

**Università degli Studi di Padova**  
**Dipartimento di Ingegneria dell'Informazione**  
**Tesi di Laurea**



# **Implementazione di Sincronizzazione Temporale Distribuita in Reti di Sensori Wireless**

RELATORE: *Prof. Luca Schenato*

LAUREANDO: *Federico Fiorentin*

# Obiettivo

---



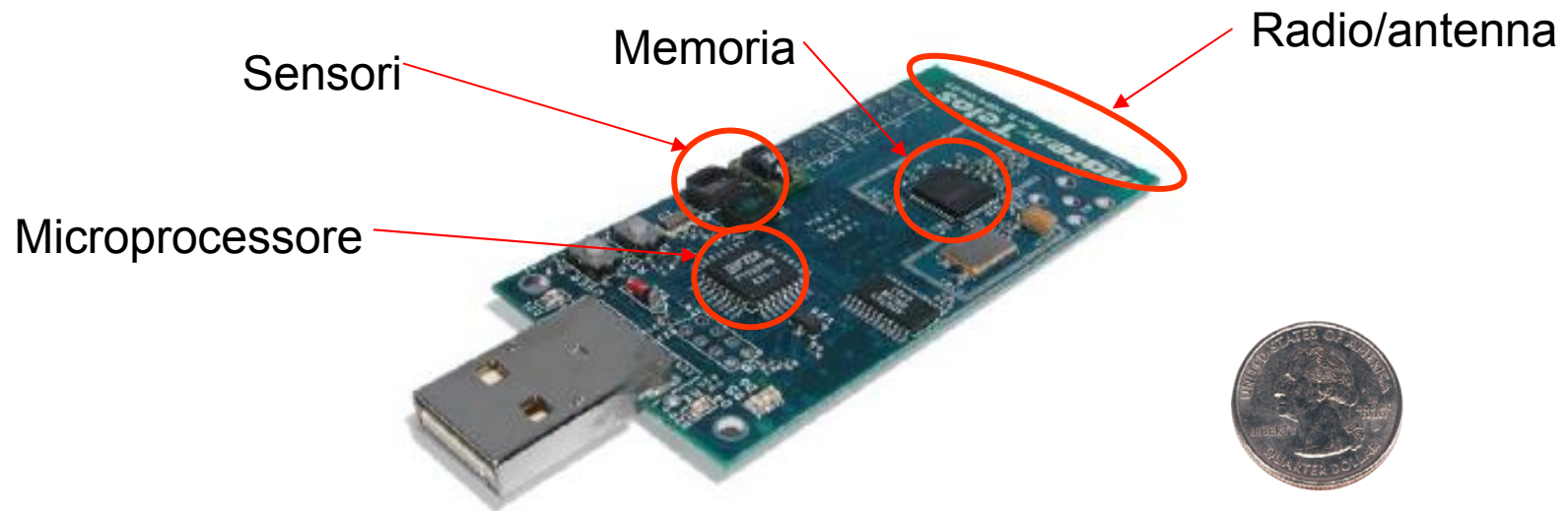
# Contributo

---

- Implementazione dell'Algoritmo
- Confronto con le soluzioni presenti in Letteratura
- Analisi sperimentale al variare delle condizioni di funzionamento

# Wireless Sensor Network (WSN)

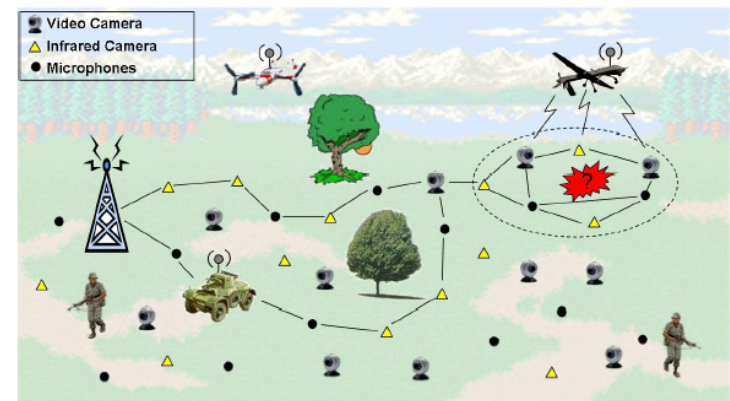
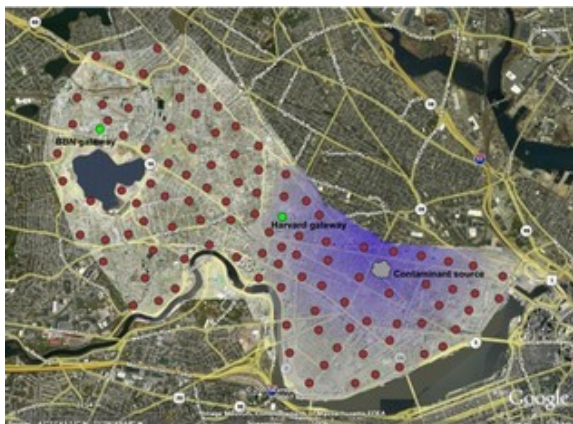
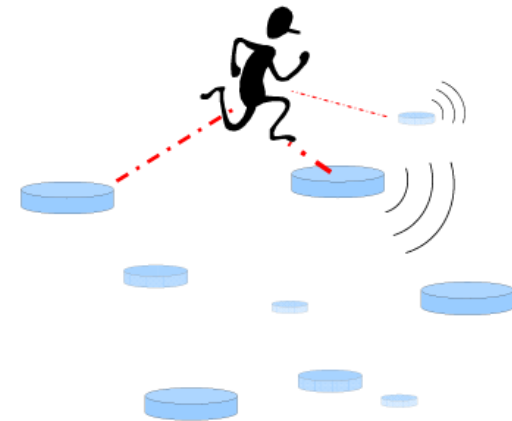
- Unità di piccole dimensioni



- Alimentazione a Batterie
- Comunicazione Multi-Hop

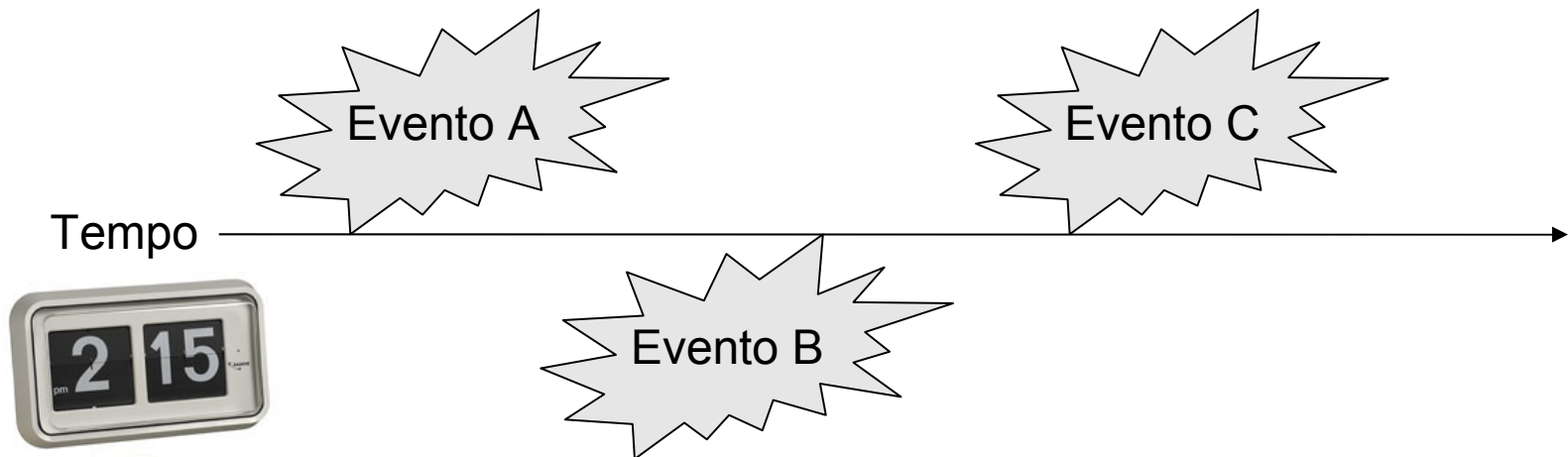
# Wireless Sensor Network (WSN)

- Sensori Wireless ( $\sim 10^2 - 10^3$ )
- Reti ad hoc per il monitoraggio di un'area
  - Random placement
- Bassa manutenzione



# È importante la Sincronizzazione Temporale?

Il concetto di tempo è fondamentale



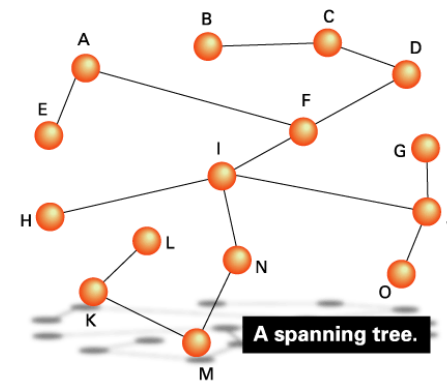
- Il tempo fisico è necessario per stabilire che relazione hanno gli eventi monitorati
- Indispensabile per coordinare azioni di gruppo:
  - moltiplicazione del canale di comunicazione
  - power scheduling

# Soluzioni esistenti

- Global Positioning System (GPS)
- Network Time Protocol (NTP) e IEEE1588 (PTP)

## Per WSN:

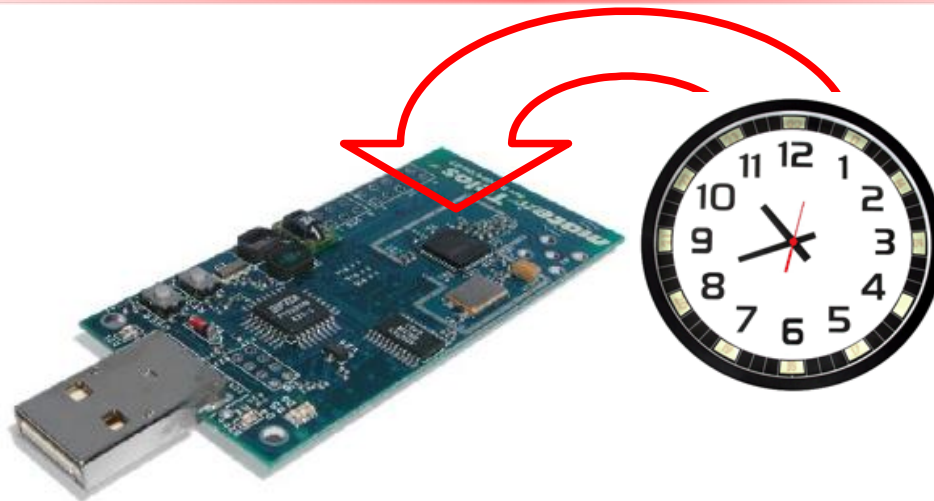
- Reference Broadcast Synchronization (RBS)
- Timing-sync Protocol for Sensor Networks (TPSN)
- Flooding Time Synchronization Protocol (FTSP)



- RFA
- Solis et al. Algorithm
- Simeone et al. Algorithm

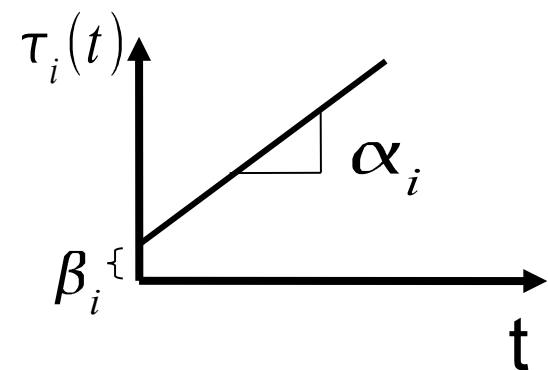
**Distribuiti**

# Caratteristiche del Clock



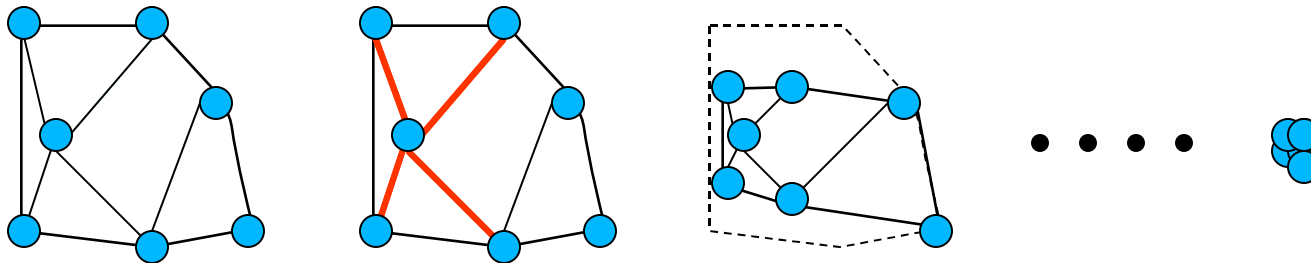
$$\tau_i(t) = \alpha_i t + \beta_i$$

Deriva                      Offset



# Average TimeSync

- Clock di riferimento virtuale (no radice o nodi speciali)
- Riferimento virtuale = media orologi vicini (consensus)

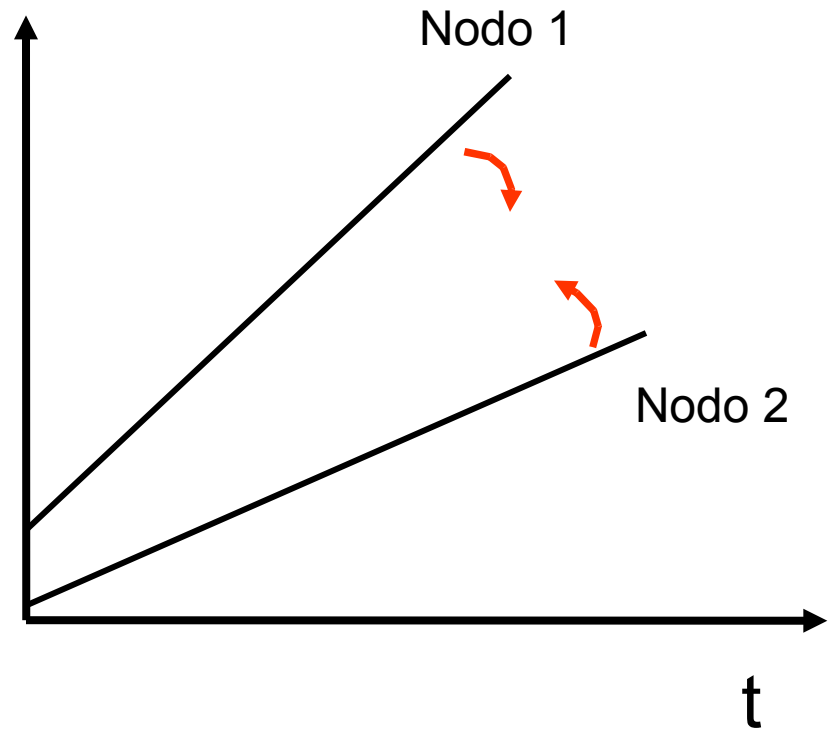


- Algoritmo = 2 algoritmi di consensus in cascata
  - Consensus su velocità dei clocks (compensazione deriva)
  - Consensus su offset (compensazione cond. Iniziali)



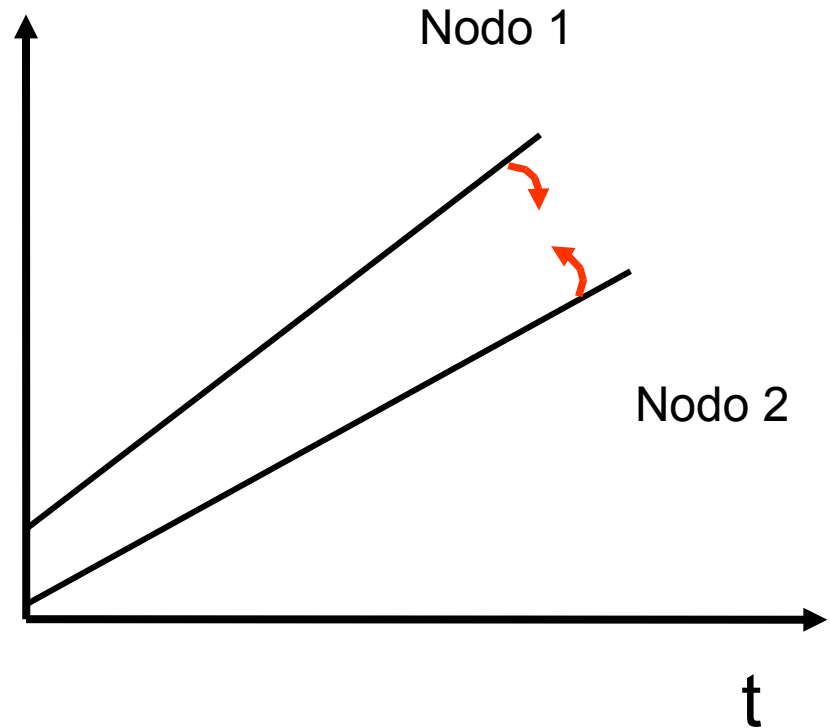
# Average TimeSync

## 1. Compensazione della Deriva



# Average TimeSync

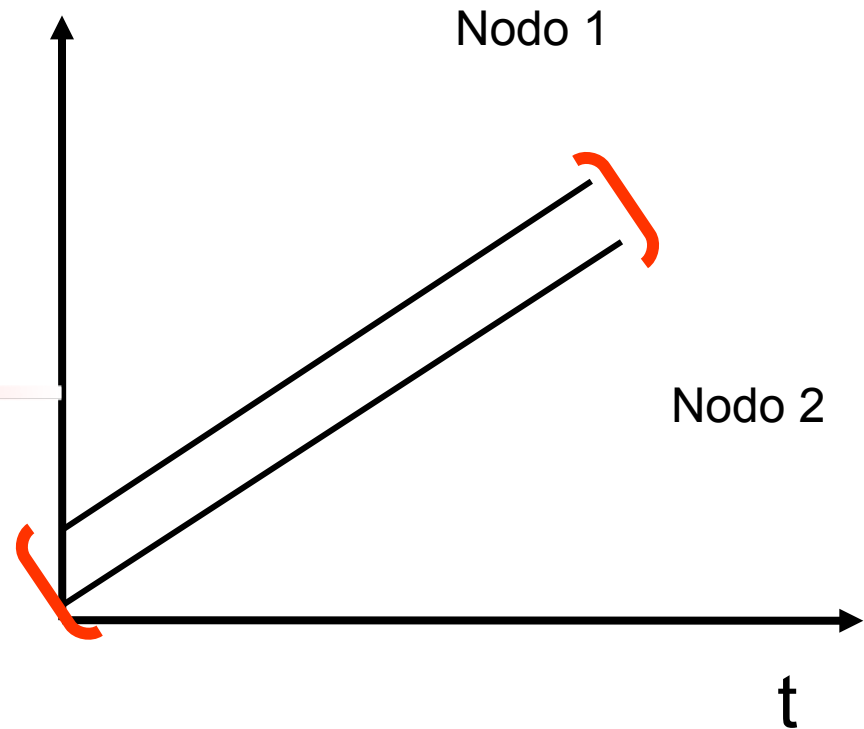
## 1. Compensazione della Deriva



# Average TimeSync

- Compensazione della Deriva

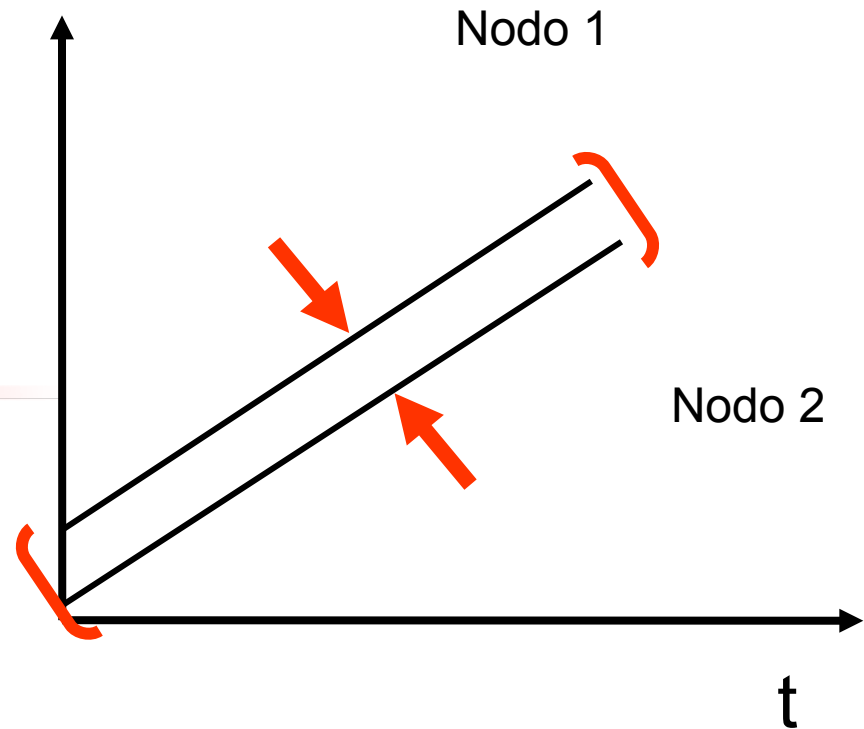
2. Compensazione dell'Offset



# Average TimeSync

- Compensazione della Deriva

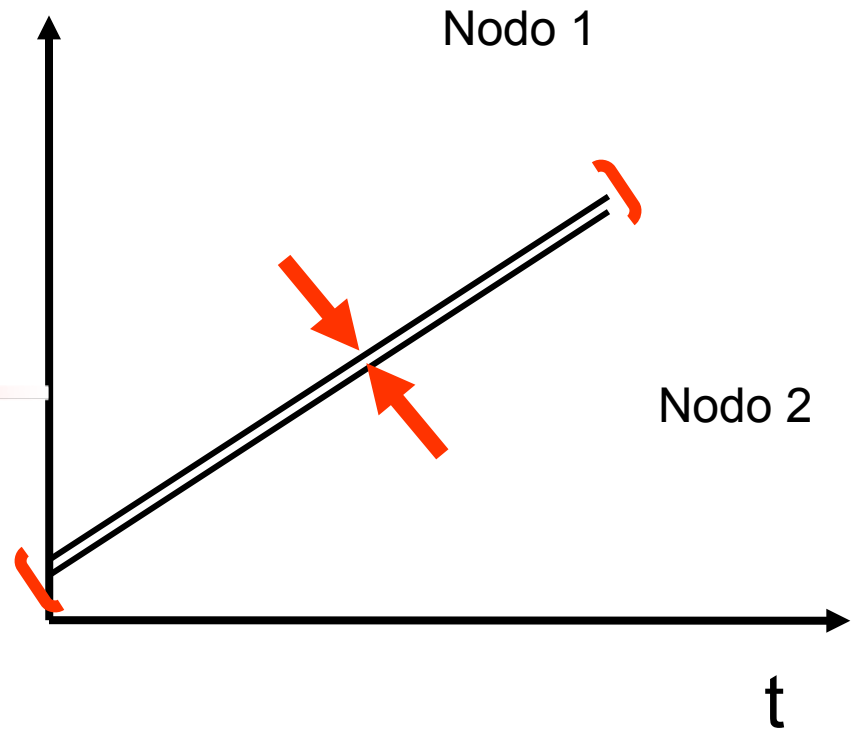
## 2. Compensazione dell'Offset



# Average TimeSync

- Compensazione della Deriva

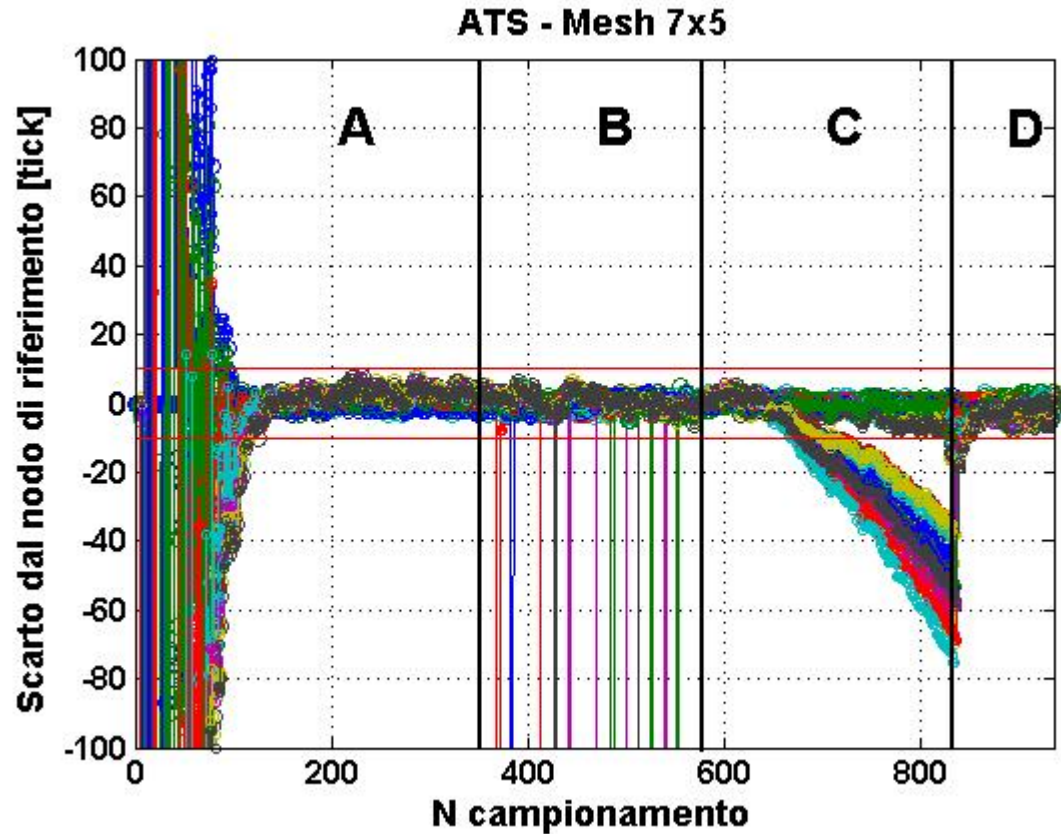
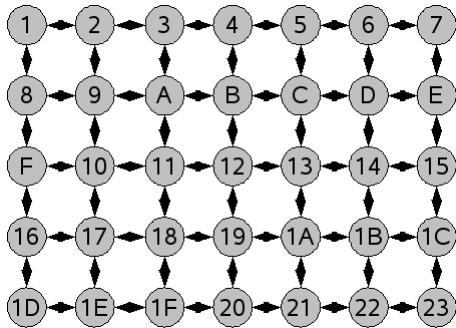
## 2. Compensazione dell'Offset



# *Average TimeSync* Caratteristiche

- Asincrono
  - no sequenza comunicazioni prestabilita
  - ok perdita pacchetti
- Distribuito:
  - no conoscenza topologia globale rete
  - no mantenimento gerarchia
- Leggero:
  - minima memoria richiesta
  - basso carico comunicazione
  - solo somme e moltiplicazioni
- Adattivo: derive possono cambiare
- Robusto ai ritardi: MAC layer time-stamping

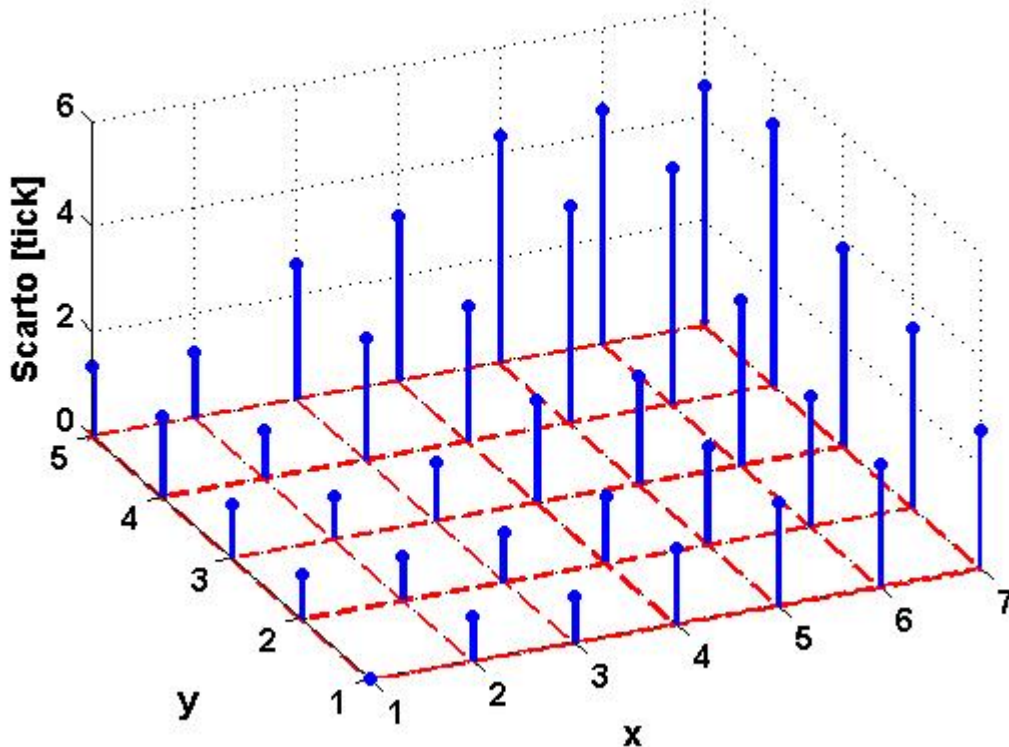
# ATS - Mesh di nodi 7x5



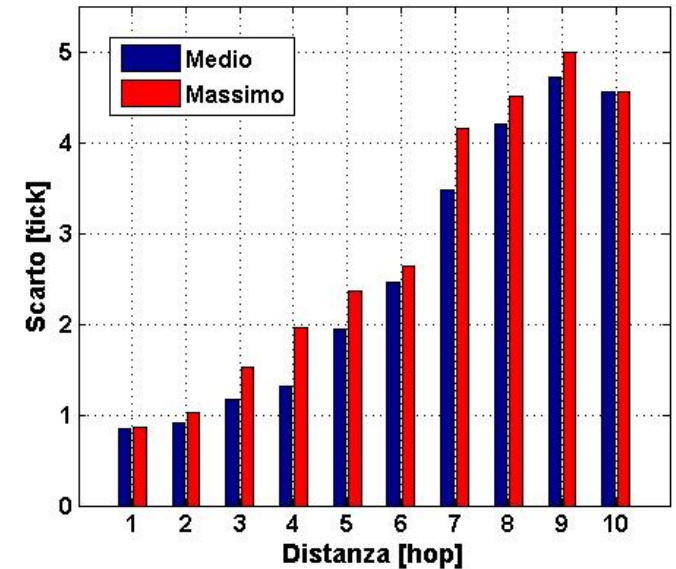
L'errore di sincronizzazione è mediamente 7.3 tick (circa 200  $\mu$ s)

# ATS – Errore Locale

## Scarto Semplice Medio Assoluto dei Tempi Globali dal Nodo 1



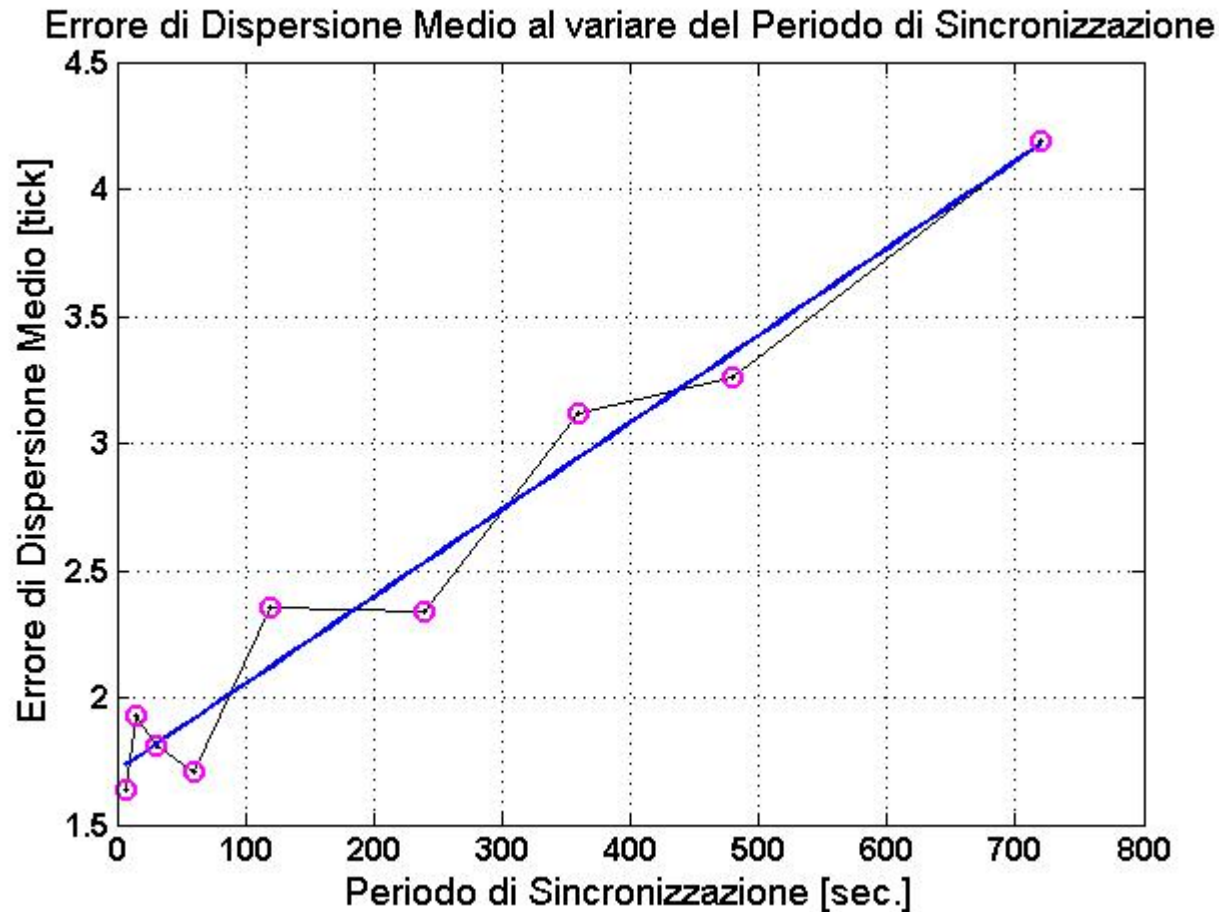
## Scarto Semplice Medio Assoluto dei Tempi Globali al variare della Distanza dal Nodo 1



Nodi vicini (in termini di hop) hanno un Errore di Sincronizzazione Inferiore

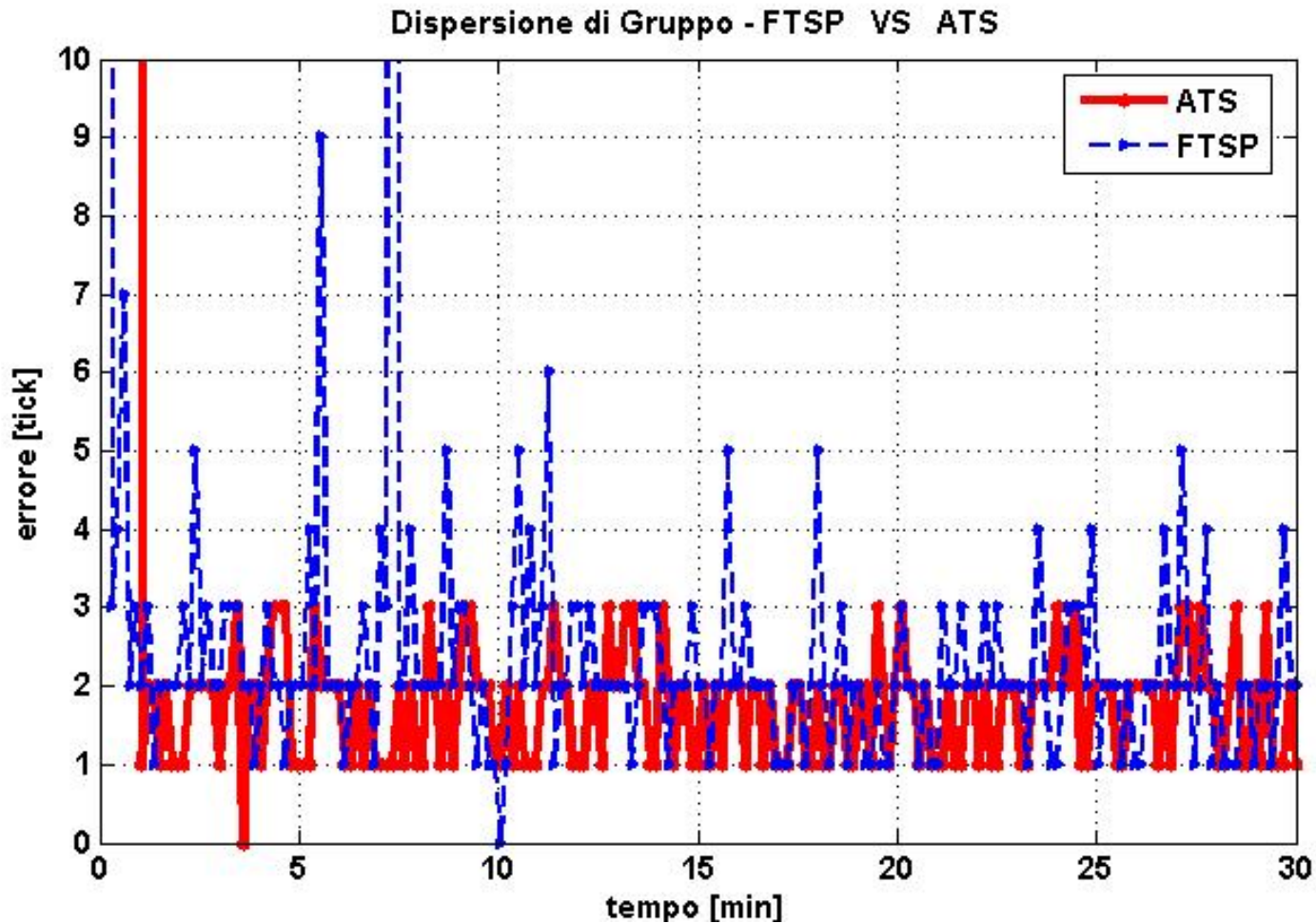


# Errore Vs Periodo di Sincronizzazione



Andamento lineare dell'Errore al variare del Periodo di Sincronizzazione

# ATS Vs FTSP



# Conclusioni

## *Average TimeSync:*

- Prestazioni comparabili con stato dell'arte
- Migliore efficienza energetica
- Migliore tolleranza ai guasti
- Adatto per reti di ogni dimensione e grandezza
- Errore locale indipendente da dimensione rete

# Prospettive

- Ottimizzazione dei pacchetti trasmessi (**piggybacking**)
- Analisi dei **pacchetti persi**
- Utilizzo del servizio di sincronizzazione per **Energy Scheduling** e **TDMA**



# Confronto con algoritmi precedenti

	Skew	Complessità	Canale	Memoria	Scalabilità	Topologia
RBS	Si	$(mn^2)$	$(m + mn)$	$O(n)$	Poca	Si (cluster)
TPSN	No	$(4m(n - 1))$	$(m + mn)$	$O(1)$	Buona	Si (albero)
TinySync	Si	$(4m(n - 1))$	$(m + mn)$	$O(1)$	Buona	Si (albero)
FTSP	Si	$(2mn)$	$(mn)$	$O(1)$	Ottima	Si (nodo radice)
RFA	No	$(2mn)$	$(m + mn)$	$O(n)$	Ottima	No
Solis et al	Si	$(2mn)$	$(mn(n - 1))$	$O(n)$	Ottima	No
ATS	Si	$(m(n + k))$	$(mn)$	$O(k)$	Ottima	No

Il risparmio del numero di pacchetti necessari per la sincronizzazione è del 50% su RBS, del 20% su TPSN e TS/MS.

Comparabile con FTSP, RFA, Solis et al. Algorithm, ma:

- FTSP ha una dipendenza dal nodo radice
- RFA non compensa la deriva
- Solis et al. inferiore per prestazioni