RUTGERS
WINLAB | Wireless Information Network Laboratory

Timeliness in Lossless Block Coding

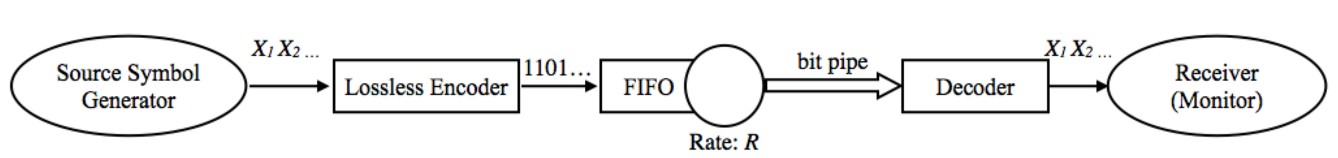
Jing Zhong
Advisor: Prof. Roy D. Yates
WINLAB, Rutgers University

Project Outlines

- We examine lossless data compression from an average delay perspective.
- Status updating system model:
  - A source generates time-stamped status update messages that are transmitted through a communication system to one or more receivers.
  - Updates must be as timely as possible.
- **Timeliness metric**: status update age.
- **Real-time lossless source coding** is a problem of status updating.
  - We evaluate the average status age in lossless sequential block coding systems.
  - We connect average age to the source coding error exponent with delay.
  - We propose the age-optimal block coding scheme.

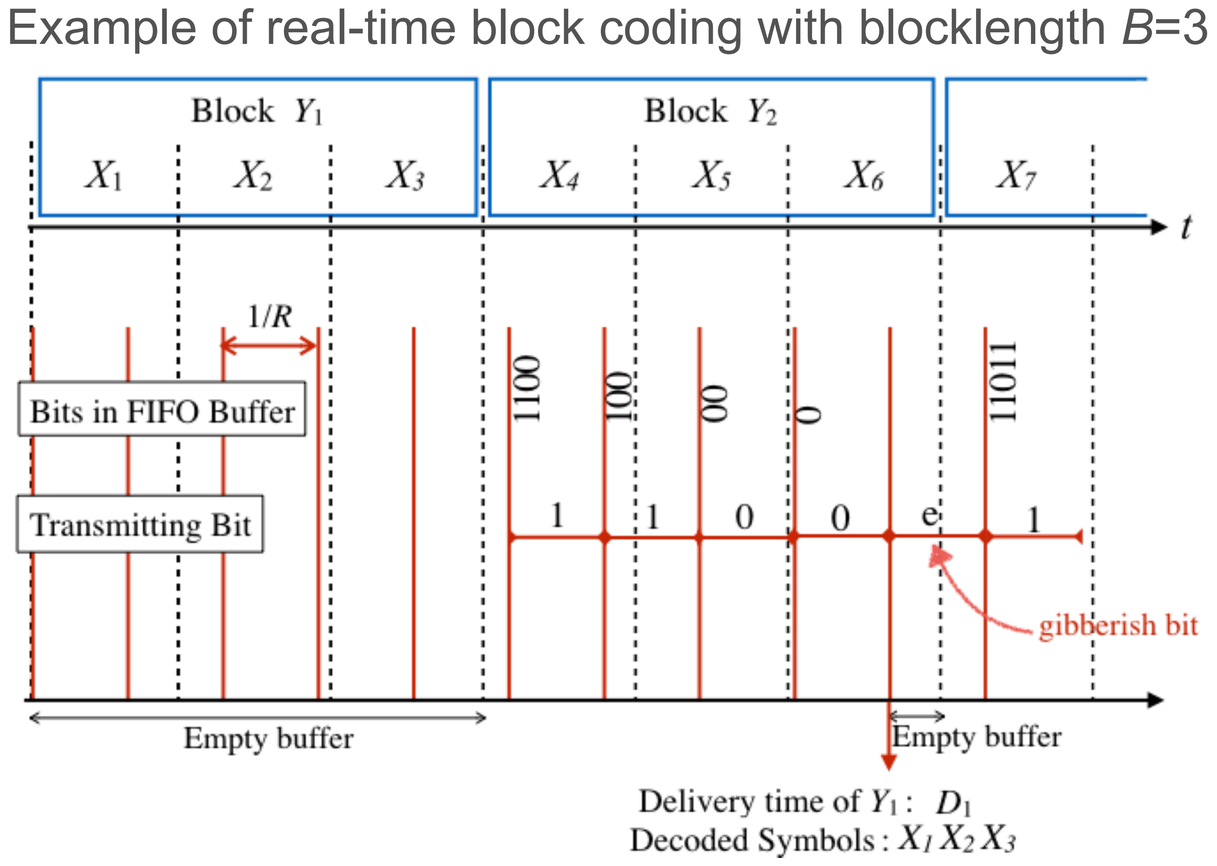
System Model

- An encoder receives input symbols one per unit time from an i.i.d. source and submits binary codewords to a FIFO buffer that transmits bits at a fixed rate to a decoder.

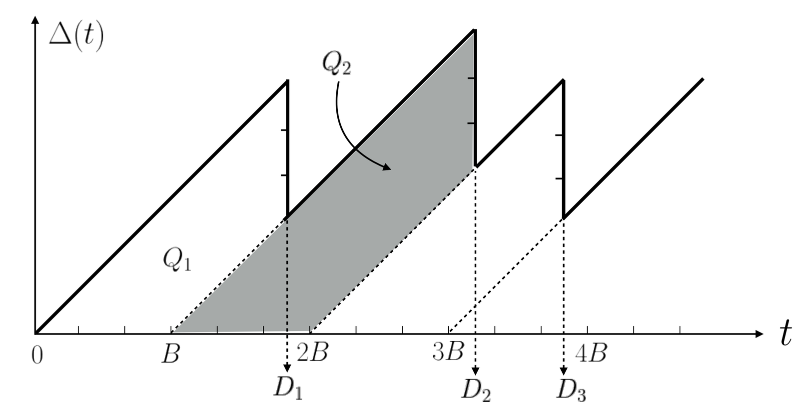


  - Each input symbol at the encoder is viewed as a status update by the source.
- System performance is characterized by the **average status update age**:
  - The number of time units (symbols) the decoder output lags behind the encoder input.

Example of real-time block coding with blocklength B=3



Status Age Analysis



Example of status age for block coding schemes

- Average age for block coding:
  - The average status update age is the normalized area under the sawtooth.

$$\Delta = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}(D_k - kB) + \frac{B}{2} = \mathrm{E}[T] + \frac{B}{2}.$$

average system time of symbol k — average waiting time + average service time

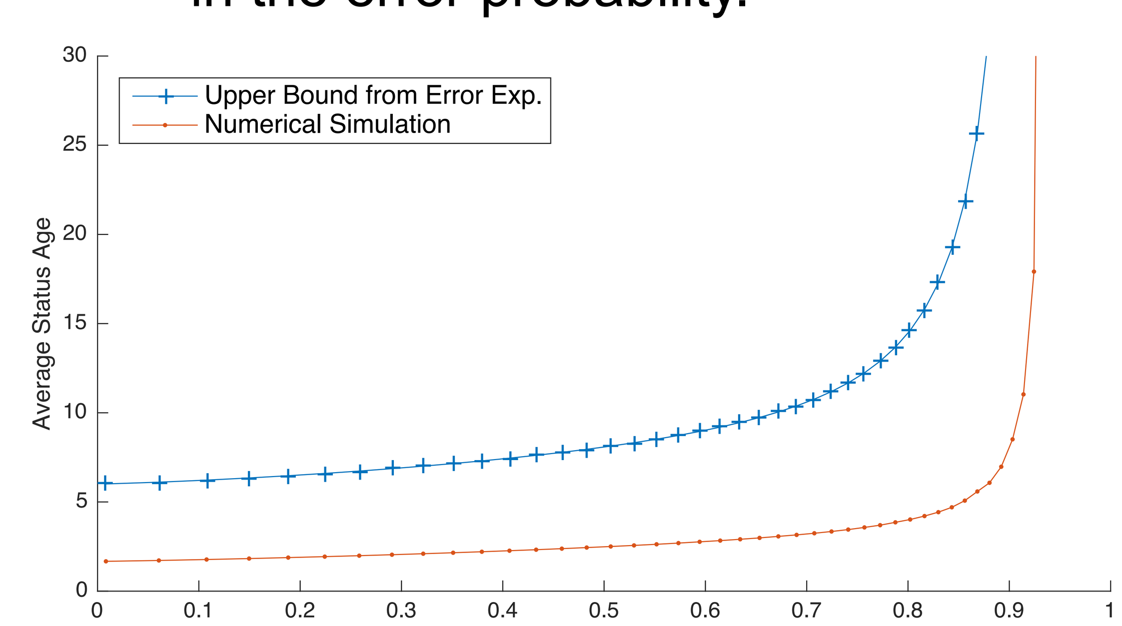
  - D/G/1 queue model:
    - service time is proportional to the length of the encoded sequence.
    - the average waiting time can be solved numerically.
      - waiting times form a discrete Markov chain.
      - obtain the stationary distribution of M.C. from the discrete distribution of the service time.

- Connection to source coding error exponent:
  - a delay constrained error exponent is achievable if and only if

$$\Pr[\hat{x}_{n-\delta}(n) \neq x_{n-\delta}] \leq K2^{(-\delta E_S(R)-\epsilon)}, \quad \text{for all } \delta \text{ and } n > \delta.$$

$$\Delta \leq \frac{K\,2^{2E_S(R)}}{(2^{E_S(R)}-1)^2} \triangleq f(K, E_S(R)).$$

  - the error exponent describes the convergence rate of the error probability.
  - this bound also scales with the constant term K in the error probability.



$P_X(a) = p$
$P_X(b) = \frac{1-p}{2}$
$P_X(c) = \frac{1-p}{2}$

Prefix-free block coding example with blocklength B=2:
- Encode aa to a single bit 0, and anything else to length-4 codewords.
- Fix channel rate R and vary the source entropy.
- The error exponent does not lead to an accurate description of the status age for small delay and small blocklength.

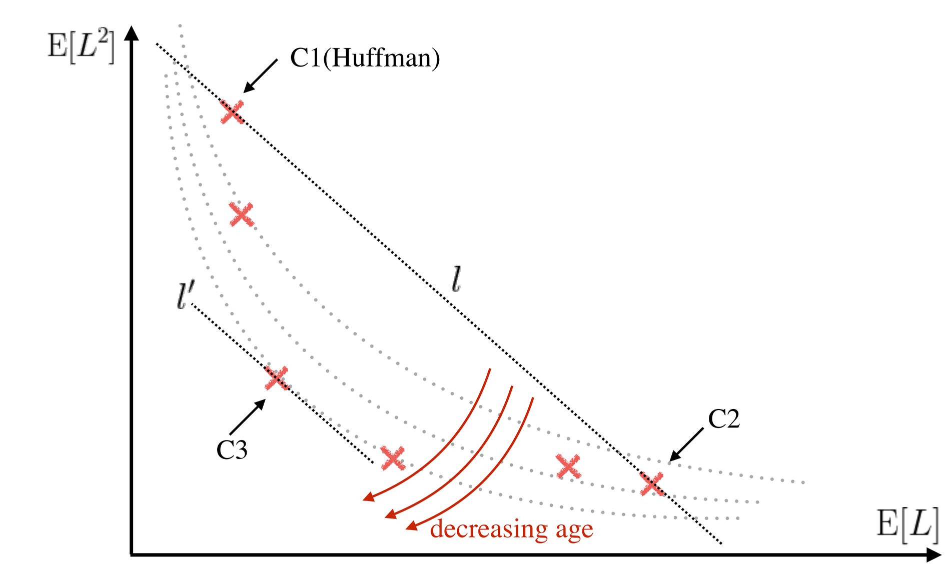
Age-optimal Block Code

- Alternative upper bound on the average waiting time of G/G/1 queue:

$$\Delta \leq \frac{\mathrm{Var}[L]}{2R(BR-\mathrm{E}[L])} + \frac{\mathrm{E}[L]}{R} + \frac{B}{2} = \frac{\mathrm{E}[L^2]-\mathrm{E}^2[L]}{2R(BR-\mathrm{E}[L])} + \frac{\mathrm{E}[L]}{R} + \frac{B}{2}.$$

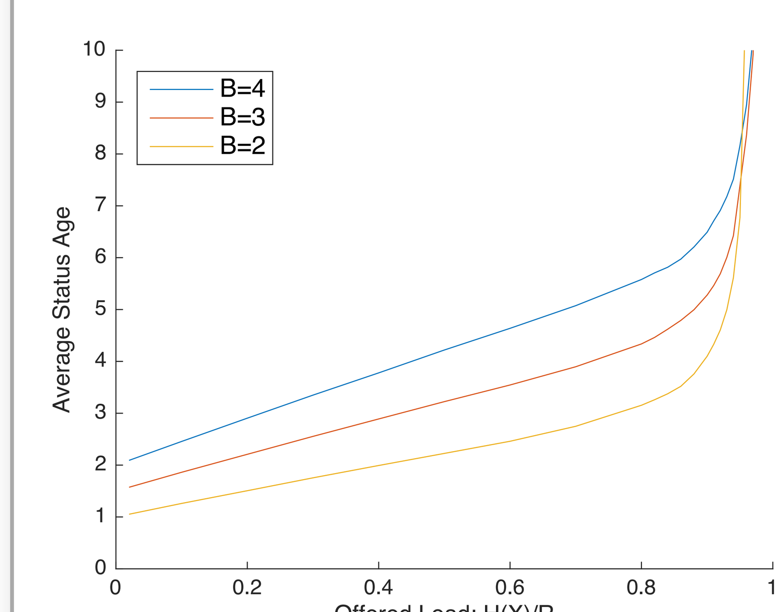
quasi-linear function of expected code length & second moment of code length

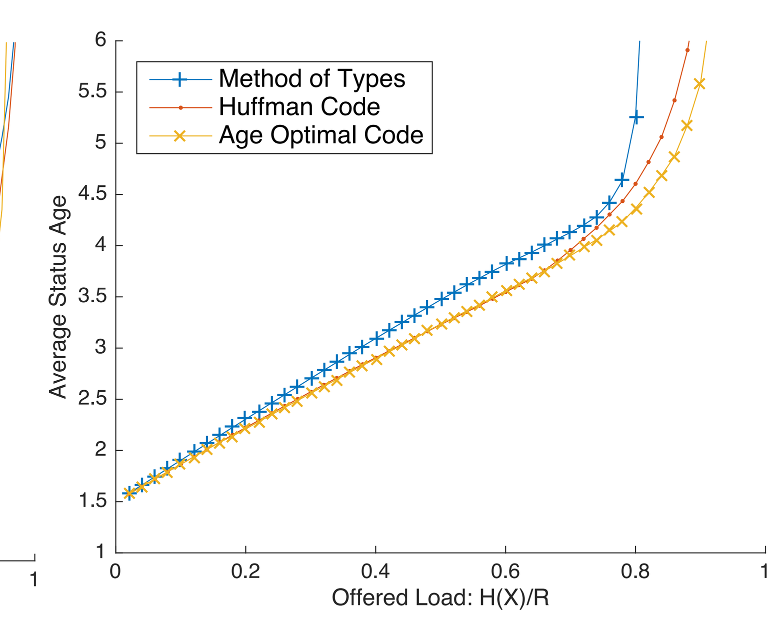
- Age minimized block code:
  - search over all the possible codebooks lying on the lower left corner of convex hull iteratively.



The search process starts from two end points: Huffman code C1 and the minimum second moment code C2.

- Computes the analytical average age function iteratively and returns the optimal code tree that gives the minimum.





Comparison of different blocklengths:
- Encoding large blocks can result in high status age when the load is low.
- Small blocks encoding yields inefficient compression, which increases the offered load on the FIFO buffer and causes queueing delays.

- Error exponent is optimized by the scheme uses method of types.
- Huffman code is only optimal when the system is idle.
- With same blocklength, age-optimal code outperforms Huffman code significantly when the system is heavily loaded.

Future Work

- Techniques to handle large blocklengths.
- Sources with memory.
- Age-optimal universal coding scheme.
- Coding with shared network resources.

December 2015

WINLAB