Shannon revisited: New separation principles for wireless multimedia

Prof. Mihaela van der Schaar
Multimedia Communications and Systems Lab
Electrical Engineering Department, UCLA
http://medianetlab.ee.ucla.edu/
UCLA Multimedia communications Lab

Delay-critical systems and networks
- Multimedia compression & proc
- Rigorous cross-layer design
- Mission-critical networks & systems
- Energy-efficient multimedia sys
- Real-time stream mining

Designs for networked communities
- New classes of games & learning
- Network economics
- Policing in networks and systems
- Design and incentives in Social Nets
Delay-critical networking and processing

Multimedia compression, processing and networking

Rigorous methods for cross-layer design (dynamic environments)

MPEG, Philips

NSF, Intel, HP, Microsoft

Delay-critical Networking and Online Learning

Real-time Stream Mining

NSF, ONR, Intel, Cisco

IBM, NSF
Goal: Designing Large-Scale Multimedia Mining Applications in Distributed Processing Environments [NSF, IBM]

Challenges:

• **High Volume of data**: faster than a database can handle

• **Complex Analytics**: correlation from multiple sources and/or signals; video, audio or other non-relational data types

• **Delay-critical**: responses required in a specified time

• **Other system requirements**:
  – Scalable to the number of flows
  – Resource variability
  – Failure Tolerance
    • Data cannot be stored and reprocessed
    • Requirements on graceful degradation under failure
  – Distributed computation by self-interested agents
Stream Computing: New Paradigm

**Traditional Computing**

**Historical fact finding with data-at-rest**
- Batch paradigm, pull model
- Query-driven: submits queries to static data
- Relies on Databases, Data Warehouses

**Stream Computing**

**Real time analysis of data-in-motion**

**Streaming data**
- Stream of structured or unstructured data-in-motion

**Stream Computing**
- Analytic operations on streaming data in real-time
Stream mining - Semantic concept detection

Smarter cities

Aerial Recon. Images

Input Stream

Dynamic Stream

Ground Recon. Images

Distributed, Real-time Stream Processing

Operating System and Transport

Hardware Configuration

Streams Middleware

Resource-Adaptive Analytic Placement, Optimization

Taxonomy

Urban

Gathering

Road

Convoy

Roadside Bomb

Flag-burning

Protest

Unknown

Bagging Models

Intelligence Analysts

Scenes and Activities

Aerial Recon. Images

Ground Recon. Images

Protest

Road

Convoy
Stream mining - Online Healthcare Monitoring

- Contextual Data Sources
  - Biometric Sensor Data
  - Distributed, Real-time Stream Processing

- Clinical, Insurance
  - Proactive Outbreak Detection
  - Trending Analysis
  - Clinical Decision

- Wellness, Citizen
  - Wellness Services
  - Third Party Consulting
  - Self Management

- Census, CDC
  - Realtime Health Census
Stream mining - Analysis for social networks

- Graph = nodes (people, e.g. bloggers) + links (interactions)
  - Each node includes a temporal sequence of ‘documents’ (blog posts, tweets, …)

1. Identify relevant content
   **Now:** keyword search

2a. Identify key influencers
   **Now:** page rank, SNA measures, …

2b. Characterize viral potential
   **Now:** use of follower statistics

3. Characterize objective vs subjective content
   **Now:** lexical and pattern-based models

4a. Topic evolution & emergence
   **Now:** word co-occurrence, clustering

4b. Classify new partially-observed documents
   **Now:** unsupervised clustering

Distributed, Real-time Stream Processing
Multi-disciplinary research effort

- Parallel and Grid Computing
  - High volume data stream processing
- Content-level routing, Topology formation and Event messaging
- Signal Processing
  - Real-time adaptive analytics
    - Stream data aggregation, filtering, compression, processing
  - Incremental learning
  - Cross-layer design
    - System and Analytics
- Distributed system designs for autonomous and self-interested agents
Information processing and economics

New Classes of Engineering Games
Network economics

Policing in networks (Intervention)
Design and Incentives in Networked Communities

NSF, IBM
Shannon revisited: New separation principles for wireless multimedia


• F. Fu and M. van der Schaar, "Structure-Aware Stochastic Control for Transmission Scheduling".

• F. Fu and M. van der Schaar, "Structural Solutions to Dynamic Scheduling for Multimedia Transmission in Unknown Wireless Environments".
Our research focus

• To develop a rigorous mathematical framework that enables us to analyze and design multi-user environments and applications, where autonomous users interfere with each other when sharing a common set of resources.

• Aim: construct a new theory for architecting next-generation distributed networks and systems under informational and/or delay-constraints.

[NSF Career, 2004]
Networked multimedia apps are booming!

- Existing network environments provide limited support for delay-sensitive applications
- My research has been dedicated in the past 14 years to enabling efficient real-time multimedia communication
- **What’s new?**
  Development of a rigorous, unified framework for the optimal design and deployment of delay-sensitive multimedia communication
Challenges

**Challenge 1:** Unknown, dynamic environments

- Dynamic source and channel conditions
- Statistical knowledge of dynamics - unknown
**Challenge 2**: Multimedia traffic is highly heterogeneous

- Different delay deadlines, importance, and dependencies
Challenges

Original VIDEO

Packet-based Network

Source

Traffic

Transmitter

Encoder Buffer

Transmission Strategy

Receiver

Decoder Buffer

Receiving Strategy

Challenge 3: Multi-user coupling/interference (dynamic!)
Problem 1: Minimize average delay for *homogeneous traffic* in dynamic networks

Existing solutions – *Network control theory*

  - Queue is stable, but delay performance is suboptimal (for low delay apps)
  - Network environment is considered known – not true in practice!

Existing solutions – *Stochastic control theory*

- Markov decision process (MDP) formulation [Berry 2002, Borkar 2007, Krishnamurthy 2006]
  - Statistic knowledge of the underlying dynamics is required
- Online learning [Krishnamurthy 2007, Borkar 2008]
  - Slow convergence and large memory requirement
Problem 2: Maximize quality of delay-sensitive applications with *heterogeneous traffic*

Existing solutions - *Multimedia communication and networking*

- Joint source and channel coding, scheduling, prioritization etc.
- Rate distortion optimization (RaDiO) [Chou, 2001, Frossard 2006, Girod 2006, Ortega 2009]

- **Limitations**
  - Myopic optimization
  - Only explores heterogeneity of the media data, but do not consider network dynamics and resource constraints
  - Only known environments considered
  - Linear transmission cost (e.g. not suitable for energy-optimization)
  - No systematic solutions available - problem seemed too hard?
Problem 3: *Multi-user* transmission sharing of network resources

Existing solutions - *Network optimization theory*

- Network utility maximization (NUM) [Chiang 2007, Katsaggelos 2008]
  - Uses static utility function without considering the network dynamics
  - No delay guarantees
  - No learning ability in unknown environments

Existing solutions – *Network control theory*

  - Queue is stable, but delay performance is suboptimal (for low-delay apps)
  - Does not consider heterogeneous media data
What can we learn from Shannon about how to address these challenges?

Shannon’s separation principle:
- Source code with minimal rate to satisfy desired source distortion
- Channel code with minimal rate to reduce channel errors

Despite separate design of source and channel codecs, optimality is achieved!
Separation principles

Shannon’s separation principle is not useful for delay-critical communication designs, because it is valid only for:

- Stationary source and channel
- Unlimited delay (arbitrary long block codes)
- Point-to-point communication systems

However, the idea of “separation principles” will become useful if we can separate at the right places!
Thus: new separation principles are needed!

This work:
- develops designs that separate into the “right” sub-problems
- efficiently solves the separated sub-problems
<table>
<thead>
<tr>
<th><strong>Challenges</strong></th>
<th><strong>Solutions</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic system</td>
<td>Stochastic control</td>
</tr>
<tr>
<td>Unknown dynamics</td>
<td>Online learning</td>
</tr>
<tr>
<td>Learning efficiency</td>
<td><strong>Separation principles</strong></td>
</tr>
<tr>
<td>Heterogeneity</td>
<td></td>
</tr>
<tr>
<td>Multi-user coupling</td>
<td></td>
</tr>
</tbody>
</table>
## Key results

<table>
<thead>
<tr>
<th>Previous state-of-art methods</th>
<th>Our improvements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy-efficient data transmission*</td>
<td>Stability constrained optimization [Neely 2006]</td>
</tr>
<tr>
<td>Wireless video transmission</td>
<td>Rate-distortion optimization [Chou 2001]</td>
</tr>
<tr>
<td>Multi-user video transmission</td>
<td>Network utility maximization [Chiang 2007]</td>
</tr>
</tbody>
</table>

*minimize the average delay
Roadmap

- **Separation principle 1**: separate the foresighted decision problem from the computation of unknown dynamics

- **Separation principle 2**: separate the foresighted decision problems across data packets

- **Separation principle 3**: separate the foresighted decision problems across users
Energy-efficient data transmission

- Point-to-point time-slotted communication system
- System variables
  - Backlog (queue length): $x_t$
  - Channel state: $h_t$ Finite state Markov chain
  - Data arrival process: $a_t$: i.i.d.
- Decision at each time slot
  - Amount of data to transmit (transmission rate): $y_t$, $0 \leq y_t \leq x_t$
- Energy consumption: $\rho(h_t, y_t)$, convex in $y_t$, e.g. $\rho_t(h_t, y_t) = \sigma^2 \frac{2^y - 1}{h_t}$.

**Assumption 1:** $u(x, y)$ is bounded, supermodular and jointly concave in $x, y$;  
**Assumption 2:** $\rho(h, y)$ is increasing and convex in $y$ for any given $h \in \mathcal{H}$. 
Foresighted optimization formulation

- Why we need foresighted optimization?
- Markov Decision Processes (MDP)
- Online learning

\[
\max_y \{ u(s, y) \} + \mathbb{E}_w V(f(s, y, w))
\]

Current utility

State-value function

Queue length
Channel condition
Heterogeneity

State: \( s \)

Action: \( y \)

State: \( s' = f(s, y, w) \)

Current time slot
Next time slot

Dynamics: \( w \)
Foresighted optimization formulation

- Foresighted optimization (MDP) formulation
  - State: \( (x_t, h_t) \)
  - Action: \( y_t \)
  - Policy: \( \pi: (x_t, h_t) \rightarrow y_t \)
  - Utility function: \( u(x_t, h_t, y_t) = -(x_t - y_t + \lambda \rho(h_t, y_t)) \).

- Objective (optimize delay and energy consumption tradeoff)

\[
V(x_t, h_t) = \max_\pi \mathbb{E} \sum_{k=t}^{\infty} \alpha^{(k-t)} \{ u(x_k, h_k, \pi(x_k, h_k)) \}
\]

\( \alpha \in [0, 1) \) is discount factor.

- If dynamics are known – solution for Bellman’s equations is known
  - Policy iteration, value iteration
Challenges for solving the foresighted optimization

**Bellman’s equation:**

\[
V(x, h) = \max_{\pi} \{ u(x, h, \pi(x, h)) + \alpha \mathbb{E}_{a, h'} | h V(x - \pi(x, h) + a, h') \}
\]

- Statistical knowledge of the underlying dynamics - **unknown**
  - Unknown traffic characteristics
  - Unknown channel (network) dynamics
- Coupling between the maximization and expectation
Exploit nature of foresighted decision - Separation Principle 1

Post-decision state separates foresighted decision from dynamics.

State-value function

\[ V(x_t, h_t) \]

Decision \( y_t \)

Exogenous dynamics

\[ V(x_t, h_t) \xleftarrow{\text{Foresighted decision}} U(\tilde{x}_t, h_t) \xrightarrow{\text{Post-decision state-value function}} V(x_{t+1}, h_{t+1}) \]

Foresighted decision

\[ V(x, h) = \max_y \{ u(x, h, y) + \alpha U(x - y, h) \} \]

Expectation over dynamics

\[ U(x, h) = \mathbb{E}_{a, h' | h} V(x + a, h') \]

Post-decision state separates foresighted decision from dynamics.
Online learning – using Separation Principle 1

\[ U(x, h) = \mathbb{E}_{a, h' \mid h} V(x + a, h') \]

\[ V(x, h) = \max_y \{ u(x, h, y) + \alpha U(x - y, h) \} \]

- Online adaptation

\[ U_t(x, h_{t-1}) = (1 - \beta_t) U_{t-1}(x, h_{t-1}) + \beta_t V_t(x, h_t) \quad \text{e.g. } \beta_t = 1/t \]

Online update \quad Time-average

\[ \pi, V \quad U \]

Foresighted decision

\[ V_t(x, h_t) = \max_{y \in \mathcal{Y}} \{ u(x, h_t, y) + \alpha U_{t-1}(x - y, h_t) \} \]

**Theorem:**
Online adaptation converges to the optimal solution when \( t \to \infty \)

Expectation is independent of backlog \( x \to \text{batch update} \) (fast convergence).

Batch update incurs high complexity. Curse of dimensionality 😞
Structural properties of optimal solution

\[
U(x, h) = \mathbb{E}_{a, h' \mid h} V(x + a, h')
\]
\[
V(x, h) = \max_y \{ u(x, h, y) + \alpha U(x - y, h) \}
\]

- Structural properties of optimal solution
  - Assumption: \( u(x, h, y) \) is jointly concave and supermodular* in \((x, y)\).

\[\pi(x, h) \text{ is monotonic in } x \]
\[\pi, V \]
\[U \]
\[U(x, h) \text{ is concave in } x \]

Foresighted decision

How can we utilize these structural properties in online learning?

\[u(x', h, y') - u(x', h, y) \geq u(x, h, y') - u(x, h, y) \text{ if } x' \geq x, y' \geq y\]
For each channel state $h$, we approximate the post-decision state-value function such that the worst-case performance degradation is bounded and performance-complexity tradeoffs can be easily performed.
Online learning with adaptive approximation

\[ \hat{U}_t(x, h_{t-1}) = A_\delta (1 - \beta_t) \hat{U}_{t-1}(x, h_{t-1}) + \beta_t V_t(x, h_t) \]

\[ V_t(x, h_t) = \max_{y \in \mathcal{Y}} \{ u(x, h_t, y) + \alpha \hat{U}_{t-1}(x - y, h_t) \} \]

**Theorem:** Online learning with adaptive approximation converges to an \( \varepsilon \)-optimal solution, where \( \varepsilon = \frac{\delta}{1 - \alpha} \)

Variant: Update \( U(x, h) \) and \( \pi(x, h) \) every \( T \) time slots
Performance of learning with approximation

Rayleigh fading channel
Average channel gain $\frac{h^2}{\sigma^2} = 0.14$  
$\#channel\ state=8$  
$\alpha = 0.95$
Comparison with stability-constrained optimization

  - Objective - trade-off between Lyapunov drift and energy consumption
    \[
    \min \lambda \rho(h_t, y_t) + (x_t - y_t)^2 - x_t^2
    \]
    Lyapunov drift
  
  \[
  \begin{array}{ll}
  \text{Utility function} & u(x_t, h_t, y_t) \\
  \text{Post-decision state value function} & U(x_t - y_t, h_t)
  \end{array}
  \]

  - Does not consider the channel state transition and the transmission cost
  - Does not consider the effect of the utility function on post-decision state value function
  - Only ensures queue stability, but results in poor delay performance
Comparison with stability-constrained optimization

Stability constrained optimization
Minimize Lyapunov drift $\Rightarrow$ Minimize delay

Our proposed solution
Minimize queue size $=$ Minimize delay
Comparison to Q-learning

- Q-learning: learn directly state-value function (no separation applied)
- Online learning based on Separation Principle 1
Roadmap

• Separation principle 1 – separation of foresighted decision and computation of unknown dynamics

• Separation principle 2 – separation of foresighted decision across data packets for heterogeneous multimedia

• Separation principle 3 – separation of foresighted decision across users for multi-user communication
Heterogeneous media data

Media data representation:

- Each DU has the following attributes:
  - Arrival time: time at which the DU is ready for processing: $t_i$
  - Delay deadline: $d_i$
  - Size: $L_i$
  - Distortion impact: $q_i$ per packet
- Interdependency between DUs: *Augmented Directed Acyclic Graph (A-DAG)*

- Utility function: video quality (PSNR) vs. energy tradeoff
Context

- Context \((c_t)\) at each time slot \(t\)
  - Include the DUs whose deadlines are within a time window \(W\)
    e.g. \(W = 3\)
Foresighted optimization

State: \( (c_t, x_t, h_t) \) \( x_t = (x^2_t, x^3_t, x^4_t, x^5_t) \)

- Multi-DU Foresighted decision

\[
\begin{align*}
\max_{y_t, i \in c_t} \{ u(c_t, x_t, h_t, y_t) + \alpha U(c_t, x_t - y_t, h_t) \} \\
\text{Current utility} & \quad \text{Post-decision state-value function}
\end{align*}
\]

where \( u(c_t, x_t, h_t, y_t) = -\sum_{i \in c_t} q_i y_i - \lambda \rho(h_t, \sum_{i \in c_t} y_i) \)

- Which DU should be transmitted first?
- How much data should be transmitted for each DU?
Priority-based scheduling

- Prioritization
  - Based on distortion impacts, delay deadlines and dependencies
Separation principle 2:
Separate foresighted decision across DUs

- Theorems
  - If there is only one DU with the highest priority, it is optimal to transmit the data in this DU by solving the foresighted optimization.
  - If there are multiple DUs with the same priorities, it is optimal to first solve the foresighted optimization for each DU and transmit the data from the DU with highest long-term utility.

Single-DU foresighted decision:

\[ V_t^i = \max_{y_t^i \in \mathcal{Y}(h_t)} \left\{ \tilde{u}_i(x_t, h_t, \sum_{j \triangleleft i} y_t^j, y_t^i) + \alpha U_i(c_t, x_t - y_t^i, h_t) \right\} \]

\[ j \triangleleft i : \text{DU } j \text{ has higher priority than DU } i. \]

One dimensional concave function given \( c_t \) and \( h_t \).
It can be updated using the proposed online learning.

Multi-DU foresighted decision → Multiple single-DU foresighted decision
Separation principle 2: Separate foresighted decision across DUs

Multi-DU foresighted decision → Multiple single-DU foresighted decision
Simulation results for video transmission

- Dynamics of both source and channel considered (separation principle 2)
- Dynamics of only source considered
- No dynamics considered (myopic)

[Graph showing PSNR vs. consumed energy for different solutions]

Foreman
Roadmap

• Separation principle 1 – separation of foresighted decision and computation of unknown dynamics

• Separation principle 2 – separation of foresighted decision across data packets for heterogeneous multimedia

• Separation principle 3 – separation of foresighted decision across users for multi-user communication
Delay-sensitive multi-access communication

**Goal:** maximize sum of long-term utilities across all users

\[
\max_{y_t, \forall t} \mathbb{E} \sum_{t=0}^{\infty} \sum_{i=1}^{M} \alpha^t \sum_{i=1}^{M} u_i(x_t^i, h_t^i, y_t^i)
\]

s.t. \([y_t^1, \ldots, y_t^M] \in \Pi(h_t), \forall t \geq 0\)

Resource constraint
(e.g. transmission time constraint in TDMA)
Foresighted optimization formulation

- Formulate as Multi-user MDP (MUMDP) and perform foresighted decision

\[ V(x_t, h_t) = \max_{y_t} \left\{ \sum_{i=1}^{M} u_i(x_i^t, h_i^t, y_i^t) + \alpha U(x_t - y_t, h_t) \right\} \]

Goal: decouple post-decision state value function across users
Separation Principle 3: Decomposition of post-decision state-value function

- Introduce scalar resource price $\lambda$, and compute post-decision state-value function $U(x_t, h_t; x_{t+1}, h_{t+1})$ individually, based on single-user MDP:
Separation Principle 3: Decomposition of post-decision state-value function

- Introduce scalar resource price $\lambda$, and compute post-decision state-value function $U_i^\lambda(x_t^i, h_t^i)$ individually, based on single-user MDP:
Multi-user resource allocation

- Resource allocation

\[
\max_{y_t \in \Pi(h_t)} \sum_{i=1}^{M} \left\{ u_i(x_t^i, h_t^i, y_t^i) + \alpha U_i(x_t^i - y_t^i, h_t^i) \right\}
\]

- Sub-gradient method to update resource price

Sub-gradient

User \( i \)

Network coordinator

Resource allocation

Current allocation

Future allocation

Resource price \( \lambda \)
Resource price update

- Subgradient method to update resource price

The resource price is updated by

$$\lambda_{k+1} = [\lambda_k + \beta_k \left( \sum_{i=1}^{M} Z^i - \frac{1}{1 - \alpha} \right)]^+$$

where $Z^i$ is the expected consumed resource by user $i$ and is individually computed by user $i$. 

![Diagram showing network coordinator updating $\lambda$ based on subgradient method with arrows between users and coordinator]
Relationship of our multi-user framework to existing methods

**Longest Queue Highest Possible Rate (LQPHR)** -
- Assign higher rate to longer queue to achieve optimal average delay
- Assume symmetric i.i.d. channels
Results for multi-user transmission

- Each user uses multiple queues to represent video data
- Markov chain model for Rayleigh fading channel
- TDMA-type channel access
Other applications developed in our lab based on the same framework

- **Cross-layer optimization** via layer separation
  - Each layer performs dynamic optimization individually
  - Message exchange across layers
- **Media-TCP:** Congestion control for real-time multimedia transmission
- **Energy-efficient multimedia processing**
- **Wireless video over cooperative, multi-hop and mission-critical networks**
- **Distributed, real-time multimedia data mining applications**
- **Scalable and reconfigurable video coding** – via layer separation

Thank you Claude!
Our research website:

http://medianetlab.ee.ucla.edu