Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2014

X. Zhang
G. Yan
Huawei Technologies

October 18, 2013

Algorithm for Ordered Metric Adjustment
draft-zxd-rtgwg-ordered-metric-adjustment-00

Abstract

Upon link down event or link up event, each device in network individually schedules route calculation. Because of different hardware capabilities and internal/external environments, the time to update forwarding entries on these devices are disordered which can cause a transient forwarding loop. This document introduces a method to prevent forwarding loop by adjusting link metric gradually for several times.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The internet is the most popular network, it is a distributed system, Depend on its configuration, each network device communicates with its neighbor, calculate routes and generate the FIB individually, finally, the packet will be forwarded hop by hop. But due to the difference of each device hardware capabilities and internal/external environments, the route calculation cannot be scheduled at same time, the micro-loop occur, and some mechanisms are already provided in IETF to resolve this issue, like ordered FIB.

This document tries to provide a different method to resolve this issue.

In figure 1, there are some forwarding loop scenarios:

o Upon link BA down event, for the destination A, if B updates its forwarding entry before G, a transient forwarding loop occurs between B and G. Node failure MAY be treated as multiple links' failure, such as B fails, the links GB, BA, BC go to down. For the destination A, if G updates its forwarding entry before I, a transient forwarding loop occurs between I and G.

- o Upon link BA up event, for the destination A, if G updates its forwarding entry before B, a transient forwarding loop occurs between B and G. Node failure recovery MAY be treated as multiple links’ failure recovery, such as B recovers, the links GB, BA, BC go to up. For the destination A, if I updates its forwarding entry before G, a transient forwarding loop occurs between I and G.

```
        D-------E
        |       |
        |       |
        C       F
        |       |
        |       |
        B-------A
        |       |
        |       |
        G       J
       / \     /
      /   \   /
     H      I
```

Figure 1 Topology (all links with metric 10 except links AF and AJ with metric 100)

2. Overview of Algorithm

This document introduces a method to prevent forwarding loop by adjusting link metric gradually. There are two cases to be considered here:

- o Link up event: The link metric will be decreased from maximum to configuration value. Node failure recovery MAY be treated as multiple links’ failure recovery.

- o Link down event: The link metric will be increased from configuration value to maximum. Node failure MAY be treated as multiple links’ failures.

2.1. Link up event

As we know, the optimal paths from other nodes to node R can be represented as the RSPF tree with root R. We assume that the link XR between node X and R goes to up, some nodes MAY switch their optimal paths to R and the new optimal paths include the link XR. If the metric of link XR is small enough, X will be the children of R in RSPF tree and the nodes of the sub-tree under X on the RSPF tree will switch their optimal paths to R, other nodes’ optimal paths are not changed. The nodes whose optimal paths to the R are changed are denoted by set of S. If the nodes in S switch their optimal paths when link XR goes to up, some forwarding loop MAY exist as described in section 1. In order to prevent the forwarding loop, we can control the nodes in S to switch optimal paths gradually instead of switching all the nodes at the same time. For the case of link up event, the metric of link XR is decreased gradually by several times. Once the metric of link XR is adjusted, one node or several nodes any two of which do not have forwarding loop will switch optimal path to R. Until the metric of link XR is decreased to configuration value, all the nodes in S switch their optimal paths to R. We give an example to describe the procedure of adjusting link metric as follows:

In Figure 1, suppose the link BA is down. All the paths of the other nodes to node A can be represented as RSPF(Reverse Shortest Path First) tree with root A(as figure 2 below).

```
             A
            / \
           /   \
          J     F
         /       \
        /         \
       I           E
      /             \
     /               \
    G                 D
   /|                  \
  / |                   \
 B  H                    C
```

Figure 2 Reverse Shortest Path First Tree

After link BA going to up, node B will start to adjust the metric of link BA and repeat adjusting several times as below:

- The metric of link BA is set to 121. For the destination A, only B’s optimal path has changed. The following figure 3 describes the RSPF tree with root A after the first adjustment.

```
               A
              /|\
             / | \
            J  B  F
           /       \
          /         \
         I           E
        /             \
       /               \
      G                 D
      |                  \
      |                   \
      H                    C
```

Figure 3 Reverse Shortest Path First Tree (The metric of link BA is 121)

- The metric of link BA is set to 101. For the destination A, the node C, G and H’s optimal paths have changed. The following figure 4 describes the RSPF tree with root A after this adjustment.

```
                A
               /|\
              / | \
             J  B  F
            /  / \  \
           /  /   \  \
          I  C     G  E
                    \  \
                     \  \
                      H  D
```

Figure 4 Reverse Shortest Path First Tree(The metric of link BA is 101)

- The metric of link BA is set to 81. For the destination A, the node D and I’s optimal paths have changed. The figure 5 below describes the RSPF tree with root A after this adjustment.

```
                   A
                  /|\
                 / | \
                J  B  F
                  / \  \
                 /   \  \
                C     G  E
                |     |\
                |     | \
                D     I  H
```

Figure 5 Reverse Shortest Path First Tree(The metric of link BA is 81)

o The metric of link BA is set to 61. For the destination A, the node E and J’s optimal paths have changed. The figure 6 below describes the RSPF tree with root A after this adjustment.

```
                A
                |\
                | \
                B  F
               / \
              /   \
             C     G
             |     |\
             |     | \
             D     I  H
             |     |
             |     |
             E     J
```

Figure 6 Reverse Shortest Path First Tree(The metric of link BA is 61)

o The metric of link BA is set to 10 which is configuration value. For the destination A, node F’s optimal path has changed. The figure 7 below describes the RSPF tree with root A after this adjustment.

```
                   A
                   |
                   |
                   B
                  / \
                 /   \
                C     G
                |     |\
                |     | \
                D     I  H
                |     |
                |     |
                E     J
                |
                |
                F
```

Figure 7 Reverse Shortest Path First Tree(The metric of link BA is 10)

As shown in the example above, one or more nodes’ optimal paths to node A will be affected when the metric of link is adjusted every time. The nodes not affected keep the outgoing interfaces and next hops unchanged. There is no forwarding loop during this adjustment.(as in Figure 3 H, G, etc.).

Once the link metric is adjusted, the new LSP/LSA will be generated with the new metric information and will be flooded to the same area/ level, and all the nodes in the same area/level will calculate routes. So all the nodes need enough time to flood new LSP/LSA and calculate routes before the next adjustment of link metric. Usually, it needs a few seconds to tens of seconds delay to start a new adjustment. The delay between different adjustments will avoid to affect each other.

Similarly, for the destination B, we can use the same method to adjust the metric of link AB to avoid forwarding loop.

2.2. Link down event

In Figure 1, supposing the link BA goes to down, for the destination A, it can avoid forwarding loop by increasing metric of link BA from configured value to maximum. The process of adjustment is reverse to the process of link up event. And the RSPF tree with root A is changed from figure 7 to figure 2 gradually.

The adjustment of metric just involves the nodes connected to the failure link, other nodes just do normal route calculation. So the method is very convenient for incremental deployment.

3. Algorithm Sections

In figure3, the metric of the link BA is set to 121, node B switches an optimal path to A, while the other nodes do not switch their optimal paths. In fact the metrics ranged from 121 to 129 of the link BA is also valid to ensure that only the node B has to switch to an optimal path to A. The following algorithm 3.1 is used to calculate a reasonable adjustment metric range for each node to switch its optimal path without forwarding loop.

We first assume the failure link is L(for example, link BA in figure 1), and its configuration metric value is K. The destination node of link L is root node(for example, node A) which is referenced in the following sections.

3.1. Calculating the adjustment range of link metric for each node

1. Supposing the link L(for example, link BA in figure 1) is down, the metric of link L can be considered as maximum. It is easy to use RSPF algorithm to calculate the distance of each node i to root node. The distance from each node i to the root is recorded as D(i, max).

2. Supposing the link L is up, the metric of link L is set to configured value. It is easy to use RSPF algorithm to calculate the distance of each node i to root which is the destination node of link L. The distance from each node i to the root is recorded as D(i, min).

3. If node i switches its optimal path to root node, the reasonable upper metric of link L can be adjusted is COST(i, max), COST(i, max) = D(i, max) - D(i, min)+K.

4. If node i switches its optimal path to root node, the reasonable lower metric of link L can be adjusted is COST(i, min), COST(i, min) = MAX{COST(j, max)}, where j is the son of node i in case of link L being up.

5. When the link L’s metric is set in the range of (COST(i, min), COST(i, max)), node i can be switched to its optimal path to the root node without forwarding loop.

3.2. Determine existing forwarding loop or not between two direct nodes

F is the parent node, S is its son node. if COST(F, max) equals COST(S, max), when F switches to the new optimal path to root because of link L’s metric’s adjustment, S will switches simultaneously with F without forwarding loop.

3.3. Algorithm of multiple nodes simultaneously switch optimal path without forwarding loop

1. Initialize three queues: TentList, CandList, OutPutList.

2. The destination node of link L is recorded as root.

3. Push the root node to TentList.

4. Get the node N from TentList, where N is the node whose COST(i, min) is maximum in TentList. if TentList is empty, we cannot get any node, then this algorithm terminates.

5. Move node N to the tail of OutPutList.

6. Push every son node Si of N to CandList, where COST(Si, max) does not equal COST(N, max).

7. For each node mi in TentList, if COST(mi, max) > COST(N, min), remove the node mi from TentList.

8. When mi is deleted from TentList, push every son node Sj of mi to CandList, where COST(Sj, max) does not equal COST(mi, max).

9. Move all the nodes from CandList to Tentlist, then CandList is empty.

10. goto 4.

11. When the algorithm finishes, OutPutList stores node N1, N2,... Ns,

    * In case of link L going to up, the adjustment process of metric of link L is COST(N1, min)+1, COST(N2, min)+1,... COST(Ns, min)+1, and configuration value K.

    * In case of link L going to down, the adjustment process of metric of link L is configuration value K, COST(Ns, min)+1, ... COST(N2, min)+1 and COST(N1, min)+1.

4. IANA Considerations

This document includes no request to IANA.

5. Security Considerations

This document is not currently believed to introduce new security concerns.

6. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[RFC5715] Shand, M. and S. Bryant, "A Framework for Loop-Free Convergence", RFC 5715, January 2010.

[RFC6976] Shand, M., Bryant, S., Previdi, S., Filsfils, C., Francois, P., and O. Bonaventure, "Framework for Loop-Free Convergence Using the Ordered Forwarding Information Base (oFIB) Approach", RFC 6976, July 2013.

Authors' Addresses

Xudong Zhang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: zhangxudong@huawei.com

Gang Yan
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: yangang@huawei.com