Network Working Group
Internet Draft
Expiration Date: June 2006
File name: draft-ietf-rtgwg-microloop-analysis-01.txt

Alex Zinin
Alcatel
October 2005

Analysis and Minimization of Microloops in Link-state Routing Protocols

draft-ietf-rtgwg-microloop-analysis-01.txt

Status of this Memo

Abstract

Link-state routing protocols (e.g. OSPF or IS-IS) are known to converge to a loop-free state within a finite period of time after a change in the topology. It is normal, however, to observe short-term loops during the period of topology update propagation, route recalculation, and forwarding table update, due to the asynchronous nature of link-state protocol operation. This document provides an analysis of formation of such microloops and suggests simple mechanisms to minimize them.

1 Introduction

Link-state routing protocols, such as [OSPF] and [ISIS] converge to a loop-free state within a finite period of time after a topology change. Additional changes postpone the convergence, but do not get in its way.

During the period of convergence, however, link-state protocols exhibit short-term routing table inconsistencies caused by the protocol's asynchronous nature. These incornsistencies may cause short-term packet loops, also known as microloops. For example, see a sample network in Figure 1.

```
+--+    1    +--+
|A |---------|B |
+--+         +--+
 |  \  10      |
5|   ------    |1
 |         \   |
+--+   10   \+--+
|E |---------|C |
+--+         +--+
    \_      /
   5  \    /1 (failure)
       +--+
       |D |
       +--+
```

Figure 1. Microloop example

We are interested in routers A and B and their best paths towards D. Before failure, B's best path to D is B-C-D with cost 2, and A's best path is A-B-C-D with cost 3. When link C-D fails, both C and D announce their link state information with link C-D missing. Within a finite period of time, both A and B shall receive the topology updates and converge on them, installing new best paths: A-E-D (10) for A, and B-A-E-D (11) for B. However, if, due to the timing differences, B calculates and installs its new best path through A before A has a chance to switch from B to E, a microloop will form between A and B for the duration of time required for A to complete its routing table update.

Similar microloops may form when other topological changes happen in the network, for example, when a new link or a node is added, a link cost is changed, etc. In summary, whenever a topological change in the network results in changes of the shortest path three (SPT) for more than one node, it is possible for the network to exhibit temporary loops.

This document provides an analysis of microloop formation. Specifically, we categorize different types of reconvergence scenarios, and explore their properties. We then show that in certain scenarios microloops do not form, in others they can be eliminated using simple techniques described in this document, and define scenarios where more sophisticated loop avoidance mechanisms may be necessary.

It is useful to understand the relationship between [IPFRR] and the technique described here. The two mechanisms play complimentary roles to each other: deploying [IPFRR] without micro-loop prevention only partially addresses the goal of minimizing packet loss during network reconvergence, since packets will be lost due to microloops. On the other hand, micro-loop prevention described in this document relies on [IPFRR] local failure protection, as routers will keep forwarding traffic down the old path until the new next-hops are known to be safe.

2 Analysis

To analyse the behavior of a network during reconvergence, we look at a given router and its neighbors before failure and during the transition to the new routes. More specifically, we analyse whether switching to the new routing information can result in loop formation or not.

2.1 Terminology

The following terms are used in the draft.

Dopt(X,Y)
Integer function defined as the cumulative cost of the least-cost path from node X to node Y in a topology graph. Normally calculated by link-state routing protocols using Dijkstra algorithm as part of regular route calculation procedures. This is the same as "Distance_opt(A,B)" defined in [IPFRR-FW]

Downstream neighbor
Neighbor N of router S is considered S's downstream neighbor for destination D, if Dopt(N, D) < Dopt(S, D)

Primary neighbor
Neighbor N of router S is considered S's primary neighbor for destination D, if the path via N is such that D(S, D) is minimized, i.e. N provides a shortest path to D according to the SPF calculation.

Loop-free neighbor

Neighbor N of router S is considered S's loop-free neighbor for destination D, if Dopt(N, D) < Dopt(N, S) + Dopt(S, D). Note that a loop-free neighbor may be, for example, router's primary before and/or after failure.

2.2 Next hop safety condition

We start the analysis of single-hop loops with the following observation:

After a topology change, there are precisely two situations when a microloop between routers X and Y can form:

a) Before failure, X uses Y as its next-hop. After failure, Y uses X as its next-hop. Y updates its routes based on the new topology before X.

b) Exact opposite of the previous case. Before failure, Y uses X as its next-hop. After failure, X uses Y as its next-hop. X updates its routes based on the new topology before Y.

Formulating this for a given calculating router S (either X or Y in the above example) switching to a new primary Pn, a microloop may occur between S and Pn only if Pn was forwarding through S before failure.

Based on the above, we can define a general safety condition for any neighbor N (whether new primary or not) of router S that has just learned about a topology change. Note that the condition must satisfy the topological criteria above, and be non-recursive, i.e. not lead to loops if both S and N follow it.

Next-hop safety condition:

After a topology change, it is safe for router S to switch to neigbor N as its next-hop for a specific destination if the path through N satisfies both of the following criteria:

1. S considered N as its loop-free neighbor based on the topology before change AND

2. S considers N as its downstream neighbor based on the topology after change.

The first requirement ensures that N has not been forwarding traffic to S before the change occured and both S and N used old topology. The second requirement makes sure N does not

forward traffic to S when N learns the new topology. Note again, that N is S's any neighbor, and may or may not be used by S as its new primary or a temporary safe neighbor.

The difference in the conditions before and after failure is there to make sure that S and N do not recursively consider each other as safe next-hops when they learn about the failure.

2.3 Transition types

Here, we analyse different types of scenarios that a given router may find itself in after learning about a topology change.

For each destination affected by a topological change, the network will have three major types of nodes categorized by the degree of safety of their old primary, new primary, and other neighbors. (Note that we do not yet consider ECMP, which will be discussed in section 3.2.)

Type A

Routers whose new primary next-hops after the topology change are safe and transition to them will not create a microloop. Two subtypes are recognized:

A1: Routers whose primaries haven't changed as a result of the topology change

A2: Routers whose new primary satisfies the safety condition

Type B

Routers whose new primary next-hops after the topology change do not satisfy the safety condition, but that have at least one other neighbor that does. Note that such a neighbor can be the router's old primary (type B1) or a neighbor that is neither old nor new primary (type B2).

Type C

Routers that have no neighbor that satisfies the safety condition.

It is clear that nothing special needs to be done for type-A routers as they either do not need to modify their routes or can immediately switch to the new primary next hops.

It can also be shown that if type-B routers do not immediately switch to their new primaries, but use their safe next-hops for some time, switching to the new primaries later will not create loops, provided that their downstream routers have also switched to the safe hops or have already switched to the new primaries.

> NOTE: The above analysis applies to single-hop loops. Multi-hop loops, possible in networks with asymmetric link costs could be prevented by using a tighter safety condition. However, as shown by simulations on real-life network topologies, doing so would decrease micro-loop coverage and thus result in increased number of unprevented single-hop loops.

The following section formally defines the mechanism.

3. Loop prevention mechanism

3.1 Basic procedures

The essense of the mechanism defined here, also known as "path locking via safe neighbors" (PLSN), can be informally summarized as follows. Upon a topology change, for each destination:

- Each router in the network assesses safety of its new primary next-hops.

- If the new primaries are safe, they are used immediately, otherwise, partial
  ordering of updates is introduced:

  o   If non-primary safe neighbors are found, they are used for a period of time, thus locking traffic to a safe path while the new primaries complete their transition to the new routes

  o   If no safe neighbors are found, the forwarding path is locked on the old next-hop for some time to give the new primary enough time to complete route updates.

For a description of several architectural constants used in this document (named as "DELAY_xxx"), refer to section 3.4.

On receiving a topology update, the router delays its SPF calculation by DELAY_SPF time in order to collect the remaining updates that

relate to the same topological event (e.g. update from the router connected to the second end of a point-to-point link in case of a link failure, or updates from other neighbors of a failed node).

Upon expiration of DELAY_SPF, the router calculates the new SPT, the new routes, checks the safety status of each neighbor relative to each affected destination using the conditions in section 3.1, and applies the following logic for each route depending on the type of role it finds itself in:

Type A:
The route SHALL be updated with the new primary next-hops without an additional delay.

Type B:
The route SHALL be updated with one or more temporary next-hops that satisfy the safety condition without an additional delay. These temporary next-hops SHALL be used for the duration of DELAY_TYPEB. After DELAY_TYPEB, the route SHALL be updated with the new primary next-hops.

Type C:
The route's old (primary) next-hops SHALL continue to be used for DELAY_TYPEC. After DELAY_TYPEC, the route SHALL be updated with the new primary next-hops.

If, after expiration of DELAY_SPF, the router receives a topology update sooner than DELAY_STABLE after the previous one, the router MUST fall back to the regular convergence mechanisms by prematurely expiring DELAY_TYPEB or DELAY_TYPEC timers if they are still running (thus causing immediate installation of the new primary next-hops), MUST recalculate its routing table as soon as practical, and MUST refrain from using the mechanisms described here until it has seen no topological updates for at least DELAY_STABLE. This is a safeguard mechanism to ensure that procedures described here are applied only when a single failure is experienced and that the network converges in a situation where multiple topological events or network instabilities are experienced.

[ISIS] includes the concept of an Overload bit (OL) that indicates a node in the network that shouldn't be used as transit. A similar notion is introduced in OSPF by [STUB] using LSInfinity link costs. Honoring these conventions, implementations of this document MUST NOT use neighbors with the OL bit set in IS-IS or announcing links to the calculating router with LSInfinity cost in OSPF as temporary safe neighbors.

NOTE: In OSPF, if S's neighbor N is a stub router, the S->N link, visited first by the SPF algorithm, will normally have a real link cost, and it is the backwards link N->S announced by N that will have its cost set to LSInfinity. Implementations have to account for this details when satisfying the above requirement.

3.2 Equal Cost Multipath Considerations

In situations where more than one primary next-hop is available after the topology change, there are several possible combination of their safety properties:

1) All new next-hops satisfy the safery condition (a pure type-A situation)

2) Some of the new next-hops satisfy the safety condition, some of them do not (a combination of type-A and type-B)

3) None of the new next-hops satisfy the safety condition, however, there's at least one other neighbor that satisfies it (a safe non-primary next-hop, causing new primaries to be type-B)

4) None of the new next-hops satisfy the safety condition, and there is no other neighbor that satisfies it (a pure type-C situation).

For situations 1, 3, and 4 above, the implementation merely follows the basic procedures described in section 3.1

For situation 2 (an A/B combination), the implementation:

1) SHALL update the route with the new next-hops that satisfy the safety condition without an additional delay

2) SHALL add the remaining new next-hops after DELAY_TYPEB.

Note that one could potentially use temporary safe neighbors in situation 2 above, however this specification does not recommend this to avoid unnessesary traffic rerouting and hence packet reordering.

3.3 Local Failure and IP Fast Reroute Considerations

After detecting a local failure and initiating the local repair process if IPFRR is supported, the router directly attached to the point of failure follows the procedures described in this document--it delays its SPF calculation to collect updates from other routers, calculates new routes, and classifies the next-hops.

For routers implementing IPFRR, the difference with routers that learn about the failure from the routing protocol updates, is that one or more of the repairing router's old next-hops has become unavailable, and hence cannot be considered as the temporary safe next-hops for type-B operation. Also, if the router was able to locally repair the failure, and the new primary next-hops do not satisfy the safety condition, the router should consider itself in the middle of type-B operation with the temporary safe neighbor engaged as part of IP Fast Reroute operation.

Another distinct situation is when the router does not support IPFRR or could not repair the failure, the new primary next-hops do not satisfy the safety condition, and there's no other neighbor that does, i.e. a type-C situation. Unlike other routers in the network, the router directly connected to the network does not have the old next-hop any more, and cannot continue using it. Immediately switching to the new next-hops, on the other hand, may result in a micro-loop. In this situation, the router MUST discard traffic forwarded along the affected route for the duration of DELAY_TYPEC, and then update the routes. Implementations MAY have a configuration option to allow switching immediately to the new next-hops for situations where this type of a micro-loop is not a concern. If implemented, this option MUST be disabled by default.

As a result, there are the following possible scenarios:

1) If the new primary next-hops satisfy the safery condition, the router updates the routes without an additional delay.

2) Otherwise, if the failure could be repaired locally by IP Fast Reroute, the router continues to use the repair path for DELAY_TYPEB and updates the routes with the new primary next-hops after it expires.

3) Otherwise (new next-hops are not safe, and IPFRR is not supported or the failure couldn't be repaired), the router discards traffic for DELAY_TYPEC and updates the routes with the new primary next-hops after its expiration.

3.4 Architectural Constants

The following architectural constants have been used in the description of the algorithm above:

DELAY_SPF
The delay between the moment the router receives the first topology update after a period of stability and the moment it starts its routing table recalculation. This delay is

necessary to collect multiple updates originated by different routers that relate to the same topological event.

DELAY_STABLE

Period of time, during which the network topology is considered to be stable if the router receives no topological updates. When the first update after DELAY_STABLE is received, all other updates that fit within DELAY_SPF are considered as related to a single topological event.

DELAY_TYPEB and DELAY_TYPEC

Periods of time used by the router to delay installation of new primary next-hops after a topology change when the router has (type-B) or has not (type-C) a safe neighbor to temporary divert the traffic to in the meantime.

While correctness and effectiveness of the algorithm described here does not depend on the actual values assigned to the architectural constants, it does depend on the relationship between them, and the assumption that all routers in the same network use the same values.

To satisfy these constrains, and yet allow these delays to be decreased as implementations continue to improve towards faster convergence, this document defines the architectural constants as configurable, specifies the required relationship between the values, and the default values that should be used by the implementations.

The following equations define the relationship between the constants that needs to be maintained in order for the mechanism described here to provide desireable results:

 DELAY_SPF > update-propagation-time

 DELAY_STABLE > DELAY_TYPEB > DELAY_TYPEC > fault-propagation-time

where:

o   update-propagation-time is the time it is expected to take routers in the network to detect the failure, and originate and propagate new link-state information.

o   fault-propagation-time is update-propagation plus the time it is expected to take routers in the network to calculate the new SPT, check the safety condition of the neighbors, and install required FIB entries.

Because fault-propagation-time includes update-propagation-time, and DELAY_SPF (since every router will delay its SPF according to this

document):

fault-propagation-time > DELAY_SPF + update-propagation-time

and hence the equations above can be converted to one:

DELAY_STABLE > DELAY_TYPEB > DELAY_TYPEC > (DELAY_SPF + update-propagation-time)

The implementations SHOULD use the following default values for the architectural constants:

Constant	Default val
DELAY_SPF	500 msec
DELAY_TYPEC	2 sec
DELAY_TYPEB	4 sec
DELAY_STABLE	10 sec

4 Coverage analysis

The above algorithm minimizes the probability of loop formation. More specifically, loops will only be possible when two neighboring routers both experience the type C condition after the topology change. Appendix A shows that transitions between A-A, A-B, A-C, and B-C routers are loop-free.

While this mechanism does not remove all possible micro-loops, it addresses the majority of them in topologies with a reasonable level of physical redundancy. Topologically, micro-loop coverage provided by this algorithm is very similar to that provided by [IPFRR]. This is due to the fact that similar construct are used by both mechanisms.

5 Backwards Compatibility Analysis

Effectiveness of the mechanism described here relies on the assumption that all routers in the network support it.

In a situation where some routers do not support the describer mechanism, the network will continue to converge properly fundamentals of the routing system are not changed. When a topology change event occurs in such a network, Type-A and Type-B routers will not substantially change the convergence patterns, as they will switch to routers that are guaranteed to forward traffic correctly after DELAY_SPF. (Note that routers today already implement a delay

similar to DELAY_SPF.) Type-C routers, when mixed with routers not supporting this mechanism, may induce longer than usual micro-loops (up to DELAY_TYPEC), however this delay is in the same order of magnitude as in most deployed networks today.

6 Security Considerations

The mechanism described in this document does not modify any routing protocol messages, and hence no new threats related to packet modifications or replay attacks are introduced. The mechanism changes certain delays used in node-local algorithms and introduces partial event ordering after a topology change has occured. This, however, does not introduce new security risks. For type-B situations, traffic to certain destinations can be temporarily routed via next-hop routers that would not be used with the same topology change if this mechanism wasn't employed. However, these next-hop routers can be used anyway when a different topological change occurs, and hence this can't be viewed as a new security threat.

7 Acknowledgements

The author would like to thank Don Fedyk, Chris Martin, Alex Audu, Olivier Bonaventure, Stefano Previdi, and other members of the IETF RTGWG for their useful comments. Special thanks go to Alia Atlas, Mike Shand, and Steward Bryant, who were instrumental in development of this mechanism, such as fine-tuning the safety condition, simulating the mechanism, proof-reading the document, and without whom this work wouldn't be possible.

8 References

8.1 Normative References

[OSPF] J. Moy. OSPF version 2. Technical Report RFC 2328, Internet Engineering Task Force, 1998.

[ISIS] ISO, "Intermediate system to Intermediate system routeing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)," ISO/IEC 10589:1992.

[IPFRR] Atlas, A., "Basic Specification for IP Fast-Reroute: Loop-free Alternates", Internet Engineering Task Force, Work in Progress, draft-ietf-rtgwg-ipfrr-spec-base-03.txt

8.2 Informative References

[IPFRR-FW] Shand, M., S. Bryant, "IP Fast Reroute Framework",

Internet Engineering Task Force, Work in Progress, draft-ietf-rtgwg-ipfrr-framework-04.txt

[STUB] Retana, A., et al, OSPF Stub Router Advertisement, RFC 3137, Internet Engineering Task Force, 2001.

Author's Address

Alex Zinin
Alcatel
701 E Middlefield Rd
Mountain View, CA 94043
E-mail: zinin@psg.com

Appendix A. Loop formation analysis

S is the calculating router discovering the failure through a link-state update. P is the old primary, NP is the new primary.

```
    BF:
                        <------
            [P]----------------[S]----------------[NP]
               ...>?

    AF:

                                  ------>
            [P]----------------[S]----------------[NP]
                                              ?<...
```

To analyze possible loop formation, we need to check the following:

1) if it is possible for P to start forwarding packets to S before S switches to NP

2) if it is possible for NP to be forwarding packets back to S before or after S starts using it

Assumptions are that type-As switch-over to NP immediately, and type-Bs and type-Cs wait certain amount of time so that:

```
DELAY_TYPEB > DELAY_TYPEC > fault-propagation-time
```

1. S is type A:

BF analysis:

1.1 If P is another type-A, then S cannot be its new primary, since S has not been P's LFA before (since it's been fwd'ing through P). Hence, P will not route through S AF, and the will be no loops between P and S.

1.2 If P is a type-B, then S hasn't been P's LF neighbor BF, and P will not forward through S at least for DELAY_TYPEB, which gives S enough time to switch to NP. After DELAY_TYPEB P may start using S as it's new primary.

1.3 If P is a type-C, then it hasn't been forwarding traffic to S BF, and will not use S as its new primary at least for DELAY_TYPEC, which should give S enough time to switch to NP.

1.4 Consequently, no loops will form between a type-A node and it's old primary before the type-A nodes switches to its new primary.

AF analysis:

1.5 Regardless of its type, NP has not been forwarding packets to S BF and will not do so AF by definition of type-A.

1.6 Consequently, no loops will form between a type-A node and it's new primary before or after the type-A nodes switches to it.

2. S is type B:

BF analysis:

2.1 If P is a type-A, then similarly to 1.1 above, there will be no routes between P and S.

2.2 If P is another type-B, then similarly to 1.2, S will not be used by P for at least DELAY_TYPEB, and S will have enough time to switch to its safe hops or NP.

2.3 If P is a type-C, then similarly to 1.3, S hasn't been receiving traffic from P BF, and will not AF for at least DELAY_TYPEC, which should give S enough time to switch to its safe hops or NP.

2.4 Consequently, no loops will form between a type-B node and it's old primary before the type-B nodes switches to its new primary.

AF analysis:

2.5 If NP is a type-A, then because of the DELAY_TYPEB NP must have

had enough time to switch to its new NP, which cannot be S by definition of SPT considering that NP is S's new nexthop in the SPT AF.

2.6 If NP is another type-B, then because of DELAY_TYPEB, NP must have had enough time to switch from its old primary and can equally likely be routing through either its safe hops, or its new primary. Neither of the two can be S by definition of a downstream node (for safe hops) and SPT (for new primary).

2.7 If NP is a type-C, then because DELAY_TYPEB > DELAY_TYPEC, NP must have had enough time to switch to its new primary, which can't be S by definition of SPT and considering that NP is S's nexthop in the SPT AF.

2.8 Consequently, no loops will form between a type-B node and it's new primary before or after the type-A nodes switches to it.

3. S is type C:

BF analysis:

3.1 If P is a type-A, then similarly to 1.1 before, S has not been P's LF neighbor before and hence won't be its new primary, so no loops will form between P and S.

3.2 If P is a type-B, then similarly to 1.2, S will not be used by P for at least DELAY_TYPEB, and because DELAY_TYPEB > DELAY_TYPEC, S will have enough time to switch to NP.

3.3 If P is another type-C, then it hasn't been using S as its primary BF, but it is possible for P to consider S as its new primary AF and to install routes before S after their DELAY_TYPEC expires. Hence, a microloop is possible between P and S.

3.4 Consequently, a microloop between a type-C node and its old primary is possible only if the old primary is also a type-C node and it considers S as its new primary AF. Note that DELAY_TYPEC only delays probably loop formation, but does not increase its duration, as both neighboring routers are using the same delay.

AF analysis:

3.5 If NP is a type-A, then because of the DELAY_TYPEC NP must have had enough time to switch to its new NP, which cannot be S by definition of SPT considering that NP is S's new nexthop in the SPT AF.

3.6 If NP is a type-B, then because of DELAY_TYPEC, NP must have had enough time to switch to its safe hops, which can't be S by

definition of a downstream node and considering that NP is S's new SPT next-hop.

3.7 If NP is another type-C, a loop is possible if S's DELAY_TYPEC expires before that on NP and NP has been using S as its primary BF.

3.8 Consequently, a microloop between a type-C node and its new primary is possible only if the new primary is also a type-C node and S was NP's primary BF.

4. Given the above analysis, it can be noted that, for a given failure, presence of single type-C nodes in the network does not create microloops.
It is the C-C combination that introduces this potential.

IPR Disclaimer

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Statement