

8. Time Synchronization Via Sensing

- **Background:** Time synchronization is an essential part of networked embedded devices. It enables various distributed applications like smart homes, automated agriculture and autonomous vehicles. However, many devices at the edge are resource constrained and existing timing services are a burden on their resources. On the other hand most of these devices are equipped with sensors and send periodic sensing data to gateway servers. We want to leverage this time-stamped sensor data received at the gateway to synchronize the sensing devices
- **Goal:** Develop a time sync protocol using timestamped sensor data from two embedded devices. The data will be received by a raspberry pi device, and the synchronization protocol will run on the raspberry pi device
- **Deliverables:**
 - Characterize the network delay between raspberry pi and the edge device
 - Estimate the relative clock drift between the participating devices
- **Hardware:** ESP32 Things, Raspberry Pi
- **References:**
 - HAEST: Harvesting Ambient Events to Synchronize Time across Heterogeneous IoT Devices
 - Automated Synchronization of Driving Data Using Vibration and Steering Events
 - Exploiting Smartphone Peripherals for Precise Time Synchronization
- **Team Size:** 2
- **Prerequisite:** Experience with programming embedded microcontrollers (e.g. 8051, AVR), Bluetooth communication, Basic Machine Learning Algorithms

10

Kian Hesemeyer- kahesemeyer@umass.edu

Kyle Greenwood- khgreenwood@umass.edu

Deliverables & Timeline

- Motivation, design goals, deliverables, system blocks, hw/sw requirements, team members responsibilities, project timeline, references
 - We like this project because we believe it applies to a lot of the concepts we will be using in our senior design projects. We also like that it involves a combination of hardware and software . Both of us have experience working with AVR and various ESP 32 devices.
 - The design goal is to have a central computer (raspberry pi) reading from two different sensor devices that send data back. We will analyze the constant stream of data to identify a common event such as an item falling or door closing. Then using the time stamps from each sensor the central computer can synchronize the two incoming data streams
 - Deliverables
 - Characterize the network delay between raspberry pi and the edge device
 - Estimate the relative clock drift between the participating devices
 - Hardware requirements
 - Raspberry pi that is connected and reads in sensor data from multiple sensors (IMU, sound, or distance)

- Software requirements
 - Read and analyze data from each sensor with timestamps from each report
 - Identity an event based off sensor readings
 - Align the
- Project timeline
 - Week 1: Research
 - Week 2: Obtain resources
 - Week 3: Assemble parts
 - Week 4-5: Read data from sensors
 - Week 6-7: Analyze signals and identify events
 - Week 8: Start developing synchronization protocol
 - Week 9: Complete synchronization protocol and analysis
 - Week 10: Write paper
- References
 - HAEST: Harvesting Ambient Events to Synchronize Time across Heterogeneous IoT Devices
 - Automated Synchronization of Driving Data Using Vibration and Steering Events
 - Exploiting Smartphone Peripherals for Precise Time Synchronization
 - Mention team member's lead roles: setup, software, networking, writing, research, algorithm design
- Responsibilities and roles
 - Kian: research, hardware, algorithm design (event analysis)
 - Kyle: research, networking, algorithm design (synchronization)
- System block:

