Microsoft Research 微软亚洲研究院

Improving Spectrum Efficiency with μACKs

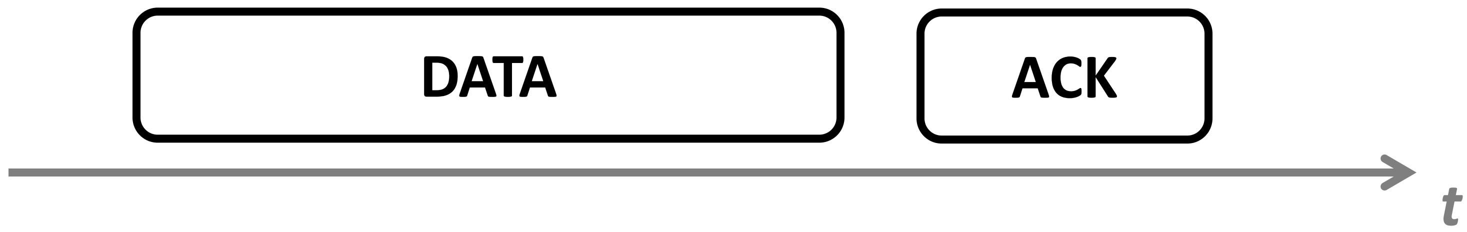
Jiansong Zhang[†#], Haichen Shen[†], Kun Tan[†], Ranveer Chandra[*], Yongguang Zhang[†] and Qian Zhang[#]

[†]Microsoft Research Asia

[*]Microsoft Research Redmond

[#]HKUST

Feedback in Wireless Networks

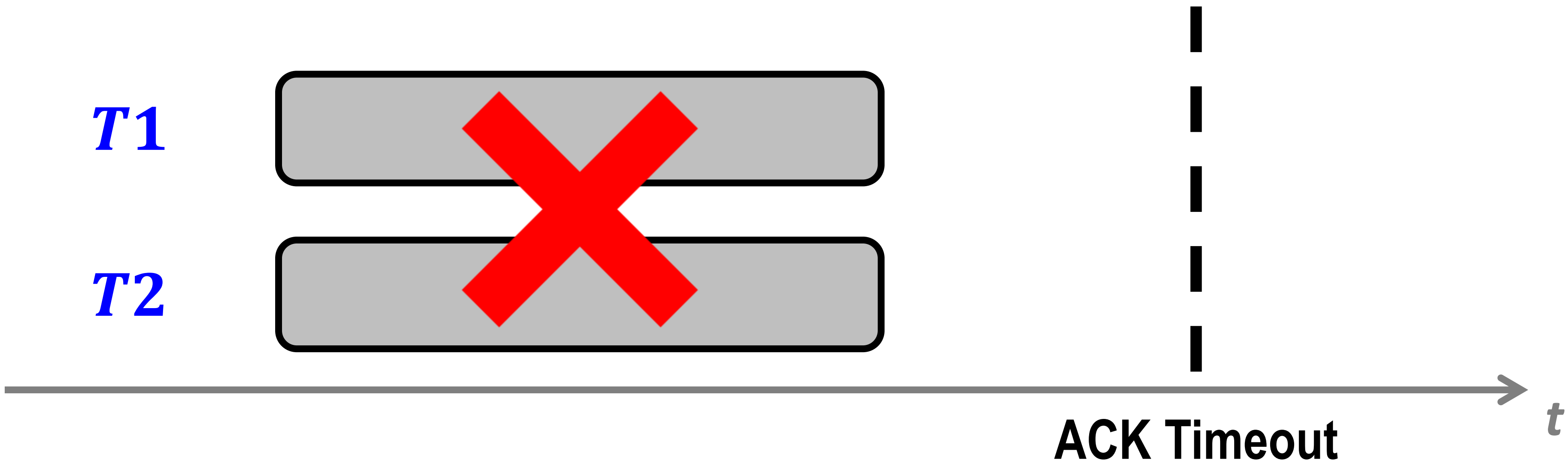


- Feedback is critical for network protocols
  - Confirm reception / detect loss (i.e. ACKs)

- Current network protocols are primarily based on frame level feedback

Frame-level Feedback Considered Harmful in Wireless

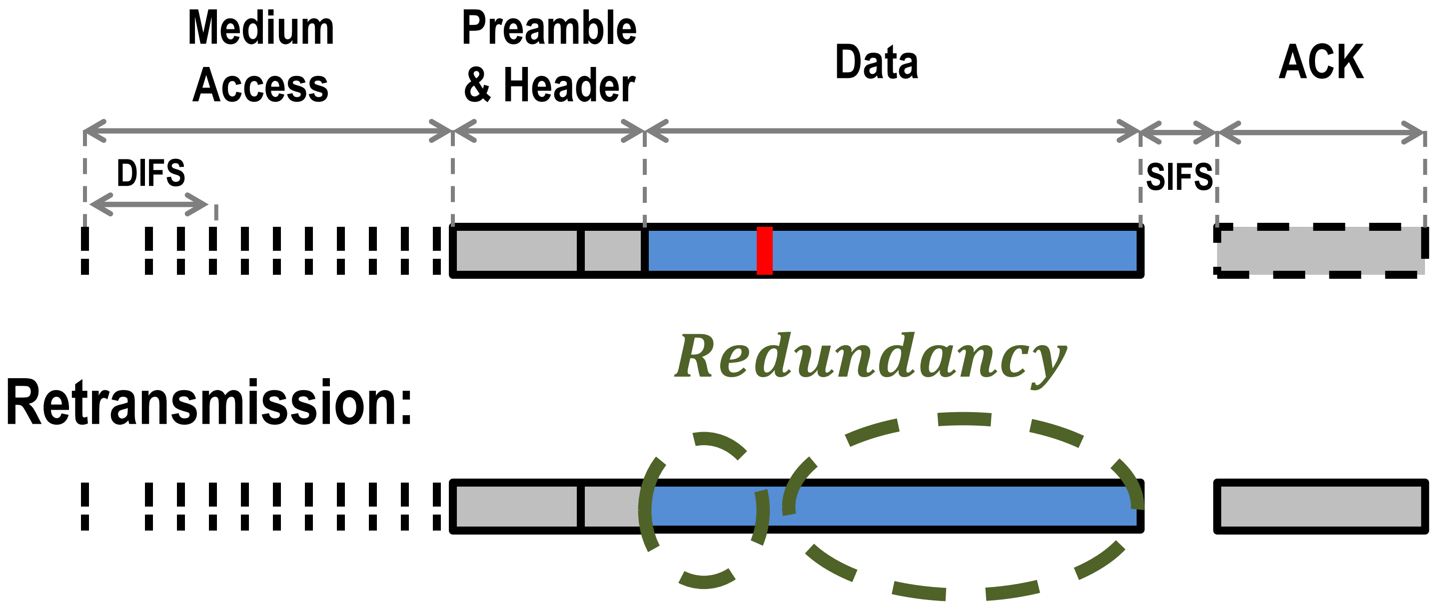
Example 1: Collision detection based on ACK



- May be too late
  - Feedback received after all damage has been done

Frame-level Feedback Considered Harmful in Wireless

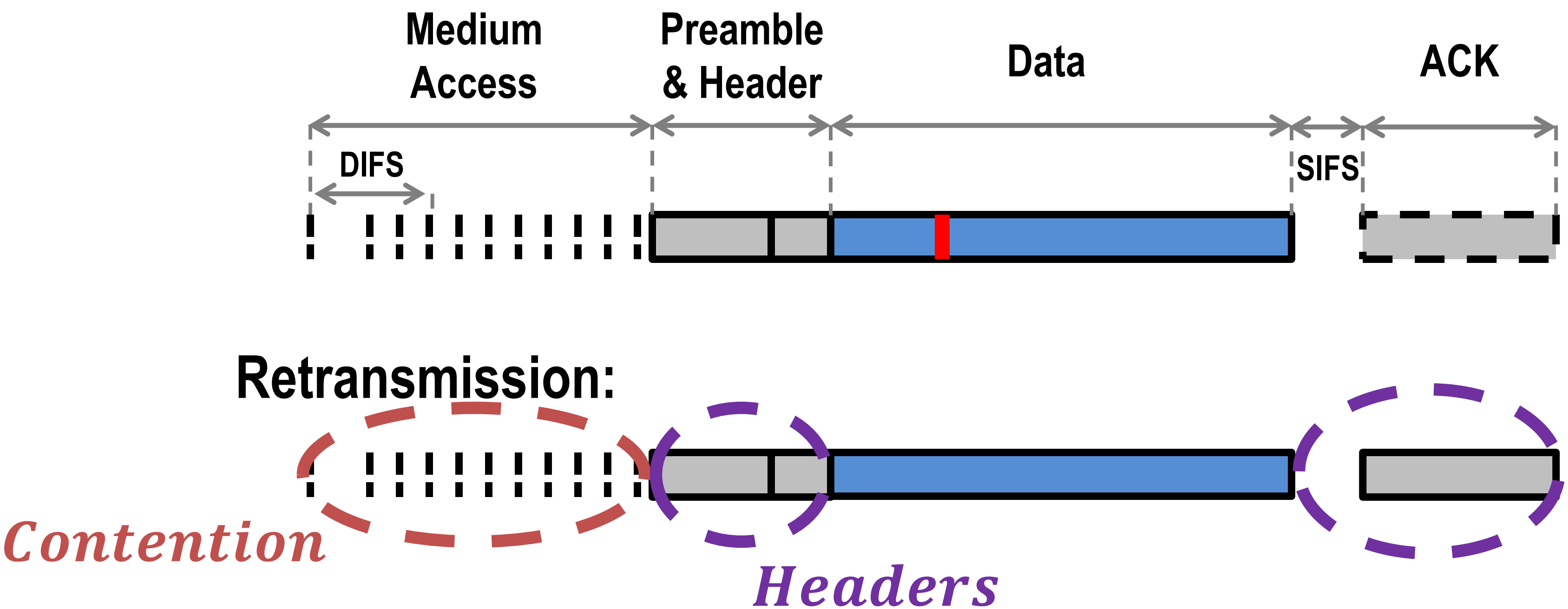
Example 2: Frame retransmission is inefficient



- May contain limited information

Frame-level Feedback Considered Harmful in Wireless

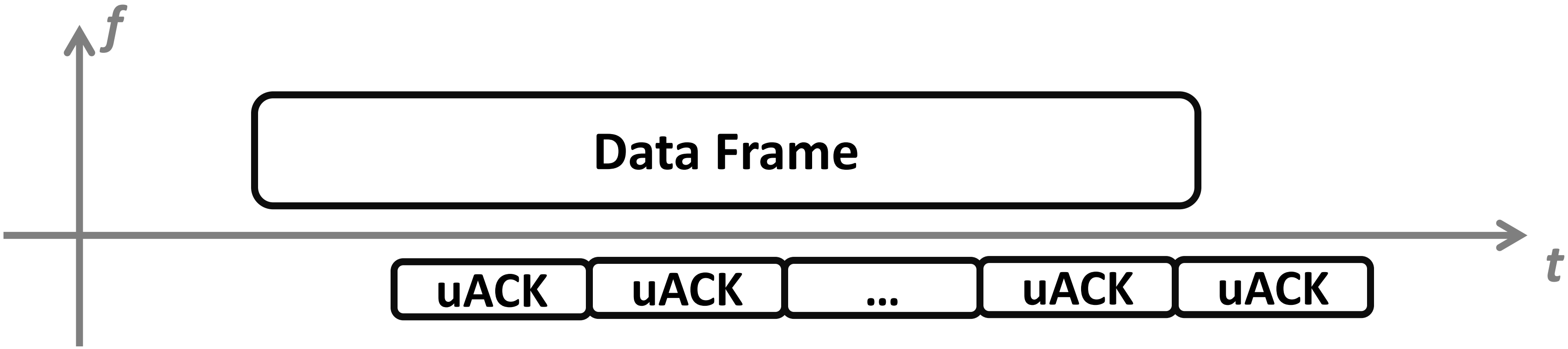
Example 2: Frame retransmission is inefficient



- May contain limited information
- May be costly to re-establish transmission context

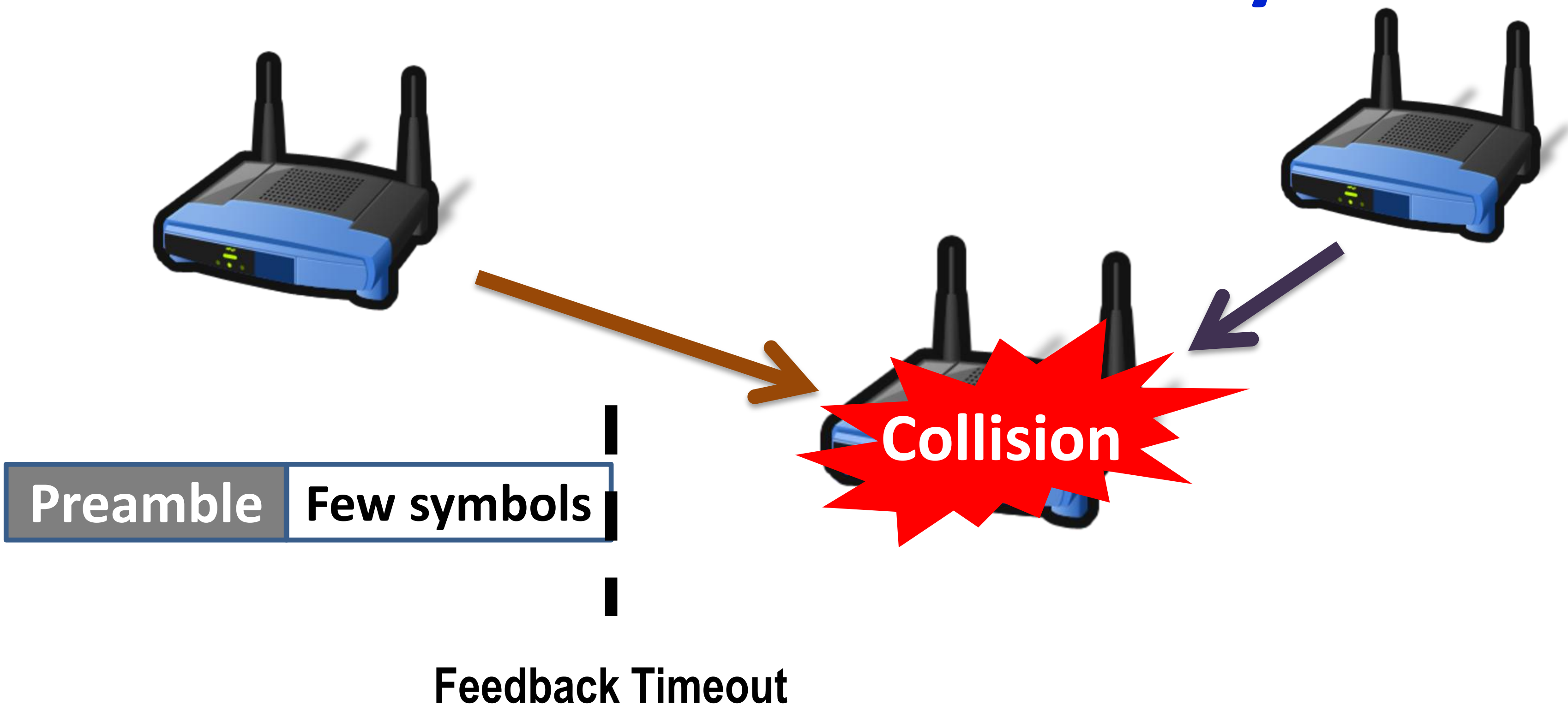
We should do symbol level feedback

µACK Towards Symbol-level Feedback



- Two Tightly synchronized radio chains
  - ❑ Wide-band forward channel
  - ❑ Narrow-band feedback channel
- Tiny acknowledgement symbols

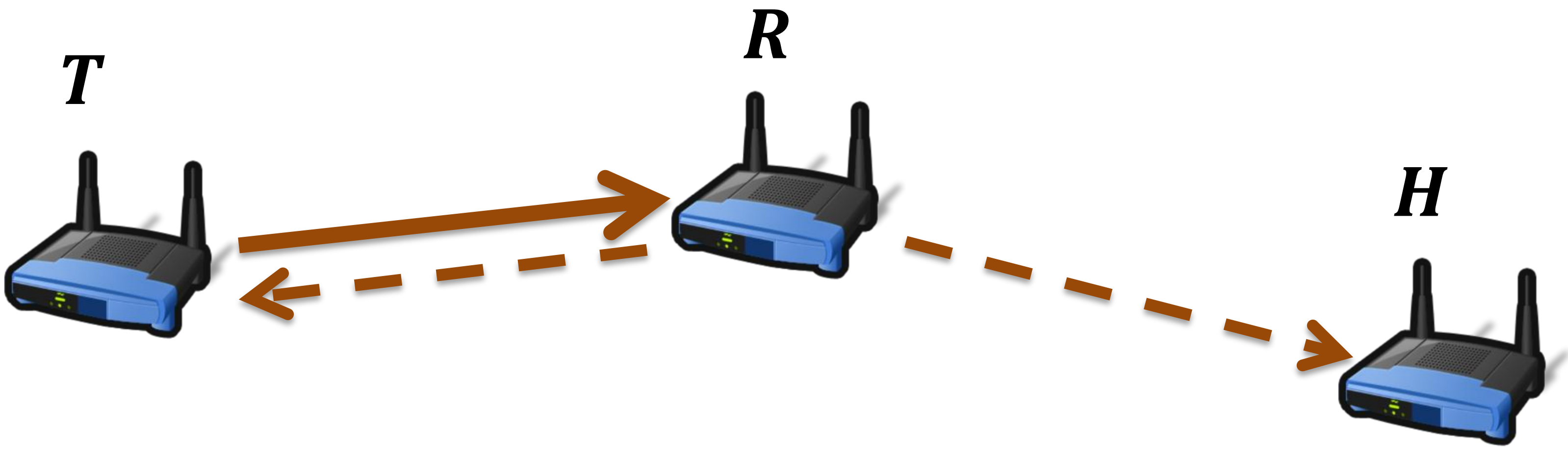
µACK Application 1 – Collision Detection and Early Backoff



- Early collision detection by feedback timeout

µACK Application 2 – Hidden & Exposed Terminal Mitigation

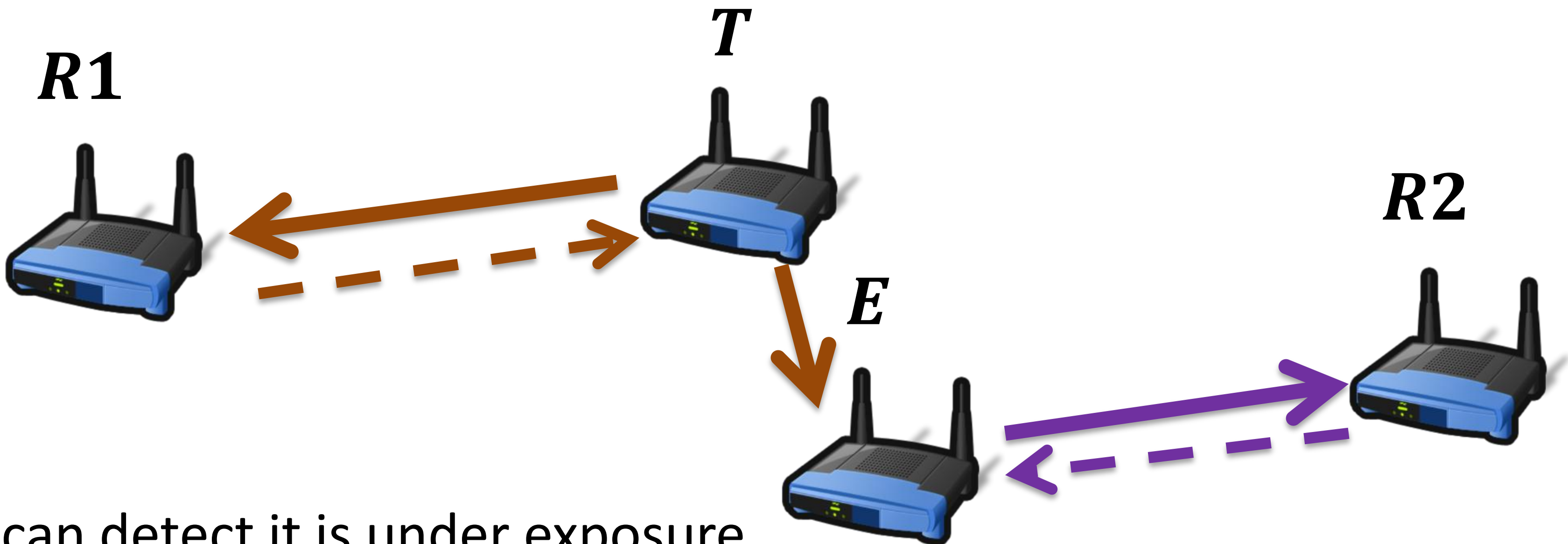
Hidden Terminal:



μACK from R prevents H from colliding

µACK Application 2 – Hidden & Exposed Terminal Mitigation

Exposed Terminal:

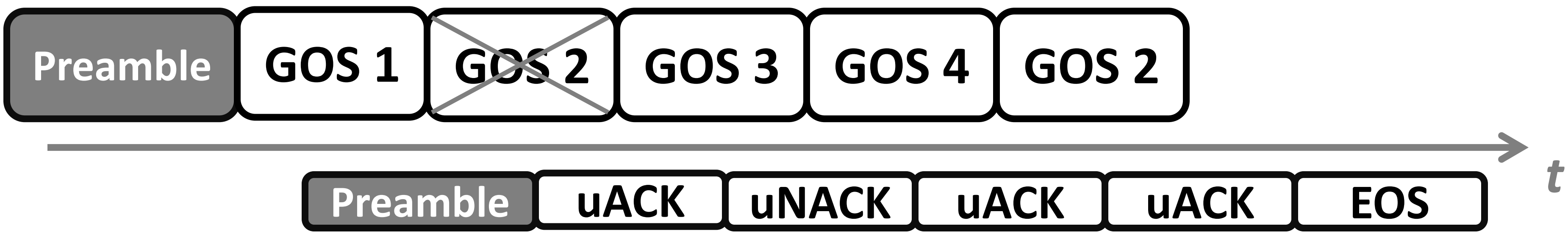


E can detect it is under exposure

- μACK is an extended busy tone

µACK Application 3 – In Frame Retransmission

GOS: group of symbols **EOS:** end of stream



- Retransmission appends to original frame

μACK Benefits Wireless in Various Ways

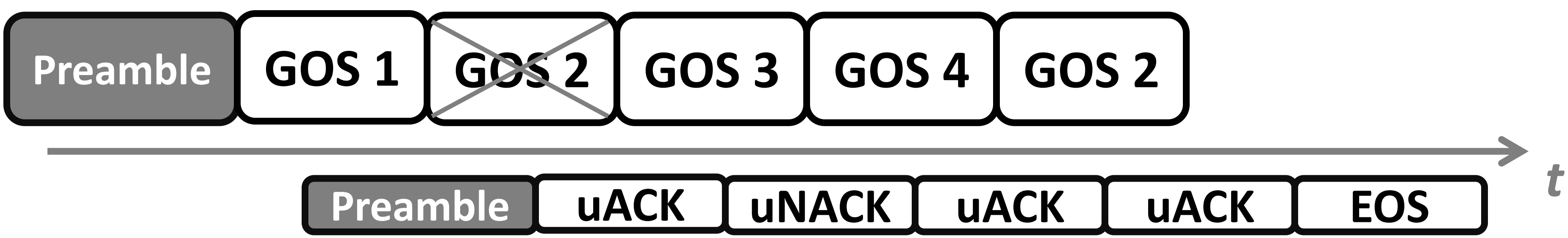
- Application 1:
  - ❑ Collision Detection and Early Backoff
- Application 2 (extended):
  - ❑ Hidden & Exposed Terminal Mitigation
- Application 3:
  - ❑ In-frame Retransmission

μACK Benefits Wireless in Various Ways

- Application 1:
  - ❑ Collision Detection and Early Backoff
- Application 2 (extended):
  - ❑ Hidden & Exposed Terminal Mitigation
- Application 3:
  - ❑ In-frame Retransmission

In-frame Retransmission Details

- Design questions
  - ❑ What is the symbol group size?
  - ❑ What is μACK physical layer?
  - ❑ How to determines a group of symbol is correct?



GOS: group of symbols **EOS: end of stream**

Data Symbol Group Size

- Symbols in a group are fate-sharing
  - GOS length **<** coherent time of the channel
- Tradeoff between redundant bits and feedback channel requirement
  - Larger GOS → more redundant bits, and less feedback bandwidth
- Design choice
  - 20μs GOS → 5 OFDM symbols
  - 1MHz feedback channel ~ 5% for 20MHz data channel

µACK PHY

- Simple spectrum spreading PHY
  - ❑ Feedback symbol time is 20μs (the length of GOS)
  - ❑ Four bits per symbol (encode 3 states)
  - ❑ Channel width is 1MHz (50% guard band) → Bandwidth 500KHz → Chip rate is 500Kcps
  - ❑ Ten chips per symbol

Symbol name	Symbol binary ($b_3b_2b_1b_0$)	Chip values
ACK	1100	0111100010
NACK	1001	0011001101
EOS	0110	1100110110

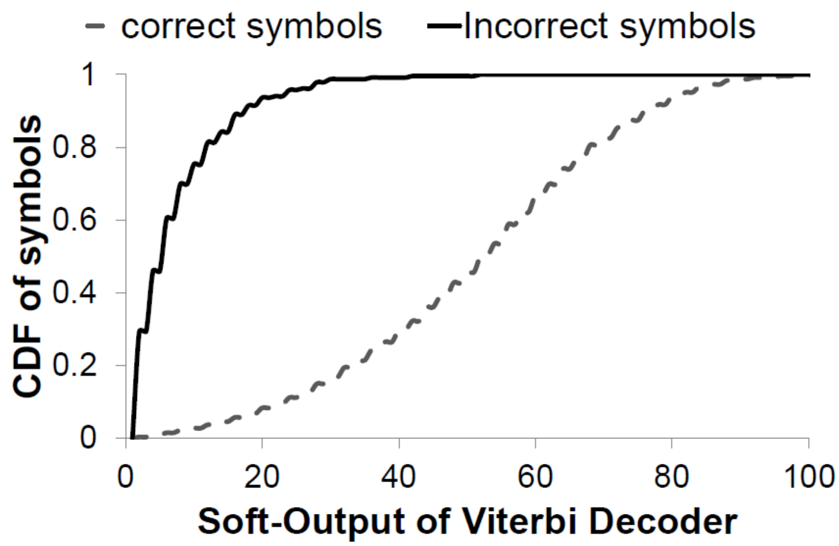
Error Detection

- Two methods
  - Segment CRC (additional overhead)
  - PHY hints

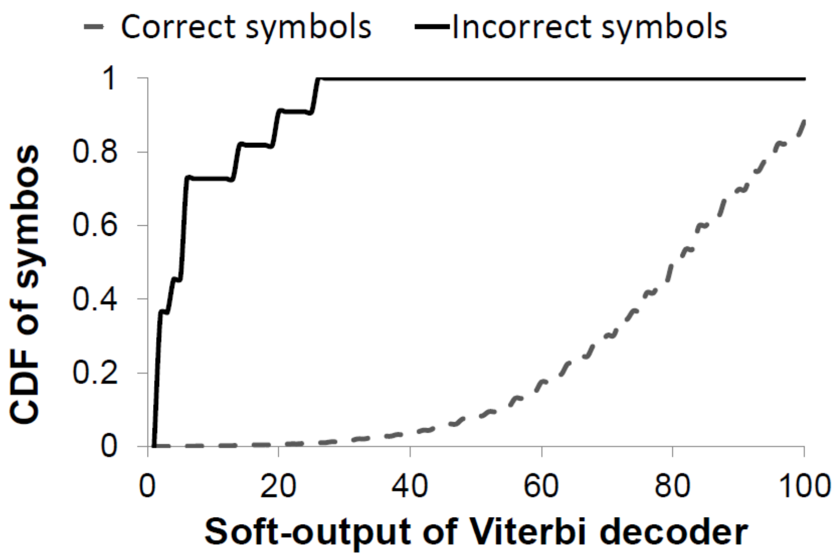
We found PHY hints becomes less reliable in some cases …

PHY hints become unreliable on marginal SNR

24Mbps, 10dB (marginal)

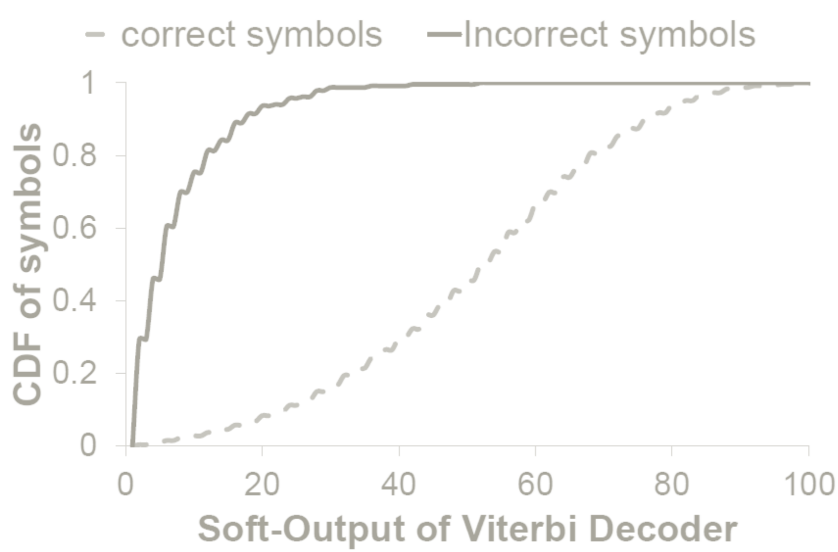


24Mbps, 12dB (higher)

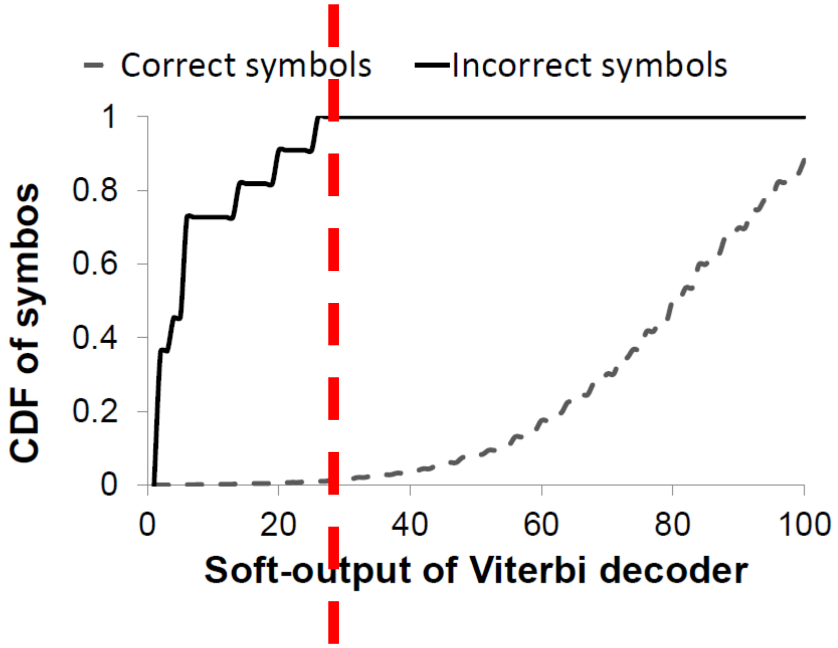


PHY hints become unreliable on marginal SNR

24Mbps, 10dB (marginal)

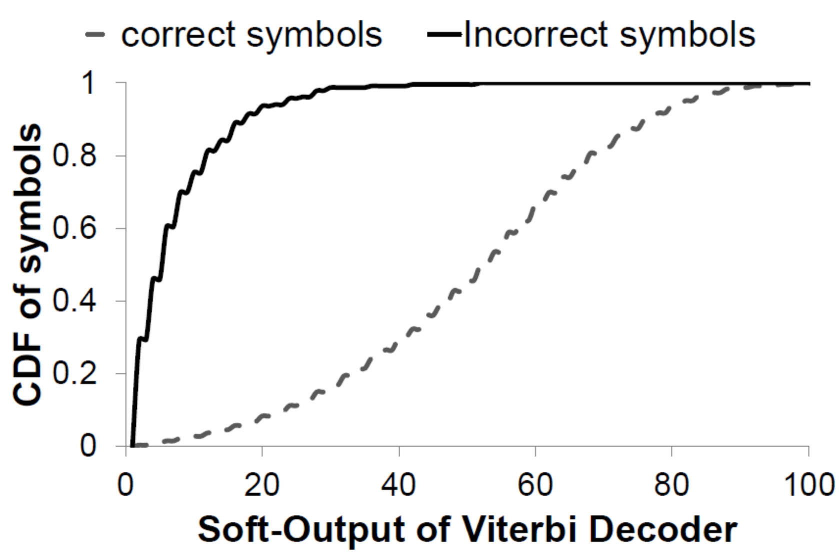


24Mbps, 12dB (higher)

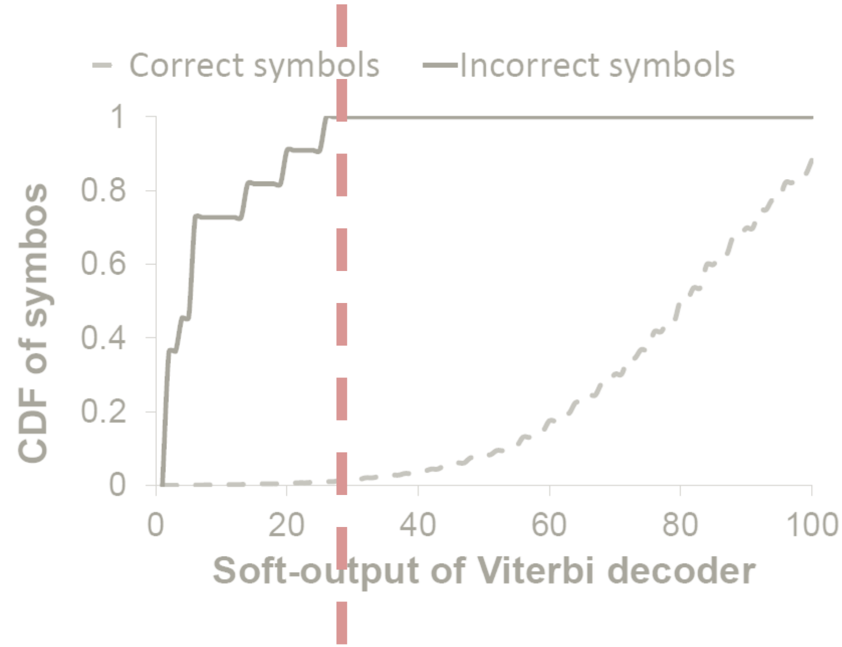


PHY hints become unreliable on marginal SNR

24Mbps, 10dB (marginal)

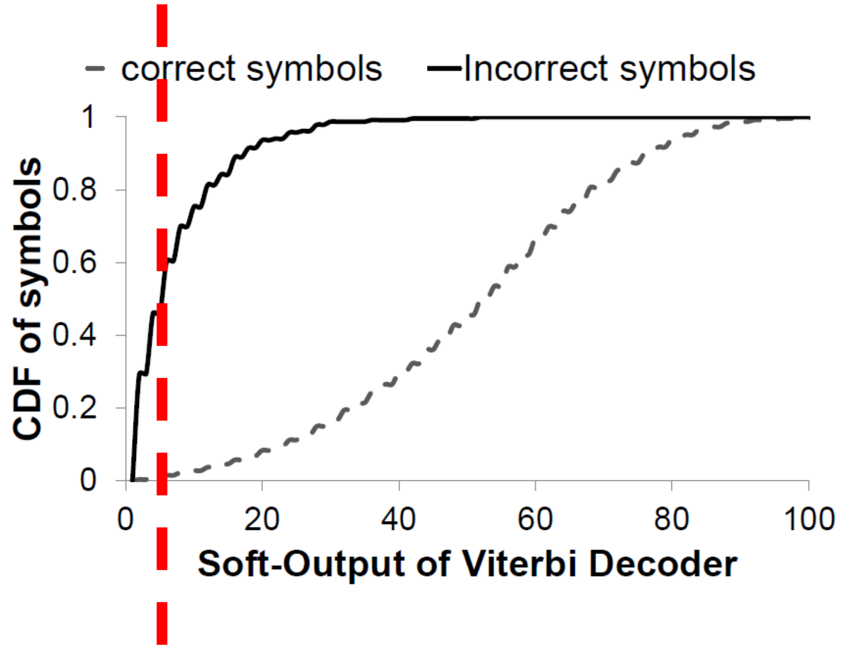


24Mbps, 12dB (higher)



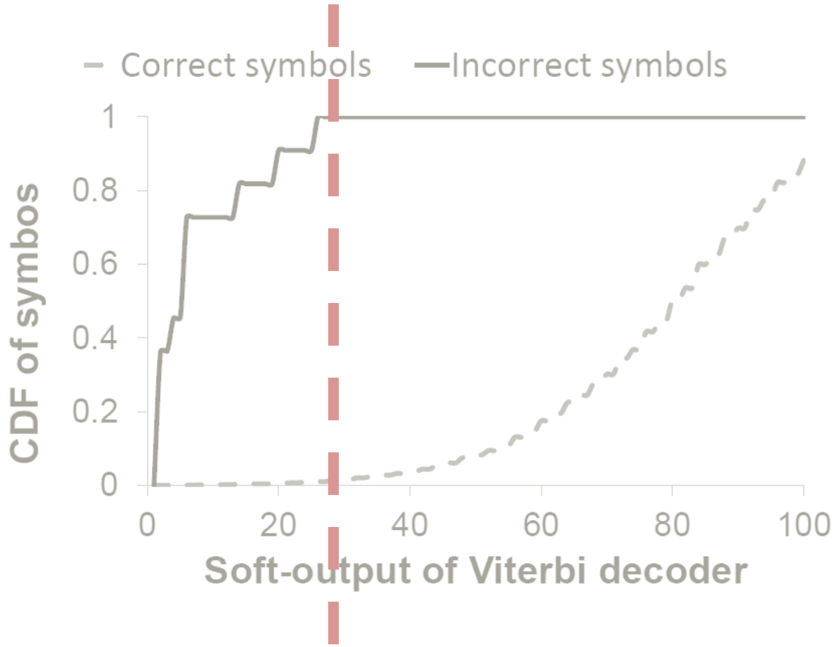
PHY hints become unreliable on marginal SNR

24Mbps, 10dB (marginal)



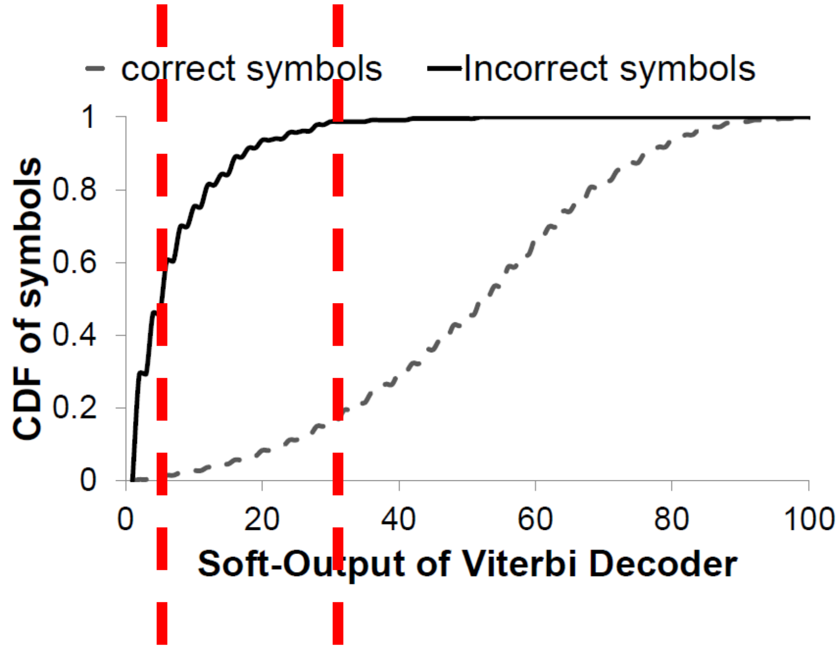
False negative

24Mbps, 12dB (higher)



PHY hints become unreliable on marginal SNR

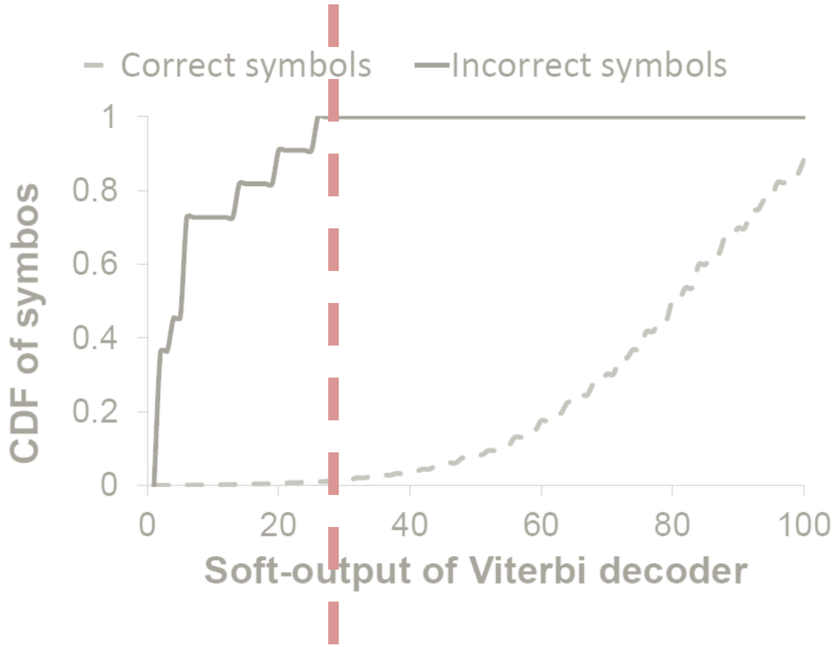
24Mbps, 10dB (marginal)



False negative

False positive

24Mbps, 12dB (higher)

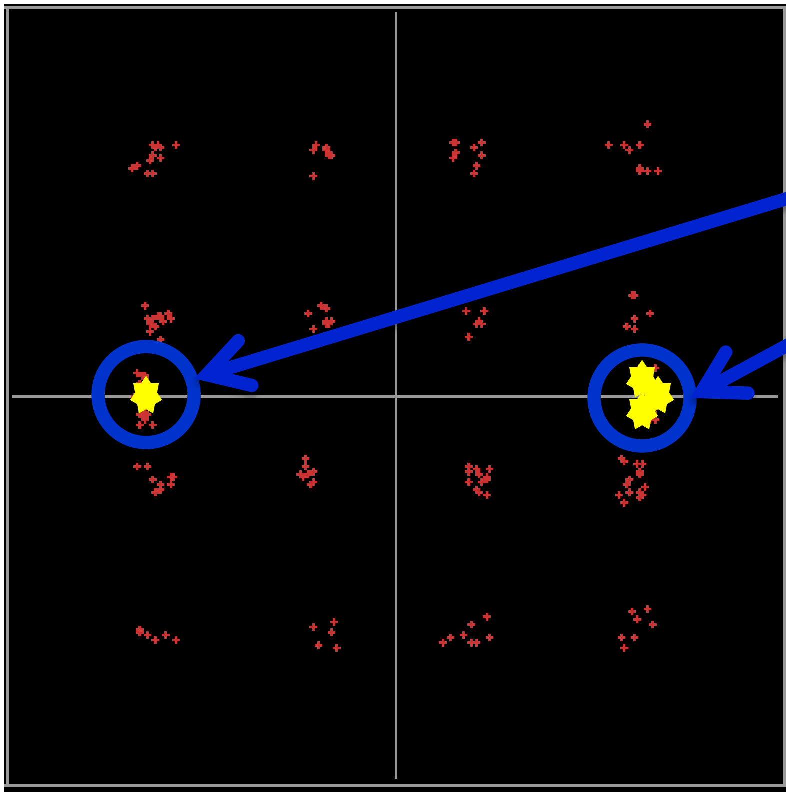


We explicitly embed CRC in each GOS

Segment CRCs add additional overhead

Can we avoid the overhead?

Pilot Side-Channel

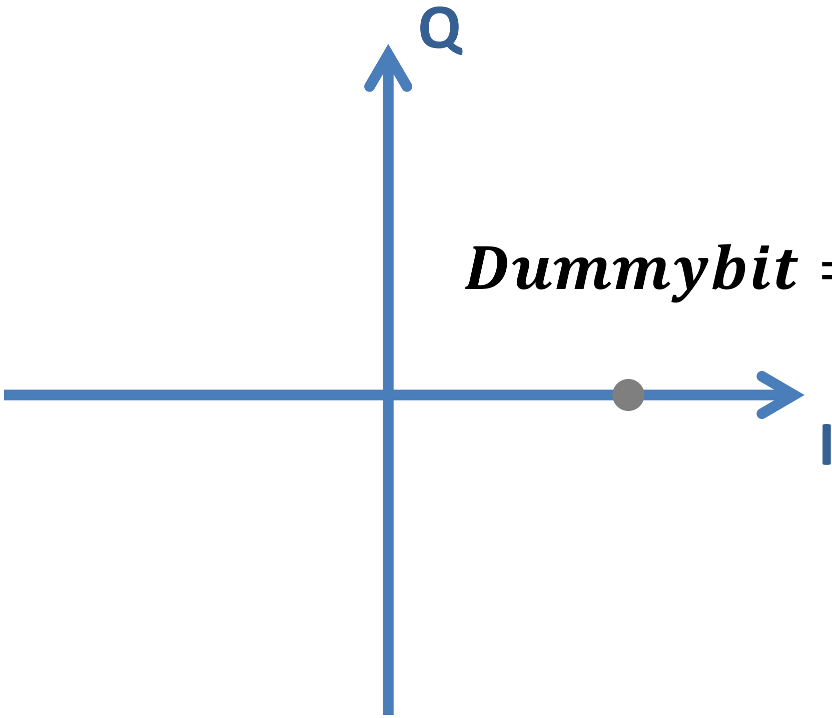


Dummy-bit Pilots

- Encode information in the pilots
  - ❑ Embed 16 bits in a GOS
  - ❑ Hamming (16, 11) code
  - ❑ CRC-10

Pilot Side-Channel

- How?
  - Differential BPSK (similar to 802.11b)



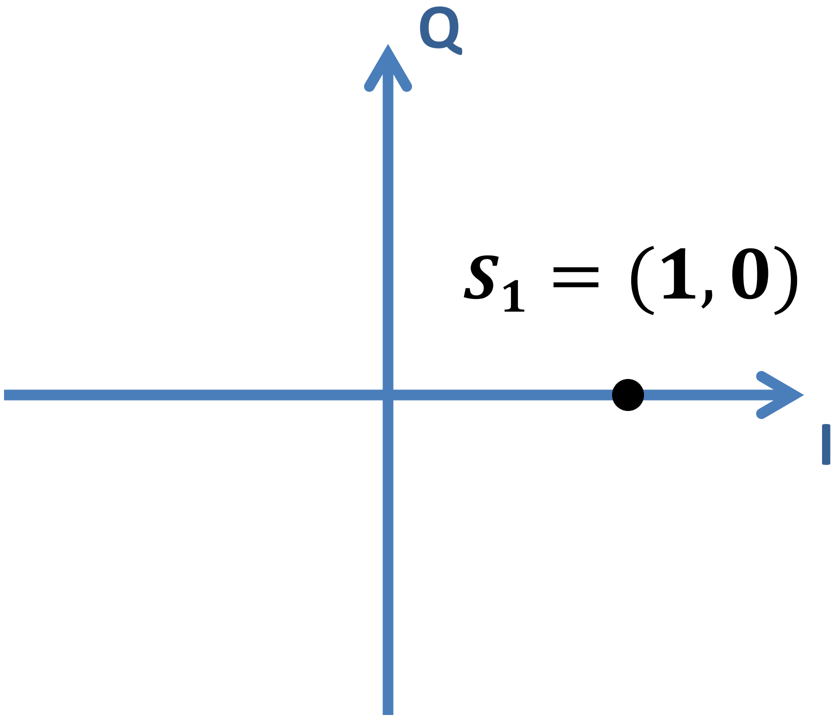
$$\boldsymbol{Dummybit = (1, 0)}$$

Example:

Symbol	Encoded	(I, Q)
s_0		$(1, 0)$

Pilot Side-Channel

- How?
  - Differential BPSK (similar to 802.11b)



Example:

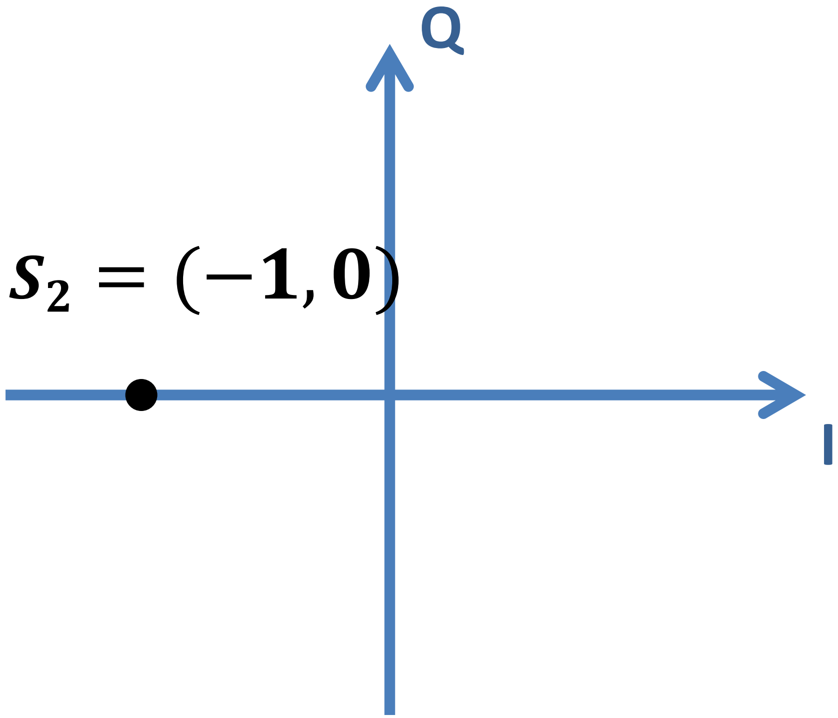
Symbol	Encoded	(I, Q)
S_0		$(1, 0)$
S_1	0	$(1, 0)$

Pilot Side-Channel

- How?
  - Differential BPSK (similar to 802.11b)

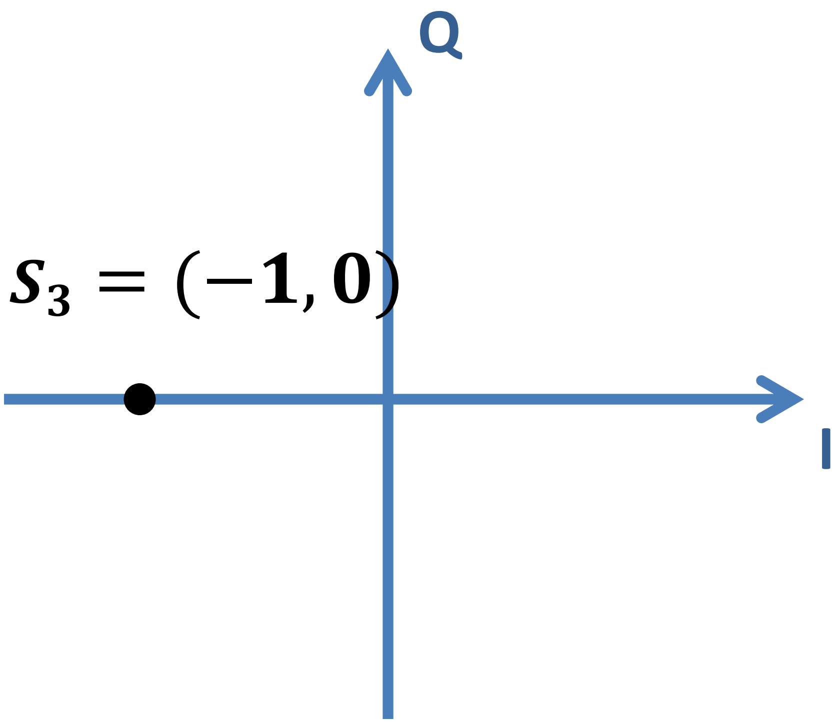
Example:

Symbol	Encoded	(I, Q)
S_0		$(1, 0)$
S_1	0	$(1, 0)$
S_2	1	$(-1, 0)$



Pilot Side-Channel

- How?
  - Differential BPSK (similar to 802.11b)

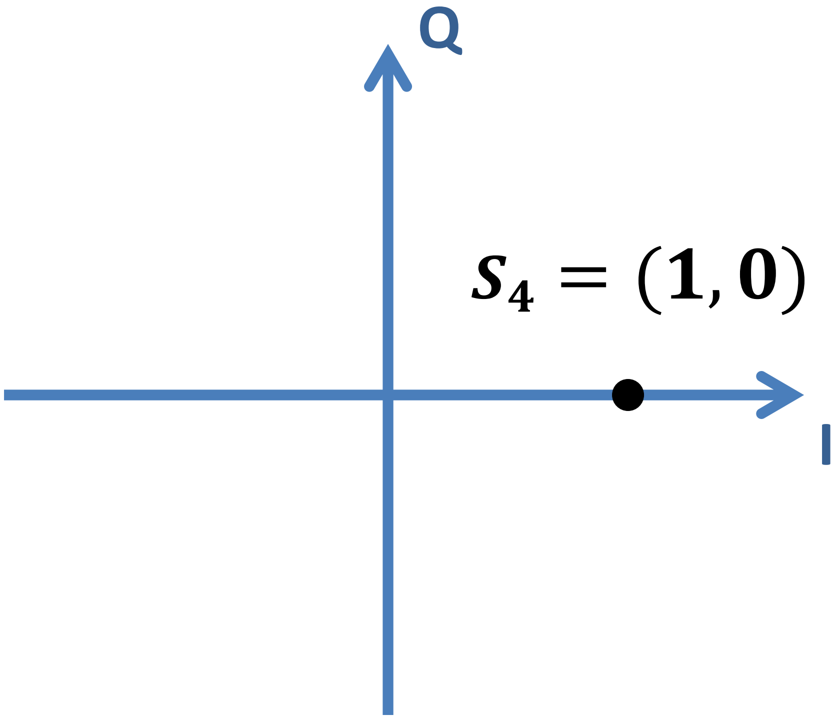


Example:

Symbol	Encoded	(I, Q)
S_0		$(1, 0)$
S_1	0	$(1, 0)$
S_2	1	$(-1, 0)$
S_3	0	$(-1, 0)$

Pilot Side-Channel

- How?
  - Differential BPSK (similar to 802.11b)

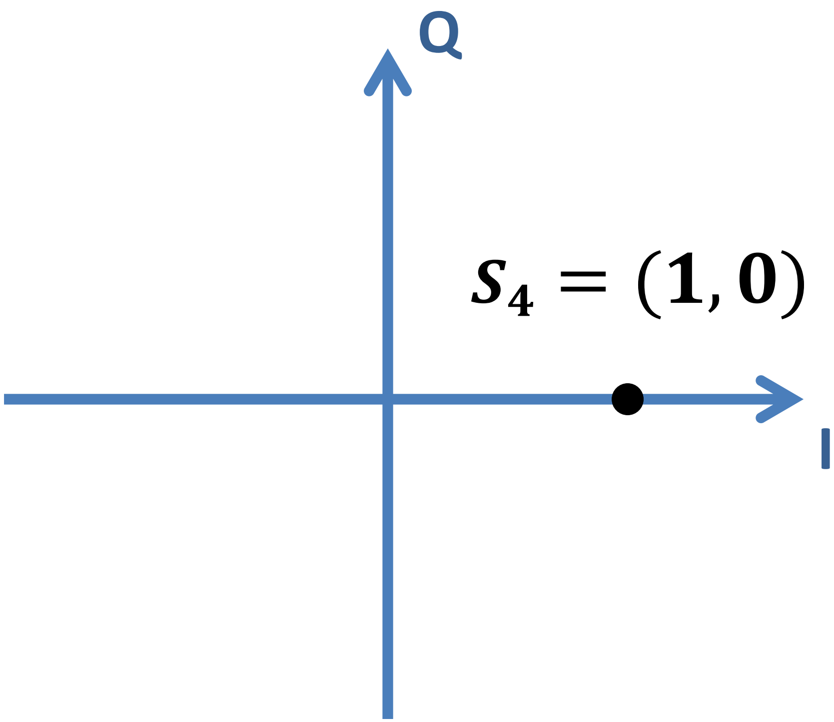


Example:

Symbol	Encoded	(I, Q)
S_0		$(1, 0)$
S_1	0	$(1, 0)$
S_2	1	$(-1, 0)$
S_3	0	$(-1, 0)$
S_4	1	$(1, 0)$

Pilot Side-Channel

- How?
  - Differential BPSK (similar to 802.11b)



Example:

Symbol	Encoded	(I, Q)
S_0		$(1, 0)$
S_1	0	$(1, 0)$
S_2	1	$(-1, 0)$
S_3	0	$(-1, 0)$
S_4	1	$(1, 0)$
…	…	…

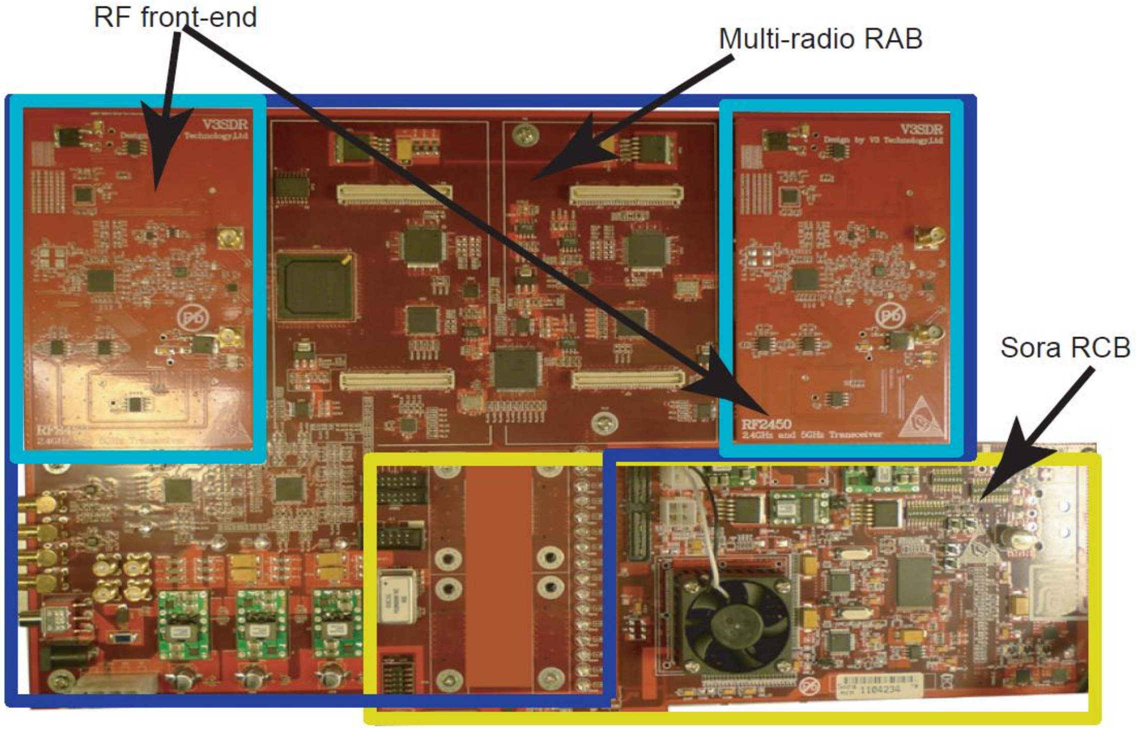
Decision Directed Pilot Tracking

- Pilots should be decoded first before used for channel tracking
  - No performance loss if pilots are correctly decoded
  - No performance loss even if pilots are not correctly decoded

- Normal pilots are inserted at beginning of an GOS
  - Pilot decision error will not propagate to next GOS

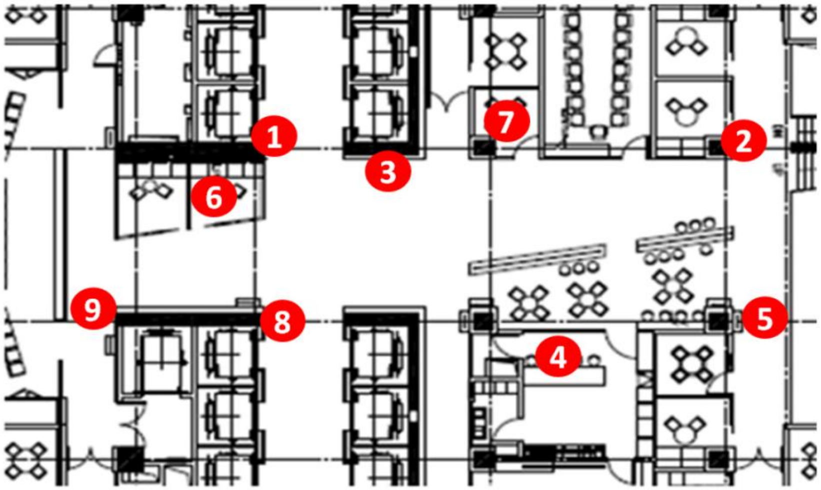
Sora Based Implementation

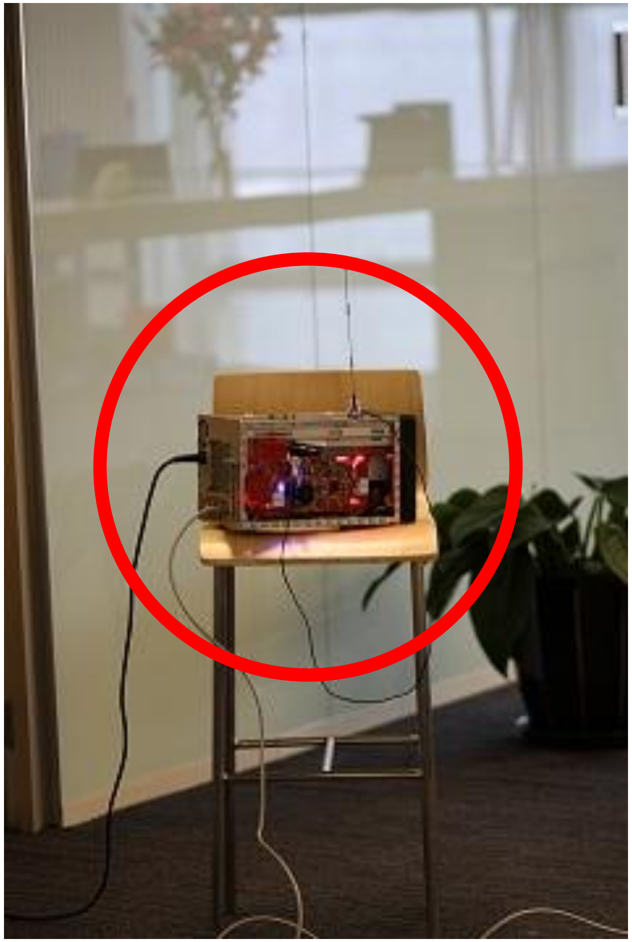
- Extend Sora
  - ❑ Multi-radio board
  - ❑ Direct symbol transmission to radio



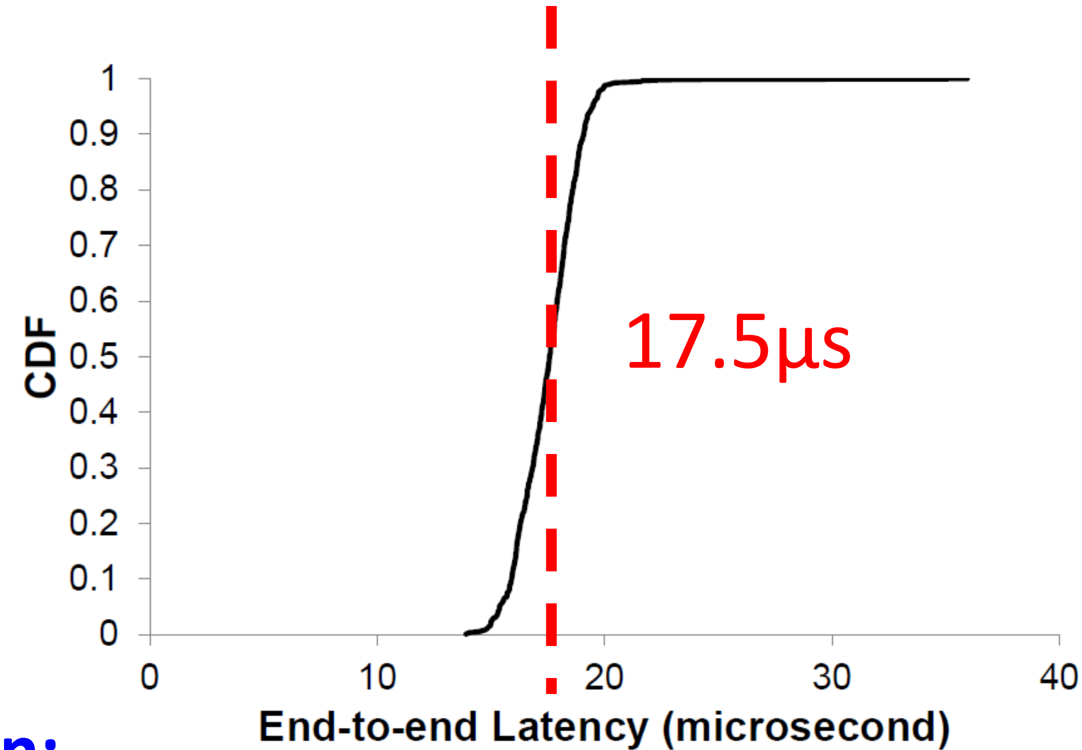
Performance Evaluation

- Is µACK feasible?
  - ❑ Micro-benchmarks
- What is the benefit of µACK?
  - ❑ Wired single link
  - ❑ 9 node real network





End-to-end Latency of µACK

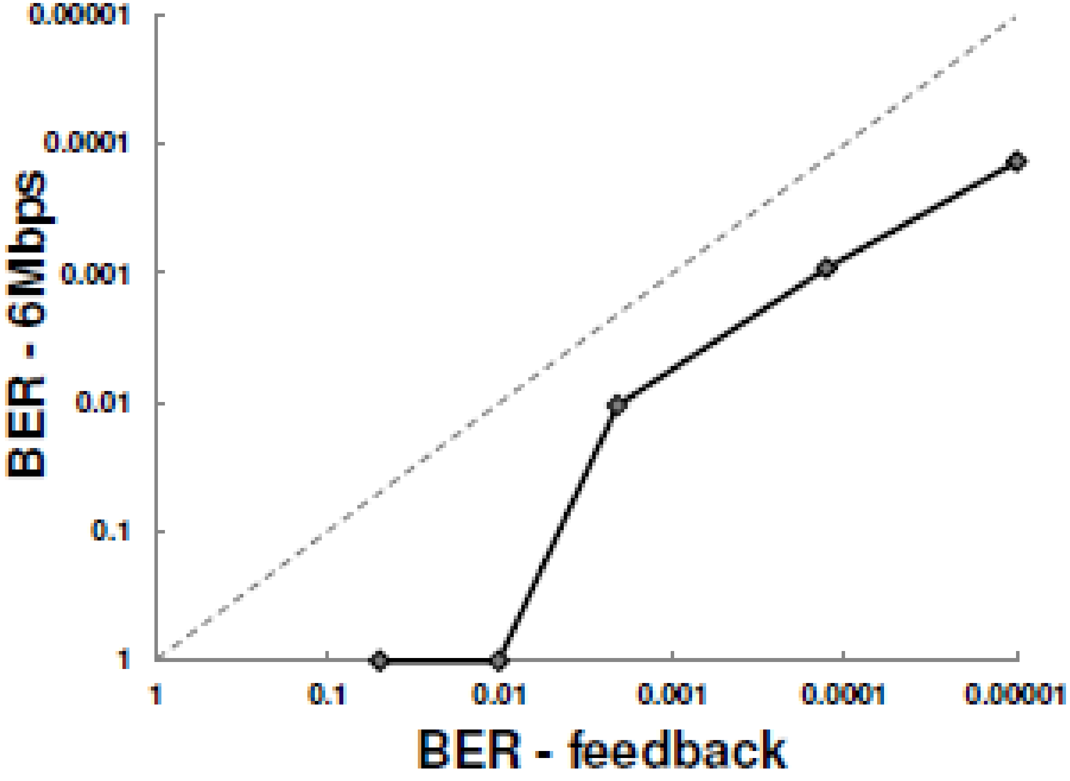


Breakdown:

Viterbi Decoding	µACK modulation	Hardware
7.5µs	1.96µs	9.103µs

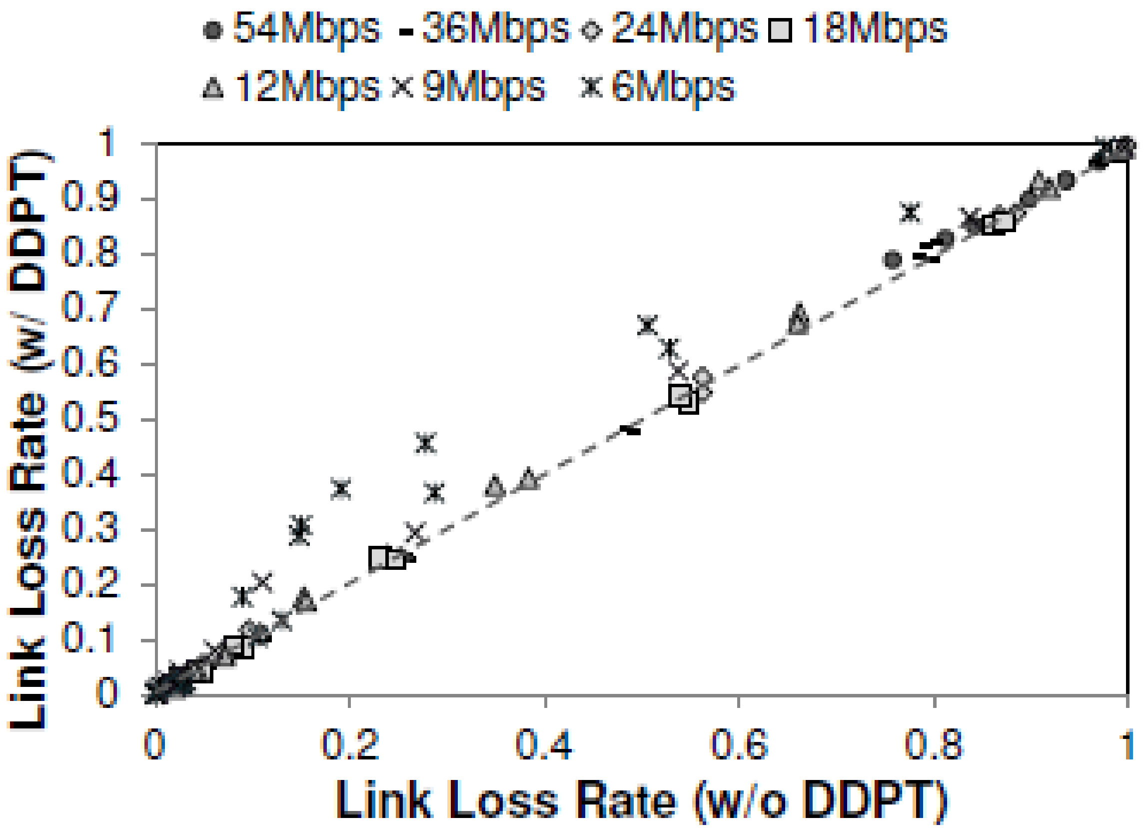
µACK PHY Performance

- **µACK vs. 802.11 6Mbps**

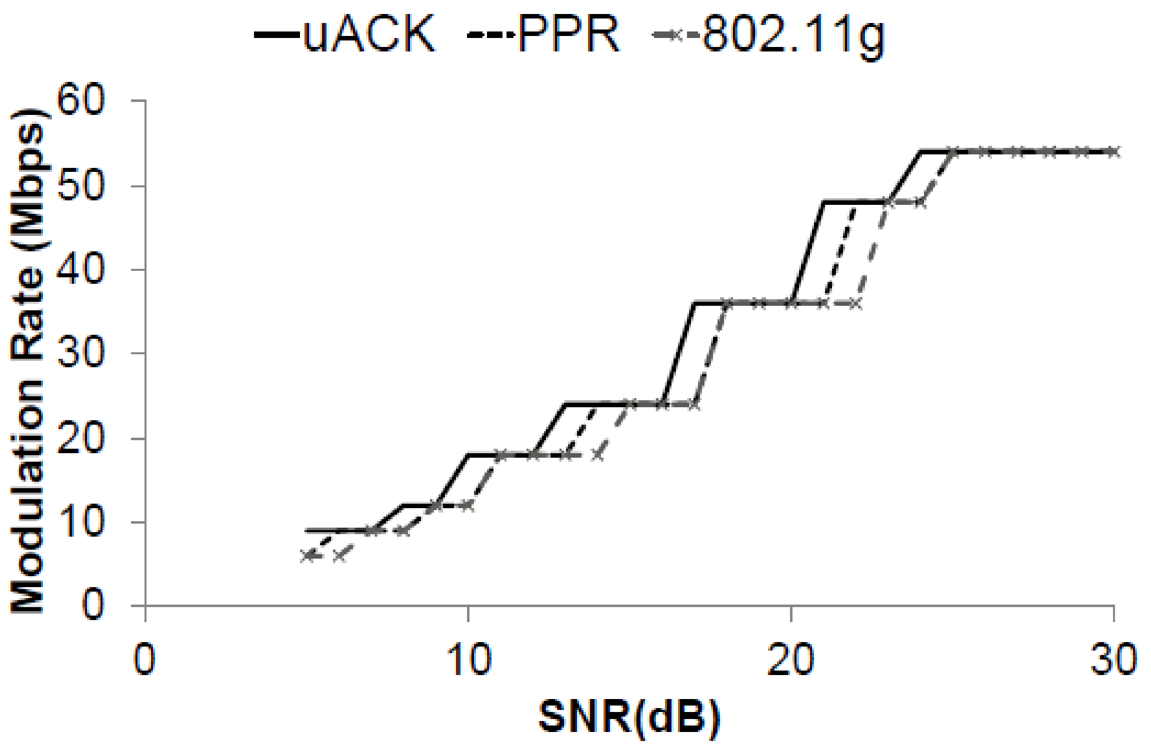


DDPT Performance

- **DDPT vs. Normal**



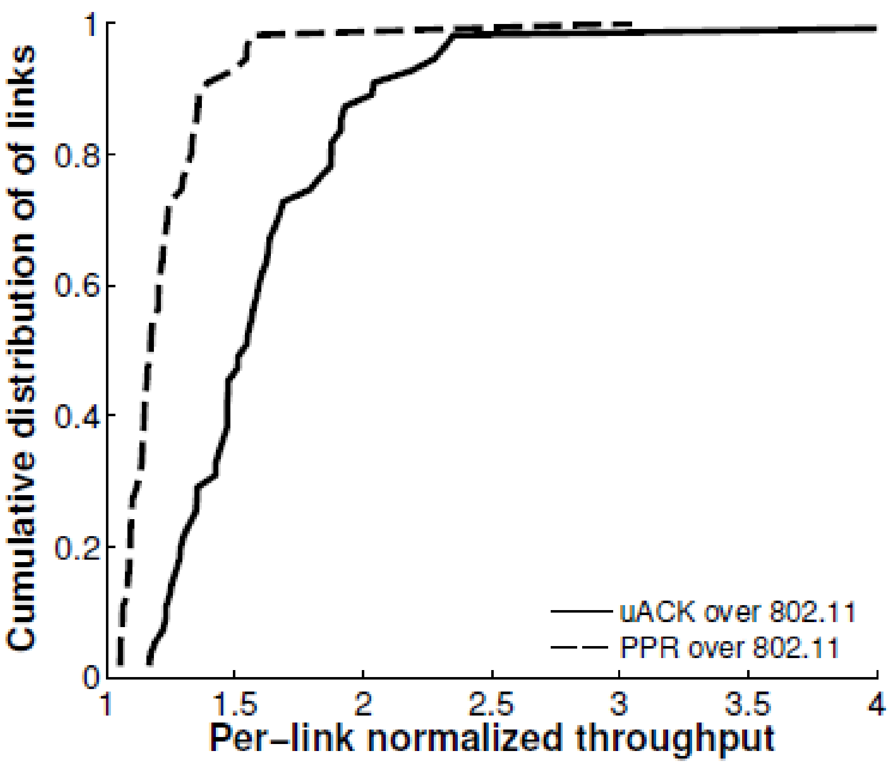
µACK on Wired Single Link



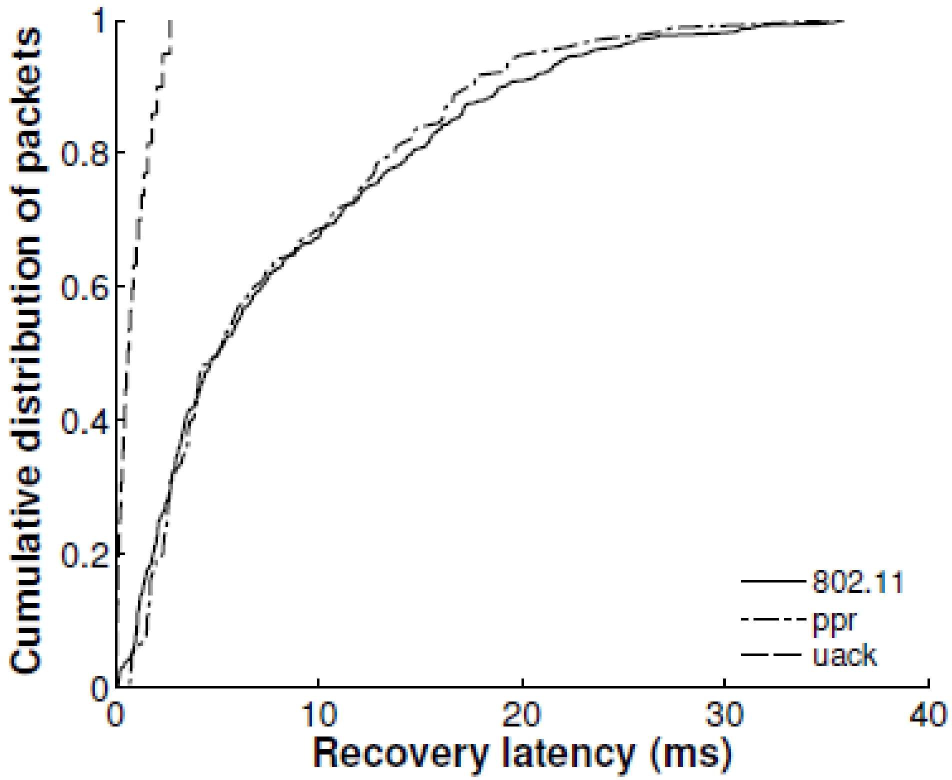
- μACK sender aggressively use higher data rates.
- Up to 220% over 802.11a, up to 30% over PPR

Trace-based Emulation

Throughput



Latency



Related Work

- Hybrid ARQs
  - ❑ Complementary to μACK
- Partial Packet Recovery
- CSMA/CN
- Rate adaptation
  - ❑ μACK shows by reducing loss recovery overhead, one can use more aggressive rates
  - ❑ μACK also enables in-frame rate adaptation
- Busy-tone schemes (DBTMA)
  - ❑ μACK can serve as an extended busy tone

Conclusion

- μACK enables sending fine-grained feedback
  - Collision detection
  - Mitigation of hidden & exposed terminal problem
  - In-frame loss recovery

- μACK is feasible & significantly improves spectrum efficiency
  - Reduces retransmission overhead
  - Increases transmission rate
  - Improves collision management