Chapter 02

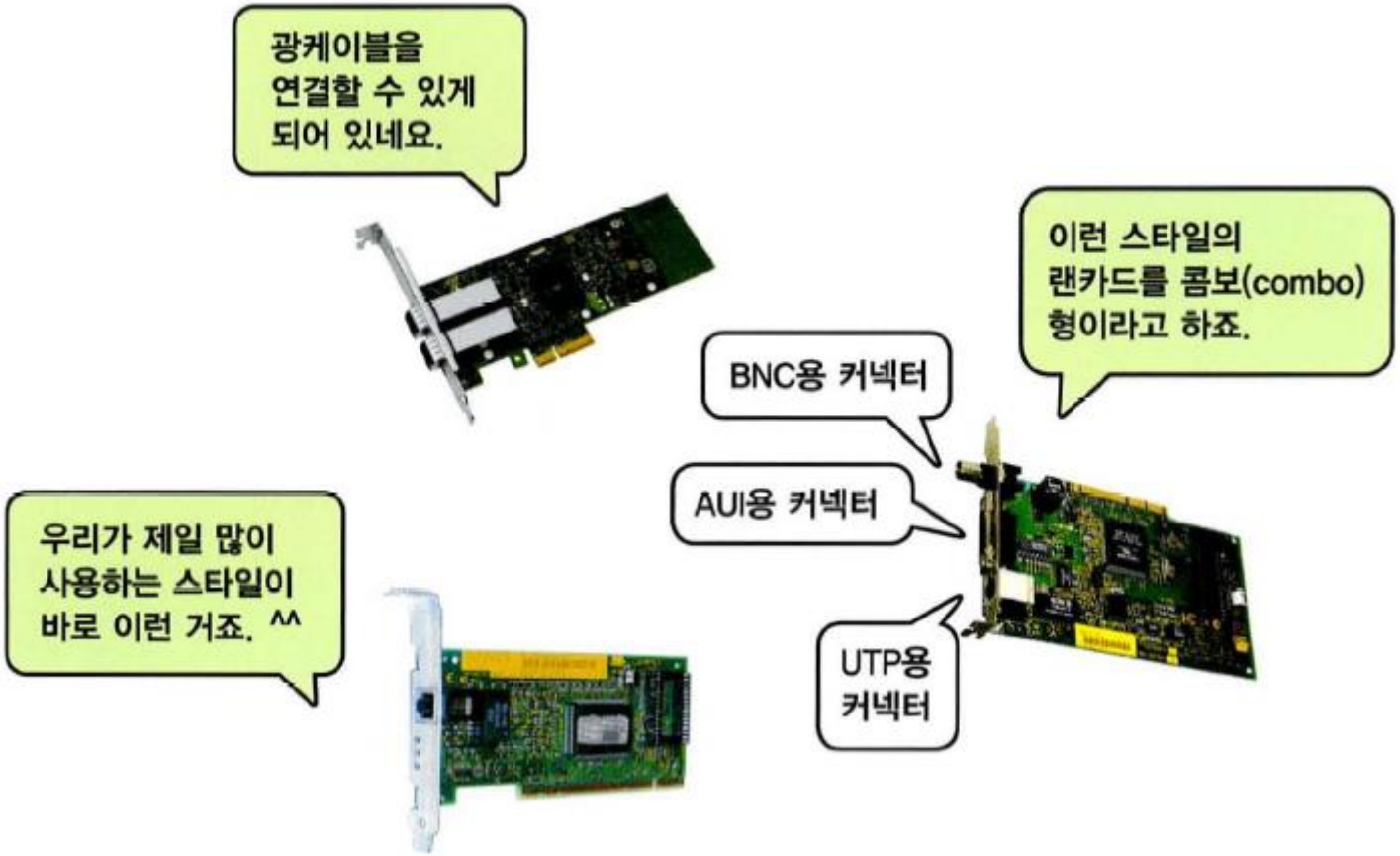
네트워크 장비의 개요

1. 랜카드

- NIC(Network Interface Card)이다.
- 보통 '랜카드'라고 하는 우리 주변에서 가장 많이 볼 수 있는 네트워크 장비이다.
- 랜카드는 유저의 데이터를 케이블에 실어서 허브나 스위치, 혹은 라우터 등으로 전달해주고 자신에게 온 데이터를 CPU에게 전달해주는 역할을 한다.
- 랜카드는 우선 어떤 환경에서 사용하는가에 따라서 이더넷용 랜카드와 토큰링용 랜카드, 그리고 FDDI, ATM용 랜카드 등으로 구분한다.
- 하지만 대부분의 환경에서는 이더넷용을 사용하기 때문에 아마 여러분 PC에 설치된 랜카드나 사무실에 설치된 랜카드의 90% 이상은 이더넷용 랜카드일 것이다.

1. 랜카드

- 어디에 설치하는가에 따라서 데스크톱용 랜카드와 PCMCIA 방식이라고 하는 노트북용 랜카드가 있다.



1. 랜카드

- 데스크톱용 랜카드를 선택하는 경우에는 또 하나 생각해야 할 것이 있다.
- 바로 PC의 버스(Bus) 방식이다.
- PC의 버스 방식은 크게 3가지 정도로 나누어 볼 수 있는데, 현재 가장 많이 사용하고 있는 방식이 PCI 방식이고 이전까지는 ISA 방식을 많이 사용했다.
- 자주는 아니지만 간혹 서버급 PC에서는 EISA(Enhanced ISA) 방식의 버스도 있다.
- 이더넷 랜카드의 경우는 속도에 따라 크게는 10메가, 100메가, 10/100메가, 1기가 등으로 나눌 수 있다.
- 몇 년 전까지만 해도 10Mbps용 랜카드가 일반적이었다.
- 그런데 요즘은 100Mbps 또는 1Gbps용 랜카드가 거의 대부분이다.
- 이는 네트워크 대역폭, 즉 네트워크상에서 날아다니는 데이터의 양이 그만큼 더 많아졌다는 것을 의미한다.

1. 랜카드

- 랜카드에 접속하는 케이블의 종류에 따라서 TP 포트를 가진 랜카드, BNC나 AUI 포트를 가진 랜카드, 광케이블과 접속하는 랜카드 등의 종류로 나누어 볼 수 있다.
- 예전에는 주로 AUI 타입의 커넥터와 BNC용 커넥터가 있는 방식을 많이 사용했다.
- 그리고 나서 한동안은 오른쪽에 보이는 콤보(combo) 방식을 많이 쓰기도 했다.
- AUI와 BNC, 그리고 UTP를 모두 골라서 연결할 수 있는 타입이 바로 이 방식의 랜카드이다.
- 한마디로 과도기적인 랜카드라고 볼 수 있다.
- 하지만 요즘에는 AUI나 BNC를 사용하는 사람이 거의 없는 편이니 주로 아래 보이는 UTP 타입을 쓰는 것이 일반적이다.

2. 허브

- 허브는 직사각형의 상자에 구멍이 뚫려있는 모양으로 되어 있다.
- 이 구멍이 몇 개 뚫려있느냐에 따라서 '몇 포트 허브다'라고 이야기하고, 이 구멍의 숫자가 바로 몇 대의 장비를 연결할 수 있는지를 결정하게 된다.
- 즉 랜카드가 설치된 각각의 PC들은 케이블을 타고 바로 이 허브에 연결된다.
- 그리고 같은 허브에 연결된 PC끼리는 서로 통신이 가능하다.
- 그렇다면 예를 들어 구멍 10개짜리 허브가 있는데, 18대의 PC를 연결하려면 어떻게 해야 할까?
- 첫 번째 해결 방법은 구멍이 18개 이상 되는 허브를 따로 1대 사는 것이다.
- 간단하긴 하지만 이 방법을 시용할 경우 기존에 가지고 있던 구멍 10개짜리 허브를 시용하지 못하기 때문에 절대 권하지 않는 방법이다.
- 두 번째 방법은 구멍이 10개인 허브 1대를 더 산 후 이 2대를 서로 연결하고 2대의 허브에 PC를 연결하는 방법이다.
- 즉 허브는 서로 연결을 하게 되면 마치 1대의 허브처럼 동작이 가능하다.

2. 허브

- 허브 역시 랜카드처럼 이더넷용과 토큰링용이 있고, 이더넷 허브도 속도에 따라 그냥 허브(10Mbps)와 패스트(100Mbps) 허브가 있다.
- 100Mbps 랜카드를 설치했는데 허브는 10Mbps용을 사용한다면 당연히 통신 속도는 10메가이기 때문에 랜카드에 맞는 허브를 선택하는 것이 중요하다.
- 허브는 네트워크에서 약방에 감초처럼 없으면 안 되는 가장 기본이 되는 장비 중 하나이다.
- 랜카드, 케이블, 그리고 허브만 있으면 일단 내부에서는 허브에 접속된 모든 PC가 서로 통신이 가능하다.

2. 허브

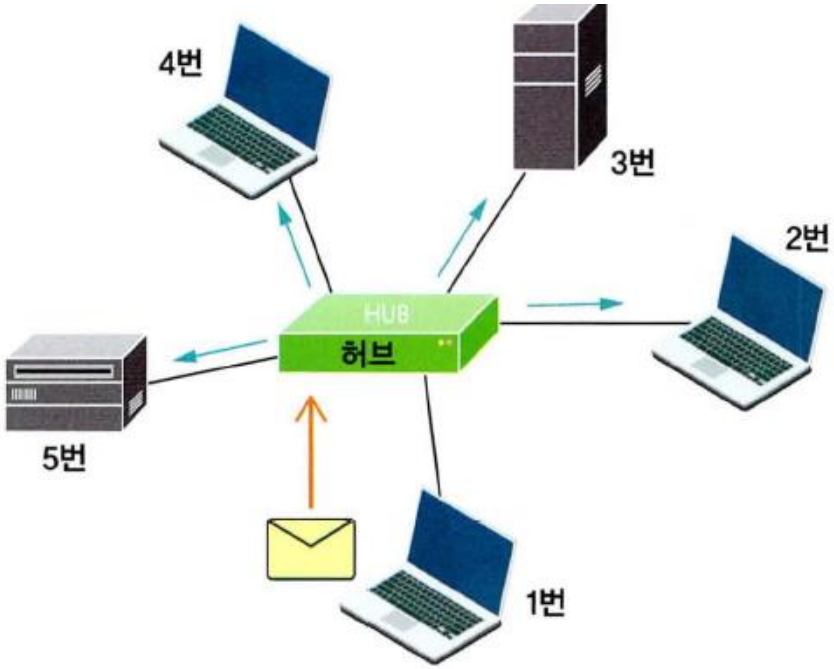
- 그럼 허브는 무슨 일을 할까?
- 허브를 한마디로 이야기하면 '멀티포트(Multiport) 리피터(Repeater)'라고 말할 수 있다.
- 멀티포트는 말 그대로 포트가 많이 붙어있다는 뜻이고, 리피터는 들어온 데이터를 그대로 재전송한다는 의미를 가지고 있으니까 허브는 포트가 여러 개 달린 장비인데, 이것은 한 포트로 들어온 데이터를 나머지 모든 포트로 뿌려준다는 것이다.
- 여기서 잠깐 리피터에 대해서 알아보자.
- 네트워크에서 데이터를 전송하는 경우 케이블에 따라서 전송 거리에는 제약이 있다.
- 예를 들어 우리가 현재 가장 많이 사용하고 있는 UTP 케이블의 경우는 최대 전송 거리가 100 미터이다.
- 즉 장비와 장비 사이가 100미터 이상 떨어져 있는 경우에는 통신이 불가능하다.

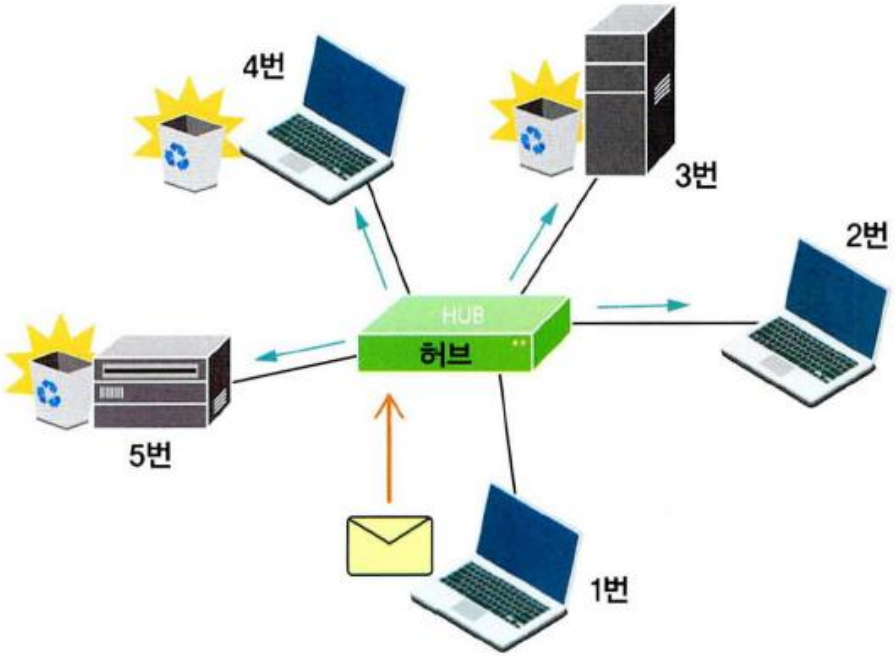
2. 허브

- 그렇다면 이 경우 케이블이 갈 수 있는 최대 거리 이상 떨어진, 예를 들어 두 장비 간의 거리가 150미터인 경우 두 장비 간을 UTP 케이블로 연결하려면 어떻게 해야 할까?
- 그때 중간에서 들어온 데이터를 다른 쪽으로 전달해 주는 역할을 하는 것이 바로 리피터이다.
- 음성 통신의 경우 엠프라고 하는 것이 멀리 떨어진 곳에 소리를 전달하기 위해 쓰이는 것처럼 데이터는 중간에 리피터가 있어서 한쪽에서 들어온 데이터를 그대로 다른 쪽으로 전달해주는 것이다.
- 따라서 중간에 리피터를 두고 두 장비는 케이블을 통해서 리피터로 연결하면 둘 간의 통신이 가능해지게 된다.
- 허브는 바로 이런 리피터의 기능도 가지고 있다.

2. 허브

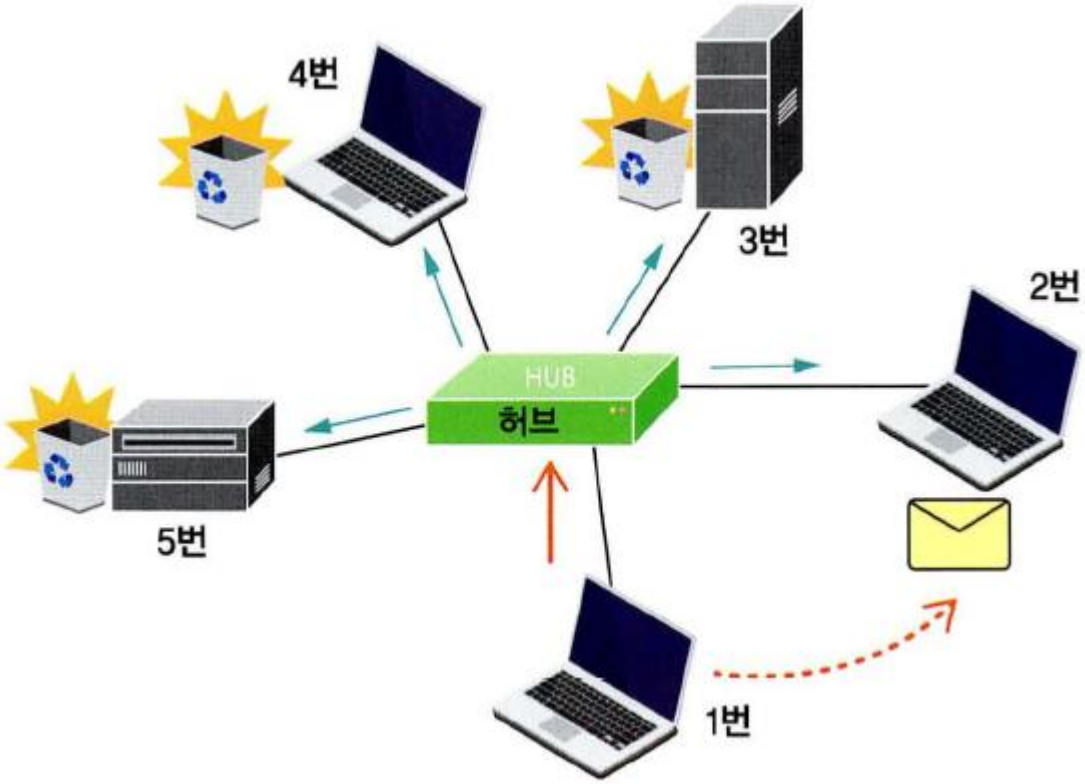
- 허브에 1번에서 5번까지의 PC가 붙어 있는데, 만약 1번 PC가 2번 PC에게 데이터를 전송하는 경우





2. 허브

- 허브에 1번에서 5번까지의 PC가 붙어 있는데, 만약 1번 PC가 2번 PC에게 데이터를 전송하는 경우



3. 허브의 한계

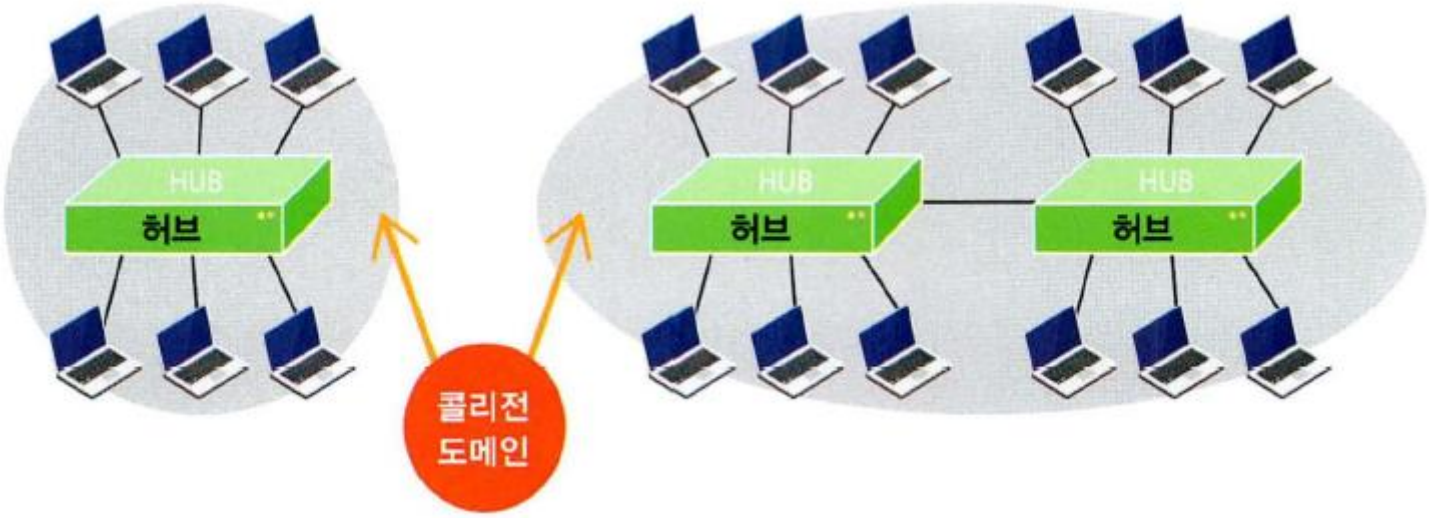
- 허브를 고를 때 무엇을 먼저 확인해 봐야 할까?
- 가장 중요한 것은 안정성이다.
- 또한 사후 AS 역시 중요한 요소이다.
- 나름대로 약간은 이름이 있는 메이커를 선택하는 것이 편리할 것이다.
- 허브에 연결된 모든 PC들은 서로 간에 통신이 가능하다.
- 또한 이 모든 PC들은 하나의 콜리전 도메인(Collision Domain) 안에 있기 때문에 어느 한순간에는 한 PC만 데이터를 보낼 수 있는 것이다.
- 따라서 이러한 기능을 수행하는 허브를 우리는 'Shared(셰어드, 즉 공유 방식) 허브'라고 한다.
- 즉 10Mbps의 속도를 그 허브에 연결된 모든 PC들이 공유한다는 것이다.
- 따라서 우리가 10Mbps의 허브에 20대의 PC를 연결해서 쓴다면 실제는 10Mbps를 20으로 나눈 만큼의 속도를 각자 쓰고 있는 것이다.

3. 허브의 한계

- 그런데 우리가 데이터 양이 아주 많은, 예를 들어 화상 회의나 멀티미디어 등에 모든 PC들이 계속 사용된다면 아무래도 10Mbps Shared HUB로는 무리가 될것이다.
- 자, 그래서 랜카드를 100Mbps로 바꾸고, 케이블도 100Mbps용으로 바꾸고, 허브도 100Mbps용으로 바꾸었다고 가정해보자.
- 더 빨라질까?
- 물론 조금 더 빨라지겠지만 우리가 원하는 속도는 아닐 것이다.
- 왜냐하면 100Mbps 허브 역시 어느 한순간에는 한 PC만 네트워크상에 데이터를 실어 보낼 수 있기 때문이다.

3. 허브의 한계

- 또 한 가지는 한 번의 충돌이 발생하면 그 허브에 붙어있는 모든 PC들이 영향을 받기 때문이다.
- 게다가 만약 허브 한 대를 추가해서 허브 두 대가 서로 연결되어 있고, 그 두 대의 허브에 붙어있는 모든 PC들은 하나의 콜리전 도메인 안에 있기 때문에 더욱 더 충돌이 자주 발생할 수밖에 없다.



3. 허브의 한계

- 또 한 가지는 한 번의 충돌이 발생하면 그 허브에 붙어있는 모든 PC들이 영향을 받기 때문이다.
- 게다가 만약 허브 한 대를 추가해서 허브 두 대가 서로 연결되어 있고, 그 두 대의 허브에 붙어있는 모든 PC들은 하나의 콜리전 도메인 안에 있기 때문에 더욱 더 충돌이 자주 발생할 수밖에 없다.

4. 지능형 허브

- 우리가 보통 허브를 이야기할 때 허브의 종류를 나누는데, 그게 바로 '인텔리전트 (Intelligent) 허브'와 '더미 (Dummy) 허브'이다.
- 굳이 하나 더 나누자면 '세미인텔리전트(Semi-Intelligent) 허브' 이다.
- 인텔리전트 허브란, 말 그대로 지능형 허브이다.
- 보통 우리가 인텔리전트 허브와 더미 허브를 나누는 가장 중요한 요소로는 NMS(네트워크 관리 시스템)를 통해서 관리가 되는가이다.
- 즉 인텔리전트 허브는 NMS에서 모든 데이터를 분석할 수 있을 뿐 아니라 제어도 가능하다.
- 말 그대로 앉아서 멀리 있는 허브의 동작을 감시하고 조정까지 할 수 있다.

4. 지능형 허브

- 그런데 이 기능은 정말 대형 네트워크에서 NMS를 쓰는 경우에나 필요하지, 사무실에서 PC 몇대 쓰는 경우라든지, 아니면 PC방에서 사용하는 경우에도 과연 필요할까?
- 답은 '전혀 필요치 않다' 이다.
- 따라서 인텔리전트라고 무조건 좋은 건 아니라는 것이다.
- 값만 비쌀 뿐이다.
- 하지만 인텔리전트 허브는 위의 기능 이외에도 몇 가지 기능을 가지고 있다.
- 예를 들어보자.
- 허브에 연결된 한 PC에 문제가 발생했다.
- 그래서 그 PC는 계속 이상한 데이터를 허브로 끊임없이 보낸다고 가정해 보자.
- 그럼 어떻게 될까?
- 계속해서 충돌이 발생하게 되면서 다른 모든 PC는 통신이 불가능한 상태로 빠져들게 된다.

4. 지능형 허브

- 바로 이더넷의 CSMA/CD란 특징 때문이다.
- 이 경우 문제의 PC를 찾아내서 PC를 끄지 않는 이상에는 그 문제를 해결할 수가 없다.
- 하지만 인텔리전트 허브의 경우는 문제의 PC가 연결된 포트를 찾아내어 자동으로 Isolation(현 네트워크에서 분리시켜서 따로 고립시킴)시켜 버린다.
- 즉 문제가 계속되는 포트는 방출시켜 버리는 것이다.
- 따라서 그 한 PC는 통신이 불가능하게 되겠지만, 나머지는 그 PC로부터 영향을 받지 않으므로 정상적인 통신이 가능하다.
- 또 분리된 포트는 허브에서 램프로 표시되기 때문에 바로 어떤 PC인지 알게 되어 조치가 가능하다는 것이다.
- 이 기능을 Auto Partition이라고 하는데, 이 기능은 요즘은 더마 허브에서도 있는 경우가 많다.

4. 지능형 허브

- 또한 세미(semi) 더미 허브란 일단 더미 허브인데 인텔리전트 허브와 연결하면 자기도 인텔리전트 허브가 된다.
- 따라서 혼자 있을 때는 더미 허브, 그리고 인텔리전트 허브랑 같이 있으면 인텔리전트 허브가 되는 허브를 말한다.
- 그림에 허브의 사진이 있다.

4. 지능형 허브

- 이 허브는 스택이 가능한 스태커블 허브여서 스택을 위한 케이블로 서로 연결되어 있는 것을 볼 수 있다.
- 아래에 있는 그림은 네트워크 관리 화면이다.
- 즉 허브의 상태를 웹 브라우저를 통해서 그대로 확인할 수 있을 뿐 아니라 포트의 상태나 데이터 양의 감시까지도 가능하다.
- 물론 이 경우에는 허브 관리에 필요한 값을 세팅해주어야 한다.
- 관리용 IP 주소라든지 암호, 관리 옵션 등을 넣어주어야 멀리 떨어진 곳에서도 관리가 가능하게 된다.
- 이처럼 관리 기능을 제공해 주는 허브를 '인텔리전트 허브'라고 한다.

5. 스위치

- 아무리 빠른 속도를 내는 허브를 쓴다고 하더라도 어느 한순간에는 한 PC만 데이터를 보낼 수 있다고 했다.
- 즉 허브에 연결된 한 PC에서 발생하는 충돌이 다른 PC들에도 영향을 주는 충돌 도메인(영역)이 그 허브에 연결된 모든 PC들이라는 뜻이다.
- 그러므로 바로 충돌 도메인이 너무 커지는 상황을 항상 조심해야 한다.
- 충돌 도메인이 너무 커지게 되면 충돌에 의해 영향을 받는 PC가 너무 많아지면서 통신의 속도가 점점 떨어지게 된다.
- 이러한 문제를 해결하기 위해서 충돌 영역을 나누어 줄 수 있는 장비가 나왔는데, 이 장비가 바로 브리지(Bridge) 또는 스위치(Switch)이다.
- 스위치가 나오기 전까지는 이 역할을 브리지 혼자 다 해주었지만, 이제 브리지보다 빠른 스위치가 나왔으니 브리지는 그 자리를 스위치에게 물려주고 사라져가는 추세이다.

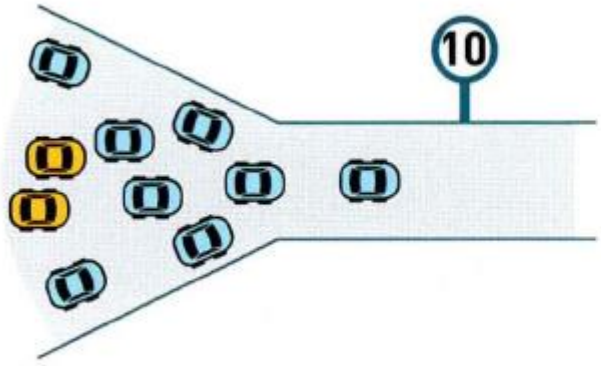
5. 스위치

- 스위치는 예를 들어 1번 포트에 연결된 PC가 2번 포트에 연결된 PC와 데이터를 주고받는 동안에도 3번 포트에 연결된 PC와 4번 포트에 연결된 PC가 서로 데이터를 주고받을 수 있게 하는 장비이다.
- 이걸 전문적인 용어로는 '포트별로 충돌 도메인이 나뉘어 있다'라고 말한다.
- 즉 1번과 2 번 사이에서 통신이 일어나면 나머지 모든 PC들은 기다려야만 하는 허브와는 달리 다른 PC들도 동시에 통신이 가능하다.
- 이것이 스위치와 허브의 가장 큰 차이이다.
- 그래서 우린 스위치의 경우 각각의 포트에 연결된 PC가 독자적으로(Dedicated 하게) 10Mbps 또는 100Mbps의 속도를 갖는다고 이야기한다.
- 스위치는 허브에 비해서 데이터를 처리하는 방법이 우수할 뿐만 아니라 데이터의 전송 에러 등을 복구해 주는 기능 등 여러 가지 기능을 가지고 있다.

5. 스위치

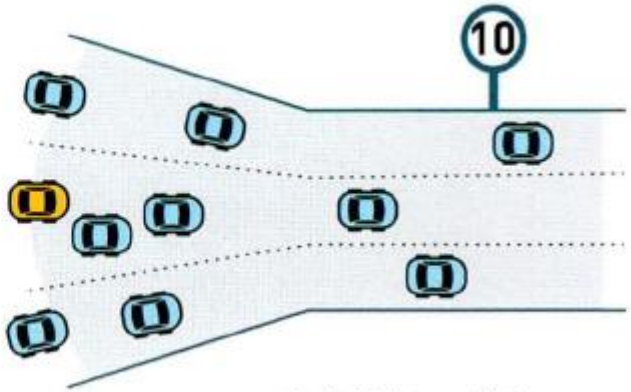
■ 허브와 스위치의 그림 비교

허브(Shared Ethernet)



한 번에 한 노드만
보낼 수 있다.

스위치(Switched Ethernet)



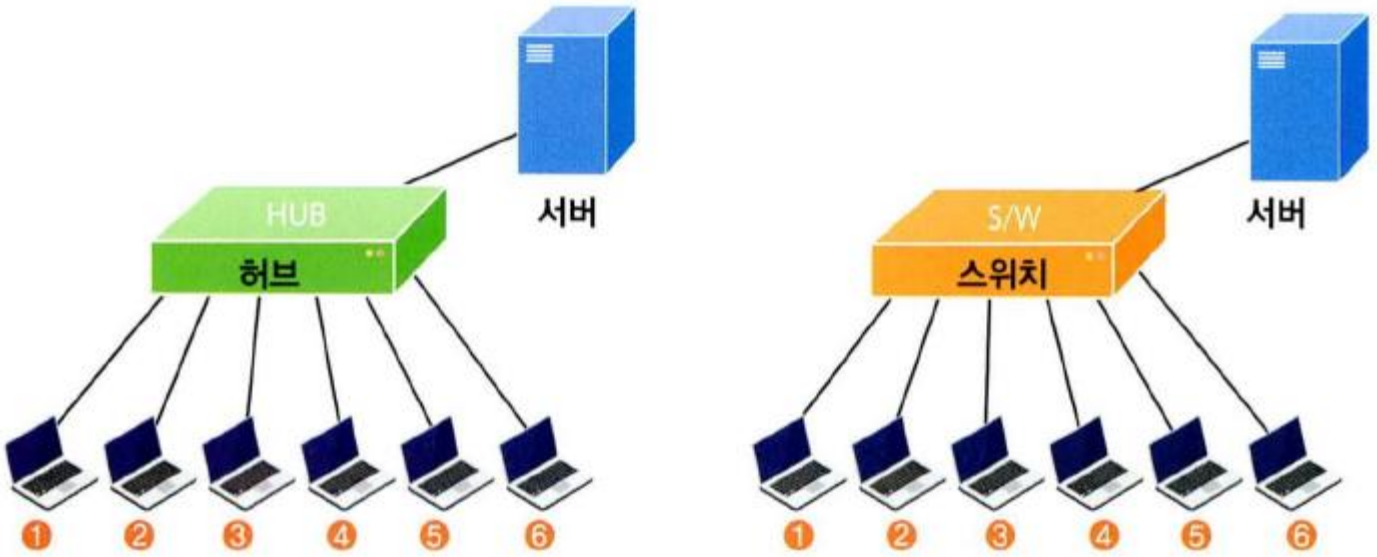
여러 개의 노드에서
동시 통신이 가능하다.

5. 스위치

- 하지만 허브는 허브대로 장점이 있다.
- 일단 스위치보다 싸다는 것이 가장 큰 장점이다.
- 또 데이터 처리 속도가 일반적으로 스위치에 비해 빠르다.
- 그럴 수밖에 없는 게 들어온 데이터에게 별로 해줄 일이 없기 때문이다.
- 들어오는 대로 그냥 내보내기만 하면 된다.
- 또 한 가지는 스위치를 사용하는 게 좋으냐, 허브를 사용하는 게 좋으냐를 결정할 때 그 네트워크에서 어떤 데이터가 돌아다니느냐 하는 것도 알아두어야 한다.
- 예를 들어 채팅이나 메일 정도를 쓰는 경우는 네트워크상에 트래픽이 적기 때문에 PC들을 스위치에 붙이는 건 아무래도 낭비일 것이다.
- 이 정도의 트래픽이라면 허브로도 충분하다.

5. 스위치

- 허브와 스위치로 구성된 네트워크



6. 브리지

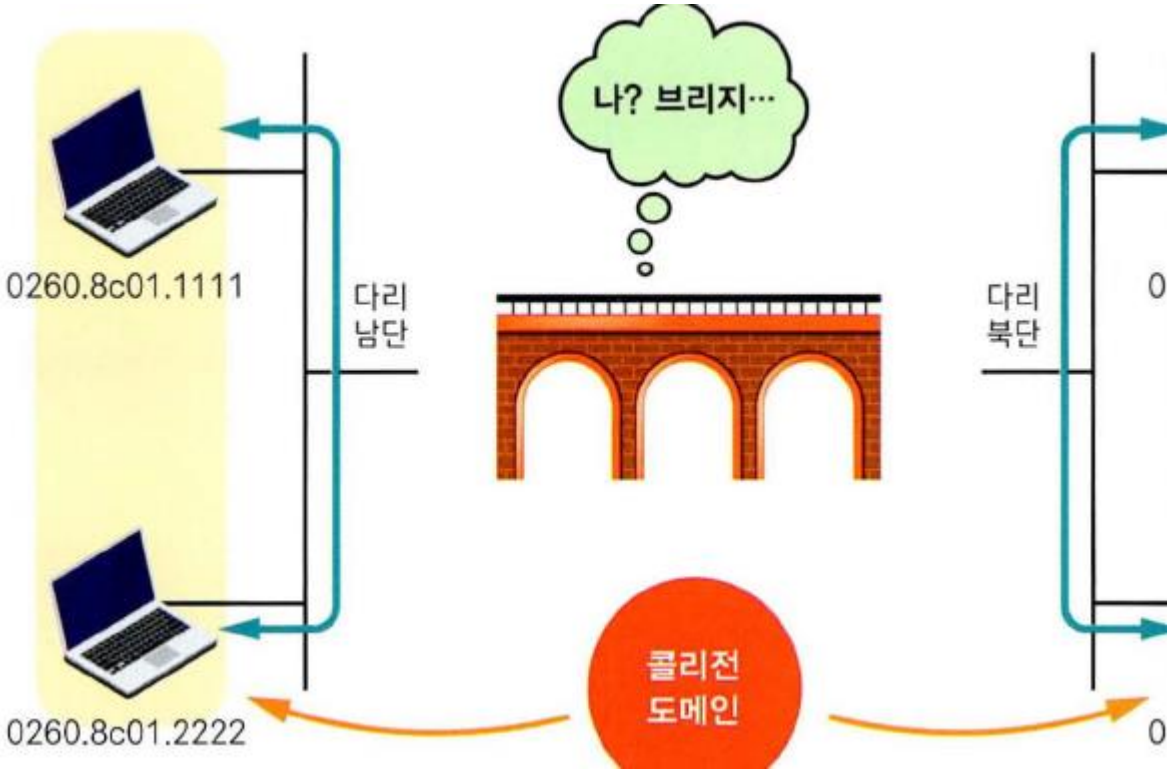
- 허브는 모든 PC들이 하나에 붙어있기 때문에 이 허브로 연결된 PC들 간의 통신에는 문제가 생길 수 있다.
- 어떤 문제냐면, 허브에 연결된 PC들 중 두 PC만 통신을 하게 되면 그 순간에는 다른 PC의 통신이 불가능하다.
- 충돌 문제는 앞에서도 몇 번 말씀드렸지만 작은 규모의 네트워크에서는 문제가 안 되지만, 네트워크의 규모가 조금만 커지게 되면 말썽을 일으키게 된다.
- 그렇다면 이렇게 네트워크 규모가 커지고 통신량이 증가할 때 충돌 도메인을 나누어 주기 위해 무엇을 사용해야 할까?
- 이때는 허브로써는 감당이 안 되기 때문에 한 수 높은 스위치나 브리지를 사용해야 한다.
- 브리지와 스위치는 사촌간이라고 볼 수 있다.
- 하는 일이 서로 비슷하다.

6. 브리지

- 그럼 브리지는 이런 충돌 도메인을 어떻게 나눠줄까?
- 브리지는 허브로 만들어진 충돌 도메인 사이를 반으로 나누고 중간에 다리를 놓는다.
- 그렇게 되면 다리 남단은 다리 남단끼리, 다리 북단은 다리 북단끼리 동시에 통신이 가능하게 된다.
- 즉 다리 남단에서 두 PC 간에 통신이 일어나는 사이에 다리 북단에 있는 PC들끼리도 통신이 가능하다는 것이다.
- 그리고 만약 다리 남단에 있는 PC와 다리 북단에 있는 PC가 통신하고자 하는 경우에만 다리를 건너서 통신이 이루어진다.
- 이것이 바로 브리지이다.

6. 브리지

- 그림을 살펴보면 0260.8c01.1111 이라는 맥 어드레스 (MAC Address)를 가진 PC가 0260.8c01.2222라는 맥 어드레스를 가진 PC에게 통신을 하는 중에도 0260.8c01.3333이라는 맥 어드레스를 가진 PC는 0260.8c01.4444라는 맥 어드레스를 가진 PC와 통신이 가능하다.



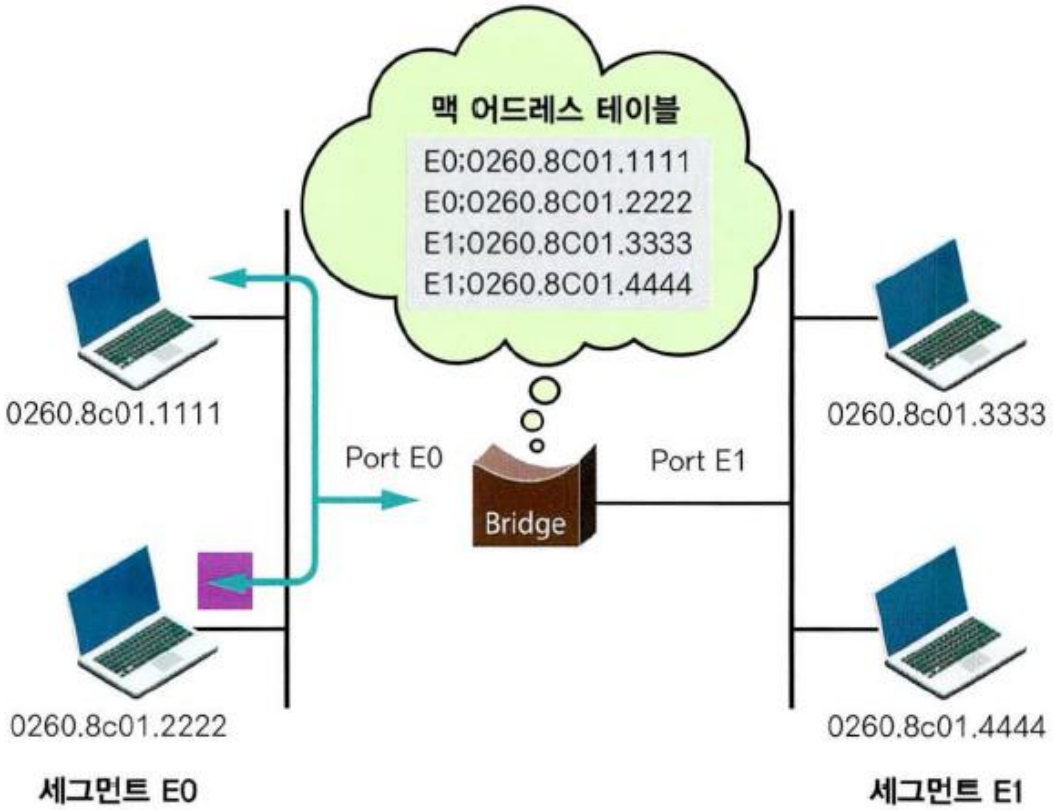
7. 브리지/스위치의 기능

- 브리지나 스위치는 다음 5가지 일을 한다.
  - Learning, 배운다
  - Flooding, 모르면 들어온 포트를 제외한 다른 모든 포트로 뿌린다.
  - Forwarding, 해당 포트로 건네준다.
  - Filtering, 다른 포트로는 못 건너가게 막는다.
  - Aging, 나이를 먹는다.

7. 브리지/스위치의 기능

Learning

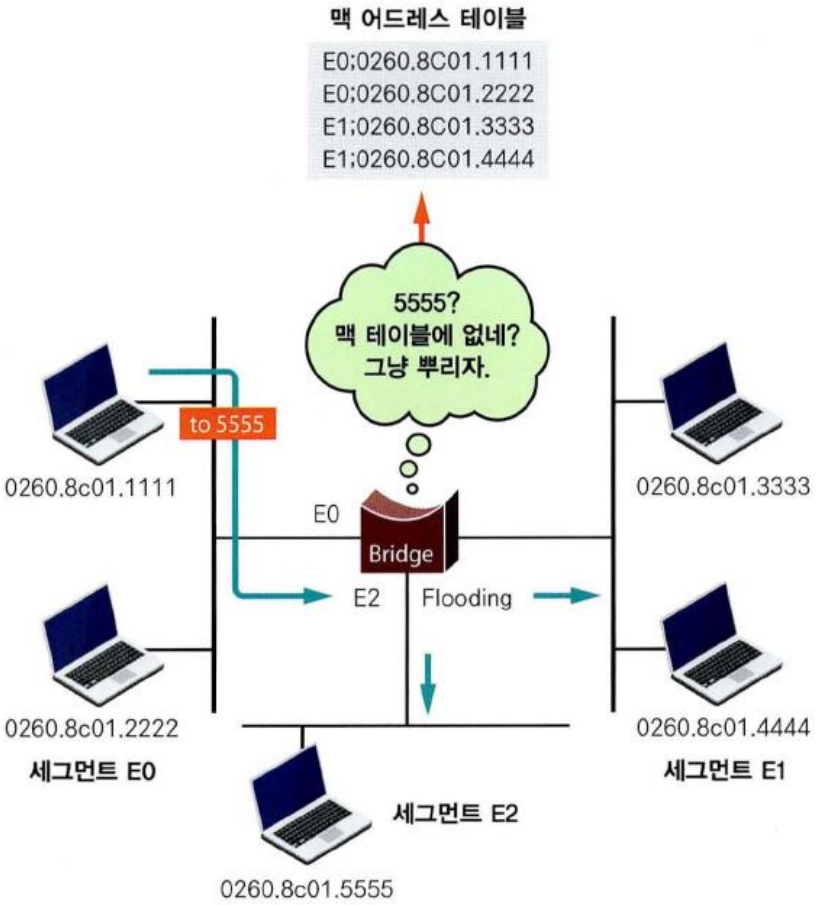
- 브리지나 스위치는 자신의 포트에 연결된 'A'라는 PC가 통신을 위해서 프레임을 내보내면 그때 이 PC의 맥 주소를 읽어서 자신의 맥 주소 테이블('브리지 테이블'이라고 한다.)에 저장한다.
- 그리고 나중에 어떤 PC가 'A'에게 통신할 경우에는 자신의 브리지 테이블을 참고해서 다리를 건너게 할 것인지 아니면 못 건너가게 할 것인지를 결정한다.



7. 브리지/스위치의 기능

Flooding

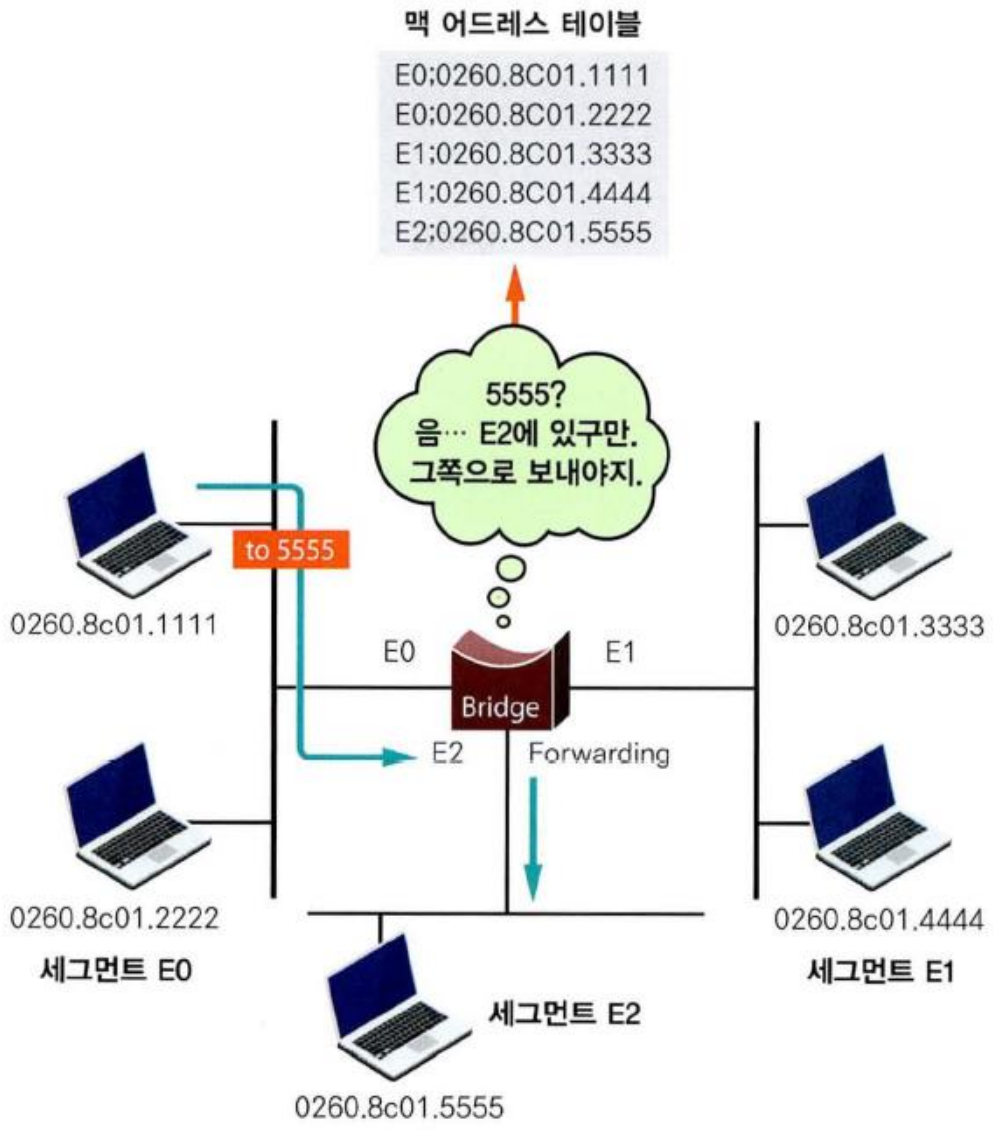
- 그냥 들어온 포트를 제외한 나머지를 모든 포트로 뿌리는 것을 의미한다.
- 들어온 프레임이 찾아가는 주소를 보니 그 주소가 만약 브리지가 가지고 있는 브리지 테이블에 없는 주소라면 어떻게 할까?
- 이때 사용되는 것이 바로 Flooding이다.



7. 브리지/스위치의 기능

Forwarding(포워딩)

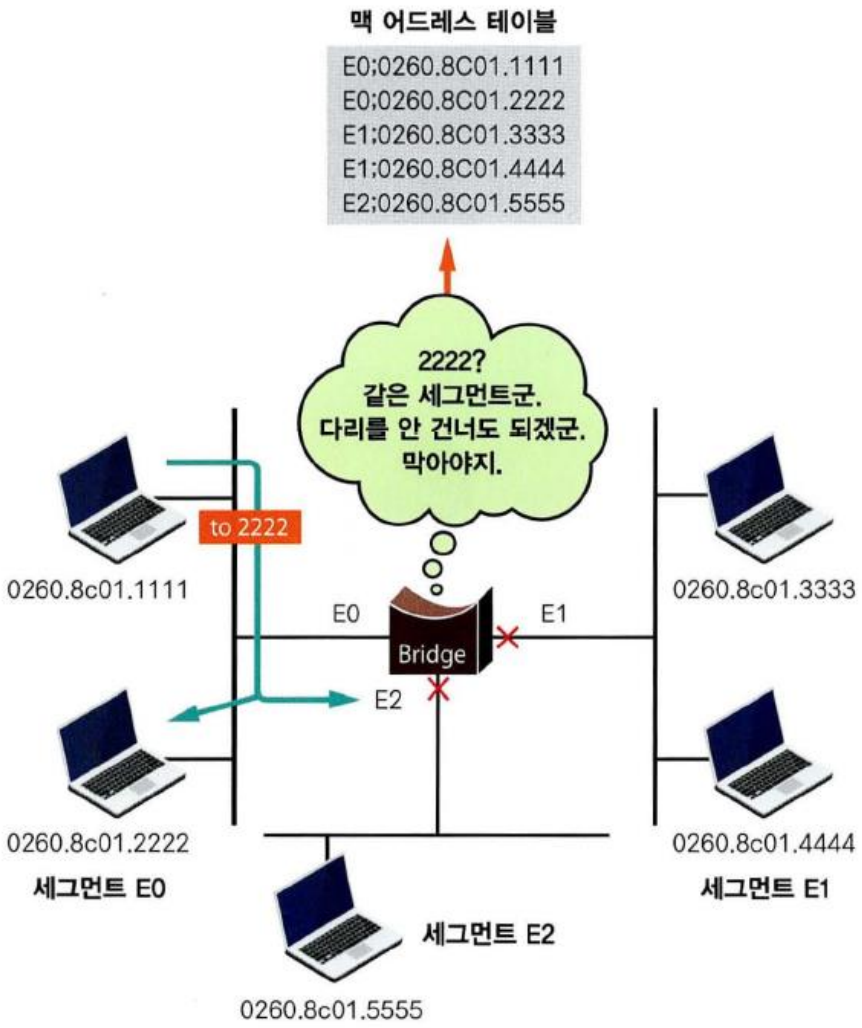
- Forwarding은 브리지가 목적지의 맥 어드레스를 자신의 브리지 테이블에 가지고 있고 이 목적지가 출발지의 맥 주소와 다른 세그먼트에 존재하는 경우에 일어난다.
- 한마디로 목적지가 어디 있는지를 알고 있는데 그 목적지가 다리를 건너가야만 하는 경우에 Forwarding이 발생한다.
- Forwarding은 이전에 배운 Flooding이 모든 포트로 프레임을 뿌리는 것과는 달리 오직 해당 포트쪽으로만 프레임을 뿌려준다.



7. 브리지/스위치의 기능

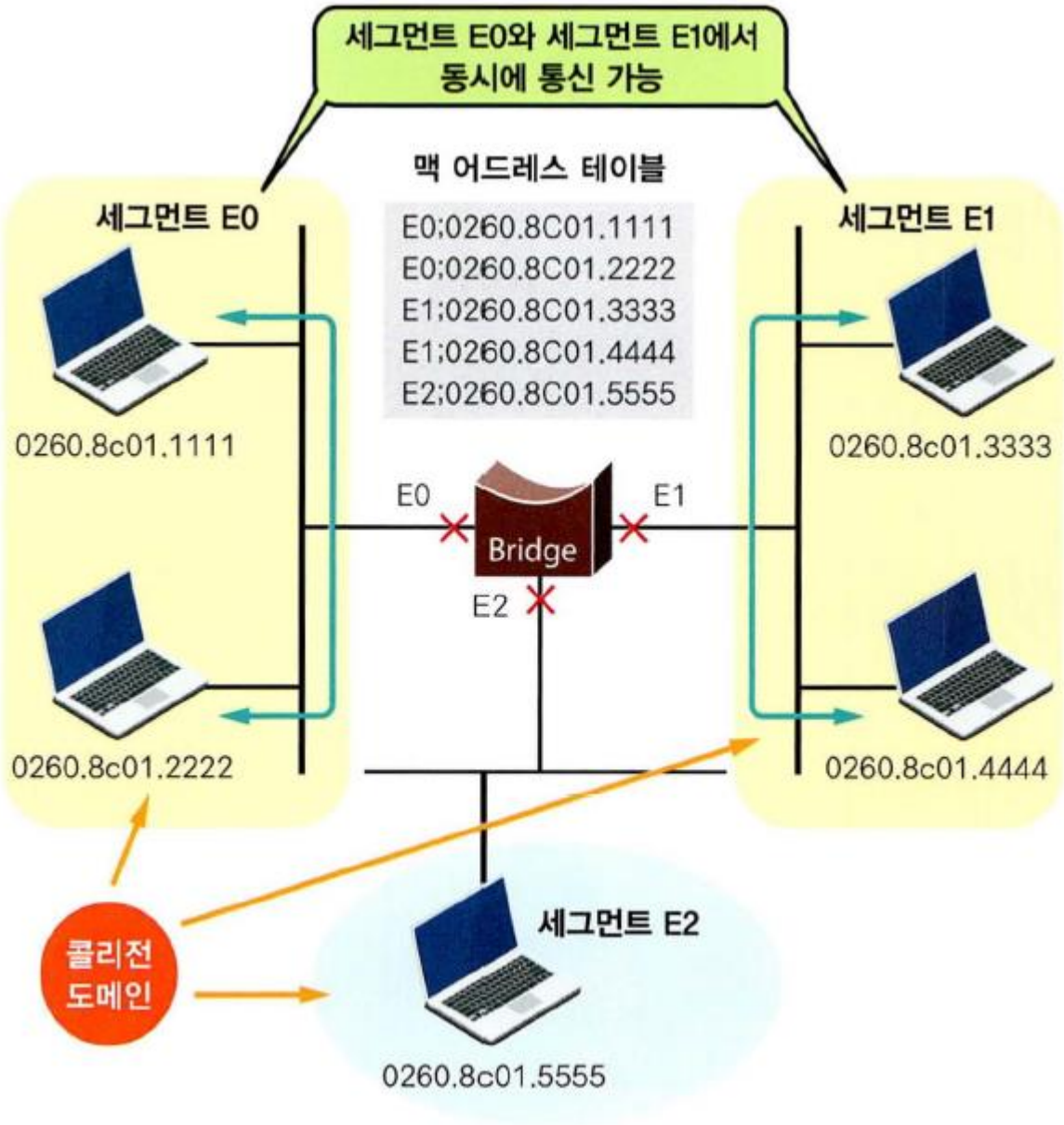
Filtering(필터링)

- Filtering(필터링)은 브리지를 못 넘어가게 막는다는 것을 뜻한다.
- 그럼 어떤 경우에 필터링이 발생할까?
- 필터링은 브리지가 목적지의 맥 주소를 알고 있고, (즉 브리지 테이블에 목적지 맥 주소가 들어있는 경우) 출발지와 목적지가 같은 세그먼트에 있는 경우이다.
- 이 경우에는 브리지를 건너가지 않아도 통신이 일어날 수 있다.
- 따라서 브리지는 다리를 막는 필터링을 실시하게 된다.
- 브리지의 이러한 Filtering(필터링) 기능 때문에 허브와는 다르게 충돌 도메인을 나누어 줄 수 있는 것이다.



7. 브리지/스위치의 기능

- **Filtering(필터링)**
  - 아래 그림을 보면 브리지의 필터링 때문에 양쪽 세그먼트에서 동시에 통신이 가능하다는 것을 알 수 있다.
  - 즉 브리지는 충돌 도메인을 이렇게 나누어 줄 수 있다.
  - 만약 이 자리에 브리지 대신 허브가 놓였다면 이런 통신은 불가능해 질 것이다.



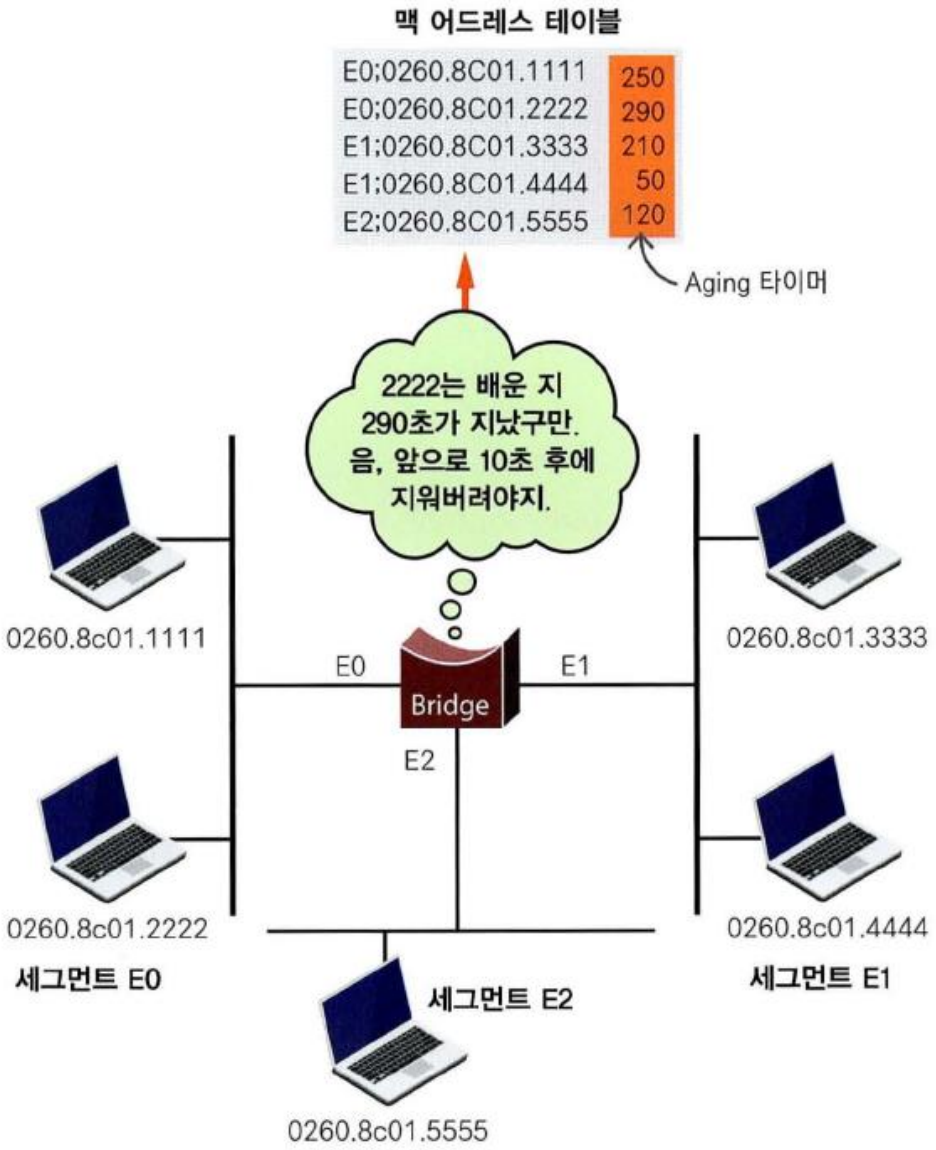
7. 브리지/스위치의 기능

Aging

- Aging이란, 말그대로 나이를 먹는다는 것이다.
- 지금까지 배운 대로 브리지는 학습 능력이 있다고 했다.
- 이것을 Learning이라고 한다.
- 브리지는 출발지의 맥 주소를 외운 후 이것을 브리지 테이블이란 곳에 저장한다고 했다.
- 그렇다면 얼마동안이나 저장할 수 있을까?
- 평생 아니면 1년 동안?
- 어차피 브리지 테이블은 한정되어 있기 때문에 평생 저장하는 것은 불가능하다.
- 만약 한 번 배운 맥 주소를 평생 저장한다면 금방 브리지 테이블이 다 차버릴 것이고, 그 다음에는 배워도 저장할 곳이 없어서 기억을 못하게 될 것이다.
- 따라서 브리지 테이블도 우리의 두뇌처럼 어느 정도 시간이 지나고 나면 이 정보를 브리지 테이블에서 지우게 된다.
- 다시 새로운 맥 어드레스를 기억해야 하기 때문이다.
- 그 시간은 디폴트로는 5분, 즉 300초이다. (물론 이 값은 조정이 가능하다.)
- Aging이란 것은 바로 이것에 관련된 타이머이다.

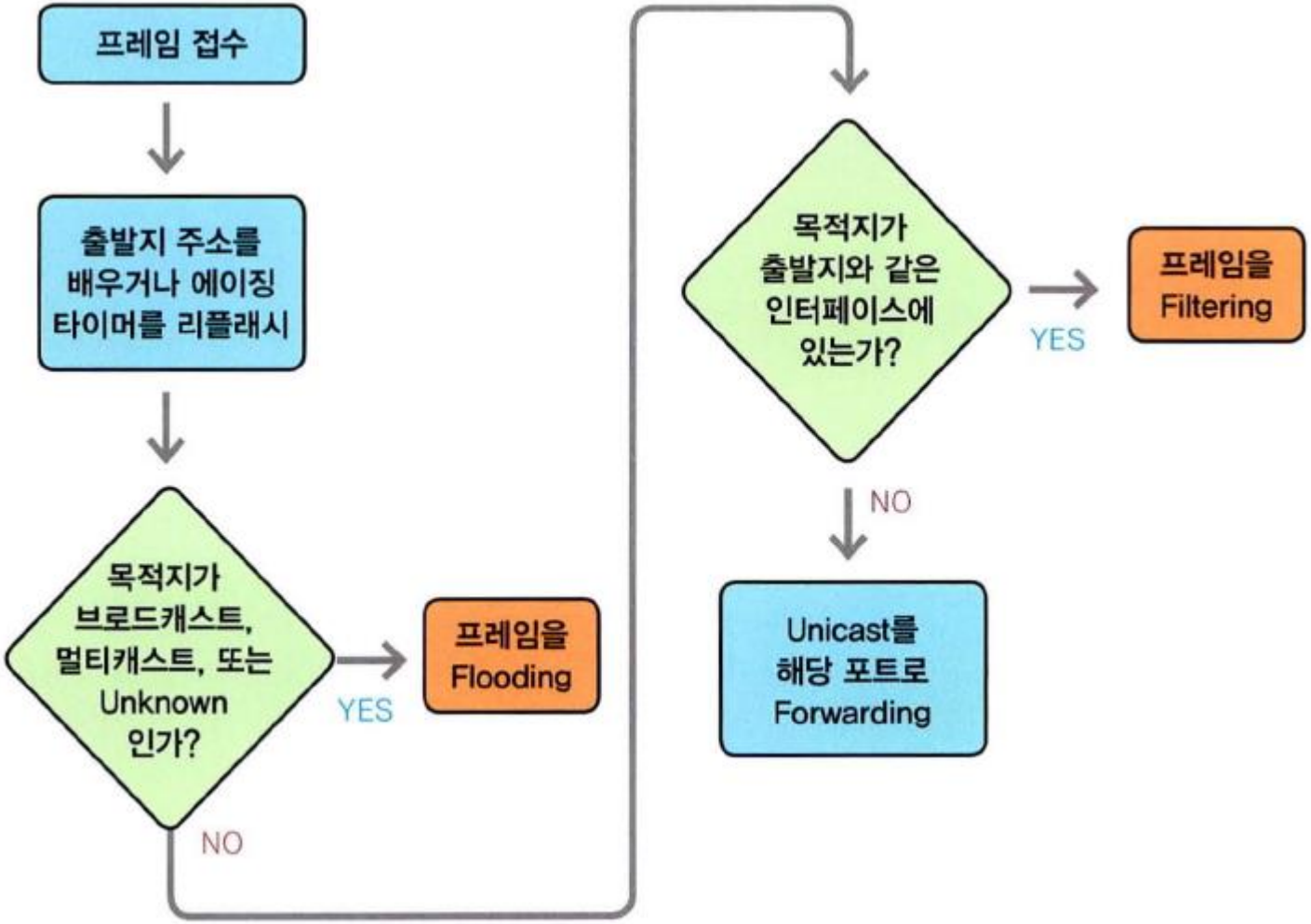
7. 브리지/스위치의 기능

- **Aging**
  - 브리지에서 Aging 타이머



7. 브리지/스위치의 기능

■ 브리지에서 프레임의 흐름



7. 브리지/스위치의 기능

브리지와 스위치의 차이점

① 스위치는 처리 방식이 하드웨어로 이루어지기 때문에 소프트웨어적으로 프레임을 처리하는 브리지에 비해서 훨씬 빠르다는 차이점이 있다. 즉 브리지의 경우는 프레임의 처리 방식이 소프트웨어적 프로그램에 의해서 처리되는 방식을 취하지만, 스위치의 경우는 처리 절차를 미리 칩에 구워서 하드웨어 방식으로 만드는 ASIC 방식이기 때문에 프레임 처리 속도가 브리지에 비해서 훨씬 빠르다.

② 브리지는 포트들이 같은 속도를 지원하는 반면, 스위치는 서로 다른 속도를 연결해줄 수 있는 기능을 제공한다. 예를 들어 스위치는 10메가 포트와 100메가 포트가 한 장비에 같이 있게 되는데, 이는 서로 다른 속도를 연결해주는 기능을 수행한다.

③ 스위치는 브리지에 비해 제공하는 포트 수가 훨씬 많다. 즉 브리지는 대부분 2개에서 3개 정도의 포트를 가지고 있는 반면, 스위치는 몇십 또는 몇백 개의 포트를 제공할 수 있다.

④ 스위치의 경우는 Cut- through, 또는 Store-and-forward 방식을 사용하는 데 비해서 브리지는 오로지 Store-and- forward 방법만을 사용한다.

7. 브리지/스위치의 기능

브리지와 스위치의 차이점

스토어-앤-포워드(Store-and-forwarding) 방식

- 이 방식은 스위치나 브리지가 일단 들어오는 프레임을 전부 받아들인 후 처리를 시작하는 방식이다.
- 프레임을 모두 받아들이고 나서 이 프레임이 제대로 다 들어왔는지, 에러는 없는지, 또 출발지 주소는 어디인지, 목적지 주소는 어디인지를 파악해서 처리를 해주는 방식이다.
- 만약 이때 에러가 발견되면 브리지나 스위치는 이 프레임을 버리고 재전송을 요구하기 때문에 에러 복구 능력이 뛰어나다.
- 따라서 이런 방식은 회선에 에러가 자주 발생하거나 출발지와 목적지의 전송 매체가 다른 경우에는 자주 사용되는 방식이다.

컷스루(Cut-through) 방식

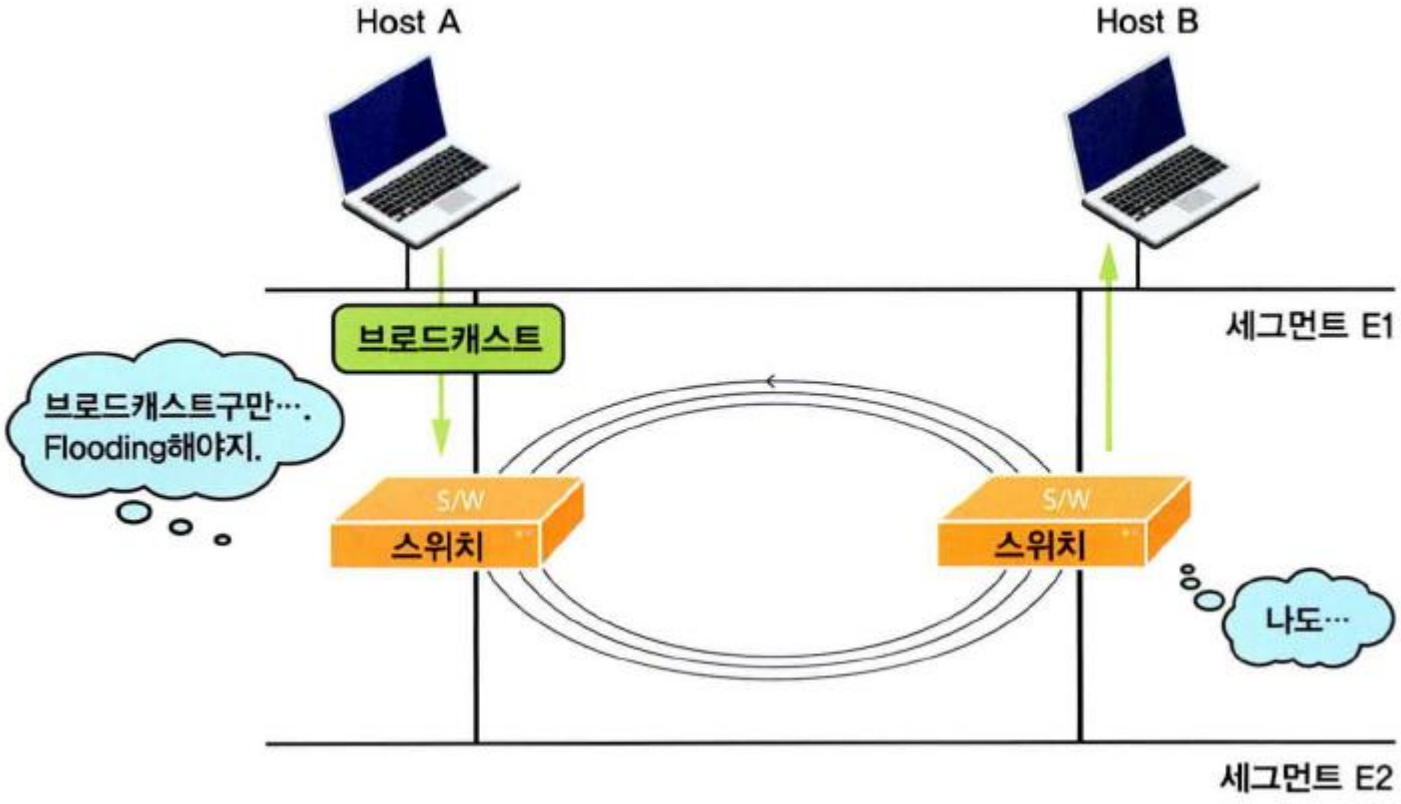
- 이 방식은 스위치가 들어오는 프레임의 목적지 주소만 본 후 바로 전송 처리를 시작하는 방식이다.
- 따라서 앞에서 배운 Store-and-forward 방식처럼 프레임이 다 들어오기를 기다리지 않고 앞에 들어오는 목적지 주소만을 본 후 바로 목적지로 전송하기 때문에 처음 48비트만을 보게 된다.
- 따라서 이전 방식에 비해서 훨씬 빨리 처리한다는 장점을 가지고 있지만, 프레임에 발생됐을지도 모를 에러를 찾아내기가 어렵기 때문에 에러 복구 능력에는 약점을 가지고 있다.
- 이 2가지 방식의 장점만을 결합한 또 다른 스위칭 방식이 있는데, 이것을 'Fragment-free 방식'이라고 한다.

프래그먼트-프리 (Fragment-free) 방식

- 이 방식은 앞에서 배운 Store-and-forward 방식과 Cut-through 방식의 장점을 결합한 방식이다.
- 즉 전체 프레임이 다 들어올 때까지 기다릴 필요가 없다는 측면에서는 Cut-through 방식을 닮았지만, 컷스루처럼 처음 48비트만을 보는 것이 아니라 처음 512비트를 보게 된다.
- 따라서 에러 감지 능력이 컷스루에 비해서는 우수하다고 할 수 있다.

8. Looping

- 루핑(Looping)은 프레임이 네트워크상에서 무한정으로 뱅뱅 돌기 때문에 이더넷의 특성상 네트워크가 조용해야 데이터를 전송할 수 있는 다른 녀석들이 계속 네트워크가 조용해지기를 기다리기만 할 뿐 데이터 전송은 불기능해지는 상태를 말하는데, 브리지나 스위치의 디자인에서는 가장 주의해야 할 사항이다.



8. Looping

- 루핑이 발생되면 물론 다른 데이터를 전송할 수가 없다.
- CSMA/CD의 특성 상 한 세그먼트 안에서 어느 한 순간에는 오직 한 녀석만이 통신을 할 수 있다는 규칙 때문에 그렇다.
- 따라서 네트워크가 무용지물 상태로 빠지게 된다.
- 이와 같이 루핑은 네트워크를 치명적인 상태에 빠뜨릴 수 있다.
- 자, 그렇다면 이런 루핑은 어떻게 하면 막을 수 있을까?
- 물론 사람이 네트워크를 구성하면서 모든 목적지의 경로를 하나만 있도록 만들어주면 아예 루핑은 생기지 않을 것이다.
- 하지만 늘 사람이 모든 걸 해줄수는 없다.
- 또 일부러 연결을 이중으로 하는 경우도 있으니까 무조건 못하게 하는 것도 문제가 있다.
- 따라서 자동으로 루핑을 막아주는 알고리즘이 필요한데 이 알고리즘을 '스패닝 트리 알고리즘(Spanning Tree Algorithm)'이라고 한다.

9. 폴트 톨러런트(Fault Tolerant)와 로드 밸런싱(Load Balancing)

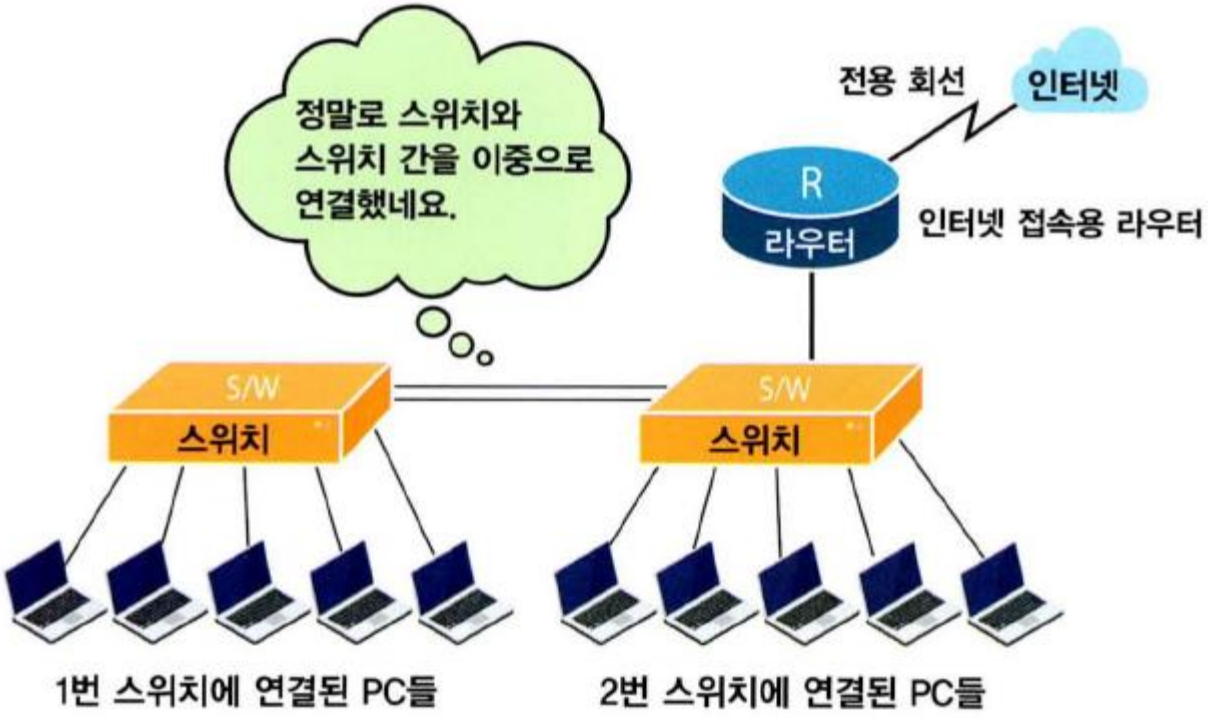
- 폴트 톨러런트란, 장애 대비책으로 대부분 이중 구조를 의미하고 전체 네트워크가 하나의 지점에서 발생한 장애로 인해 영향을 받는 것을 방지하기 위한 대책이다.
- 네트워크를 디자인할 때 폴트 톨러런트는 꼭 잊지 말아야 할 이슈이다.
- 로드 밸런싱은 말 그대로 로드를 분산하는 것이다.
- 예를 들어 하나의 인터넷 회선을 이용한 인터넷 접속 대신 두 개의 인터넷 회선을 사용하는 것이다.
- 이렇게 되면 데이터가 두 라인 중 하나를 선택해서 이용하기 때문에 로드가 분산되는 효과를 얻을 수 있다.
- 즉 속도가 두 배가 되는 것이다.
- 그러다가 회선 하나가 끊어지면 다른 회선으로 이전할 수 있는데, 이럴 경우 로드 밸런싱과 폴트 톨러런트를 겸하게 된다.

10. 스패닝 트리 알고리즘

- 만약 어떤 사람이 게임방에 두 대의 스위치를 설치했다.
- 그럼 라우터는 스위치의 포트 중 하나에 연결되고 스위치에서는 각 PC들이 연결될 것이다.
- 또 서로 간의 통신이 이루어져야 하기 때문에 두 대의 스위치 간에도 연결을 해야 한다.
- 자, 이때 이 사람이 스위치 간의 연결이 하나밖에 없으면 속도도 느리고, 또 혹시 이 연결이 끊어지면 그 스위치에 붙어있는 PC들이 통신을 못하니까 스위치 간에 링크(연결)를 두 개로 만들었다.

10. 스패닝 트리 알고리즘

- 그럼 어떻게 될까?
- 주인 생각대로 속도가 두 배로 빨라지고(스위치 간에 연결이 두 개니까) 또 하나의 링크가 끊어져도 다른 하나가 살아있으니까 문제가 없도록 해줄까?
- 정답은 '아니오'이다.



10. 스패닝 트리 알고리즘

- 자, 이때 스패닝 트리가 세팅되어 있으면 스패닝 트리는 지동으로 루핑을 검색해서 이런 루핑이 발생할 수 있는 상황을 미리 막아주는 역할을 한다.
- 어떻게 그렇게 만들까?
- 그건 스위치 간의 두 개의 링크 중 하나를 끊어 놓는 것이다.
- 따라서 실제 링크는 두 개이지만 데이터는 한 쪽으로만 다니게 하는 것이다.
- 그럼 루핑은 발생하지 않을 것이다.
- 자, 그렇다면 무엇 때문에 링크를 하나 더 연결할까?
- 이 링크는 지금 사용하는 하나의 링크가 끊어졌을 때를 대비하는 것이다.
- 만약 사용중인 링크가 끊어지게 되면 그 때 대기하던 나머지 하나가 살아나서 데이터 전송을 맡아준다.

10. 스패닝 트리 알고리즘

- 스패닝 트리 알고리즘(Spanning Tree Algorithm) 이란, 스위치나 브리지에서 발생할 수 있는 루핑을 미리 막기 위해 두 개 이상의 경로가 발생하면 하나를 제외하고 나머지 경로를 자동으로 막아두었다가 기존 경로에 문제가 생기면 막아놓은 경로를 풀어서 데이터를 전송하는 알고리즘이다.
- 모든 스위치는 이 스패닝 트리 알고리즘을 지원한다.
- 참고로 스패닝 트리 알고리즘에 의해서 현재의 링크가 끊어졌을 때 대기하고 있던 다른 링크가 다시 살아나서 연결을 해주는 데 걸리는 시간은 약 1분 이상이 소요된다.
- 그러니까 사용자들은 1분 이상을 네트워크가 끊어진 상태로 기다려야만 한다.
- 따라서 요즘의 스위치들은 여러 가지 다양한 기능을 가지고서 이러한 전통적인 스패닝 트리 알고리 즘의 약점을 보완하고 있다.

10. 스패닝 트리 알고리즘

- 예를 들어 시스코의 이더 채널 (Ether-Channel) 기술은 여러 개의 링크가 마치 하나의 링크처럼 인식되게 하는 기술이다.
- 따라서 게임방 주인이 이더 채널이 지원되는 스위치를 구매했다면 평소에도 두 배의 속도를 낼 뿐만 아니라 하나의 링크가 끊어져도 기다리는 시간이 전혀 없이 링크가 유지되는 장점이 있다.
- 이러한 이더 채널은 속도에 따라서 패스트 이더 채널(Fastether Channel)과 기가 이더 채널(Giga Ether Channel) 등이 있고 최대 8개의 링크를 묶어서 만들 수 있게 되어 있다.
- 또 업링크 패스트(Uplink Fast)라는 기술은 전통적인 스패닝 트리에서 링크의 복구 시간이 1분 이상 걸리는 점에 착안해서 이 복구 시간을 약 2~3초 안에 가능하도록 만든 기술이다.

11. 라우팅/스위칭

- **왜 라우팅이 필요할까?**
  - 가격: 라우터가 스위치보다 비싸다.
  - 속도: 이것도 스위치가 우세하다. 라우터는 내부에 서 처리하는 일이 많아서 스위치보다 패킷을 처리하는 속도가 느릴 수밖에 없다.
  - 구성의 편리함: 스위치가 훨씬 구성이 쉽다. 스위치는 대부분 전원만 공급해주면 사용이 가능하지만, 라우터는 그렇지 않다. 라우팅 프로토콜도 정해 주어야 하고, 네트워크도 설정해주어야 한다. 필터링이니 보안이니 정말 구성해주어야 할 것이 많다.

- **자, 그렇다면 가격 싸고, 속도 빠르고, 구성 편리한 스위치만 쓰면 되지, 뭐하러 라우터란 것을 쓸까?**

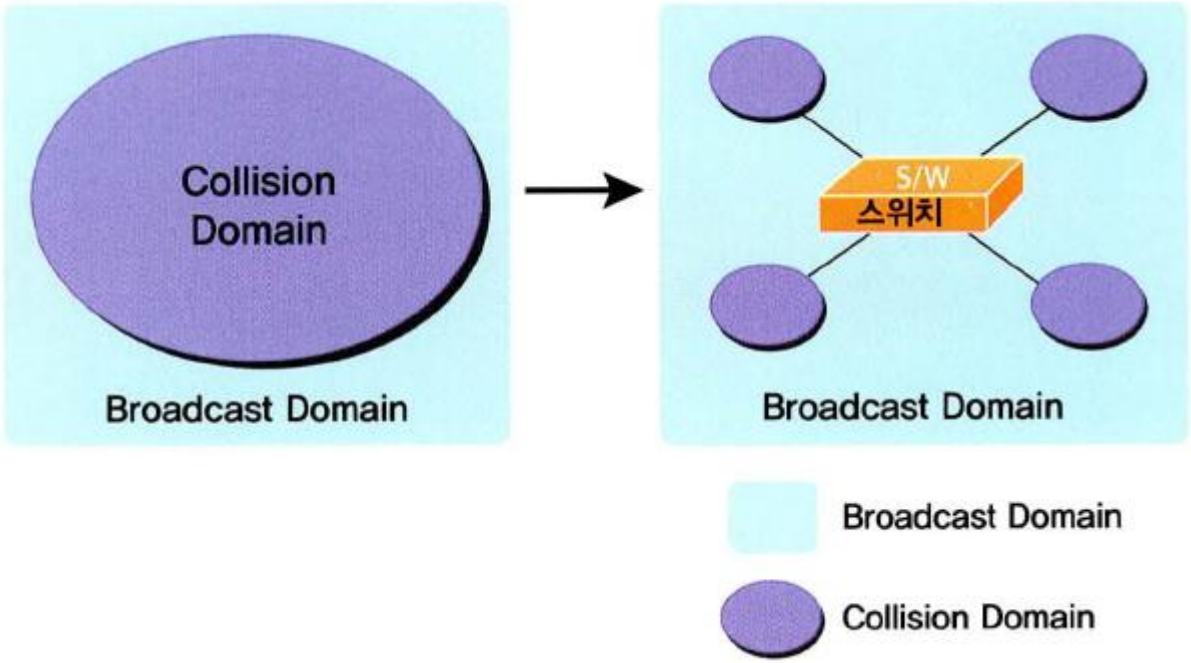
11. 라우팅/스위칭

- 바로 스위치로는 풀 수 없는 한계가 있다.
- 첫 번째 이유는 브로드캐스트이다.
- 만약 우리가 사용하는 인터넷 전체가 하나의 브로드캐스트 영역(도메인)이라고 생각해보면 어떤 일이 벌어질까?
- 저 멀리 독일에 있는 PC가 한 번 켜졌다 꺼져도 이 브로드캐스트가 우리나라에 있는 PC까지 전달된다.
- 또 통신을 할 때 상대편의 맥 주소를 찾기 위해 ARP(Address Resolution Protocol)를 사용하는데, 전 세계의 PC들이 이 ARP를 하루에 몇 번이나 사용할까?
- 만약 이런 상황이 발생한다면 우리가 제대로 네트워크를 사용하는 것은 상상도 할 수 없을 뿐만 이나라 PC 자체도 사용이 불가능해진다.

11. 라우팅/스위칭

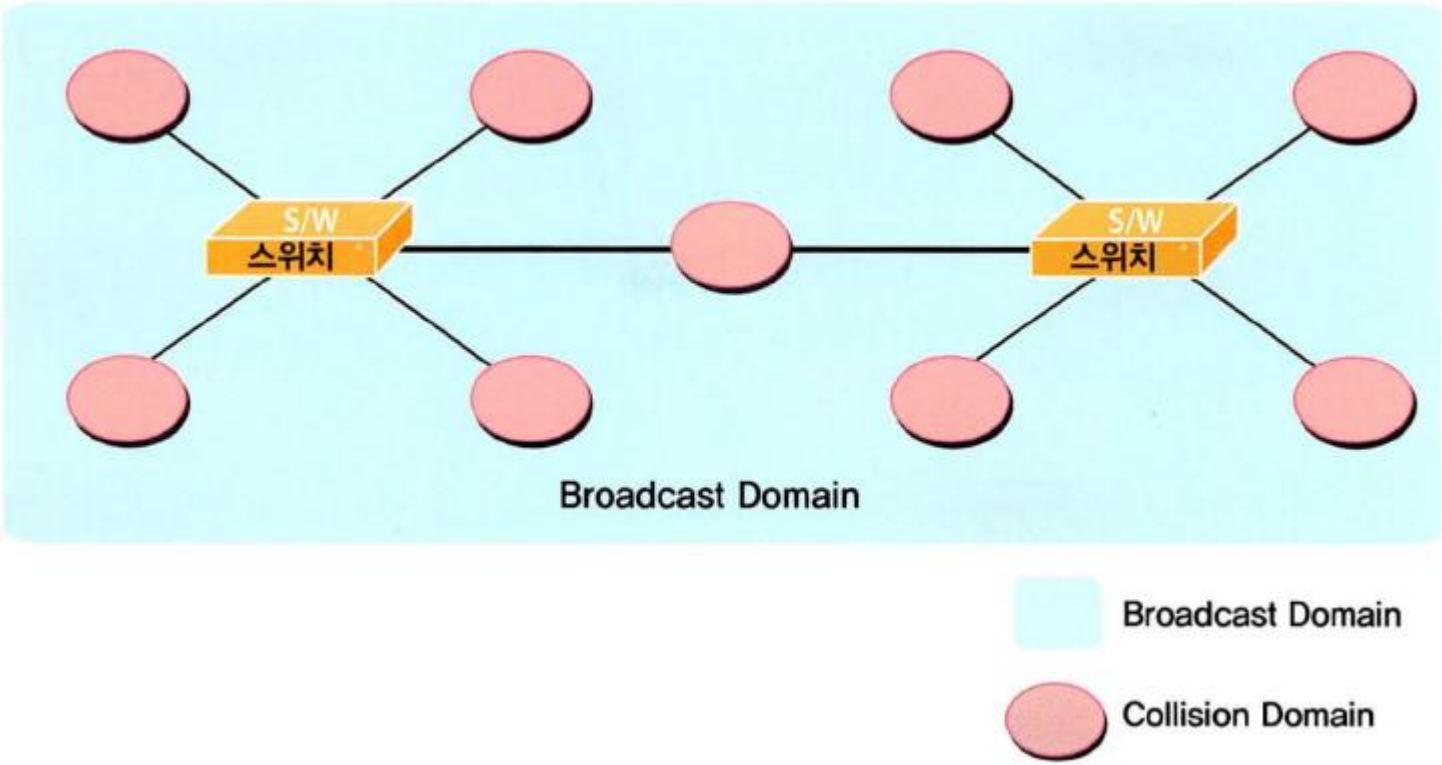
- 따라서 브로드캐스트 영역(도메인)을 나누는 것은 정말 중요한 일이다.
- 이러한 브로드캐스트 영역(도메인)을 나눠주기 위해서는 라우터가 꼭 필요하다.

- 스위치를 통한 충돌 도메인 나누기



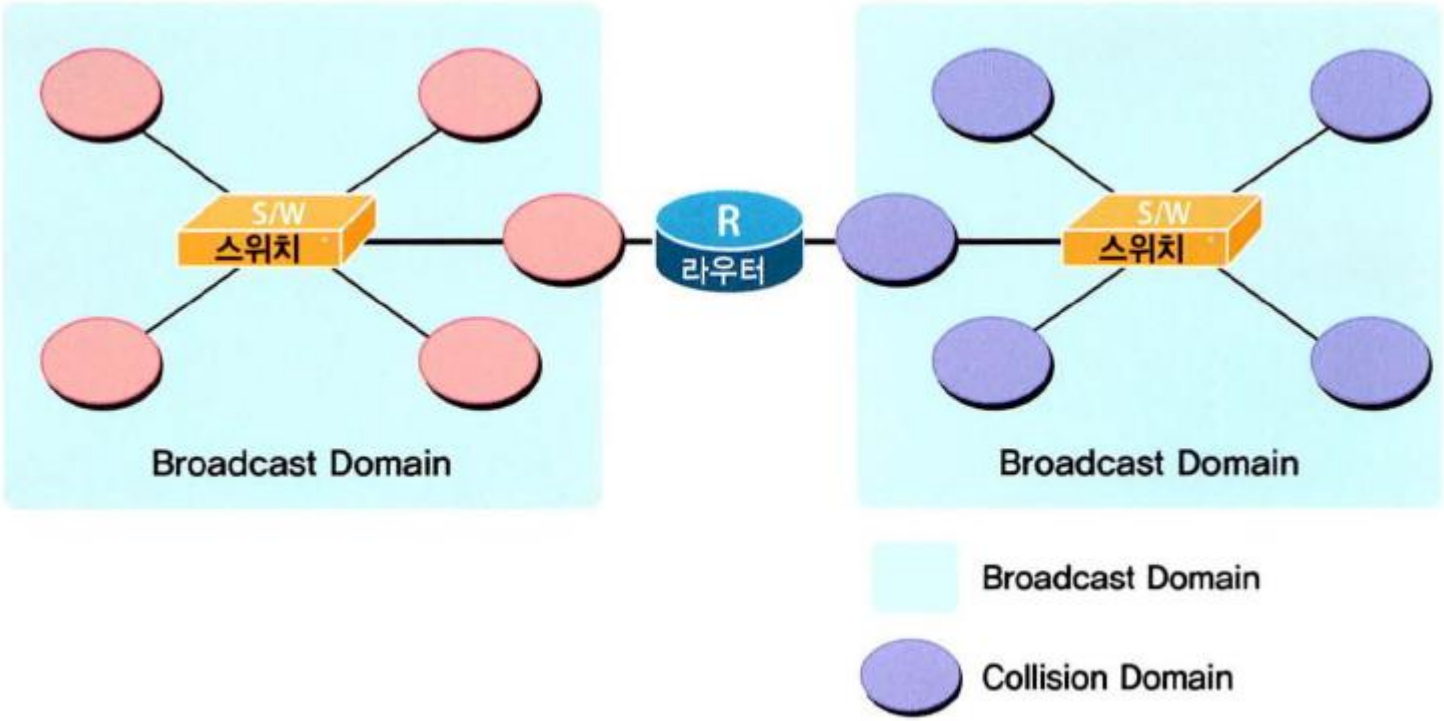
11. 라우팅/스위칭

- 그럼 이렇게 좋은 스위치를 계속 많이 구매해서 모든 네트워크를 다 스위치로 구성하면 어떻게 될까?



11. 라우팅/스위칭

- 스위치가 있던 자리에 라우터를 가져다 놓았더니 브로드캐스트 도메인이 반으로 나누어 진 것을 알 수 있다.



11. 라우팅/스위칭

- 또한 라우터는 스위치가 보장 못한 보안 기능, 즉 패킷 필터링 기능을 제공한다.
- 따라서 네트워크 주소에 따라 전송을 막았다 풀었다 하는 필터 기능을 제공해서 불필요한 트래픽이 전송되는 것을 막는다.
- 아시겠지만 이러한 보안 기능은 요즘 들어 점점 더 중요한 이슈로 떠오르고 있다
- 또 하나 라우터가 제공해주는 기능은 바로 '로드 분배'이다.
- 즉 여러 개의 경로를 가지고 있기 때문에 데이터가 여러 경로를 타고 날아갈 수 있다.
- 따라서 한쪽 경로에 문제가 생겨도 바로 다른 경로를 타고 날아갈 수 있다.
- 물론 스위치도 로드 분배가 가능하지만 이것은 굉장히 제한적이다.
- 라우터는 그 외에도 프로토콜이나 데이터의 크기 중요도 등 여러 상황에 따라 트래픽의 전송 순서를 조정해주는 QoS(Quality of Service) 기능도 제공한다.

Thank You