# Time Synchronization in a Network of Bluetooth Low Energy Beacons

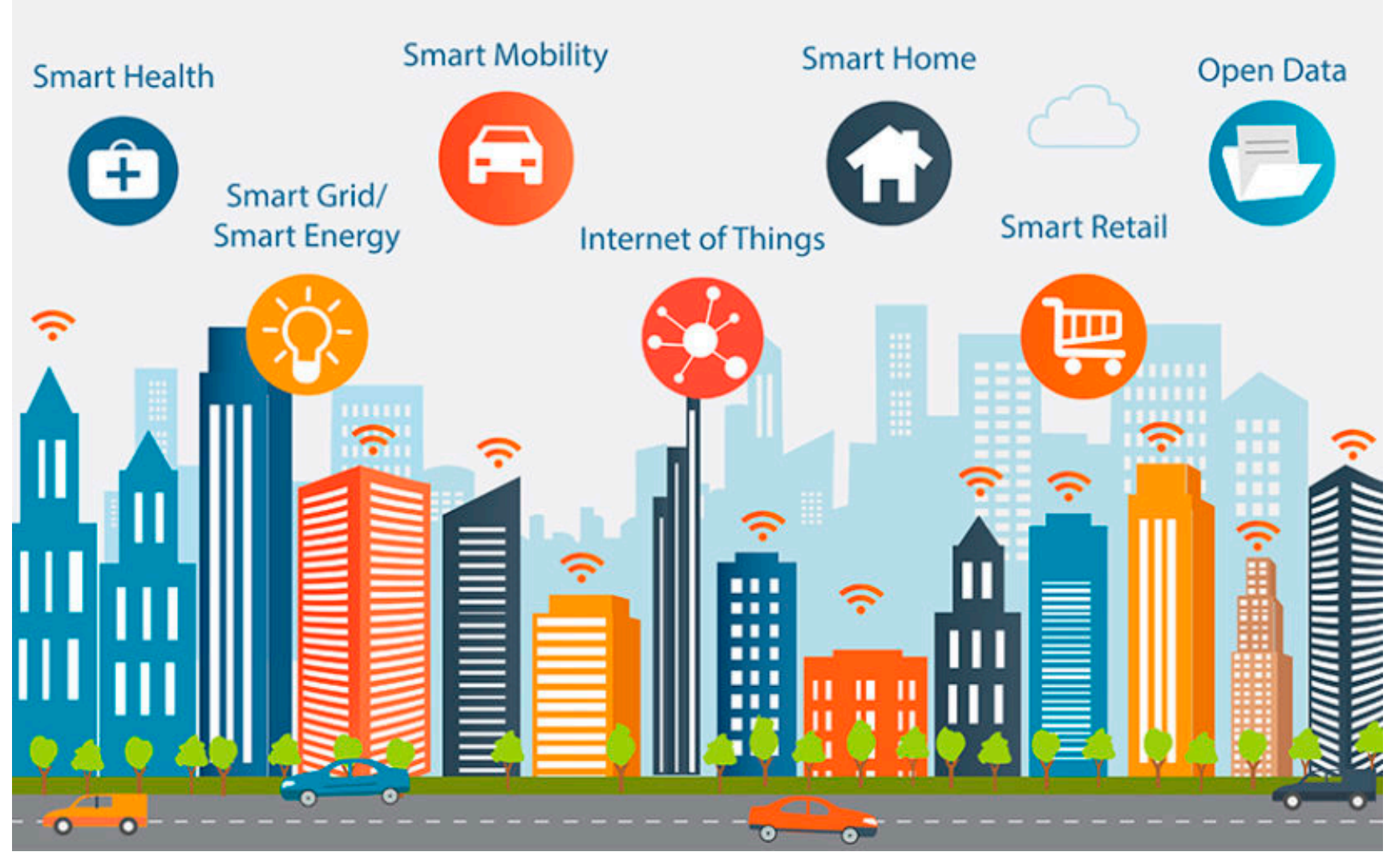
Farzad Asgarian, and Khalil Najafi

*Electrical Engineering & Computer Science, University of Michigan, Ann Arbor, MI, USA*

## Motivation / Application

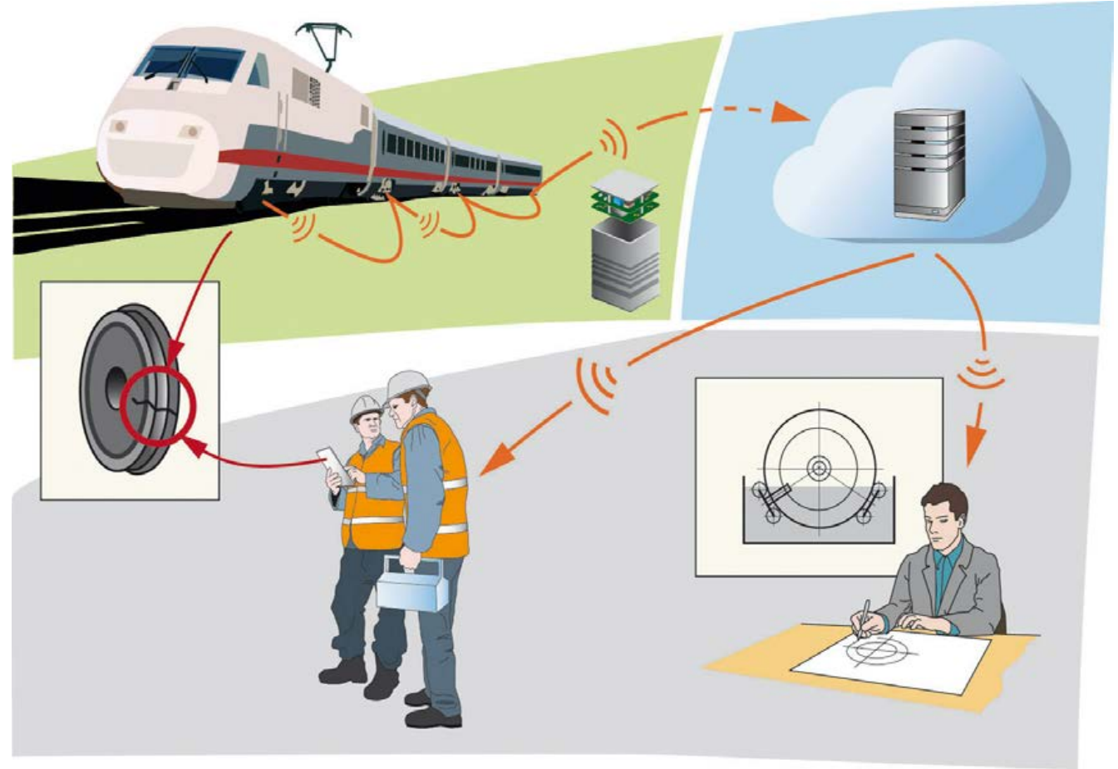
### ❖ *Wireless Sensor Networks* (WSN) are

- widely used as a tool for monitoring the physical world and human body
- expected to be integrated into the Internet of Things (IoT)



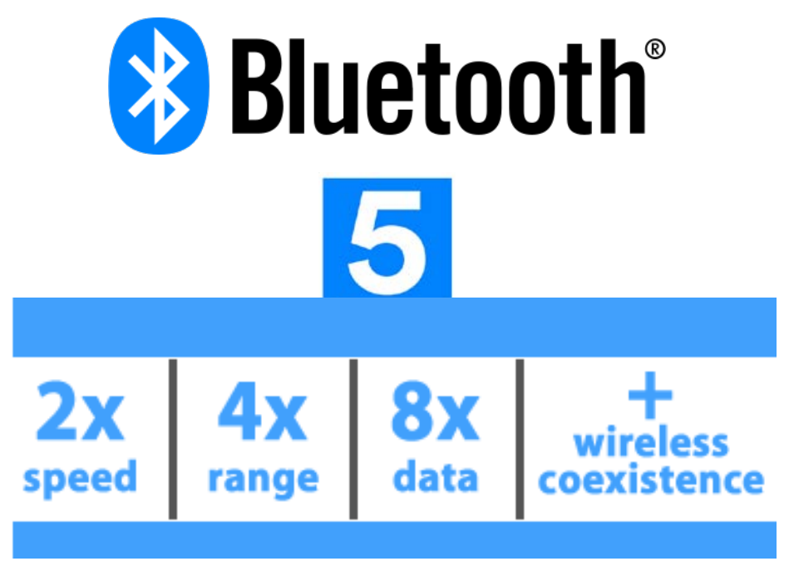
### ❖ *Time Synchronization*

- is a vital feature in many applications such as health and usage monitoring systems (HUMS)
- can improve scalability and efficiency



### ❖ *Bluetooth Low Energy* (BLE) is targeting IoT applications with multiple topologies:

- Beacon (Broadcast) → connectionless IoT
- Mesh → large scale networks



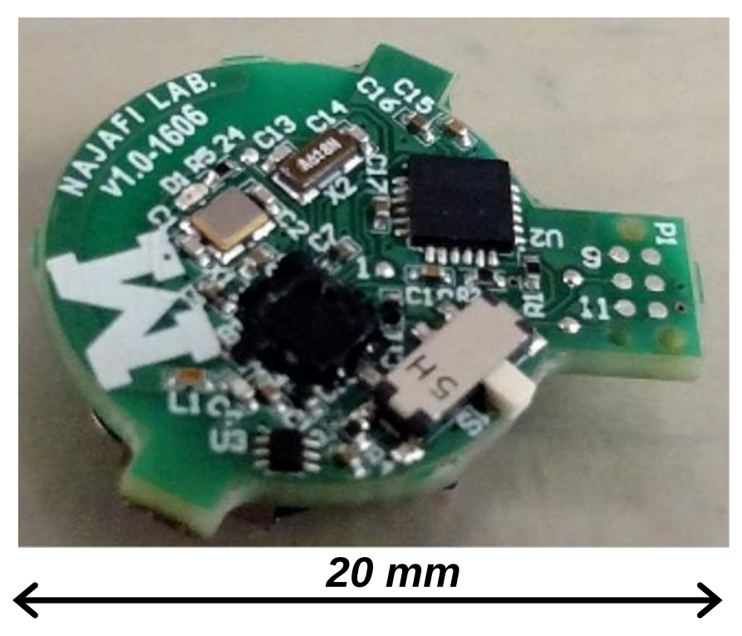
**☒ No Synchronization defined in BLE protocol**

- 10 µs error with 100 ms re-synchronization reported in *Cheapsync* [1-2]

## *BlueSync* Implementation

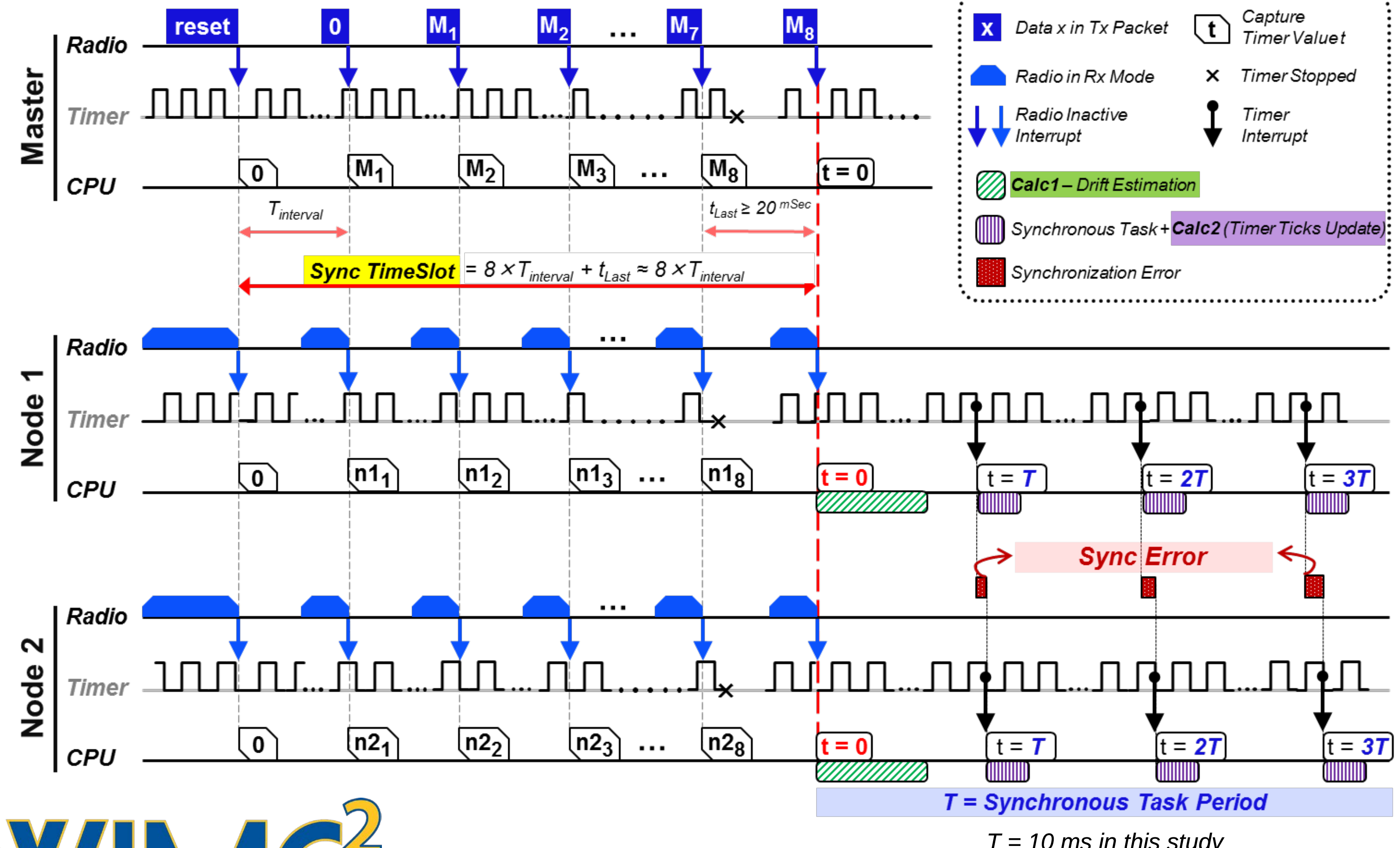
### ❖ *Target Platform*

- Sensor nodes with one of nRF5 BLE system-on-chips (SoCs)
- nRF51: Cortex-M0 16 MHz, 16-bit timers (16 MHz)
- nRF52: Cortex-M4 64 MHz, 32-bit timers (16 MHz)



| Challenges | Solutions |
|---|---|
| Single channel scanning | API Timeslot |
| Accurate Timestamping | PPI (Programmable Peripheral Interconnect) |
| Updating advertising data with timestamps | Delay sync |

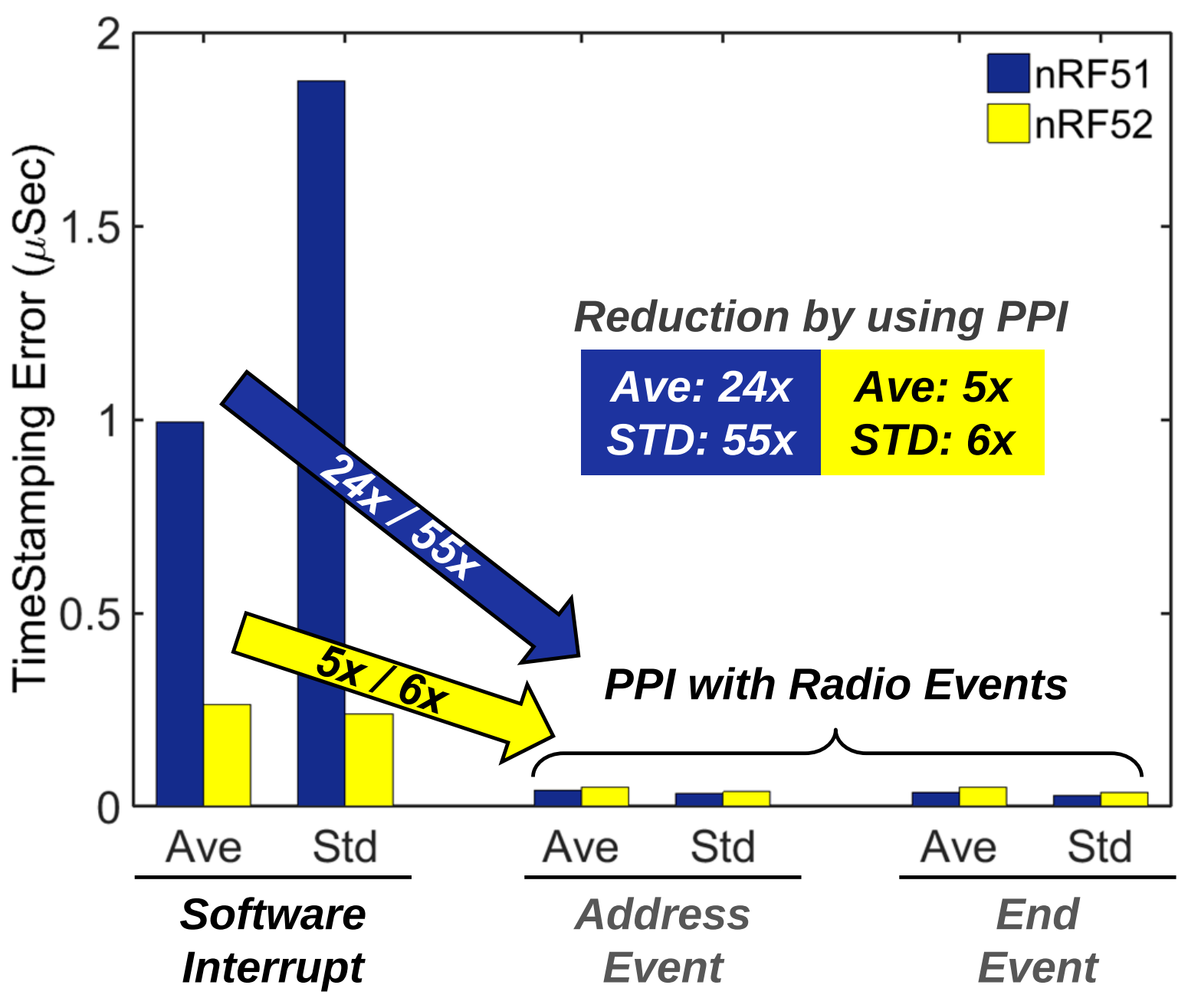
**BlueSync Protocol with 8 Synchronization Packets**



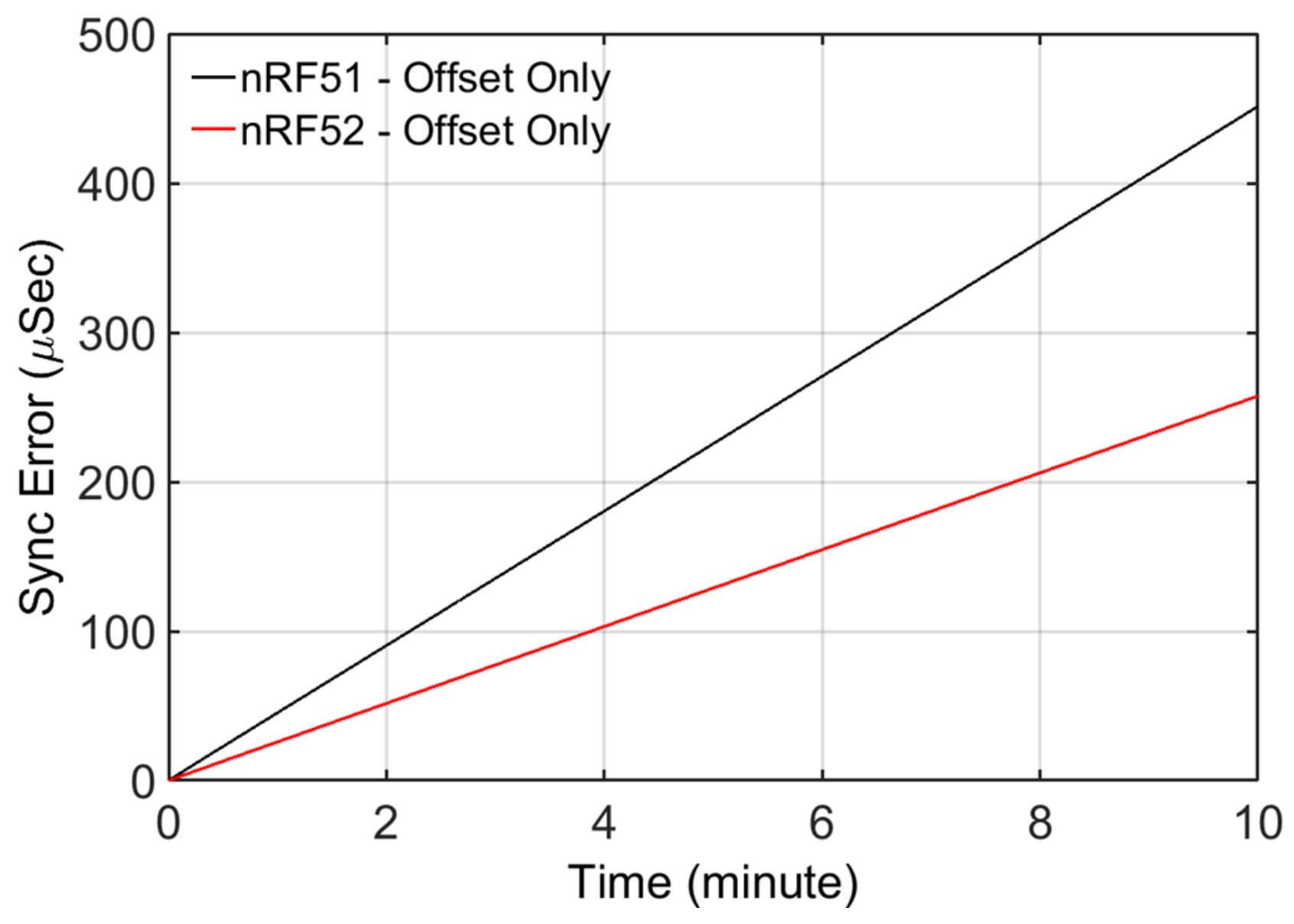
*T = 10 ms in this study*

## Results

### ❖ *Timestamping Error with 16 MHz Clock*



### ❖ *Sync Error without Drift Management*



### ❖ *Applying Drift Estimation Techniques*

- Linear Regression (LR)

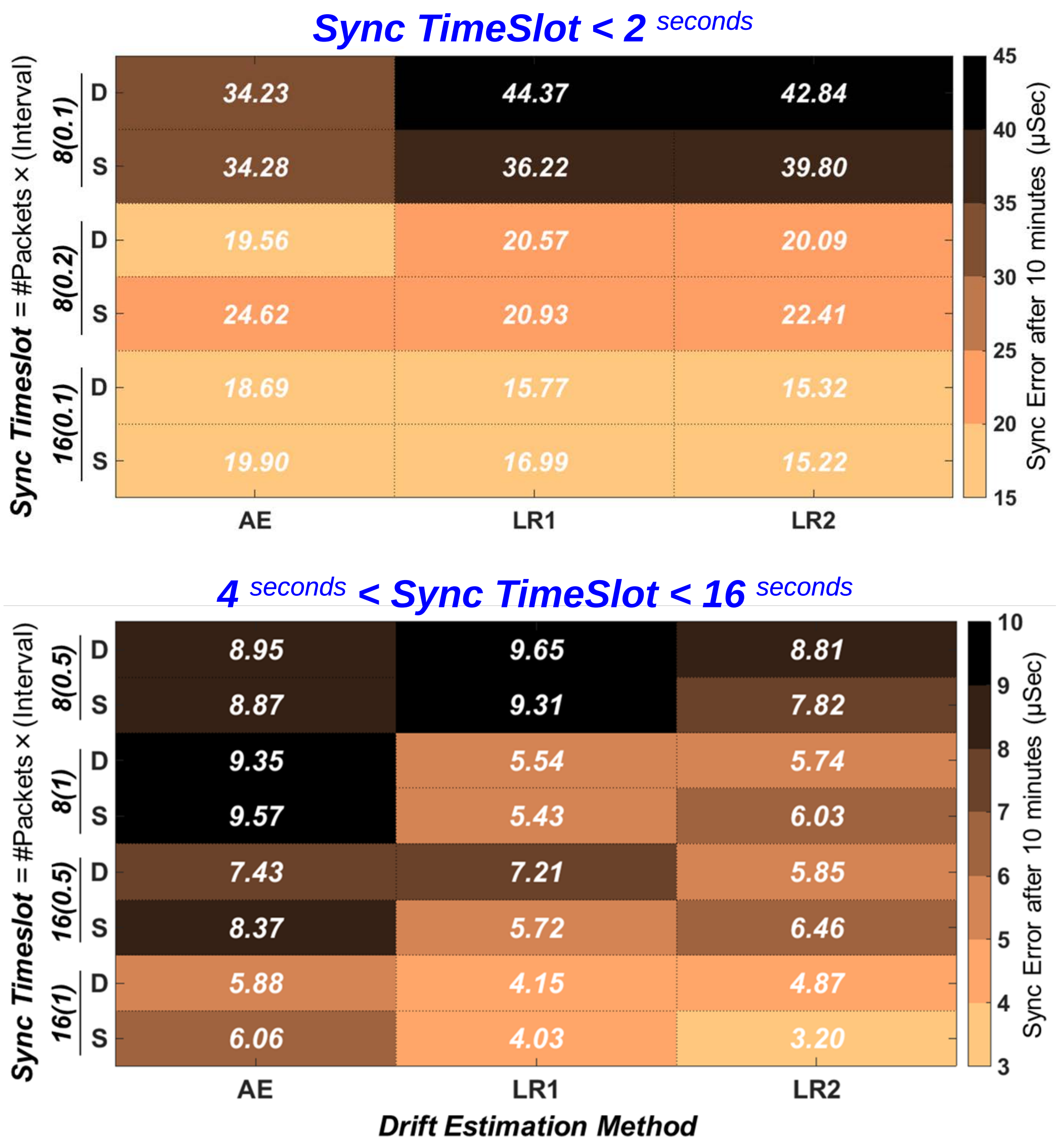
$$slope = \frac{n\sum_{i=1}^{n} M_i - \sum_i M_i \sum_i S_i}{n\sum_i {M_i}^2 - (\sum_i M_i)^2},$$

$$offset = \frac{\sum_i S_i \sum_i {M_i}^2 - \sum_i M_i \sum_i M_i S_i}{n\sum_i {M_i}^2 - (\sum_i M_i)^2}$$

- Average Error (AE)

$$C_{AE} = \frac{1}{n}\sum_{i=1}^{n} \frac{(S_i - S_{i-1}) - (M_i - M_{i-1})}{M_i - M_{i-1}}$$

- Results for Single (S) and Double (D) precision floating point calculations
- Measured every 10 ms
- No resynchronization during 10-minutes



*Sync TimeSlot < 2 seconds*

| Sync Timeslot |  | AE | LR1 | LR2 |
|---|---|---|---|---|
| 8(0.1) | D | 34.23 | 44.37 | 42.84 |
|  | S | 34.28 | 36.22 | 39.80 |
| 8(0.2) | D | 19.56 | 20.57 | 20.09 |
|  | S | 24.62 | 20.93 | 22.41 |
| 16(0.1) | D | 18.69 | 15.77 | 15.32 |
|  | S | 19.90 | 16.99 | 15.22 |

*4 seconds < Sync TimeSlot < 16 seconds*

| Sync Timeslot |  | AE | LR1 | LR2 |
|---|---|---|---|---|
| 8(0.5) | D | 8.95 | 9.65 | 8.81 |
|  | S | 8.87 | 9.31 | 7.82 |
| 8(1) | D | 9.35 | 5.54 | 5.74 |
|  | S | 9.57 | 5.43 | 6.03 |
| 16(0.5) | D | 7.43 | 7.21 | 5.85 |
|  | S | 8.37 | 5.72 | 6.46 |
| 16(1) | D | 5.88 | 4.15 | 4.87 |
|  | S | 6.06 | 4.03 | 3.20 |

**100x smaller error compared to [1-2] and**
**10x better than FTSP [3] with same measuring conditions**

## Figure of Merit (FOM)

$$FOM = \frac{t}{error} \times \frac{1}{\underbrace{nT_P}_{\text{Sync Timeslot}}} \times \frac{1}{\underbrace{T_1 + N \times T_2}_{\text{Calc times}}}, N = \frac{t}{T_{Task}}$$

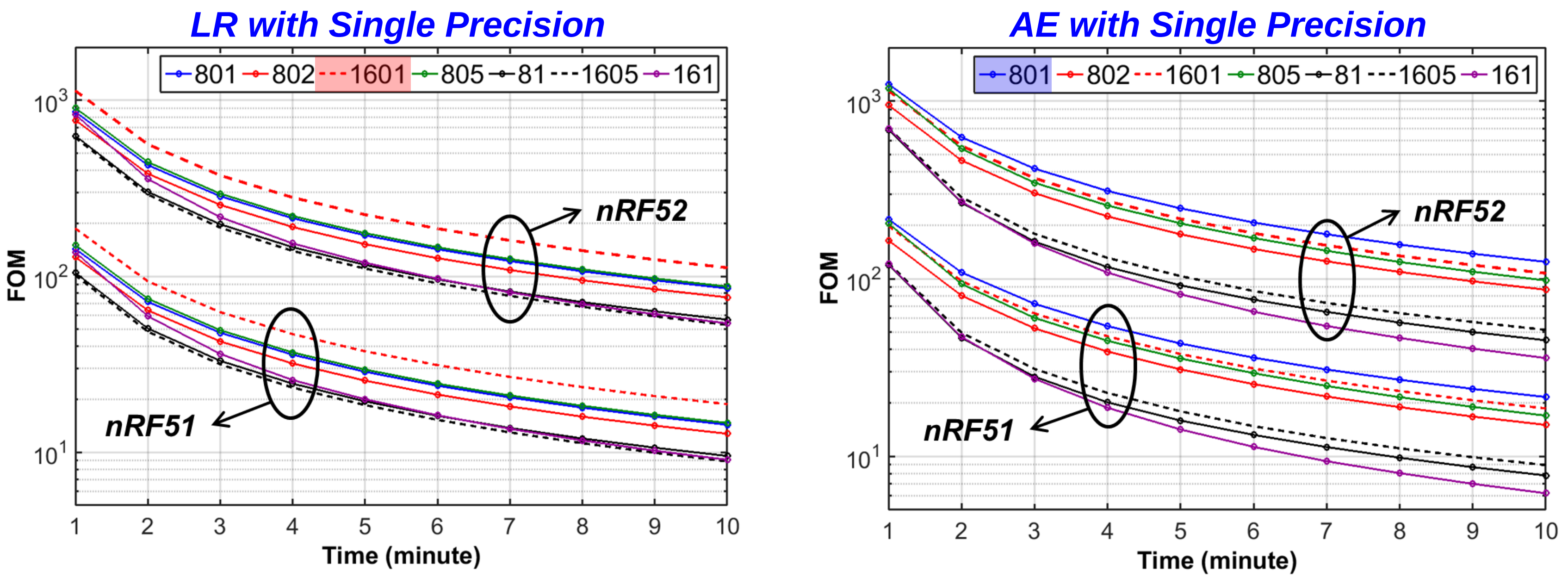
*error*: average maximum synchronization error after $t$ seconds

*n*: # of synchronization packets with interval $T_P$

*N:* # of interrupts during $t$

$T_1/T_2$: times require for error estimation / ticks adjustments.

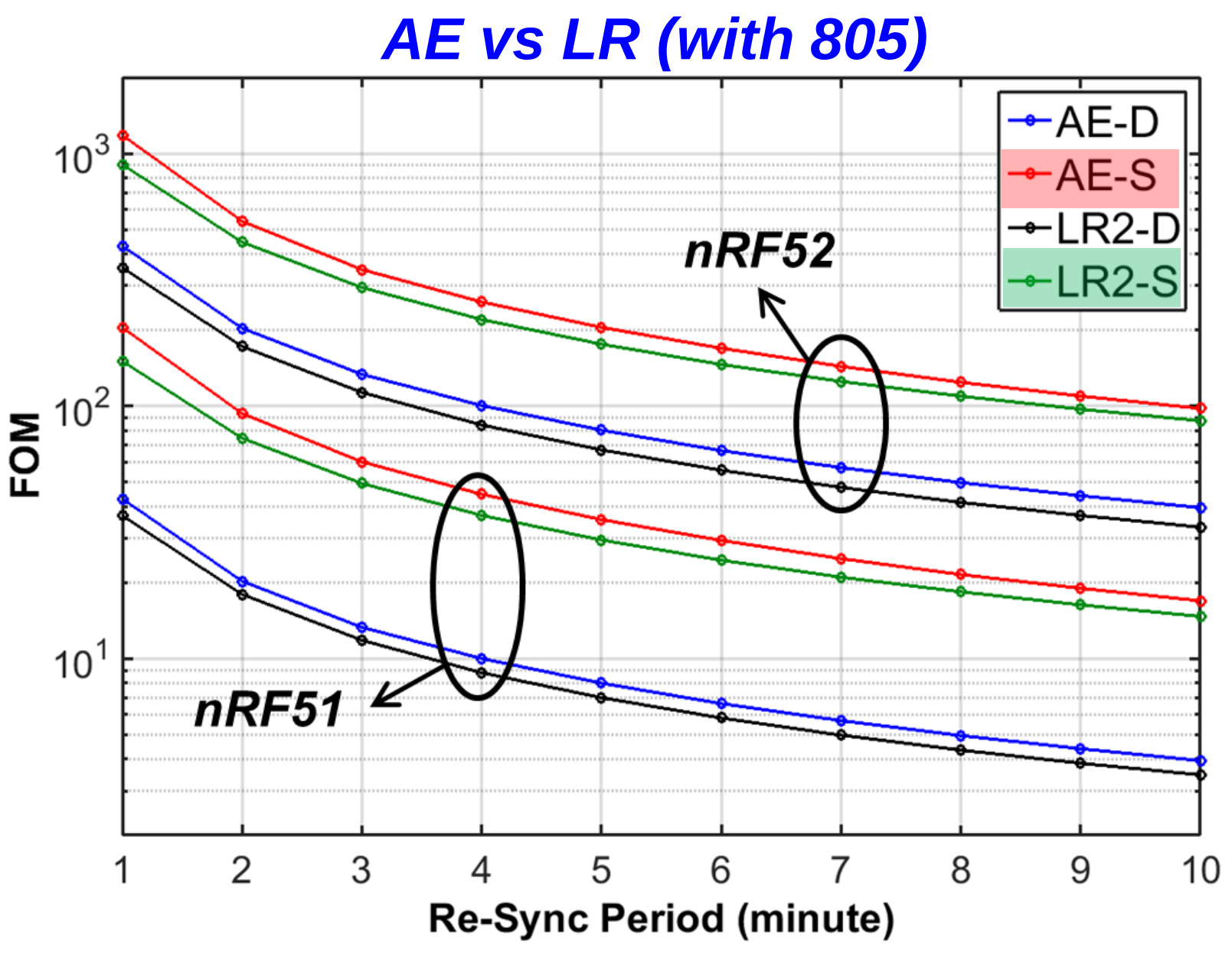
### ❖ *Sync Speed & Accuracy tradeoff*



- 1601 (LR) and 801 (AE) have the highest FOM

### ❖ *Methods & Accuracy tradeoff*

- Results with single precision have higher FOMs in both methods
- AE has slightly higher FOMs with both floating point types



## References

[1] Sabarish Sridhar, Prasant Misra, Gurinder Singh Gill, and Jay Warrior. 2016. Cheepsync: a time synchronization service for resource constrained bluetooth le advertisers. IEEE Communications Magazine 54, 1 (2016), 136–143.

[2] Sabarish Sridhar, Prasant Misra, and Jay Warrior. 2015. CheepSync: a time synchronization service for resource constrained bluetooth low energy advertisers. In Proceedings of the 14th International Conference on Information Processing in Sensor Networks. ACM, 364–365.

[3] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. 2004. The flooding time synchronization protocol. In Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM, 39–49.