Santa Clara University
Internet of Things
RESEARCH LAB
SIOTLAB

IEEE Wireless Communications and Networking Conference (WCNC)

Traffic Characterization for Efficient TWT Scheduling in 802.11ax IoT Networks

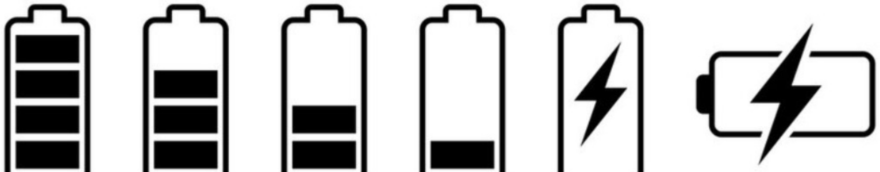
Jaykumar Sheth, Vikram K. Ramanna, and Behnam Dezfouli

Glasgow, Scotland, UK
March 2023

Introduction

- **The adoption of WiFi technology, especially for IoT connectivity**
  - WiFi technology provides **higher communication rates** compared to technologies such as Bluetooth and ZigBee
  - WiFi communication utilizes **unlicensed spectrums**
    - Deployments are considerably less expensive than cellular technologies
  - **An omnipresent infrastructure for connectivity**
    - **Distributed and customer-oriented deployment of WiFi networks** in residential and enterprise settings
    - Low deployment costs

Introduction



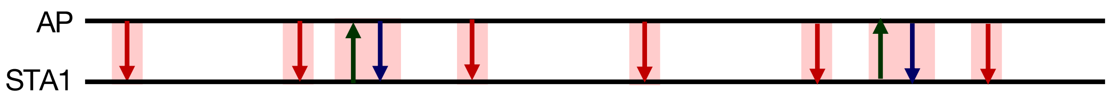
- **Many IoT devices rely on limited energy resources such as batteries or energy harvesting**

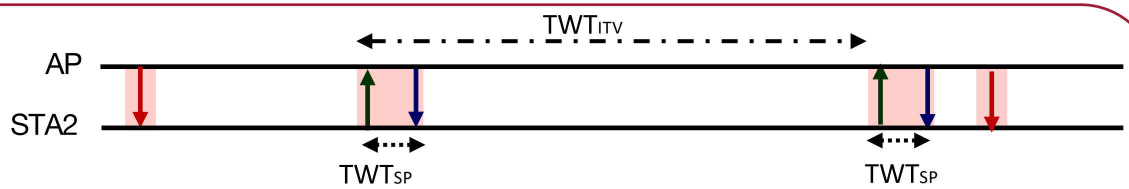
- **The 802.11 standard offers various power-saving modes:**
  - Power Save Mode (PSM)
  - Adaptive PSM (APSM)
  - Adaptive Power Save Delivery (APSD)

Introduction

802.11
b, g, n, ac

- STA wakes up every beacon





802.11 ax

- A new functionality introduced called Target Wake Time (TWT)
- Allows negotiation of sleep schedule
- STA does not need to wake up every beacon

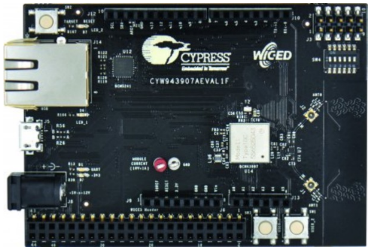
Notation	Meaning
	Beacons
	Uplink packets
	Downlink packets
	STA awake duration
TWT_{ITV}	TWT Interval
TWT_{SP}	TWT Service Period

- TWT allows STA to switch to low-power sleep state for a prolonged period
- How to accurately characterize traffic for allocating TWT_{SP} schedule?

Traffic Pattern Analysis

We study the following IoT applications (interference-free environment)

- **Sensing**
  - An RTOS development kit (CYW54907) collecting accelerometer readings
  - Periodically collects 3920 samples (equivalent to 5880 bytes), prepares packets, and then sends them via a TCP connection

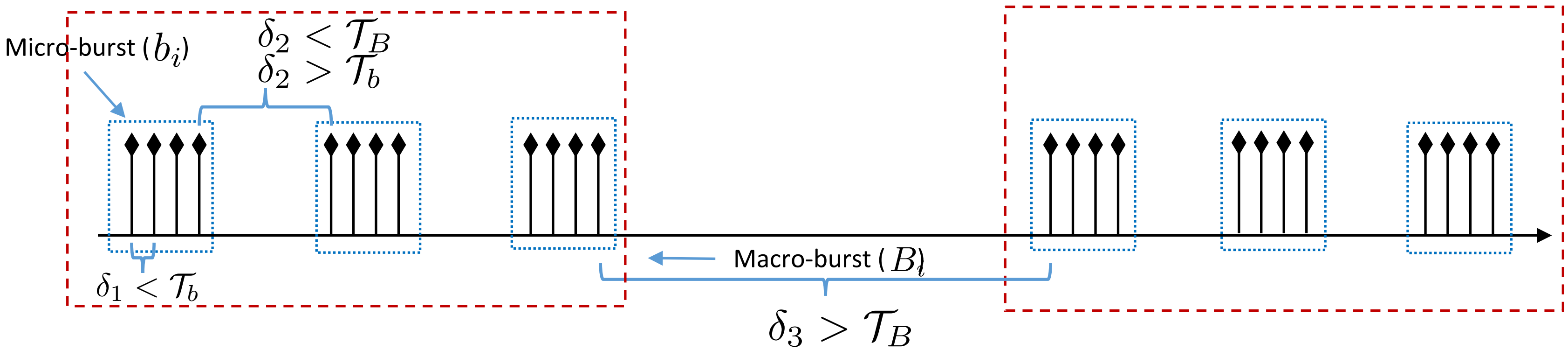


- **Camera**
  - A security camera using RPi camera module (version 2) attached to a RPi 3B+
  - Captures and sends images via a TCP connection
    - Each image is processed by the H.264 codec

- **Video Streaming**
  - A YouTube video is streaming on an Amazon Echo Show device

Traffic Pattern Analysis

- To build a generalized traffic analysis framework, we consider three inter-packet intervals



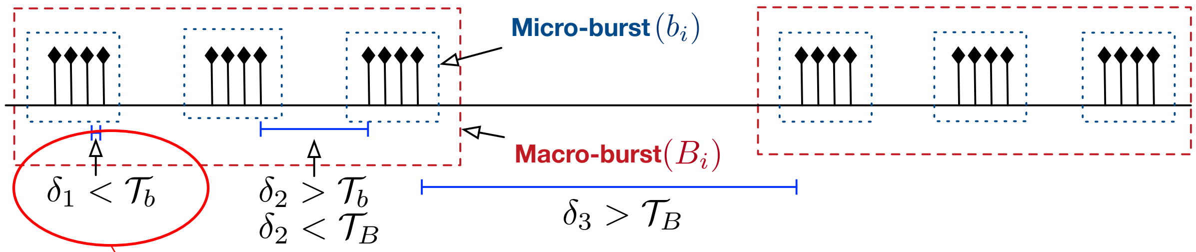
Traffic components:

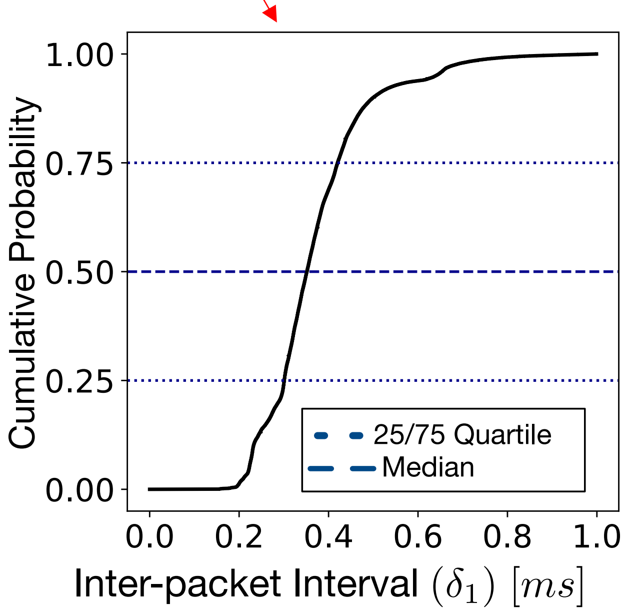
- Macro-burst $\Longrightarrow$ Power save mechanism employed
- Micro-burst $\Longrightarrow$ Contention or traffic generation pattern
- Packets within a Micro-burst $\Longrightarrow$ Packet preparation delay or Transmission parameters of 802.11

Traffic Pattern Analysis

Sensing

- Within a micro-burst, the mean interval between packets (δ_1) is about 400 μs





- Collecting 3920 samples from accelerometer
  - Equivalent to (3920 * 12 bits)/8 = 5880 bytes

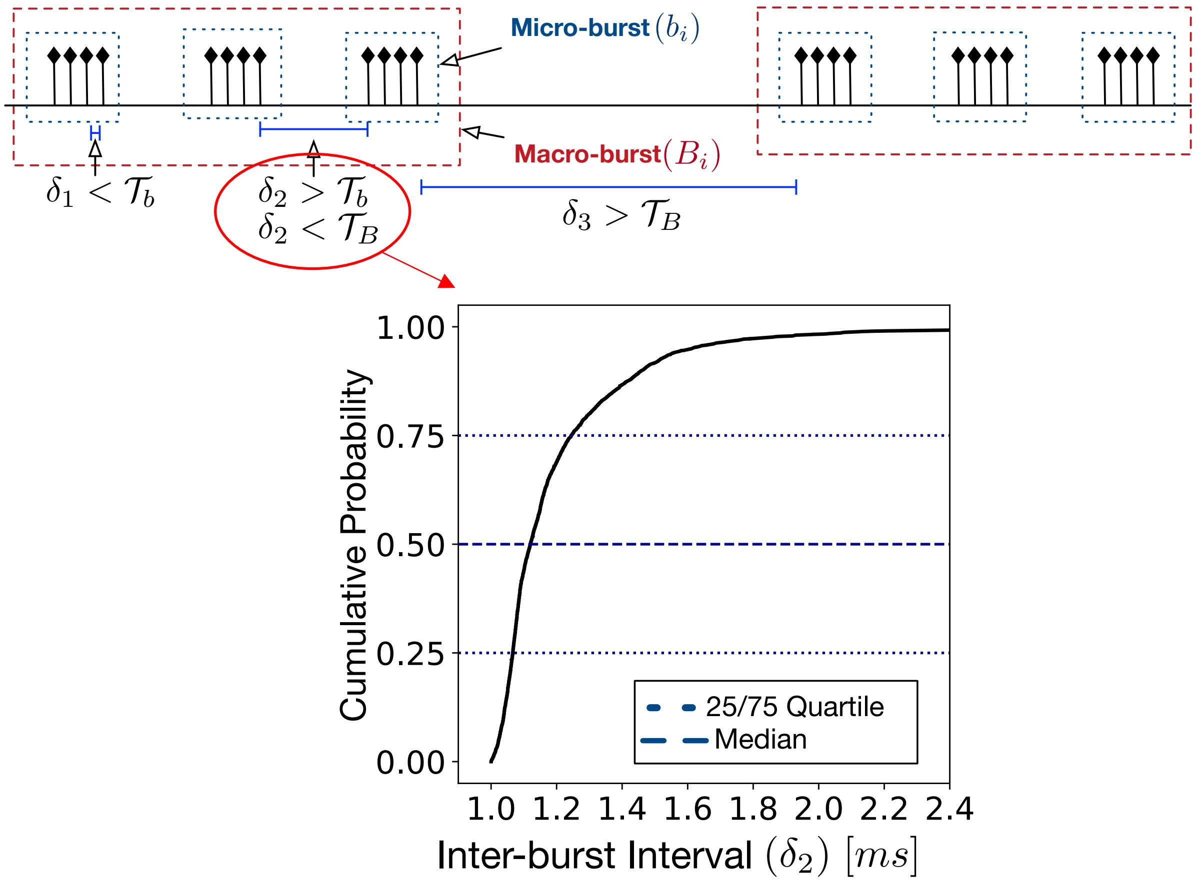
Main factors affecting inter-packet interval

- Prepare multiple packets
- Transfer packets from driver to NIC
- Send multiple packets – backoff and channel access contention

Traffic Pattern Analysis

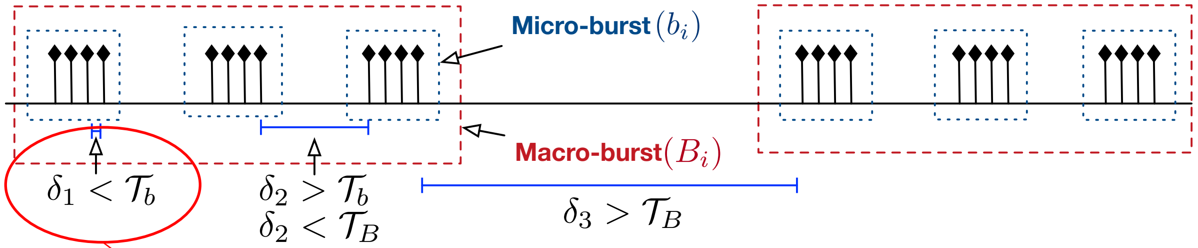
Sensing

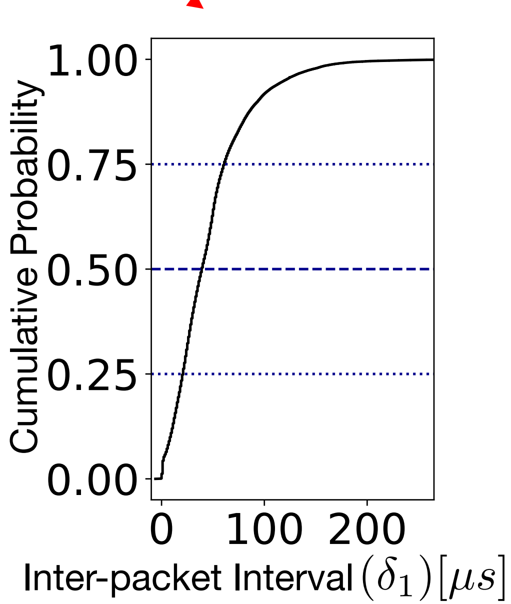
- The interval between micro-bursts is due to collecting 3920 samples
  - Communication between the processor and ADC over the Serial Peripheral Interface (SPI) to collect samples



Traffic Pattern Analysis

Camera

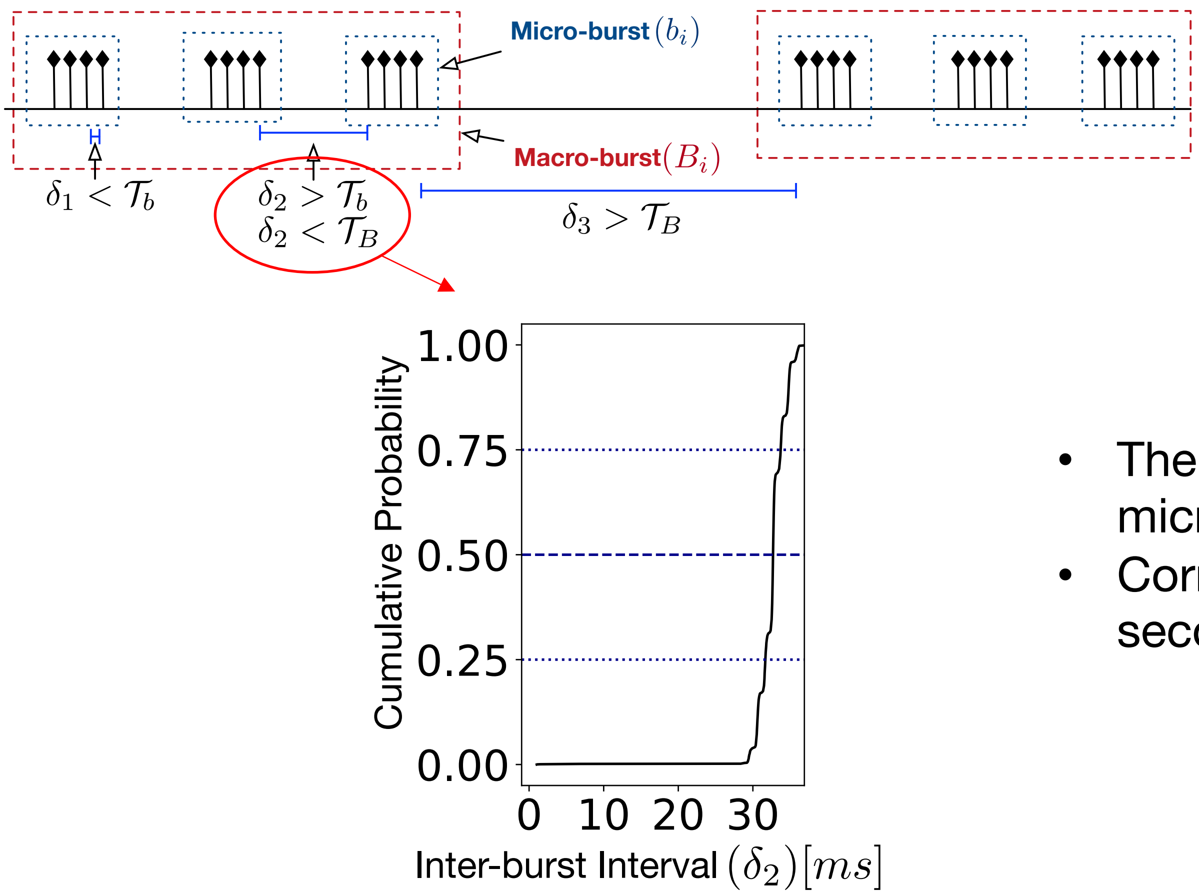




- The camera captures a frame
- Prepares multiple packets to send the frame

Traffic Pattern Analysis

Camera



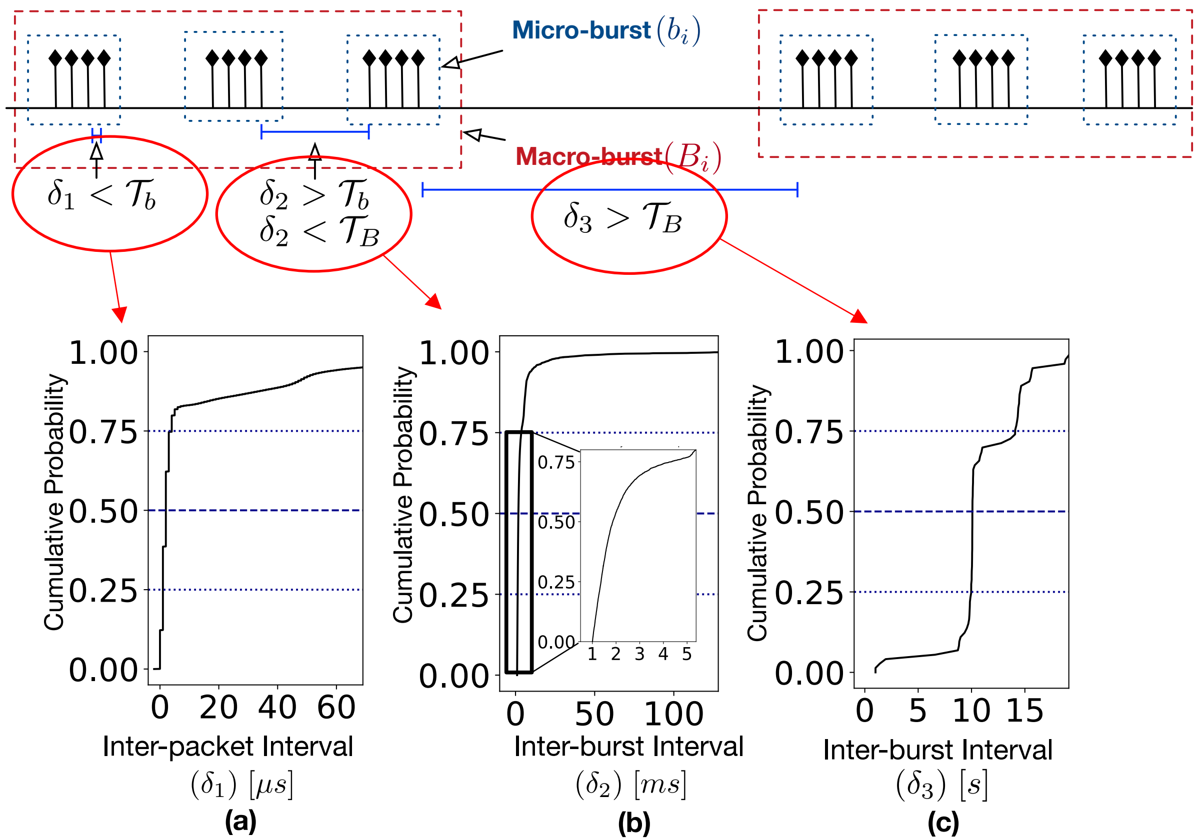
(b)

- The interval between microbursts (δ_2) is about 33 ms
- Corresponds to 30 frames per second

Traffic Pattern Analysis

Video Streaming

- The mean interval between packets of a micro-burst (δ_3) is 9 us
- The mean interval between micro-bursts (δ_2) is 2 ms
- The mean interval between macro-bursts (δ_1) is 10 seconds



Traffic Pattern Analysis

- In sum, these experiments show that traffic characterization can be used for:
  - Allocating TWT service periods based on each STA's demands
  - Utilizing inter-packet intervals by other STA to enhance throughput
  - Enhancing packet aggregation performance

Existing Methods

- Channel Utilization (CU) estimation
- Buffer Status Report (BSR)
- Packet sniffing

- Do not provide high accuracy

Gradual TWT adjustment

- High communication overhead
- Cannot quickly address traffic variations

Assumptions about STAs

- Unrealistic
- Impossible

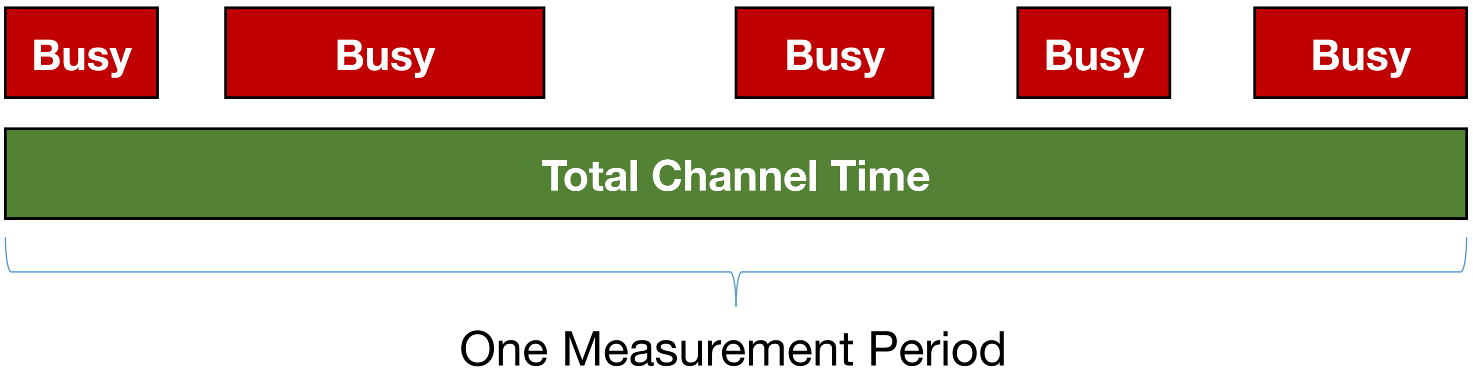
Traffic classification (e.g., using machine learning)

- High processing overhead
- Difficult implementation

Existing Methods

Channel Utilization (CU)

- Channel Utilization (CU) is defined as $t_{activity}/t_{total}$



- CU values can be collected from the driver via various methods such as the ‘proc’ file system (`procfs`) in Linux
- **The information provided by CU is cumulative**
  - **It cannot be used to characterize per-STA traffic patterns**

Existing Methods

Packet Sniffing

- Using one or more additional NICs to sniff packets

- Shortcomings:
  - The timestamps of sniffed packets do not represent the actual packet generation instances by STAs
  - Additional hardware (NIC) and processing resources
  - Mismatch between sniffed packets and those exchanged by the AP's main NIC

Existing Methods

Buffer Status Report (BSR)

```
> Qos Control: 0x0c10
v HT Control (+HTC): 0x0000750f
    .... .... .... .... .... .... .... ...1 = VHT: True
    .... .... .... .... .... .... .... ..1. = HE: True
  v Aggregate Control: 0x1d43
      Control ID: 3: Buffer status report
    v Buffer Status Report: 0x000001d4
        .... .... .... .... .... .... .... 0100 = ACI Bitmap: 0x4
        .... .... .... .... .... .... ..01 .... = Delta TID: 0x1
        .... .... .... .... .... .... 11.. .... = ACI High: 0x3
        .... .... .... .... .... ..01 .... .... = Scaling Factor: 0x1
        .... .... .... ..00 0000 00.. .... .... = Queue Size High: 0x00
        .... ..00 0000 00.. .... .... .... .... = Queue Size All: 0x00
> CCMP parameters
```

- **Introduced in 802.11ax (a.k.a., WiFi 6)**
- **Queue Size All (QSA)** field of BSR conveys the cumulative amount of data in all queues
- BSR also conveys information such as the Queue size of the highest-priority Access Category (AC)

Existing Methods

Buffer Status Report (BSR)

Shortcomings of using BSR for traffic characterization:

- We selected several COTS 802.11ax NICs and empirically analyzed them
  - Intel AX200 and Realtek RTL8852A **transmit BSR intermittently**
    - Based on the amount of traffic queued
  - In contrast, Compex WLT639 **includes a BSR in every packet**

- None of the evaluated AP and STAs support requesting or generating BSR manually
- Also, we observed that for those 802.11ax devices that include BSR in each packet, **all the MPDUs included in an A-MPDU report the same value**
  - Even though the payloads they are carrying have been generated at different time instances.
  - The reported value is the state of queues before the transmission of A-MPDU

Source-assisted Traffic Characterization (SATRAC)

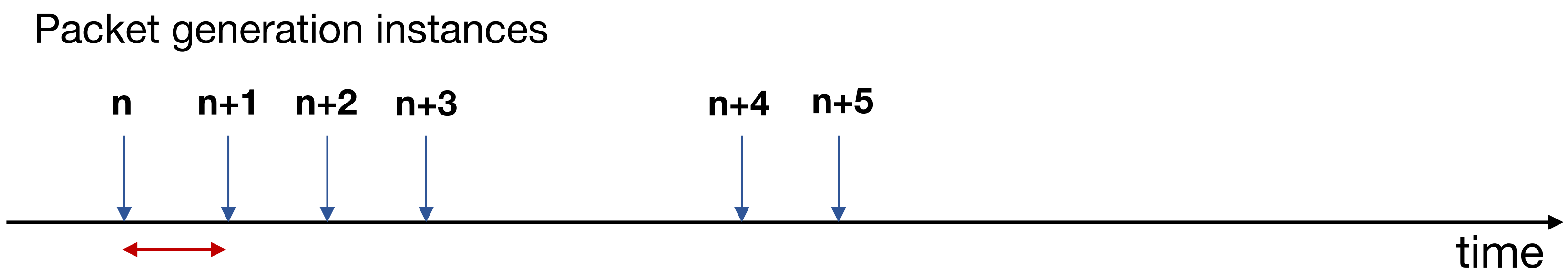
Overview

- Basic idea:
  - Keep track of packet generation time instances in each STA
  - Convey to AP
  - The AP can construct the traffic pattern of the STA
- This method is unaffected by packet preparation delay, channel access contention, interference, and packet loss
- Each STA modifies packets in their protocol stack's data-path
  - Adds timing information
  - Similar to in-band network telemetry (INT)

Source-assisted Traffic Characterization (SATRAC)

Overview

- Instead of including an absolute timestamp in each packet, **we include only a 2-byte value**
- This value encodes **the the difference between the generation time of the current packet and the previous packet of the same flow**

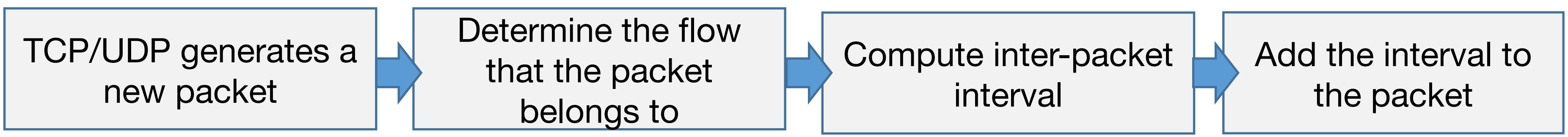


- The interval between each two consecutive packets generation instances is encoded as a 2-byte value and added to the second packet
- Here, the interval between n+1 and n is included in packet n+1

Source-assisted Traffic Characterization (SATRAC)

Overview

- Each STA computes a unique 5-tuple hash value for each flow and keeps track of the timestamp of the last generated packet

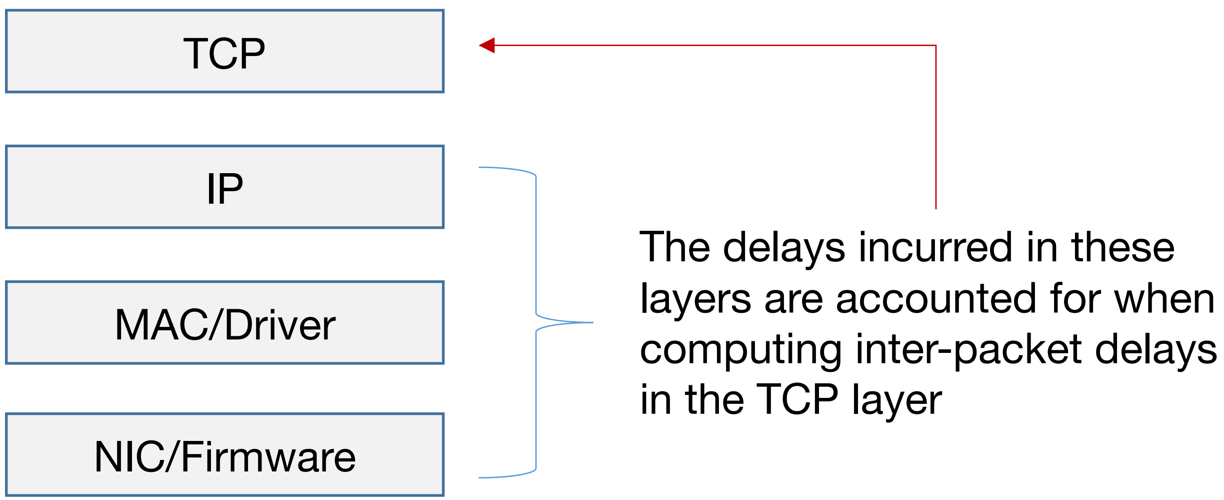


- Where is this information added to?
  - TCP Options field, or
  - IPv4 Options field, or
  - IPv6 Next Header

Source-assisted Traffic Characterization (SATRAC)

Packet Preparation Delays

- Add timing information in the TCP layer when the TCP protocol prepares the TCP header
- To account for packet preparation delay, we add the delays caused by the IP layer, MAC layer, and driver-to-NIC handoff to the timing information



Source-assisted Traffic Characterization (SATRAC)

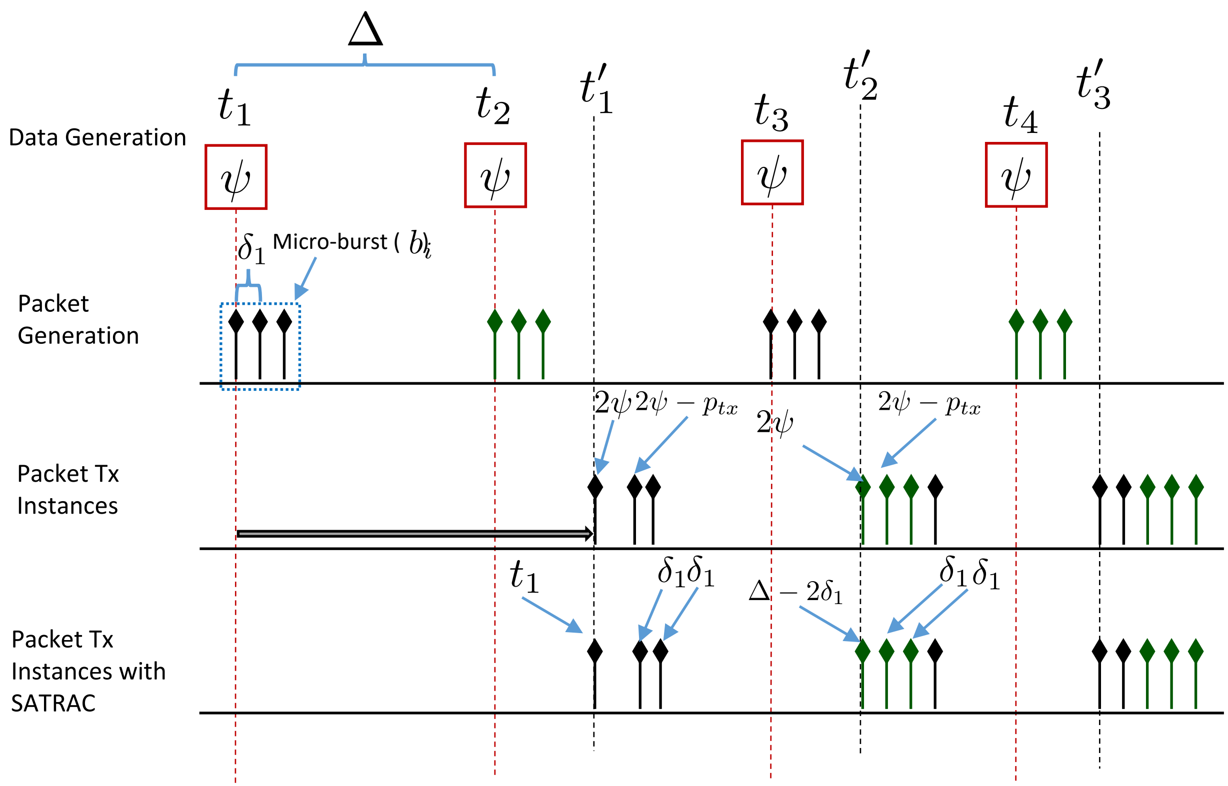
eBPF

- We leverage **eBPF** and build an application-agnostic middleware for **setting the TCP Options field**

- Since APs run Linux, a similar eBPF program extracts and parses the values included in TCP Options field of packets received from STA to characterize uplink traffic

Source-assisted Traffic Characterization (SATRAC)

Comparison of Traffic Characterization Methods



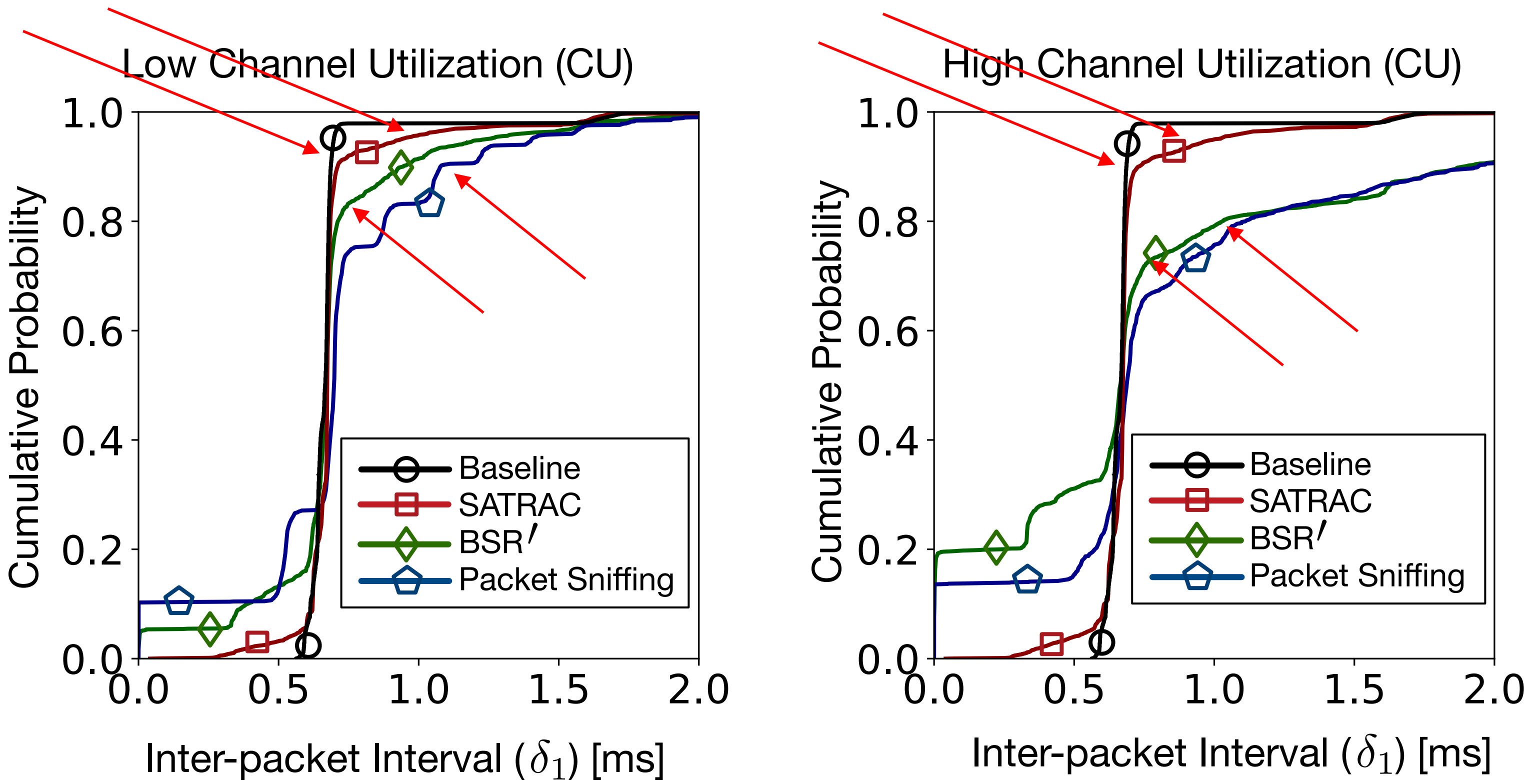
Source-assisted Traffic Characterization (SATRAC)

Empirical Comparison

- An 802.11ax (WiFi 6) testbed: one AP and multiple STAs
- We consider two CU scenarios
  - **Low**: the measured CU is around 15%
  - **High**: the measured CU is around 70%
- A STA runs a program that generates and sends a 1400-byte message every 500 us
  - Voice Access Category – no packet aggregation
- To establish a **baseline** for accuracy comparison, we denote the actual data generation instances by the application as *baseline*

Source-assisted Traffic Characterization (SATRAC)

Empirical Evaluation of Timing Accuracy



(a) **(b)**

Source-assisted Traffic Characterization (SATRAC)

Empirical Comparison of TWT Allocation Efficiency

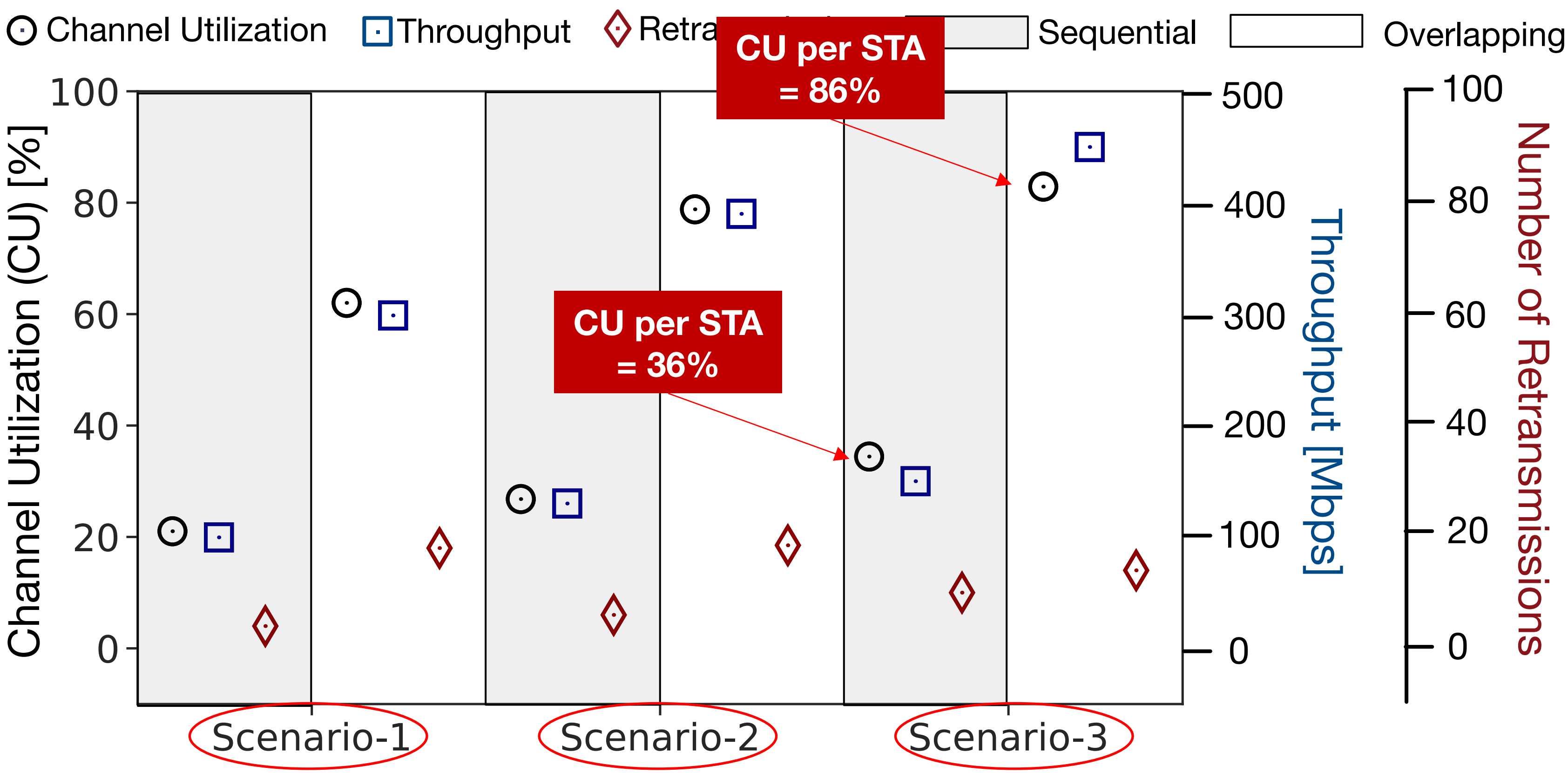
- An 802.11ax (Wi-Fi 6) testbed: one AP and multiple STAs

- **Sequential Allocation of TWT Service Period**
  - Similar to the existing works, we assign non-overlapping service periods to the STA

- **Overlapping Allocation of TWT Service Periods using SATRAC**
  - We enable the AP to characterize traffic using SATRACT
  - Determine the possibility of higher channel utilization, and assign overlapping service periods to the STA

Source-assisted Traffic Characterization (SATRAC)

Empirical Comparison of TWT Allocation Efficiency



Conclusion

- To meet applications' demands while enhancing energy efficiency and throughput
  - Traffic characterization is required for the allocation of TWT service periods to IoT STAs

1. We empirically studied traffic burstiness and the causes of inter-packet delays in WiFi-based IoT networks
2. Analyzed the shortcomings of existing traffic characterization methods
3. Introduced a novel approach based on packet modification in STAs' protocol stack
4. Empirically measure the accuracy of the proposed method and analyzed its effect on TWT allocation

- Not only for TWT allocation, but the proposed method can also be used for Resource Units (RU) allocation

Contact Details

Jaykumar Sheth
jsheth@alumni.scu.edu

Vikram Ramanna
vramannna@scu.edu

Behnam Dezfouli
bdezfouli@scu.edu