

Exstart/Exchange 상태에서 Open Shortest Path First Neighbors 고착 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

[배경 정보](#)

[Exstart 상태](#)

[교환 상태](#)

[인접 디바이스가 Exstart/Exchange 상태에 머물러 있음](#)

[솔루션](#)

[관련 정보](#)

소개

이 문서에서는 OSPF(Open Shortest Path First) 인접 디바이스가 Exstart 및 Exchange 상태에서 중단되는 상황을 해결하는 방법을 설명합니다.

사전 요구 사항

요구 사항

사용자는 기본 OSPF 작업 및 컨피그레이션, 특히 OSPF 네이버 상태에 대해 잘 알고 있는 [것이 좋습니다](#).

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco 2503 라우터
- Cisco IOS[®] Software 릴리스 12.2(24a) - 두 라우터에서 실행

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

표기 규칙

문서 규칙에 대한 자세한 내용은 Cisco 기술 팁 표기 규칙을 참조하십시오.

배경 정보

인접성 형성을 위한 OSPF 상태는 Down, Init, 2-way, Exstart, Exchange, Loading 및 Full입니다. OSPF(Open Shortest Path First) 인접 디바이스가 Exstart/Exchange 상태에 머물러 있는 이유는 여러 가지가 있습니다. 이 문서에서는 Exstart/Exchange 상태를 초래하는 OSPF 네이버 간의 MTU 불일치에 초점을 맞춥니다. OSPF 문제 해결 방법에 대한 자세한 내용은 [OSPF 문제 해결을 참조하십시오](#).

Exstart 상태

두 개의 OSPF 인접 라우터가 양방향 통신을 설정하고 DR/BDR 선택을 완료하면(다중 액세스 네트워크에서) 라우터가 Exstart 상태로 전환됩니다. 이 상태에서 인접 라우터는 기본/하위 관계를 설정하고 DBD 패킷이 교환되는 동안 사용할 초기 DBD(데이터베이스 설명자) 시퀀스 번호를 결정합니다.

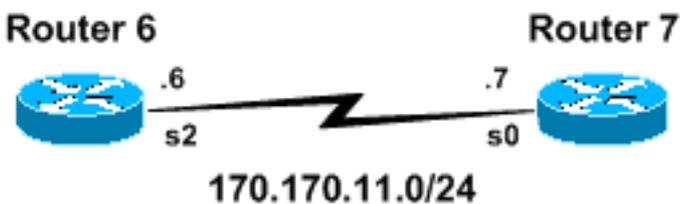
교환 상태

일단 Primary/Subordinate 관계가 협상되었습니다(라우터 ID가 가장 높은 라우터가 기본 라우터가 됨). 인접 라우터는 Exchange 상태로 전환됩니다. 이 상태에서 라우터는 전체 링크 상태 데이터베이스를 설명하는 DBD 패킷을 교환합니다. 라우터는 또한 링크 상태 요청 패킷을 전송하며, 인접 디바이스에서 보다 최근의 LSA(링크 상태 알림)를 요청합니다.

일반적인 OSPF 인접성 구축 프로세스 중에 OSPF 인접 디바이스가 Exstart/Exchange 상태를 통해 전환되지만 OSPF 인접 디바이스가 이 상태로 유지되는 것은 정상적인 상황이 아닙니다. 다음 섹션에서는 OSPF 인접 디바이스가 이 상태에서 중단되는 가장 일반적인 이유를 설명합니다. 여러 [OSPF 상태](#)에 대한 자세한 내용은 OSPF 네이버 상태를 참조하십시오.

인접 디바이스가 Exstart/Exchange 상태에 머물러 있음

이 문제는 Cisco 라우터와 다른 벤더 라우터 간에 OSPF를 실행하려고 할 때 가장 자주 발생합니다. 최대 전송 단위(MTU) 설정이 neighboring 라우터 인터페이스가 일치하지 않습니다. MTU가 높은 라우터가 인접 라우터에 설정된 MTU보다 큰 패킷을 전송하면 인접 라우터는 패킷을 무시합니다. 이 문제가 발생하면 `show ip ospf neighbor` 명령은 이 그림과 유사한 출력을 표시합니다.



라우터 6과 라우터 7이 프레임 릴레이를 통해 연결

이 섹션에서는 이 문제를 실제로 재현하는 방법에 대해 설명합니다.

이 그림의 라우터 6과 라우터 7은 프레임 릴레이를 통해 연결되며 라우터 6은 OSPF에 재배포된 5개의 고정 경로로 구성되었습니다. 라우터 6의 직렬 인터페이스는 기본 MTU가 1500인 반면 라우터 7의 직렬 인터페이스는 MTU가 1450입니다. 각 라우터 컨피그레이션이 테이블에 표시됩니다(필요한 컨피그레이션 정보만 표시됨).

라우터 6 컨피그레이션

라우터 7 컨피그레이션

```
interface Serial2
```

```
!--- MTU is set to its default value of 1500.
```

```
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay lmi-type ansi
!
interface Serial2.7 point-to-point
ip address 10.170.10.6 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 101
!
router ospf 7
 redistribute static subnets
 network 10.170.10.0 0.0.0.255 area 0
!
ip route 192.168.0.10 255.255.255.0 Null0
ip route 192.168.10.10 255.255.255.0 Null0
ip route 192.168.10.0 255.255.255.0 Null0
ip route 192.168.37.10 255.255.255.0 Null0
ip route 192.168.38.10 255.255.255.0 Null0
```

```
interface Serial0
mtu 1450
no ip address
no ip directed-broadcast
encapsulation frame-relay
frame-relay lmi-type ANSI
!
interface Serial0.6 point-to-p
ip address 172.16.7.11
255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 11
!
router ospf 7
 network 172.16.11.6 0.0.0.255
0
```

각 라우터에 대한 **show ip ospf neighbor** 명령 출력은 다음과 같습니다.

```
router-6# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.7.11	1	EXCHANGE/ -	00:00:36	172.16.7.11	Serial2.7

```
router-6#
```

```
router-7# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.170.10.6	1	EXSTART/ -	00:00:33	10.170.10.6	Serial0.6

```
router-7#
```

이 문제는 교환 상태에서 라우터 6이 1450바이트보다 큰 DBD 패킷(라우터 7의 MTU)을 보낼 때 발생합니다. 이 **debug ip packet** 및 **debug ip ospf adj** 명령을 실행하여 OSPF 인접성 프로세스를 확인할 수 있습니다. 라우터 6 및 7의 1~14단계 출력은 다음과 같습니다.

1. 라우터 6 디버그 출력:

```
<<<ROUTER 6 IS SENDING HELLOS BUT HEARS NOTHING, STATE OF NEIGHBOR IS DOWN>>>
00:03:53: OSPF: 172.16.7.11 address 172.16.7.11 on Serial2.7 is dead 00:03:53: OSPF:
172.16.7.11 address 172.16.7.11 on Serial2.7 is dead, state DOWN
```

2. 라우터 7 디버그 출력:

```
<<<OSPF NOT ENABLED ON ROUTER7 YET>>>
```

3. 라우터 6 디버그 출력:

```
<<<ROUTER 6 SENDING HELLOS>>>
00:03:53: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7), len 64, sending
broad/multicast, proto=89 00:04:03: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7), Len
64, sending broad/multicast, proto=89
```

4. 라우터 7 디버그 출력:

```
<<<OSPF NOT ENABLED ON ROUTER7 YET>>>
```

5. 라우터 7 디버그 출력:

```
<<<OSPF ENABLED ON ROUTER 7, BEGINS SENDING HELLOS AND BUILDING A ROUTER LSA>>>
00:17:44: IP: s=172.16.7.11 (local), d=224.0.0.5 (Serial0.6), Len 64, sending
broad/multicast, proto=89 00:17:44: OSPF: Build router LSA for area 0, router ID
```

172.16.7.11, seq 0x80000001

6. 라우터 6 디버그 출력:

```
<<<RECEIVE HELLO FROM ROUTER7>>>
00:04:04: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5, Len 64, rcvd 0, proto=89 00:04:04:
OSPF: Rcv hello from 172.16.7.11 area 0 from Serial2.7 172.16.7.11 00:04:04: OSPF: End of
hello processing
```

7. 라우터 6 디버그 출력:

```
<<<ROUTER 6 SEND HELLO WITH ROUTER7 ROUTERID IN THE HELLO PACKET>>>
00:04:13: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7), Len 68, sending
broad/multicast, proto=89
```

8. 라우터 7 디버그 출력:

```
<<<ROUTER 7 RECEIVES HELLO FROM ROUTER6 CHANGES STATE TO 2WAY>>>
00:17:53: IP: s=10.170.10.6 (Serial0.6), d=224.0.0.5, Len 68, rcvd 0, proto=89 00:17:53:
OSPF: Rcv hello from 10.170.10.6 area 0 from Serial0.6 10.170.10.6 00:17:53: OSPF: 2 Way
Communication to 10.170.10.6 on Serial0.6, state 2WAY
```

9. 라우터 7 디버그 출력:

```
<<<ROUTER 7 SENDS INITIAL DBD PACKET WITH SEQ# 0x13FD>>>
00:17:53: OSPF: Send DBD to 10.170.10.6 on Serial0.6 seq 0x13FD opt 0x2 flag 0x7 Len 32
00:17:53: IP: s=172.16.7.11 (local), d=224.0.0.5 (Serial0.6), Len 52, sending
broad/multicast, proto=89 00:17:53: OSPF: End of hello processing
```

10. 라우터 6 디버그 출력:

```
<<<ROUTER 6 RECEIVES ROUTER7'S INITIAL DBD PACKET CHANGES STATE TO 2-WAY>>>
00:04:13: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5, Len 52, rcvd 0, proto=89 00:04:13:
OSPF: Rcv DBD from 172.16.7.11 on Serial2.7 seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450
state INIT 00:04:13: OSPF: 2 Way Communication to 172.16.7.11 on Serial2.7, state 2WAY
```

11. 라우터 6 디버그 출력:

```
<<<ROUTER 6 SENDS DBD PACKET TO ROUTER 7 (PRIMARY/SUBORDINATE NEGOTIATION - ROUTER 6 IS
SUBORDINATE)>>>
```

```
00:04:13: OSPF: Send DBD to 172.16.7.11 on Serial2.7
seq 0xE44 opt 0x2 flag 0x7 Len 32
00:04:13: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7),
Len 52, sending broad/multicast, proto=89
00:04:13: OSPF: NBR Negotiation Done. We are the SLAVE
```

12. 라우터 7 디버그 출력:

```
<<<RECEIVE ROUTER 6'S INITIAL DBD PACKET (MTU MISMATCH IS RECOGNIZED)>>>
00:17:53: IP: s=10.170.10.6 (Serial0.6), d=224.0.0.5, Len 52, rcvd 0, proto=89 00:17:53:
OSPF: Rcv DBD from 10.170.10.6 on Serial0.6 seq 0xE44 opt 0x2 flag 0x7 Len 32 mtu 1500
state EXSTART 00:17:53: OSPF: Nbr 10.170.10.6 has larger interface MTU
```

13. 라우터 6 디버그 출력:

```
<<<SINCE ROUTER 6 IS SUBORDINATE SEND DBD PACKET WITH LSA HEADERS, SAME SEQ# (0x13FD) TO
ACK ROUTER 7'S DBD. (NOTE SIZE OF PKT)>>>
```

```
00:04:13: OSPF: Send DBD to 172.16.7.11 on Serial2.7
seq 0x13FD opt 0x2 flag 0x2 Len 1472
00:04:13: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7),
Len 1492, sending broad/multicast, proto=89
```

14. 라우터 7 디버그 출력:

```
<<<NEVER RECEIVE ACK TO ROUTER7'S INITIAL DBD, RETRANSMIT>>>
00:17:54: IP: s=172.16.7.11 (local), d=224.0.0.5 (Serial0.6), Len 68, sending
broad/multicast, proto=89 00:18:03: OSPF: Send DBD to 10.170.10.6 on Serial0.6 seq 0x13FD
opt 0x2 flag 0x7 Len 32 00:18:03: OSPF: Retransmitting DBD to 10.170.10.6 on Serial0.6 [1]
```

이 시점에서 라우터 6은 라우터 7의 초기 DBD 패킷에 대한 ACK를 계속 시도합니다.

```
00:04:13: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5,
Len 68, rcvd 0, proto=89
00:04:13: OSPF: Rcv hello from 172.16.7.11 area 0 from
Serial2.7 172.16.7.11
00:04:13: OSPF: End of hello processing
```

```
00:04:18: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5,
Len 52, rcvd 0, proto=89
00:04:18: OSPF: Rcv DBD from 172.16.7.11 on Serial2.7
seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450 state EXCHANGE
```

```
00:04:18: OSPF: Send DBD to 172.16.7.11 on Serial2.7
seq 0x13FD opt 0x2 flag 0x2 Len 1472
00:04:18: IP: s=10.170.10.6 (local), d=224.0.0.5
(Serial2.7), Len 1492, sending broad/multicast, proto=89
```

```
00:04:23: IP: s=10.170.10.6 (local), d=224.0.0.5
(Serial2.7), Len 68, sending broad/multicast, proto=89
```

```
00:04:23: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5,
Len 52, rcvd 0, proto=89
00:04:23: OSPF: Rcv DBD from 172.16.7.11 on Serial2.7
seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450 state EXCHANGE
```

라우터 7의 DBD 패킷이 라우터 7 MTU에 비해 너무 크기 때문에 라우터 7은 라우터 6에서 ACK를 가져오지 않습니다. 라우터 7은 DBD 패킷을 반복적으로 재전송합니다.

```
0:17:58: IP: s=172.16.7.11 (local), d=224.0.0.5
(Serial0.6), Len 52, sending broad/multicast, proto=89
00:18:03: OSPF: Send DBD to 10.170.10.6 on Serial0.6
seq 0x13FD opt 0x2 flag 0x7 Len 32 00:18:03: OSPF:
Retransmitting DBD to 10.170.10.6 on Serial0.6 [2]
```

```
00:18:03: IP: s=172.16.7.11 (local), d=224.0.0.5
(Serial0.6), Len 52, sending broad/multicast, proto=89
00:18:03: IP: s=10.170.10.6 (Serial0.6), d=224.0.0.5,
Len 68, rcvd 0, proto=89
00:18:03: OSPF: Rcv hello from 10.170.10.6 area 0 from
Serial0.6 10.170.10.6
00:18:03: OSPF: End of hello processing
```

```
00:18:04: IP: s=172.16.7.11 (local), d=224.0.0.5
(Serial0.6), Len 68, sending broad/multicast, proto=89
```

```
00:18:03: OSPF: Send DBD to 10.170.10.6 on Serial0.6
seq 0x13FD opt 0x2 flag 0x7 Len 32 00:18:03: OSPF:
Retransmitting DBD to 10.170.10.6 on Serial0.6 [3]
```

```
00:18:08: IP: s=172.16.7.11 (local), d=224.0.0.5
(Serial0.6), Len 52, sending broad/multicast, proto=89
router-7#
```

라우터 6은 MTU가 더 높기 때문에 라우터 7의 DBD 패킷을 계속 수락하고 이를 승인하려고 시도하며 EXCHANGE 상태를 유지합니다.

라우터 7은 MTU가 더 낮기 때문에 라우터 6의 ACK와 함께 DBD 패킷을 무시하고 초기 DBD 패킷을 계속 재전송하며 EXSTART 상태를 유지합니다.

9단계와 11단계에서 라우터 7과 라우터 6은 기본/하위 협상의 일부로 설정된 플래그 0x7과 함께 첫 번째 DBD 패킷을 전송합니다. 이후 Primary/Subordinate 라우터 ID가 높기 때문에 라우터 7이 기본 라우터로 선택됩니다. 13단계와 14단계의 플래그는 라우터 7이 기본(플래그 0x7)이고 라우터 6이 하위(플래그 0x2)임을 분명히 보여 줍니다.

10단계에서 라우터 6은 라우터 7 초기 DBD 패킷을 수신하고 상태를 양방향(2-way)으로 전환합니다.

12단계에서 라우터 7은 라우터 6 초기 DBD 패킷을 수신하고 MTU 불일치를 인식합니다. 라우터 7은 DBD 패킷의 인터페이스 MTU 필드에 해당 인터페이스 MTU를 포함하므로 MTU 불일치를 인식할 수 있습니다. 라우터 6 초기 DBD는 라우터 7에서 거부됩니다. 라우터 7은 초기 DBD 패킷을 재전송합니다.

13단계에서는 라우터 6이 subordinate는 라우터 7 시퀀스 번호를 채택하고 LSA의 헤더가 포함된 두 번째 DBD 패킷을 전송합니다. 그러면 패킷의 크기가 증가합니다. 그러나 라우터 7은 라우터 7 MTU보다 크기 때문에 이 DBD 패킷을 수신하지 않습니다.

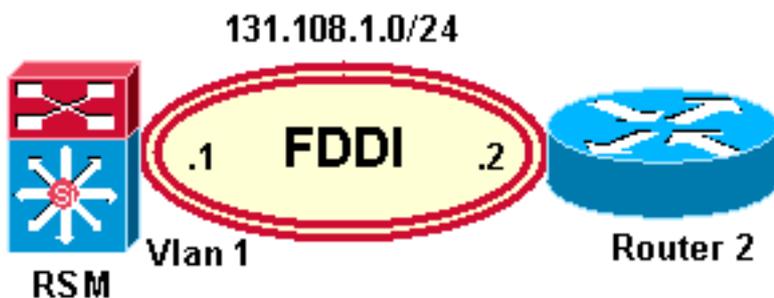
13단계 이후에도 라우터 7은 초기 DBD 패킷을 라우터 6에 계속 재전송하는 반면 라우터 6은 기본 시퀀스 번호 뒤에 오는 DBD 패킷을 계속 전송합니다. 이 루프는 무기한 계속되므로 두 라우터 중 하나가 exstart/exchange 상태에서 전환되지 않습니다.

솔루션

이 문제는 일치하지 않는 MTU로 인해 발생하므로 해결 방법은 라우터 MTU를 인접 디바이스 MTU와 일치하도록 변경하는 것입니다.

참고: Cisco IOS Software Release 12.0(3)에는 인터페이스 MTU 불일치 감지가 도입되었습니다. 이 탐지는 OSPF를 통해 DBD 패킷의 인터페이스 MTU를 광고하며, 이는 OSPFRFC [2178](#), 부록 G.9에 따릅니다. 라우터가 수신할 수 있는 MTU보다 큰 MTU를 광고하는 DBD 패킷을 수신하면 라우터는 DBD 패킷을 무시하며 인접 디바이스 상태는 Exstart를 유지합니다. 이것은 인접성의 형성을 방지합니다. 이 문제를 해결하려면 링크의 양쪽 끝에서 MTU가 동일한지 확인합니다.

Cisco IOS Software 12.01(3)에서는 `ip ospf mtu-ignore interface configuration` 명령도 도입되어 MTU 불일치 감지를 해제했습니다. 그러나 이는 이 다이어그램에 표시된 대로 드문 경우에만 필요합니다.



FDDI(Fibre Distributed Data Interface) 포트

이전 다이어그램은 Cisco Catalyst 5000의 FDDI(Fiber Distributed Data Interface) 포트와 라우터 2의 FDDI 인터페이스에 연결된 RSM(Route Switch Module)을 보여줍니다. RSM의 가상 LAN(VLAN)은 MTU가 1500인 가상 이더넷 인터페이스이며 라우터 2의 FDDI 인터페이스는 MTU가 4500입니다. 패킷이 스위치의 FDDI 포트에서 수신될 때 백플레인으로 이동하며 FDDI에서 이더넷 변환/조각화가 스위치 자체 내에서 발생합니다. 이는 유효한 설정이지만 MTU 불일치 감지 기능이 있으면 라우터와 RSM 사이에 OSPF 인접성이 형성되지 않습니다. FDDI와 이더넷 MTU가 다르므로 `ip ospf mtu-ignore`이 명령은 RSM의 VLAN 인터페이스에서 OSPF의 MTU 불일치 감지를 중지하고 인접성을 형성하는 데 유용합니다.

MTU 불일치가 가장 일반적인 이유이지만 OSPF 인접 디바이스가 Exstart/Exchange 상태에 갇히는 유일한 이유는 아닙니다. 이 문제는 DBD 패킷을 성공적으로 교환할 수 없기 때문에 가장 자주 발생합니다. 그러나 근본 원인은 다음 중 하나일 수 있습니다.

- MTU 불일치
- 유니캐스트가 끊어졌습니다. Exstart 상태에서 라우터는 Primary 및 Subordinate를 선택하기 위해 인접 디바이스로 유니캐스트 패킷을 전송합니다. 이는 point-to-point 링크가 없는 경우(이 경우 멀티캐스트 패킷 전송). 가능한 원인은 다음과 같습니다. 고도로 이중화된 네트워크의 ATM(Asynchronous Transfer Mode) 또는 프레임 릴레이 환경에서 잘못된 VC(Virtual Circuit) 매핑이 발생했습니다. MTU 문제: 라우터가 특정 길이의 패킷만 ping할 수 있음을 의미합니다. 액세스 목록은 유니캐스트 패킷을 차단합니다. NAT는 라우터에서 실행되며 유니캐스트 패킷을 변환합니다.
- PRI와 BRI/다이얼러 간의 네이버입니다.
- 두 라우터의 라우터 ID가 동일합니다(잘못 구성됨).

또한 OSPFRFC [2328](#), 섹션 10.3에는 Exstart/Exchange 프로세스가 다음 이벤트(내부 소프트웨어 문제로 인해 발생할 수 있음)에 대해 시작된다고 명시되어 있습니다.

- 시퀀스 번호가 일치하지 않습니다. 예기치 않은 DD 시퀀스 번호입니다. "I" 비트가 예기치 않게 설정되었습니다. DBD 패킷에서 받은 마지막 옵션 필드와 다른 옵션 필드입니다.
- BadLSReq네이버는 교환 프로세스 중에 인식되지 않는 LSA를 전송합니다. 교환 프로세스 중에 인접 디바이스에서 찾을 수 없는 LSA를 요청했습니다.

OSPF가 네이버를 형성하지 않는 경우, 문제를 해결하기 위해 물리적 미디어 및 네트워크 하드웨어와 같이 앞서 언급한 요인을 고려하십시오.

관련 정보

- [OSPF 네이버 상태](#)
- [OSPF 네이버 문제 설명](#)
- [Technical Support - Cisco Systems](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.