Wireless Communication Systems
@CS.NCTU

Lecture 11: Full-Duplex Communications

Instructor: Kate Ching-Ju Lin (林靖茹)

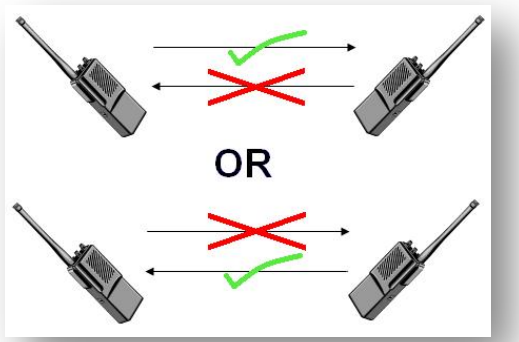
Outline

- What's full-duplex
- Self-Interference Cancellation
- Full-duplex and Half-duplex Co-existence
- Full-duplex relaying

What is Duplex?

- Simplex

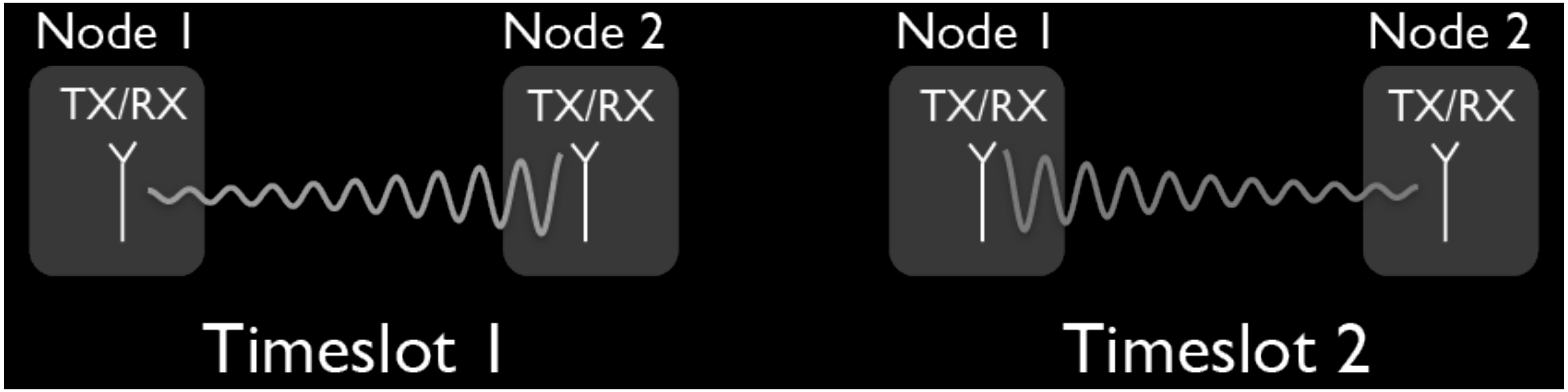
- Half-duplex



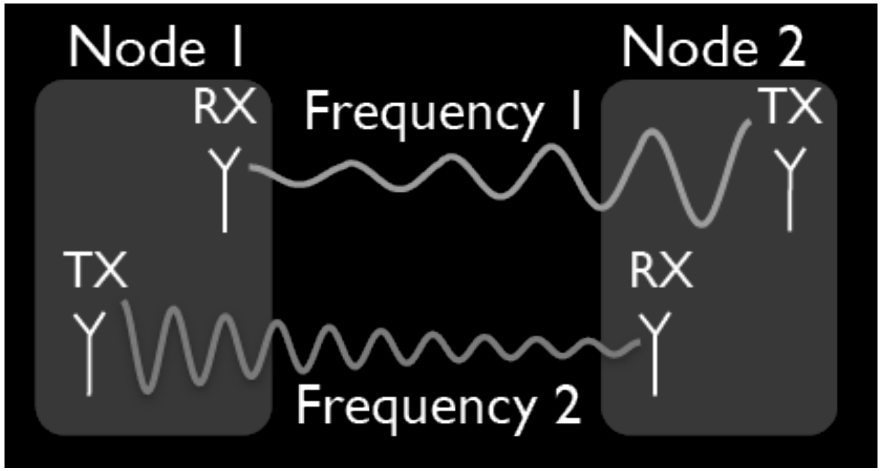
- Full-duplex

How Half-duplex Works?

- Time-division half-duplex

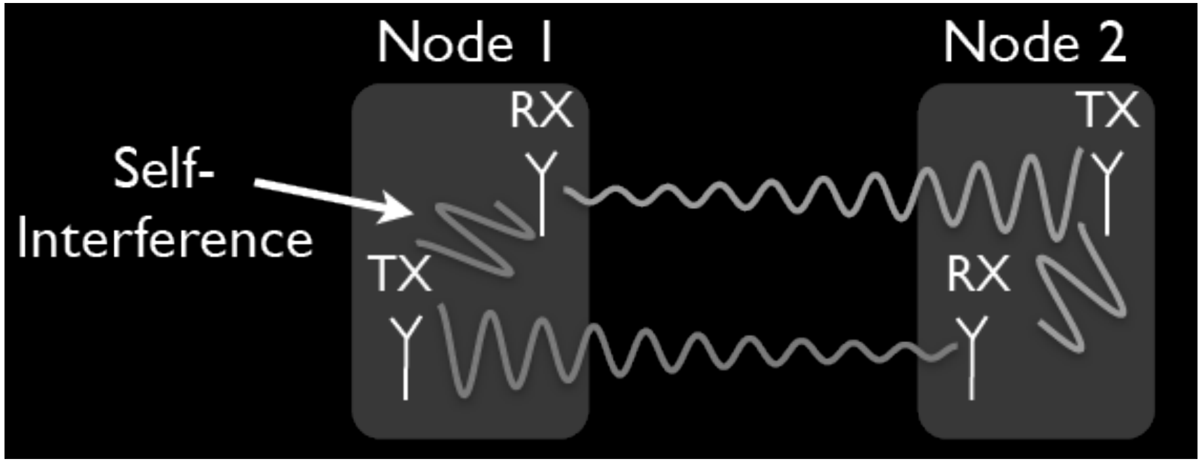


- Frequency-devision half-duplex



Co-Channel (In-band) Full-duplex

Very strong self-interference (~70dB for 802.11)



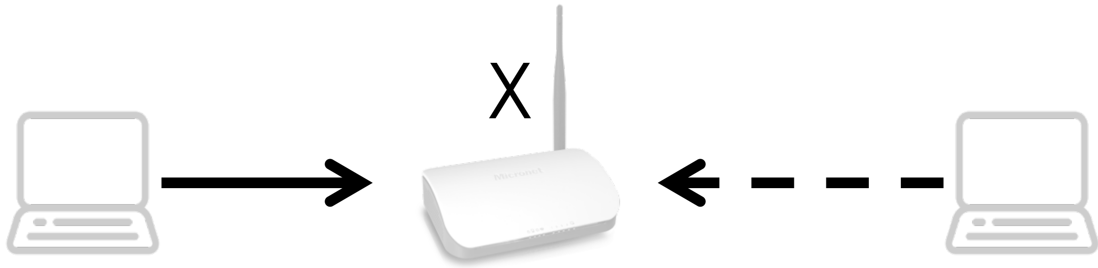
- The transmitted signals will be an interference of the received signals!
- But, we know what we are transmitting → Cancel it!

Benefits beyond 2x Gain

- Can solve some fundamental problems
  - Hidden terminal
  - Primary detection for cognitive radios
  - Network congestion and WLAN fairness
  - Excessive latency in multihop wireless

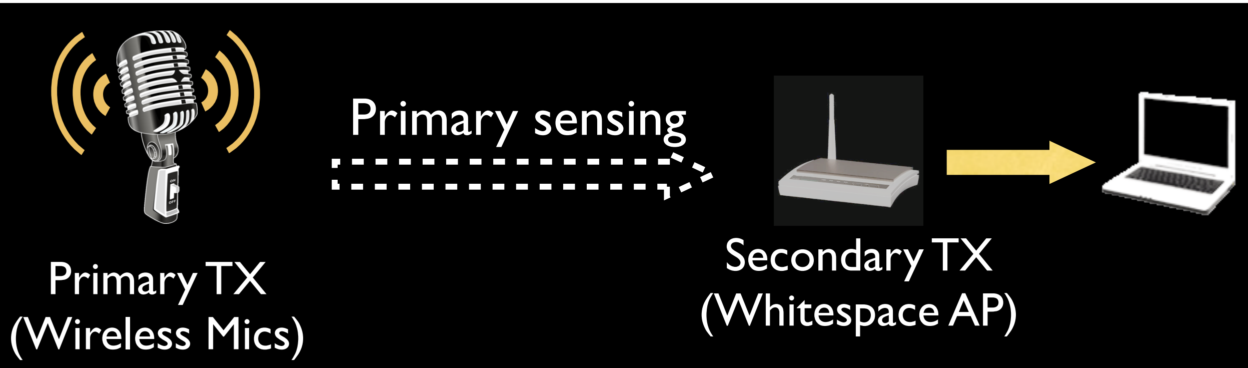
Mitigating Hidden Terminal

- Current networks have hidden terminals
  - CSMA/CA cannot solve this
  - Schemes like RTS/CTS introduce significant overhead



- Full-duplex solves hidden terminals
  - Since both slides transmit at the same time, no hidden terminals exist

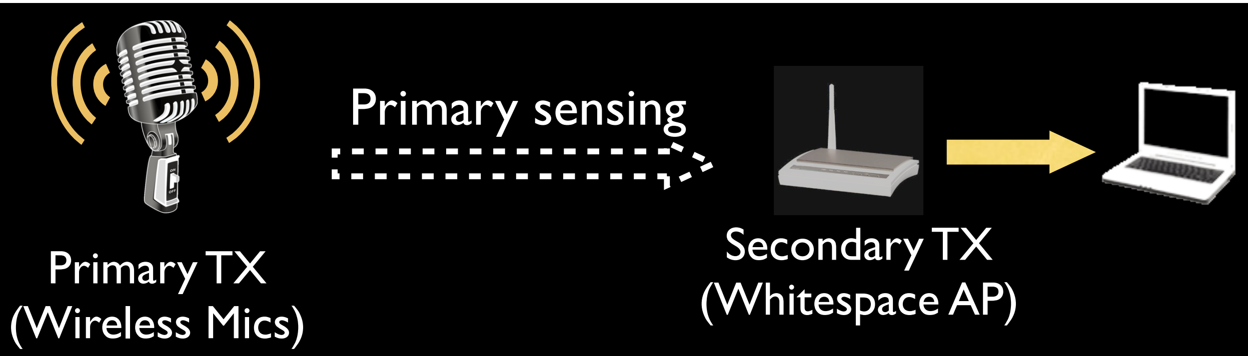
Primary Detection in Whitespaces



Secondary transmitters should sense for primary transmissions before channel use

Traditional nodes may still interfere during transmissions

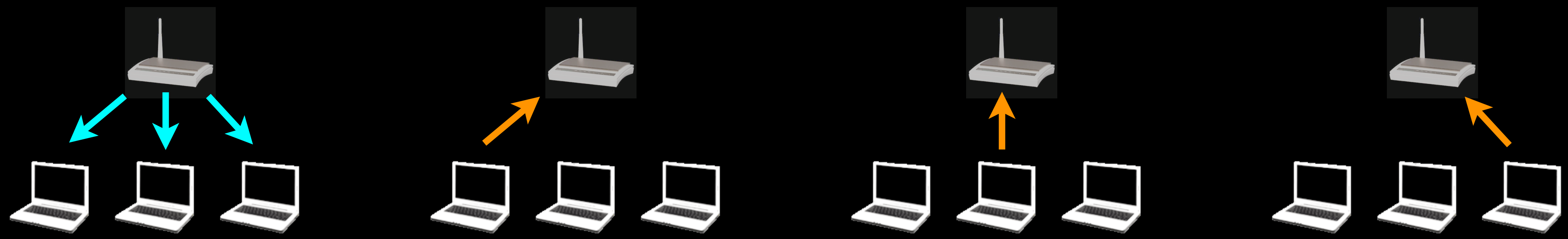
Primary Detection in Whitespaces



Secondary transmitters should sense for primary transmissions before channel use

Full-duplex nodes can sense and send at the same time

Network Congestion and Fairness

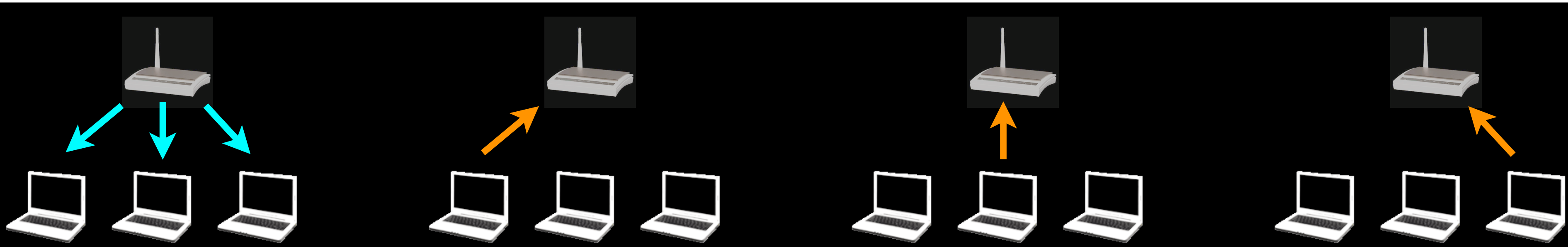


Without full-duplex:

- 1/n bandwidth for each node in network, including AP

Downlink Throughput = 1/n   Uplink Throughput = (n-1)/n

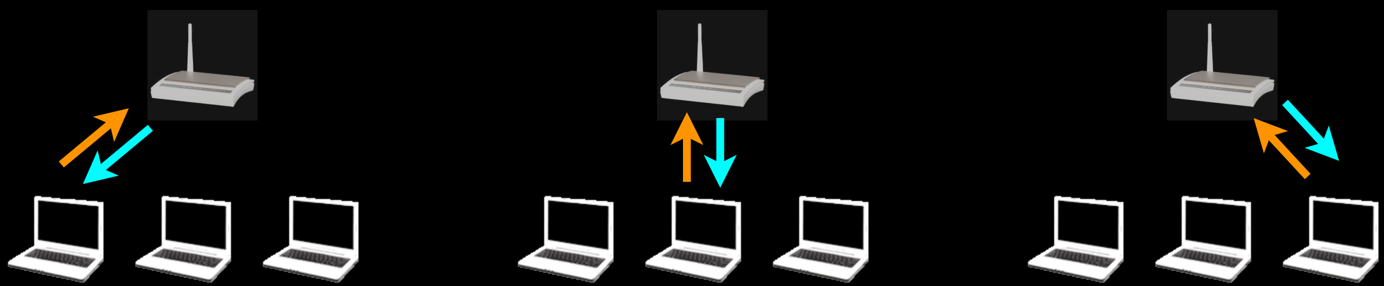
Network Congestion and Fairness



Without full-duplex:

- 1/n bandwidth for each node in network, including AP

Downlink Throughput = 1/n   Uplink Throughput = (n-1)/n



With full-duplex:

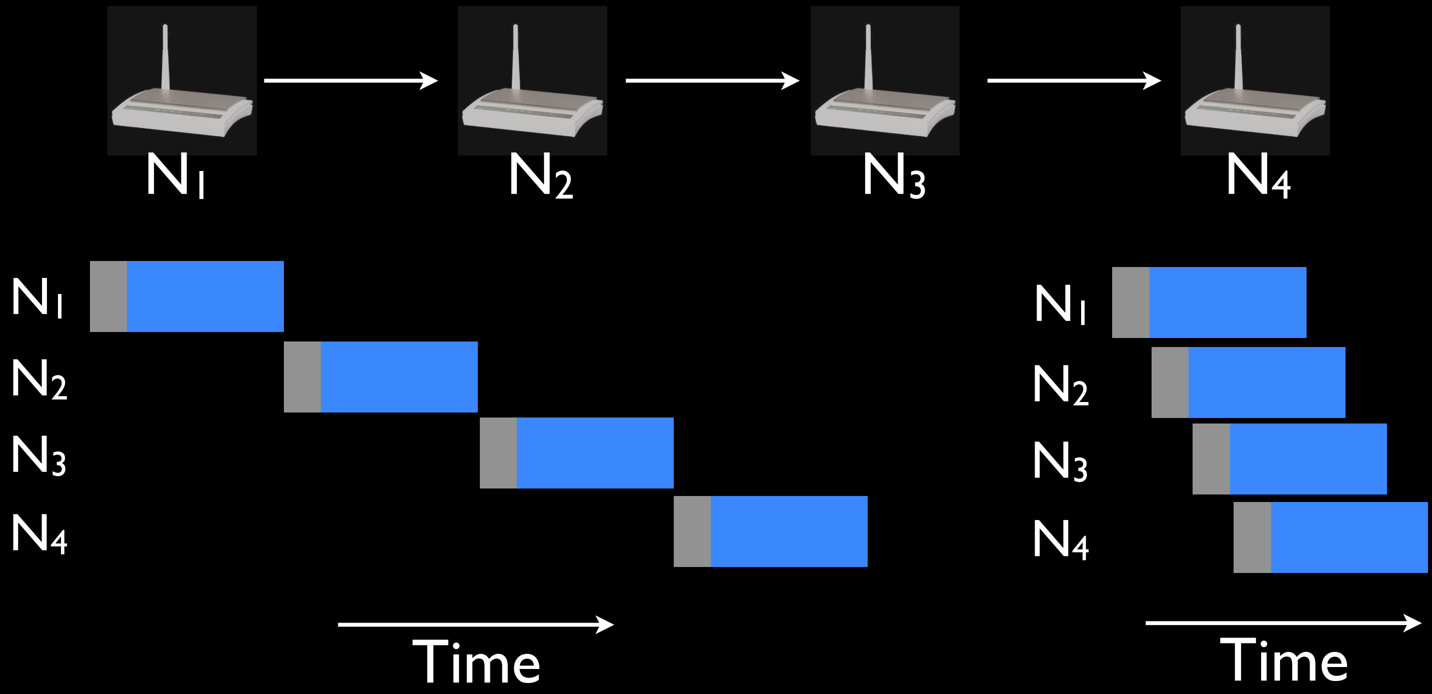
- AP sends and receives at the same time

Downlink Throughput = 1   Uplink Throughput = 1

Reducing Round-Trip Time

Long delivery and round-trip times in multi-hop networks

Solution: Wormhole routing

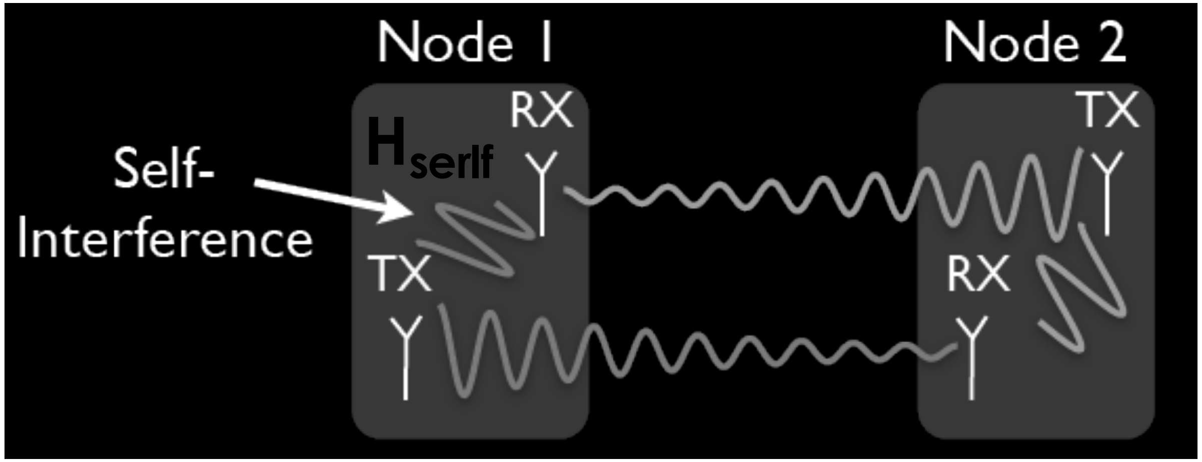


Half-duplex Full-duplex

Outline

- What's full-duplex
- Self-Interference Cancellation
- Full-duplex and Half-duplex Co-existence
- Full-duplex relaying

Self-Interference Cancellation



$$Y = Hx + \cancel{H_{self} x_{self}} + n$$

Wanted signals

Unwanted
self-interference

Challenge1: self-interference is much stronger than wanted signals, i.e., $|H_{self}|^2 \gg |H|^2$

Challenge 2: hard to learn real H_{self}

Self-Interference Cancellation

- **Analog interference cancellation**
  - RF cancellation (~25dB reduction)
  - Active
- **Digital interference cancellation**
  - Baseband cancellation (~15dB reduction)
  - Active
- **Antenna cancellation**
  - Passive

What Makes Cancellation Non-Ideal?

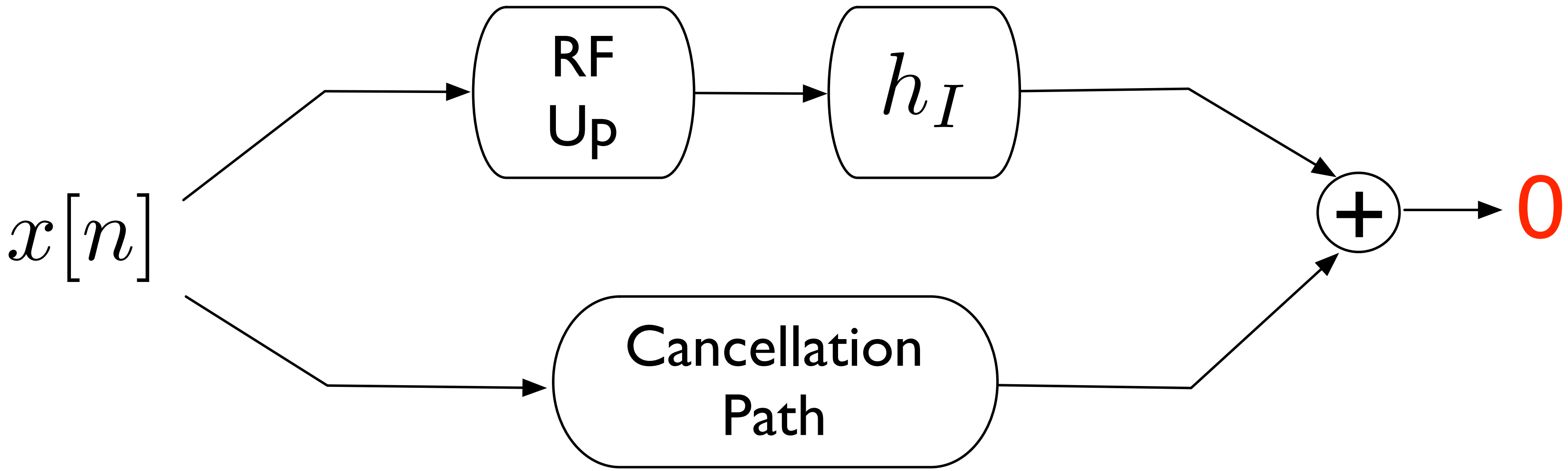
- Transmitter and receiver phase noise
- LNA (low-noise amplifier) and Mixer noise figure
- Tx/Rx nonlinearity
- ADC quantization error
- Self-interference channel

Noise figure (NF) is the measure of degradation of SNR caused by components in a RF chain

Analog Cancellation

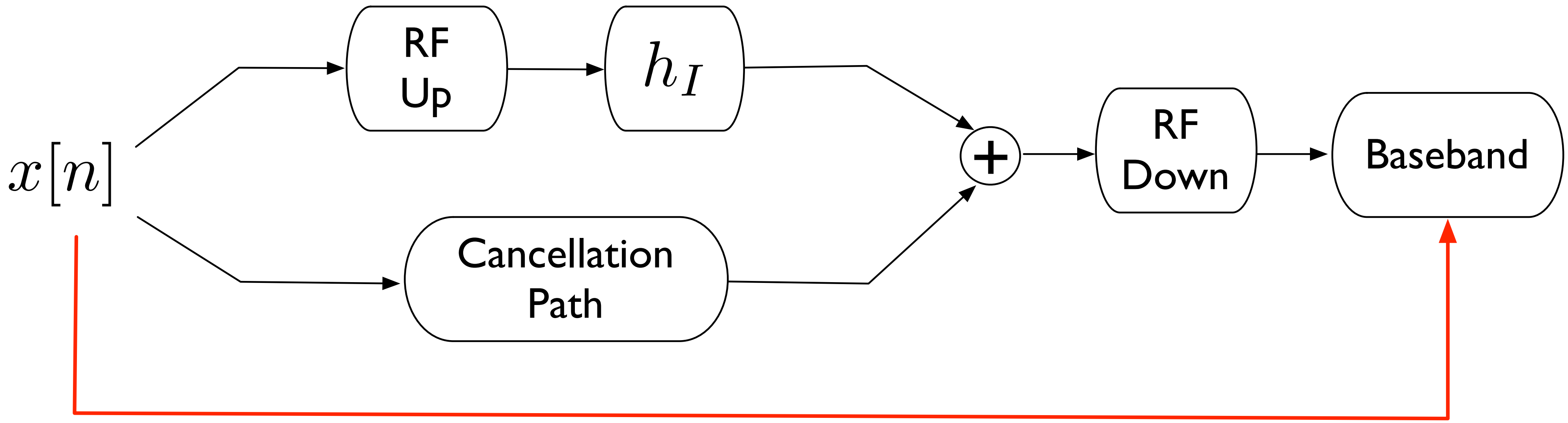
- Why important?
  - Before digital cancellation, we should avoid saturating the Low Noise Amplifier and ADC
  - Eg., Tx power = 20 dBm and LNA with a saturation level -25dB → at least need -45 dB of analog cancellation
- Major drawback
  - Need to modify the radio circuitry
  - Should be added after RF down-converter but before the analog-to-digital converter, usually not accessible

Analog Cancellation



- Objective is to achieve exact 0 at the Rx antenna
- Cancellation path = negative of interfering path
- These techniques need analog parts

Digital Cancellation



- Cancel interference at baseband
- Conceptually simpler – requires no new "parts"
- Useless if interference is too strong (ADC bottleneck)

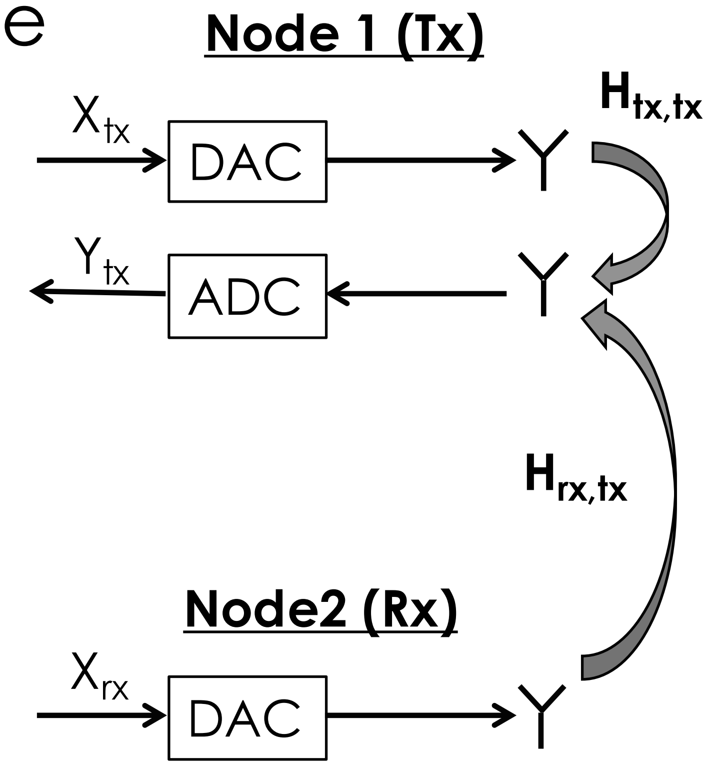
How Digital Cancellation Works?

- Assume only Tx is transmitting
  → Tx receives self-interference

- Estimate the self-channel

- When Rx starts transmitting
  → Tx now receives

- Cancel self-interference by



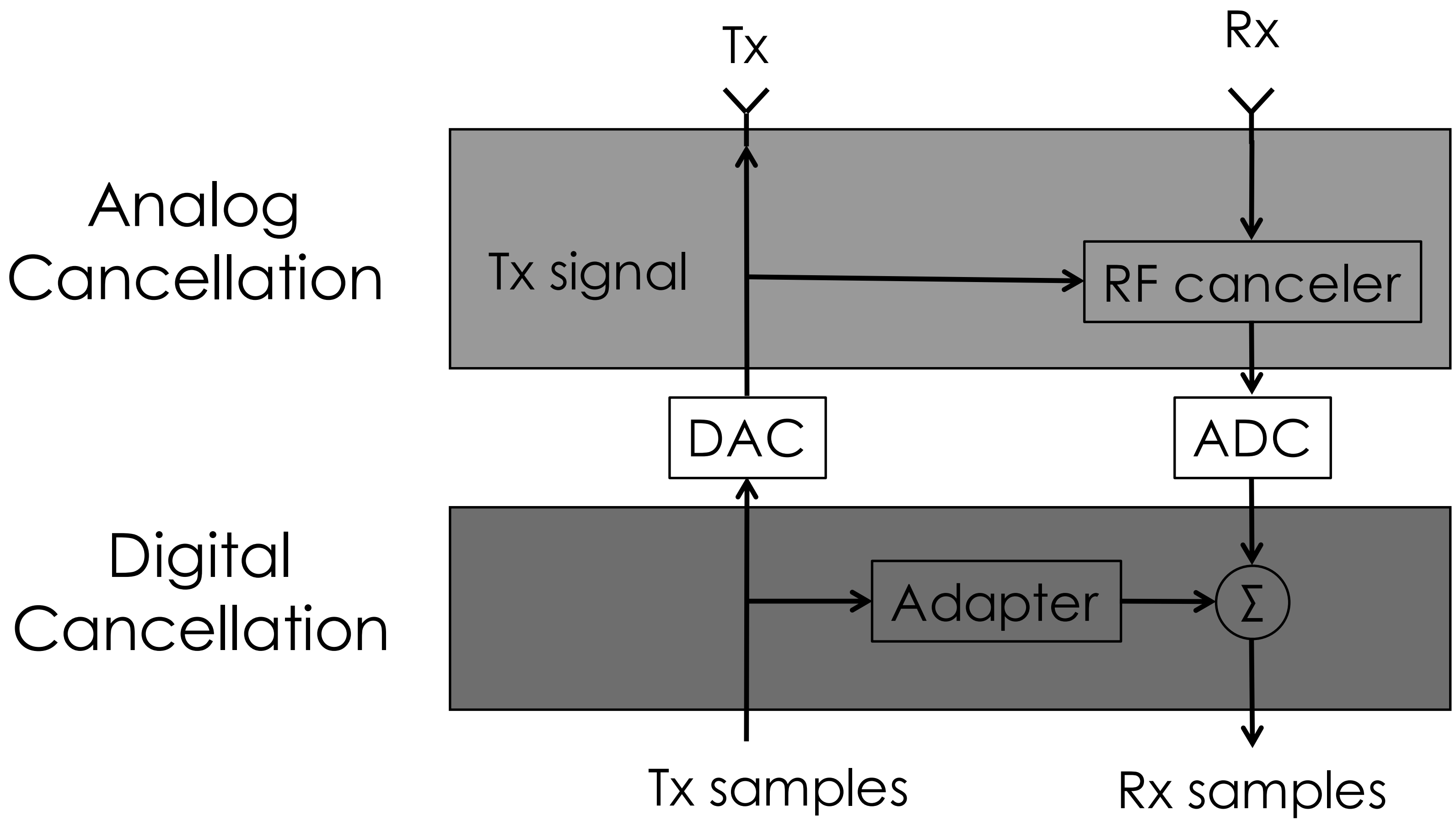
Digital Cancellation for OFDM

- Cancel for each subcarrier separately

$$Y_{\text{rx}}[k] \approx Y[k] - \hat{H}[k]_{\text{tx,tx}} X_{\text{tx}}[k] = H_{\text{rx,tx}}[k] X_{\text{rx}}[nk] + n$$

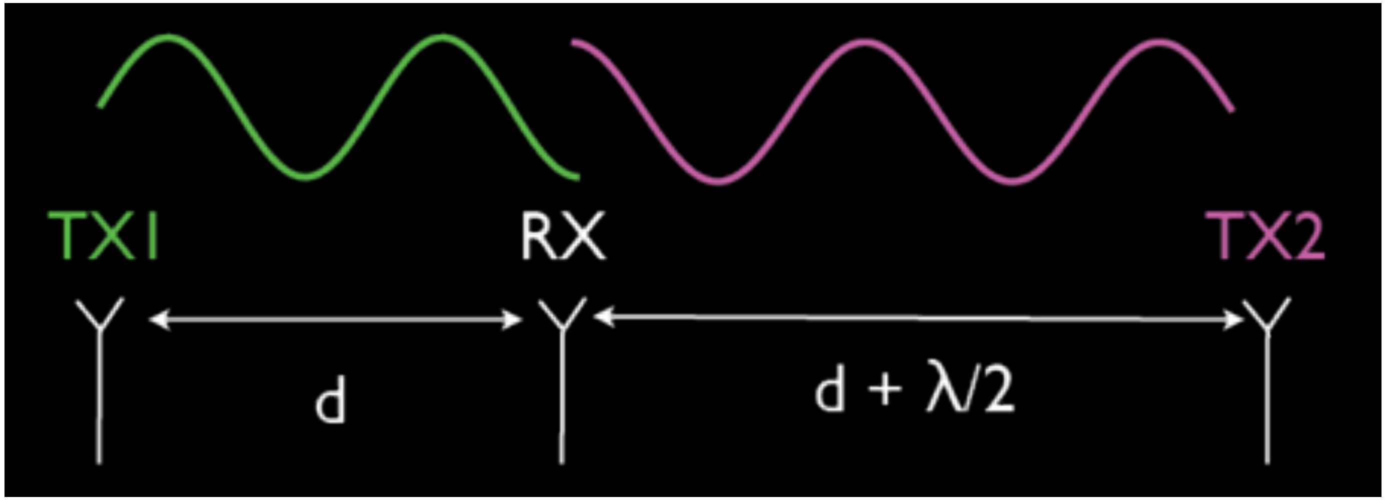
- But, can't just perform cancellation in the frequency domain → Why
  - Hard to do iFFT → Cancellation → FFT in real-time
- How can we do digital cancellation for each subcarrier in the time-domain?
  - See FastForward [Sigcomm'14]

Combine RF/Digital Cancellation



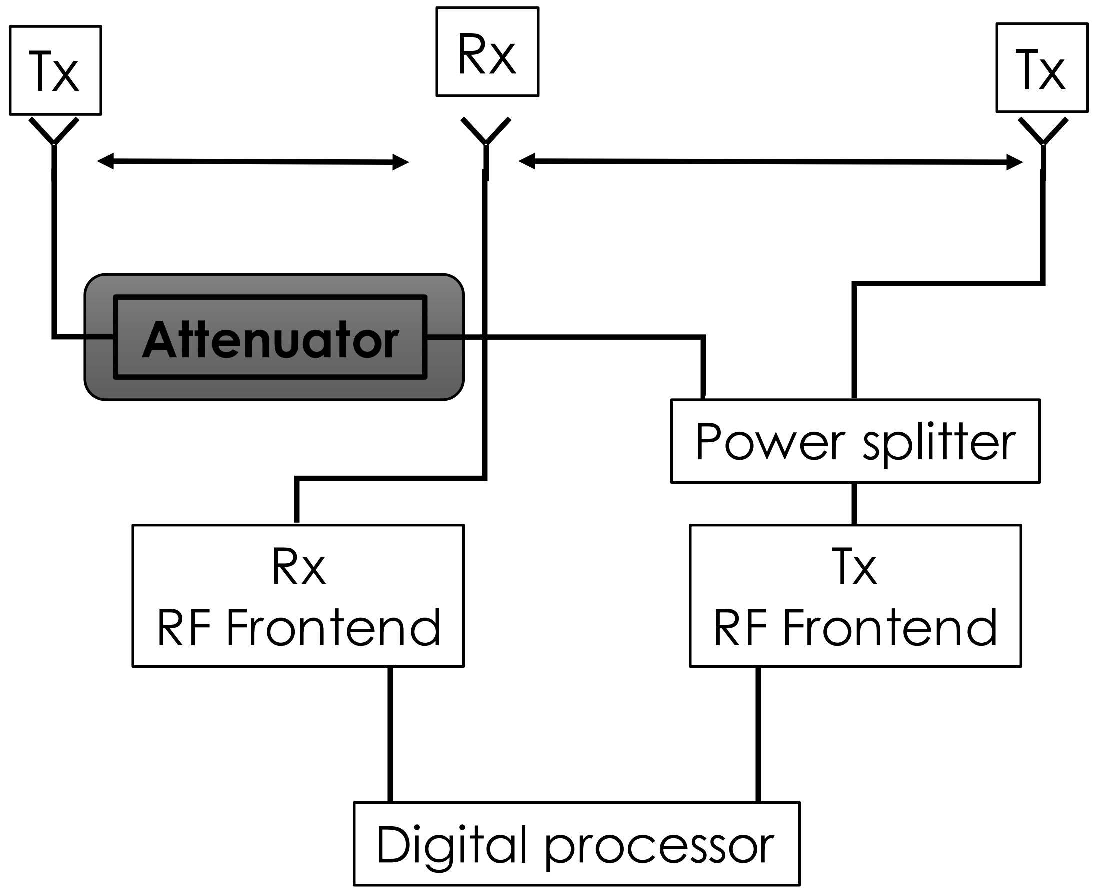
Antenna Cancellation

- Separate the antennas such that the two signals become deconstructive
  - The distance different = λ/2

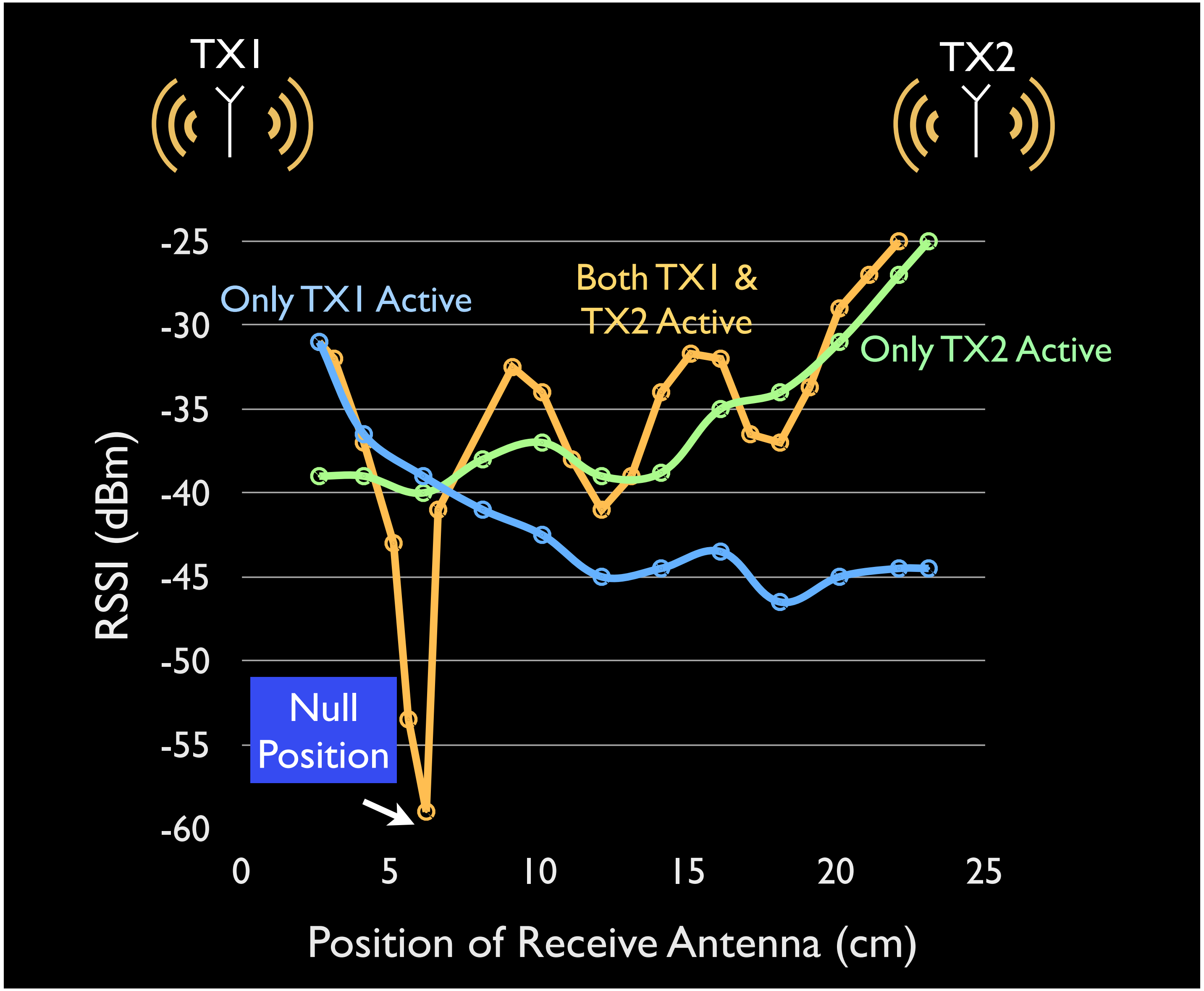


~30dB self-interference cancellation combined with analog/digital cancellation
→ 70 dB

Antenna Cancellation: Block Diagram



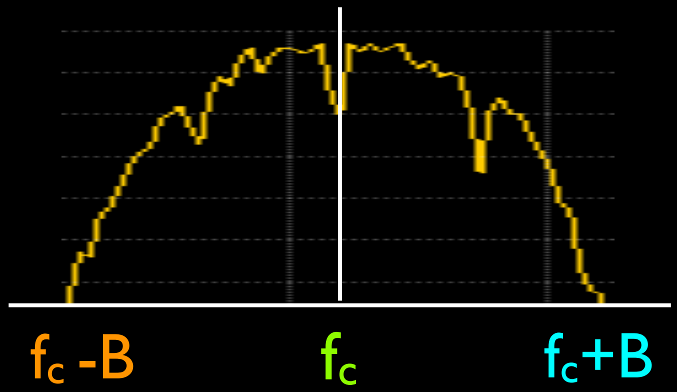
Performance

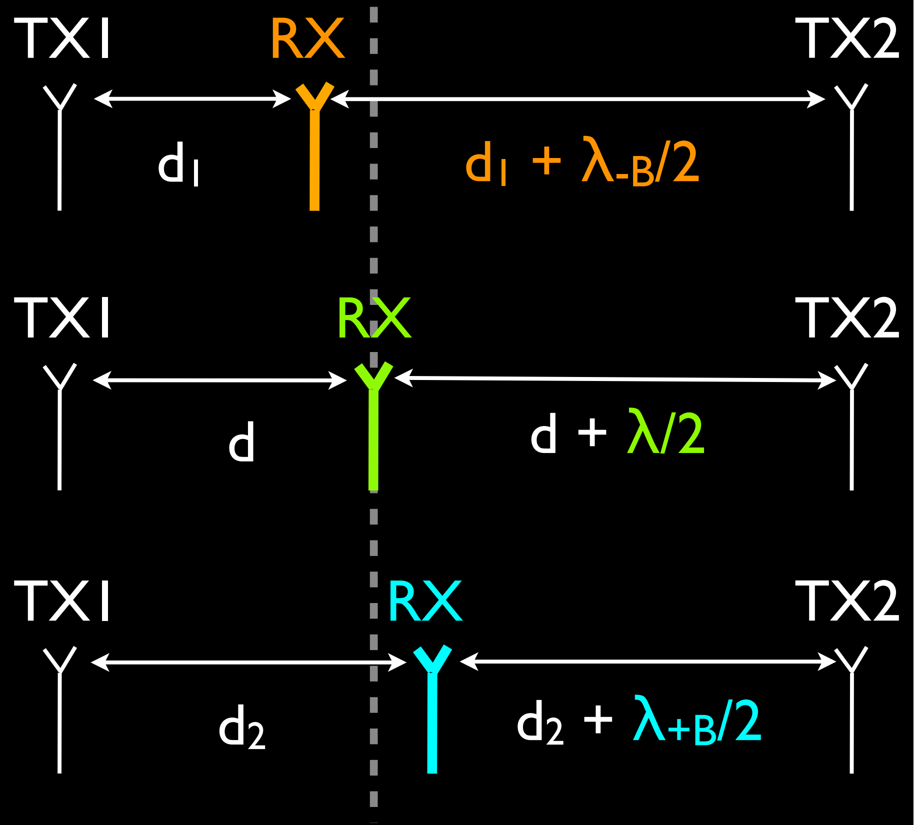


Impact of Bandwidth

A λ/2 offset is precise for one frequency

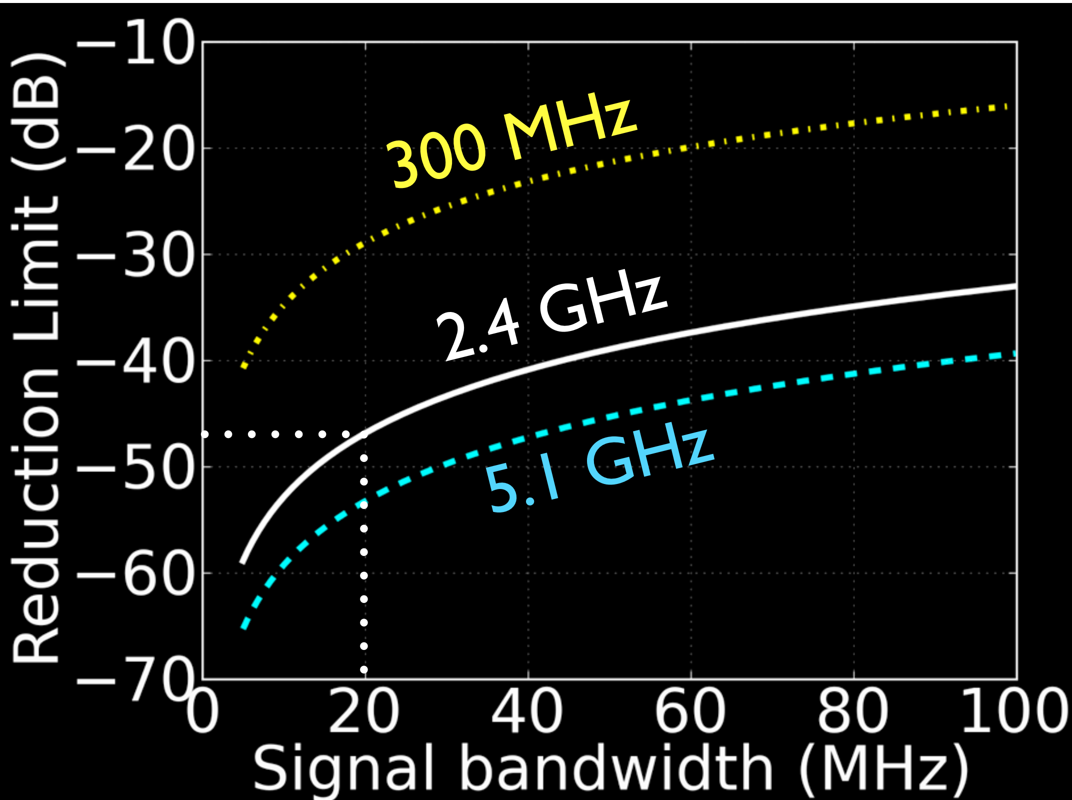
not for the whole bandwidth





WiFi (2.4G, 20MHz) => ~0.26mm precision error

Bandwidth vs. SIC Performance

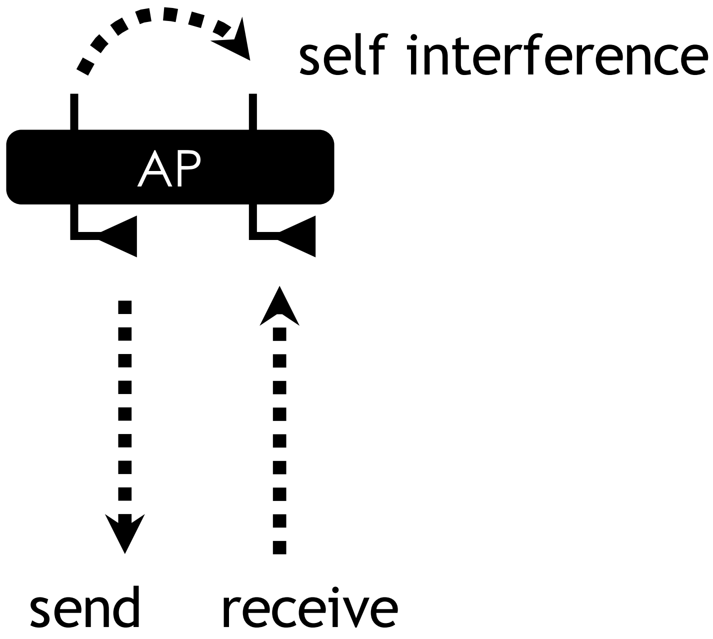


- WiFi (2.4GHz, 20MHz): Max 47dB reduction
- Bandwidth ⬆ => Cancellation ⬇
- Carrier Frequency ⬆ => Cancellation ⬆

Outline

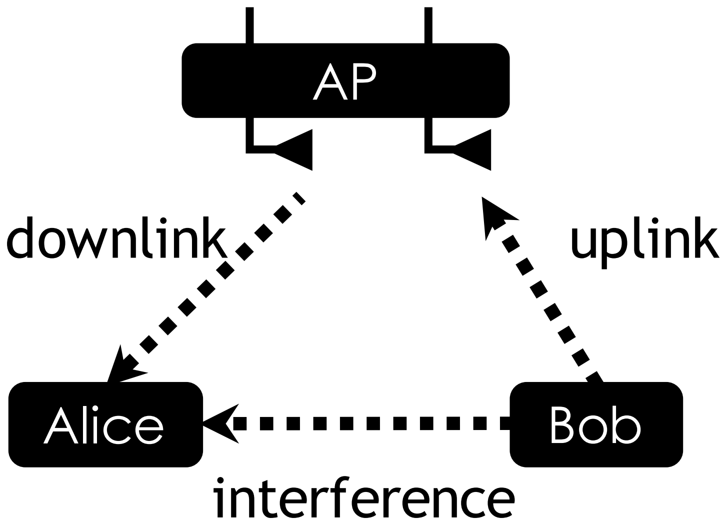
- What's full-duplex
- Self-Interference Cancellation
- Full-duplex and Half-duplex Co-existence
- Full-duplex relaying

Full-Duplex Radios



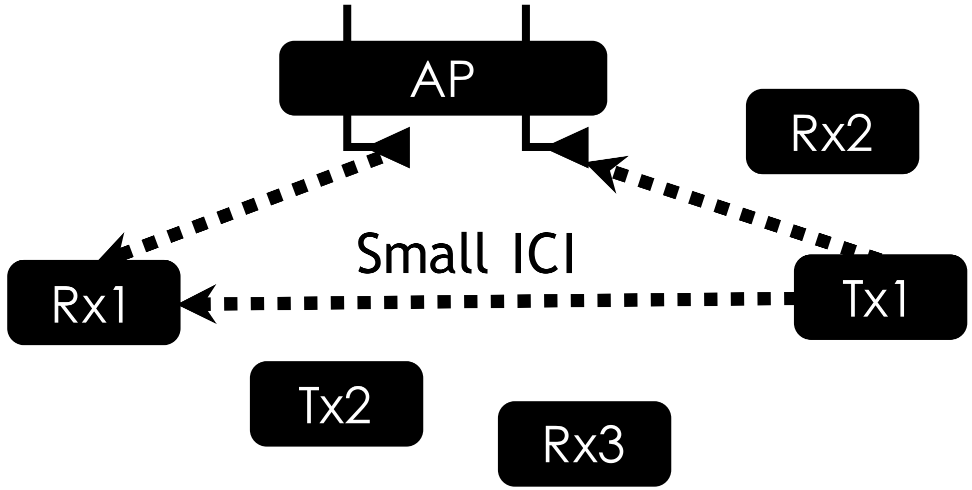
- Transmit and receive simultaneously in the same frequency band
- Suppress self-interference (SI) [Choi et al. 2010, Bharadia et al. 2013]

Three-Node Full-Duplex



- Commodity thin clients might only be half-duplex
- Inter-client interference (ICI)
  - Uplink transmission interferes downlink reception

Access Control for 3-Node FD



- ICI might degrade the gain of full-duplex
  - Appropriate client pairing is required
  - Always enabling full-duplex may not good due to inter-client interference
  - Switch adaptively between full-duplex and half-duplex

Existing Works

- Only allow hidden nodes to enable full-duplex [Sahai et al. 2011]
  - Favor only part of clients, e.g., hidden nodes
- Pair clients based on historical transmission success probability [Singh et al. 2011]
  - Statistics takes time and might not be accurate due to channel dynamics
- Schedule all the transmissions based on given traffic patterns [Kim et al. 2013]
  - Need centralized coordinator and expensive overhead of information collection

Our Proposal: Probabilistic-based MAC

- Flexible adaptation
  - Adaptively switch between full-duplex and half-duplex
- Fully utilizing of full-duplex gains
  - Assign a pair of clients a probability of full-duplex access
  - Find the probabilities so as to maximize the expected overall network throughput
- Distributed random access
  - Clients still contend for medium access based on the assigned probability in a distributed way

Candidate Pairing Pairs

- Full-duplex pairs
  - Only allows those with both clients with non-negligible rates
  - $\mathcal{C}_{\text{full}} \triangleq \{(i,j) : i,j \in \mathcal{N},\ i \neq j, r_d^{(i,j)}, r_u^{(i,j)} > \epsilon\}$
- Half-duplex *virtual* pairs
  - Let '0' denote the index of a virtual empty node
  - $\mathcal{C}_{\text{half}} \triangleq \{(i,j) : i = 0 \text{ or } j = 0, r^{(i,j)} > \epsilon\}$
- All candidate pairs
  - $\mathcal{C} \triangleq \mathcal{C}_{\text{full}} \cup \mathcal{C}_{\text{half}}$

Assign each pair a probability $p^{(i,j)}$

Linear Programming Model

$$\mathcal{P}_1: \quad \max \sum_{(i,j)\in\mathcal{C}} p^{(i,j)} r^{(i,j)}$$

Expected total rate

$$\text{subject to} \quad \sum_{j\in\{j:(i,j)\in\mathcal{C}\}} p^{(i,j)} \geq \eta_d^{(i)}, \forall i\in\mathcal{N}$$

Downlink fairness

$$\sum_{i\in\{i:(i,j)\in\mathcal{C}\}} p^{(i,j)} \geq \eta_u^{(j)}, \forall j\in\mathcal{N}$$

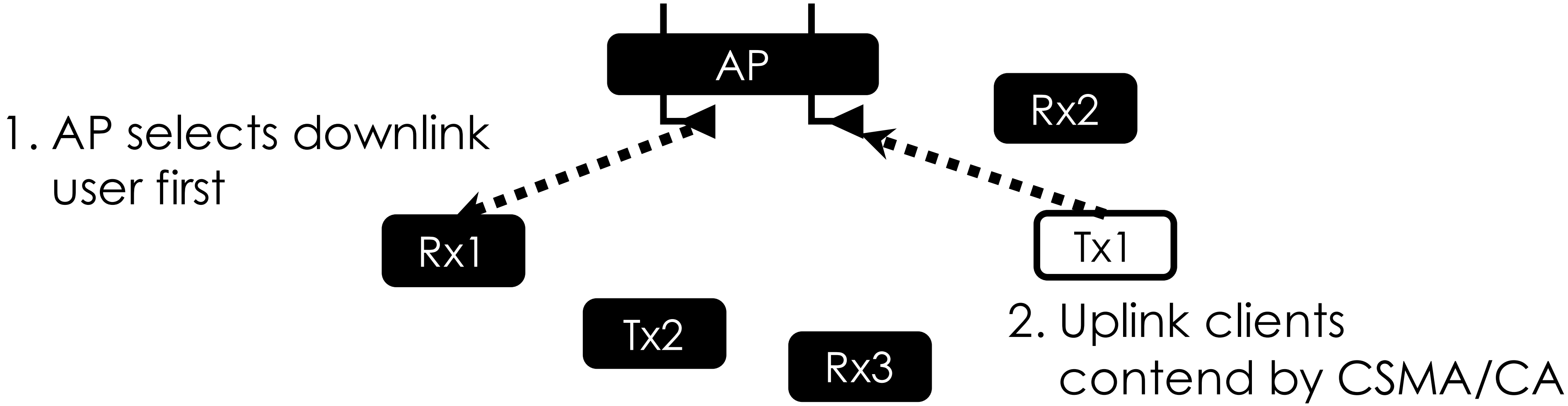
Uplink fairness

$$\sum_{(i,j)\in\mathcal{C}} p^{(i,j)} = 1$$

Sum probability

$$\text{variables:} \quad p^{(i,j)} \in \mathbb{R}_{\geq 0}, \forall (i,j)\in\mathcal{C}$$

Probabilistic Contention



- AP selects downlink user *i* with probability

$$p_d^{(i)} = \sum_{j \in \{j:(i,j) \in \mathcal{C}\}} p^{(i,j)}$$

- Given downlink user *i*, uplink users adjust its priority by changing its contention window to

$$CW_u^{(i,j)} = \lceil 1/p_u^{(i,j)} \rceil, \text{ where } p^{(i,j)}/p_d^{(i)}$$

Outline

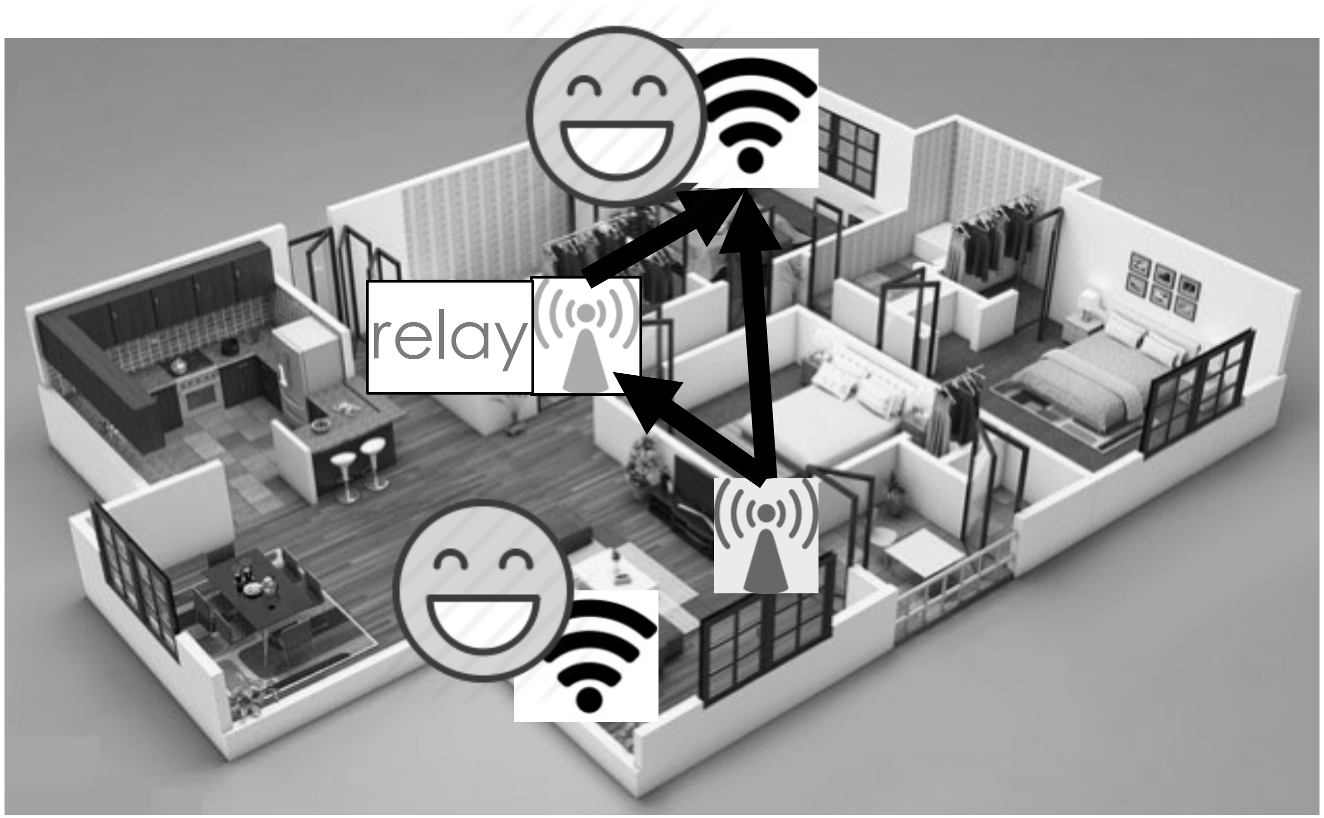
- What's full-duplex
- Self-Interference Cancellation
- Full-duplex and Half-duplex Co-existence
- Full-duplex relaying

Today's Wireless Networks

- Ideally, 802.11ac and 802.11n support up to 780 Mb/s and 150 Mb/s, respectively
- In reality, signals experience propagation loss

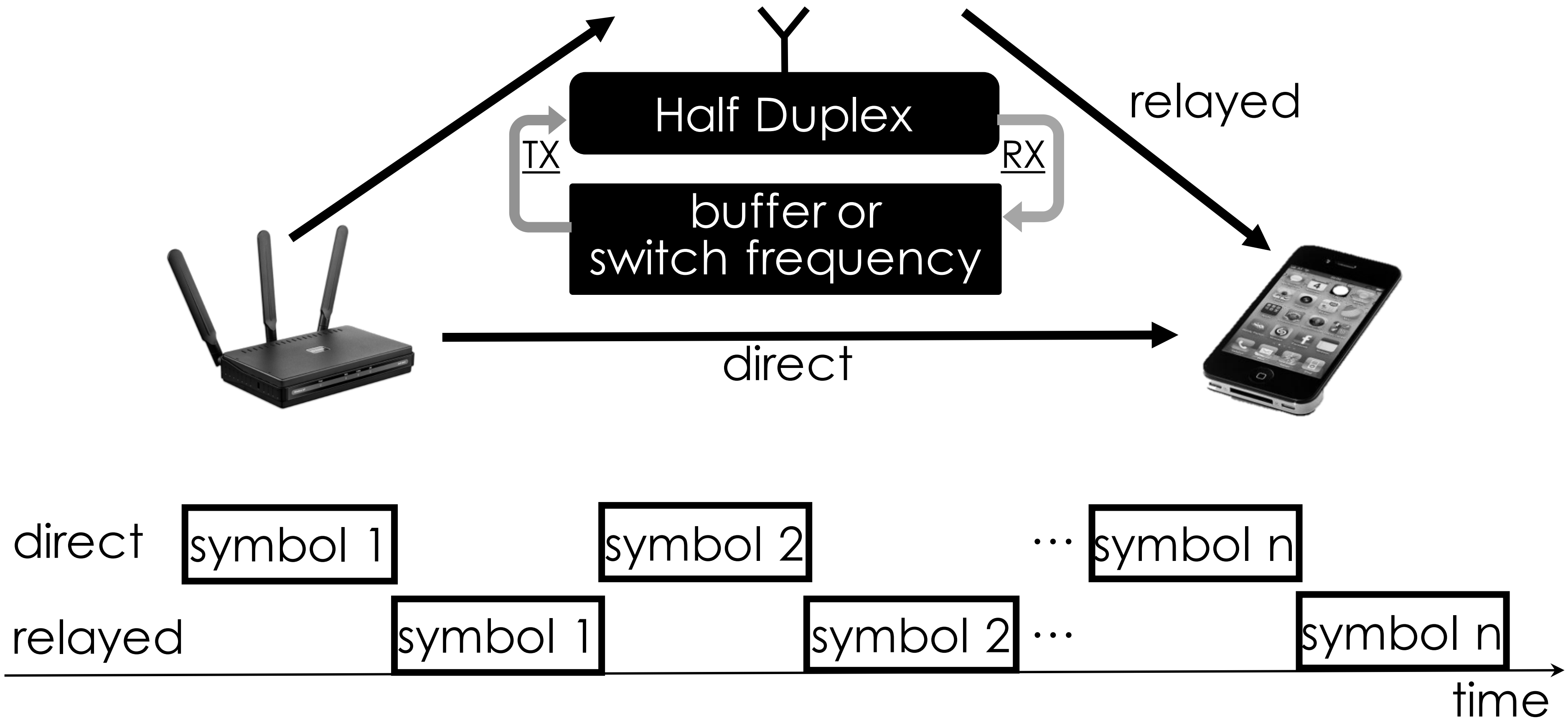
What Can We Do?

- Increase capacity and coverage using relay



Traditional Half-Duplex Relaying

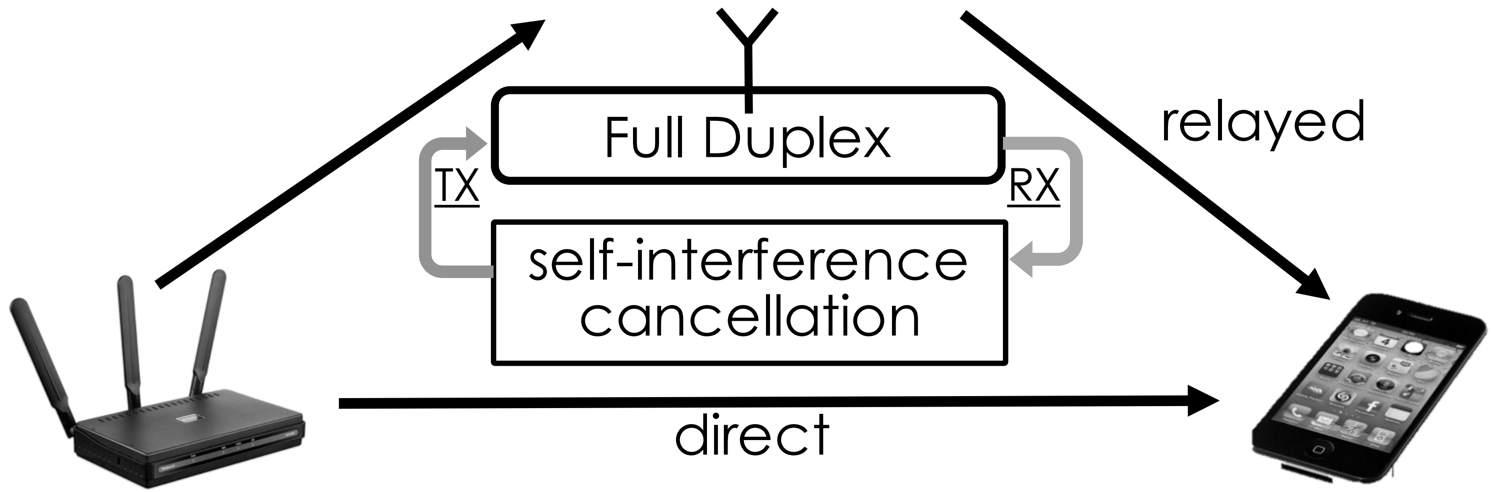
TX and RX in a time/frequency division manner

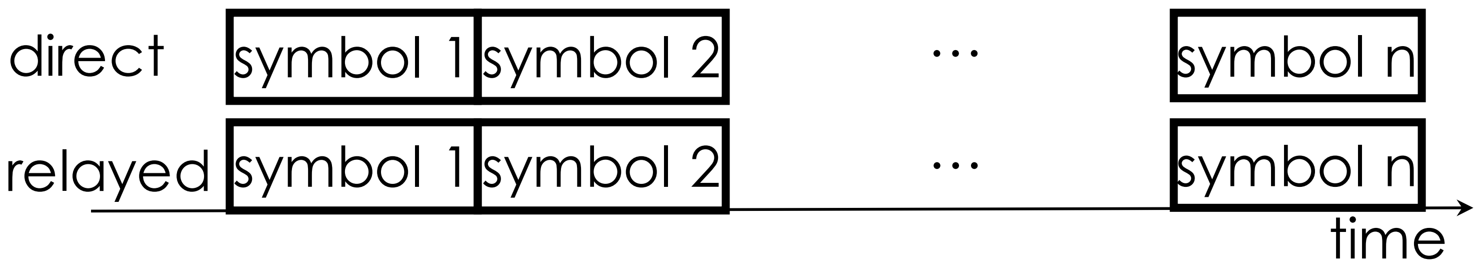


Improve SNR, but also halve the bandwidth

Full-Duplex Relaying!

Simultaneous TX and RX on the same frequency

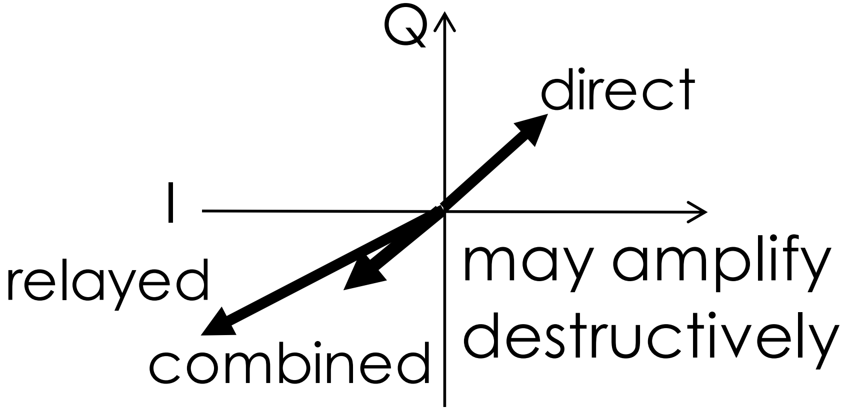




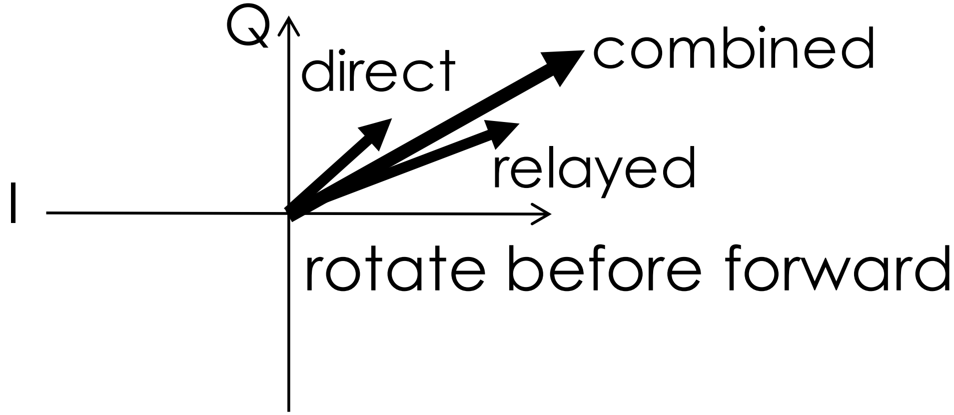
Improve SNR without halving the bandwidth

[FastForward, SIGCOMM'14]

1. Amplify-and-forward or Construct-and-forward



<u>rather decrease the SNR</u>



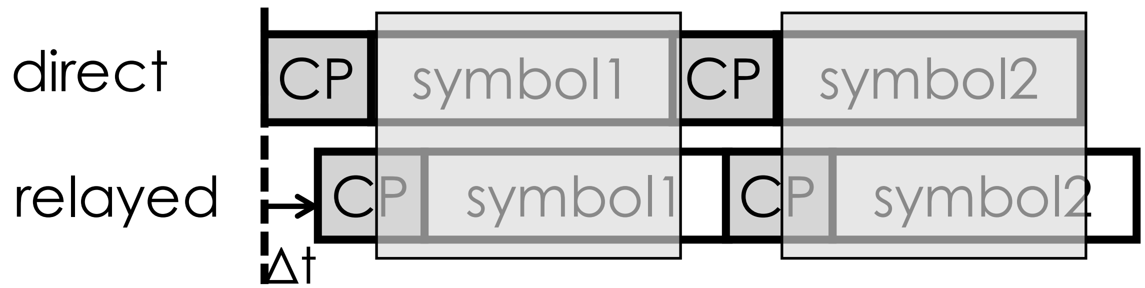
<u>amplify constructively</u>

2. Demodulate-and-forward

[DelayForward, MobiHoc'16]

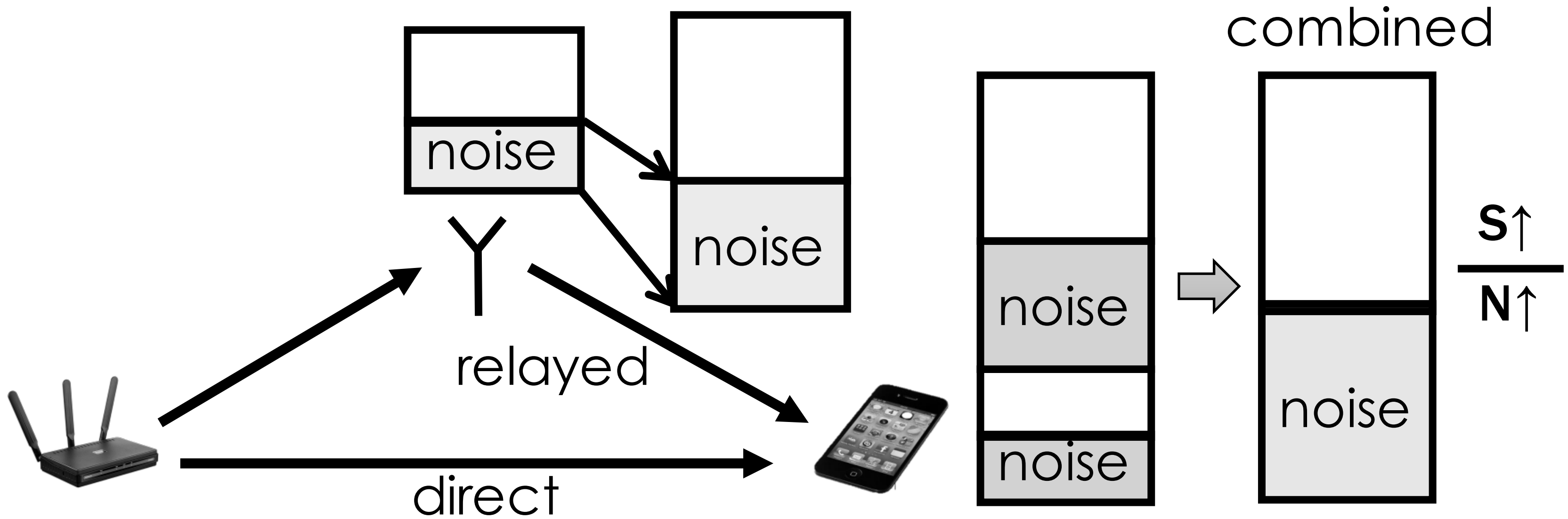
Pros and Cons of Amplify-and-Forward

✔Negligible processing delay at relay



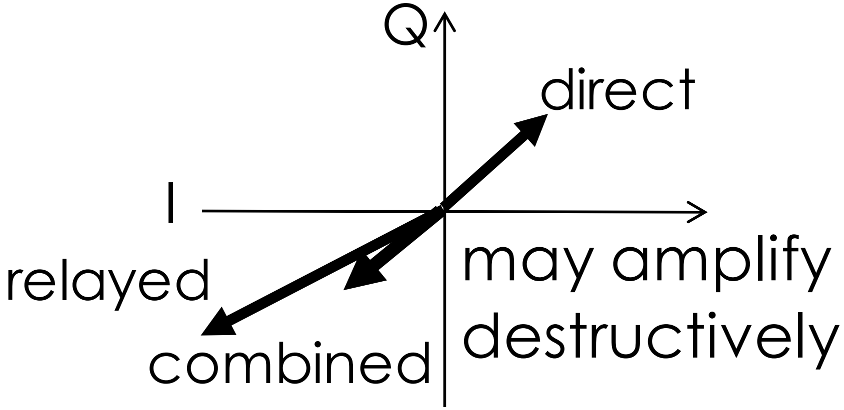
Still decodable with OFDM

✘ Also amplifying the noise at the relay

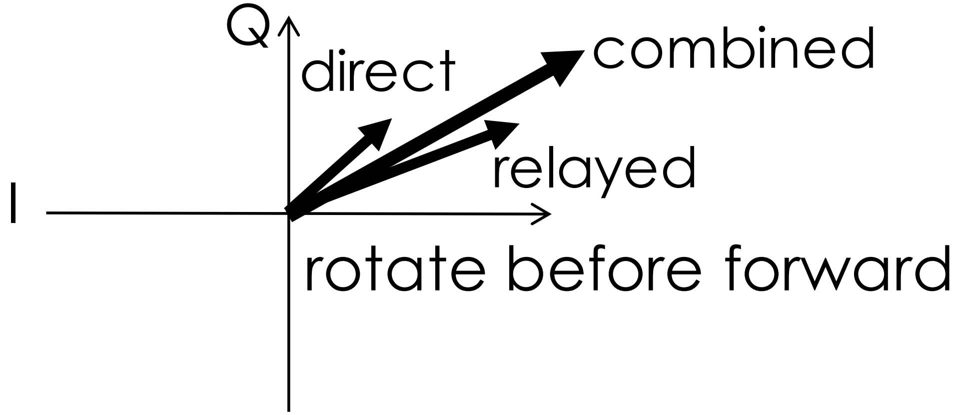


[FastForward, SIGCOMM'14]

1. Amplify-and-forward or Construct-and-forward

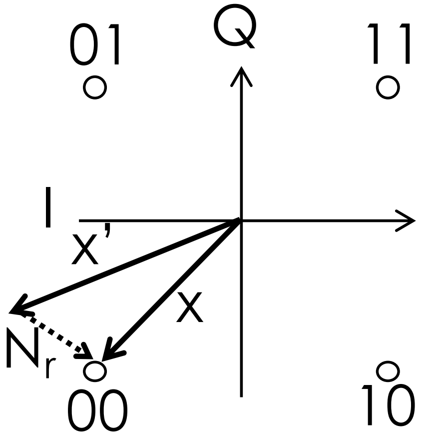


rather decrease the SNR



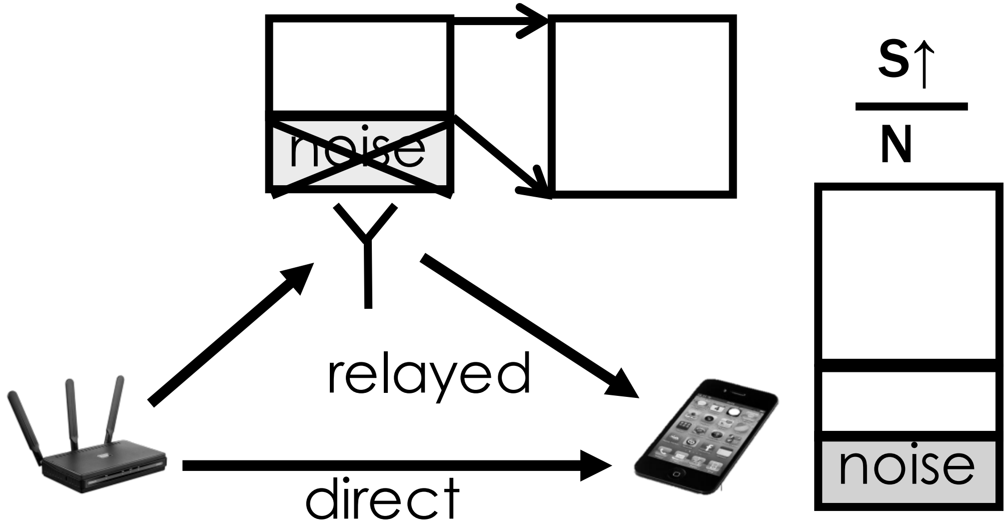
amplify constructively

2. Demodulate-and-forward



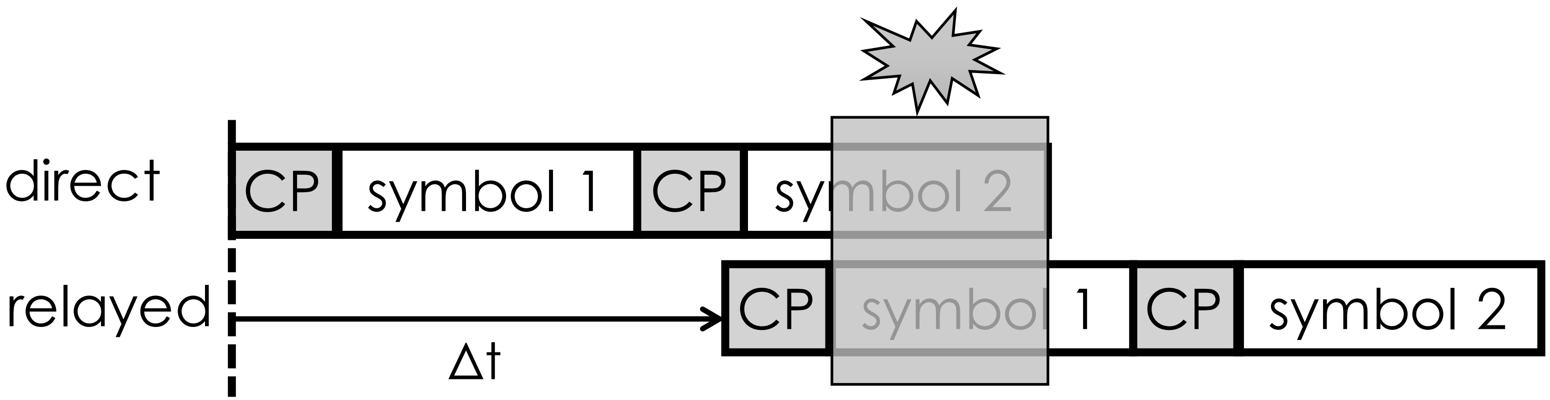
denoise at the relay

only amplify the signal



Challenges: Mixed Symbols

- Demodulation takes a much longer time
  - Receive the whole symbol → FFT → demodulation → modulation → IFFT
- It's unlikely to fast forward within a CP interval

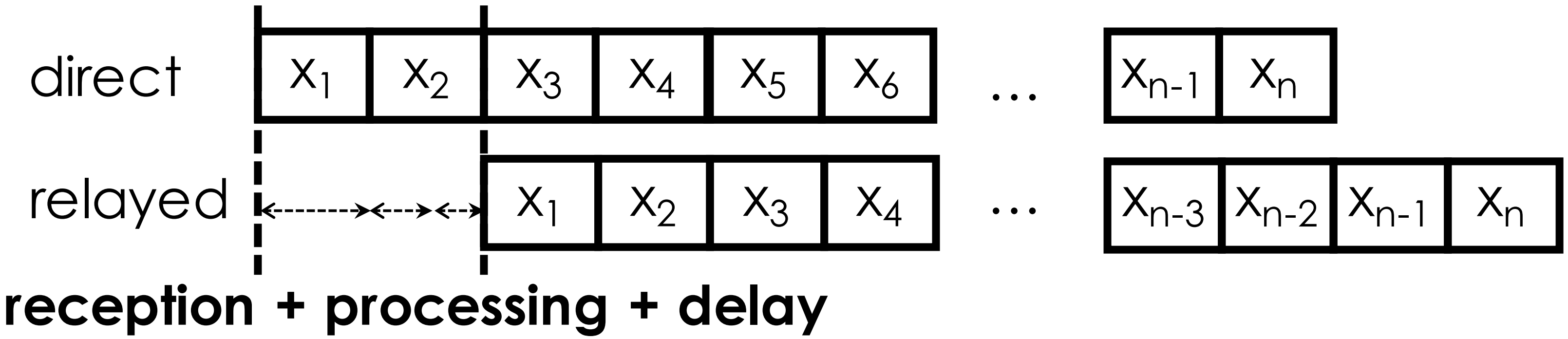


Inter-symbol interference at the destination

Need to recover from mixed symbols

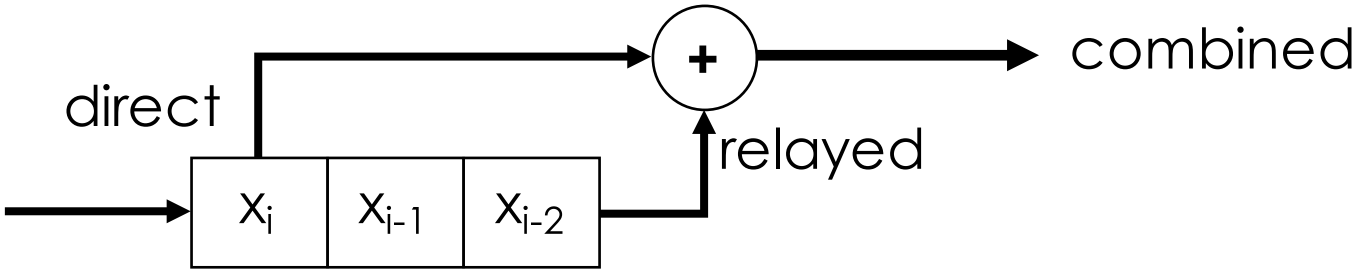
How to Ensure Decodability?

- Introduce delay to enable symbol-level alignment

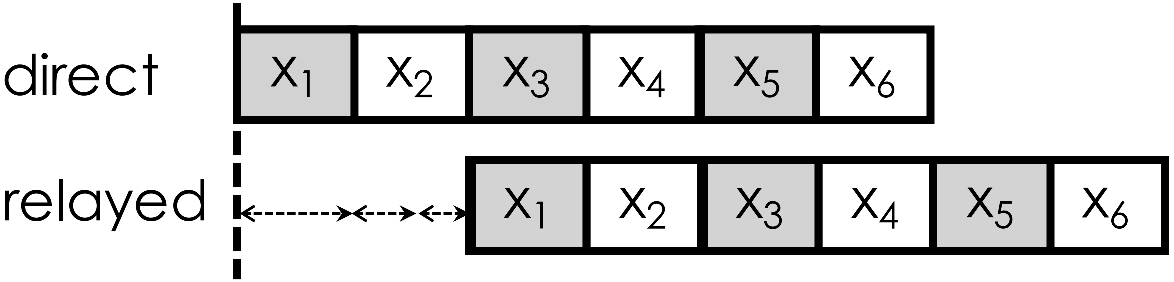


- Structure of combined signals is analogous to **convolutional code**

→ Viterbi-type Decoding



Viterbi-Type Decoding



$y_1 = h_{sd}x_1$ [redacted]

$y_3 = h_{sd}x_3 + h_{rd}x_1$

$y_5 = h_{sd}x_5 + h_{rd}x_3$

$y_7 =$ [redacted] $h_{rd}x_5$

Decode [y_1, y_3, y_5, y_7] and [y_2, y_4, y_6, y_8] separately because they are independent