SMPTE /EBU Joint Task Force: Time & synchronization

Transmission of a video synchronization signal over Ethernet/IP Network

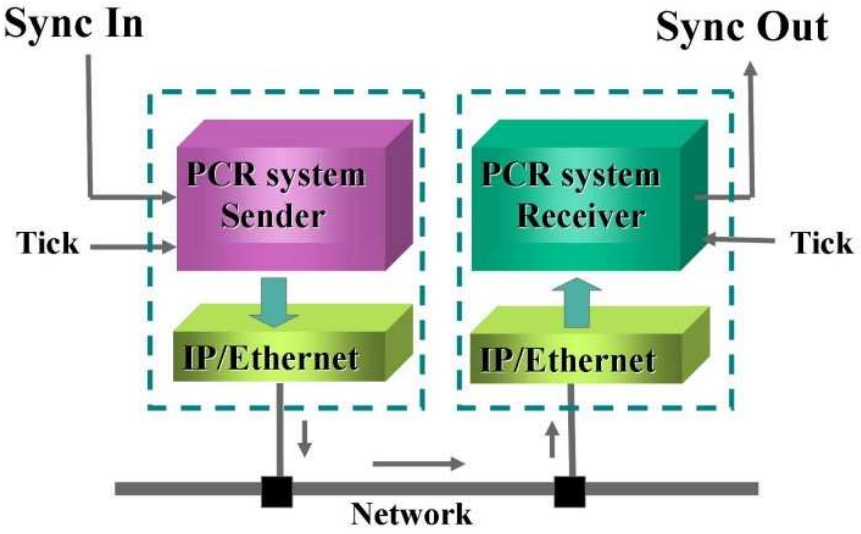
Serge DEFRANCE
Thierry TAPIE
Gaël MACÉ
CR / CP&M Lab. (Rennes / France)

802.1 AS – October 2008

THOMSON
images & beyond

Video Genlock over Ethernet/IP

- **A PCR based clock recovery application over a Ethernet/IP network.**
  - widely used in A/V professional environment



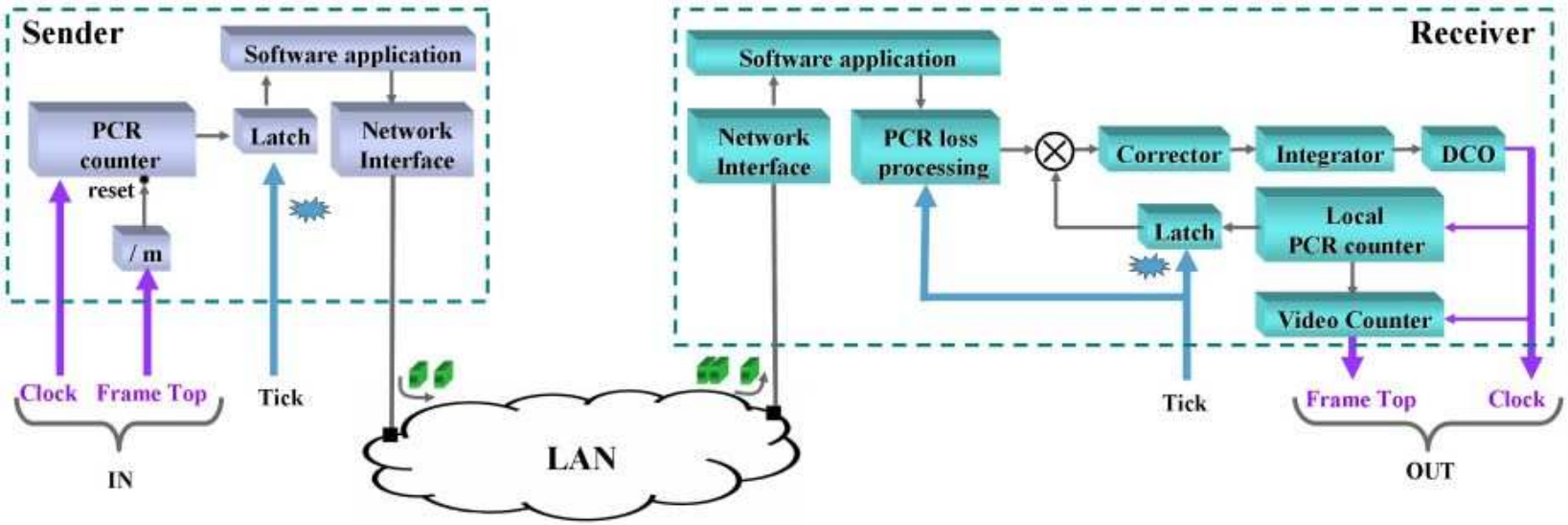
But traditional clock recovery solution using PCR technique (like MPEG Systems) is dedicated to channel transmission with **constant packet transmission delay and low jitter**.

Proposed solution (1/2)

- **A common regular tick signal.**

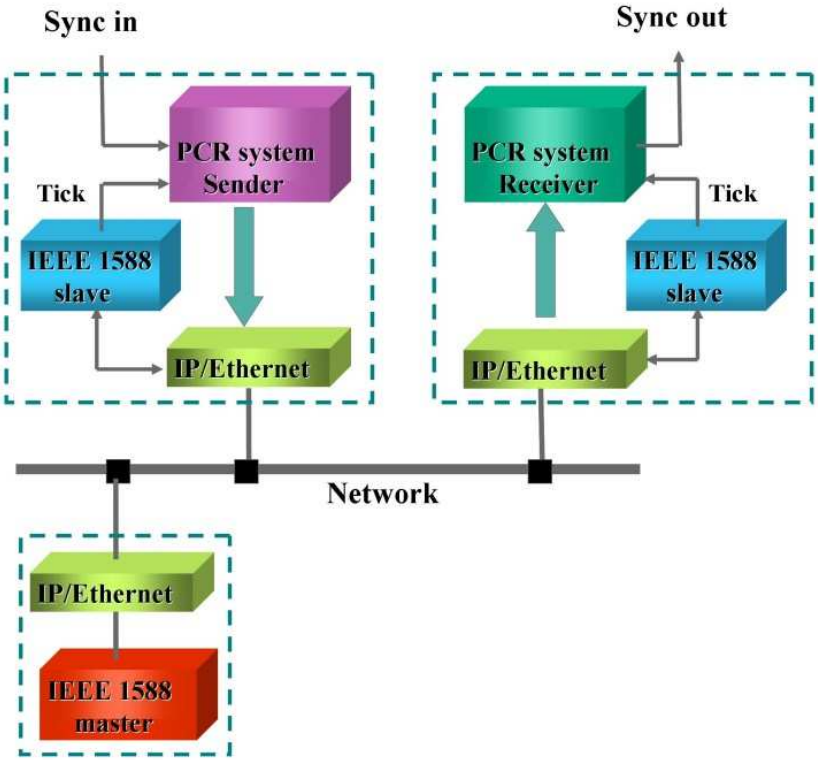
  The ticks signals have the same period and the same phase on every equipment. The ticks instants will determine the PCR instant comparison instead of PCR reception time instant comparison.

- **Asynchronous and independents Ticks and Frame Top signals.**



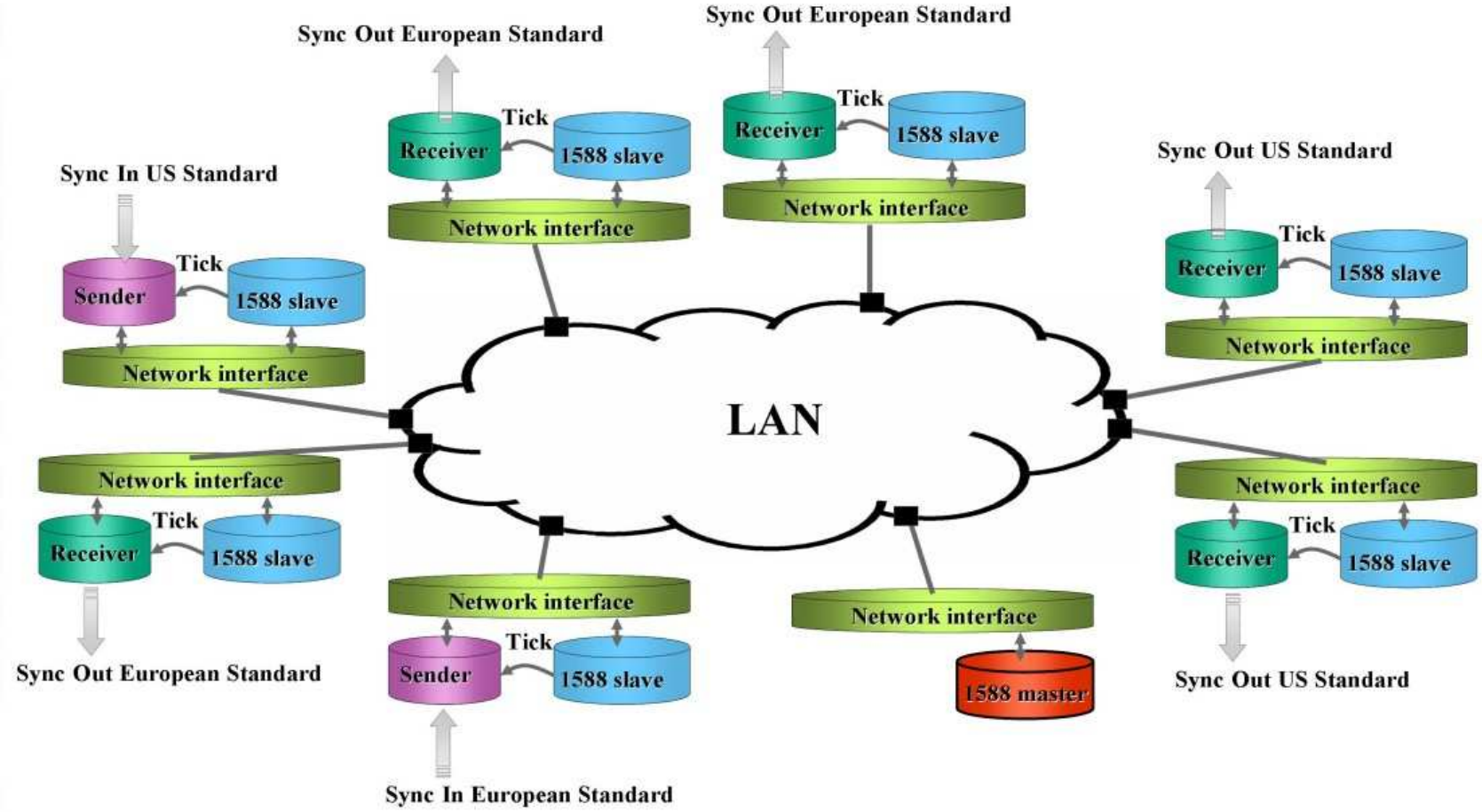
Proposed solution (2/2)

- **802.1 AS: the tick signal layer**
  - standardized synchronization service able to support the generation of a cyclic Tick signal with the same phase on each equipment
  - Even it has to be very accurate (class A application), **IEEE 802.1AS has not to cope with video requirements; this is supported by the PCR system layer.**



Multi standards transmission

Several A/V synchronizations from different standards can be transmitted at the same time. The system does not care about who is the 802.1 AS master.



802.1 AS – Application Interface for a Video Synchronization System

Serge DEFRANCE
Thierry TAPIE
Gaël MACÉ
CR / CP&M Lab. (Rennes / France)

802.1 AS - 2008

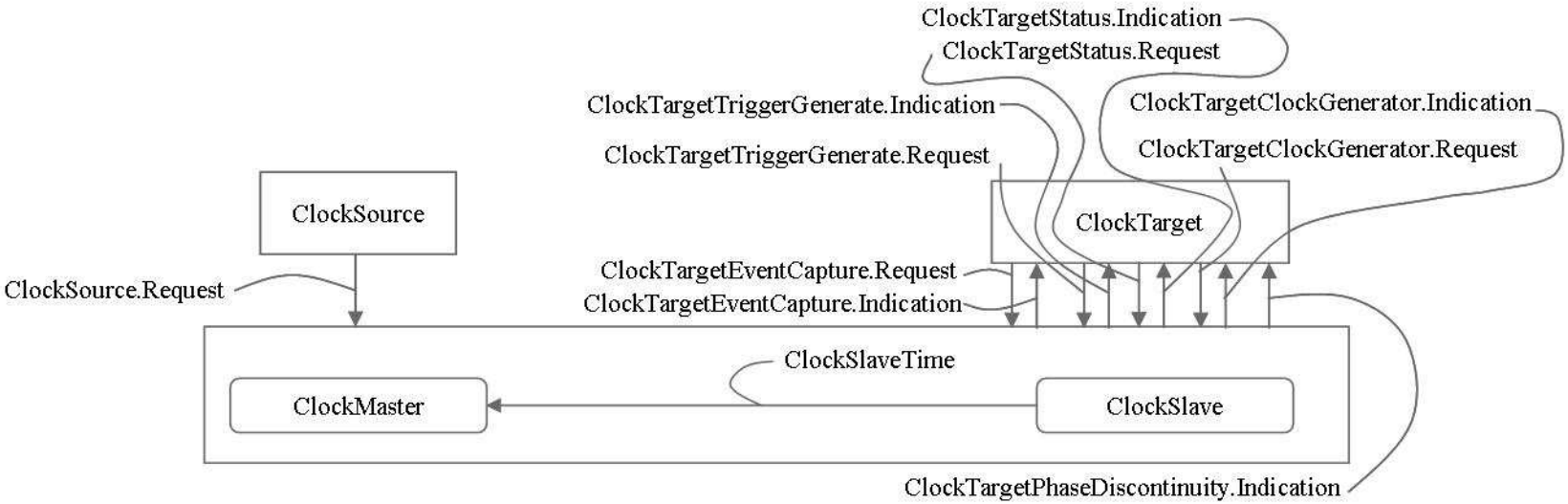
THOMSON
images & beyond

802.1 AS : Application Interface (1/2)

- **Need of APIs allowing to :**
  - Indicate accurately Coordinated Measurement Instants (CMI, sampling triggers)
  - Notify the validity of the CMI or any other error condition
  - The accuracy of the indication (related to the sourceTime specified in the request and according to the ClockSlave resolution) impacts the timing performance of the upper layers.

802.1 AS : Application Interface (2/2)

- **Current 802.1 AS D4.0 draft defines such interfaces:**



ClockTargetPhaseDiscontinuity Interface (1/2)

- **Mandatory interface** which avoid the need of disruption indication within the `ClockTargetTriggerGenerate` and `ClockTargetClockGenerator` interfaces.
- When a discontinuity occurs, any running interface call (to `ClockTargeteventCapture`, `ClockTargetTriggerGenerate` and `ClockTargetClockGenerator` shall be stopped to avoid any erroneous indication

```
ClockTargetPhaseDiscontinuity.indication {
        disruption,
        deltaPhase,
        deltaFrequency
}
```

ClockTargetPhaseDiscontinuity Interface (2/2)

```
ClockTargetPhaseDiscontinuity.indication {
        disruption,
        deltaPhase,
        deltaFrequency,
}
```

Boolean disruption: as defined in 802.1AS D4.0, § 9.6.2

ScaledNs deltaPhase: The amplitude of the phase's change related to the indicated discontinuity.

```
if lastGmPhaseChange != lastGmPhaseChangeOld then
        deltaPhase = lastGmPhaseChange – lastGmPhaseChangeOld
```

Double deltaFrequency: The amplitude of the frequency's change related to the indicated discontinuity

```
if lastGmFrequencyChange != lastGmFrequencyChangeOld then
        deltaPhase = lastGmPhaseChange – lastGmPhaseChangeOld
```

ClockTargetStatus Interface

- Proposal for a **new interface** which shall be mandatory to allow the ClockTarget to get the characteristics of the ClockSlave entity of a time-aware system.

```
ClockTargetStatus.request {
}

ClockTargetStatus.indication {
          ClockSlaveStatus,
          ClockSlaveResolution
}
```

`Integer8 ClockSlaveState`: specified the current synchronization status of the 802.1AS layer, as:
`{ freerunning, GM_locked, GM_acquisition, … }`

`UScaledNs ClockSlaveResolution:` specify the resolution of the Clock Slave Entity and allow the Clock Target entity to set correctly the parameters of the `ClockTargetTriggerGenerate` and `ClockTargetClockGenerator` interface

question: does it exist global variables relevant to these parameters?

ClockTargetTriggerGenerate Interface

- **Mandatory interface** if the `ClockTargetClockGenerator` interface is not implemented.

```
ClockTargetTriggerGenerate.request {
        sourceTime
}
ClockTargetTriggerGenerate.indication {
        errorCondition
}
```

`ExtendedTimestamp sourceTime`: as defined in 802.1AS D4.0, § 9.4.2

`Boolean errorCondition`: as defined in 802.1AS D4.0, § 9.4.2.

In addition of the §9.4.3.1 note, errorCondition could be also TRUE:

- If a clockSlave disruption has occurred before the requested sourceTime (A `ClockTargetPhaseDiscontinuity.indication` shall have been invoked)
- if the ClockSlave's resolution doesn't allow to indicate accurately the requested sourceTime.

ClockTargetClockGenerator Interface (1/2)

- **Mandatory interface** if the `ClockTargetTriggerGenerate` interface is not implemented.

```
ClockTargetClockGenerator.request {
        clockPeriod,
        sourceTimePhase
}
```

`TimeInterval clockPeriod:` as defined in 802.1AS D4.0, § 9.5.2

`ExtendedTimestamp sourceTimePhase`: as defined in 802.1AS D4.0, § 9.5.2

ClockTargetClockGenerator Interface (2/2)

```
ClockTargetClockGenerator.indication {
        sourceTime
        errorCondition
}
```

`ExtendedTimestamp sourceTime`: as defined in 802.1AS D4.0, § 9.5.3. As described in previous slides (Video Synchronization system), this parameter could be NULL as far as generating only a tick signal is concerned, and because the media clock (video genlock) is asynchronous regarding the 802.1AS clockSlave

`Boolean errorCondition`:

errorCondition could be TRUE:

- If a clockSlave disruption has occurred before the requested sourceTime (A `ClockTargetPhaseDiscontinuity.indication` shall have been invoked)
- If the ClockSlave's resolution doesn't allow to handle accurately the requested clockPeriod.

Discussion

Thanks

Any comment or question is welcome