

IP 방송제작 기술 8

PTP와 SMPTE 타임코드 II

글. 조인준 KBS 미디어기술연구소 차장

지난 연재에서 SMPTE 타임코드에 관해 설명해 드렸습니다. SMPTE 타임코드를 설명하는 과정에서 Drop Frame Timecode와 Non-drop frame Timecode의 차이점에 관해 비중 있게 다루었습니다. 이번 연재에서는 PTP를 이용하여 SMPTE 타임코드를 생성하는 원리와 과정에 대한 간단히 설명해 드릴 것이며, 이 원리는 Drop Frame Timecode와 Non-drop frame Timecode의 생성에 모두 적용되는 것입니다. SMPTE 타임코드는 그 형식에 따라 LTC(Linear or Longitudinal Time Code)와 VITC(Video Interval Time Code)가 있습니다.



그림 1. LTC와 VITC

[그림 1]은 LTC와 VITC의 차이점을 보여줍니다. [그림 1]과 같이 LTC는 오디오 신호를 통해서 저장/전달되고, VITC는 비디오 신호를 통해 저장/전달되며, 신호의 데이터 포맷이 다르다는 것을 제외하면 PTP 타임을 이용하여 두 타임코드를 발생시킬 때, 그 생성원리는 같다고 할 수 있습니다. 그러므로 타임코드의 형식이 LTC인지 VITC인지는 논외로 하고 SMPTE 타임코드를 생성하는 원리에 관해서만 설명을 이어가겠습니다.

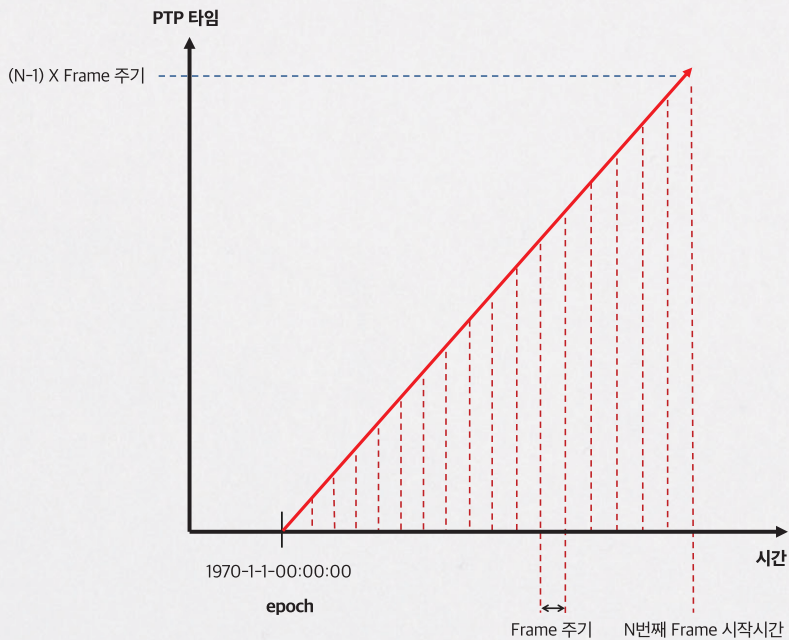


그림 2. PTP 타임과 비디오 Frame 시작시간

지난 연재에서 네트워크 내 Grandmaster와 방송 장비 (Slave Clock) 간 PTP 타임 동기화를 기반으로 epoch에서 시작된 특정 frame rate를 갖는 가상의 무한 비디오를 가정하여 **(N-1)×frame 주기**의 식을 적용하면 [그림 2]와 같이 N번째 프레임이 시작하는 PTP 타임을 간단히 알 수 있고, 이 PTP 타임에 N번째 프레임을 출력하면 자연스럽게 네트워크 내 비디오 장비 간 프레임 동기화가 이루어지게 됩니다. 프레임 동기화가 이루어지면 Grandmaster와 방송 장비(Slave Clock)의 내부 시계로 측정된 PTP 타임을 완벽히 일치하게 되고, Slave Clock의 PTP 타임을 통해서 자체적으로 현재 프레임이 epoch으로부터 몇 프레임 경과한 프레임인지 계산이 가능하며, 이 경과한 프레임 수를 기반으로 해당 프레임과 연결된 SMPTE 타임 코드를 계산할 수 있습니다. 조금 더 구체적으로 PTP 타임으로 SMPTE 타임코드를 계산하는 방법을 설명드리겠습니다. 최대한 수식의 사용을 피하기 위해 계산 절차와 개념 위주로 설명하겠습니다.

[SMPTE 타임코드 계산 절차]

1. PTP 프로토콜로 Grandmaster와 동기화된 개별 방송장비 내부 Clock이 주는 타임은 epoch으로 부터 현재까지 경과한 시간을 나타내므로, 이 시간을 비디오 프레임 주기로 나누어 소수점 이하를 버림하면 현재 출력 중인 비디오 프레임이 epoch으로부터 몇 번째 프레임인지 알 수 있고, 올림을 하면 다음에 출력할 비디오 프레임이 epoch으로부터 몇 번째 프레임인지 알 수 있다.

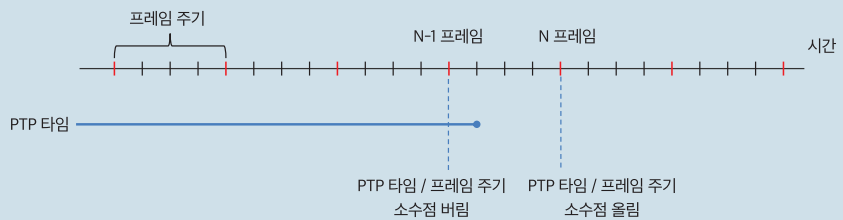


그림 3. PTP 타임과 프레임 수 계산

2. 타임코드의 계산은 타임코드의 출력에 선행하므로 위에서 설명한 계산 방법으로 다음에 출력할 비디오 프레임이 epoch으로부터 몇 번째 프레임인지 알면 비디오 프레임의 주기를 이용하여 타임코드의 시, 분, 초 및 프레임을 구할 수 있다.

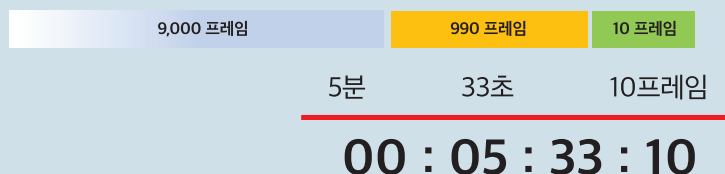


그림 4. 프레임 수와 타임코드

[그림 4]의 예를 들어 초당 30 프레임의 영상에서 다음에 출력할 비디오 프레임이 10,000번째 프레임이면, $10,000/30$ 을 통해 다음 프레임 시작 시각이 epoch으로부터 333.33333...초 경과하였으며, $333.33333... / 60 = 5.55555...$ 를 통해 epoch으로부터 5.55555...분 경과했음을 알 수 있습니다. 5.55555...분은 5분은 넘지만 6분이 지난 것은 아니므로 $5 \times 60 \text{초} \times 30 \text{ 프레임/초} = 9,000 \text{ 프레임}$ 으로 계산됩니다. 10,000 프레임에서 5분이 차지하는 프레임이 9,000 프레임이므로 $10,000 - 9,000 = 1,000 \text{ 프레임}$ 을 통해서 타임코드의 나머지 초와 프레임을 계산할 수 있습니다. **1,000 프레임 = 33초 × 30 프레임/초 + 10 프레임**이므로 10,000 프레임은 epoch의 타임코드를 00 : 00 : 00 : 00 라고 하면 00 : 05 : 33 : 10 이 됩니다.

PTP 타임을 이용해 SMPTE 타임코드를 구하는 원리는 만년 달력을 계산하는 S/W의 원리와 유사하다고 생각하면 친숙하게 느껴질 것 같습니다. 지금 예를 든 30fps와 같은 Non-drop Frame Timecode가 아닌 Drop Frame Timecode도 다소 복잡하지만 PTP 타임을 통해서 계산할 수 있습니다. 물론 이전 연재에서 설명해 드린 바와 같이 Drop Frame Timecode는 0, 10, 20, 30, 40, 50분을 제외한 모든 분의 0초에서 0과 1 두 프레임의 타임코드를 건너뛰기 때문에 계산식이 더 복잡하기는 합니다. 하지만 위에서 설명을 위해 예를 든 것과 같이 장황한 계산이 아닌 몇 개의 함수들로 깔끔하게 정리된 계산식을 사용하므로 구현이 복잡하지는 않습니다.

그렇다면 PTP 타임을 이용하여 SMPTE 타임코드를 생성하는 방법에 대한 설명이 이제 다 끝난 걸까요? 그렇다면 좋겠지만 애석하게도 사정은 더 복잡합니다. 우선 지난 연재에서도 말씀드렸듯이 초당 29.97, 59.94 등의 프레임을 사용하는 영상의 경우 Drop Frame Timecode를 사용하더라도 타임코드와 실제 시간을 완벽히 일치시킬 수 없습니다. 시간이 지날수록 타임코드와 실제 시간의 오차가 벌어져서 초단위로 타임코드와 실제 시간의 차이가 커지고, 오랜 시간 계속 방치하면 분 단위, 시간 단위까지 타임코드와 실제 시간의 차이가 차츰 벌어지게 됩니다. 그래서 SMPTE에서는 Jam이라는 것을 정의한다고 이전에 설명해 드렸습니다. Jam은 타임코드와 실제 시간의 오차가 커지기 전에 일종의 리셋과 같이 타임코드를 실제 시간에 맞추는 작업입니다. 그런데 Jam을 해야 하는 경우는 타임코드가 태생적으로 가진 오차를 리셋하는 것 이외에도 실제 시간의 연속성이 깨질 때 타임코드를 이에 맞추는 것도 포함합니다. 실제 시간에 어떤 연속성 파괴 문제가 발생할 수 있을까요? 대표적인 케이스가 윤초입니다. 윤초는 원자시계와 지구 자전에 따른 태양시의 오차를 맞추기 위해 시간에서 가감하는 1초입니다. 또한, 지금은 우리나라에서 잘 시행하지 않지만 과거에 있었던 서머타임 같은 것도 시간의 연속성에 문제를 일으킵니다. 서머타임이 무엇인지 모를 수 있는 젊은 독자들을 위해서 간단히 설명하면 여름철에 표준시보다 1시간 시계를 앞당겨 일과를 일찍 시작하게 하고, 수면에 들기 전까지 어둠이 짧아져 전기를 절약하는 효과 등을 노린 제도입니다. 서머타임을 시행하게 되면 시작하는 날에 사람들이 수동으로 시계를 한 시간 앞당기고, 해제되는 날에는 수동으로 시계를 한 시간 늦춥니다. 이런 인위적 시간의 조작으로 인해 서머타임을 실시하면 시간의 연속성이 깨질 수밖에 없습니다. 그래서 윤초나 서머타임 등의 사건이 발생하면 타임코드를 사용하는 방송국에서는 Jam 메카니즘을 이용하여 이를 극복합니다.

Jam은 타임코드와 실제 시간의 오차를 특정 순간에 리셋하는 특성을 가지므로 개별 장비들이 사전에 Jam의 시점에 관한 정보를 취득하여 미리 준비해야 합니다. 그렇다면 Slave Clock인 개별 방송 장비는 언제 Jam이 일어날지 어떻게 알고 Jam을 준비할까요? Jam이 일어나는 시점은 누군가가 그 시점을 개별 장비에 알려주지 않는 이상 개별 장비에서 Grandmaster가 전송하는 PTP 타임만 보고는 알 수 없습니다. Jam에 관련된 정보를 개별 장비에 보내는 방법 중에 가장 효율적인 방법은 시스템 관리자가 Jam 관련 정보나 스케줄을 Grandmaster에 입력하면 Grandmaster가 이를 개별 장비인 Slave Clock에

보내는 방법입니다. 이를 위해 SMPTE는 [그림 5]와 같이 매초 또는 필요할 때마다 Jam 관련 정보를 개별 장비에 보내기 위한 메시지 포맷을 정의했습니다.

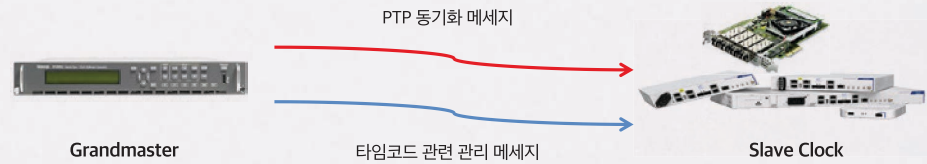


그림 5. Jam 관리를 위한 타임코드 관리 메시지

[그림 5]와 같이 Grandmaster가 PTP 동기화 메시지 이외에도 타임코드 관련 관리 메시지를 보내므로 개별 장비는 언제 Jam이 일어날지 사전에 알 수 있고, 사전에 알려진 정확한 시간에 Jam을 수행합니다. Jam 수행의 조금 더 구체적인 예를 들면 다음과 같습니다.

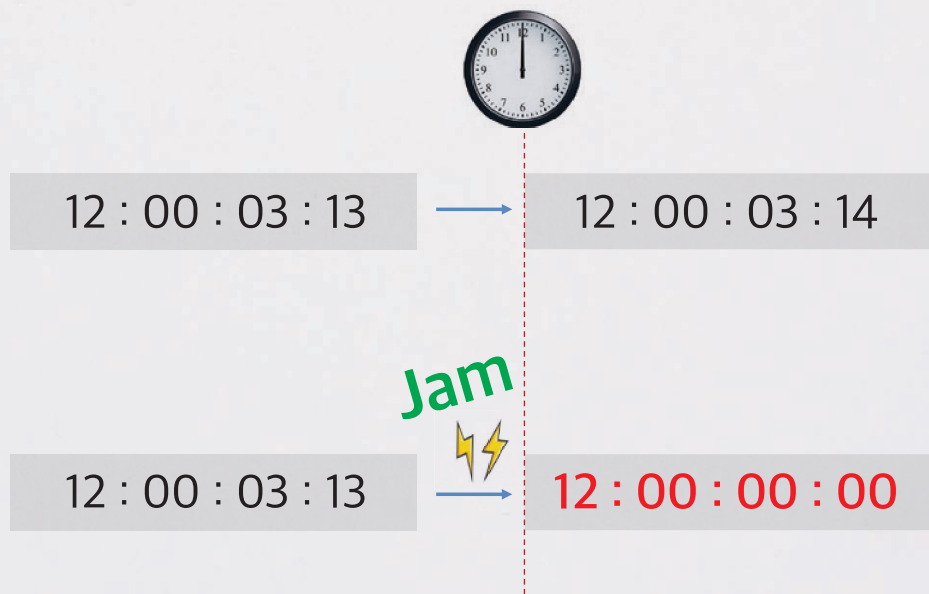


그림 6. Jam 발생에 의한 타임코드 변경

이전의 연재에서 Drop Frame Timecode를 쓰면 실제 시간과 타임코드로 나타내지는 시간에 오차가 발생할 수 밖에 없는 원인을 설명드렸습니다. 이 오차 때문에 실제 시간은 12:00:00(시:분:초)라도 타임코드는 12:00:03:14(시:분:초:프레임)와 같이 오차가 발생할 수 있습니다. 만약 PTP 타임을 보내는 Grandmaster가 PTP 타임으로 12:00:00에 Jam을 수행한다고 사전에 개별 장비에 알려 놓으면 개별 장비는 [그림 6]의 상단과 같이 12:00:03:13 → 12:00:03:14로 타임코드를 증가시키지 않고 [그림 6]의 하단과 같이 12:00:03:13 → 12:00:00:00으로 타임코드에서 3초 13 프레임을 제거합니다. Jam이라는 것이 사실 내용상으로는 대단한 것이 아니지만 윤초나 서머타임 같은 것들까지 고려하면 신경 써서 관리되어야 하는 부분 이긴 합니다.

지금까지 SMPTE 타임코드의 생성 원리를 개념 위주로 설명해 드렸습니다. 작게나마 독자 여러분께 도움이 되는 지식이 되었으면 좋겠습니다.

다음 연재에서는 IP 방송 제작시스템에서 비디오가 어떻게 IP 신호로 전달되는지에 관한 이야기를 시작해보겠습니다. 📺