OSPF, MTU and LSA Packing Tech Note

Document ID: 116119

Contributed by Luc De Ghein, Cisco TAC Engineer.
Jul 29, 2013

Contents

Introduction

This document describes the interaction of Open Shortest Path First (OSPF) packets, maximum transition unit (MTU), Link State Advertisements (LSAs), and Link State (LS) Update packets in the context of Cisco bug ID CSCse01519.

OSPF Packet Size

Links on routers have a MTU. Outgoing packets, such as OSPF packets, cannot be larger than the interface MTU.

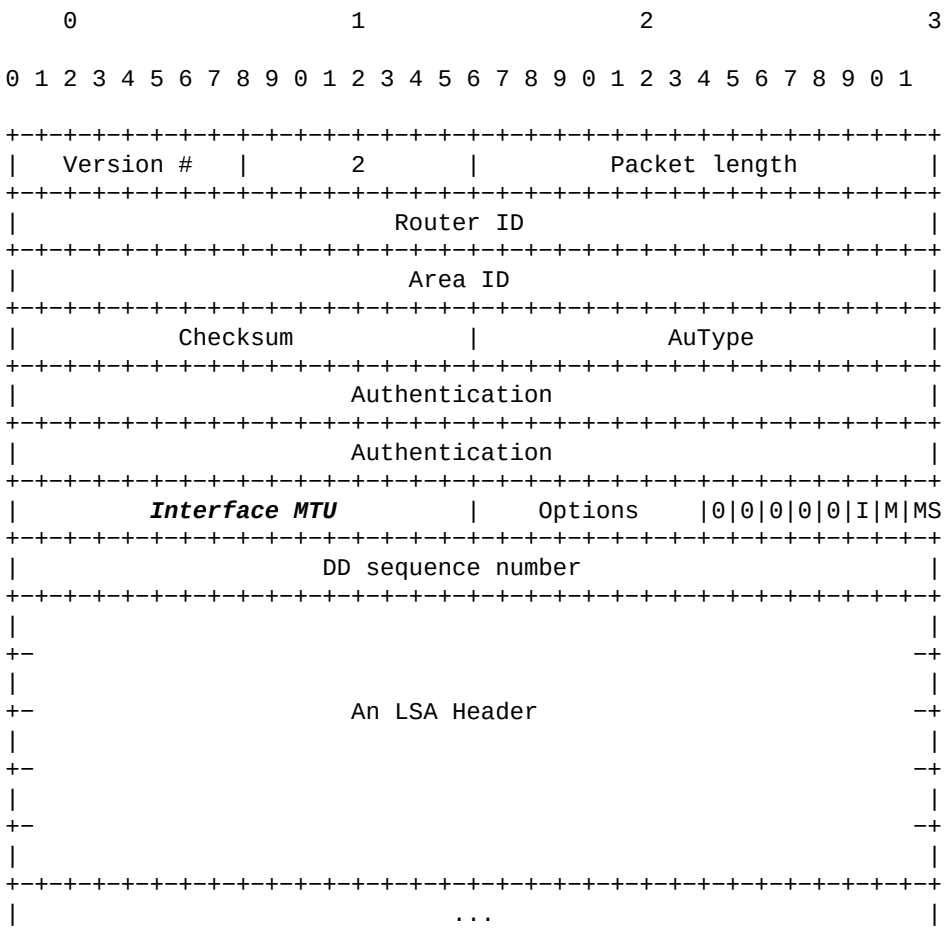
Request for Comments (RFC) 2328 documents version 2 of the OSPF protocol. Appendix A.1 of RFC 2328 describes the Encapsulation of OSPF packets in this manner:

> OSPF runs directly over the Internet Protocol's network layer. OSPF packets are therefore encapsulated solely by IP and local data−link headers.
>
> OSPF does not define a way to fragment its protocol packets, and depends on IP fragmentation when transmitting packets larger than the network MTU. If necessary, the length of OSPF packets can be up to 65,535 bytes (including the IP header). The OSPF packet types that are likely to be large (Database Description Packets, Link State Request, Link State Update, and Link State Acknowledgment packets) can usually be split into several separate protocol packets, without loss of functionality. This is recommended; IP fragmentation should be avoided whenever possible.

There can be one or more LSAs in an LS Update packet. Many LSAs in one LS Update packet is known as packing LSAs into a LS Update packet.

MTU in DBD Packet

The Database Description (DBD) packet, also specified in RFC 2328, describes the contents of the OSPF link−state database:



Appendix A.3.3. of RFC 2328 describes the interface MTU as:

> The size in bytes of the largest IP datagram that can be sent out the associated interface, without fragmentation.

Routers that are attached to a link exchange their interface MTU value in DBD packets when the OSPF adjacency is initialized.

Section 10.6 of RFC 2328 states:

> If the Interface MTU field in the Database Description packet indicates an IP datagram size that is larger than the router can accept on the receiving interface without fragmentation, the Database Description packet is rejected.

When the ***debug ip ospf adj*** command is used, you can see the arrival of these DBD packets.

In this example, there is a mismatch in MTU values between two OSPF neighbors. This router has MTU 1600:

```
OSPF: Rcv DBD from 10.100.1.2 on GigabitEthernet0/1 seq 0x2124 opt 0x52 flag 0x2
   len 1452 mtu 2000 state EXSTART
OSPF: Nbr 10.100.1.2 has larger interface MTU
```

The other OSPF router has interface MTU 2000:

```
OSPF: Rcv DBD from 10.100.100.1 on GigabitEthernet0/1 seq 0x89E opt 0x52 flag 0x7
   len 32 mtu 1600 state EXCHANGE
OSPF: Nbr 10.100.100.1 has smaller interface MTU
```

The DBD packets are retransmitted continuously until the OSPF adjacency is eventually torn down.

```
OSPF: Send DBD to 10.100.1.2 on GigabitEthernet0/1 seq 0x9E6 opt 0x52 flag 0x7
   len 32
OSPF: Retransmitting DBD to 10.100.1.2 on GigabitEthernet0/1 [10]
OSPF: Send DBD to 10.100.1.2 on GigabitEthernet0/1 seq 0x9E6 opt 0x52 flag 0x7
   len 32
OSPF: Retransmitting DBD to 10.100.1.2 on GigabitEthernet0/1 [11]
%OSPF-5-ADJCHG: Process 1, Nbr 10.100.1.2 on GigabitEthernet0/1 from EXSTART to
   DOWN, Neighbor Down: Too many retransmissions
```

OSPF Behavior and Packing LSAs into a LS Update Packet

Before Cisco Bug ID CSCse01519

Before Cisco bug ID CSCse01519, OSPF in the Cisco IOS® software built OSPF packets no larger than1500 bytes, regardless of the interface MTU. So, if the interface MTU was larger than 1500 bytes, OSPF still packed only up to 1500 bytes into an OSPF packet. This was somewhat inefficient because OSPF could send larger packets on the link and achieve greater throughput.

Note: There was one exception to this scenario. If one LSA held more than 1500 bytes, OSPF built that packet, no matter the size, because OSPF cannot fragment one LSA. The IP stack of the router then fragmented the packet in order to fit the MTU of the outgoing interface. This typically occurred when an OSPF router had many links, and the router LSA became larger than the link MTU.

Similarly, if the MTU of the outgoing interface was smaller than 1500 bytes, the OSPF process still built or packed OSPF packets up 1500 bytes, and the IP stack of the router fragmented the packet into smaller IP packets in order to fit the MTU of the outgoing link. This typically occurred with an IPSec tunnel between two routers that were running OSPF. The added overhead of the encapsulation bytes of the tunnel led to an MTU that was smaller than 1500 bytes. OSPF built OSPF packets up to 1500 bytes and the packets were then fragmented before the router transmitted them. This was an additional inefficiency.

After Cisco Bug ID CSCse01519

After Cisco bug ID CSCse01519, OSPF in IOS can pack OSPF packets to be larger than 1500 bytes. This occurs if the MTU of the outgoing interface is larger than 1500 bytes. Transmissions are more efficient because more information can be packed into one larger packet. In other words, if one OSPF router needs to transmit many external LSAs to an OSPF neighbor, it can pack more external LSAs into one LS Update packet if that router runs IOS with Cisco bug ID CSCse01519 implemented.

Cisco bug ID CSCse01519 also allows OSPF to build packets smaller than 1500 bytes. In some scenarios, the MTU between two OSPF neighbors is smaller than 1500 bytes. In the previous example with an IPSec tunnel, OSPF transmits OSPF packets that are smaller than 1500 bytes and avoids IP fragmentation; again, the exception is the case of an LSA that is larger than the interface MTU.

Cisco Bug ID CSCse01519

When you upgrade an OSPF router, you may discover an OSPF MTU issue caused by Cisco bug ID CSCse01519.

Overview

Many networks have OSPF neighbors which are connected through a Layer 2 (L2) switched network, or transport network, comprised of L2 VPN service or a Synchronous Digital Hierarchy/Synchronous Optical

Network (SDH/SONET) network. These transport networks can have different MTU settings than the routers that are running OSPF.

Although the MTU setting should be correct on all routers and should reflect the true MTU, there are often mistakes that go unnoticed.

This is an example network with two routers that are running OSPF. Router 1 (R1) and router 2 (R2) are connected through an L2 switch.

Figure 1 : Example network

In this example, the routers have GigabitEthernet interfaces with an MTU set to 2000. The MTU of the L2 switch is only 1500 bytes.

If the size of the data traffic is never larger than 1500 bytes, you can use IOS without Cisco bug ID CSCse01519 because the OSPF packets are never larger than 1500 bytes. However, if there is an LSA that is 1800 bytes, for example, the OSPF process on R1 or R2 builds a LS Update packet larger than 1500 bytes and transmits it, but the packet is dropped by the L2 switch between the routers.

If the OSPF database on R2 has enough networks, the locally originated LSAs are so large that a LS Update packet might be larger than the interface MTU.

- If these networks are originated by the covering network command, the networks appear in the router LSA of R2. R2 builds a router LSA that is larger than 2000 bytes and transmits it, but IP fragments it to 2000 bytes, the interface MTU. The L2 switch however drops these packets. OSPF then retransmits this packet endlessly, and the OSPF adjacency state is never full. So, the issue is immediately discovered, even when you are running IOS without Cisco bug ID CSCse01519.
- If these networks are originated by the ***redistribute connected*** command, the networks appear in external LSAs. OSPF tries to pack external LSAs into one LS Update packet that is up to 1500 bytes in size. In this case, because the interface MTU is 2000 bytes, the OSPF adjacency reaches the 'FULL' state. The issue of an inadequate underlying MTU is not immediately discovered. The issue will be discovered when one router is upgraded to IOS with Cisco bug ID CSCse01519.

Scenario

Assume that both routers run an IOS version without Cisco bug ID CSCse01519.

When the OSFP adjacency builds, notice that R1 never receives an OSPF packet larger than 1500 bytes, although the MTU of the interfaces is 2000.

Enable the ***debug ip ospf packets*** command.

```
OSPF: rcv. v:2 t:1 l:48 rid:10.100.1.2
      aid:0.0.0.0 chk:72CF aut:0 auk: from GigabitEthernet0/1
```

```
...
OSPF: rcv. v:2 t:4 l:1468 rid:10.100.1.2
      aid:0.0.0.0 chk:8389 aut:0 auk: from GigabitEthernet0/1
OSPF: rcv. v:2 t:4 l:136 rid:10.100.1.2
...
```

In this debug output, 'l:1468' is the length of the OSPF packet, so you can see that the largest OSPF packet was 1468 bytes. 't:4' indicates that the OSPF packet is type 4, which is a Link State Update packet. This table from section 4.3 of RFC 2328 defines the different OSPF packet types:

Type	*Packet name*	*Protocol function*
1	Hello	Discover/maintain neighbors
2	Database Description	Summarize database contents
3	Link State Request	Database download
4	Link State Update	Database update
5	Link State Ack	Flooding acknowledgment

The OSPF adjacency reaches the 'FULL' state.

```
R1#show ip ospf neighbor gigabitEthernet 0/1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.1.2        0   FULL/  -        00:00:34    10.1.1.2        GigabitEthernet0/1

R2#show ip ospf neighbor gigabitEthernet 0/1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.100.1      0   FULL/  -        00:00:34    10.1.1.1        GigabitEthernet0/1
```

Next, upgrade IOS on R2 to an IOS version with Cisco bug ID CSCse01519.

```
R2#show ip ospf neighbor gigabitEthernet 0/1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.100.1      0   LOADING/  -     00:00:33    10.1.1.1        GigabitEthernet0/1

R2#show ip ospf neighbor gigabitEthernet 0/1 detail
Neighbor 10.100.100.1, interface address 10.1.1.1
    In the area 0 via interface GigabitEthernet0/1
    Neighbor priority is 0, State is LOADING, 5 state changes
    DR is 0.0.0.0 BDR is 0.0.0.0
    Options is 0x12 in Hello (E-bit L-bit )
    Options is 0x52 in DBD (E-bit L-bit O-bit)
    LLS Options is 0x1 (LR)
    Dead timer due in 00:00:39
    Neighbor is up for 00:00:49
    Index 1/1, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
    Number of retransmissions for last link state request packet 9
    Poll due in 00:00:00

R2#show ip ospf neighbor gigabitEthernet 0/1 detail
Neighbor 10.100.100.1, interface address 10.1.1.1
    In the area 0 via interface GigabitEthernet0/1
    Neighbor priority is 0, State is LOADING, 5 state changes
    DR is 0.0.0.0 BDR is 0.0.0.0
    Options is 0x12 in Hello (E-bit L-bit )
    Options is 0x52 in DBD (E-bit L-bit O-bit)
    LLS Options is 0x1 (LR)
```

```
    Dead timer due in 00:00:33
    Neighbor is up for 00:02:06
    Index 1/1, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
    Number of retransmissions for last link state request packet 25
    Poll due in 00:00:03

%OSPF-5-ADJCHG: Process 1, Nbr 10.100.100.1 on GigabitEthernet0/1 from LOADING
to DOWN, Neighbor Down: Too many retransmissions
```

The OSPF adjacency is stuck in 'LOADING' state and does not reach the 'FULL' state. Retransmissions occur until OSPF reaches its limit of 25 retransmissions. OSPF tries to establish the adjacency again, the same issue reoccurs, and the loop continues endlessly.

Thus, the upgrade on R2 uncovers a previously hidden issue: the underlying MTU is smaller than the one used by the OSPF routers.

When the switch changes MTU to 2000, an OSPF packet larger than 1500 bytes ('l:1980') is transmitted with no problem.

```
R1#
OSPF: rcv. v:2 t:3 l:1980 rid:10.100.1.2
      aid:0.0.0.0 chk:AC5B aut:0 auk: from GigabitEthernet0/1
```

In order to check underlying MTU issues, always ping the OSPF neighbor IP address with a size equal to the MTU and the DF (don't fragment) bit set.

In order to discover the value of the underlying MTU, perform a ping, and sweep the size. Count the number of exclamation marks (!) in the output in order to determine the correct MTU. In this example, the last echo reply from the ***ping*** command has size 1500 bytes.

```
R2#ping
Protocol [ip]:
Target IP address: 10.1.1.1
Repeat count [5]: 1
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: yes
Source address or interface:
Type of service [0]:
Set DF bit in IP header? [no]: yes
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]: yes
Sweep min size [36]: 1460
Sweep max size [18024]: 1540
Sweep interval [1]:
Type escape sequence to abort.
Sending 81, [1460..1540]-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
Packet sent with the DF bit set
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.............................
...........
Success rate is 49 percent (40/81), round-trip min/avg/max = 1/1/4 ms
```