An Empirical Evaluation of VoIP Playout Buffer Dimensioning for

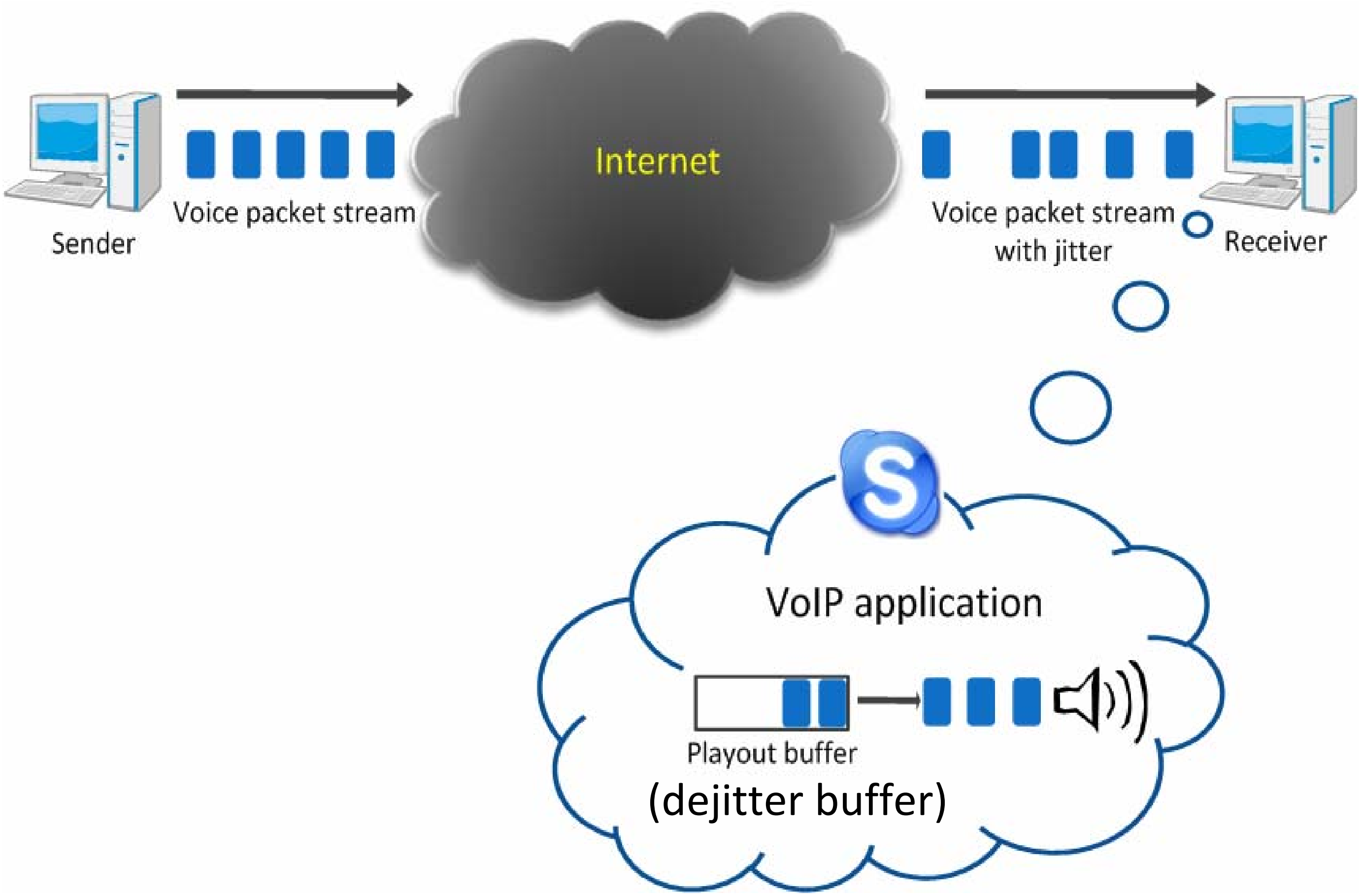
Kuan-Ta Chen, Academia Sinica

Chen-Chi Wu and Chin-Laung Lei, National Taiwan University

Chun-Yin Huang, National Taiwan Ocean University

Playout Buffering

- Voice needs to be played smoothly, but network may inject delay variance (jitters)



The Tradeoff

- As buffering sacrifices **delay** in exchange for a higher chance that **voice packets arrive on time**
- Larger buffer size →
  - better speech quality
  - lower conversational interactivity
- Smaller buffer size →
  - worse speech quality
  - higher conversational interactivity

Playout Buffer Dimensioning

- Deciding the **optimal** buffer size
  - maintain a **balance** between conversational interactivity and speech quality
- It's challenging because of so many factors
  - Network delay
  - Network delay jitter
  - Network loss
  - Codec implementation
  - Redundancy control
  - Error recovery
  - Transport protocol, etc

changes over time

Proposals on Buffer Dimensioning

- Many algorithms have been proposed
  - The **k** largest delay among the previous **m** delays [Naylor'85]
  - Inflate buffer size when packets arrive late, and shrink buffer size over time [Stone'95]
  - Weighted sum of EWMA of delay and delay jitter [Ramjee'94]
  - Automatic adjustment of EWMA weights [Narbutt'03]
  - Weight adjustment within talk bursts [Liang'03, Sreenan'03]
  - …

We are curious about …

Skype: 405 million registrars (15 million online)

Whether a gap exists between research community and software practitioners?

In other words …

Do commercial products really adopt any of these proposals?

What’s the performance of commercial products (in terms of buffer dimensioning)?

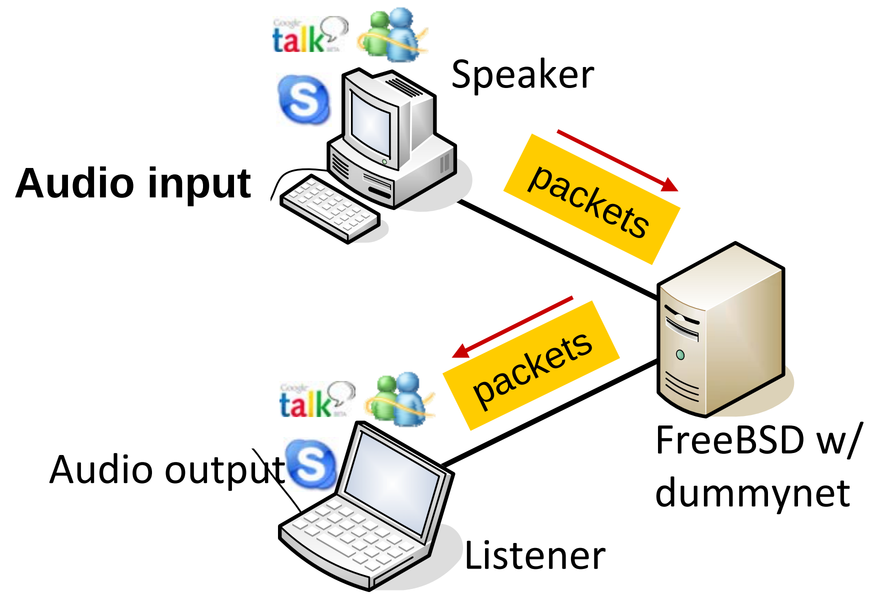
Our Contribution

- A systematic measurement **methodology** for measuring VoIP playout buffer size
- Show that the real-life VoIP applications **do not** adjust their buffer sizes appropriately
  - based on QoE measures
- A regression-based algorithm to compute the **optimal buffer size** given a network condition
  - Light-weight computation; thus can be applied in run time

Talk Progress

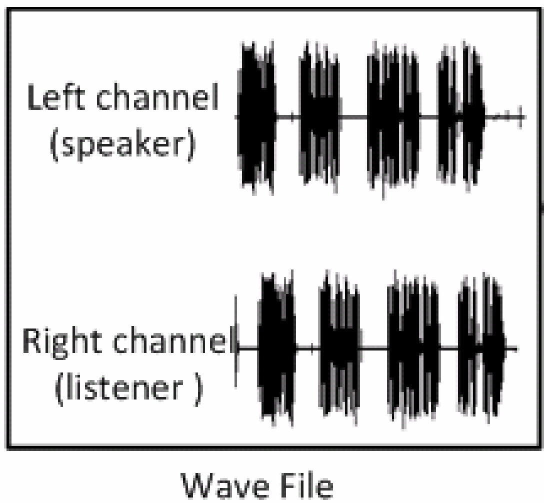
- Overview
- Measuring Applications' Buffer Size
  - Experiment Methodology
  - Measurement Results
- Deriving Optimal Buffer Size
  - Methodology
  - Derived Buffer Sizes
  - Evaluation of Applications' Dimensioning Algorithms
- Conclusion & On-going Work

Experiment Methodology



- Dummynet for controling network conditions (delay, jitter, and packet loss)
- Use a recording card (ESI Maya44) to ensure time-synchronized audio recordings from two hosts

Buffer Size Estimation



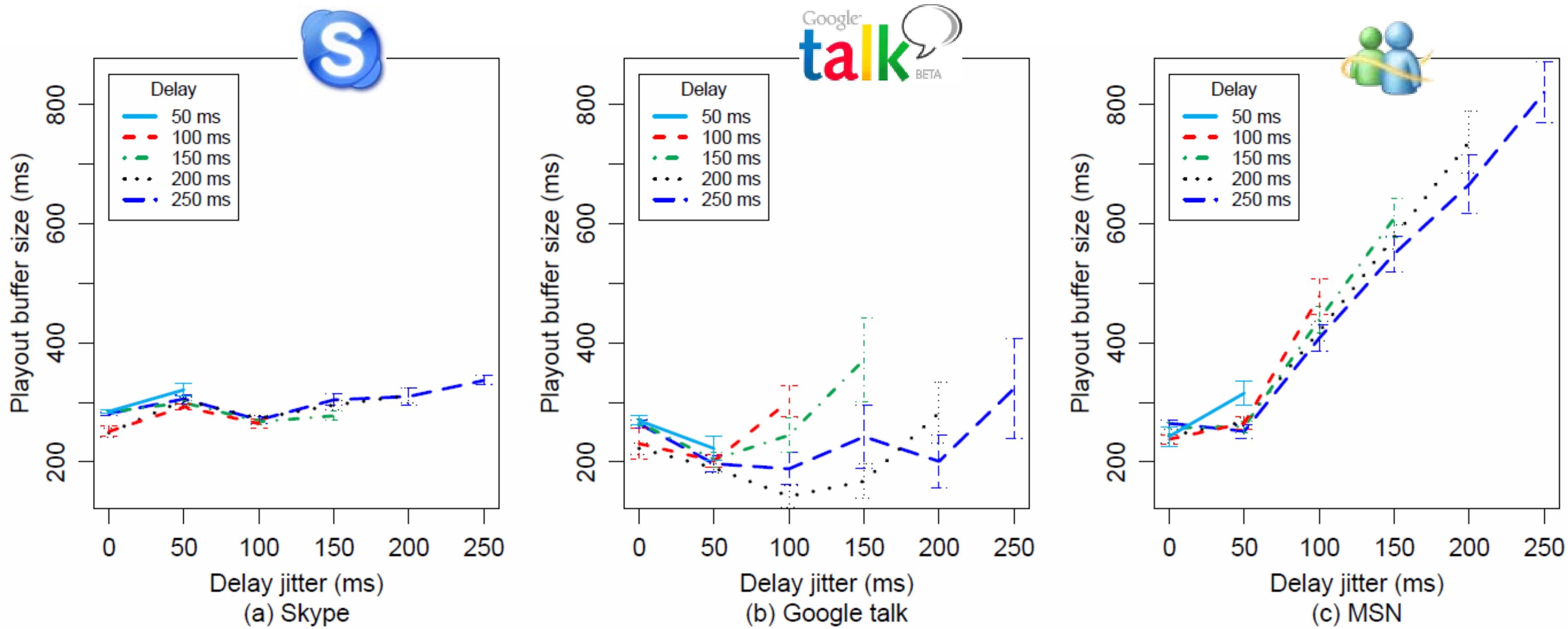
Finding best alignment (based on cross-correlation coeffcients) → end-to-end delay

- End-to-end delay components
  - Network delay (dummynet)
  - Coder delay + packetization delay (assumes 50 ms)
  - Playout buffering delay (unknown)
- Buffer size = e2e delay – network delay – 50 ms

Experiment Settings

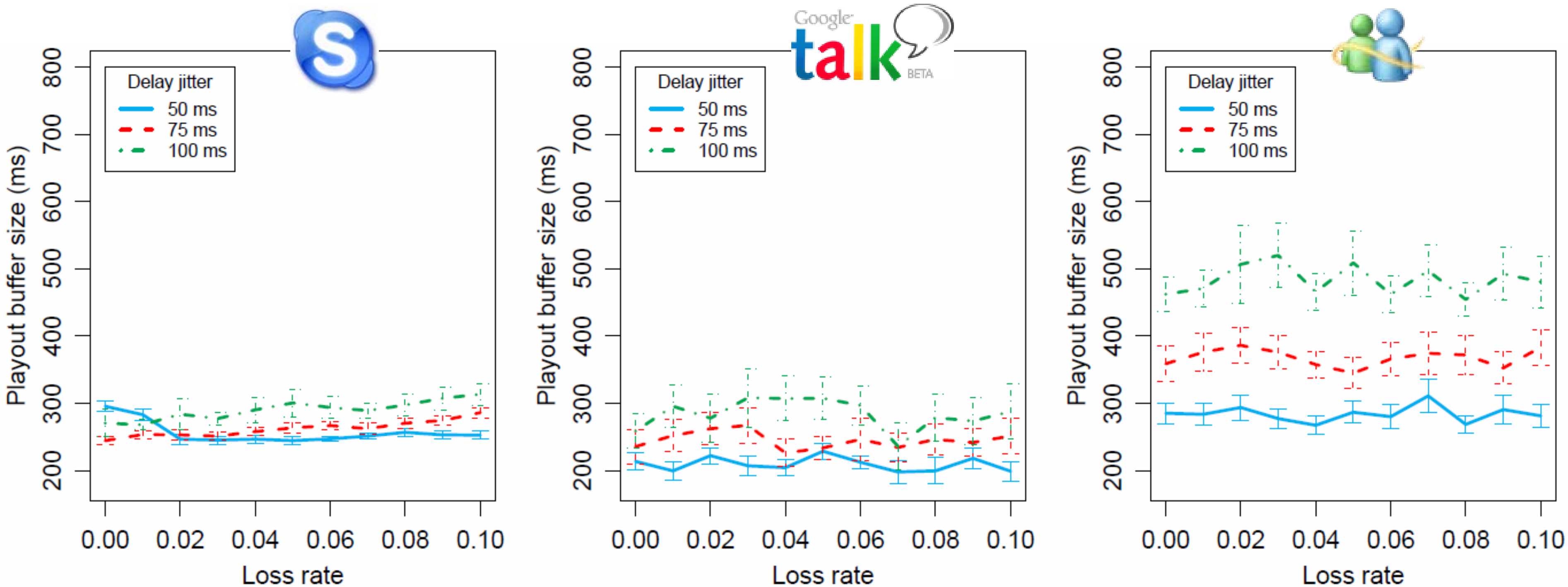
- Application: Skype, Google Talk, MSN Messenger
- Network delay & jitter: 0 ms, 25 ms, 50ms, …, 200 ms
- Network loss rate: 0%, 1%, …, 10%
- 10 VoIP calls for each app/network setting
  - Each call lasts 240 seconds

Effect of Delay and Jitter



- Skype maintains the same buffer size
- Google Talk slightly adjusts the buffer size according to delay and jitter
- MSN Messenger’s buffer size grows linearly as the jitter increases

Effect of Packet Loss



(a) Skype (b) Google talk (c) MSN

- All three applications do not adapt buffer size to packet loss

Having seen the different behavior of the applications, …

Which one application's playout dimensioning is best?

Is the best one optimal?

Talk Progress

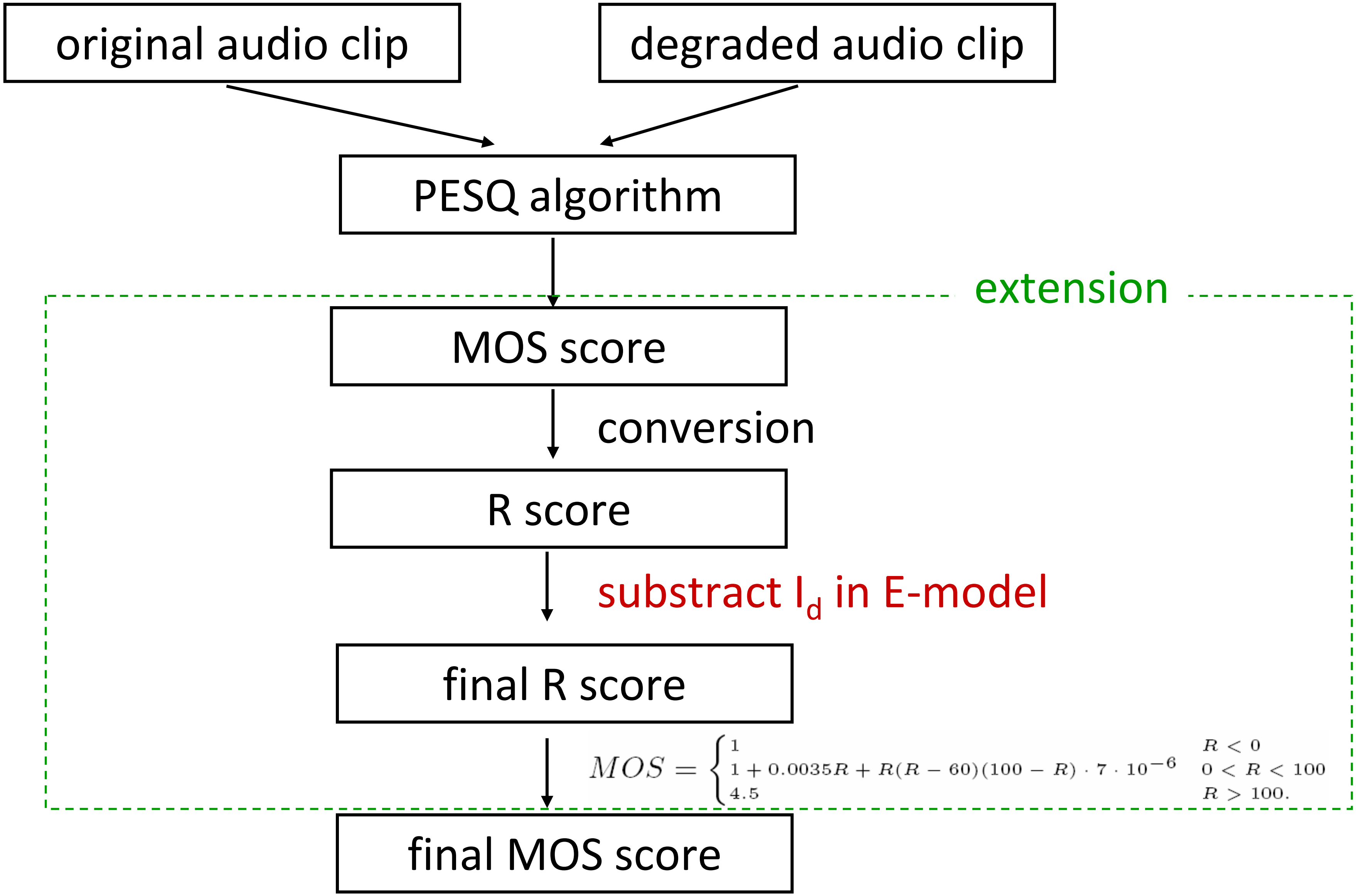
- Overview
- Measuring Applications' Buffer Size
  - Experiment Methodology
  - Measurement Results
- Deriving Optimal Buffer Size
  - Methodology
  - Derived Buffer Sizes
  - Evaluation of Applications' Dimensioning Algorithms
- Conclusion
- On-going Work

Deriving Optimal Buffer Size - QoE

- How to define the “optimal” buffer size?
  - Buffer size that yields the best quality of experience (QoE)

- How to measure the QoE of a VoIP call?
  - PESQ (ITU-T P.862, Perceptual Evaluation of Speech Quality)
    - measures **listening quality**
    - signal level, accurate
  - E-Model (ITU-T G.107)
    - measures **overall quality** (listening + interactivity)
    - network level, lightweight but not accurate in listening quality

QoE Assessment Model [Ding'03]

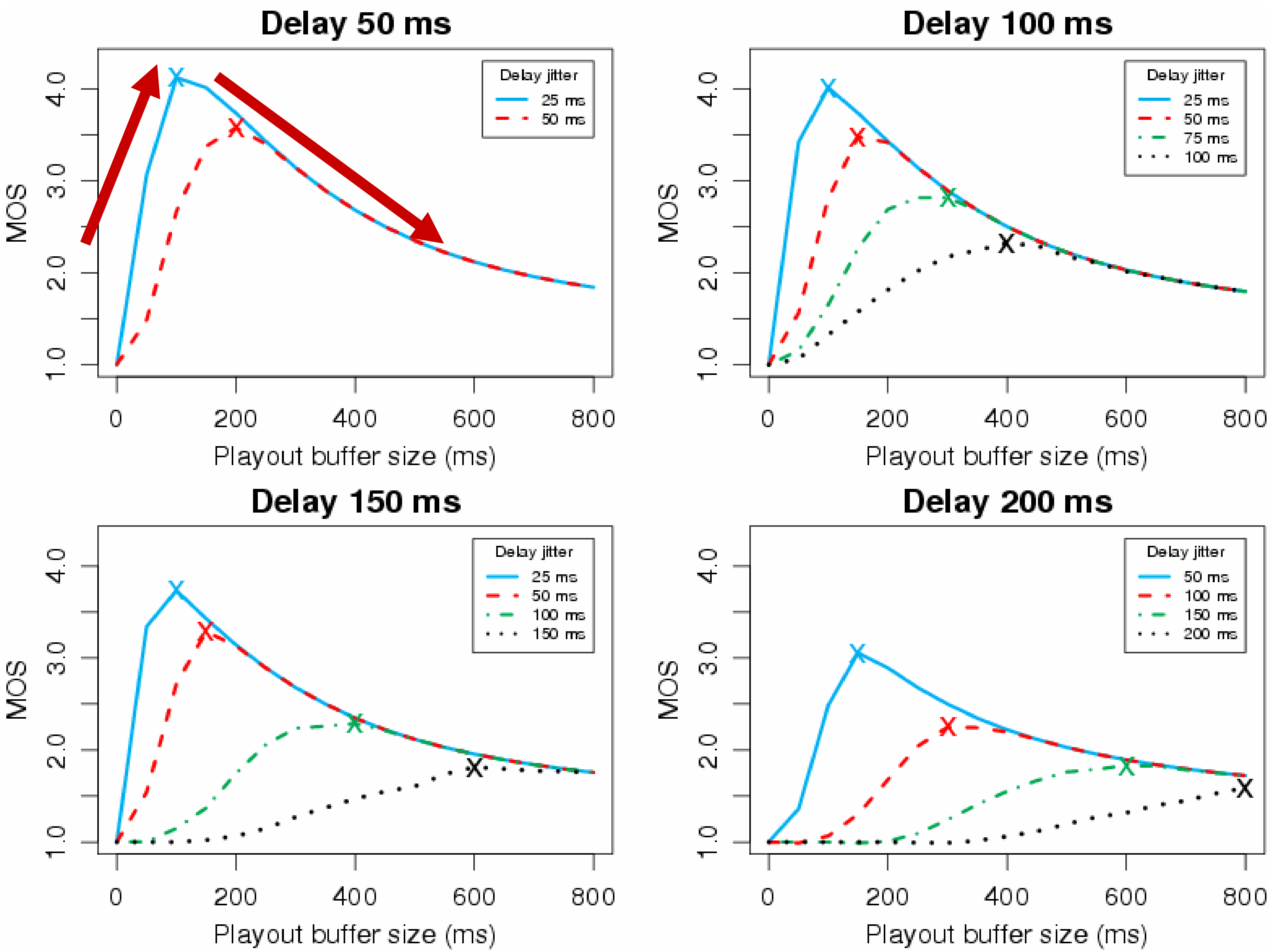


Deriving Optimal Buffer Size - Simulation

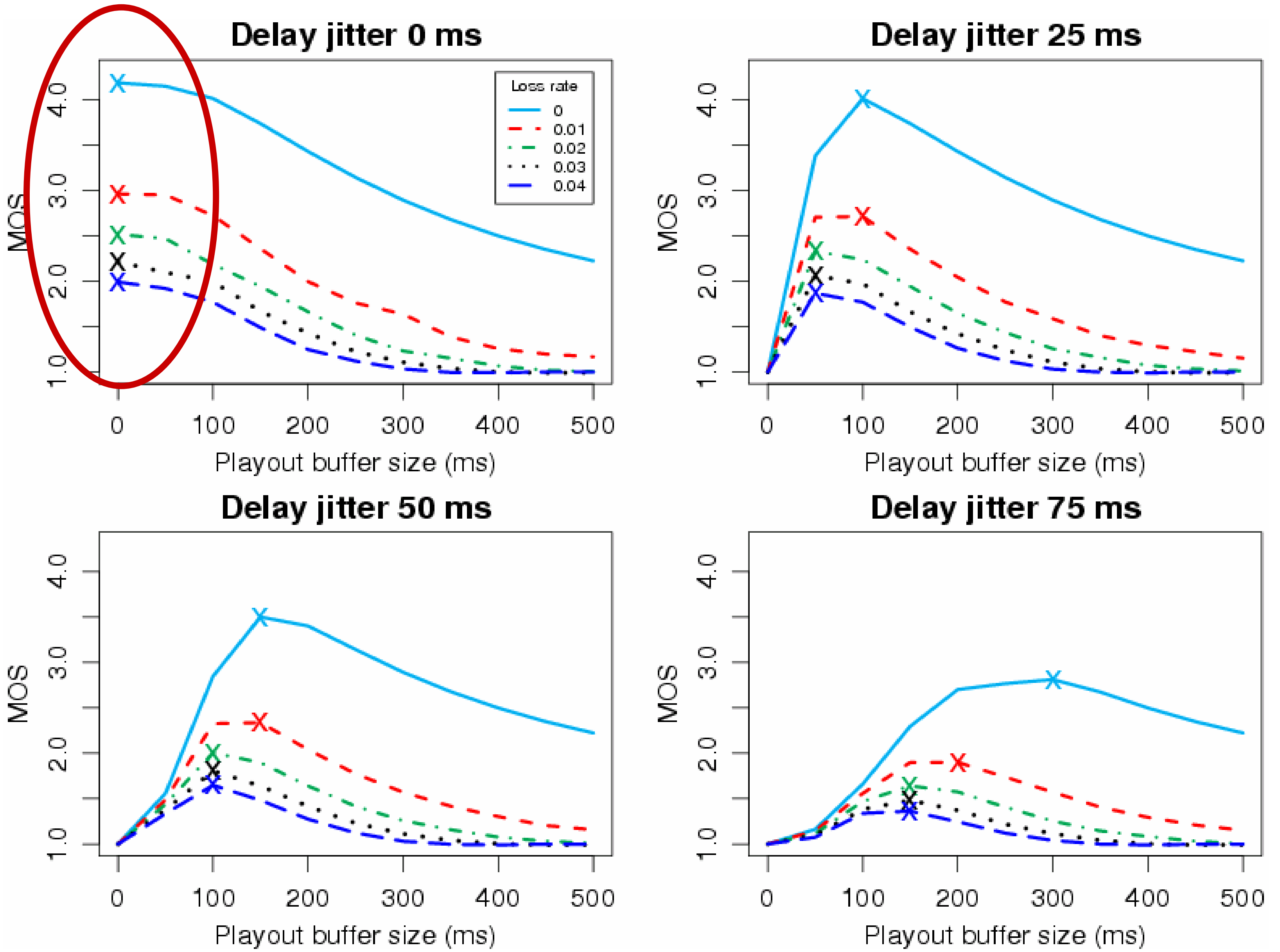
For each (buffer size, network setting)

1. Encode an audio clip into a sequence of VoIP frames
2. Impairment at the network
   - network delay & jitter
   - packet loss (Gilbert model)
3. Packet discarding at the receiver
   - drop a packet if its arrival time is later than scheduled time (sent time + playout buffer size)
4. Decode the result frames (a subset of original frames) to a **degraded** audio clip
5. Compute average QoE scores

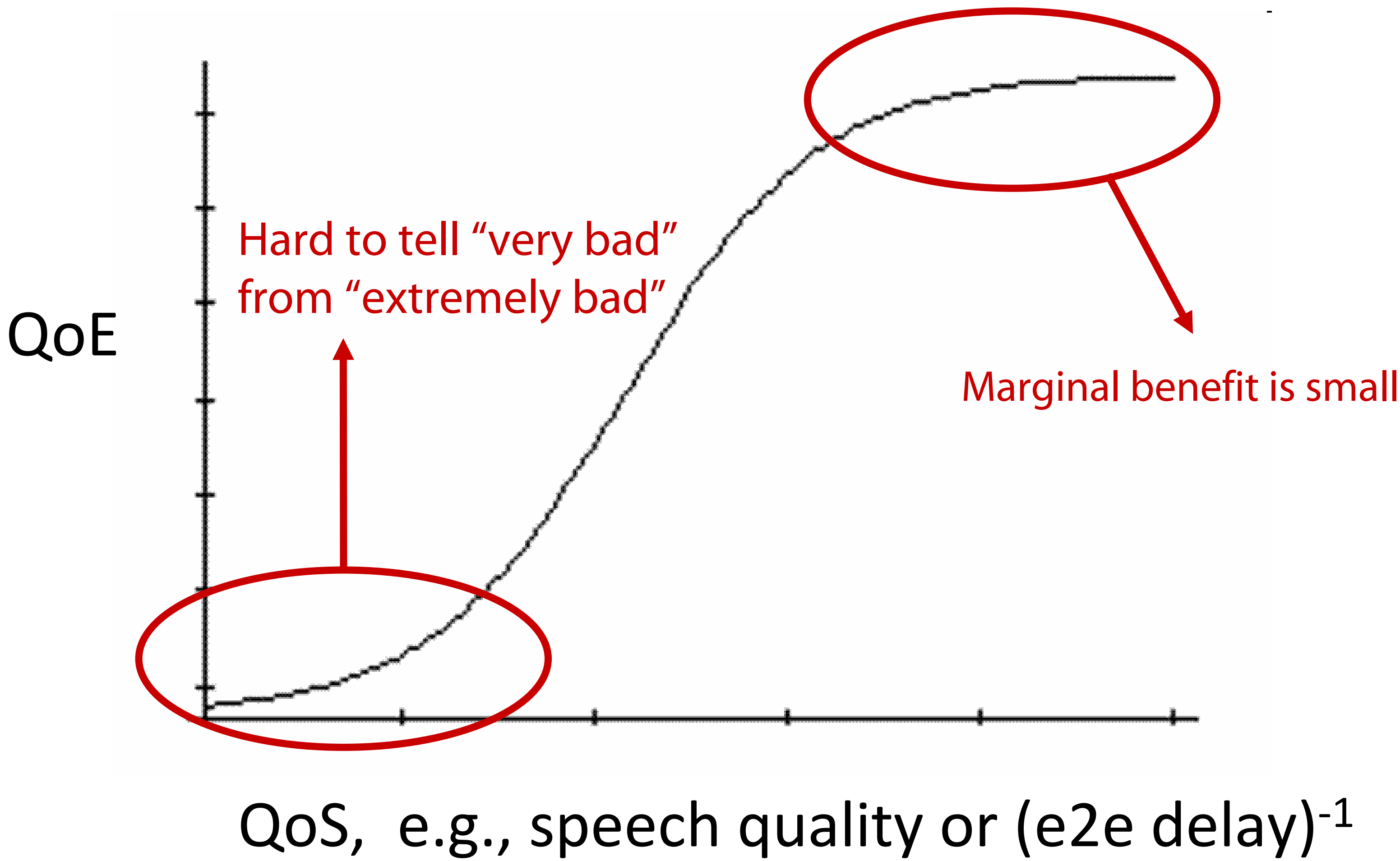
Derived Optimal Buffer Size (1)



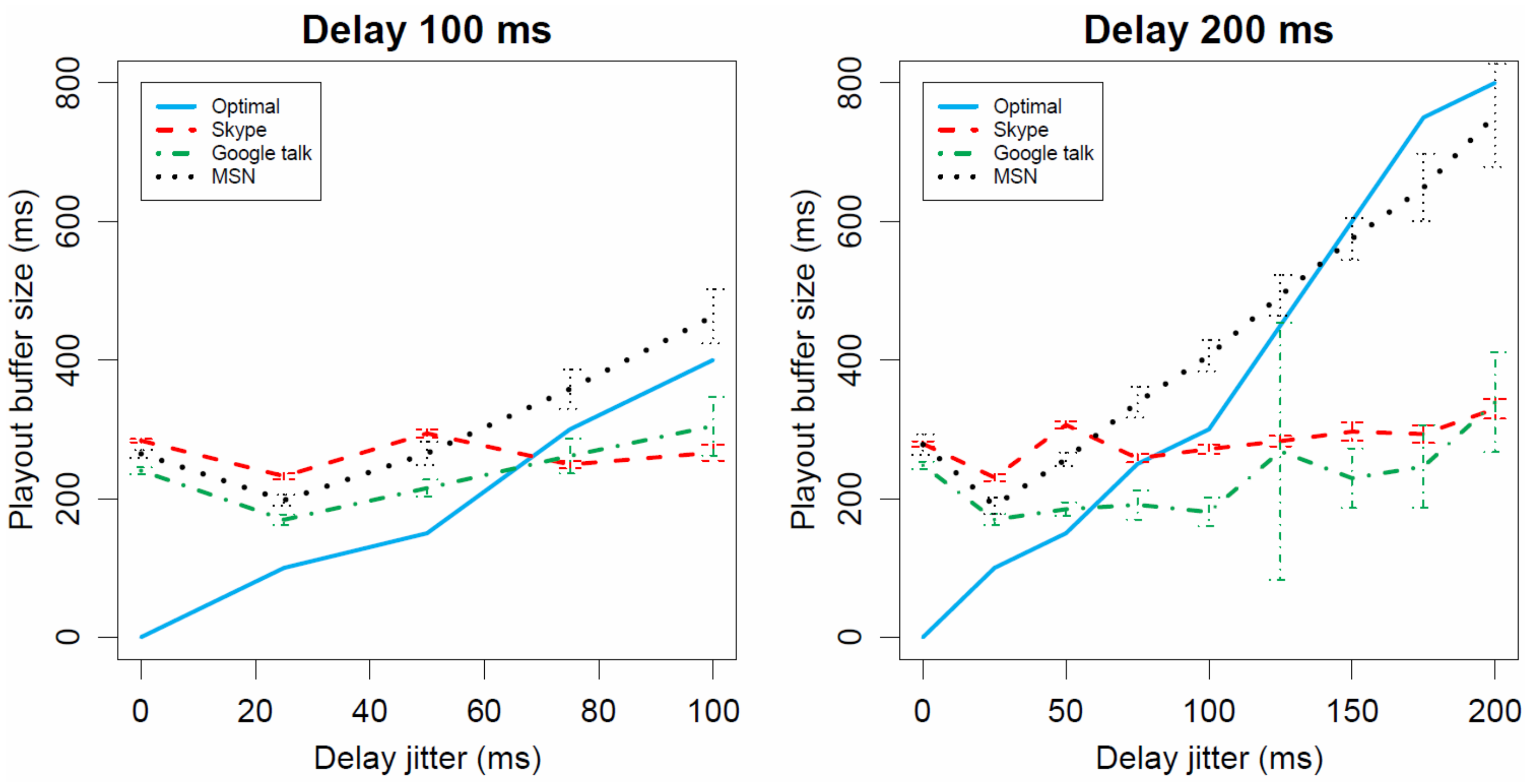
Derived Optimal Buffer Size (2)



A typical relationship between QoS and QoE



Comparing Real-Life Applications with Optimal Settings



- MSN Messenger's buffer dimensioning algorithm is better than those of Skype and Google Talk

Modeling Optimal Buffer Size

- Using a linear regression to model the optimal buffer size given a network setting

$$
\begin{aligned}
&(const.) + coef_{delay} \cdot delay + \\
&coef_{delay \cdot jitter} \cdot delay \cdot jitter + \\
&coef_{delay \cdot jitter \cdot plr} \cdot delay \cdot jitter \cdot plr
\end{aligned}
$$

Variable	Coef	Std. Err.	t	$Pr > \|t\|$
$(constant)$	157	20	7.54	$< 2 \times 10^{-9}$
$delay$	-1.05	0.21	-4.78	$< 2 \times 10^{-5}$
$delay \cdot jitter$	0.02	< 0.01	17.25	$< 2 \times 10^{-16}$
$delay \cdot jitter \cdot plr$	-0.57	0.04	-11.65	$< 5 \times 10^{-15}$

Talk Progress

- Overview
- Measuring Applications' Buffer Size
  - Experiment Methodology
  - Measurement Results
- Deriving Optimal Buffer Size
  - Methodology
  - Derived Buffer Sizes
  - Evaluation of Applications' Dimensioning Algorithms
- Conclusion & On-going Work

Conclusion

- Our results show that MSN Messenger performs the best in terms of buffer dimensioning
  - Proprietary codec are used in Skype
  - Other factors may dominate the final quality perceived by users
- Results from the research community seem not be applied in real-life VoIP applications
  - methods not generalizable enough?
  - methods not inutitive enough?
  - methods not practical enough? (e.g., hard to implement)
  - other explanations?
- A regression modeling appraoch to compute the optimal buffer size in run time

On-going Work

- More factors in measuring applications' buffer dimensioning behavior
  - frame size, redundancy control, loss burstiness, speech codec, …
- More factors in deriving optimal buffer size
  - transport protocol (TCP in addition to UDP), speech codec, …
- Real-life network experiments to evaluate the regression-based buffer dimensioning algorithm

Thank You!

Kuan-Ta Chen

http://www.iis.sinica.edu.tw/~ktchen