vDelay: A tool to measure capture-to-display latency and frame rate

Omer Boyaci,
Andrea Forte, Salman Abdul Baset and Henning Schulzrinne

Introduction

- Measures capture-to-display latency and frame rate of a video chat session
- Do not require access to source code or network stream
- No modifications to the executables
- Sender and receiver side vDelay applications are written in Java so it is platform independent.
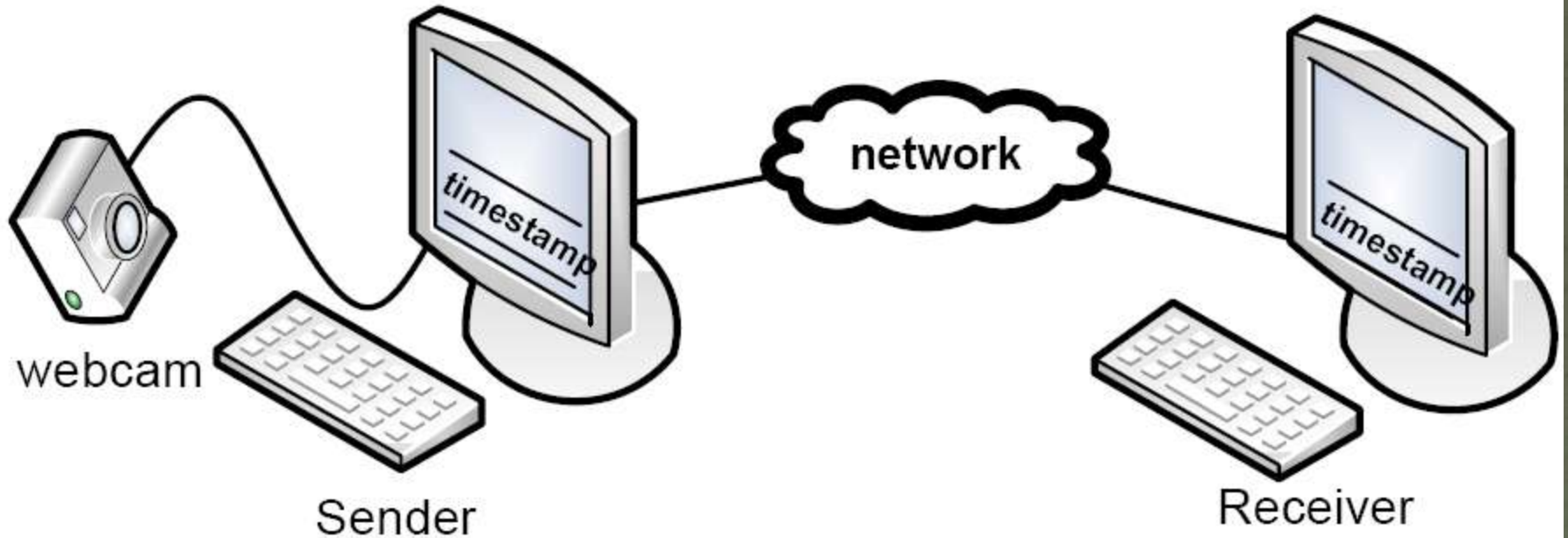
vDelay

Figure 1: vDelay setup.

vDelay

Application Layer (Java)
Renders the timestamp

Operating System
(Frame Buffer)

Display
(LCD)

Webcam

Video Chat App
(Encoding)

Receiver side

Application Layer
(Java)
Captures the timestamp

Display
(LCD)

Operating System
(Frame Buffer)

Video Chat App
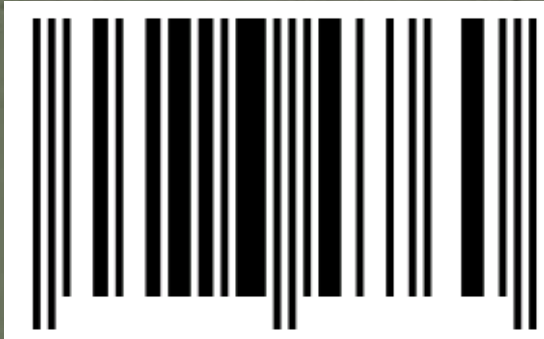(Decoding)

Frame

4

vDelay

Screenshot of the receiver side vDelay application.
FPS, CDL, and FRR statistics are shown at the top of the image.
The barcode received from the caller agent is also visible.

vDelay (Tools)

Application Layer (Java)
Renders the timestamp

- EAN-8 barcodes
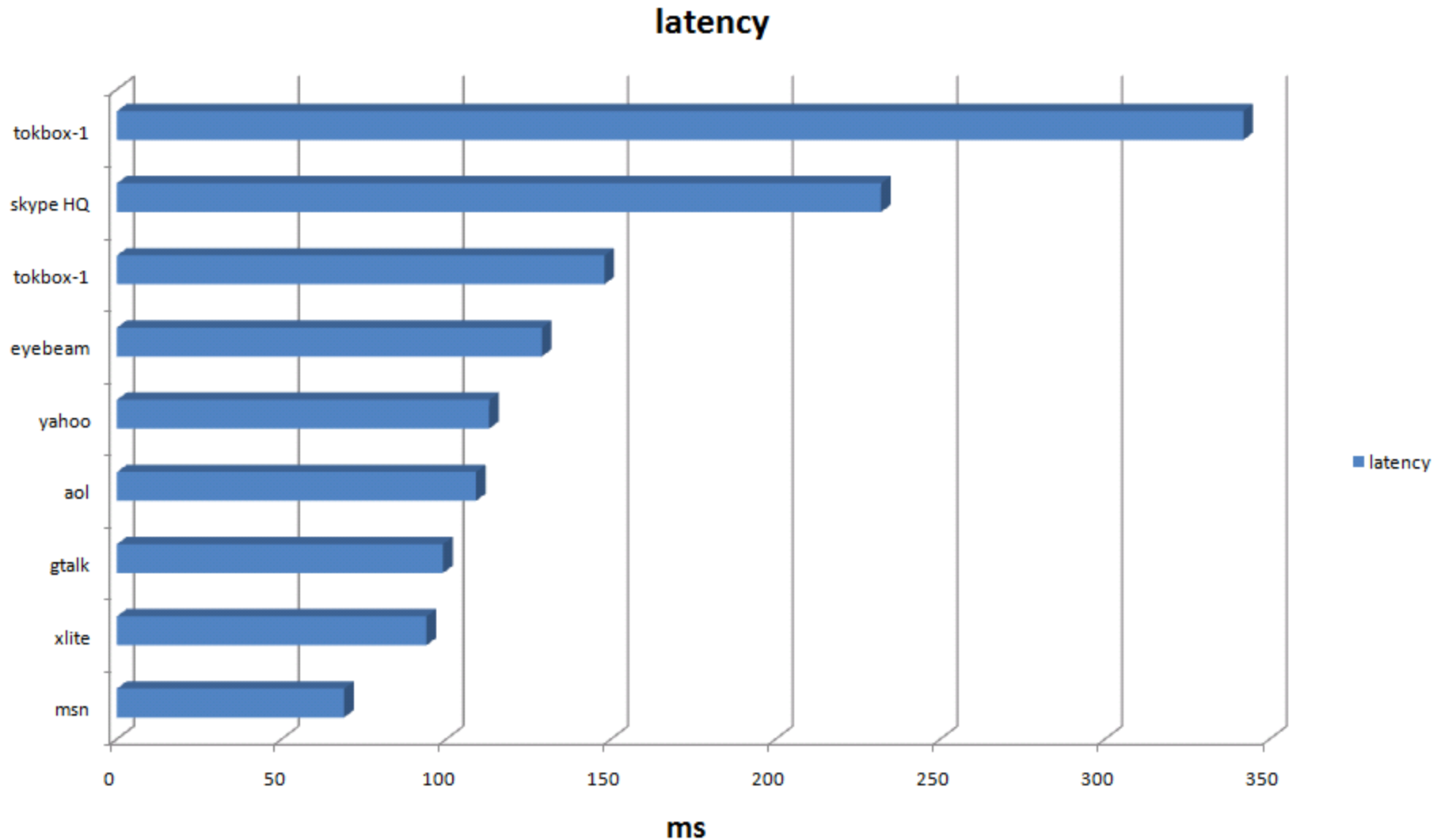Barcode4J library (10000 barcodes)

Application Layer
(Java)
Captures the timestamp

- Screen Capture (Java)(~~Aero~~)
- Checks whether new frame or not
- If new ->Zxing library (Google Code)
- Reports
    - Frame per second
    - Capture-to-display latency
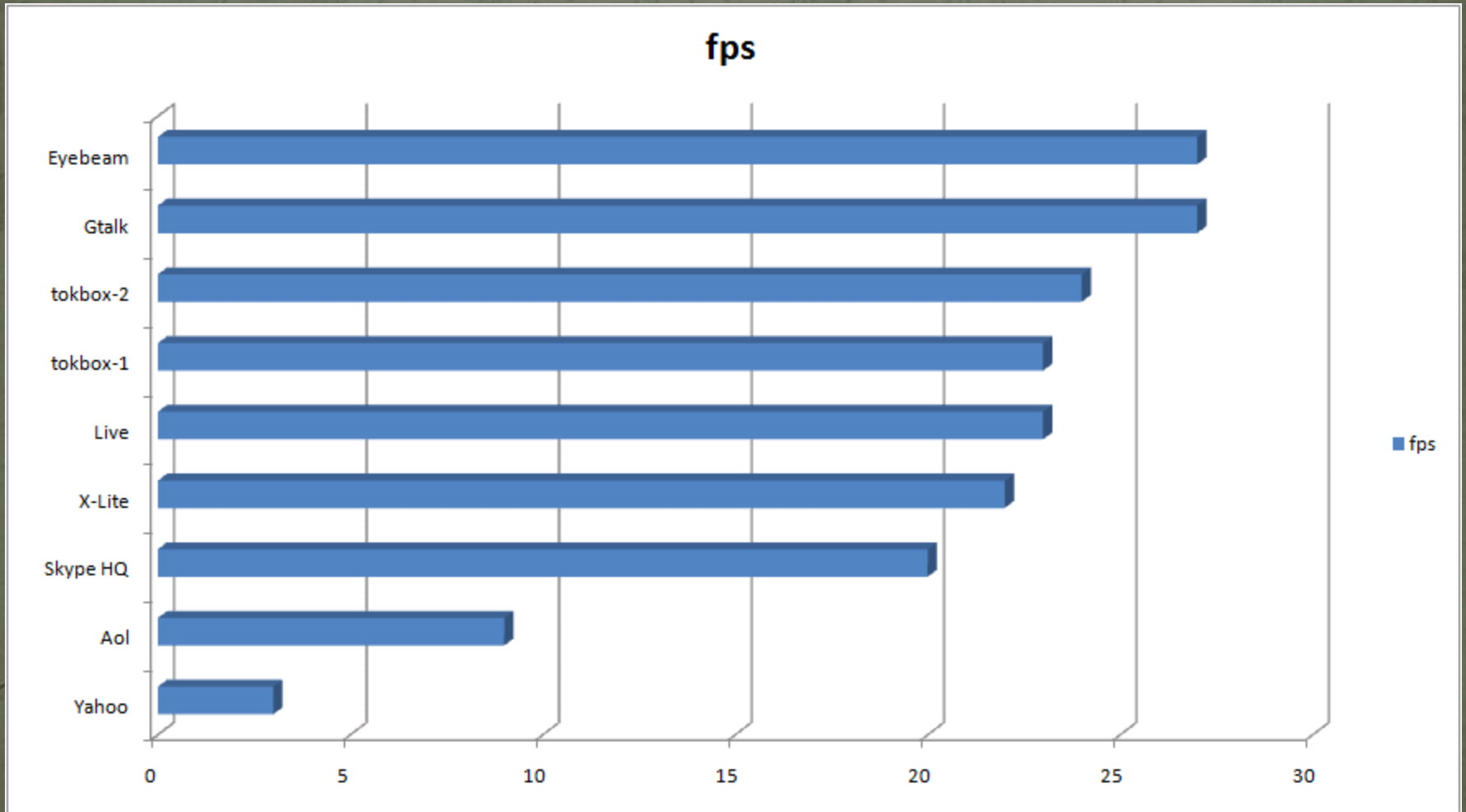
6

vDelay: Important factors

- Shutter speed of the webcam
  - Lower is better (1ms worked for us)
- Gain (Aperture)
  - Should be set to maximum to decrease shutter speed
- LCD vertical refresh rate
  - Higher is better (60Hz in our case)
- LCD latency
  - Lower is better (5ms in our case)
- Video quality
  - Reading barcodes are easier for HQ video
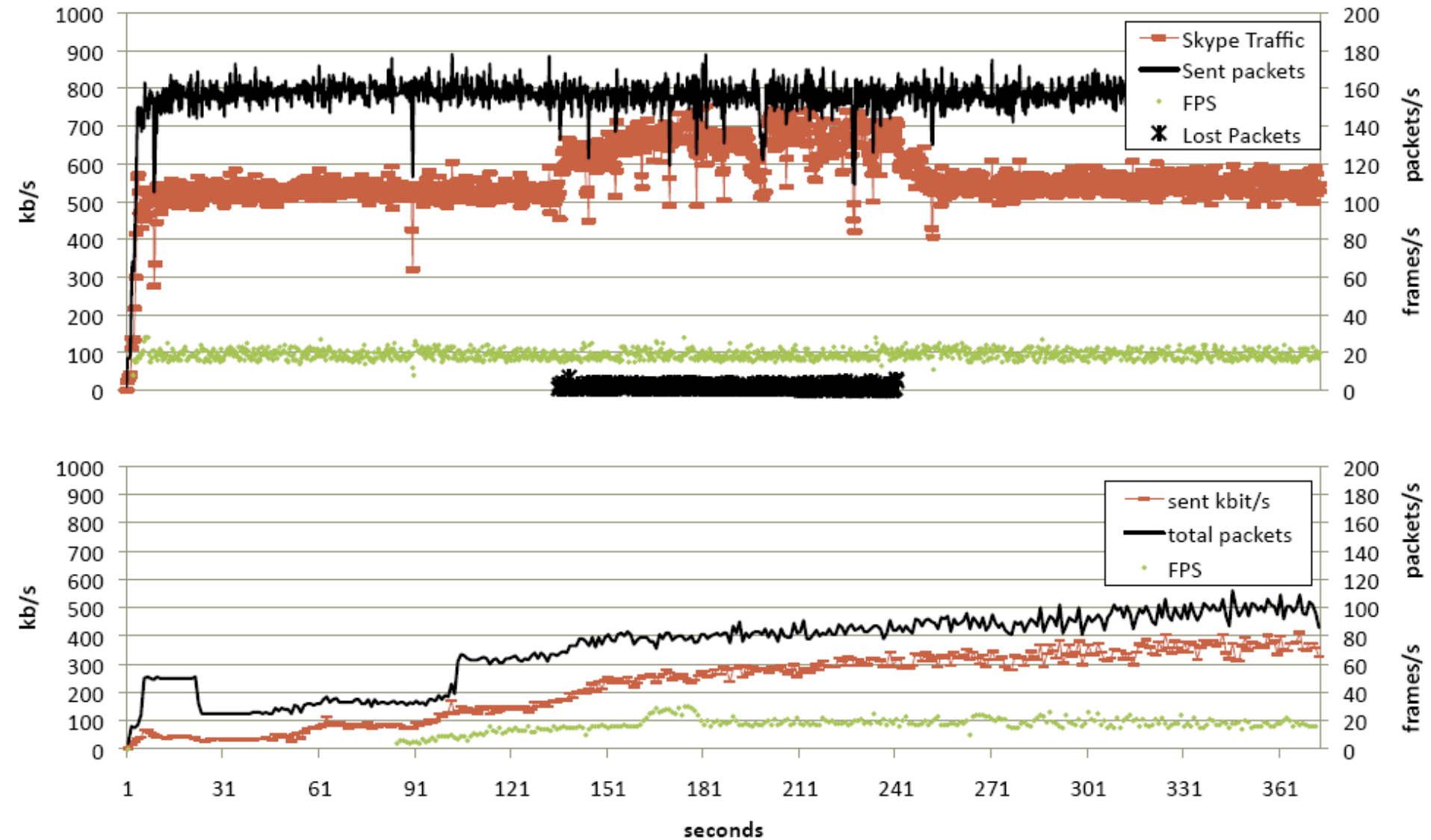
vDelay: Results (Latency)

vDelay: Results (Frame per sec)

Skype traffic under congestion

Conclusion

- Video chat applications have very different congestion control mechanism.

- Big DSL/cable modem buffers increase network latency in case of congestions.

- Capture-to-display latency of video chat applications varies from 70ms to 300ms.