Optimal Stopping of the Context Collection Process in Mobile Sensor Networks

Christos Anagnostopoulos[1], Stathes Hadjiefthymiades[2], Evangelos Zervas[3]

[1] University of Glasgow, School of Computing Science, UK
[2] National and Kapodistrian University of Athens, Dept. of Informatics & Telecommunications, Greece
[3]TEI Athens, Dept. Electronics, Greece

The considered setting

Consider a **Mobile Sensors Network** (MSN) with

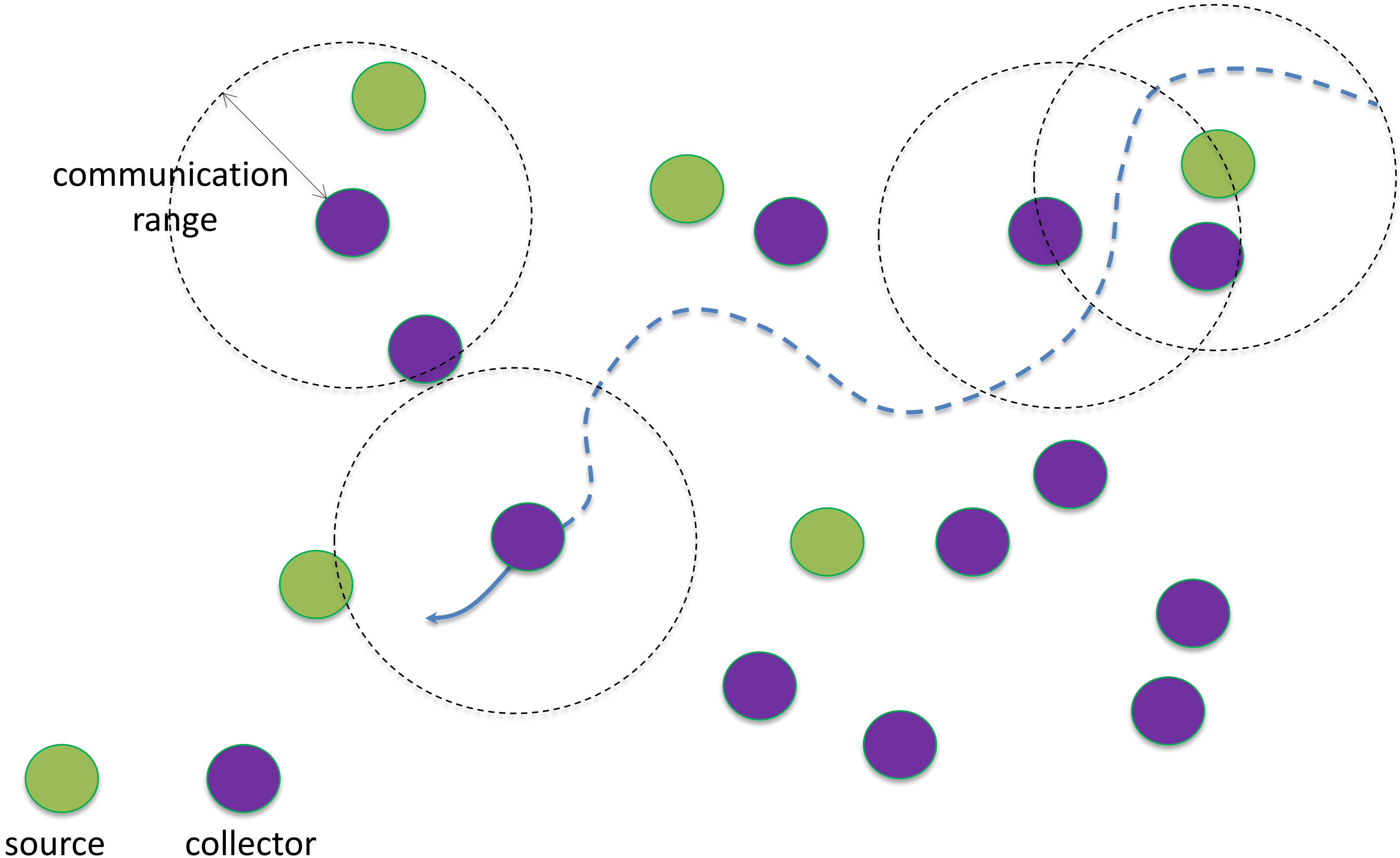
- **sources**, *i.e.*, sensors that produce *context* (*e.g.*, humidity)
- **collectors**, *i.e.*, mobile nodes that receive, store and forward context to their neighbors.

Context is **quality-stamped**, *e.g.*, freshness.

The context *quality indicator* **decreases** with time.

The **aim** of a collector is to **gather** as many high-quality pieces of context as possible from sources and/or collectors.

The considered setting



The problem

The **collectors** in a MSN:

- *forage* for high quality context and, then, *deliver* it to mobile context-aware applications;

- *undergo* a context collection process by exchanging data with neighboring collectors and/or sources **in light of** receiving context of <u>better</u> quality and/or <u>new</u> context;

- *cannot prolong* this process forever, since context quality decreases with time, thus, delivered context might be unusable for the application.

Some definitions

Context c is represented as:

$$c = \langle p, u, x_u \rangle$$

where:

p is contextual parameter/type (*e.g.*, temperature),

u is contextual value (*e.g.*, 30 °C),

x_u is quality indicator of value u

Context quality indicator

Indicator $x_u \in [0, 1]$ indicates *freshness* of u.

- $x_u = 1$ indicates that u is of **maximum quality**.
- $x_u = 0$ indicates that u is **unusable**.

Context at time t is called **fresh** if $x_u(t) > 0$; otherwise it is called **obsolete**.

Context quality indicator

Indicator $x_u(t)$ at time $t > 0$ is updated as follows:

$$x_u(t) = x_u(t-1) - 1/z \ , \qquad x_u(0) = 1,\ z \neq 0$$

- z is the **validity horizon** for parameter p in which value u is considered usable.
- z is application specific, *e.g.*, $z = 10$min if p is temperature, $z = 1$min if p is wind-speed.

Notice: Alternative quality indicator functions can be, for instance, the inverse exponential function

Rationale

Consider a **collector** which has collected a set of N **fresh** pieces of context, $C = \{c_1, c_2, \ldots, c_N\}$, referred to as *local context*.

Let collector receive context q from a neighbor collector.

Collector increases its **local context** in **type** and/or **quality** as follows:

- If q is obsolete then collector **discards** q;
- If q is fresh and there is some local context c with the same type of q but less fresh than q then collector **replaces** c with q;
- If q is fresh and there is no other local context of the same type, collector **inserts** q into C;

Degree of completeness

Local context C is quantified through **degree of completeness (DoC)**, Y, defined as the random variable [1]:

$$Y = N \cdot \sum_{k=1}^{N} X_k$$

- N is the *current* number (**quantity**) of collected pieces of context; $N \in \{0, 1, 2, \ldots, m\}$, $m > 0$.
- X_k is the *current* **quality** indicator of the kth contextual parameter in C.

[1] C. Anagnostopoulos, S. Hadjiefthymiades, 'Multivariate context collection in mobile sensor networks', Computer Networks, Elsevier, 57(6):1394–1407, April 2013

Degree of completeness

When the collector decides to **stop** the collection process at some time, it wants to achieve the highest expected value of Y.

Hence, the collector has to find an optimal stopping time *t* of the collection process which maximizes:

$$E[Y_t] = E[N_t \cdot \sum_{k=1}^{N_t} X_k^t]$$

Optimal Stopping Theory (OST)

- Choose the **best** time to **take** a decision of performing a certain action.

- **Observe** the current state of a system and decide whether to:
  - **continue** the process or
  - **stop** the process, and incur a certain cost.

…the *discounted sum* problem, the *odds* algorithm, the *secretary* problem, the *parking* problem, the *asset-selling* problem, etc.

Application to context collection problem

- **Decision**
  - *When* to **stop** collecting pieces of context from neighboring collectors/sources and deliver them to the application.
- **Cost**
  - *Quality* of local pieces of context decreases with time.
  - *Serving* obsolete context to the application.

- **Approach**
  - *Adoption* of the OST **discounted sum** problem

Discounted sum problem in context collection

The decision of the collector at time t is:

- **stop** and **deliver** local context to the application, or
- **continue** the process and **update** local context

Let us define a *tolerance threshold* $\theta \in (0, m^2)$ such that:

If $Y > \theta$ **Then** local context is significantly adequate for the collector's requirements in terms of **quantity** and **quality.**

Discounted sum problem in context collection

Consider the indicator function:

$$I_t = \begin{cases} 1, \text{if } Y_t > \theta \\ 0, \text{otherwise} \end{cases}$$

and the *cumulative* sum up to time t: $S_t = \sum_{n=1}^{t} I_n$

The problem is to **decide** how large S_t should get before the collector stops, *i.e.*, we have to determine a time t such that the *supremum*

$$\mathbf{sup}_t \mathbf{E}[\, \beta^t S_t \,]$$

is attained, $0 < \beta < 1$.

Discounted sum problem in context collection

Optimal Stopping Rule: Observe I_t value at time *t* and stop at the *first* time for which it holds true that:

$$S_t \geq \frac{\beta}{1-\beta}\left(1 - F_Y(\theta)\right)$$

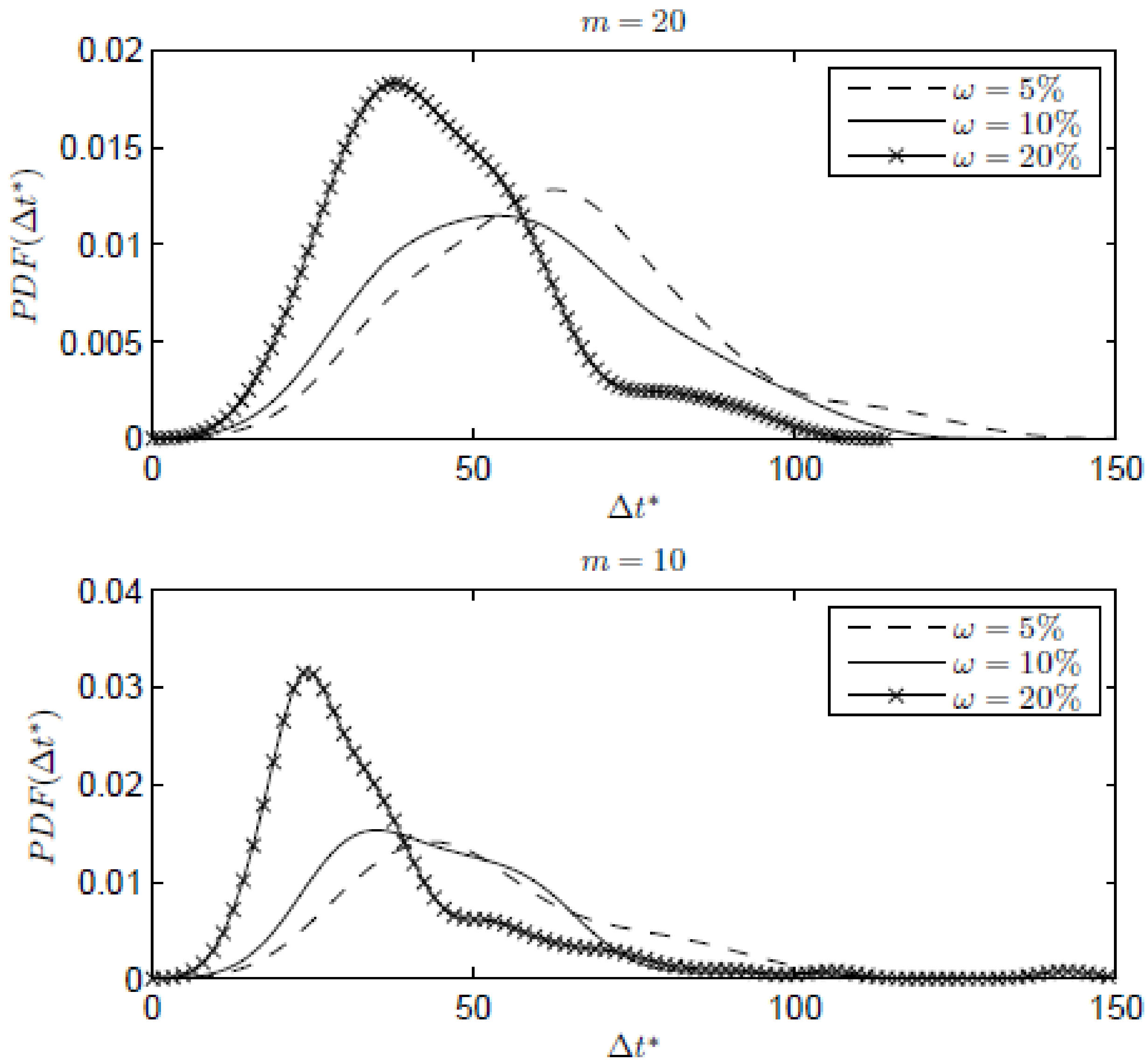
- $F_Y(y)$ is the cumulative distribution function of Y.
- β is *discount factor* indicating the **necessity** of collector to take a decision;
  - collector requires a rather extended time horizon for deciding on deliver context when β is high.

Performance & Comparative Assessment

- **Simulation setup**
  - MSN of 100 nodes; number of sources $\omega = \{5, 10, 20\}$
  - Mobility model: Random Waypoint
  - Validity horizon $z \sim U(2,10)$min.
  - Tolerance threshold $\theta \in [0.2, 0.7]$
  - Maximum quantity of contextual parameters $m = \{10, 20\}$
- **Comparison Schemes**
  - **Scheme C**: Randomized policy: collectors stop the process at a random time instance
  - **Scheme B**: Finite-Horizon policy [1]: collectors stop the process based on a pre-defined deadline T; adoption of OST
- **Metric**
  - Normalized average value of DoC delivered to the application;
    - the higher DoC is, the higher context **quality** and **quantity** is delivered to the application

Performance & Comparative Assessment

The Probability Density Function (PDF) of the **decision delay** Δt^*, *i.e.*, interval between following collection processes, for diverse number of contextual parameters *m*.



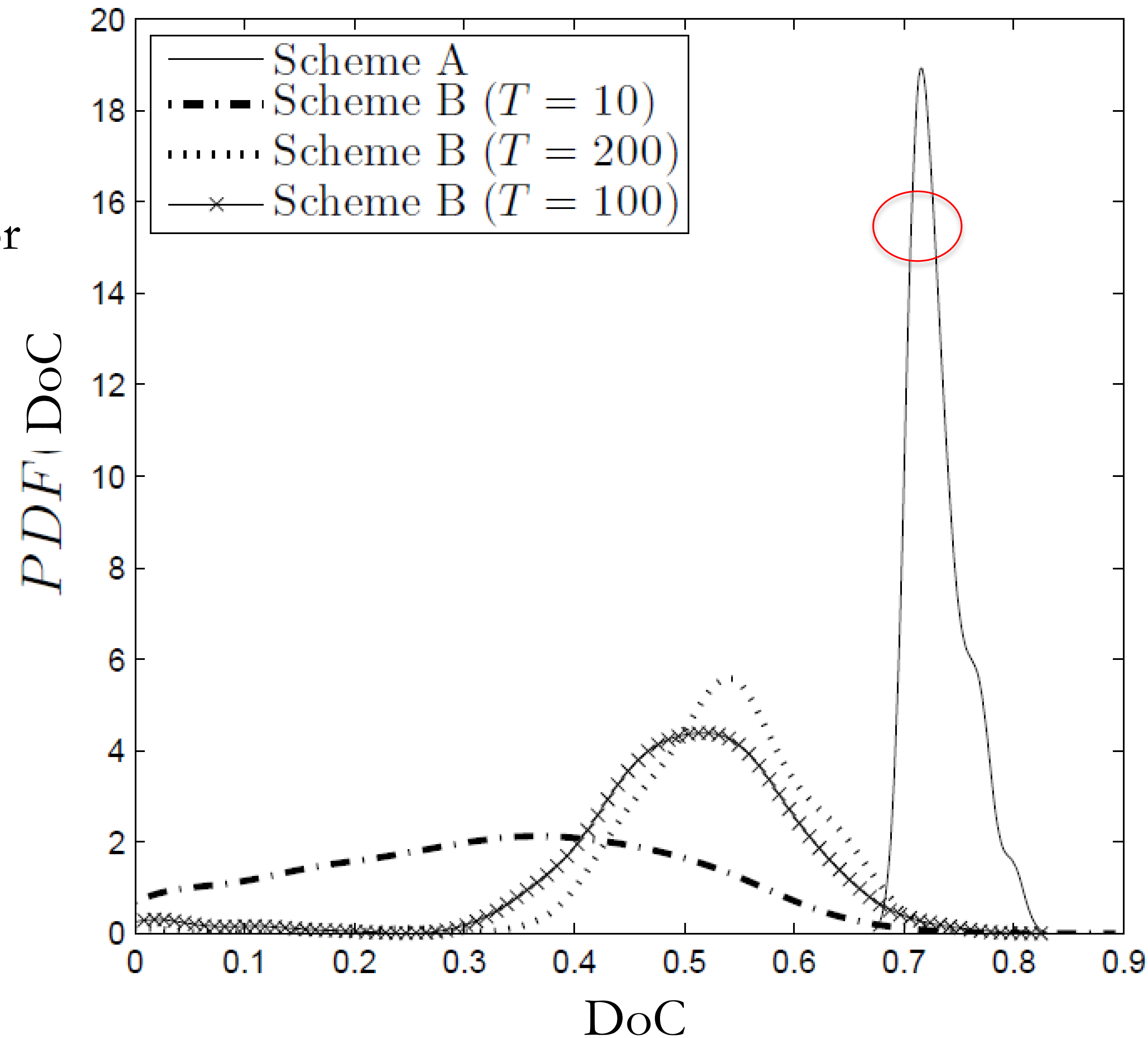
Performance & Comparative Assessment

THE DoC FOR SCHEMES A, B, & C

Scheme	B			A			C
$m = 10$							
$\omega(\%)$	$T = 10$	100	200	$\beta = .5$	.9	.98	
5	.33	.52	.44	.64	.72	.71	.175
10	.33	.55	.57	.65	.74	.74	.175
20	.35	.55	.56	.66	.75	.74	.175
$m = 20$							
$\omega(\%)$	$T = 10$	100	200	$\beta = .5$	.9	.98	
5	.30	.42	.45	.62	.63	.63	.17
10	.31	.48	.48	.64	.63	.62	.17
20	.32	.50	.50	.64	.64	.63	.17

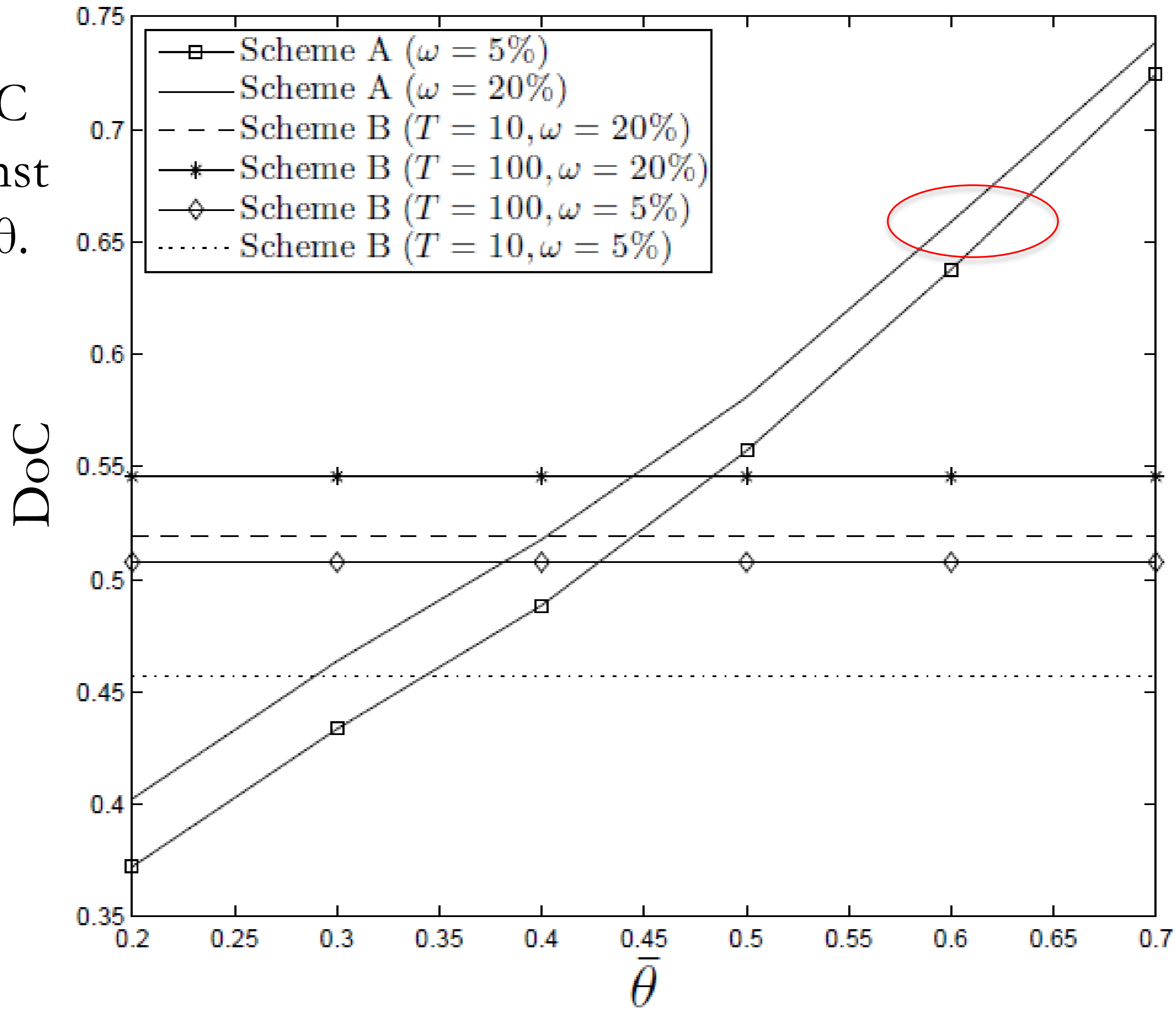
Performance & Comparative Assessment

The PDF of normalized DoC for all schemes.



Performance & Comparative Assessment

The normalized DoC for all schemes against tolerance threshold θ.



Conclusions

- A solution to the *context collection problem* based on Optimal Stopping Theory;
- Collectors *autonomously* take time-optimized context delivery decisions **without** a deadline;
- Collectors deliver context of high quality and quantity within **short** delays;
- Our scheme performs **well** when dealing with applications which require context of high quality and quantity (*i.e.*, high tolerance threshold)

Thank you!