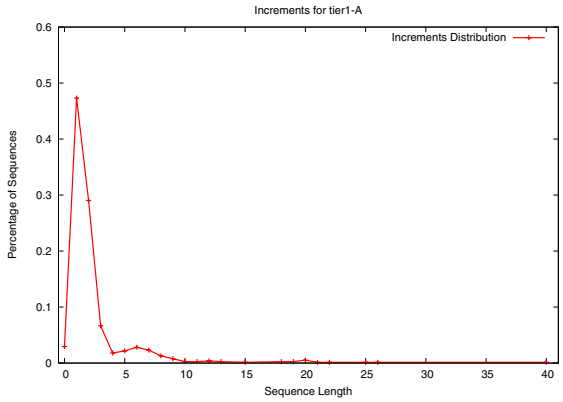

Fig. 6: Reroute Metric Sequence length distribution for the second Tier-1 ISP

97.3% of the links can be shutdown without forwarding loops by using Reroute Metric Sequences whose length is shorter than 10 and 99.3% with metrics sequences shorter then 20. 5 links require longer metric sequences, with a worst case length of 40 for one link.

Assuming a worst-case convergence time of 5 seconds after a link metric update, applying the solution would let an operator wait for less than a minute to shut down a link without loosing packets in most of the cases. As the solution is applied in the case of planned, non-urgent topological change, the delaying of the actual link shut down seems to be short compared to the obtained gain. When a sudden topological changes occurs while the solution is applied on a link somewhere else in the network, the network monitoring tool should stop the modification of the link metric and restart the computation of a valid Metric Reroute Sequence according to the new topology.

Shutting down a link is a worst-case event for the solution. We also performed analysis where the metric of each link is doubled to consider a case where a metric is updated for traffic engineering purposes. For Geant, the maximum length of a sequence was 3. In the second topology, one sequence had a length of 12, and 92% of the sequences where shorter than 3, with the target metric included. In the third topology, the maximum length of a sequence was 22, and the length of most of the remaining sequences was shorter than 5.

VI. Related Work

The problem of avoiding transient loops during IGP convergence that follows topology changes has mainly been studied by considering extensions to routing protocols. Extensions have been defined for link-state [10] and distance vector protocols [11], [12]. More recently, two different solutions have been proposed to avoid transient loops during the convergence of OSPF or ISIS. The first solution, proposed by Bryant et al. in [13] avoids transient loops after a link shutdown by delaying the computation of the SPF on the routers in function of their distance from the failure. A drawback of this timer-based approach is that some routers may have to wait a

long time before updating their FIB, but it does not require new OSPF/IS-IS messages. In [14], we proposed a distributed solution based on messages encoded inside the IS-IS Hello messages exchanged between routers. Our solution allows to converge faster than the timer-based solution, but requires small changes to the protocol [15]. These two solutions have been merged recently [16], but a few years will pass before the IETF standardizes extensions to OSPF and IS-IS and operators are actually able to deploy them. The main advantage of the solution proposed in this paper is that it can be implemented today in a network management system and does not require any changes to routers and protocols. In [17], M. Shand et al. discuss the idea of repeatedly incrementing a link metric by one to reach a forwarding state where the link is no longer used. This solution was rejected by the IETF due to the number of increments that would be required to shut a link down. The solution presented in this paper only performs the metric updates that are necessary to avoid forwarding loops, so that the idea is now applicable.

In [18], the authors propose to avoid transient loops by adding state in the interfaces of the routers, so that these can infer that a packet is caught in a transient forwarding loops based on its source, its destination and the interface on which it is received. This solution is attractive but requires complex modifications to routers software, and does not deal with asymmetrical link metrics.

The problem of gracefully changing the network topology without disrupting traffic has been addressed in MPLS networks using traffic engineered tunnels. In these networks, RSVP-TE [19] is used to create and modify the MPLS tunnels between an ingress and an egress router. When a traffic engineered tunnel must be modified, for example to follow a different path, RSVP-TE allows to change the tunnel without loosing any packet.

VII. Conclusion

In this paper, we proposed a solution that can be applied now by ISPs to avoid transient forwarding loops during a maintenance operation performed on a link. The solution allows an operator to reconfigure the metric of a link, shut down a link, or set up a link in the network without loosing a single packet. Compared to the solutions proposed before, the main advantage of the solution is that it does not require any modification to the intra-domain routing protocol, as the solution relies on sequences of metric reconfigurations such that each step of the sequence does not disrupt the consistency in the forwarding of packets accross the network. We do not intend to implement such a solution in the routers themselves, but rather in a network management tool that would issue SNMP requests to the node being the head-end of the link to be reconfigured. The provided applicability analysis performed on real ISP topologies shows that the solution never requires a large number of link metric reconfigurations to shut a link down or bring it back up. This is fortunate as the consequence is that applying the solution will not lead to a tremendous delaying of the actual shut down of the link being maintained. Hence, it will not be an important constraint for an operator to use the solution, even if the gain of using it is important. As stringent SLAs are a reality that ISPs currently face, we think that the solution is attractive as it will help them to avoid forwarding loops by themselves while the long lasting standardization process of a protocol built-in solution terminates and implementations reach the market.

Acknowledgements

We would like to thank Clarence Filsfils and Stewart Bryant for their useful suggestions and comments.

References

[1] 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.

[2] 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.

[3] R. Teixeira and J. Rexford, "Managing routing disruptions in internet service provider networks," *IEEE Communications Magazine*, March 2006.

[4] P. Pongpaibool, R. Doverspike, M. Roughan, and J. Gottlieb, "Handling ip traffic surges via optical layer reconfiguration," *Optical Fiber Communication*, 2002.

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

[6] N. Shen and H. Smit, "Calculating interior gateway protocol (IGP) routes over traffic engineering tunnels," October 2004, rFC3906.

[7] 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.

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

[9] 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

[10] 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.

[11] J. Garcia-Luna-Aceves, "Loop-free routing using diffusing computations," *IEEE/ACM Transactions on Networking*, vol. 1, no. 1, 1993.

[12] A. Retana, R. White, and D. Slice, *EIGRP for IP: Basic Operation and Configuration*. Addison Wesley, 2000.

[13] S. Bryant, C. Filsfils, S. Previdi, and M. Shand, "IP Fast Reroute using tunnels," May 2004, internet draft, draft-bryant-ipfrr-tunnels-00.txt, work in progress.

[14] P. Francois and O. Bonaventure, "Avoiding transient loops during IGP convergence in IP networks," in *IEEE INFOCOM'2005*, Miami, Florida, USA, March 2005.

[15] 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.

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

[17] S. Bryant and M. Shand, "A framework for loop-free convergence," December 2006, internet draft, draft-ietf-rtgwg-lf-conv-frmwk-00.txt, work in progress.

[18] 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.

[19] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, " RSVP-TE: Extensions to RSVP for LSP Tunnels," December 2001, RFC 3209.