Cisco Modular Load Balancer 이해하기 (CSM & ACE)

Cisco Korea

이 창 주, Systems Engineer

changjoo@cisco.com

Contents

Prologue

본 문서는 Cisco의 차세대 L4~L7 Load Balancing Module인 ACE(Application Control Engine)에 대한 이해를 돕기 위해 작성 되었습니다.

설명 중 사용된 제품은 CSM이지만, CSM에만 적용되는 개념 보다는 Module형 Load Balancer가 어떻게 동작하고 관련된 주요 Issue들은 어떤 것들인지를 이해할 수 있도록 설명을 했기 때문에, 여기서 얻은 개념을 가지고 ACE에 접근하면 훨씬 더 이해가 빠를 것입니다.

내용은 CSM이나 그 외 Service Module에 아주 초보적인 경험만을 갖고 계신 분들을 대상으로 설명을 했으며, 스위치 내에서 일어나는 Frame Forwarding 원리, MAC address와 IP Address와의 상관 관계, VLAN의 개념 등의 CCNA 수준의 지식을 갖추신 분이 이해할 수 있도록 가급적 쉽게 작성하였습니다.

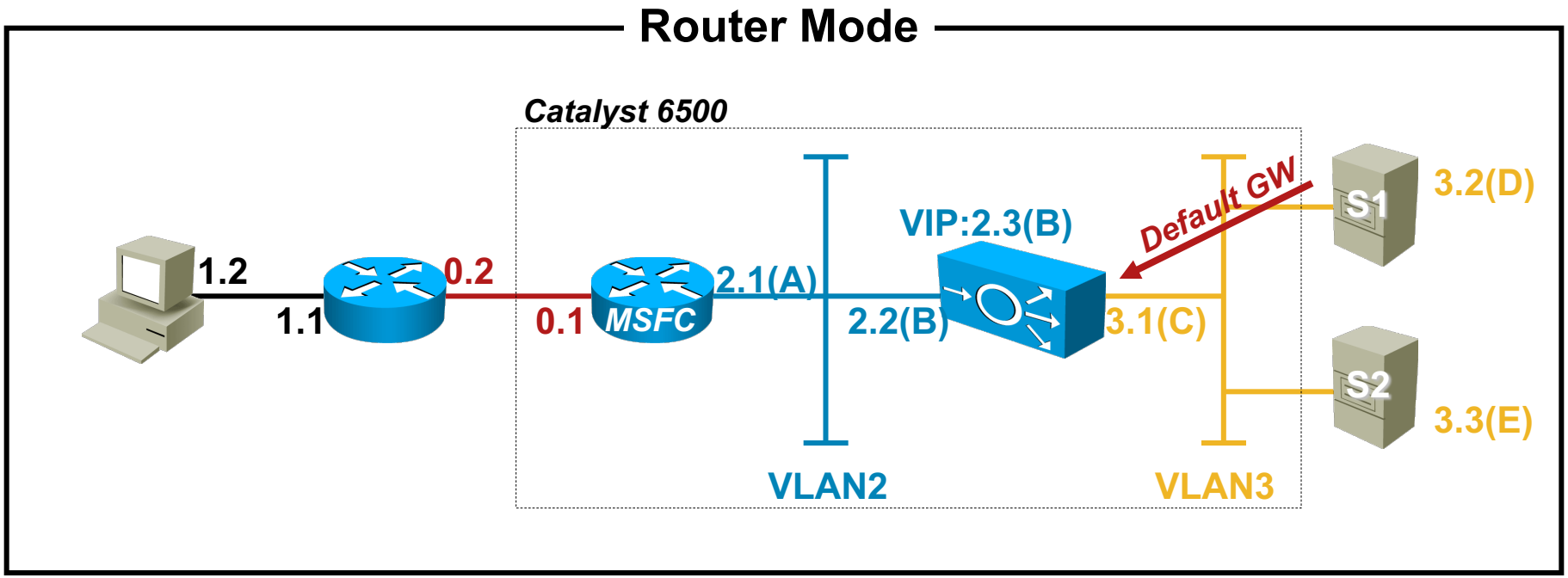
Catalyst 6500을 많이 다뤄 본 엔지니어, 또는 FWSM과 같은 Service Module을 다뤄본 분이라면 어렵지 않게 읽어 나가실 수 있을 것이고, 경험이 많으신 분들은 너무 쉽게도 느껴질 수 있을 것입니다.

아무쪼록 본 문서를 통해 Cisco의 Module형 Load Balancer와 관련된 구성을 Design하거나 Troubleshooting 하실 때 들이시는 시간을 조금이나마 줄여드릴 수 있길 바랍니다.

I. Module 형 L4의 기본 동작 Mode

1. Router Mode

Router Mode : Catalyst 6500과의 연동

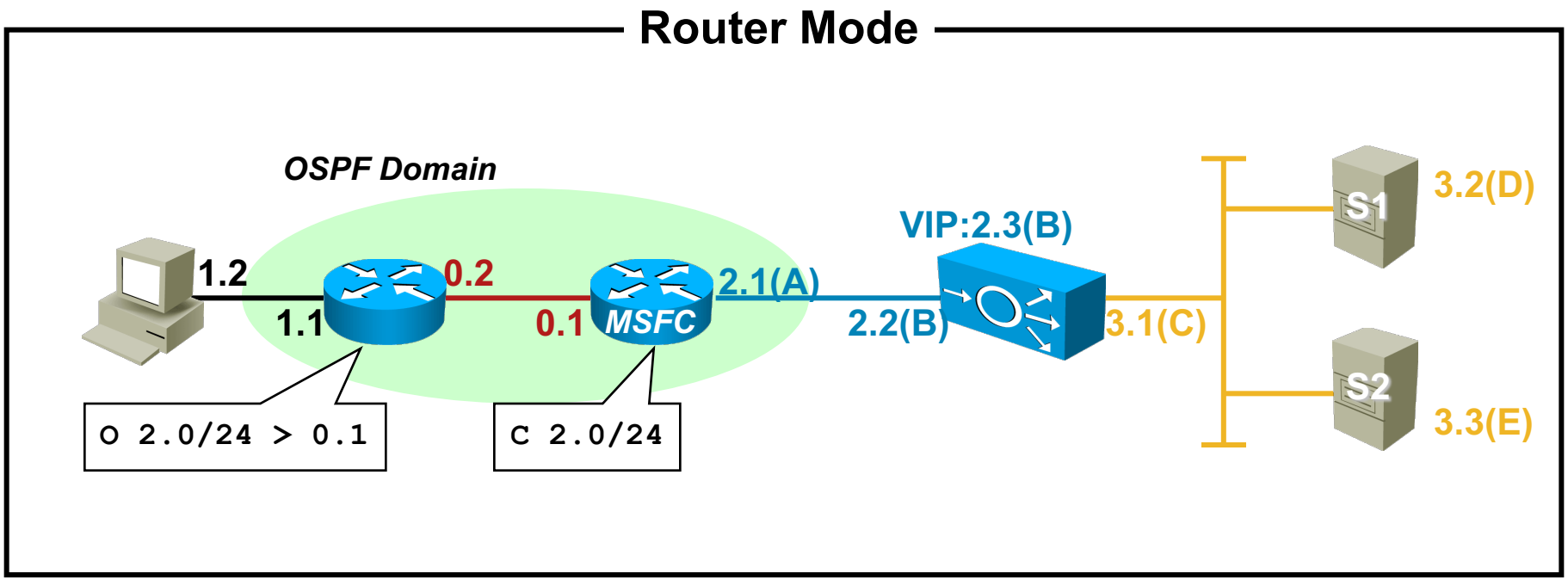


위의 구성은 가장 단순한 CSM 구성입니다. Router Mode이기 때문에 CSM 바깥쪽 IP 대역(2.0/24)과 Real Server의 IP 대역(3.x/24)을 다르게 설정 합니다.

숫자는 IP를, 괄호 안의 문자는 MAC Address를 나타내며 각각의 VLAN이 색깔 별로 표기되어 있습니다.
그림이 조금 복잡해 질 테니, 다음 페이지에서부터 VLAN2는 간단히 표현하도록 하겠습니다.

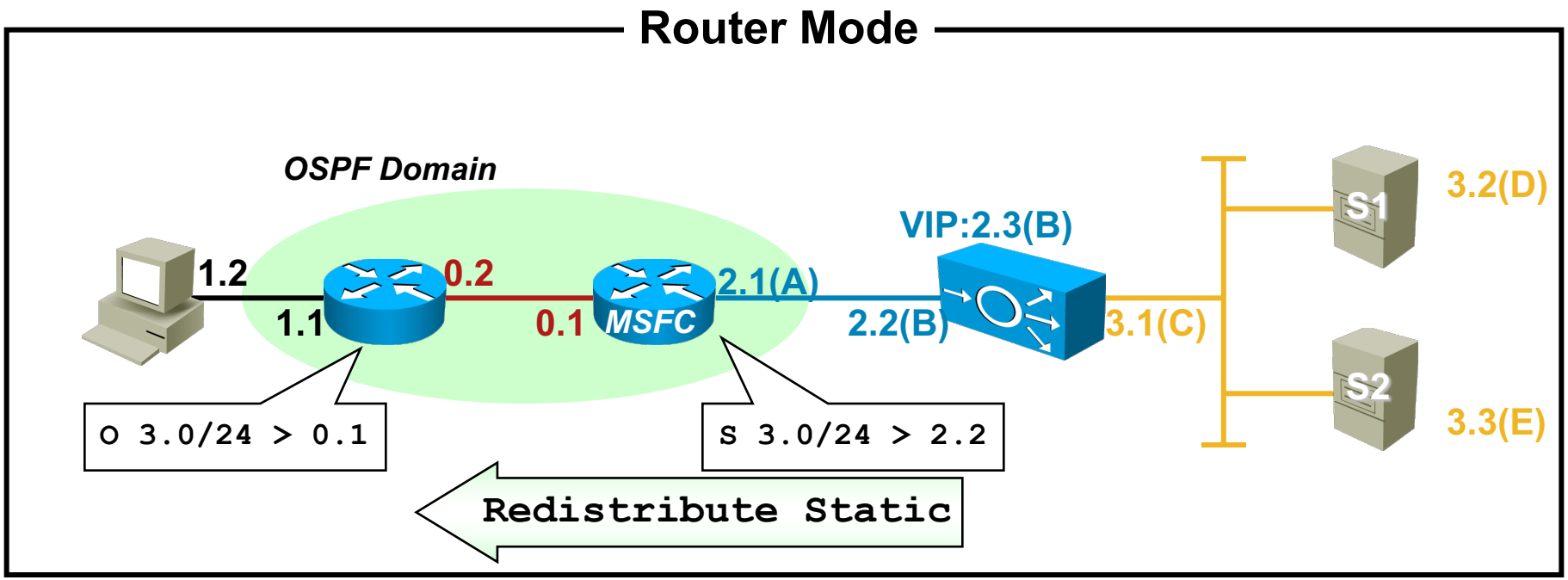
참.. L4에서 2.2와 2.3의 MAC이 왜 “B”로 같은 지 궁금하신가요? 2.3은 말 그대로 Virtual IP인데, 2.2가 설정된 Interface와 같은 곳에서 ARP 반응을 해야 하기 때문에 그렇습니다. 이해가 안되시더라도 일단 넘어 가죠.

Router Mode : Virtual IP 대역의 Routing



이 경우에는 2.0/24가 MSFC의 Connected Network이기 때문에 이를 Dynamic Routing 영역에 넣어 주기만 하면 추가적인 Static Route없이도 Client에서 VIP로 접속이 가능합니다. Redistribute connected를 사용해도 괜찮겠네요.

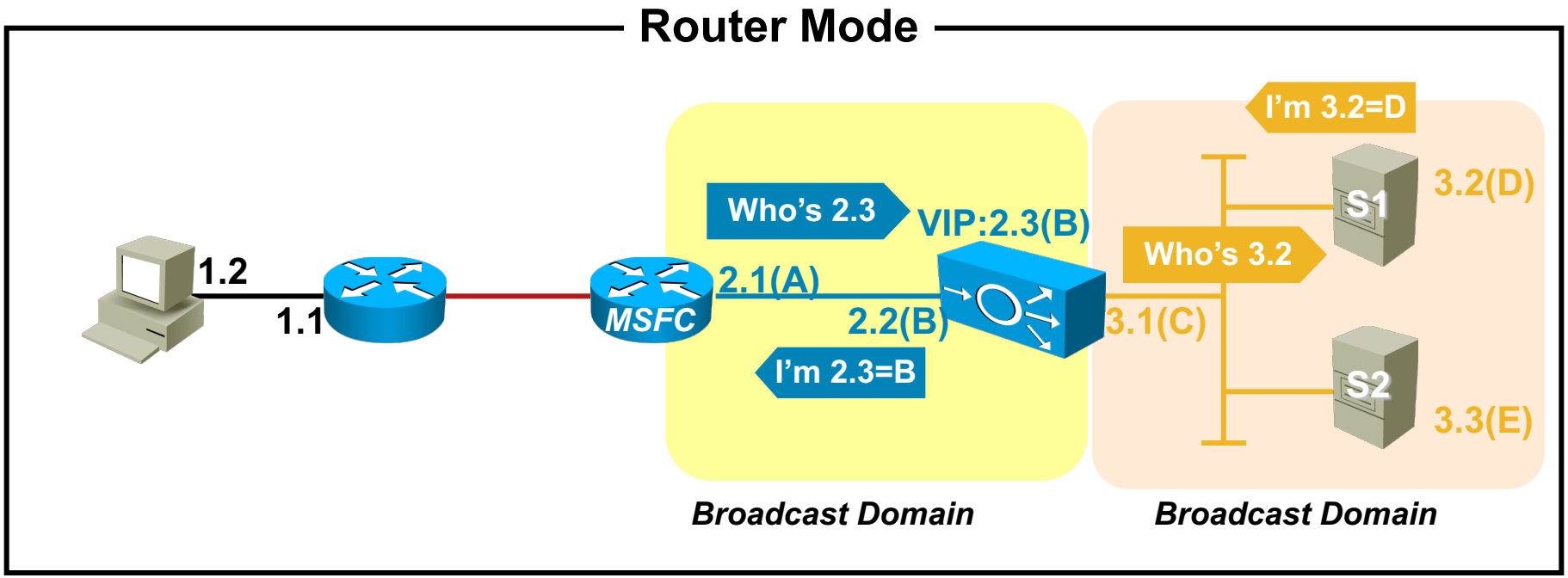
Router Mode : Rear Svr. IP 대역의 Routing



CSM을 Router Mode로 설정한 후, Real IP로 접속을 하기 위해서는 Router에서 3.0/24에 대해 CSM으로 Static Route를 설정하고 Dynamic Routing Domain으로 Redistribute를 시켜야 합니다.

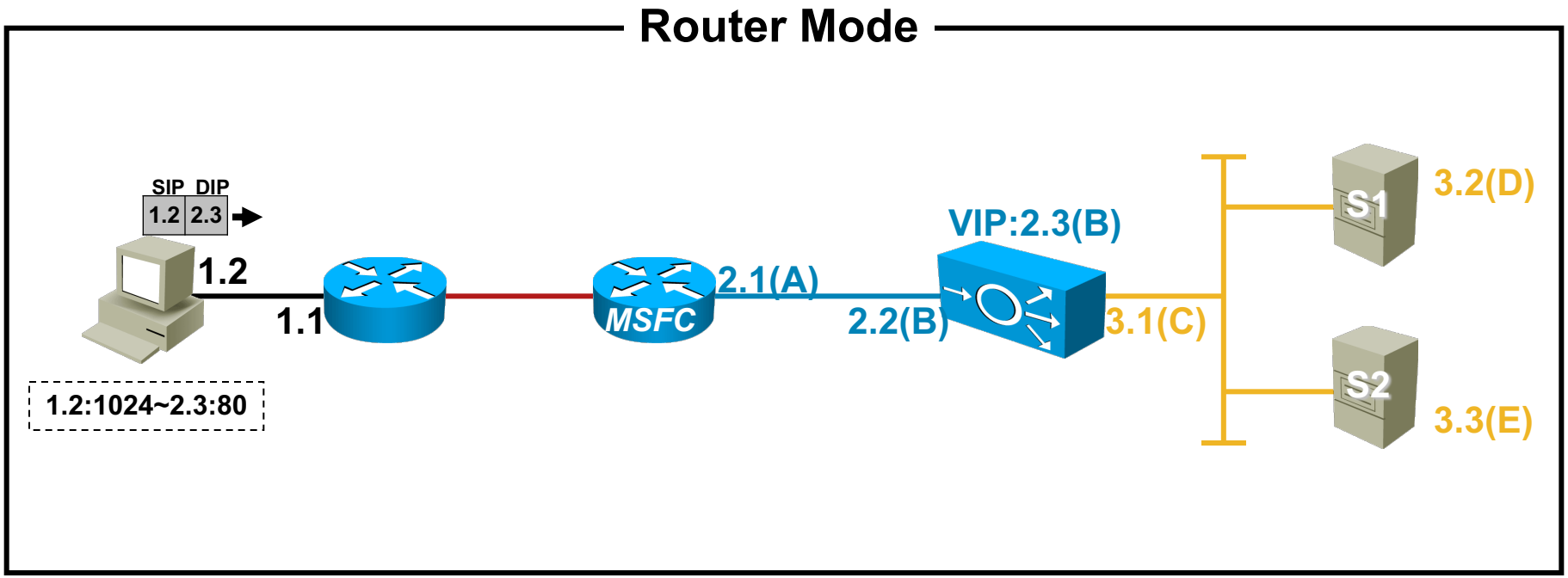
물론 Real IP로의 Direct Traffic을 허용하는 설정을 CSM에 별도로 해주긴 해야 하는데, 지금은 일단 그렇다는 정도만 알고 지나가면 됩니다.

Router Mode : ARP Mechanism



Server의 Virtual IP인 2.3에 대해서 MSFC가 ARP Request를 보내면 CSM이 2.3에 대한 ARP Reply를 보내줍니다. MSFC와 CSM은 각각의 Broadcast Domain에서 ARP를 사용하여 필요한 MAC을 찾게 됩니다.

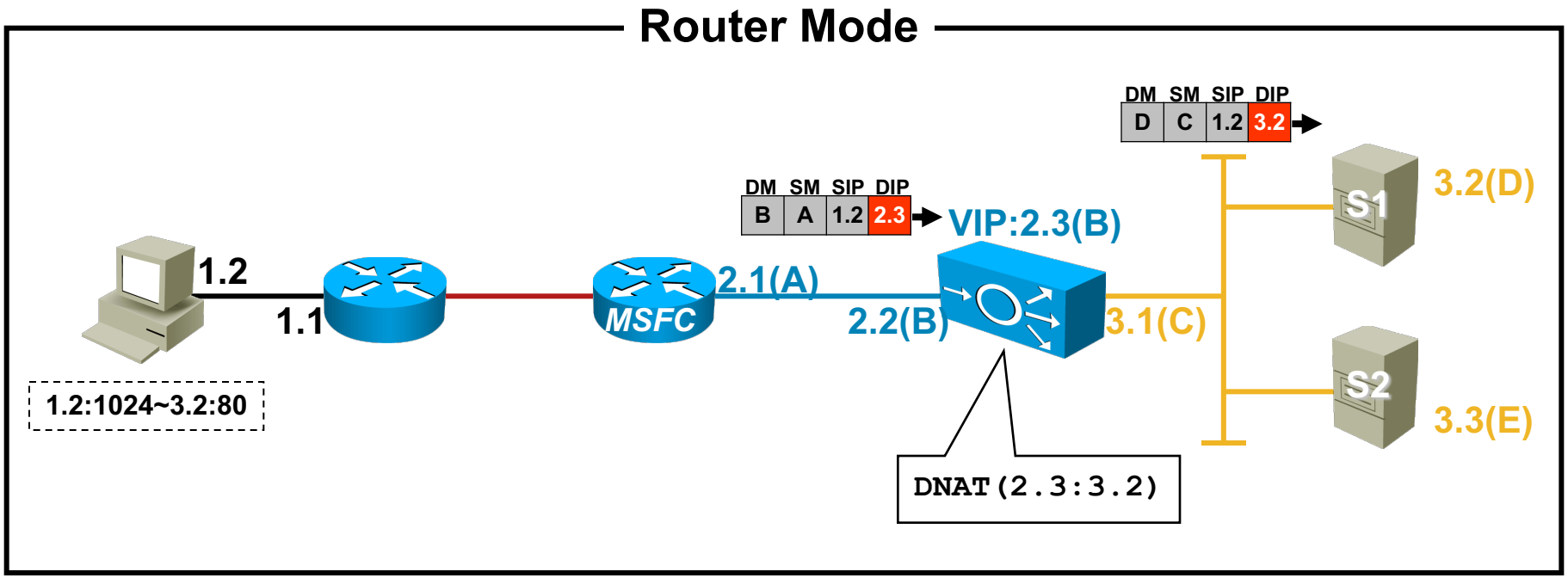
Router Mode : Load Balancing 구현 원리



Client가 VIP(2.3)에 접속을 하려고 하면 “나의 Source IP:Source Port는 무엇이고 Destination IP:Destination Port는 무엇이다”라는 정보를 생성하게 됩니다. 이것을 TCP Socket라고 부르며 내가 요청을 보낸 Destination IP가 아닌 다른 곳에서 응답을 하게 되면 그 응답은 무시하게 됩니다.

여기서 반드시 내가 요청을 보낸 Destination에서 응답이 와야 한다는 점이 중요합니다.

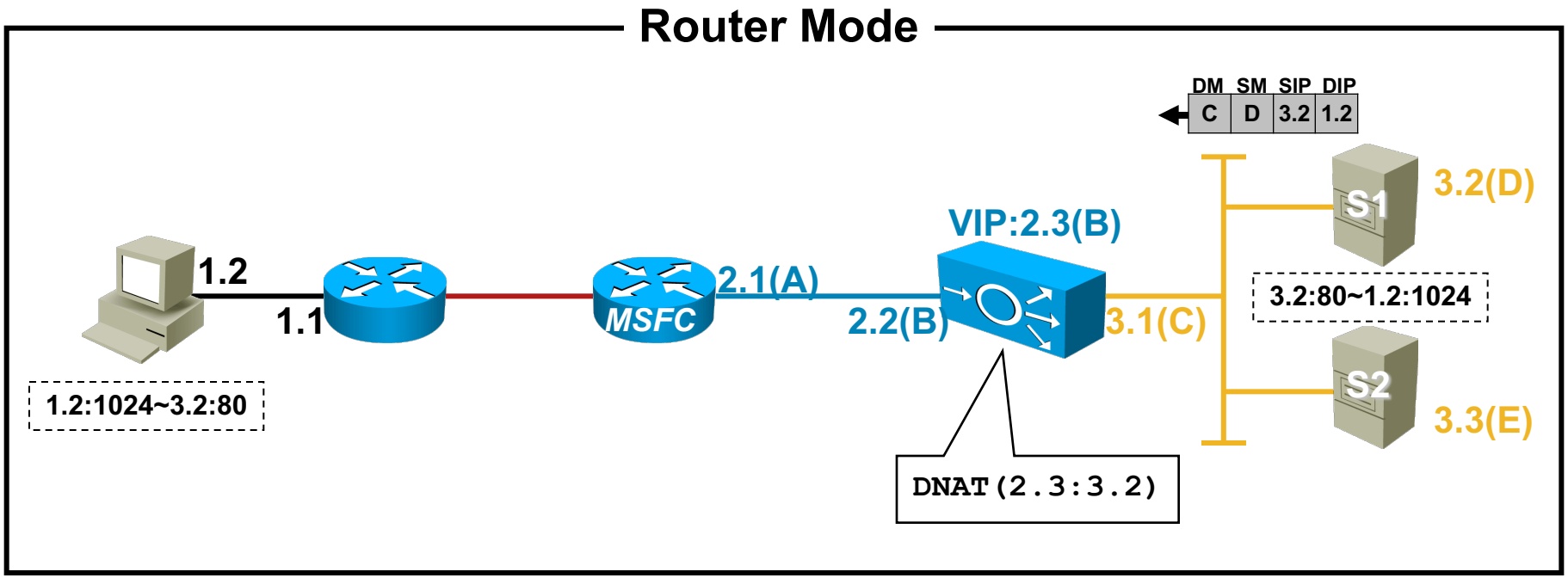
Router Mode : Load Balancing 구현 원리



이 Packet이 CSM에 도착을 하면, L4는 Destination IP Address인 2.3을 Load Balancing 메커니즘에 따라 적절히 Real Server의 IP로 NAT시키고(Destination NAT 또는 NAT Server), 해당 VLAN 쪽에 맞게끔 SMAC, DMAC을 Rewrite해서 보냅니다.

MAC을 Rewrite하는 것은 Router 역할을 하는 모든 장비의 공통점이지만 VIP를 Real IP로 바꾸는 바로 이 DNAT(혹은 NAT Server)의 원리로 인해 Load Balancing이 가능하며, CSM에서 Virtual Server를 설정하면 Default로 이 기능이 적용됩니다.("`nat server`" 명령어가 Default로 들어가 있습니다.)

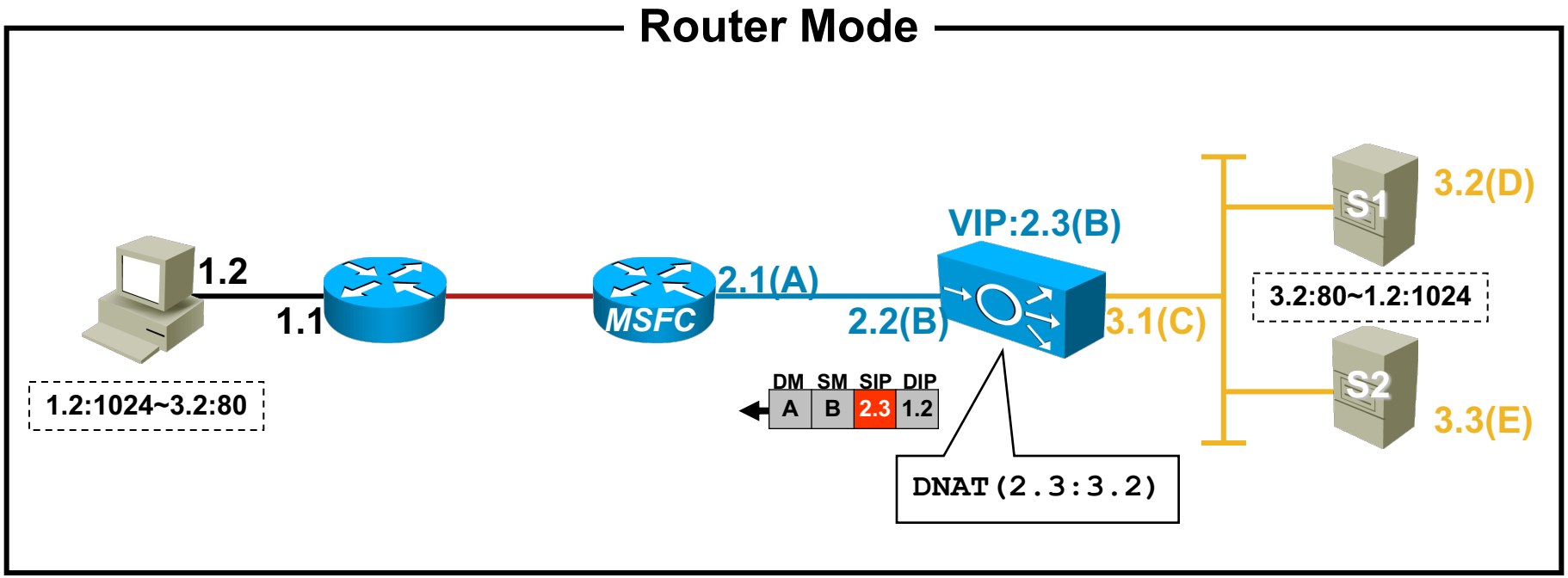
Router Mode : Load Balancing 구현 원리



Client의 요청을 받은 Real Server(S1)는 마찬가지로 자신의 TCP Socket을 생성하고 Client에게 응답을 줍니다.

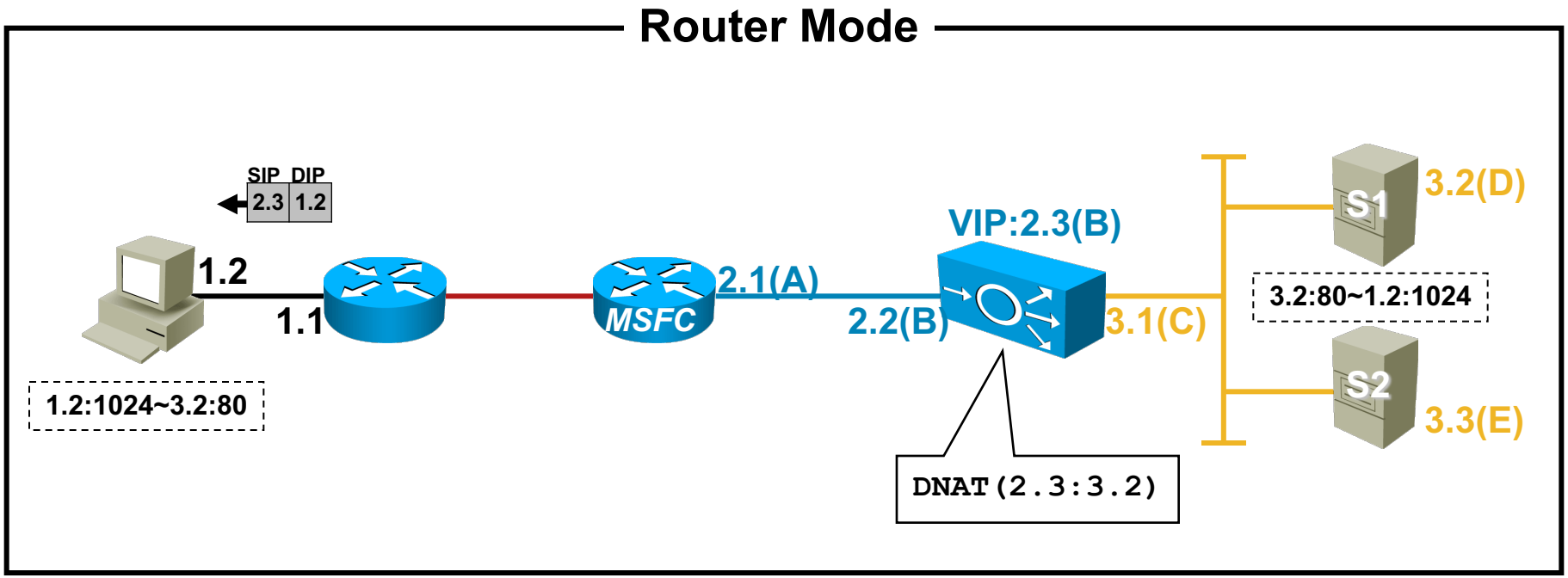
그런데, 이때, 응답 Packet을 보낼 Destination IP(1.2)가 Real Server의 IP 대역(3.x/24)과 다르므로 Real Server는 자신의 Default Gateway, 즉 CSM으로 이 패킷을 보냅니다.

Router Mode : Load Balancing 구현 원리



이 Packet을 받은 CSM은 자기가 이미 갖고 있는 NAT Table을 확인하여 변환된 IP를 다시 복구 시킨 후에 Next Hop, 즉 MSFC로 Frame을 만들어 보내 줍니다.

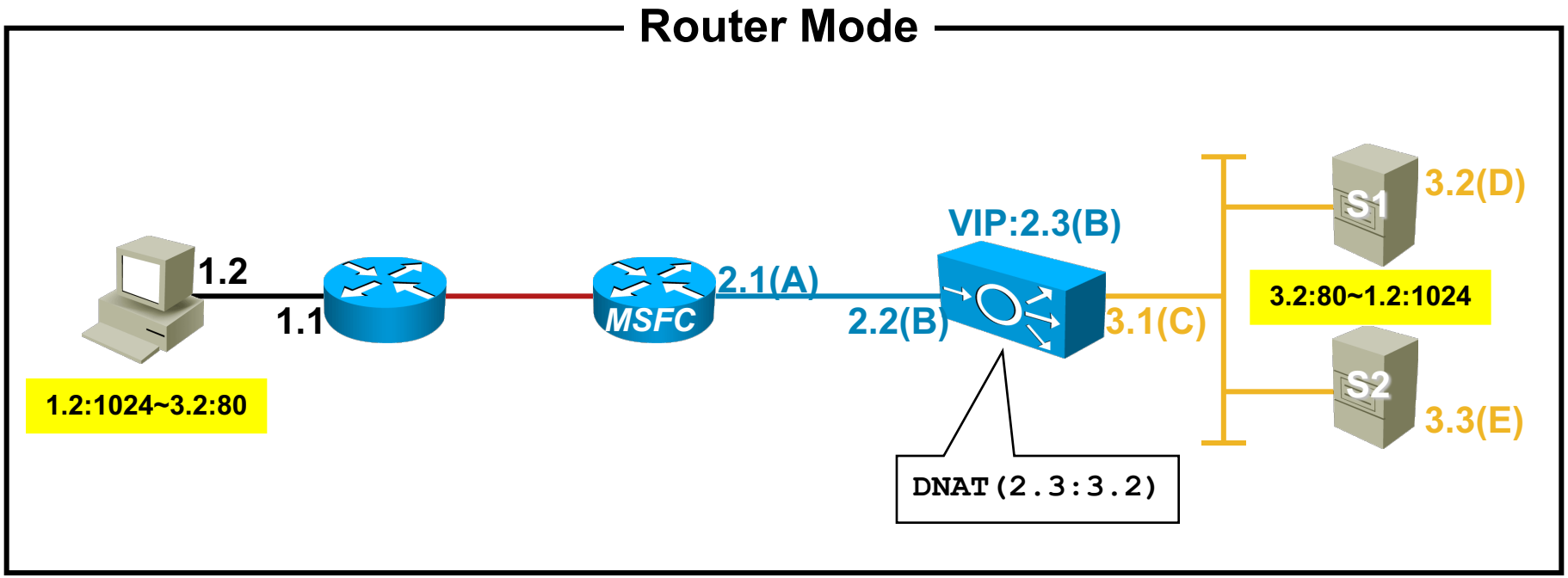
Router Mode : Load Balancing 구현 원리



이 응답을 받은 Client에게는 자신이 요청을 보냈던 바로 그 IP(2.3)에서 응답을 받은 것으로 판단 하므로 이 응답 Packet을 수락합니다.

이러한 과정을 통해 TCP의 3-Way Handshake(SYN, SYN/ACK, ACK)를 거치게 되면 Client와 Real Server 에서 완전한 TCP Session을 완성(Established)하게 되고 이 생성된 Session을 통해 Data를 주고 받게 됩니다.

Router Mode : Load Balancing 구현 원리

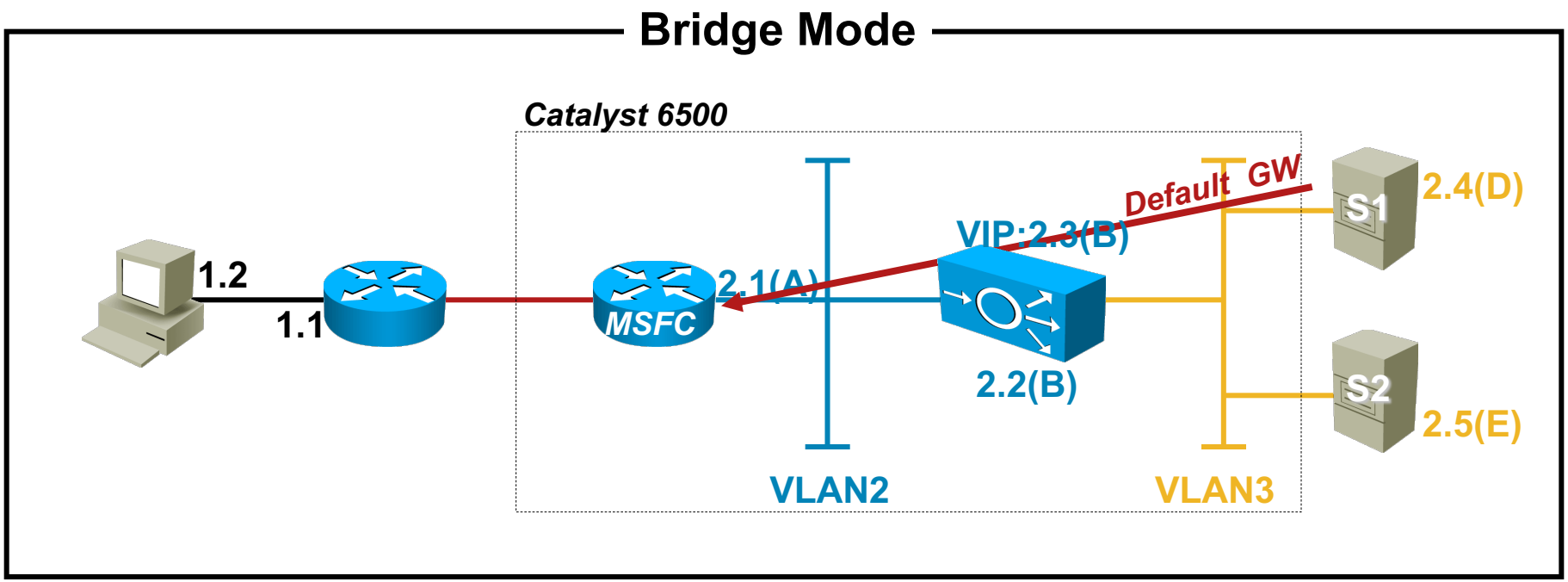


물론 CSM은 TCP Session이 생성된 후에 오고 가는 Traffic에도 계속 NAT를 수행할 뿐만 아니라,
새로운 Session을 처리하거나 오래된 Session을 삭제하는 등의 기능을 계속합니다.

지금까지는 VIP와 Real Server의 IP 대역이 다른 Router Mode에 대해 설명 드렸는데,
만약 두 IP 대역이 같다면 CSM을 어떻게 설정 해야 할까요?

2. Bridge Mode

Bridge Mode : Catalyst 6500과의 연동

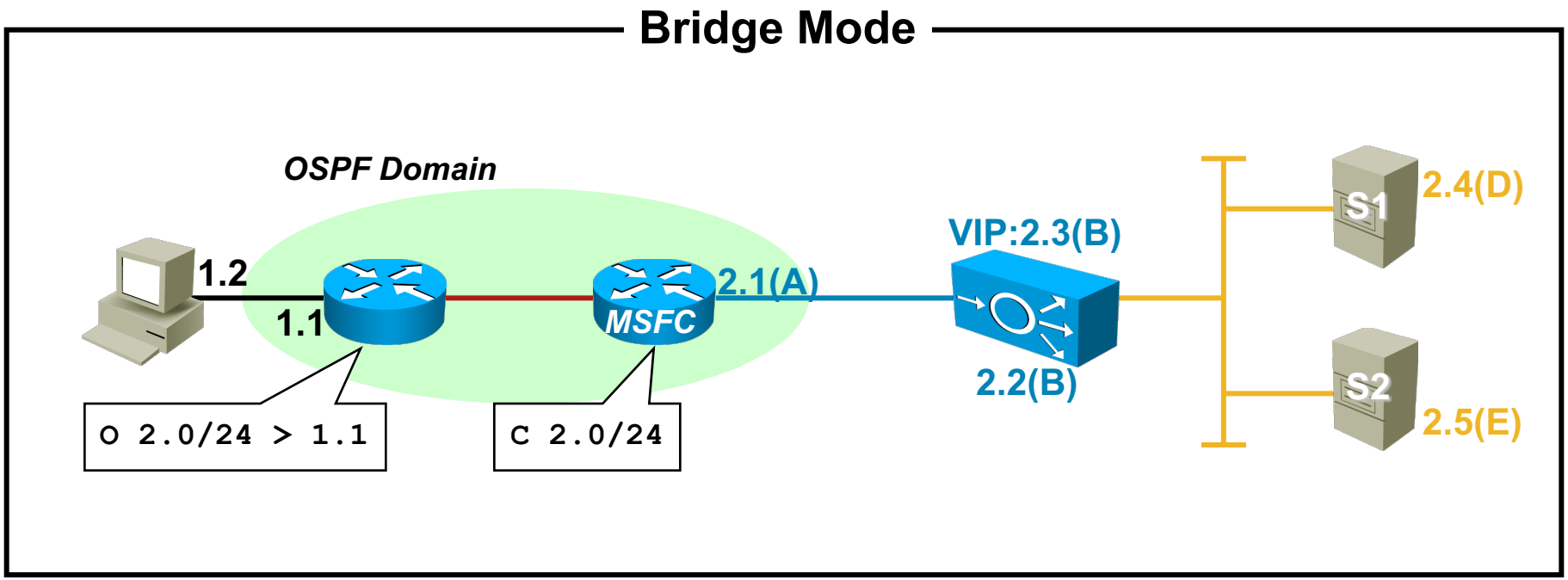


그림처럼 Real Server의 IP 대역과 VIP의 대역이 같은 경우에는 CSM을 Bridge Mode로 구현하면 됩니다. CSM이 VLAN2와 VLAN3를 Bridging하도록 하는 구조이며, Real Server들의 Default Gateway도 MSFC(2.1)가 됩니다.

CSM의 경우에는 Client VLAN 2의 IP와 Server VLAN 3의 IP를 같게 설정하면(2.2로) 자동으로 그 두 VLAN에 대해서 Bridge Mode로 동작하고,

ACE의 경우에는 별도로 Bridge Group을 생성한 후에 ACE의 두 Interface VLAN (여기서는 VLAN2와 VLAN3)을 그 Bridge Group에 넣어 주면 마찬가지로 그 두 VLAN에 대해서 Bridge Mode로 동작합니다.

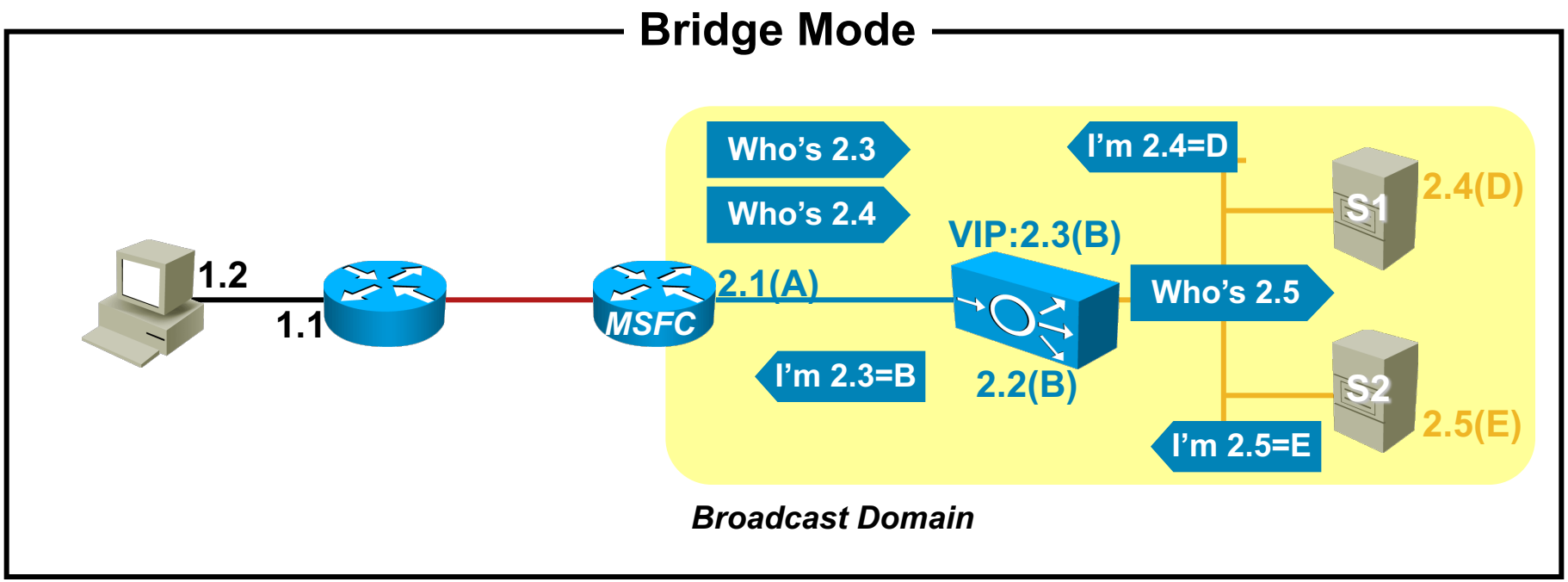
Bridge Mode :VIP&Rear Svr. IP의 Routing



Bridge Mode로 설정을 하면, Real Server의 IP 대역에 대해서도 별도의 Static Route가 필요 없어서 Dynamic Routing만으로도 VIP에 접속 하는데 문제가 없습니다.

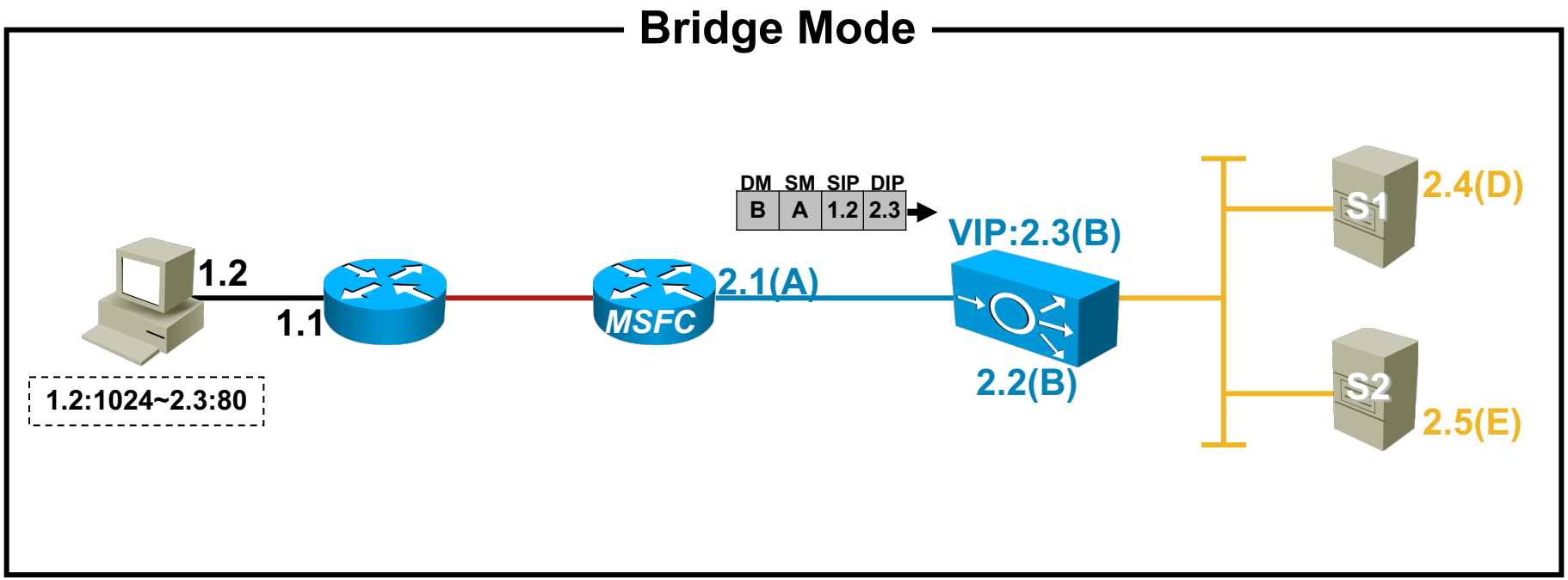
그림에 나타난 2.2 IP는 Bridge Mode를 설정하기 위한(앞에서 잠시 설명 드렸죠?)를 위한 IP일 뿐이며 Load Balancing과는 직접적인 관련이 없습니다만, 중복되는 IP는 당연히 피해서 할당을 해 줘야 합니다. L2 Switch에 Management IP를 할당하는 것 정도로 이해하면 될 것 같습니다.

Bridge Mode : ARP Mechanism



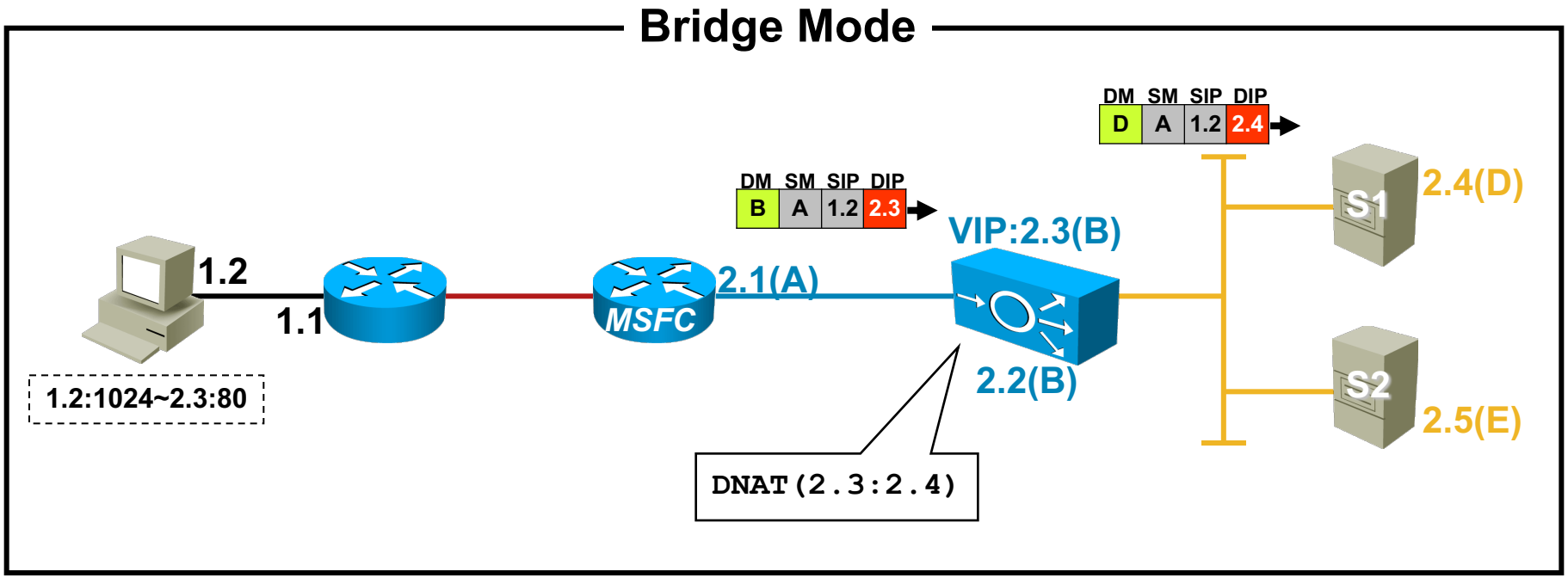
MSFC가 VIP(2.3)의 MAC Address(B)를 찾는 방식도 Router Mode와 마찬가지로 ARP Mechanism에 의한 것입니다. 물론 Real Server의 MAC Address를 찾는 것도 마찬가지구요.

Bridge Mode : Load Balancing 구현 원리



Client가 Router Mode에서 설명했던 것과 동일한 과정을 통해 Packet을 보내면, 이 Packet은 MSFC를 거쳐 CSM에 도달하게 됩니다.

Bridge Mode : Load Balancing 구현 원리

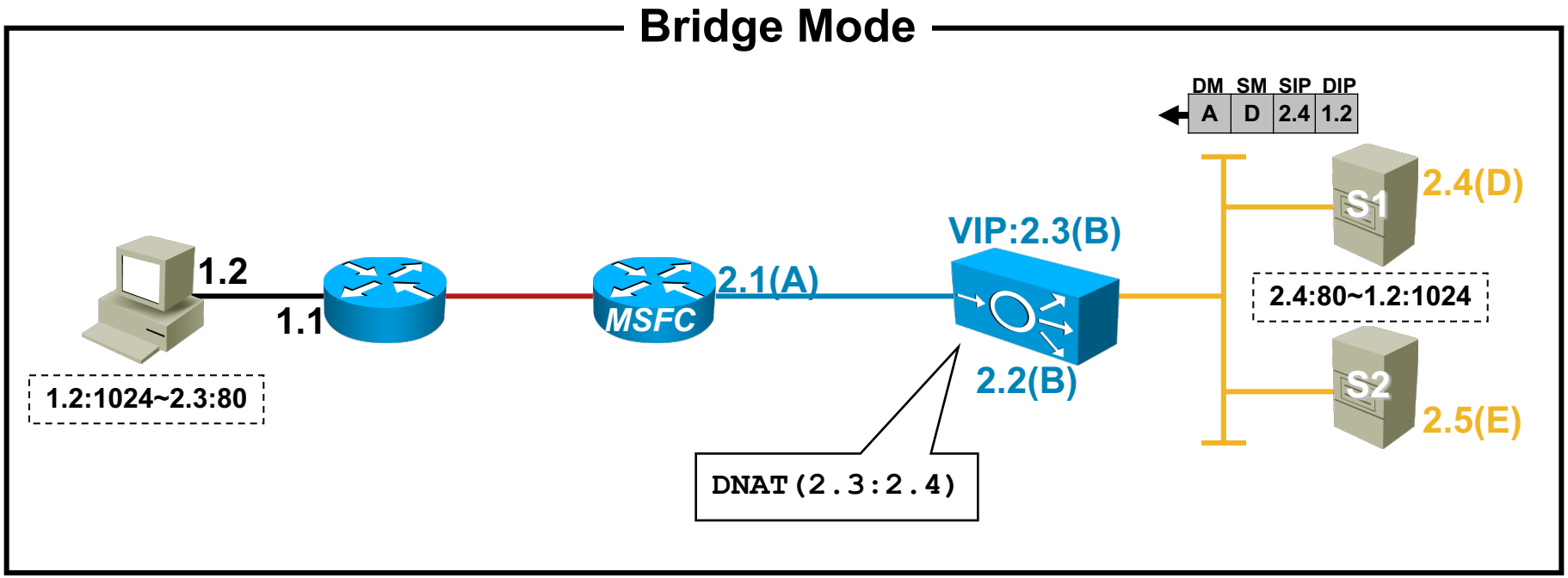


자, 이제 Load Balancing을 하기 위해서는 아까도 설명한 것처럼 Load Balancing Mechanism에 따라, Destination IP에 대해서 NAT을 해야 합니다. 그림에서도 2.30이 2.4로 바뀐 것을 알 수 있겠죠?

그런데 여기서 하나 더, Bridge Mode로 구현하면 비록 “Bridge”라고 하더라도 Destination MAC Address를 CSM에서 Rewrite합니다. 이 때 사용되는 MAC Address는 Load Balancing Mechanism에 따라, Server1으로 가야 하면 “D”를, Server2로 가야 하면 “E”를 사용하는 식으로 선택됩니다.

그림에서 왼쪽에 있는 Frame의 DMAC이 “B” 였는데, CSM을 지난 후 “D”로 바뀐 것을 볼 수 있습니다.

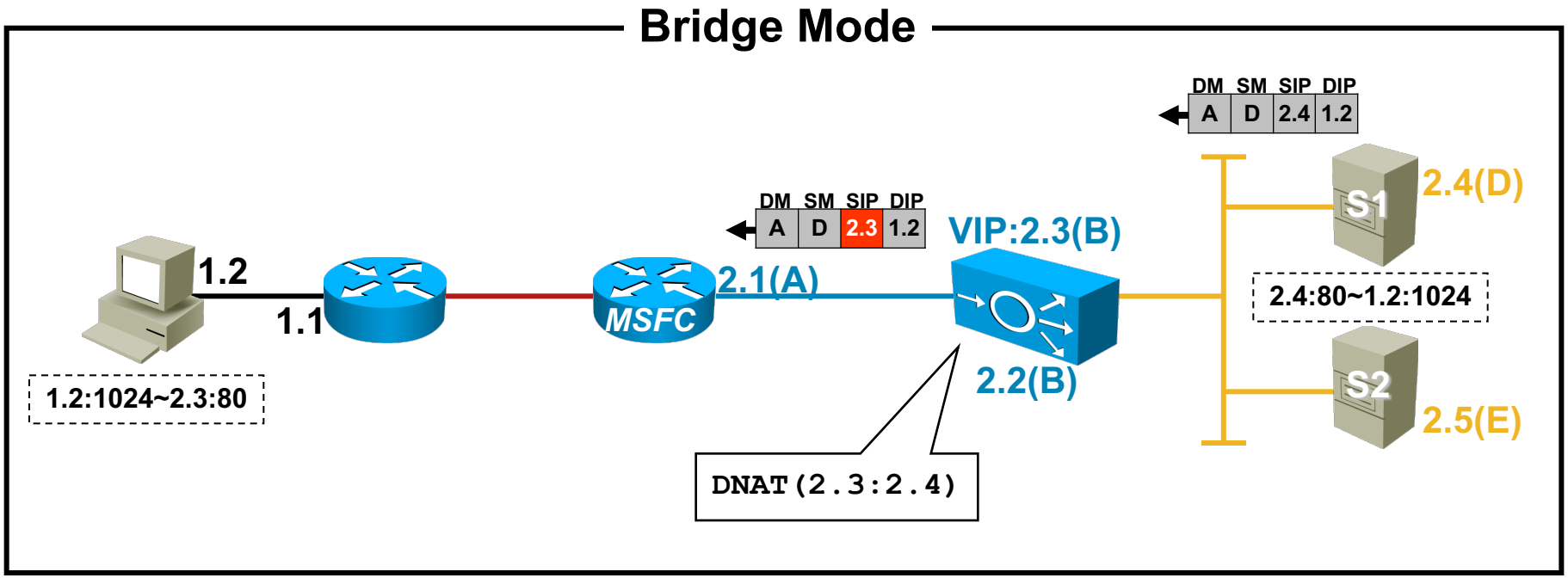
Bridge Mode : Load Balancing 구현 원리



자, 이제 Real Server가 응답을 할 차례입니다. Real Server는 자기 IP(2.4)와 다른 대역에 있는 1.2로 Packet을 보내야 하므로 자기 자신의 Default Gateway인 MSFC를 향해 Frame을 만들어 보낼 것입니다.

당연히 MSFC의 IP에 Mapping된 MAC Address(A)로 Frame을 만들어 보내겠죠?

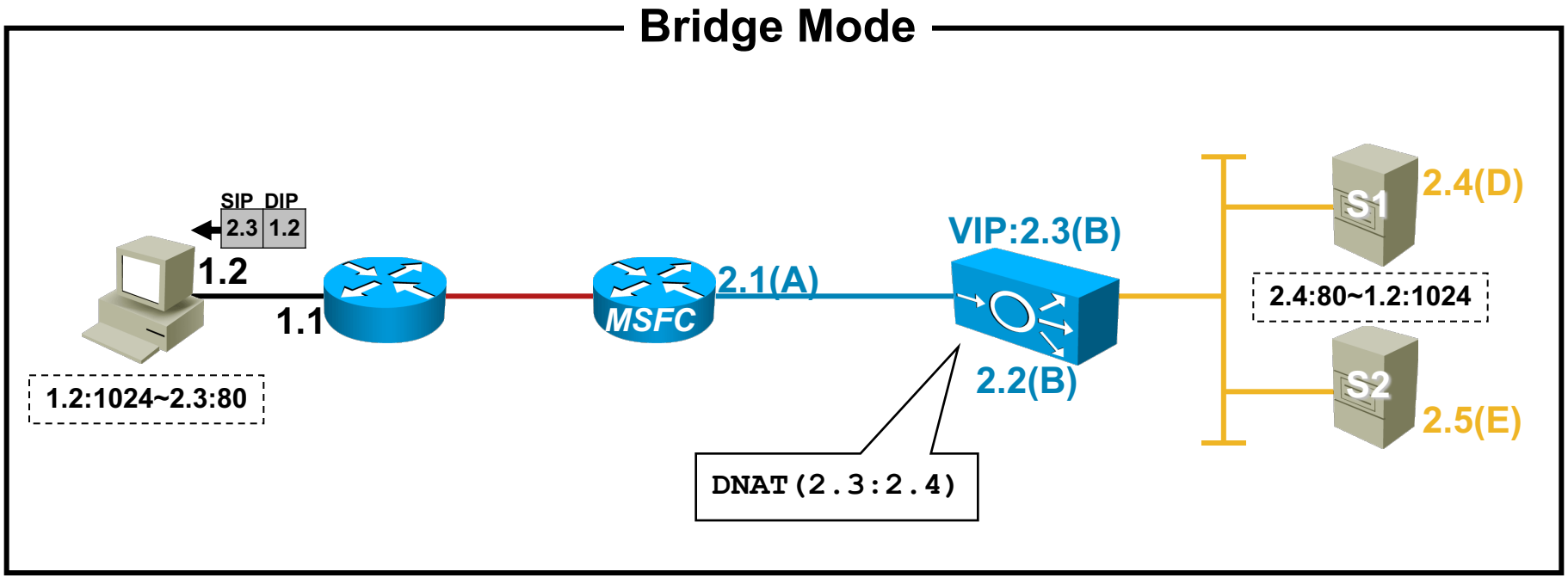
Bridge Mode : Load Balancing 구현 원리



그림에서 보이듯이 CSM이 비록 Bridge Mode로 구성되어 있더라도 Traffic의 흐름 상 Inline 상에 위치 하기 때문에 이 응답 Frame은 CSM을 통과하게 됩니다.

따라서 Real Server가 MSFC로 Frame을 보내더라도 CSM이 가운데서 이 응답 Traffic을 감지하고, 다시 원래 요청 받았던 IP(2.3)로 변환을 합니다.

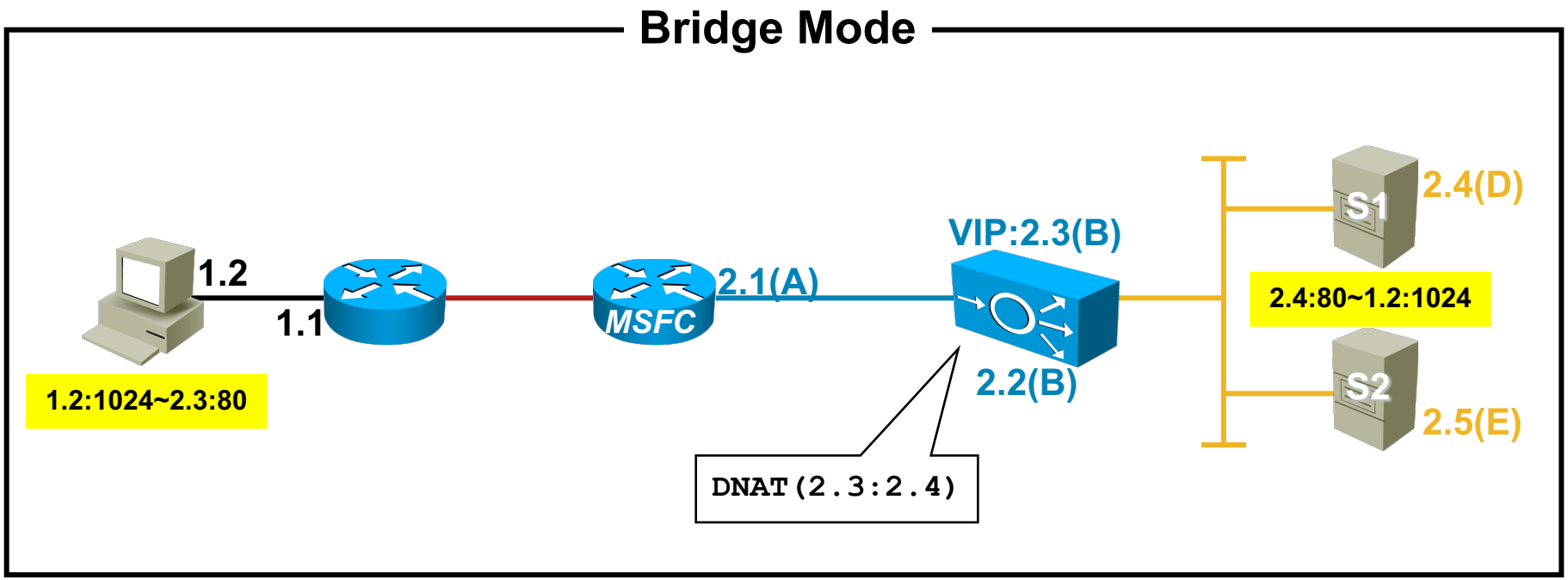
Bridge Mode : Load Balancing 구현 원리



이 응답을 받은 Client에게는 자신이 요청을 보냈던 바로 그 IP(2.3)에서 응답을 받은 것으로 판단 하므로 이 응답 Packet을 수락합니다.

이러한 과정을 통해 TCP의 3-Way Handshake(SYN, SYN/ACK, ACK)를 거치게 되면 Client와 Real Server 에서 완벽한 TCP Socket을 완성(Established)하게 되고 이 생성된 Session을 통해 Data를 주고 받게 됩니다.

Bridge Mode : Load Balancing 구현 원리



물론 TCP Session이 생성된 후에 지나가는 Traffic에도 L4는 계속 Destination NAT를 할 뿐만 아니라, 새로운 Session을 처리하거나 오래된 Session을 삭제하는 등의 기능을 계속합니다.

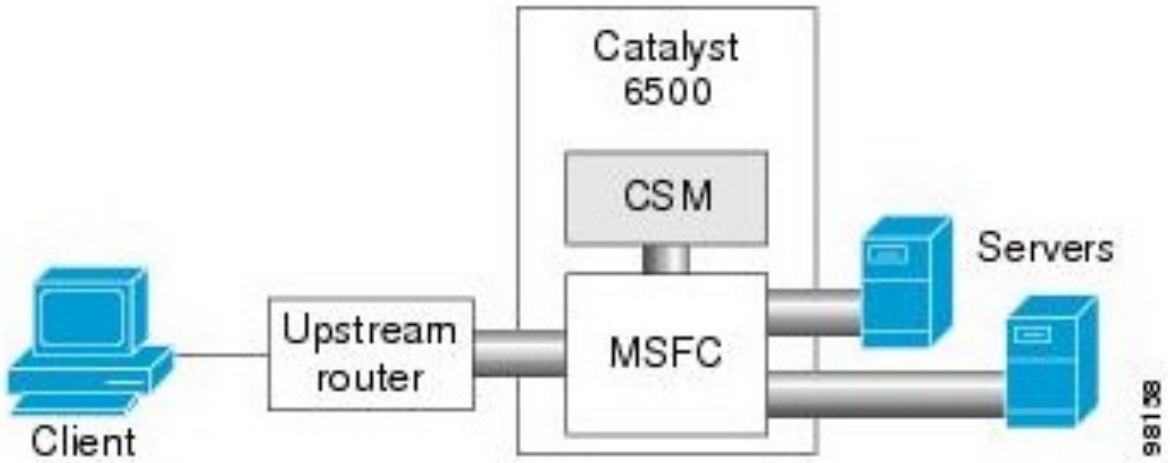
자, 여기까지 문제 없이 이해하신 분들은 아마 CSM을 한번 이상 설치를 해 보신 분들일 겁니다.

그럼 지금 부터는 이 기본 개념을 응용한 두 가지 기법을 더 알아보도록 하겠습니다.

II. Module 형 L4의 응용 동작 Mode

1. One-Arm Mode

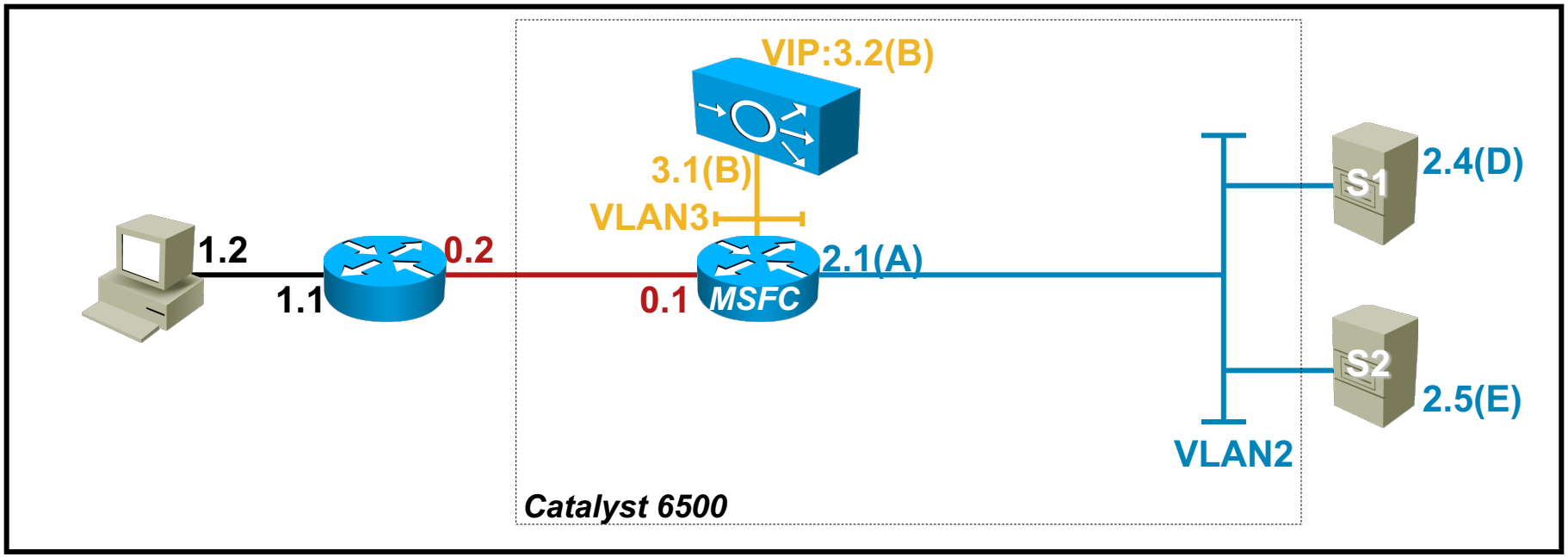
One-Arm Mode : 개요



One-Arm Mode는 문자 그대로 CSM의 VLAN이 한쪽만 설정이 되어 있는 것을 뜻합니다. 이 경우에 CSM은 Load-Balancing할 Traffic만을 처리하게 되며, 나머지 Traffic은 CSM과 전혀 무관하게 처리됩니다.

One-Arm Mode는 Documentation에 따라 Aggregate Mode라고 부르기도 합니다.

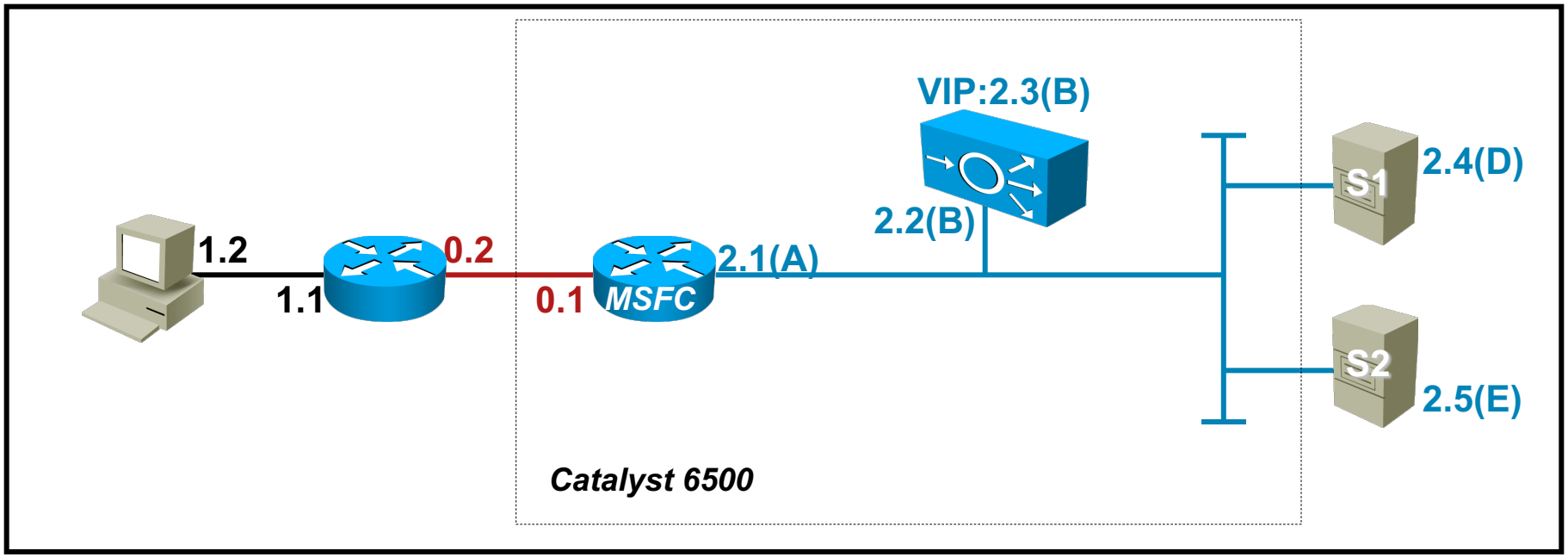
One-Arm Mode : VIP와 Real IP 대역이 다른 경우 Catalyst 6500과의 연동



One-Arm Mode를 구성하는 방식은 두 가지가 있는데, 위의 그림처럼 VIP가 Real Server IP와 다른 대역에 있을 수도 있고, 다음 페이지처럼 같은 대역에 위치할 수도 있습니다.

이 의미가 앞에서 설명한 Router Mode, Bridge Mode와 비슷하게 들릴 수도 있는데, 사실은 전혀 다른 개념입니다.

One-Arm Mode : VIP와 Real IP 대역이 같은 경우 Catalyst 6500과의 연동



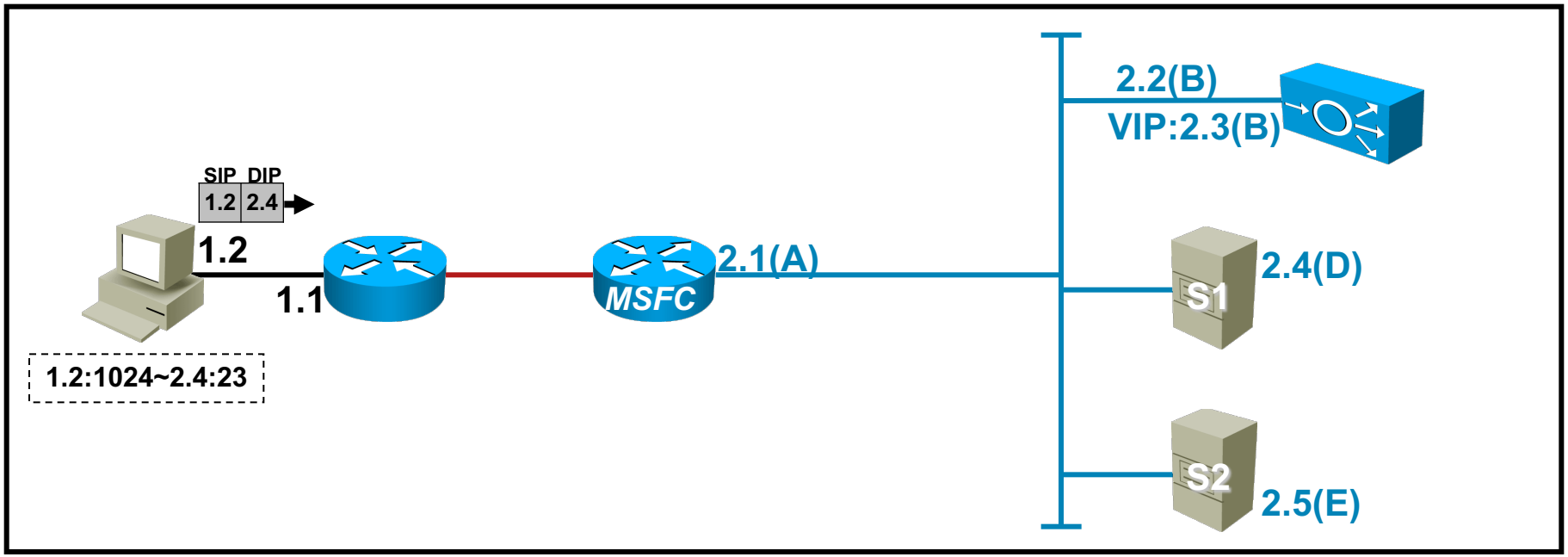
이런 구성이라면 일반적으로 VIP와 Real Server IP를 같은 대역으로 하는 경우겠죠.

어? 이거 쉬운 거 아냐?라고 하실 수도 있는데, 실은 그렇질 않습니다. 이제 그 이유를 설명할 텐데, 편의상 이 페이지에 있는 VIP와 Real IP 대역이 같은 경우를 예로 들어 설명을 드리겠습니다.

앞의 구성으로 하더라도 원리는 동일하니, 나중에 시간을 내서 한번 연구해 보시기 바랍니다.

참.. 그 전에, 여러분의 이해를 돕기 위해서 CSM의 위치를 약간 바꿔 놓는 것이 좋겠네요.

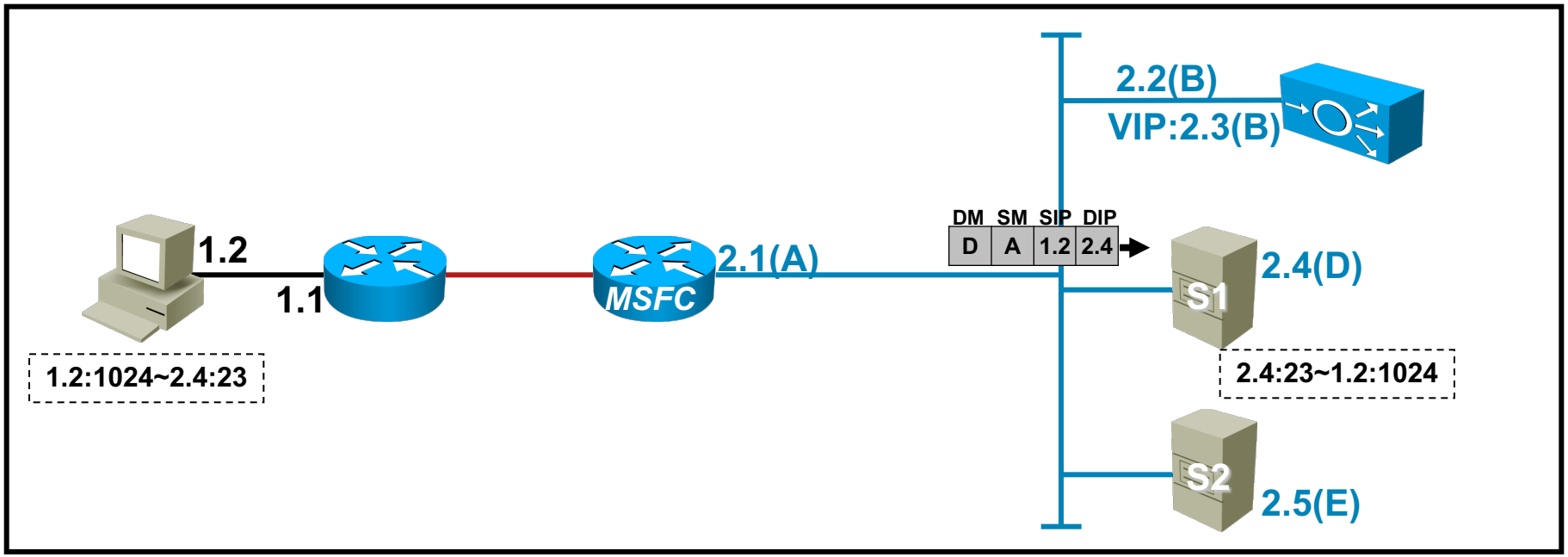
One-Arm Mode : 일반적인 Traffic의 Flow



조금 쉽게 보이나요? 사실 앞의 그림과 똑 같은 그림입니다.

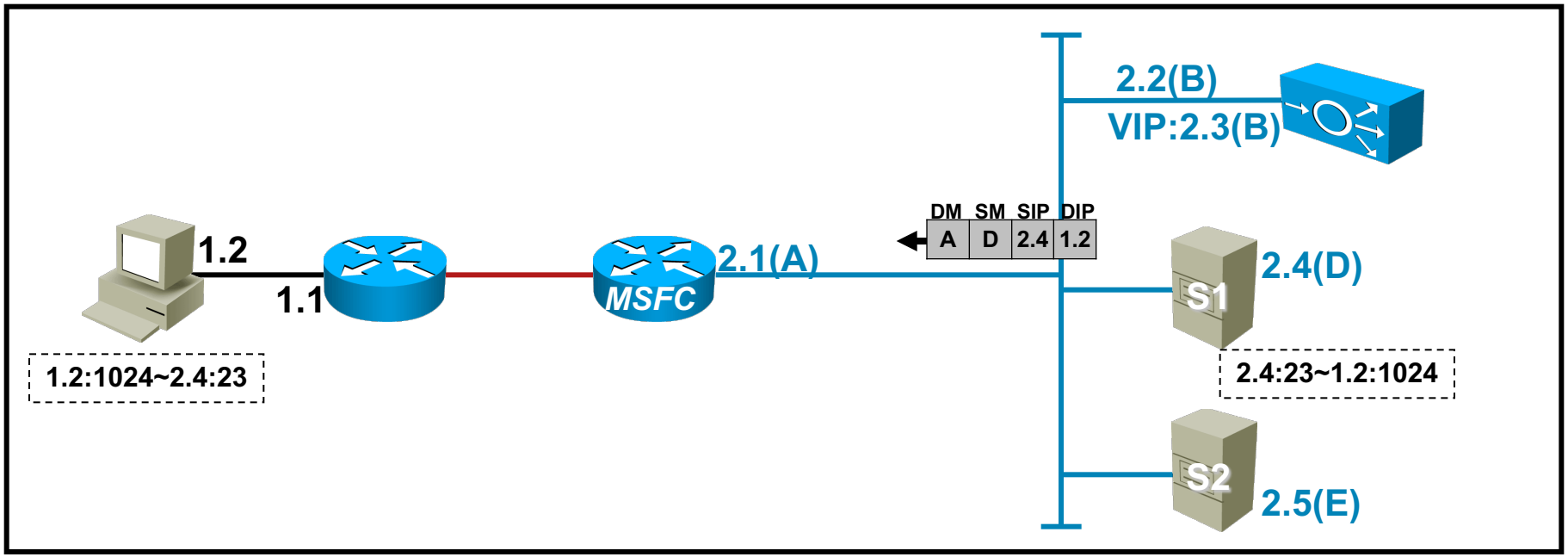
먼저 Client가 Real Server(2.4)로 직접 접속을 한다고 가정해 보겠습니다. 아마 관리자가 Server에 Telnet으로 접속해서 뭔가 작업을 하고 싶은 모양입니다.

One-Arm Mode : 일반적인 Traffic의 Flow



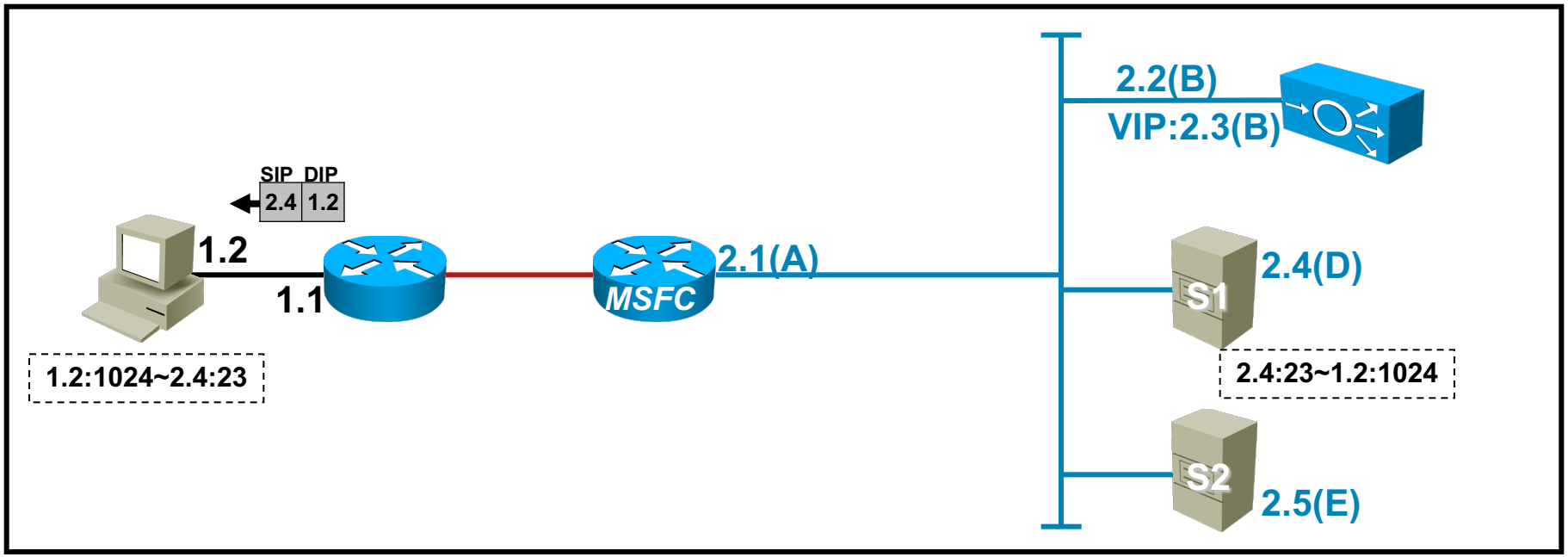
MSFC는 Real Server 2.4에 대한 MAC을 알고 있으므로 바로 Real Server로 Frame을 만들어 보냅니다.
이 Traffic은 CSM과 전혀 무관하므로 바로 Switching Path를 통해 Real Server로 전달됩니다.

One-Arm Mode : 일반적인 Traffic의 Flow



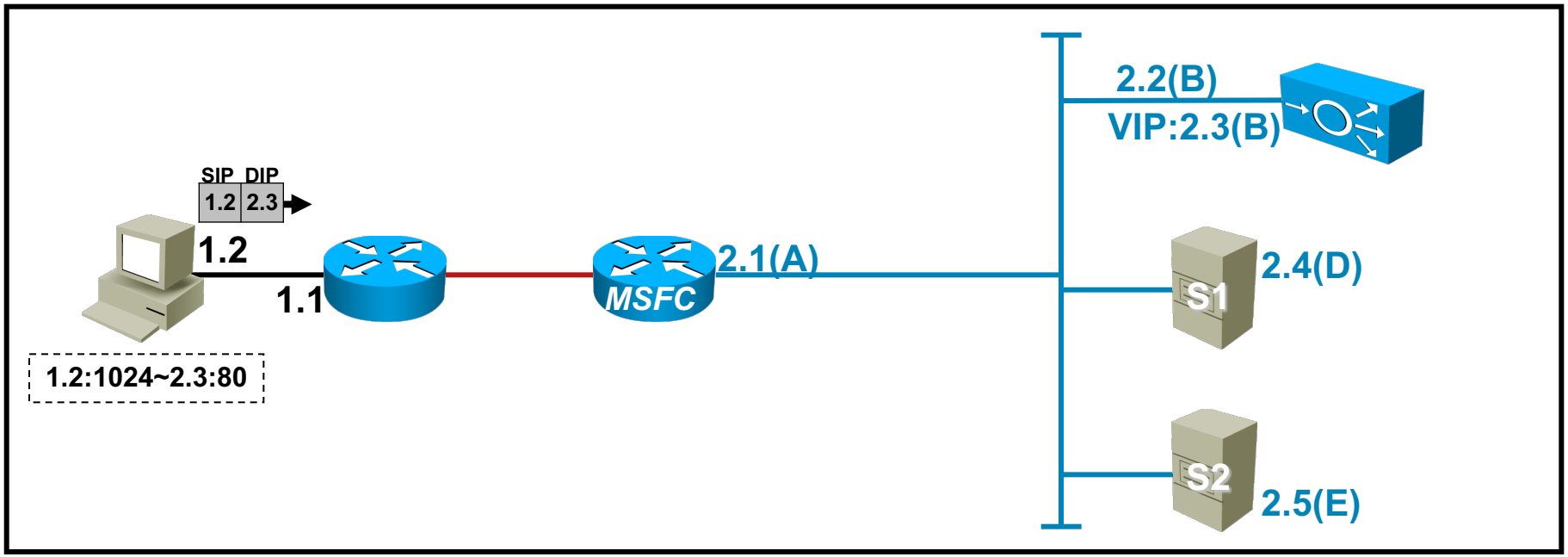
Real Server 2.4도 마찬가지로 자신의 Default Gateway와 직접 통신할 수 있으므로 이 Traffic은 CSM을 통과하지 않고 바로 MSFC로 보내집니다.

One-Arm Mode : 일반적인 Traffic의 Flow



Client는 자기가 요청을 보냈던 IP인 2.4에게서 정상적으로 응답을 받았다고 판단하고 TCP 3-Way Handshake 과정을 계속 진행합니다.

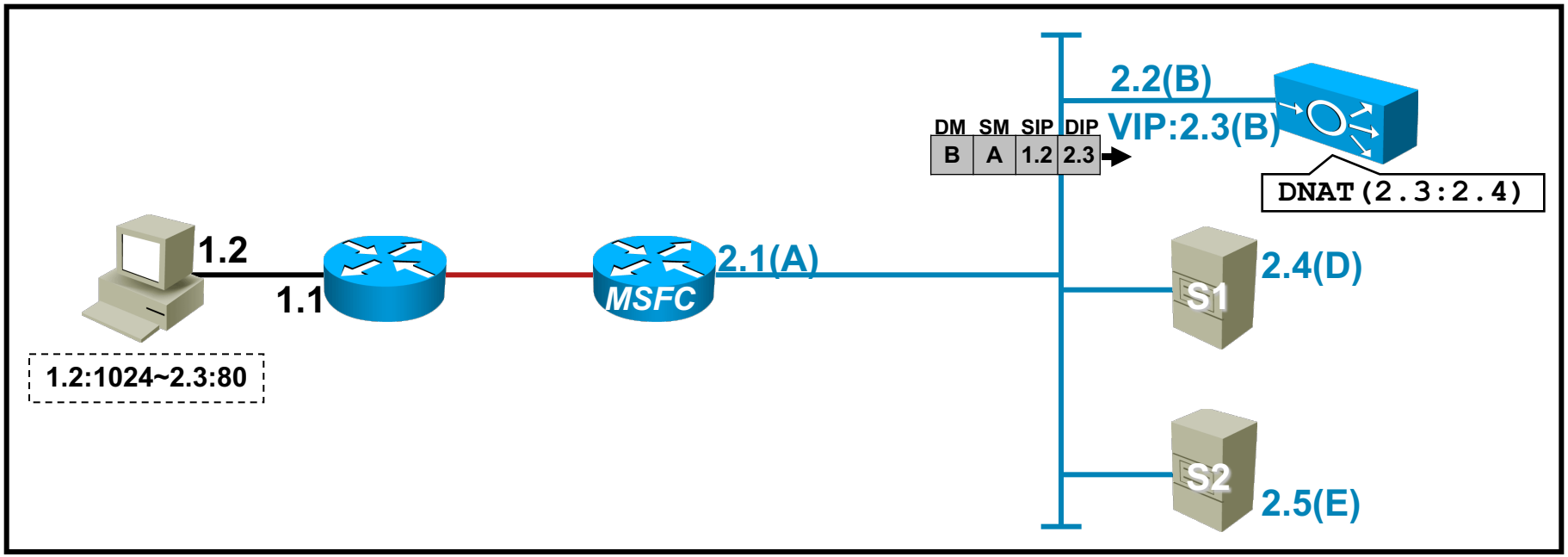
One-Arm Mode : VIP로 접속할 때의 Issue



이번에는 Client가 VIP로 HTTP 접속 하는 경우를 생각해 보겠습니다. 자꾸 반복되는 Client 부분은 설명을 생략하고 싶은데, 여기서는 Client의 입장이 매우 중요해서 다시 한번 설명을 드리려고 합니다.

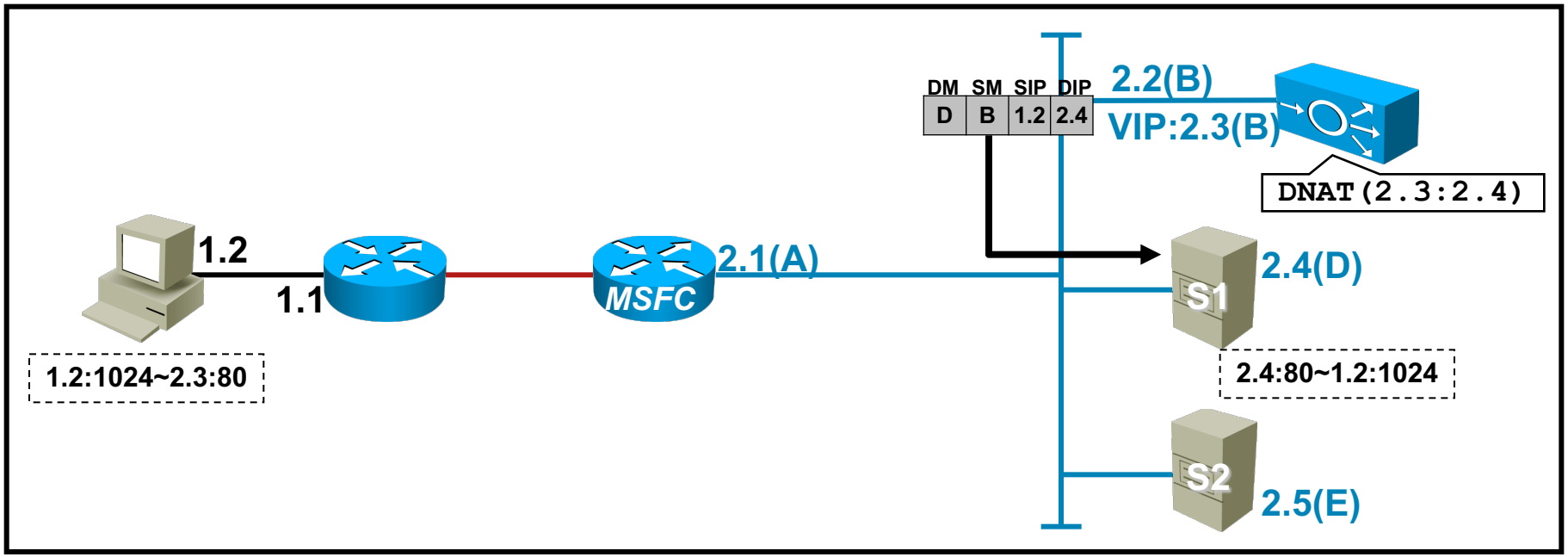
먼저, Client가 앞의 경우와 동일한 방식을 따라 요청 Packet을 보냅니다.

One-Arm Mode : VIP로 접속할 때의 Issue



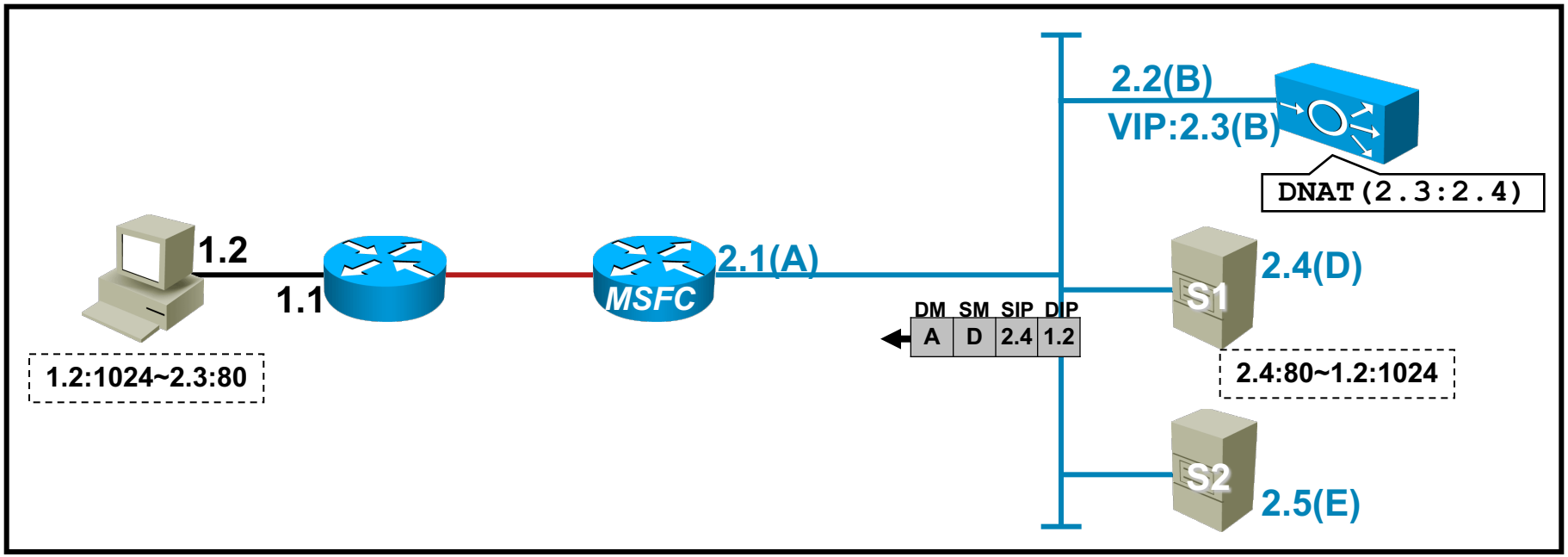
자, MSFC는 Destination IP가 2.3인 Traffic을 CSM에게 보냅니다.

One-Arm Mode : VIP로 접속할 때의 Issue



CSM은 Load Balancing을 하기 위해 Connection Table을 만들고 Destination NAT를 합니다.
물론 처리된 Traffic을 Forward하기 위해 적절히 Real Server의 MAC(D)을 이용해서 MAC Rewriting도 해야겠군요.

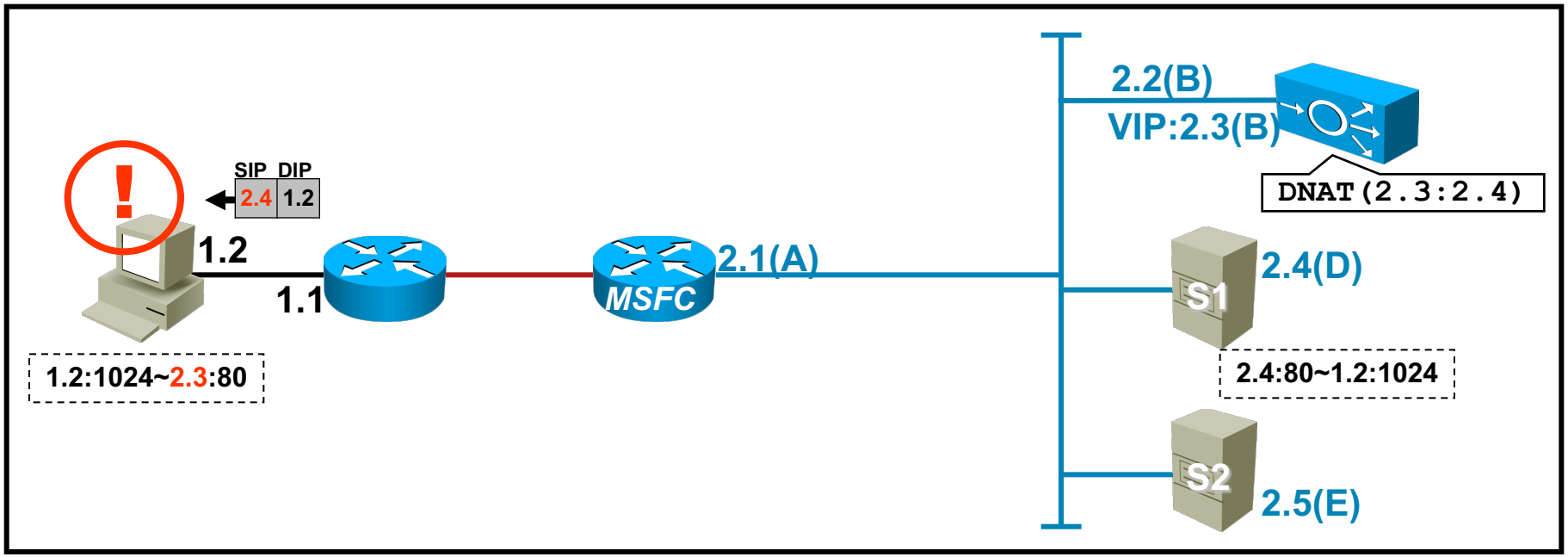
One-Arm Mode : VIP로 접속할 때의 Issue



자, 이제는 Real Server가 응답을 할 차례입니다.

그런데 Real Server는 자기가 응답을 줘야 할 Destination IP(1.2)가 자기가 속한 Network(2.0/24)이 아닌 것을 보고 자기의 Default Gateway, 즉 MSFC로 Frame을 만들어 보냅니다.

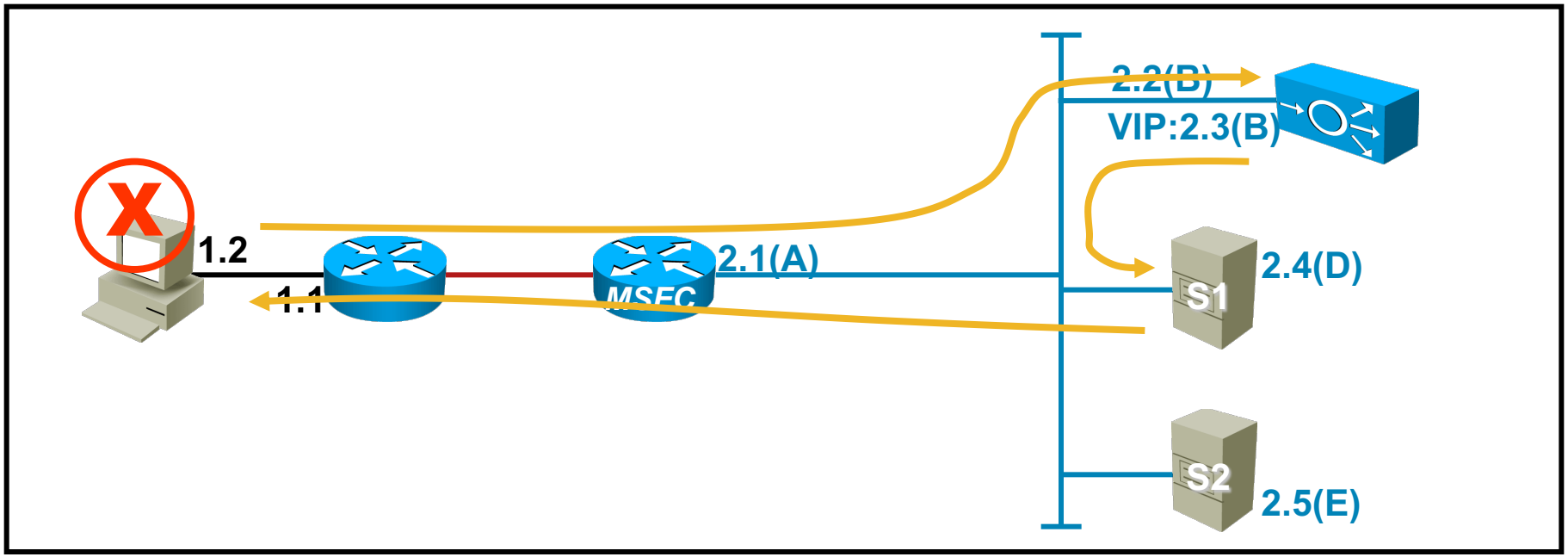
One-Arm Mode : VIP로 접속할 때의 Issue



자, 이제 응답 Packet이 Client에 도달 했습니다. 앗! 그런데, Packet의 Header를 곰곰이 보니 나에게 Packet을 보내준 IP(2.4)가 내가 요청한 IP(2.3)와 다르죠? 이렇게 되면 Client는 내가 SYN을 보내지 않은 곳으로부터 SYN/ACK를 받은 셈이므로 정상적으로 3-Way Handshake를 할 수 없습니다.

어디서 문제가 발생했나 다시 한번 자세히 살펴 보니, 요청 Packet에 대해서는 DNAT이 이뤄졌고 응답 Packet에 대해서는 아무런 처리가 되지 않고 Client에 도달을 했군요.

CSM으로 들어간 Traffic은 반드시 CSM을 통해 나와야 함

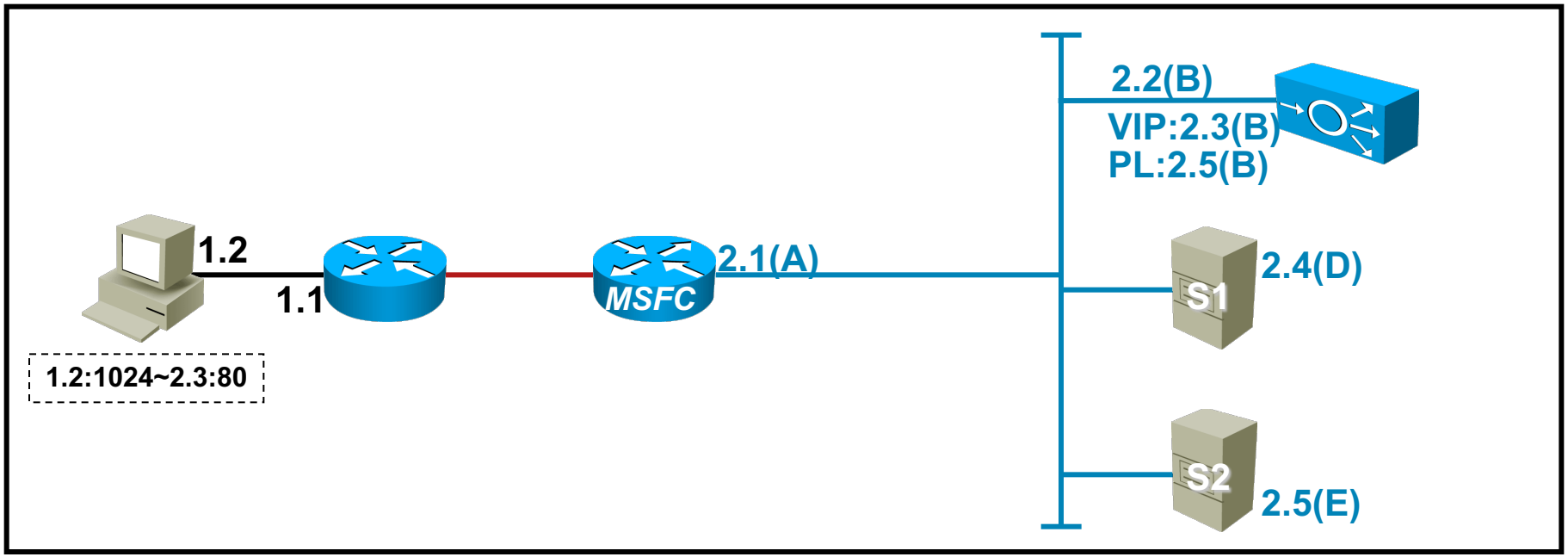


그래서 결과적으로 CSM으로 들어간 Traffic이 CSM을 거쳐서 나오지 않으면 문제가 된다고들 하는 것입니다.

정확히 말하면, CSM에서 문제가 발생하는 것이 아니라 Client에서 문제가 발생해서 Session이 수립되지 않는 거죠.

그럼 이 문제를 어떻게 하면 좋을까요? 일단 Return Traffic이 다시 CSM으로 들어갈 수만 있게 해 주면 될 겁니다.

One-Arm Mode : Load Balancing 구현 원리

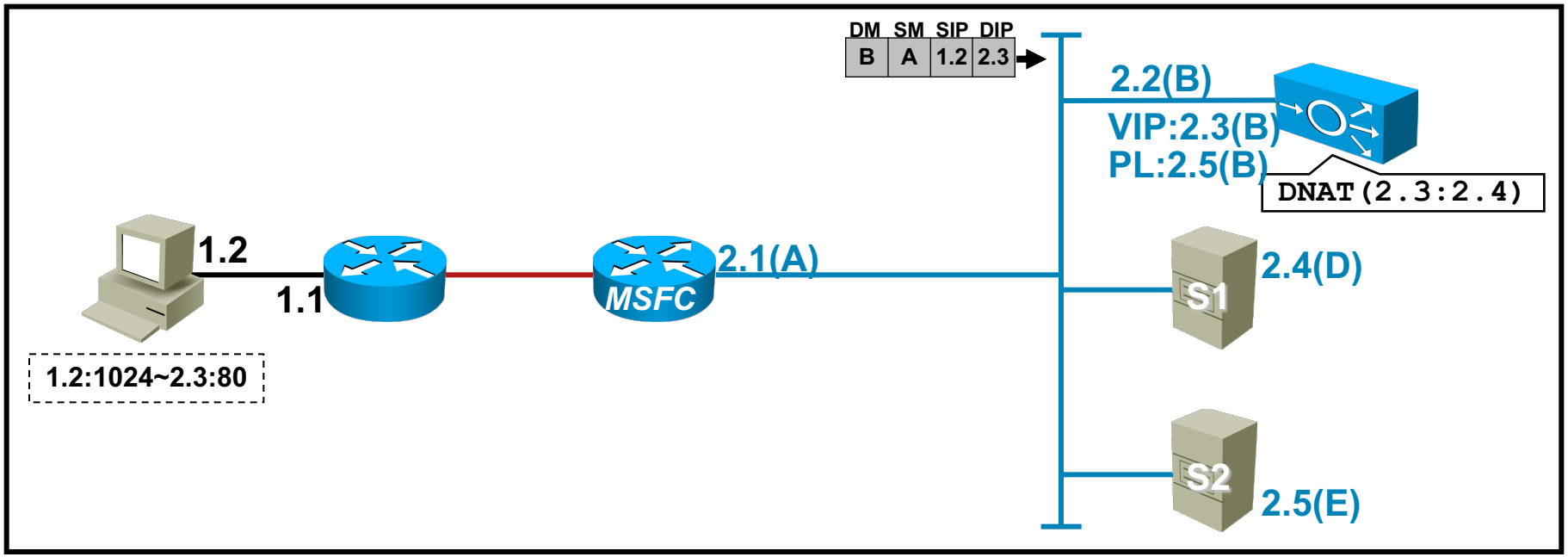


이 문제를 해결하는 한가지 방법은 CSM에서 Source NAT을 하는 것입니다. Source NAT을 하기 위해서 별도로 하나의 IP Pool을 만들었습니다. Pool 이름은 “PL”이며 2.5가 정의되어 있습니다.
그럼 이 2.5 IP의 MAC Address는 어떻게 될까요?

네, 맞습니다. 여전히 동일한 MAC Address인 “B”를 사용하게 됩니다.

자, 그럼 다시 한번 Packet Flow를 따라가 보도록 하겠습니다. 준비되셨나요?

One-Arm Mode : Load Balancing 구현 원리

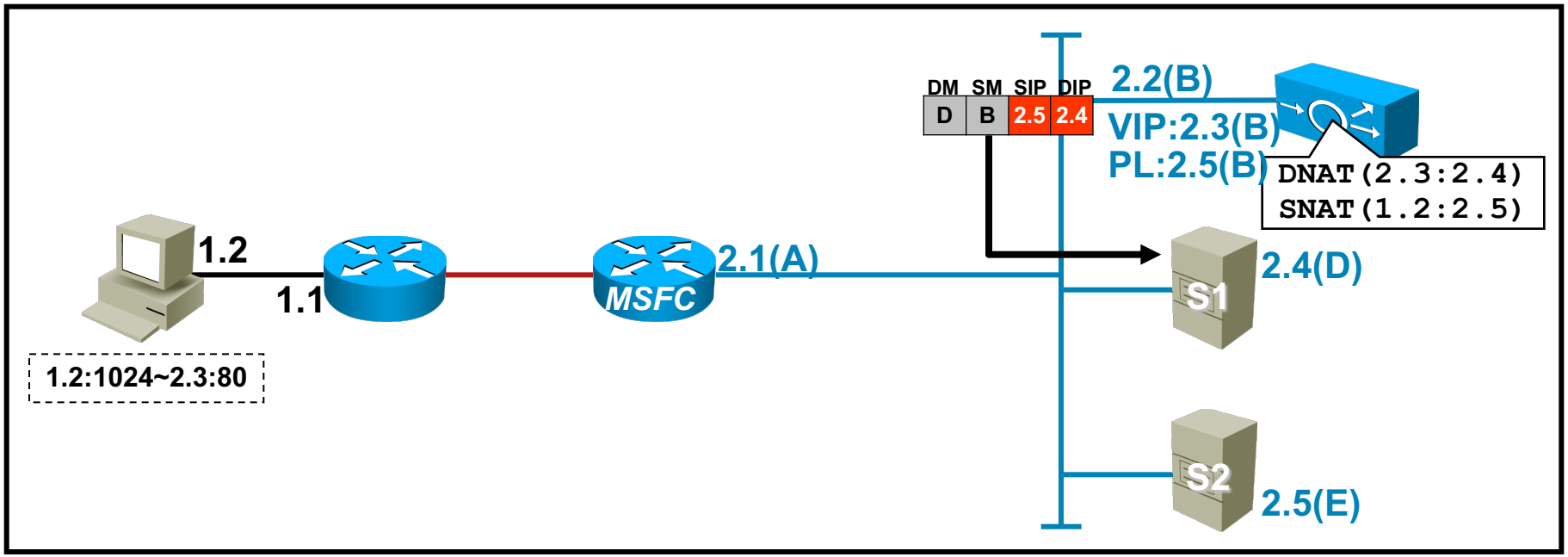


이번에는 진짜로 Client 부분은 빼고 설명을 하겠습니다! Client의 Socket 정보 정도만 한번 확인 하세요.

CSM은 Load Balancing을 하기 위해 기본적으로 Destination NAT를 해야 한다고 계속 말씀 드렸습니다.
물론 처리된 Frame을 Real Server로 Forward하기 위해 적당히 MAC Address도 바꿔 줘야 할 겁니다.

이번에는 여기에다 우리가 정의한 IP Pool을 이용하여 Source(Client)의 IP를 바꿔 주는 것이 하나 추가됩니다.
(CSM Configuration에는 serverfarm 아래 "`nat client PL`"이라는 Command로 들어갑니다.)

One-Arm Mode : Load Balancing 구현 원리

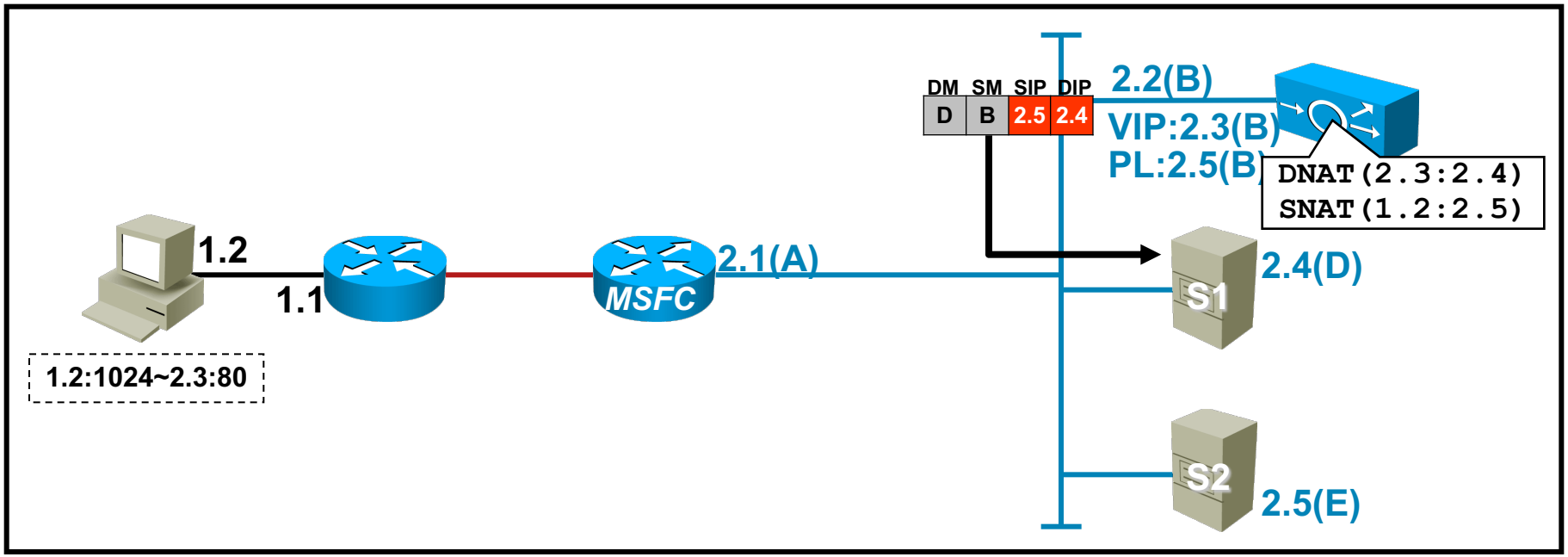


자, CSM이 Packet을 처리하면서 해당하는 Source NAT/Destination NAT Table이 생성되었습니다.

앞에서 설명 드린 대로 Packet Header를 보면, DIP였던 2.3은 Real Server의 IP인 2.4로 바뀌었군요,
SIP였던 1.2는 우리가 설정한 IP Pool인 PL의 IP, 즉, 2.5로 바뀐 것도 볼 수 있을 겁니다.

또한 SMAC은 Frame을 내 보내는 자기의 Interface MAC인 B가 되고,
DMAC은 Server 1의 MAC인 D가 된 것을 알 수 있습니다.

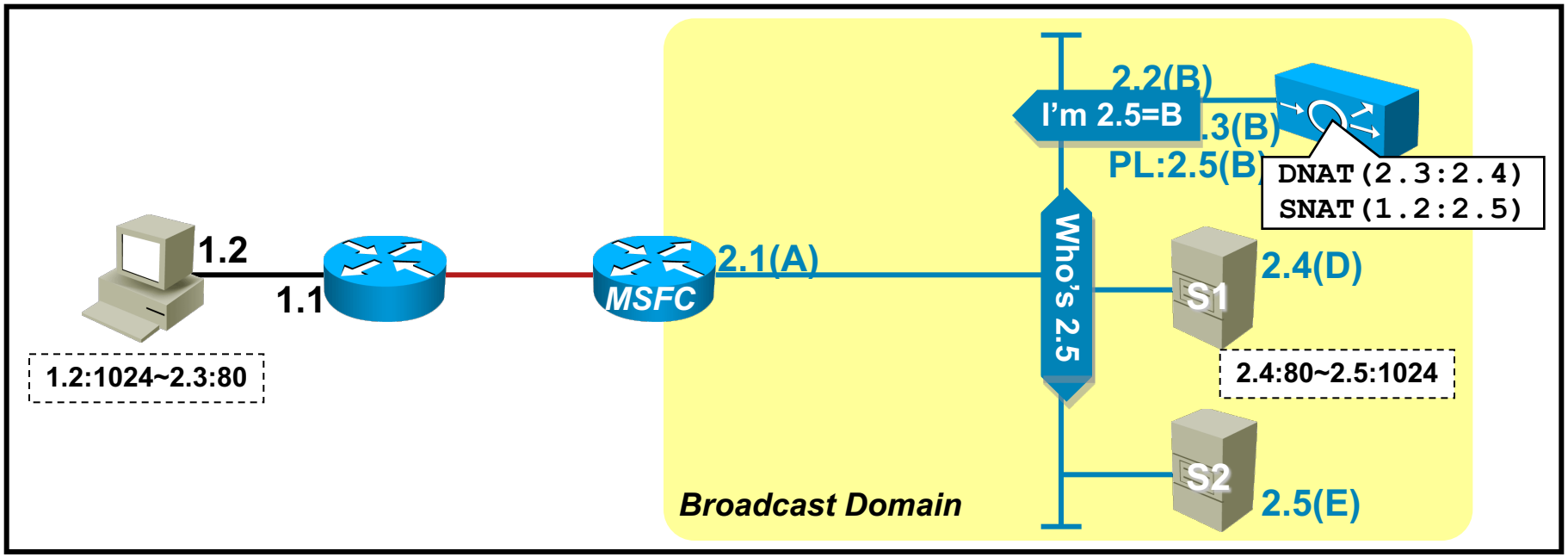
One-Arm Mode : Load Balancing 구현 원리



자, 이제는 Real Server가 응답을 할 차례입니다. 여기서 다시 질문!
Real Server는 자기가 받은 Traffic이 L4에서 온 것인지 알 수 있을까요, 없을까요?

정답은 있다/없다 둘 다 맞습니다. 설명을 하자면, 자기가 받은 Packet이 L4로부터 온 것인지는 모르지만,
아무튼 그 쪽으로 다시 보내줘야 한다는 것은 알거든요. 왜 그런지 자세히 한번 살펴 볼까요?

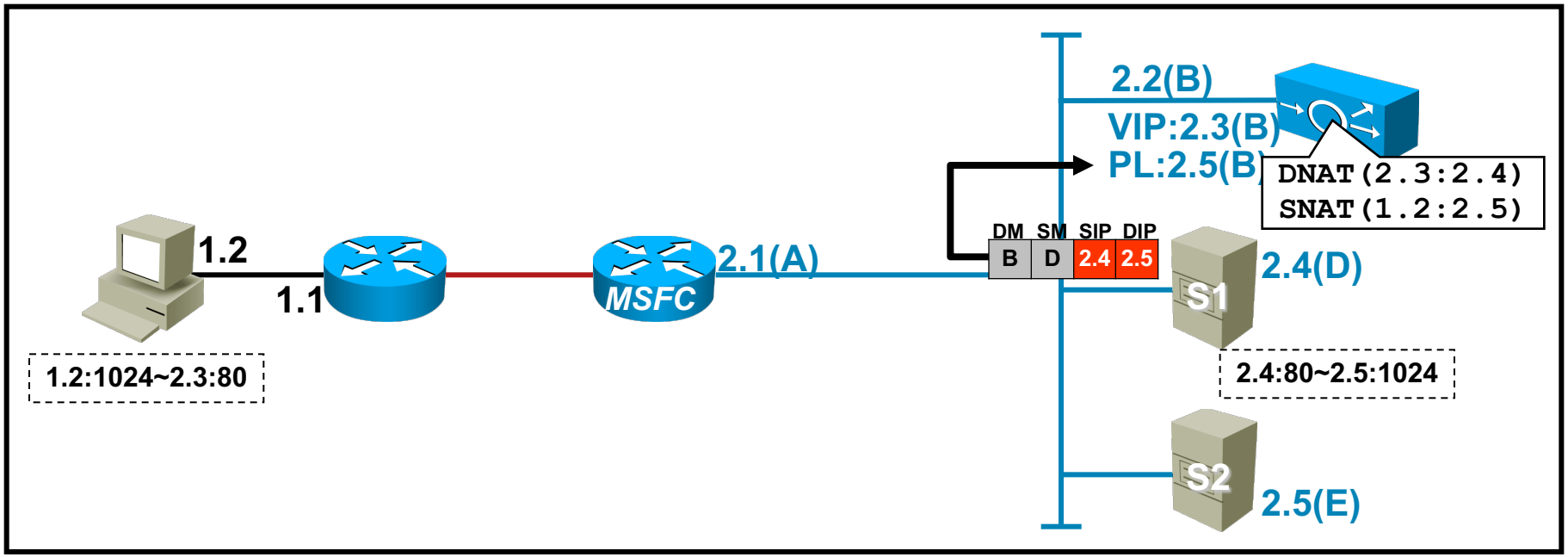
One-Arm Mode : Load Balancing 구현 원리



Real Server 2.4는 자기에게 접속을 요청한 Client가 2.5라고 간주하고 Socket 정보를 만듭니다. 따라서, 이 Real Server는 원래의 Client IP인 1.2가 아니라 2.5에게 응답을 주어야 합니다.

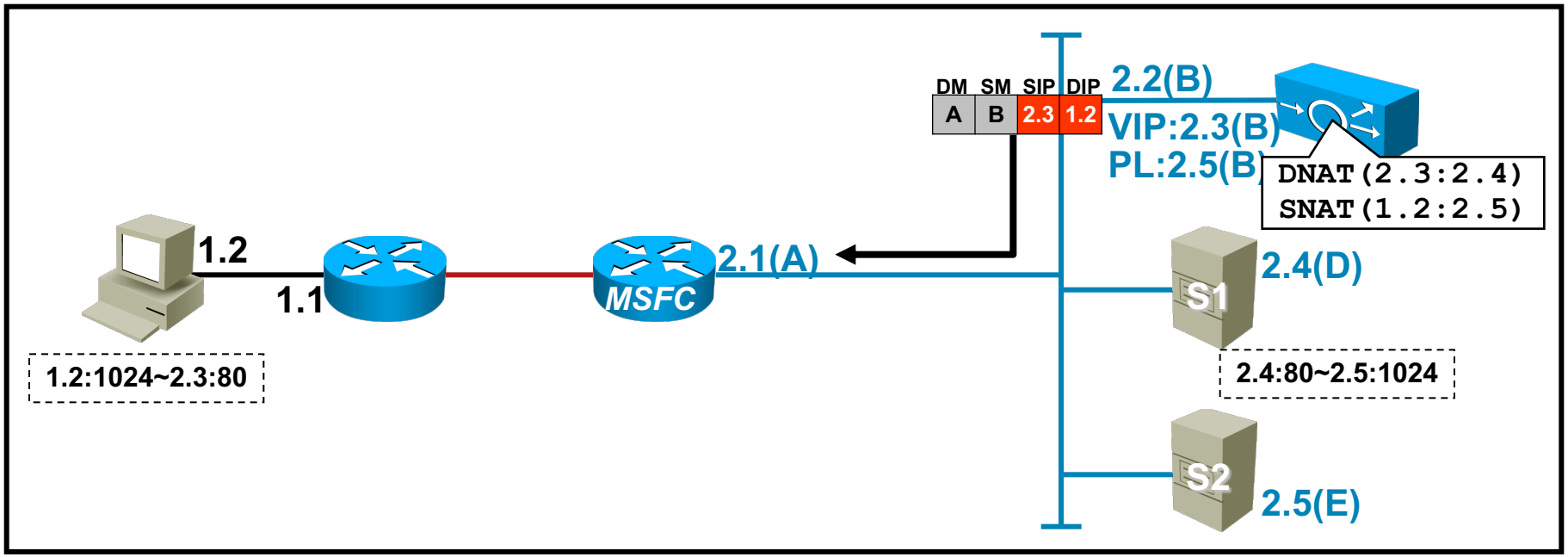
자, 그럼 이 2.4 Server는 2.5의 MAC Address를 알아야 2.5로 Frame을 만들어 보낼 수가 있겠죠?
이제 여기서 또 우리 친구 ARP가 등장을 하는데, 2.4 Server는 자기가 속한 Broadcast Domain으로 ARP Request를 뿌리게 되고 그 응답은 2.5 IP를 갖고 있는 CSM이 2.5=B로 해 줍니다.

One-Arm Mode : Load Balancing 구현 원리



그래서 Packet Header를 만들 때, SIP는 2.4, DIP는 2.5로 채우고, 해당 MAC을 사용하여 Frame을 만들게 됩니다.
결국 응답 Packet이 일단 L4로 다시 들어오게 하는 데는 성공을 했습니다. Congratulations!!

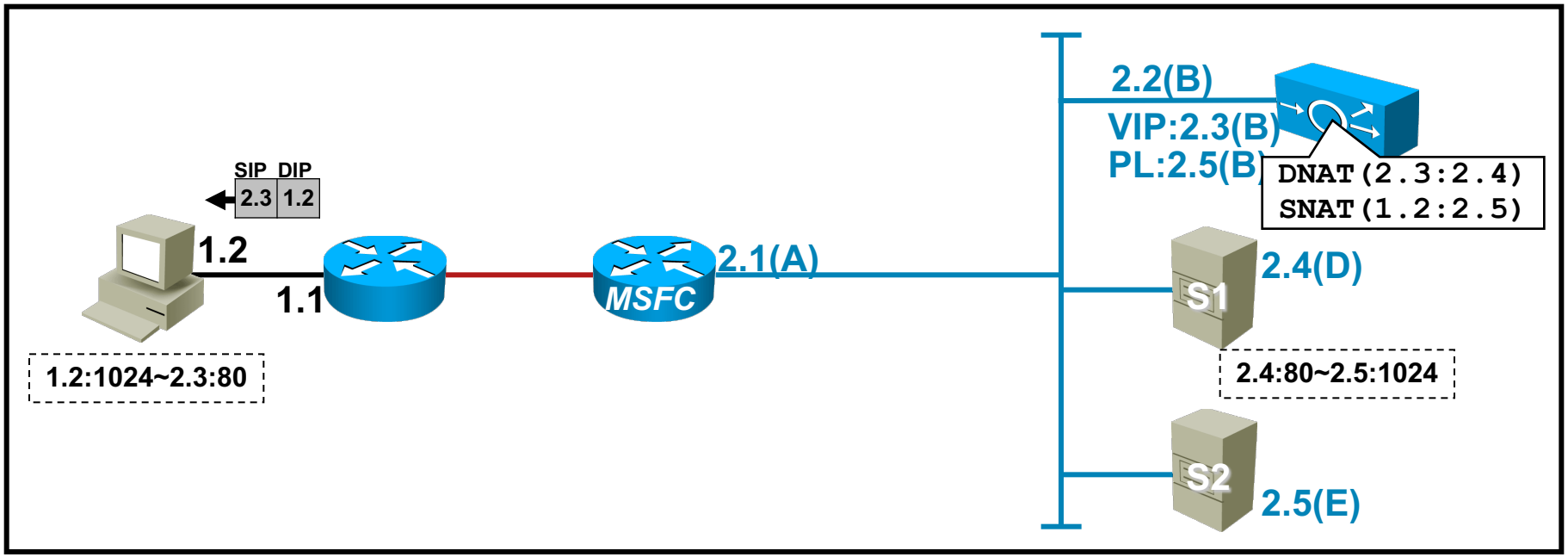
One-Arm Mode : Load Balancing 구현 원리



자, 이제 마지막 한 Step 남았습니다. 조금 복잡하니 집중하세요~

CSM은 자기가 갖고 있는 두 NAT Table을 이용하여 Real Server로부터 받은 Packet의 Source와 Destination을 모두 다 변환 시킨 후 MSFC로 보냅니다.

One-Arm Mode : Load Balancing 구현 원리

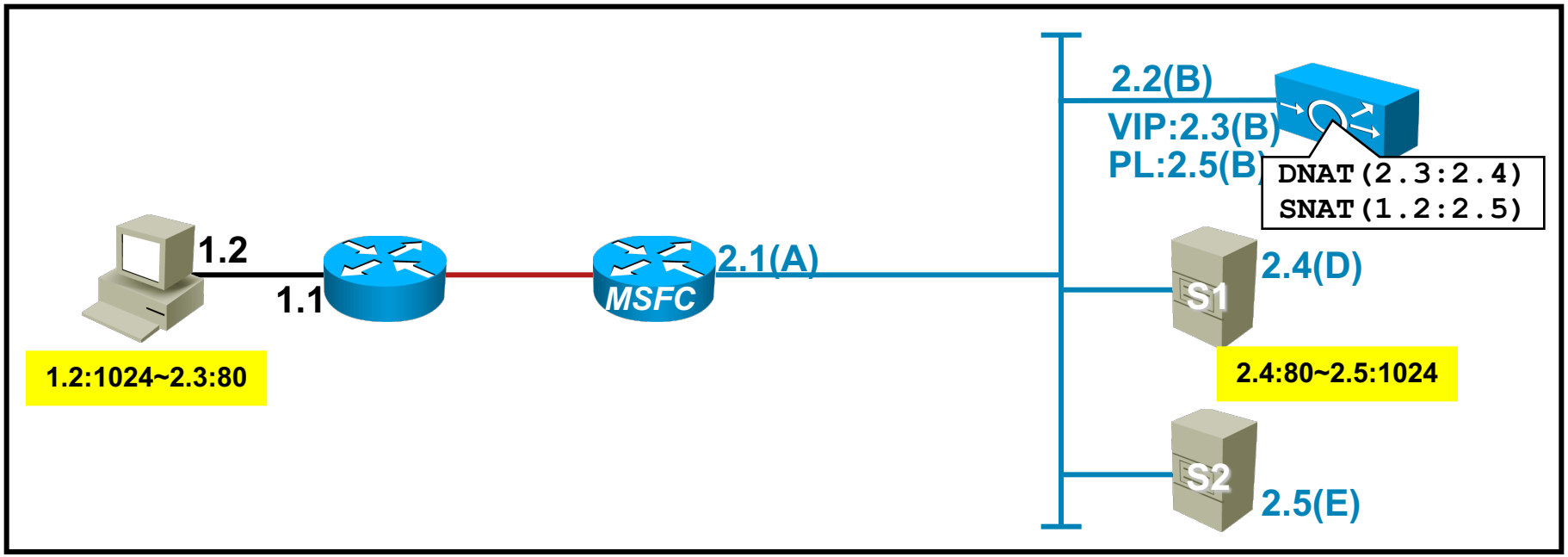


이제 우여곡절 끝에 응답 Packet이 Client에 도달 했습니다. 이번에는 문제가 없어야 할 텐데요…

받은 Packet의 Header를 보니, 이번에는 내가 요청을 보냈던 2.30이 맞군요!

이제 Client는 이 세션에 대해서 3-Way Handshake를 완료하고 Data를 전송하는 일만 남았습니다.

One-Arm Mode : 별로 권고되지 않는 구성



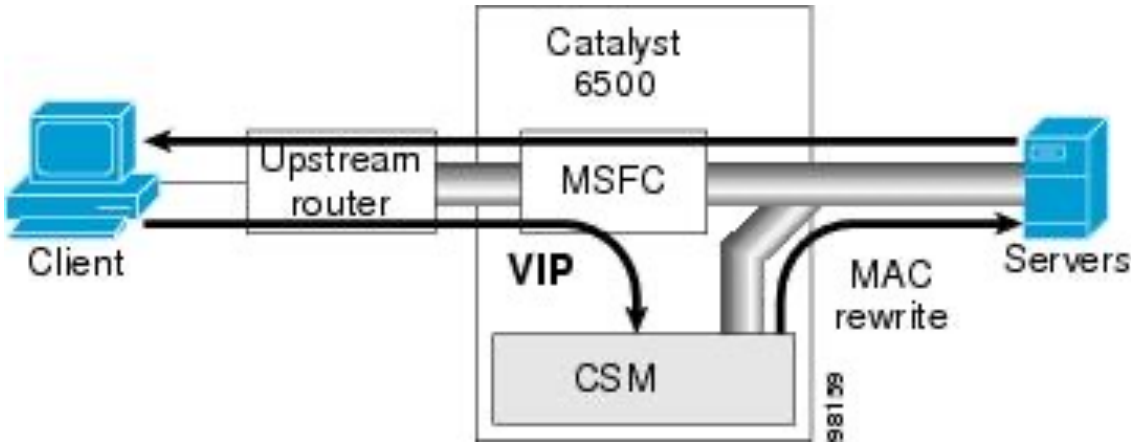
"근데요, IP Pool을 자세히 보니,20/24 대역을 쓰는데, 그럼 Client는 250대 정도 밖에 접속 못하는 거 아닙니까?"
그렇죠,잘 보셨습니다. 그래서 CSM에서는 Client PAT을 함께 지원 하고 있습니다. 해결이 되죠?

"하지만 머가 이리 복잡하죠? Server와 Client의 Socket 정보도 완전히 다르고, 좀 지저분하네요."
네, 역시 잘 보셨습니다. 그래서 이 One-Arm Mode는 일반적으로 권고 되지 않는 디자인입니다.

"아니, 그럼 쓰지도 않는 걸 왜 이렇게 구구절절 설명을 했나요?"
여러분께서 이 원리를 잘 이해하고 계시면 CSM 관련된 Troubleshooting 뿐 아니라, Real Server-to-VIP 통신 문제 등과 같은 여러 가지 Design Issue를 해결하는데 많은 도움이 됩니다. 입에는 쓰지만 몸에는 좋은 약입니다.

2. DSR(Direct Server Return) Mode

DSR(Direct Server Return) 소개

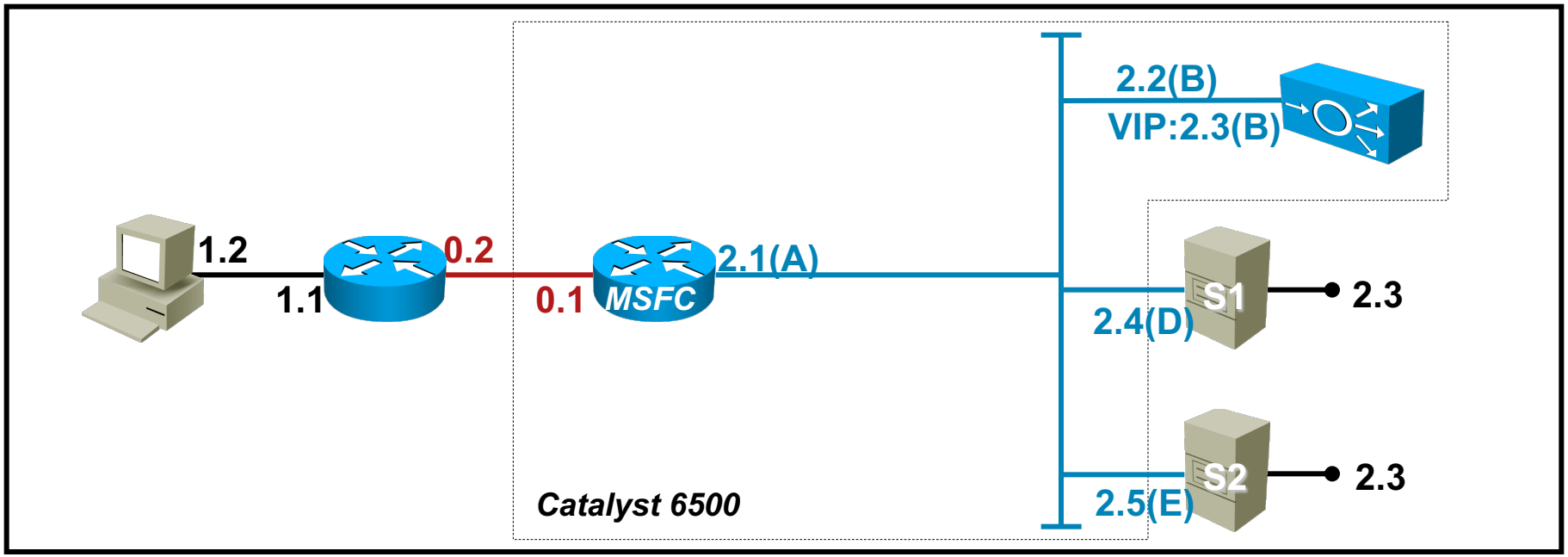


이 그림은 Cisco.com의 Documentation에서 인용한 그림인데요 DSR이 이해하기 쉽게 그려져 있습니다.

DSR은 문자 그대로 Server가 직접 응답을 주는 구조인데요, “아니! 지금까지 계속 L4로 들어간 Traffic은 다시 L4로 나와야 한다고 해 놓고선 이게 무슨 소린가?” 하시는 분들도 계실 겁니다.

하지만 이제 앞으로 몇 장 슬라이드를 통해 설명드릴 내용을 잘 따라 오시면 곧 “아하~”하시는 순간이 올 겁니다.

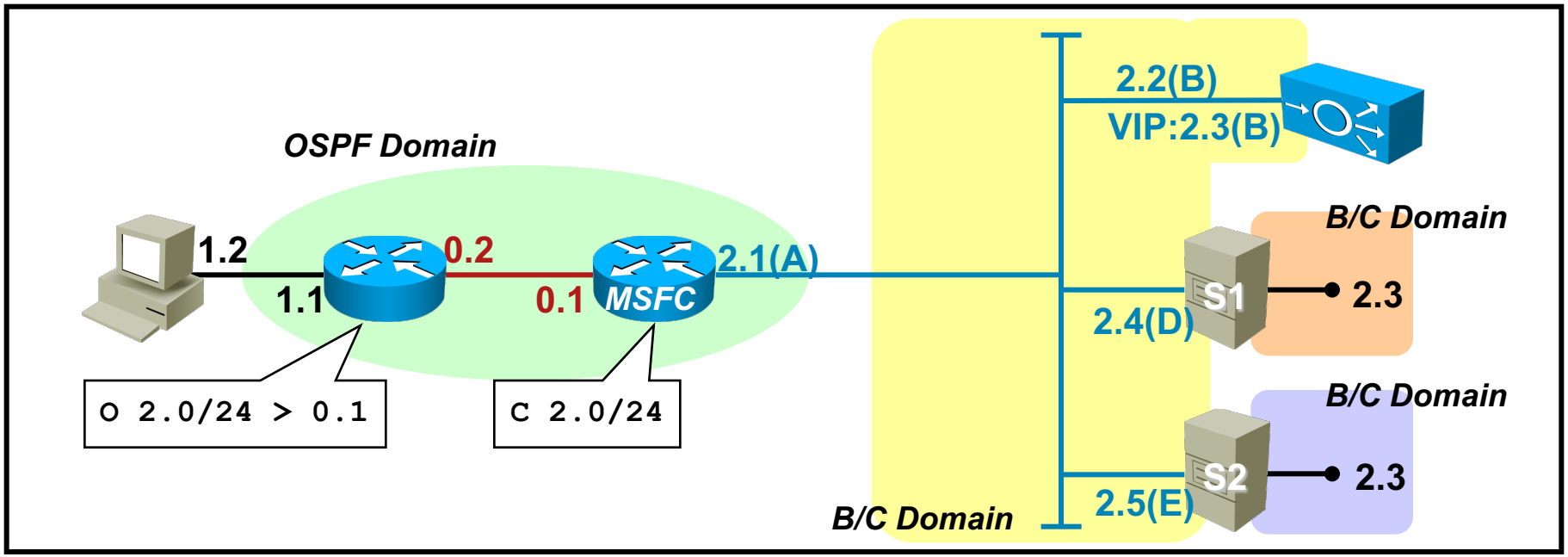
DSR : VIP와 Real IP 대역이 같은 경우



앞에 있는 그림을 네트워크 엔지니어 입장에서 다시 그려보면 위와 같습니다. 이제 척 보면 아셔야 합니다. 그런데, 여러분 잘 보시면 뭔가 약간 다른 점이 있는데 보이시나요?

네, Real Server에 Loopback Interface가 하나씩 있고, 이상하게도 여기에 VIP가 설정 되어 있는 것을 발견하셨을 겁니다. 신기하죠? 바로 이것이 DSR의 중요한 Trick 중에 하나입니다.

DSR : Routing Design & ARP Mechanism

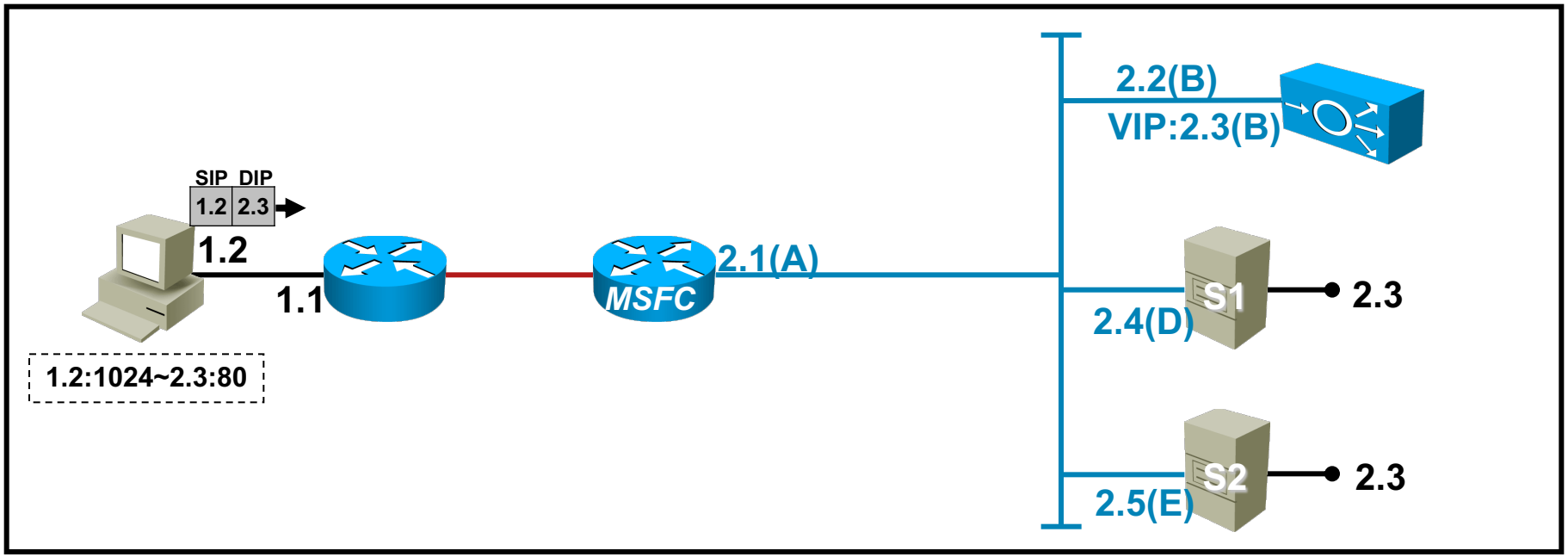


혹시 Virtual IP가 중복해서 사용되는 것에 대해 거부감을 느끼시는 분들을 위해 Routing에 대해서 좀 더 확실히 짚어 보고 가겠습니다.

이 경우에는 2.0/24가 Dynamic Routing에 의해 전파 되므로 별도의 Static Route 설정이 필요없습니다.

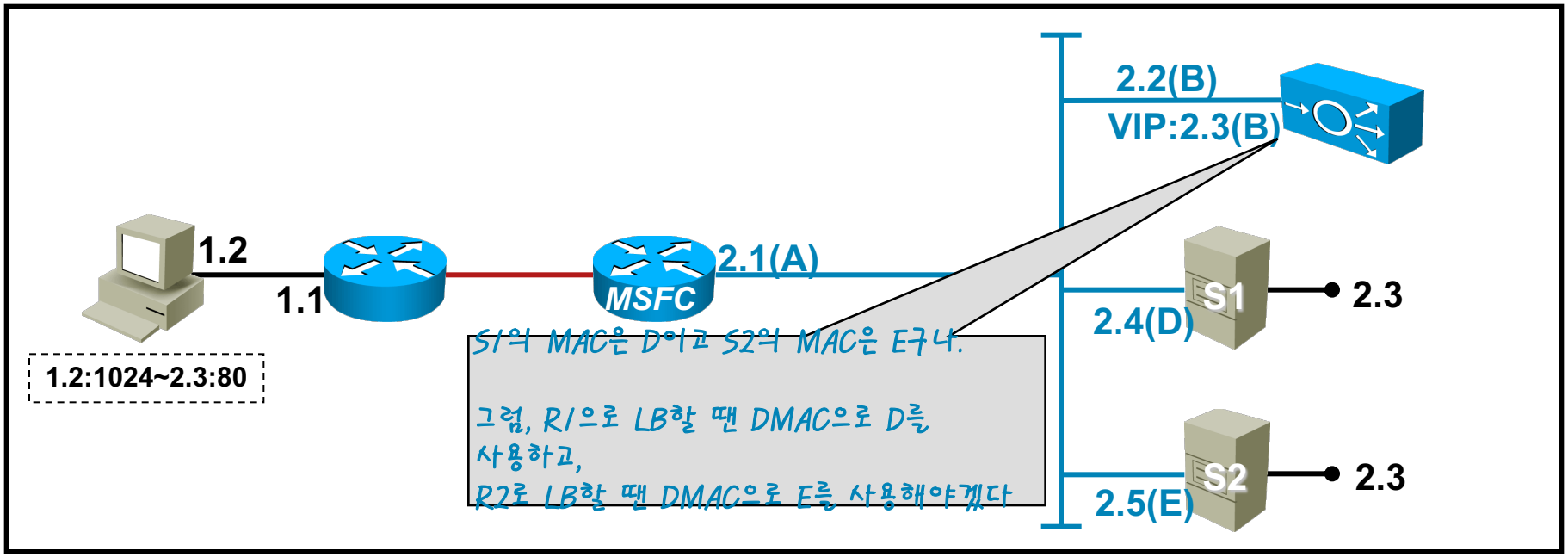
2.3 IP에 대한 ARP 충돌을 염려하시는 분들도 계시겠습니다만, 2.3 IP를 갖고 있는 모든 Interface가 각각의 Broadcast Domain으로 분리 되어 있어서 IP 충돌은 발생하지 않습니다.

DSR : Load Balancing 구현 원리



Client가 VIP로 HTTP 접속을 시도하고 있습니다. SIP는 자기 자신인 1.2. DIP는 VIP인 2.3이 되겠죠?

DSR : Load Balancing 구현 원리

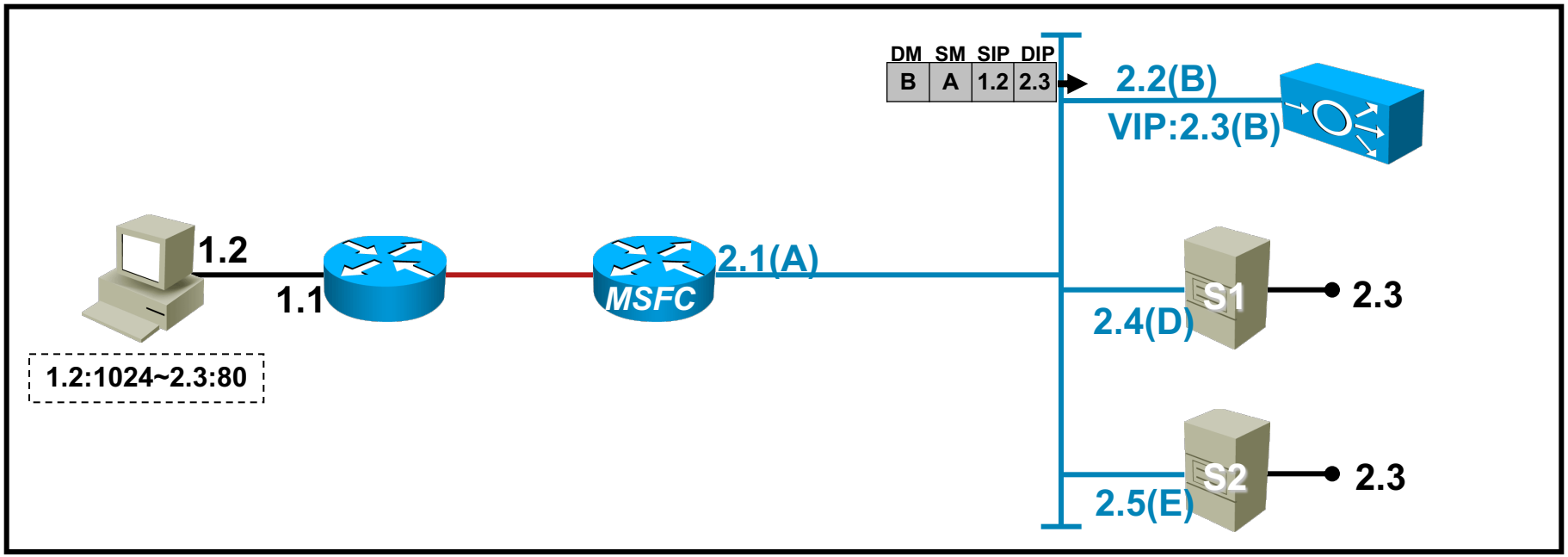


자, 여러분께 제가 지금까지 여러분께 자세히 설명을 안 드린 것이 하나 있는데, 이제 설명을 드려야 할 것 같습니다.

CSM이 Load Balancing을 위해 요청 Packet안에 있는 Destination IP를 Real Server의 IP로 바꾼다고 했는데, Destination MAC Address는 이 Destination IP에 해당하는 MAC이 사용되는 것이 아니라 CSM에 등록된 Real Server들의 MAC Address를 DIP와 관계 없이 Load Balancing 해서 사용을 합니다.

즉, S1으로 LB할 때는 S1의 MAC(D)을 사용해서 Encapsulation하고 S2로 LB할 때는 S2의 MAC(E)를 사용해서 Encapsulation해서 Frame을 보낸다는 의미입니다.(p.11에 간단히 설명한 것보다는 조금 더 깊은 내용입니다)

DSR : Load Balancing 구현 원리

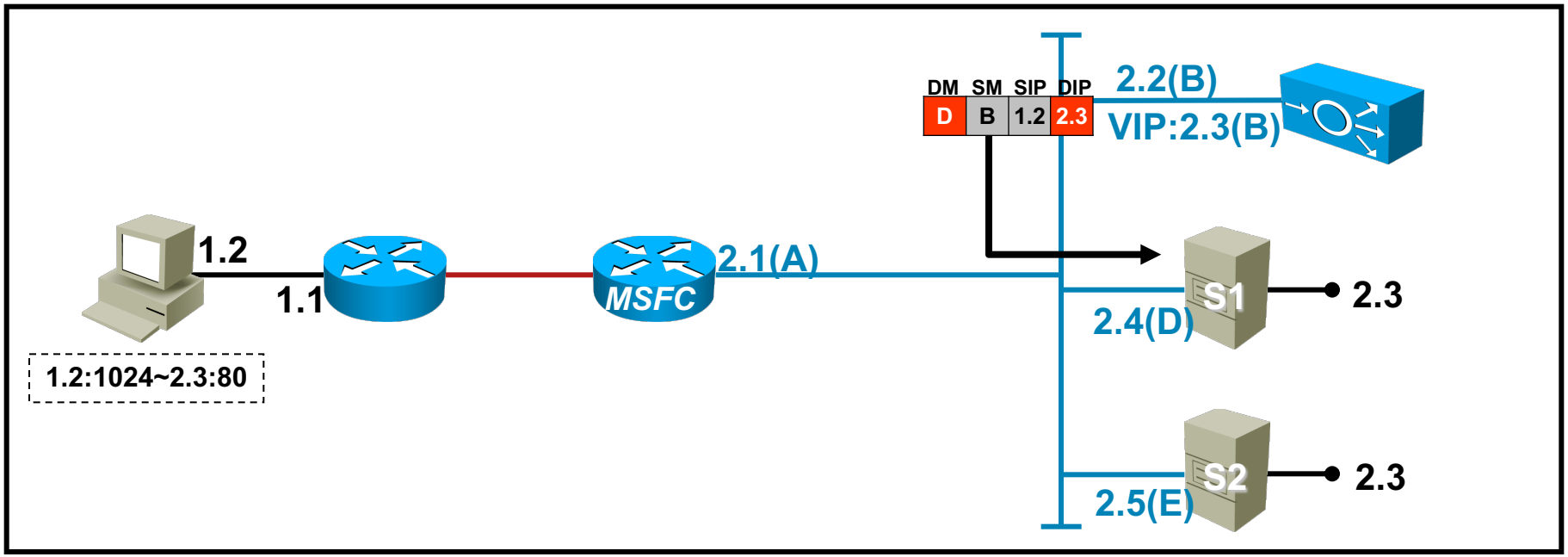


자, 이제 Packet이 CSM에 도달했습니다. 일반적인 경우라면 Destination IP(2.3)를 Load Balancing Mechanism에 의해 Real Server의 IP(2.4)로 바꿔야 할 텐데, DSR을 하실 때는 이것을 하시면 안됩니다. CSM의 Configuration에는 "`no nat server`"라고 입력을 해 줘야 합니다.

"아니, 지금까지 Load Balancing 을 하려면 DIP를 Real IP로 바꿔야 한다면서요?"
네, 맞습니다. 하지만 DSR을 하실 때 만큼은 이 기능을 꺼 줘야 합니다.

그럼 DIP를 바꾸지 않고 어떻게 Load Balancing이 가능할까요?
지금 바로 이해가 안되더라도 일단은 다음 페이지로 저와 함께 넘어가 주시기 바랍니다.

DSR : Load Balancing 구현 원리

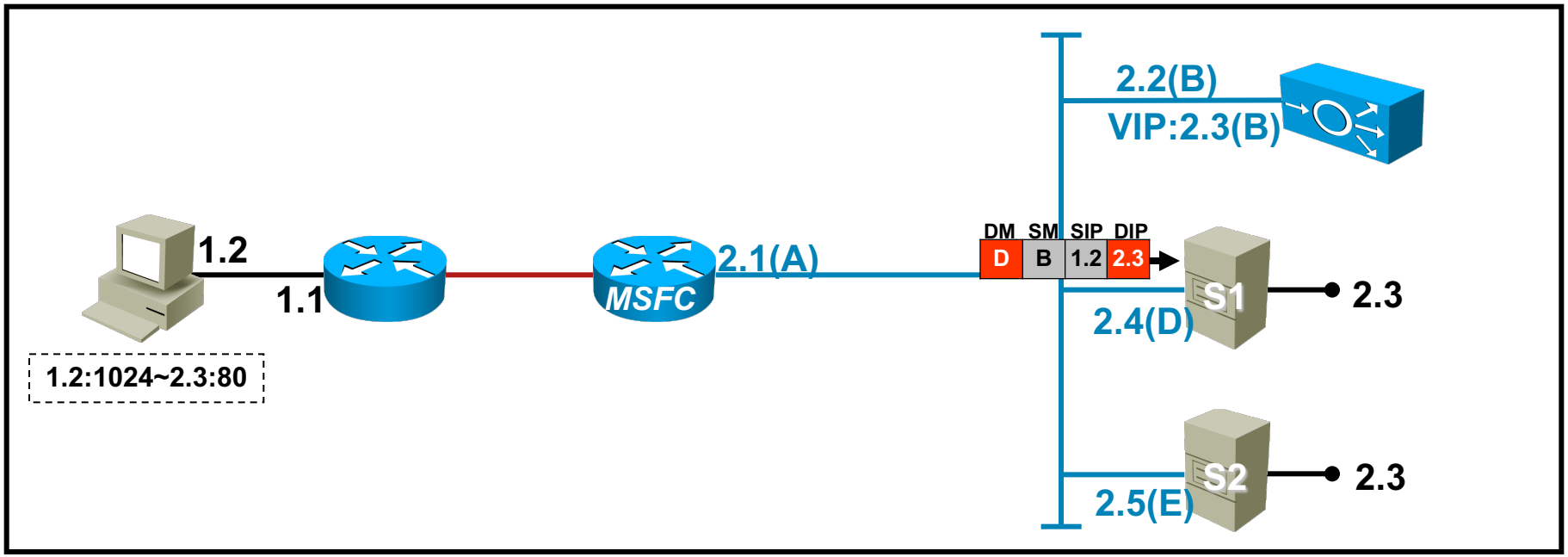


앞의 설명에 이어서, S1으로 Load Balancing 되는 Packet의 L3/L2 Header를 만들어 보겠습니다.

우선 SIP는 Client의 IP인 1.2입니다. 여기까지는 괜찮죠?
다음에 DIP는 바꾸지 않고 놔둡니다. 그냥 2.3이 됩니다. 껄끄러워도 지금은 그냥 놔둡니다.

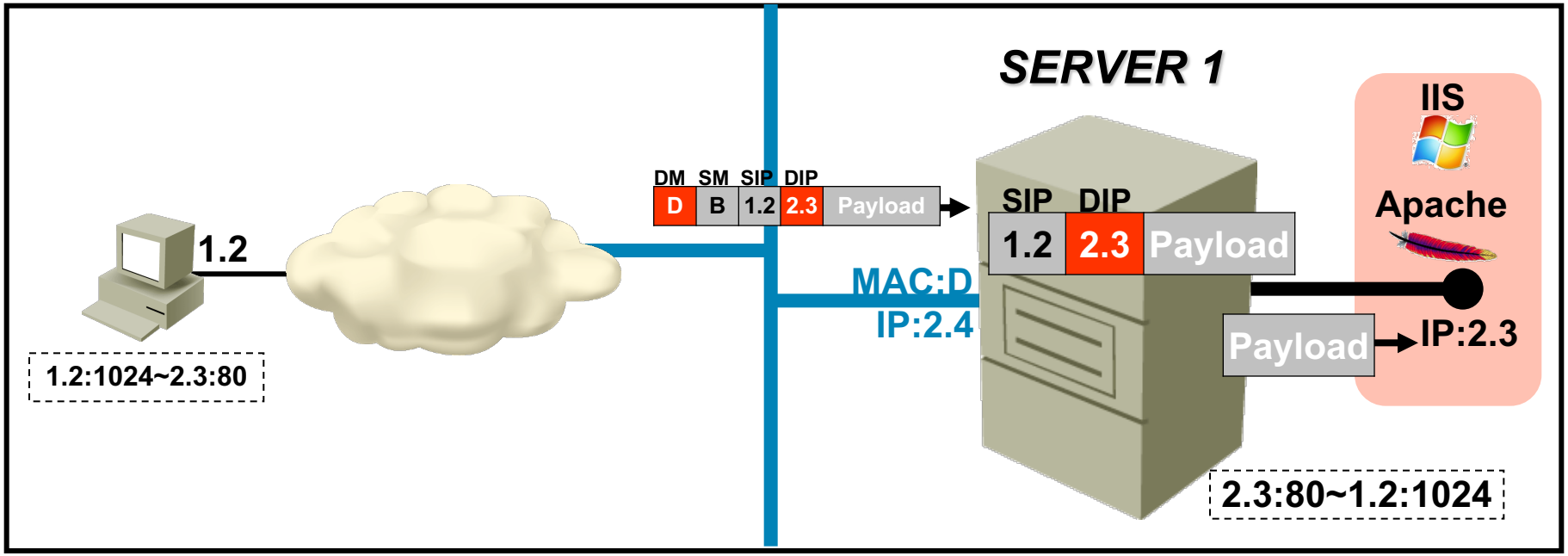
DMAC은 앞에 앞에 페이지 설명에 의하면 2.3의 MAC을 쓰는 것이 아니라 S1의 MAC을 쓴다고 했습니다. “D”네요.
SMAC은 항상 Frame이 나가는 Interface의 MAC을 사용하게 되어 있으니 “B”가 되겠네요.

DSR : Load Balancing 구현 원리



이 Frame은 Switching Path를 통해 S1으로 들어갔습니다. 어쨌든 여기까지 Load Balancing은 이루어 졌습니다.

DSR : Load Balancing 구현 원리

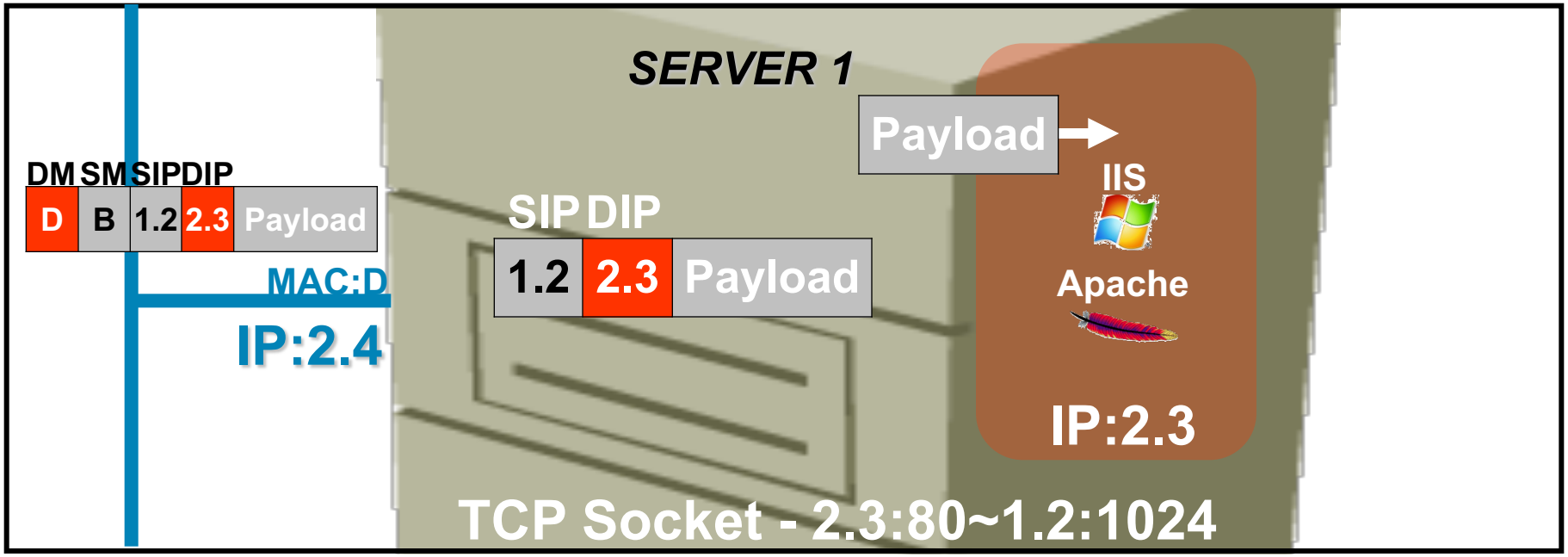


자 이제는 Server1을 유심히 들여다 볼 차례입니다. 이 Packet을 받은 Server는 “아, 1.2라는 Client가 나의 2.3 IP와 Session을 맺으려고 하는구나”하고 판단을 할 것입니다.

하지만 Loopback에 설정된 2.3 IP는 결국 다른 Server가 아닌 자기 자신이므로 이 Server는 1.2와 자기의 Loopback IP인 2.3과 Socket 정보를 생성하게 됩니다.

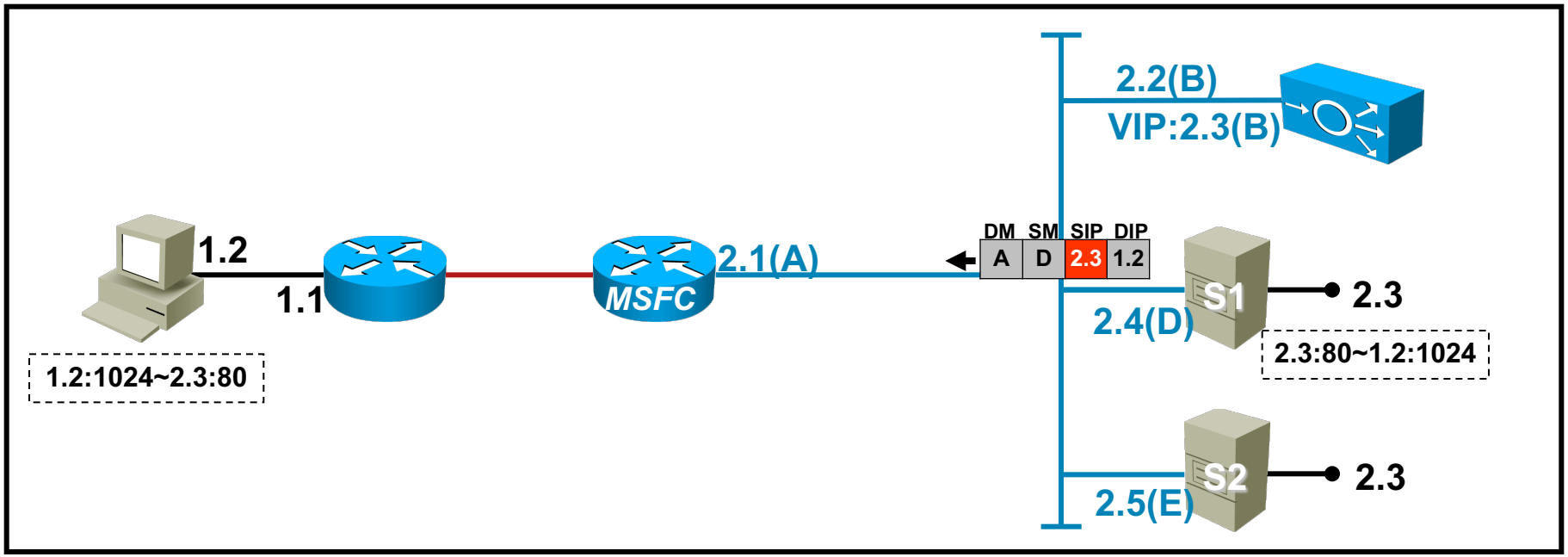
물론 이 경우에는 Web Server Daemon이 Physical IP인 2.4가 아니라 Loopback IP인 2.3에 대해서 올라와 있어줘야 하겠죠.

DSR : Load Balancing 구현 원리



혹시 시력이 안 좋으신 분들 위해 한 컷 추가…

DSR : Load Balancing 구현 원리

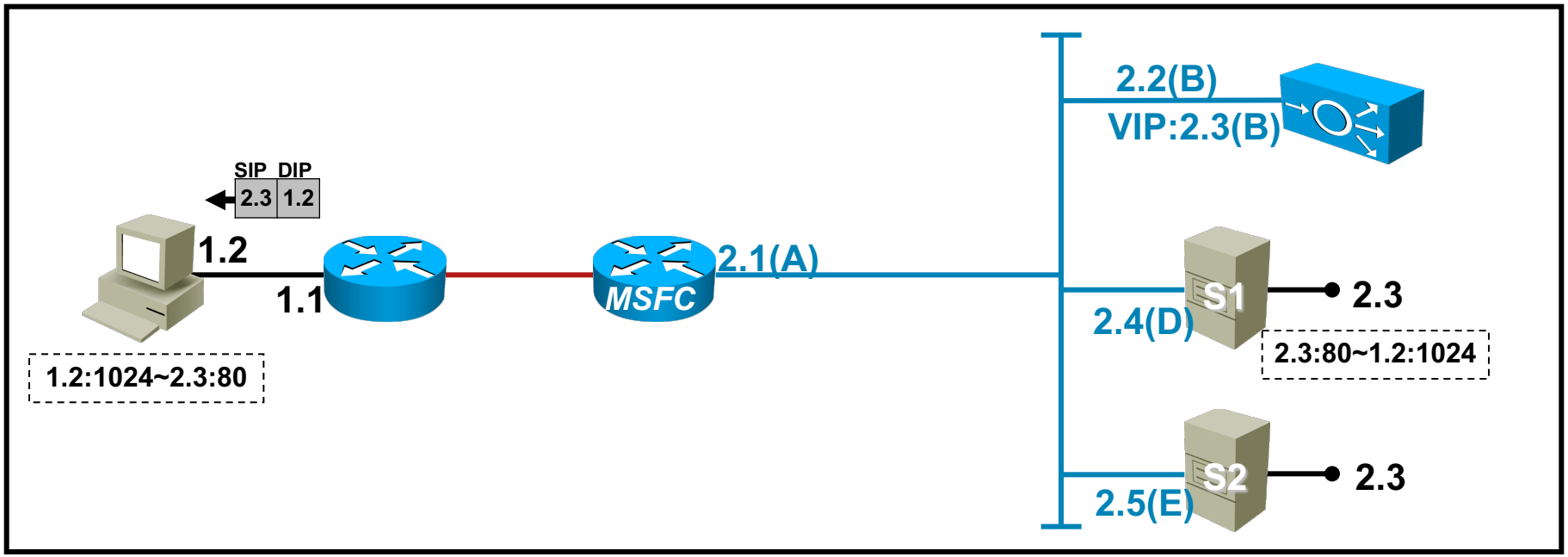


자, 이제 Server가 응답을 할 차례가 되었습니다. Server는 자기의 Socket 정보를 이용해서 응답 Packet을 만들고 ARP를 기반으로 Frame을 만들어 낼 것입니다.

따라서 Source IP는 2.3(2.4가 아닌 것에 유의), Destination IP는 1.2, 그리고 Destination MAC은 자기의 Default Gateway의 MAC인 A, Source MAC은 Frame을 내 보내는 Physical Interface의 MAC인 D로 하게 됩니다.

DMAC	SMAC	SIP	DIP
A	D	2.3	1.2

DSR : Load Balancing 구현 원리

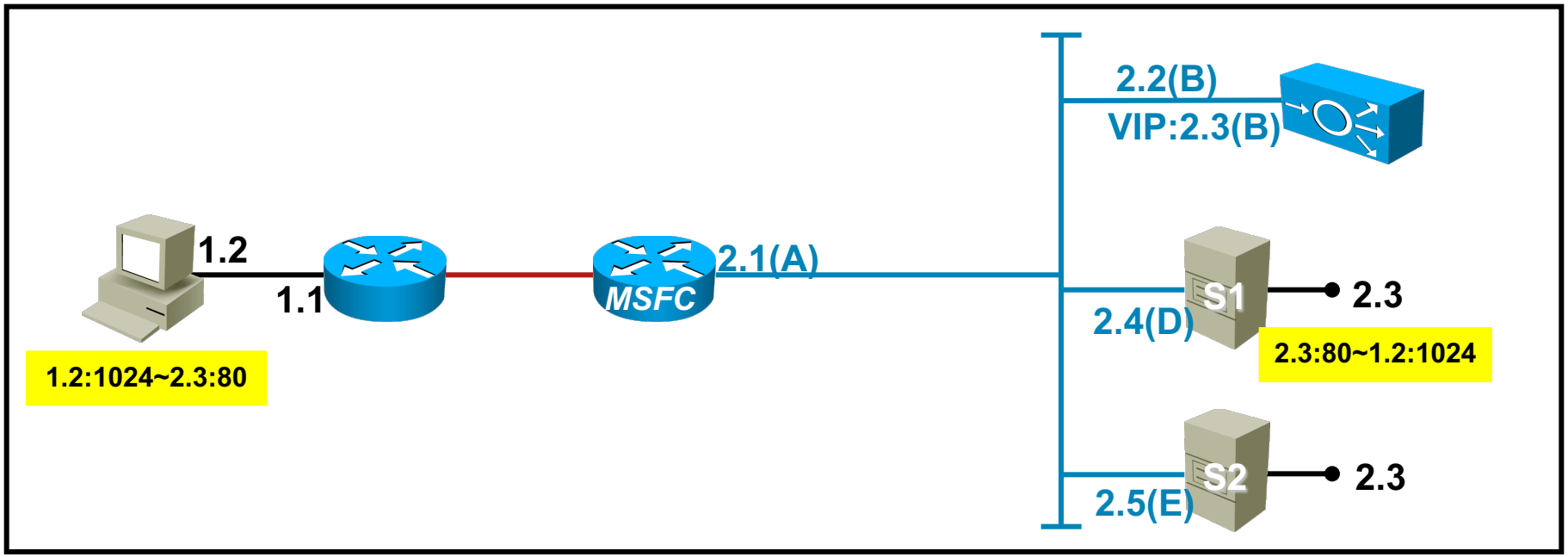


헥헥.. 힘드시죠? 이제 거의 다 왔네요. 글 쓰는 저도 손과 머리에 쥐납니다.^^

드디어 응답 Packet이 Client에게 도달했습니다. 자… 그럼 과연 Client가 요청 Packet을 보낸 Destination IP로부터 응답이 왔을까요?

우리가 요청을 보냈던 IP는 2.3, 그리고 응답을 준 Source IP도 2.3.
네, 결국 정확한 IP로부터 응답을 받았고 이제는 TCP 3-way Handshake를 마치는 일만 남았습니다.

DSR : Load Balancing 구현 원리



이 DSR 구성은 장점도 단점도 많은 구성입니다.

One-Armed 구성과 마찬가지로 Load Balancing할 Traffic만 CSM으로 보내어 고성능을 추구하면서도 Server와 Client에서 NAT 없이 정확이 Real IP로 다 보이므로, Client의 IP를 추적할 때 아주 좋습니다.

반면에 단점은 CSM이 Connection을 Control하지 못하므로 인해 생길 수 있는 보안 상의 취약점이 발생할 수 있고 CSM의 Hardware 상의 제한 때문에 그다지 높은 성능의 DSR을 구현할 수 없었습니다.
하지만 CSM의 고성능 차기 모델로 출시된 ACE의 경우에는 전혀 문제 없이 DSR을 구현하여 사용할 수 있습니다.

Epilogue

CSM이나 ACE 또는 그 밖에 CSS나 타사 L4 장비를 다뤄 보신 분들은 내용을 읽으시면서 많은 내용들 특히 다양한 구성 방식이 언급되어 있지 않은 것을 발견하셨을 수도 있습니다.

예를 들면, VIP를 Client VLAN 쪽이 아닌 Server VLAN의 IP 대역에 두는 구성, VIP를 Server VLAN도 Client VLAN도 아닌 제 3의 IP 대역에 설정하고 Static Routing 한 후에 Redistribute하는 방법도 Field에서 많이 사용하는 방법입니다.

또한 MSFC를 아예 Traffic Path에서 제외 시키고 CSM/ACE와 외부 Device가 직접 통신하도록 하는 방법, 그래서 CSM/ACE에서 직접 Static Route를 설정하고 마치 Router와 같은 역할을 하게 하는 MSFC-not Involved 구성도 고수 분들께서 많이들 사용하는 Mode입니다.

CSM은 이미 수많은 Reference를 통해 기능과 안정성이 입증 되었고, 16Gbps라는 고용량으로 Upgrade된 ACE2.0도 현재 금융권을 중심으로 수요가 급증하고 있습니다.

개인적으로 추천을 하자면, Router/Switch 기술에 익숙하신 SE분들은 이제 Firewall 또는 이 Load Balancer 영역으로 관심을 돌려 보시기를 바랍니다. 네트워크를 Layer 4 이상, Application에서 바라볼 수 있는 넓은 시야와 지식을 갖추시게 될 것입니다.


CISCO