innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Wireless sensor networks: Time Synchronization

Time Sync ??

- All nodes in the network have a common view of time

Why Time Sync ??

BITS Pilani

- Target Tracking
- Speed estimation
- Event Detection
- Voice & Video Sync
- Security
- MAC-TDMA
- Local Clocks with crystal instability tend to drift

Specific examples

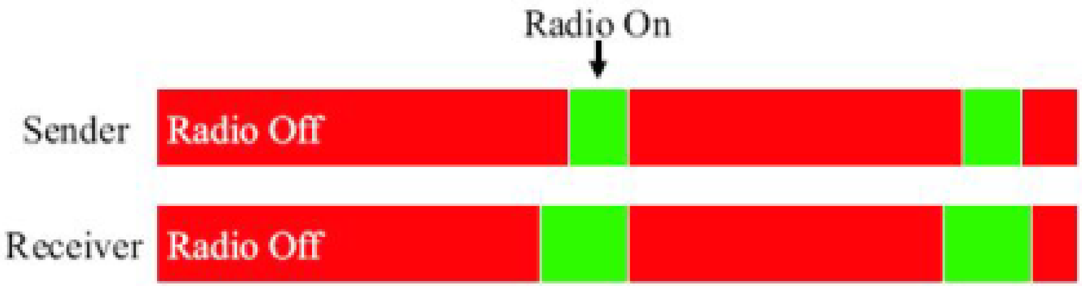
BITS Pilani

- **Link to the physical world!**
  - **When does an event take place?**
- **Key basic service of sensor networks**
  - **Fundamental to data fusion.**
- **Crucial to the efficient working of other basic services**
  - **Localization, Calibration, In-network processing etc.**
- **Several protocols require time synchronization**
  - **Cryptography, MAC, Topology management ….**

Case study…

BITS Pilani

- **Energy-efficient Radio Scheduling**
  - **TDMA, Guard band**



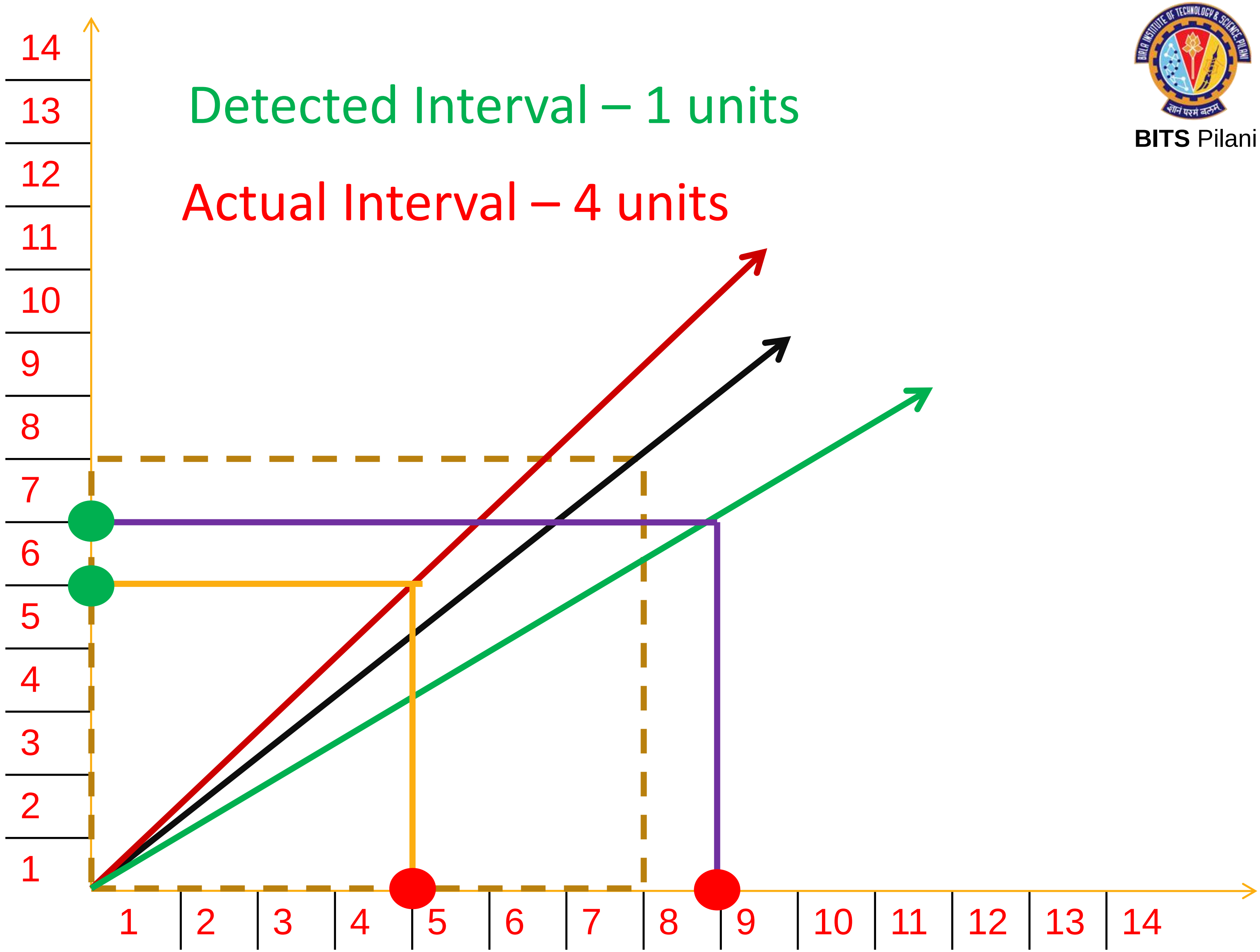
Reasons for drift

BITS Pilani

- Temperature – few ppm in PC
- Frequency noise (10^{-4} - 10^{-6})
- Phase Noise
- Asymmetric Delay

Detected Interval – 1 units

Actual Interval – 4 units



Time Sync Protocol - Requirements

- Robust
- Precision
- Energy Aware
- Server-Less
- Light Weight
- Tunable
- Immediacy

Performance Metrics and Fundamental Structure

BITS Pilani

Precision:

maximum synchronization error for deterministic algorithms, mean error /stddev /quantiles for stochastic ones

Energy costs,

e.g. # of exchanged packets, computational costs

Memory requirements

Fault tolerance:

what happens when nodes die?

Why not GPS ?

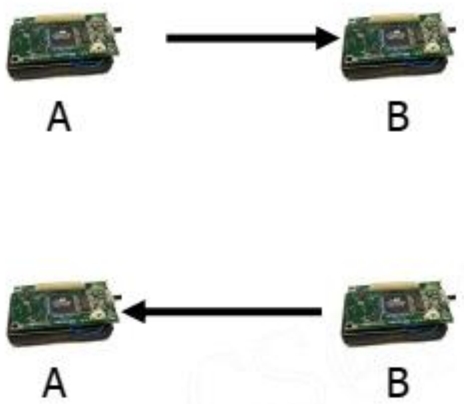
BITS Pilani

- **Cost**
  - **300$ (achieve < 20ns phase error to UTC)**
- **Practical Limitations**
  - **Cannot be used under special environment where is no free line of sight to the GPS satellites**
    - **e.g. dense foliage or inside buildings**
- **Policy Limitations**
  - **Military Applications**

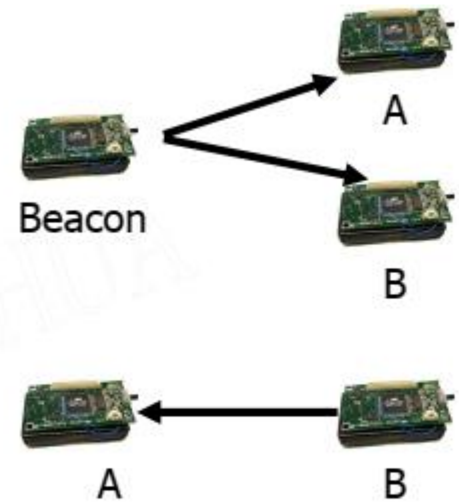
Wireless Sensor Network – Time Sync Protocols

Time Sync Types

- Sender – Receiver synchronization
- Receiver – Receiver synchronization



S – R



R – R

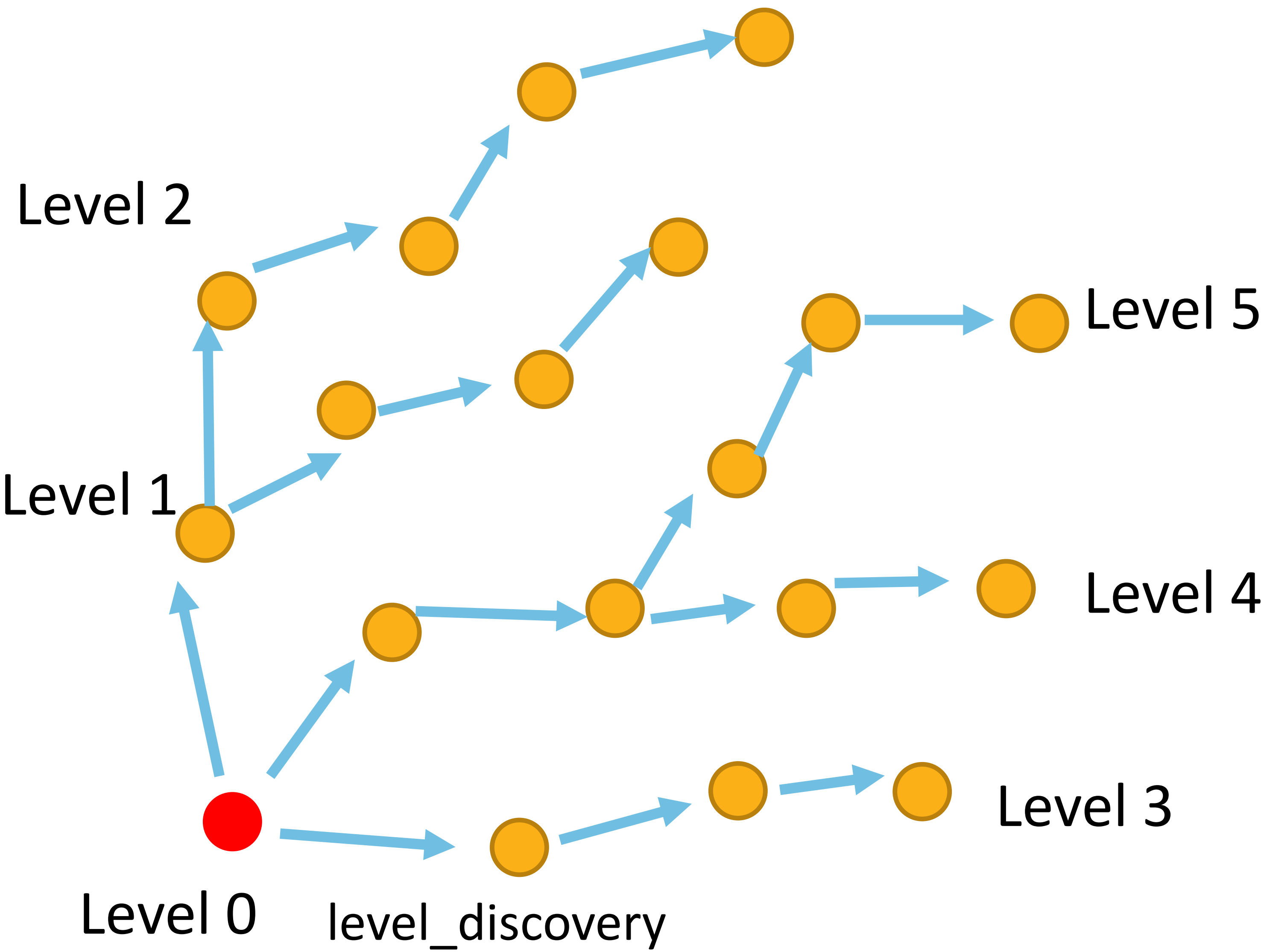
Sender – Receiver Synch

Time Sync Protocol for Sensor Networks (TPSN)

- Level Discovery
- Sync

TPSN – Level discovery

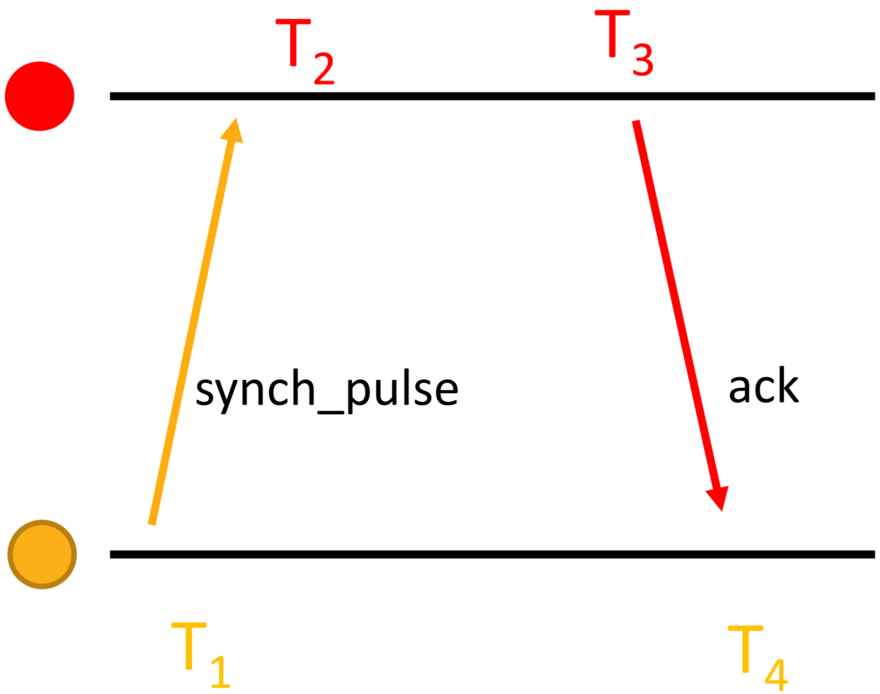
BITS Pilani



TPSN - Sync

- Root node initiates – time sync
- Level 1 nodes – each wait for random time
  - Reduce probability of collision at MAC

TPSN – Sync



$T_2 = T_1 + \Delta + d$

Δ - clock drift

d – propagation delay

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

$$d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

TPSN - Sync

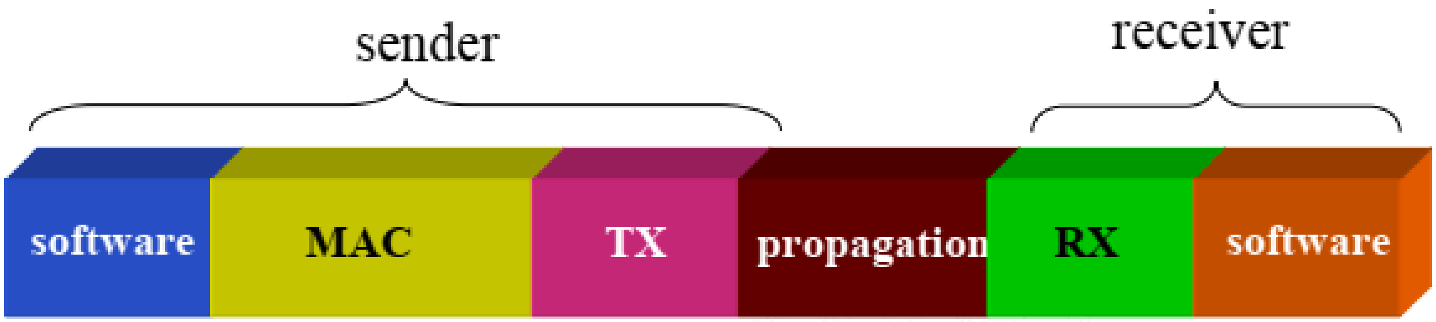
BITS Pilani

- Nodes at Level 2 will be able to hear sync pulses of nodes at Level 1
- Wait for a random time
- Attempt to sync with nodes at Level1

TPSN - Issues

- Unable to hear level_discovery from higher level nodes – then wait and send level request
- Hear from different nodes – different levels – pick smallest level
- No response to sync pulse as node at higher level dead – send level request at higher energy levels

Sources of error



ALL DELAYS ARE VARIABLE !

Sources of error

BITS Pilani

- **Send time**
  - Kernel processing
  - Context switches
  - Interrupt Processing
- **Access time**
  - Specific to MAC protocol
    - E.g. in Ethernet, sender must wait for clear channel
- **Transmission Time**
- **Propagation time**
  - Very small in WSNs, can be omitted
- **Reception time**
- **Receive time**

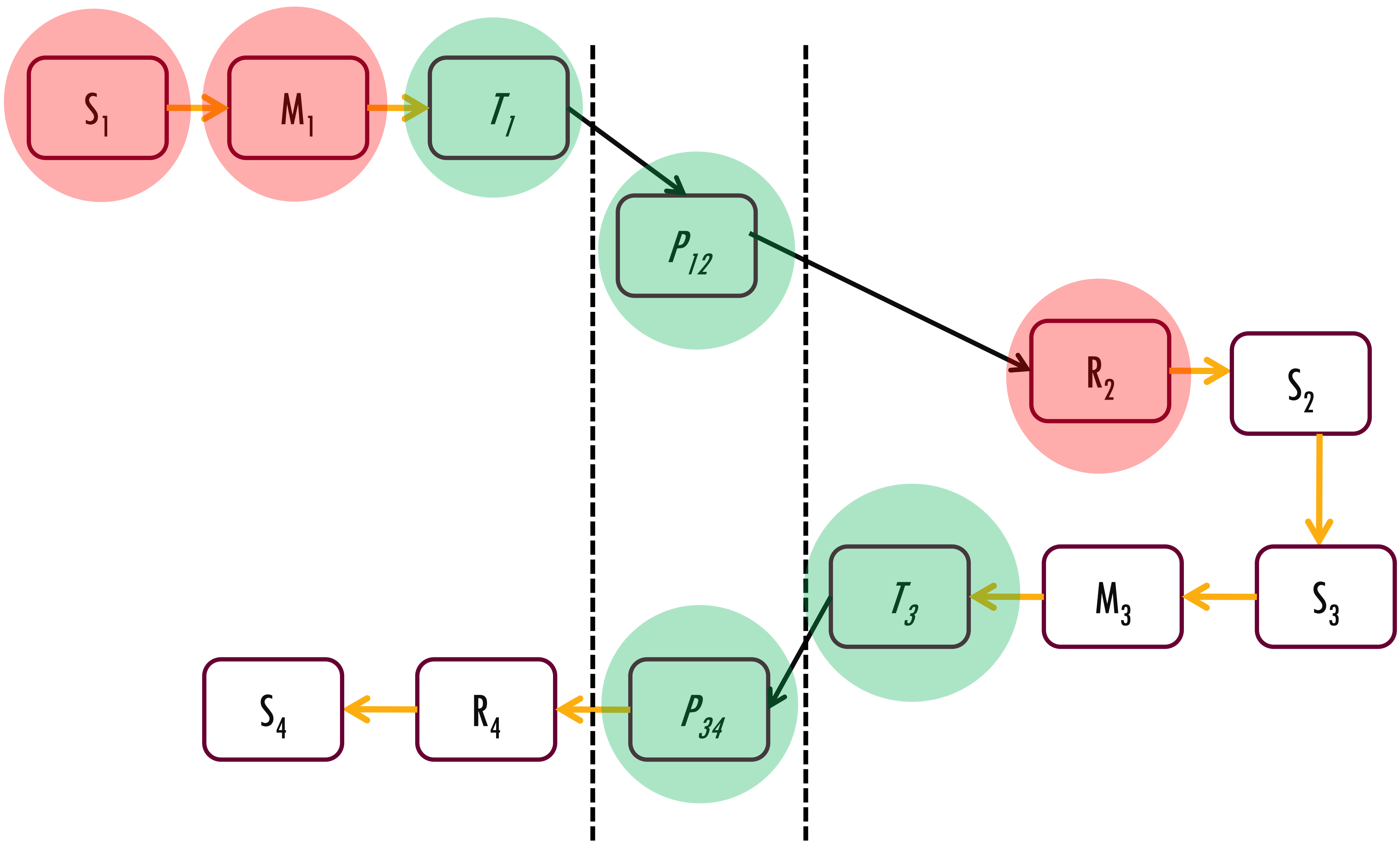
Common denominator:

(1) non-determinism

(2) difficult to estimate

Send, access, receive

TPSN - Inaccuracies



Flooding Time Synchronization Protocol (FTSP)

- Introduction

Solves one of the problems of TPSN

- Linear regression is used in FTSP to compensate for clock drift

Flooding Time Synchronization Protocol (FTSP)

Network Model

- Every node in the network has a unique ID

- Each synchronization message contains three fields:

TimeStamp

RootID

SeqNum

-The node with the smallest ID will be only one root in the whole network

Flooding Time Synchronization Protocol (FTSP)

- The root election phase
  FTSP utilizes a simple election process based on unique node IDs

- Synchronization phase

Flooding Time Synchronization Protocol (FTSP)

The root election phase

When a node does not receive new time synchronization messages for a number of message broadcast periods

The node declares itself to be the root

Whenever a node receives a message, the node with higher IDs give up being root

Eventually there will be only one root

Flooding Time Synchronization Protocol (FTSP)

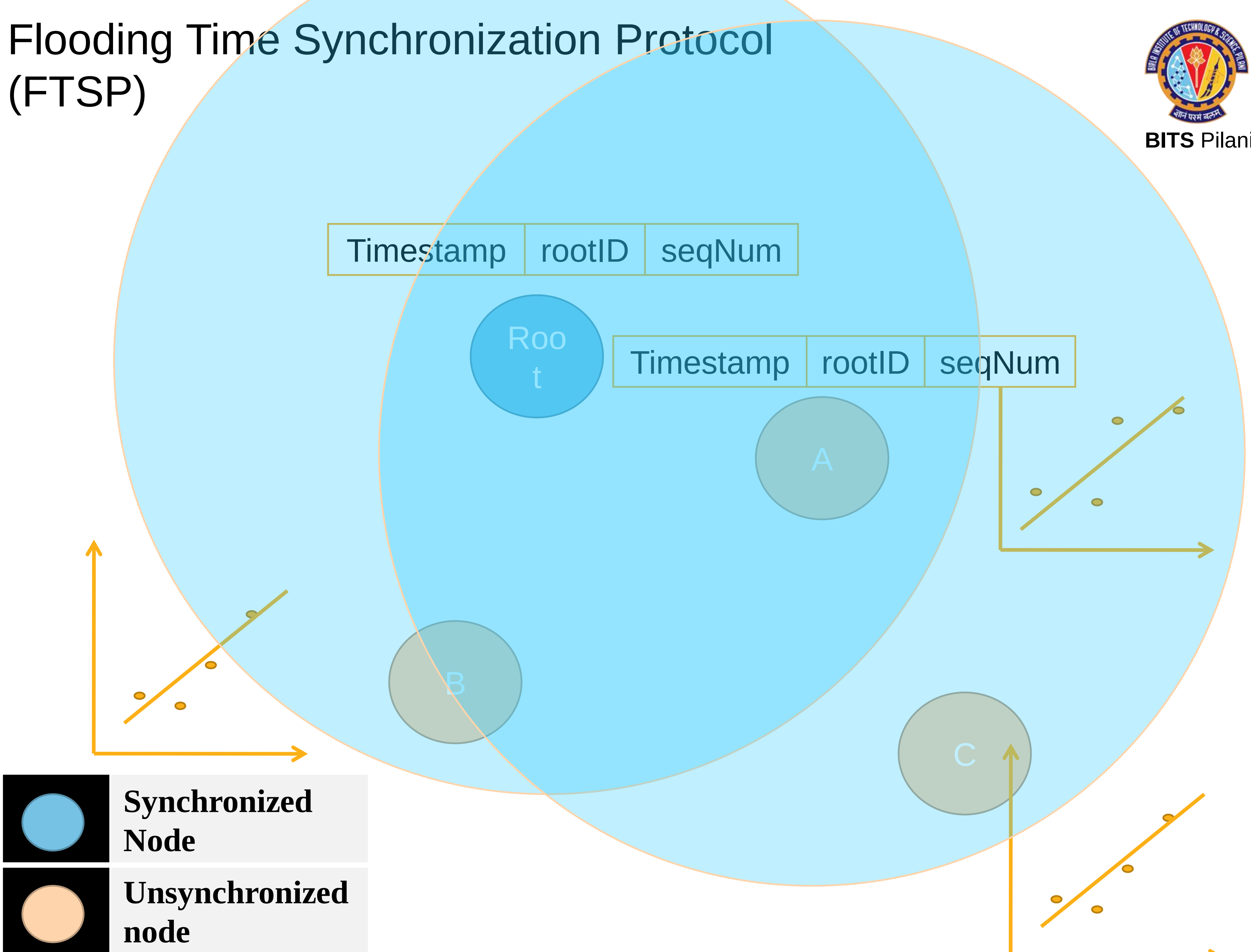
Synchronization phase

Root and synchronized node broadcast synchronization message

Nodes receive synchronization message from root or synchronized node

When a node collects enough synchronization message, it estimates the offset and becomes synchronized node

Flooding Time Synchronization Protocol (FTSP)



BITS Pilani

Wireless Sensor Network – Time Sync Protocols

Time Sync Types

BITS Pilani

- Sender – Receiver synchronization
- Receiver – Receiver synchronization

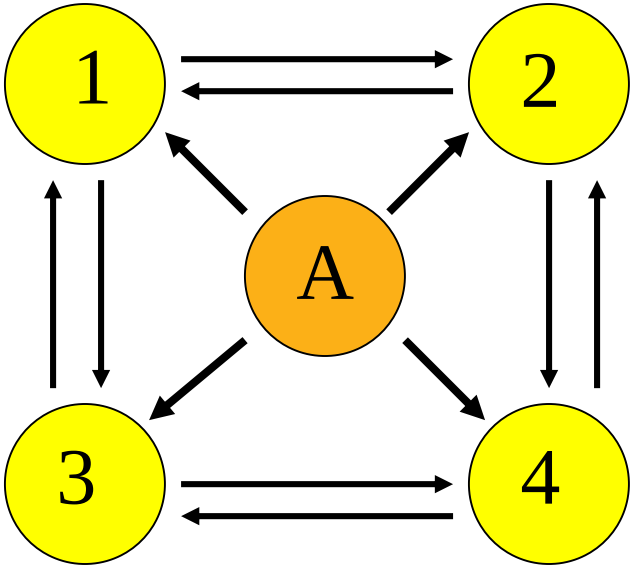
Receiver – Receiver Synch

Receiver Broadcasting Service (RBS)

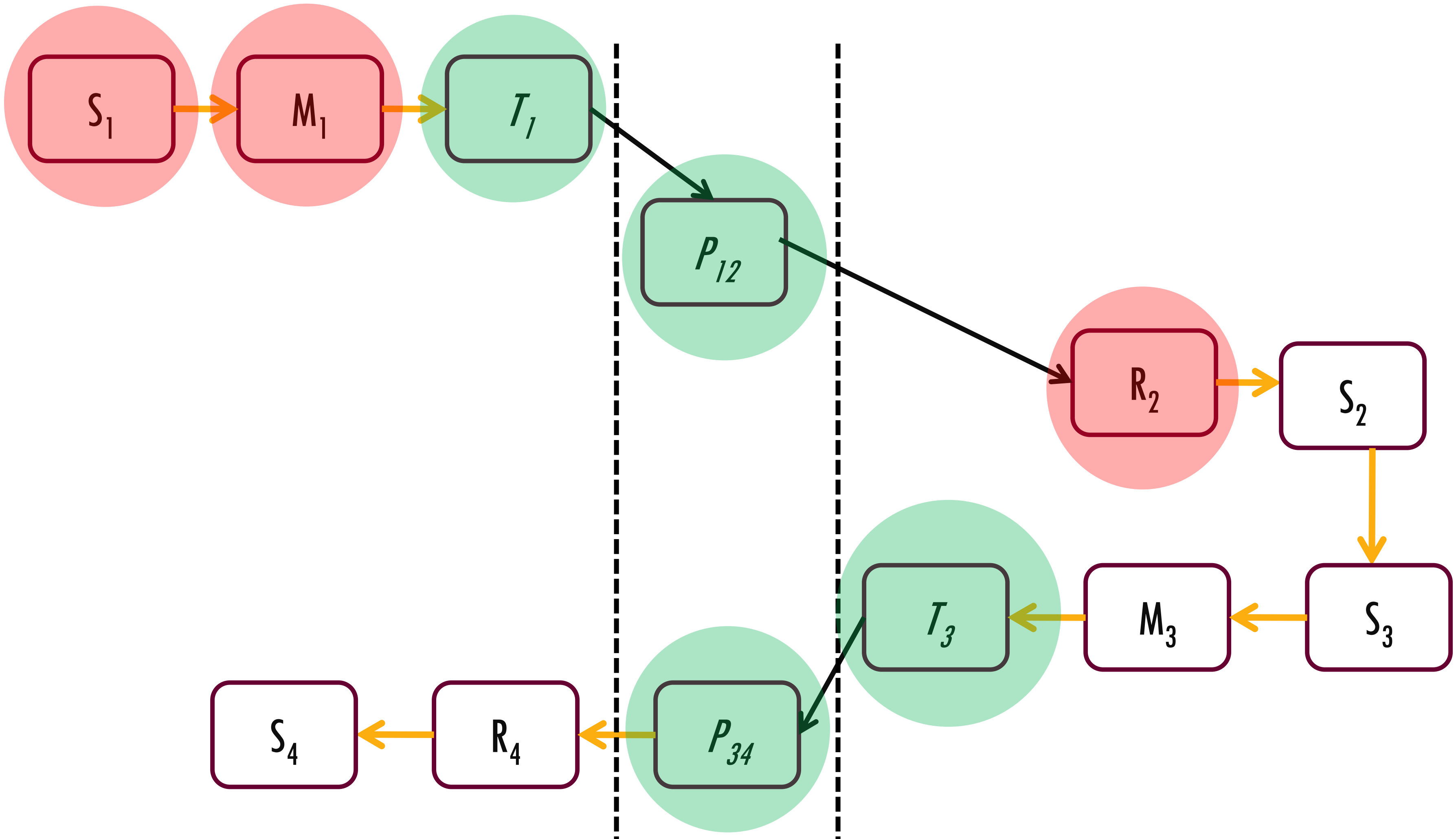
BITS Pilani

- Three stages
- Transmitter broadcasts clock time
- Each rx records the time that the ref was rxed-local clock
- Receivers exchange observations

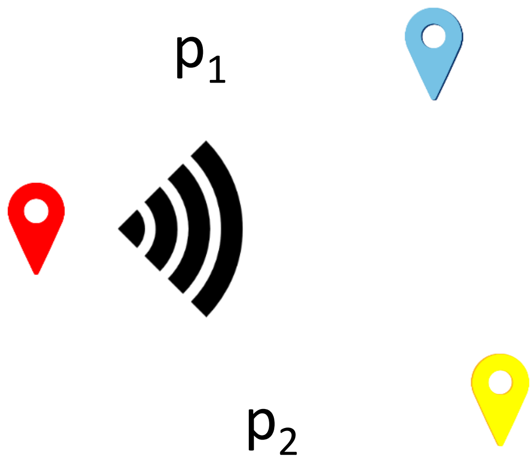
RBS



TPSN - Inaccuracies



Inaccuracies Removed ??



$$r_1 = T + p_1$$

$$r_2 = T + p_2$$

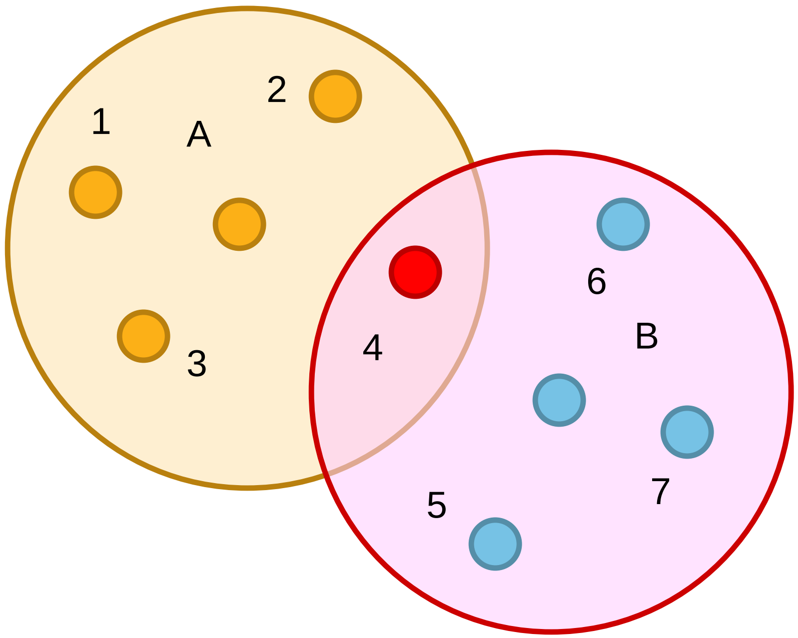
Propagation delay - negligible

$$T_2 = T_1 + \Delta +$$

Is RBS extremely accurate ?

- No as both skew and offset contribute to lack of sync
- Offset – as each node may start at different time
- Works well in single – hop
- Requires Time Translation in case of multi-hop

Time Translation



P_A sent

E_1 occurs 2 units later

E_2 occurs

P_B sent 4 units later

P_B sent 10 units before P_A

$E_1 = P_A + 2$

$E_2 = P_B - 4$

$P_A = P_B + 10$

$E_1 – E_2 = P_B + 10 + 2 – P_B + 4$

$E_1 – E_2 = 16$

Time Sync Protocols

BITS Pilani

- Type1 : Time servers
- Type2 : Translate time thro’ ntk
- Type3 : Self-organize to sync clock