Wireless Communication Systems
@CS.NCTU

Lecture 8: Successive Interference Cancellation

Instructer: Kate Ching-Ju Lin (林靖茹)

Agenda

- Successive Interference Cancellation
- ZigZag decoding

SRN and SNR$_{dB}$

- SNR $= \frac{P_{\text{signal}}}{P_{\text{noise}}}$
- Unit of power: watt

  Logarithmic unit of power: decibel (**dBm**)

  $P_{\text{dB}} = 10 \log_{10} P$

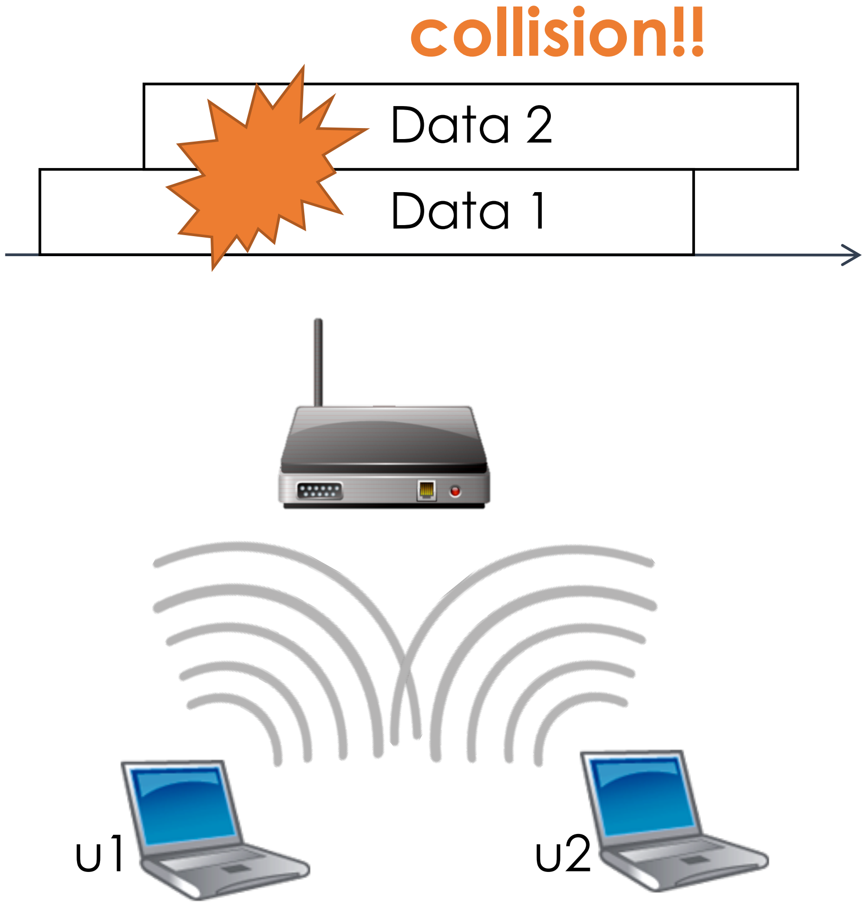
- $\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right)$

$$= 10 \log_{10}(P_{\text{signal}}) - 10 \log_{10}(P_{\text{noise}})$$

$$= P_{\text{signal,dBm}} - P_{\text{noise,dBm}}$$

Example:
signal = -70 dBm
noise = -90 dBm
SNR = -70 - (-90) = 20 dB

Scenario



$$y = h_1 x_1 + h_2 x_2 + n$$

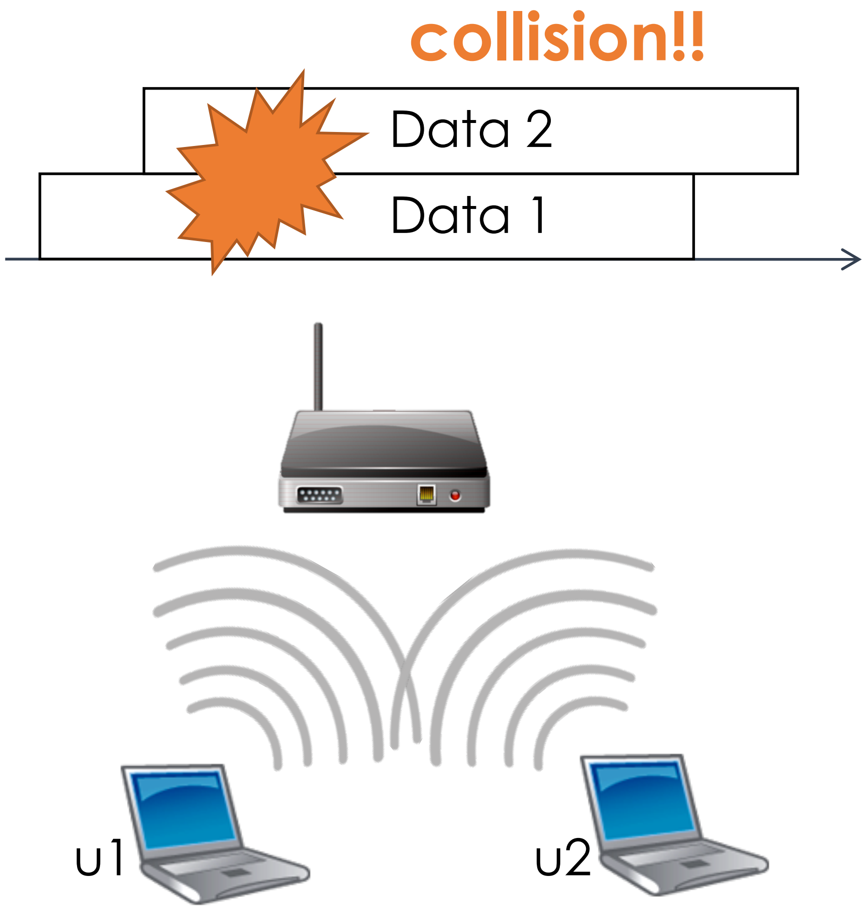
interference ($h_2 x_2$) noise (n)

$$\text{SINR} = \frac{P_1}{P_2 + N_0}$$

Reliably decode when the rate is no larger than capacity

$$R \leq C = \log(1 + \frac{P1}{P_2 + N_0})$$

Scenario



$y = h_1x_1 + h_2x_2 + n$

interference (h_2x_2) noise (n)

$$\text{SINR} = \frac{P_1}{P_2 + N_0}$$

Example:
signal = -70 dBm
Interference = -75 dBm
noise = -90 dBm
SNR ~= -70 - (-75) = 5 dB

Can still decode if selecting a very low bit-rate

SIC Decoding

- Successive Interference Cancellation (SIC)
  1. **Decode one** user first in the presence of interference
     $x'_2 = y/h_2 = x_2 + \mathbf{h_1x_1/h_2} + n/h_2$
  2. **Re-encode** the recovered data to remove the noise
     (demodulate x'_2 and re-modulate it)
  3. **Subtract** the re-encoded data from the received signal
     $y' = y - \mathbf{h_2x_2} = h_1x_1+n$
  4. Decode the second user
     $x'_1 = y'/h_1$

$y = h_1x_1 + h_2x_2 + n$

Capacity Region without SIC

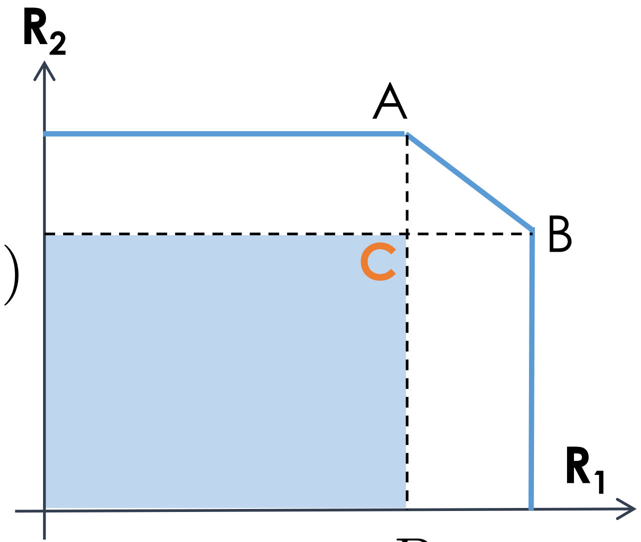
- $y_1 = h_1x_1 + (h_2x_2 + n)$
- $y_2 = h_2x_2 + (h_1x_1 + n)$

$$R_1 \leq \log(1 + \frac{P_1}{P_2 + N_0})$$

$$R_2 \leq \log(1 + \frac{P_2}{P_1 + N_0})$$

Maximal sum-rate: point C

$$\log(1 + \frac{P_2}{P_1 + N_0})$$



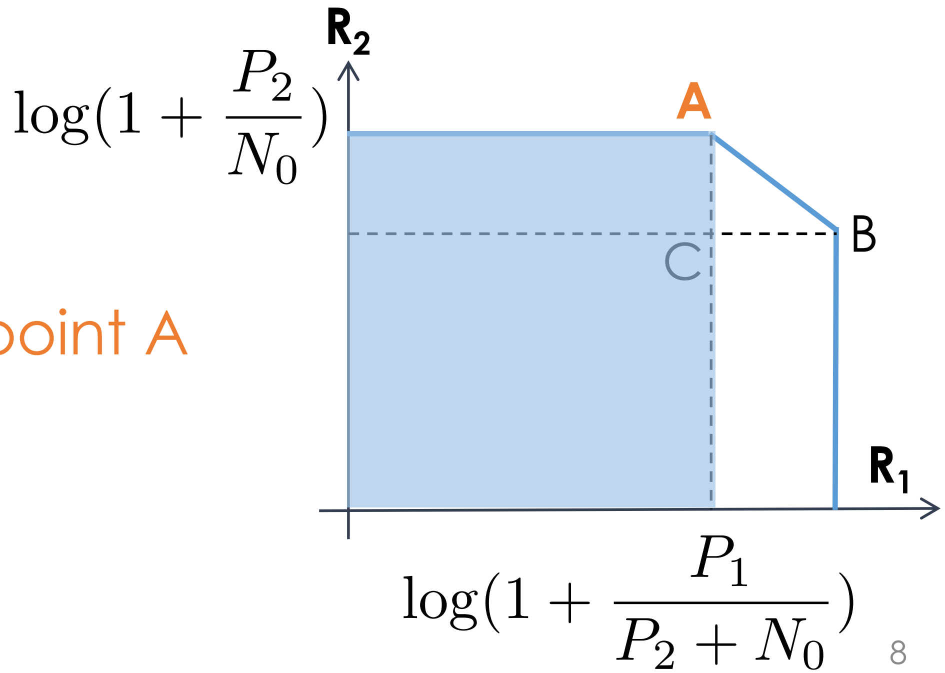
$$\log(1 + \frac{P_1}{P_2 + N_0})$$

Capacity Region of SIC

- Decoding order: user 1 → user 2
  - If we decode u1 in the presence of interfering u2, and then decode u2
  - $y_1 = h_1x_1 + (h_2x_2 + n)$
  - $y_2 = h_2x_2 + n$

    → Get single-user rate

    → Maximal sum-rate: point A

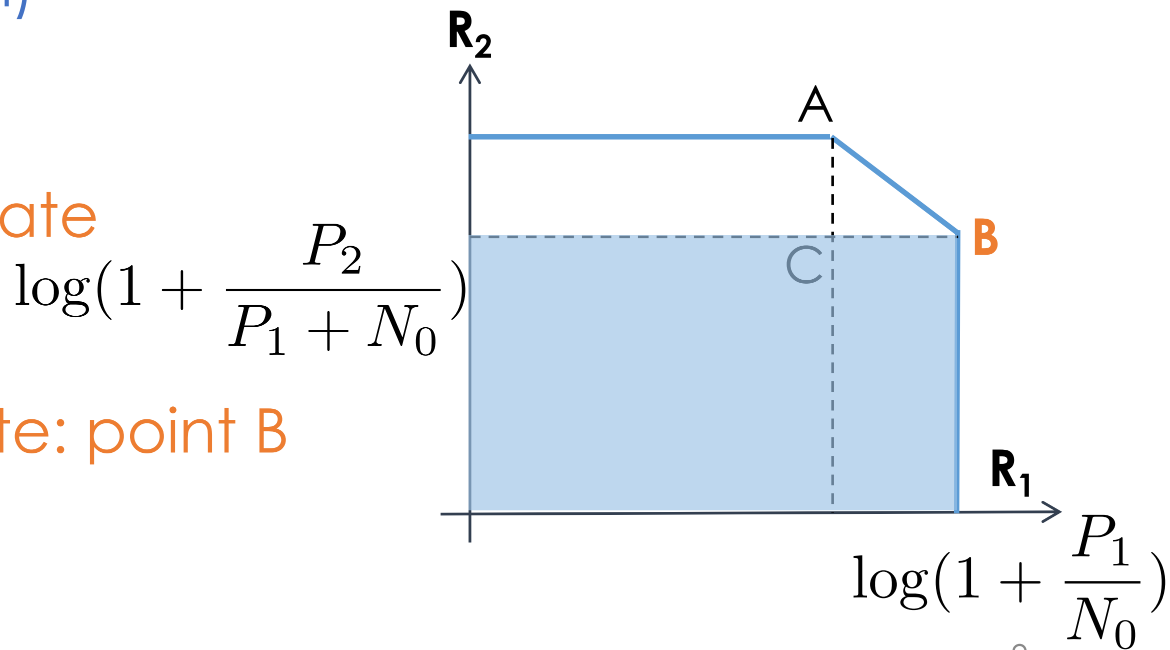


Capacity Region of SIC

- Decoding order: user 2 → user 1
  - If we decode u2 in the presence of interfering u1, and then decode u1
  - $y_2 = h_2x_2 + (h_1x_1 + n)$
  - $y_1 = h_1x_1 + n$

  → Get single-user rate

  → Maximal sum-rate: point B



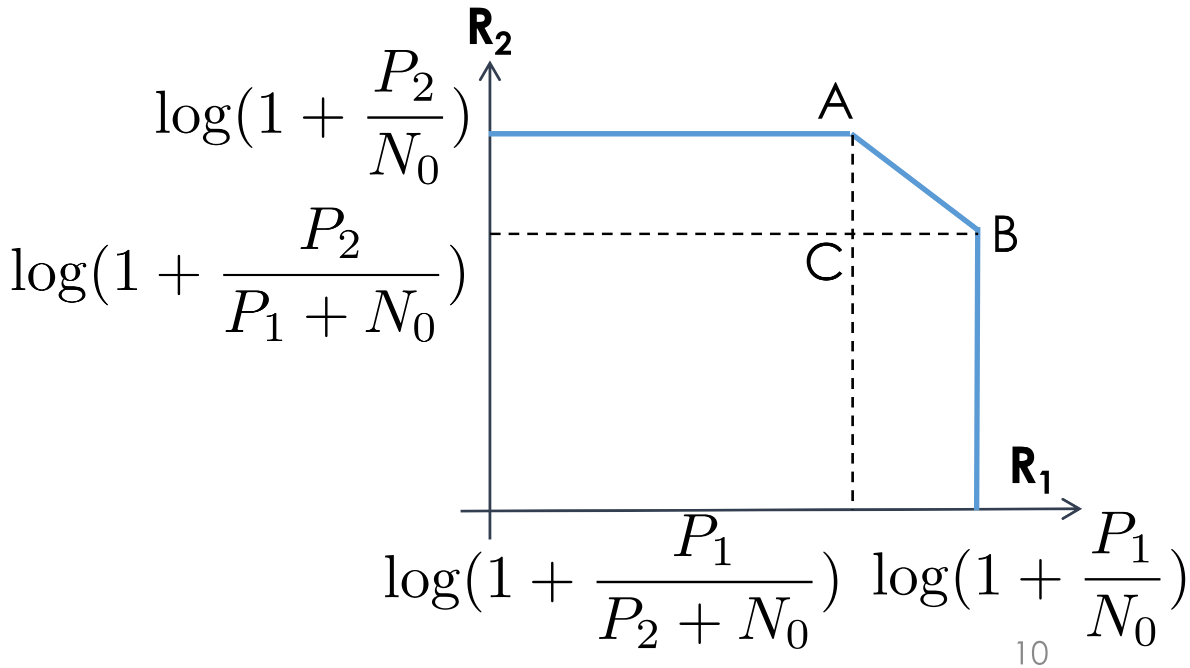
Capacity Region of SIC

- To ensure reliable decoding, the rates (R1, R2) need do satisfy three constraints:

$$R_1 + R_2 \leq \log(1 + \frac{P_1 + P_2}{N_0})$$

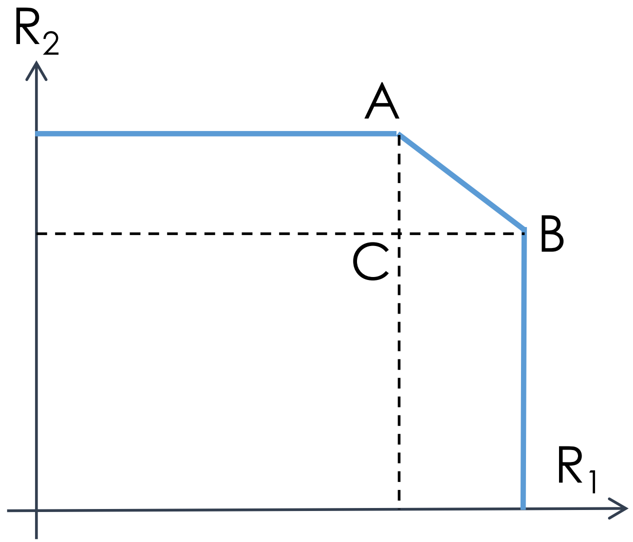
$$R_1 \leq \log(1 + \frac{P_1}{N_0})$$

$$R_2 \leq \log(1 + \frac{P_2}{N_0})$$



Capacity Region of SIC

- User 1 achieves its single-user bound (point B) while user 2 can get a non-zero rate
  - $R_2^* = \log(1 + \frac{P_1 + P_2}{N_0}) - \log(1 + \frac{P_1}{N_0}) = \log(1 + \frac{P_2}{P_1 + N_0})$
  - Namely, decode u2 in the presence of interfering u1
- Segment AB contains all the optimal sum-rate, and can be achieved via time-sharing
  - Pareto optimal



Decoding Order

- If the goal is to maximize the sum-rate, any point on AB is equally fine
- If we want to ensure max-min fairness such that the weak user get its best possible rate
  - Decode the stronger user first
- To minimize the total transmit power or increase the capacity in an interference-limited system
  - Decode the stronger user first

With SIC,
the near-far problem (SNR2 < SNR1) becomes an advantage
→ a far user now becomes decodable if SNR2 << SNR1

SIC for Multiple Users

$$y = h_1x_1 + h_2x_2 + \ \ldots + h_Nx_N + n$$

- Repeat the following procedure iteratively
  1. Decode any user $x_i = y/h_i$
  2. Re-encode x_i (demodulate and re-modulate)
  3. Subtract the re-encoded signal from y
- The user decoded earlier is interfered by more users

Use SIC in MIMO Decoding

- Standard Zero Forcing (ZF) decoding
  - SNR reduction due to channel correlation
    $SNR_{ZF} = SNR_{orig} * \sin^2(\theta)$
  - In 2x2 system, both streams suffer from SNR reduction if they are both decoded using ZF
- Combine ZF with SIC
  - 2x2 example
  - Decode x_2 using ZF
  - Decode x_1 using SIC

Decode x_2 Using ZF

orthogonal vectors

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix} x_1 + \begin{pmatrix} h_{12} \\ h_{22} \end{pmatrix} x_2 + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

* h_{21}

* - h_{11}

+)

$$y_1 h_{21} - y_2 h_{11} = (h_{12} h_{21} - h_{22} h_{11}) x_2 + n'$$

$$x_2' = \frac{y_1 h_{21} - y_2 h_{11}}{h_{12} h_{21} - h_{22} h_{11}}$$

$$= x_2 + \frac{n'}{h_{12} h_{21} - h_{22} h_{11}}$$

$$= x_2 + \frac{n'}{\vec{h}_2 \cdot \vec{h}_1^{\perp}}$$

Decode x_1 Using SIC

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix} x_1 + \begin{pmatrix} h_{12} \\ h_{22} \end{pmatrix} x_2 + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

- Re-encode x_2
- Removing x_2 and we get

$$y_1 = h_{11} x_1 + n_1$$
$$y_2 = h_{21} x_1 + n_2$$

- Use traditional SISO decoder

$$x_1 = \frac{y_1}{h_{11}} \text{ or } x_1 = \frac{y_2}{h_{21}}$$

ZF-SIC Decoding

- Combine ZF with SIC to improve SNR
  - Decode one stream and subtract it from the received signal
  - Repeat until all the streams are recovered
  - Example: after decoding x_2, we have $y_1 = h_1x_1+n_1$ → decode x_1 using standard SISO decoder
- Why it achieves a higher SNR?
  - The streams recovered after SIC can be projected to a smaller subspace → lower SNR reduction
  - In the 2x2 example, x_1 can be decoded as usual without ZF → no SNR reduction (though x2 still experience SNR loss)

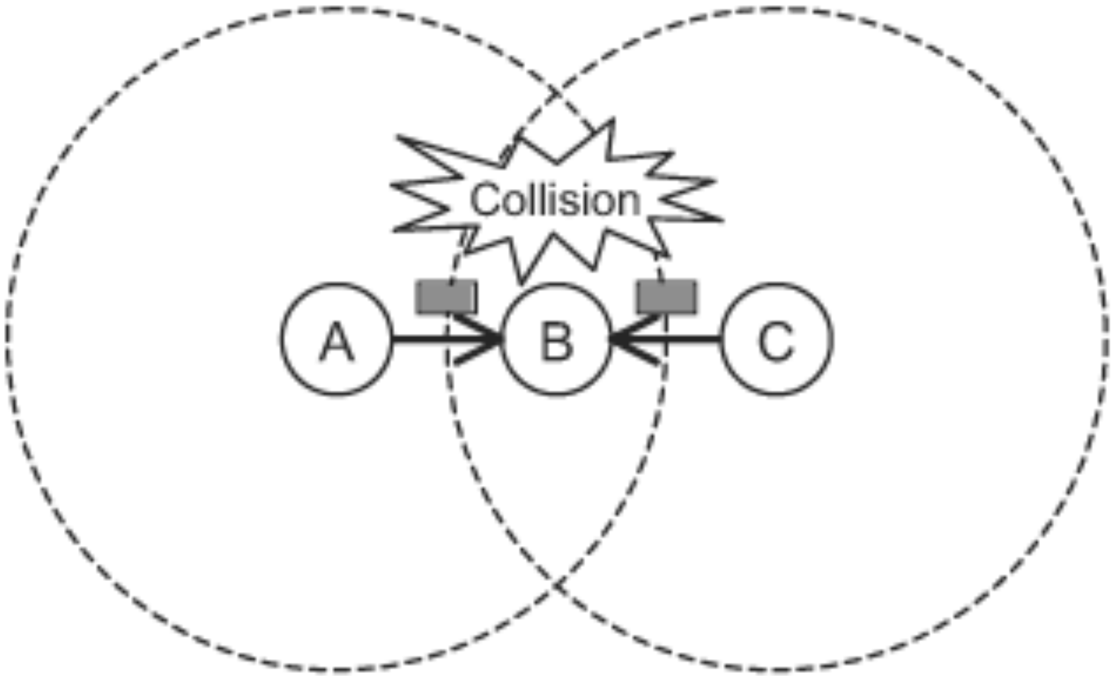
Wireless Communication Systems
@CS.NCTU

Lecture 8: Successive Interference Cancellation

ZigZag Decoding (SIGCOMM'08)

Lecturer: Kate Ching-Ju Lin (林靖茹)

Hidden Terminal



- Two nodes hidden to each other transmit at the same time, leading to collision

ZigZag

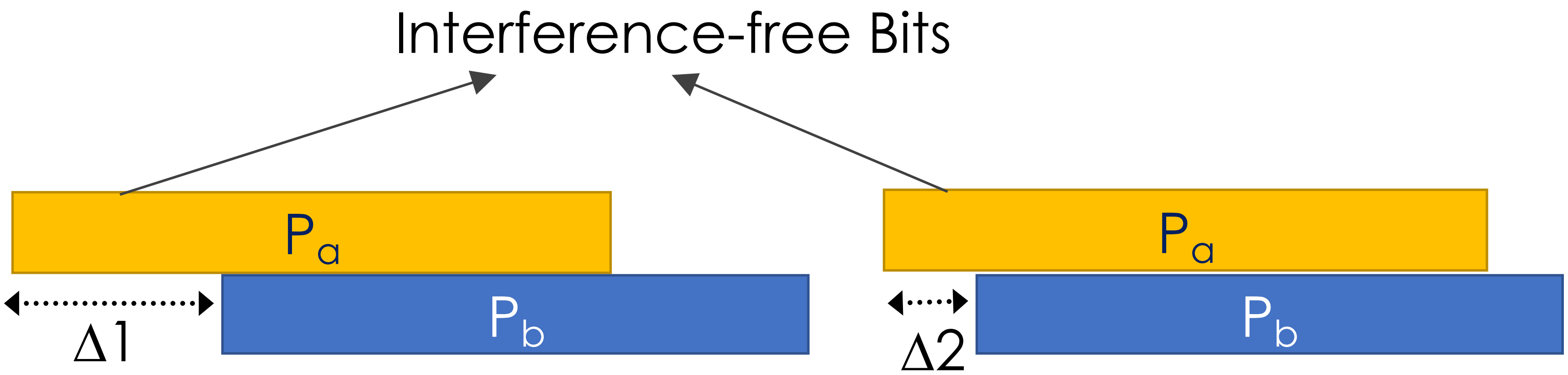
Exploits 802.11's behavior

- Retransmissions

  → Same packets collide again

- Senders use random jitters

  → Collisions start with interference-free bits

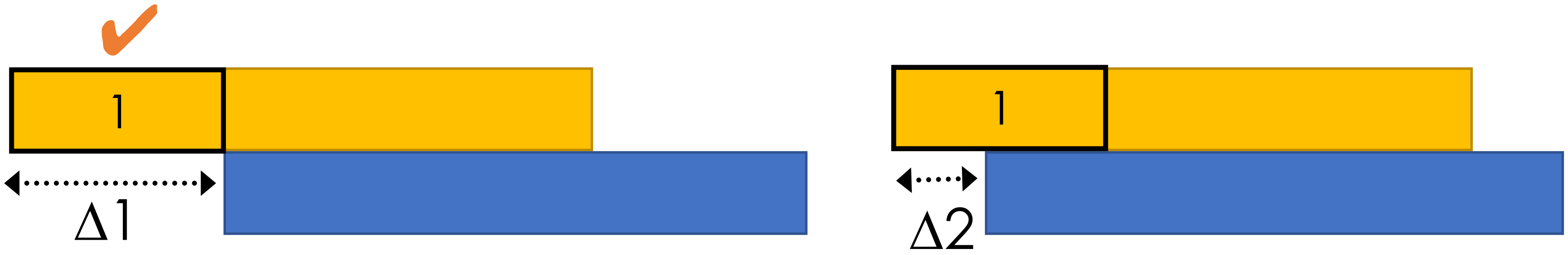


How does ZigZag Work?

$\Delta 1 \neq \Delta 2$

- Find a chunk that is interference free in one collision and has interference in the other

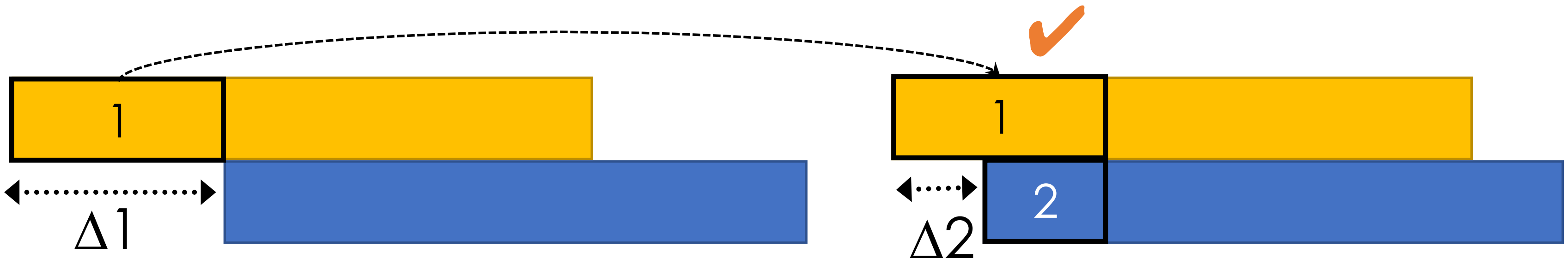
How does ZigZag Work?



Δ1 ≠Δ2

- Find a chunk that is interference free in one collision and has interference in the other
- Decode the interference-free chunk and subtract it from the other collision

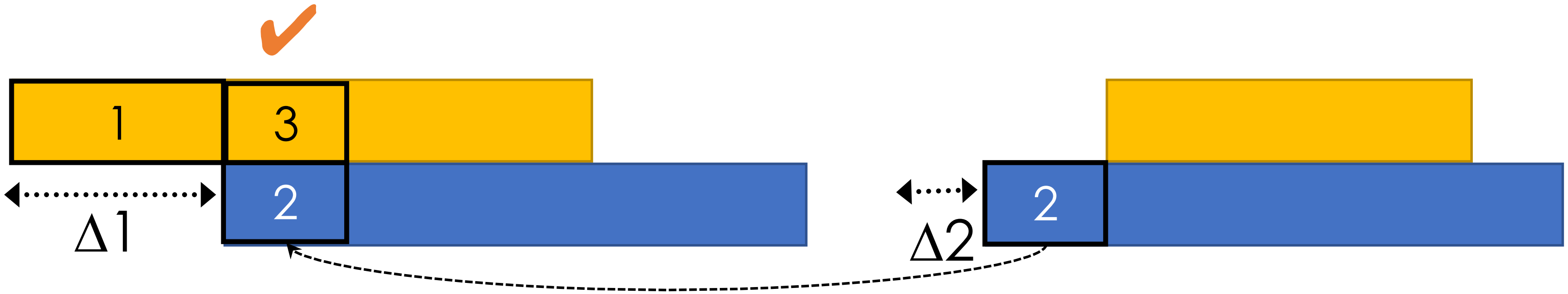
How does ZigZag Work?



Δ1 ≠Δ2

- Find a chunk that is interference free in one collision and has interference in the other
- Decode the interference-free chunk and subtract it from the other collision

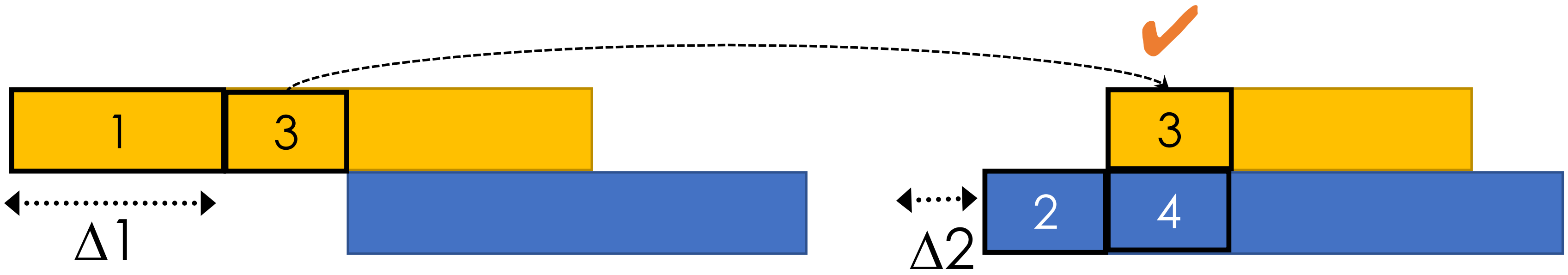
How does ZigZag Work?



Δ1 ≠Δ2

- Find a chunk that is interference free in one collision and has interference in the other
- Decode the interference-free chunk and subtract it from the other collision

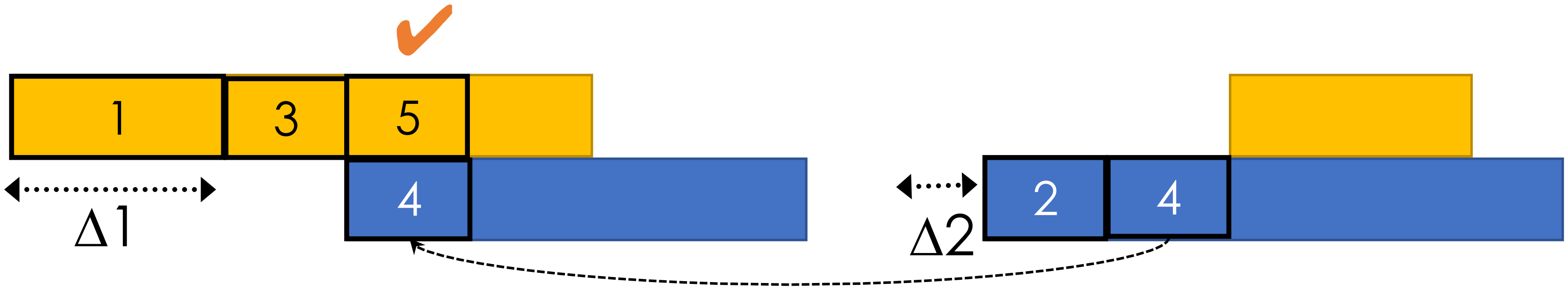
How does ZigZag Work?



Δ1 ≠Δ2

- Find a chunk that is interference free in one collision and has interference in the other
- Decode the interference-free chunk and subtract it from the other collision

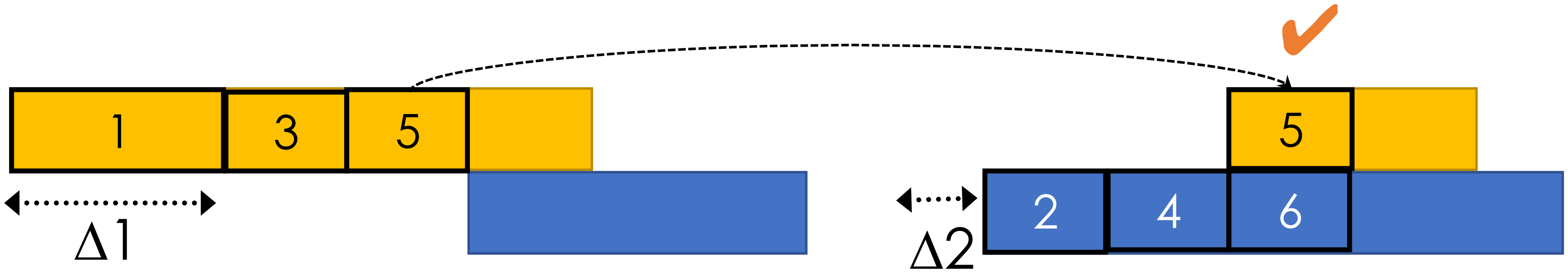
How does ZigZag Work?



$\Delta 1 \neq \Delta 2$

- Find a chunk that is interference free in one collision and has interference in the other
- Decode the interference-free chunk and subtract it from the other collision

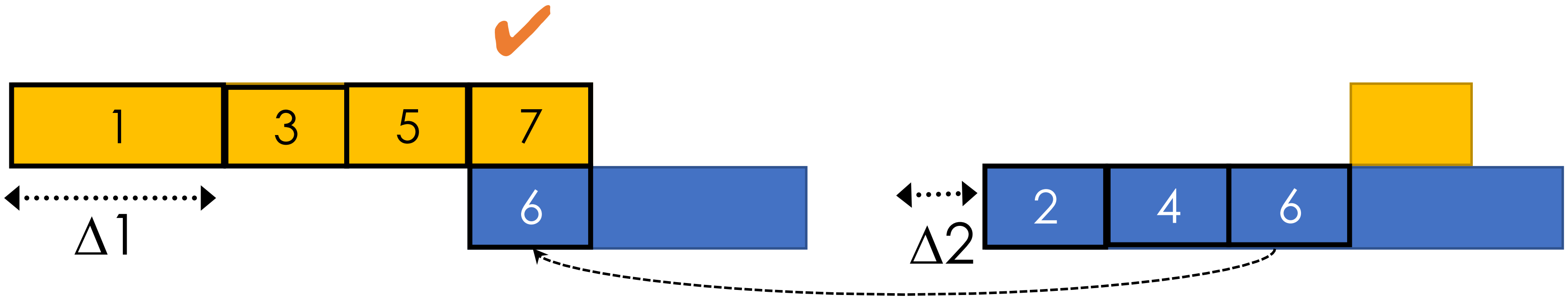
How does ZigZag Work?



Δ1 ≠Δ2

- Find a chunk that is interference free in one collision and has interference in the other
- Decode the interference-free chunk and subtract it from the other collision

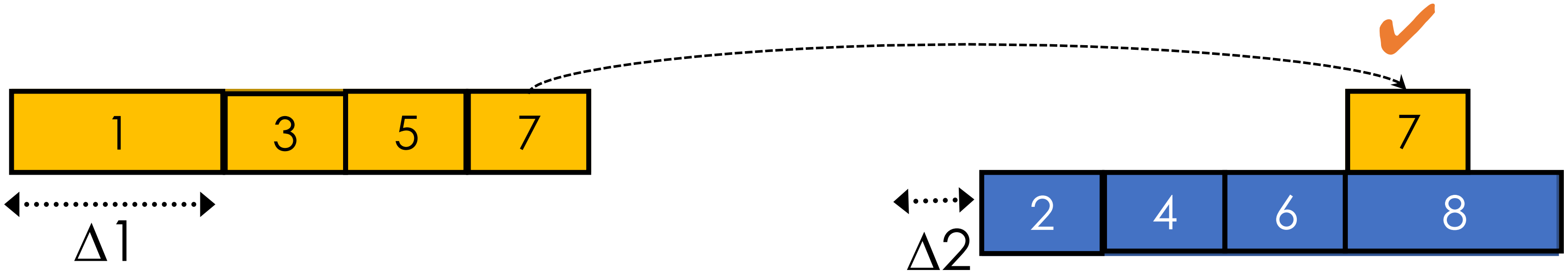
How does ZigZag Work?



Δ1 ≠Δ2

- Find a chunk that is interference free in one collision and has interference in the other
- Decode the interference-free chunk and subtract it from the other collision

How does ZigZag Work?



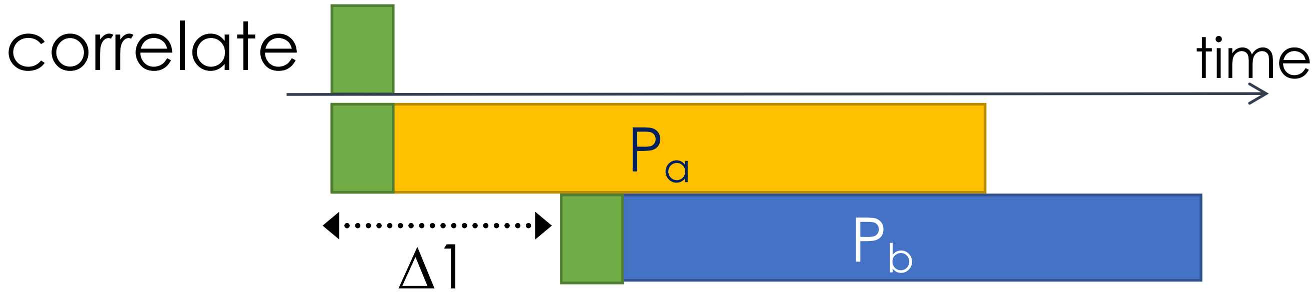
Δ1 ≠Δ2

- Deliver 2 packets in 2 timeslots
- As efficient as if the packets did not collide

Practical Issues

- How does the receiver know it is a collision and where they start?
- What if the channel has changes in the second collision?
- How to deal with error propagation?

Detecting Collisions

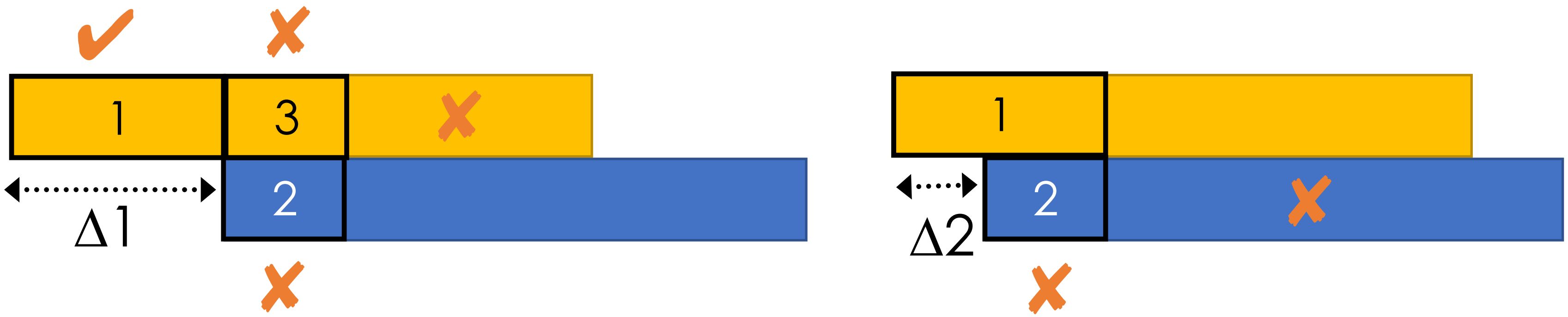


- Preamble correlation
  - Detect collision and the offset value Δ
  - Work despite interference because correlation with an independent signal (random data samples) is zero

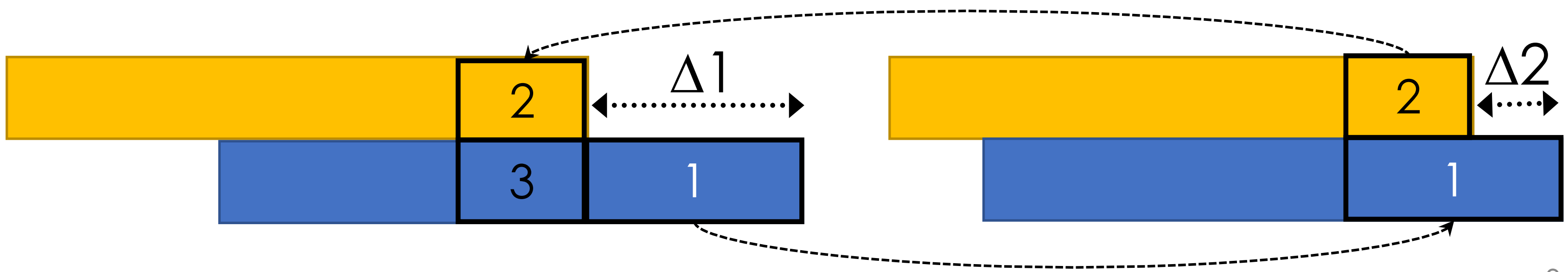
Signal Subtraction

- Channel's attenuation or phase may change between collisions
- Can't simply subtract a chunk across collisions
- Subtract as conventional SIC
  - **Decode** chunk in one collision into bits
  - Demodulate and **re-modulate** bits to get channel-free signal
  - Apply the **channel learned from the other collision** to encode the signal
  - **Subtract** it!

What if decoding errors happen?



- Error can propagate across chunks
- Cannot completely avoid the problem, but can reduce this probability via leveraging **time diversity**
  - Get two independent decodings: forward and backward



When will ZigZag Fail?

- The offsets in the two collisions happen to be the same
- A packet is sent at different bit-rates (modulation and coding schemes) in the two collisions
- Packets are modulated with OFDM
  - Symbols cannot be reliably converted the frequency domain when the colliding packets are not aligned in the symbol level
  - Lead to **inter-symbol interference**

Wireless Communication Systems
@CS.NCTU

Lecture 8: Successive Interference Cancellation

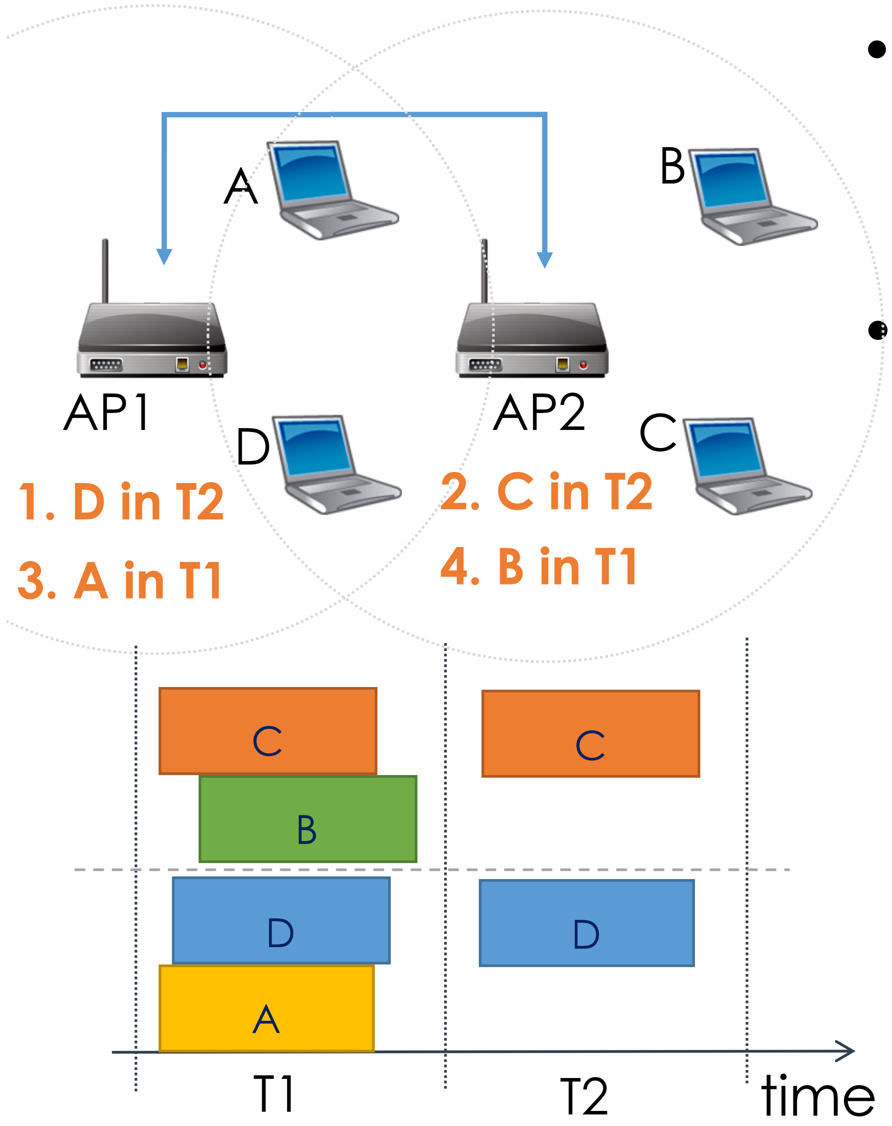
Symphony: Cooperative Packet Recovery over the Wired Backbone(MOBICOM'13)

Lecturer: Kate Ching-Ju Lin (林靖茹)

Basic Ideas

- Allow multiple APs to cooperatively recover their collided packets
- Exchange decoded bits via the wired backbone
- Leverage the property that not all the APs will hear the same set of packets
  - An AP hears an interference-free packet can initiate SIC decoding

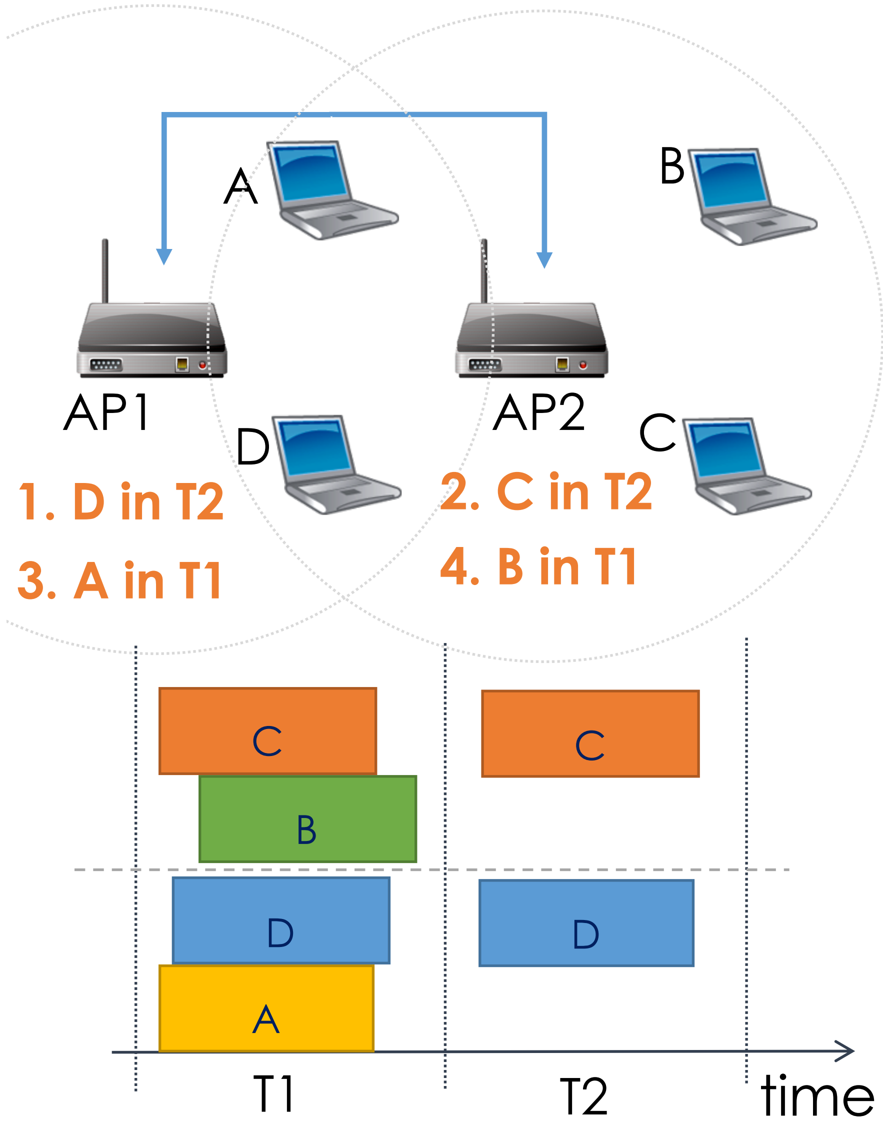
Example



- Clients
  - In T1, four nodes transmit
  - In T2, C and D retransmit
- APs

1. AP1 decodes the interference-free packet from D in T2
2. AP1 forwards the bits of D to AP2 s.t. it can uses SIC to recover C in T2 via SIC
3. AP1 uses SIC to subtract D in T1 and decode A
4. AP1 forwards the bits of A to AP2 s.t. it can recover B in T1

Example



- Deliver 4 packets in two slots
- TDMA: need 4 slots

Challenges

- Determine the decoding order so as to minimize the amount of traffic forwarded via the wired backbone
- Specify which clients should transmit in which time slots so as to maximize the number of transmissions
- Deal with imperfect time synchronization among APs and the latency over the backbone

Wireless Communication Systems
@CS.NCTU

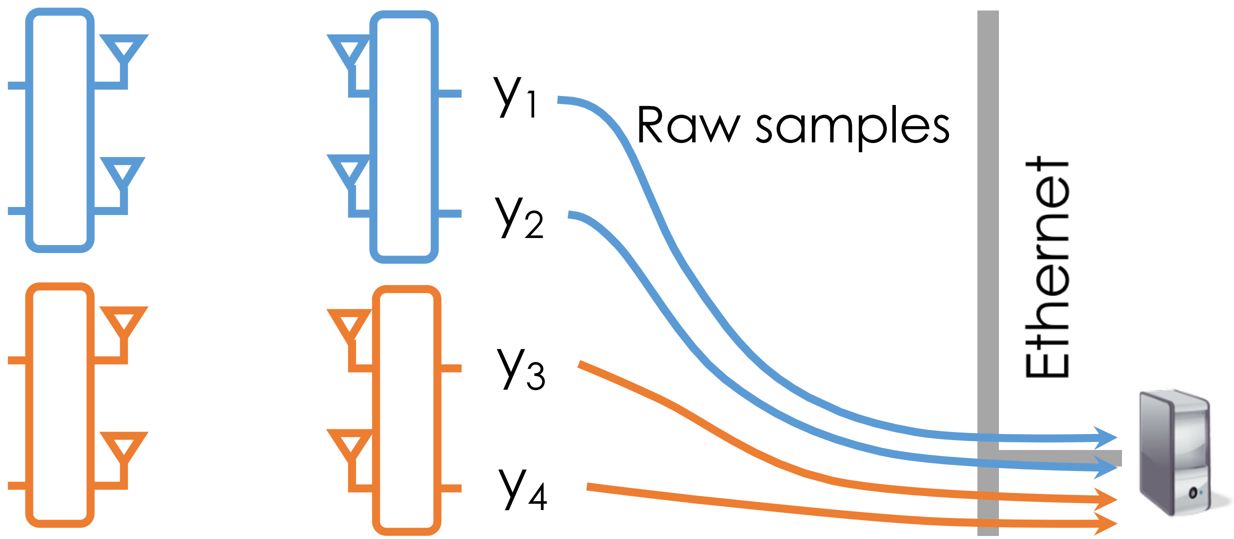
Lecture 5: Multi-User MIMO (MU-MIMO)

Interference Alignment and Cancellation (SIGCOMM'09)

Lecturer: Kate Ching-Ju Lin (林靖茹)

Naïve Cooperative MIMO

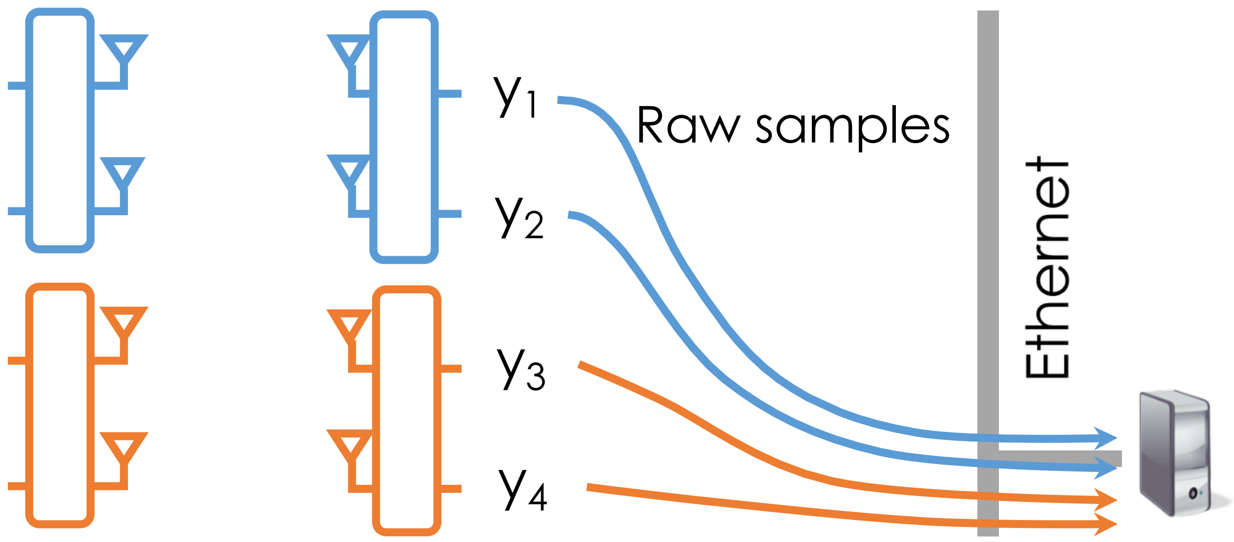
- Say we combine two 2-antnena APs as a 4–antenna virtual AP
- Naïve solution:
  - Connect the two APs to a server via Ethernet
  - Each physical AP sends every received raw signal (complex values) to the server over Ethernet



Naïve Cooperative MIMO

Impractical overhead:
For example, a 3 or 4-antenna system needs 10's of Gb/s

- Naïve solution:
  - Connect the two APs to a server via Ethernet
  - Each physical AP sends every received raw signal (complex values) to the server over Ethernet



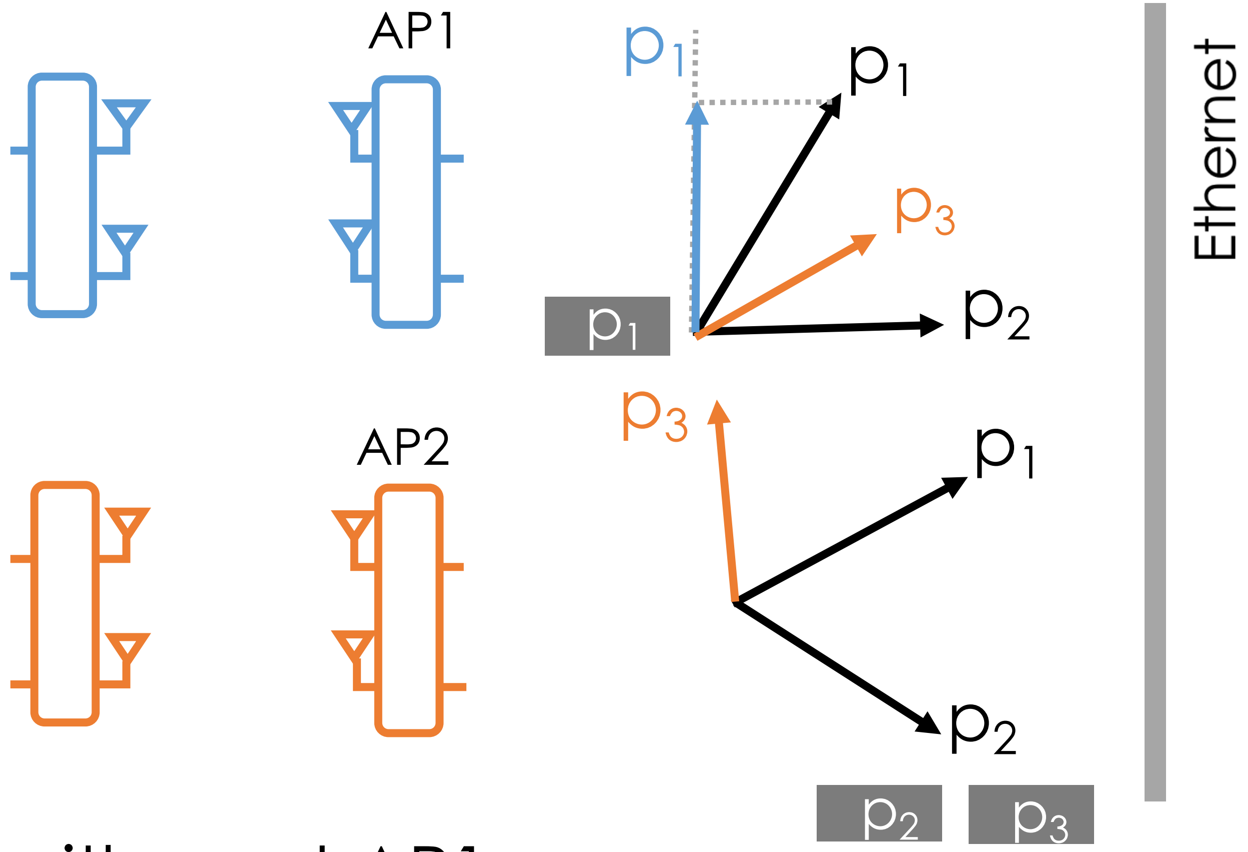
How to Minimize Ethernet Overhead?

- High-level idea:
  1. Decode some packets in certain AP
  2. Forward the decoded packets through the Ethernet to other APs
  3. Other APs decode the remaining packets
  4. Repeat 1-3 until all packets are recovered

How to Minimize Ethernet Overhead?

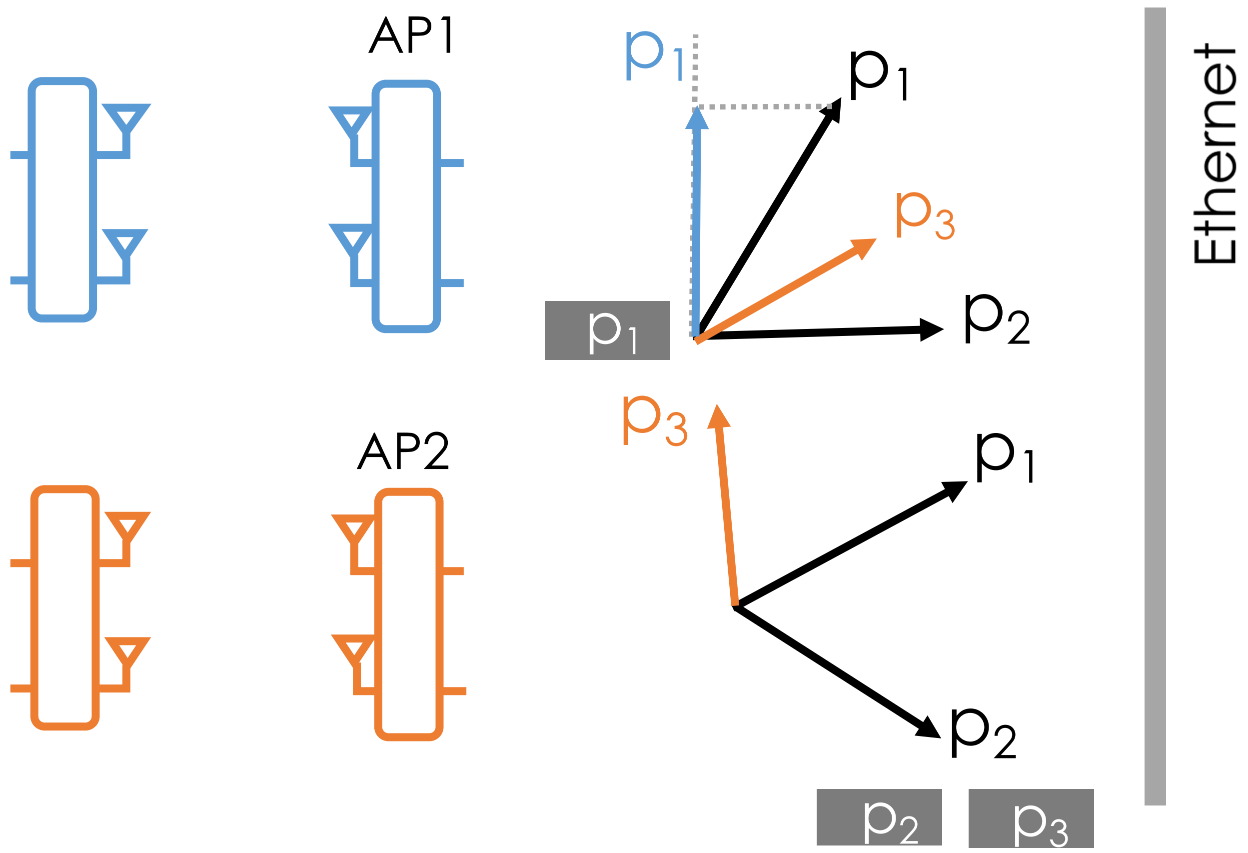
- Advantage:
  - The size of data packets is much smaller than the size of raw samples → minimize overhead
- Challenge:
  - In theory, an N-antenna AP cannot recover M concurrent transmissions if M>N
  - How can an N-antenna AP recover its packet from M concurrent transmissions (M>N)?
    → **Interference Alignment and Cancellation**

Interference Alignment and Cancellation



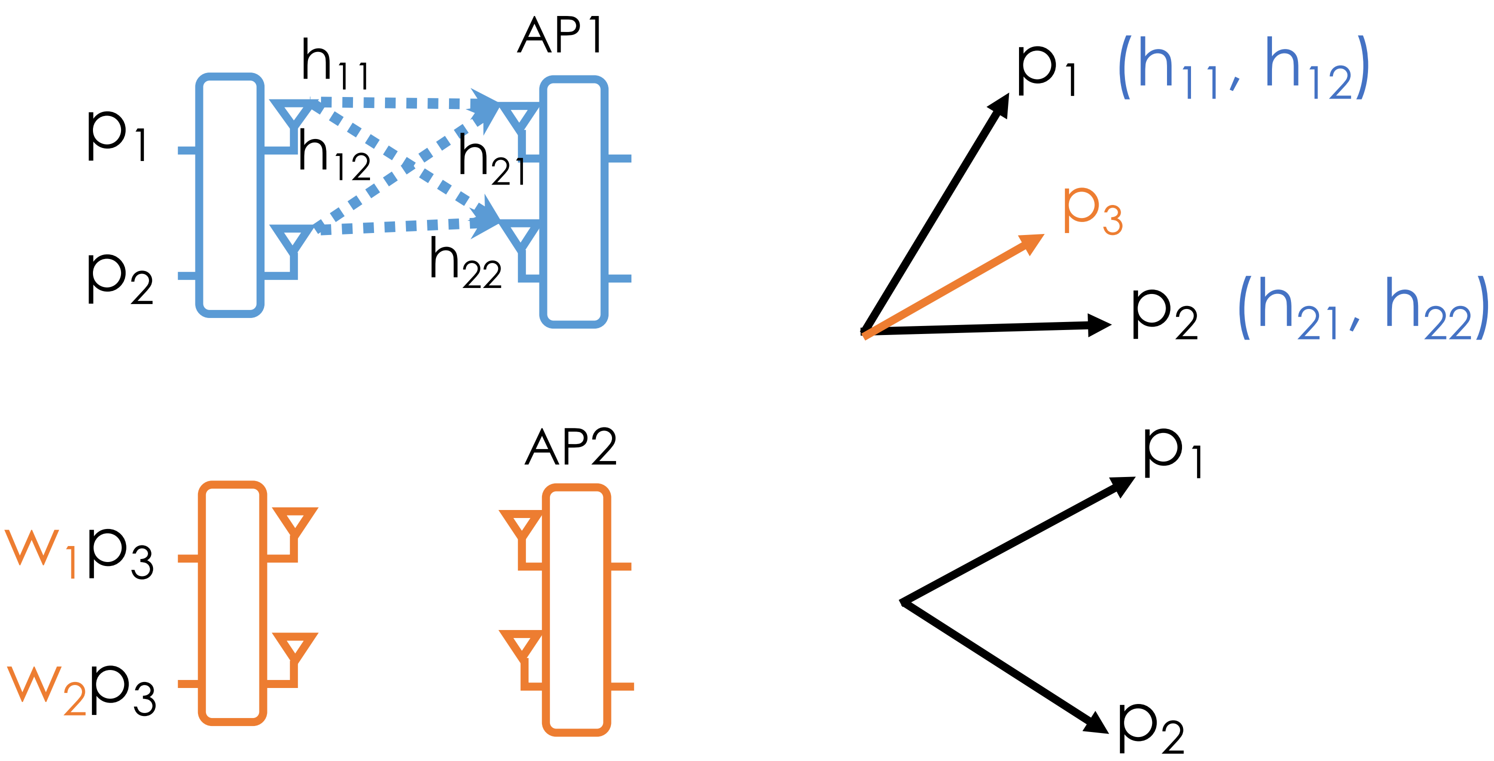
- Align p_3 with p_2 at AP1
- AP1 broadcasts p_1 on Ethernet
- AP2 subtracts/cancels $p_1 \rightarrow$ decodes p_2, p_3

Interference Alignment and Cancellation



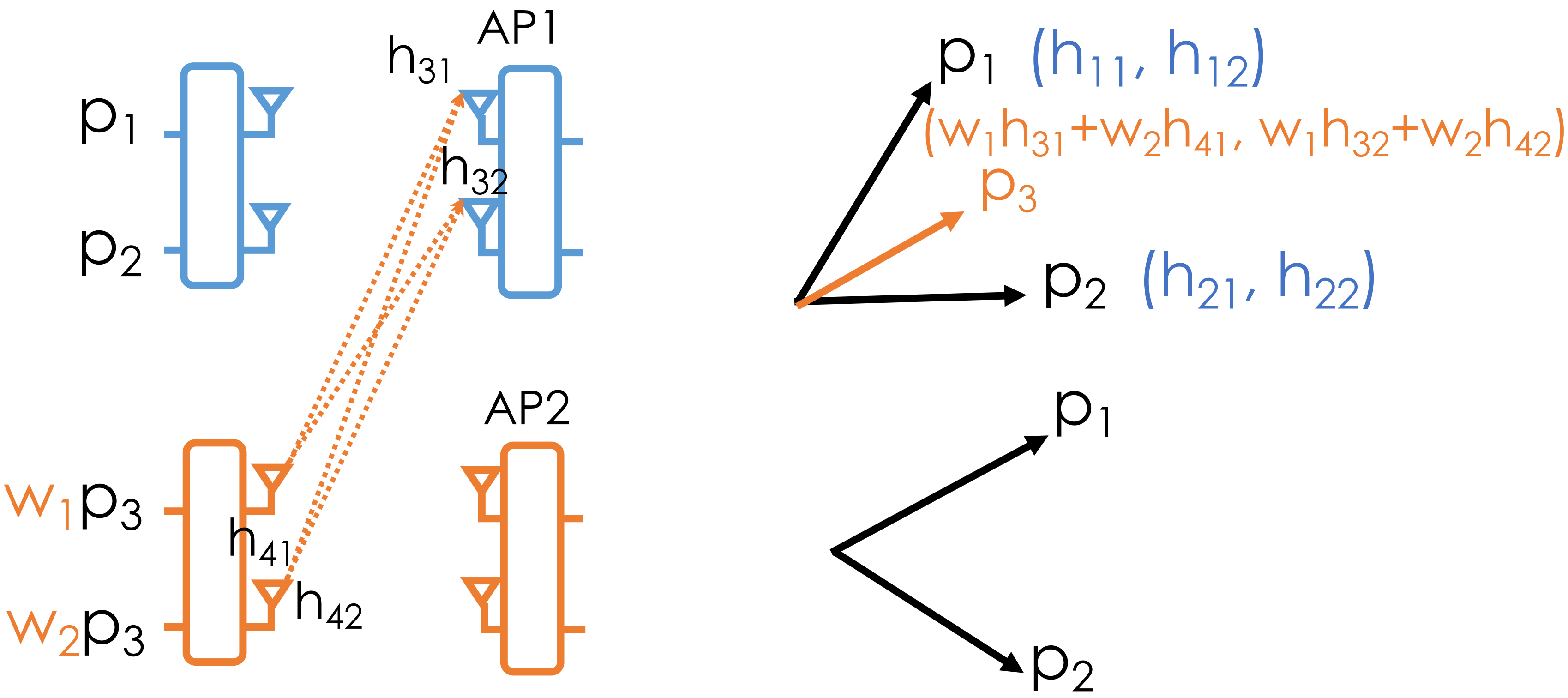
Only forward **1 data packet** through the Ethernet!

How to Align?



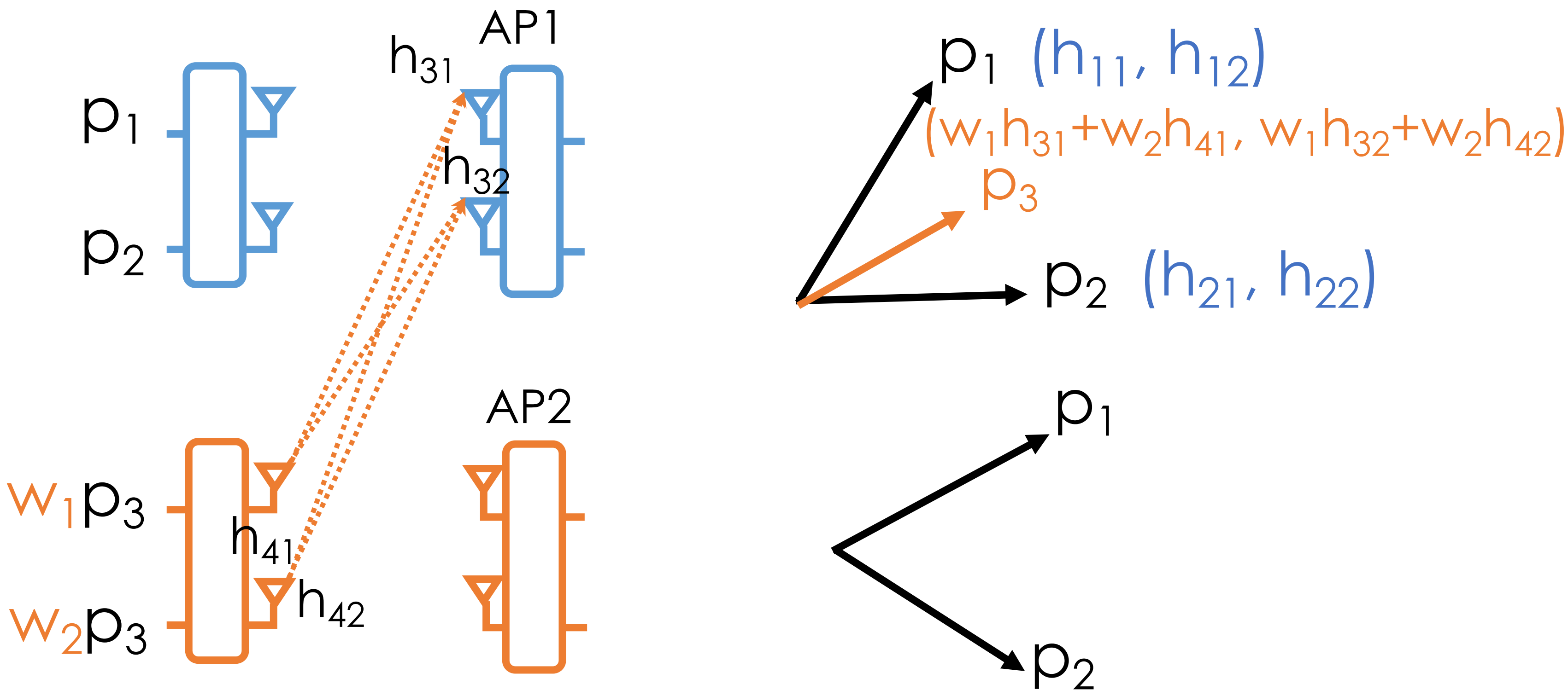
1. Learn the direction we need to align
   – Client 2 aligns p_3 along (h_{21}, h_{22}) at AP1

How to Align?



2. Precode p_3 by (w_1, w_2)
3. AP2 receives p_3 along the direction $(w_1h_{31}+w_2h_{41}, w_1h_{32}+w_2h_{42})$

How to Align?



4. Since AP1 tries to decode p_1, we align the interference p_3 along the direction of p_2
   → Let $(w_1h_{31}+w_2h_{41})/(w_1h_{32}+w_2h_{42})=h_{21}/h_{22}$

Infinite number of solution?
No! **power constraint** $w_1^2+w_2^2=P_{max}$

How to Remove Interference?

- For example, how can AP2 remove the interference from p_1?
- Cannot just subtract the bits of p_1 from the received packet
  - Should subtract interference signals as received by AP2
- How? → Similar to SIC
  - AP2 re-modulates p_1's bits
  - AP2 estimate the channel from client 1 to AP2 and apply the learned channel on the re-modulated signals of p_1
  - Subtract it from the received signal y

How to Generalize to M-Antenna MIMO?

Theorem

In a M- antenna MIMO system, IAC delivers

- *2M concurrent packets on uplink*
- *max{2M-2, 3M/2} concurrent packets on downlink*

e.g., M=2 antennas { 4 packets on uplink
3 packets on downlink

See the paper for the details!