Signal Processing and Coding for Distributed Sensing and Storage

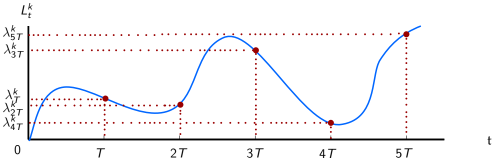
Xiaodong Wang
Electrical Engineering Dept.
Columbia University, New York

Outline

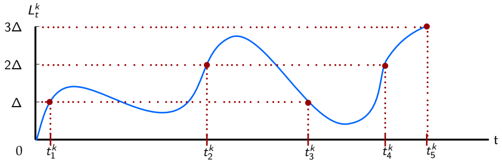
- Distributed spectrum sensing via level-triggered sampling
- Coding and resource allocation for distributed wireless cloud

Uniform Sampling vs. Event-triggered Sampling

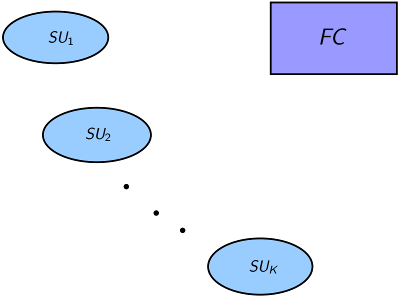
- Uniform in-time sampling:
  - sample with period T
  - deterministic sampling



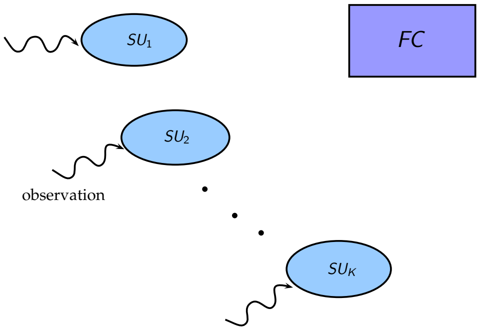
- Event-triggered sampling:
  - sample whenever an event occurs (e.g., a level is passed)
  - dynamic sampling $\rightarrow$ samp. times dictated by the signal (random)



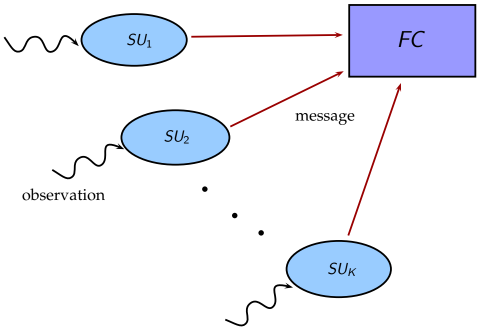
Cooperative Spectrum Sensing



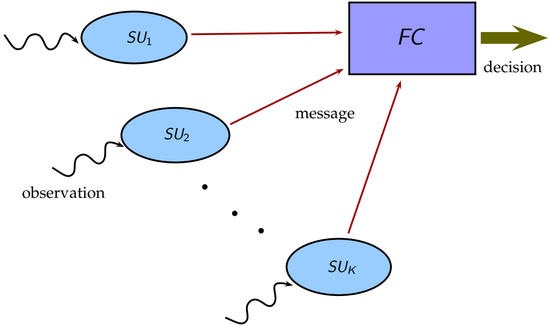
Cooperative Spectrum Sensing



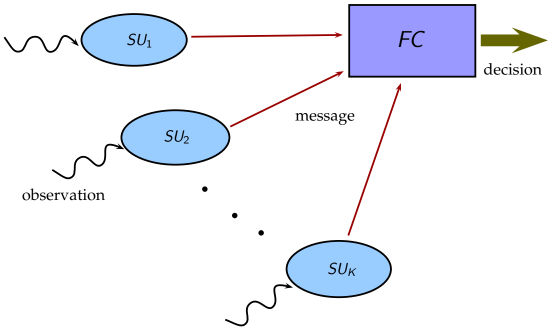
Cooperative Spectrum Sensing



Cooperative Spectrum Sensing

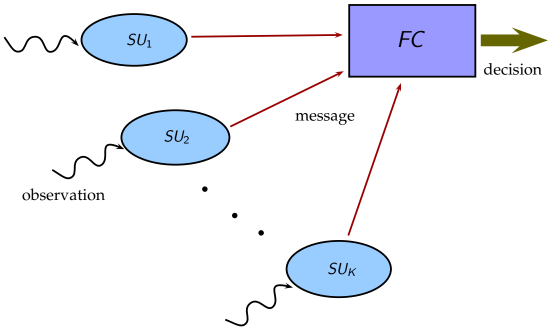


Cooperative Spectrum Sensing



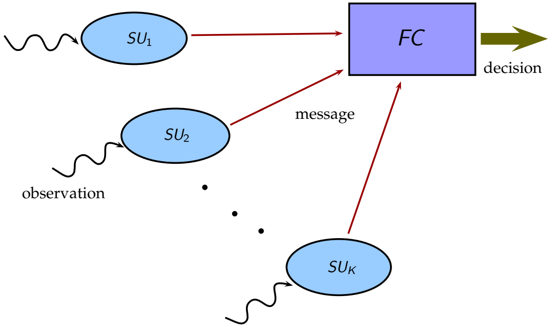
Decision mechanism: fixed sample size / sequential

Cooperative Spectrum Sensing



Decision mechanism: fixed sample size / sequential

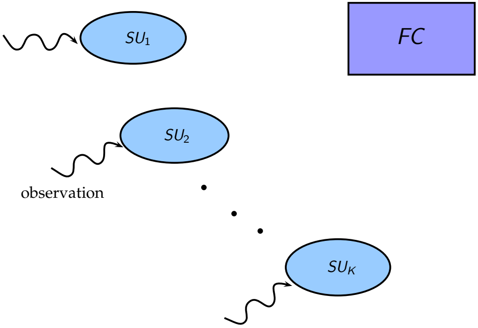
Cooperative Spectrum Sensing



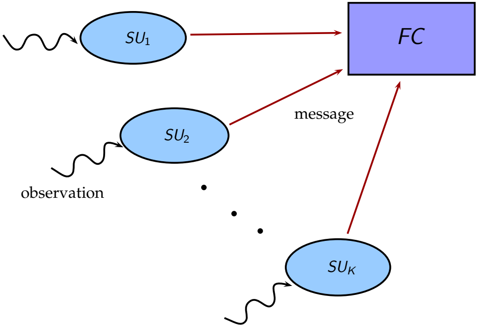
Decision mechanism: fixed sample size / sequential

SPRT

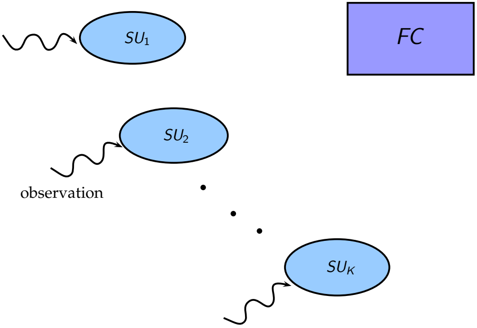
Motivation



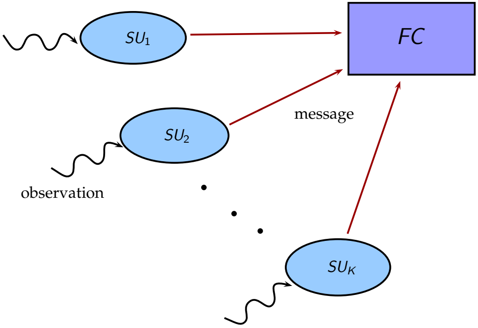
Motivation

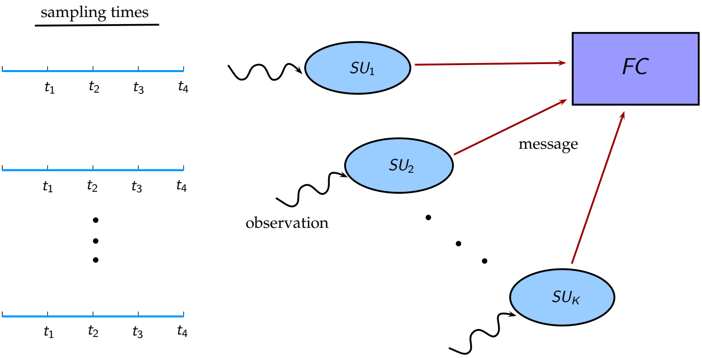


Motivation



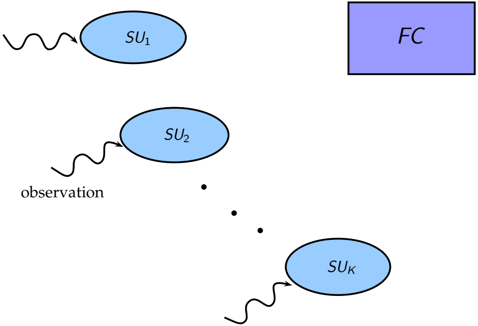
Motivation



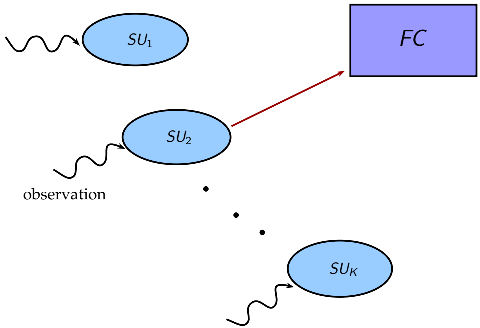


global clock (synchronous communication)

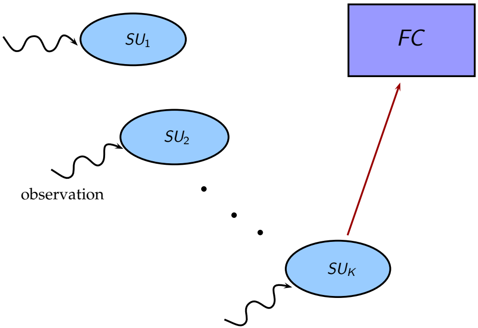
Motivation



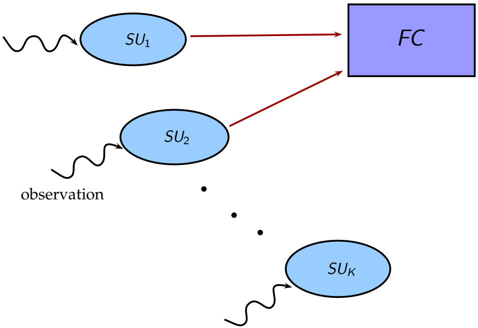
Motivation



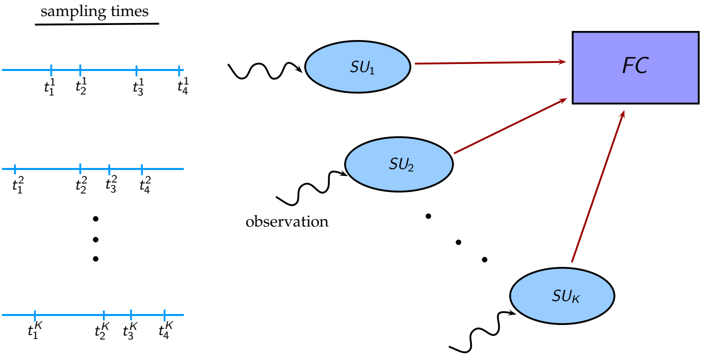
Motivation



Motivation



Motivation



dynamic sampling (asynchronous communication)

Spectrum Sensing via SPRT

Having observations $\{y_1^k, \dots, y_t^k\}_{k=1}^K$ at SUs, we perform the following hypothesis test,

$$\begin{array}{ll} \mathsf{H}_0: & \{y_1^k, \dots, y_t^k\} \sim f_0, \ k = 1, \dots, K \\ \mathsf{H}_1: & \{y_1^k, \dots, y_t^k\} \sim f_1, \ k = 1, \dots, K \end{array} . \tag{4}$$

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

$$L_t^k \triangleq \log \frac{f_1(y_1^k, \dots, y_t^k)}{f_0(y_1^k, \dots, y_t^k)} = \sum_{n=1}^{t} \underbrace{\log \frac{f_1(y_n^k)}{f_0(y_n^k)}}_{\ell_n^k} = L_{t-1}^k + \ell_t^k \tag{5}$$

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

SPRT

$$\mathcal{S} = \inf\{t > 0 : L_t \notin (-B, A)\}, \tag{6}$$

$$\delta(\mathcal{S}) = \begin{cases} 1, & \text{if } L_{\mathcal{S}} \geq A, \\ 0, & \text{if } L_{\mathcal{S}} \leq -B. \end{cases} \tag{7}$$

Spectrum Sensing via SPRT

Thresholds A, B are selected so that SPRT satisfies following constraints with equality.

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

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!

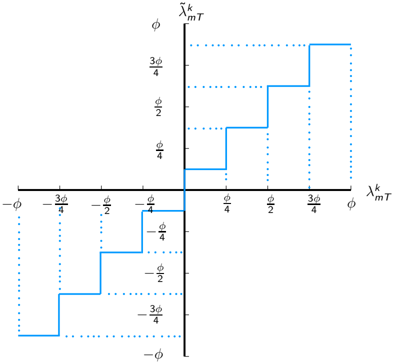
Objective

Decentralized schemes $\equiv$ low rate info. transmission from SUs to FC

Decentralized Q-SPRT Scheme

Secondary Users

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



- send $\tilde{\lambda}^k_{mT}$ to FC using $\log_2 \tilde{r}$ bits

Decentralized Q-SPRT Scheme

Fusion Center

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

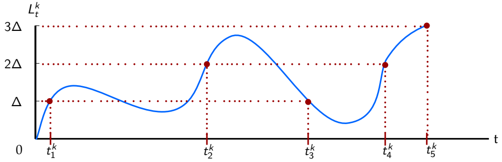
$$\tilde{L}_{mT} = \tilde{L}_{(m-1)T} + \sum_{k=1}^{K} \tilde{\lambda}_{mT}^{k} \qquad (9)$$

- applies the SPRT idea using $\tilde{L}_{mT}$ and $\tilde{A}, -\tilde{B}$ as the two thresholds.

Decentralized Scheme based on Event-triggered Sampling

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 $\lambda_n^k = L_{t_n^k}^k - L_{t_{n-1}^k}^k$ to FC (either Δ or $-\Delta$)

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

Remark

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

Fusion Center

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

$$\hat{L}_{t_n^k}^k = \sum_{j=1}^{n} \hat{\lambda}_j^k = \hat{L}_{t_{n-1}^k}^k + b_n^k \Delta = \sum_{j=1}^{n} b_j^k \Delta \tag{11}$$

Remark

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

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

$$\hat{L}_t = \sum_{k=1}^{K} \hat{L}_t^k = \Delta \sum_{k=1}^{K} \sum_{n:t_n^k \leq t} b_n^k \tag{12}$$

- applies the SPRT idea using $\hat{L}_t$ and $\hat{A}, -\hat{B}$ as the two thresholds.

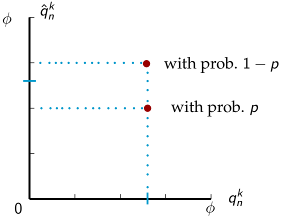
Enhancement: Overshoot Quantization at SUs

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.



p is chosen so that

$e^{\hat{L}_{t_n}}$ and $e^{-\hat{L}_{t_n}}$ are supermartingales,

which greatly simplifies the performance analysis of the scheme

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

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

Performance Analysis

Definition

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

$$1 \leq \frac{\mathsf{E}_i[\mathcal{T}]}{\mathsf{E}_i[\mathcal{S}]} = 1 + o_{\alpha,\beta}(1); \tag{14}$$

and *order-2 asymptotically optimal* if

$$0 \leq \mathsf{E}_i[\mathcal{T}] - \mathsf{E}_i[\mathcal{S}] = \mathsf{O}(1) \tag{15}$$

where $(\mathcal{S}, \delta_{\mathcal{S}})$ is the optimum SPRT.

Cont.-time results:

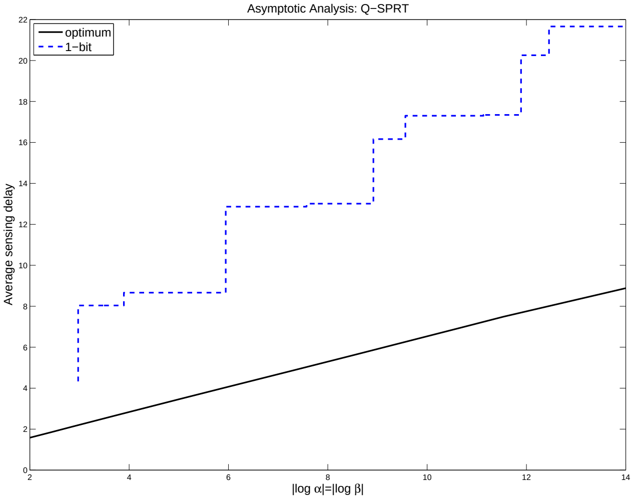
- Q-SPRT is **not** even order-1 asymp. optimal with **any fixed** number of bits.
- RLT-SPRT is **order-2** asymp. optimal with only **one** bit.

Comparisons

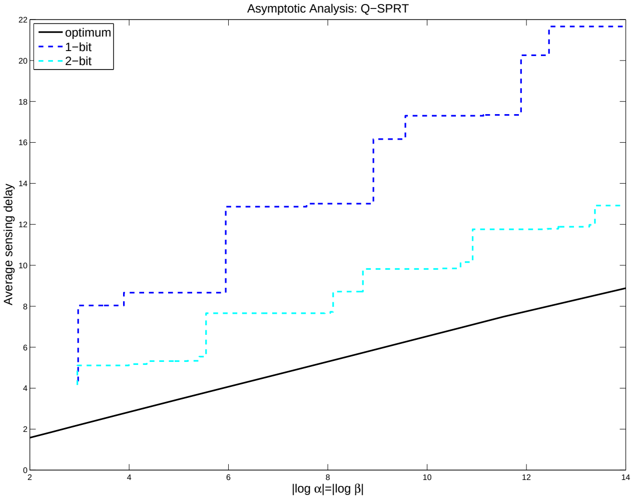
In order to make fair comparisons, Δ is adjusted so that average frequency of received messages by the FC is the same for Q-SPRT and RLT-SPRT.

- 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 asymp. optimality when $T \to \infty$ with a rate slower than $|\log \alpha|$.
- In contrast, by controlling T, Q-SPRT can not enjoy any form of asymp. optimality.

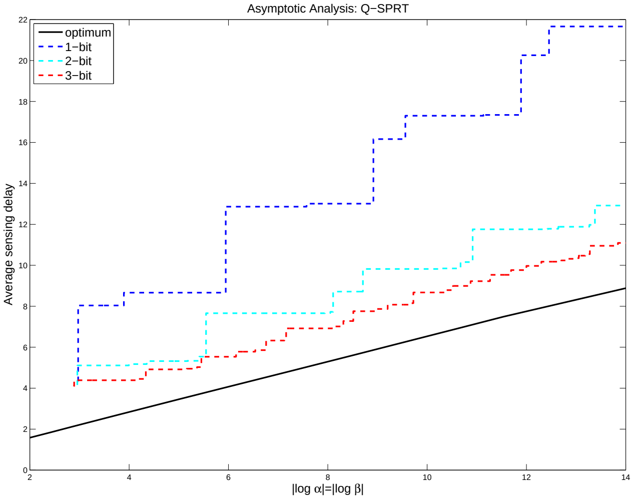
Simulations: Q-SPRT



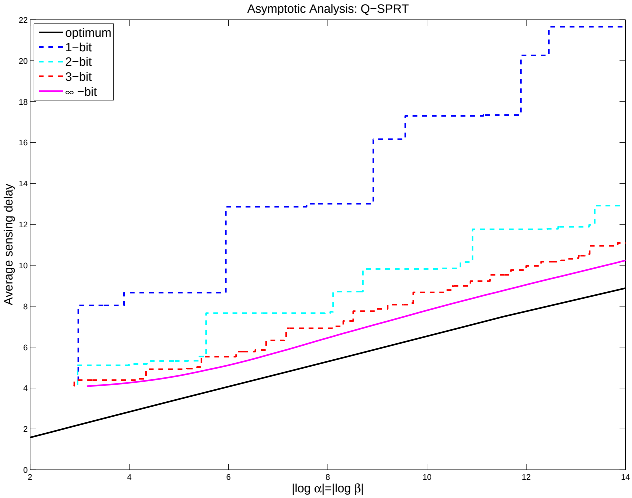
Simulations: Q-SPRT



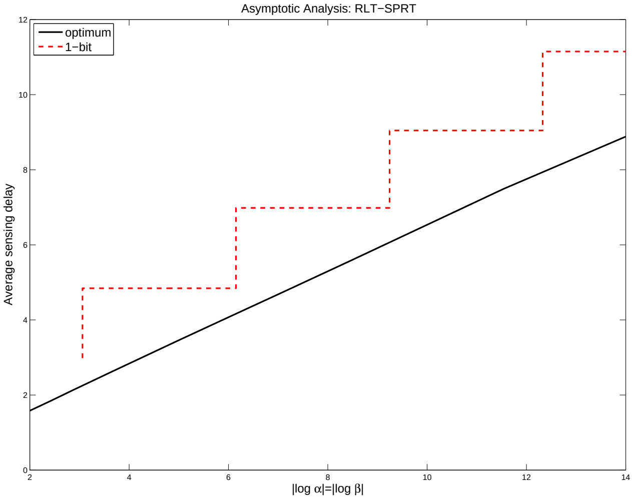
Simulations: Q-SPRT



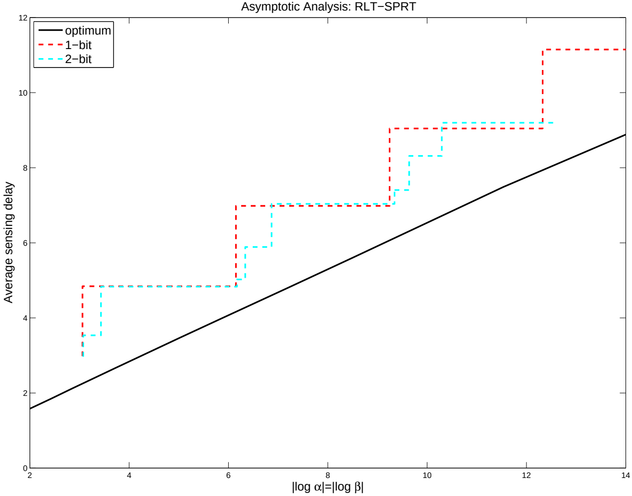
Simulations: Q-SPRT



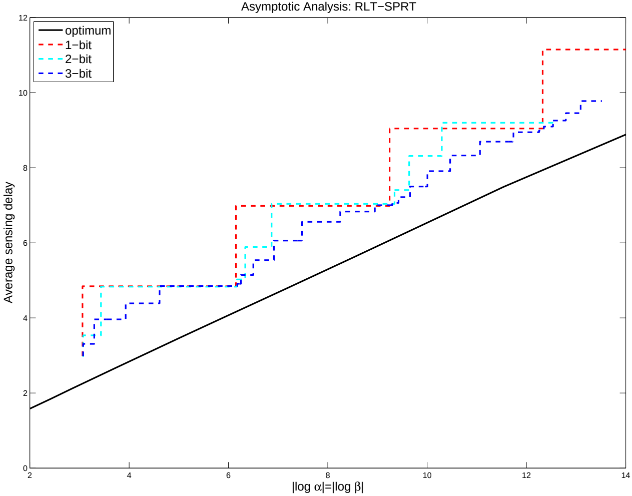
Simulations: RLT-SPRT



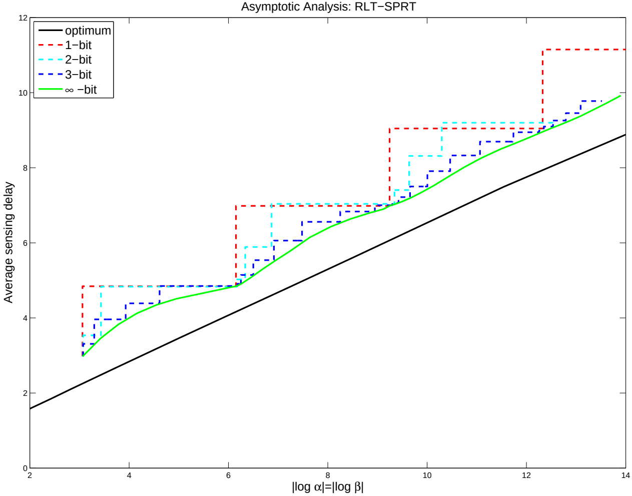
Simulations: RLT-SPRT



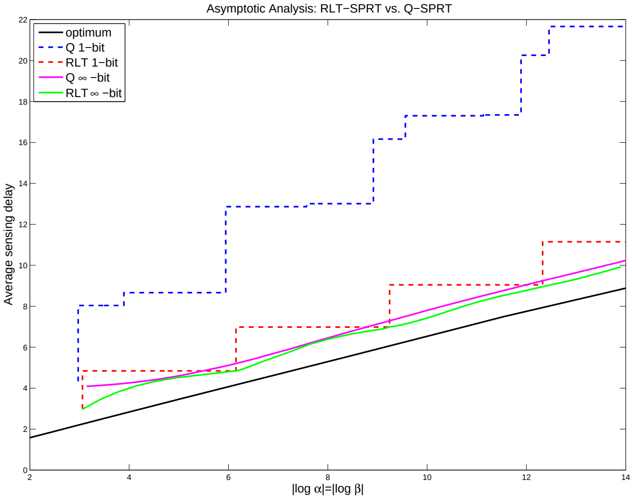
Simulations: RLT-SPRT



Simulations: RLT-SPRT



Simulations: RLT-SPRT vs. Q-SPRT



Conclusion

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 509147 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)

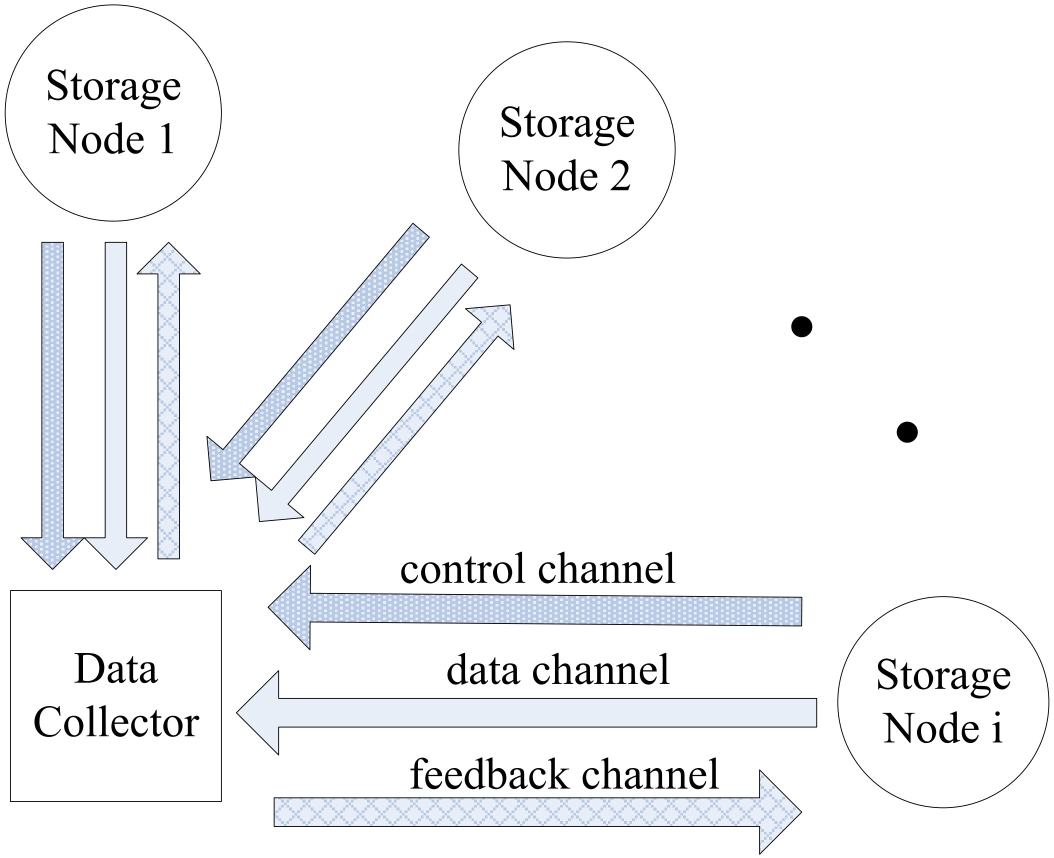
Distributed Wireless Clouds

- 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



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

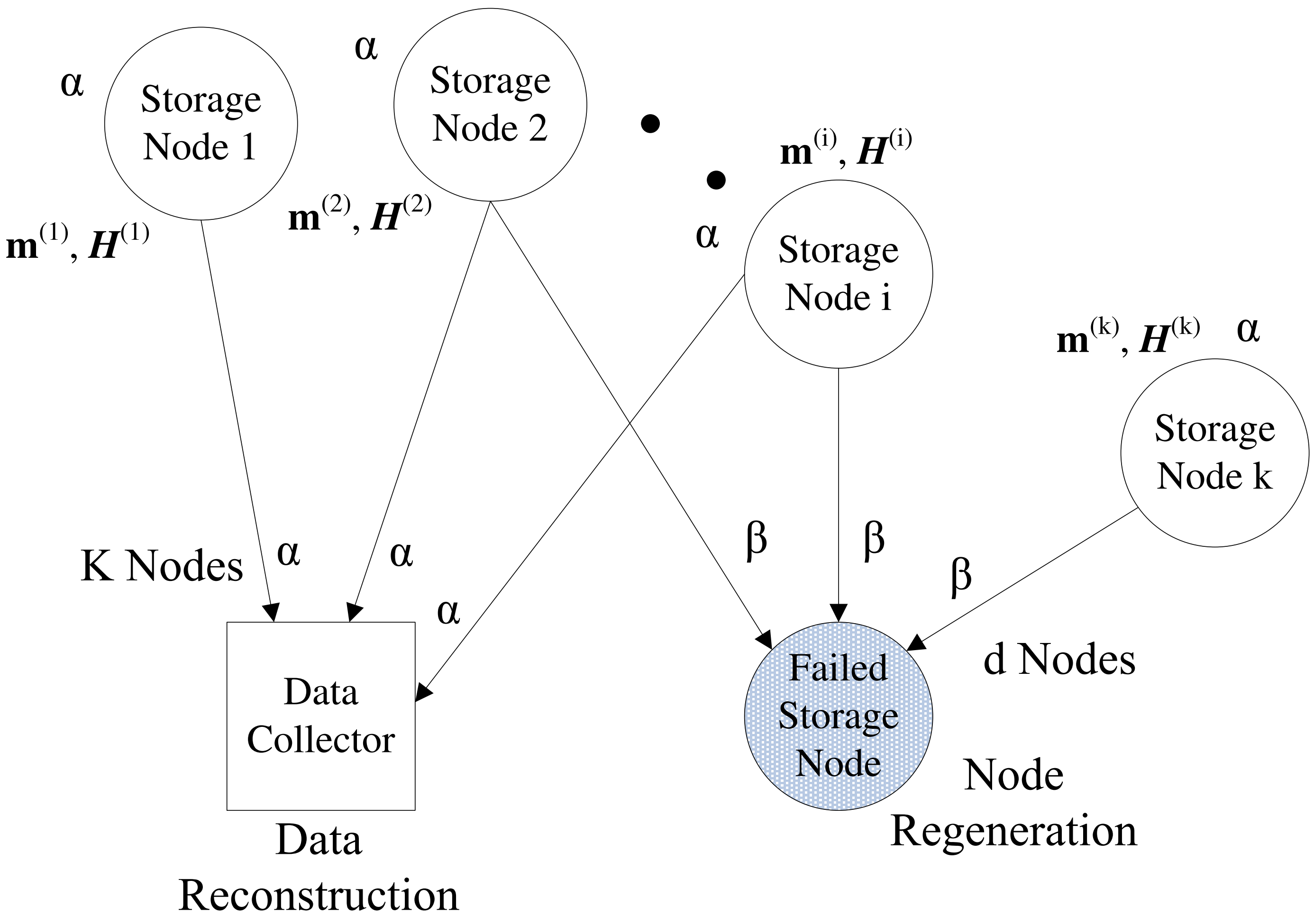
Distributed storage:

- 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, α, β) **regenerating code:**

- totally S storage nodes, each storing α symbols;
- reconstruct original file,
  - via downloading all $K\alpha$ symbols from any K nodes;
- repair a failed node,
  - via downloading β symbols each from any d surviving nodes

Data Reconstruction and Node Regeneration



Wireless Distributed Storage Data Coding

Distributed storage setting:

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

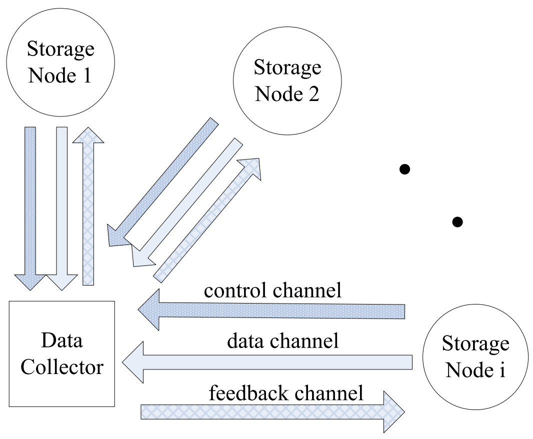
Wireless network setting:

- 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



wireless distributed storage network structure

Partial Downloading Linear Combination

Partial downloading formulation:

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

Downloading original symbols:

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

Wireless Cloud Resource Allocation Formulation

Wireless resource allocation:

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

Problem formulation:

minimize transmission power s.t. data reconstructable

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

Difficulty: how to analyze the data reconstructability

- transform the full rank constraint to ...

Data Reconstructability (MSR)

Full rank constraint:

- transform it using μ_i: downloading μ_i symbols from node i;

Minimum storage regeneration (MSR):

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

Wireless Cloud Resource Allocation (MSR)

Relaxed resource allocation problem:

- $\sum_{i \in \mathcal{S}} \mu_i \geq M$, $\mu_i \leq \alpha \Rightarrow$ remove constraint $\mu_i \leq \alpha$;
- problem reformulation
  - minimize transmission power, s.t. totally downloading M symbols;
  - $\min \sum_{j=1}^{N} P_j$; 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 $\mu_i \leq \alpha$, $i \in \mathcal{S}$ violated
- rarely happens in simulation scenarios

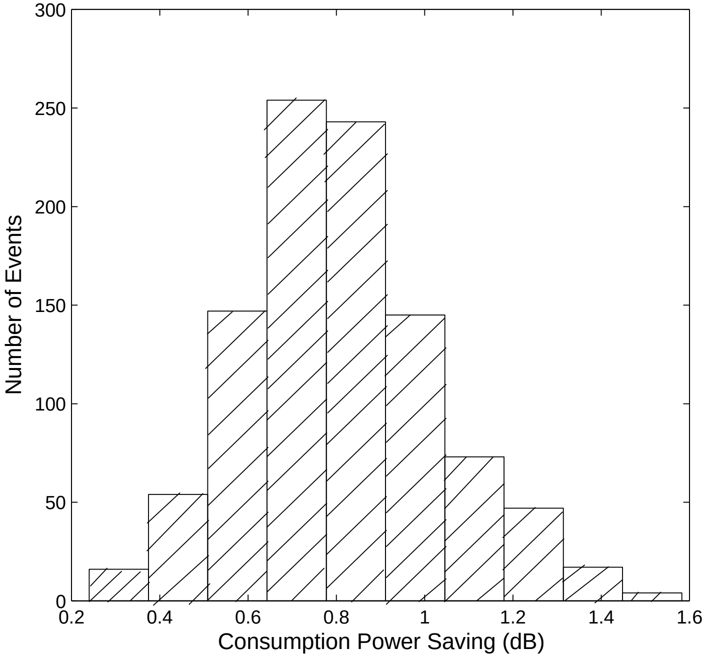
Wireless Cloud Resource Allocation Results (MSR)

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:

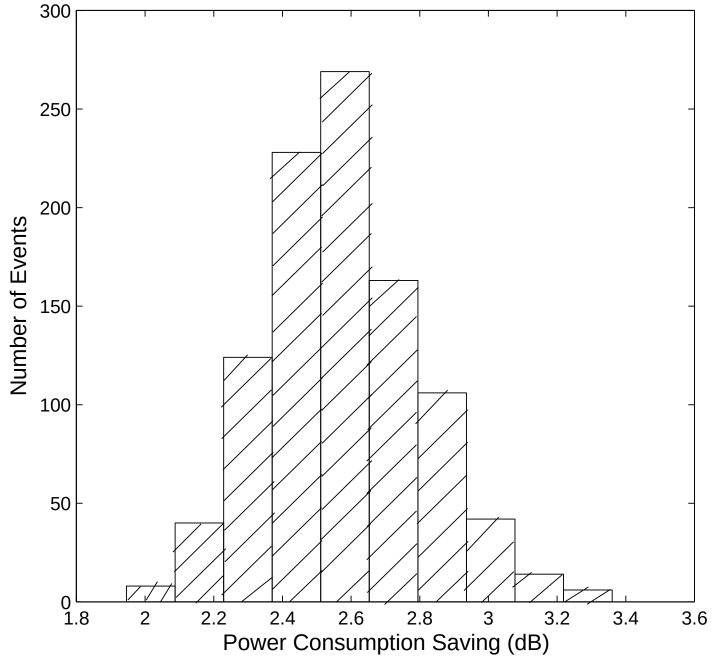
- more complicated reconstructability condition;
- downloaded symbols: $M(< K\alpha)$
  - $K\alpha$ required by full downloading;

Resource allocation:

- relax to $\sum_{i \in \mathcal{S}} \mu_i = M$;
- 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 $\sum_{i \in \mathcal{S}} \mu_i \geq M$ symbols suffice data reconstruction;
- needs γ symbols for failed node repair;

Bound for failed node repair γ:

- 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 $\sum_{i \in \mathcal{S}} \mu_i \geq M$ 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 $\gamma = 6$;