Tuning Skype Redundancy Control Algorithm for User Satisfaction

Te-Yuan Huang, Kuan-Ta Chen, Polly Huang

Proceedings of the IEEE Infocom Conference
Rio de Janiero, Brazil, April 2009

Motivation

- Voice traffic is sensitive to network impairment
- Why is VoIP sending rate important?
  - Most important factors for user satisfaction [5]:
    - Sending rate and its variation
    - High and stable voice quality
    - (MLC: note, their measure is "call survival rate". Based on empirical measurement of net traffic.)
- Why is adapting sending rate difficult?
  - Aggressively? → cause congestion, worse for all
  - Conservatively? → quality lower than users want

Goals

- Skype – one of most popular VoIP apps
  - 2008, had 338 millions users [footnote 1]

Q1: How does Skype adapt voice traffic?

Q2: Is Skype mechanism good enough?

Q3: How can Skype's policy be improved?
(Look to apply to VoIP in general)

Related Work

- Skype's voice traffic is governed by [4]:
  - Bit rate
  - Framing time (aka sample rate)
  - Redundant data
- PC-PSTN calls (Public Switched Telephone Network)
  - G.729 – constant bit rate, framing time
- PC-PC calls
  - iSAC – likely implemented by 3rd party, so bitrate and framing time fixed

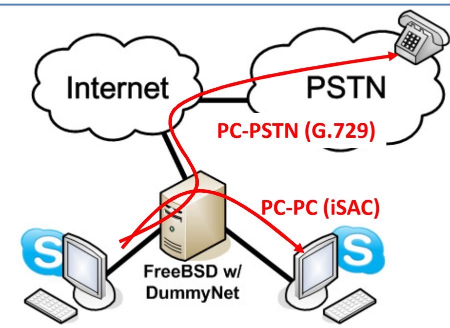
Only redundant data is controlled by Skype

Outline

- Motivation (done)
- Related Work (done)
- How does Skype adapt redundancy? (next)
- Is Skype's mechanism good enough?
- How can Skype do better?
- Conclusion

Experiment Setup

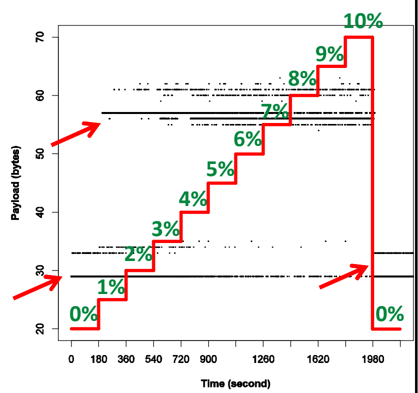
Voice Input : American English Speech Samples from Open Speech Repository [20]



Only UDP sessions via tcpdump
60 seconds after starting
Loss increased by 1% every 180 seconds

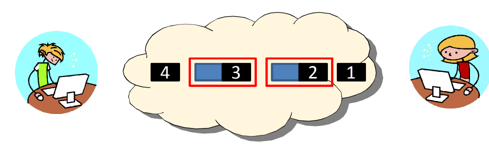
Observation

- G.729 (PC-PSTN)
  - Constant bit rate
  - Constant framing time



Redundancy Ratio

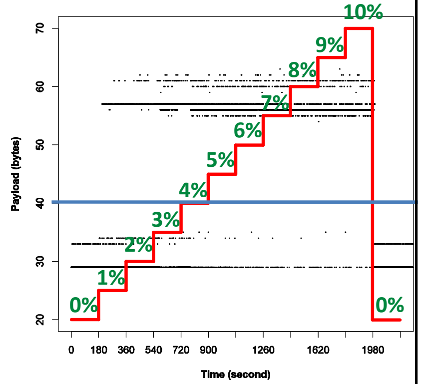
- Definition
  - Fraction of packets that carry redundant voice data



Redundancy Ratio = 2/4 = 0.5

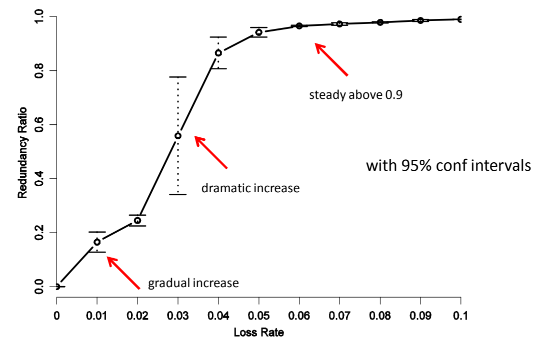
Estimate Redundancy Ratio of G.729

- G.729 (PC-PSTN)
  - Constant bit rate
  - Constant framing time



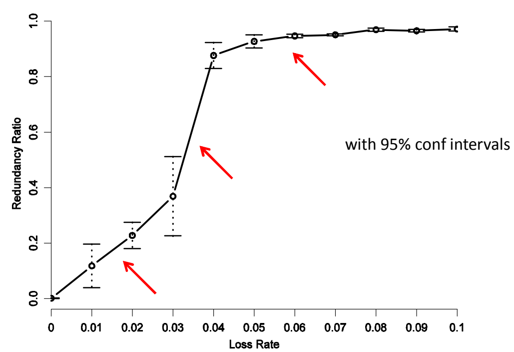
Identify Redundancy Ratio (1 of 2)

- Redundancy Ratio of G.729 (PC-PSTN Calls)



Identify Redundancy Ratio (2 of 2)

- Redundancy Ratio of iSAC (PC-PC Calls)



Outline

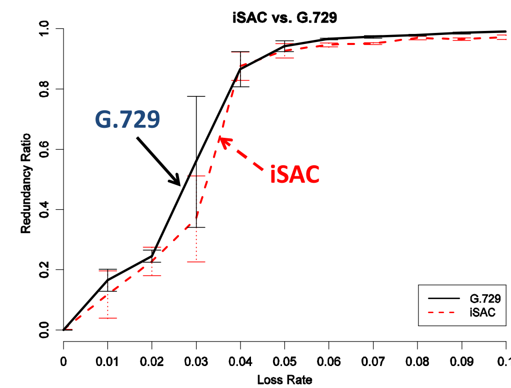
- Motivation (done)
- Related Work (done)
- How does Skype adapt redundancy? (done)
- Is Skype's mechanism good enough? (next)
- How can Skype do better?
- Conclusion

Skype's Redundancy Control Algorithm

- Broadly, three factors that may affect VoIP quality
  - Available bitrate
  - Codec
  - Burstiness of network loss
- Adapt to network loss rate?

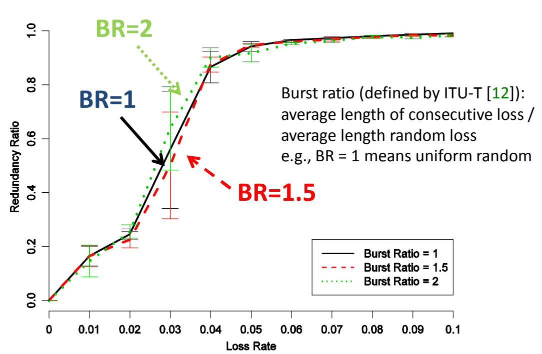
  → Based on [4], switches rates before packet loss reaches 1% so changing bitrate doesn't affect redundancy algorithm
- Adapt to other factors?
  - Codec
  - Network loss burstiness

Effect of Codec



Effect of Network Loss Burstiness

- G.729 (PC-PSTN, but iSAC similar)



Outline

• Motivation	(done)
• Related Work	(done)
• How does Skype adapt redundancy?	(done)
• Is Skype's mechanism good enough?	(done)
• How can Skype do better?	(next)
• Conclusion	

Optimal Redundancy Control Policy

- Redundancy purpose → to maintain voice quality under loss rates
  - Adapt to different loss rates
- What's the *optimal policy*?
  - Minimum amount of redundancy data needed to sustain same audio quality under different network conditions
- Develop an emulator to derive
  - Manipulate voice frames generated from encoder to emulate different network conditions
  - Grade voice quality of degraded audio clips after transmitting under given network conditions

(details next)

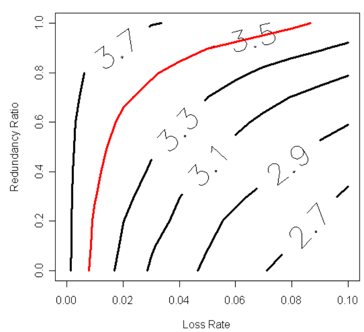
Emulation Flow

Voice Input : American English Speech Samples from Open Speech Repository

1. Encode audio clip into voice frames: G.711 and G.729
   - Can't use iSAC since closed
2. Simulate network loss with Gilbert model
3. Determine if piggybacking based on ratio (simple duplicate)
4. Decode into voice frames
5. Use PESQ [14] to compare original to degraded → MOS

Repeat for range of redundancy ratios
Consider minimum redundancy ratio to sustain given MOS
→ *optimal redundancy ratio*

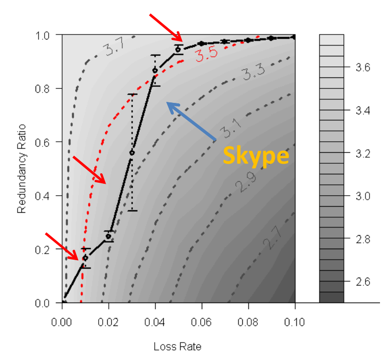
Optimal Redundancy Ratio – G.729



Combinations of redundancy for loss that gives MOS
(Fig 7a: G.711 is somewhat different than G.729 – needs more redundancy)

Skype versus Optimal – G.729

Assume desired MOS 3.5



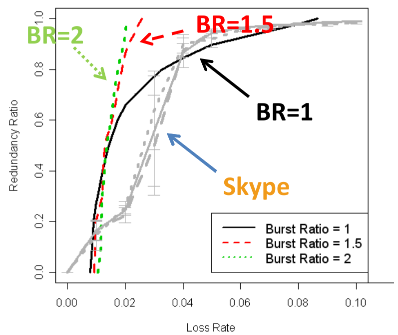
Higher for very low loss and very high loss

Lower for mid-range loss

Optimal can better balance

Burst Ratio = 1

Optimal Redundancy for Burst Ratio



Under bursty loss, need more aggressive redundancy ratio

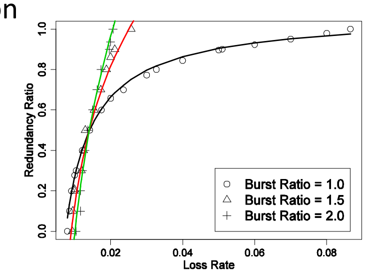
Skype does not adapt to loss burstiness

G.729, MOS=3.5

Modeling Optimal Redundancy Ratio

- Based on targeted voice quality
  - e.g., MOS 3.5
- Take codec and burstiness into consideration
- Polynomial regression

Example: optimal policy for G.729, targeted MOS=3.5



R^2 as high as 0.98

Conclusion

- Explored how Skype adapts redundancy for voice traffic → *redundancy ratio*
- Propose general model for optimal redundancy ratio policy
  - Consistent user satisfaction
  - Extensible to general VoIP software
- Skype's policy does not factor in codec and loss patterns
  - But should!

  → Skype not optimal for some ranges of loss, even more so for bursty loss

Future Work?

Future Work

- Kinds of redundancy
  - Not just piggybacking
- Video
- Other VoIP