



Ali  
Abedi



Tim  
Brecht



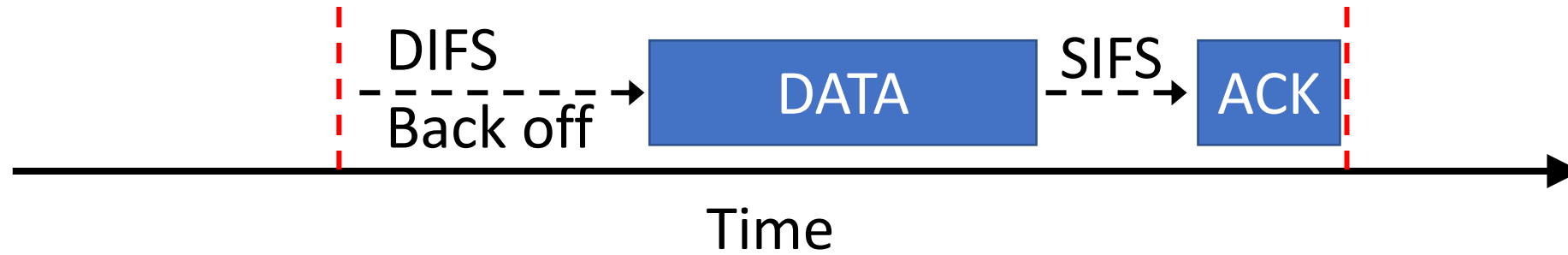
Omid  
Abari

# PNOFA: Practical, Near-Optimal Frame Aggregation for Modern 802.11 Networks

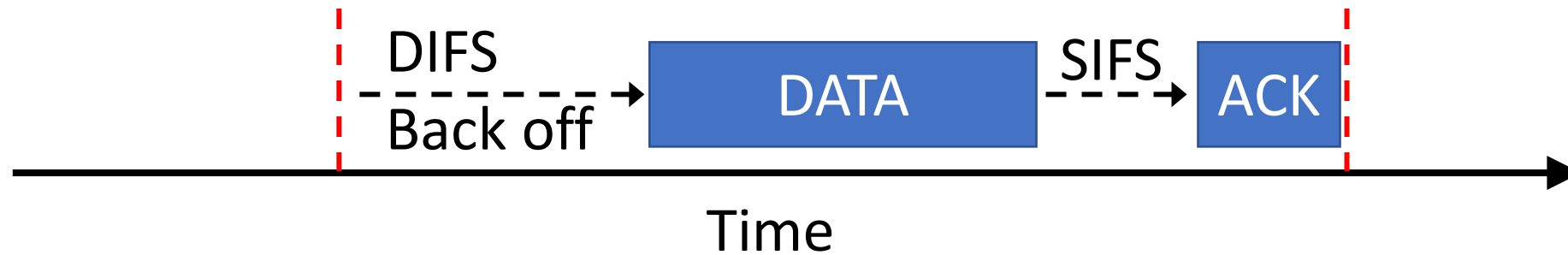


UNIVERSITY OF  
**WATERLOO**

# Background



# Background



Physical-layer bit rate:  $\infty$  bps  $\rightarrow$  MAC-layer throughput  $\approx$  50 Mbps

# Background

DATA ACK

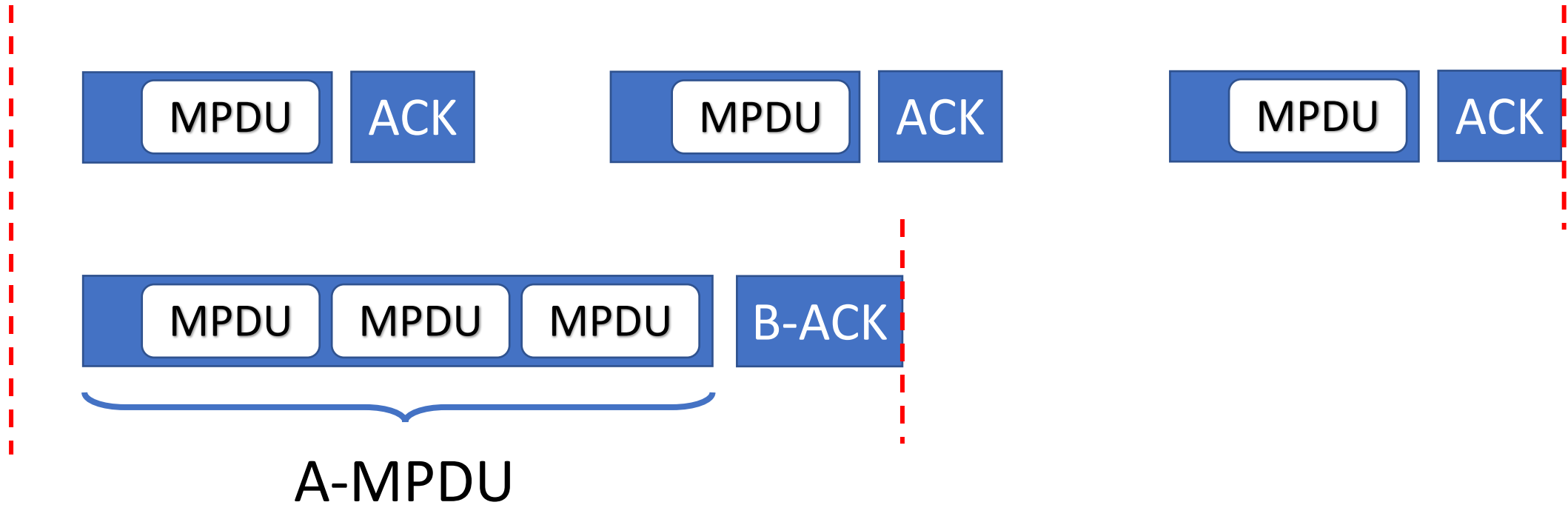
DATA ACK

DATA ACK

# Background

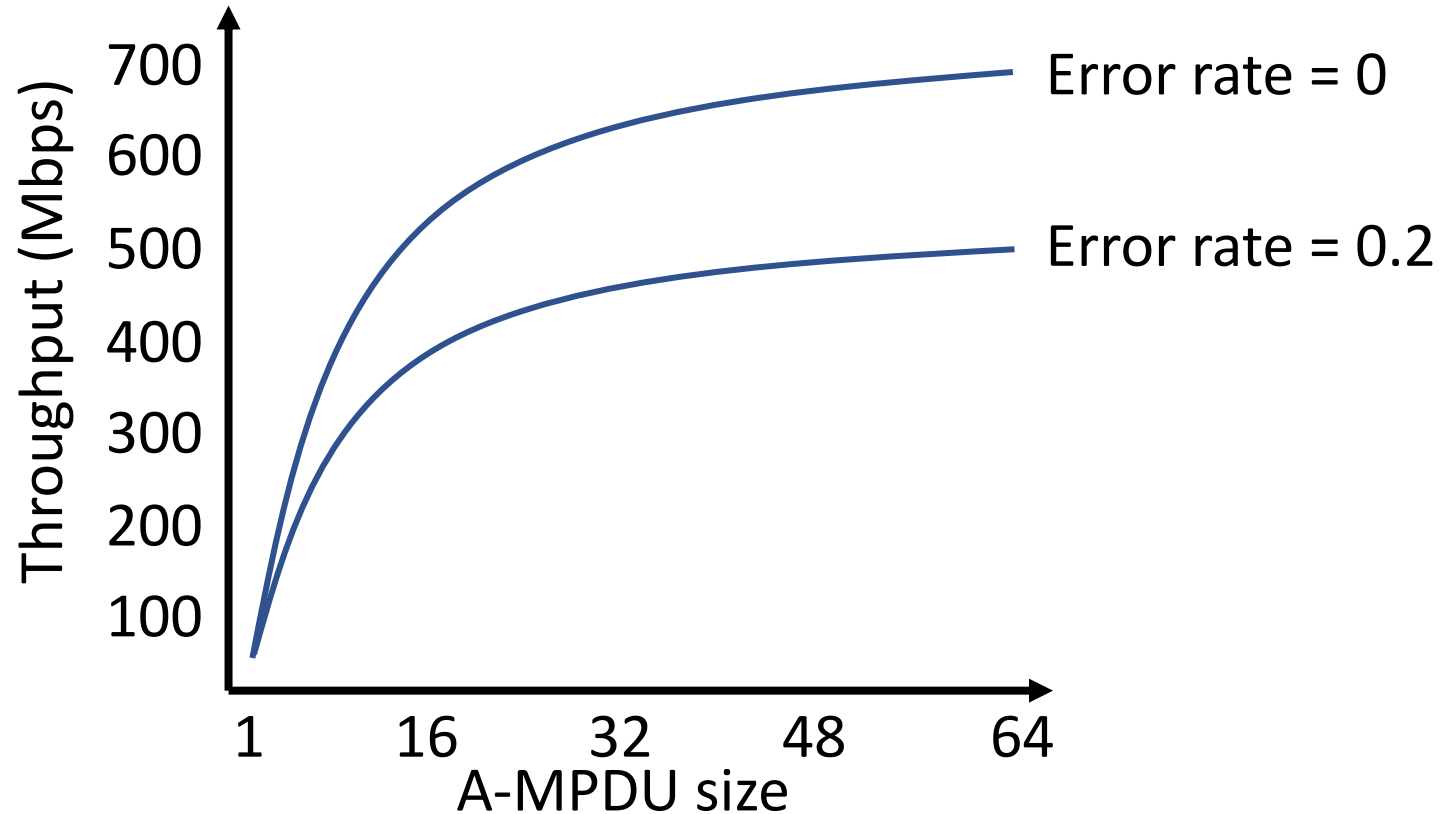


# Background



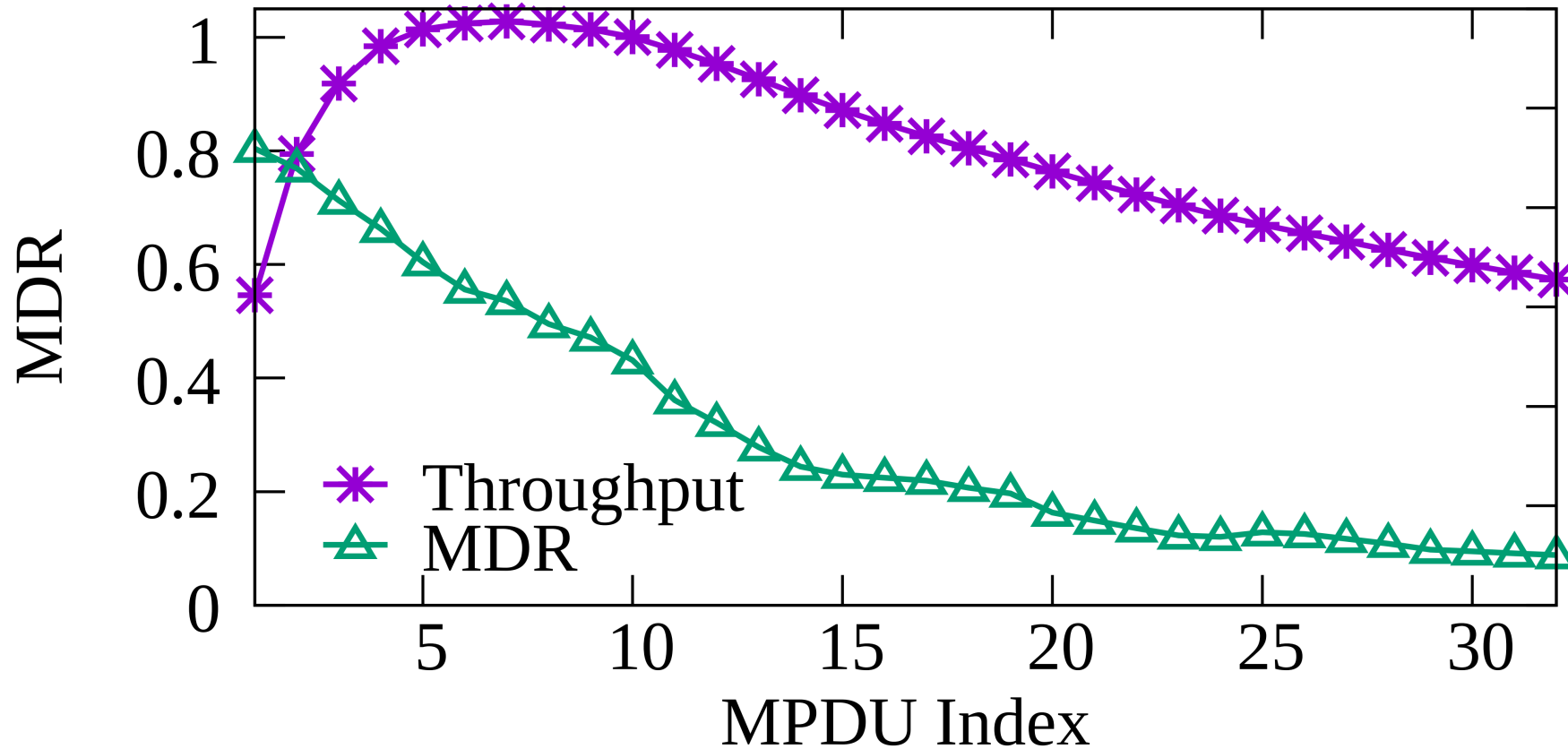
How many MPDUs should be aggregated?

# Throughput vs #MPDUs



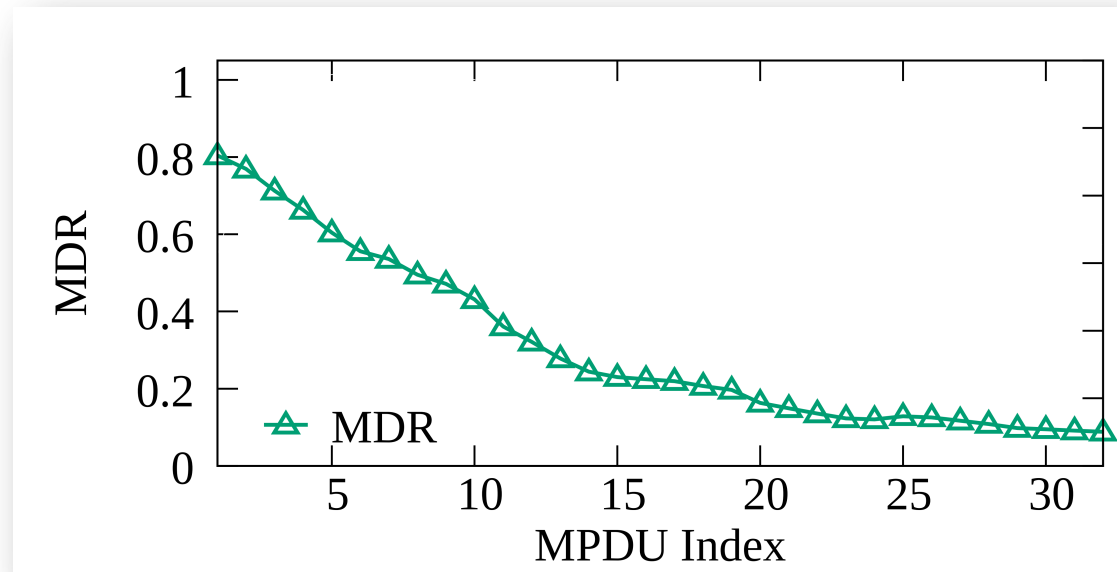
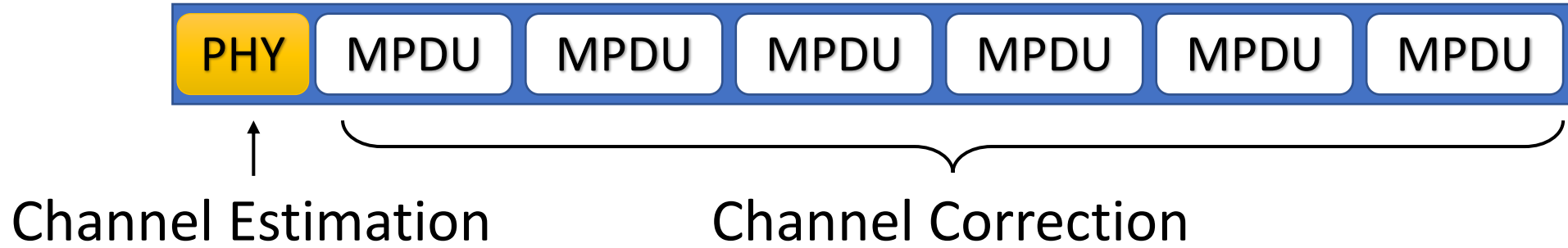
Should we aggregate as many as possible?

# Frame Aggregation is More Complicated!

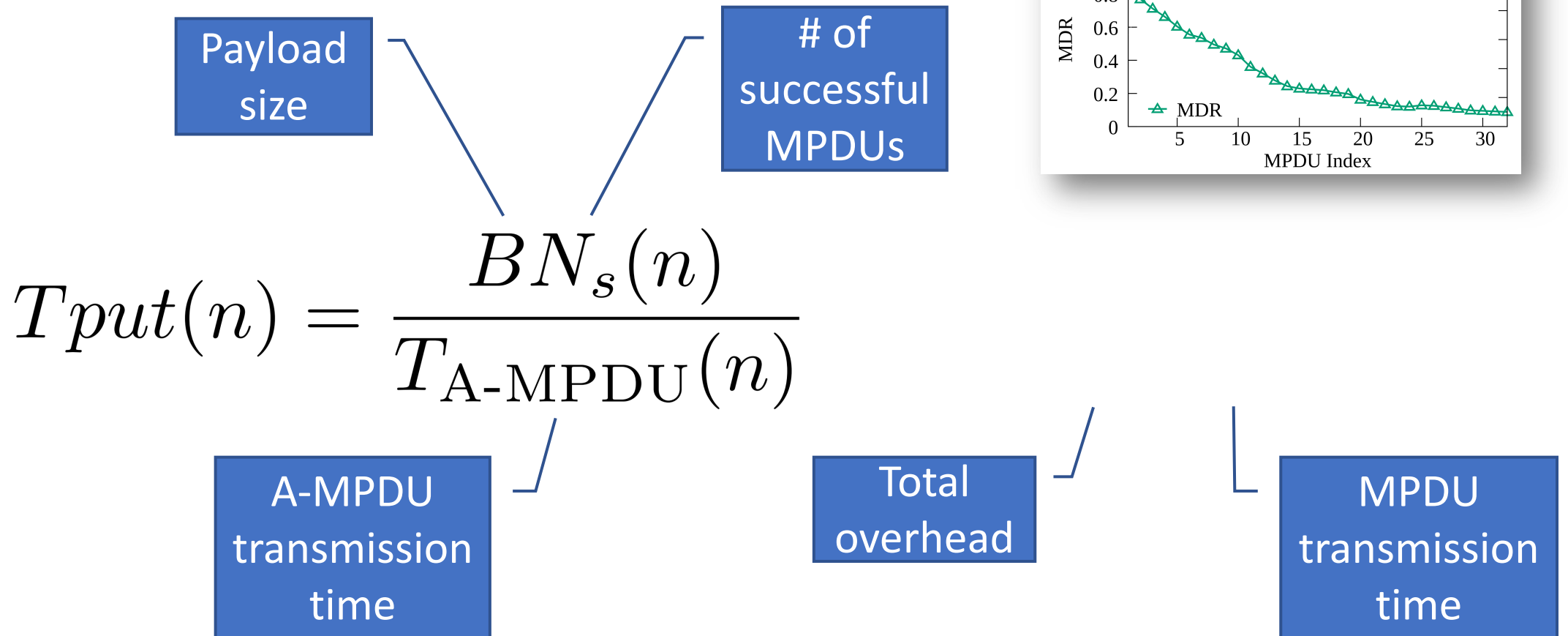




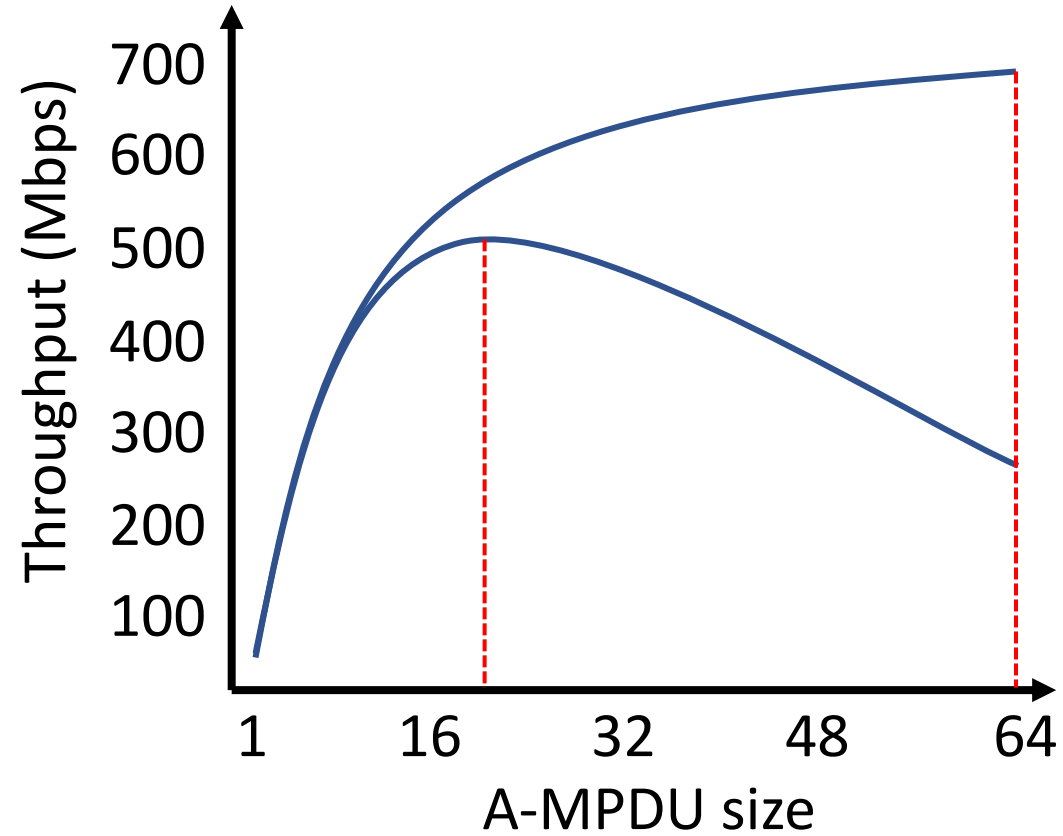
# Channel Correction Limitation



# Modeling Optimal A-MPDU Length

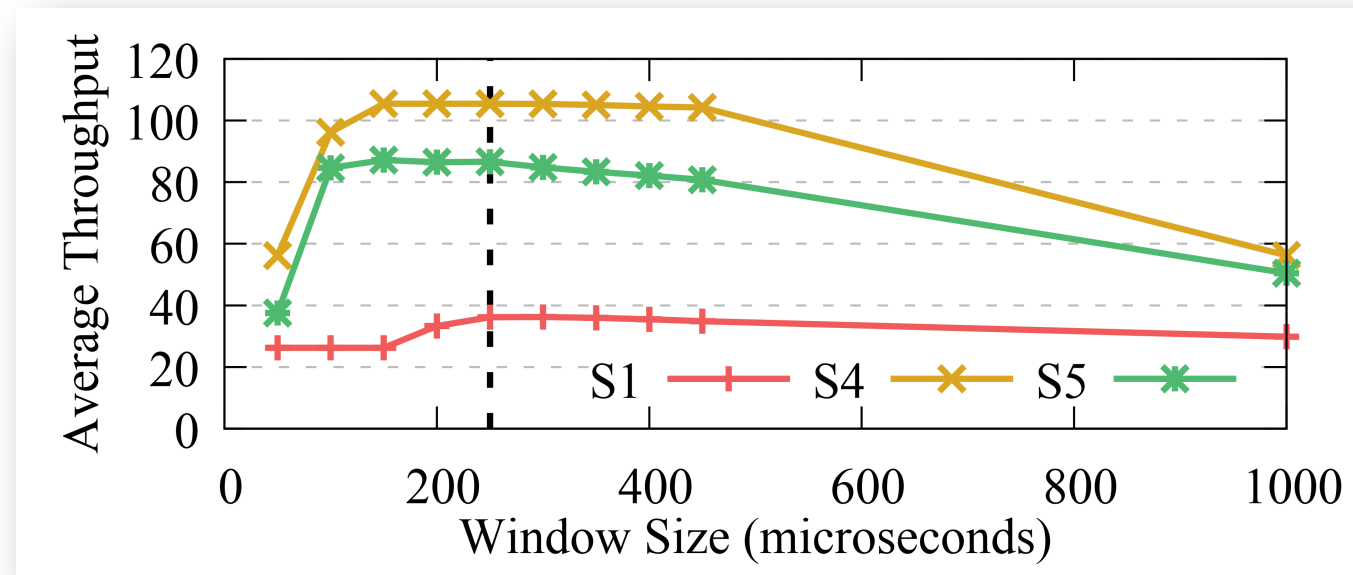
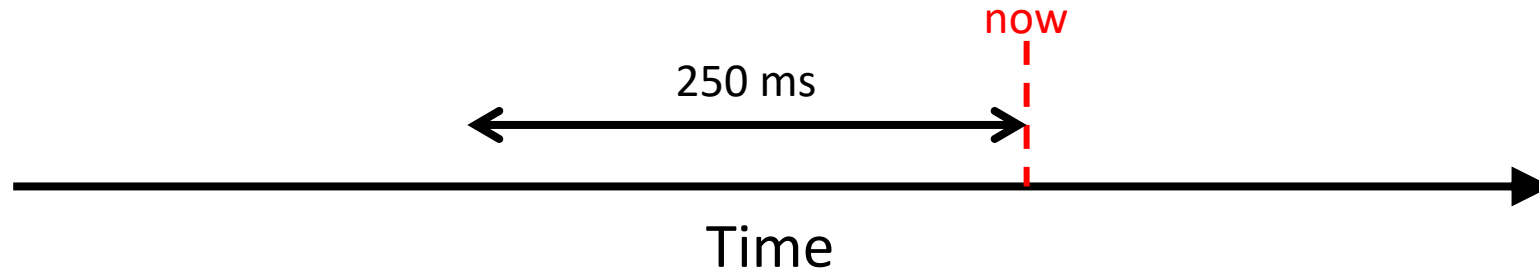


# Statistically Optimal Algorithm



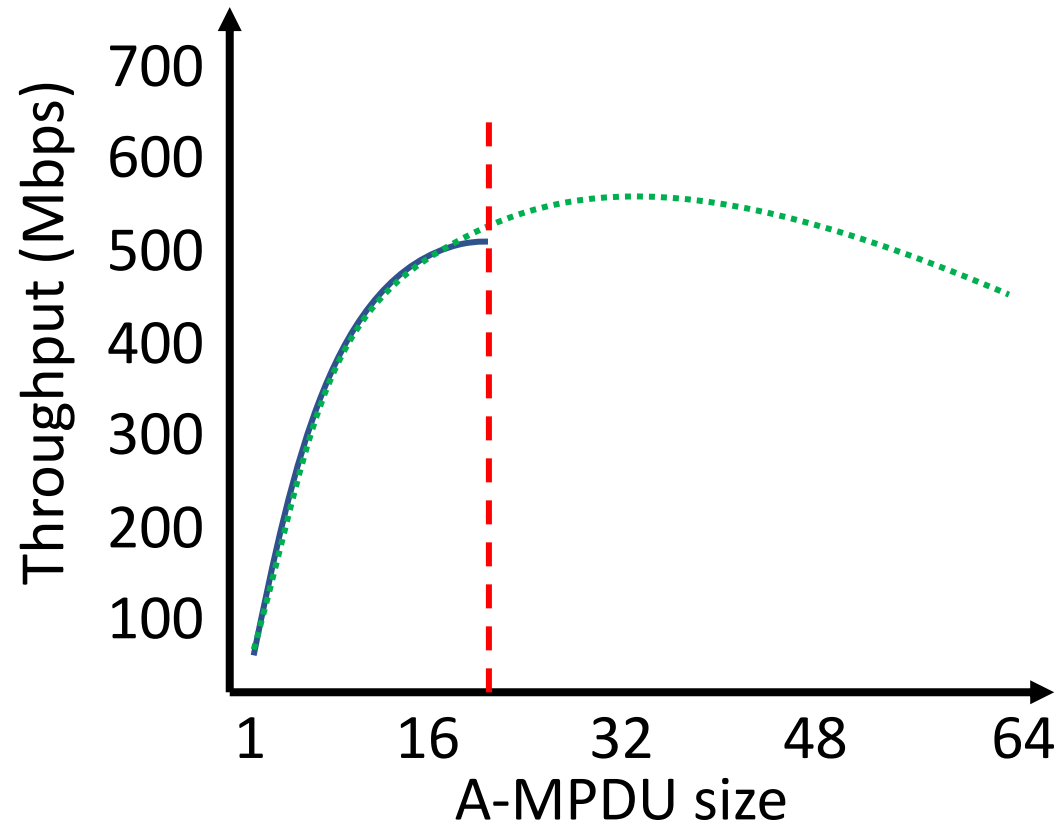
# Practical, Near-Optimal Frame Aggregation

- How does PNOFA estimate MDRs?



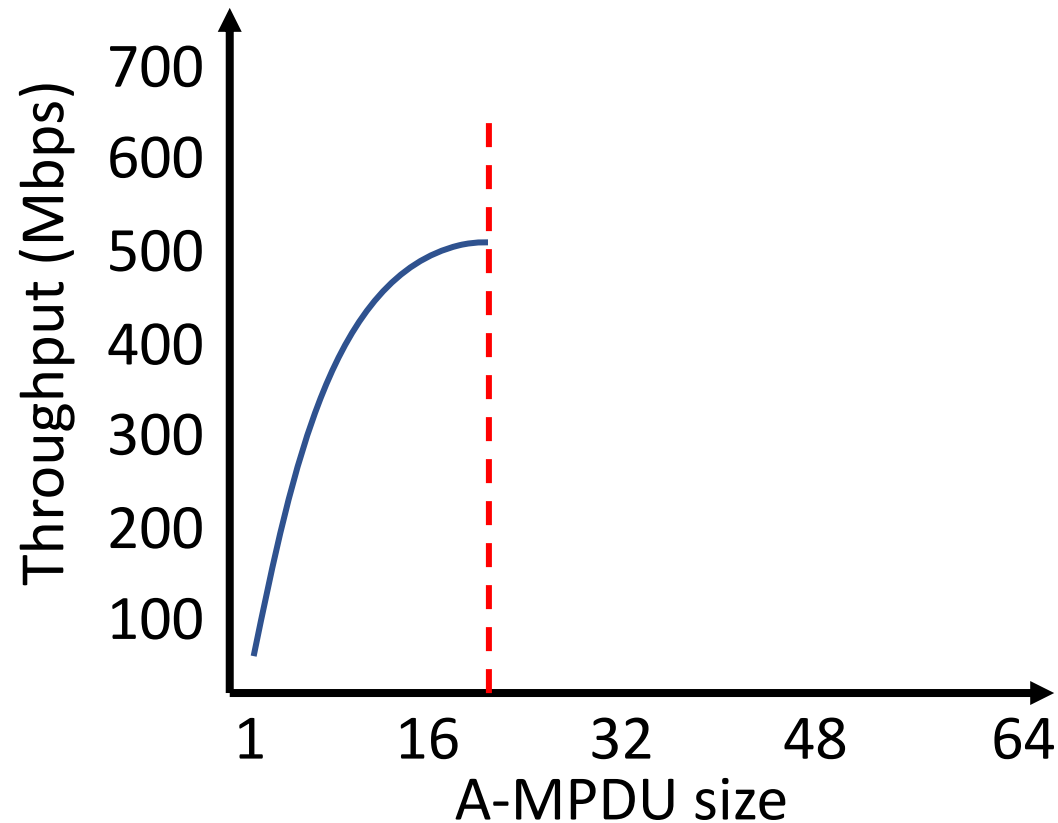
# Practical, Near-Optimal Frame Aggregation

- Operation of PNOFA



# Practical, Near-Optimal Frame Aggregation

- How to know when to increase the A-MPDU size?



# Evaluation of PNOFA

- Trace-based evaluation
  - Comparison with statistically optimal algorithm
  - Comparison with state-of-the-art algorithms
- Experimental evaluation
  - Implementation on Google Wifi APs

# Related Work

## MoFA

---

- Dynamically adjusts A-MPDU size
  - Compares the error rate in the first & seconds half of A-MPDU

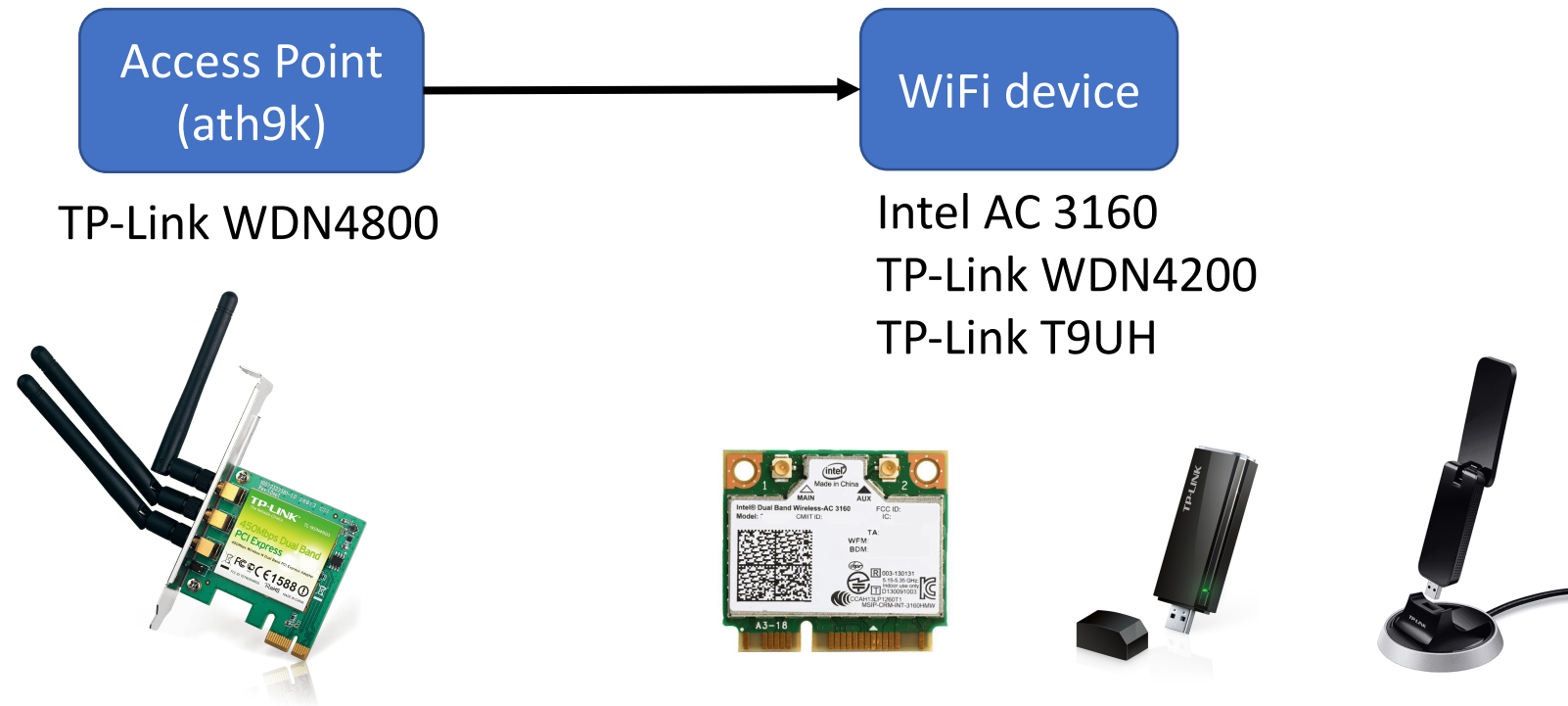
## STRALE

---

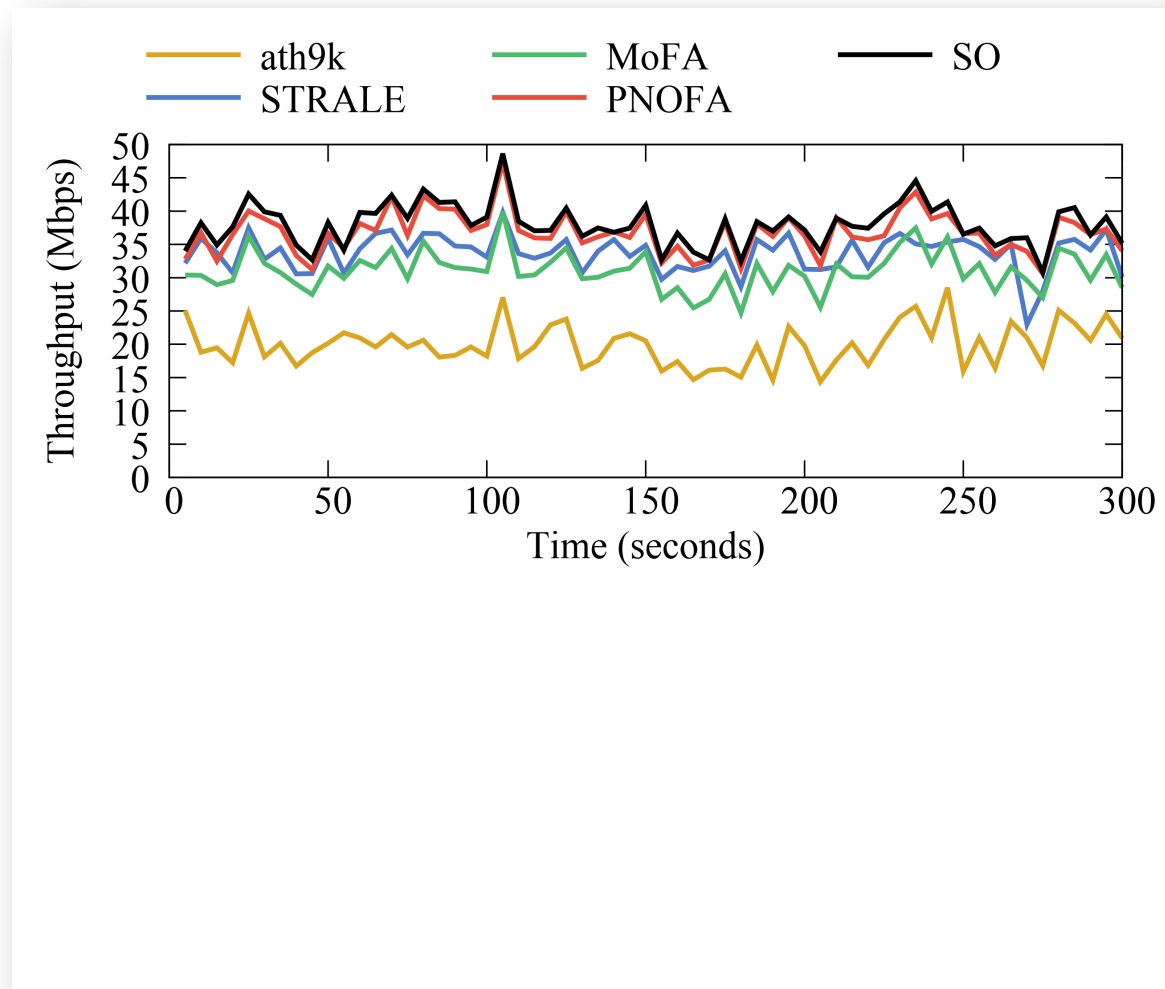
- Handles channel correction limitation by
  - Adjusting A-MPDU size similar to MoFA
  - Adjusting transmission rate

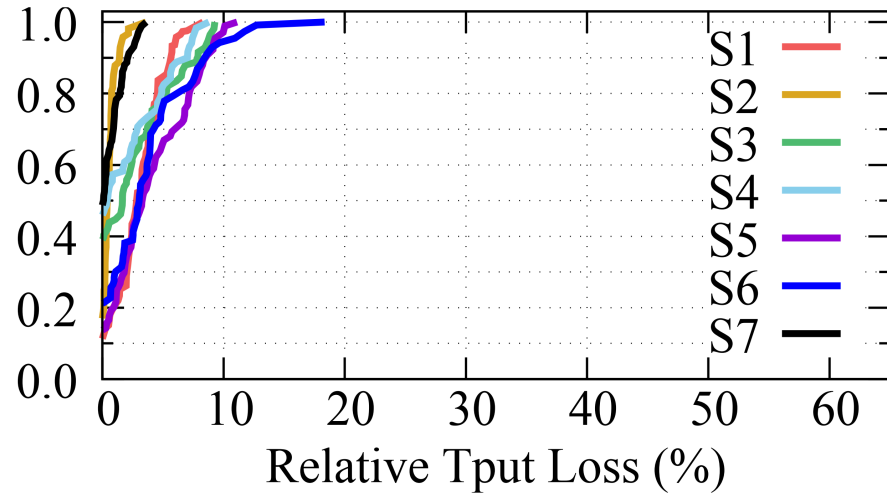


# Trace-Based Evaluation: Scenarios

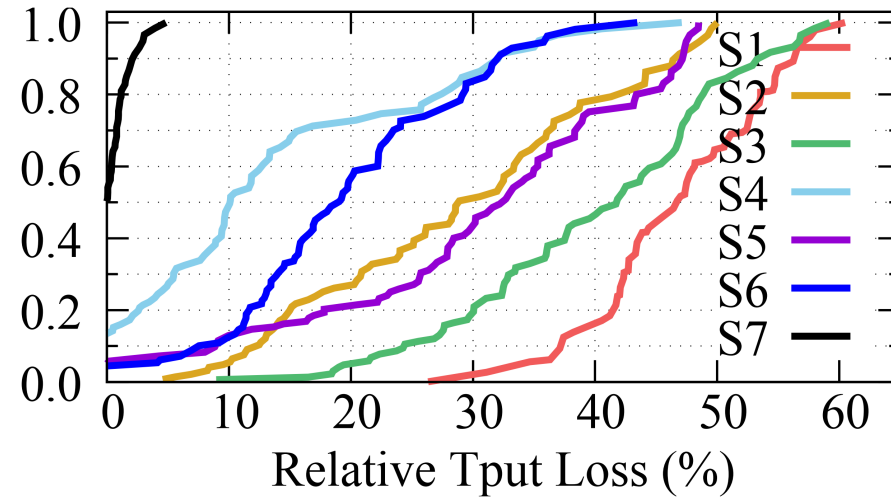


# Trace-Based Evaluation: Results

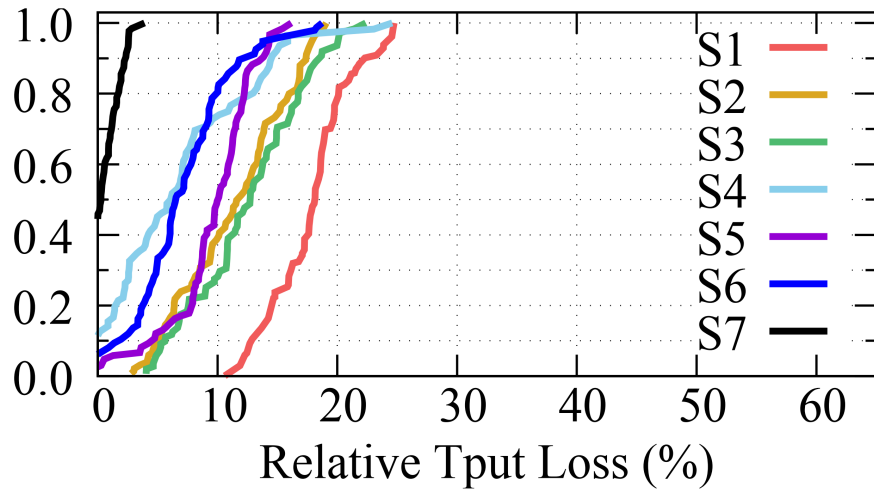




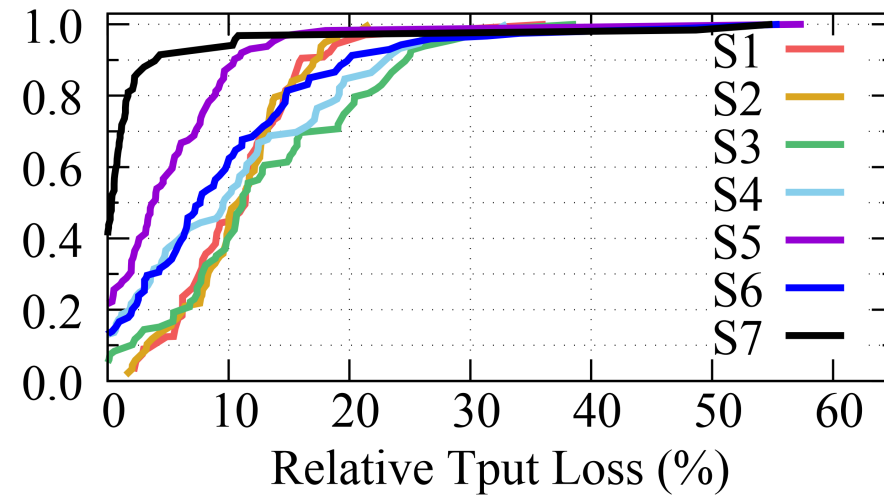
(a) PNOFA



(b) ath9k

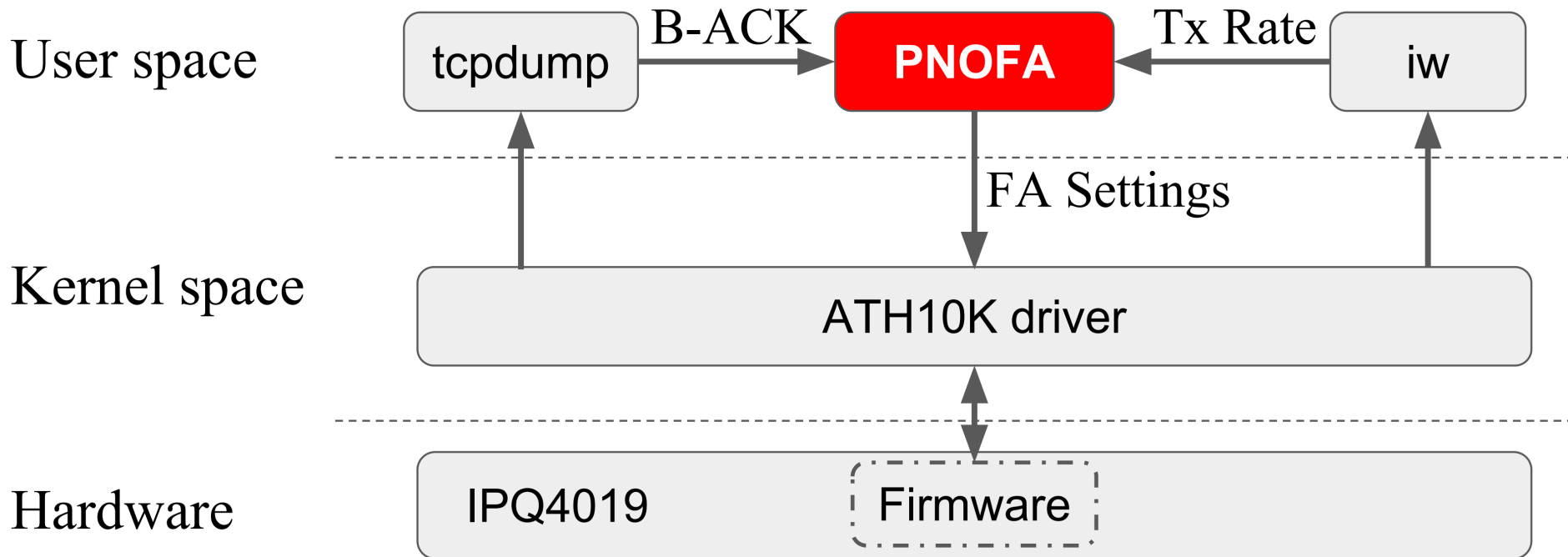


(c) MoFA



(d) STRALE

# Experimental Evaluation: Implementation



# Experimental Evaluation: Methodology

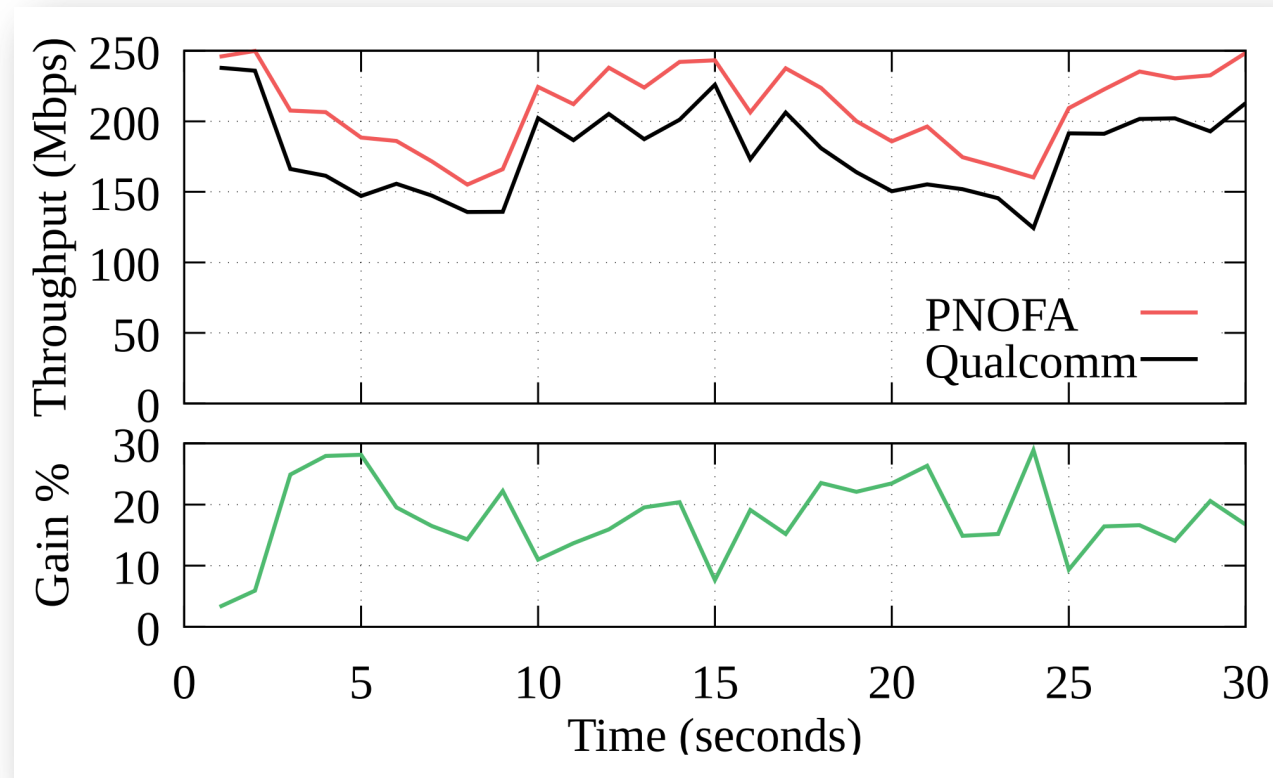


PNOFA



Moved at walking speed

# Experimental Evaluation: UDP Results



17% average improvement

# Conclusions

- PNOFA
  - Near optimal performance
  - Practical (implemented on Google Wifi using tcpdump)
  - Outperforms state-of-the-art algorithms