Signal Processing and Coding for Distributed Sensing and Storage

Xiaodong Wang
Electrical Engineering Dept.
Columbia University, New York
Outline

- Distributed spectrum sensing via level-triggered sampling
- Coding and resource allocation for distributed wireless cloud
Uniform Sampling vs. Event-triggered Sampling

**Uniform in-time sampling:**
- sample with period $T$
- deterministic sampling

**Event-triggered sampling:**
- sample whenever an event occurs (e.g., a level is passed)
- dynamic sampling $\rightarrow$ samp. times dictated by the signal (random)
Cooperative Spectrum Sensing

Decision mechanism: fixed sample size / sequential SPRT

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Cooperative Spectrum Sensing

- Decision mechanism: fixed sample size / sequential
- Yilmaz, Moustakides, Wang
- Spectrum Sensing via Event-triggered Sampling
Cooperative Spectrum Sensing

- Decision mechanism: fixed sample size / sequential SPRT

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
Cooperative Spectrum Sensing

Decision mechanism: fixed sample size / sequential

Yilmaz, Moustakides, Wang
Cooperative Spectrum Sensing

Decision mechanism: fixed sample size / sequential
Cooperative Spectrum Sensing

Decision mechanism: fixed sample size / sequential

Yilmaz, Moustakides, Wang
Decision mechanism: fixed sample size / sequential SPRT
Motivation

- Observation message sampling times
- Global clock (synchronous communication)

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
Motivation

- Observation times: \( t_1, t_2, t_3, t_4 \)
- Event-triggered sampling

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
Motivation

- SU_1
- SU_2
- SU_K
- FC

- Observation

Sampling times: t_1, t_2, t_3, t_4

Global clock (synchronous communication)

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Motivation

- Observation
- Sampling times

SU_1 → FC
SU_2 → FC

SU_K → FC

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
Motivation

- FC
- SU_1
- SU_2
- SU_K

Sampling times

\[ t_1, t_2, t_3, t_4 \]

Observation

Message

Global clock (synchronous communication)

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Motivation

Observation sampling times $t_1$, $t_2$, $t_3$, $t_4$ and dynamic sampling (asynchronous communication)

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Motivation

- $SU_1$
- $SU_2$
- $SU_K$
- $FC$

Observation:
- $t_1$
- $t_2$
- $t_3$
- $t_4$

Dynamic sampling (asynchronous communication)

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Motivation

- Observation sampling times
- Dynamic sampling (asynchronous communication)

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
Motivation

- FC
- SU_1
- SU_2
- SU_K

observation

t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4

dynamic sampling (asynchronous communication)

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
Motivation

Observation sampling times:

- $t_1^1$ to $t_1^4$
- $t_2^1$ to $t_2^4$
- $t_3^1$ to $t_3^4$
- $t_4^1$ to $t_4^4$

Dynamic sampling (asynchronous communication)

$SU_1$ to $SU_K$ to $FC$
Having observations \( \{y^k_1, \ldots, y^k_t\}_{k=1}^K \) at SUs, we perform the following hypothesis test,

\[
H_0 : \{y^k_1, \ldots, y^k_t\} \sim f_0, \ k = 1, \ldots, K \\
H_1 : \{y^k_1, \ldots, y^k_t\} \sim f_1, \ k = 1, \ldots, K
\]  \tag{4}

Each SU computes its own log-likelihood ratio (LLR) and sends it to the FC.

\[
L^k_t \triangleq \log \frac{f_1(y^k_1, \ldots, y^k_t)}{f_0(y^k_1, \ldots, y^k_t)} = \sum_{n=1}^{t} \log \frac{f_1(y^k_n)}{f_0(y^k_n)} = L^k_{t-1} + \ell^k_t
\]  \tag{5}

FC computes the global LLR, \( L_t = \sum_{k=1}^{K} L^k_t \), and applies SPRT to make a sensing decision.

**SPRT**

\[
S = \inf \{ t > 0 : L_t \notin (-B, A) \},
\]

\[
\delta(S) = \begin{cases} 
1, & \text{if } L_S \geq A, \\
0, & \text{if } L_S \leq -B.
\end{cases}
\]  \tag{7}
Thresholds $A, B$ are selected so that SPRT satisfies following constraints with equality.

$$P_0(\delta_S = 1) \leq \alpha \quad \text{and} \quad P_1(\delta_S = 0) \leq \beta \quad (8)$$

There are two serious practical weaknesses of SPRT in our problem.

- Local LLRs must be sent to the FC at Nyquist-rate.
- Infinite number of bits is required to represent local LLRs.

Substantial communication overhead is incurred between SUs and FC!

**Objective**

Decentralized schemes $\equiv$ low rate info. transmission from SUs to FC
Decentralized Q-SPRT Scheme

Secondary Users

- sample local LLRs uniformly at time instants $T, 2T, \ldots, mT, \ldots$
- quantize sampled values using a finite number of levels, $\tilde{r}$ (e.g. uniform mid-riser quantizer)

\[
\tilde{\lambda}^k_{mT} \rightarrow FC \text{ using } \log_2 \tilde{r} \text{ bits}
\]
Decentralized Q-SPRT Scheme

Fusion Center

- **synchronously** receives quantized info. from SUs
- updates the approximation of the global running LLR

\[
\tilde{L}_{mT} = \tilde{L}_{(m-1)T} + \sum_{k=1}^{K} \tilde{\lambda}_{mT}^k
\]  

applying the SPRT idea using \(\tilde{L}_{mT}\) and \(\tilde{A}, -\tilde{B}\) as the two thresholds.
Secondary Users

- sample local LLR process $L_t^k$ at a sequence of random times $\{t_n^k\}$

- send the information of the threshold that is crossed by $\lambda_n^k = L_{t_n^k}^k - L_{t_n^{k-1}}^k$ to FC (either $\Delta$ or $-\Delta$)

$$b_n^k = \text{sign}(\lambda_n^k) \quad (10)$$

Remark

Each SU performs a local SPRT with thresholds $\Delta$ and $-\Delta$. 

Yilmaz, Moustakides, Wang  
Spectrum Sensing via Event-triggered Sampling
Decentralized Scheme based on Event-triggered Sampling

Fusion Center

- approximates the local incremental LLR as $\hat{\lambda}_n^k = b_n^k \Delta$.

\[
\hat{L}_{tn}^k = \sum_{j=1}^{n} \hat{\lambda}_j^k = \hat{L}_{tn-1}^k + b_n^k \Delta = \sum_{j=1}^{n} b_j^k \Delta
\]  

(11)

Remark

If $\lambda_n^k$ hits exactly one of the boundaries $\pm \Delta$, then we have exact recovery ($\hat{L}_{tn}^k = L_{tn}^k$).

- adds all the received bits transmitted by all SUs up to time $t$ and then normalizes the result with $\Delta$.

\[
\hat{L}_t = \sum_{k=1}^{K} \hat{L}_t^k = \Delta \sum_{k=1}^{K} \sum_{n:t_n^k \leq t} b_n^k
\]  

(12)

- applies the SPRT idea using $\hat{L}_t$ and $\hat{A} = -\hat{B}$ as the two thresholds.
Enhancement: Overshoot Quantization at SUs

A very important source of performance degradation: $L_t^k - \hat{L}_t^k$

**Idea**

use additional bits to quantize over(under)shoots, $q_n^k \triangleq |\lambda_n^k| - \Delta$.

- Divide $[0, \phi]$ uniformly into $\hat{r}$ subintervals.
- Transmit either lower or upper end of the corresponding subinterval by random selection.

![Diagram showing quantization with prob. 1 - p and prob. p](image)

- FC at time $t_n$ receives $(b_n, \hat{q}_n)$, and updates its approx. running LLR.

$$\hat{L}_{t_n} = \hat{L}_{t_{n-1}} + b_n(\Delta + \hat{q}_n) \quad (13)$$

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
**Definition**

Any sequential scheme \((\mathcal{T}, \delta_T)\) satisfying the error prob. bounds as \(\alpha, \beta \to 0\), is said to be **order-1 asymptotically optimal** if

\[
1 \leq \frac{E_i[\mathcal{T}]}{E_i[S]} = 1 + o_{\alpha,\beta}(1);
\]  

and **order-2 asymptotically optimal** if

\[
0 \leq E_i[\mathcal{T}] - E_i[S] = O(1)
\]

where \((S, \delta_S)\) is the optimum SPRT.

Cont.-time results:

- Q-SPRT is **not** even order-1 asymp. optimal with **any fixed** number of bits.
- RLT-SPRT is **order-2** asymp. optimal with only **one** bit.
In order to make fair comparisons, $\Delta$ is adjusted so that average frequency of received messages by the FC is the same for Q-SPRT and RLT-SPRT.

- RLT-SPRT needs significantly less bits than Q-SPRT in order to enjoy order-2 asymptotic optimality.
- For fixed $s$, RLT-SPRT achieves order-1 asymptotic optimality when $T \to \infty$ with a rate slower than $|\log \alpha|$.
- In contrast, by controlling $T$, Q-SPRT cannot enjoy any form of asymptotic optimality.
Asymptotic Analysis: Q−SPRT

Average sensing delay

|log α| = |log β|

optimum
1-bit

Spectrum Sensing via Event-triggered Sampling

Yilmaz, Moustakides, Wang
Simulations: Q-SPRT

Asymptotic Analysis: Q-SPRT

Average sensing delay

Yilmaz, Moustakides, Wang  Spectrum Sensing via Event-triggered Sampling
Simulations: Q-SPRT

Asymptotic Analysis: Q−SPRT

Average sensing delay

|\log \alpha| = |\log \beta|

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Simulations: Q-SPRT

Asymptotic Analysis: Q−SPRT

|\log \alpha| = |\log \beta|

Average sensing delay

<table>
<thead>
<tr>
<th>optimum</th>
<th>1-bit</th>
<th>2-bit</th>
<th>3-bit</th>
<th>\infty-bit</th>
</tr>
</thead>
</table>

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Simulations: RLT-SPRT

Asymptotic Analysis: RLT–SPRT

|\log \alpha| = |\log \beta|

Average sensing delay

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Simulations: RLT-SPRT

Asymptotic Analysis: RLT−SPRT

| log α | = | log β |

Average sensing delay

optimum
1-bit
2-bit
Simulations: RLT-SPRT

Asymptotic Analysis: RLT−SPRT

Average sensing delay

|log α|=|log β|

Yilmaz, Moustakides, Wang

Spectrum Sensing via Event-triggered Sampling
Simulations: RLT-SPRT

Asymptotic Analysis: RLT−SPRT

Average sensing delay

\[ |\log \alpha| = |\log \beta| \]

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
Simulations: RLT-SPRT vs. Q-SPRT

Asymptotic Analysis: RLT−SPRT vs. Q−SPRT

|log α| = |log β|

- optimum
- Q 1-bit
- RLT 1-bit
- Q ∞-bit
- RLT ∞-bit

Average sensing delay

Yilmaz, Moustakides, Wang
Spectrum Sensing via Event-triggered Sampling
The proposed decentralized (low-rate transmission) scheme based on non-uniform samplers

- asynchrony among SUs
- order-2 asymp. optimality
  - only 1 bit in cont.-time
  - significantly less number of bits ($-\log_2 T$) than Q-SPRT in disc.-time
- order-1 asymp. optimality using a constant num. of bits when av. comm. period is controlled

Its uniform sampling counterpart (Q-SPRT)

- no optimality using const. num. of bits
- order-2 optimality when num. of bits is allowed to increase at a rate $O(\log |\log \alpha|)$
- no optimality when num. of bits kept constant and av. comm. period changed
Coding for Distributed Storage in Wireless Clouds

Chen Gong

Qualcomm Research San Diego, Qualcomm Inc.

Prof. Xiaodong Wang

Department of Electrical Engineering, Columbia University
Data Centers

- Server clusters that store and process all the data in the Internet
- There were 509,147 data centers worldwide in 2011
- Consume vast amounts of energy - more than 2% of US electricity
  - Power to run and repair servers, and for cooling systems
  - Backup power generators use diesel cause air pollution
- Consequences if a data center breaks down (electricity failure)
Desired Properties of Distributed Storage

- Reliability against disk failures
- Recovery with minimum cost
- Simple updates when data changes
- Easy accessibility without blocking
- Easy failed node repair
  - no data recovery efficiency loss after failed node repair
Trade-offs in Distributed Storage

- Reliability vs. Storage
  - Replication is the most commonly used redundancy
  - \((n, k)\) MDS Codes - any \(k\) out of \(n\) sufficient for data recovery
- Storage vs. Repair Bandwidth
  - Locally Repairable Codes - To restore a failed disk by accessing minimum number of working disks
- Accessibility vs. Storage
  - Coding gives lower blocking probability than replication for the same storage (Energy Cost)
Distributed Wireless Clouds

- Information storage and retrieve in Mobile Wireless Cloud
- Mobile Wireless Cloud without Infrastructure
  - Military communication networks
  - Wireless cloud of Vehicles and Ships
  - Emergency cases: earthquake, tsunami, infrastructures destroyed
- Large amount of data exceeding the infrastructure capacity
- Security reasons, information must be stored locally
Technique Overview

- A file split into several parts,
  - coded symbols across the split parts,
  - stored in various data storage nodes;
- A data collector, reconstruct the original files:
  - via downloading data from the storage nodes
- Download part of the data from each storage node
  - amount of downloaded data depends on wireless link strength
- Orthogonal wireless channels for symbol downloading
Basic Structure for Wireless Storage Networks

- Storage Node 1
- Storage Node 2
- Data Collector
- Storage Node i

- control channel
- data channel
- feedback channel

wireless distributed storage network structure
Failed Node Regeneration

- A failed storage node downloads symbols from other nodes;
- Exactly recover the coded data symbols it stored;
- Similar procedure as that for data reconstruction.
Distributed Storage Modeling for Wireless Cloud

Distributed storage:

• store a file in a distributed manner, *in several nodes*

• two operations:
  - reconstruct the original file
  - repair the storage in a failed node

\((S, K, d, \alpha, \beta)\) *regenerating code:*

• totally \(S\) storage nodes, each storing \(\alpha\) symbols;

• reconstruct original file,
  - via downloading all \(K\alpha\) symbols from *any* \(K\) nodes;

• repair a failed node,
  - via downloading \(\beta\) symbols each from any \(d\) surviving nodes
Data Reconstruction and Node Regeneration

Storage Node 1
\( m^{(1)}, H^{(1)} \)

Storage Node 2
\( m^{(2)}, H^{(2)} \)

Storage Node i
\( m^{(i)}, H^{(i)} \)

Storage Node k
\( m^{(k)}, H^{(k)} \)

Failed Storage Node

Data Collector

Data Reconstruction

K Nodes

\( \alpha \)

\( \beta \)

d Nodes

Node Regeneration
Wireless Distributed Storage Data Coding

Distributed storage setting:

- original file:
  \[ s = [s_1, s_2, \ldots, s_M] \];
- each storage node \( i \) stores:
  \[ m^{(i)} = s^T H^{(i)}; \]

Wireless network setting:

- a data collector (DC)
- \( N \) orthogonal channels
  \[ c(|g_j^{(i)}|^2 P_j) = \frac{WT}{B} \log_2 \left( 1 + \kappa \frac{|g_j^{(i)}|^2 P_j}{\sigma^2} \right) \]

Full downloading and partial downloading:

- full downloading
  - power grows exponentially with the capacity
- partial downloading
Partial Downloading Linear Combination

Partial downloading formulation:

- downloading $\mu_i \leq \alpha$ symbols from storage node $i$
- downloading linear combination $s^T H^{(i)} A^{(i)}$
  - $A^{(i)}_{\alpha \times \mu_i}$ linear combination matrix;
- downloading symbols: $s^T \left[ H^{(i)} A^{(i)} \right]_{i \in S}$
  - $s$ reconstructable iff $\left[ H^{(i)} A^{(i)} \right]_{i \in S}$ of rank $M$

Downloading original symbols:

- **Theorem:** $\left[ H^{(i)} A^{(i)} \right]_{i \in S}$ of rank $M \Rightarrow \left[ \bar{H}^{(i)} \right]_{i \in S}$ of rank $M$,
  - $\exists \bar{H}^{(i)}$, $\alpha \times \mu_i$ submatrix of $H^{(i)}$
- downloading original symbols suffices
  - no need to use linear combination (matrix $A^{(i)}$)
Wireless Cloud Resource Allocation Formulation

Wireless resource allocation:

- data reconstructable: \(\left[\mu_1, \mu_2, \ldots, \mu_S\right]\), \(M \times \mu_i\) submatrix \(\bar{H}^{(i)}\) of \(H^{(i)}\)
  - \(\begin{bmatrix} \bar{H}^{(i)} \end{bmatrix}_{i \in S}\) of rank \(M\)
- \(\beta_j^{(i)} = 1\) if DC downloads from storage node \(i\) using channel \(j\)
  - \(X_j = c\left(P_j \sum_{i \in S} \beta_j^{(i)} |g_j^{(i)}|^2\right)\), \(\mu_i = \sum_{j=1}^{N} \beta_j^{(i)} X_j\) \(\ldots\ldots(1)\)

Problem formulation:

minimize transmission power s.t. data reconstructable
- \(\min \sum_{j=1}^{N} P_j\); s.t. data reconstructable, \((1)\), \(\sum_{i \in S} \beta_j^{(i)} \leq 1\).

Difficulty: how to analyze the data reconstructability
- transform the full rank constraint to ...
Data Reconstructability (MSR)

Full rank constraint:
- transform it using $\mu_i$: downloading $\mu_i$ symbols from node $i$;

Minimum storage regeneration (MSR):
- MSR: $M = K\alpha$, minimum downloading for data reconstruction
  - $\left[ H^{(i)} \right]_{i \in \mathcal{R}}$ of rank $M$ for any $|\mathcal{R}| = K$;
- Simple necessary condition: number of downloaded symbols $\geq M$
  - this is also sufficient
- **Theorem:** For any $\sum_{i \in S} \mu_i \geq M$, $\mu_i \leq \alpha$, there exists $\mu_i \times \alpha$ submatrix $\bar{H}^{(i)}$ of $H^{(i)}$, such that $\left[ \bar{H}^{(i)} \right]_{i \in \mathcal{S}}$ is of rank $M$.
  - keep adding linearly independent symbols, plus some stuck processing
Wireless Cloud Resource Allocation (MSR)

**Relaxed resource allocation problem:**

- \[ \sum_{i \in S} \mu_i \geq M, \mu_i \leq \alpha \Rightarrow \text{remove constraint } \mu_i \leq \alpha; \]
- problem reformulation
  - minimize transmission power, s.t. totally downloading \( M \) symbols;
  - \[ \min \sum_{j=1}^{N} P_j; \text{ s.t. } \sum_{j=1}^{N} X_j = M. \]
- two-step optimal solution
  - each channel \( j \) allocated to the best user, \( \max_i |g_j^{(i)}|^2 \);
  - optimal greedy algorithm for symbol allocation.

**Local adjustment:**

- in case that the constraint \( \mu_i \leq \alpha, i \in S \) violated
- rarely happens in simulation scenarios
Wireless Cloud Resource Allocation Results (MSR)

System setup:
- node number $S = 16$, channel number $N = 64$, noise $\sigma^2 = 0.25$, coefficients $\kappa = 0.5$, $\frac{WT}{B} = 0.25$;
- MSR: $M = 16$, $K = \alpha = 4$;

Partial downloading:
- 1000 channel realization,
- total transmission power
  - mostly more than 0.6dB performance gain over full downloading
Wireless Cloud Resource Allocation (MBR)

Partial downloading:
- more complicated reconstructability condition;
- downloaded symbols: $M(<K\alpha)$ - $K\alpha$ required by full downloading;

Resource allocation:
- relax to $\sum_{i \in S} \mu_i = M$;
- optimal greedy solution + local adjustment (rare);

Results:
- 1000 channel realization,
- total transmission power - mostly around 2.5dB performance gain over full downloading
Performance Comparison with Existing Schemes

Existing schemes - flexible downloading:

- any $\sum_{i \in S} \mu_i \geq M$ symbols suffice data reconstruction;
- needs $\gamma$ symbols for failed node repair;

Bound for failed node repair $\gamma$:

- MSR point $\alpha = 4$, $M = 16$: bound $\gamma \geq 7$, partial downloading $\gamma = 7$;
- MBR point $\alpha = 6$, $M = 18$: bound $\gamma \geq 8$, partial downloading $\gamma = 6$;
  - only $\sum_{i \in S} \mu_i \geq M$ not suffices, but usually suffices for wireless setting

Explicit coding schemes for failed node repair:

- MSR point: flexible downloading $\gamma = 10$, partial downloading $\gamma = 7$;
- MBR point: flexible downloading $\gamma = 12$, partial downloading $\gamma = 6$;