An Experimental Evaluation of Voice Quality over DCCP

Horia Vlad Balan*
International University Bremen

Lars Eggert*
Nokia Research Center

Saverio Niccolini
NEC Network Laboratories

Marcus Brunner
NEC Network Laboratories

* research performed at NEC Network Laboratories

IEEE INFOCOM 2007
Anchorage, AL, USA
May 6-12, 2007

Motivation

- voice-over-IP (VoIP) telephony becoming popular (SIP, Skype, etc.)
- most VoIP apps use either UDP or TCP for voice transmission
- problem: TCP retransmissions add delay during loss events
    - retransmissions are unnecessary – voice codecs deal with loss
- problem: UDP has no congestion control (and apps don’t either)
    - unfair behavior towards other traffic
    - no reduction in bandwidth use under persistent congestion
- a different transport protocol may be more suitable to support this type of communication

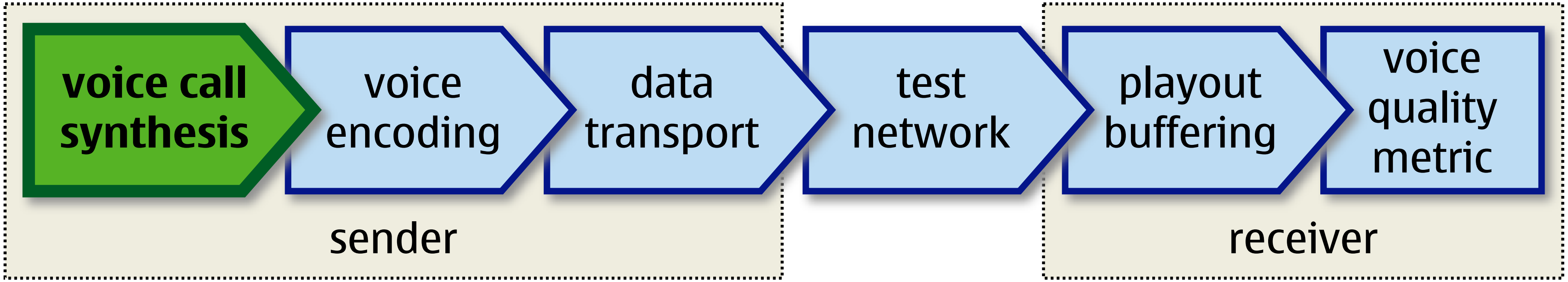
Datagram Congestion Control Protocol (DCCP)

- recent IETF transport protocol framework
  - congestion-controlled but unreliable data transmission
  - “congestion-controlled UDP”

- DCCP offers different congestion control schemes (CCIDs)
  - CCID2 – TCP-like windowing scheme 
  - CCID3 – TCP-Friendly Rate Control (TFRC) 
    - TFRC SP – TFRC for small packets (work in progress)
    - TFRC FR – TFRC with “faster restart” (work in progress)

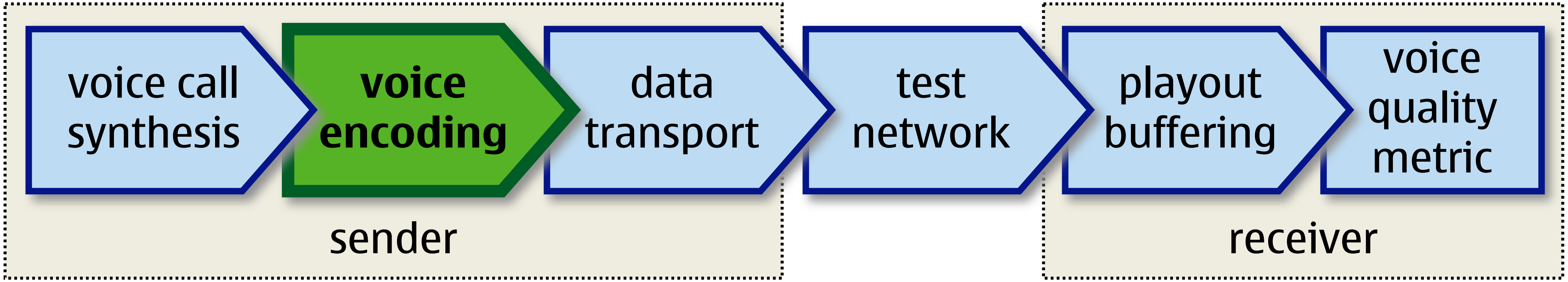
- TFRC SP and FR are targeted at voice transmission
  - how well do they perform?

Experimental Setup: Voice Call Synthesis



- sender synthesizes random voice calls
- by interleaving talkspurts & pauses using a decaying exponential distribution (Sriram/Whitt, 1986)
    - average length of talkspurt = 1 sec
    - average length of pause = 1.5 sec
- talkspurt audio taken from a speech recording (Bush on creation of DHS)
- each call is 100 talkspurt/pause cycles, i.e., average call length is 250 seconds

Experimental Setup: Voice Encoding



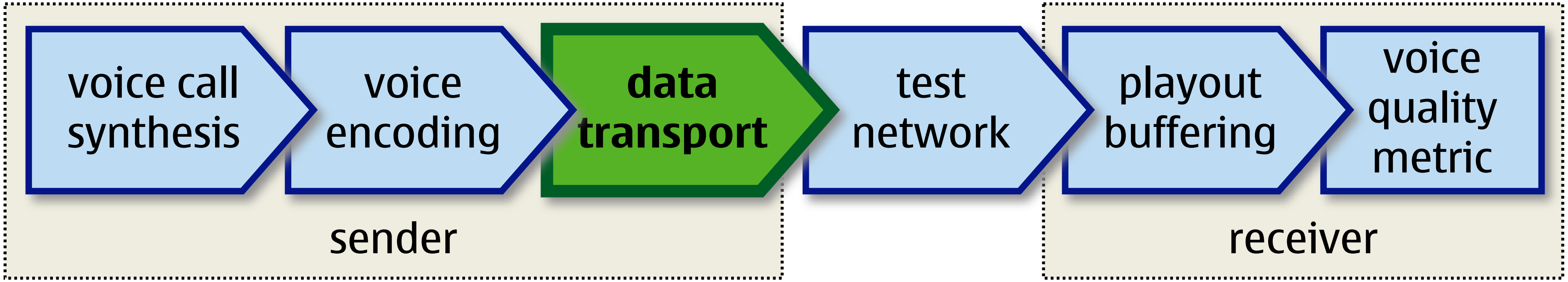
- sender encodes voice into audio frames for transmission across the network

- experiments use two different configurations of the Speex codec
  - emulate G.711
  - emulate G.729
  - both with voice activity detection

Codec	Audio Bandwidth [kbps]	Sample Period [ms]	Frame Size [Bytes]	Frames/ Packet	Data Bandwidth [kbps]
G.711	64	20	160	1	95.2
G.729	8	10	10	2	39.2

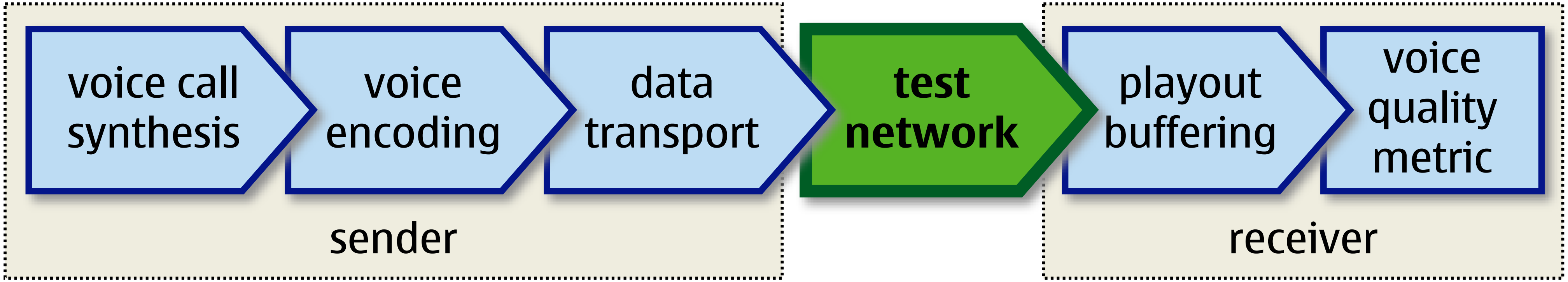
- talk will only present G.729 results (for full results, see paper)

Experimental Setup: Data Transport



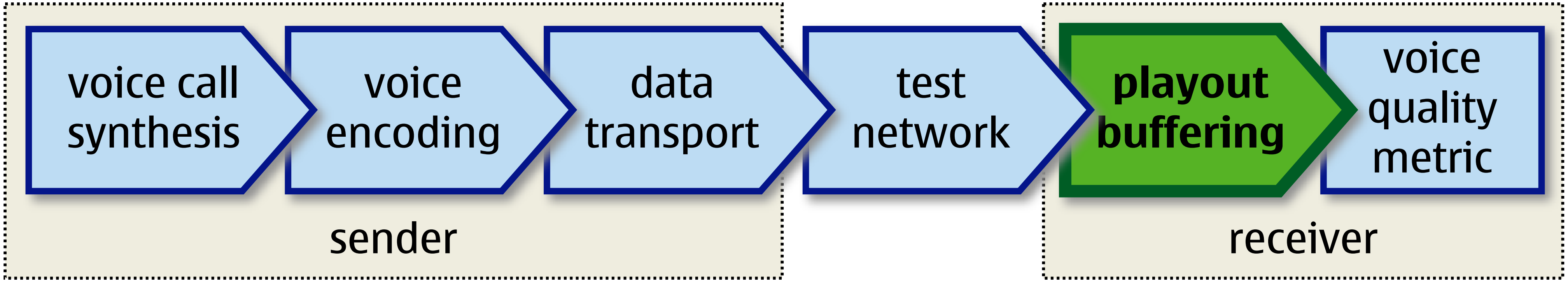
- sender transmits audio frames over several transport protocols
    - UDP
    - TCP (with Nagle disabled)
    - TFRC (DCCP CCID3)
    - TFRC small packet variant (TFRC SP)
    - TFRC SP with “faster restart” optimization (TFRC SP+FR)

Experimental Setup: Network Emulation



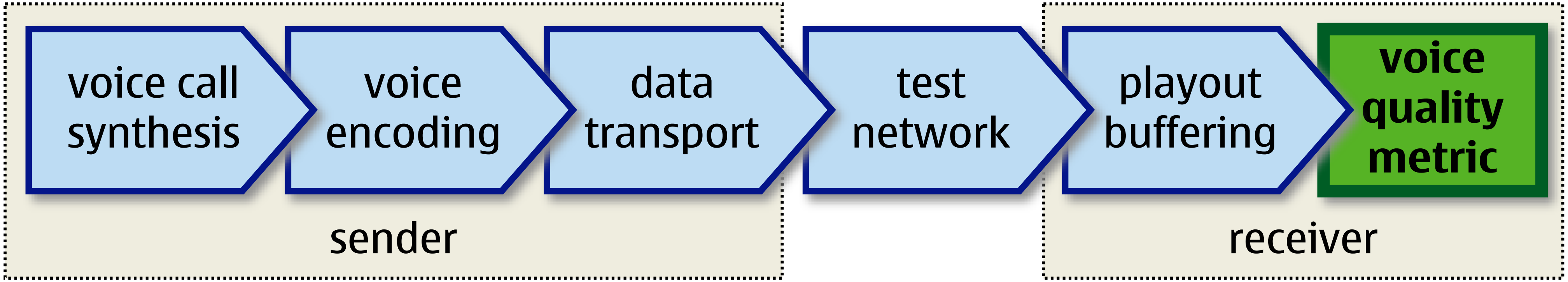
- data transmission occurs over a one-hop network
  - using the KAME BSD DCCP prototype (with some bugs fixed)
- DummyNet router emulates varying path delays and loss rates
  - path delay varies from 0 to 400 ms
  - loss rates vary from 0.01% to 10%
- no bandwidth limitation!

Experimental Setup: Playout Buffering



- goal: investigate impact of transport protocols on audio quality
  - factor out the impact of different playout algorithms
- receiver computes best possible playout sequence (offline)
  - one that leads to the highest possible audio quality for the received voice frames (Moon/Kurose/Towsley, 1998)

Experimental Setup: Voice Quality Metric

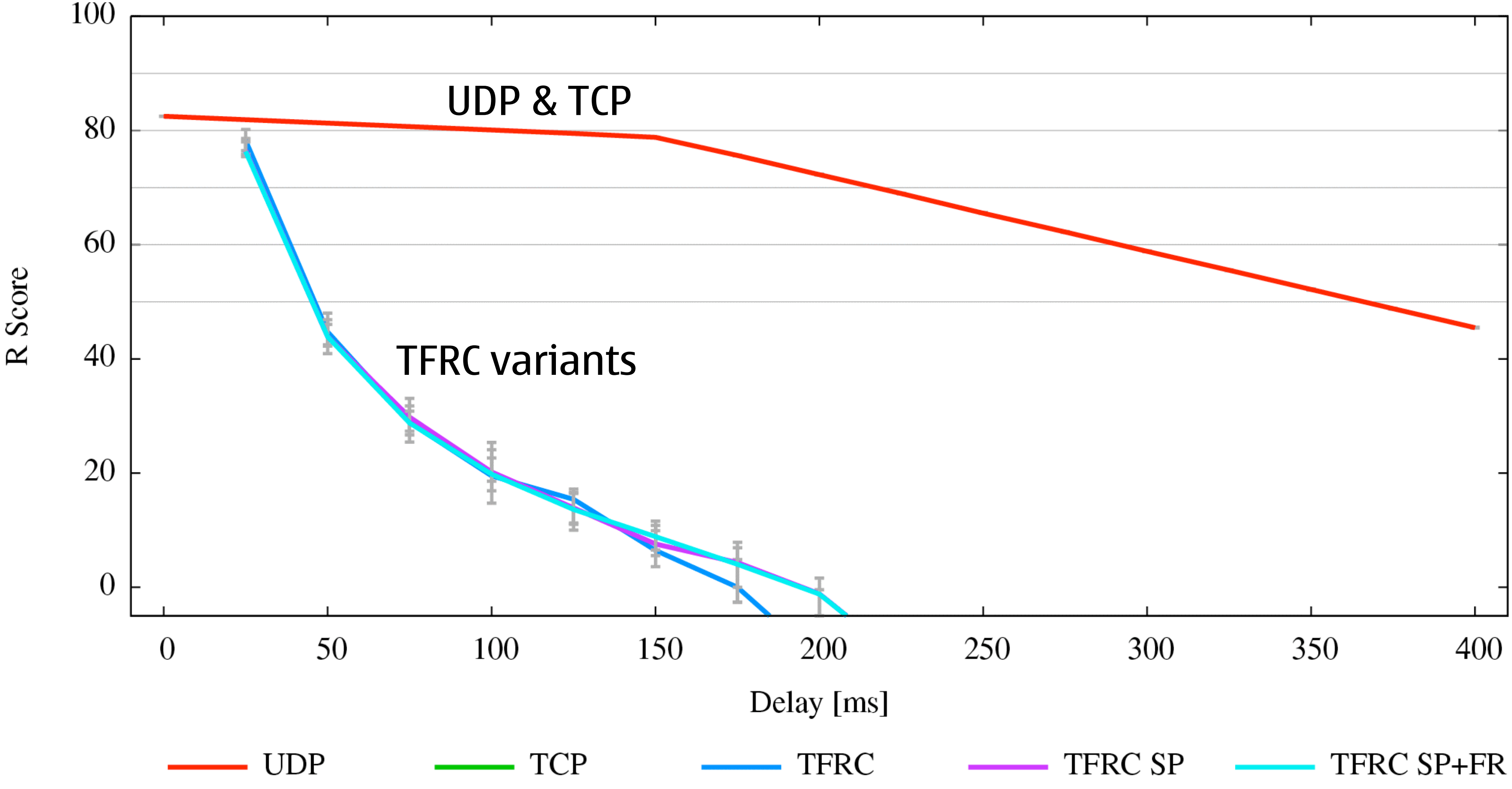


- experiments use the ITU-T E-Model to compute the R-Score over a received audio frame sequence
- R-Score approximates the Mean Opinion Score (MOS) when calibrated to specific codecs

R Score	MOS Score	Perceived Quality
90 – 100	4.34 – 4.50	Best
80 – 90	4.03 – 4.34	High
70 – 80	3.60 – 4.03	Medium
60 – 70	3.10 – 3.60	Low
50 – 60	2.58 – 3.10	Poor

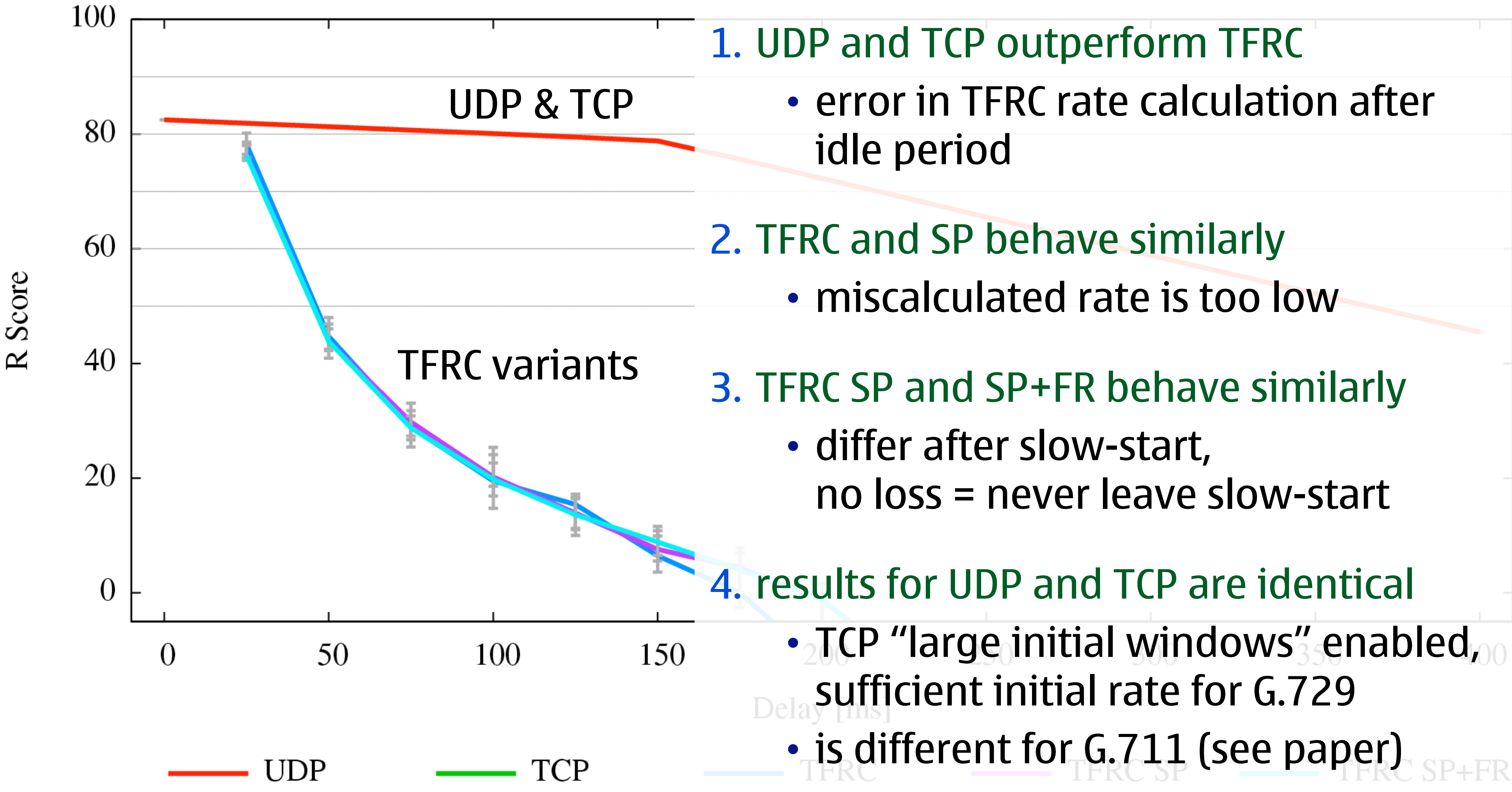
Codec	Frames/Packet	λ_1	λ_2	λ_3
G.711	1	0	30.00	15
G.729	1	10	47.82	18

Experimental Results: Varying Delay, No Loss



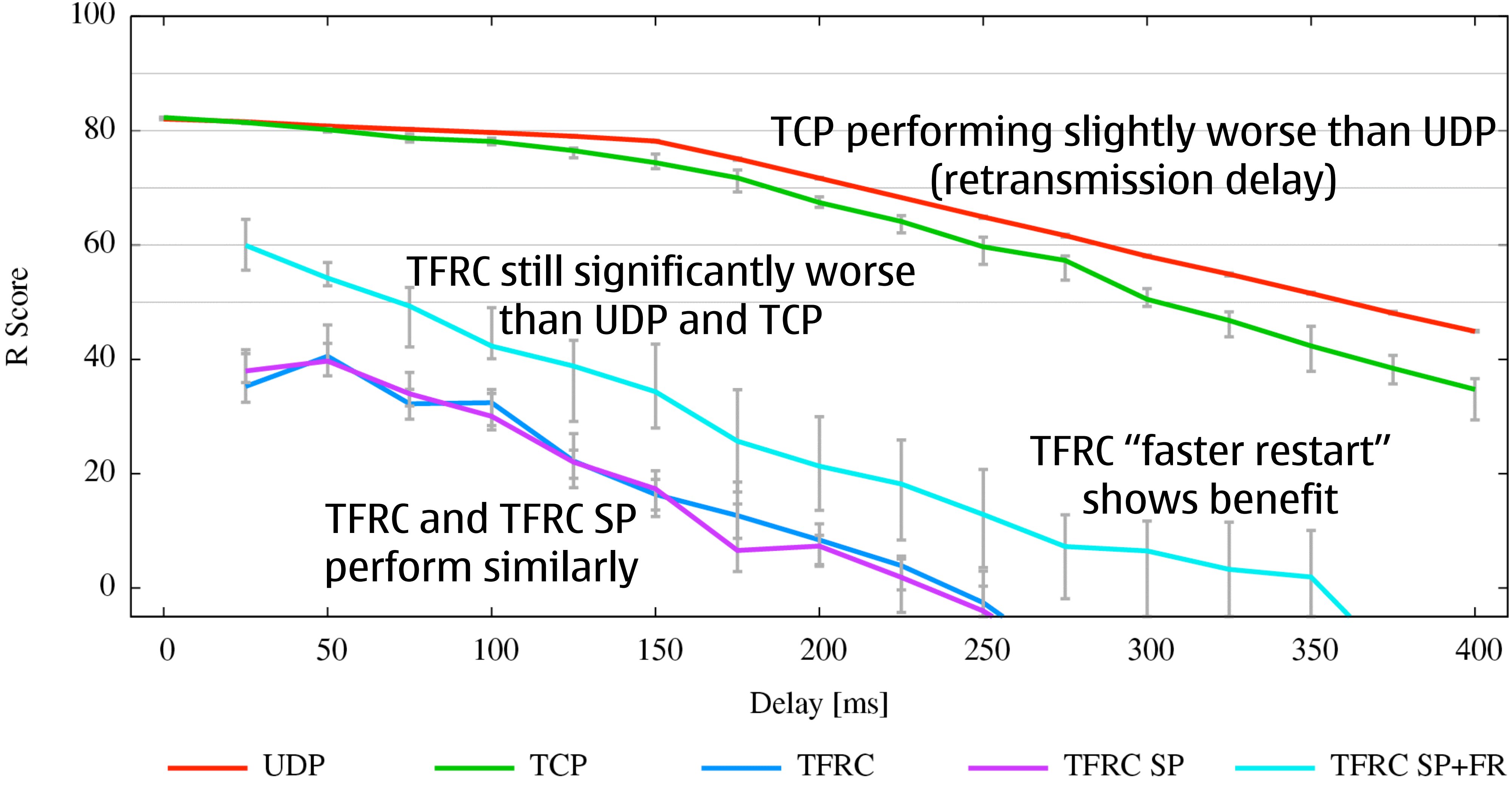
(all plots show medians over 15 runs; error bars show interquartile gap)

Experimental Results: Varying Delay, No Loss

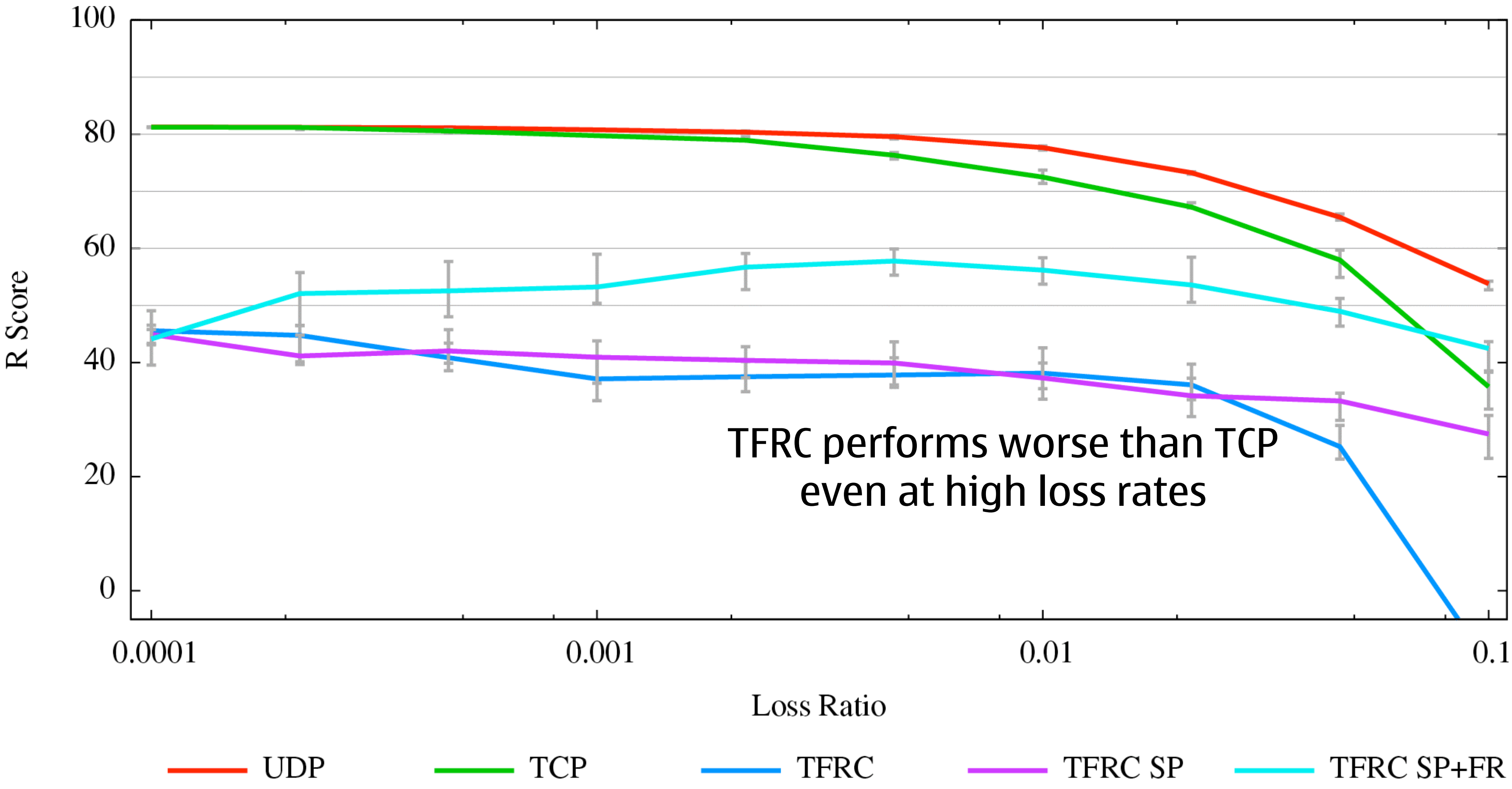


1. UDP and TCP outperform TFRC
   - error in TFRC rate calculation after idle period
2. TFRC and SP behave similarly
   - miscalculated rate is too low
3. TFRC SP and SP+FR behave similarly
   - differ after slow-start, no loss = never leave slow-start
4. results for UDP and TCP are identical
   - TCP "large initial windows" enabled, sufficient initial rate for G.729
   - is different for G.711 (see paper)

Experimental Results: Varying Delay, 0.1% Loss

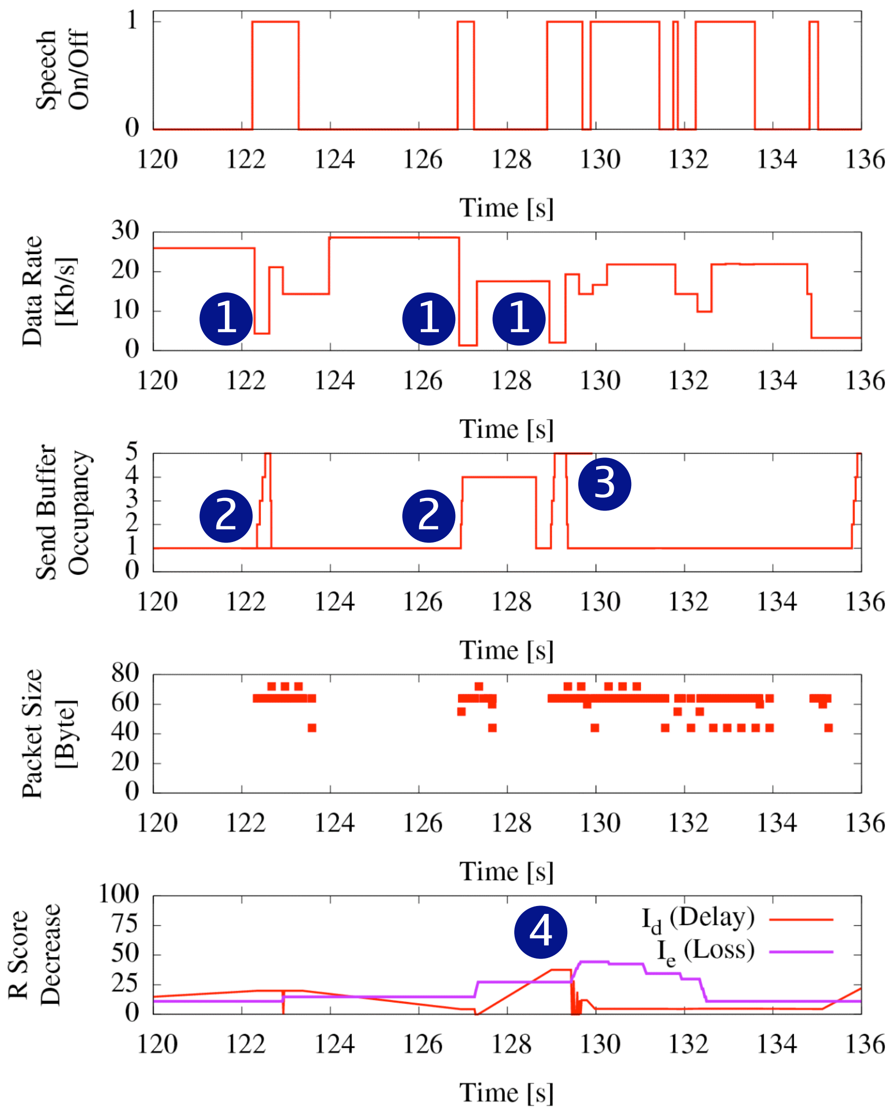


Experimental Results: 50ms Delay, Varying Loss



Analysis of Results

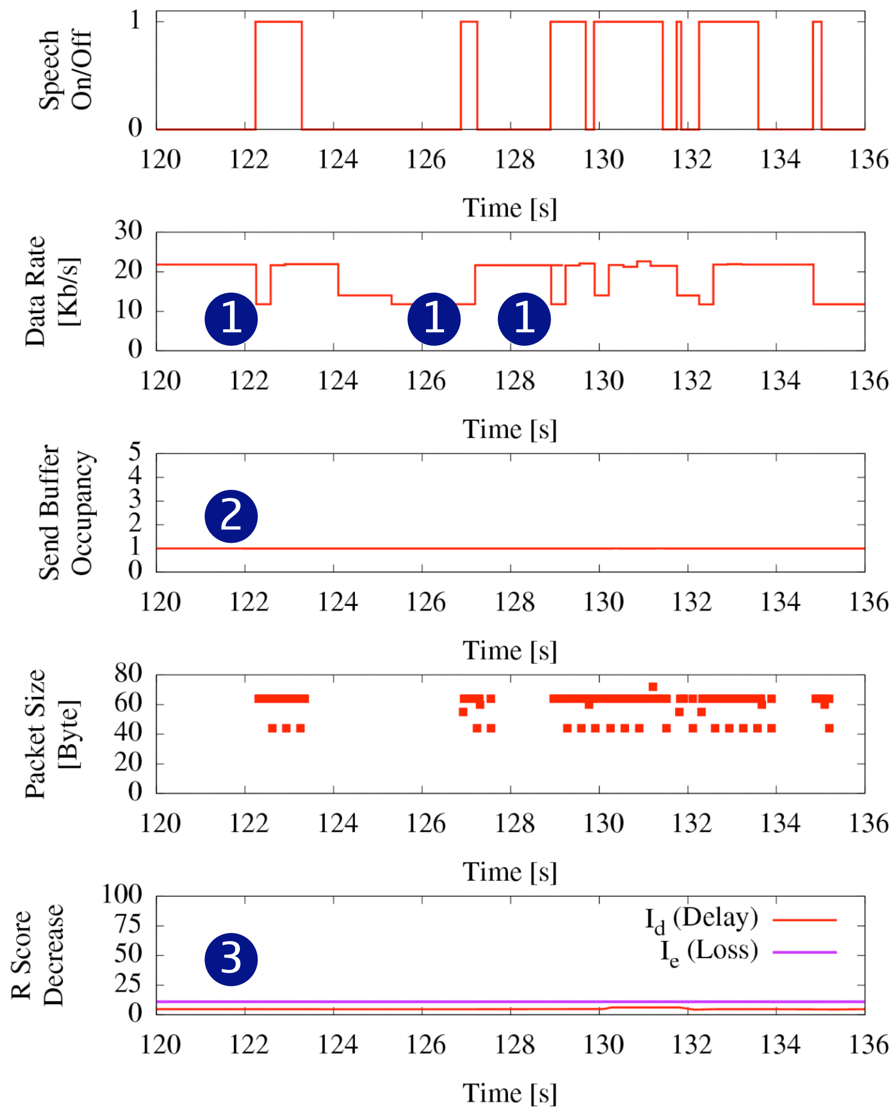
- all variants of TFRC are hampered by the miscalculation of the send rate after an idle period
  1. allowed rate drops sharply
  2. send buffer fills
  3. drops may occur
  4. impairments due to loss & delay increase
- additionally, TFRC slow-start is much slower that TCP
- finally, with data-limited apps, initialization of the TFRC loss history may be inaccurate



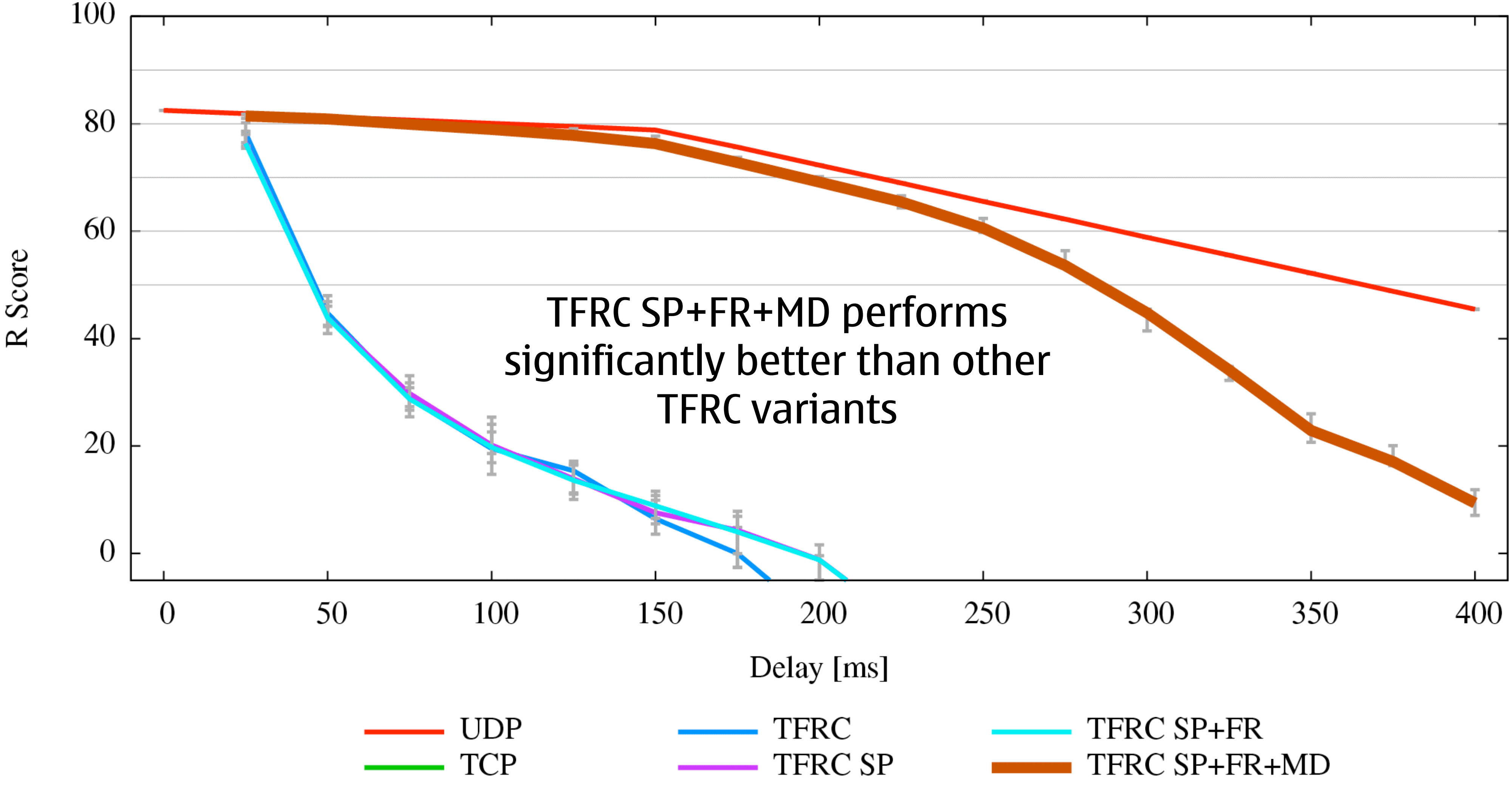
TFRC Improvement: TFRC SP+FR+MD

- improvement to TFRC SP+FR
  - maintains a minimum rate of 8 packets/RTT
    (same bandwidth use as TCP)
  - corrects the rate calculation after an idle period
  - corrects loss history initialization

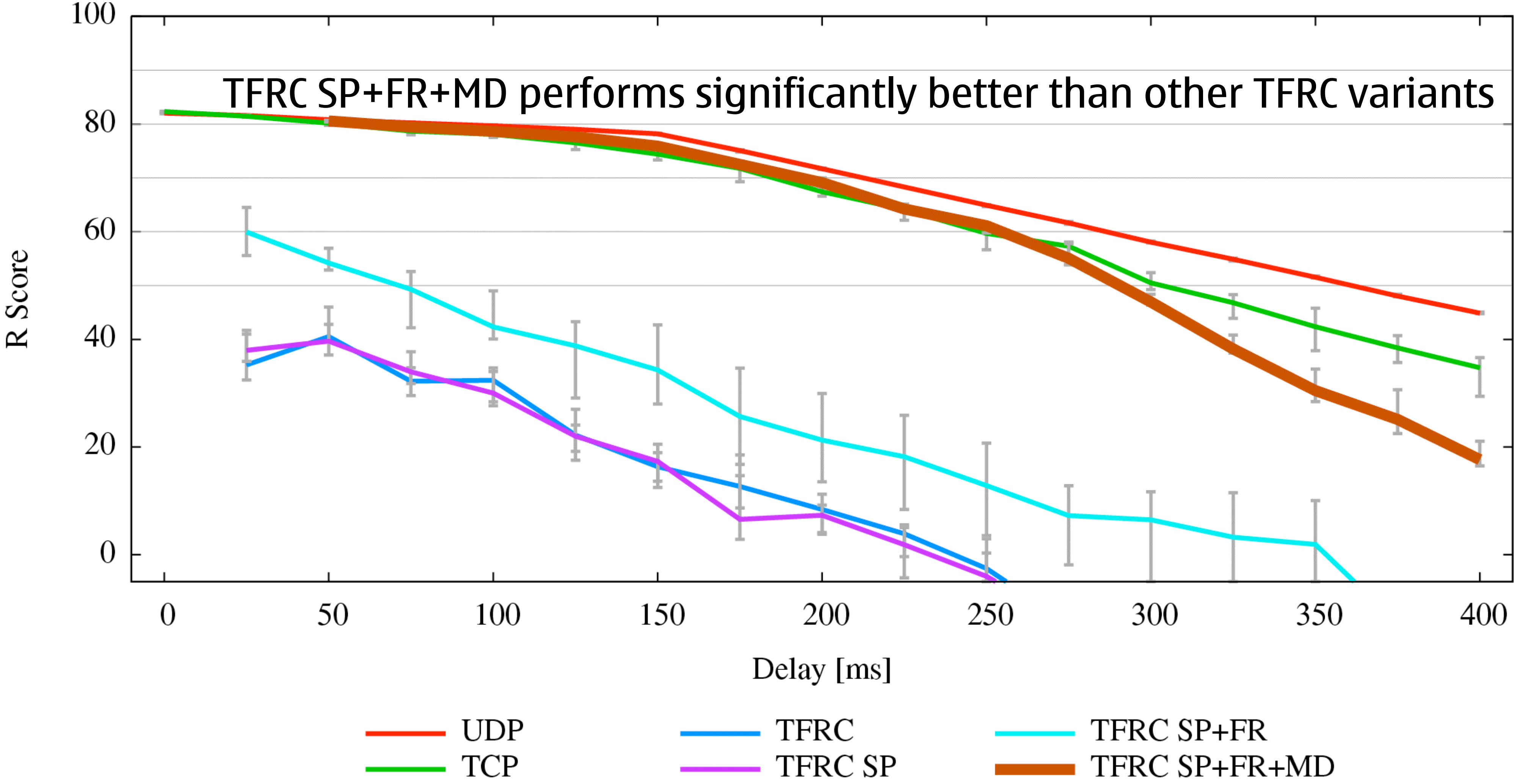
- result (same sample as before)
  1. maintains minimum rate
  2. no queuing delay or drops
  3. no delay or loss impairment



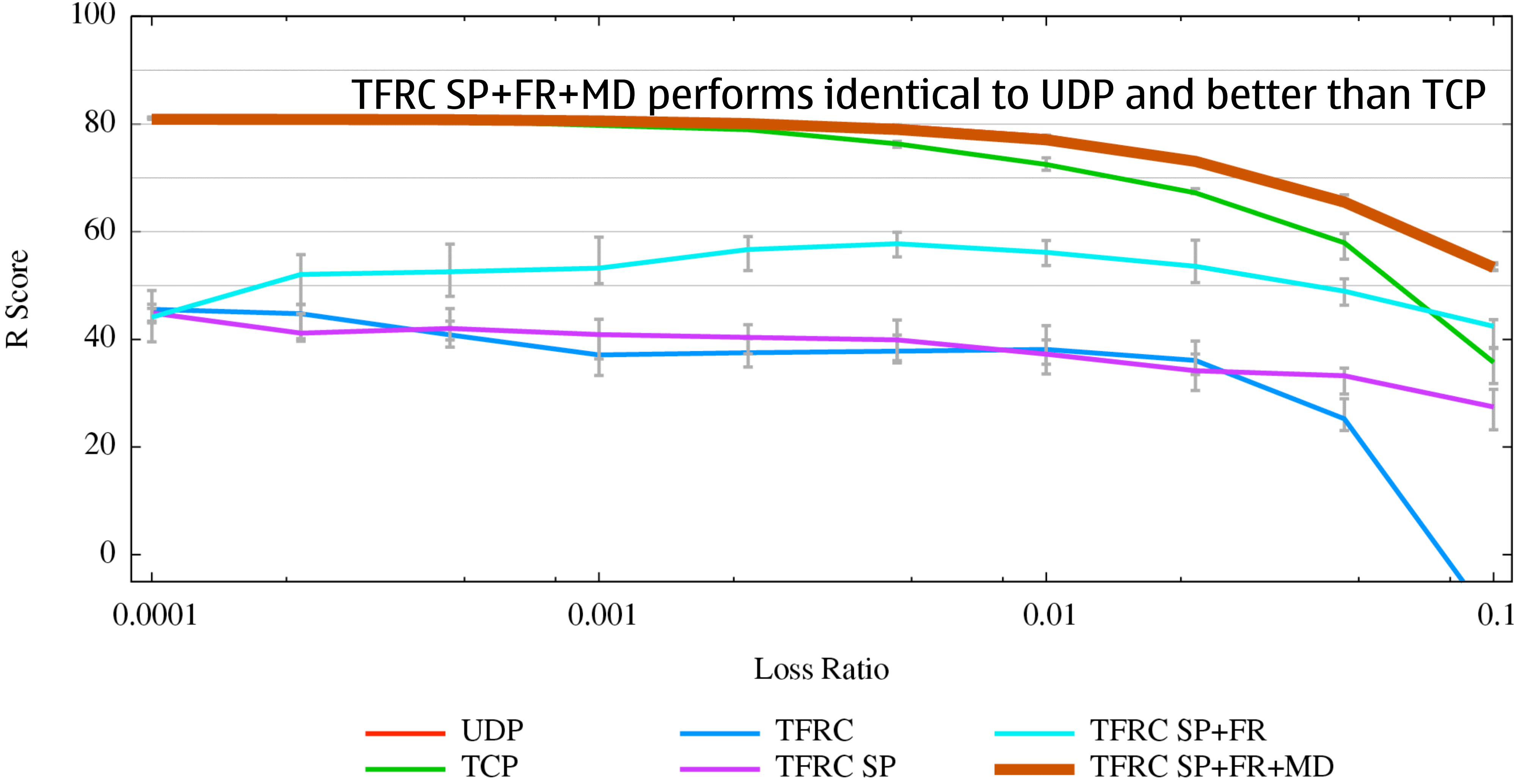
Experimental Results: Varying Delay, No Loss



Experimental Results: Varying Delay, 0.1% Loss



Experimental Results: 50ms Delay, Varying Loss



Conclusion

- extensive experimental analysis of voice quality over DCCP and other transport protocols
- identified design limitations that severely impact voice quality
  - original TFRC assumptions don't fit voice
    - large packets, continuous transmission, high-datarate
  - TFRC is less aggressive than a modern standard TCP
    - because it is based on a model of a simplified TCP Reno under limiting assumptions
- designed an improved TFRC variant for voice traffic and experimentally validated its effectiveness
  - contributed improvements to the IETF DCCP design process

Questions?