# Self-organized Synchronization in Wireless Network

**Jiang Yu and Olav Tirkkonen**

Department of Communications and Networking

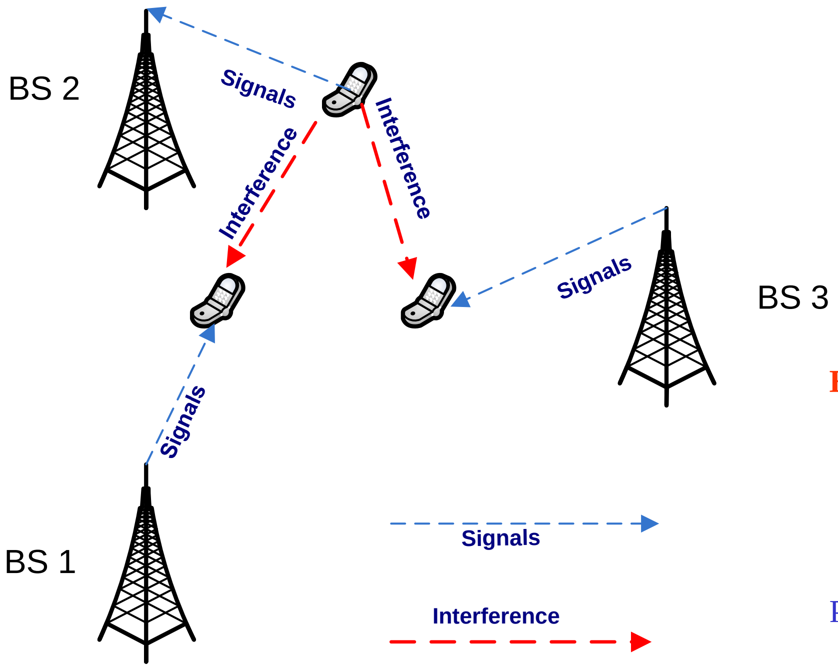
Helsinki University of Technology (TKK)

# Contents

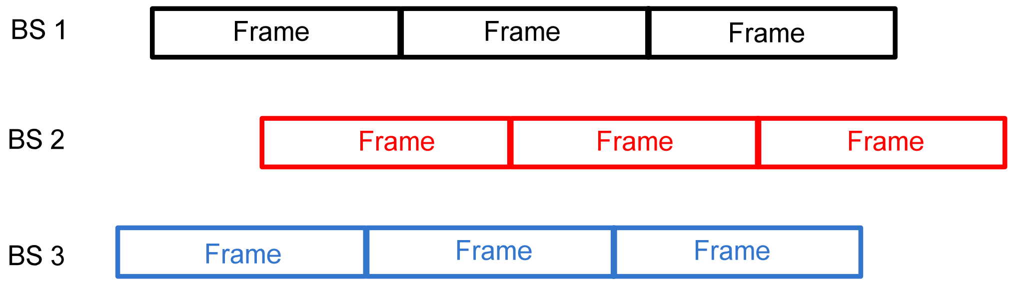
- Problems
- System model
- Firefly inspired algorithm
- Circular averaging with random selection method
- Simulation results
- Conclusions

# Problem

- The Local Area wireless network has the self-organized feature
  - No centralized controller which is able to decide everything
- Problem of no agreement for the timing of frame in TDD system
  - Each cell has different frame timing -> uplink and downlink signals are transmitted at the same time between adjacent cells -> high interference -> poor system performance
  - Resource information exchange, handover and resource allocation are difficult
- To solve this problem
  - All the frames of adjacent cells start almost at the same time
  - Need to find a self-organized synchronization method without centralized control




BS: Base Station




**Event Synchronization V.S. Time synchronization**

Time synchronization: all BSs/nodes share a common notion time that can be mapped back onto a real work clock

Event synchronization: all BSs/nodes agree on a time period start, which means that all frames start at the same time

Previous Synchronization methods for distributed network

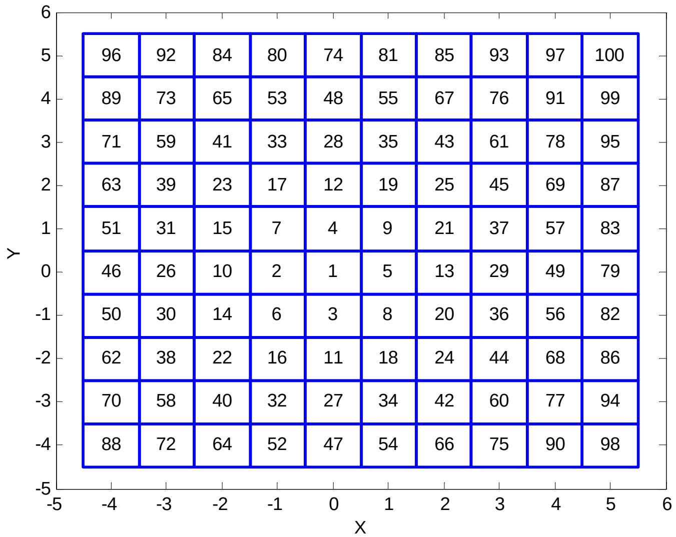
Network Time Protocol: not suitable for wireless network

GPS aided: need additional hardware and line of sight

Firefly-inspired algorithms

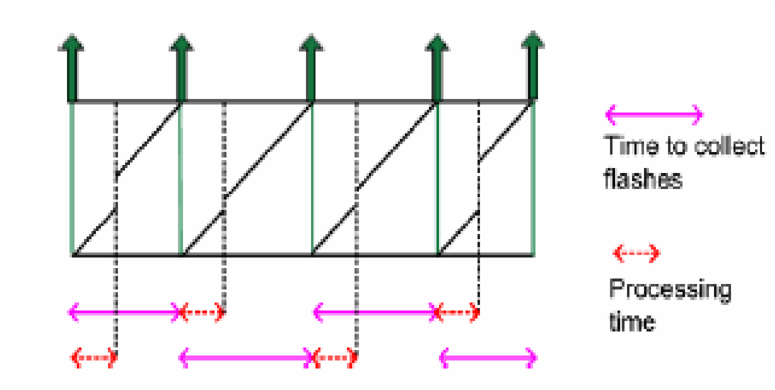
# System model

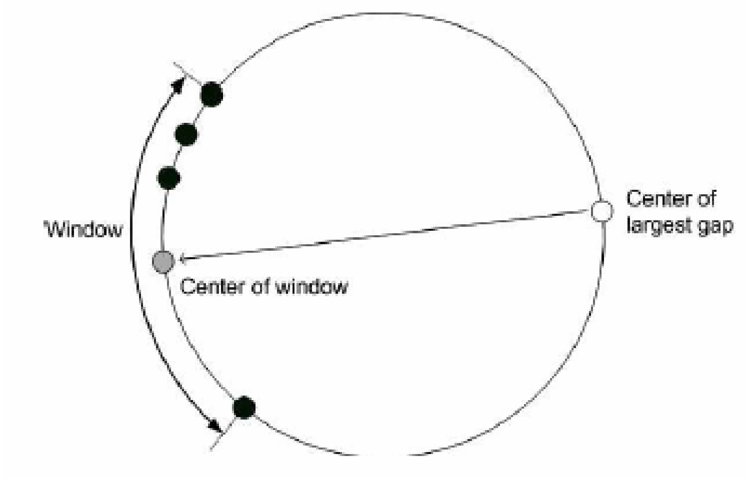
- Network model
  - Topology: regular square lattice
  - The set of BSs/nodes: $S$
  - Interaction with nearest neighbors, two way connection
  - Frame timing is modeled by phase variable, which is quantized into N time slots $\phi = 2\pi n / N, \quad n = 0, \ldots, N-1$
  - All BSs have the same period of frame $T$
  - The phase variable fulfills a full circle from 0 to $2\pi$, then jumps to 0
  - Reason to use time slot:
    - finite accuracy to measure the time difference among BSs
    - accurate property of synchronization algorithms
- Example



# System model

- Timing exchange protocol
  - information is exchanged by ’firing’
  - May contain BS ID or not
    - Firefly-inspired
    - Circular averaging with random selection
  - Time for information collecting
  - Processing time (much smaller than T)
- Smallest window covering a set of points



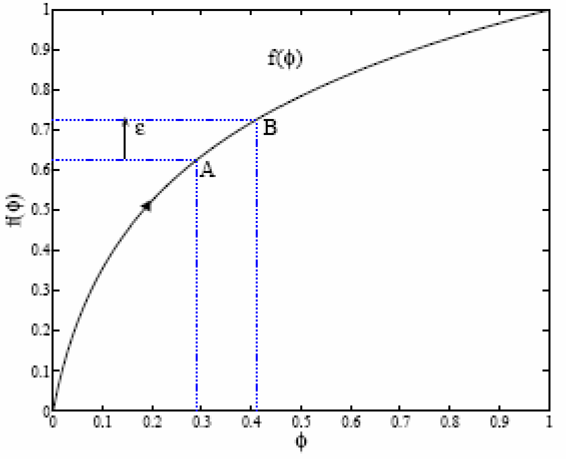



# Firefly-inspired algorithms

- Pulse coupled oscillator (PCO)
  - Mirollo and Strotagz proposed the model of connected PCO [1]
  - coupled differential equations

$$\dot{\phi}_i = \frac{1}{T} + \sum_{j \in \mathcal{N}_j} \left(\min\left[f^{-1}(f(\phi_i) + \epsilon),\ 2\pi\right] - \phi_i\right) \delta(\phi_j - 2\pi)$$

  - $\phi_i \in [0, 2\pi]$ is periodic
  - $T$ is period of $\phi_i$
  - $f(\phi)$ is firing function, $f > 0, \dot{f} > 0, and\ \ddot{f} < 0$
  - $\varepsilon > 0$ is a jump constant



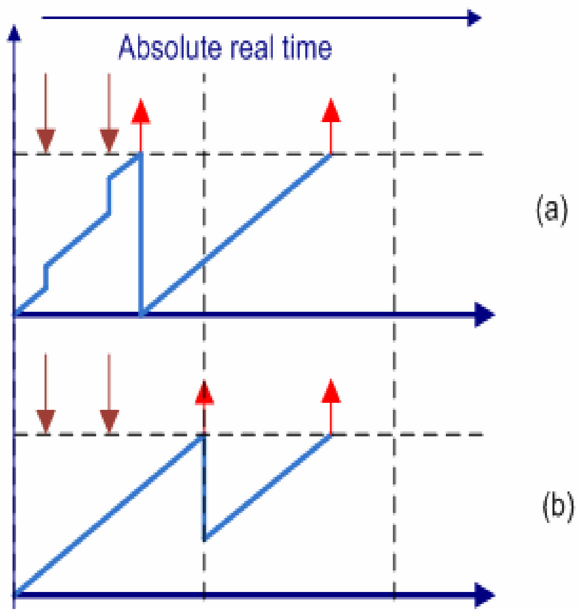

  - To simplify, $f(\phi) = \ln(\phi)$,

$$\Delta\phi = f^{-1}(f(\phi_0) + \varepsilon) - \phi_0 = (e^{\varepsilon} - 1)\phi_0$$

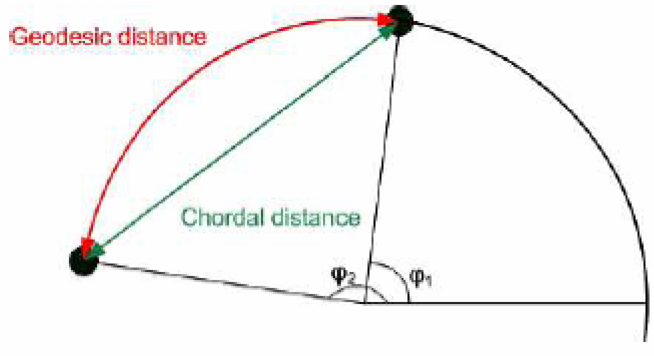
with Taylor expansion, $\Delta\phi = \varepsilon\phi_0$

# Firefly-inspired algorithms

- ReachBack Firefly Algorithm (RFA) [2]
  - problems of previous model:
    - fully coupled network
    - no delay of firing information
  - RFA:
    - BS will not jump immediately after firing of its neighbor
    - collect all firings from previous time period
    - react all at once
    - can be implemented for fully coupled or only nearest connected network
  - Select $\varepsilon$
    - too large: 'overshoot' , preventing convergence
    - too small: speed of convergence is very slow
    - need to test to find the best choice



# Circular averaging with random selection method




- Figures of merit
  - Metrics on the circle
    - Geodesic distance
    - Chordal distance
  - Considered norms of distance vector
    - one-norm (mean) $\|d\|_1 = \frac{1}{M}\sum_{j=1}^{M} d_j$
    - two-norm (RMS) $\|d\|_2 = \sqrt{\mathbf{d}^{\mathbf{T}}\mathbf{d}/M}$
    - infinite-norm (max) $\|d\|_\infty = \max_j \left(d_j\right)$

Combine these 2 distances and 3 norms, we have a class of algorithms to average a set of circular numbers

# Circular averaging with random selection method

- Randomized method
  - Group timings of neighbors
    - presence of group: at least 2 neighbors have same timing
    - Example: [1 30 46 85] no group
      [1 30 30 85] 3 groups with size 1, 1 and 2
  - Allocate probabilities in terms of the size of group
  - Choose one group as new timing
- Circular averaging with random selection
  - if all neighbors have different timings (no group appears), use circular averaging; else use random group selection
  - choose randomly between circular averaging and pure randomized algorithm
- based on previous results, 6 algorithms are considered
  - Geodesic 2-norm with weighted group selection (Geo2R)
  - Geodesic inf-norm with weighted group selection (GeoInfR)
  - Chordal 2-norm with weighted group selection (Cho2R)
  - Random selection between Geodesic 2-norm and weighted group selection (RGeo2R): 90% for circular and 10% for group
  - Random selection between Geodesic inf-norm and weighted group selection (RGeoInfR)
  - Random selection between Chordal 2-norm and weighted group selection (RCho2R)

# Simulation results

- Performance of combined algorithms and RFA
  - 100 nodes, 4 neighbors or 8 neighbors scenario
  - time slots N=1000 and 1000 periods to update
  - use 4 windows to investigate the accuracy: 1, 10, 50 and 100
  - for comparison reason, RFA is also tested

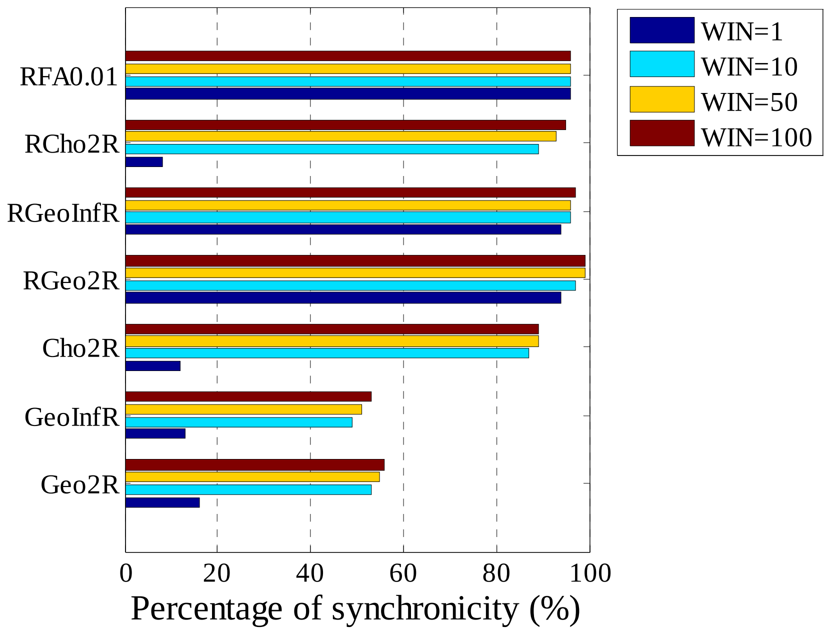
**Table 1. Rate of synchronicity for RFA.**

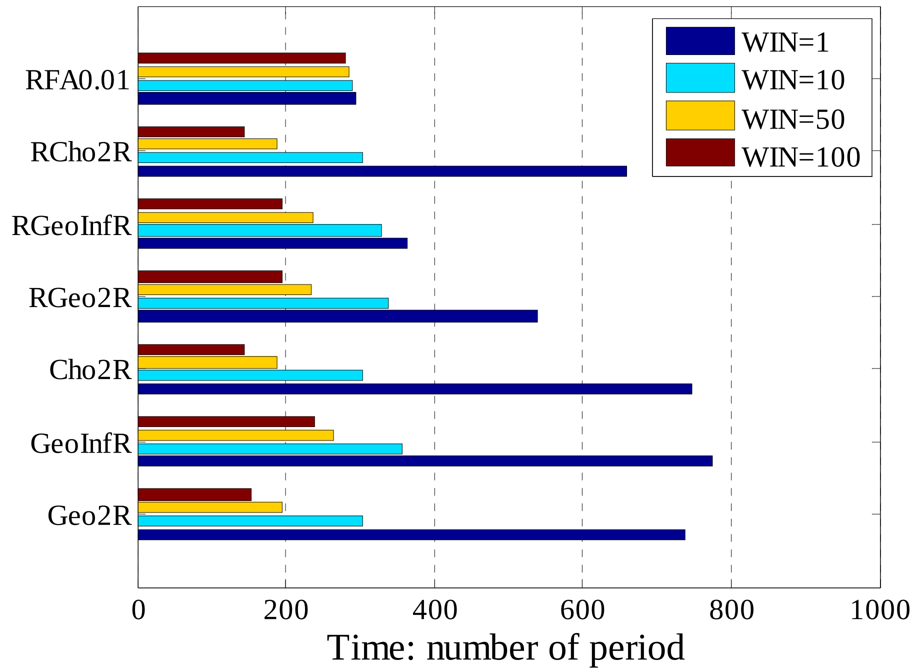
| Neighbors | $\epsilon$ | | | | |
|---|---|---|---|---|---|
| | 0.04 | 0.03 | 0.02 | 0.01 | 0.008 |
| 4 | 93% | 94% | 95% | 96% | 92% |
| 8 | 99% | 99% | 100% | 99% | 98% |

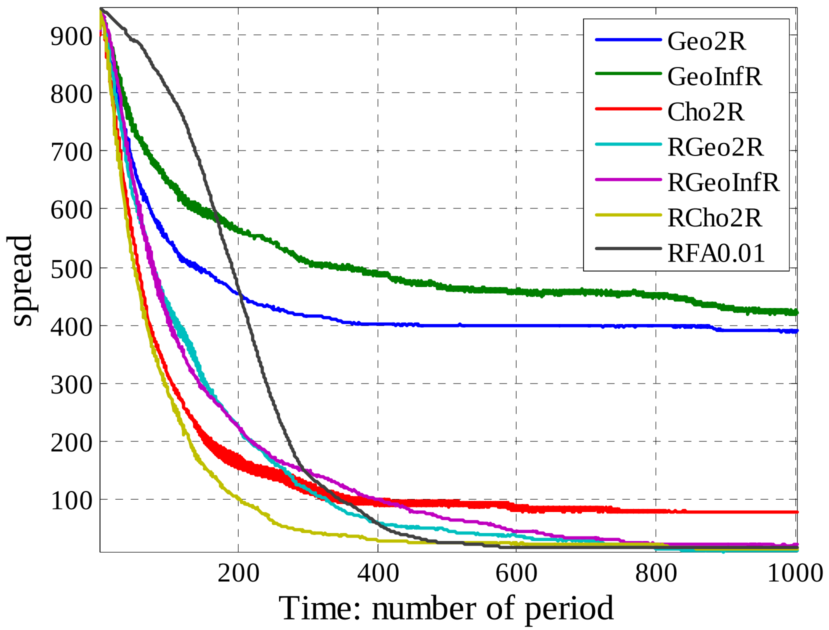
To get reasonable $\varepsilon$ value, the percentages of synchrony for RFA are tested.

For 4 neighbors scenario, $\varepsilon = 0.01$

For 8 neighbors scenario, $\varepsilon = 0.02$

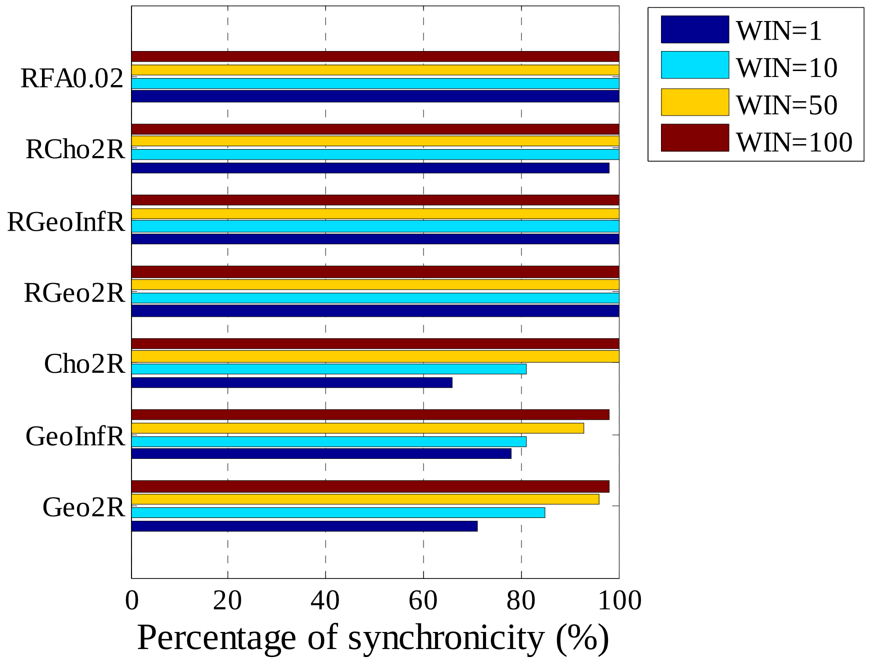


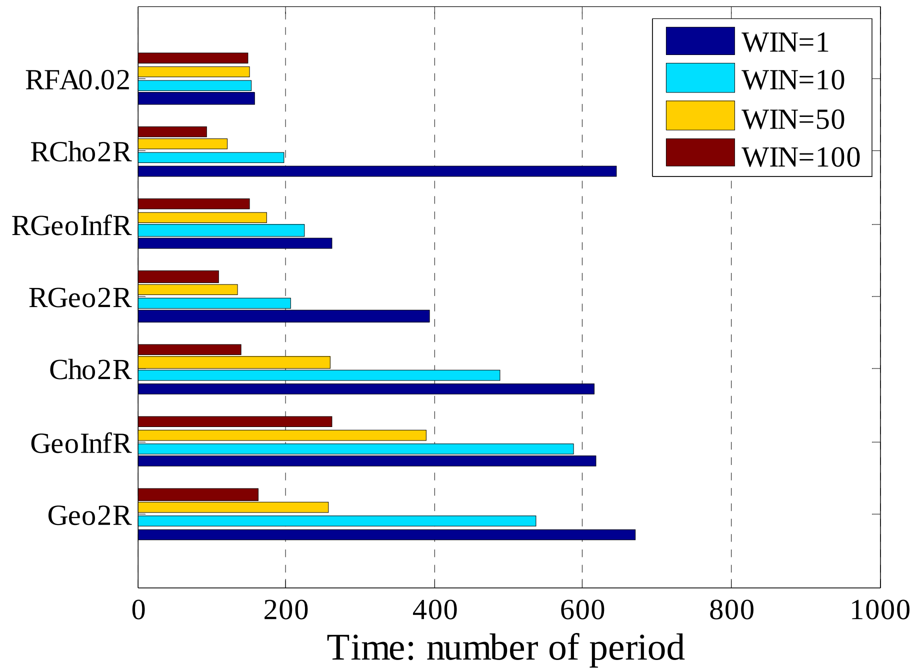



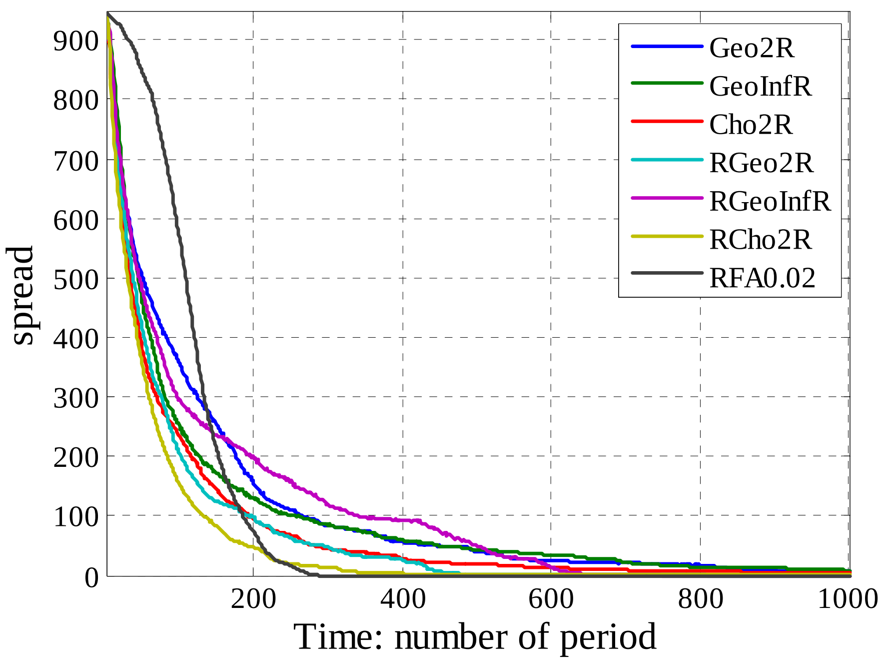




## 4 neighbors scenario

Proposed algorithms have better performance for gross accuracy: speed and percentage of synchrony



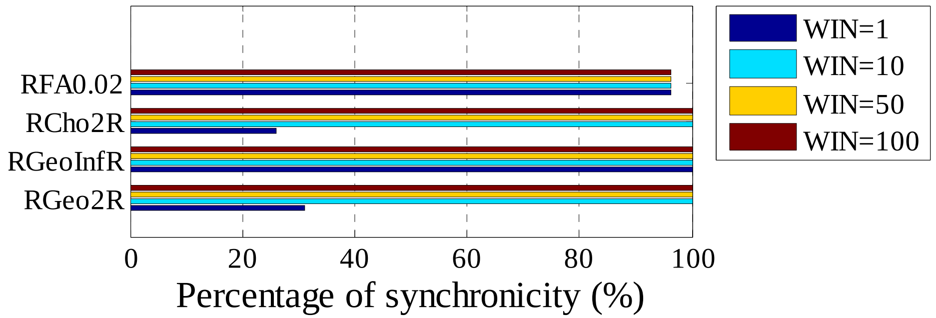



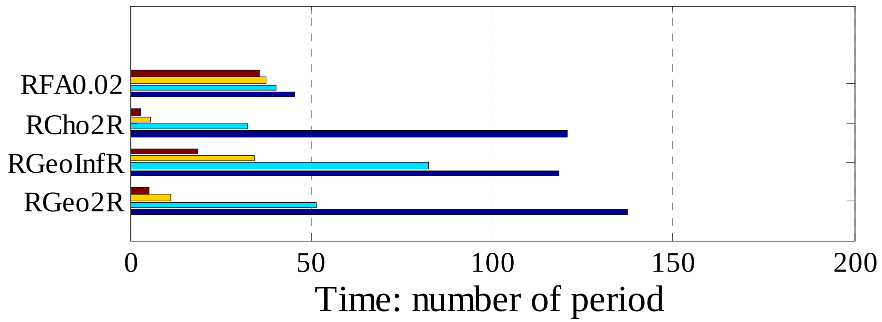




## 8 neighbors scenario

The more connected the network, the better of the performance

- Robustness against node addition
  - 8 neighbors, 100 BSs, N=1000
  - the number of joining BSs K is randomly selected from [1,10]. Before addition, the other (100-K) BSs are already synchronized
  - RCho2R, RGeoInfR, RGeo2R and RFA 0.02 are tested
  - 200 periods are used for updating the system




Within 200 periods updating, RFA can not converge 100%.

RGeoInfR has the best performance

# Conclusions

- For all algorithms, the more connected of the network, the better performance
- RFA needs to test jumping constant for different scenario or different topology
- The proposed circular averaging with random selection algorithms have better performance with gross accuracy (converge quickly)
- When new BSs join the system, the proposed algorithms are more robust
- The RFA may disturb all BSs in the system, while our algorithms just disturb neighbors
- RFA could be more sensitive to delay and missing of ’firing’ information, while the proposed algorithms can just use the old information.
- Multiple avenues to improve the circular averaging with random selection algorithms can be seen: the probability to use random selection and the probability to choose each group can be optimized.

# Reference

[1] R. E. Mirollo and S. H. Strogatz. Synchronization of pulsecoupled biological oscillators. *SIAM J. Appl. Math.*, 50(6):1645–1662, Dec. 1990.

[2] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proc. SensSys*, 2005.