The Need For Differentiated "Quality Of Service"
A Information Theoretic Approach

Anant Sahai and Sanjoy Mitter
MIT LIDS
{sahai,mitter}@lids.mit.edu

1. Layering and Motivation
   *Our goal: find a suitable toy*

2. Toy problem:
   Mean-squared tracking of an unstable vector source over an erasure channel with noiseless feedback

3. Review of “anytime” capacity and separation theorem

4. Impossibility without differentiated treatment

5. Possible with differentiation through priorities

6. Conclusion
Layering And Motivation

- Layering enforces abstractions (equivalence relations)
- Source/Channel separation theorem justifies layering in terms of bits and the rate of bitstreams
- “Real-time” applications seem to demand more than just rate
- Different streams demand different treatment (audio, video, control, email)
- Why?
Traditional Answers

1. *(Biological/Cultural)* Human users have preferences
2. *(Economic/Business)* Enabling price discrimination
   - Need to turn a profit and/or control congestion
   - Differentiate service to have different prices (and vice versa)
3. *(Theoretical/Technical)* Delay Matters
   - Information Theory takes limit of large end-to-end delays
   - Real world wants relatively small delays for complexity and other purposes
   - To get good performance we must do some kind of “joint coding”
   - Since sources are different, so must be the services
Our Goal

- Traditional answer philosophically unsatisfying
  Why are human preferences that way? Could it be otherwise? Is it really only about large profits or small delays? What does “real-time” truly mean?

- Want a more classically information theoretic toy situation
  1. No humans in the loop
  2. Mathematically tractable source
  3. Expected per-letter additive difference distortion measure
  4. Mathematically tractable channel
  5. Allow limit of large (but finite) delays
  6. Allow infinite computational resources
  7. Still end up sharply distinguishing between nondifferentiated service and differentiated service
Our Toy Problem

- Markov source $X_{t+1} = AX_t + W_t$ with suitable $A$, $X_0 = 0$, and $W_t$ having bounded support $\|W_t\|_\infty \leq \frac{\omega}{2}$.

- Can we design encoding and decoding systems that go in the boxes above so that the estimates $\hat{X}_t$ track the source $X_t$ in a mean squared sense:

  $$\sup_{t>0} E \left[ \|X_t - \hat{X}_t\|^2 \right] \leq D < \infty$$

- $\hat{X}_t$ must be generated by time $t+d$. We are allowed to choose $d$ large, but finite.
The Specifics

- Binary Erasure Channel with erasure probability $e = 0.27$

- Unstable $A$ matrix:

$$A = \begin{pmatrix}
1.178 & 0 & 0.04 & 0 & 0.04 \\
1.08 & 1.058 & 0.36 & 0 & 0.36 \\
0.36 & 0 & 1.178 & 0 & 0.12 \\
-0.12 & 0 & -0.04 & 1.058 & -0.04 \\
-0.12 & 0 & -0.04 & 0 & 1.018
\end{pmatrix}$$

- Transformed coordinates $\tilde{X}_t = TX_t$, we have diagonal dynamics:

$$\tilde{A} = \begin{pmatrix}
1.258 & 0 & 0 & 0 & 0 \\
0 & 1.058 & 0 & 0 & 0 \\
0 & 0 & 1.058 & 0 & 0 \\
0 & 0 & 0 & 1.058 & 0 \\
0 & 0 & 0 & 0 & 1.058
\end{pmatrix}$$

if

$$T = \begin{pmatrix}
3 & 0 & 1 & 0 & 1 \\
0 & 1 & -2 & 0 & 3 \\
0 & 0 & 1 & 2 & 1 \\
0 & 0 & 0 & 1 & -1 \\
1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

- So $\tilde{\Omega} \leq 6\Omega$ is the transformed bound on noise.

- Can think of $\{\tilde{X}_t\}$ as 5 scalar sources
The Requirements

• Sequential Rate Distortion theory tells us that the total rate

\[ R \geq (\log_2(1.258) + 4 \log_2(1.058)) \]

for finite mean-squared error

• For finite mean-squared error on \( X_t \), we need to have finite mean-squared error on \( \tilde{X}_t \) and hence on each component thereof.

• Scalar separation theorem for unstable Markov processes gives:

\[
C'_{\text{anytime}}(2 \log_2(1.258)) > \log_2(1.258) \\
C'_{\text{anytime}}(2 \log_2(1.058)) > \log_2(1.058)
\]

for finite mean-squared error on the fast and slow components.

• Have to use “any-time” capacity. For erasure channel with noiseless feedback, let \( \eta \) range over \((0, \infty)\):

\[
C_{\text{anytime}}(\eta - \log_2(1 + (2^n - 1)e)) = 1 - \frac{1}{\eta} \log_2(1 + (2^n - 1)e)
\]
Motivating Case: Erasure With Feedback

- Probability $e$ that a sent bit gets erased. Repeat till it gets through.
- Input buffer stays stable as long as $R < 1 - e$
- If we wait $d$, the appropriate bit is not available iff the input buffer exceeds $d$. This is an exponentially decreasing function of $d$.
- Interpreted as variable delay messages, no bit errors, though we might have to wait.
Generalize: Anytime Capacity

- Have a fixed encoder, but let the decoder be parameterized by the delay. Want a good estimate “anytime” we ask for one.
- “Reliable Transmission” means every bit is eventually correctly received. Parameterize by the rate at which \( P(S_t \neq \hat{S}_t) \) goes to zero as delay increases.

\[
C_{\text{anytime}}(\alpha) = \sup \left\{ R \mid \exists (\mathcal{E}, \mathcal{D}, K) \forall d > 0 \\
\text{Rate} = R, \text{Delay} = d, \\
P_{\text{error}}(\mathcal{E}, \mathcal{D}, d) \leq K2^{-\alpha d} \right\}
\]
Comparison With Classical Notions

\[
C = \sup_R \left\{ \begin{array}{l}
\forall \epsilon > 0 \ \exists (\mathcal{E}, \mathcal{D}, d') \ \forall d > d' \\
\text{Rate} = R, \text{Delay} = d, \\
P_{\text{error}}(\mathcal{E}, \mathcal{D}, d) \leq \epsilon
\end{array} \right. \\
\exists (\mathcal{E}, \mathcal{D}, K) \ \forall d > 0 \ \forall \epsilon \geq K2^{-\alpha d}
\right\}
\]

\[
C_{\text{anytime}}(\alpha) = \sup_R \left\{ \begin{array}{l}
\forall \epsilon > 0 \ \exists (\mathcal{E}, \mathcal{D}, d') \ \forall d > d' \\
\text{Rate} = R, \text{Delay} = d, \\
P_{\text{error}}(\mathcal{E}, \mathcal{D}, d) \leq \epsilon
\end{array} \right. \\
\exists (\mathcal{E}, \mathcal{D}, d') \ \forall d > d' \ \forall \epsilon > 0
\right\}
\]

\[
C_{\text{zero}} = \sup_R \left\{ \begin{array}{l}
\forall \epsilon > 0 \ \exists (\mathcal{E}, \mathcal{D}, d') \ \forall d > d' \\
\text{Rate} = R, \text{Delay} = d, \\
P_{\text{error}}(\mathcal{E}, \mathcal{D}, d) \leq \epsilon
\end{array} \right. \\
\exists (\mathcal{E}, \mathcal{D}, d') \ \forall d > d' \ \forall \epsilon > 0
\right\}
\]

- Anytime capacity is stronger than regular Shannon capacity, but weaker than zero-error capacity.

\[
C_{\text{zero}} \leq C_{\text{anytime}}(\alpha) \leq C
\]

- Closely related to the idea of error-exponents, but stronger since the anytime encoder \( \mathcal{E} \) must be the same regardless of the delay allowed in decoding.

- Just as in the erasure case, can think of the anytime decoder as producing a variable length message \([N_t, (i_{jt}, \hat{S}_{jt})_{j=1}^{N_t}]\) at each time step.

- Unlike the erasure case, in general the anytime decoder will sometimes give a new corrected estimate for a previously estimated bit. It even sometimes corrects the corrections.
Unstable Source/Channel Separation Theorem

*Direct:* If there is an $\epsilon > 0$ for which the channel has

$$C_{\text{anytime}}(2 \log_2 a + \epsilon) > \log_2 a$$

then a scalar Markov process with parameter $a$ driven by bounded noise can be tracked with finite expected squared-error across the channel.

*Converse:* If a scalar Markov process with parameter $a > 1$ driven by bounded noise can be tracked with finite expected squared-error across a noisy channel, then for all $\epsilon > 0$ the channel has

$$C_{\text{anytime}}(2 \log_2 a) > \log_2 a - \epsilon$$

- The direct part also shows that it is possible to track in the mean-squared sense by appropriately cascading a source code with an anytime channel code.

- This theorem can be thought of as a separation theorem with an explicit and fundamental notion of “Quality of Service” given by the parameter $a$.

- Faster dynamics (larger $a$) need more than just an increased bit-rate, they need better QoS $a$. 
No Differentiated Service

- Treat all bits alike for reliable transmission. Everyone gets the same $\alpha$
- Combined rate $R \geq (\log_2(1.258) + 4\log_2(1.058))$ for finite mean-squared error on each component
- At that $R$, the best $\alpha \leq 0.646 < 2\log_2(1.258)$ This is too slow for the fast component!

$$
\sup_{t>0} E \left[ \| X_t - \hat{X}_t \|^2 \right] = \infty
$$
regardless of end-to-end delay $d$ or choice of source encoders!
- Encode each scalar source separately using a causal code with $R_1 = \frac{1}{3} > \log_2(1.258)$ and $R_{2,3,4,5} = \frac{1}{12} > \log_2(1.058)$. (Combined $R = \frac{2}{3}$)

- Discriminate between the bitstreams during reliable transmission. Give “high priority” to the fast source and “low priority” to the slow ones.
Causal Recursive Scalar Source Code

Window around $\hat{X}_t$ known to contain $X_t$

grows by a factor of $a > 1$ because of the dynamics and
also by constant from driving noise $|W_t| \leq \frac{Q}{2}$

giving rise to a larger window of uncertainty regarding $X_{t+1}$

send $n$ bits and cut decoder’s window by a factor of $2^{-n}$
giving a new window around the updated estimate $\hat{X}_{t+1}$

- Let $S_i$ be the $i$-th set of $n$ bits coming from the source code. Interpret them as an integer between $-2^{n-1}$ and $2^{n-1}$. 

$$\hat{X}_t = \sum_{i=1}^t \frac{\Delta_i}{2^n} S_i a^{t-i}$$

- As long as rate $R > \log_2 a$, we can keep $\Delta_t \leq M(R) < \infty$ bounded for all $t > 0$

- Over noiseless digital channels:

$$\sup_{t>0} E \left[ (X_t - \hat{X}_t)^2 \right] \leq (M(R))^2 < \infty$$
Priority Channel Encoding

- Input bitstreams have deterministic timing and feedback is instantaneous, so encoder and decoder are always in sync.
- “High priority” stream pre-empts the “Low priority” ones.
- “Low priority” streams are treated fairly among each other.
- Infinite buffers.
Evaluating High Priority $\alpha$

• Since it preempts everything, it is effectively alone
• Setup Markov chain for input queue state:

\[
\begin{align*}
    p_{0,0} &= 3e^2(1 - e) + 3e(1 - e)^2 + (1 - e)^3 \\
    p_{i,i+1} &= e^3 \\
    p_{i,i} &= 3e^2(1 - e) \\
    p_{i,i-1} &= \begin{cases} 
        3e(1 - e)^2 + (1 - e)^3 & \text{if } i = 1 \\
        3e(1 - e)^2 & \text{if } i > 1 
    \end{cases} \\
    p_{i,i-2} &= (1 - e)^3
\end{align*}
\]

• Evaluating asymptotics gives:

\[
\begin{align*}
    P_{\text{error}}(\text{Delay} = d) &\leq P(\text{Buffer State} > dR_1) \\
    &\leq K \left( \frac{2e^3}{1 + 2e^3 + (1 - e)\sqrt{1 + 2e - 3e^2 - 3e^2}} \right)^{\frac{d}{3}} \\
    &= K2^{-\frac{1}{3}\log_2(\frac{1+2e^3+(1-e)\sqrt{1+2e-3e^2-3e^2}}{2e^3})d}
\end{align*}
\]

which for $e = 0.27$ results in a fast enough

\[
\alpha \approx 1.799 > 2 \log_2(1.258) \approx 0.662
\]
Evaluating Low Priority $\alpha$

- The combined size of all the buffers is the same as it would be for a single $R = \frac{2}{3}$ system. Setup Markov chain:

$$
p_{0,0} = 3e(1 - e)^2 + (1 - e)^3
$$

$$
p_{i,i+2} = e^3
$$

$$
p_{i,i+1} = 3e^2(1 - e)
$$

$$
p_{i,i} = 3e(1 - e)^2
$$

$$
p_{i,i-1} = (1 - e)^3
$$

- Calculate steady state distributions and realize that for large combined buffer lengths, the dominant term comes from the low priority ones.

- Evaluating:

$$
P_{\text{error}}(\text{Delay} = d) \leq P(\text{Combined Length} > d(R_2 + R_3 + R_4 + R_5))
$$

$$
\leq K \left( \frac{2e^2}{2e^2 + \sqrt{4e - 3e^2} - 3e} \right)^{\frac{d}{3}}
$$

$$
= K2^{-\frac{1}{3} \log_2(\frac{2e^2 + \sqrt{4e - 3e^2} - 3e}{2e^2})}d
$$

which for $e = 0.27$ results in

$$
\alpha \approx 0.285 > 2 \log_2(1.058) \approx 0.163
$$

so our scheme is fast enough for all the slow components as well!
Conclusions and Further Work

• We had to make no *a-priori* assumptions on delay and chose a standard additive difference distortion measure

• Everything was classical except the source

• We have a simple vector source example for which regardless of the end-to-end delay bound we allow, the expected mean-squared-error will be infinite if we do not differentiate between bitstreams

• If we differentiate and use a priority-based system, the expected mean-squared-error is finite

• QoS can be studied in Information Theoretic “Asymptopia”

• Is this just a toy? Or are unstable autoregressive processes prototypical for “real-time” data streams?

My forthcoming Ph.D. dissertation covers all this material in more depth. For more information, contact: sahai@lids.mit.edu
Direct Part Proof

- Just cascade the source code with the anytime channel code as follows:

\[ \begin{align*}
E \left[ (\hat{X}_t - X_t)^2 \right] & \leq \sum_{n=0}^{\infty} Ma^{2n} P(\hat{S}_{t-n-d} \neq S_{t-n-d}) \\
& \leq D_0 + 2^{-(2\log_2 a+\epsilon)d} \sum_{n=0}^{\infty} Ma^{2n} K 2^{-(2\log_2 a+\epsilon)n} \\
& = D_0 + 2^{-(2\log_2 a+\epsilon)d} \sum_{n=0}^{\infty} MK 2^{-\epsilon n} \\
& = D_0 + \frac{MK 2^{-(2\log_2 a+\epsilon)d}}{1 - 2^{-\epsilon}} < \infty
\end{align*} \]
Converse Part

- Idea: Use the joint source/channel encoder together with a simulated scalar Markov source driven by the bits to be transported rather than random bounded noise.

- Intuitively, we are able to use non-ergodic nature of the source (ie. it does not “forget” the past) to our advantage.

- Technically, we use a Cantor-set like construction to embed our bits into the real-valued source. They are extracted by using a modified A/D conversion procedure.