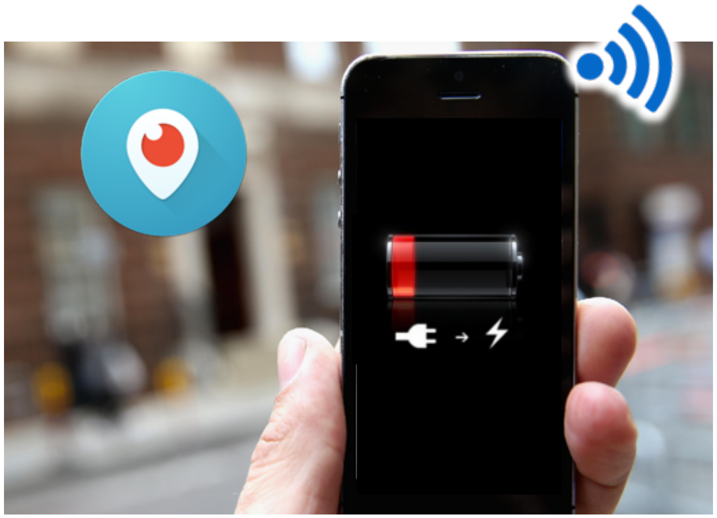
Trimming the Smartphone Network Stack

Yanzi Zhu, Yibo Zhu*, Ana Nika,
Ben Y. Zhao, and Heather Zheng

SAND Lab, UCSB *Microsoft Research
yanzi@cs.ucsb.edu

Networking Consumes Energy



Live Broadcasting

Media Streaming

- Existing energy measurements† show that networking costs ~50% energy of a daily app

† Chen, X., and et al. "Smartphone Energy Drain in the Wild: Analysis and Implications." In *Proc. of SIGMETRICS* (2015).

Reducing Networking Energy

- Networking subsystem as a black box
  - ON/OFF power management
  - Downclocking (MobiCom’11, NSDI’13, MobiCom’14)
  - …

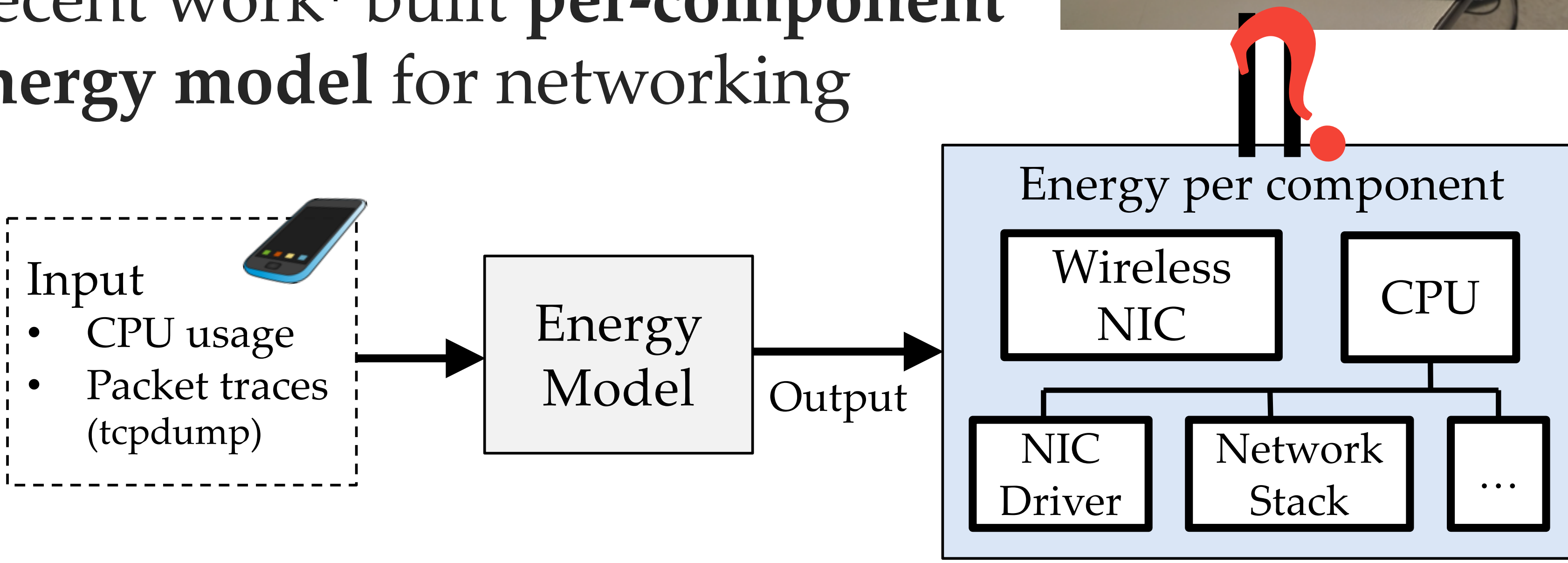
- What happens inside the box?

- Challenge: lack of componentized energy analysis

Componentized Energy Model

- Power meter measures energy of the entire phone, not individual components

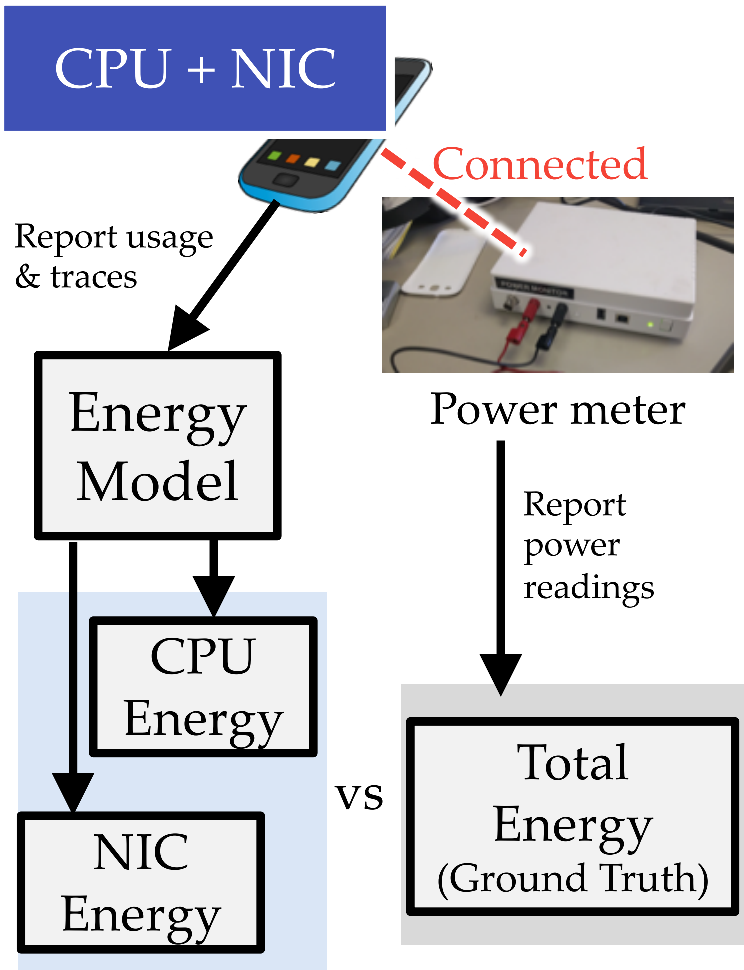
- Recent work† built **per-component energy model** for networking



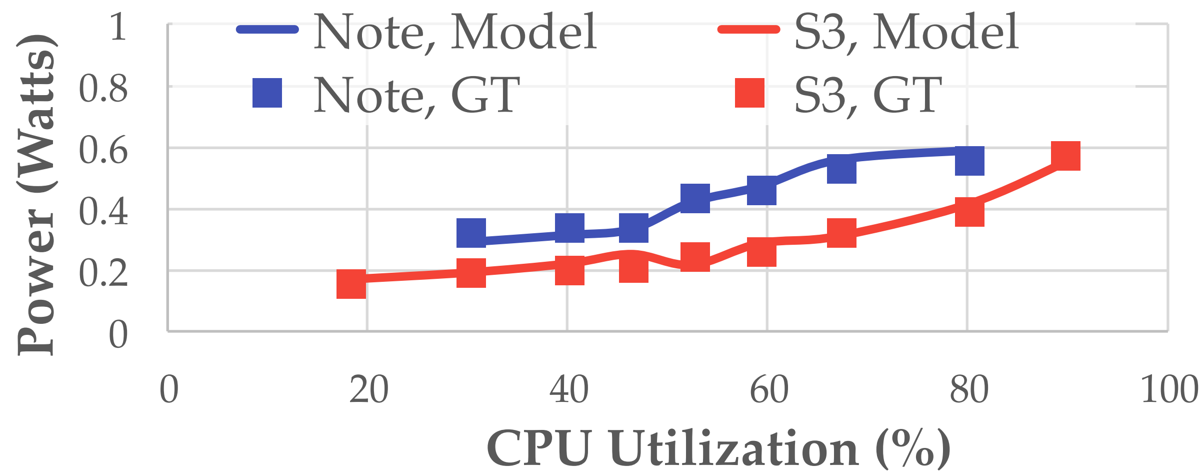
† Nika, A., and et al. "Energy and performance of smartphone radio bundling in outdoor environments." In *Proc. of WWW* (2015).

Validating Energy Model

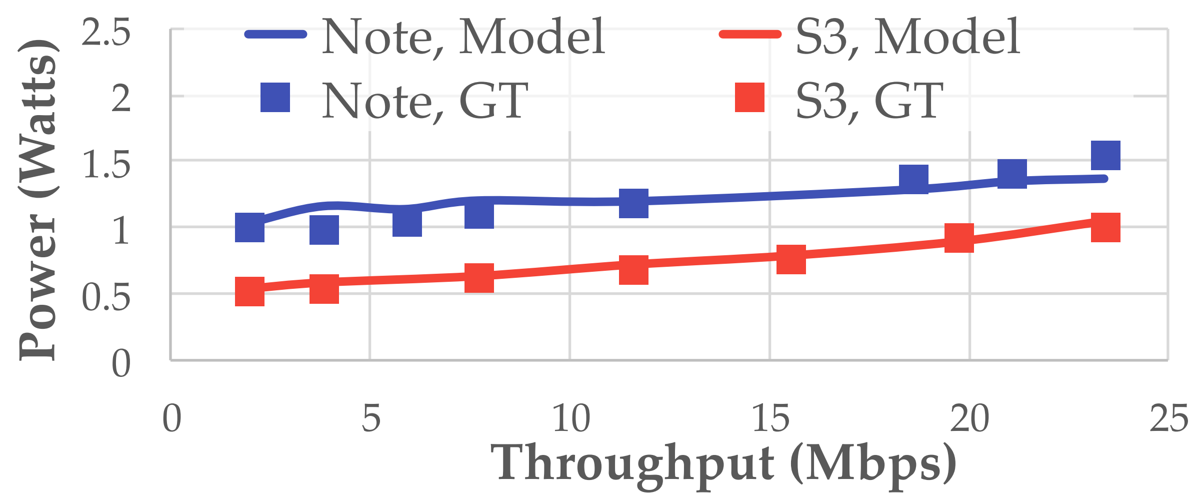
Two Phones
(S3 & Note)



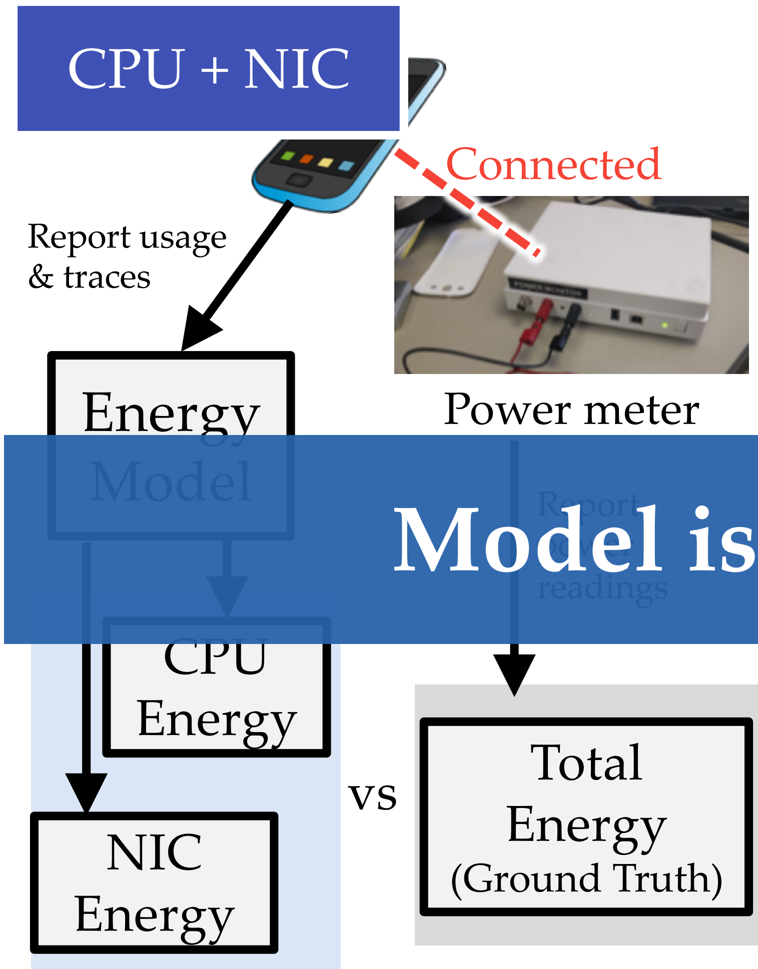
Test 1: Run CPU only

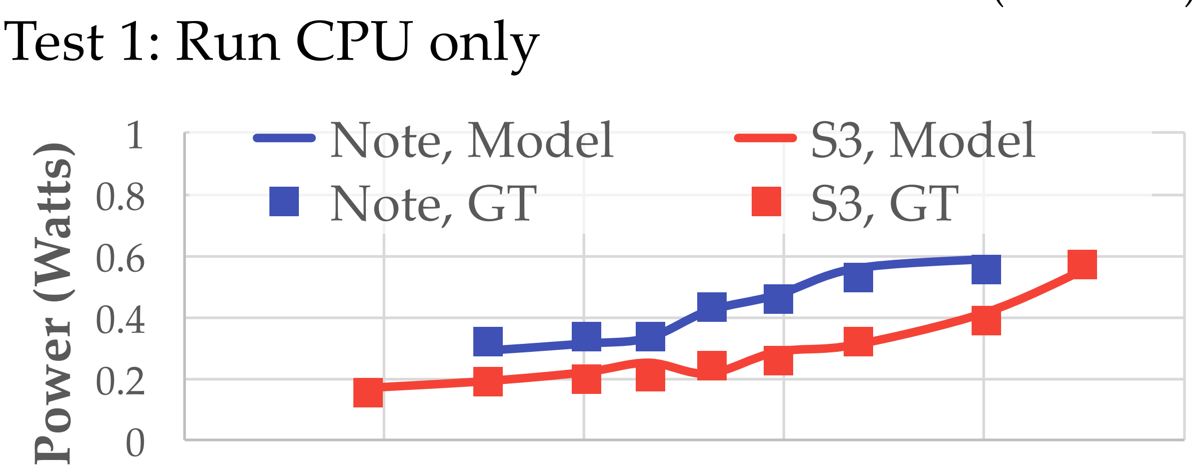


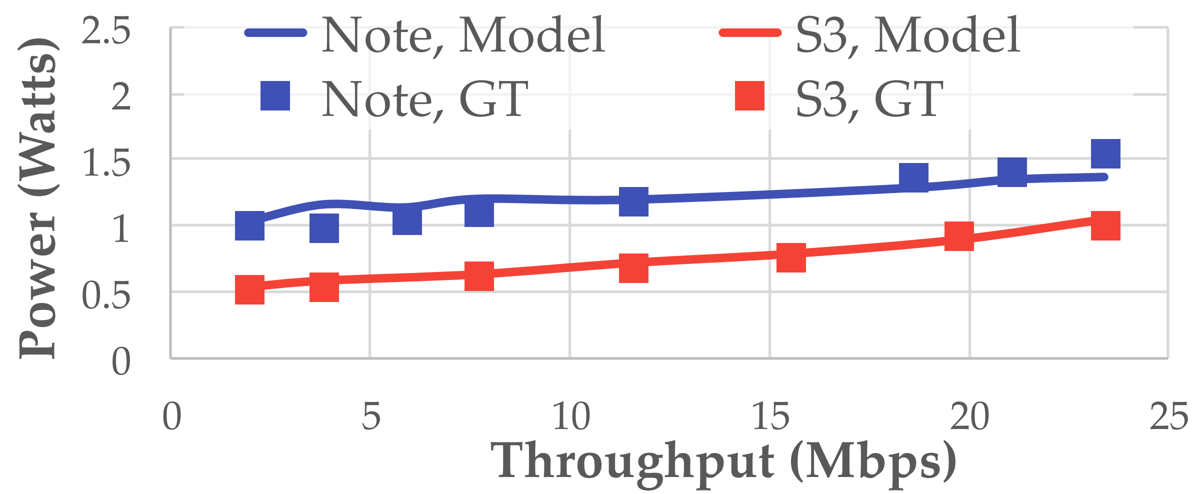
Test 2: Run NIC + CPU



Validating Energy Model



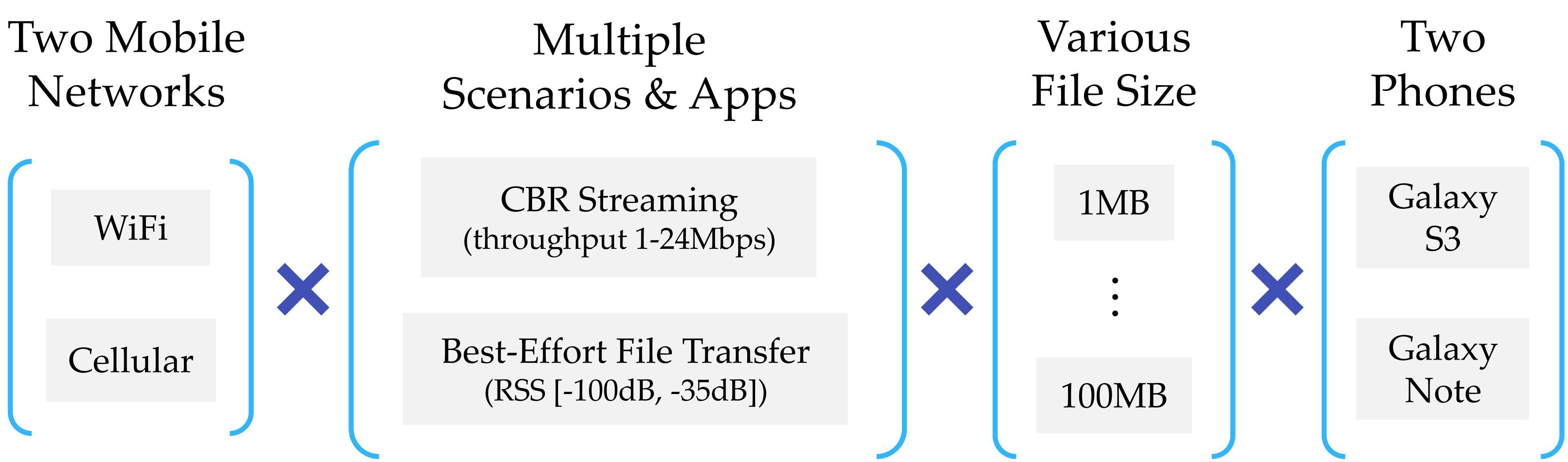




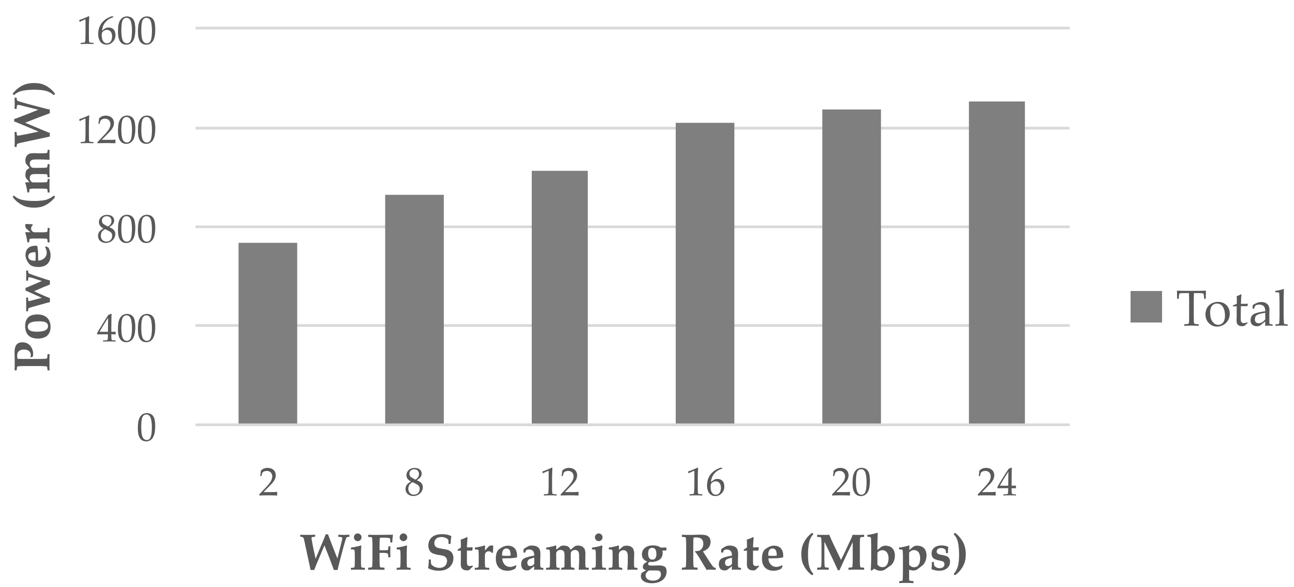
Model is accurate (error < 10%)

Model-Based Energy Analysis

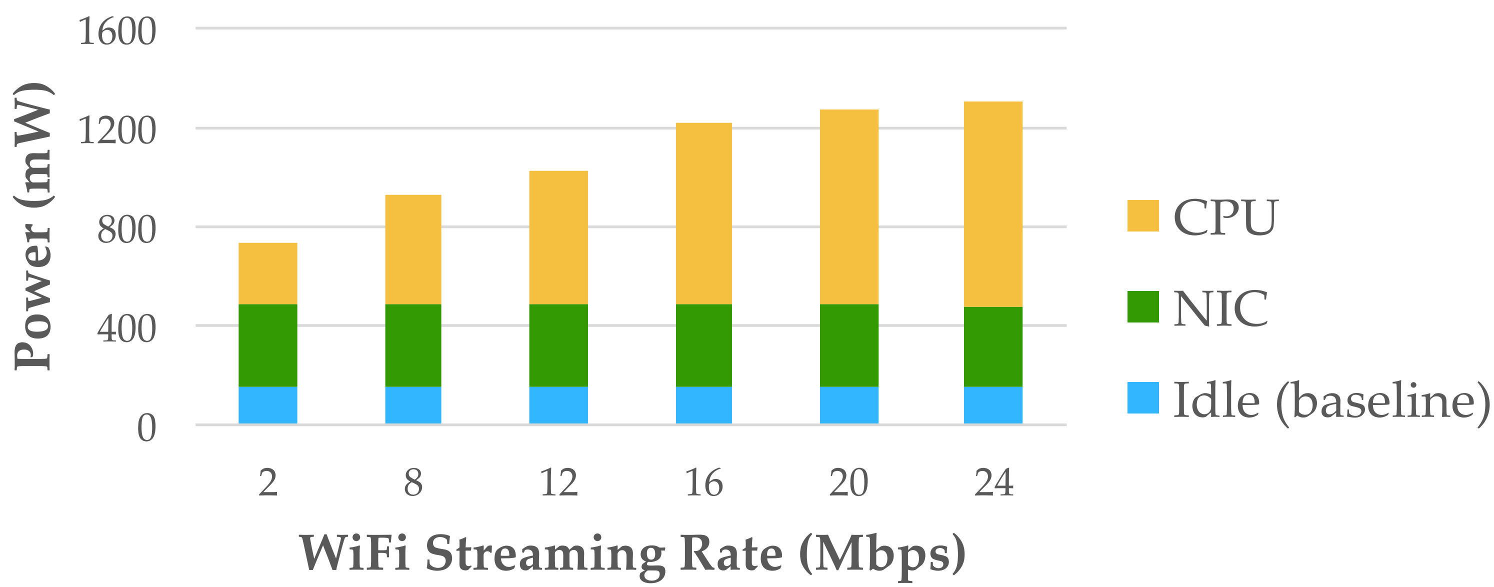
- Isolating *networking*
  - Screen turned off, no other apps, etc.
  - Minimal logging overheads (<5% CPU usage)
- Extensive experiments



Key Finding 1: Large CPU Energy Cost



Key Finding 1: Large CPU Energy Cost

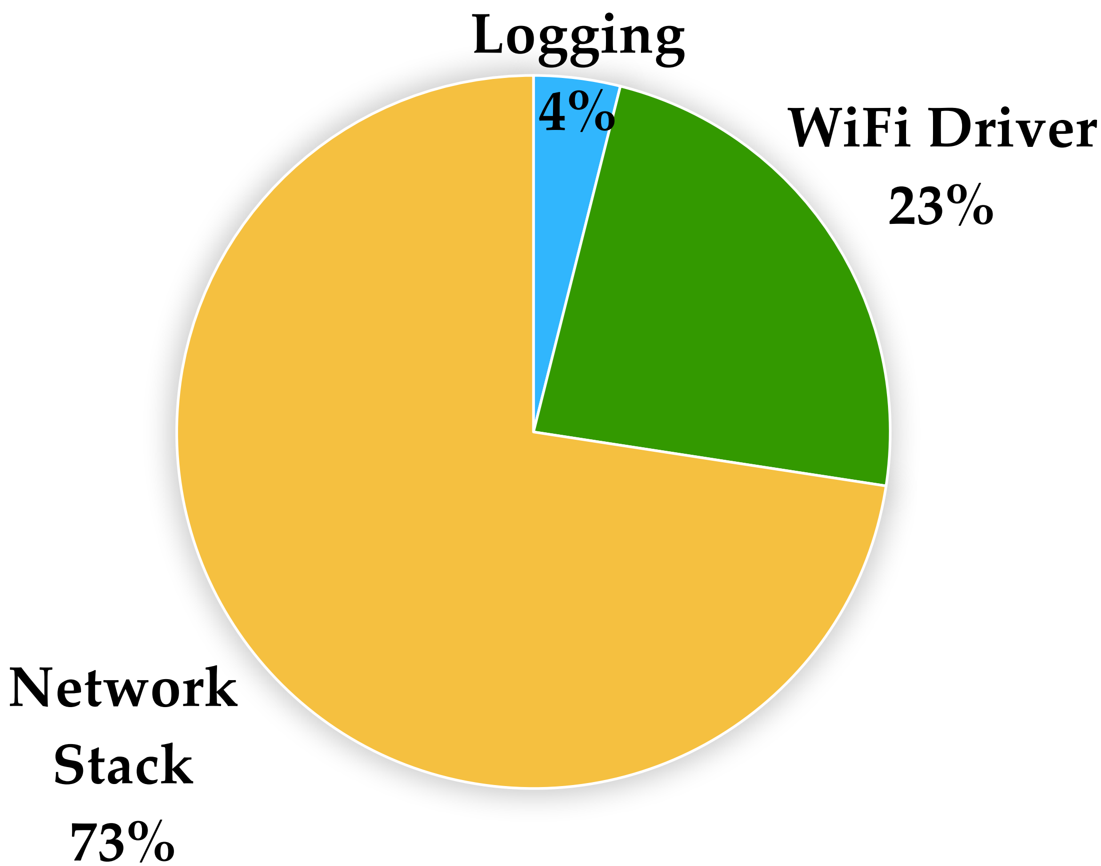


- CPU draws considerable amount of power
  - Scale with streaming rate
- CPU takes up to 60% energy (WiFi) and 20% (LTE)
  - Up to 800mW (WiFi) and 600mW (LTE)
  - WiFi NIC consumes 200–900mW†

† Nika, A., and et al. “Energy and performance of smartphone radio bundling in outdoor environments.” In *Proc. of WWW* (2015).

A Deeper Look at CPU for Networking

- Energy breakdown of processes



Key Takeaways

1. CPU consumes significant energy in networking

2. Network stack processing consumes a lot of CPU

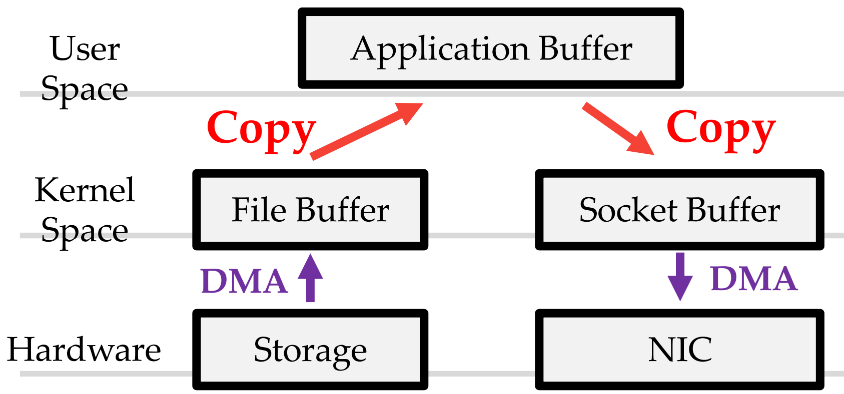
Rest of This Talk

- Cut CPU usage by trimming the network stack
  - Reduce memory copies: one-copy
  - Reduce TCP protocol processing: TCP offloading

Method 1: Reduce Memory Copies

Convention: Two-Copy

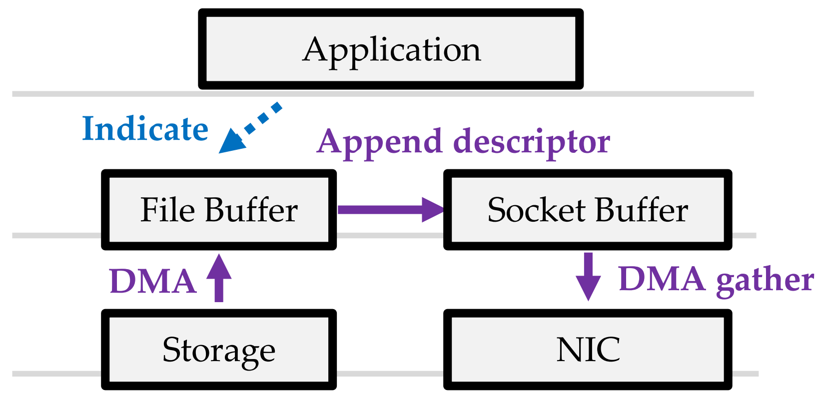
(upload)



- Memory copies:
  kernel → user space → kernel
- Unnecessary because streaming apps do not modify data

Zero-Copy

(upload)

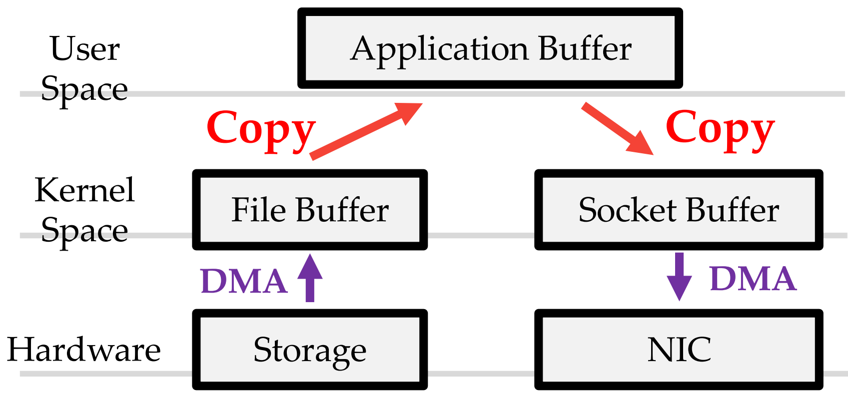


- Zero-copy requires NIC's support (memory gather operation)

Method 1: Reduce Memory Copies

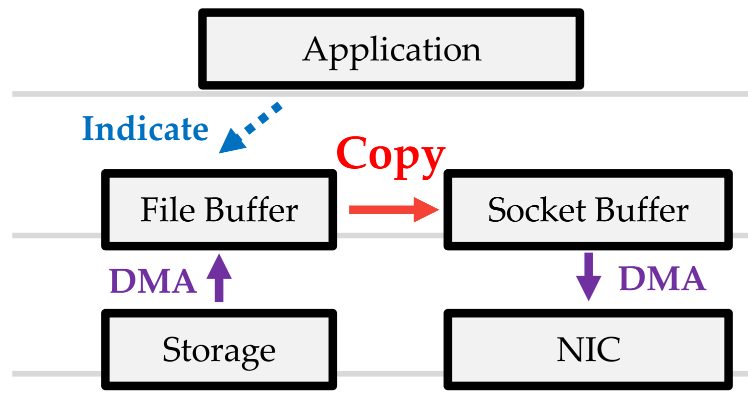
Convention: Two-Copy

(upload)



One-Copy

(upload)

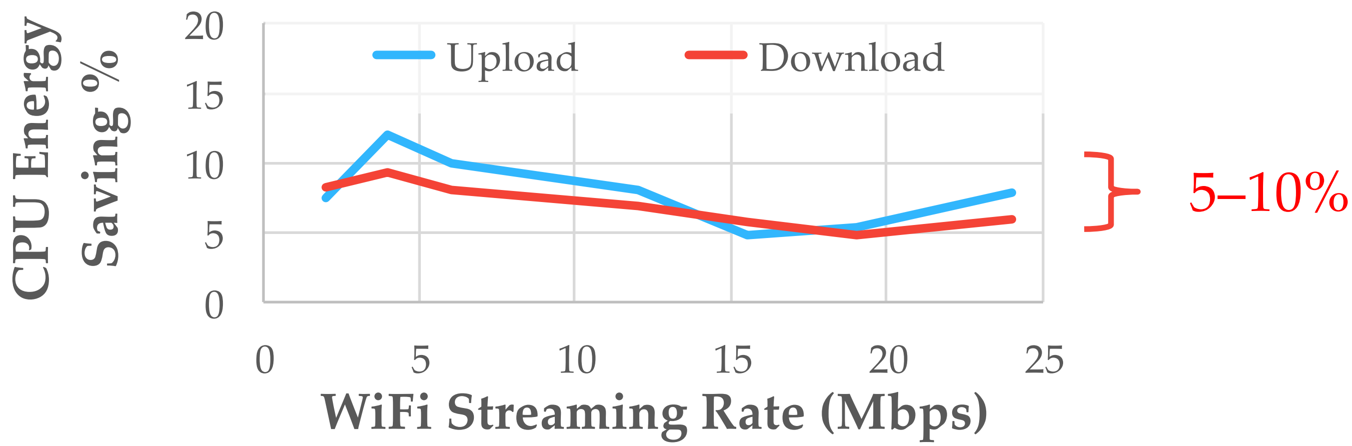


- Memory copies:
  kernel → user space → kernel
- Unnecessary because streaming apps do not modify data

- Zero-copy requires NIC's support (memory gather operation)
- Not available in today's smartphones
- **We use one-copy**

Energy Savings of One-Copy

- Metric: average CPU energy saving %
  - $\text{Savings} = 1 - \frac{\text{Energy}(CPU_{1-copy})}{\text{Energy}(CPU_{2-copy})}$



- Overall <10% savings (WiFi & LTE)
  - Throughput is the bottleneck, not memory copies
- Emulate high throughput via *loopback* interface
  - 30–40% savings at 150Mbps (Note) or 50Mbps (S3)

Outline

- Componentized energy analysis
  - Finding 1: CPU costs a lot in networking
  - Finding 2: network stack costs the most in CPU
- Reduce memory copies: one-copy
- Reduce TCP protocol processing: TCP offloading
- Conclusion

Method 2: Offload TCP to AP

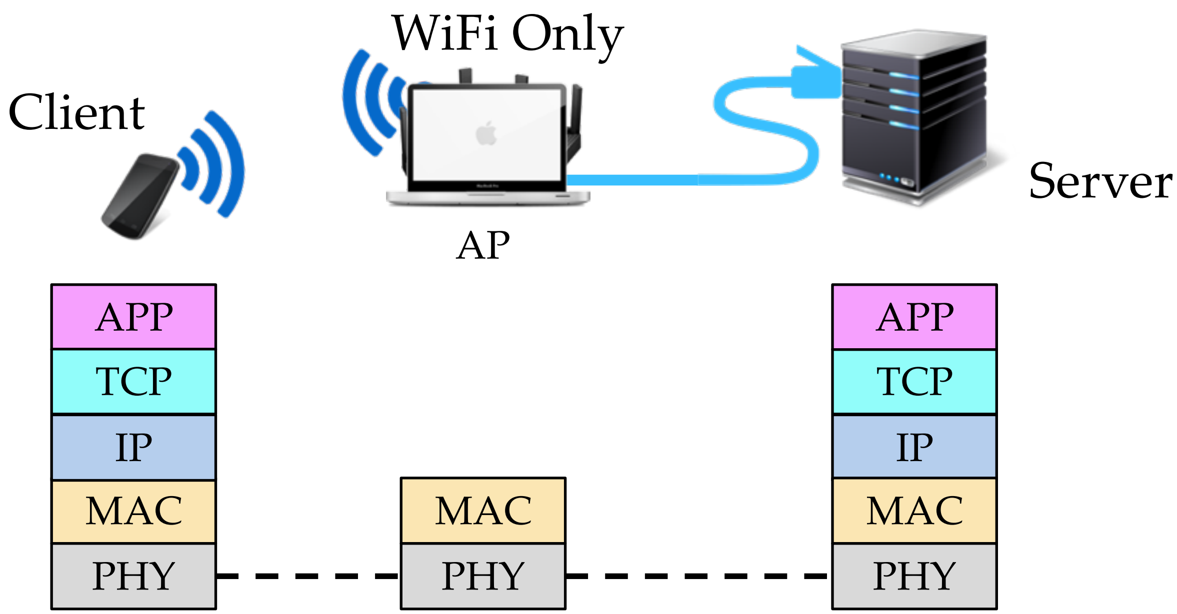
- TCP isn't energy efficient
- But we still want it
- Idea: move TCP processing to AP, i.e. TCP offloading
  - Applicable to *private* and *trusted* environments (*e.g.*, home, office)
  - If not, do not offload

✓

✓

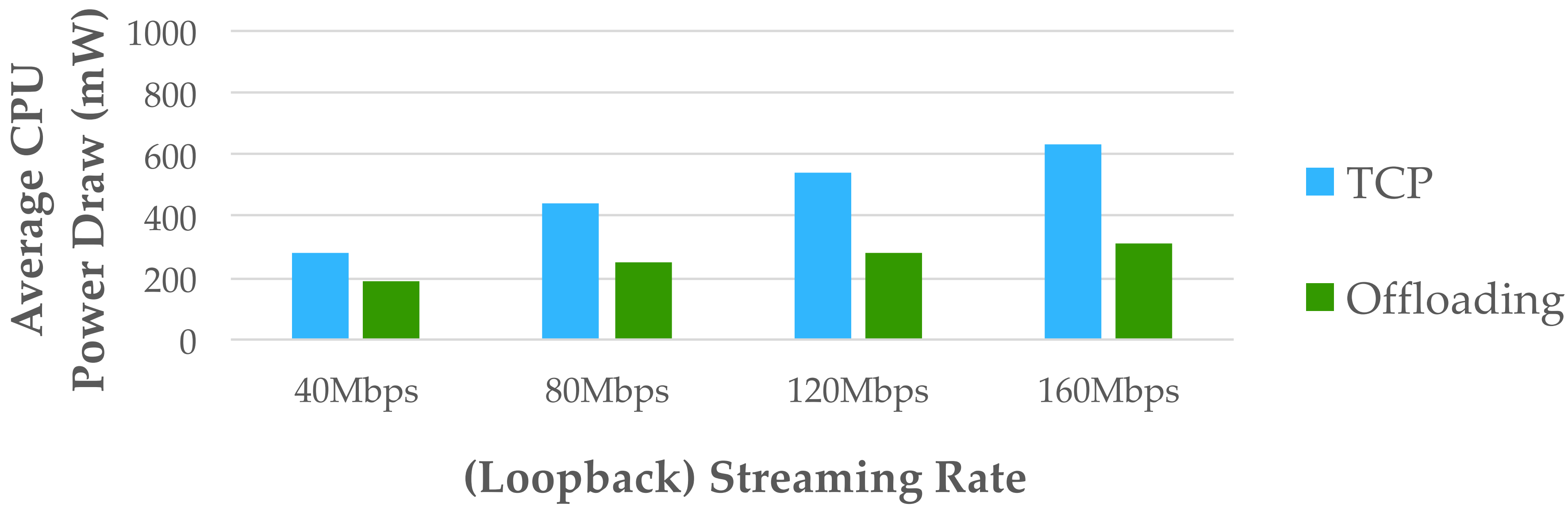
✗

TCP Offloading



- AP as a proxy
  - Handle TCP/IP stack processing
- Device w/ thin link layer
- Raw link-layer frames in last hop
  - Append a flow identifier in link-layer header

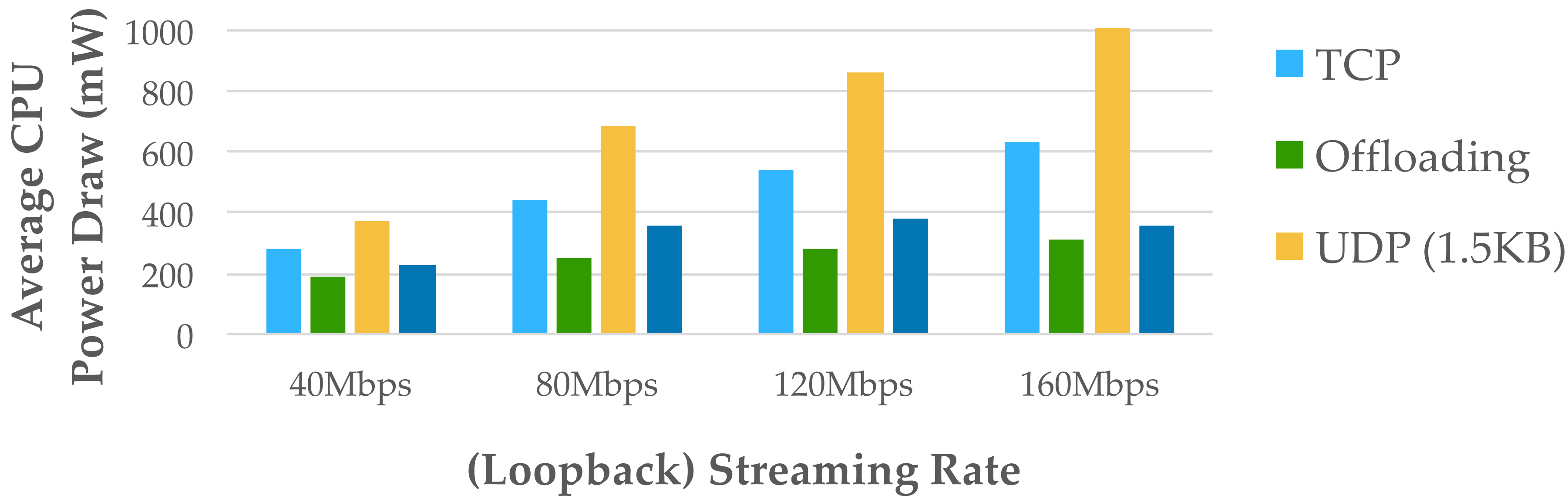
Energy Savings of TCP Offloading



- TCP consumes substantial CPU power
  - WiFi NIC costs between 200mW and 900mW†
- Offloading energy savings scale with throughput
- Up to 60% CPU energy (loopback), up to 40% (WiFi)

† Nika, A., and et al. "Energy and performance of smartphone radio bundling in outdoor environments." In *Proc. of WWW* (2015).

TCP Offloading vs UDP



- UDP worse than TCP
- Datagram keeps message boundary
  → Per-packet system call
  – Especially at high throughput

Conclusion

- Reducing CPU usage is important for energy-efficient networking
- One-copy is good at high throughput
  - What about zero-copy?
- Offloading outperforms TCP and UDP in energy cost
  - Practical deployment
  - Need private and trusted environment
  - Need reliable last-hop link

Thank you!

Questions?