

# Modeling Video Traffic Source for RMCAT Evaluations

draft-zhu-rmcat-video-traffic-source

Xiaoqing Zhu, Sergio Mena,  
and Zaheduzzaman Sarker

# Outline

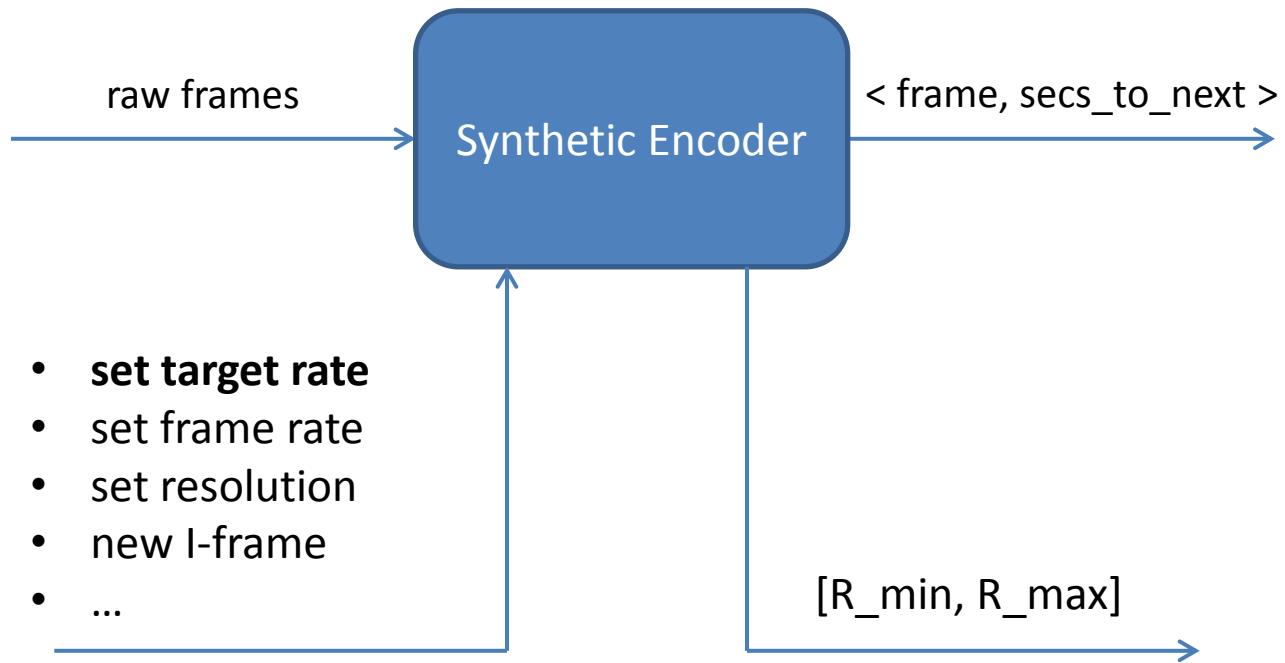
- Why?
- **What's new?**
- What's next?

**WHY?**

# Synthetic Traffic Source

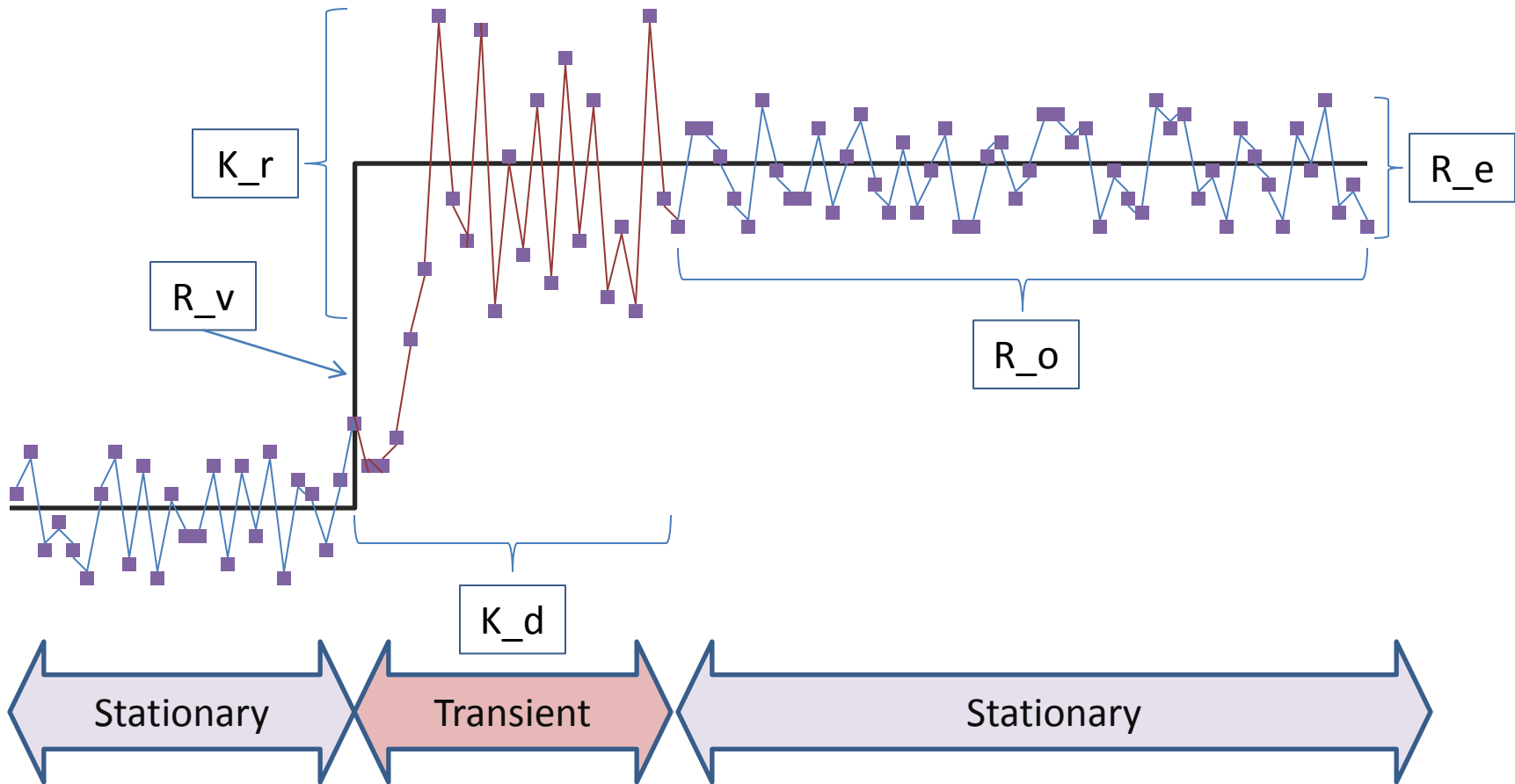
- Why bother?
  - Working with real codecs
    - cumbersome
    - repeatable tests?
  - Working with simplistic models:
    - E.g., “perfect codec”
    - **benign** → test works but, will it fly?
  - Strike **balance** between two extremes

# System Model



See Fig. 1 in draft-zhu-rmcat-video-traffic-source

# Statistics-Based Model



- Presented at IETF-88 & IETF-91

<http://www.ietf.org/proceedings/88/slides/slides-88-rmcat-2.pdf>

<http://www.ietf.org/proceedings/91/slides/slides-91-rmcat-0.pdf>

# Trace-Based Model

- Presented at IETF-88 & IETF-91:
  - <http://www.ietf.org/proceedings/88/slides/slides-88-rmcat-9.pdf>
  - <http://www.ietf.org/proceedings/91/slides/slides-91-rmcat-0.pdf>
- Offline:
  - video sequence encoded at constant target rate
  - repeat various rates:  $[R_{\min}, R_{\max}]$  with step  $L$
- Online:
  - Choose  $r, r+L$  around target rate  $r_v$  and interpolate
  - If  $r_v$  is not in  $[R_{\min}, R_{\max}]$ , then scale
  - Can be extended for variable frame rate, or resolution

**WHAT'S NEW?**



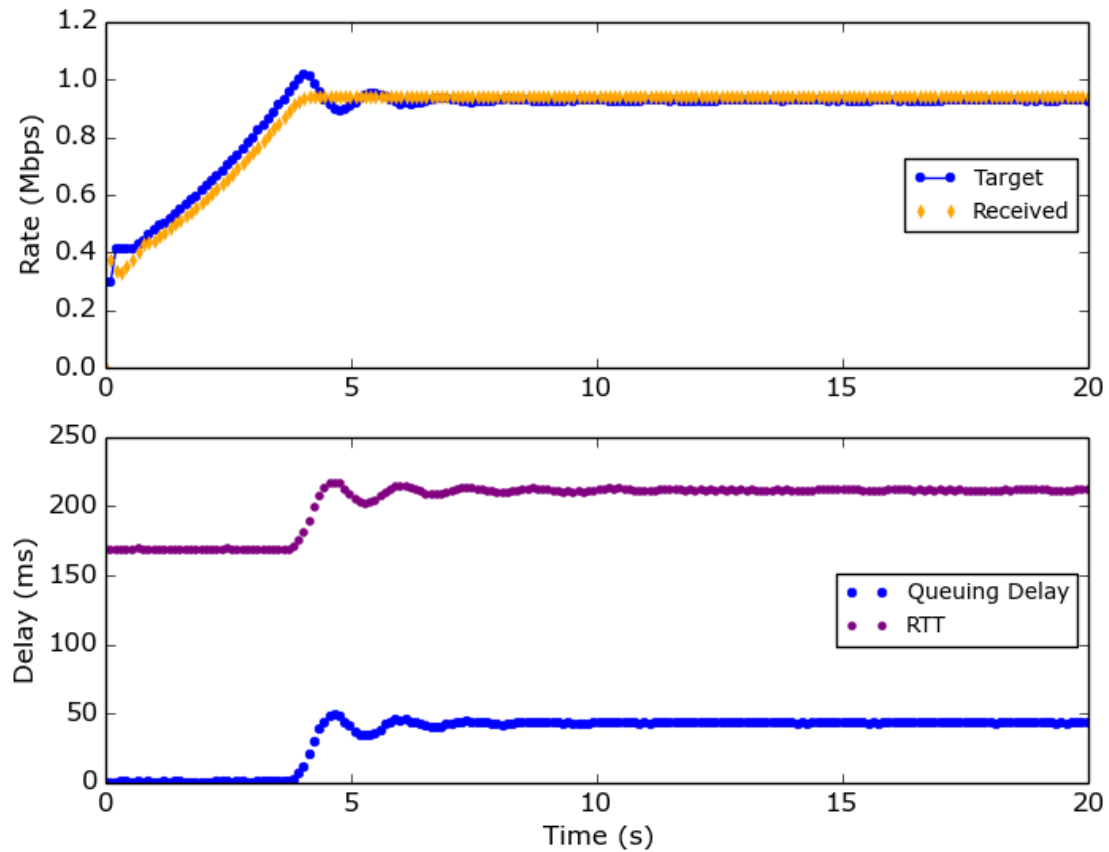
# Updates to the Draft

- Section 4: simplified system model figure
- Section 5: added a table listing all parameters
- Section 7: added discussions on:
  - Pros & cons of the two models
  - Potentially, how to combine them
- Section 8: added implementation status
  - Our implementation: *Syncodecs*
  - Open sourcing process

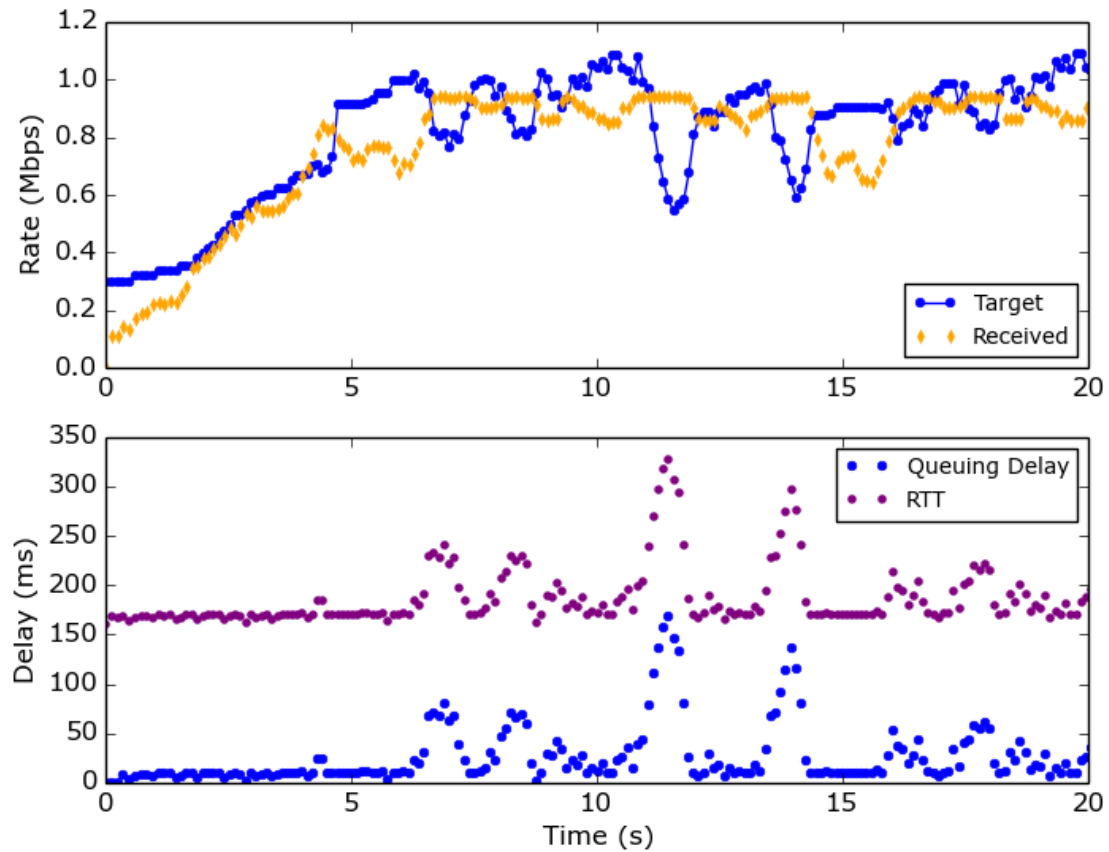
# Syncodecs. Open Source

- Codecs implemented as C++ iterators
  - system-agnostic
  - well-known, simple interface
    - “++codec”: advance frame
    - “\*codec”: provides a tuple  
< frame, seconds\_to\_next >
- Hierarchy of codecs
  - Simple: “perfect codec”
  - More sophisticated: “trace-based codec”
- Open sourcing
  - Ongoing (at last stages)
  - Following Cisco’s legal procedure

# Perfect Codec. Example



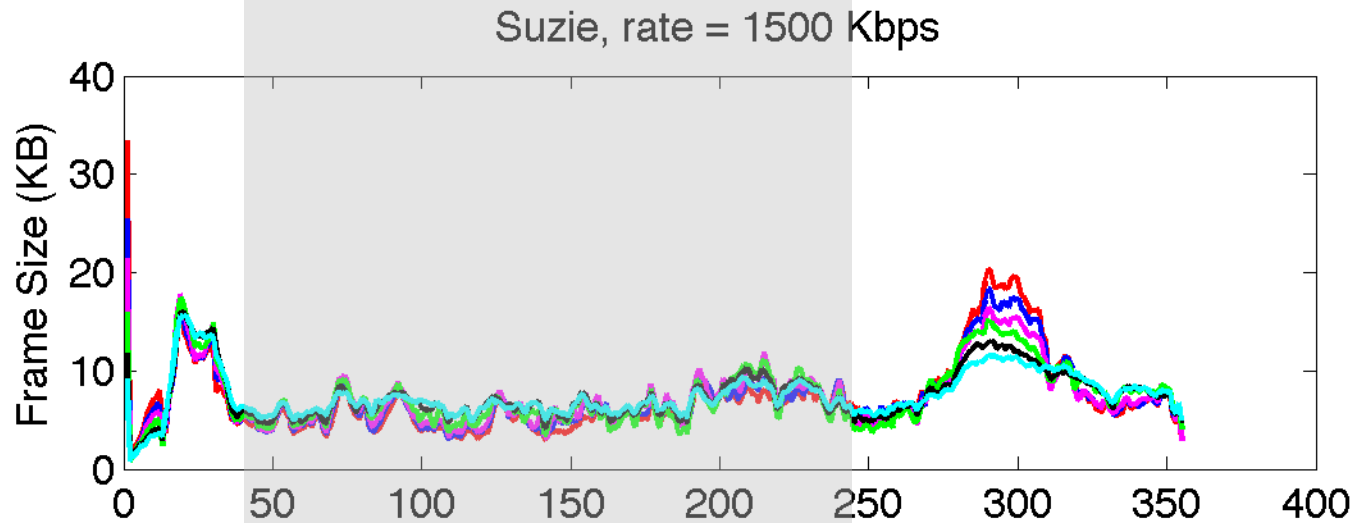
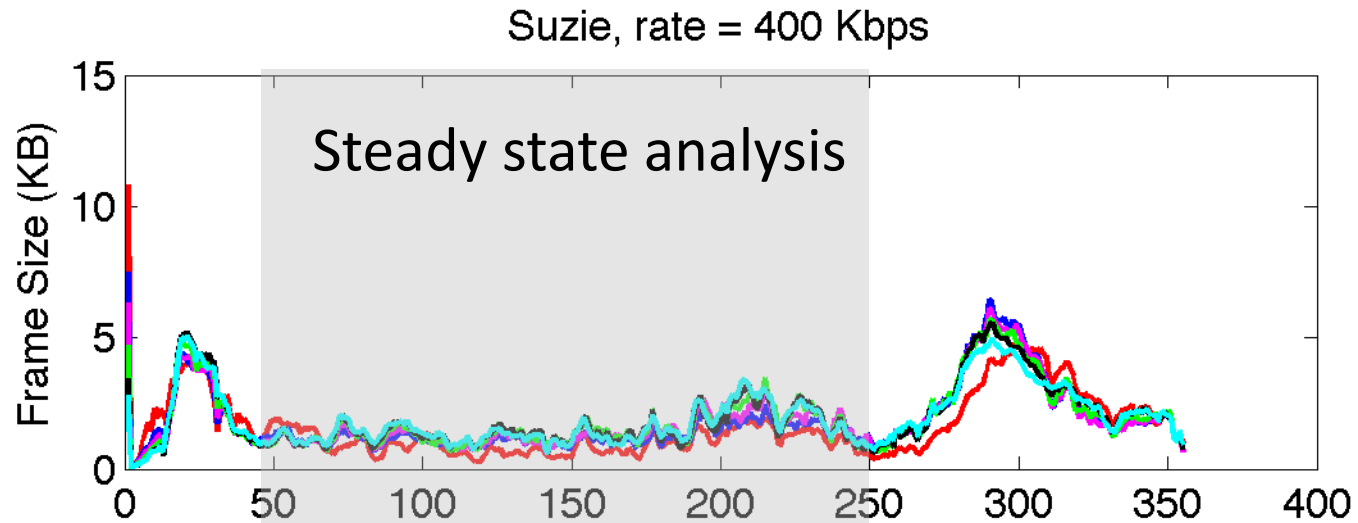
# Trace-based Codec. Example



# Video Traffic Data Gathering

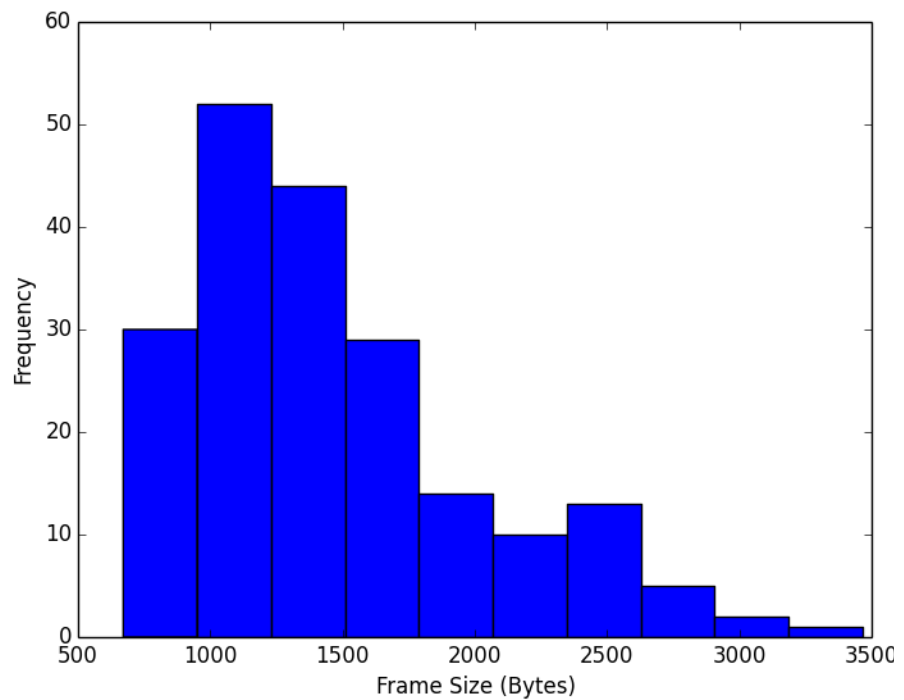
- Source of video:  
<http://www.elementaltechnologies.com/resources/4k-test-sequences>
- Sequences: *Foreman, News, Suzie*
- Encoder: x264 (lookahead = 1 in rate control)
- Encoded resolutions & frame rate
  - 1080p, 720p, 540p, 360p, 240p, 180p
  - 25 fps
- Encoded rates: 100 ~ 1500Kbps
- Frame structure: I frame followed by all P frames

# Example Traffic Trace: *Suzie*



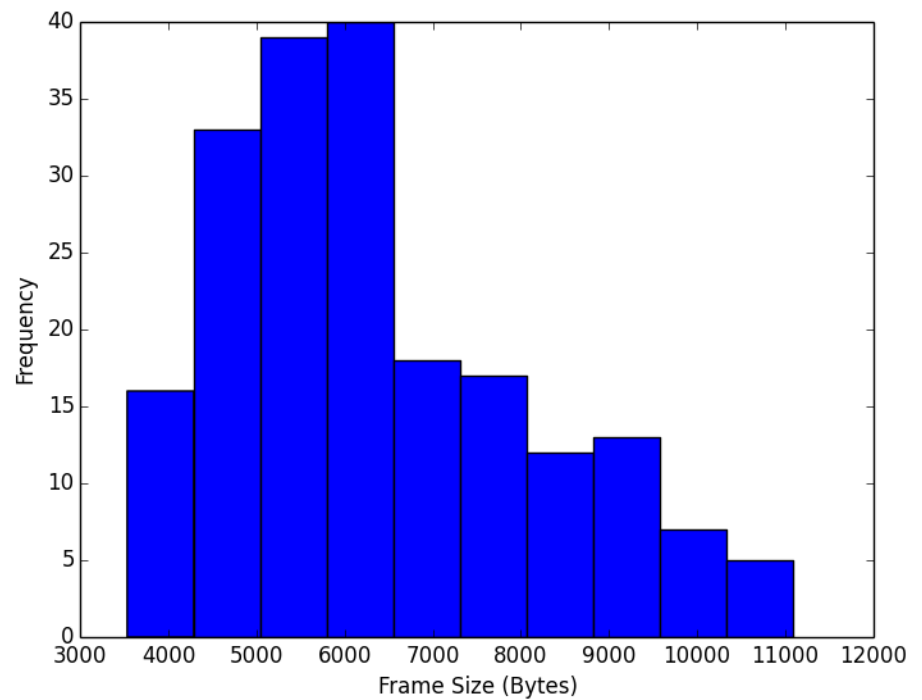
# Distribution of Frame Size: *Suzie*

400Kbps @ 360p



AVG: 1181.3B | STD: 281.1B

1500Kbps @ 360p



AVG: 5867.2 B | STD: 1218.8 B

## Observations:

- Shape of distribution closer to Gamma than Normal/Uniform;
- Deviation from the mean ~ 30-50%

**WHAT'S NEXT?**



# Next Steps

- Adoption as WG item?
- Statistics-based model
  - Data gathering and analysis for transient
  - Open source: add to Syncodecs C++ class family
- Combine both models
  - Stationary state → trace-based behavior
  - Transient state → statistics-based (bursts)
- Interested in extensions to:
  - Changing frame resolutions
  - Changing frame rates
  - Both: tradeoff frame rate/resolution

Thank you

Questions?