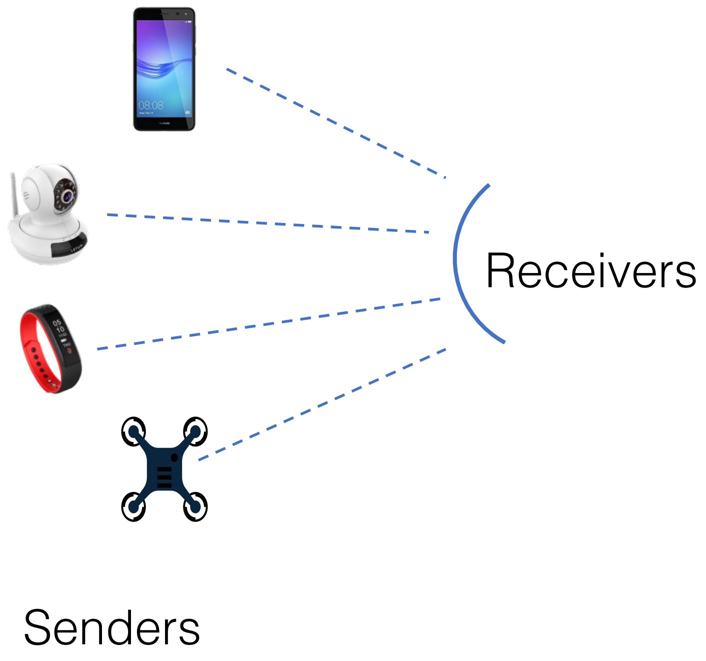
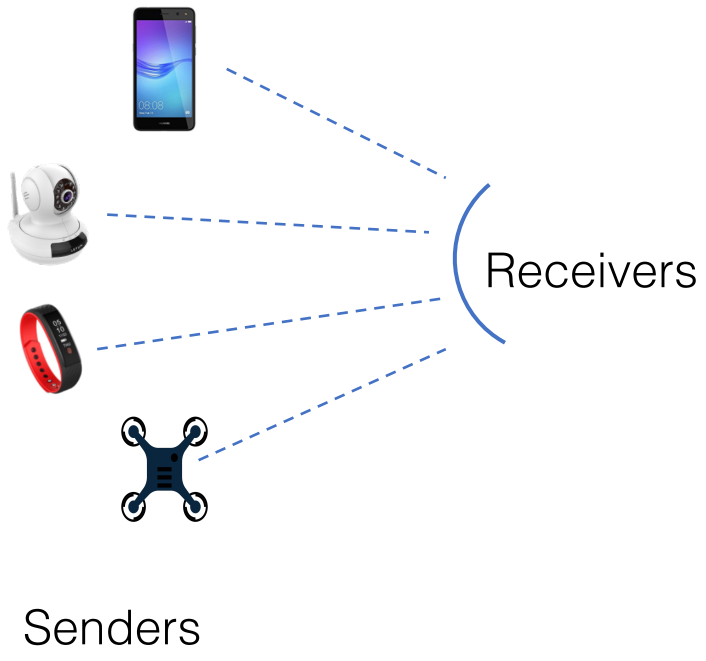
Understanding Timing Errors in Datasets

Sandeep Singh Sandha
Networked & Embedded Systems Lab
University of California, Los Angeles

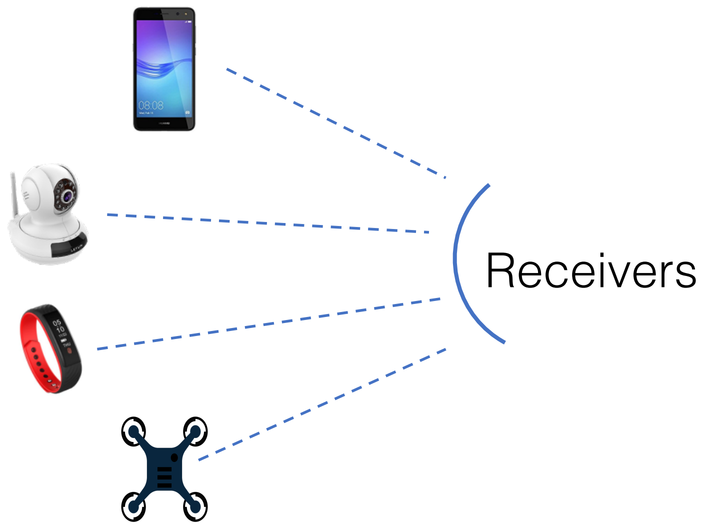
Data: Collected from Edge Devices



Data Timestamp Uncertainties



Data Timestamp Uncertainties



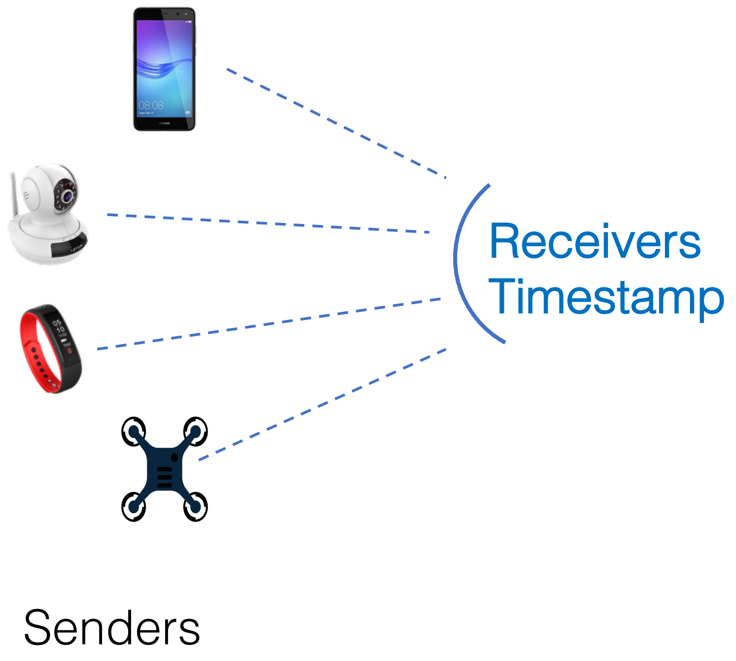
Senders Timestamp

Timestamp at the sender: Time synchronization is needed.

1. Clock's oscillation frequency depends on temperature, age, etc.
2. Elapsed time is an integration of frequency, so open loop time diverges.

Data timestamp error: sync error, system software mismanagement, drift and temperature.

Data Timestamp Uncertainties



Timestamp at the sender: Time synchronization is needed.

1. Clock's oscillation frequency depends on temperature, age, etc.
2. Elapsed time is an integration of frequency, so open loop time diverges.

Data timestamp error: sync error, system software mismanagement, drift and temperature.

Timestamp at the receiver: Data transfer from senders to receivers.

Data timestamp error: variable networking delays, receiver timestamp delays, sync error at receiver.

Outline: Timing Errors in Datasets

Focus on: Data timestamp uncertainty

- Quantify the timestamp uncertainty in smartphones [1].
- Impact on deep learning classifier accuracy [1].
- Improve deep learning classifier resiliency to timestamp uncertainty [1, 2].
- Improve time across smartphones [3].

*Equal Contribution

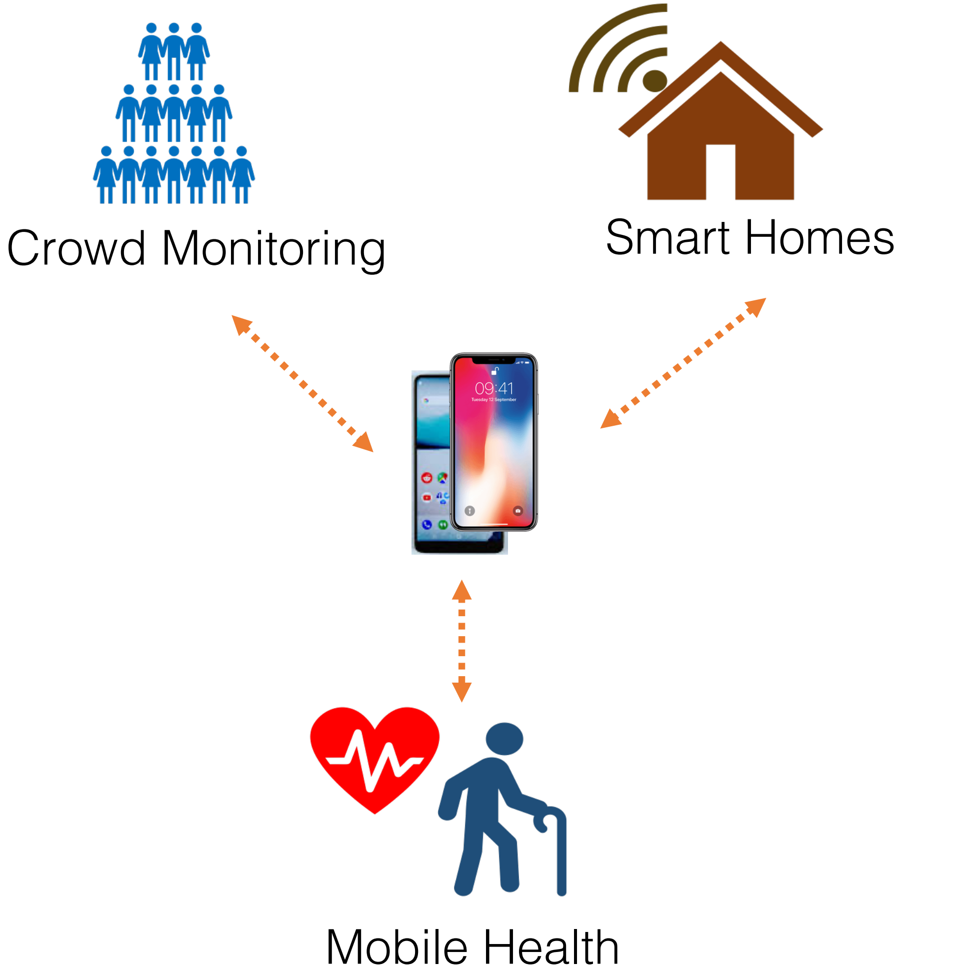
[1] Sandeep Singh Sandha*, Joseph Noor*, Fatima Anwar and Mani Srivastava, "Time Awareness in Deep Learning-Based Multimodal Fusion Across Smartphone Platforms," 2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI-20).

[2] Swapnil Sayan Saha*, Sandeep Singh Sandha*, Mani Srivastava, "Deep Convolutional Bidirectional LSTM for Complex Activity Recognition with Missing Data," Human Activity Recognition Challenge - Smart Innovations, Systems and Technologies, Ch. 4, Springer Singapore (2020).

[3] Sandeep Singh Sandha*, Joseph Noor*, Fatima Anwar, Mani Srivastava, "Exploiting Smartphone Peripherals for Precise Time Synchronization," 2019 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS-19)

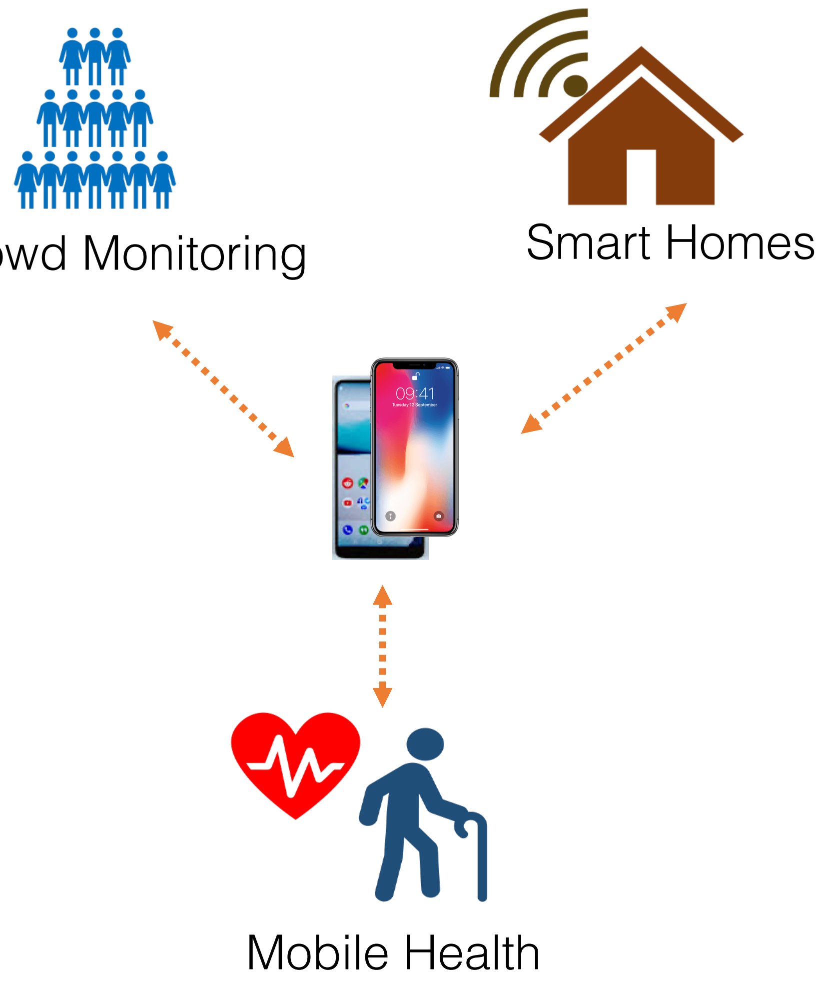
Use Case: Multimodal Fusion across Smartphones

Emerging applications



Use Case: Multimodal Fusion across Smartphones

Emerging applications use multimodal fusion.



Fusion improve application performance [Ngiam11, Eitel15, Radu18, Ortega19]

Multimodal fusion: Assumes modalities are synchronized

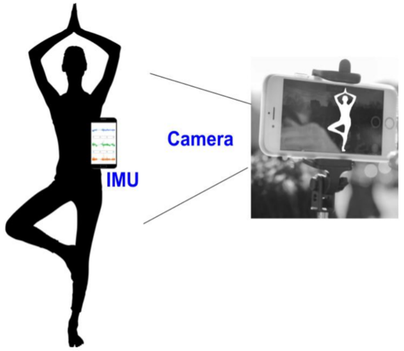
[Ngiam11] Ngiam, Jiquan, et al. "Multimodal deep learning." In International Conference on Machine Learning, 2011.
[Eitel15] Eitel, Andreas, et al. "Multimodal deep learning for robust RGB-D object recognition." 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015.
[Radu18] Radu, Valentin, et al. "Multimodal deep learning for activity and context recognition." Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2018.
[Ortega19] Ortega, Juan DS, et al. "Multimodal fusion with deep neural networks for audio-video emotion recognition." arXiv preprint arXiv:1907.03196 (2019).

Use Case of Synchronized Time at the Edge

Deep learning-based multimodal fusion

Fusion improve classifier accuracy

Labelling of modalities



Camera labelling IMU modalities [Cook20, Plotz12]

Coordinate actions

Ampme: Play music louder [AMPME20]

[Plotz12] Plotz, Thomas, et al. "Automatic synchronization of wearable sensors and video-cameras for ground truth annotation--a practical approach." 2012 16th international symposium on wearable computers. IEEE, 2012.
[Cook20] https://abc-research.github.io/cook2020/
[AMPME20] https://www.ampme.com/

Time at the Edge

The most straightforward approach to align the modalities is to use the timestamps.

Are the timestamps across smartphone reliable ?

Time at the Edge

Researchers previously noted that time across smartphone is unreliable, but they have not quantified and characterized these errors, nor studied the reasons for these errors.

Research	Remarks
Lazik et al. [Lazik15], Yan et al. [Yan17]	❖ Observe poor clock in smartphones and wearables. ❖ [Lazik15] uses a network of beacons to synchronize smartphones. ❖ [Yan17] uses powerline radiation to synchronize wearables.
MNTP [Mani16]	❖ Studies Network Time Protocol (NTP) errors in mobile devices (Laptops). ❖ NTP is also used to the discipline smartphone's clock.
Fridman et al. [Fridman16], Plotz et al. [Plotz12]	❖ Observes data streams are out of sync across devices. ❖ Synchronize data using common events.

[Plotz12] Plotz, Thomas, et al. "Automatic synchronization of wearable sensors and video-cameras for ground truth annotation--a practical approach." 2012 16th international symposium on wearable computers. IEEE, 2012.
[Lazik15] Lazik, Patrick, et al. "Ultrasonic time synchronization and ranging on smartphones." In 21st IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 108-118. IEEE, 2015.
[Mani16] Mani, Sathiya Kumaran, et a. "Mntp: Enhancing time synchronization for mobile devices." In Proceedings of the 2016 Internet Measurement Conference, pp. 335-348. 2016.
[Fridman16] Fridman, Lex, et al. "Automated synchronization of driving data using vibration and steering events." Pattern Recognition Letters, 2016.
[Yan17] Yan, Zhenyu, et al. "Application-layer clock synchronization for wearables using skin electric potentials induced by powerline radiation." Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems. 2017.

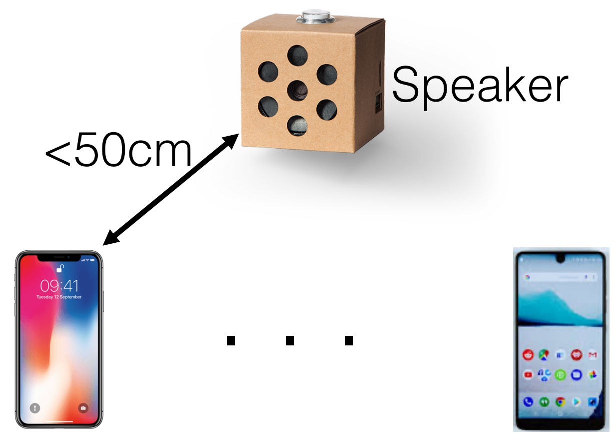
Study: System Clock Accuracy

13 Devices

ID	Device	OS	Year	SIM?
I1	iPhone 6	iOS 12.1.4	2014	N
I2	iPad Pro 9"	iOS 12.1.4	2016	N
I3	iPhone 7+	iOS 12.1.4	2016	N
I4	iPhone 6S	iOS 12.1.4	2015	N
I5	iPhone 6	iOS 12.1.4	2014	Y
A1	Nexus 5X	Android 8.1.0	2015	Y
A2	Nexux 7 Tab	Android 6.0.1	2012	N
A3	Huawei P9	Android 7.0	2016	N
A4	OnePlus A1	Android 5.1.1	2014	N
A5	Samsung GTS2	Android 7.0	2015	N
A6	Nexus 5X	Android 8.1.0	2015	Y
A7	Nexus 7 Tab	Android 6.0.1	2012	N
A8	Pixel 3	Android 9.0	2018	Y

- The study was conducted in March 2019.
- A patch was submitted to Google.
- Changes have been done to new Android versions.

Timestamp a common audio event



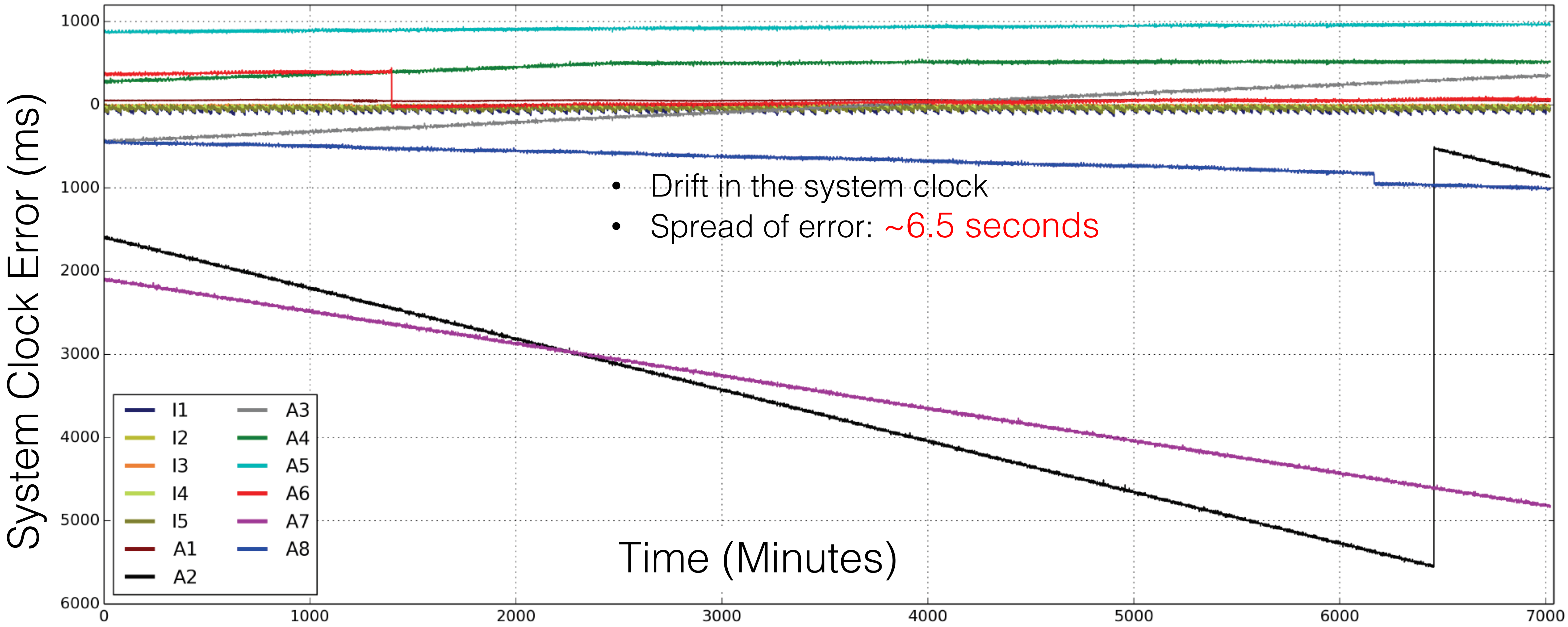
- Generates periodic chirp (~20 Seconds)
- Baseline: Average of NTP clients from all phones

NTP variability: ~10 ms [Mani16].
Audio latency: ~(10 ms - 40 ms).

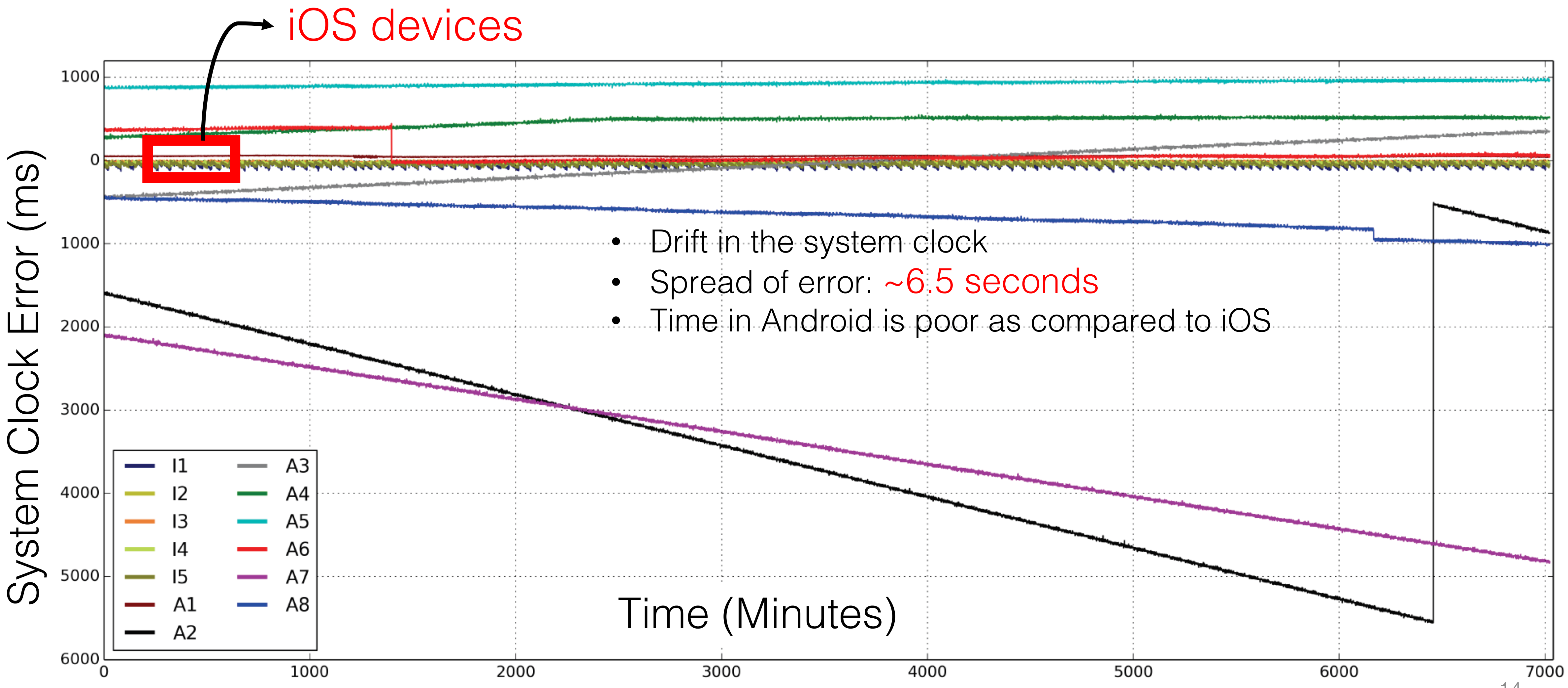
[Mani16] Mani, Sathiya Kumaran, et al. "Mntp: Enhancing time synchronization for mobile devices." Proceedings of the 2016 Internet Measurement Conference. 2016.
Audio Latency: https://superpowered.com/

5 Day Study

System Clock Error = Recorded Timestamp of chirp- Baseline

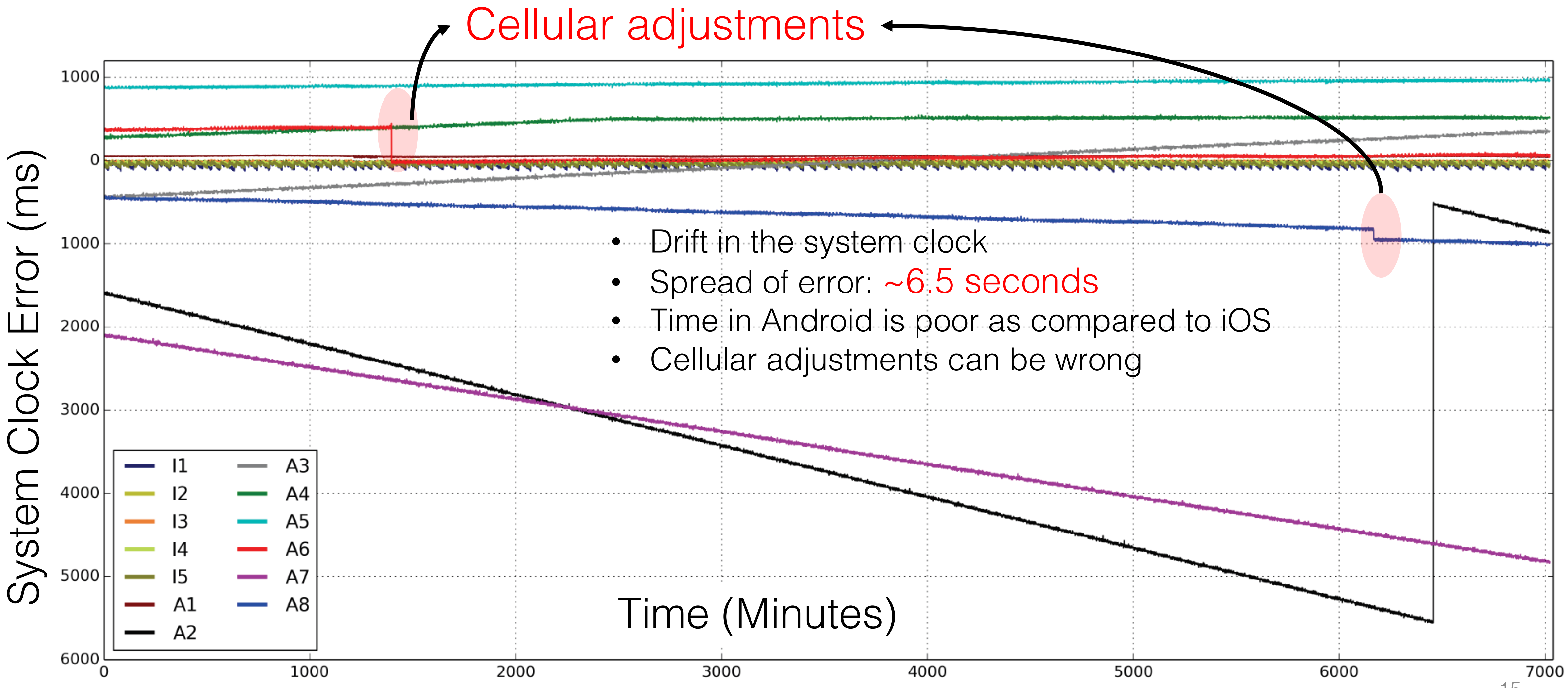


5 Day Study



- Drift in the system clock
- Spread of error: ~6.5 seconds
- Time in Android is poor as compared to iOS

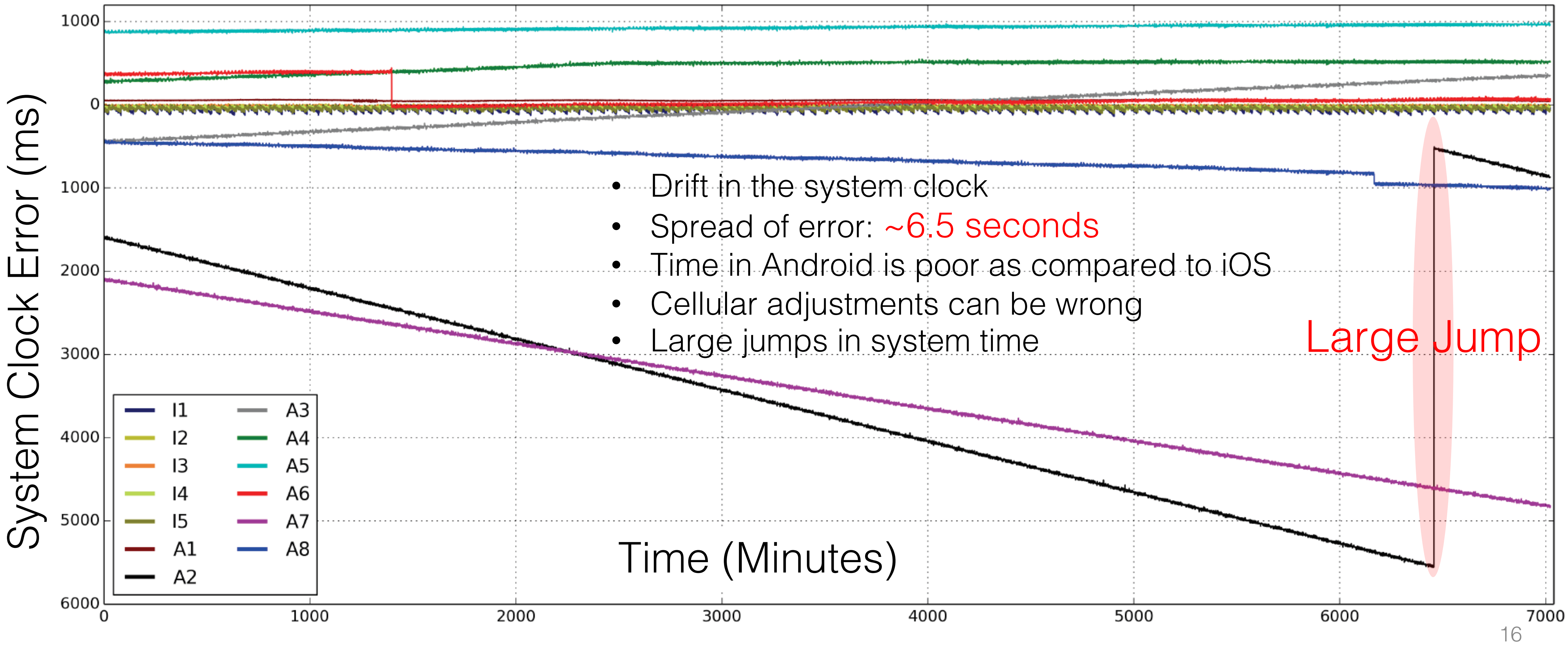
5 Day Study



Cellular adjustments
System Clock Error (ms)
1000
0
1000
2000
3000
4000
5000
6000
0
1000
2000
3000
4000
5000
6000
7000
Time (Minutes)
I1
I2
I3
I4
I5
A1
A2
A3
A4
A5
A6
A7
A8
• Drift in the system clock
• Spread of error: ~6.5 seconds
• Time in Android is poor as compared to iOS
• Cellular adjustments can be wrong

5 Day Study

- Drift in the system clock
- Spread of error: ~6.5 seconds
- Time in Android is poor as compared to iOS
- Cellular adjustments can be wrong
- Large jumps in system time



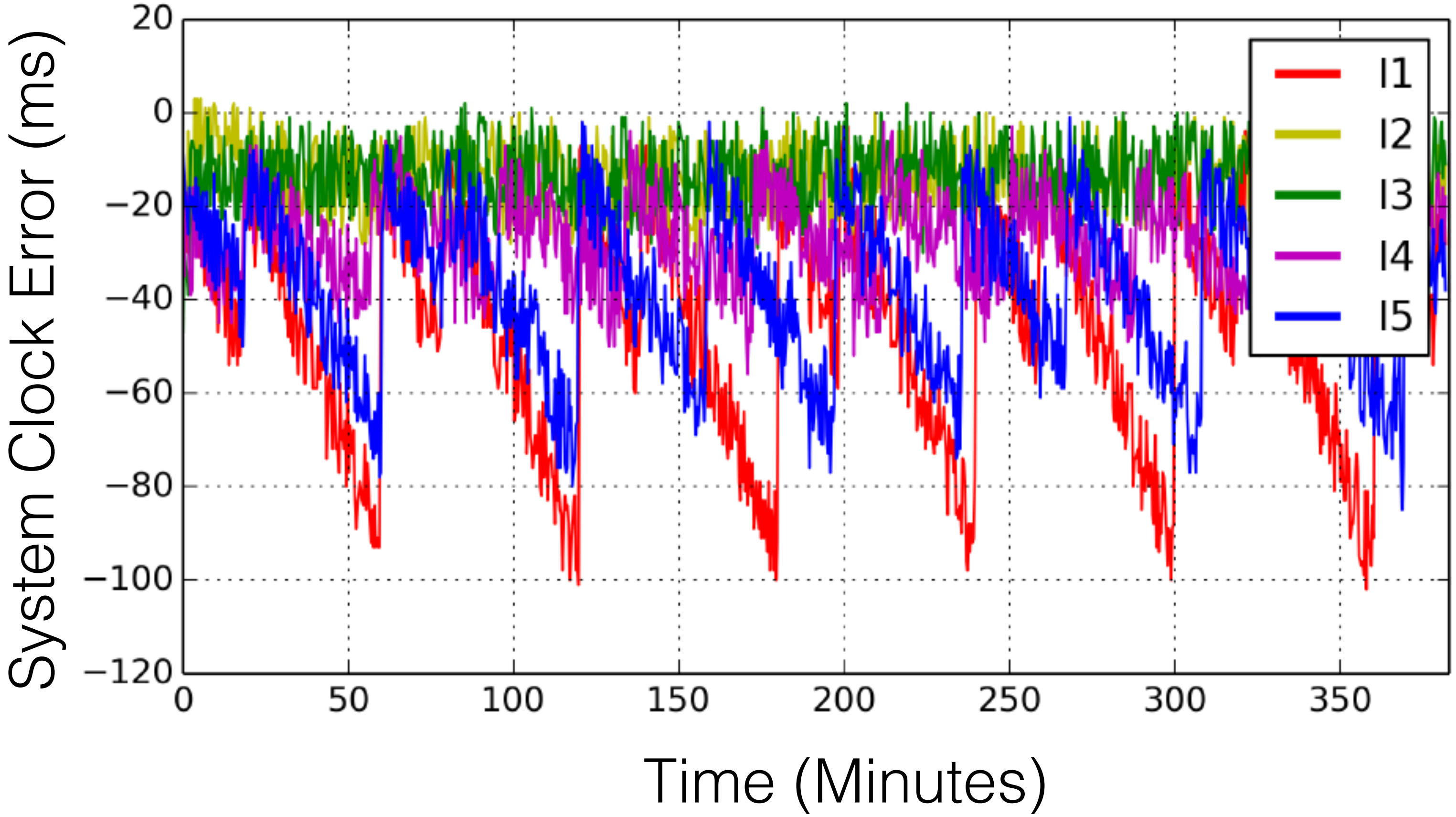
5 Day Study: iOS

iOS

- Error within ~100ms
- Aggressive synchronization

Android

- Error >5000ms



5 Day Study

iOS
- Error within ~100ms
- Aggressive synchronization

Android
- Error ~5000ms

Why are Android timing errors so high?

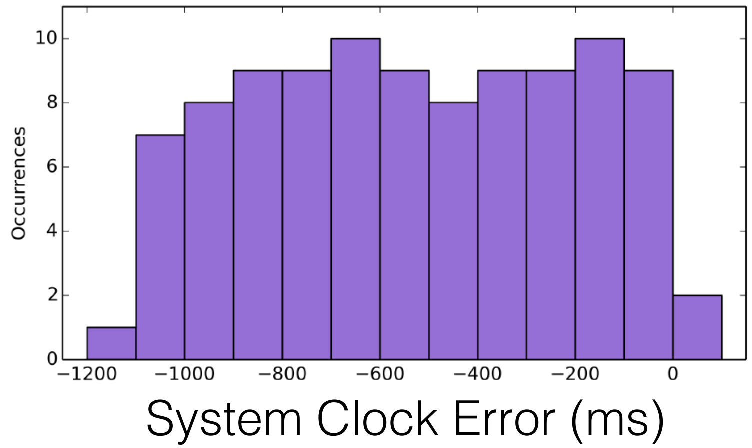
Android Timing Stack

- Android System Time
  - Two update mechanisms: NITZ, and NTP
  - NITZ has priority, and it directly updates the system time.
  - NTP done every 24 hour.
  - NTP updates system time only if the error is more than 5000ms => Large jumps

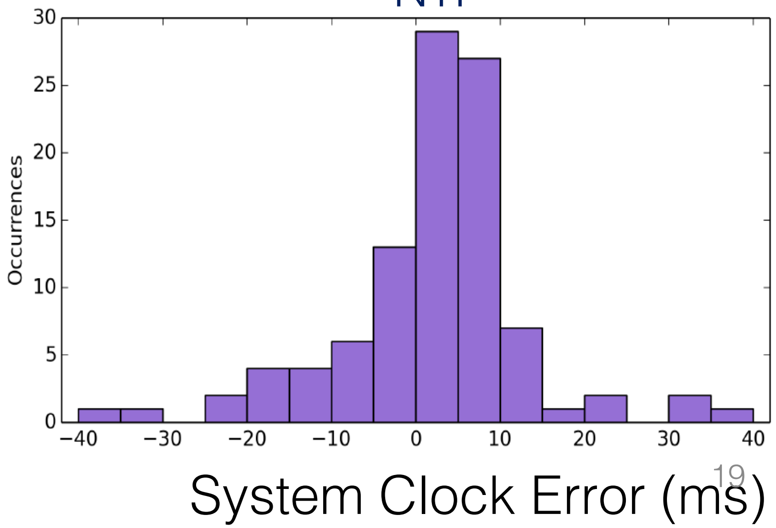
```
if (DBG) Log.d(TAG, "Ntp time is close enough = " + ntp);
```

- Changes in Android 10:
  - 5000ms is modified to 2000ms.

NITZ (Cellular Protocol)

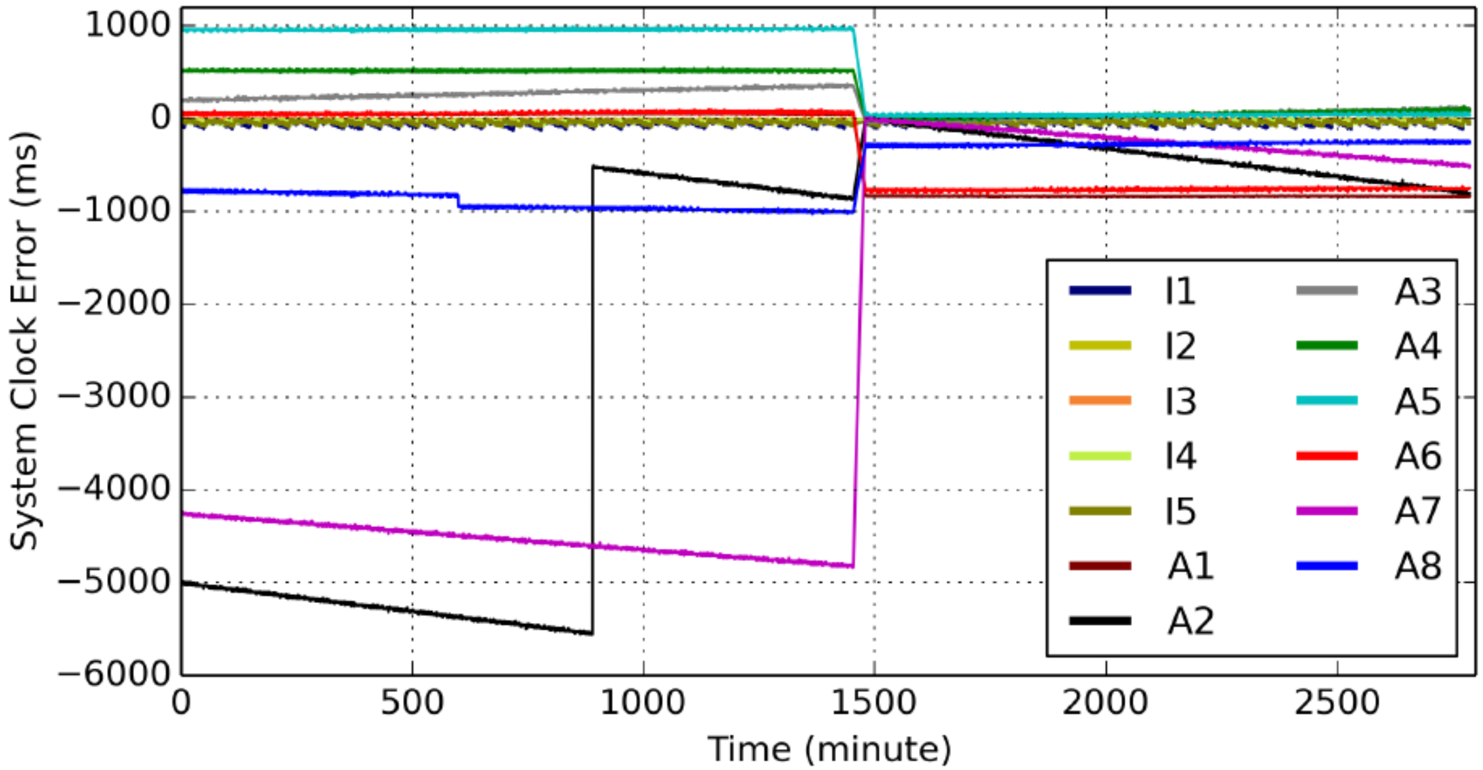


NTP



Simple Trick: Restart all Phones

- Phones with SIM have error ranging from 300 to 800ms.
- Phones using NTP have offset < 40ms



Impact of Timing Errors on Multimodal Fusion

Use case: Human activity recognition

Dataset https://github.com/nesl/CMActivities-DataSet

- Audio from one smartphone and IMU from another

Activity	Number of Videos	Duration (sec)
Go Upstairs	162	1338
Go Downstairs	161	1113
Walk	119	1143
Run	115	891
Jump	73	995
Wash Hand	73	1070
Jumping Jack	90	958

Tianwei Xing*, Sandeep Singh Sandha*, Bharathan Balaji, Supriyo Chakraborty, Mani Srivastava, "Enabling edge devices that learn from each other: Cross modal training for activity recognition,"EdgeSys-18

Impact of Timing Errors on Multimodal Fusion

Use case: Human activity recognition

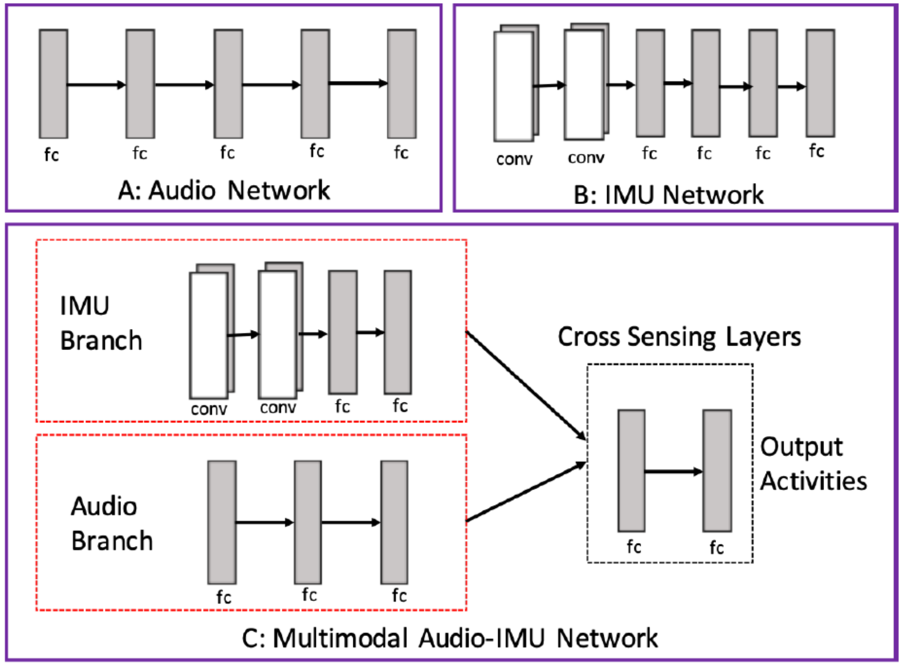
Dataset https://github.com/nesl/CMActivities-DataSet

- Audio from one smartphone and IMU from another

Baseline Accuracy

Networks	Audio	IMU	Multimodal Audio-IMU
Test Accuracy	91.34%	90.10%	96.12%

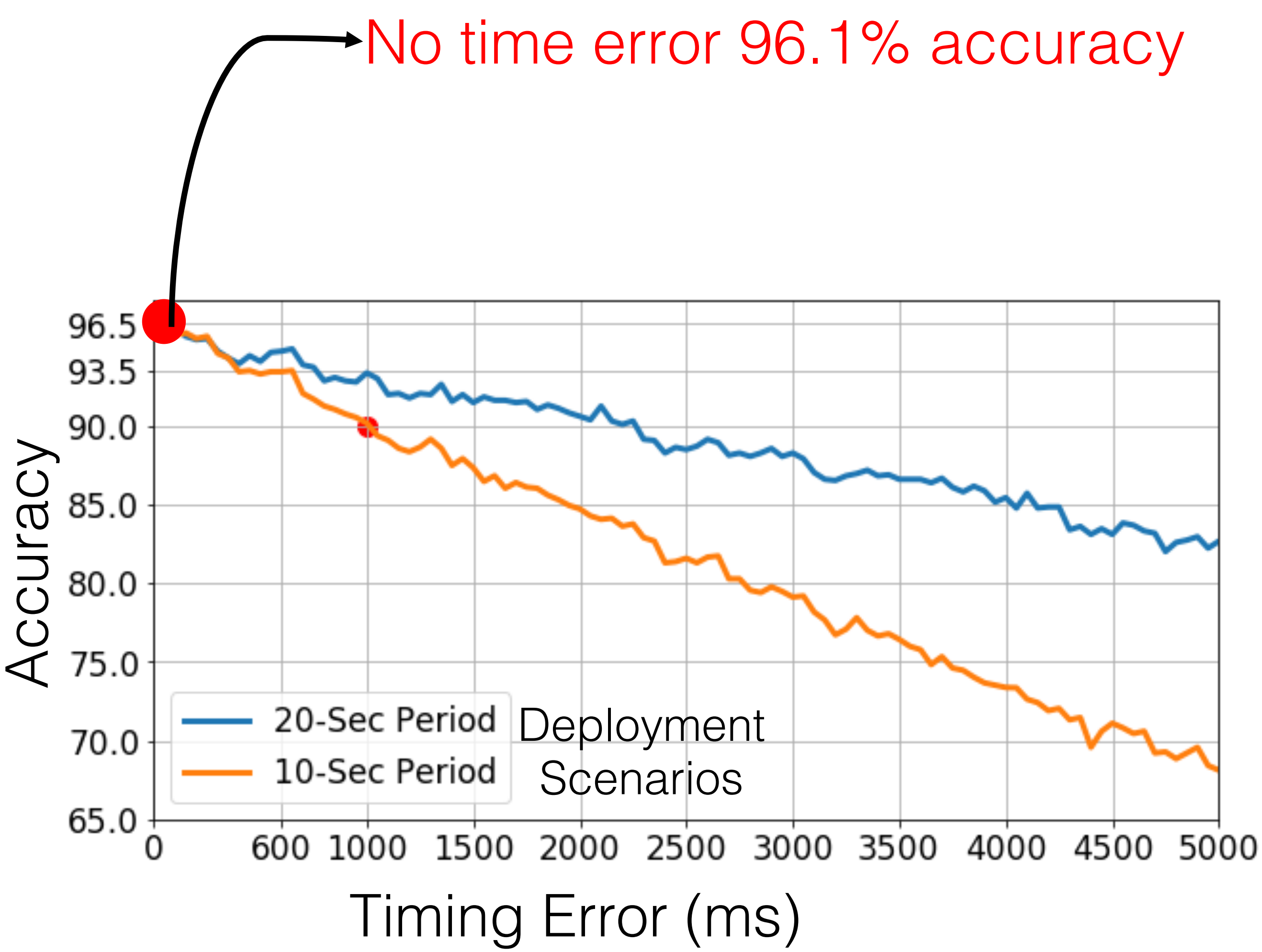
Feature level fusion [Ngiam11, Radu18]

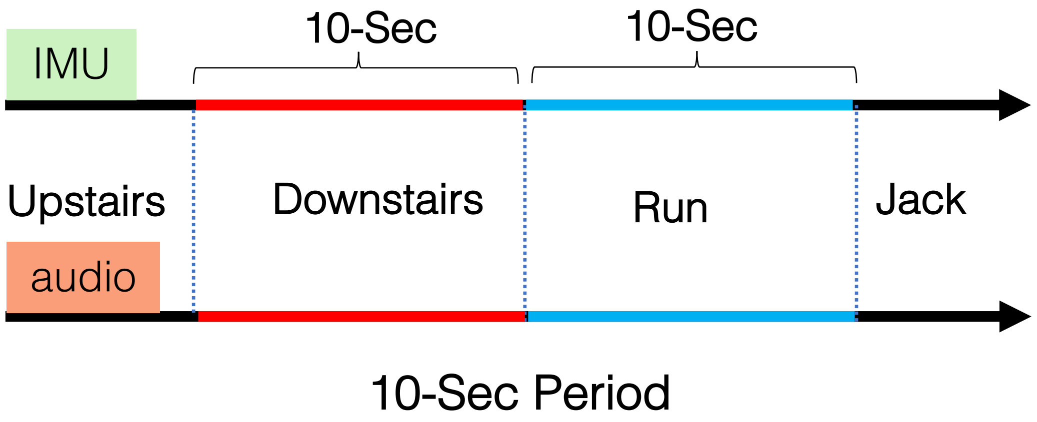


Multimodal fusion improves accuracy by ~5%.

[Ngiam11] Ngiam, Jiquan, et al. "Multimodal deep learning." In International Conference on Machine Learning, 2011.
[Radu18] Radu, Valentin, et al. "Multimodal deep learning for activity and context recognition." Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2018.

Impact of Timing Errors on Fusion Classifier

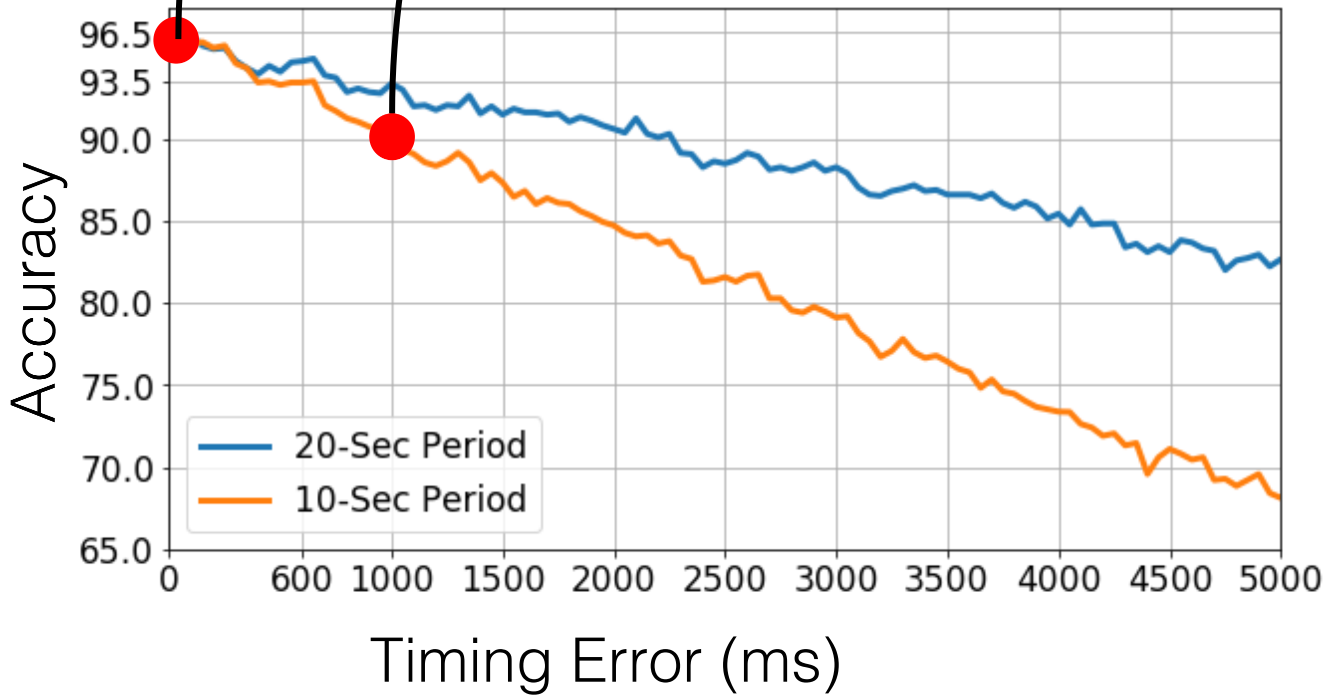


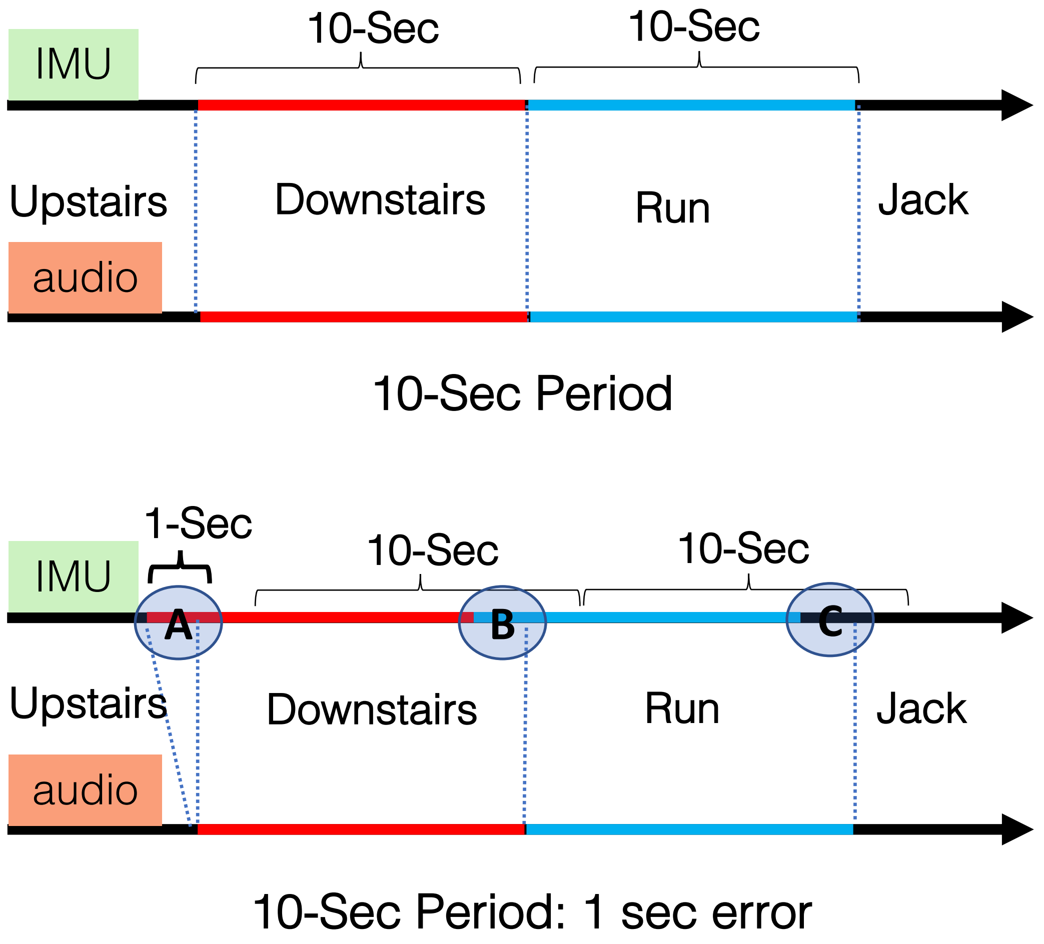


Impact of Timing Errors on Fusion Classifier

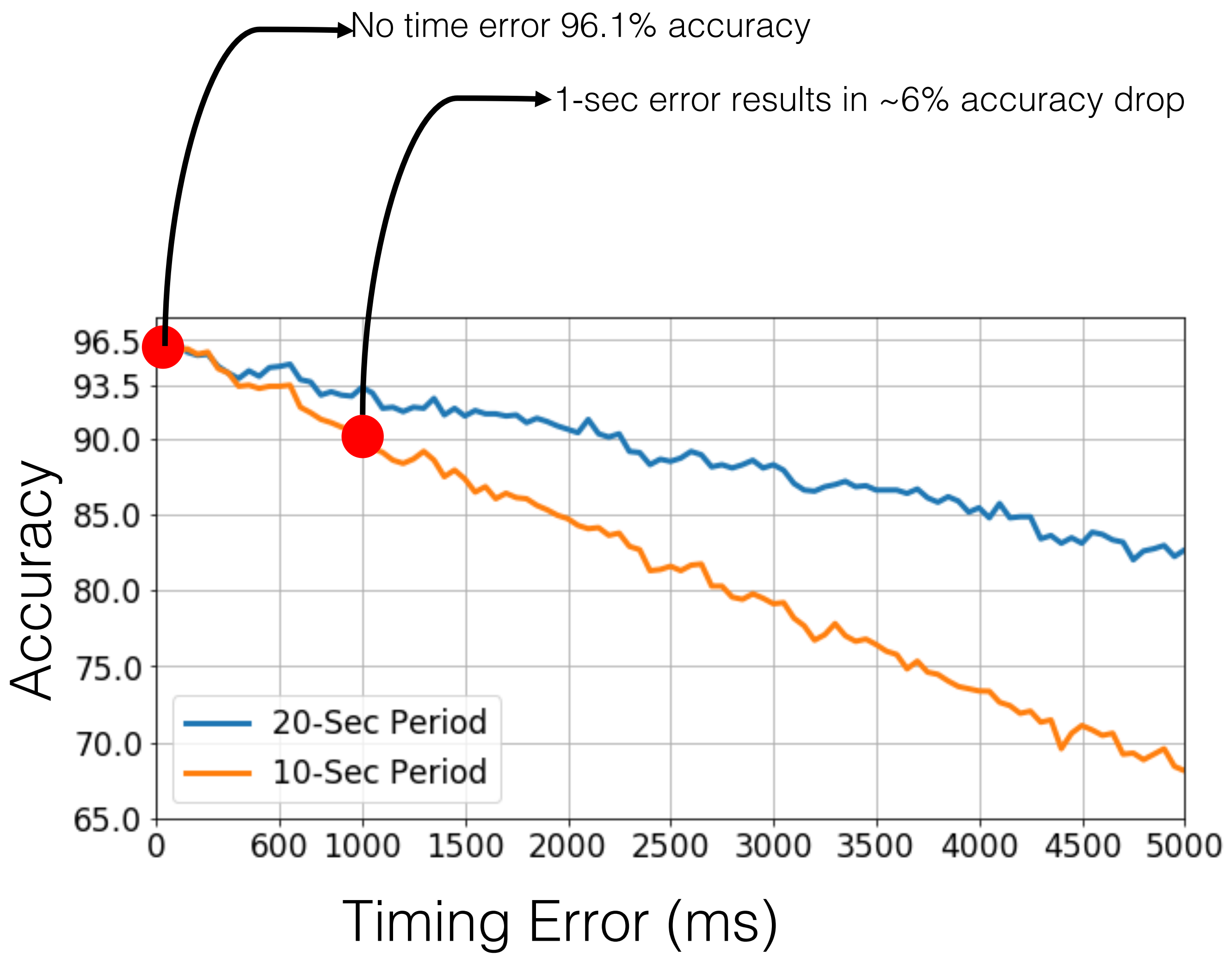
No time error 96.1% accuracy

1-sec error results in ~6% accuracy drop





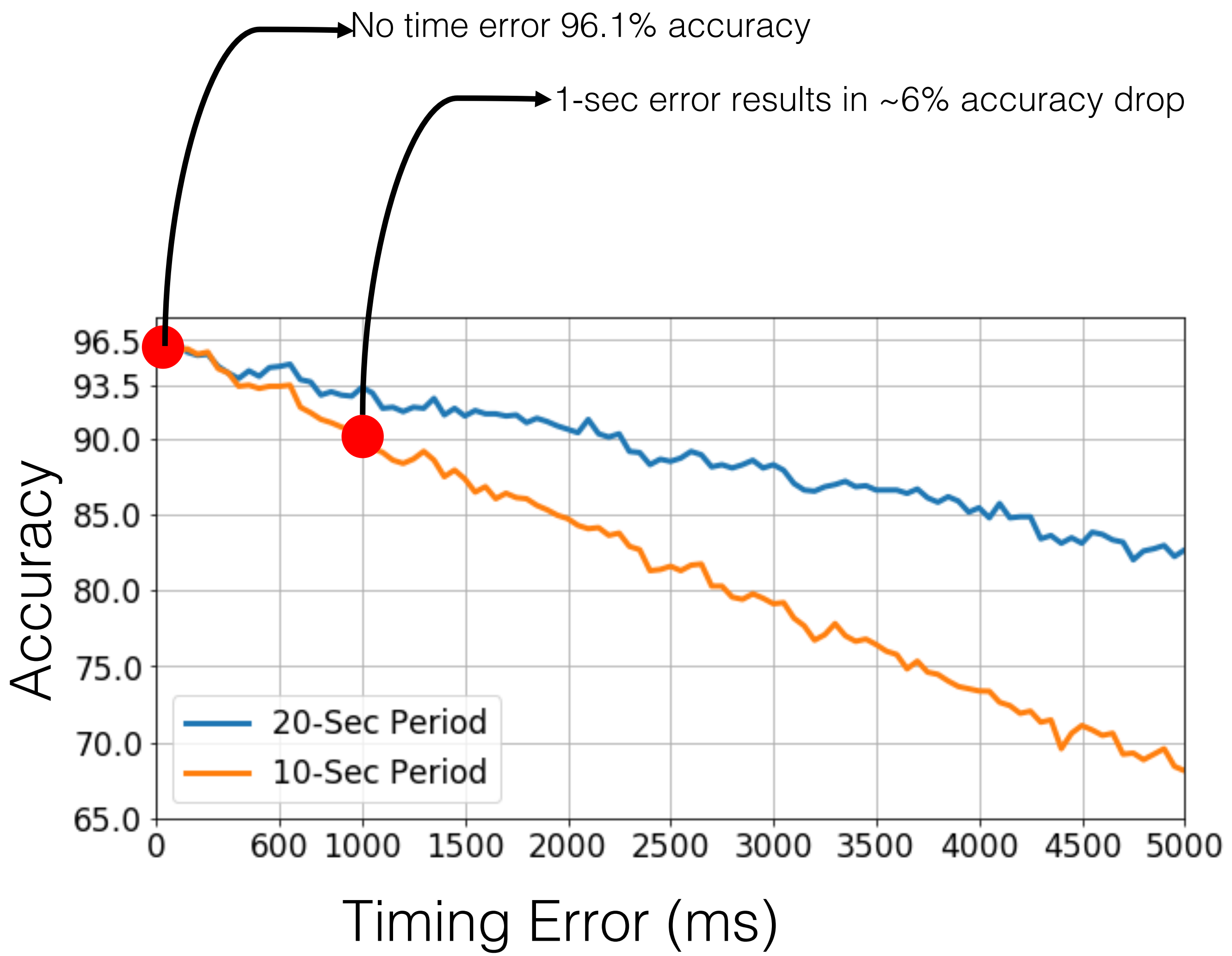
Impact of Timing Errors on Fusion Classifier



Time-Aware Fusion

1. Improve the clock across devices.
2. Modify training pipeline for the imperfect timestamps

Impact of Timing Errors on Fusion Classifier

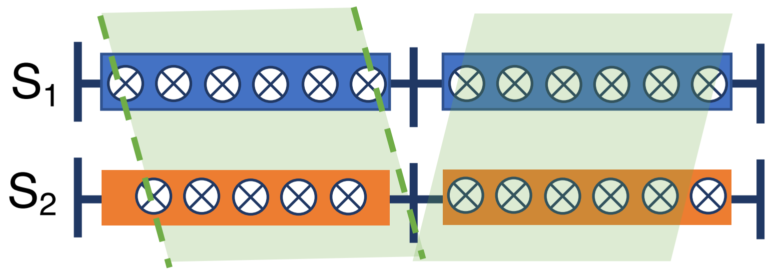


Time-Aware Fusion

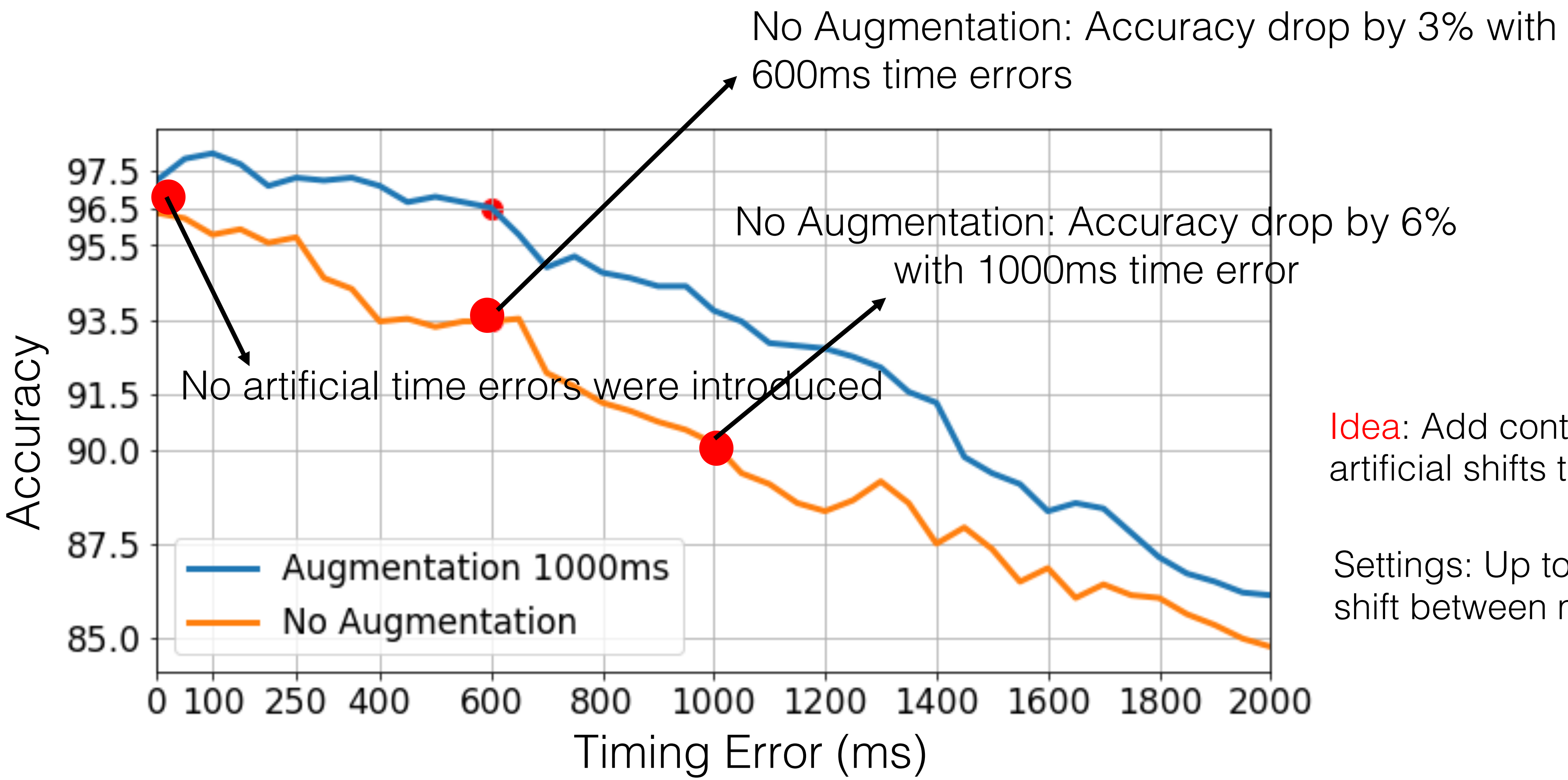
1. Improve the clock across devices.
2. Modify training pipeline for the imperfect timestamps

Time-Shift Data Augmentation

Idea: Add controlled artificial shifts during training.



Time-Shift Data Augmentation



Idea: Add controlled artificial shifts to the data

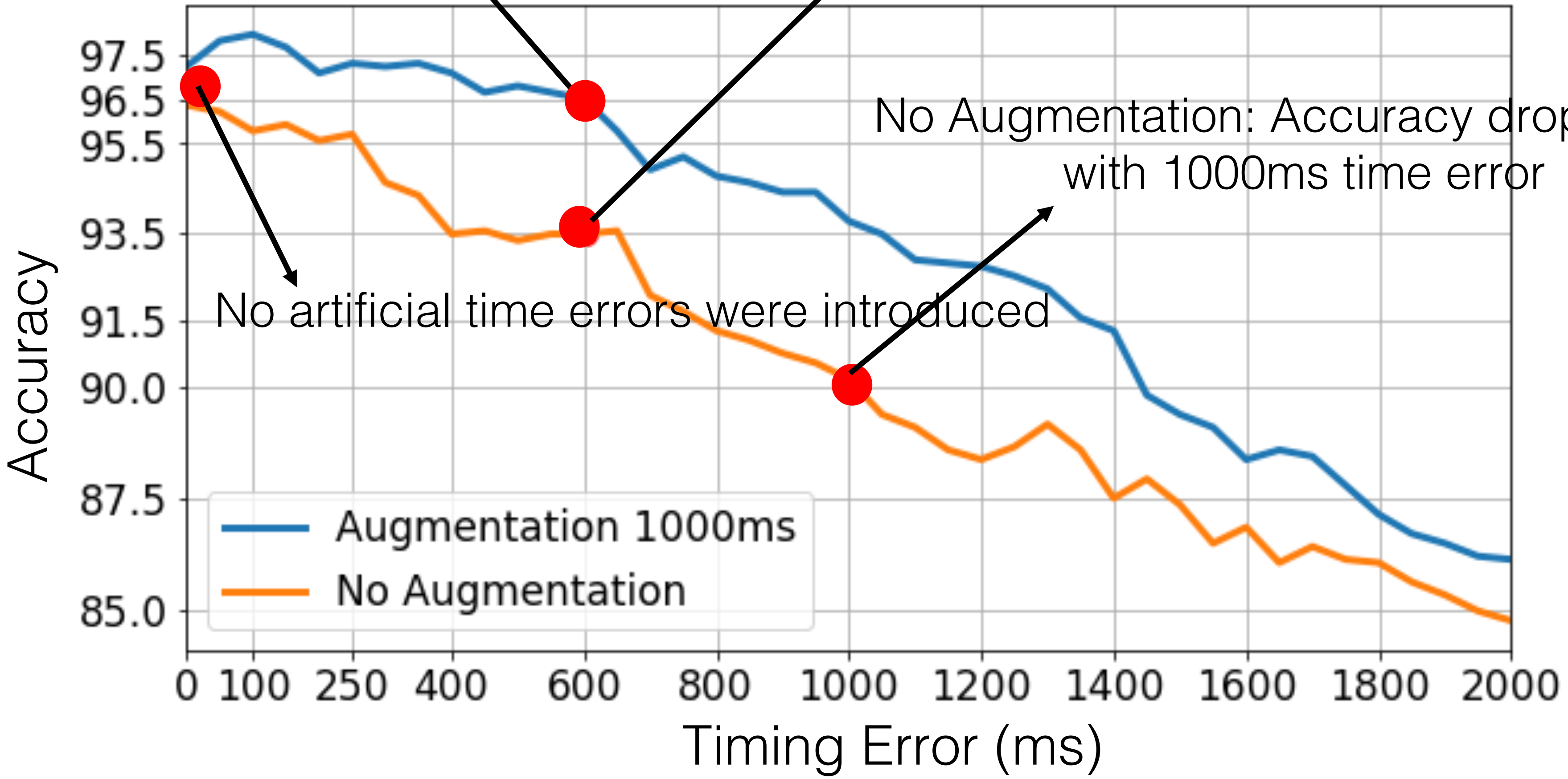
Settings: Up to 1000ms shift between modalities

Time-Shift Data Augmentation

1000ms Augmentation can handle ~600ms time errors

No Augmentation: Accuracy drop by 3% with 600ms time errors

No Augmentation: Accuracy drop by 6% with 1000ms time error

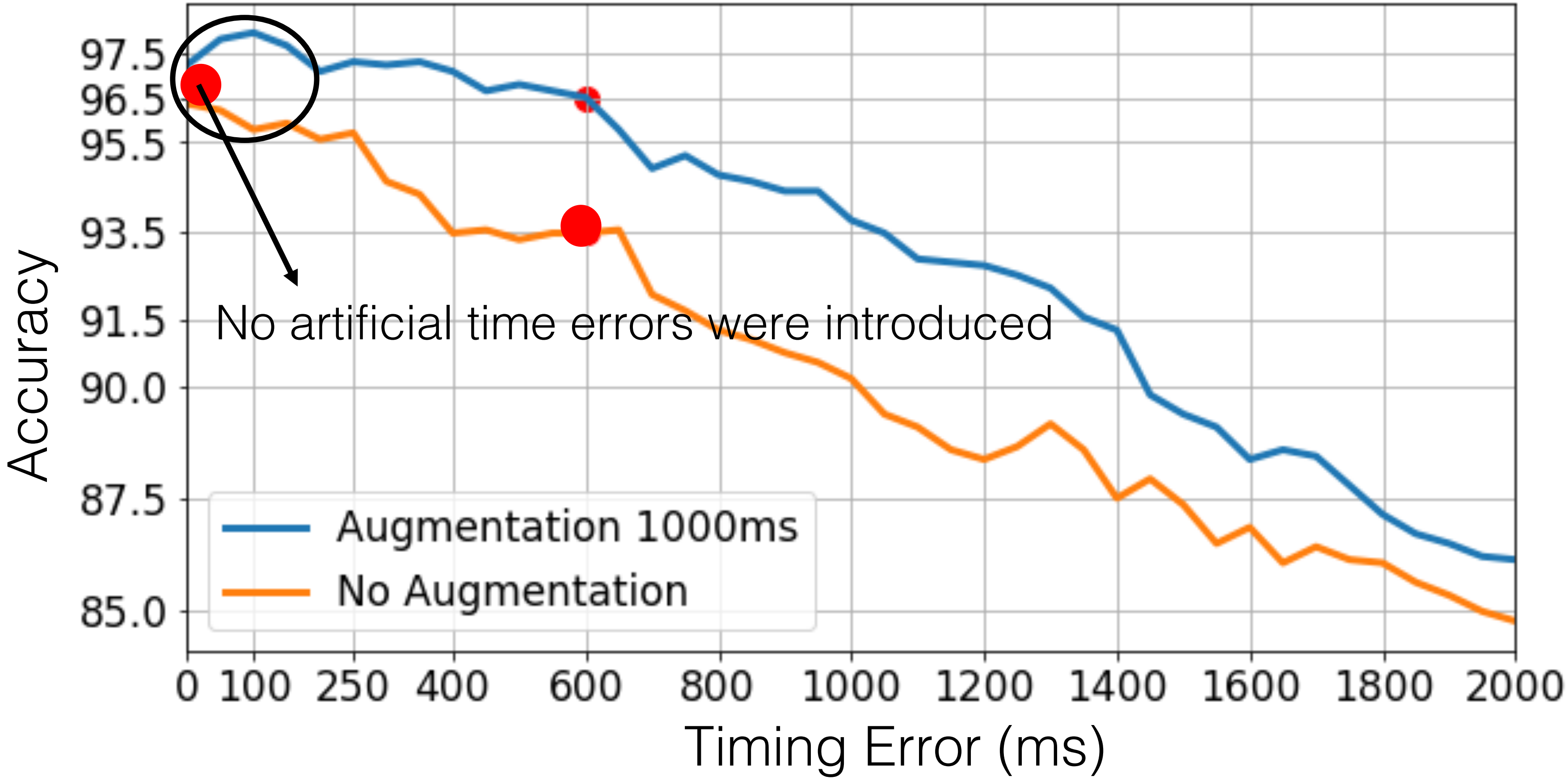


Idea: Add controlled artificial shifts to the data

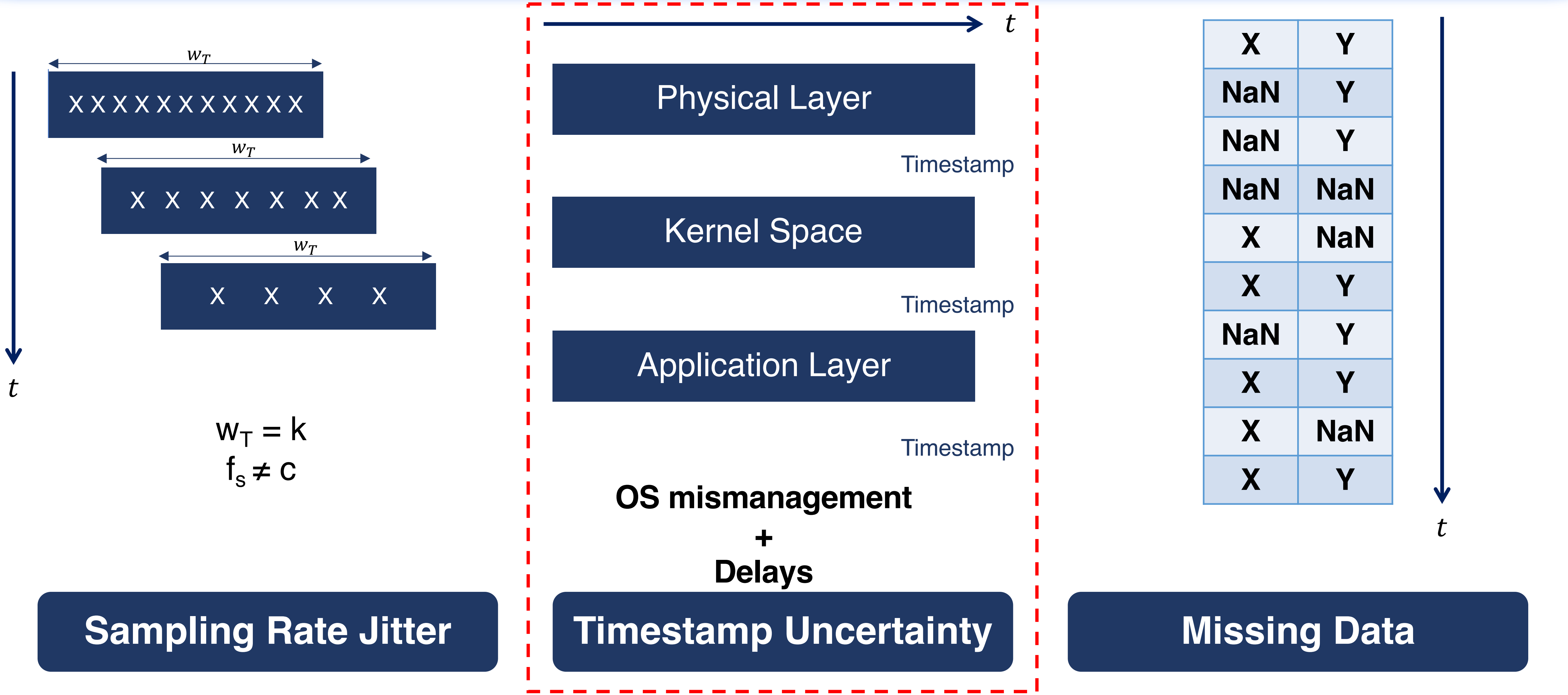
Settings: Up to 1000ms shift between modalities

Time-Shift Data Augmentation

Timing errors are inevitable in the data



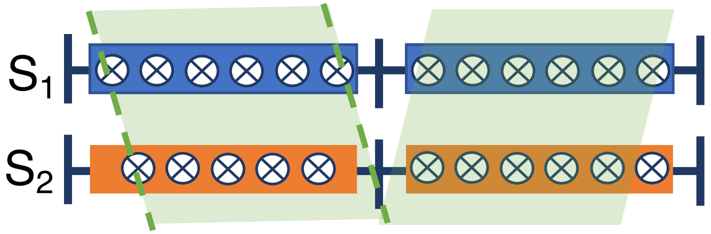
Non-Determinism in Temporal Properties in Open-Source Datasets



Non-Determinism in Temporal Properties in Open-Source Datasets

Cooking Activity Recognition Challenge

- 4 accelerometers (2 wrist watches, 2 smartphones)
- **3 distinct macro** and **10 distinct micro-activities**
- Data: 288 files each with 30 seconds of data.



Artificially shifts

- **Sandwich** - cut, wash, take, put, other
- **Fruit salad** - cut, take, peel, add, mix, put, other
- **Cereal** - cut, take, pour, peel, put, open, other

Accuracy	Without Time aug.	With Time aug.
Macro activity	77%	83%
Micro activity	48%	72%

Submission: 3rd out of 78 registered teams.

Swapnil Sayan Saha*, **Sandeep Singh Sandha***, Mani Srivastava, "Deep Convolutional Bidirectional LSTM for Complex Activity Recognition with Missing Data," Human Activity Recognition Challenge - Smart Innovations, Systems and Technologies, Ch. 4, Springer Singapore (2020).

Code: https://github.com/nesl/Robust-Deep-Learning-Pipeline

How to Improve Time at the Edge ?

Review: Time Synchronization Approaches

Sync. approach	Applicability to smartphones
Reference broadcasts [Elson02], TPSN [Ganeriwal03], FTSP [Maróti04], PulseSync [Lenzen14]	❖ Proposed for wireless sensor networks. ❖ Theoretically possible across smartphone. ❖ Require shared wireless connection, specialized timestamping and timing stack control.
Skin electric potential [Yan17], Powerline radiation [Rowe09], Radio data system [Li11]	❖ Exploit ambient signals like powerline radiation, FM broadcast. ❖ Require specialized external hardware. ❖ Limited generalizability.
PTP [Eidson02]	❖ Hardware timestamping and based on wired networks.
NTP [Mills91]	❖ Widely used to maintain system clock.

[Mills91] Mills, David L. "Internet time synchronization: the network time protocol." IEEE Transactions on communications 39.10 (1991): 1482-1493.
[Elson02] Elson, Jeremy, et al. "Fine-grained network time synchronization using reference broadcasts." ACM SIGOPS Operating Systems Review 36.SI (2002): 147-163.
[Eidson02] Eidson, John C., et al. "IEEE-1588™ Standard for a precision clock synchronization protocol for networked measurement and control systems." Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting. 2002.
[Ganeriwal03] Ganeriwal, Saurabh, et al. "Timing-sync protocol for sensor networks." Proceedings of the 1st international conference on Embedded networked sensor systems. 2003.
[Maróti04] Maróti, Miklós, et al. "The flooding time synchronization protocol." Proceedings of the 2nd international conference on Embedded networked sensor systems. 2004.
[Li11] Li, Liqun, et al. "Exploiting FM radio data system for adaptive clock calibration in sensor networks." Proceedings of the 9th international conference on Mobile systems, applications, and services. 2011.
[Lenzen14] Lenzen, Christoph, et al. "PulseSync: An efficient and scalable clock synchronization protocol." IEEE/ACM Transactions on Networking 23.3 (2014): 717-727.
[Yan17] Yan, Zhenyu, et al. "Application-layer clock synchronization for wearables using skin electric potentials induced by powerline radiation." Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems. 2017.
[Rowe09] Rowe, Anthony, et al. "Low-power clock synchronization using electromagnetic energy radiating from ac power lines." Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems. 2009.

What is Achievable Time Accuracy on Modern Smartphones?

- Exploit rich peripheral (audio, Wi-Fi & BLE) on Smartphones:
  - We compare synchronization accuracy.
  - We introduce Cross-Peripheral evaluation.

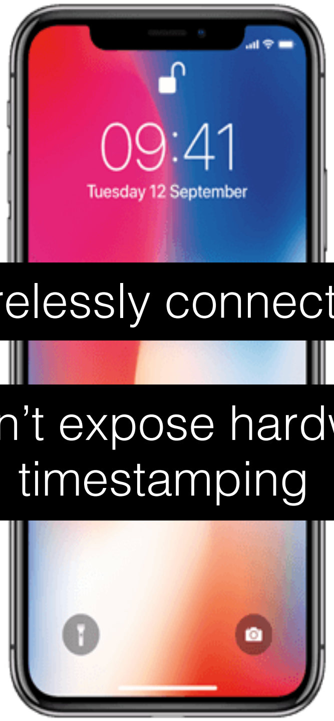
- For Application using different peripherals:
  - Is it possible to synchronize time using one peripherals?
  - Phones that are synchronized using Wi-Fi, can they play music simultaneously?

Time Sync using Smartphone Peripherals

Peripherals	Comments
IMU, Ambient Light, Proximity	❖ Sampling rate limitation (<200Hz) ❖ Hard to generate events (IMU) ❖ Applicable in local setting
Camera	❖ Sampling rate limitation (<240 Hz) ❖ Computation requirement ❖ Applicable in local setting
GPS	❖ Poor indoor availability

Time Sync using Smartphone Peripherals

Peripherals	Comments
IMU, Ambient Light, Proximity	❖ Sampling rate limitation (<200Hz) ❖ Hard to generate events (IMU) ❖ Applicable in local setting
Camera	❖ Sampling rate limitation (<240 Hz) ❖ Computation requirement ❖ Applicable in local setting
GPS	❖ Poor indoor availability
Audio	❖ Easy to generate sound events ❖ Kernel level timestamping ❖ Very high sampling rate (192 kHz) ❖ Applicable in local setting
Wi-Fi	❖ Applicable in Geo-distributed setting ❖ Requires network connectivity ❖ Global NTP servers are available
BLE	❖ Available widely across sensing platforms ❖ Applicable in local setting ❖ Energy efficient

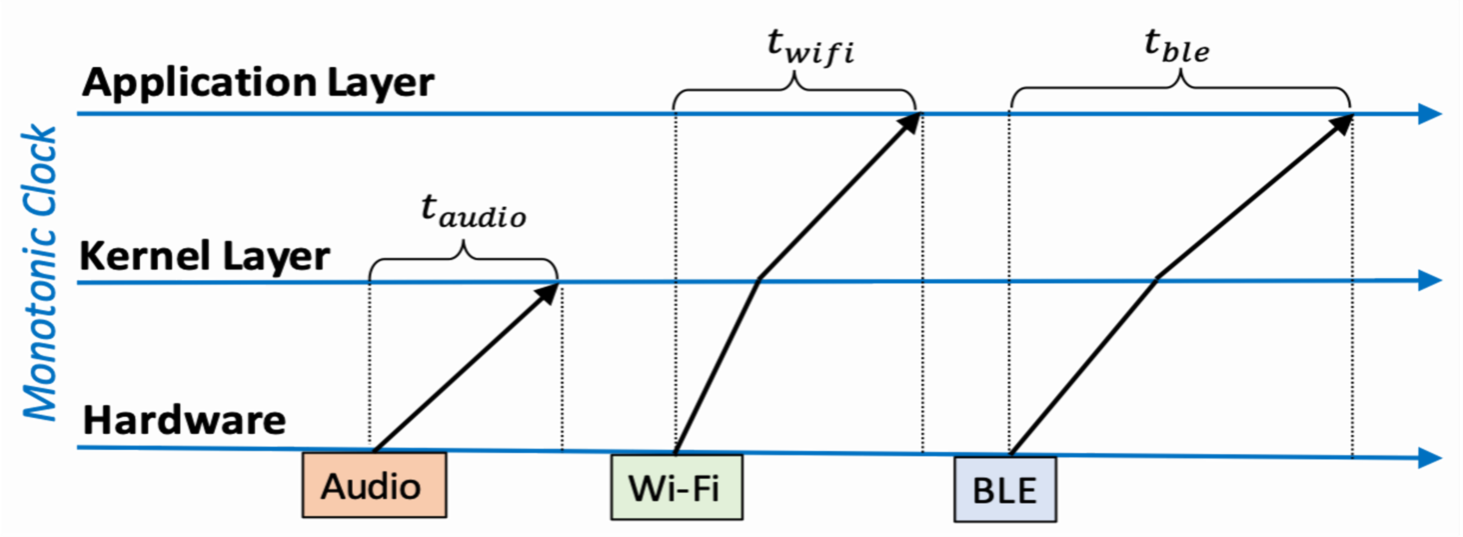


Wirelessly connected

Doesn’t expose hardware timestamping

Time Sync using Audio, Wi-Fi and BLE

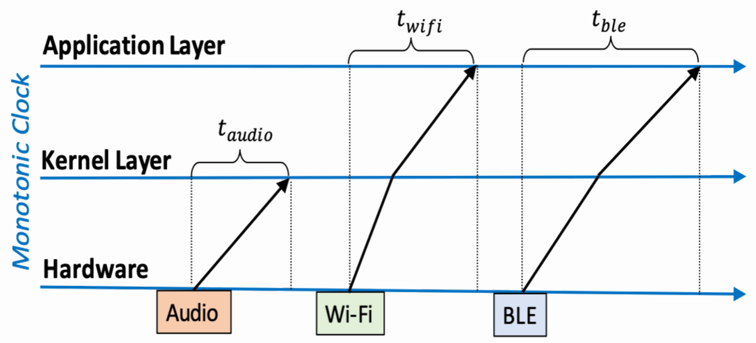
Gather Timestamps --> Calculate offset --->Discipline clock



Synchronizing smartphone, A and B using audio:

$$\underbrace{offset_{audio}}_{\text{Offset calculated by audio}} = offset_{true} + \underbrace{(t^{A}_{audio} - t^{B}_{audio})}_{\text{Difference in audio stack latencies}}$$

Accuracy of a Time Sync



Synchronizing smartphone, A and B using audio:

$$offset_{audio} = offset_{true} + (t^A_{audio} - t^B_{audio})$$

1. **Variability** [Lazik15]: Jitter in multiple offset.

Use case: Synchronize using audio and then listen/timestamp the common audio events.

2. **Cross-peripheral evaluation**: Comparing audio with Wi-Fi.

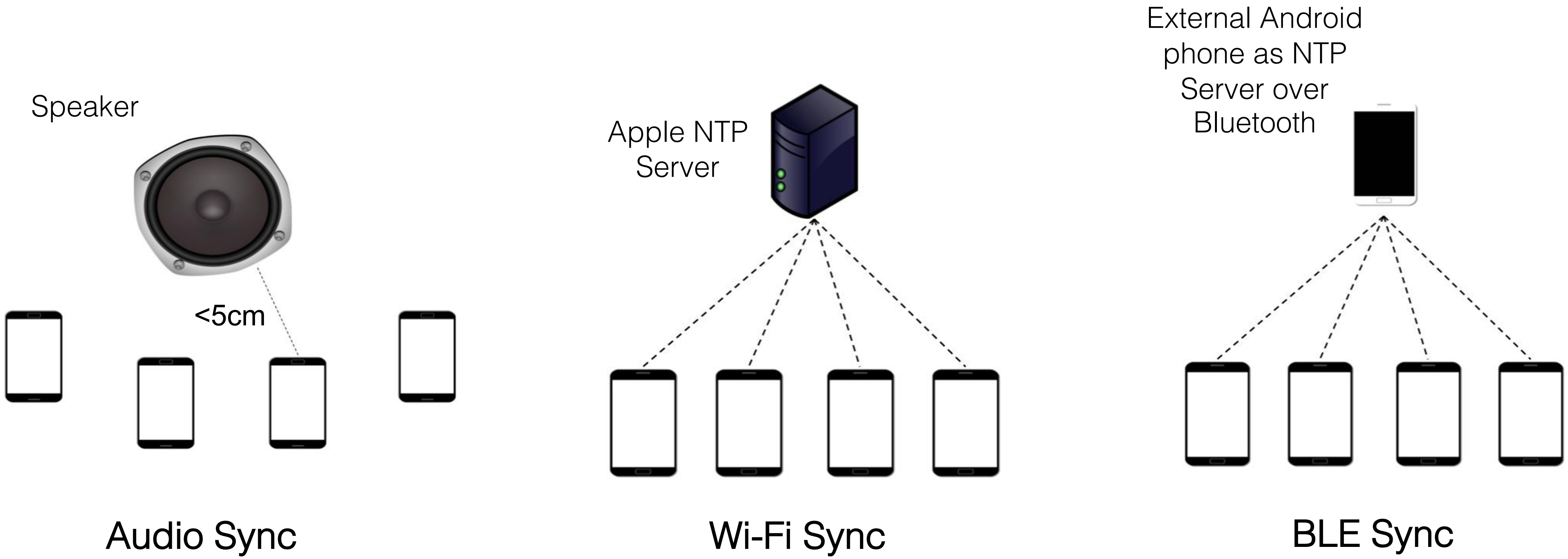
$$offset_{audio} - offset_{wifi} = \underbrace{(t^A_{audio} - t^B_{audio})}_{\text{Audio stack latencies}} + \underbrace{(t^A_{wifi} - t^B_{wifi})}_{\text{Wi-Fi stack latencies}}$$

Jitter in stack latencies.
Bias due to heterogeneous stack latencies.

[Lazik15] Patrick Lazik, et al. "Ultrasonic time synchronization and ranging on smartphones." In 21st IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 108-118. IEEE, 2015.

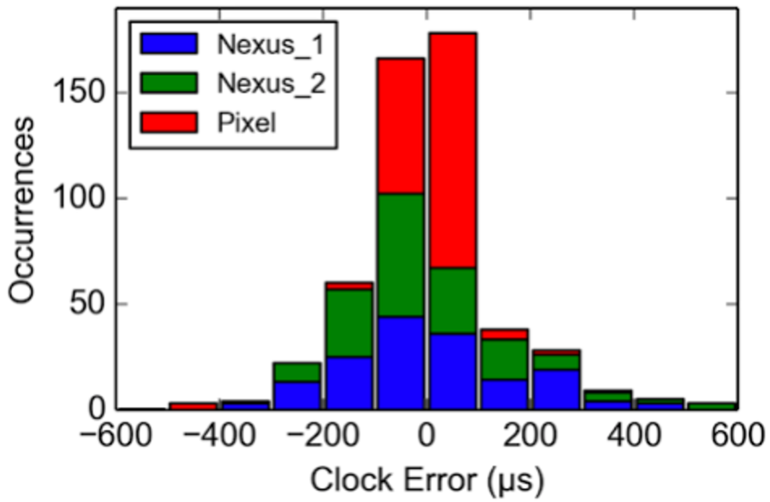
Experimental Setup

Smartphones: Two Pixel 3 and two Nexus 5x.

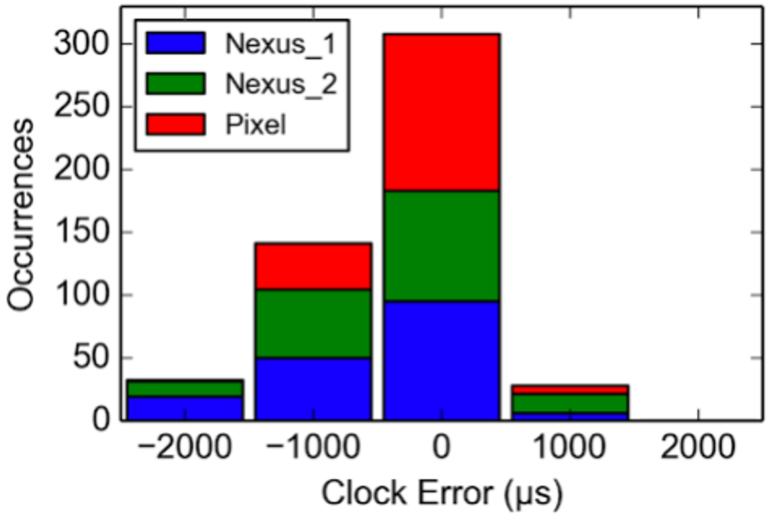


Variability Evaluation

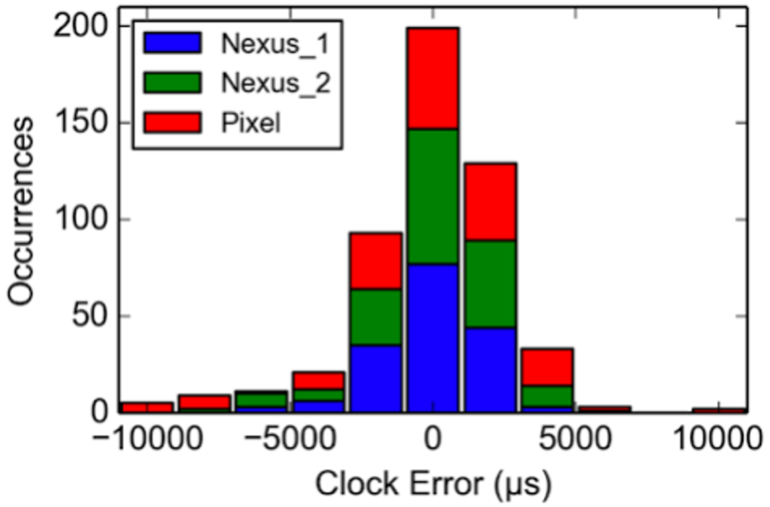
Audio



Wi-Fi



BLE



- Audio: 86% sync attempts fall within 200 microseconds
- Wi-Fi: 95% sync attempts fall within ±1 milliseconds.
- BLE: 85% sync attempts fall within ± 3 milliseconds.

Cross-Peripheral Evaluation

$$offset_{audio} - offset_{wifi} = (t^A_{audio} - t^B_{audio}) + (t^A_{wifi} - t^B_{wifi})$$

Using pixel-3 clock as reference

	Audio vs Wi-Fi	Audio vs BLE	Wi-Fi vs BLE
Pixel-3	1.51±1.50ms	2.20±5.06ms	0.51±5.06ms
Nexus 5 X1	13.12±1.50ms	13.85±4.00ms	2.15±5.46ms
Nexus 5 X2	12.39±1.50ms	12.93±3.74ms	-0.86±4.5ms

- **Similar devices**: One peripheral can be used.
- **Different devices**: Can have heterogeneous stack latencies.

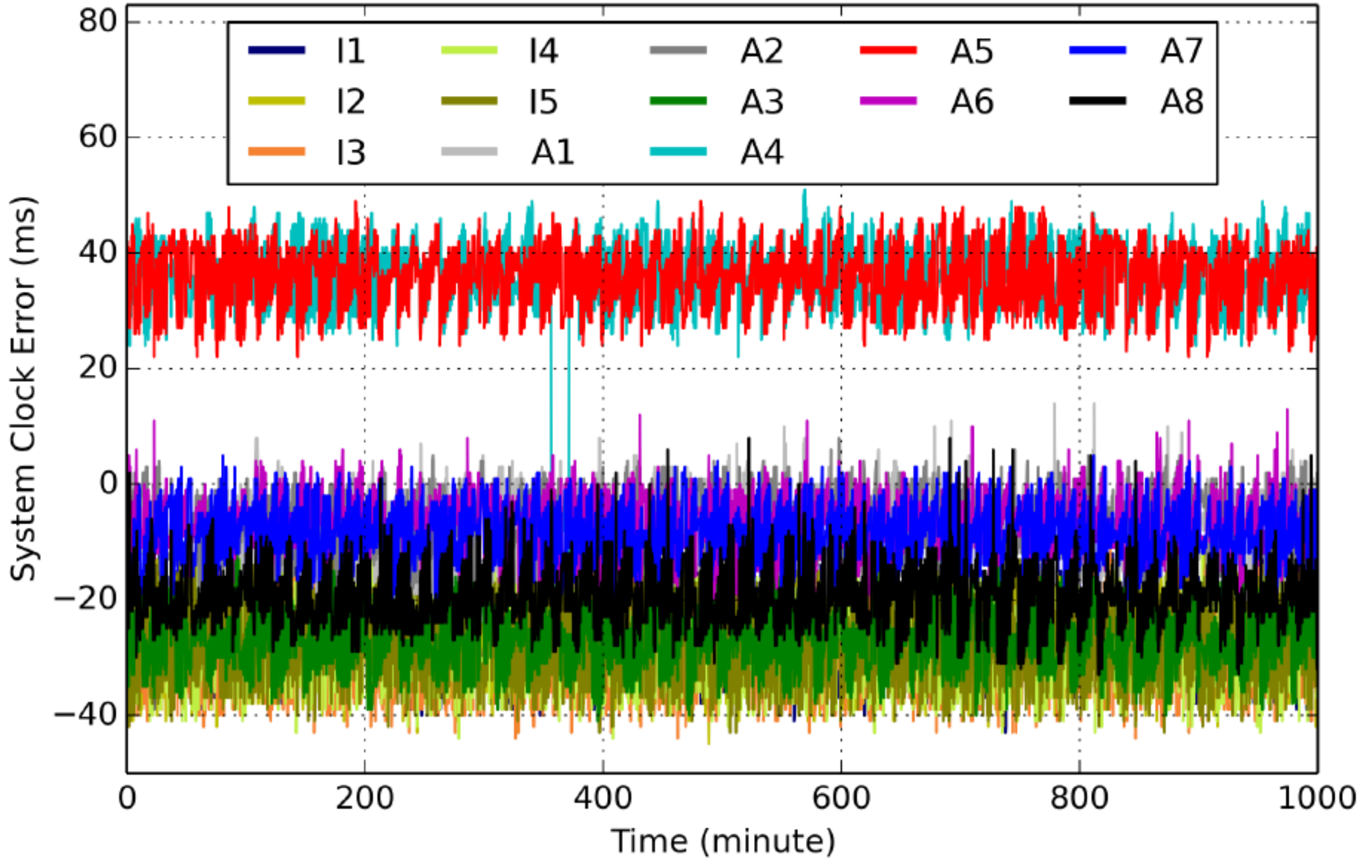
- BLE and Wi-Fi have **similar stack latencies**.
- Audio has heterogenous stack latency.
- BLE has high jitter.

Cross-Peripheral Evaluation: Audio vs Wi-Fi

- Wi-Fi used to do time sync
- Timestamp a common audio event

Devices

ID	Device	OS	Year	SIM?
I1	iPhone 6	iOS 12.1.4	2014	N
I2	iPad Pro 9"	iOS 12.1.4	2016	N
I3	iPhone 7+	iOS 12.1.4	2016	N
I4	iPhone 6S	iOS 12.1.4	2015	N
I5	iPhone 6	iOS 12.1.4	2014	Y
A1	Nexus 5X	Android 8.1.0	2015	Y
A2	Nexus 7 Tab	Android 6.0.1	2012	N
A3	Huawei P9	Android 7.0	2016	N
A4	OnePlus A1	Android 5.1.1	2014	N
A5	Samsung GTS2	Android 7.0	2015	N
A6	Nexus 5X	Android 8.1.0	2015	Y
A7	Nexus 7 Tab	Android 6.0.1	2012	N
A8	Pixel 3	Android 9.0	2018	Y



Conclusion of Smartphone Time Sync

- Peripherals can synchronize time.
  - Best accuracy: ~200 us using Audio across smartphones.
- Can we use one peripheral offset across other peripherals?
  - Yes: Smartphone having same hardware and software stack.

- Code of approaches: https://github.com/nesl/Time-Sync-Across-Smartphones
- Goodclock Library: https://github.com/nesl/GoodClock (Android, iPhone & Python supporting devices)
- GoodClock is used by **LONN Lab** (David Geffen School of Medicine) to collect brain data with accurate timestamps from patients.

  http://lonn.semel.ucla.edu/

Non-Determinism in Temporal Properties

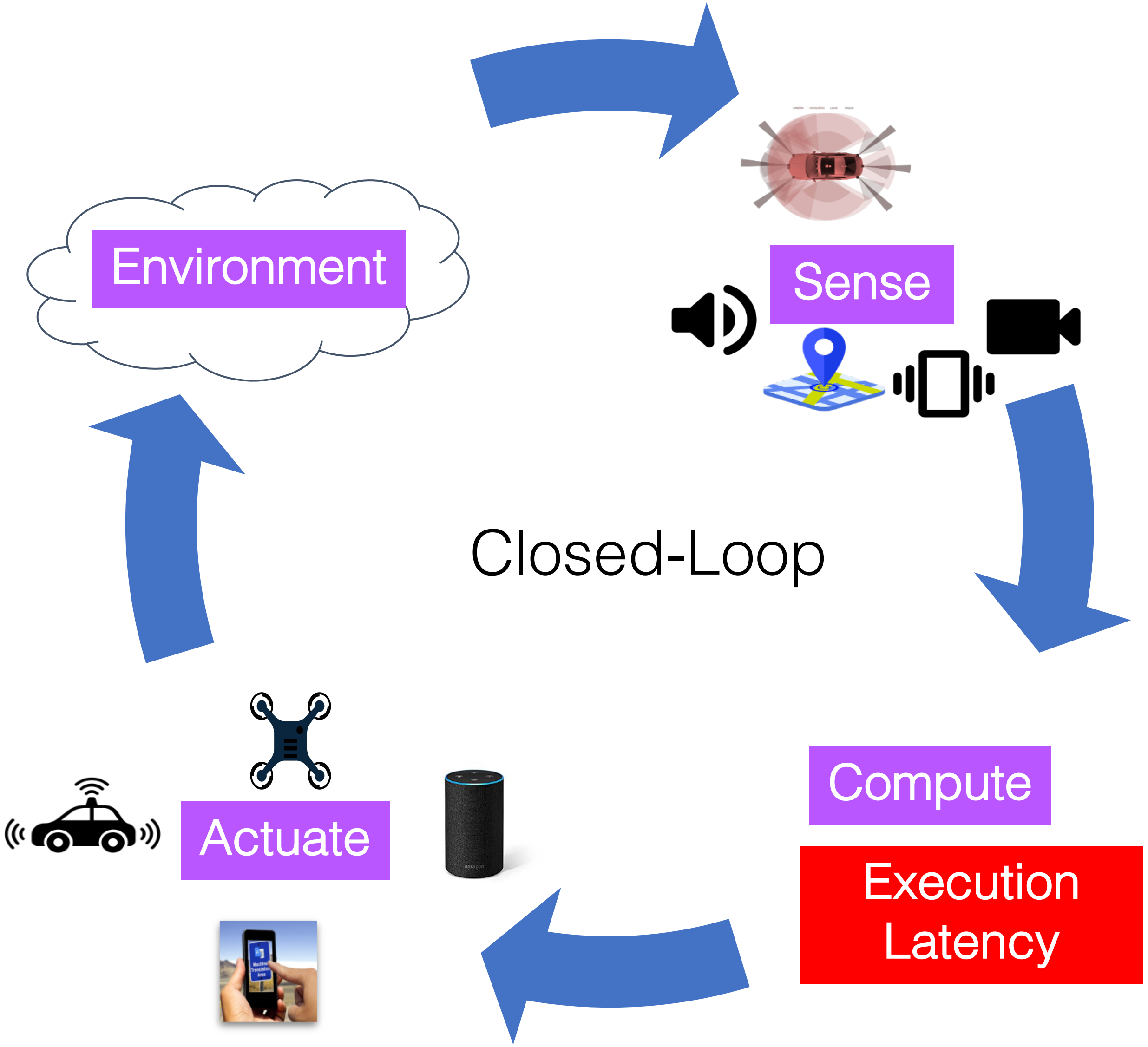
- Time synchronization errors
- OS mismanagement

→ Inaccurate clocks across devices

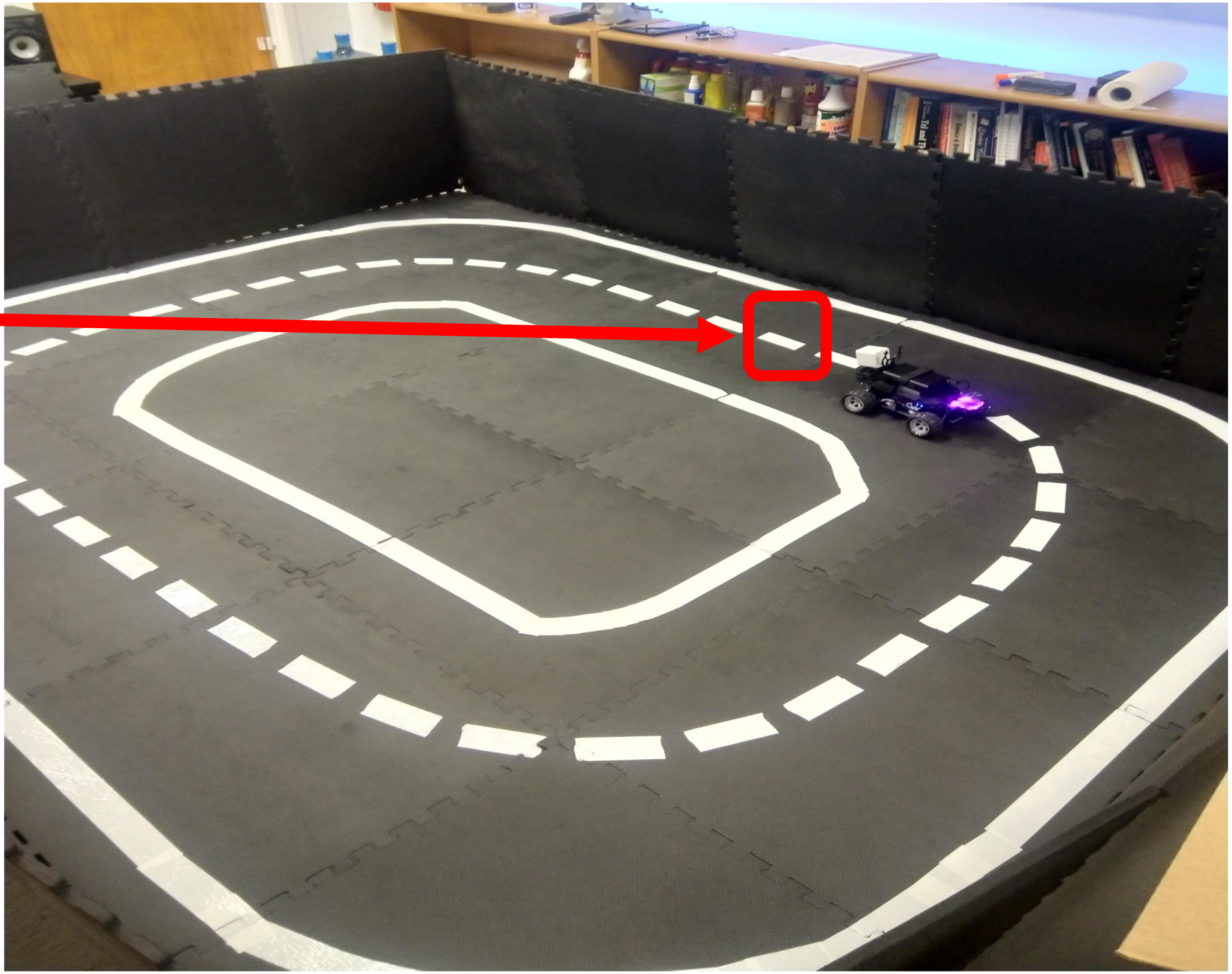
- Hardware heterogeneities
- Network variations
- Compute stochasticity

→ Sampling rates and end-to-end delay variations

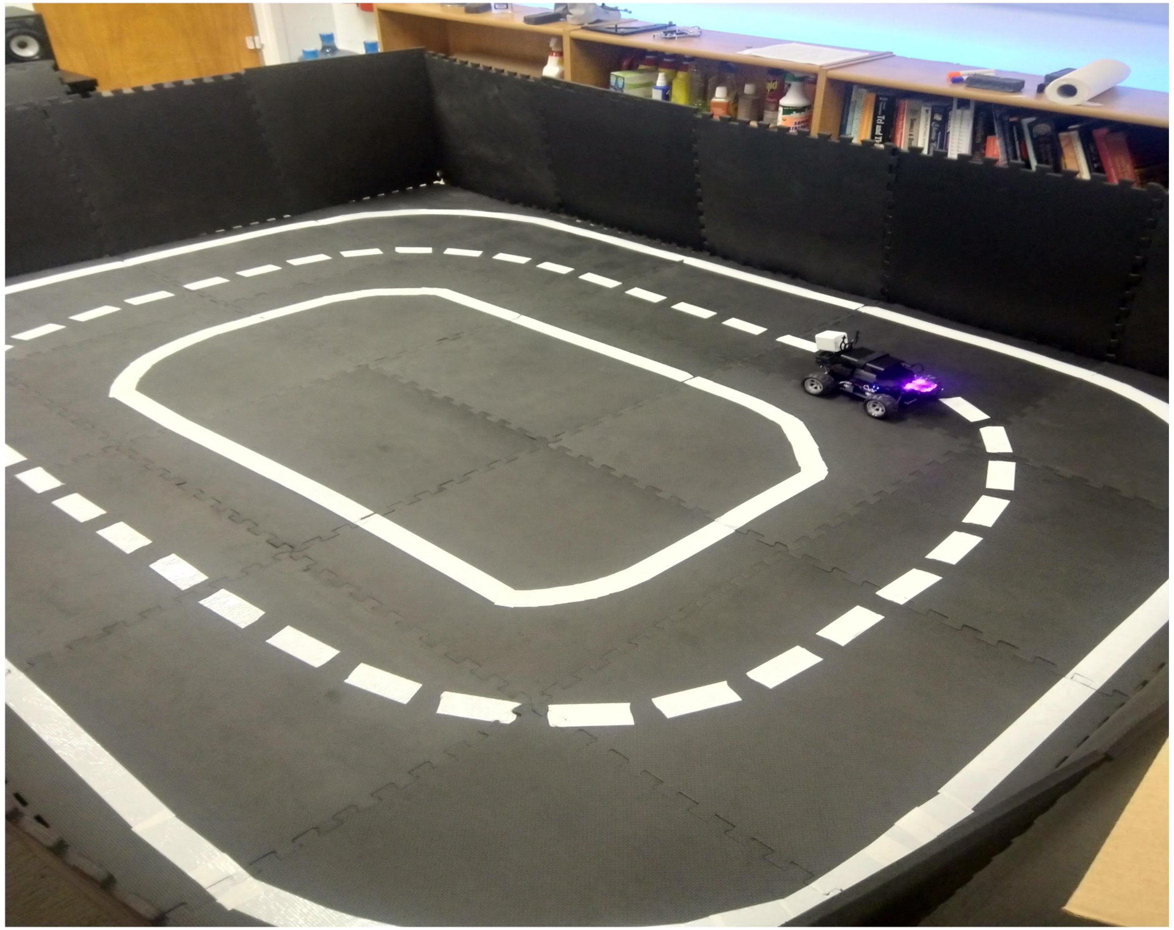
The impact of Timing on Closed-Loop System



Stay within the track



Sampling Rate and End-To-End Delay Variations



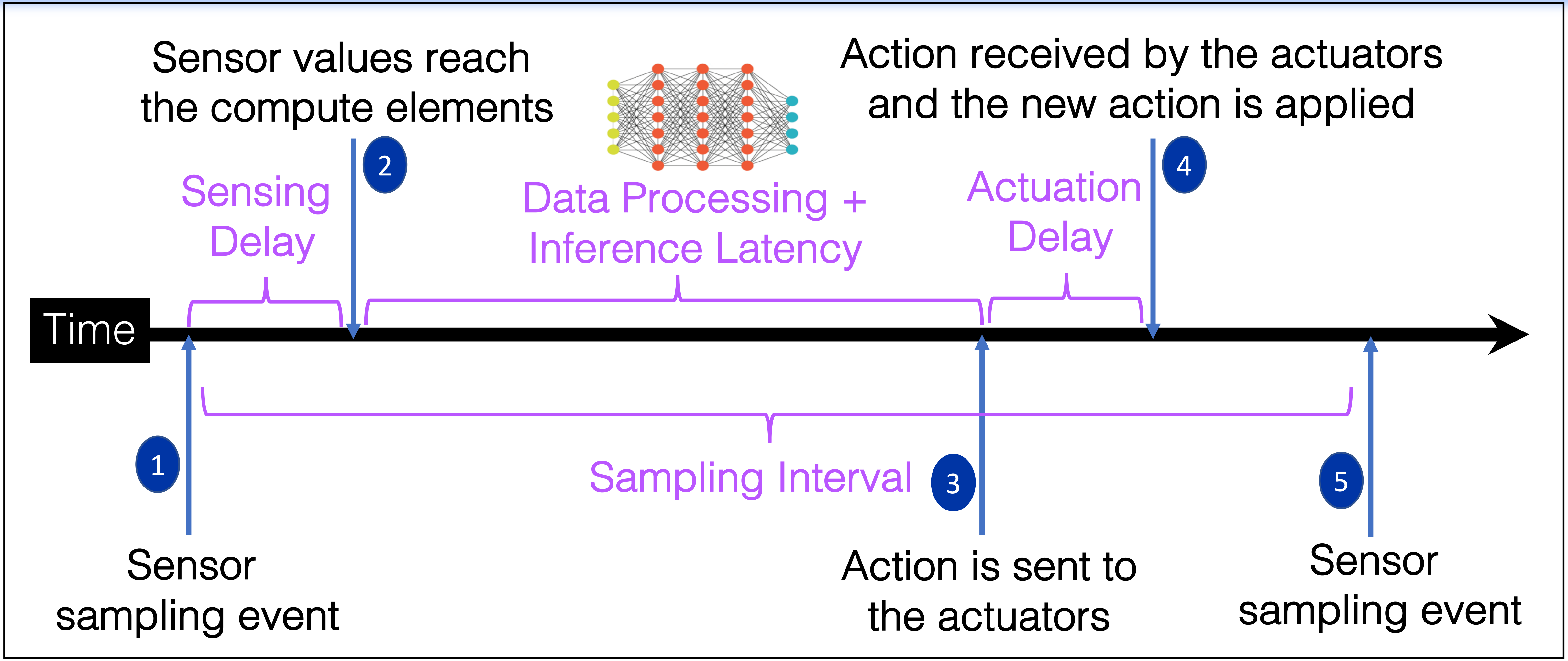
Multitude of reasons

- Multitenancy
- Hardware heterogeneity
- Thermal throttling
- Battery budget
- Compute availability
- Complexity of controller

Impact

- Action of car is delayed => crash.

Delays in a typical Deep RL: Sensing to Actuation Pipeline



Execution Latency = Data Processing + Inference Latency

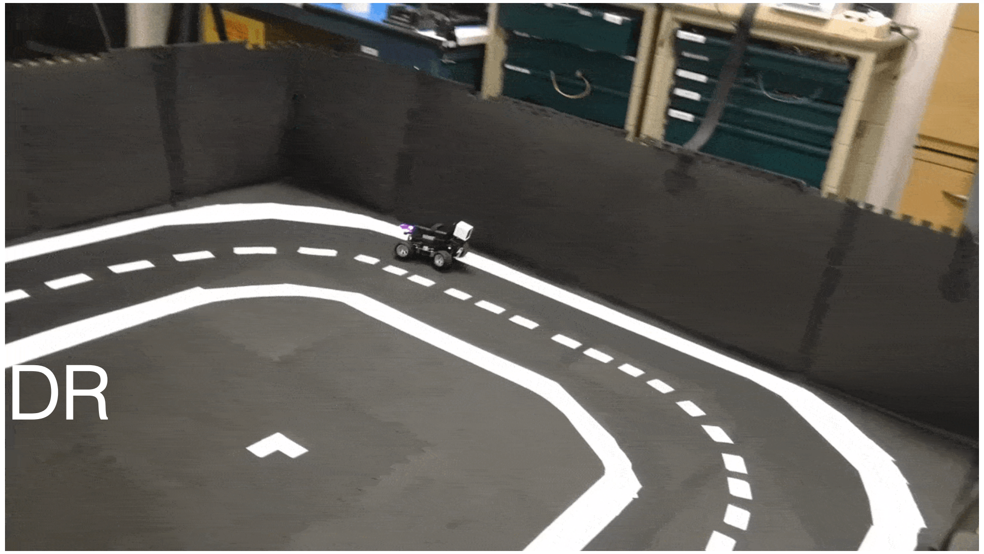
Our Approach: Time-in-State Reinforcement Learning

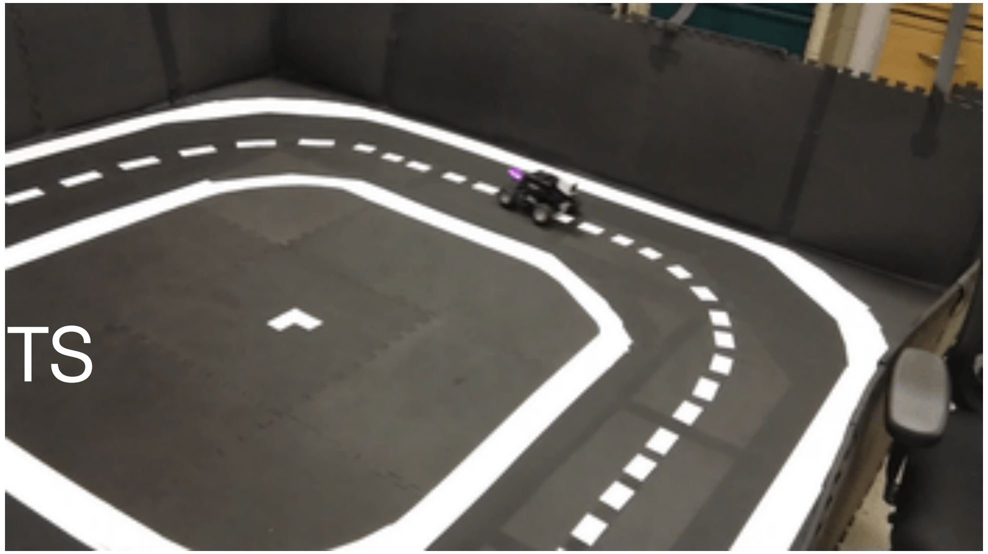
Application can monitor and adapt to the continuous changes in the execution latency and sampling rate at runtime.

Evaluation on 1/18th Scale Car

Delay	20ms	60ms	100ms
Baseline (DR)	20	11	7
Our Approach (TS)	20	17	13

Laps Completed (out of 24)





Sandeep Singh Sandha, Luis Garcia, Bharathan Balaji, Fatima Anwar, Mani Srivastava, "Sim2Real Transfer for Deep Reinforcement Learning with Stochastic State Transition Delays," Conference on Robot Learning (CoRL), 2020.

Acknowledgements

Names: left to right

- My Advisor Prof. Mani Srivastava
- Outside collaborators
    - Luis Garcia (USC), Fatima Anwar (UMass), Bharathan Balaji (Amazon) & Supriyo Chakraborty (IBM)
- Collaborators from NESL
    - Swapnil Sayan Saha, Joseph Noor & Tianwei Xing.

Q/A