Enabling Robust, Secure and Efficient Knowledge of Time Across the System Stack

(Start Date: June 2014)

Carnegie Mellon University

UCLA

UCSB

UCSD








clock_gettime()

ClockTime()

What is the "time"?

os.time()

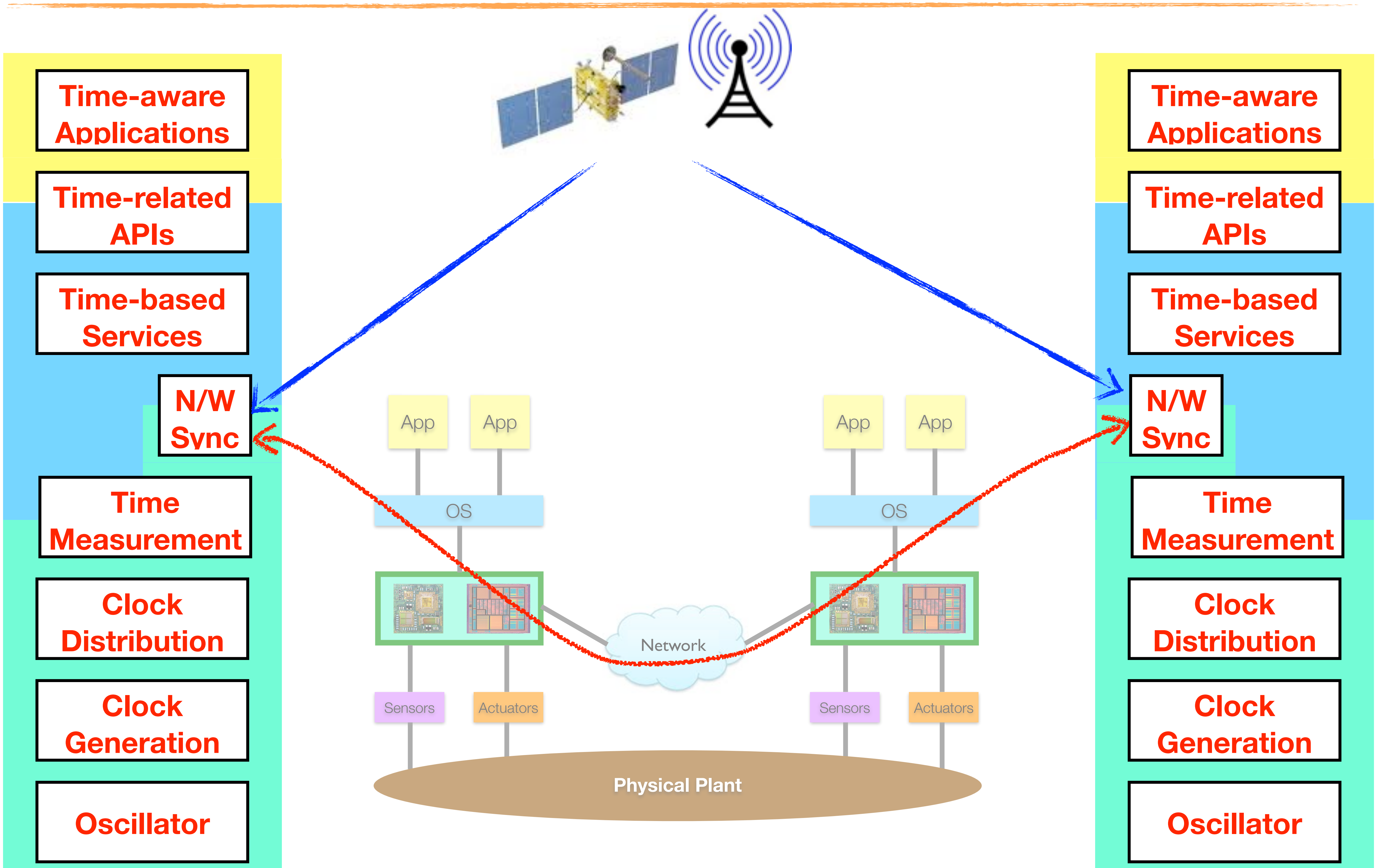
The time is 1375599686.9 + Δ

time.time()

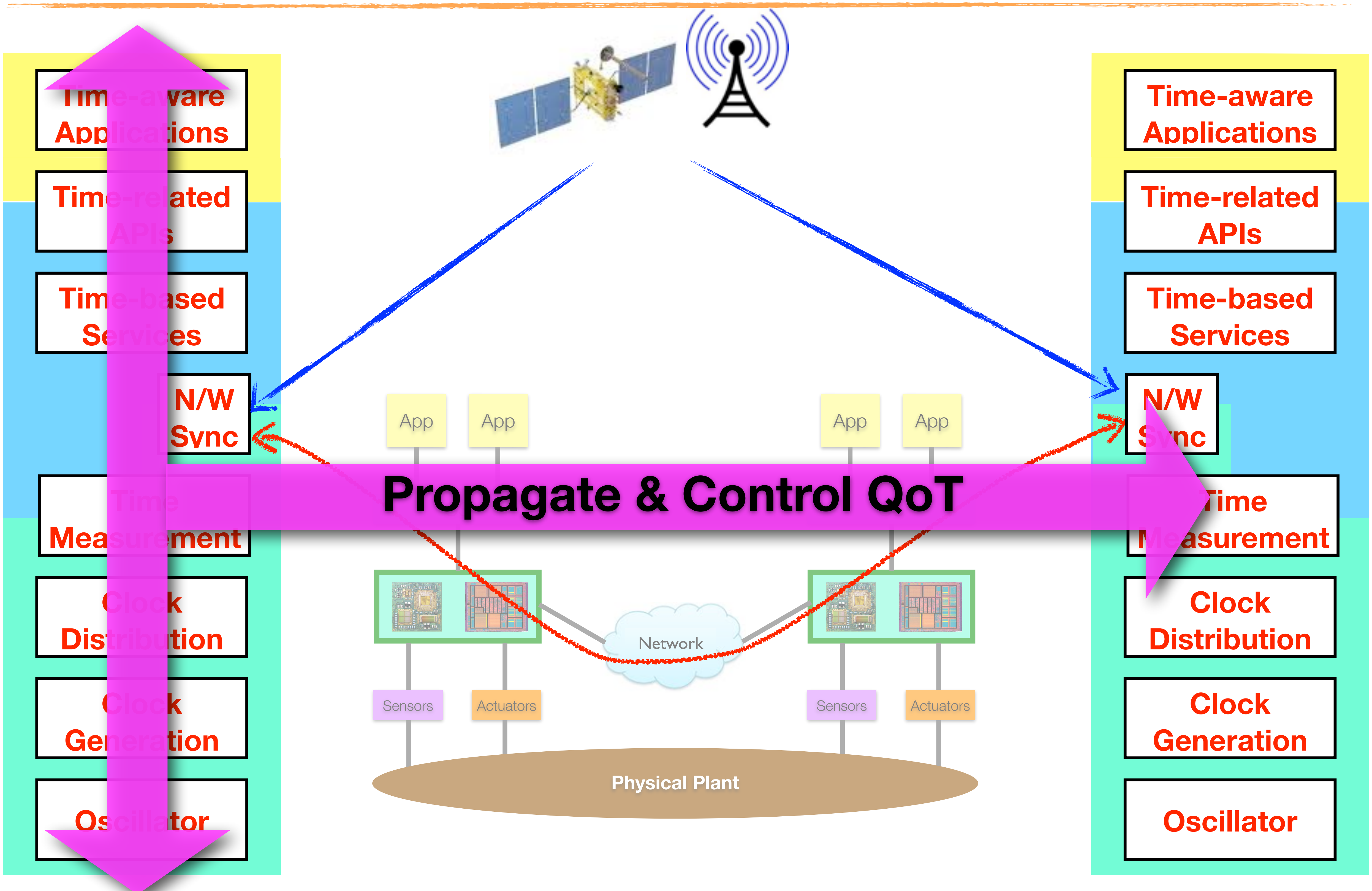
Date.now()

Our objective: transform Δ into a rich structure called "Quality of Time" (QoT) that is observed and systematically controlled throughout the system.

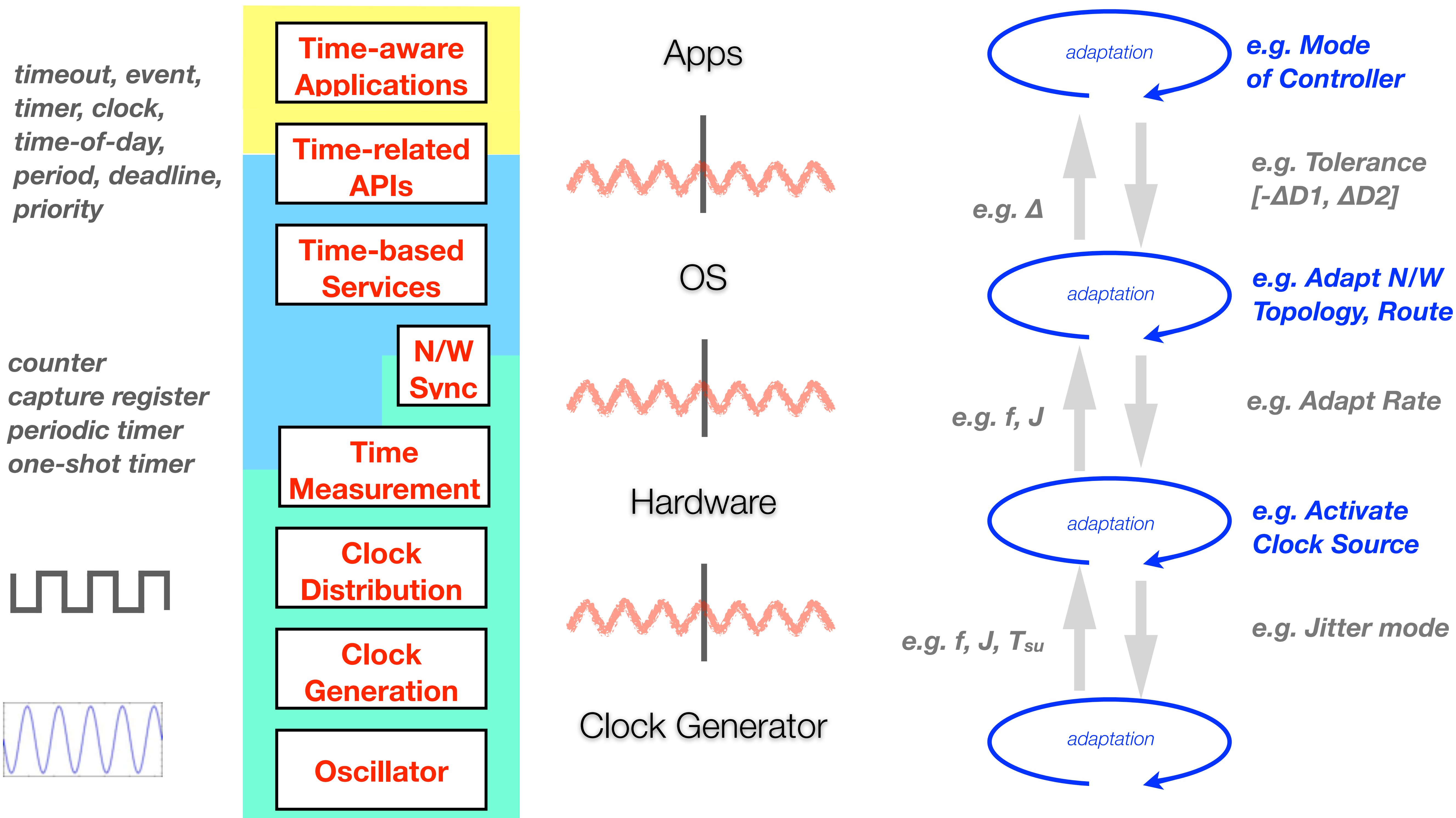
Time Information Traverses a Deep Stack



Time Information Traverses a Deep Stack

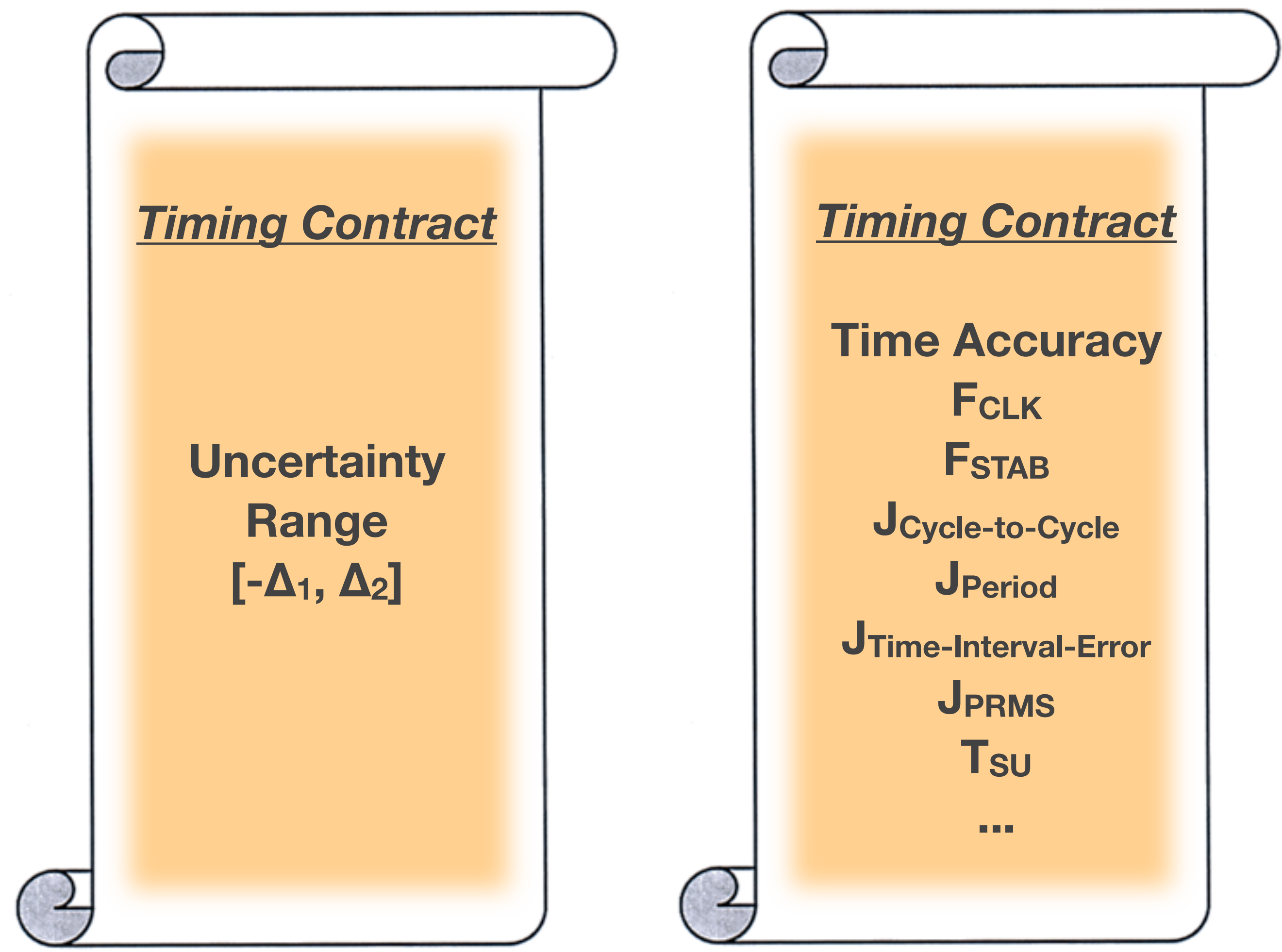


Rethinking the Time Stack
From Over-design to Sense-and-Adapt

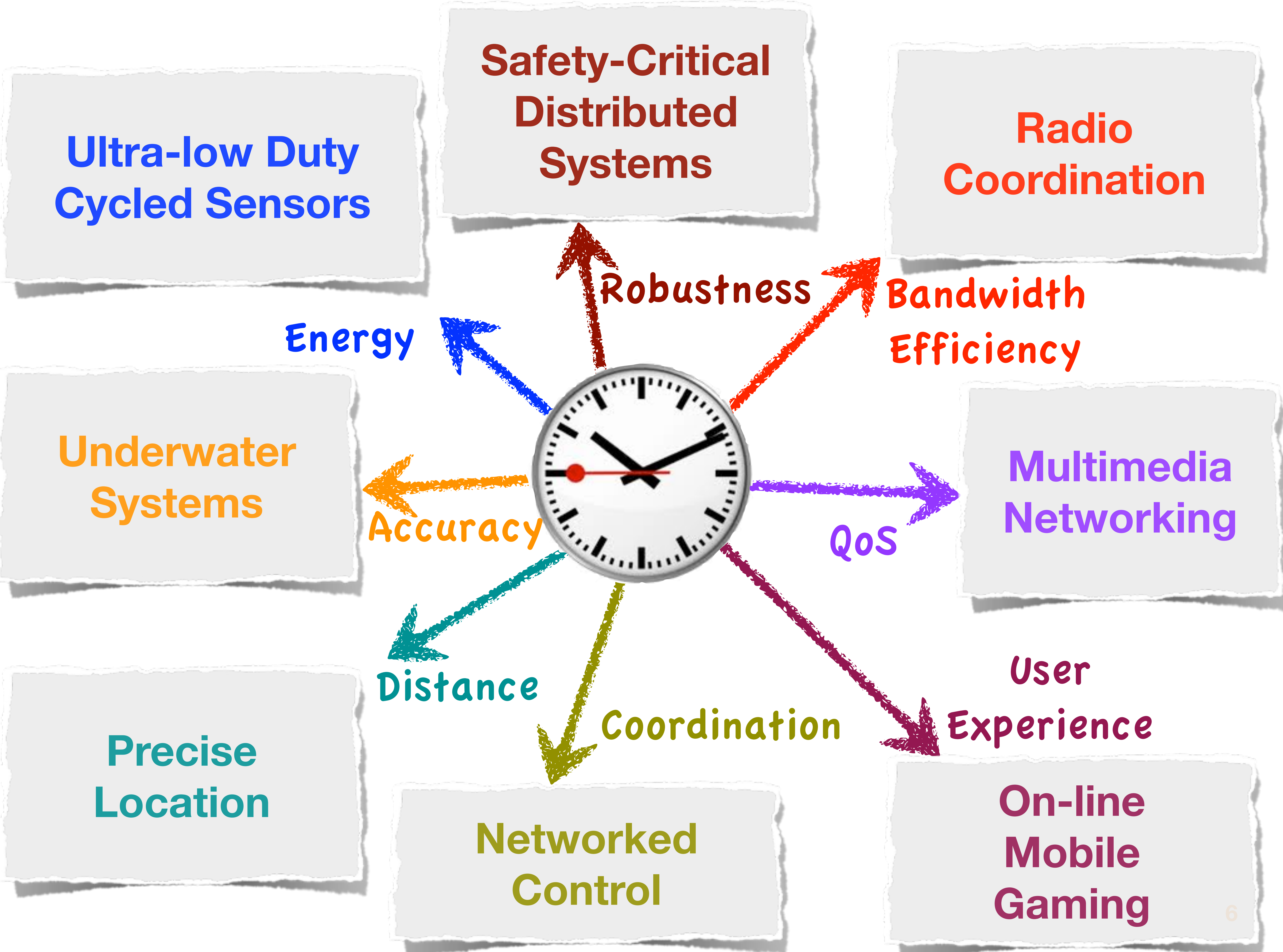


Quality of Time (QoT) Enables Contracts

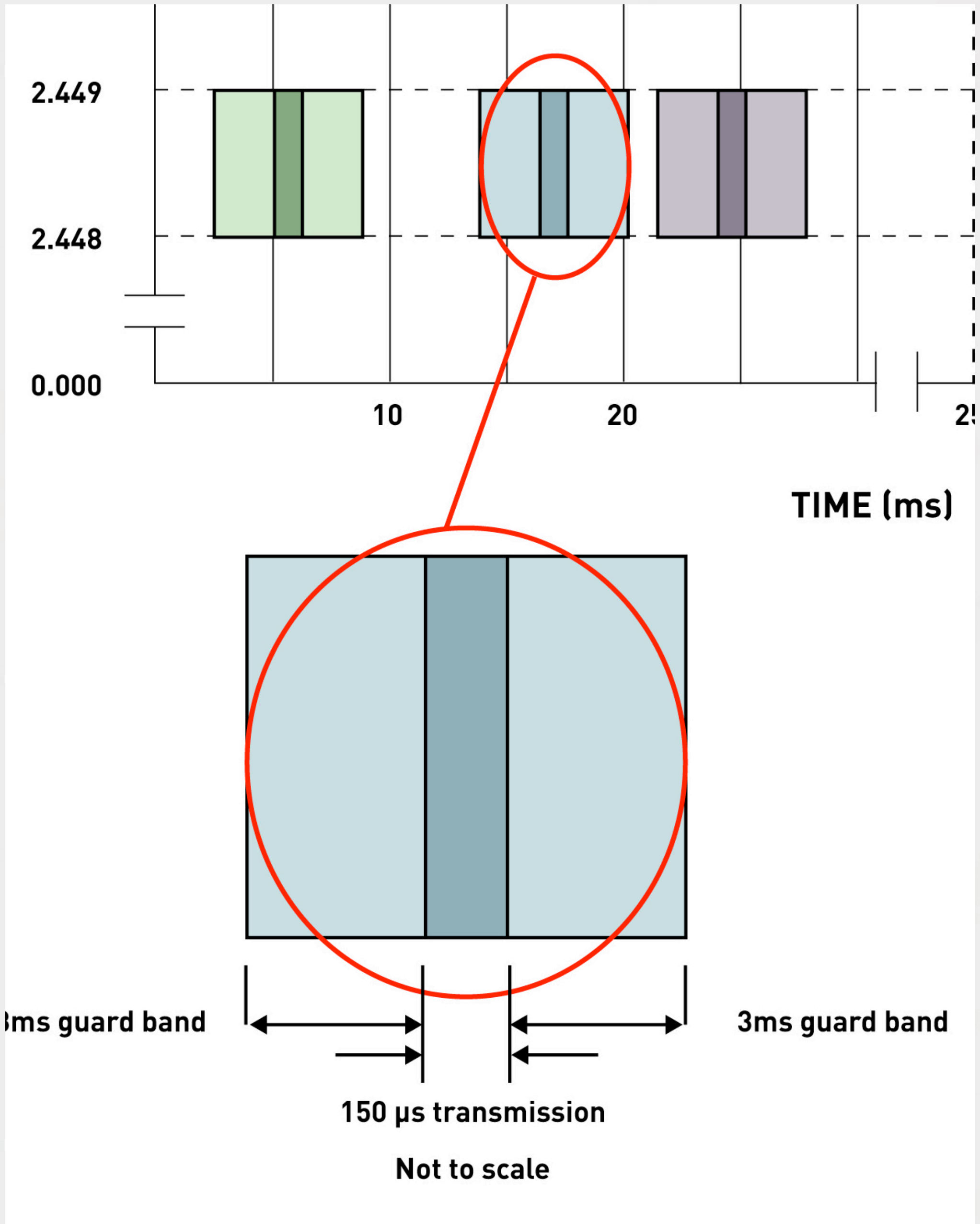
- **Basis of timing contracts**
  - Lower layer provides available QoT
  - Higher layer indicates required QoT
  - Exposed via APIs, HALs, and hardware interfaces
- **Not a single number, but statistical parameters and ranges**
- **Available QoT depends on**
  - Clock sources
  - Network paths
  - Network interface hardware
  - Timer/counter hardware
  - OS mechanisms
- **Function of $, J, bps, mm^3 etc.**



App-OS Interface *OS-HW Interface*

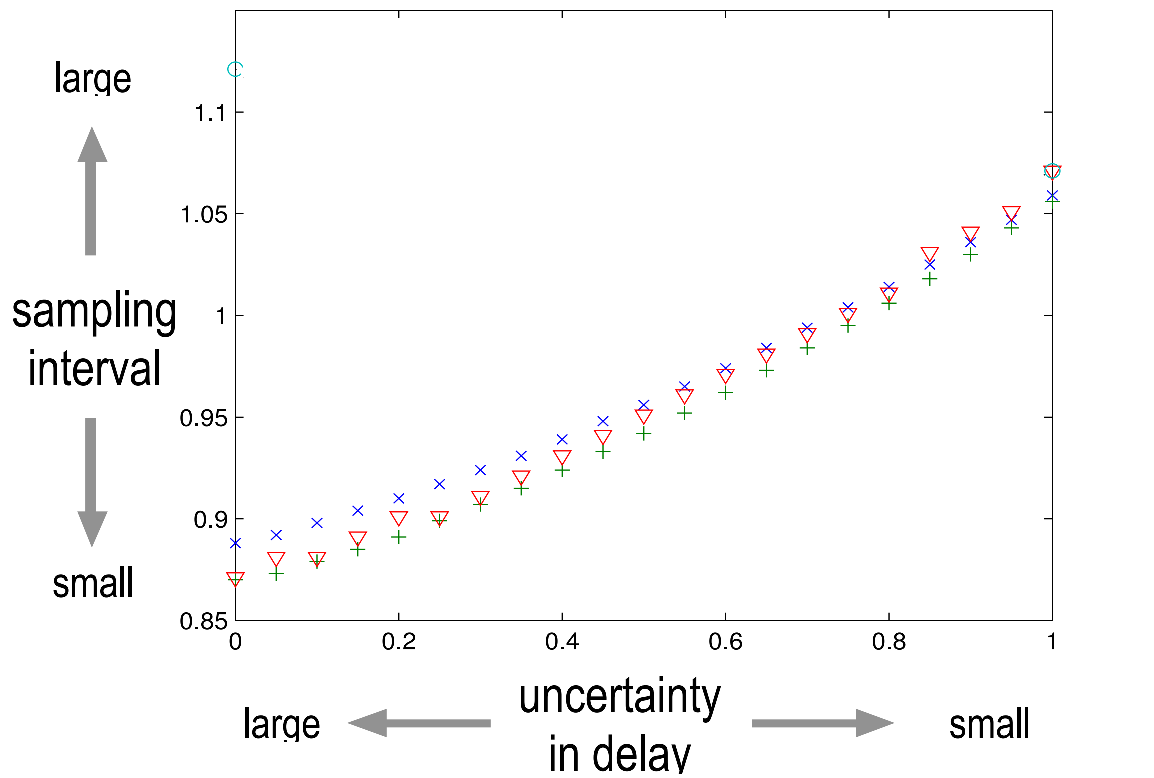

Ultra-low Duty Cycled Sensors
Safety-Critical Distributed Systems
Radio Coordination
Energy
Robustness
Bandwidth Efficiency
Underwater Systems
Accuracy
Multimedia Networking
QoS
Distance
Coordination
User Experience
Precise Location
Networked Control
On-line Mobile Gaming

Radio Coordination in Wireless

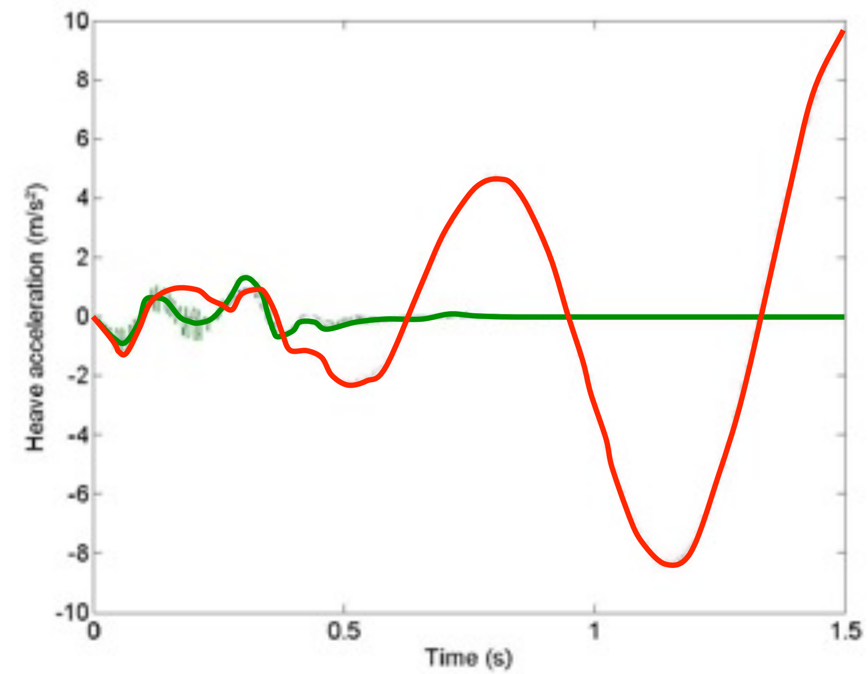


- Short-range radio protocols for D2D, wireless sensors etc.
- Examples: ANT, BLE, ZWAVE
- Low cost, power, bandwidth → High Δ → Guard bands
- Guard band proportional to relative drift & resync interval
- 3 + 3 ms guard band for each 150 us message → 2.4% efficiency
- Reduce guard band to 0.3 ms → 20% efficiency, or 8.2x more devices, power reduced 1.4-5x
- **Knowing QoT can enable more capacity and lower power**

Networked Control



- Benchmark system from [Branicky, Phillips, Zhang, ACC 2000]
- Different symbols correspond to different analysis methods



- Distributed controller for car suspension system
- Two curves correspond to different CAN bus priorities

- Uncertainty in delay measurement results in faster sampling
- **Knowledge of QoT enables more robust and efficient controller design and operation**

Precise Location in IoT

AT86RF233

- New low-power radios capable of distance measurement via phase measurement
- Jitter in clock due to PLL affects distance accuracy: low jitter PLL uses power while averaging wastes bandwidth
- Stability requirements on clock change over time
- **Dynamically scaling QoT (power vs. jitter of PLL) can reduce distance error (from cms to mms) while retaining low-power rest of the time**

Project Activities

Thrust 2

Mechanisms and Interfaces for QoT-aware Applications

Thrust 1

Autotuning Time Service for Efficient & Resilient QoT

Thrust 3

Circuit and Architecture Support for QoT Adaptation

Thrust 4

Experimental Testbeds

Thrust 5

Education & Outreach

Project Activities

Thrust 2

Mechanisms and Interfaces for QoT-aware Applications

Thrust 1

Autotuning Time Service for Efficient & Resilient QoT

Thrust 3

Circuit and Architecture Support for QoT Adaptation

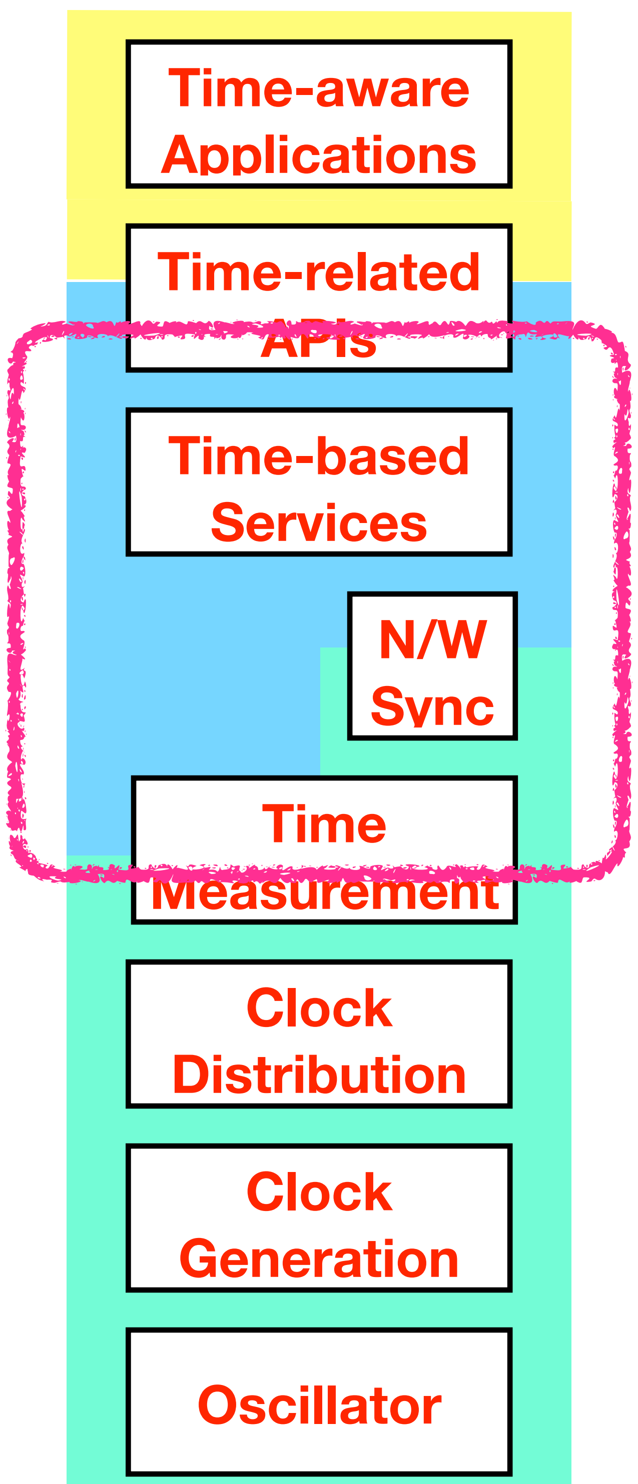
Thrust 4

Experimental Testbeds

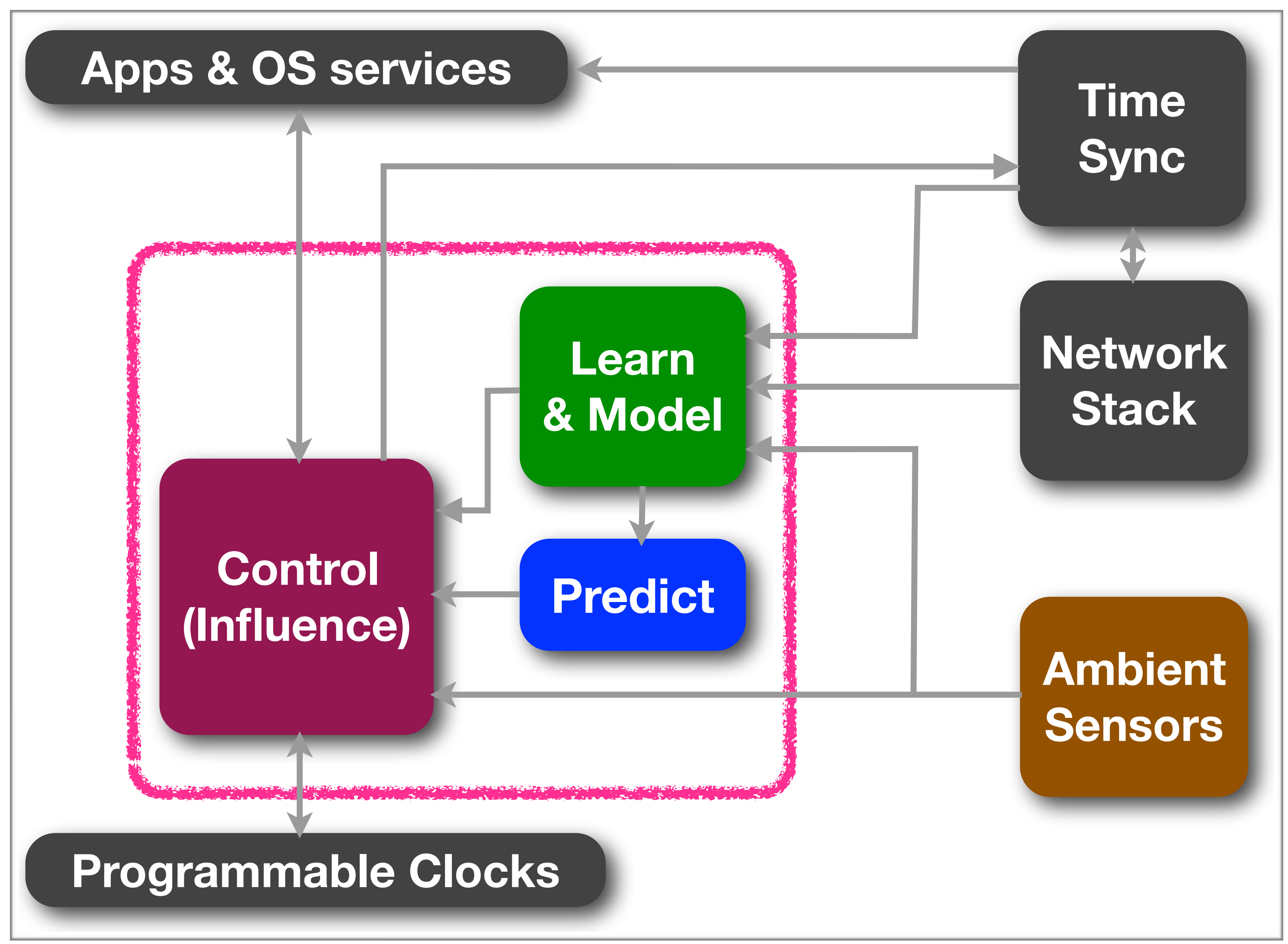
Thrust 5

Education & Outreach

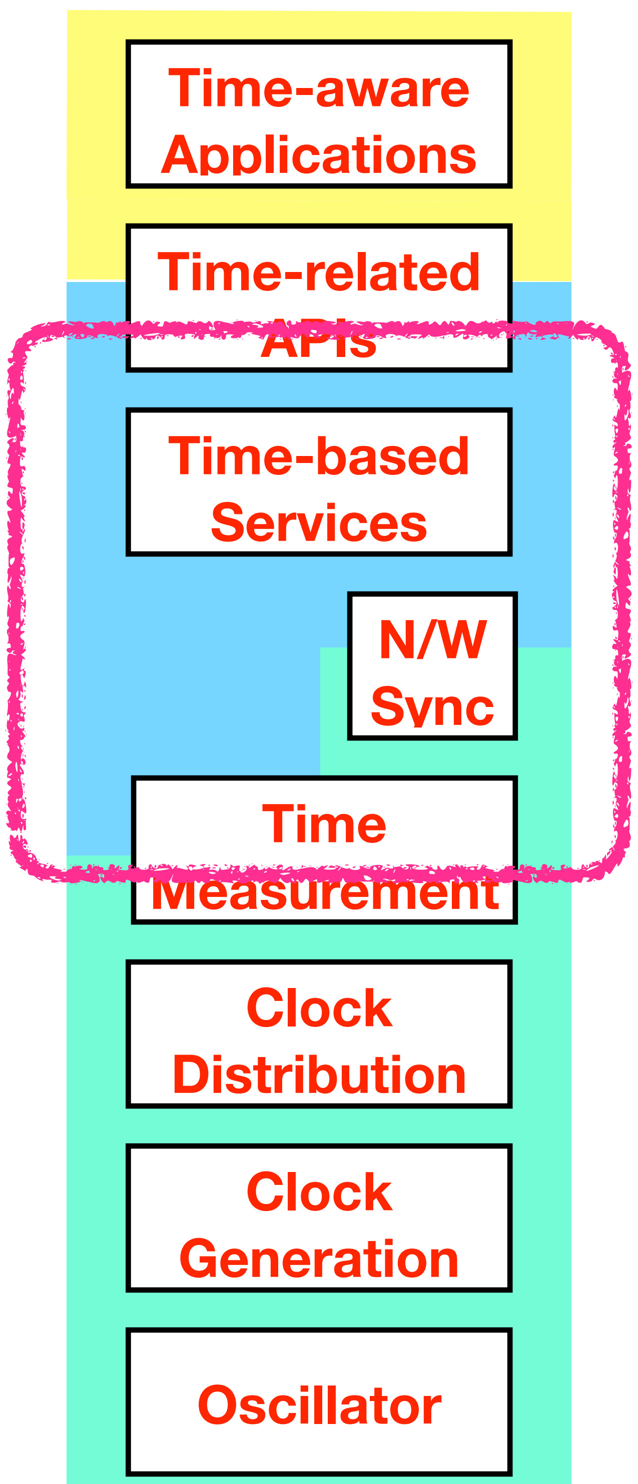
Autotuning Time Service



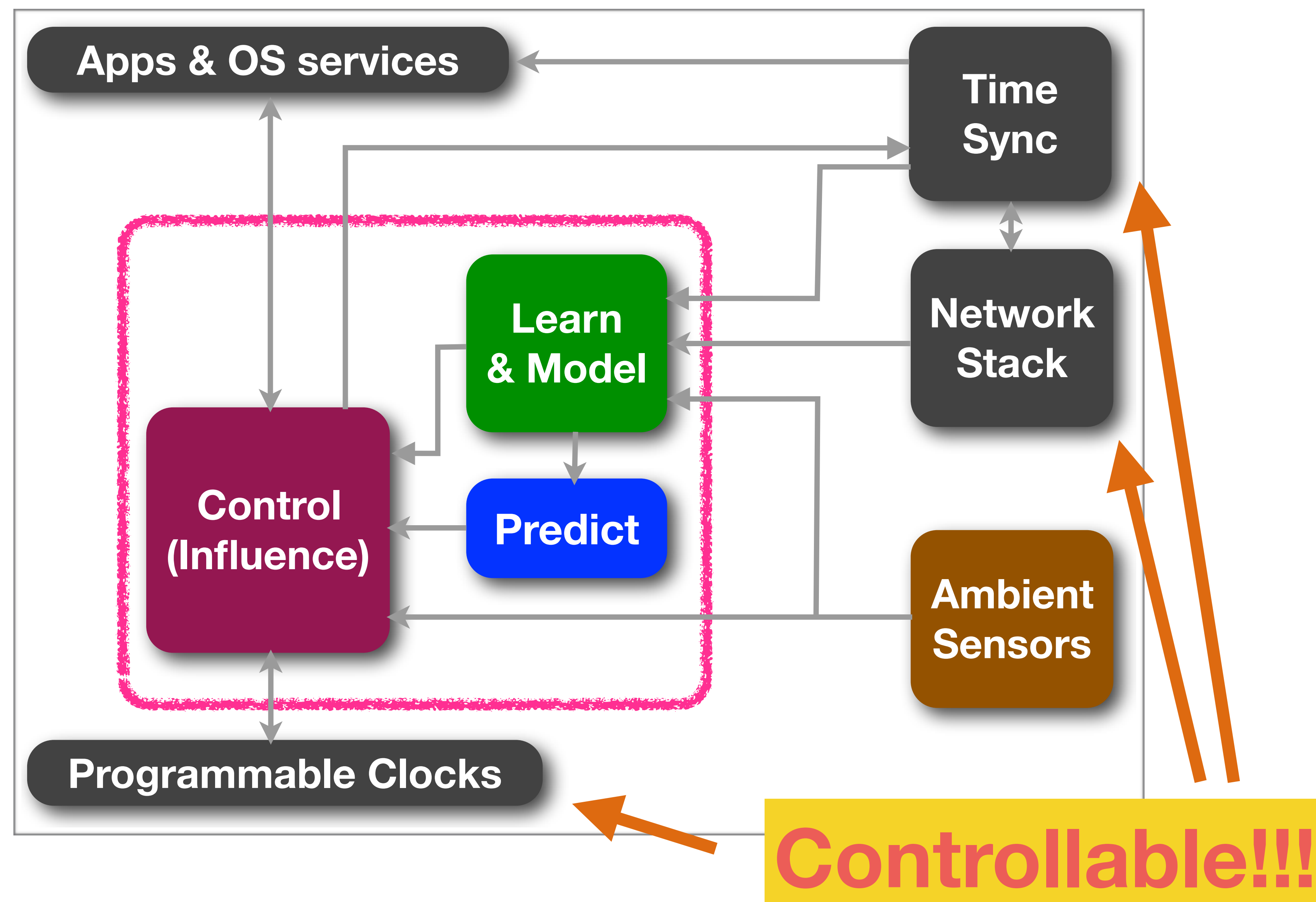
QoT = f (clock sources, network, ambient)



Autotuning Time Service

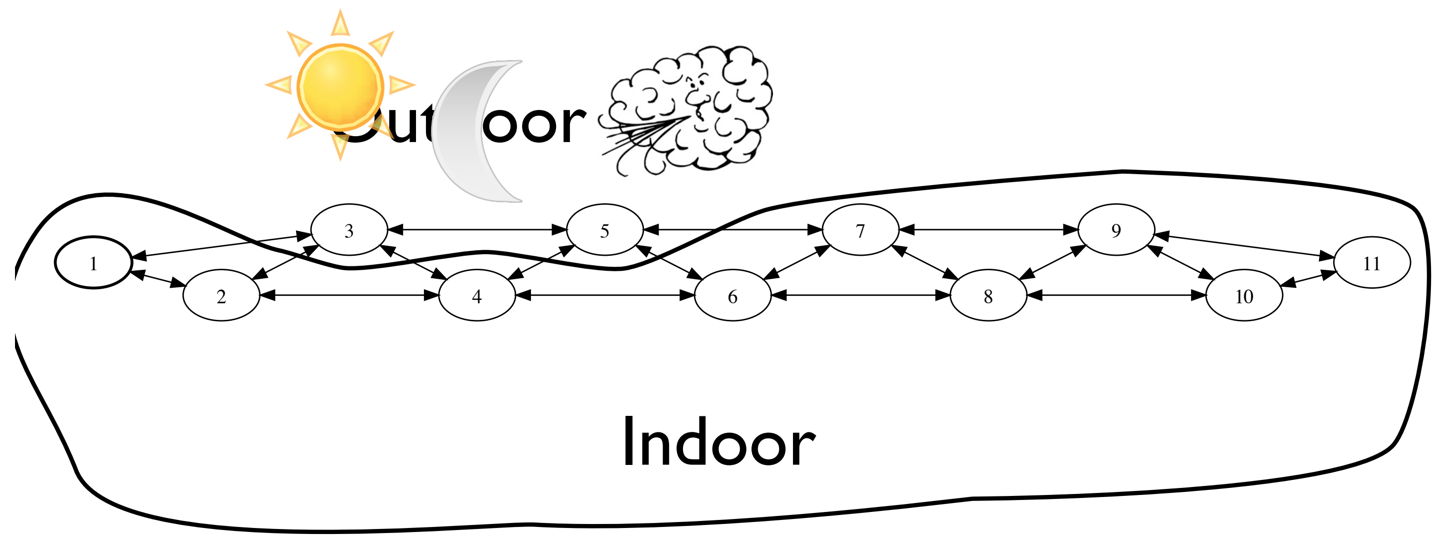


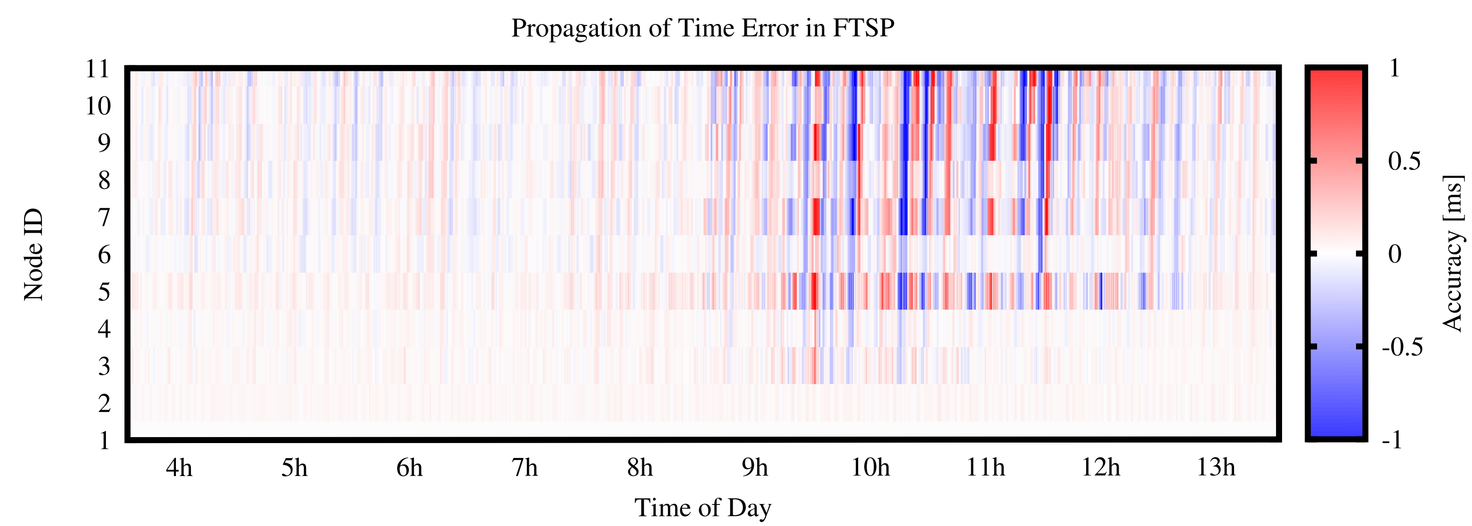
QoT = f (clock sources, network, ambient)

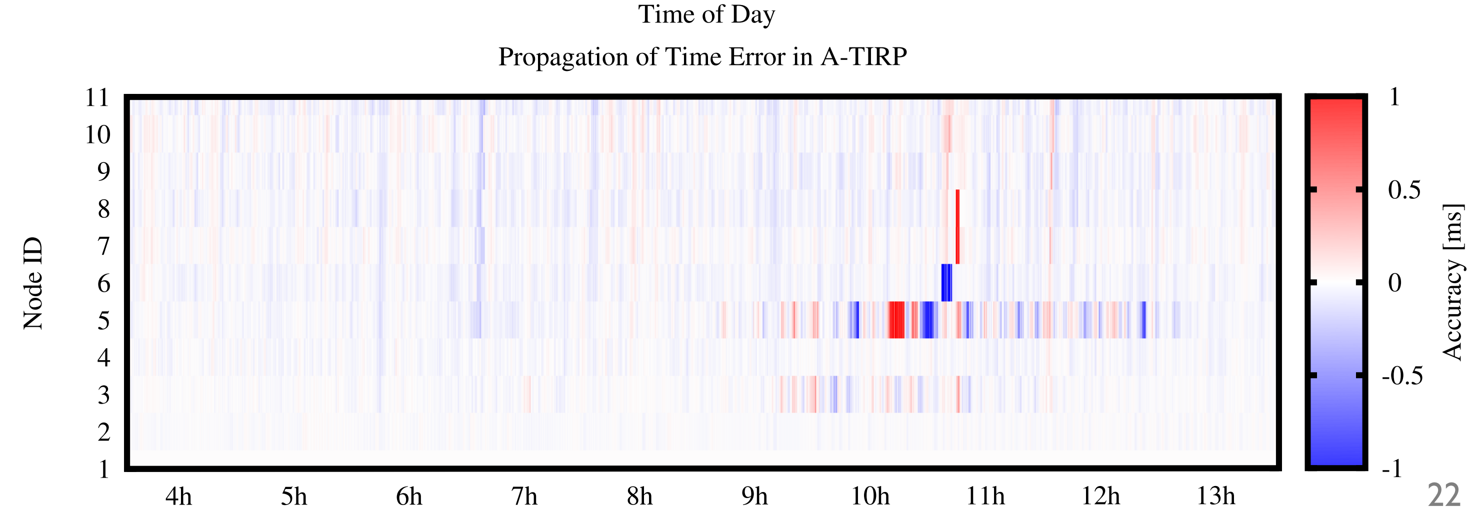


Tunable Time Sync for Multihop Wireless

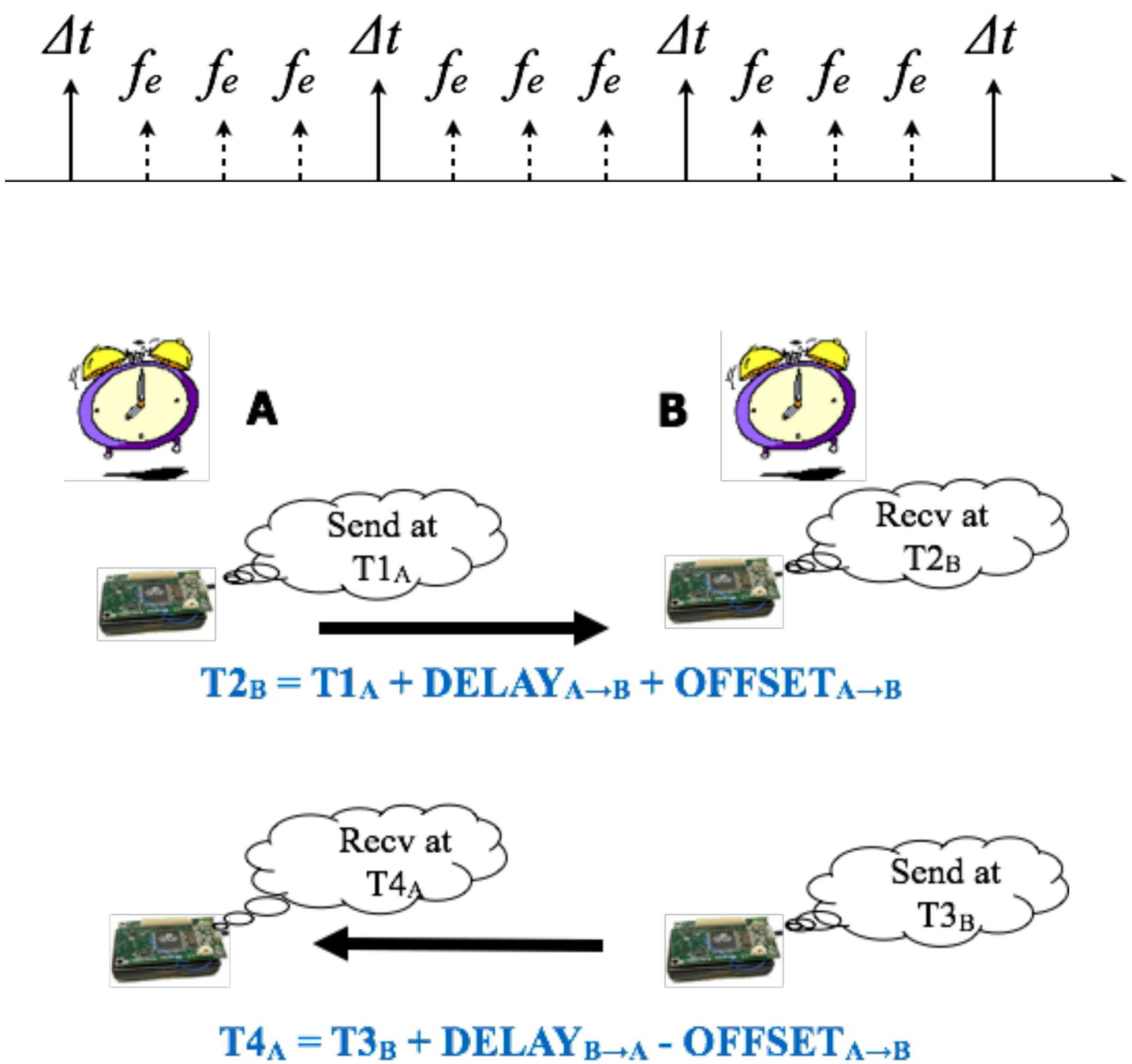
- Time in multi-hop networks
  - Nodes have different QoTs
  - Paths have different characteristics
  - Intermediate nodes may misbehave
  - Path subject to “meaconing” attacks
  - Errors propagate multiplicatively
  - Standard routing uses link quality metric: suboptimal for QoT
- Time Information Routing Protocol (TIRP)
  - Selects the node with stable clocks and routes tine information around nodes with unstable clocks
- Next steps
  - General QoT metrics
  - Optimally fuse multiple references,
  - Adapt messaging based on node and path characteristics





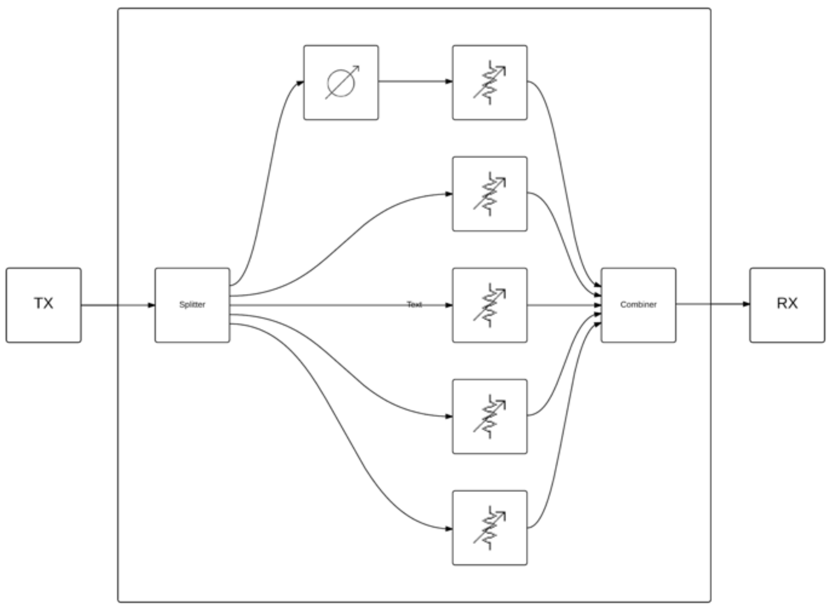


Tunable Time Sync for Multihop Wireless (contd.)

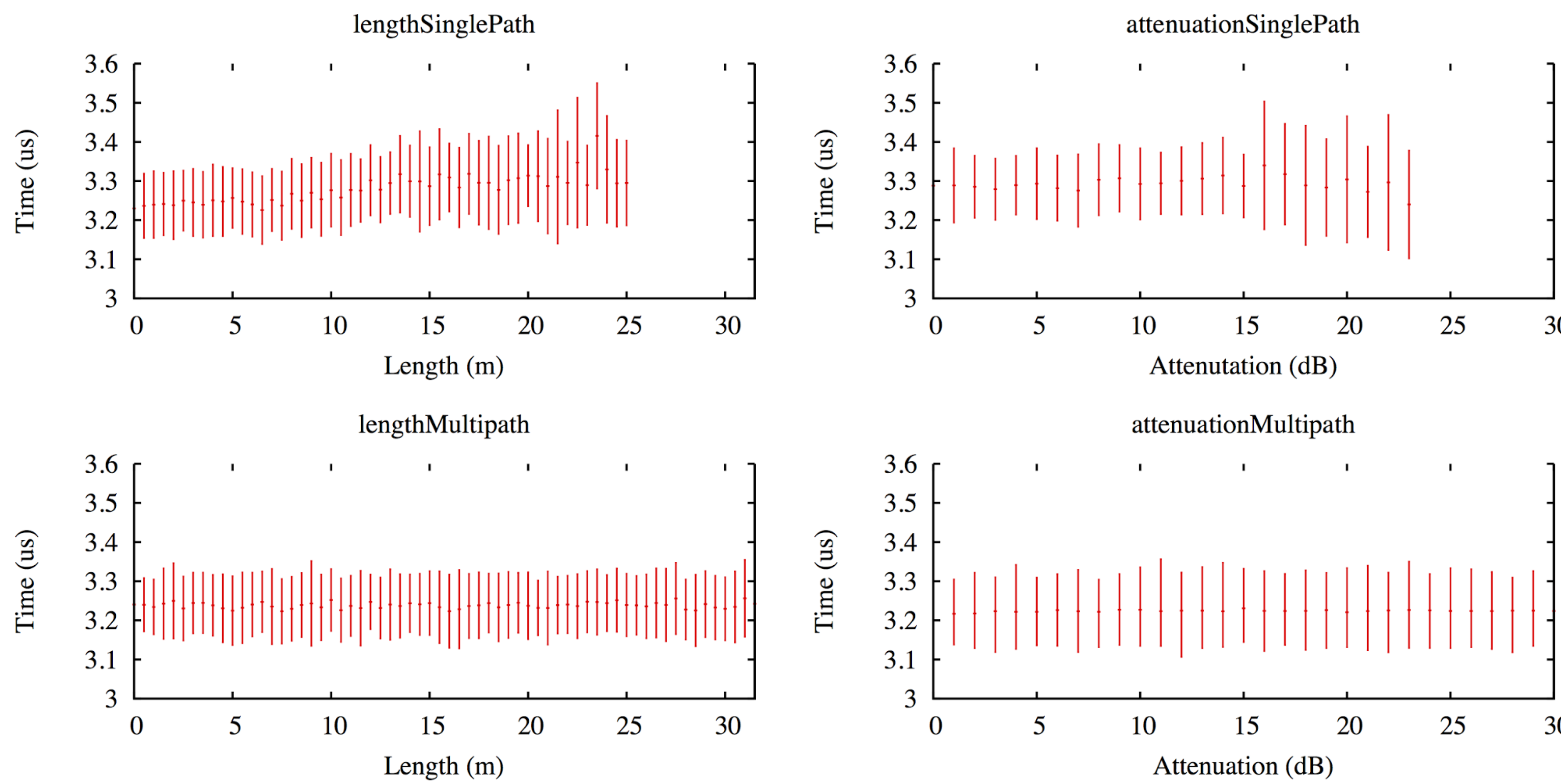


Tunable Parameters
Message Handshake (Uni- vs. Bi-Directional)
Pairwise vs. Broadcast (Uni- vs. Bi-Directional)
Messaging Schedule
Drift Model (Complexity, Window)
Routing Topology (Static vs. Dynamic)

Measuring Impact of RF Channel on QoT



Five-path channel emulator with programmable path lengths, phases, and attenuations.



- TI CC2420 with SFD signal, and an RF Channel emulator for different distances, interference, and multi-path scenarios
- Even in heavy multi path environments, and at high path attenuation, SFD accuracy is stable down to a few hundred ns.

Project Activities

Thrust 2

Mechanisms and Interfaces for QoT-aware Applications

Thrust 1

Autotuning Time Service for Efficient & Resilient QoT

Thrust 3

Circuit and Architecture Support for QoT Adaptation

Thrust 4

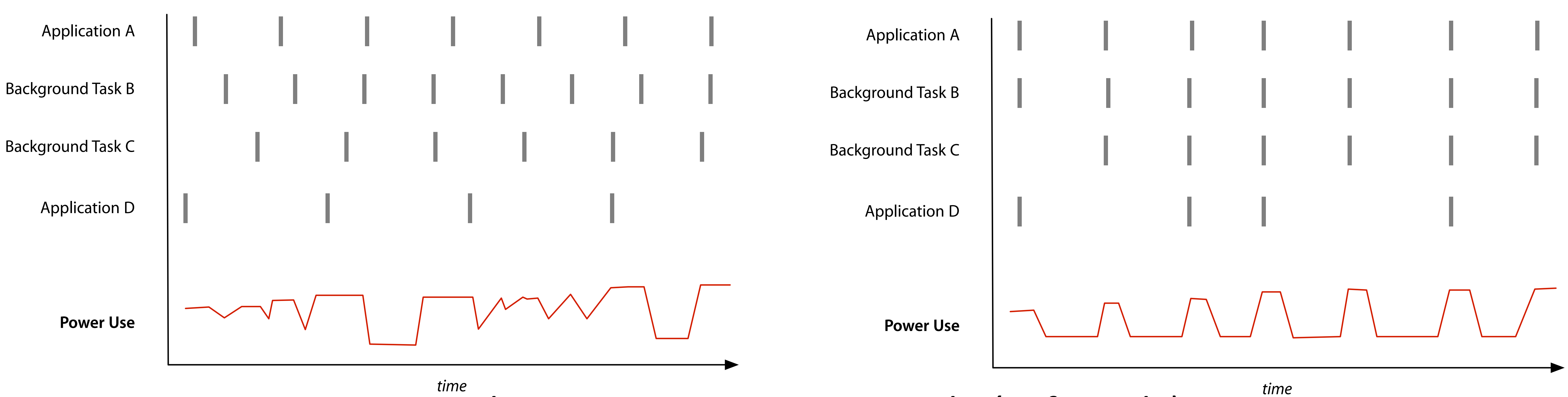
Experimental Testbeds

Thrust 5

Education & Outreach

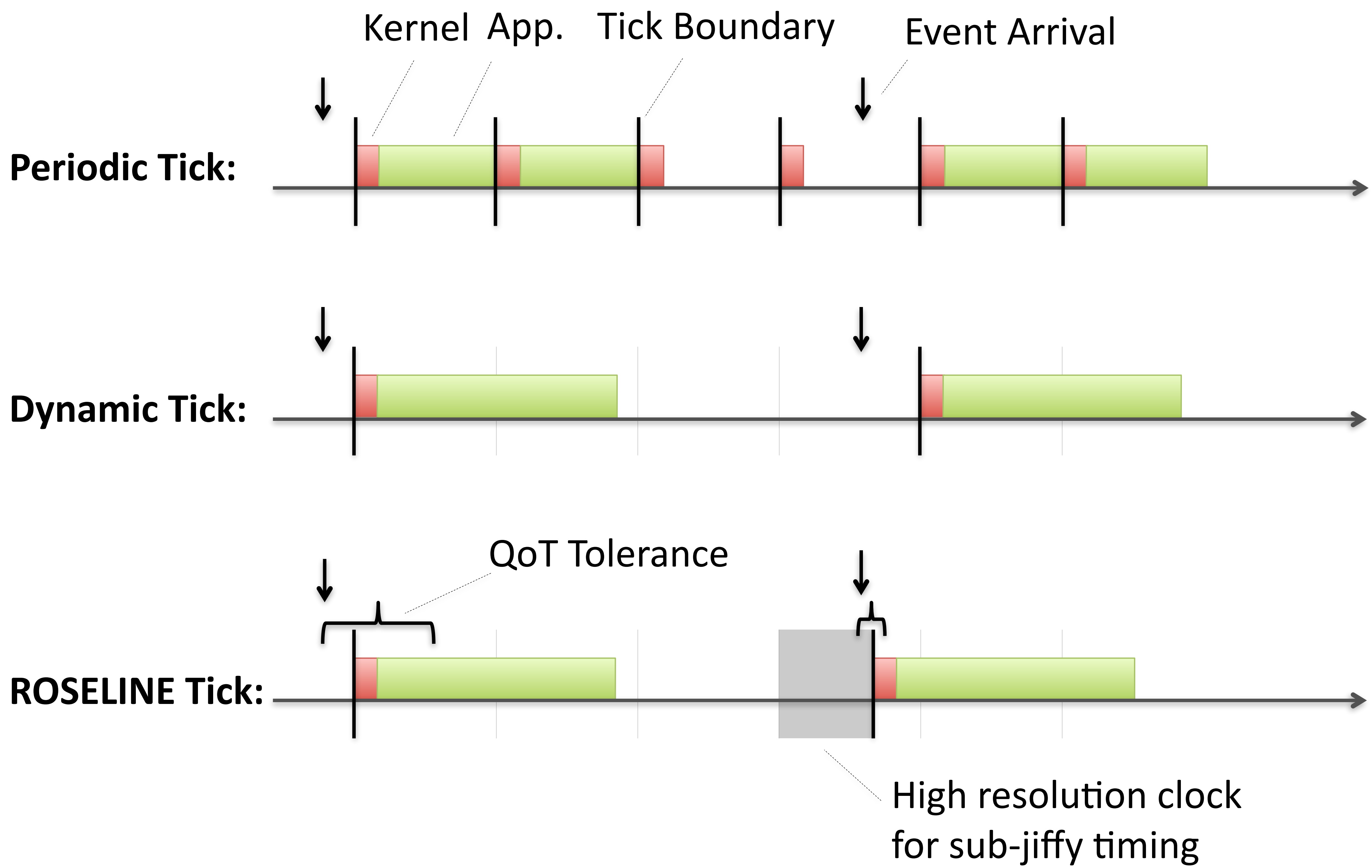
QoT at the Application-System Interface

- Applications express rich timing properties
  - Period, deadline, worst-case execution: {T, D, C}
  - Permit tolerance in specification: $\{T, [-\Delta T_1, \Delta T_2], D, [-\Delta D_1, \Delta D_2], C\}$
- OS and compiler use tolerance to *schedule resources*, *control clock source* and *adapt time synchronization protocol settings*
- Temporal fault management can be implemented as a part of exception handling in PL

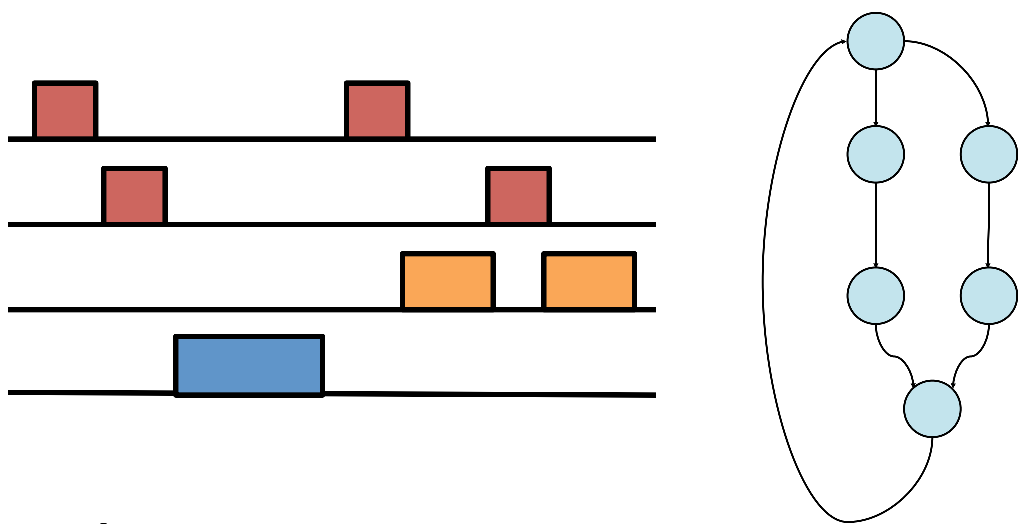


Timer Coalescing in Mac OS X Mavericks (Ref: Apple)

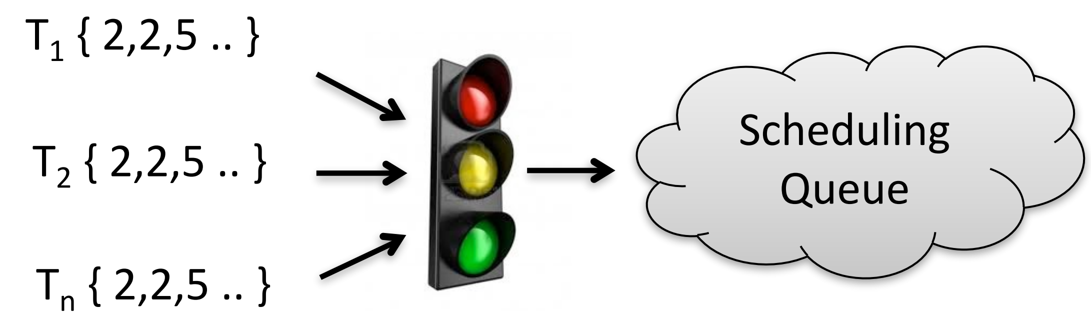
Rethinking the OS Tick



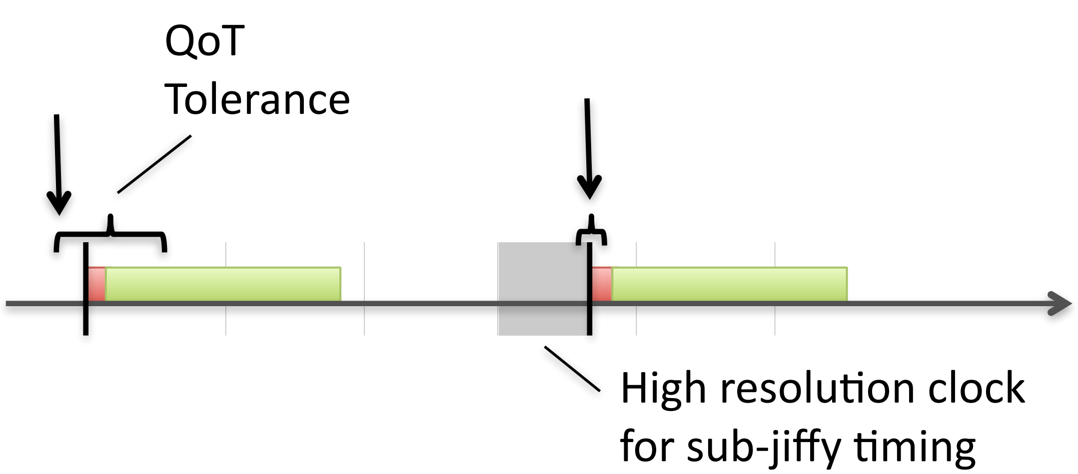
QoT-aware OS APIs and PL Abstractions



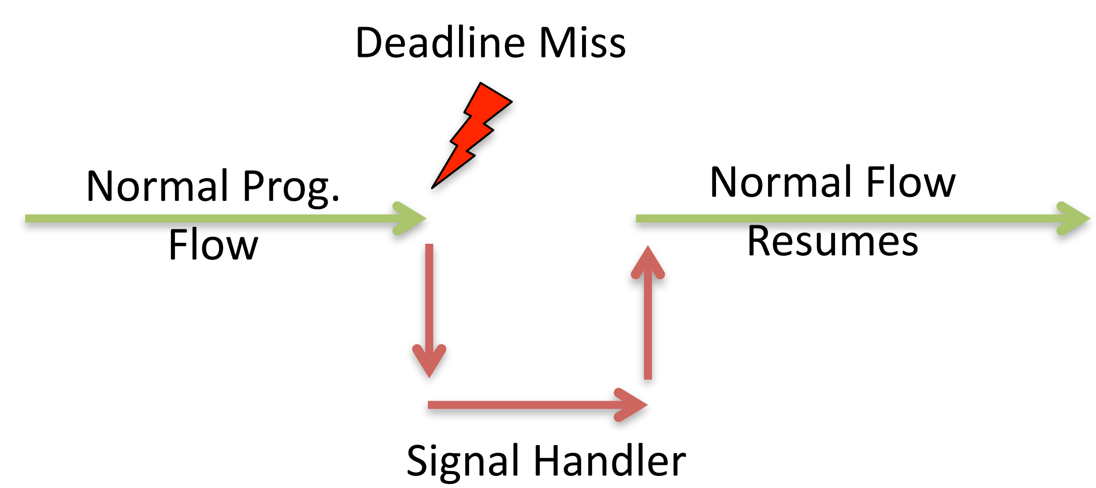
Tasking Timing Requirements



Admission Control

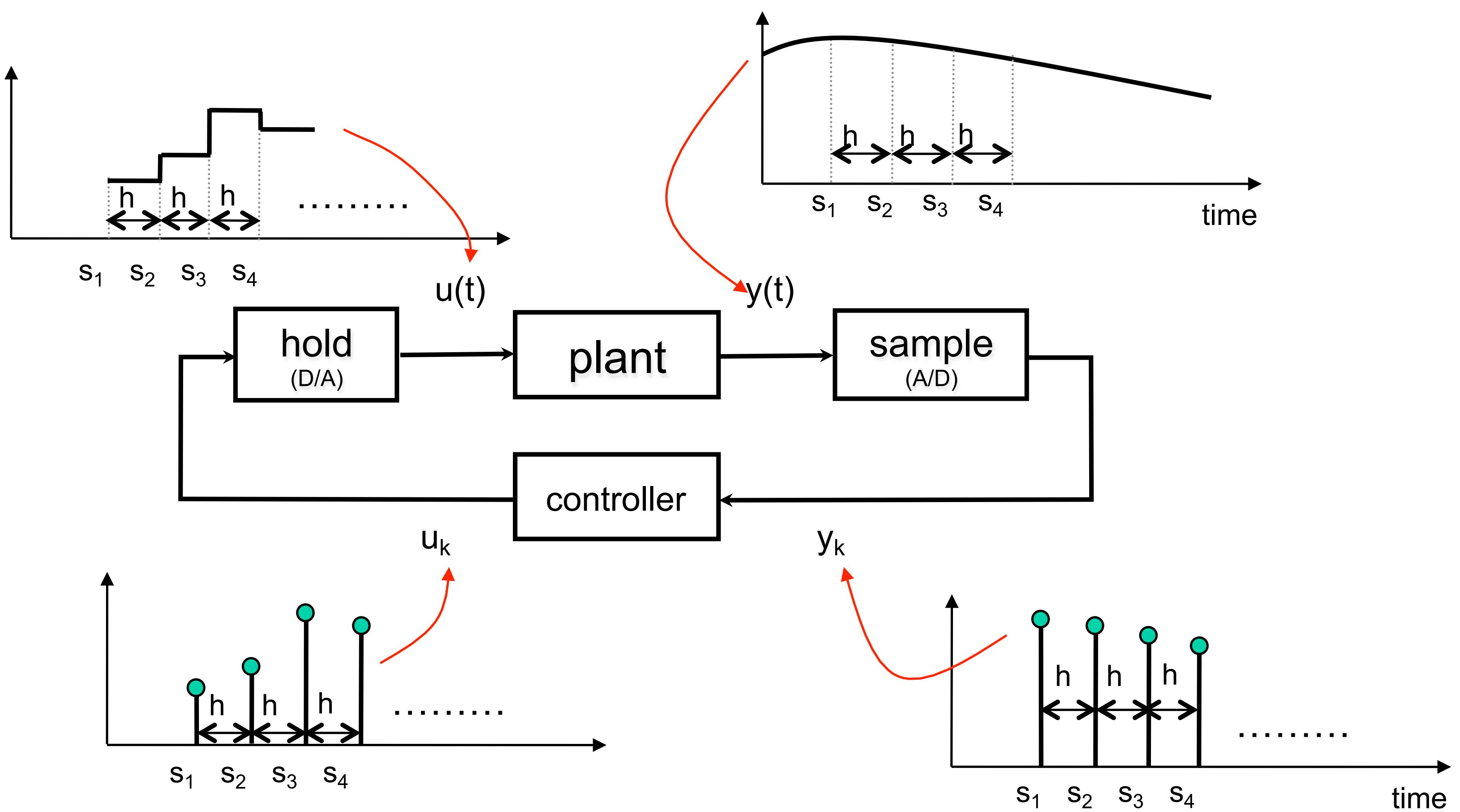


Precise Scheduling and Time Stamping



Temporal Failure Handlers

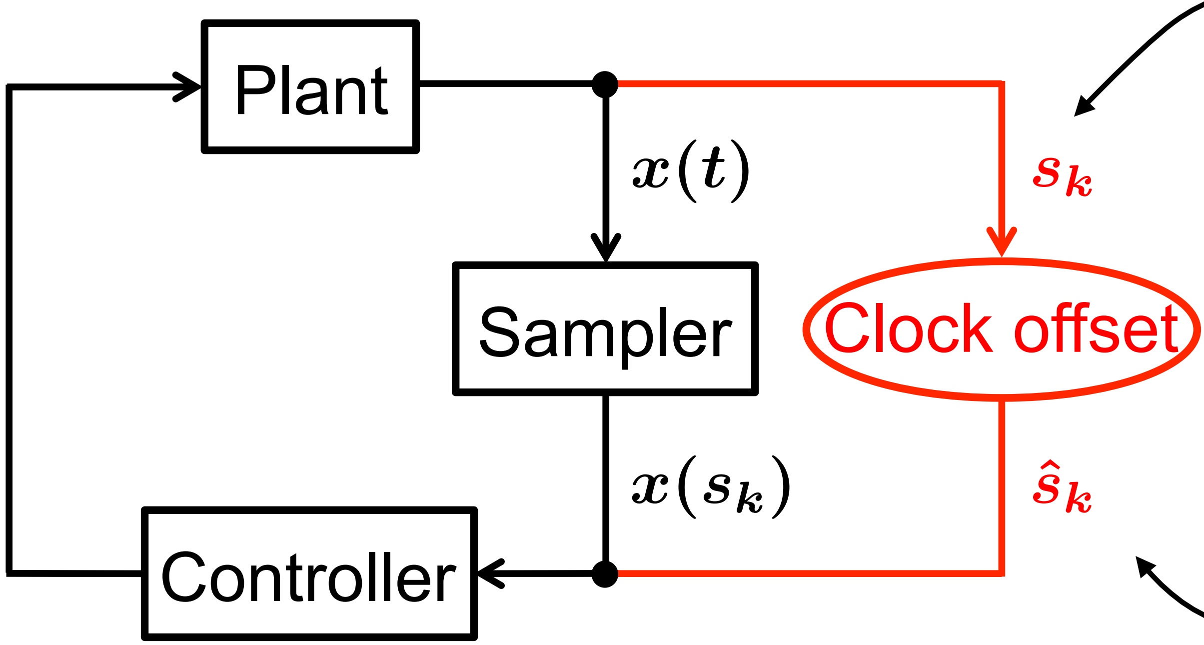
Digital Control Applications: Classical Synchronous Model

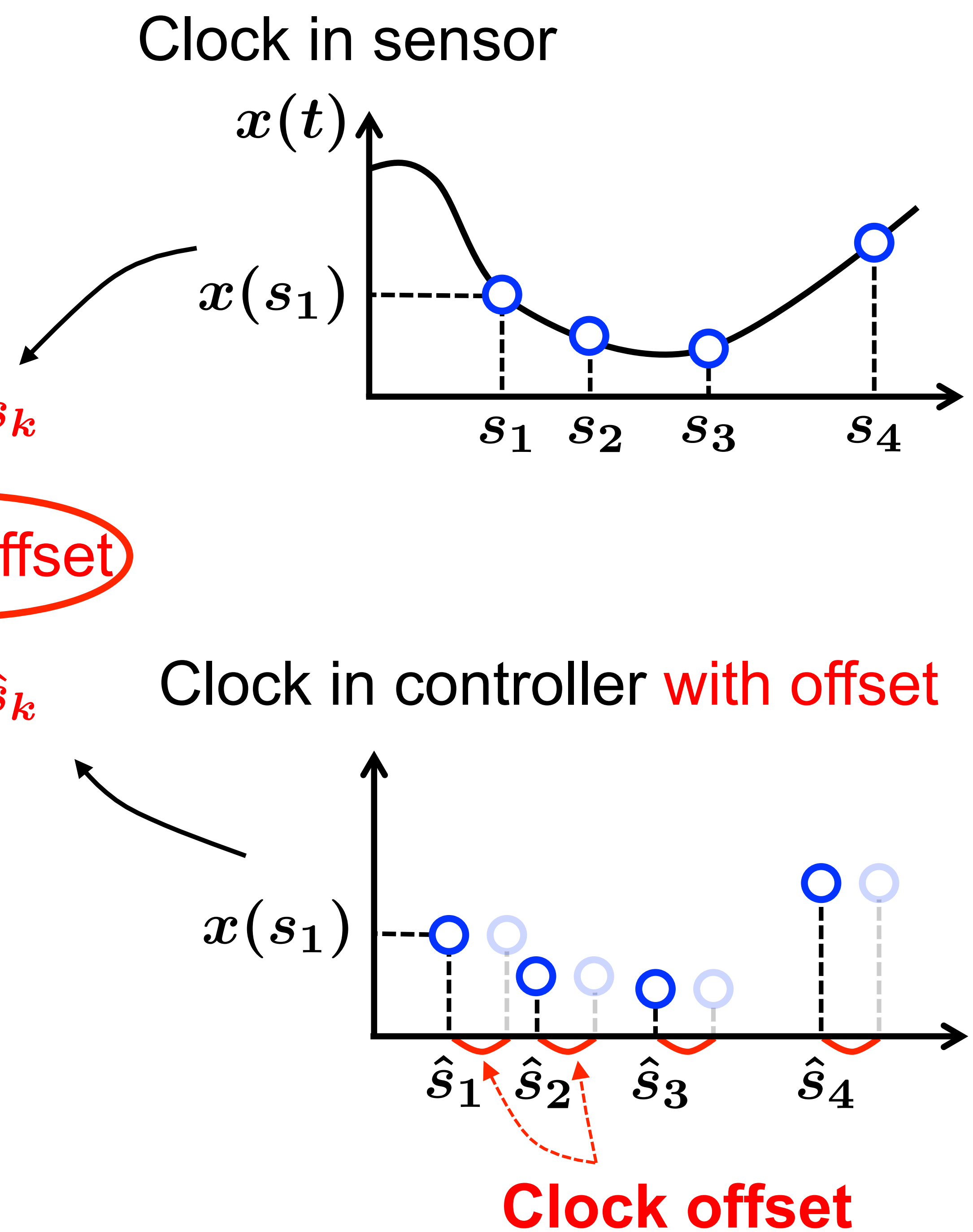


- Constant sampling time
- Sampling synchronized across nodes
- Negligible transmission delays

$$s_{k+1} - s_k = h$$

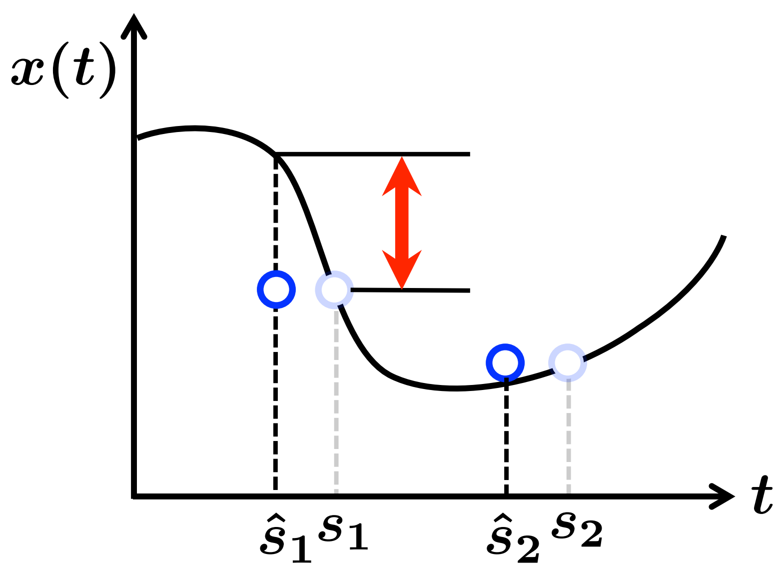
Impact of Time Synchronization Errors
(due to Sensor Interface and Network)





How large an offset can tolerated without leading to instability?

Impact of Time Synchronization Errors



Example:

$\dot{x} = ax + u$

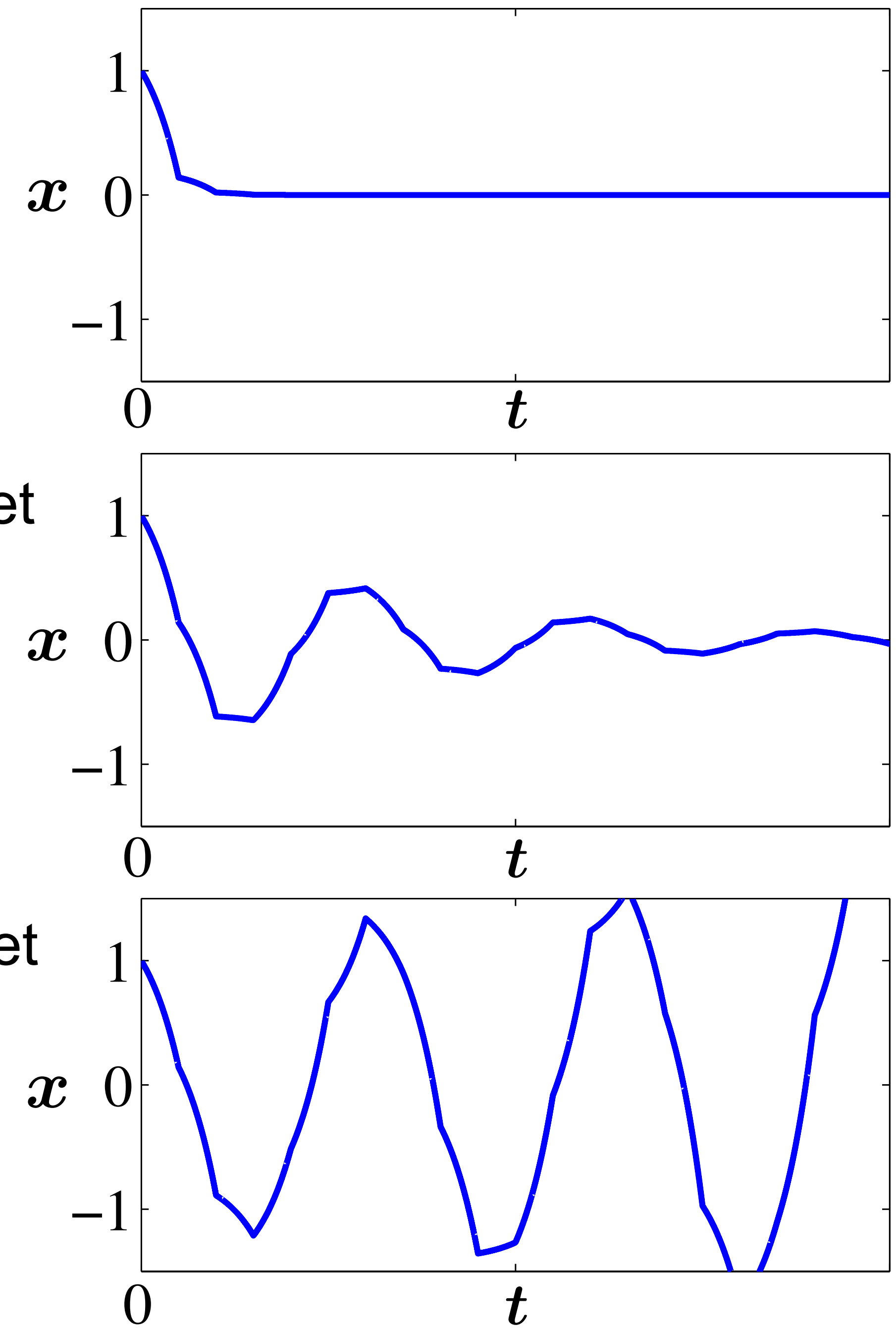
u : periodically updated static state feedback

How large an offset can tolerated without leading to instability?

No offset

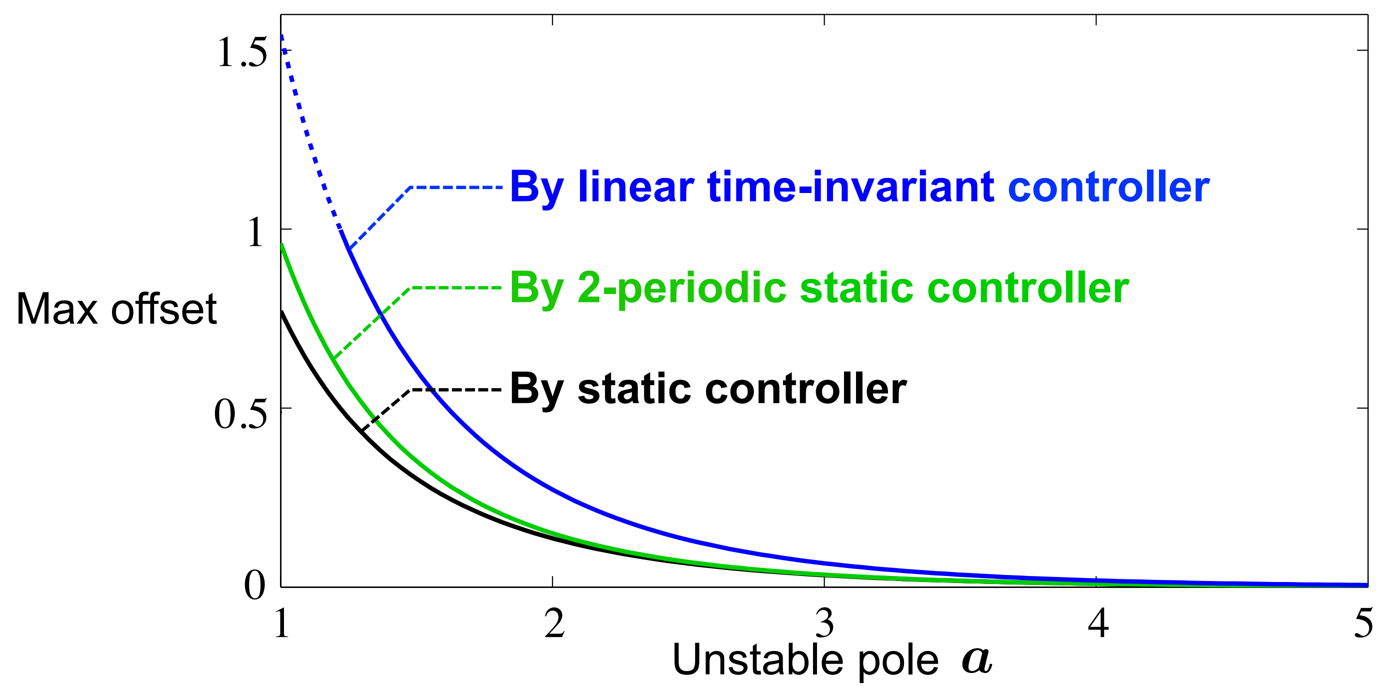
Small offset (20% of T_s)

Large offset (30% of T_s)



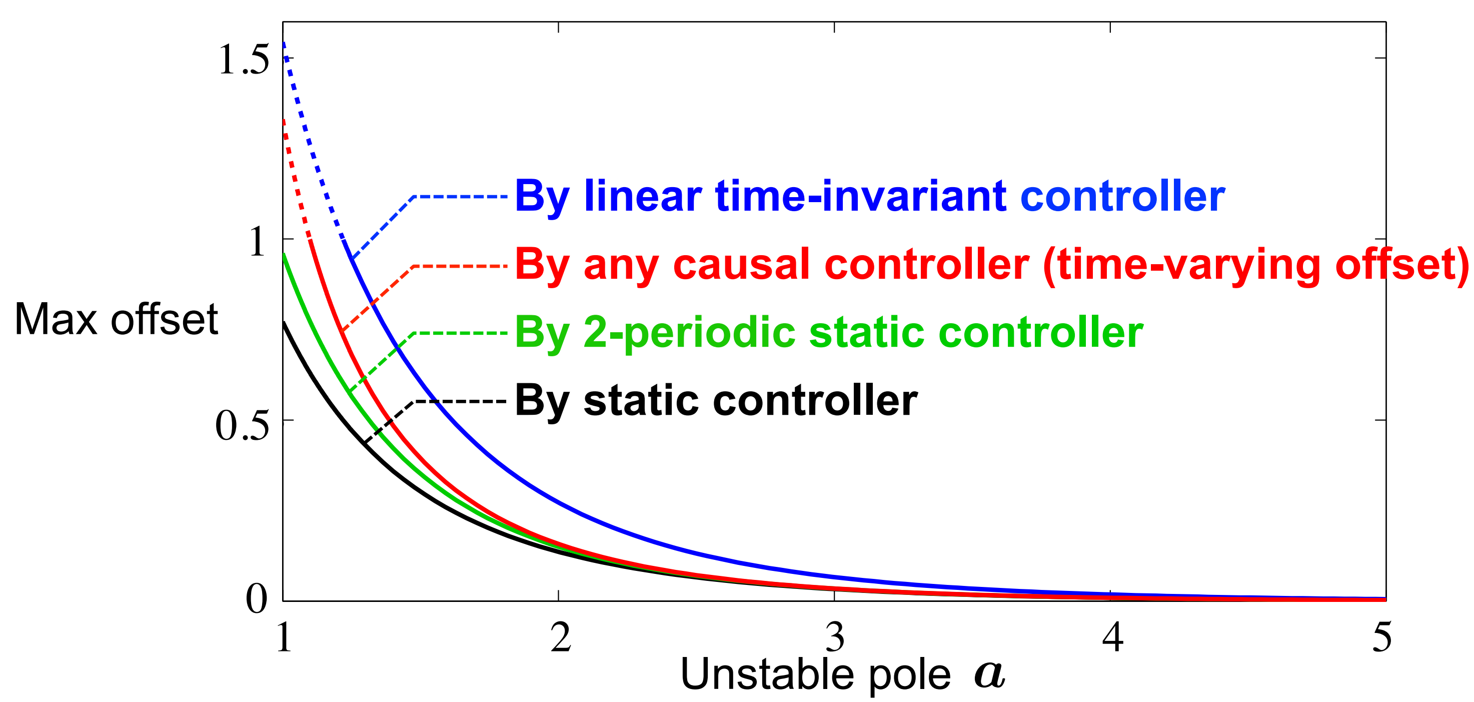
Initial Result: Maximum Allowable Clock Offset

Example: Scalar system $\dot{x} = ax + u$, Sampling interval $= 1$



Constant Offset

Example: Scalar system $\dot{x} = ax + u$, Sampling interval $= 1$



Variable Offset

Project Activities

Thrust 2

Mechanisms and Interfaces for QoT-aware Applications

Thrust 1

Autotuning Time Service for Efficient & Resilient QoT

Thrust 3

Circuit and Architecture Support for QoT Adaptation

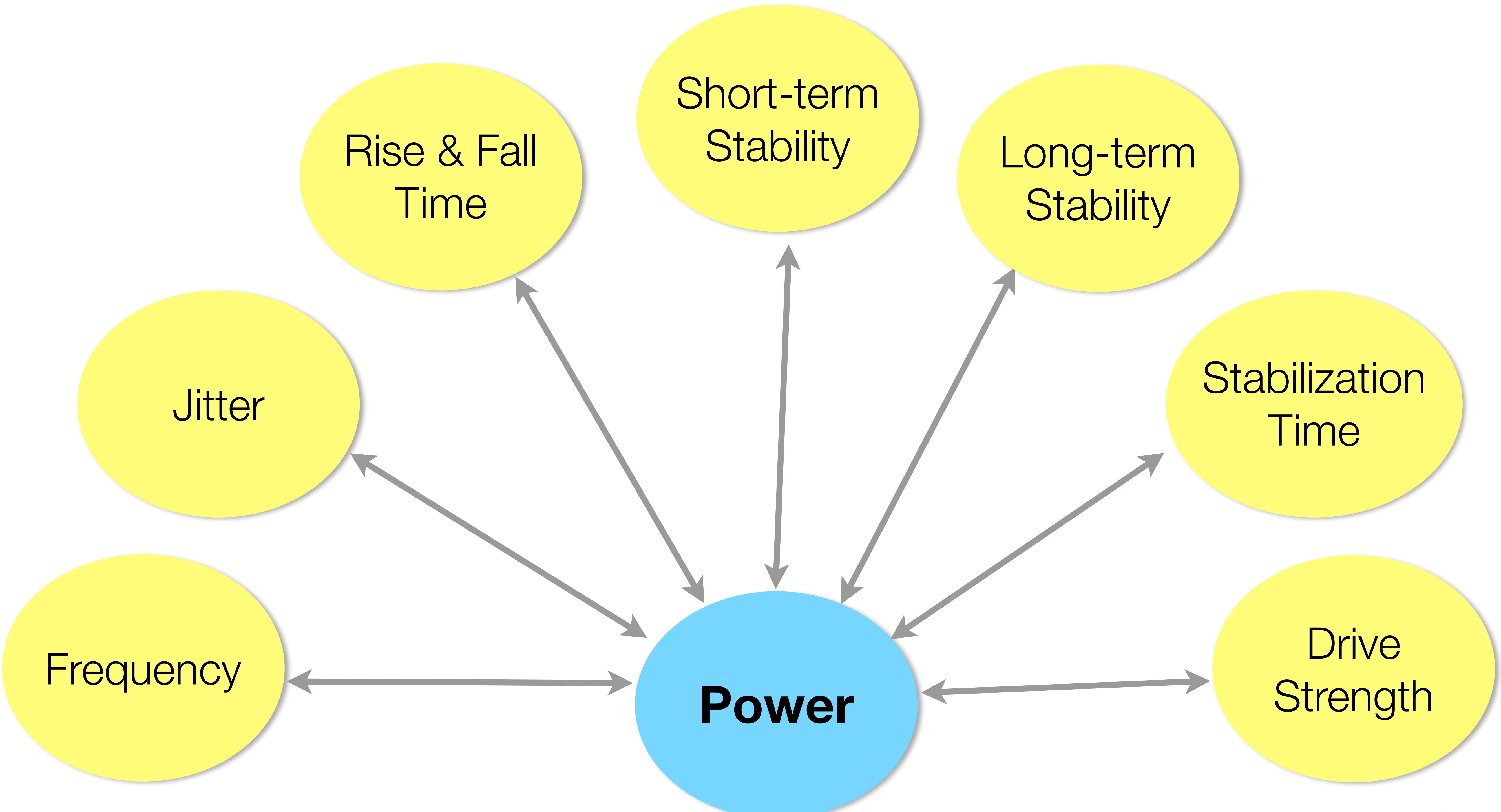
Thrust 4

Experimental Testbeds

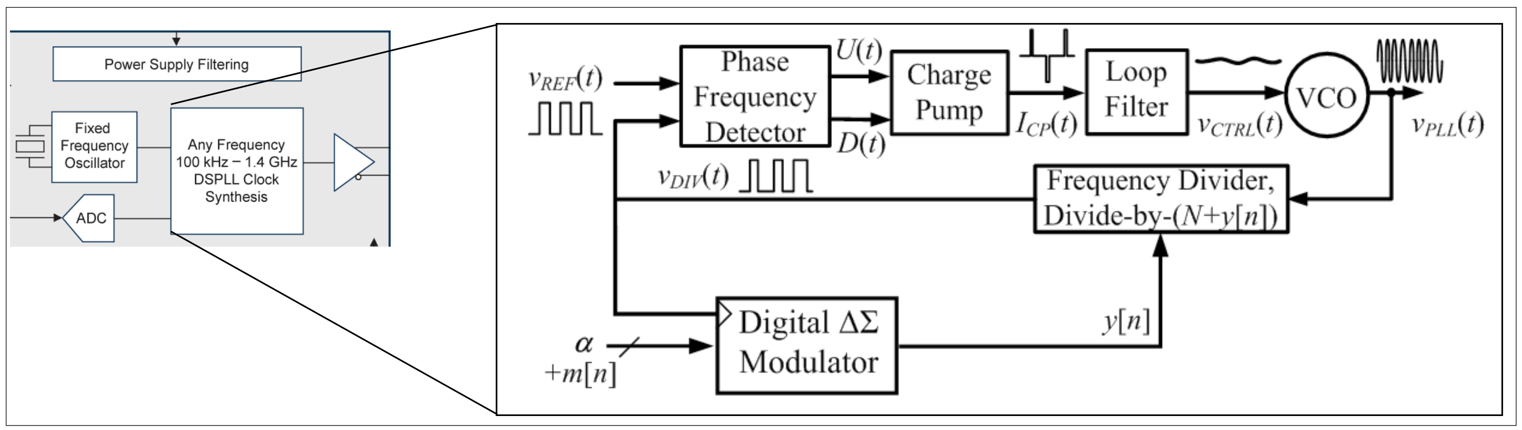
Thrust 5

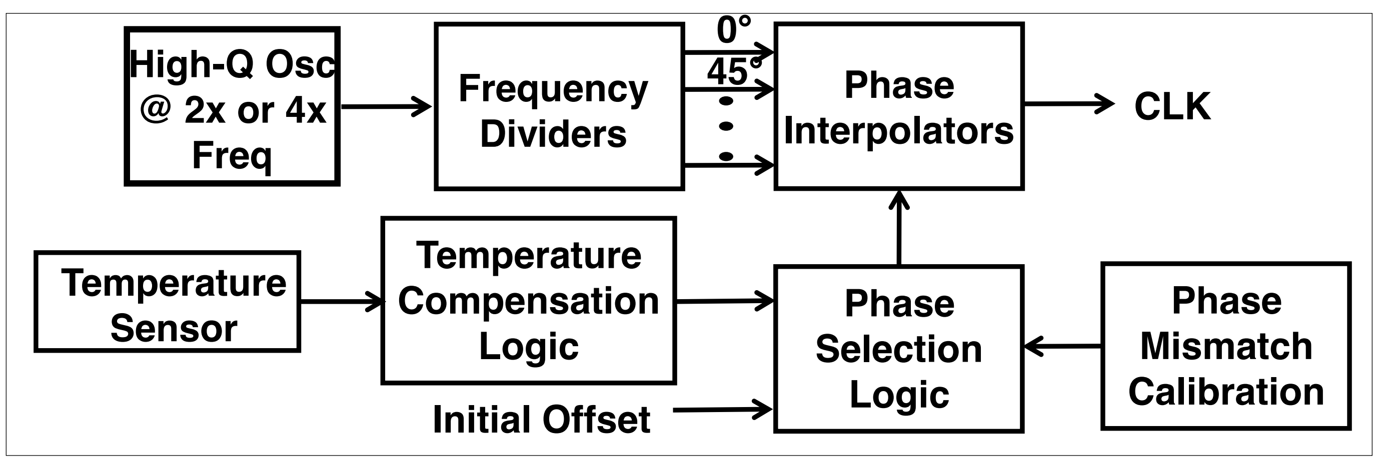
Education & Outreach

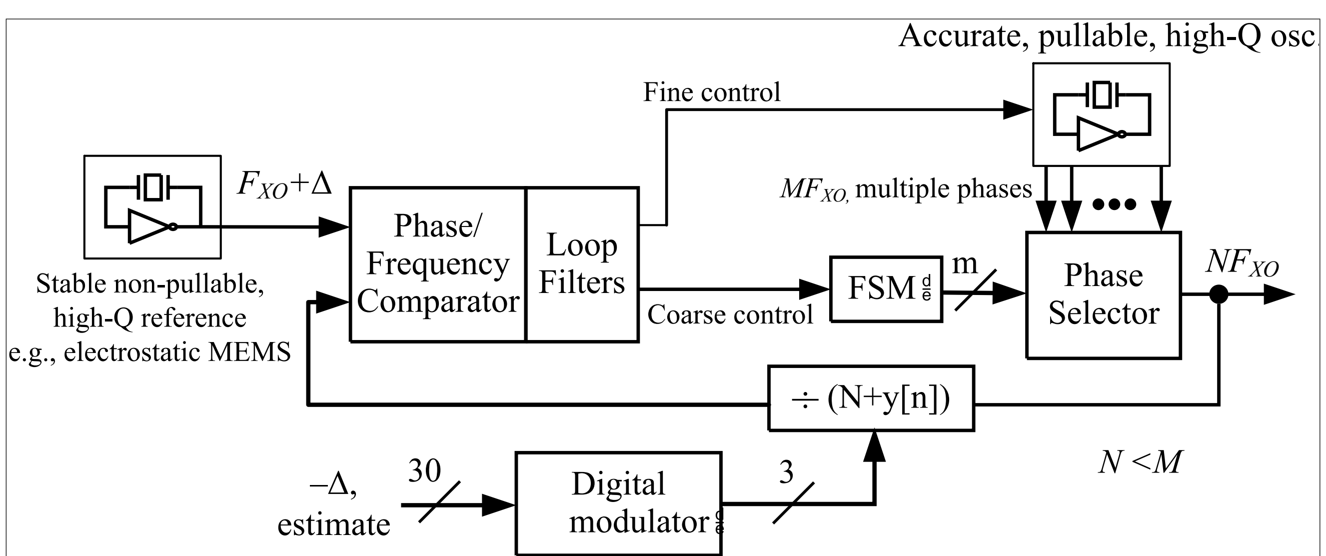
Highly Programmable Clock Generators with *Dynamically Tunable QoT*



Roseline Approach: Efficient Controllability at the Clock Sources







Highly Programmable VCTCXO

- Digital frequency programmability
- power-performance programmability in the fractional-N PLL
- Means of conveying "quality information" to higher layers

PLL-Less Synchronization

- Fractional-N PLL has programmability and compensation, but adds noise and power; High-Q oscillators have very limited pulling range
- Open loop phase switching

Multi-resonator Clocks

- Stability vs. Pullability tension
- Synchronize a stable (e.g. electrostatic MEMS) with a pullable (e.g. piezoelectric MEMS) using a fractional-N PLL
- MEMS Resonators & Si Integration

Project Activities

Thrust 2

Mechanisms and Interfaces for QoT-aware Applications

Thrust 1

Autotuning Time Service for Efficient & Resilient QoT

Thrust 3

Circuit and Architecture Support for QoT Adaptation

Thrust 4

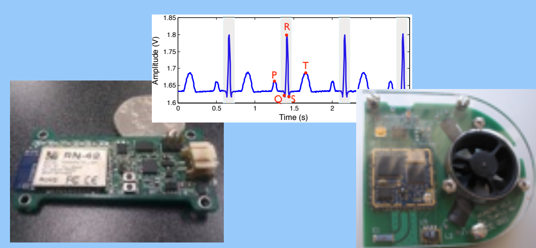
Experimental Testbeds

Thrust 5

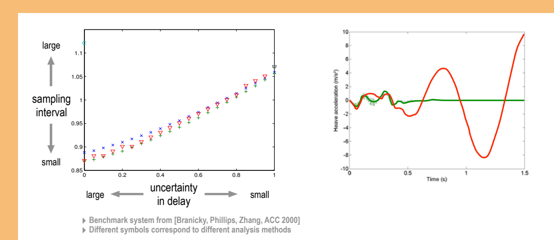
Education & Outreach

Application Testbeds

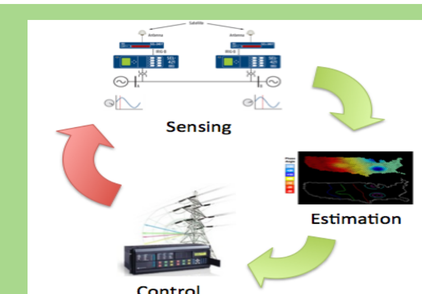
Initial Focus



Ultra-low Power Wireless Wearable/Embedded Sensors

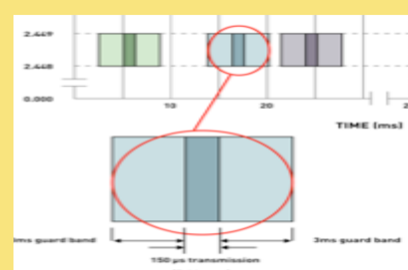


Robust Control of Distributed Systems

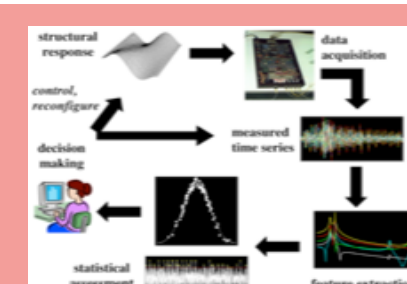


Secure PMU for Smart Electrical Grids

Accurate Underwater Location Sensing



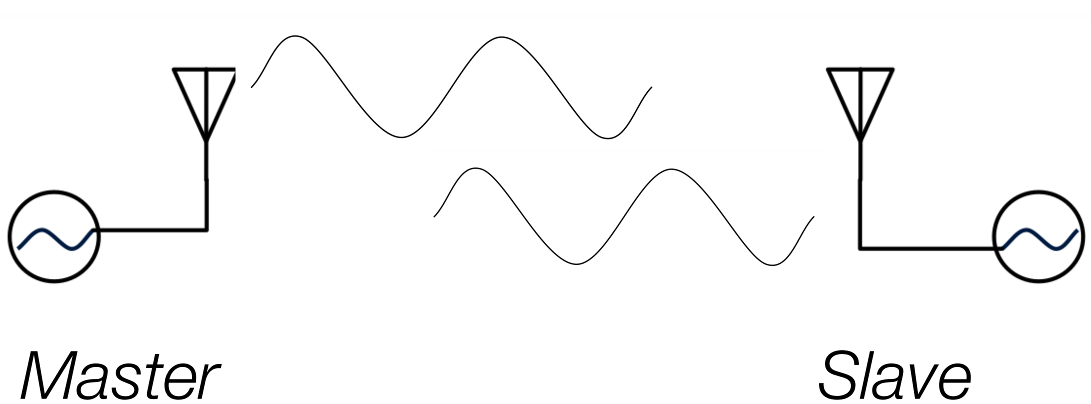
Spectrum-efficient Wireless Devices

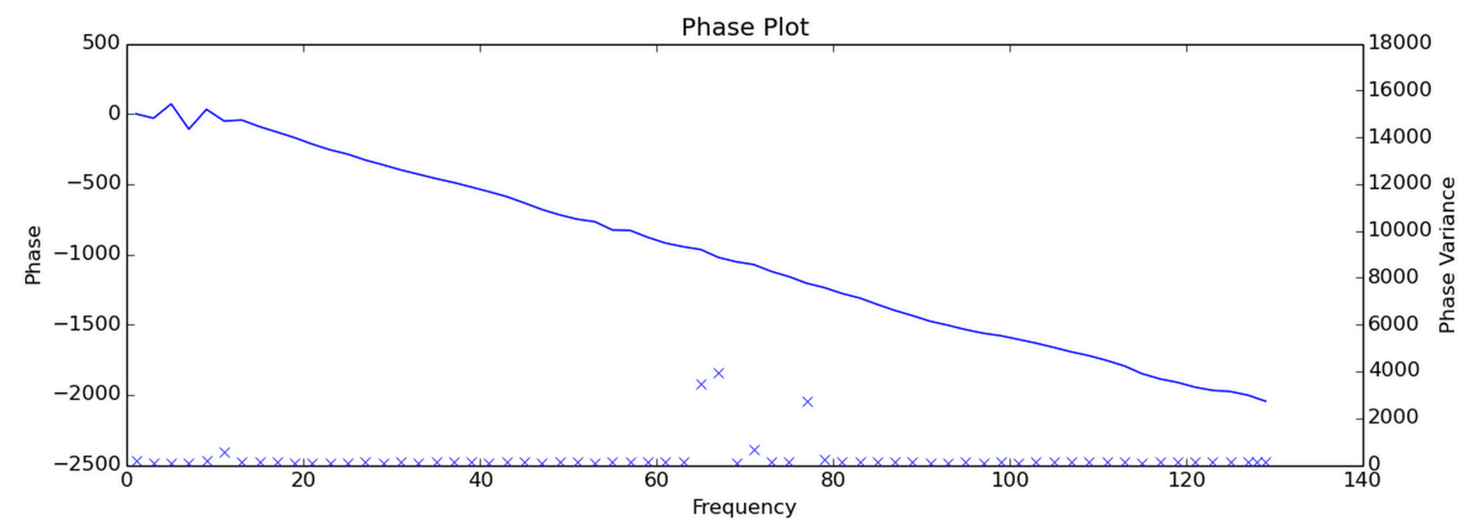


Real-time Structural Health Monitoring

Example: Wireless Sensors with RF Precise Phase Difference Measurement

- Active reflector for precise inter-node distance measurement





- Rapid and precise clock frequency synchronization
- Current activities: (i) Phase synchrony, (ii) Joint time and location, (iii) Impact of clock quality
- Promise: capacity increase in communication (distributed MIMO) and sensing capabilities (distributed beamforming, precisely synchronized sampling)

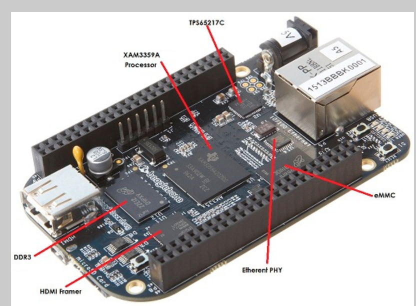
Patwari, Schmid

Engaging Roseline Team Through Experimental Infrastructure

QoT-aware Apps

Control Algorithms
PL Support

QoT-aware APIs



Linux Node

QoT-aware OS Mechanisms
(Tick Management, Scheduling)

Linux RK

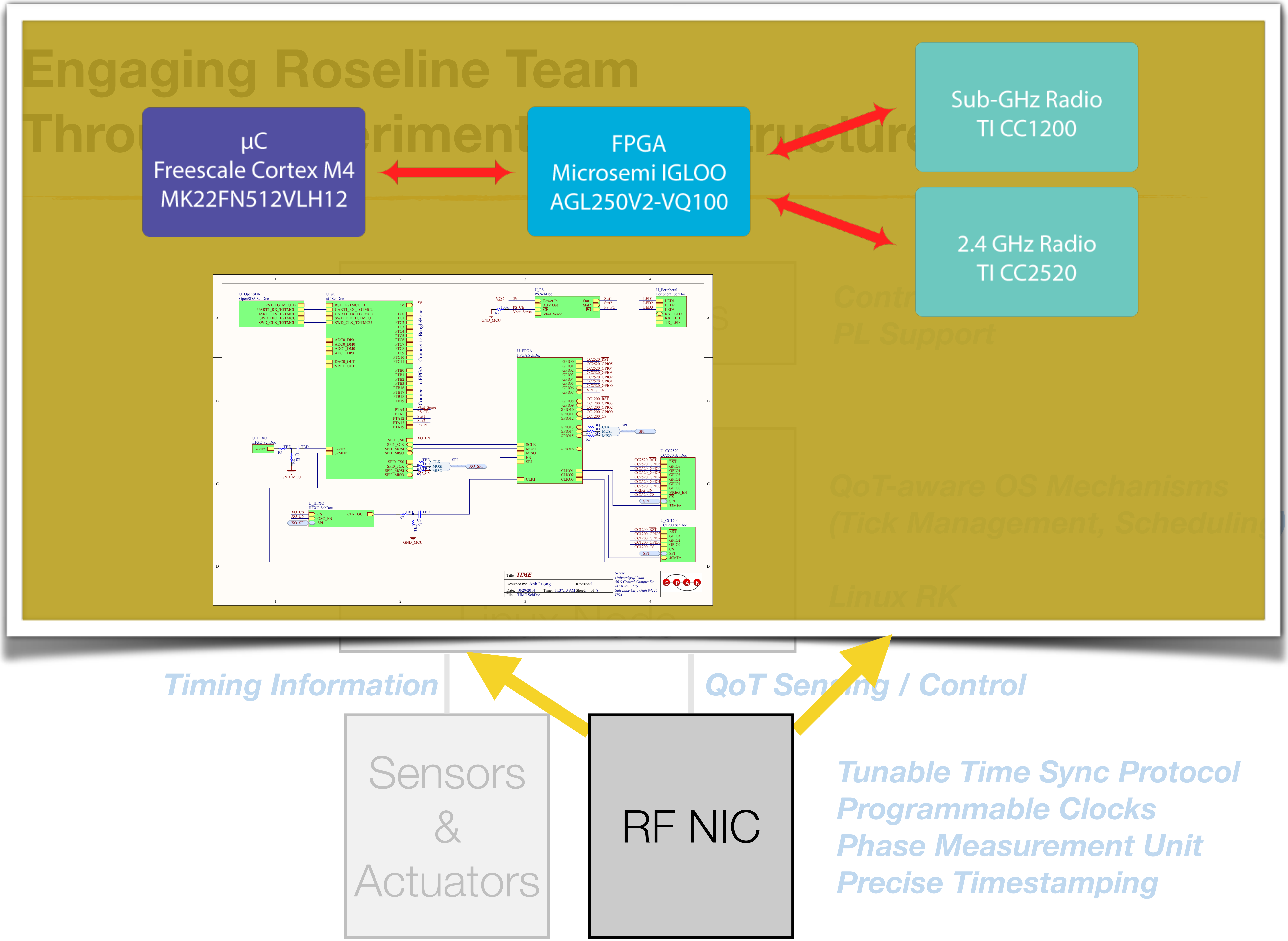
Timing Information

QoT Sensing / Control

Sensors & Actuators

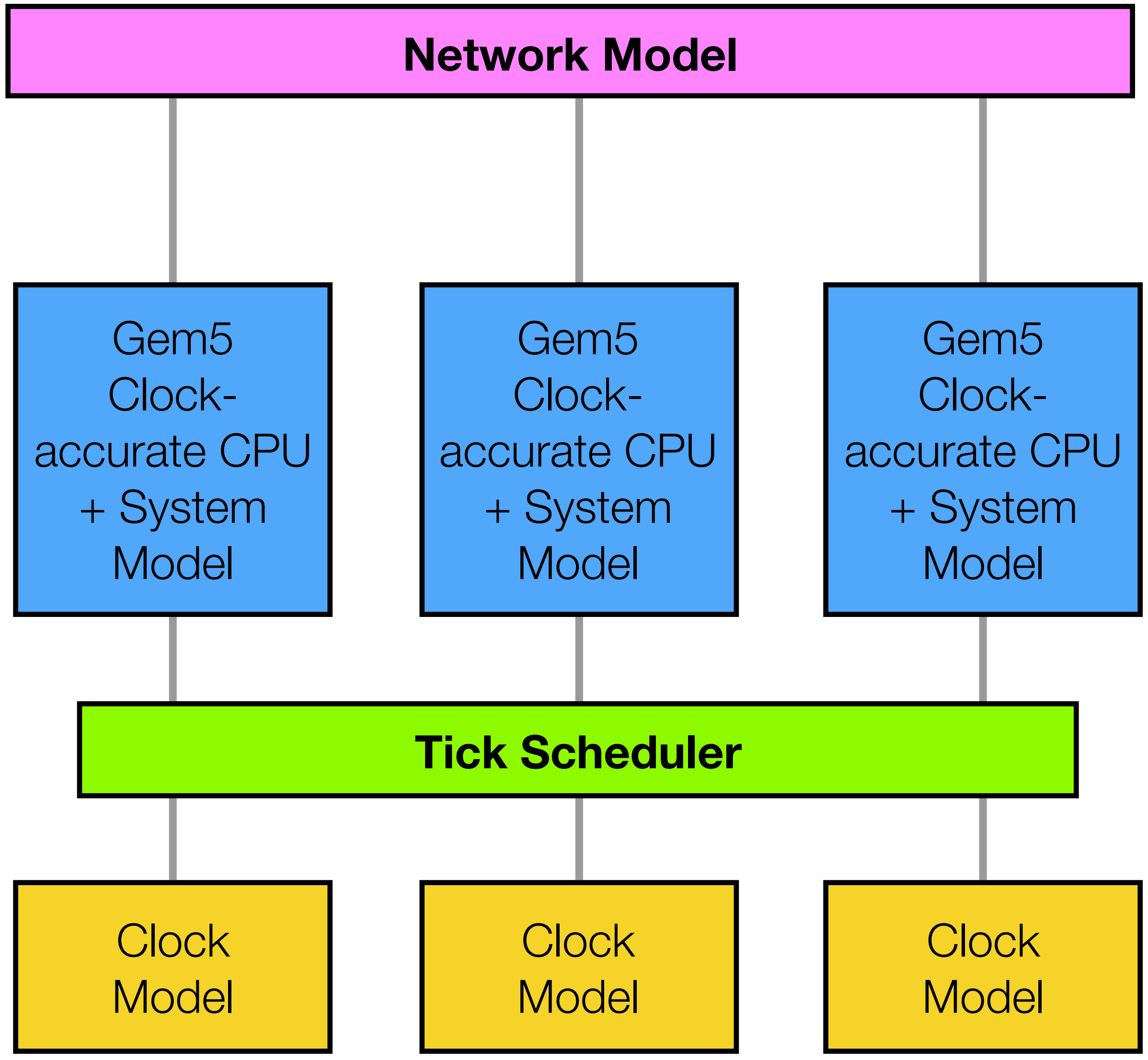
RF NIC

Tunable Time Sync Protocol
Programmable Clocks
Phase Measurement Unit
Precise Timestamping


Engaging Roseline Team
Through Experimental Infrastructure
µC
Freescale Cortex M4
MK22FN512VLH12
FPGA
Microsemi IGLOO
AGL250V2-VQ100
Sub-GHz Radio
TI CC1200
2.4 GHz Radio
TI CC2520
Timing Information
QoT Sensing / Control
Sensors
&
Actuators
RF NIC
Tunable Time Sync Protocol
Programmable Clocks
Phase Measurement Unit
Precise Timestamping

Roseline Simulation Platform

- **Need:** study performance impact of clock and network variations, and develop software mechanisms in simulation
- **Problem:** simulators assume ideal clocks and global time!
- **Approach:** simulation environment where clock and network variations can be modeled, with Linux OS and apps
    - Based on gem5, a modular platform encompassing system-level architecture as well as processor microarchitecture



Project Activities

Thrust 2

Mechanisms and Interfaces for QoT-aware Applications

Thrust 1

Autotuning Time Service for Efficient & Resilient QoT

Thrust 3

Circuit and Architecture Support for QoT Adaptation

Thrust 4

Experimental Testbeds

Thrust 5

Education & Outreach

Bring Appreciation for the Physical into Computer Science

K-9: Inspire Passion

- *Gupta/Naderi's myLab @ UCSD: Combine Art and Engineering with focus on Girls 7-15 (e.g. Girls Hat Day)*
- *Schmid's @ Utah: elementary school via CDC project on sensor-based disease tracking*
- *ROSELINE will add simple actuators & control to the learning experience*

THE myLab PROGRAM

10-12: Challenge

- *Los Angeles Computing Circle & HSSRP @ UCLA, COSMOS @ UCSD, Dos Pueblos Internship @ UCSB*
- *Strong emphasis on physically-based computing*
- *Grad and Undergrads as mentors*
- *ROSELINE will add modules on time-aware computing & networking*
- *Metrics: # going to STEM majors, tracking & mentoring mechanisms*

LACC

COSMOS

UG & Grad: Research Interest & Reusable Artifacts

- *Collaboratively developed course modules and ROSELINE CPS toolkit*
- *CPS Education Initiative via CPS-VO*
- *Research experience for younger undergraduate students*
- *On-line course (Coursera)*
- *Metrics: # of UGs going toMS/PhD*

CPS Community: Engage

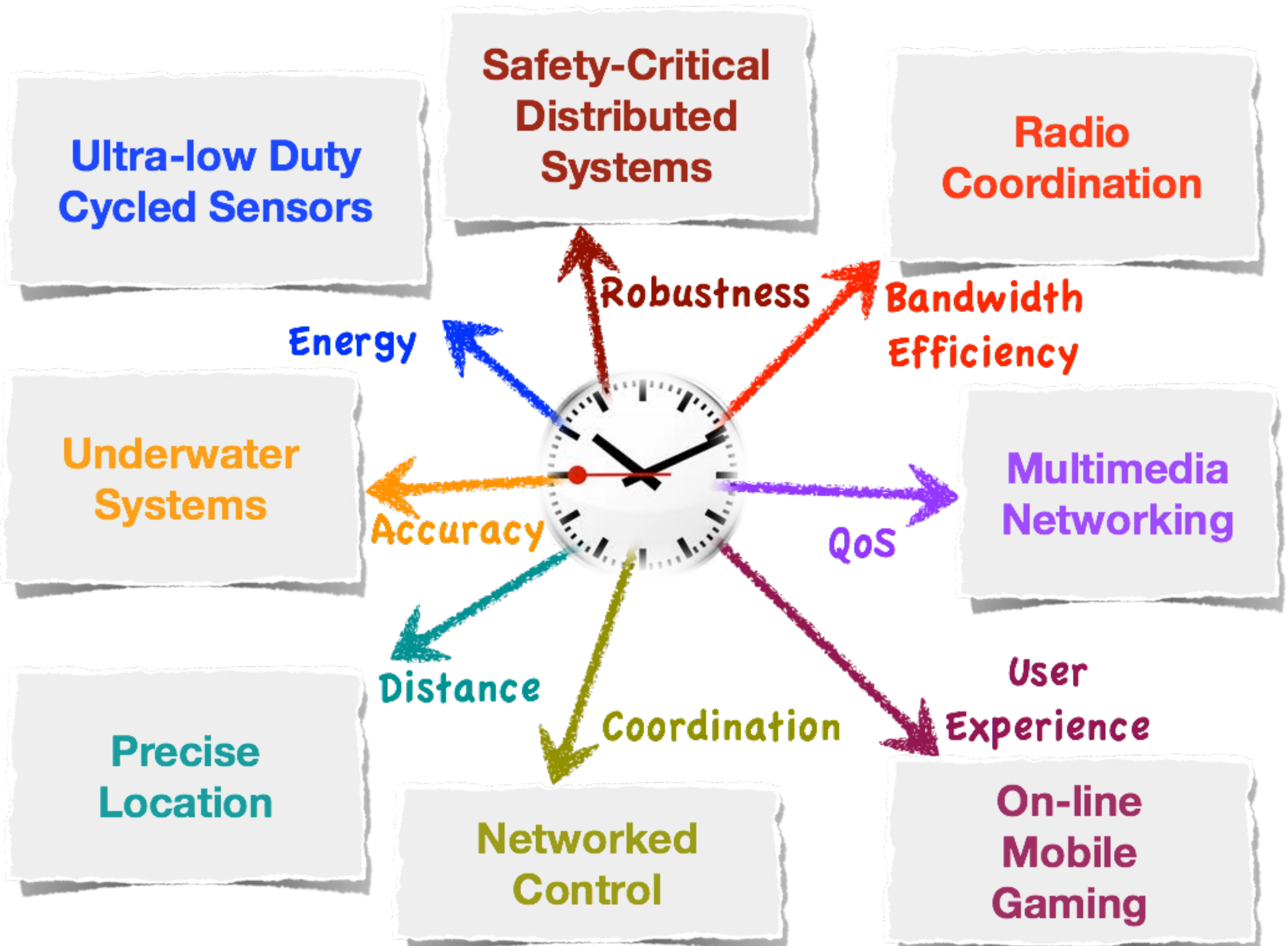
- *Dissemination of tools, designs, data, testbeds, & courseware via CPS-VO*
- *Workshop @ CPSWeek on cross-layer time issues*
- *Industry engagement: Atmel, NI, SiTime, Qualcomm*
- *Other collaborations: IST, IIITD, LANL*
- *External invitees to ROSELINE Meetings*

SiTime

INSTRUMENTS

Interacting with TAACCS @ NIST/CMU

Roseline: Defining QoT and Making It Visible & Controllable across the System



Improve robustness, bandwidth, energy, buffer, location, and user experience

Enable time-centric apps in systems & domains with intrinsic time variability

Advance state-of-the-art in clocking circuits and platform architectures

Drive CPS research with a deeper understanding of time & its system trade-offs

Thank you!