Partial Network Coding: Theory and Application for Continuous Sensor Data Collection

Dan Wang, Jiangchuan Liu
Simon Fraser University
Qian Zhang
Hong Kong Univ. of Sci. and Tech.

Outline

- Introduction and Motivation
- Partial Network Coding
- Practical Consideration
- Experiment Results
- Conclusion and Future Work

Outline

- Introduction and Motivation
- Partial Network Coding
- Practical Consideration
- Experiment Results
- Conclusion and Future Work

Introduction

We consider the following applications in sensor networks

- Raw data (no aggregation)
- Sensors record when event occurs
- Server collects data occasionally
- In each collection, data should be retrieved fast

Typical Examples

Habitat monitoring: Great Duck Island

Data Collection: Chemical Plants

Characteristics:

- Sensors should store data temporarily
- Data collection should be done efficiently

Requirements

Server: Ubiquitous data access

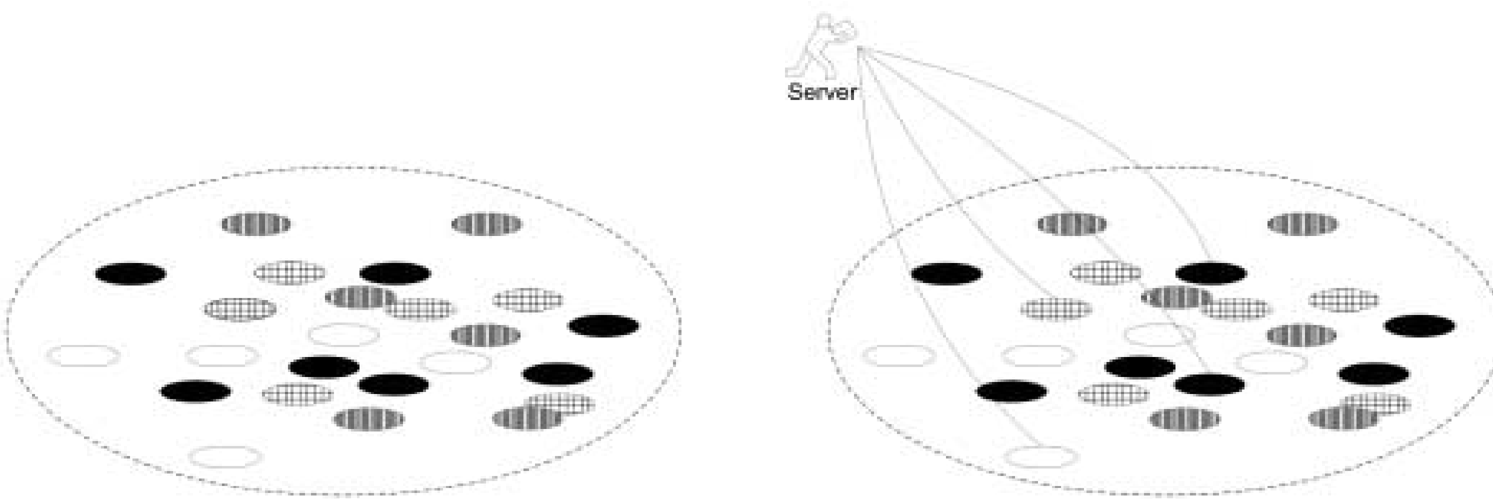
The server may randomly contact a few (but enough) sensors and each sensor uploads data

Sensors:

- limited storage space
- no knowledge of the storage of other sensors and the data uploaded by other sensors

Straight Forward Solution: Random Collection

This scheme may introduce huge duplication



Ubiquitous Data Access [ADMK05][DPR05]

Network coding (random linear coding) based distributed storage

Let c_i, $0 < i < N$ denote original data pieces, the combined data segments are $f = \sum_i \beta_i \times c_i$

The coded data segments are equivalent to each other in decode-ability

i.e., by random access any N combined data, the original data segments are recovered by solving a set of linear equations

Problems and Motivation

Can it be used for continuous data collection?

Original data segments are generated and will become obsolete; the system needs to remove obsolete data segments

Network coding is easy to combine new data, but how to remove existing ones?

- Decoding is required
- Impossible for storage-constraint sensors

Outline

- Introduction and Motivation
- Partial Network Coding
- Practical Consideration
- Experiment Results
- Conclusion and Future Work

Example: No Coding (Non-NC)

We have a sensor network with 6 sensors, s_0, s_1, … , s_5, and each sensor has 2 pieces of memory. We are interested in 4 most-recent data, c_3, c_2, c_1, c_0

A possible no-coding random storage:

s_0: c_2, c_1,	s_1: c_1, c_0
s_2: c_3, c_0,	s_3: c_3, c_1
s_4: c_3, c_2,	s_5: c_2, c_0

Example: Network Coding (NC)

A possible storage with network coding
  notice that the size of f_i is the same as c_i

s_0: $f_0=5c_3+2c_2+3c_1+4c_0$, $f_1=7c_3+2c_2+3c_1+4c_0$
s_1: $f_0=3c_3+2c_2+10c_1+c_0$, $f_1=c_3+2c_2+5c_1+c_0$
s_2: $f_0=2c_3+5c_2+7c_1+4c_0$, $f_1=c_3+5c_2+6c_1+3c_0$
s_3: $f_0=c_3+8c_2+9c_1+4c_0$, $f_1=2c_3+8c_2+9c_1+4c_0$
s_4: $f_0=2c_3+5c_2+c_1+4c_0$, $f_1=12c_3+6c_2+c_1+4c_0$
s_5: $f_0=c_3+8c_2+9c_1+4c_0$, $f_1=8c_3+8c_2+8c_1+4c_0$

Partial Network Coding

In stead of $f = \sum_i \beta_i \times c_i$, we have

$$\mathcal{B} = \{ f^k \mid f^k = \sum_{j=k}^{N-1} \beta_j c_j, k \in [0,\ldots, N-1], \beta_j \in F_q \}$$

This leads to a triangle-like coding base (we omit the coefficients)

$$f^0 = [c_3, c_2, c_1, c_0]$$
$$f^1 = [c_3, c_2, c_1]$$
$$f^2 = [c_3, c_2]$$
$$f^3 = [c_3]$$

An example with coding base 4 (cardinality)

Storing Coded Data

Let the buffer space of each sensor be B, then the storage of the sensors are

$$\mathcal{S} = \{ f_i^k \mid f_i^k \in \mathcal{B}, 0 \le i \le B \}$$

Example: Partial Network Coding (PNC)

A possible storage with partial network coding

s_0: $f_0=[c_3, c_2, c_1, c_0]$, $f_1=[c_3, c_2]$
s_1: $f_0=[c_3, c_2, c_1]$, $f_1=[c_3, c_2]$
s_2: $f_0=[c_3, c_2]$, $f_1=[c_3]$
s_3: $f_0=[c_3, c_2, c_1, c_0]$, $f_1=[c_3, c_2, c_1]$
s_4: $f_0=[c_3, c_2, c_1]$, $f_1=[c_3]$
s_5: $f_0=[c_3, c_2, c_1, c_0]$, $f_1=[c_3, c_2]$

Advantages

From the above example, if c_4 is recorded, then the storage of the system will change to

s_0: $f_0=[c_4,]$, $f_1=[c_4, c_3, c_2]$

s_1: $f_0=[c_4, c_3, c_2, c_1]$, $f_1=[c_4, c_3, c_2]$

s_2: $f_0=[c_4, c_3, c_2]$, $f_1=[c_4, c_3]$

s_3: $f_0=[c_4]$, $f_1=[c_4, c_3, c_2, c_1]$

s_4: $f_0=[c_4, c_3, c_2, c_1]$, $f_1=[c_4, c_3]$

s_5: $f_0=[c_4]$, $f_1=[c_4, c_3, c_2]$

Data replacement scheme

Algorithm Data Replacement (c_n)

c_n : newly recorded data

for i = 1…B

if cardinality(f_i) < N

$f_i = ß_n \times c_n + f_i$

else

$f_i = ß_n \times c_n$

Comments on Data Replacement

- Data replacement needs to maintain uniform distribution of the cardinality
  - This is essential for stable performance

- Theorem 1: If the cardinality is distributed uniformly in the sensor network, then after data replacement, the distribution of cardinality is still uniform

Performance of PNC

Can we still decode easily?

Success Probability: the probability of obtaining all the original data pieces

Theorem 2: the success probability of PNC is at least no worse than Non-NC

Compare with NC?

After collecting some number of segments, can we guarantee decode?

Possible Improvement

The probability of obtaining c_n and c_0 is not the same

c_0 is more difficult to collect

- We request each sensor to upload the combined data segment with the largest cardinality
- We extend the cardinality of the system, so that more combined data segments have c_0

How much is enough?

Storage at Each Sensor

$$f_0 = [c_{N-1}, \cdots, c_{N-\sqrt{N}-1}, \cdots, c_{2\sqrt{N}-1}, \cdots, c_0, \cdots, c_{-\sqrt{N}}]$$

$$f_1 = [c_{N-1}, \cdots, c_{N-\sqrt{N}-1}, \cdots, c_{2\sqrt{N}-1}, \cdots, c_0]$$

$$f_2 = [c_{N-1}, \cdots, c_{N-\sqrt{N}-1}, \cdots, c_{2\sqrt{N}-1}]$$

……

$$f_{\sqrt{N}} = [c_{N-1}, \cdots, c_{N-\sqrt{N}-1}]$$

Lemma 3: By extending the full cardinality to N+sqrt(N) and the buffer to sqrt(N), at any time, each sensor is guaranteed to have a data segment of cardinality greater than N

Performance of PNC

Corollary 5: the success ratio of PNC with $B = \sqrt{N} + 1$ and $W = N + \sqrt{N}$ is identical with NC

With a sub-linear sacrifice of buffer size and communication cost, we guarantee the decode-ability of PNC to be the same as NC

Outline

- Introduction and Motivation
- Partial Network Coding
- Practical Consideration
- Experiment Results
- Conclusion and Future Work

Practical Concerns

Overhead

- Communication overhead – coefficient vectors need to be uploaded
- Computation overhead on data replacement algorithm – very small

Clustering

If the memory is small, e.g., <<sqrt(N), sensors can form clusters to store sqrt(N) combined data cooperatively

Outline

- Introduction and Motivation
- Partial Network Coding
- Practical Consideration
- Experiment Results
- Conclusion and Future Work

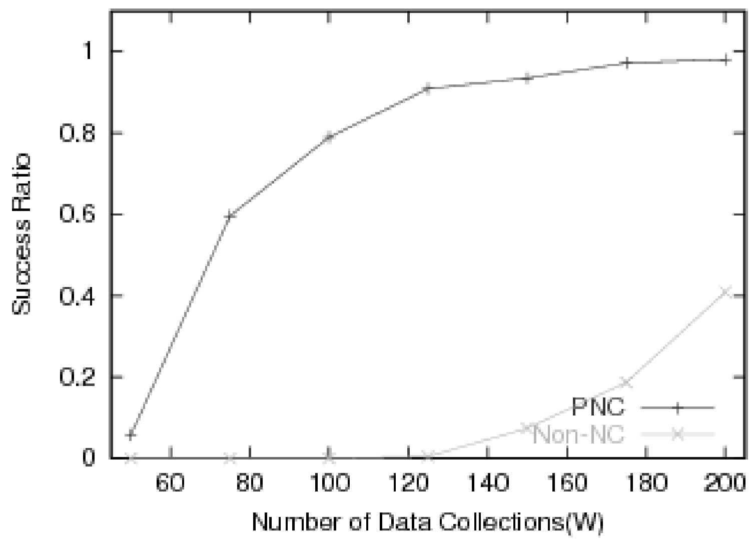
System Settings

By default N = 50; each data point is an average of 1000 experiments.

Evaluation metric:

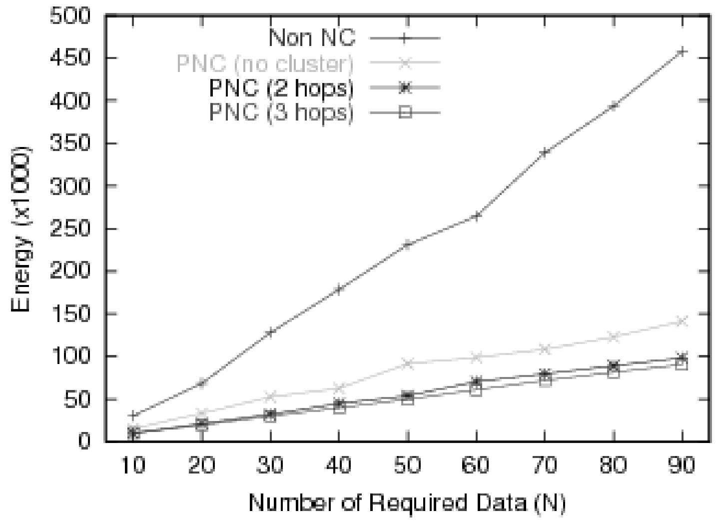
- Success Ratio: the probability of obtaining all the original data pieces
- Energy: the energy used to collect all original data pieces

Performance Evaluation



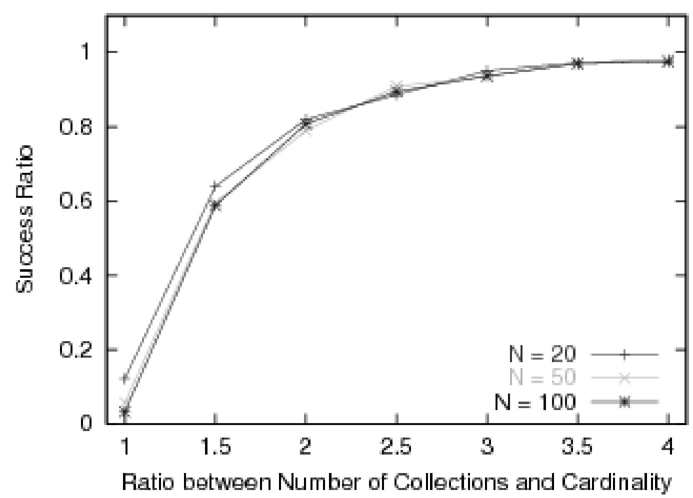
PNC vs Non-NC

Performance Evaluation



Energy Comparison

Performance Evaluation



Variate N

Summary

We propose a new coding scheme PNC

- capable of decoding free data removal; PNC provides insight on the improvement of conventional NC
- The performance of PNC is close to NC; sacrifice of sublinear factors

Future Work

PNC encourages us to think more…

- Network coding becomes more beneficial when more data pieces are combined; our work shows the opposite direction is worth consideration as well
- It may be useful for other streaming applications
- Do we reach the potential of PNC?

References

- [ADMK05] S. Acedanski, S. Deb, M. Medard, and R. Koetter, "How Good is Random Linear Coding Based Distributed Storage", in *Proc. NETCOD'05*, Italy, Apr. 2005.
- [DPR05] A. Dimakis, V. Prabhakaran and K. Ramchandran, "Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes", in *Proc. IPSN"05*, Los Angeles, CA, Apr. 2005.
- [WZL06] D. Wang, Q. Zhang and J. Liu, "Partial Network Coding and Continues Data Collection in Sensor Networks", in *Proc. IWQoS'06*, New Haven, CT, June, 2006.