Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 18, 2011

H. Chen
Huawei Technology, Inc.
September 14, 2010

Intelligent OSPF Link State Database Exchange
draft-chen-ospf-intelligent-db-exch-01.txt

Abstract

This document presents an intelligent database exchange mechanism for the database exchange procedure in OSPFv2 and OSPFv3. This mechanism is backward compatible. It eliminates the unnecessary database exchanges through using the existing reachability information calculated from the link state database and the un-reachability information about routers recorded. It significantly speeds up the establishment of the full adjacency between two routers in some situations.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

If a full adjacency is to be formed between two OSPF routers, their link state databases will be synchronized through the database exchange procedure described in OSPFv2 [RFC2328] and OSPFv3 [RFC2740]. Each of the routers sends the other the summary of its database through a set of Database Description packets containing the header of every LSA in its database. For every LSA header received in the Database Description packets, the router compares it with the corresponding LSA instance in its database and sends the other router a request for the LSA if the LSA instance in its database is older. The adjacency becomes full when the router finishes sending the summary of its database and processing all the Database Description packets from the other router and gets all the LSAs it requested.

1.1. Eliminating Whole Link State Database Exchange

In some situations, the whole link state database exchange is unnecessary. For example, in the case illustrated in the figure below, we suppose that full adjacencies between routers RT3 and RT4, RT4 and RT5, and RT5 and RT6 have been established, and a new full adjacency between RT3 and RT6 is to be formed over the link directly connecting them. In this case, RT3 and RT6 are reachable each other through RT4 and RT5. They do not need to exchange their link state databases at all since they are reachable each other and already have the same database.

```
                    +
                    | 3+---+                     N12      N14
                  N1|--|RT1|\ 1                    \ N13 /
                    |  +---+ \                     8\ |8/8
                    +         \ ____                 \|/
                               /    \   1+---+8    8+---+
                              *  N3  *---|RT4|------|RT5|
                               \____/    +---+      +---+
                     +         /   |                  |6
                     | 3+---+ /    |                  |
                   N2|--|RT2|/1    |1                 |6
                     |  +---+    +---+6            6+---+
                     +           |RT3|==============|RT6|
                                 +---+              +---+
```

Figure 1: A Full Adjacency between RT3 and RT6 to be Formed

For a large database, eliminating the unnecessary database exchange will significantly speed up the establishment of the full adjacency and save lots of link bandwidth for the transmission of the

unnecessary Database Description packets and CPU cycles for the processing of the packets.

1.2. Eliminating Part of Link State Database Exchange

In some other cases, some part of the database exchange is not needed. For example, in the topology shown in the figure below, when the only connection between two routers Rt4 and RT5 is down for a short time and then up again, most parts of their link state databases are the same and only small parts of the databases may be different. In this case, it is not necessary for these two routers to exchange the parts of their databases that are the same.

```
                    +
                    | 3+---+                     N12      N14
                  N1|--|RT1|\ 1                    \ N13 /
                    |  +---+ \                     8\ |8/8
                    +         \ ____                 \|/
                               /    \   1+---+8    8+---+
                              *  N3  *---|RT4|======|RT5|
                               \____/    +---+      +---+
                     +         /   |                  |6
                     | 3+---+ /    |                  |
                   N2|--|RT2|/1    |1                 |6
                     |  +---+    +---+6            6+---+
                     +           |RT3|              |RT6|
                                 +---+              +---+
```

Figure 2: Link between RT4 and RT5 goes Down and then Up

Another example is in a Mobile Ad-Hoc Network (MANET) where nodes are mobile. When a mobile node moves out of a transmission range and into another range in the same OSPF area in a short time, the adjacencies to the nodes in the old range will be down and the new adjacencies to some nodes in the new range will be established. In this situation, most of their databases are the same.

This document describes a solution, called an intelligent database exchange mechanism, for eliminating the unnecessary database exchanges and processes during the establishment of a full adjacency between two routers. This solution is backward compatible.

2. Terminology

This document uses terminologies defined in RFC 2328, and RFC 2740.

3. Conventions Used in This Document

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.

4. Intelligent Link State Database Exchange

The intelligent OSPF Link State database exchange mechanism eliminates the unnecessary Database Description packets exchanges and processes through using the existing reachability information calculated from the link state database and the un-reachability information about routers recorded.

When two OSPF routers are going to bring up a full adjacency between them, each of them checks whether it is reachable to the other through using its route table calculated from its link state database. If it can reach the other router, it does not need to send the other any Database Description packets that contain the summary of its database since two routers have the same database. In this case, all the unnecessary Database Description packets exchanges and processes are eliminated. The full adjacency between two routers is formed almost immediately.

If one router can not reach to the other now but it could reach the other at time t and the difference between the current time and t is less than the LSA maximum age MaxAge (3600 seconds), then it does not need to send the other the header of every LSA in its database through Database Description packets. It just needs to send the other the headers of the LSAs that have been changed in its database since time t at which the other router became unreachable. During the intelligent database exchange, when one router detects that the other router was restarted after time t through some way such as comparing the router LSA generated by the other router with its corresponding LSA instance in its link state database, it stops its current intelligent database exchange and starts a new normal database exchange with the other router.

During the intelligent database exchange between two routers, if one router detects that the other router becomes unreachable, it stops its current intelligent database exchange and starts a new normal database exchange with the other router.

5. Changes to OSPF Protocols

The changes to the OSPF protocols include three parts. The first

part is to modify the creation of the neighbor database summary list. The second is to change the processing of a Database Description packet contents. The third is to add some data structures and procedures.

5.1. Changes to Creation of Database Summary List

In the OSPF protocols, when a local router is going to form a full adjacency with a neighboring router, it creates the neighbor database summary list that has the contents of its entire area link state database. The area link state database consists of the router-LSAs, network-LSAs and summary-LSAs contained in the area structure, along with the AS-external-LSAs contained in the global structure. The intelligent database exchange mechanism modifies the creation of the summary list as follows:

If the local router can reach the neighboring router now, then the neighbor database summary list is empty; otherwise (i.e., the router can not reach the neighboring router now), if the router could reach the neighboring router at time t and the difference between the current time and t is less than the LSA maximum age MaxAge, then the neighbor database summary list must have the contents of the LSAs that have been changed in its link state database since time t at which the neighboring router became unreachable; otherwise, the neighbor database summary list has the contents of its entire area link state database.

5.2. Changes to Processing of DD Packet Contents

In the original OSPF protocols [RFC2328, RFC2740], when the local router accepts a received Database Description Packet as the next in sequence, one part of processing of the packet contents is that the router looks up the LSA contained in the packet in its database to see whether it also has an instance of the LSA. If it does not, or if the database copy is less recent, the LSA is put on the Link state request list so that it can be requested in Link State Request Packets. This part of processing of the packet contents is modified as follows:

If the local router can reach the neighboring router now, it does not look up any LSA contained in the packet in its database to see whether it also has an instance of the LSA or if the database copy is less recent. Otherewise, the local router follows the processing of the packet contents as described in OSPF protocols.

5.3. New Data Structures and Procedures

In addition to the above modification, some new data structures and procedures should be added to provide the following functions:

1. Record function, which records the information about:

   * Every tuple <r, tu>, where r is the remote router in the entire area that became unreachable from the local router at time tu and the difference between the current time and tu is less than the LSA maximum age MaxAge (3600 seconds).

   * The time tc for every LSA at which the LSA is changed if there exist some remote routers that became unreachable less than MaxAge ago. It is not necessary to record time tc for any LSA if there is not any remote router that became unreachable within the past MagAge (3600 seconds).

2. Retrieve function, which provides the information about:

   * For a given unreachable remote router r, the time tu at which the router r became unreachable.

   * For a given time tu, all the LSAs in the link state database that have been changed since then. Normally tu is the time when a remote router became unreachable.

3. Delete function, which removes the information about:

   * Every tuple <r, tu> when the remote router r becomes reachable or the difference between the current time and tu (where tu is the time when r became unreachable) is equal to or greater than the LSA maximum age MaxAge (3600 seconds).

   * The time tc for every LSA recorded if there is not any remote router that are unreachable.

During the full adjacency establishment between two routers, either of them may restart to form the adjacency in the normal way as described in the OSPF protocols [RFC2328] and [RFC2740] if it detects that the reachability between them is broken.

6. Some Details about Implementations of New Procedures

This section describes some implementation options for the record function. The implementations of the retrieve and delete functions depend on the implementation of the record function to some extend

and should be trivial after the details of the implementation of the record function are determined.

One implementation finds the failure time and LSA change time and records them accordingly. It finds time tu for every remote router r at which r became unreachable because some failures happened at time tu and records the information about <r, tu>. It also finds time tc for every LSA at which the LSA is changed and records the information about tc for the LSA.

Another implementation reuses some of the existing information in the link state database such as LSA age, and finds some other information such as failure time and records them.

6.1. Finding and Recording Failure Time and LSA Change Time

This subsection specifies two methods for finding the failure time. One is more accurate. The other is simpler. In addition, it describes a method for calculating the time for every LSA at which it is changed.

If a remote router r becomes unreachable from reachable after the shortest path first algorithm is run against the link state database, then tuple <r, tu> is recorded, where tu is the time at which a failure such as a link down happened that leads to the router r unreachable. Failures can be classified into two classes: link/ interface failures and node failures. When an OSPF router detects a failure, it will regenerate and flood LSAs for the failure. Suppose that ti is the maximum time delay for the OSPF router to detect an interface failure and tn is the maximum time delay for the OSPF router to detect a node failure. For an LSA, assume that tp is the time delay for the LSA to reach the local router from its origin.

6.1.1. Finding and Recording Failure Time and LSA Change Time

For a point to point link failure in the network, two router LSAs with the link down are regenerated and flooded. One is by each of two end routers of the link. For a broadcast link or NBMA link failure, one network LSA and one or more router LSAs with the information about link down are generated and flooded. The network LSA is generated by the designated router and the router LSAs are generated by the routers attached to the link.

Among these LSAs, suppose that tr is the earliest time at which one of the LSAs is received. tp is less than the age of the LSA, which can be used as an estimated value of tp. We can also estimate tp as (255 - TTL of the LSA update packet). We may select the smaller one between these two estimated values for tp. In all these cases, the

exact tp is less than its estimated value. The time tu at which the interface failed can be estimated as tu = (tr - ti - tp). The estimated value for tu is earlier than the exact time at which the failure happened. This guarantees that all the LSAs that are changed after the exact tu will be included in the neighbor database summary list if a full adjacency is to be established with router r.

For n (n > 1) link failures, the time at which each of these n link failures happened can be calculated as above. The time tu is the earliest time among all these n times. If we can identify m (1 <= m <= n) link failures among these n link failures that results in the remote router r unreachable, we only need to calculate the time at which each of those m link failures happened. In this case, the time tu is the earliest time among all those m times.

For one node failure in the network, every node that has a full adjacency with the failed node will regenerate and flood the LSA with the link to the failure node down. Among these LSAs, suppose that tr is the earliest time at which one of these LSAs is received and tp is the time delay for this LSA to reach the local router from its origin, then the time tu at which the node failed can be estimated as tu = (tr - tn - tp). The estimated value for tu is earlier than the exact time at which the failure happened. This guarantees that all the LSAs that are changed after the exact tu will be included in the neighbor database summary list if a full adjacency is to be established with router r.

For k (k > 1) node failures in the network, there are at most k groups of LSAs will be regenerated and flooded in the network. Every LSA in one of these groups contains the information about the link to the same failed node down. For each group of LSAs, the time at which the node failed can be estimated in the same way as above for one node failure. The time tu at which the remote router r became unreachable because of k node failures is estimated as the earliest time among all the estimated node failure times.

In the case that there are link and node failures in the network, two times are estimated in the ways described above. One time is the time at which the earliest link failure happened. The other is the time at which the earliest node failure happened. The time tu at which the remote router r became unreachable because of link and node failures is estimated as the earlier between these two times.

6.1.2. A Simpler Method for Finding Failure Time

One simpler way for finding failure time uses all the changed LSAs received. It derives the failure time from every LSA in the way similar to the ones described above and then selects the earliest

time as tu. For every changed LSA received at time tr, the failure time derived from this LSA is (tr - max(ti, tn) - tp), where ti, tn and tp are defined above and tp is calculated in the same way as described above.

This method is simpler than the method described in the above subsection. But the failure time it estimated may not be as accurate as the one the previous method calculated. Thus the LSAs changed between these two times will be included in the neighbor database summary list when a full adjacency is to be created to router r and this failure time is used as the unreachable time for router r. However, it is not necessary to include these LSAs in the summary list.

6.1.3. Finding and Recording LSA Change Time

If there exist some remote routers that became unreachable less than MaxAge ago, for an LSA, we use the time tr at which the LSA is received as the time tc at which the LSA is changed for recording the time tc for the LSA. This time may be a little bit later than the exact time at which the LSA is changed. But it guarantees that the LSA will be included in the neighbor database summary list if a full adjacency is to be established with a router r that became unreachable before the exact tc.

There are a number of ways for recording the LSA change time. One way is to add a new field similar to the field for LSA age into the data structure for storing LSA and store the estimated change time tc into this new field for each changed LSA.

Another way for recording the LSA change time is to add a link field into the data structure for storing an LSA, a new array and a variable for the index of the array into the data structure for storing an area. The link field is used to link all the LSAs that have the same change time together. The size of the array is the number of time units in MaxAge (3600 seconds). A time unit can be one second, 1/10 second or other smaller time unit. This depends on the implementation. The array and the variable for the index can be considered as a relative clock. The index variable starts from 0, and goes up by one every time unit. When it reaches the size of the array, it starts from 0 again. If the value of the index variable is k at the current time, the time tj represented by the element of the array at index j is tj = (current time - time unit * d), where d = (k - j) if k >= j and d = (array size - j + k) if k < j. The element of the array at index j is a pointer that points to the header of the linked list of all the LSAs that are changed at the time tc, where tc = tj or (tc + one time unit) > tj if tc < tj (i.e., rounding up tc is tj).

6.2. Reusing LSA Age, Finding and Recording Adjust Time

For every remote router r that became unreachable at time tu, tu can be estimated in one of the ways described above. For every changed LSA received at time tr after time tu, its age is less than 255, which is the maximum time to live value in the IP packet containing the LSA. Thus we use tu and reuse the age of an LSA to decide whether the LSA is put into the summary list. When a full adjacency to router r is to be created, every LSA that satisfies the condition "the age of the LSA < the current time - tu + 255" (i.e., "tu - 255 < the current time - the age of the LSA") must be put into the neighbor database summary list. This will guarantee that all the LSAs that were changed after the router r became unreachable are included in the summary list. However, some LSAs that were changed before time tu may be included. In order to reduce the number of the LSAs changed before time tu to be included, we need to find the earliest time te in which the changed LSA was regenerated in term of LSA age. Thus we can use te in place of tu - 255 to decide whether an LSA must be put into the summary list. When a full adjacency to router r is to be created, every LSA that satisfies the condition "te < the current time - the age of the LSA" must be put into the neighbor database summary list.

For each changed LSA received at time tr-i after time tu and before time (tu + 255), the time at which the LSA was regenerated is estimated as (tr-i - ta-i), where ta-i is the age of the LSA. The time te is the earliest one among all (tr-i - ta-i).

For the case that another remote router r' became unreachable later during the calculation of the time te for router r, we stop finding te and s tart to find the earliest time te' for the remote router r' in the same way as described above and mark that the time te relies on the time te'. When a full adjacency to router r is to be created, the time te is calculated as follows. It is the earliest time among the te partially calculated and the te' that the te depends on.

7. **Security** Considerations

The mechanism described in this document does not raise any new security issues for the OSPF protocols.

8. IANA Considerations

This document specifies a backward compatible intelligent link state database exchange mechanism for OSPFv2 and OSPFv3, which does not require any new number assignment.

9. Acknowledgement

The author would like to thank Richard Li, Yang Yu, Steve Yao, Quintin Zhao and others for their valuable comments on this draft.

10. References

10.1. Normative References

[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.

[RFC2740] Coltun, R., Ferguson, D., and J. Moy, "OSPF for IPv6", RFC 2740, December 1999.

10.2. Informative References

[RFC5243] Ogier, R., "OSPF Database Exchange Summary List Optimization", RFC 5243, May 2008.

Author's Address

Huaimo Chen
Huawei Technology, Inc.
Boston, MA
US

Email: Huaimochen@huawei.com