Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet

Jonathan Rosenberg
dynamicsoft
jdrosen@dynamicsoft.com

Henning Schulzrinne
Columbia University
hgs@cs.columbia.edu

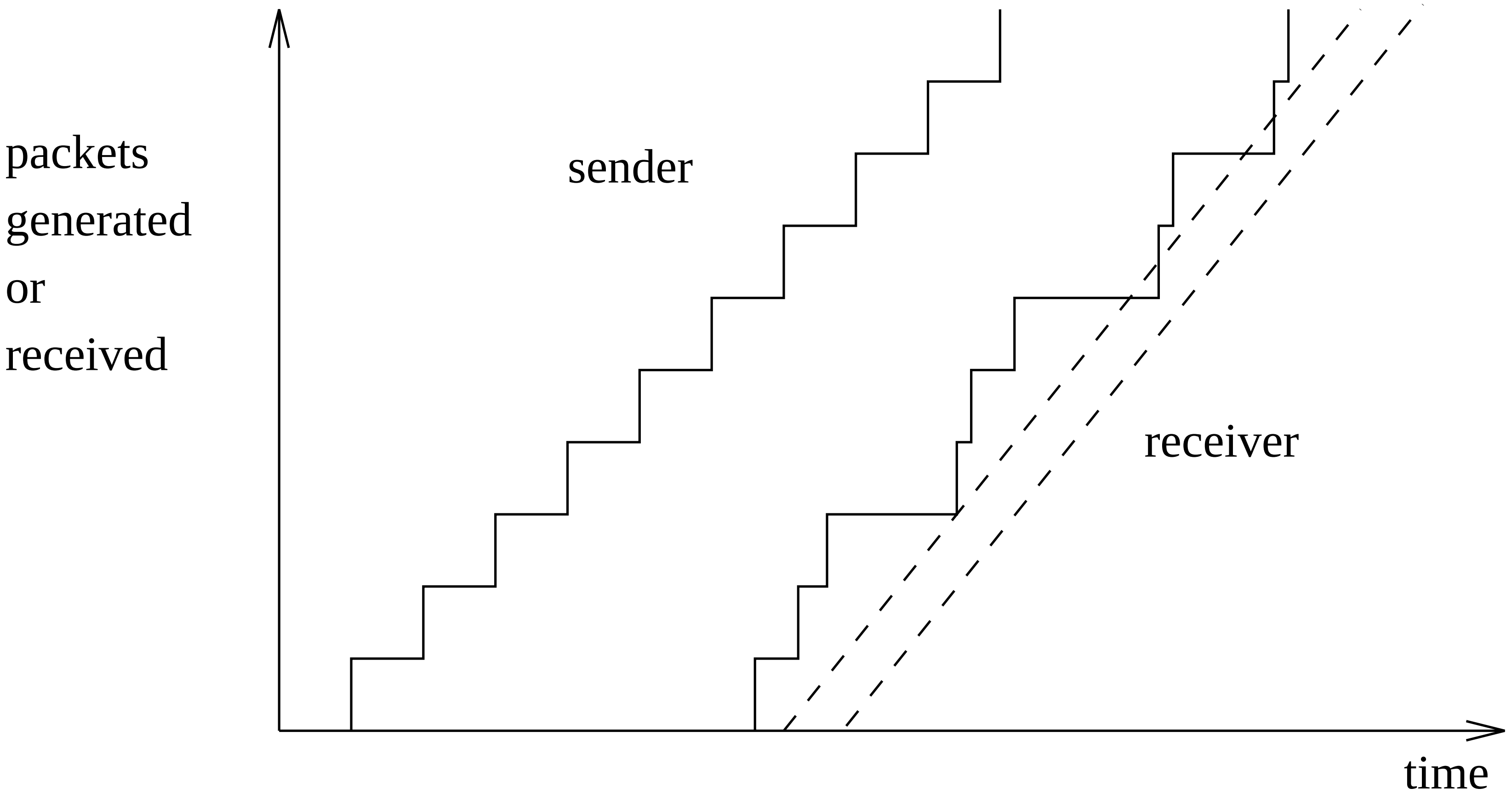
Lili Qiu
Cornell University
lqiu@cs.cornell.edu

IEEE Infocom 2000

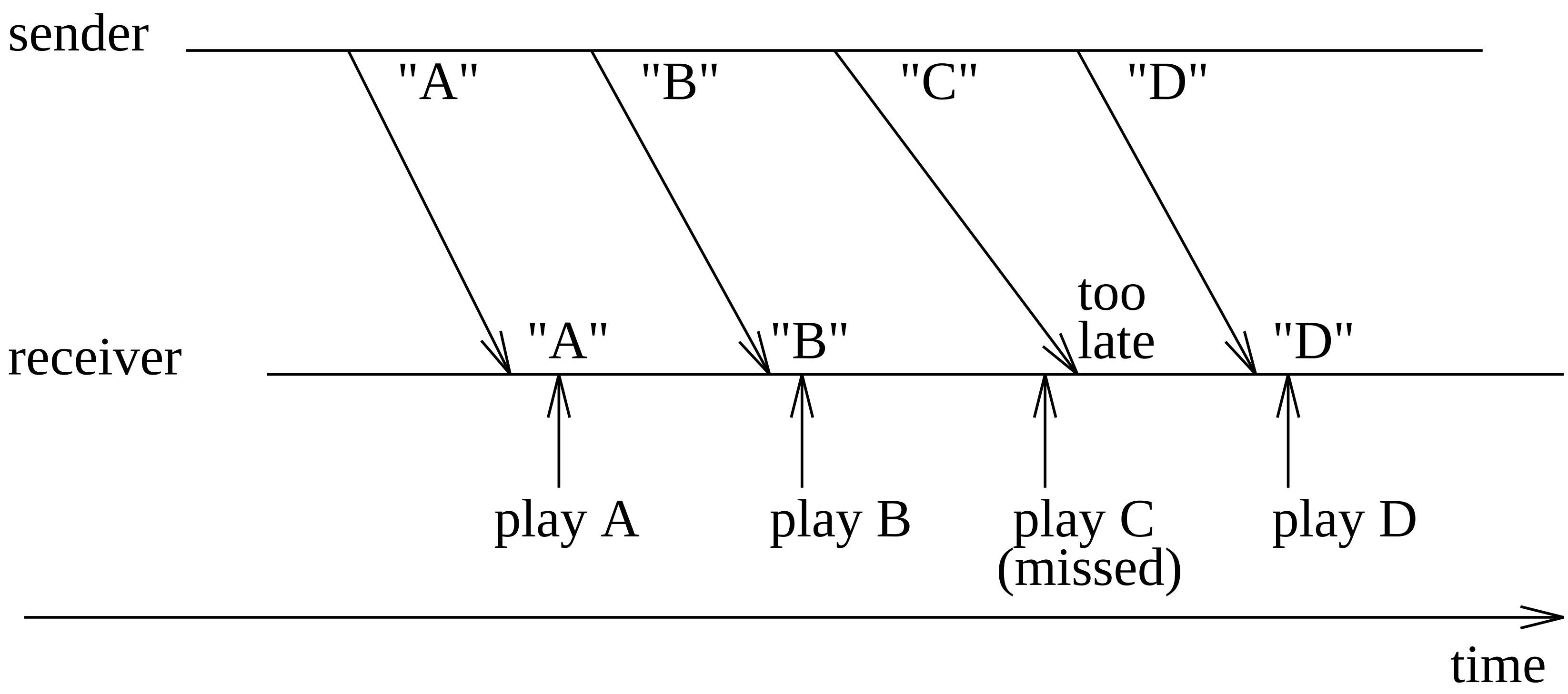
Background

- Voice quality on Internet poor for two reasons
  - Packet loss - highly variable from 0 to 100 %
  - Delay and jitter
- Packet loss compensated in many ways
  - Repair at receiver
  - Interleaving
  - Packet Forward Error Correction (FEC)
- Jitter compensated through adaptive playout buffers

Adaptive Playout Buffers



Adaptive Playout Buffers



Jitter Compensation Algorithms

All $p_i = t_i + \hat{d}_i + \mu \hat{v}_i$, $\mu = 4$

Exp-Avg: $\hat{d}_i = \alpha \hat{d}_{i-1} + (1 - \alpha) n_i$

Fast Exp-Avg: different α for delay $>$ or $<$ average

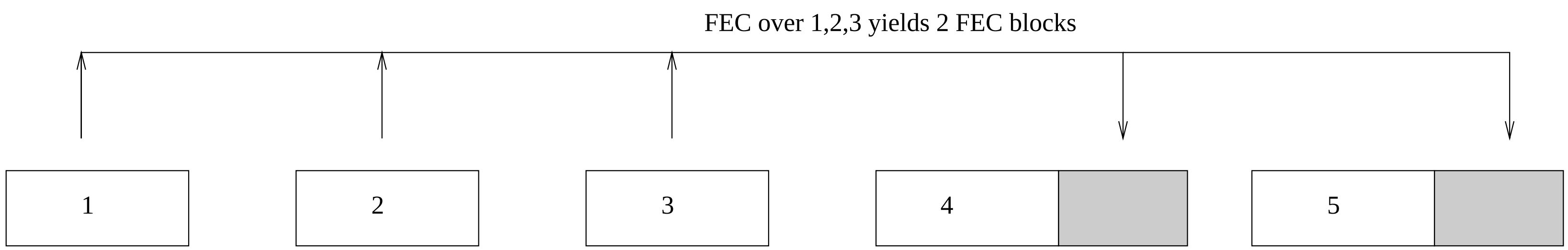
Min-Delays: minimum delay in talkspurt

Spk-Delay: spike detection

Window: quantile of delays among last O(1000) packets

Packet FEC

- Send additional data to recover past packets
  - Redundant codecs
  - Parity, Reed-Solomon applied to entire packet



Important Observation

- Playout buffer adaptation and FEC studied in isolation
- There is a coupling!
  - FEC requires delay in order to operate
  - Jitter buffers add delay
  - Actual playout delay must combine both
- Determination of playout delay must be based on both delay and loss and account for FEC and jitter

Problem Statement

- What is the desired performance metric for system?
  - Existing playout buffers generally seek to pick minimal delay that results in minimal loss
  - Existing speech codecs still work well for loss up to 5%
  - Our approach: choose minimum delay that results in a desired loss rate
- Problem statement
  - Determine playout buffer algorithm that is FEC aware, and is able to choose the minimal delay which results in some desired application loss rate after FEC is used

Virtual Delay Algorithms

- Class of algorithms
- Basic idea:
  - Define virtual delay d_V as the earlier of
    * Time of arrival
    * Earliest possible recovery time with FEC
  - Use existing playout buffer algorithms, but drive them with d_V instead of actual delay
    * Natural consequence of layering of FEC and playout buffers

Previous Optimal Algorithm

- Optimal algorithm
  - Offline
  - Examines delays of all arrived packets, and computes smallest delay that allows target application loss rate
  - Unrealizable
- Idea: make it realizable through delay
  - After talkspurt N, compute ideal delay for talkspurts $N - K$ to N. Let this be D_{opt}
  - For talkspurt $N + 1$, let playout delay be exponential weighted average of D_{opt}

Analytical Algorithm

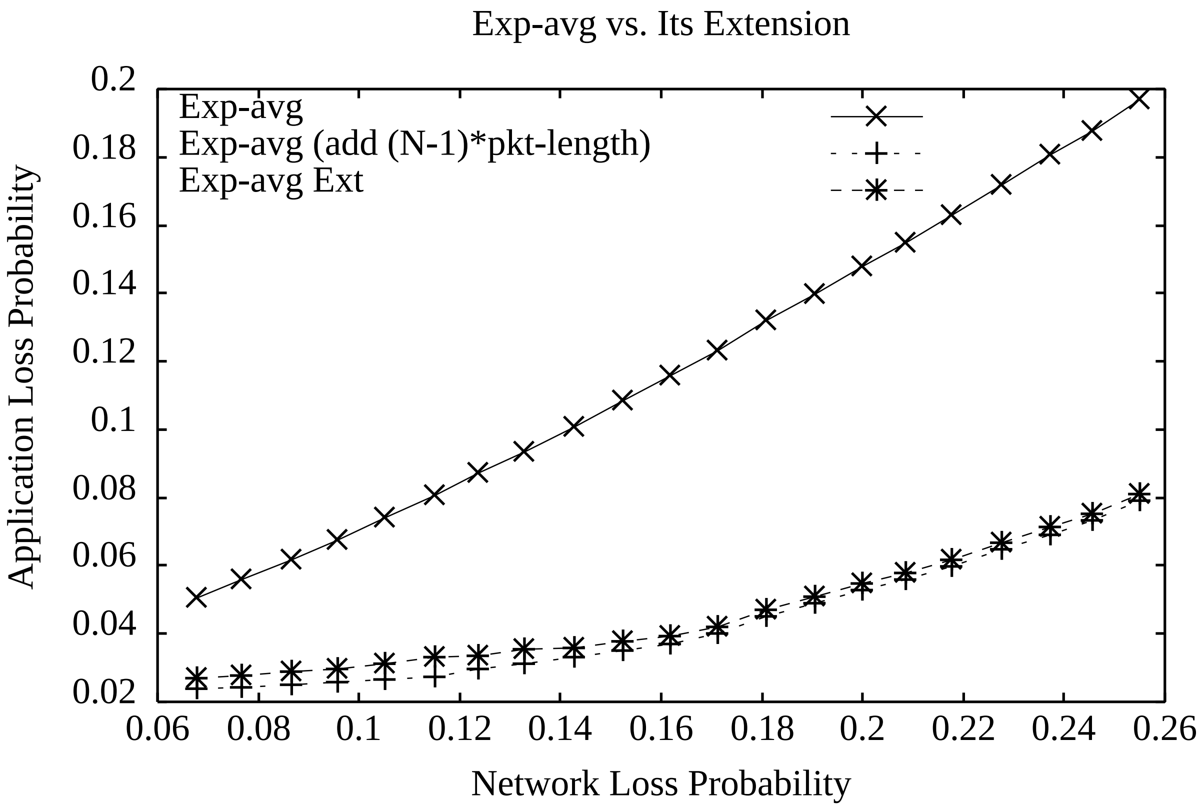
- Develop model of impact of loss and network delay r.v.'s on application loss rate given a specific FEC algorithm and playout delay D
- Measure packet loss and network delays
- Use model to determine application loss rate vs. playout delay given current packet loss and network delays
- Invert function to compute playout delay given desired application loss rate

Simulation Results I

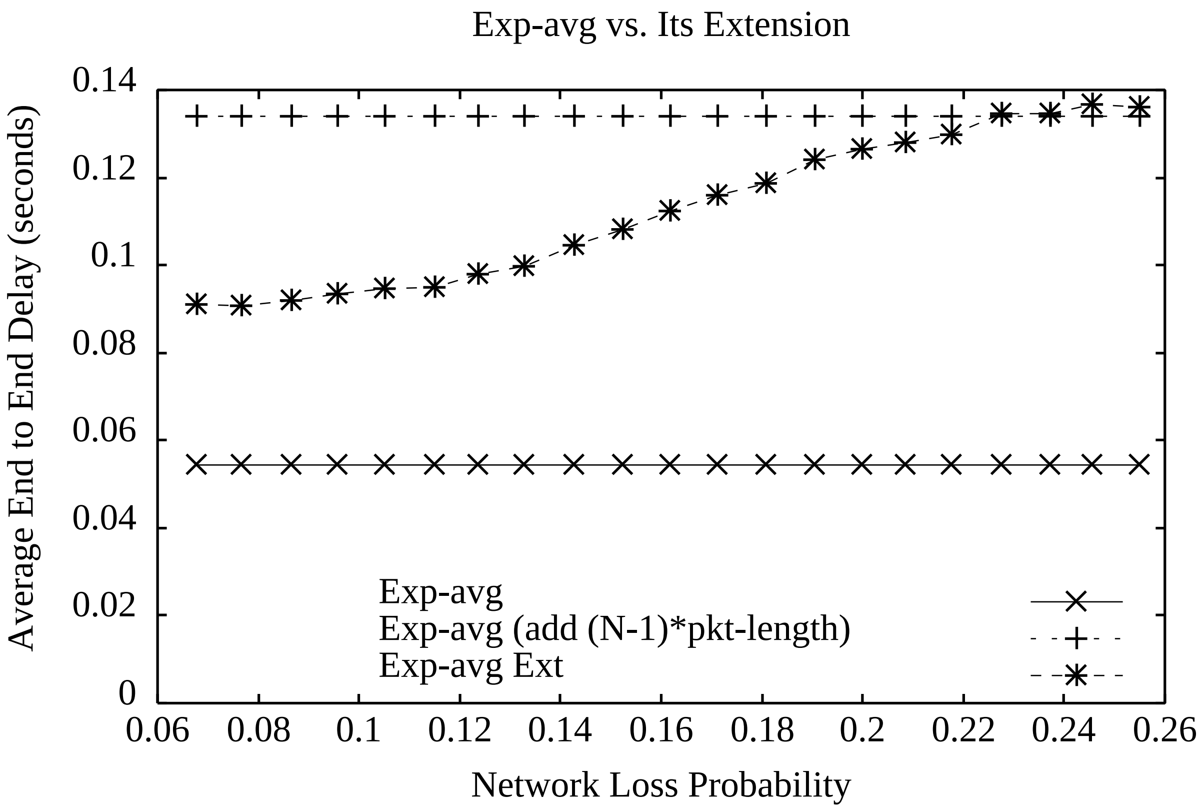
- Compare performance of adaptively virtual algorithms vs. existing uncoupled algorithms
- Two uncoupled algorithms
  - First ignores FEC
  - Second adds $(N-1)T_p$ to result of normal playout buffer
- For each item, left plot shows application loss probability vs. network packet loss probability
- Right plot shows average playout delay vs. loss probability

Simulation Results I: Loss

- Adaptively virtual exponential average



Simulation Results I: Delay



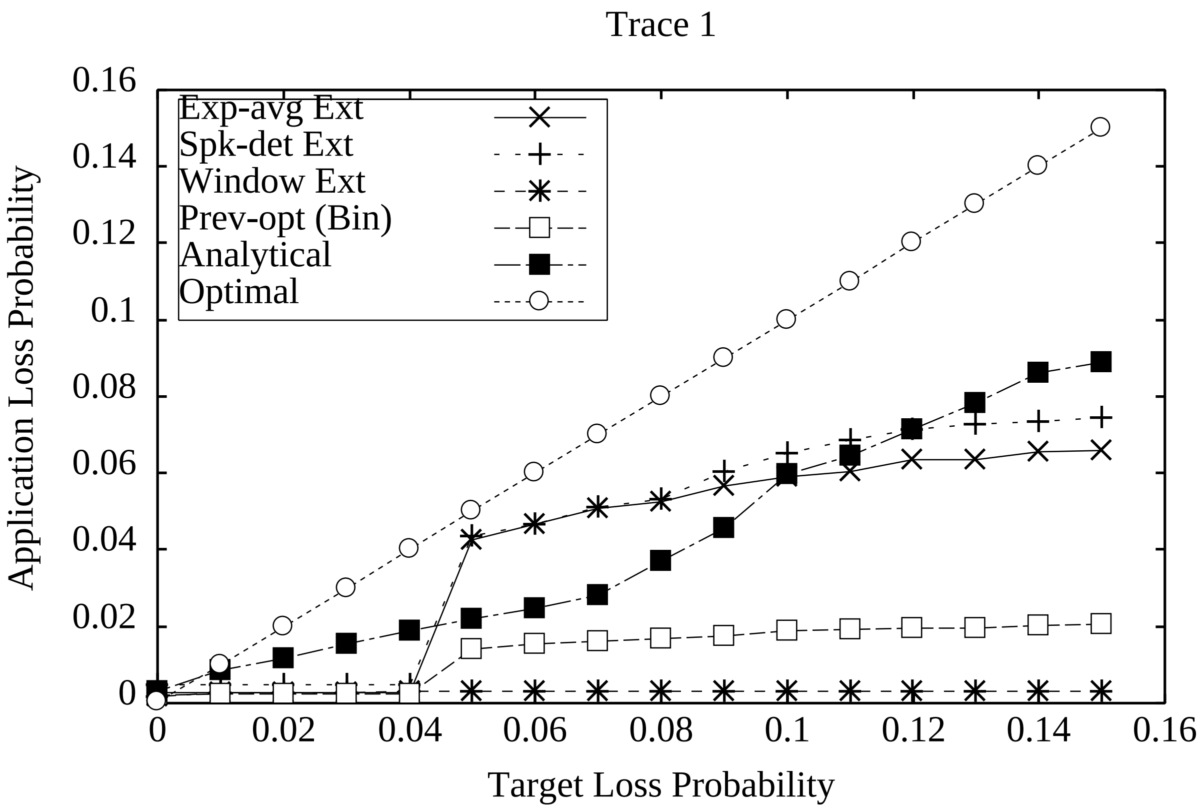
Observations on Performance

- First decoupled algorithm always underestimates needed delay, resulting in high loss
- Second algorithm always overestimates needed delay, resulting in low loss but high delay
- Our algorithms adapt well

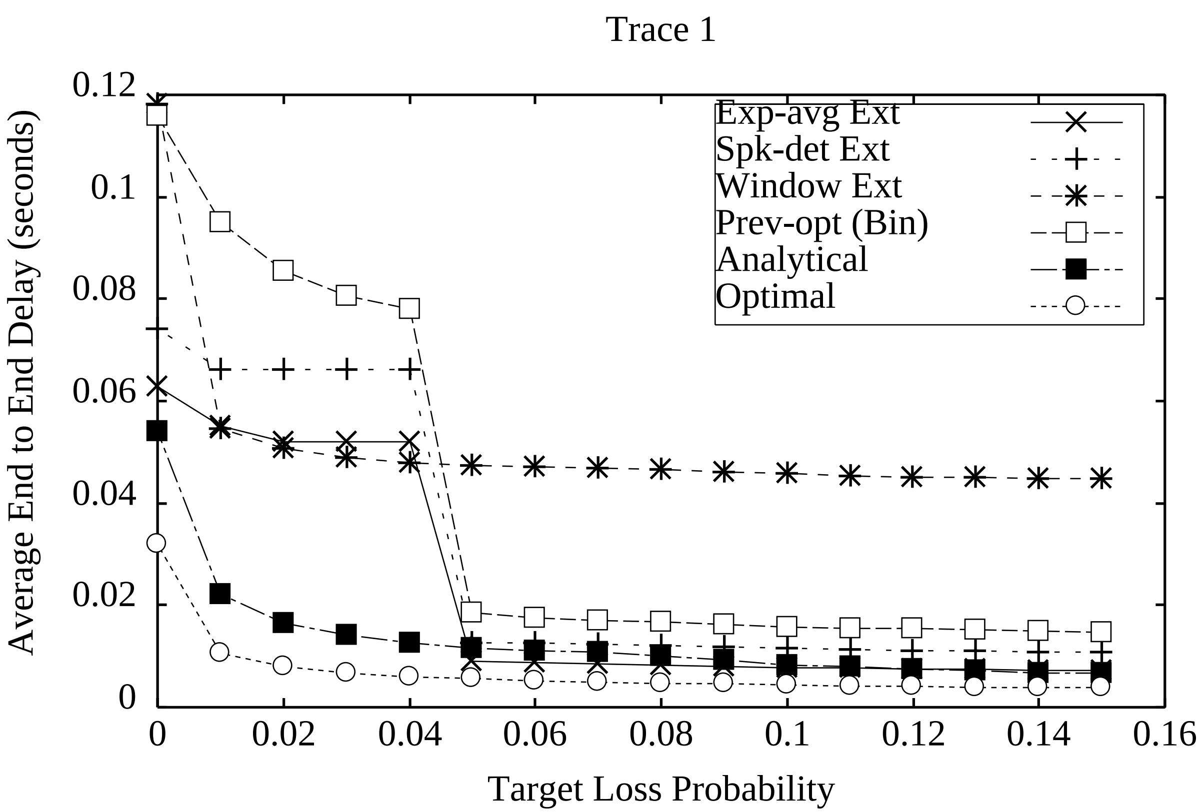
Simulation Results II

- Show ability of algorithms to maintain target application loss rate of 7% as network packet loss probability varies
- Left plot shows application loss probability vs. network packet loss probability
  - Linear is ideal
- Right plot shows average playout delay vs. loss probability
- Observations
  - Analytical does best job overall

Simulation Results II:Loss



Simulation Results II: Delay



Conclusions

- There is a need for algorithms which incorporate FEC awareness into playout buffer adaptation
- Better performance metric is ability of algorithms to meet a specific loss target rather than 0 loss
- Adaptively virtual algorithms incorporate awareness by taking traditional algorithms and driving them with virtual delay
- Previous optimal and analytical algorithms also integrate FEC awareness
- Performance results show all algorithms outperform decoupled algorithms
- Performance results show most algorithms do a good job meeting specific loss targets; no clear winner