An Empirical Evaluation of VoIP Playout Buffer Dimensioning in Skype, Google Talk, and MSN Messenger

Chen-Chi Wu, Kuan-Ta Chen, Yu-Chun Chang, and Chin-Laung Lei

ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)
Williamsburg, VA, USA
June 2009

Introduction (1 of 2)

- VoIP increasingly important
  - Started with inexpensive use at home with friends and family
  - Now businesses between corporations
- Sound quality can be comparable to traditional telephones
- Skype reports: 405 million registered users, 15 million online users [footnote 1]
- Reliable service and quality a priority for ISP and VoIP providers

Introduction (2 of 2)

- Many factors impacting quality
- (This class talks about a lot of them!)
  - Codec, Transport protocol, Redundancy and Error Control, and Playout Buffer
- This work focuses on the *Playout Buffer*

Buffering Basics

- Sacrifice speech conversational interactivity for better sounding quality playout
  - "Smoother" sound, plus could repair loss
- Typically, transmit packets every 30 ms, but can arrive later than 30 ms from previous (delay jiiter)
  - Results is silent periods, noise, unclear speech (depending upon loss concealment)
- So, *playout buffer* holds packet temporarily in order to allow more packets to arrive on time

Buffering Challenge

- How to determine best playout buffer size to use?
- Larger buffer leads to better sounding voice quality, but lower interactivity and vice versa
- Optimal size affected by network delay, delay jitter, repair and compression (codec) implementations
  - And network factors may change over time, so buffer size should too!

Buffering in Practice

- Academics proposed many algorithms [9-11, 13]
- Most adjust buffer based on linear combination of network delay and jitter
  - Combinations vary with network measurements
- But what algorithms are used in practice?
- Analyze 3 popular VoIP applications: *Skype*, *Google Talk*, *MSN Messenger*
  - Do they differ?
  - Do they adjust?
  - How close to "optimal"?

Outline

- Introduction
- Related Work
- Experiments
- Results
- Optimal
- Conclusion

Related Work (1 of 2)

- [11]: Authors use weighted exponential moving average of delay and standard deviation to determine buffer
  - weights are hard-coded
- [10]: extends [11] by adapting the weights according to magnitude of events
  - Both [10] and [11] by simulation
- [9,13]: extend by adjusting during talk spurt so can adapt to changes in network more quickly
- Above, all academic systems

→ What is used in practice?

Related Work (2 of 2)

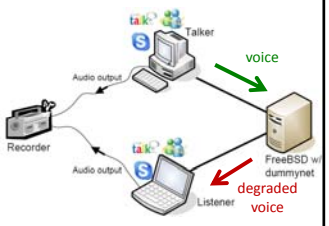
- To assess, Perceptual Evaluation of Speech Quality (PESQ) [8]
  - Compare original to degraded, and map to Mean Opinion Score (MOS), value 1-5.
- E-Model has arithmetic sum of impairments of delay, equipment and compression [7]
  - R = 94 – *i*(delay) – *i*(loss) → *R factor*, can map to MOS
- Neither is sufficient. PESQ does not use delay, E-model not accurate nor combines delay and quality
- [5] combines both
  → Use their technique (later)

Outline

- Introduction
- Related Work
- Experiments
- Results
- Optimal
- Conclusion

Experiment Methodology

- Free BSD w/dummynet as router
  - Control loss, delay, jitter (stddev of delay)
  - Link is 1 Mb/s
- 2 PCs running Windows XP with Skype, Google Talk, MSN Messenger
  - One PC "talker" the other "listener"
- Play recording on talker, send to listener
  - Recording from Open Speech Repository [3]
- Record both talker and listener speech
  - Compare to get degradation



- Each "call" 240 seconds
- 10 calls at each setting

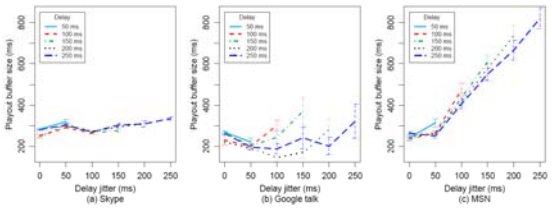
Buffer Size Estimation

- Have two audio samples. Compare to determine delay (use cross-correlation coefficient [1])
  - (MLC: not validated as a technique?)
- Note, not sure of sample interval, compression, etc. ("black box")
  - But, estimate to be 50 msec based on literature
- May not be totally accurate, but want to see how commercial VoIP applications adjust

Outline

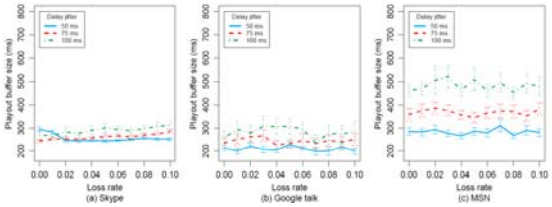
- Introduction
- Related Work
- Experiments
- Results
- Optimal
- Conclusion

Network Delay and Jitter



- Delay:
  - Skype doesn't adjust
  - MSN doesn't adjust
  - Google may (fig b, trendlines differ).
- Jitter:
  - Skype flat, so doesn't adjust
  - Google adjusts slightly, lots of variance
  - MSN adjusts linearly

Network Loss Rate



- All flat, so no apparent adaptation

Outline

- Introduction
- Related Work
- Experiments
- Results
- Optimal
- Conclusion

QoE Measurement Model

- Based on [5] …
- Given original and degraded clips
- Apply PESQ to get MOS
- Convert MOS to R score
  - Using formula in ITU-T G.107 [7]
- Compute *delay impairment* (I_d) from E-model

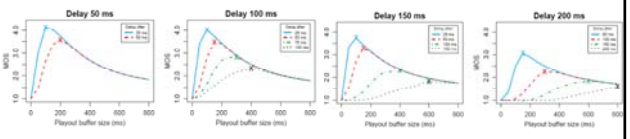
  $I_d = 0.024 \times d$ if $d < 177.3$

  $I_d = 0.024 \times d \times (d - 177.3)$ if $d > 177.3$
- Subtract I_d from R score to get R'
- Convert R back to MOS

Determining Optimal Buffer Size

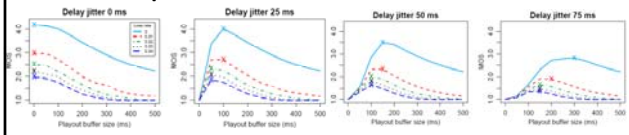
- Yields best quality (QoE, previous slide)
- Encode audio clips from open speech repository [3] to VoIP using [2]
  - Use G.711, popular codec
- Simulate any loss (using Gilbert model)
- Add delay (Gamma distribution)
  - If later than buffer size, drop
    - (MLC: what policy is this?)
- Decode any resulting stream
- Apply QoE to determine quality

Optimal Buffer Size with Delay and Jitter



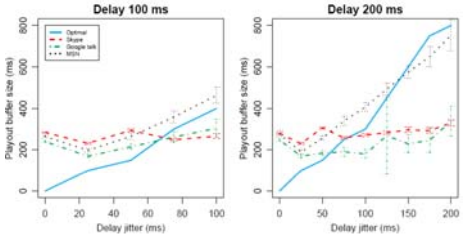
- As expected, MOS decreases with delay (sanity check)
- MOS varies a lot with buffer size
  - Important to get buffer size right
- Optimal indicated by 'X'
  - As jitter increases, more delay is necessary

Optimal Buffer Size with Loss



- Method: Delay all 100 ms, add loss
- Loss degrades MOS (sanity check)
- With jitter, optimal point shifts left with higher loss
  - May be different with repair (future work)

Optimal for Skype, Google, MSN



- (They don't adjust for loss, so no further analysis)
- All are conservative (~220 ms buffer) with no jitter
- MSN adapts best with jitter, others too conservative

Model for Determining Optimal Buffer Size

- Can derive optimal via simulations
  - But lot of work, not real-time
- Try regression to determine under network scenario

Optimal buffer = $(constant) + coef_{delay} \cdot delay +$
$coef_{delay \cdot jitter} \cdot delay \cdot jitter +$
$coef_{delay \cdot jitter \cdot plr} \cdot delay \cdot jitter \cdot plr.$

- Delay – average network delay, jitter – std of delay, plr - packet loss rate
- For G.711, coefficients are below, R^2 is 0.885 (good)

Variable	Coef
$(constant)$	157
$delay$	−1.05
$delay \cdot jitter$	0.02
$delay \cdot jitter \cdot plr$	−0.57

Conclusions

- Investigate if gap between academic research and practice exists
  - MSN Messenger, Skype, Google Talk
- MSN best in terms of buffer dimensioning
- Skype, does not adjust much at all
- Provide algorithm to compute optimal based on QoE metric and model

Future Work?

Future Work

- More factors
  - Frame size
  - Repair
  - Codec
- Use optimal dimensioning model in system
  - Real-life experiments to evaluate