Beacon Alignment Synchronization for Increased Scalability of IEEE 802.11s Light Sleep Mode

Marco Porsch, Florian Schlegel

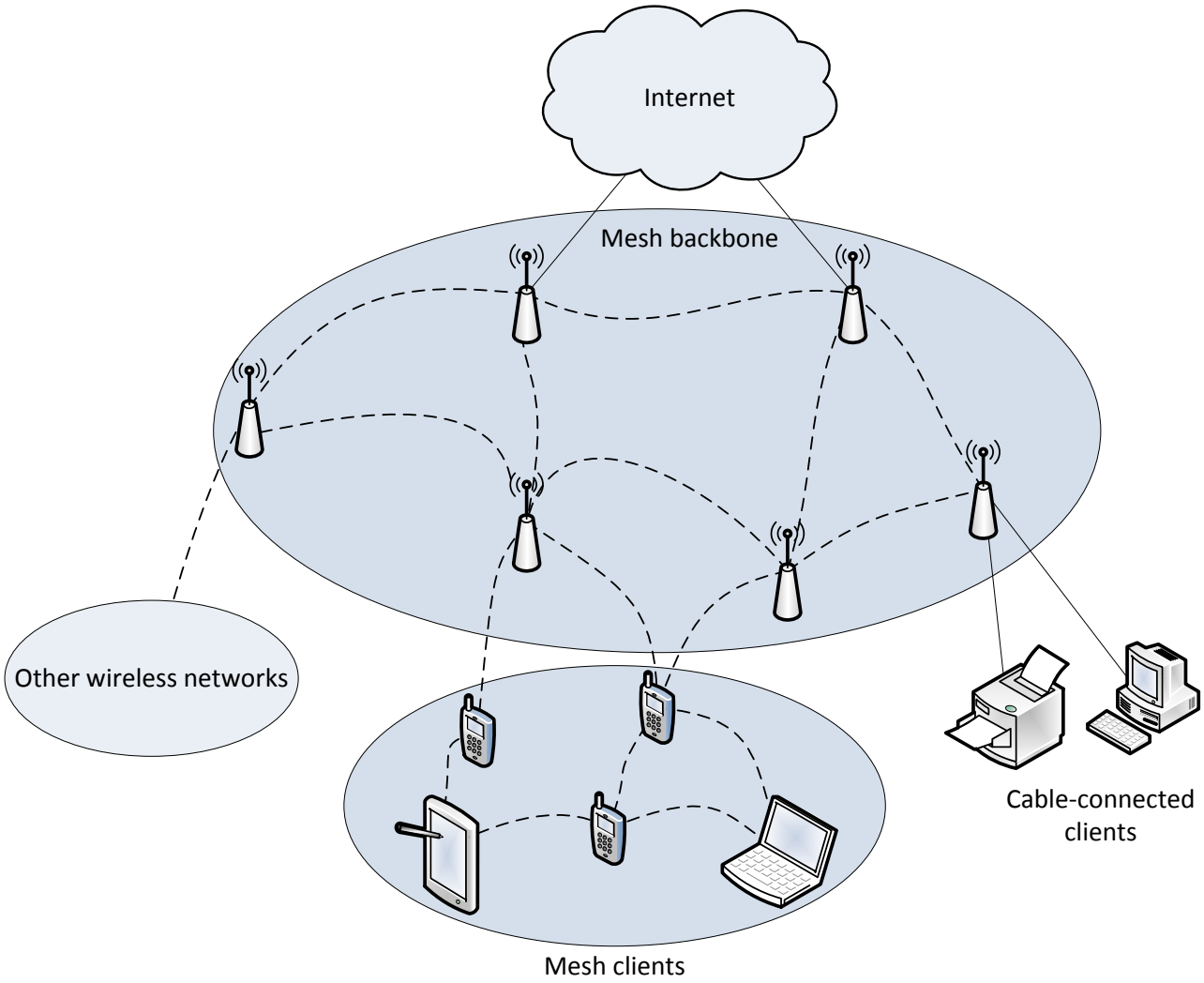
Technische Universität Chemnitz

marco.porsch@etit.tu-chemnitz.de
florian.schlegel@etit.tu-chemnitz.de

Overview

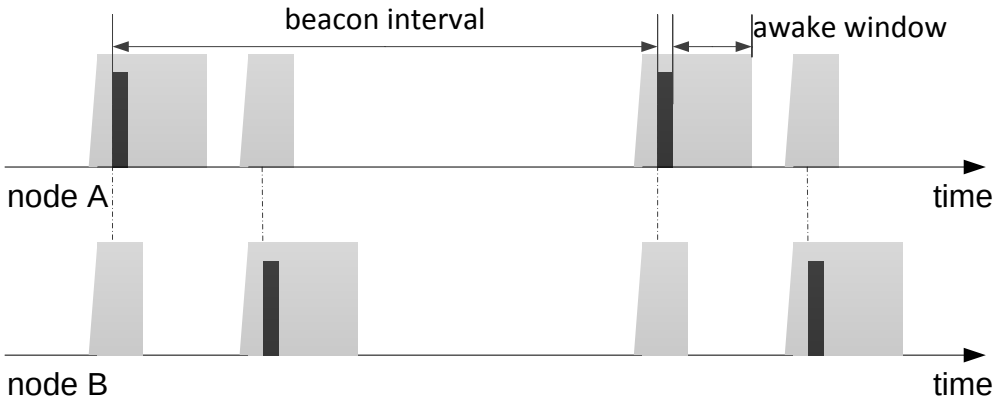
- IEEE 802.11s Synchronization and Power Save
- Beacon Alignment Synchronization
- Performance Evaluation
- Summary

IEEE 802.11s Synchronization and Power Save



IEEE 802.11s Synchronization and Power Save

- Active: no doze, no energy saving
- Deep sleep: doze state, except for sending beacon frame and following awake window
- Light sleep: nodes additionally wake up to receive peer's beacons

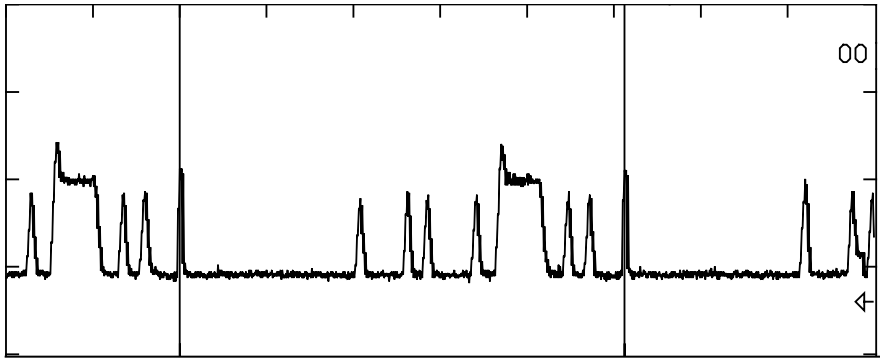


IEEE 802.11s Synchronization and Power Save

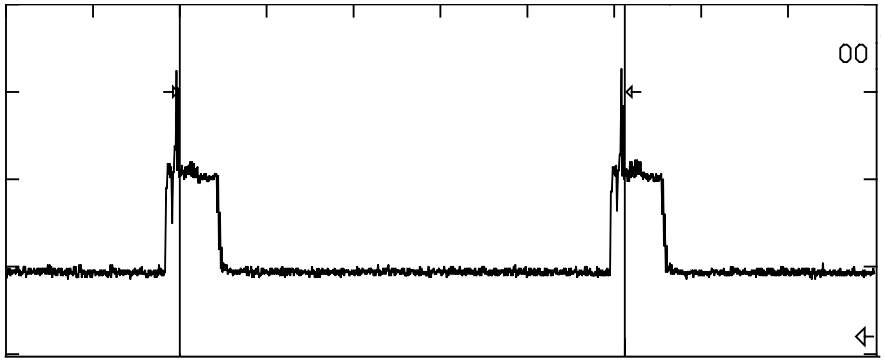
- Extensible framework; synchronization algorithm as “plugin”
- IEEE 802.11s standard defines “neighbor offset synchronization” as default:
  - does not equalize/adapt the TSF timers (as managed mode and ad hoc mode do)
  - keeps arbitrary TSF/TBTT offsets between nodes constant

Beacon Alignment Synchronization

- Problem: each wakeup for peer beacons in light sleep incorporates overhead
- Solution: shift all TBTT together
  → wake up only once for all peers



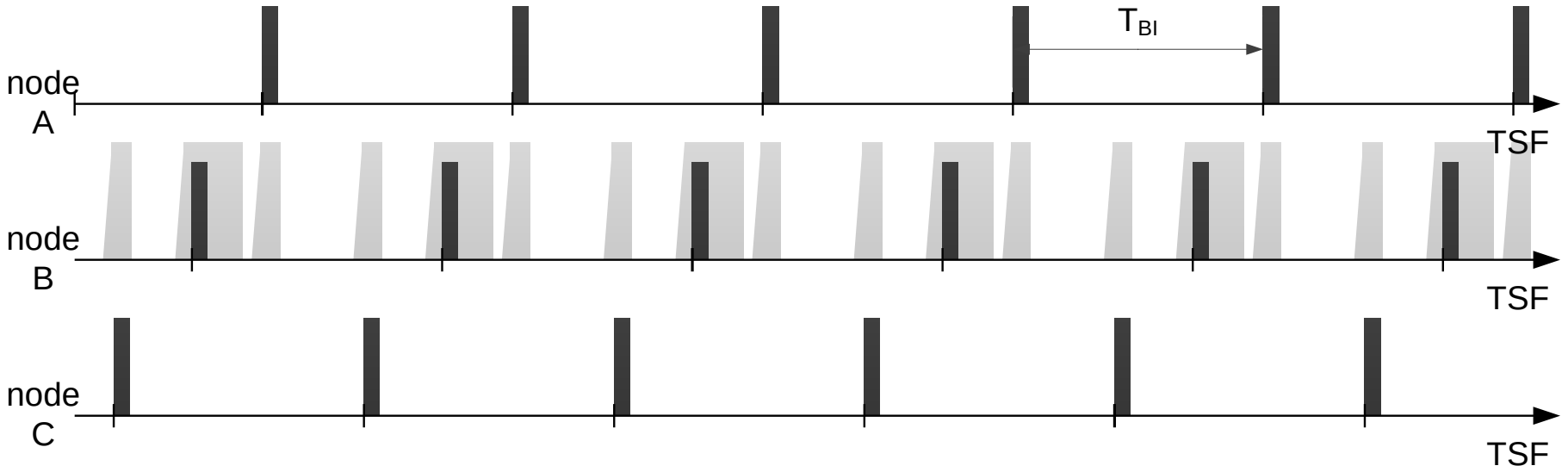
before



after

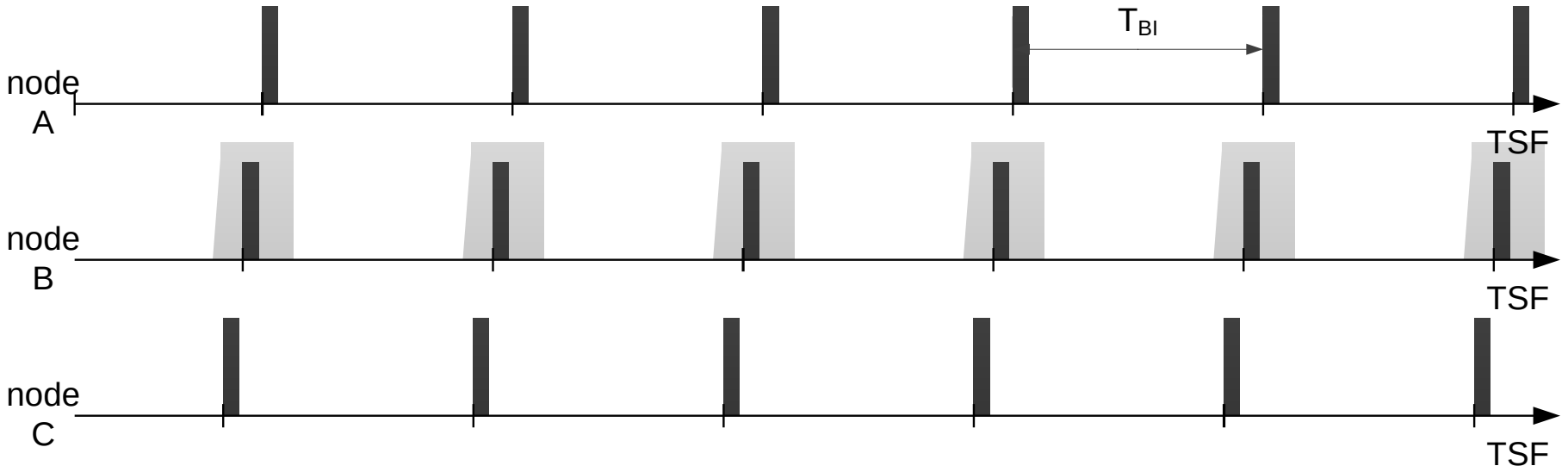
Beacon Alignment Synchronization

- Example scenario before beacon alignment starts:
  - peers A, B, C with common beacon interval
  - B in light sleep mode towards both peers



Beacon Alignment Synchronization

- Example scenario when beacon alignment procedure is finished

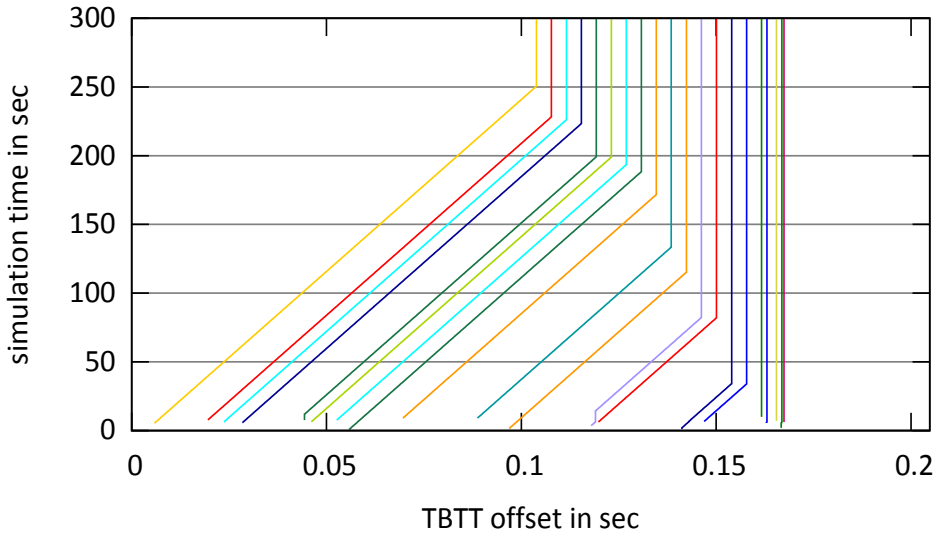


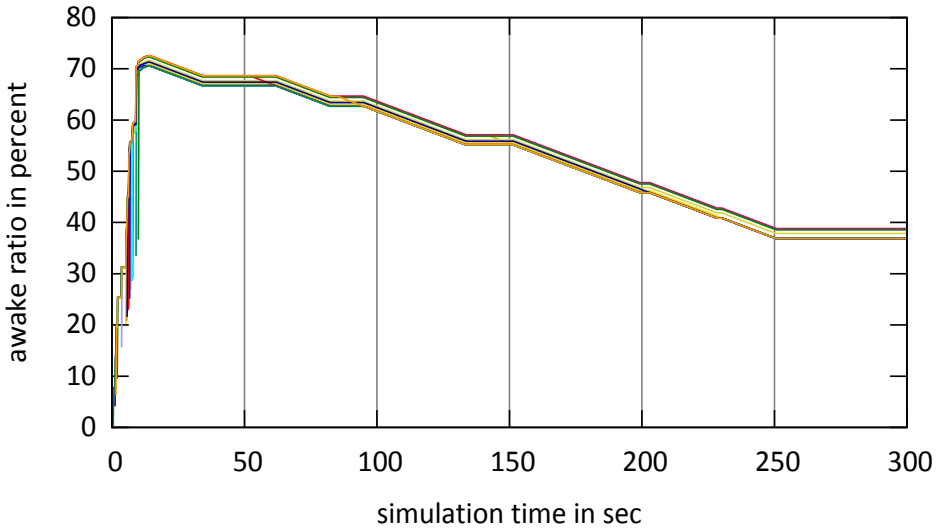
Performance Evaluation

- Beacon alignment algorithm implemented in network simulation framework “ns-3”
- Simulation parameters extracted from testbed measurements
- All peer nodes in light sleep mode towards each other

Performance Evaluation

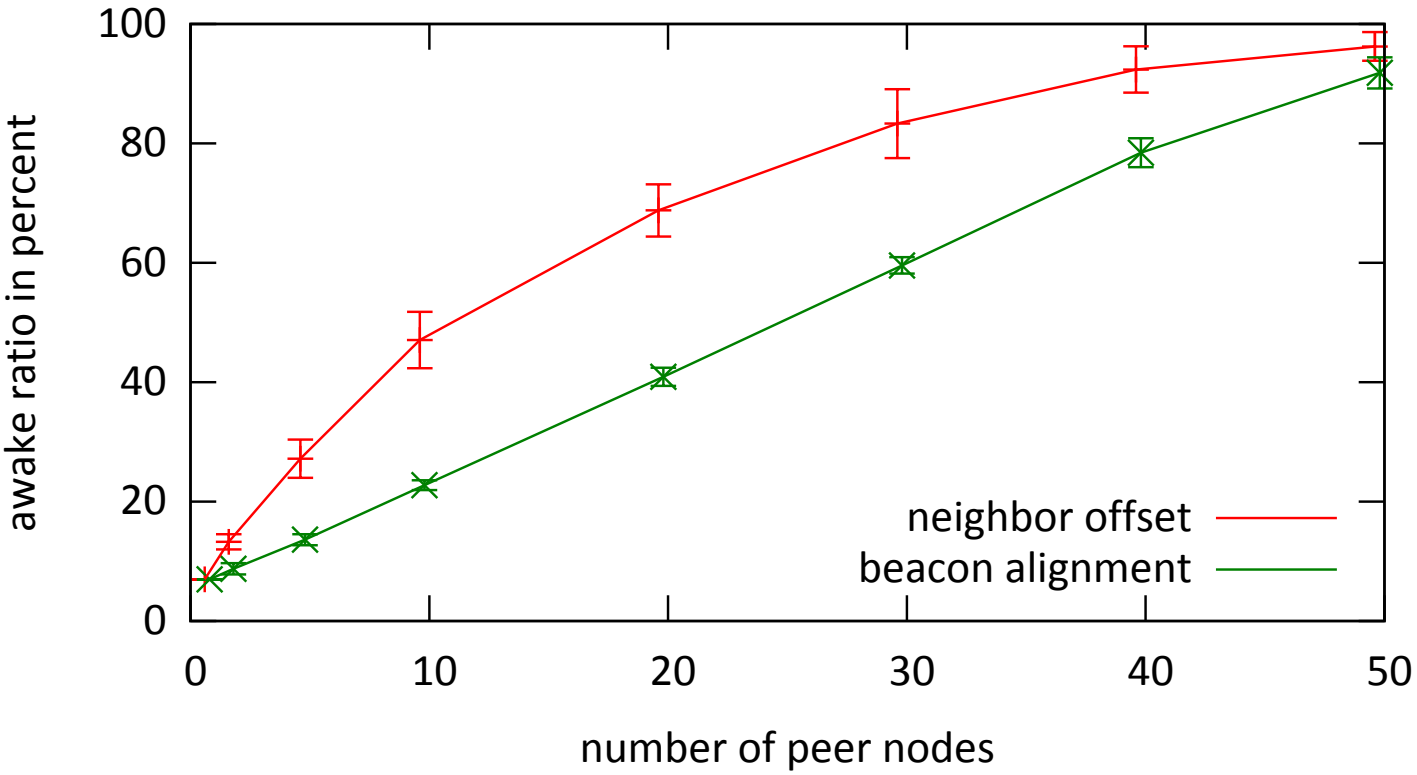
- Results of a single simulation run with 20 nodes
- Awake ratio reduced by more than 30%





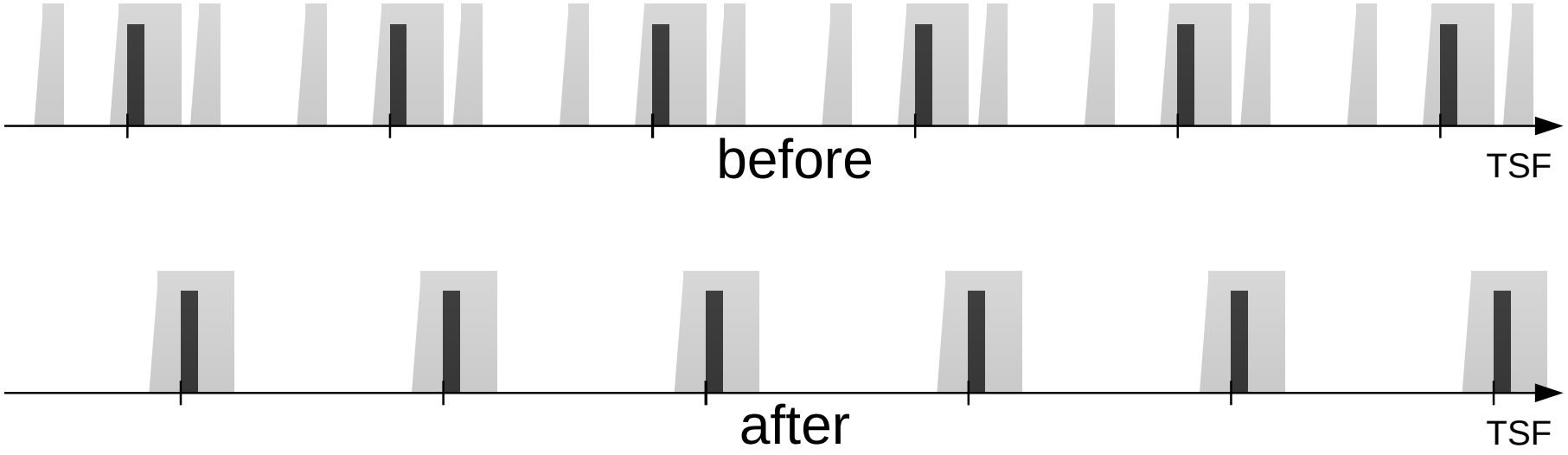
Performance Evaluation

- Simulation results with variable number of peer nodes
- Averaging over 500 node simulation runs



Summary

- Beacon alignment synchronization
  - significantly reduces awake ratio of nodes
  - orders arbitrary TBTT distribution
    → reduces variability

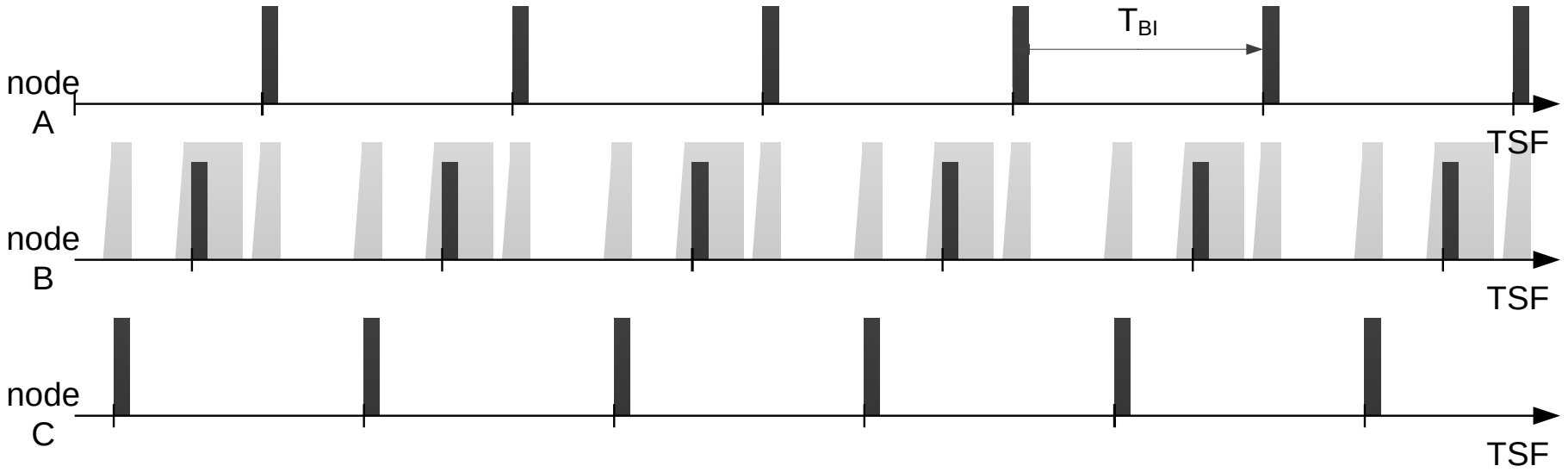


Questions?

Backup

Neighbor Offset Synchronization

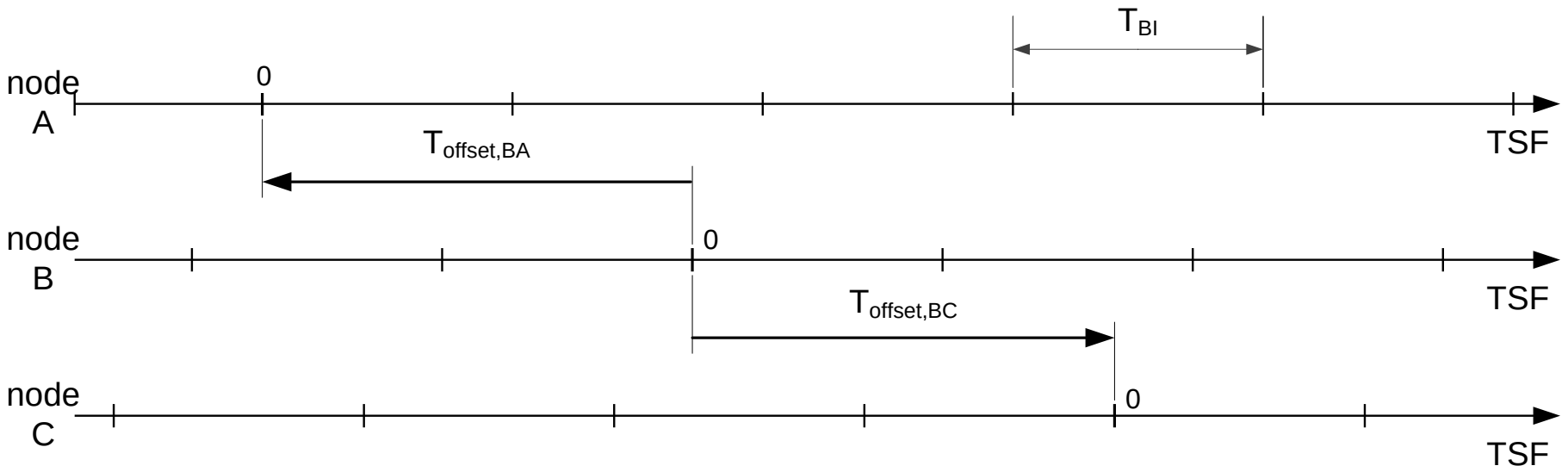
- Example scenario:
  - peers A, B, C with common beacon interval
  - B in light sleep mode towards both peers



Neighbor Offset Synchronization

- Determine T_{offset} towards peers

$$T_{offset,BA} = T_{t,A} - T_{r,B}$$



Neighbor Offset Synchronization

- Monitor and compare T_{offset} of each peer over consecutive beacon intervals

$$T_{ClockDrift} = T_{offset,old} - T_{offset,new}$$

$T_{ClockDrift} < 0$ → own clock is slower than peer's
$T_{ClockDrift} = 0$ → own clock is synchronous to peer's
$T_{ClockDrift} > 0$ → own clock is faster than peer's

Neighbor Offset Synchronization

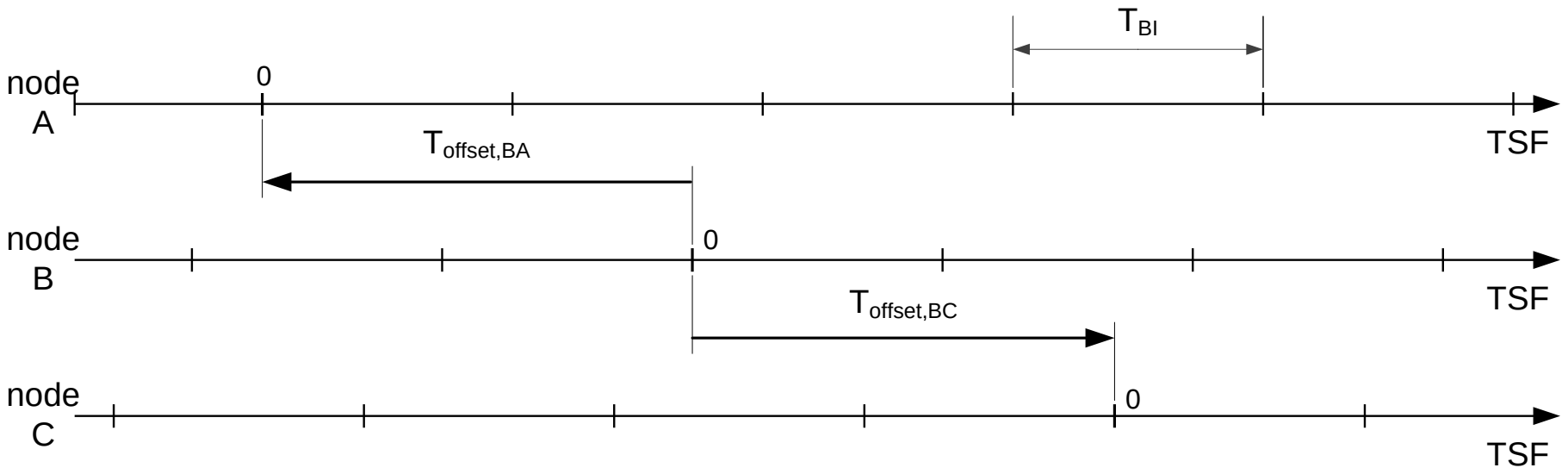
- at own TBTT, each node determines the maximum clock drift among its peers $T_{ClockDrift,max}$
- decreases its TSF by $T_{ClockDrift,max}$

→ all nodes adapt their TSF increments to the slowest clock among their peers
→ T_{offset} between peers is kept constant

Beacon Alignment Synchronization

- Determine T_{offset} towards peers
  (equally to neighbor offset synchronization)

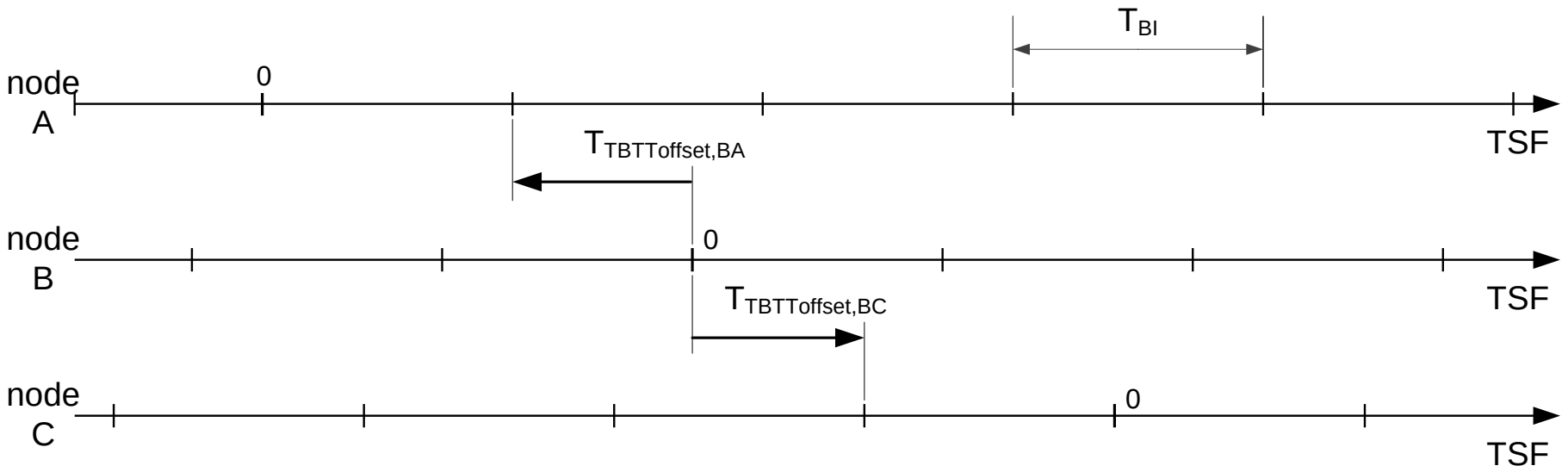
$$T_{offset,BA} = T_{t,A} - T_{r,B}$$



Beacon Alignment Synchronization

- Determine $T_{TBTToffset}$ towards peers

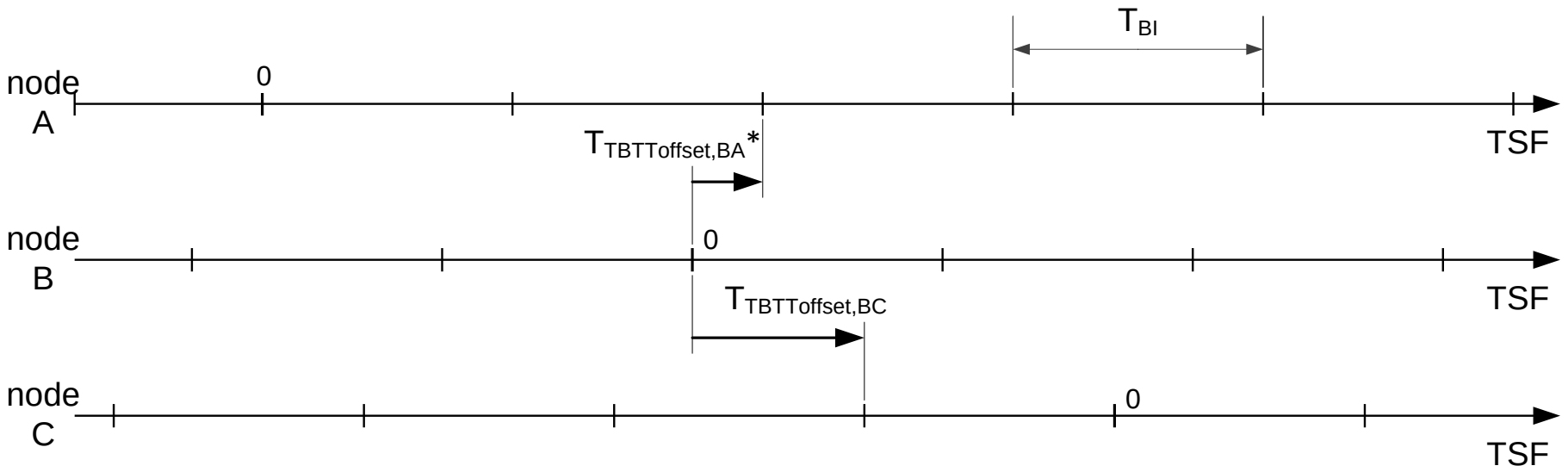
$$T_{TBTToffset,BA} = T_{offset,BA} \bmod \min(T_{BI,A}, T_{BI,B})$$



Beacon Alignment Synchronization

- Correct $T_{TBTToffset}$ towards TBTT later than the own

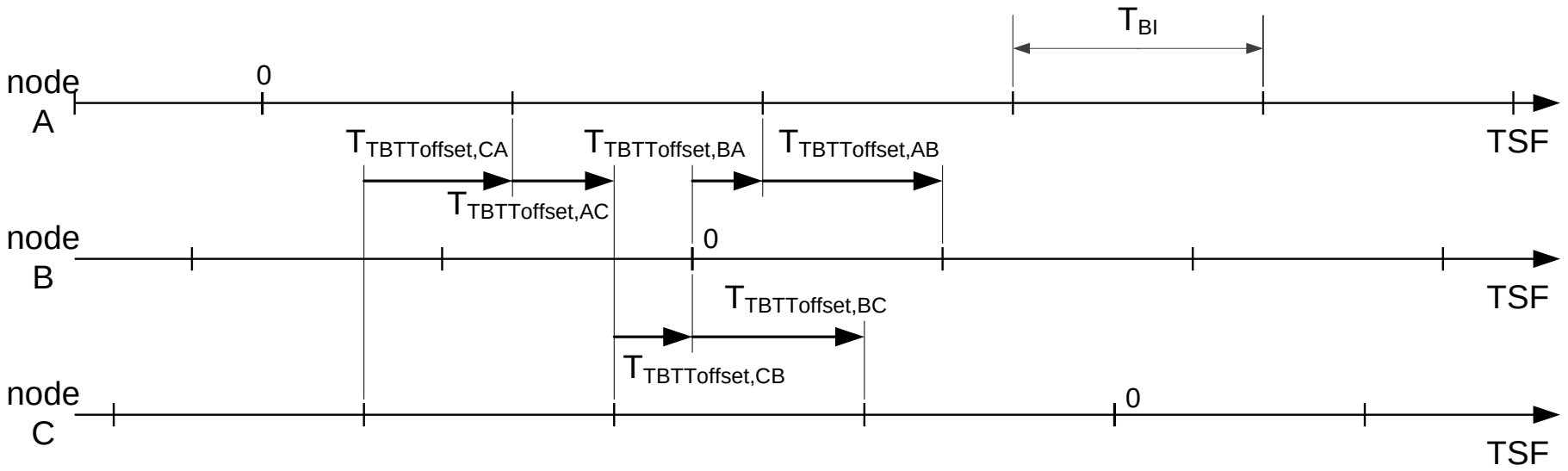
$$T_{TBTToffset,BA}^{*} = T_{TBTToffset,BA} - \min(T_{BI,A}, T_{BI,B})$$



Beacon Alignment Synchronization

- Determine $T_{TBTToffset}$ between peer nodes and from peer nodes towards self

$$T_{TBTToffset,CA} = (T_{offset,BA} - T_{offset,BC}) \bmod \min(T_{BI,A}, T_{BI,C})$$



Beacon Alignment Synchronization

- The minimum offset of each node is the offset towards the following peer's TBTT
- node with highest offset towards following peer is the "steady node"
  - all other nodes start delaying their TBTT (shifting right)