Time Sensitive Application Transport

Mohammad Shafahi

University of Amsterdam
Faculty of Science
Introduction

- Time sensitive applications are becoming more and more demanded.
- The demand of TSAs has introduced discussions in the network community, specially the NRENS community.
- The NRENS main question is “What transport technology best suites time sensitive applications?”
Research Question(s)

- What are the sources of delay and jitter in technologies?
- What are the magnitudes of each source of delay and jitter?
- What are the requirements for measuring delay and jitter?
- What are the suitable methods and tools for measuring delay and jitter?
What are TSAs?
# Jitter requirements of TSAs

<table>
<thead>
<tr>
<th>Video</th>
<th>Audio</th>
<th>Application</th>
<th>Jitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td></td>
<td>Overlay image</td>
<td>240ms</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td>lip synchronization</td>
<td>80ms</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td>Music Script</td>
<td>5ms</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td>Dialog</td>
<td>120ms</td>
</tr>
<tr>
<td>✔</td>
<td></td>
<td>Background music</td>
<td>500ms</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td>Stereo</td>
<td>5µs</td>
</tr>
</tbody>
</table>
What are the sources of delay and jitter in technologies?
Sources of delay in a network node

\[ \text{delay}_t^i = RD_t + QD_{1t} + PCD_t + QD_{2t} + TD_t + PPD_t \]
Sources of delay in a network node

$\text{delay}_t^i = RD_t + QD_1_t + PCD_t + QD_2_t + TD_t + PPD_t$
Delay in a network path

\[ \text{delay}_t = \sum_{i=1}^{N} (RD_t^i + QD1_t^i + PCD_t^i + QD2_t^i + TD_t^i + PPD_t^i) - PD_t^N \]
Delay in a network path

\[
\text{delay}_t = \sum_{i=1}^{N} (RD_t^i + QD1_t^i + PCD_t^i + QD2_t^i + TD_t^i + PPD_t^i) - PD_t^N
\]
Definition of jitter

\[ Jitter_i = \text{delay}_i - \text{delay}_{i-1} \]

\[ Jitter(\delta t) = \frac{\sum_{i=1}^{N} |jitter_i|}{N} \quad (1) \]
Definition of jitter

\[ Jitter_i = \text{delay}_i - \text{delay}_{i-1} \]

\[ Jitter(\delta t) = \frac{\sum_{i=1}^{N} |jitter_i|}{N} \quad (1) \]
## Sources of delay

<table>
<thead>
<tr>
<th>Tech/Delay</th>
<th>Propagation</th>
<th>Transmission/Reception</th>
<th>Processing</th>
<th>Queuing</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDM</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D)WDM</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SONET/SDH</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OTN</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Openflow</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>MPLS-TP</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>PBB-TE</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Optical</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diffserv</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>
What are the magnitudes of each source of delay and jitter?
# Magnitudes of delay

<table>
<thead>
<tr>
<th>Source of Delay</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propagation</td>
<td>4.9 µs/km</td>
</tr>
<tr>
<td>Transmission</td>
<td>0.01 µs/kb*</td>
</tr>
<tr>
<td>Processing</td>
<td>2 µs/kb**</td>
</tr>
<tr>
<td>Queuing</td>
<td>Used Memory</td>
</tr>
<tr>
<td></td>
<td>Service Speed</td>
</tr>
</tbody>
</table>

* For a 100Gb interface  
** Could be much less
What are the requirements for measuring delay and jitter?
One Way Delay and Jitter

\[ T_{STS} = \text{Send Time Registered} \]
\[ T_{RTS} = \text{Receive Time Registered} \]
\[ T^S = \text{Actual Send Time} \]
\[ T^R = \text{Actual Receive Time} \]
One Way Delay and jitter

\[ T^{STS} = \text{Send Time Registered} \]
\[ T^{RTS} = \text{Receive Time Registered} \]

\[ T^S = \text{Actual Send Time} \]
\[ T^R = \text{Actual Receive Time} \]
One Way Delay and jitter

$T_{STS_1}, T_{STS_2}$
$T_{S_1}, T_{S_2}$

$T_{RTS_1}, T_{RTS_2}$

$T^{STS} = \text{Send Time Registered}$
$T^{RTS} = \text{Receive Time Registered}$

$T^S = \text{Actual Send Time}$
$T^R = \text{Actual Receive Time}$
Round Trip Time and jitter

$T_{STS} = \text{Send Time Registered}$

$T_{RTS} = \text{Receive Time Registered}$

$T_S = \text{Actual Send Time}$

$T_R = \text{Actual Receive Time}$
Round Trip Time and jitter

$T_{STS} =$ Send Time Registered
$T_{RTS} =$ Receive Time Registered
$T_S =$ Actual Send Time
$T_R =$ Actual Receive Time
Round Trip Time and jitter

$T_{STS1}, T_{STS2}$

$T_{S1}, T_{S2}$

$T_{RTS1}, T_{RTS2}$

$T_{STS} = \text{Send Time Registered}$

$T_{RTS} = \text{Receive Time Registered}$

$T^{S} = \text{Actual Send Time}$

$T^{R} = \text{Actual Receive Time}$
Calculating delay and jitter

delay_i^o = delay_i^m + delay_i^n

delay_i^n = T_i^R - T_i^S

delay_i^m = T_i^S - T_i^{STS} + T_i^{RTS} - T_i^R

jitter_i^o = delay_i^o - delay_{i-1}^o

jitter_i^n = delay_i^n - delay_{i-1}^n

jitter_i^m = delay_i^m - delay_{i-1}^m
Calculating delay and jitter

\[ \text{jitter}^m(\delta t) + \text{jitter}^n(\delta t) = \sum_{i=1}^{N} \left| \text{jitter}^m_i \right| + \sum_{i=1}^{N} \left| \text{jitter}^n_i \right| \]

\[ = \frac{\sum_{i=1}^{N} \left| \text{jitter}^m_i \right| + \left| \text{jitter}^n_i \right|}{N} \]

\[ \geq \frac{\sum_{i=1}^{N} \left| \text{jitter}^0_i \right|}{N} = \text{Jitter}^0(\delta t) \]
Calculating delay and jitter

\[
delay^m(\delta t) + delay^n(\delta t) = \frac{\sum_{i=1}^N |delay^m_i|}{N} + \frac{\sum_{i=1}^N |delay^n_i|}{N}
\]

\[
\geq \frac{\sum_{i=1}^N |delay^o_i|}{N} = delay^o(\delta t)
\]
Measuring Delay and jitter

- One way delay measurement (OWD)
  - requires synchronous clock for measuring delay (Using NTP + GPS)
  - Works fine without synchronous clock for jitter

- Round Trip Time measurement (RTT)
  - Assumes your out going and return path and delay are the same/equal (Use the same!)
What are the suitable methods and tools for measuring delay and jitter?
Tools for measuring delay and jitter

- **Hardware Solutions**
  - Expensive
  - Higher accuracy

- **Software Solutions**
  - Cheep and many open source
  - Lower accuracy
Tools for measurement

- **Hardware Solutions**
  - Custom hardware
    - Passive
    - Active
  - Network devices with OAM support
- **Software**
  - Libraries
  - Applications
D-ITG

- Supports OWD and RTT measurement
- Supports multiple flows at the same time
- Supports multiple senders and receivers
- Supports custom traffic patterns
- Supports a variety of protocols
Measurement accuracy setup

Pentium D 3GHz
2 cores
8GB ram
1 Gbit Ethernet

Cat5e
x-connect 8m

Pentium Xeon
3GHz 8 cores
16GB ram
1 Gbit Ethernet
Jitter density function of Setup
Measurement accuracy setup 2

Pentium D 3GHz
2 cores
8GB ram
1 Gbit Ethernet

PowerConnect 6248

Pentium Xeon 3GHz 8 cores
16GB ram
1 Gbit Ethernet
Jitter density function of Setup

Jitter density function for vanilla_voip

Density

Jitter(us)
Estimation of delay and jitter

Central Limit theorem:

The mean of a sufficiently large number of independent random variables each with finite mean and variance will approximately have a normal distribution.
POC measurement setup

SUT University, Iran
- AMD Athlon 64
  - 3.4GHz 1 cores
  - 2GB ram
  - 1 Gbit Ethernet

UvA University, Netherlands
- Pentium Xeon
  - 3GHz 8 cores
  - 16GB ram
  - 1 Gbit Ethernet
Jitter density function of setup

<table>
<thead>
<tr>
<th>SD</th>
<th>14.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>96.8</td>
</tr>
<tr>
<td>N</td>
<td>12000</td>
</tr>
</tbody>
</table>
Conclusion

- Classified the sources of delay and provide a magnitude for them
- Provided a framework for discussion about delay and jitter
  - We have to be careful with arithmetics
- Showed that software solutions provide accuracies of microseconds in jitter measurements
- Estimate that hardware is more accurate
Further Work

- Studying hardware solutions
- Looking into different implementations of OAM in Network devices
- Adding queueing theory and system compression theories to the framework
Thanks

- Cees de laat for his great guidance
- The OS3 Research group
- My wife, fahime for her patience
Questions?
mohammad.shafahi@os3.nl
mohammad.shafahi@gmail.com