ETSI TR 103 010 V1.1.1 (2007-03)

Technical Report

Speech Processing, Transmission and Quality Aspects (STQ);
Synchronization in IP Networks -
Methods and User Perception

Reference
DTR/STQ-00068

Keywords
multimedia, quality, speech, synchronization, VoIP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Contents

Intellectual Property Rights 4
Foreword....4
Introduction 4
1 Scope 5
2 References 5
3 Definitions and abbreviations....7
3.1 Definitions....7
3.2 Abbreviations 7
4 IP-based Services and Network Configurations....8
4.1 Network models 8
5 Timestamp methods 9
5.1 RTP based media transport....10
5.1.1 RTP Control Protocol 10
5.1.2 Functionality of RTP timestamps 11
5.2 Network Time Protocol or NTP 12
5.2.1 Possible sources of errors in NTP....15
5.2.2 Achievable performance with NTP....15
5.3 Precision Time Protocol or IEEE 1588 16
5.3.1 Protocol Implementation of PTP 18
5.3.2 Limitations of PTP Protocol 19
5.3.3 Possible Sources of Error in PTP....20
5.3.4 Achievable performance with PTP 20
5.4 Clock rate and state based synchronization....20
5.4.1 Positive and negative features of SyncUTC 21
5.4.2 Achievable Performance in SyncUTC....21
6 Perception-based application requirements....22
6.1 Audio....23
6.2 Video 23
6.2.1 Animation 23
6.2.2 Audio and Video....23
6.2.2.1 Conversational Service....23
6.2.2.2 Distribution service 24
6.2.2.3 Conferencing Service 24
6.3 Data and Audio....24
6.3.1 Pointers and Audio 24
6.3.2 Still pictures and audio 24
6.3.3 Text and Audio 24
6.4 Data and video....24
6.4.1 Text and video 24
6.4.2 Still pictures and audio 25
7 Analysis of timestamp methods 25
7.1 Clock frequency stability....25
7.2 Timestamp reading errors....26
8 Media synchronization in services offered over a combination of TDM and IP networks....27
History 28

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Speech Processing, Transmission and Quality Aspects (STQ).

Introduction

Traditionally, TDM voice networks have had service performance requirements based on meeting the end user's quality of service expectations. In the TDM network, synchronization is a physical layer parameter that has to be designed to meet specific performance standards. Without proper clock synchronization, a service offered over the TDM network experiences errors, i.e. missing data that contributes to reduced service quality and availability. More and more real-time services are now being offered over internet-based networks, where timestamp based synchronization is utilized for billing, maintenance, call control, one way delay measurements and intra/inter-media stream synchronization.

Real-time applications offered over the internet include voice, video and data that have been traditionally carried over circuit switched networks. These services are offered by utilizing new equipment and new protocols exclusively designed for this purpose. The protocols permit integration of previously dissimilar voice and data services, creating new applications such as integrated voice mail and email, white boarding that combines voice call with data transfer, desktop video calling etc., from the use of a single integrated network. An entity to perform data/signalling conversion is required when these services are supported across disparate networks.

Internet services can also be offered by connecting existing TDM islands using Internet Protocol (IP) network (TDMoIP or Circuit Emulation over IP) that enables backward compatibility. In TDMoIP, data and signalling from TDM islands will be encapsulated or de-encapsulated in the inter-working functions situated at the interfaces of TDM and IP networks. Service quality requirements are expected to be the same as those for TDM service as the end user is not aware of the IP transport. The same argument can be extended to physical layer synchronization requirements.

The IP network is an asynchronous network with no knowledge of the physical layer and it was solely used for data transport until recently. With the introduction of real-time IP services, the need arose to set specifications for QoS related parameters - delay, delay variation and packet loss. Main sources of packet loss are bandwidth limitations at the edges, network congestion, clock related impairments and large delay variations that cause the jitter buffer to drop or add. ITU-T has published ITU-T Recommendation Y.1541 [1] on performance objectives for Layer 3 based network parameters - end-to-end packet delay, packet delay variation and packet loss - based on different classes of service. Efforts are ongoing to set additional service classes with reduced packet loss objectives. However, synchronization at the physical layer is a topic of discussion only for TDMoIP services in ITU-T [2] and ANSI [3] standard bodies.

The present document reviews the time stamping methods available in IP networks and assesses the performance of these in achieving satisfactory services as perceived by end user.

1 Scope

The purpose of the present document is to examine whether accuracies achievable by currently available synchronization methods in IP networks are adequate to ensure end user's quality of service expectations are met for a particular multimedia service. The present document will discuss the issues related to inter-stream synchronization when all the concerned media are offered using:

- IP networks from end-to-end.
- A combination of TDM and IP networks.

Services that will be covered, but not limited to:

- Video conferencing: uses video and voice.
- White boarding: uses data and voice.

When a multimedia service is offered in an IP-based network, the play-out mechanism involves de-multiplexing of different media in end node and subsequent play-out of the media based on the timing information carried over by the media. Depending on this timing information, one media may precede the other resulting in user dissatisfaction. For each media, the timing relationship between different packets (intra-stream synchronization) is preserved by proper presentation at the end user in which play-out buffer management plays a crucial part. Size of the play-out buffer can be a fixed value or can be adaptively set based on one way delay measurement using timestamps.

The present document gives an overview of synchronization methods by time stamping and the other means that are available to IP-based real-time applications and the effect of these methods on service performance as experienced by end user. However, actual implementation of time stamping is outside the scope of the present document. The reference point at which inter-stream synchronization will be examined is considered to be at the interface of the end device involved, where the play-out buffers are normally located.

2 References

For the purposes of this Technical Report (TR) the following references apply:

NOTE 1: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

NOTE 2: The following standards contain provisions, which, are referenced in the present document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on the present document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below, or their successors. ANSI and TIA maintain registers of currently valid national standards published by them.

[1] ITU-T Recommendation Y.1541 (2006): "Network performance objectives for IP-based services".

[2] ITU-T Recommendation G.8261/Y.1361 (2006): "Timing and synchronization aspects in packet networks".

[3] ANSI T1 draft technical report OPTXS-SYNC-2006-007R4.doc: "Synchronization of Packet Networks".

[4] IETF RFC 3388: "Grouping of Media Lines in Session Description Protocol (SDP)".

[5] IETF RFC 3261: "SIP: Session Initiation Protocol".

[6] IETF RFC 3550: "A Transport Protocol for Real-Time Applications".

[7] IETF RFC 3551: "RTP Profile for Audio and Video Conferences with Minimal Control".

[8] IETF RFC 2250: "RTP Payload Format for MPEG1/MPEG2 Video".

[9] IETF RFC 3016: "RTP Payload Format for MPEG-4 Audio/Visual Streams".

[10] IETF RFC 3497: "RTP Payload Format for Society of Motion Picture and Television Engineers (SMPTE) 292M Video".

[11] IETF RFC 3984: "RTP Payload Format for H.264 video".

[12] IETF RFC 1305: "Network Time Protocol (version 3) - Specification, Implementation and Analysis".

[13] IEEE Std 1588-2002: "Standard for Precision Time Synchronization Protocol for Networked Measurement and Control Systems".

[14] IETF RFC 4330: "Simple Network Time Protocol (SNTP) Version 4 for IPV4, IPV6 and OSI".

[15] David L. Mills: "Internet Time Synchronization: The Network Time Protocol", IEEE Transactions on Communications, Vol. 39, No.10, October 1991, pp 1482 - 1493.

[16] David L. Mills: "Adaptive Hybrid Clock Discipline Algorithm for the Network Time Protocol", IEEE Transactions on Networking, Vol. 6, No. 5, October 1998, pp 505 - 514.

[17] David L. Mills: "NTP Performance Analysis", www.eecis.udel.edu/~mills/ntp.html.

[18] Nelson Minar, MIT Media Lab: "A Survey of the NTP Network", http://www.media.mit.edu/~nelson, December 9, 1999.

[19] John C. Eidson and Kang Lee: "Sharing A Common Sense of Time", IEEE Instrumentation and Measurement Magazine, March 2003.

[20] Dirk S. Mohl: "IEEE 1588 - Precise Time Synchronization as the Basis for Real Time Applications in Automation", http://www.l-com.com/multimedia/whitepapers/wp_IEEE1588.pdf.

[21] Hans Wiebel: "PTP Timestamping Methods", 2004 Conference on IEEE 1588, September 2004.

[22] Dave Tonks: "IEEE 1588 in Telecommunications Applications", 2004 Conference on IEEE 1588, September 2004.

[23] Eidson, J.C and Hamilton, Bruce: "IEEE-1588 Node Synchronization Improvement by High Stability Oscillators", NIST workshop on IEEE 1588, September 2003.

[24] Martin Horauer: "Clock Synchronization in Distributed Systems", Ph.D. dissertation, Department of Computer Technology, University of Technology, Vienna, February 2004.

[25] Kero N.E., Muhr H., Gaderer G., Holler R., Sauter T., Horauer M.: "Embedded SyncUTC and IEEE 1588 clock synchronization for Industrial Ethernet", NIST workshop on IEEE-1588 standard, September 2003.

[26] Gaderer G., Holler R., Sauter T. and Muhr H.: "Extending IEEE 1588 to Fault Tolerant Clock Synchronization", Proceedings of International Workshop on factory communication systems, September 2004, pp 353-357.

[27] ITU-T Recommendation G.114 (2003): "One way transmission time".

[28] ITU-T Recommendation F.700 (2000): "Framework Recommendation for multimedia services".

[29] Ralf Steinmetz: "Human Perception of Jitter and Media Synchronization", IEEE Journal on Selected Areas In Communications, Vol. 14, No.1, January 1996.

[30] Yadavalli G., Masry M., Hemami S.: "Frame Rate Preference in Low Bit Rate Video", IEEE International Conference on Image Processing (ICIP 2003), Barcelona, 14-17 September 2003.

[31] ANSI T1 technical report TR-45 (1995): "Speech Packetization".

[32] ITU-R Recommendation BT.1359-1 (1998): "Relative timing of sound and vision for broadcasting".

[33] ETSI ETR 297 (1996): "Human Factors (HF); Human Factors in Videotelephony".

[34] ANSI T1.522 (2000): "Quality of Service for Business Multimedia Conferencing".

[35] ETSI TR 102 479 (V1.1.1): "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Review of available material on QoS requirements of Multimedia Services".

[36] ATSC Implementation Subcommittee Finding - Doc. IS-191 (June 2003): "Relative Timing of Sound and Vision for Broadcast Operations".

[37] ETSI TS 122 228 (V7.5.0): "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Service requirements for the Internet Protocol (IP) multimedia core network subsystem (IMS); Stage 1".

[38] ITU-T Recommendation Y.1401 (2000): "General requirements for interworking with Internet protocol (IP)-based networks".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

drift: variation in skew or the second derivative of offset between two clocks with time

IP multimedia application: application that handles one or more media simultaneously such as audio, video and data (e.g. chat, shared whiteboard) in a synchronized way from the user's point of view

NOTE 1: As defined by TS 122 228 [37].

NOTE 2: A multimedia application may involve multiple parties, multiple connections, and the addition or deletion of resources within a single IP multimedia session. A user may invoke concurrent IP multimedia applications in an IP multimedia session.

IP-based service: An IP-based service is defined as the functions, facilities, and capabilities implemented and executed above IP network services

NOTE 1: As defined by ITU-T Recommendation Y.1401 [38].

NOTE 2: It utilizes the IP Transfer Capabilities offered by a network provider.

offset: time difference between two clocks

skew: frequency difference or first derivative of the offset between two clocks with time

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ADSL	Asymmetric Digital Subscriber Line
ALF	Application Layer Framing
ALG	Application Level Gateway
ATSC	American Television Systems Committee
CSRC	Contributing Source
DTS	Decode Time Stamp
FEC	Forward Error Correction
HDTV	High Definition TeleVision
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
MPEG	Moving Picture Experts Group
NIC	Network Interface Card
NTP	Network Time Protocol

PCR	Program Clock Reference
PT	Payload Type
PTP	Precision Time Protocol
PTS	Presentation Time Stamp
QoS	Quality of Service
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTT	Round Trip Times
SCR	System Clock Reference
SDP	Session Description Protocol
SDTV	Standard Definition TeleVision
SIP	Session Initiation Protocol
SMPTE	Society of Motion Picture and Television Engineers
SR	Sender Report
SSRC	Synchronization Source
SyncUTC	Synchronized Universal Time Coordinated
TDM	Time Division Multiplex
TDMoIP	TDM over IP
VoIP	Voice over Internet Protocol

4 IP-based Services and Network Configurations

4.1 Network models

Multimedia services can be offered using end-to-end IP networks as shown by Figure 1 where all the media terminals are connected to a network that has a common reference clock. In this case, media synchronization depends on the accuracy of the timestamp method followed.

Multimedia services can also be offered by using a combination of TDM and IP networks with voice going over traditional path and video and/or data going over internet path as shown in Figure 2. TDM based media uses different synchronization methods than IP-based media making media synchronization a function of reference clocks in the respective networks and relative accuracies of the methods utilized.

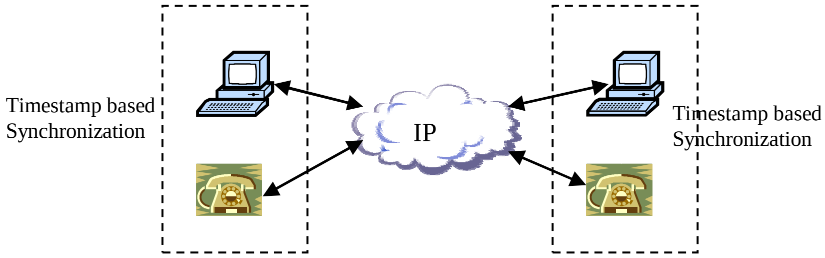

Figure 1: IP-based implementation of a multimedia service

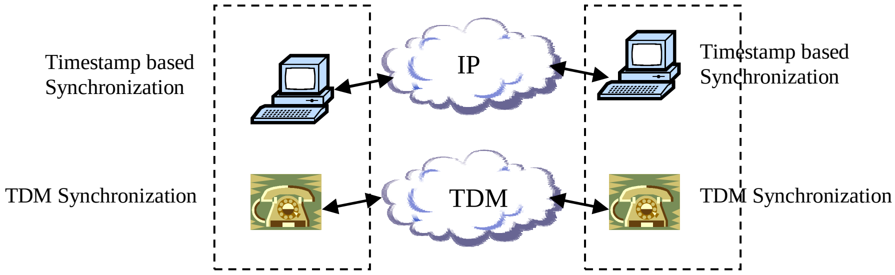

Figure 2: Multimedia Service offering over a combination of TDM and IP Networks

When a multimedia service is offered in an IP-based network, presentation of the multiple media at the end user needs temporal organization between different media components that involves resolution of intra-stream and inter-stream synchronization. Depending on the media characteristics - real-time/continuous (e.g. video, audio and animation), hybrid (e.g. audio with graphics, video with audio and image) temporal relationships need to be identified and established. Media derived from different sources experience different delays and jitter in the data transmission path.

A typical play-out mechanism that is involved in a multimedia service is illustrated in Figure 3 where the incoming packets are demultiplexed at the network end node. Timing information of the different media involved is extracted by a synchronization agent that adjusts the play-out delays such that the media are available to the end user within an acceptable time window. Based on the accuracy of the timing used, the end user may experience the involved media simultaneously or at different times.

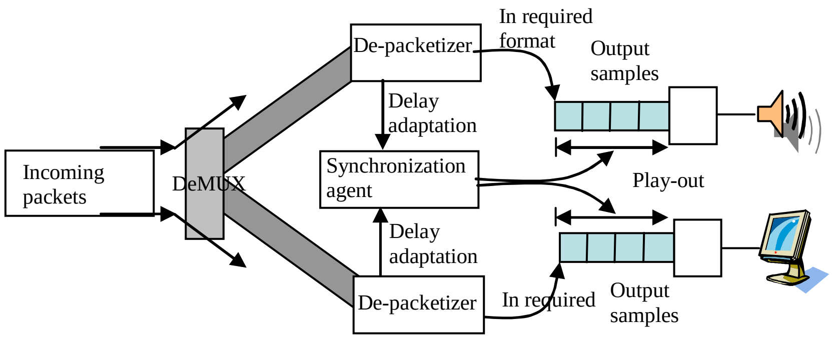

Figure 3: Media Play-out Mechanism involved in a Multimedia Service

5 Timestamp methods

Information about synchronization will be conveyed and/or negotiated between the endpoints with the use of a control protocol during session setup or any other stage. The purpose of the protocol is to indicate that synchronization is required between specific media components that constitute the service. Actual implementation of synchronization will not necessarily be a part of the control, however, it needs to be defined at the control level if not provided at the media transport level. If the feature is not supported by both or one of the endpoints, the media may be presented unsynchronized to the end user. E.g. SDP has an option to indicate which streams are needed to be synchronized [4] at the receiver. SIP [5] can use this SDP as a part of session negotiation. If the receiver can not support the feature, it ignores this attribute. Based on the system implementation, either party can refuse the session.

5.1 RTP based media transport

Applications sensitive to delay and jitter use RFC 3550 [6] based Real-time Transport Protocol (RTP) over IP networks. Some of the supported applications include interactive audio, video, multiparty conferencing, stored media distribution. RTP is used to provide delivery services - payload identification, sequence numbering, time stamping and delivery monitoring for real time data. To facilitate these services, RTP header contains Synchronization Source identifier (SSRC), Contributing Source identifier (CSRC), sequence number, payload type (PT) and timestamp. SSRC gives identification of the source of a stream of RTP packets. Based on SSRC, sequence numbers and timestamps would be allocated to different packets. All the packets that are generated by the same source will have the same SSRC. Sequence numbers increase monotonically for each RTP data packet belonging to the same source and a missing sequence number indicates a missing packet. Initial value for both categories of timestamp and sequence numbers would be random. CSRC gives the identification of the source for the payload contained in the packet. Payload type gives the type of codec used in the generation of a stream. RFC 3550 [6] states that an RTP based application is completely specified, when one or more companion documents on payload format and profile are provided. RTP also supports multicasting if provided by the underlying network. Audio and video profiles for conferencing application with minimal control are specified in RFC 3551 [7] that includes information on payload encodings, including clock frequencies to be used in different codecs, e.g. JPEG, H.261, and H.263. The codecs that are not defined in RFC 3551 [7] are covered in RFCs 2250 [8], 3016 [9], 3497 [10] and 3984 [11] give payload formats for RTP using MPEG1/MPEG2, MPEG4, SMPTE, and MPEG-4 part-10 or H.264 respectively. A payload format specifies packetization scheme, use of RTP timestamp in the receiver, media and codec specific headers, changes to RTP header, if any. Functionality of different header fields and payload formats differ from those when used in stand alone environment, e.g. MPEG-4 system has a synchronization layer to take timing issues into account in native environment. However, MPEG-4 streams transported using RTP do not utilize sync layer functionality. SMPTE uses a separate low frequency timing stream for synchronization.

Timestamp field is 32 bits long and indicates the sampling instant of the first byte of the data packet that is generated by a sampling clock at the sender. The sampling clock is designed to increment monotonically and linearly in time, even when the source is inactive. Depending on the packet size and media type, a number is added to each generated media packet e.g. if 30 ms audio packets are generated, the RTP timestamp which is initialized randomly is increased by 240 i.e. 8000 Hz x 0,03. Resolution of the clock used to generate timestamp determines the synchronization accuracy and also should be adequate for jitter estimation. Clock frequency depends on the format of the payload, e.g. for systems using video encoding, the clock frequency is 90 kHz and is the same as that of MPEG timestamps. A receiver can use this timestamp to provide synchronous play-out. However, RTP timestamps for different media usually have independent random offsets due to the use of different synchronization sources and increase at different rates so that RTP timestamps on their own are insufficient to synchronize in the receiver. Finally, where RTP timestamp rates differ from the nominal values e.g. 8 000 Hz for audio or 90 kHz for video, then such clock skew can result in a cumulative timing error or misalignment between different streams e.g. audio and video.

RTP is an Application Level Framing (ALF) based protocol and it uses mixers and translators as Application Level Gateways (ALG) between two transport clouds. Translators process different data streams independently, keeping the SSRC intact. A mixer generates a single new data stream out of several different incoming data streams and changes the SSRC to a new value identifying the mixer as the source and puts the old value of SSRC into CSRC list. A CSRC list contains list of all the synchronization sources that contributed to the generation of that particular stream. Timing adjustments between different streams is set by the mixer. Examples of translators include audio/video converters, firewalls and gateways that receive multicast data and transfer to unicast receivers. Clock skew can seriously degrade the performance of mixers.

5.1.1 RTP Control Protocol

RTP is augmented by RTCP for control. RTCP is primarily used for getting feedback on quality of data delivery. The amount of RTCP traffic allowed and the intervals between two RTCP packets need to be engineered as a part of network design as bandwidth is a precious resource. RTCP packets carry information that can be used by the end points of the media path to calculate packet loss and delay variation. For each RTP stream, the sender transmits a RTCP sender report (SR) periodically that has the information about number of packets sent in the stream, number of bytes sent, timestamp pairs containing RTP timestamp and absolute time or wall clock (also known as system time) in NTP format corresponding to RTP timestamp. To implement this aspect correctly, the two timestamps should correspond to the same instant. RTP timestamps enable a receiver to reconstruct the media stream from incoming packets. Furthermore, such a receiver can use the RTP/NTP timestamp pairs within received RTCP SR packets for inter-stream synchronization of streams that originate from the same host i.e. the RTCP SR packets generated for each media stream from a given host use a common system time and thus RTP timestamps from the various streams can be temporally aligned.

A media aware receiver uses timestamp pairs for intra-stream and inter-stream synchronization. A media un-aware receiver uses the timestamps to predict the RTP clock frequency. RTCP reports are issued based on SSRC and frequency of which determines the resolution of the statistics. A traditionally built RTP system prohibits multiplexing of packets with different SSRC identifiers into a single stream in a single RTP session, as encodings, sequence numbers and related statistics are generated based on SSRC. Also the RTCP reports do not include payload types, however, extended reports can include profile related information.

Wall clock time is typically derived from a local time source in the transmitter and the format is that of a NTP timestamp. This local time source can be based on

- Synchronized local clock using any synchronization protocols, e.g. Network Time Protocol (NTP) [12], Precision Time Protocol or IEEE 1588 [13]:
  - Timing accuracy is thus influenced by the performance of the protocols.
- Free run sources, in the absence of synchronization:
  - Timing accuracy is thus influenced by usual clock problems such as skew and drift.

If alignment or synchronization of media streams originating from different hosts is required, then some means of synchronizing the system clocks on the different hosts is required. This will then allow the mapping of RTCP SR packets from different hosts. More generally, if some means of synchronizing wall clock or system time is implemented, this will also facilitate precise one way delay measurements. This brings further benefits as outlined in clause 6.

Whichever synchronization method or reference clock is used to generate absolute time, they should be the same for all media involved in a multimedia session, as usage of different methods and clocks result in differences in high bits of NTP timestamp and large timing errors. Wall clock time used in an IP network can be derived from any available method, e.g. NTP, PTP and plays an important role in determining the service quality and user satisfaction.

In many applications, the wallclock time (system clock) is used in the receiver to adjust a dejitter buffer based on trends in delay measurement. However, in absence of synchronization of system clocks no precise knowledge of actual delays can be determined though Round Trip Times (RTT) can be determined using RTCP. Wallclock (system clock) can also be used for decoder and various other buffers including de-interleaving and initial buffers based on the codec requirement.

5.1.2 Functionality of RTP timestamps

Use of RTP timestamps in media play-out depends on the format of the media streams transported across all the involved IP networks and end systems. Two of typical configurations used in real-time media transport using RTP are illustrated in Figure 4. In Figure 4(a), RTP is used for transport of the media generated by an end network which has its own rules for time-dependent events, e.g. decoding and presentation of the data. One example of such a network can be a MPEG-2 based cable distribution plant, wherein Figure 4 (a), the transitions are from native MPEG-2 network to native RTP network and vice versa. MPEG-2 encoding uses System or Program Clock Reference (SCR or PCR respectively), Decode Time Stamp (DTS) and/or Presentation Time Stamp (PTS). SCR and PCR indicate the reference clock of the transmitter and are used in the MPEG-2 program stream and the transport stream, respectively. The MPEG-2 receiver uses these time stamps to synchronize local clock to the transmitting clock. DTS is used to indicate the time at which the data should be taken instantaneously from the decoder buffer and decoded. PTS indicates the instant at which the data should be removed from the receiver buffer, instantaneously decoded and presented for display. DTS and PTS are identical, unless specified otherwise. In these cases, use of RTP time stamp may be limited to synchronizing the receiver clock to the transmitter clock. RTP can also be used to connect two internet-based systems as given in Figure 4(b), e.g. video server and computers.

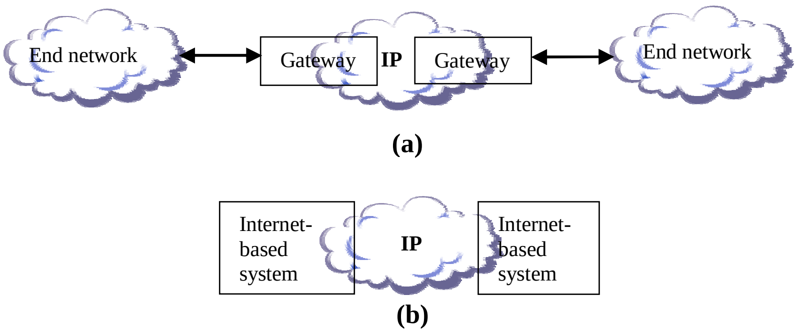

Figure 4: Typical Scenarios of Media Transport using RTP

Some codecs support data formats that send different media in different streams, where the others send a single multiplexed stream with packets from multiple media based on the application. The role of RTP time stamps changes depending on the combination of application and codec. Live media streams can be encoded and transported across IP network as RTP packets with headers that contain sequence numbers and timestamps to be used in presentation of the data to the end user. If a media frame is bigger than a single packet that can be carried by the underlying network, several packets belonging to the same frame will have the same time stamp, however, different sequence numbers. Each media is transported in a single RTP session using QoS allocated by the transport network. Based on the QoS, different media can travel by different paths. In the receiver, RTP sequence numbers are used to order the packets, subsequent timestamps and media play-out speeds determine intra-stream synchronization. The timestamp pair in the RTCP SR can be used for inter-stream synchronization of streams from the same host. As outlined in clause 5.1.1, if interstream synchronization for streams from different hosts is required, then wall clock synchronization across the different hosts is also required. Examples of the end systems that support this scenario are video servers, computers and work stations.

Multimedia can also be transported using a single IP stream that has multiplexed packets from different media. Temporal information of the packets is carried in the packet headers and some type of numbering is used to indicate order of appearance to the end-user. Use of different QoS for different media is not possible with a single stream, as the IP network is not aware of the media involved. Also, the RTP layer will not be used in the presentation of the information. In the receiver, the timestamp extracted from RTP header is used to synchronize the local clock to the transmitter clock. Decoder and play-out buffers utilize the timestamps, if any, that are encoded into the stream.

5.2 Network Time Protocol or NTP

Network Time Protocol (NTP) is one of the most popular protocols that is used for Internet synchronization and version 3 of the protocol is specified in RFC 1305 [12]. Version 4 of the protocol started as Simple Network Time Protocol (SNTPv4 - RFC 4330 [14]) with the basic difference from version 3 being a modified header to accommodate IPV6 addressing and is still in development.

In NTP, the synchronization architecture is based on a hierarchical master-slave configuration with a tree like structure of time servers and clients that are organized with primary reference source at the root and other servers of varying accuracy at different levels of the tree. The accuracies are represented by a number called stratum, determined by the hop count from the root as shown in Figure 5. Primary servers are set at stratum 1. Each succeeding level towards the leaves has a stratum one higher than the preceding level, i.e. if a server of stratum *i* is used to synchronize a client, the client will be of stratum (*i+1*) and the maximum value allowed for the stratum is 15. Primary time servers are synchronized to external reference sources that are national standards using wire-line or wireless connections or an onsite portable clock i.e. atomic standard clock.

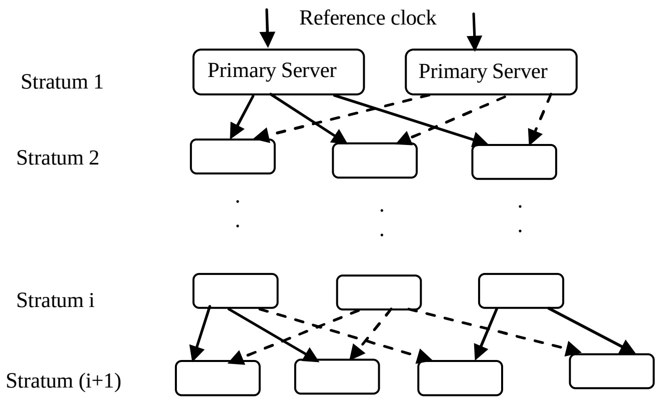

Figure 5: NTP hierarchical master-slave configuration

The protocol was developed by Mills [15] to satisfy the following performance requirements:

- The primary servers deliver time continuously.
- The primary servers provide accurate and precise time. Since the servers derive timing from national standards over wide area networks that are prone to be affected by relatively large delay variations, changing traffic load conditions, route selection mechanisms and outages, they employ data smoothing, deglitching algorithms and they use stable local oscillators to provide accurate and precise time.
- Timing supply is reliable and survivable. The time servers use redundant servers and diverse network paths to provide a reliable timing source. Also timing distribution paths are dynamically reconfigurable.
- Operation of the timing protocol is continuous and the protocol provides periodic updates at rates adequate to accommodate expected wander of the room-temperature quartz oscillators that are commonly used in ordinary computers.
- Operation of the protocol is flexible enough to accommodate a variety of hardware ranging from personal computers to super computers.

Clock synchronization of a local clock in a host/computer involves sending periodic requests to a set of time servers that respond with the local timestamps. Time offset of the NTP client will be derived from a collection of four timestamps (two of its own and two from the remote server). A list of suitable servers that can be used for time synchronization are maintained in each client. Based on the version of the protocol used, this list can be set automatically (version 4) or manually. At designated intervals, determined by a dedicated timer, the client sends a request to each server in the list and the response will contain two timestamps corresponding to receiving and sending times. From these, the client calculates round trip time, clock offset relative to each server separately, distance and dispersion (maximum error relative to a reference clock) to the root and submits the information to grooming algorithms. The grooming algorithms use this information to select the best server with the lowest stratum and synchronization distance which is used by the client for setting the clock update. Depending on the network characteristics, a local host can take a long time to set an accurate time setting. With the Internet, to achieve an accuracy of a few tens of milliseconds per day, dozens of measurements may be required over many hours depending on the version of the protocol used. To achieve an accuracy of the order of an ms/day, hundreds of measurements over many days may be required with the earlier versions of NTP [16].

NTP is built on IP and UDP. The protocol uses several variables and parameters for the calculation of the offset and roundtrip time. Some of these parameters are fixed, e.g. maximum values of stratum, maximum allowed distance for a peer to be accepted as a synchronization source, minimum and maximum values of dispersion. Variable parameters are delay, offset, dispersion to the synchronization peer, and peer timer. The first three are used in determining the accuracy of the local clock and the last one is used to control spacing of NTP messages. Some of the fields contained in a NTP message are as follows-

- Local clock related: Stratum (see note), mode of operation and precision.

NOTE: The accuracy of each server is given by an integer, called the stratum with the top most level assigned as one and each level downwards in the hierarchy assigned as one greater than the preceding level. In TDM networks, four levels of stratum are defined that are not necessarily the same as the ones used in the present document.

- Root clock source related: delay and dispersion of the local clock with respect to the primary reference source.
- Reference clock Identifier: type of reference clock used.
- Four timestamps:
  - Reference timestamp: of the local time at last update.
  - Originate timestamp: peer timestamp when the last received message was sent.
  - Receive timestamp: local time when the latest NTP message was received.
  - Transmit timestamp: local time when the above message is transmitted.

A single timestamp represents the local clock copied at that particular instant of time. The timestamps are 64 bits long with first 32 bits indicating the integer and the last 32 bits indicating the fractional part. The precision of the representation is approximately 200 picoseconds and the observed precision depends on the actual implementation of the system.

The protocol calculates clock offset of a client from a server as shown in equations (1) to (5) and the method is illustrated in Figure 6, where host A and host B are two computers. Host A is one of the suitable synchronization servers listed in Host B, which is a client in this case. When Host B transmits an NTP request message, it sends a timestamp T_1 (originate timestamp). Upon receiving NTP request, host A generates timestamp T_2 (receive timestamp), spends a finite time in processing the request and transmits a response with a timestamp, T_3. T_1 and T_4 are the time stamps generated by host B using its own clock, T_2 and T_3 are the timestamps generated by host A using its local clock.

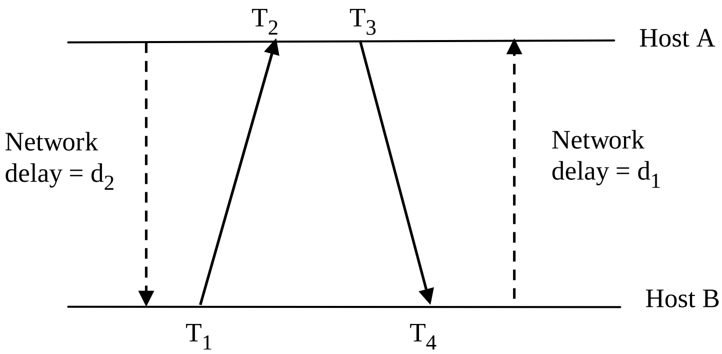

Figure 6: Request and response in NTP

$$T_2 = T_1 + d_1 + \delta_{B,A} \tag{1}$$

where, d_1 is network delay experienced in the transmission path from B to A;

$\delta_{B,A}$ is clock offset of B with reference to A.

$$T_4 = T_3 + d_2 + \delta_{A,B} \tag{2}$$

where, d_2 is network delay experienced in the transmission path from A to B;

$\delta_{A,B}$ is clock offset of A with reference to B.

$$\delta_{B,A} = \frac{(T_2 - T_1) - (T_4 - T_3) - (d_1 - d_2)}{2} \tag{3}$$

$$\text{if } d_1 = d_2,\ \delta_{B,A} = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \tag{4}$$

Similarly, Round trip time is calculated by Equation (5).

$$d_1 + d_2 = (T_2 - T_1) + (T_4 - T_3) \tag{5}$$

5.2.1 Possible sources of errors in NTP

Since NTP timestamps are calculated from roundtrip delay measurements, the timing quality can be effected by the characteristics of the network interconnecting the two hosts and errors arising from clock discipline or adjustment algorithm itself. Equation (4) is valid only when the network delays experienced by NTP message are equal in both paths of request and the response. The messages can experience unequal delays due to use of asymmetric link bandwidths in access and asymmetric propagation delays resulting from network routing. Bandwidth available to access the internet can be different in sending and receiving paths, based on the implementation. Some examples are access by ADSL modems, cable modems, and high-speed modems. The propagation delay experienced by a NTP packet include fixed packet delay, variable queuing delay and random errors, with the last two dependent on the statistical properties of inbound and outbound network paths. Asymmetric delays also occur, when the two hosts involved are connected by disparate networks, served by different service providers and when request, response paths take different routes.

As NTP calculates local clock offset from the latest four timestamps, any errors that occur in determination of timestamps also effect quality of timing based on NTP. Some of the possible sources include Operating system latencies in scheduling, measurement errors, frequency errors and the errors arising from clock discipline and selection algorithms. Measurement errors arise from precision and granularity of a local clock based on the interval used to increment internal digital counters/dividers to adjust frequency and time. The errors due to granularity are largely resolved to microsecond level for UNIX®/Windows® machines, that are of the same magnitude as those due to other uncertainties. A timestamp is determined with a reading of the local clock at any arbitrary time that is independent of this interval, thus introducing a reading error. Characteristics of the local clock such as drift, manufacturing processes, temperature variations and path transients contribute to frequency related errors. As frequency errors are functions of observation time, they are expressed from the time of the last update.

Each NTP server maintains its local clock offset, round trip delay and dispersion relative to the primary reference source located at the root of the synchronization tree. Frequency and phase errors resulting from clock discipline are additive and contribute to the overall error budget at each server. When a client sends a NTP request, before sending a response with timestamps, the server adds its accumulated errors from the time its clock was last updated. Thus the clock offset, round trip delay and dispersion increase as the hop count increases from the primary reference source. Details of these derivations are given under inherited errors in RFC 1305 [12].

5.2.2 Achievable performance with NTP

NTP was designed for UNIX® platforms and though now ported to Windows® operating system, performance on UNIX® and associated platforms is superior to that of Windows®. NTP performance is optimized under following conditions:

- Proximity to primary reference sources: With decreasing cost of GPS clocks, the feasibility of installing a primary (stratum 1) server is greatly enhanced. Consequently, the availability of Stratum 1 servers has greatly increased in recent years.
- Redundant paths: An NTP client should utilize multiple redundant paths to ensure that network path failure or asymmetries in individual paths can be identified and eliminated.
- Diverse paths: Diverse paths to the redundant servers will ensure that asymmetries cannot corrupt measurements to all servers.

NOTE 1: UNIX is a registered trademark of The Open Group.

NOTE 2: Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

If above conditions are met, recent tests have shown that UNIX® or UNIX® flavoured NTP clients can be synchronized to low single figure millisecond values over LAN (1 ms to-2 ms) with marginally worse performance over well provisioned WANs.

An NTP client is said to be in synchronization with the server, if the time and frequency offsets are within 128 ms and 500 ppm respectively [16]. NTP algorithm is designed to adjust the local clock gradually with offsets less than 128 ms and step adjustments if offsets greater than 128 ms persists for an observation interval of 900 s. The algorithm resets or reboots when the clock accuracy is greater than 1 000 s. Depending on the clock time and frequency values and version of the NTP implemented (i.e. version 3 or 4), it can take a few hours to a day for the initial convergence. From the latest survey available [17], there are 230 active Stratum 1 or primary servers and 100 000 active Stratum 2 servers that can be accessed in the public domain and many other exist in private domains. The primary servers are distributed across the globe. The NTP performance surveys involve measurements of time and frequency errors of several servers synchronized with NTP. Results represent statistical values of synchronization accuracies, irrespective of whether the servers directly derived time from a primary server or where they were located in the NTP hierarchy.

A typical NTP setup involved a Stratum 1 somewhere in the Internet, Stratum 2 near the gateway for an organization, Stratum 3 and Stratum 4 clocks on the LAN. For the coverage of a University campus, Stratum 2 clocks would be located at the ingress of the University network, Stratum 3 clocks in individual department servers and Stratum 4/5 clocks in the desktops. The minimum and maximum resynchronization or polling intervals are version dependent (64 to 16 384s in Version 3 and 64 to 131 074s in Version 4). Time offset from synchronization peer were found to be with a mean value of 8.2 ms, median of 1,8 ms and standard deviation of 18 ms [18]. Dispersion to the root or the primary server was found to be of the same magnitude as the delay of the server to the root with a mean value of 88 ms, median of 30 ms and standard deviation of 175 ms. Published measurements indicate that the offset of a Stratum 4 based NTP client from different stratum 1 clocks on the Internet would be of the order of 100 ms.

5.3 Precision Time Protocol or IEEE 1588

Precision Time Protocol (PTP) is another method used for synchronization in packet based networks that support multicasting and is defined in IEEE Standard 1588 [13]. Originally introduced for testing and automation to achieve synchronization accuracies on the order of sub microseconds in Ethernet networks, the protocol uses time stamps for synchronizing different clocks arranged in a master-slave hierarchy. The protocol was designed for small homogeneous, heterogeneous networks, for use with minimal network bandwidth, low resources, low administration and low cost. With a proper design, a PTP system synchronizes the clocks to within less than a microsecond of accuracy.

A grand master clock is located at the root of the hierarchy and is selected based on the source of time it is connected to. A typical hierarchical PTP tree is illustrated in Figure 7. The order of precedence used in the selection of a grandmaster is a clock directly synchronized to an atomic clock over a clock synchronized to a GPS receiver over a clock participating in NTP over a local oscillator. A PTP system includes several masters serving groups of local clocks or slaves. The time base established by the protocol in a PTP based system will be that of the grand master. PTP is built over IP and UDP and uses several variables in the calculations of offset of a local clock with reference to a master. The messages from the master are sent by multicast communication to the slaves to limit number of addresses and to cause lesser PTP traffic over the network. Communication from slaves to the master is sent by unicast. The protocol uses four messages - Sync, Follow_up, Delay_Req and Delay_Resp between a master and slaves with two messages - Sync and Delay_req resulting in time stamp events and the other two carrying the precise time stamps [19] to [22].

A PTP system is a distributed system spanning over few subnets in which nodes are classified as ordinary clocks (nodes with a single network port), boundary clocks (nodes with multiple network ports) and administrative nodes. A boundary clock is recommended to be located in nodes that use extensive store and forward queuing techniques to transfer a packet and devices that separate different communication technologies. The function of a boundary clock is to mitigate the effect of delay fluctuations introduced by the particular component. One example is a router in packet based networks. IEEE 1588 [13] recommends use of long term averaging and filtering techniques to remove delay variations introduced by a switch. The local clock of the boundary clock is synchronized to the grand master over one port designated as the PTP port and the remaining ports act as masters to the other PTP synchronization elements in a communication path.

PTP protocol works on an acyclic topology and IEEE 1588 [13] defines different timing zones/sets using communication paths and sub-domains. In a sub-domain, clocks synchronize to each other using PTP protocol creating localized sets of clocks. The topology allows only one master to be connected to a set. PTP messages are not shared between different sub domains. The clocks in one domain will synchronize with each other but may not synchronize with the clocks in other sub domains with a boundary clock separating the two sub-domains. If a sub domain has more than one medium or technology, the clocks will be split into different communication paths based on the technology, each being served by a boundary clock. Each communication path has connection to only one boundary clock and PTP automatically corrects the topology to acyclic if there exists a communication path connecting to more than one boundary clock.

PTP uses stratum number and clock identifier to describe the quality of a particular clock. The stratum numbers are based on the reference clock to which the clock is synchronized to. Stratum 1 clocks include GPS clocks and calibrated atomic clocks. Stratum 2 clocks are directly synchronized to Stratum 1 without using PTP. The maximum number allowed for stratum based classification is 256 with only four being currently defined in IEEE 1588 [13]. On power up, the stratum number will never be less than 3. If a clock degrades from the specified values for a given stratum, the clock can change the current number or declare it faulty. If a clock improves from the specified values the stratum and identifier fields will be upgraded. The clock identifier given in ASCII form indicates nature, expected absolute accuracy and epoch of a clock. Amongst a group of clocks with the same stratum number, selection of the best clock depends on the clock identifier. Eight identifiers have been defined with some being used more than once, which in combination with stratum numbers represents a clock with absolute accuracy varying between 25 ns and a few seconds.

The protocol selects the best clock available in a system with a clock selection algorithm using the parameters obtained from Sync messages. Some of the information carried in a typical Sync message is:

- Grandmaster related: communication technology, stratum number, clock identifier, clock variance.
- Local clock related: stratum number, clock identifier, clock variance, number of communication paths from the grandmaster.
- Parent/master related: information about the parent issuing this packet - communication technology, system identifier, estimated values of clock drift, variance.
- Timestamps: Origin time stamp.
- Sync interval.

The algorithm basically compares two clocks based on the grandmaster clock from which the timing is derived. In case of equivalent grand masters, the decision will be based on the clock that has the nearest grandmaster. If the path length to the grandmaster yields equivalent results, the algorithm moves to the comparison of stratum number, identifiers etc. In the clock selection, a clock with the higher accuracy takes precedence over a clock with lower accuracy. A master clock can also provide external timing signals to the local clocks that do not posses designated PTP slave ports. However, the medium to carry external timing signals is different from that used for the transport of Sync messages.

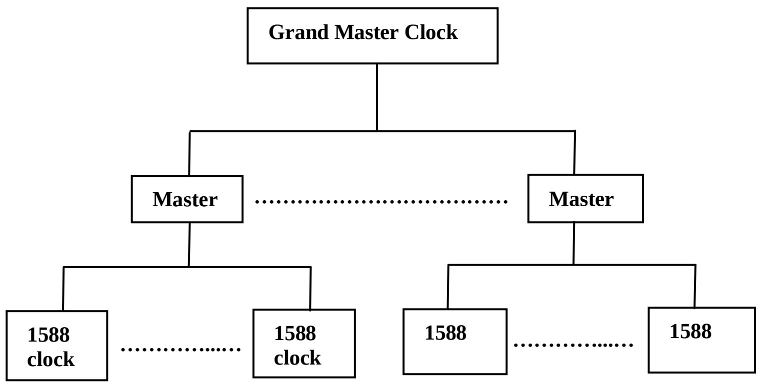

Figure 7: Master-Slave hierarchy in IEEE 1588 [13]

5.3.1 Protocol Implementation of PTP

The PTP protocol utilizes two phases for setting local clock time - offset measurement phase and delay measurement phase. In offset phase, the master periodically transmits Sync message to related slaves using multicast at pre-defined intervals (1, 2, 8, 16 and 64 s, with a default value of 2 s). Transmission of Sync messages can also be prompted by management messages. The Sync message contains an estimate of the time the message will physically leave the master. Upon the reception of the message, a slave clock measures the received time (TS_1) using the local clock. The master clock sends a Follow_up message within a pre-defined interval with the precise sending time enclosed that is accurately measured at a point close to the physical layer and indicated by TM_1, in the present document. The slave adjusts the local clock with the offset as given by (6) and the message sequence is given in Figure 8.

$$TS_1 = TM_1 + O_{sm} + d_{sm} \tag{6}$$

Where, O and d are the offset and delay respectively and the subscript *sm* denotes slave to master. The delay is assumed to be zero until the delay measurements are made. In case of any network delay between the master and the slave pair, after the offset being adjusted, the clocks will be synchronous.

In the second phase, the delay measurement involves a slave sending a Delay_Req message to the master. This message is time stamped with the time of reception, TM_2, the accurate measured value by the master and will be sent in Delay_Resp message to the slave. Delay between the master and the slave will be calculated as given by (7).

$$TM_2 = TS_2 + O_{ms} + d_{ms} \tag{7}$$

Where, the subscript *ms* denotes master to slave. By adding (6) and (7),

$$d_{sm} + d_{ms} = (TS_1 - TM_1) + (TM_2 - TS_2) \tag{8}$$

With a symmetric delay in both the directions, $d_{sm} = d_{ms}$, (9) gives one way delay between the master and slave.

$$One\ way\ delay,\ d = d_{ms} = d_{sm} = \{(TS_1 - TM_1) + (TM_2 - TS_2)\}/2 \tag{9}$$

Equation (9) will be used in the adjustment of offset of the slave clock for further offset adjustments.

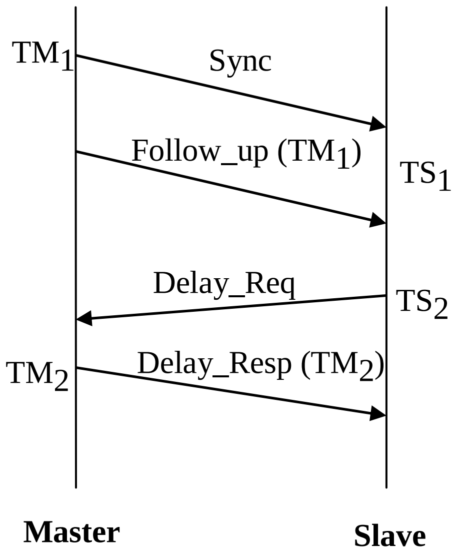

Figure 8 Offset and Delay measurements using IEEE 1588 [13] protocol

Frequency of Delay_Req messages is less when compared to that of Sync messages (usually a random value chosen between 4 s and 64 s by default). The offset adjustment takes the most recently measured delay value with the underlying assumption that the variation in delay is smaller during the measurement interval which is only possible under low network load conditions. In networks using 1588 based timing, the more the symmetry in delays between the master and the slave, the more precise the synchronization, making it a mandatory requirement that delay variation be as small as possible.

5.3.2 Limitations of PTP Protocol

Some of the factors that can influence the implementation of PTP protocol and results in degraded accuracy of PTP based timing are given in this clause.

Details about robustness of clock synchronization - redundancy of the master clocks, timing paths, failure recovery and related convergence times are not included in IEEE 1588 [13] and the specifics are left to implementation. Due to the requirement of an acyclic topology, there cannot be more than one boundary/master clock per communication path which can result in a single point of failure. Currently path reconvergence is of the order of seconds in IP networks and can disrupt the timing supply.

The delay measurements need to include latencies incurred between PTP communication path and the point at which the timestamps are generated. These latencies are the propagation times between the clock time stamp point and the communication medium for in-bound and out-bound messages - Sync and Delay_Req messages and the latencies will not be identical. The time stamps can be generated at any point in the OSI protocol stack in packet networks and with or without hardware support. Without hardware support, the timestamps are generated at application layer and are subject to delay variations of the protocol stack and the operating system. IEEE 1588 [13] specifies techniques to reduce these fluctuations. However, it is possible to generate the timestamps closer to the physical layer with hardware support resulting in increased timing precision. E.g. time stamp point can be Media Independent Interface in an Ethernet network. For external timing signal, IEEE 1588 [13] does not specify rules for the latency corrections similar to that of PTP synchronization.

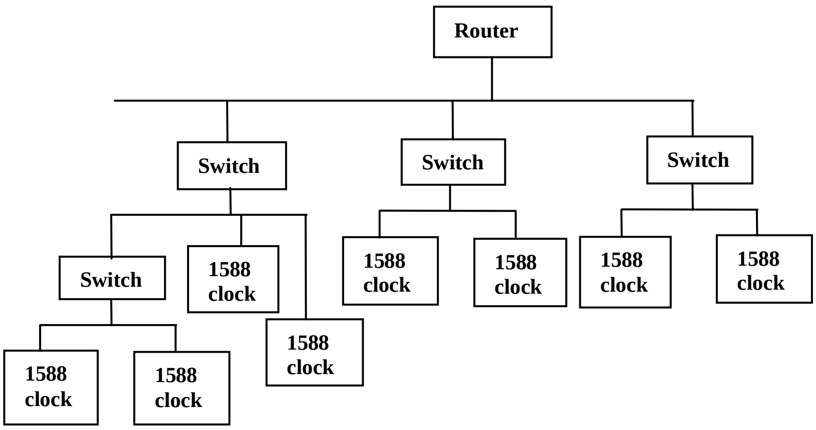

Figure 9 Typical IP network

A typical IP network can be as illustrated in Figure 9, where the subnets can be defined based on the availability of IP addresses. Long term averaging and filtering need to be employed in some of the switches to reduce the effect of large delay fluctuations that are typical of switches. In case of path reconfigurations due to failures, rerouting, etc., result in longer reconvergence times, in addition to the existing delays. Also, IEEE 1588 [13] limits the maximum number of clocks allowed in a communication path to 480 based on the implementation capacity of complete PTP functionality.

The protocol uses two sets of measurements with one set being executed more often than the other. When compared to NTP, with message interspacing and message sizes taken into account, the timing traffic is almost double in a PTP based system. If there are any low bandwidth links, synchronization traffic can be significant and should be taken into account in traffic engineering.

5.3.3 Possible Sources of Error in PTP

PTP offset measurements depend on the timestamps that are subjected to path related impairments, e.g. path symmetry, delay variation properties of the network connecting the clocks and the effect of different parameters. These impairments and the subsequent effects will be the same as in case of NTP. Any errors that effect generation of timestamps also affect the precision of the measurements. These errors include measurements errors, local clock resolution, transmission delay uncertainty and clock drift between offset adjustments.

5.3.4 Achievable performance with PTP

Use of IEEE 1588 [13] in telecommunication networks has not been studied extensively. Measurements on accuracy and precision of using this method over large networks are limited, when compared to those available with NTP based systems. However, some measurement data are available related to the expected offset from the three types of connections between a clock or a client and a master [23]. The three connections are a direct connection, a connection with a router and a connection with a switch between the client and the master. The reported measurements were with two types of clocks. The most frequently used clock was relatively inexpensive, but less stable than the other clock. All the measurements were done in the absence of any traffic other than the synchronization messages, making the effect of traffic load/delay variation on timing negligible.

For direct connections between the clock and master, the mean offset values were found to be 20 ns to 28 ns (-1 ns to 5 ns for the stable oscillator) for inexpensive and most commonly used oscillator. For clock connected via a router, the offsets were in the order of 27 ns to 32 ns (-7 ns to 10 ns for the stable oscillator). For clocks connected through a switch, the mean values of offsets fall in the range of 21 ns to 49 ns (-14 ns to 5 ns for the stable oscillator) with standard deviation being almost doubled (tripled for the stable oscillator) when clocks were connected through a switch against those connected through a router.

5.4 Clock rate and state based synchronization

One good example of this type of synchronization method is SyncUTC (Synchronized Universal Time Coordinated), which was developed based on IEEE 1588 [13] with extended features that include fault tolerance and a distributed synchronization architecture [24] to [26]. The algorithm is based on the PTP and uses the same protocol stack. The method involves keeping the rate and state of a group of local clocks in a network within specific boundaries. SyncUTC was developed with the following objectives:

- To provide internal and/or external clock synchronization:
    - Internal clock synchronization: refers to the method that minimizes the mutual deviation of all clocks in a group.
    - External synchronization: refers to the method that minimizes the deviation of all local clocks with respect to an external reference time (e.g. GPS).
- To synchronize the clock states or to keep the clock values in a group, as closely as possible.
- To synchronize the clock rates or to keep the clock speeds of all the clocks in a group, as closely as possible.
- To achieve a guaranteed precision and accuracy with the use of a deterministic algorithm that places an upper bound on the clock skew.
- To provide fault tolerance in synchronization to avoid a single point of failure.
- To mitigate the effect of delay and delay variation introduced by Ethernet switches on synchronization traffic.

Fault tolerance is achieved with the use of multiple masters against a single master in IEEE 1588 [13]. The rate and state algorithm chooses a set of clocks to be the master group which can be localized or distributed within the network. One of the clocks would be chosen as the master group speaker, which is equivalent to the master clock in PTP. By having multiple group speakers, physical layer jitter between master and slave can be reduced due to reduced number of hops in the path of synchronization traffic. The synchronization message exchange is the same as in PTP.

SyncUTC uses a proprietary Network Interface Card (NIC) that accommodates an adder based clock and all the necessary mechanisms to add time stamps to the PTP packets and to measure the network delays. The adder based algorithm adjusts both rate and state of the clock to compensate oscillator offset and drift, thus eliminating the need to employ a highly stable and expensive clock in the nodes. Due to the adder and distributed architecture, the internal precision of the clocks makes the nodes synchronized to each other, when connection to the external reference fails.

The errors introduced by delay uncertainty, due to buffering in switches, are mitigated with the inclusion of a switch add-on device with one side connected to the switch and the other side connected to the network at all the ports. The device would measure the time spent by a packet inside a switch and add the information to the packet when it is being sent out of the switch. Upon reception of a synchronization message, the receive time stamp is inserted into the corresponding field.

All clocks maintain an accuracy interval within negative and positive bounds, which is constantly maintained and updated within every node. The bounds are also used to limit the maximum clock deviation allowed for all nodes at any point, which is the inherent feature of deterministic clock synchronization. The algorithm uses an amortization phase in which a fast running clock will be slowed down until the clock rate catches up with that of the reference time.

Achievable worst-case synchronization precision for rate and state algorithm used by SyncUTC is given by Equation (10).

$$\pi = C_1\varepsilon + C_2G + C_3u + C_4P\rho \tag{10}$$

C_1, C_2, C_3 and C_4 are small integer constants and depend on the type of clock adjustment algorithms.

ε: transmission delay uncertainty; it depends on the delay variation experienced by the timestamp messages over the networks;

G: clock granularity or resolution of clock readings;

$u \leq G$: rate adjustment uncertainty i.e. timing error due to discrete rate adjustment = $1/f_{osc}$;

P: resynchronization period;

ρ: clock or oscillator drift.

For the orthogonal precision algorithm or SyncUTC, $C_1 = 4$; $C_2 = 3$; $C_3 = 11$; $C_4 = 4$. In the Equation (10), the major contribution to the clock precision comes from delay variation and frequency drift in a re-synchronization period. The algorithm was found to result in a precision of 100 ns when each of the factors - ε, G, u, Pρ are within the 10 ns.

5.4.1 Positive and negative features of SyncUTC

SyncUTC offers a fault tolerant synchronization and timing distribution due to the presence of multiple masters in the master group. The method scales well, as the accuracy increases as the number of clocks in the master group increases. Due to the use of hardware based time stamping, the effect of delay variation in the OSI stack is minimal. However, the method faces difficulties in a multi-vendor network environment, due to the special hardware that should be included in all the nodes. In the absence of the required hardware, the synchronization falls back to IEEE 1588 [13] due to the underlying protocol stack. Expensive oven controlled oscillators are not required, when an adder-based clock is used.

5.4.2 Achievable Performance in SyncUTC

The testing was only carried out for Ethernet-based LAN networks where the physical layer jitter was a major contributor to the delay uncertainty. However, the method is applicable to any packet-based network. The accuracies were found to be of the orders of 10's of nanoseconds [24].

6 Perception-based application requirements

In the absence of a synchronization method (such as NTP/PTP), the RTP and RTCP protocols facilitate reconstruction of individual streams and temporal alignment of different media streams (inter-stream synch e.g. lip synch) from the same source. With time synchronization implemented, RTP and RTCP will facilitate the time alignment of media streams from different hosts (inter-stream synchronization). This may be of use for mixing media streams. Furthermore, the existence of time synchronization will enable precise one way delays to be determined on a per packet basis which will also benefit the user as follows:

- Jitter Buffer Management: Precise delay measurements will enable the receiver to implement an informed play-out strategy. E.g. many adaptive applications make adjustments to jitter buffers based on trends in one way delays. This can result in unnecessary late losses where actual delays are low relative to the ITU-T Recommendation G.114 [27].

- Forward Error Correction (FEC): Precise knowledge of one-way delays can be used within a sender to choose an optimum FEC strategy and thus enhance the end user experience.

Finally and as outlined above, if media timestamps are increasing at a rate other than their nominal rates, this skew will present problems for the end user i.e. even if media streams are perfectly aligned at the start of a session with an offset of zero, this offset will increase linearly over time to an extent that that will become noticeable. Various techniques exist to skew detection and indeed to skew compensation.

In the following text, the term offset is used to indicate a lack of alignment or the time difference between media streams.

The QoS experienced by the user of a particular multimedia service depends on the play-out speeds and the synchronization of different media components. Based on the delay and delay variation experienced in the network, the play out of any media component can be delayed or rushed at the receiver, however, the speed need to be adjusted to maintain synchronization with the other media within acceptable limits. This clause gives level of synchronization required in real time services based on the existing standards and research for acceptable QoS. Some of the services that come under this category, taken from ITU-T Recommendation F.700 [28] are multimedia conversational service, conferencing service, audio/video distribution service, multimedia retrieval service and audio visual interactive services. Table 1 gives possible applications that fall into these service categories. Media involved can be any or all of the following:

- Audio.

- Video.

- Data: still pictures or images, pointers or cursors, text.

Table 1: Real time multimedia services and applications

Service	Applications
Multimedia conversational service	Video telephony or any telephony.
Video or audio graphic conference service	Multiparty conversation in combination with other media.
Audio/video distribution service	Applications similar to a broadcast radio or television program using various types of transport networks; Different types of on-demand audio and video services.
Multimedia retrieval service	Downloading of different types of media from a source. The users can be humans or machines.
Audio visual interactive service	Educational services, multiparty gaming.

Steinmetz conducted experiments [29] on user response to different video clips, different languages, varying speeds of audio, position and speed of pointers in the same scene. He also collected data with different groups of participants with different habits regarding amount of TV and films they watched and authored a concise report. The responses were found to be consistently the same in all the above cases. Targets were set based on the level of annoyance. When the offsets between the streams reached certain levels, the users got distracted, could not watch the video clip any more and these levels were reported to be upper limits for synchronization. Steinmetz's report gives synchronization requirements based on human perception research that was carried in the years prior to 1996.

6.1 Audio

Synchronization between two different audio tracks depends on whether the tracks are tightly or loosely coupled. For applications with tightly coupled tracks, e.g. stereo, the offsets allowed are of the order of ±11 µs spanning over 22 µs, which was derived based on the observation that one sample offset at a sample rate of 44 kHz can be heard. For applications with loosely coupled audio tracks, e.g. audio broadcast with background music, the offsets of the order of ±120 ms were found to be acceptable in Steinmetz's experiments [29].

6.2 Video

6.2.1 Animation

Involves a display of video frames to create a moving picture, where the temporal relationship between frames changes the continuity of the motion picture. The acceptable offsets are of the order of 3 video frames [29]. When the video frame rate is 25 frames/s, each frame lasts about 40 ms resulting in an acceptable offset of ±120 ms. However, a frame rate of 15 frames/s was found to be acceptable for application that include video conference, animation and sports [30]. For application on small screens, e.g. mobile phones, the preferred video frame rate may be lower.

6.2.2 Audio and Video

Performance limits for synchronization between audio and video are available in ANSI TR-45 [31] for speech packetization, ITU-R Recommendation BT.1359-1 [32] for broadcast operations, ETR 297 [33] for video telephony and ANSI T1.522 [34] for video conferencing based on user experience. ANSI TR-45 [31] and Steinmetz's work on human perception [29] report user acceptance levels for synchronization between audio and video streams. Both works involved showing recorded video clips to users with different offsets between audio and video. The reports did not consider intra-stream synchronization as it is determined by the underlying technology and the media play-out speeds. A common finding in both the works was that video leading audio was more acceptable than video lagging behind audio resulting in asymmetrical offsets. The asymmetry in tolerance thresholds can be explained based on user experience of receiving video before the audio, due to faster light propagation than the acoustic wave propagation.

ANSI TR-45 reports synchronization measurements that involved playing isolated nonsense Dutch words and taking the audience responses. Upper limits were set when the number of correctly repeated words fell below 50 %. The report concluded that synchronization difference of the order of -80 ms (audio lagging behind video) and +80 ms (audio leading video) had no effect on intelligibility. De-synchronization levels of -280 ms and 160 ms resulted in a negative effect in speech perception and these are considered to be the upper limits for offset between audio and video.

Steintmetz's experiments found that the errors in synchronization were not noticeable when timing differences between audio and video were within -80 ms (audio lagging behind video) and +80 ms (audio leading video). Severe annoyance levels were observed between -240 ms and 160 ms.

TR 102 479 [35] reports some studies that have been conducted prior to Steinmetz. These studies found that larger mismatch between audio and video, that can go up to 80 ms to 100 ms were tolerated. The numbers are different when the experiments involved human infants.

6.2.2.1 Conversational Service

Multimedia conversation service includes video telephony or any other telephony with multiple media, between two parties. The video application is available with several user dependent options [33]. An end user can decide to add video in the middle of a conversation. ETR 297 [33] specifies that the delay difference of -40 ms (audio lagging behind video) and 20 ms (audio leading video) should not be exceeded to preserve synchronization between audio and video channels in video telephony.

6.2.2.2 Distribution service

Timing requirement between sound and video for broadcasting application is set to be -185 ms and +90 ms in ITU-R Recommendation BT.1359-1 [32]. However, no reference was given for these numbers. In 2003, Systems Evaluation working group of the Advanced Television Systems Committee (ATSC) studied these numbers and concluded that the range is wider and can not achieve satisfactory audio, video synchronization for DTV applications. ATSC proposed that synchronization between audio and video to be -60 ms to +30 ms in their Doc.IS-191 [36] for DTV broadcasting that includes High Definition Television (HDTV), Standard Definition Television (SDTV) and satellite direct-to-home broadcasting.

6.2.2.3 Conferencing Service

Acceptable levels for audio and video synchronization were set to be between -80 ms and +80 ms for desktop video conferencing in ANSI T1.522 [34] based on Steinmetz's work. ANSI T1.522 [34] also specifies a perceptible value of -40 ms and +20 ms below which any changes in synchronization will not result in improved performance and stated that the perceptible level was derived from the traditional requirement for broadcast television synchronization and was based on earlier work than mentioned in the ATSC Doc.IS-91 [36].

For services that do not belong to either of these categories, synchronization between audio and video may be taken from Steinmetz's user perception measurements.

6.3 Data and Audio

Service with data as media involves still pictures or images, text and cursors or pointers.

6.3.1 Pointers and Audio

When graphics and audio are involved together, e.g. in a presentation given from a remote site and all the participants viewing from their desktops, synchronization between a shared pointer or cursor and audio becomes important as the speaker uses the pointer to show parts of the graphics belonging to the discussion. However, synchronization between audio and pointer is less observable than audio and video synchronization. Steinmetz's experiments found that synchronization between -500 ms and +750 ms is adequate with -ve values indicating pointing ahead of audio. It was noted that the viewers became distracted and could not follow the presentation when the offset spanned beyond -1 s and beyond +1,25 s [29].

6.3.2 Still pictures and audio

The combination is usually seen with slide shows where, the user perception levels depend on the latencies involved in manually changing slides on a slide projector and are of the magnitude of 500 ms with a total span of 1 s [29].

6.3.3 Text and Audio

Multimedia services that use these two media together are seen with audio annotations, e.g. online dictionary/encyclopaedia, music accompanied by notes display. Another application is in education where one language is spoken and another is displayed. User perception levels were derived by Steinmetz based on the duration of the pronunciation of short words that are of the order of 500 ms and were set at ±240 ms [29].

6.4 Data and video

6.4.1 Text and video

When subtitles need to be displayed with video, tight coupling of temporal relationships between the text and the video is expected. Based on the application, the text is also expected to be displayed closer to video. Offset levels of ±240 ms were found to be sufficient [29].

6.4.2 Still pictures and audio

Video accompanied by still pictures or images do not have the same tight correlation as above. Typical applications include video show of two different views of a scenario with one a still picture and the other a motion video segment. The offset is the same as that for audio with still pictures [29].

Table 2: Acceptable timing differences between different media

Media involved	First media ahead of the second media	First media behind the second media	Service
Audio and Audio [29] Tightly coupled Loosely coupled	 11 µs 120 ms	 11 µs 120 ms	
Video and audio	80 ms	80 ms	Conferencing [34]; Retrieval
Video and Audio	40 ms	20 ms	Conversational [33]
Video and Audio	60 ms	30 ms	Distribution [36]
Pointers and audio	500 ms	750 ms	Conferencing [29]
Still pictures and audio [29]; Still pictures and video [29]	500 ms	500 ms	Conferencing ; Retrieval
Text and audio [29]; Text and video [29]	240 ms	240 ms	Distribution; Conferencing; Retrieval

7 Analysis of timestamp methods

There are two metrics to describe the quality of timing provided by a clock - precision or timing accuracy with which it can be set and read, and frequency accuracy or stability that determines how well the timing can be maintained after being set. The achievable timing performance when using time stamp methods can be expressed by Equation (11), as a function of [12]:

- reading errors due to uncertainty in delay estimations; and
- internal clock skew since the last resynchronization/adjustment time

$$\text{Achievable performance} = \mathbf{f}(\text{reading errors, clock skew}) \qquad (11)$$

Where, **f** indicates a function and depends on the algorithm used for synchronization.

7.1 Clock frequency stability

Timing errors (or offset) result from the clock skew relative to the last adjustment. The frequency stability of the clocks embedded in the end points, from where the media is transmitted and received by an end user - computers, workstations, PDAs etc., depends on the temperature fluctuations, power supply and vibration experienced.
E.g. Ethernet specifies 0,01 % (100 ppm) as the absolute frequency accuracy. Frequency errors are classified based on the measurement time intervals. The time intervals can change from a few seconds to a few hours.

- Noise - with time interval < 1 minute:
  - Sources: voltage regulations and vibration.
  - Minimal contribution to frequency stability.
- Short term stability or Wander - with 1 minute < time interval < 1 hour:
  - Sources: ambient temperature.
  - Major contributor to frequency error as the temperature coefficients are of the order of 1 ppm/^{0}c to 2 ppm/^{0}c, sometimes going into the range of 15 ppm/^{0}c to 20 ppm/^{0}c for most of the end terminals described above.

- Long term stability or mean frequency error - time interval > 1 hour:
  - Sources: timing algorithm.
  - Not a significant contribution except for the initial transient with the onset of algorithm in the case of statistical synchronization.

7.2 Timestamp reading errors

The timestamp reading errors arise from delay uncertainty and clock resolution. In software based synchronization algorithms, timestamps are read based on the interrupt service routine of the operating system in the receiver. This results in uncertainties due to the latency variations involved in network channel access and reception interrupts. At the time of timestamp reading, there is no method to distinguish measurement errors from network delay uncertainties. Major contribution to the delay uncertainty comes from:

- the variation experienced by timestamp point to Layer 1 of the OSI stack;
- buffering involved in the transmission of the time stamp;
- delay fluctuations in the access networks, e.g. a network using radio access for one host and satellite for the other.

Based on the type of technology used to gain access to the network, the error induced can be substantial. The farther the timestamp point is from the physical layer, the greater the expected delay variation in the OSI stack. However, some timestamp based synchronization is more immune to delay variation, as statistical methods are used to calculate delays and offsets that tend to be more accurate than the other time stamping methods.

Another factor that contributes to timing precision is clock resolution. The local clock is normally represented by a combination of a counter/divider and an oscillator. The output frequency of the clock is a multiple or divider of the oscillator frequency and the interval at which the counter/divider is incremented. The exact value of the multiplying/dividing factor and clock granularity are unknown. However, the intervals at which the same clock is read are much larger than this resolution, making clock resolution a minor contributor to timestamp reading error.

From the above discussions, the terms that can make a timestamp different from the actual clock reading is given by Equation (12).

$$\Delta t_s t_c = r_e + f\tau \qquad (12)$$

Where: r_e reading error;

f frequency error of the clock oscillator;

τ time interval since this reading.

The time stamp value is given by adding Equation (12) to the clock reading at any arbitrary time. The variables r_e and f are random in nature and are bounded by different values based on the clock characteristics, algorithm and the actual implementation. E.g. frequency error is bounded the maximum frequency tolerance of the oscillator which is specified by the manufacturer based on its design. Frequency tolerance is expressed using positive and negative values centered around the frequency of operation and is expressed in parts per million or ppm or µs/s. Similarly the reading error due to the delay uncertainty can be bounded with a maximum value equal to half the RTT. For a system using the same synchronization method to derive the timestamps, for two clients located at the same synchronization hierarchy that would involve same delay uncertainties in deriving the timestamps, difference in timing of the two clients comes from the frequency factor - $f\tau$, given in (12).

An example scenario shown in Figure 10, is used to generate two media streams using two separate sources. Assume that the streams are generated at the same time. To achieve guaranteed service satisfaction, the two streams need to be played to the end user at the same time or within the appropriate bounds for the service. With RTP, the streams will be sent from Source #1 and Source #2 with different RTP timestamps based on the type of media. RTCP SRs will follow with timestamp pairs indicating the wall clock time (set based on NTP or SyncUTC or PTP) corresponding to the origination time of the streams. Once the absolute time is set in these sources, the difference in values of the local time maintained by Source #1 and Source #2 comes from the skew between the local clocks as explained above. The values of the timestamps at any arbitrary time would be given by equations (13) and (14).

$$(\Delta t_s t_c)_1 = r_e + (f\tau)_1 \text{ for Source\#1} \quad (13)$$

$$(\Delta t_s t_c)_2 = r_e + (f\tau)_2 \text{ for Source\#2} \quad (14)$$

where, the subscripts 1 and 2 represent the Sources #1 and #2 respectively. The values of the reading errors were assumed to be the same.

For sources with zero relative skew, this would result in a zero time difference at any arbitrary time of measurement. The worst case difference in absolute times maintained by the two sources occurs when they operate at the extreme edges of the boundaries set for the frequency tolerance.

For this example, the frequency tolerance is assumed to be ±100 ppm for both the sources. The worst case difference in the absolute time maintained by the sources occurs when Source #1 operates at -100 ppm and Source #2 operates at +100 ppm resulting in a difference of 200 ppm or 200 µs/s. If τ = 1 024 s, the timing of the two streams would differ by 200 ms at the receiver instead of being presented simultaneously. Based on the duration over which this relative drift exists between the two sources, some of the services listed in Table 2 would be unsatisfactory from the QoS perspective of the customer.

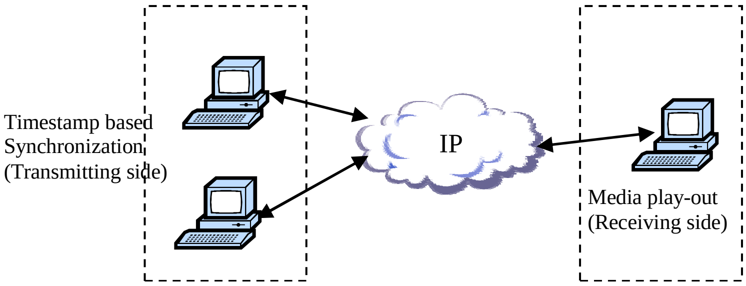

Figure 10: Example scenario

8 Media synchronization in services offered over a combination of TDM and IP networks

When a multimedia service is offered over a combination of TDM and IP networks as shown in Figure 2, synchronized media play-out is not as direct as in native IP-based services. TDM and IP networks use different timing mechanisms and provide service synchronization differently. Service synchronization in IP-based networks can be achieved using time stamps as given above that use absolute clock for a synchronized media play-out at the receiver. TDM networks use Layer 1 clocks to bound the data bit rate within standard based limits. Service synchronization is accomplished using clocks in the end points that are traceable to the same reference. When multiple media components of an offered service are carried using TDM and IP networks, currently there is no unique mechanism in existence to offer synchronized service at the receiver. The quality of such a service from end user perspective will be compromised. The following lists the factors that influence the media transport.

- Delays experienced.
- Conceptual differences in using timing to provide synchronized service.

Service synchronization can be provided to a certain level by delaying the TDM media component by an amount equivalent to that experienced by IP media component in coding/decoding and encryption/decryption processes.

History

Document history		
V1.1.1	March 2007	Publication