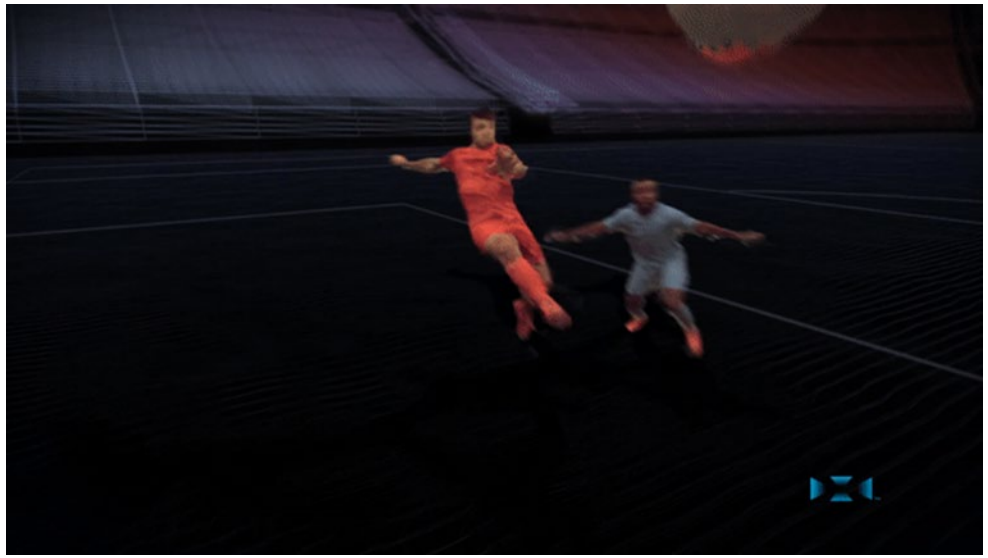
Understanding Precision Time Protocol in Today's Wi-Fi Networks

Paizhuo Chen, **Zhice Yang**

ShanghaiTech University

上海科技大学
ShanghaiTech University

Example of Freeview Point Video



Shutters are synchronized

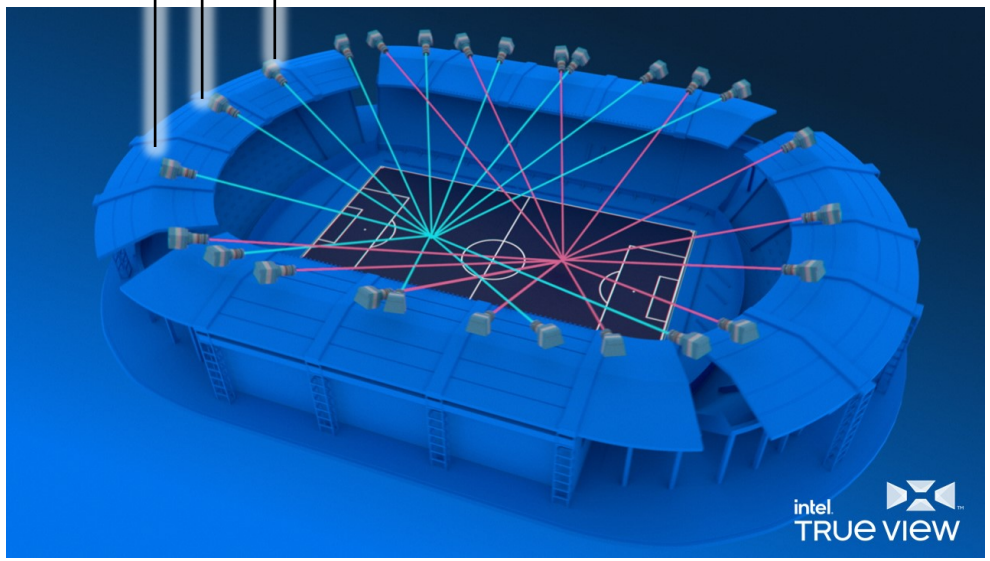


Image from Intel True View

PTP over Wi-Fi ?

- Precision Time Protocol (PTP)
  - IEEE 1588
  - The standard way of synchronizing networked systems in the LAN scale
- PTP over Wi-Fi
  - Plenty of research
    - Reported accuracy: µs-level (software), below ns (new hardware)
  - In practice
    - No open implementation
    - We tried existing software solutions, accuracy is in the ms level

PTP over Wi-Fi ?

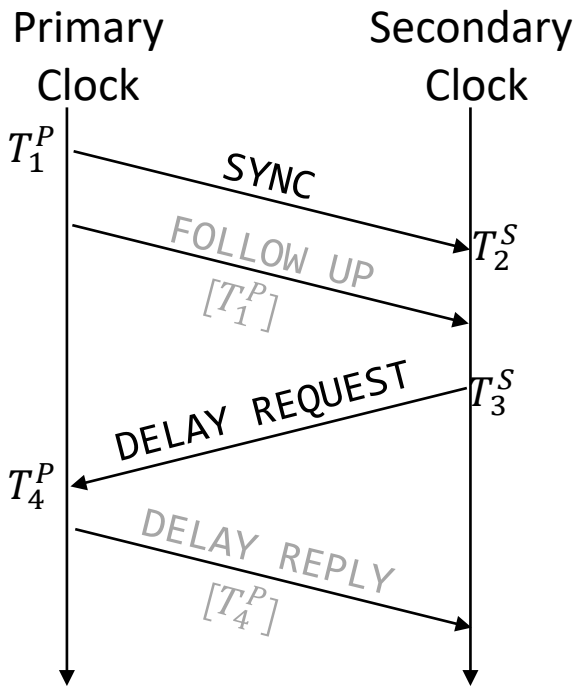
- Precision Time Protocol (PTP)
  - IEEE 1588
  - The standard way of synchronizing networked systems in the LAN scale
- PTP over Wi-Fi

1. Why do modern devices render poor sync performance ?

  - No open implementation

2. What is the best practice of applying PTP in Wi-Fi ?

PTP Overview

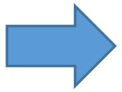


- Goal (simplified): sync the secondary clock to the primary clock
- Approach:
  - Use network packets to trigger time readings of clocks
  - Use round-trip measurement to cancel out the network delay
- Secondary clock obtains four timestamps
  - $T_1^P, T_2^S, T_3^S, T_4^P$
- Time offset to the primary clock is:
  - $\Delta = \frac{(T_1^P - T_2^S) + (T_3^S - T_4^P)}{2}$
- Assumption:
  - Timestamp is precisely bound to the packet event
    - Advance or delay results in sync error

PTP Implementation

✓Upper Part

- PTP protocol
  - Several implementations, e.g., `linuxPTP`
  - Maintain protocol state machine, clock tuning algorithm

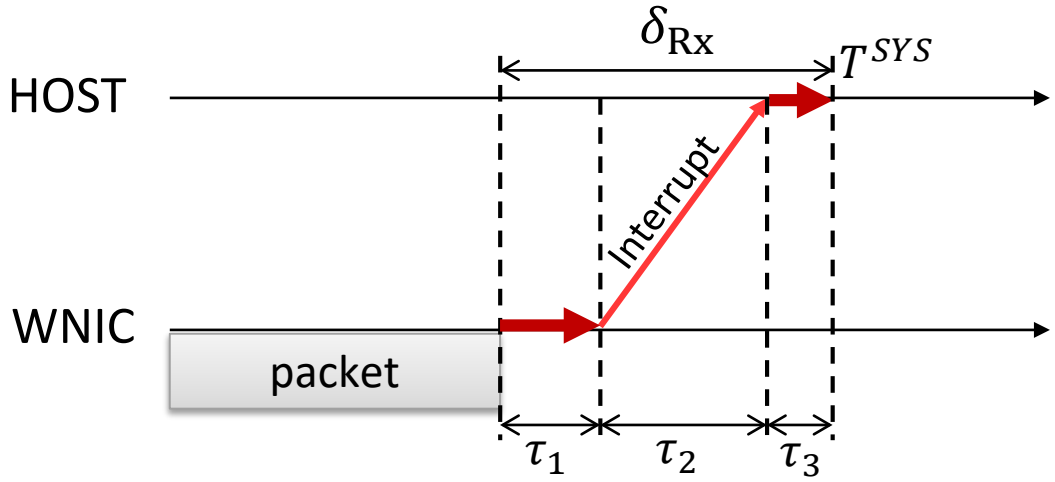


Lower Part

- Timestamping
  - Software PTP
    - Take timestamps in the software system
  - Hardware PTP
    - Take timestamps in the NIC hardware

Software PTP – Timestamp Errors

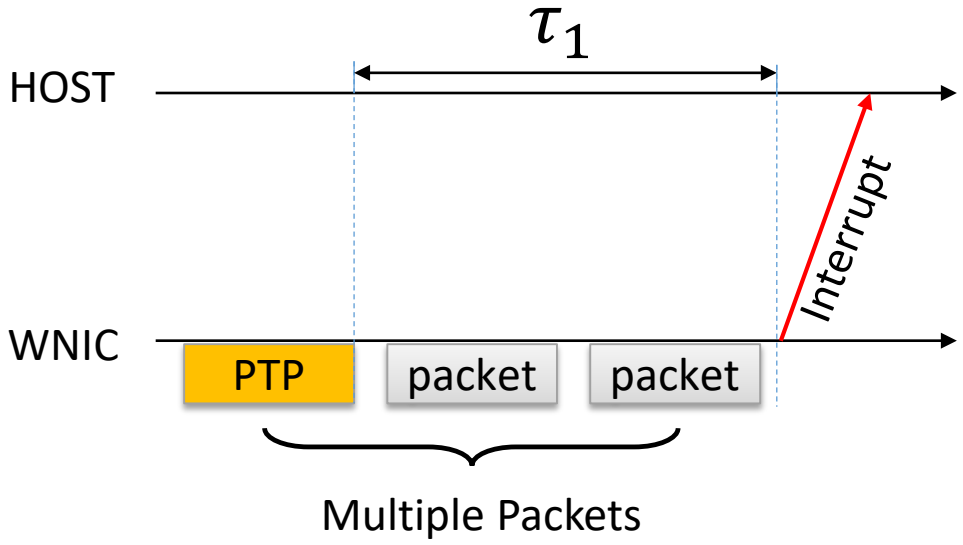
- Latency of Software Timestamps



- $\boldsymbol{\tau_1}$: NIC hardware interrupt delay
- τ_2: Interrupt propagation delay
- $\boldsymbol{\tau_3}$: Host response delay

Software PTP – Error Source 1

- Interrupt Mitigation
  - Main part of τ_1
    - Heavy network load: 1000 μs
    - Light network load: 500 μs



PC i5-8500 + AR9388

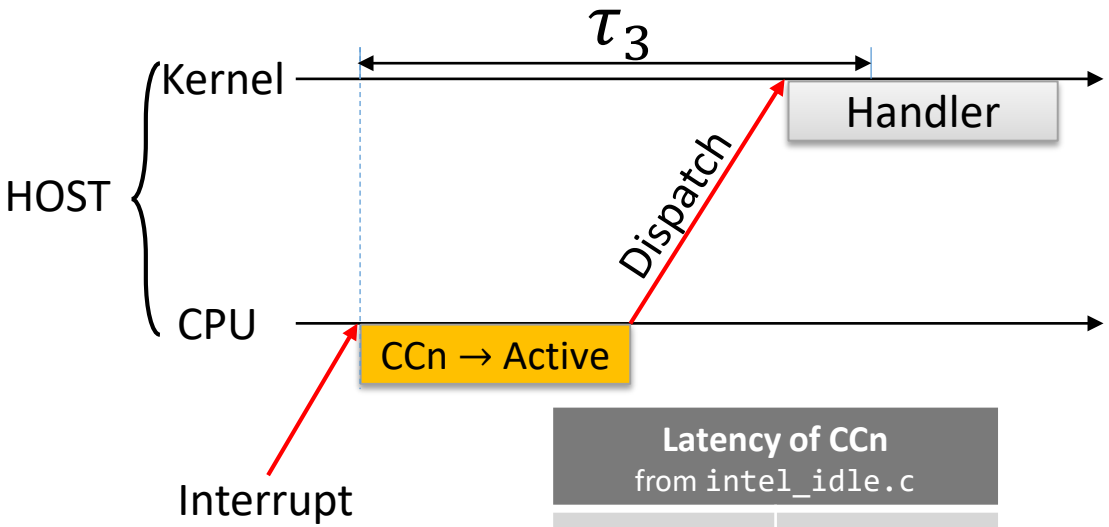
Interrupt Mitigation	Network Load (PPS)	CPU Load 0%	CPU Load 50%	CPU Load 100%
Enable	100	674.312	600.265	518.98
	1000	558.823	546.797	520.132
	5000	1093.454	1097.67	1087.88
Disable	100	175.552	105.281	20.497
	1000	60.305	42.361	20.147
	5000	37.531	23.923	19.951

Mean of total latency* $\delta_{\mathrm{Rx}} = \boldsymbol{\tau_1} + \tau_2 + \tau_3$ (μs)

*Larger value -> More sync error

Software PTP – Error Source 2

- CPU Idle Power Management
  - Main part of τ_3



Latency of CCn from `intel_idle.c`	
Active	0 μs
CC1	2 μs
CC3	70 μs
CC6	85 μs
CC8	200 μs

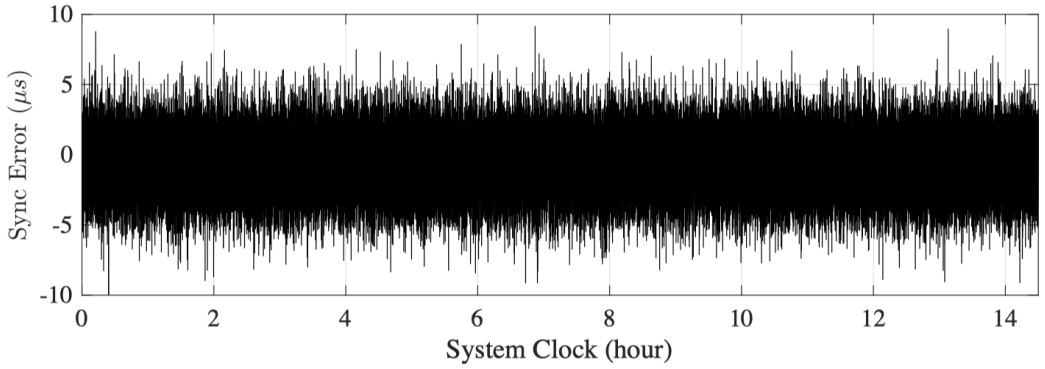
PC i5-8500 + AR9388

CPU C-state	Interrupt Mitigation	Network Load (PPS)	CPU Load 0%	CPU Load 50%	CPU Load 100%
PC C0~C8	Enable	100	674.312	600.265	518.98
		1000	558.823	546.797	520.132
		5000	1093.454	1097.67	1087.88
	Disable	100	175.552	105.281	20.497
		1000	60.305	42.361	20.147
		5000	37.531	23.923	19.951
PC C0 Only		100	19.92	19.89	20.16
		1000	19.75	19.93	19.91
		5000	19.66	19.67	19.93

Mean of total latency* $\delta_{\mathrm{Rx}} = \tau_1 + \tau_2 + \boldsymbol{\tau_3}\ (\mu s)$

*Larger value -> More sync error

Software PTP - Performance



- $\text{mean} = 0.17\ \mu s$, $\text{std} = 1.8\ \mu s$
- Random CPU/network load is applied to the Wi-Fi client during the 14-hour measurement

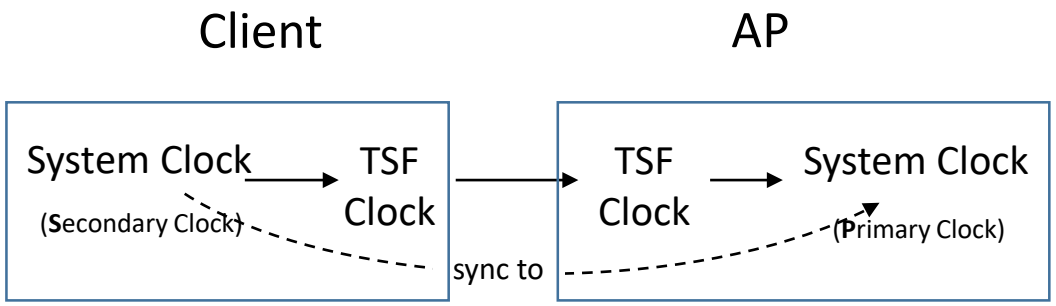
Primary Clock
(i5-8500 4GB;
Wi-Fi AP mode)

Sync to via Wi-Fi

Secondary Clock
(Jetson Xavier, ARM;
Wi-Fi Client Mode)

Hardware PTP – Overview

- Why
  - Software PTP limitations: power consumption, host platform heterogeneity, etc.
- Idea
  - Use TSF counter as the hardware PTP clock
    - TSF counter is driven by a 1 MHz tick source
    - TSF counter can be accessed by the host system
    - TSF counter timestamps packets in the WNIC hardware



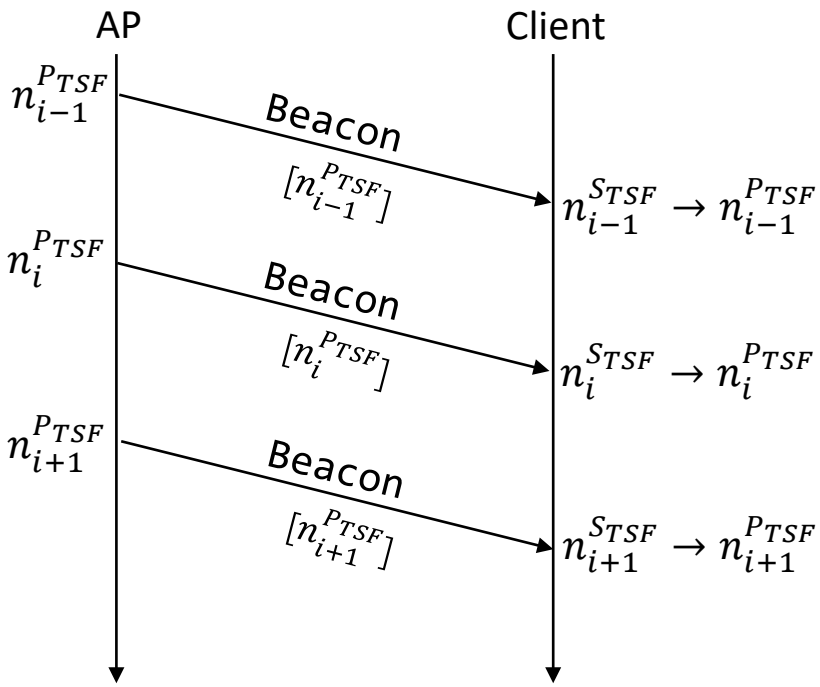
Similar to Ethernet NIC's PTP clock

TSF counter can be used as a clock

TSF counter can be used as a PTP clock

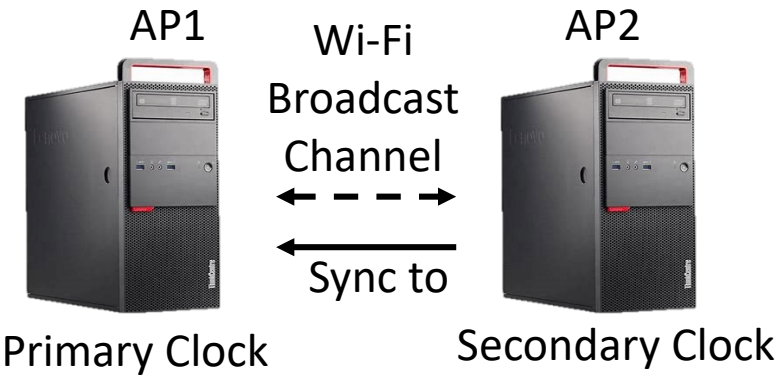
Hardware PTP – Impact of TSF

- TSF counter is designed for fulfilling the IEEE 802.11 Time Synchronization Function (TSF)
  - AP periodically broadcasts its TSF counter value in beacons
  - Client resets its TSF counter to the AP's
    - ➢ TSF clock is not a free-running clock !



Hardware PTP – Impact of TSF

- TSF counter is designed for fulfilling the IEEE 802.11 Time Synchronization Function (TSF)
  - AP periodically broadcasts its TSF counter value in beacons
  - Client resets its TSF counter to the AP's
    - ➢ TSF clock is not a free-running clock !
  - Use an AP to host the secondary clock to measure the accuracy loss
    - Broadcast PTP packets to allow overhearing between APs
    - AP's TSF clock is free running
    - No obvious performance difference
      - The beacon rate is frequent enough

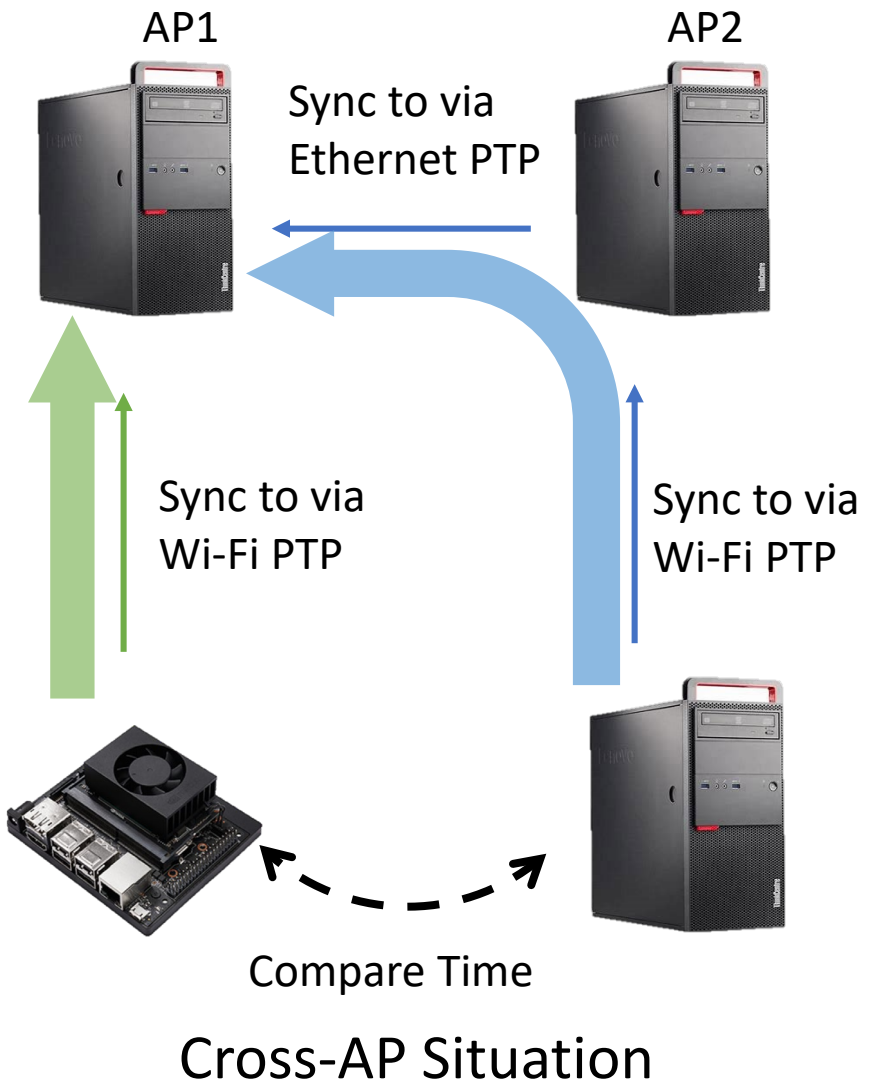


Mode	Beacon Interval (ms)	mean (µs)	std (µs)
free	102.4 (default)	0.17	0.39
no free	102.4 (default)	0.18	0.37

Sync Error Statistics

Hardware PTP - Performance

- 14-hour measurement
  - $\text{mean} = -0.09\ \mu s$, $\text{std} = 0.63\ \mu s$
- Tested under various wireless conditions
  - Mobility, signal strength
    - No impact
  - Interference
    - Harsh interference reduces accuracy due to the reduction of valid PTP packets
- Cross-AP situation
  - Mean is close to the single AP situation, but std is larger (about ×2)
    - Reason: longer sync chain



Cross-AP Situation

Hardware PTP – Implementation

- ath9k driver patch
  - Introduce a PTP hardware clock module
    - mimic the PTP hardware clock in Ethernet PTP NIC drivers
  - Tested with ath9k WNICs
- Source code is available at:
  - https://github.com/Wireless-Lab/Wi-PTP

Conclusion

- Present a timely and detailed PTP performance study on modern mobile (Wi-Fi) platforms
- Software PTP
  - Take timestamps in the driver's interrupt handler
  - Major impacting factors:
    - WNIC interruption mitigation
    - Host CPU idle power management
  - A flexible and general approach
- Hardware PTP
  - Reuse TSF counter for timestamping
  - Accurate and stable
  - No extra host configurations, but rely on hardware support
- An open source hardware Wi-Fi PTP implementation

Thank You !

Contact: Zhice Yang yangzhc@shanghaitech.edu.cn