

Modeling Video Traffic Source for RMCAT Evaluations

draft-zhu-rmcat-video-traffic-source

Xiaoqing Zhu, Sergio Mena,
and Zaheduzzaman Sarker

Outline

- Recap of draft status
- Open source implementation status
 - Update on code
 - Where to find it
 - How to use it
 - Example test results
- Next steps

draft-zhu-rmcat-video-traffic-source-02

- Purpose: manageable yet realistic evaluation of RMCAT candidates
- Last update: July 2015 (IETF-93@Prague)
- Previous presentations on traffic model and example test results:
 - <http://www.ietf.org/proceedings/88/slides/slides-88-rmcat-9.pdf>
 - <http://www.ietf.org/proceedings/88/slides/slides-88-rmcat-2.pdf>
 - <http://www.ietf.org/proceedings/91/slides/slides-91-rmcat-0.pdf>
 - <http://www.ietf.org/proceedings/93/slides/slides-93-rmcat-4.pdf>

Update on *Syncodecs*

- Code available at: <https://github.com/sergio-mena/syncodecs>
 - Stand-alone C++ classes, platform agnostic
 - Repo includes detailed documentation, examples, and sample video traces
- Supports three types of codecs:
 - Perfect Codec
 - Trace-based Codec
 - Statistical Codec (*NEW* added after Prague)
- Supports packetization and traffic shaping:
 - Packetizers have codecs' look and feel
 - Spread out packets within duration of each frame

How to Use *Syncodecs*

- Codec creation:

```
syncodecs::Codec* c =  
    new syncodecs::PerfectCodec(MAX_PKT_SIZE)
```

- Accessing current frame data:

- Frame size: `c->first.size()`

- Seconds to next frame: `c->second`

- Advancing to next frame

- Just pre-increment it (iterator-like): `++c`

- Setting current target rate:

```
c->setTargetRate(newRateInBps)
```

Relevant Codec Types

- Perfect codec (most benign):

```
syncodecs::Codec* perfectCodec =  
    new syncodecs::PerfectCodec(MAX_PKT_SIZE)
```

- Trace-based codec:

```
syncodecs::Codec* traceCodec =  
    new syncodecs::TraceBasedCodecWithScaling(  
        "/path/to/traces/dir",  
        "myAwesomeVideo", FPS)
```

- Statistical codec:

```
syncodecs::Codec* statsCodec = new syncodecs::StatisticsCodec(FPS)
```

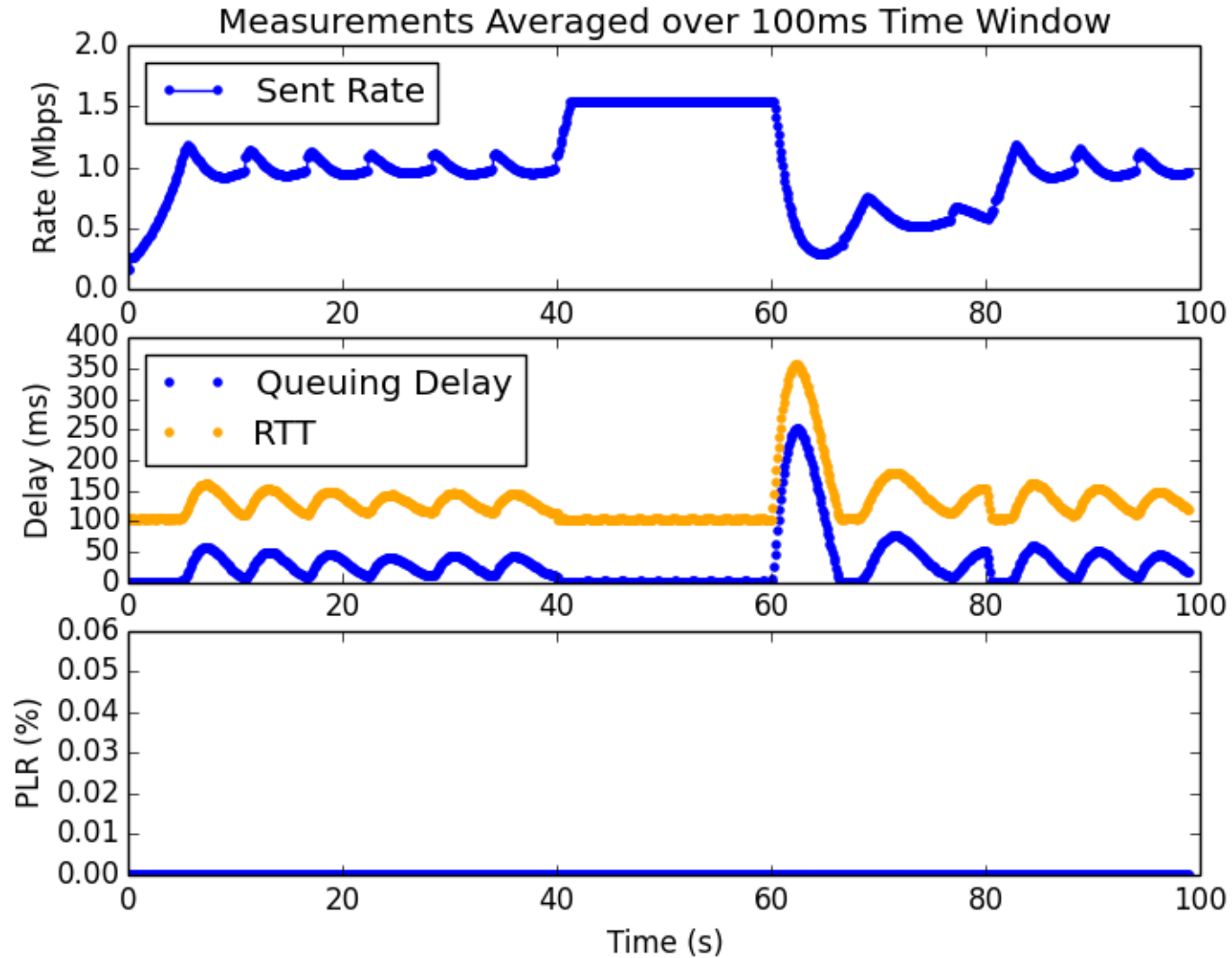
- Shaping and packetization:

```
syncodecs::Codec* wrapper =  
    new syncodecs::ShapedPacketizer(innerCodec,  
                                    MAX_PKT_SIZE)
```

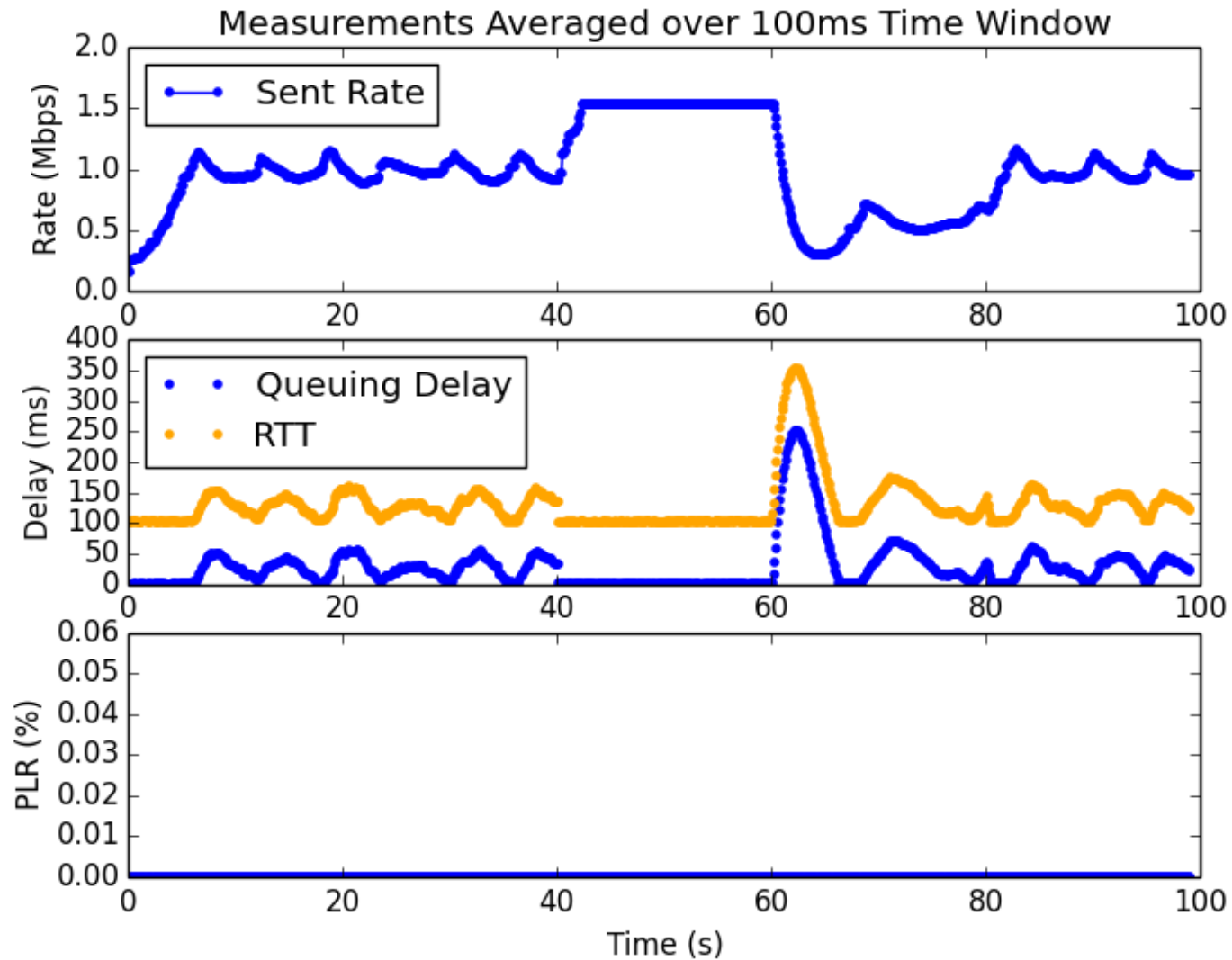
– *innerCodec* can be either *traceCodec* or *statsCodec*

- Once codec is created, usage is common

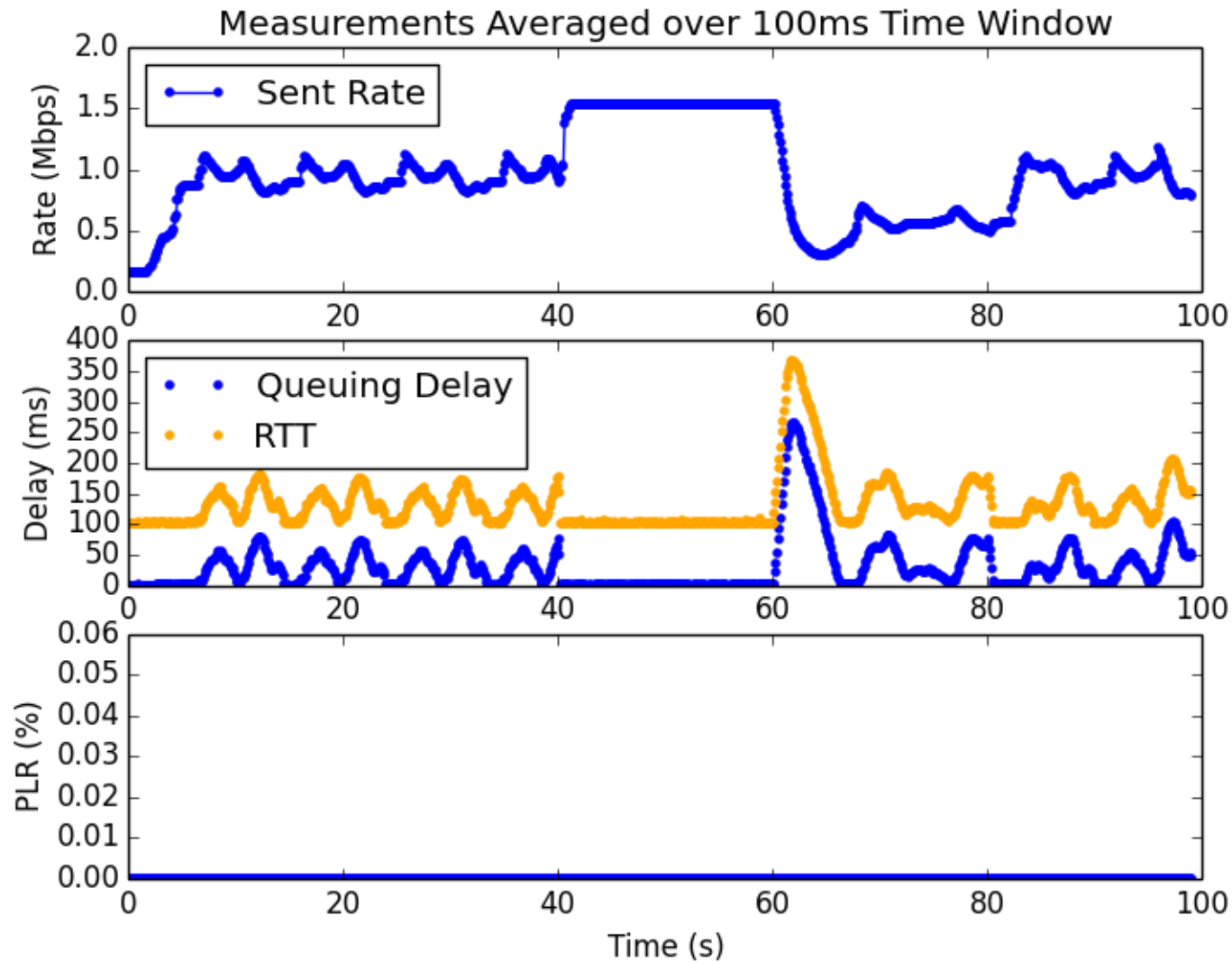
NS-3 Evaluation of NADA: Perfect Codec



NS-3 Evaluation of NADA: Statistical Codec



NS-3 Evaluation of NADA: Trace-Based Codec



Next Steps

- Future work:
 - Statistics on transient behavior of live video codec output
 - Combined model and extensions
- Adoption as WG item?
- In need of:
 - Review inputs
 - Feedback from those who use the open source code in evaluations