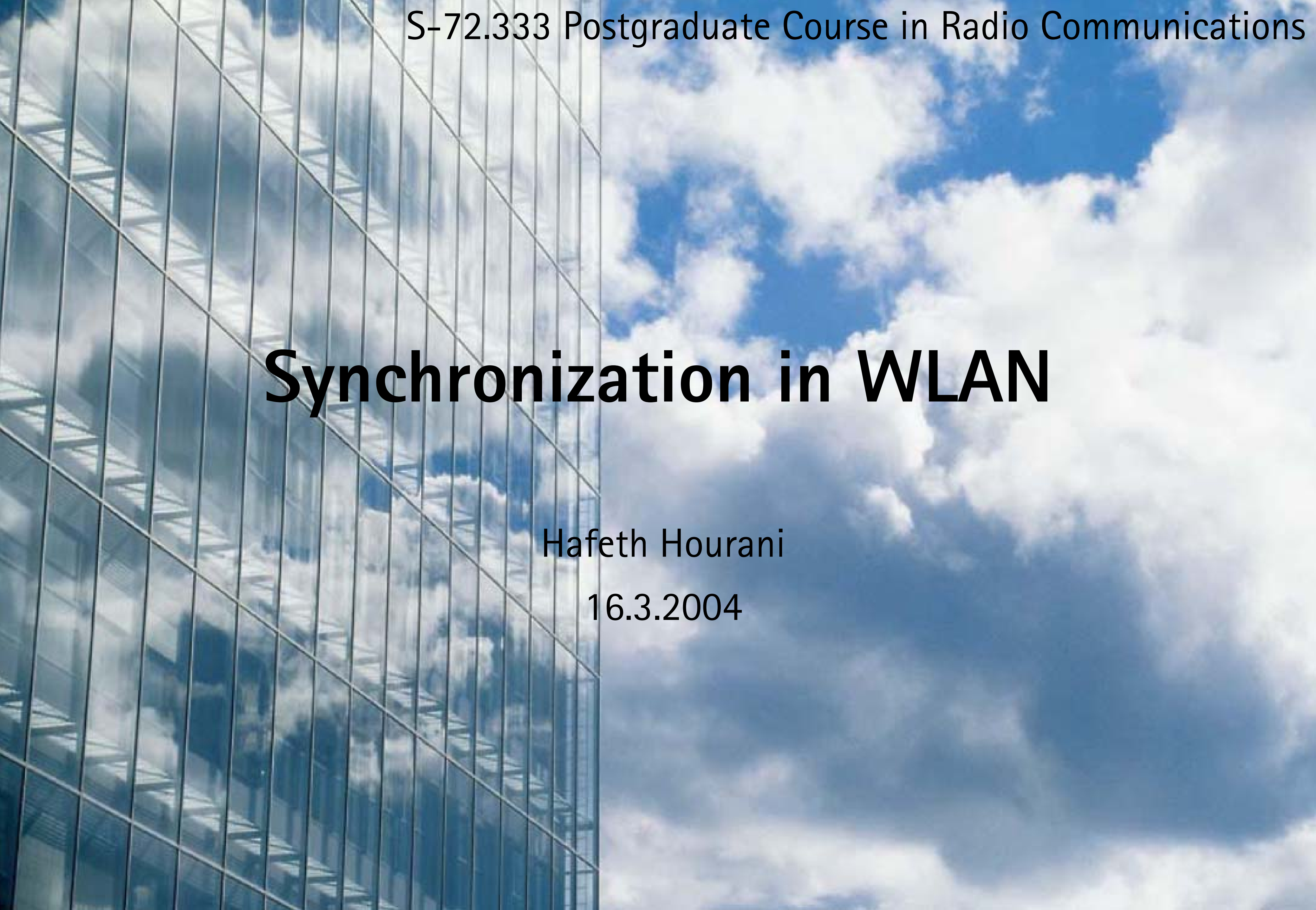


S-72.333 Postgraduate Course in Radio Communications

Synchronization in WLAN

Hafeth Hourani

16.3.2004

Outline

- Overview
- Timing Estimation
- Frequency Synchronization
- Channel Estimation
- Summary

Next . . .

- **Overview**
- Timing Estimation
- Frequency Synchronization
- Channel Estimation
- Summary

Synchronization and OFDM

- **Synchronization** is an essential task for any digital communication system
  - It is a major design problem

- **OFDM** is used for two kind of systems
  - Broadcast-type systems ➔ such as DAB & DVB
  - Packet-switched networks ➔ such as WLAN

- **Synchronization & OFDM**
  - Broadcast-type systems transmits data continuously
    - The receiver can initially spend relatively long time to acquire the signal (acquisition mode) and then switch to the tracking mode
  - WLAN systems have to use “single-shot” synchronization
    - The synchronization has to be acquired during a very short time after the start of the packet

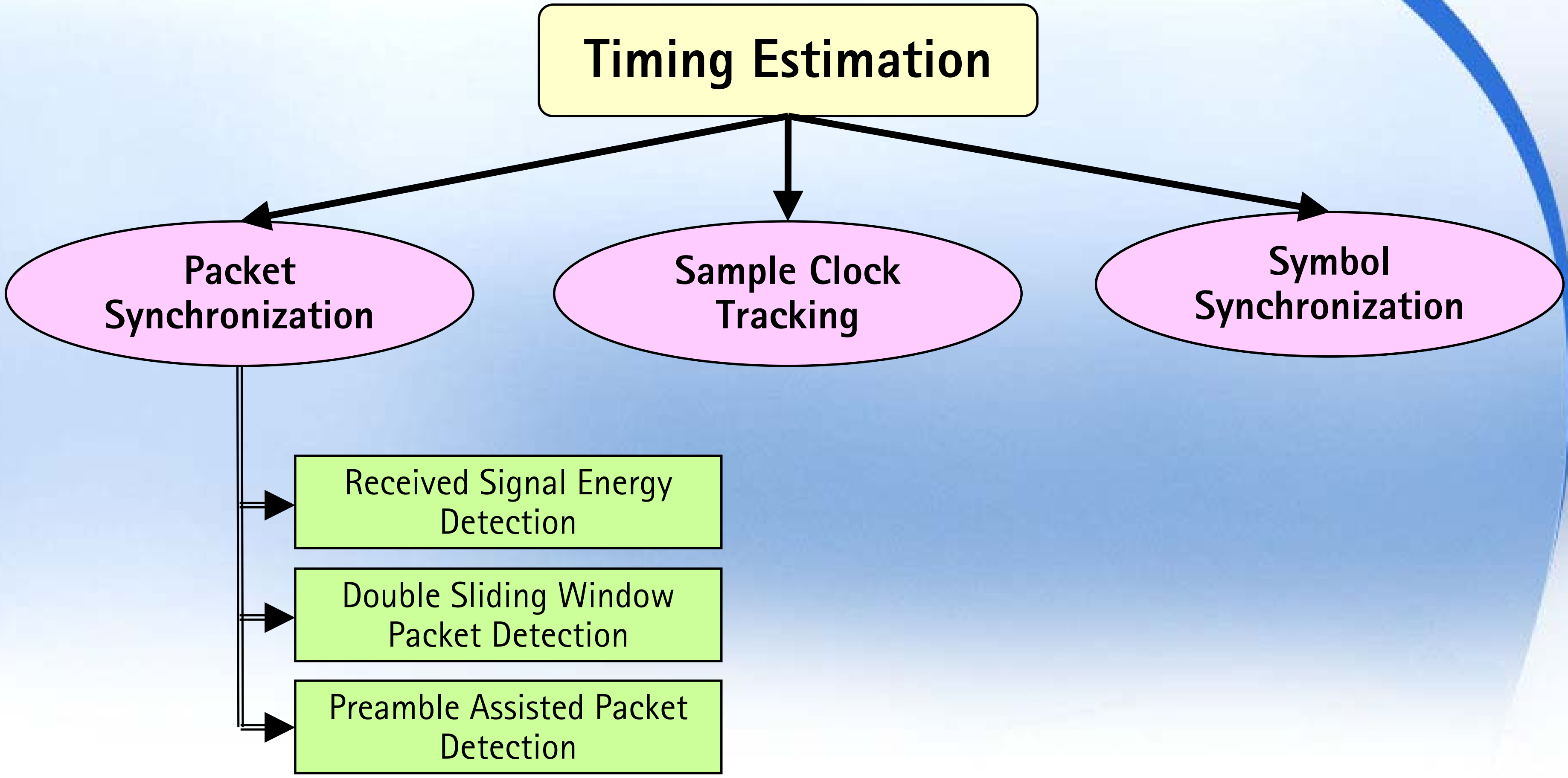
Synchronization in Packet-Switched Networks

- In WLAN, the receiver has to acquire the synchronization in very short time
  - Single-Shot Synchronization . . .
- To facilitate WLAN “single-shot synchronization”, the IEEE 802.11a includes
- Receiver training information is needed continuously
- To achieve good system throughput,

Next . . .

- Overview
- **Timing Estimation**
- Frequency Synchronization
- Channel Estimation
- Summary

Timing Estimation Overview


Timing Estimation
Packet Synchronization
Sample Clock Tracking
Symbol Synchronization
Received Signal Energy Detection
Double Sliding Window Packet Detection
Preamble Assisted Packet Detection

Packet Synchronization

- The IEEE 802.11 MAC protocol is essentially a Random Access Network.
  - The receiver does not know exactly when a packet starts

☞ Packet Detection is needed . . .

- Packet Detection

  ⇨ *The task of finding an approximate estimate of the start of the preamble of an incoming data packet*

- Packet Detection is the first synchronization algorithm that is performed

  ⇨ *The rest of the synchronization process is dependent on good packet detection performance*

Packet Detection

- Packet detection test

$$H_0 : Packet\ not\ present$$
$$H_1 : Packet\ present$$

  - Test statement

$$H_0 : m_n < Thr \Rightarrow Packet\ not\ present$$
$$H_1 : m_n \geq Thr \Rightarrow Packet\ present$$

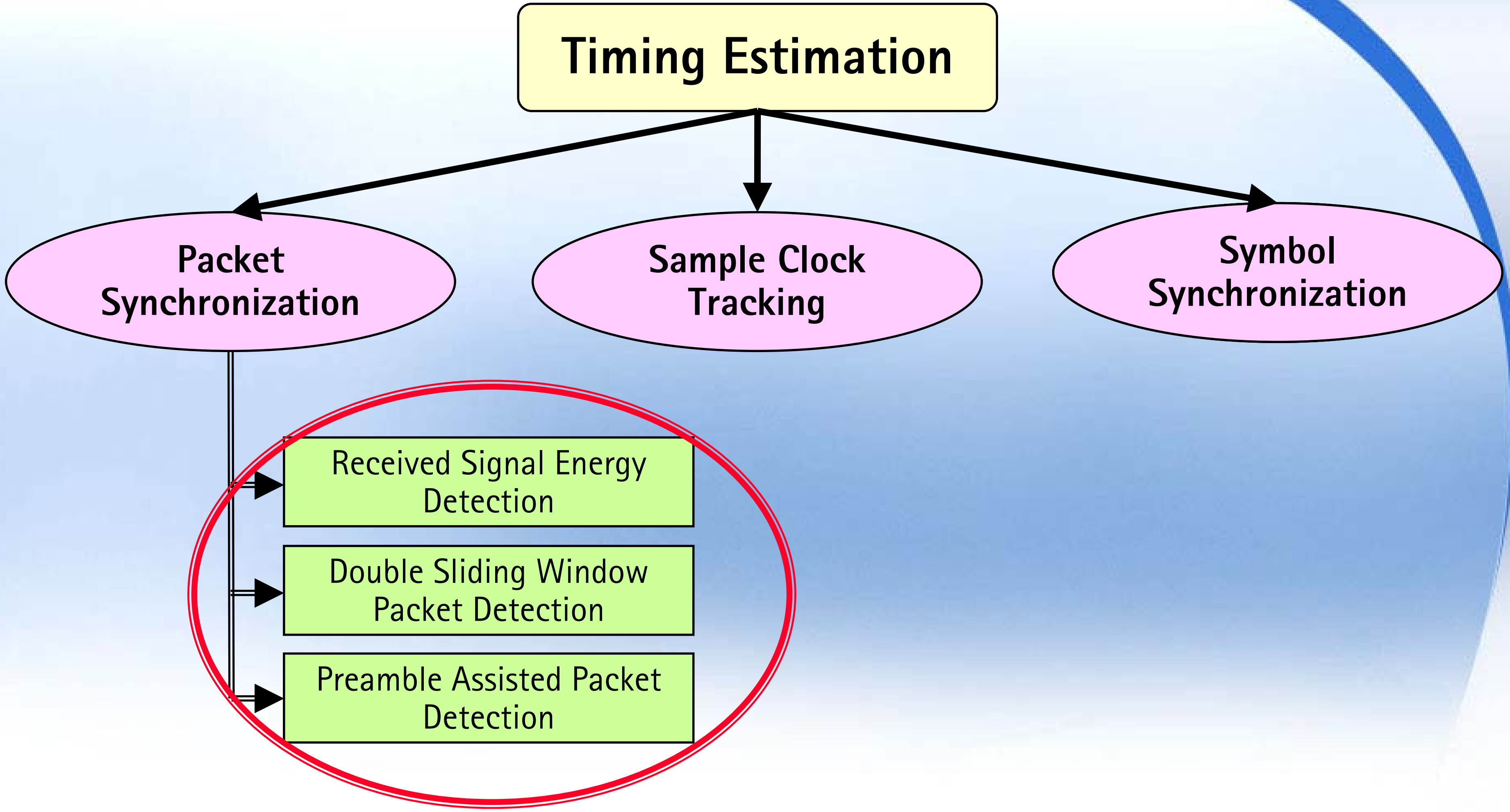
- The performance of packet detection can be measured with two probabilities:
  - Probability of detection P_D
    - The probability of detecting a packet when it is truly present
  - Probability of false alarm P_{FA}
    - The probability that the test decides that a packet is present, when actually there is none

Packet Detection Performance

- P_D should be as high as possible
- P_{FA} should be as small as possible
- In general, increasing P_D increases P_{FA} and decreasing P_D decreases P_{FA}
  - The packet detection algorithm should be a compromise between P_D and P_{FA}
- The false alarm is less severe error than not detecting a packet at all
  - After a false alarm, the receiver will try to synchronize to nonexistent packet and will detect an error
  - Not detecting a packet always results in loss of data

☞ A little higher P_{FA} can be tolerated to guarantee good P_D

Packet Detection Methods


Timing Estimation
Packet Synchronization
Sample Clock Tracking
Symbol Synchronization
Received Signal Energy Detection
Double Sliding Window Packet Detection
Preamble Assisted Packet Detection

Received Signal Energy (RSE) Detection

- The algorithm:
  - Find the start edge of the packet by measuring the received signal energy
  - When there is no signal received, the received signal $\boldsymbol{r_n}$ consists of noise only $\boldsymbol{r_n = w_n}$
  - When the packet starts, the received energy is increased by the signal component $\boldsymbol{r_n = w_n + s_n}$

☞ **The packet can be detected as a change in the received energy level**

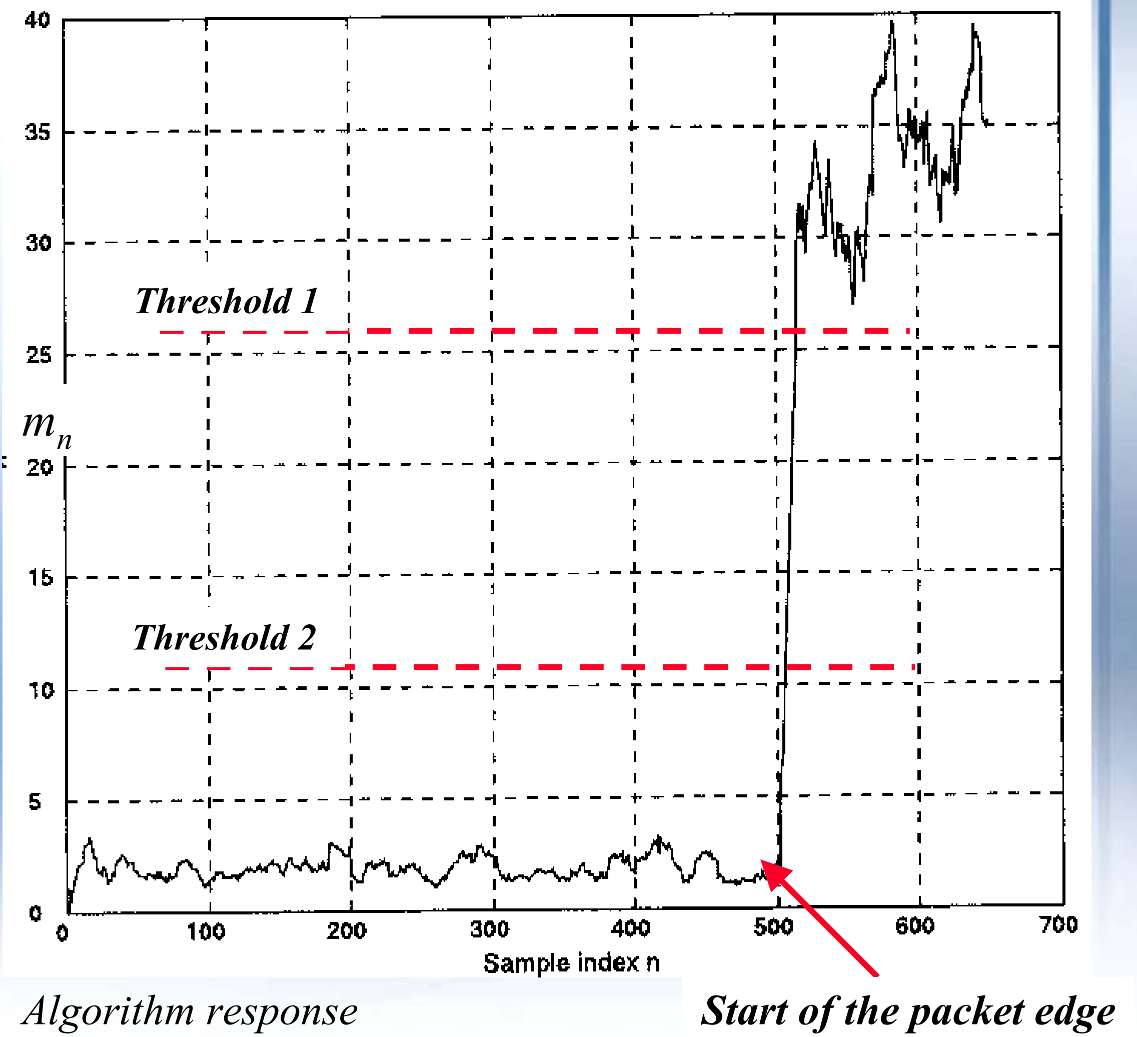
- Use the accumulated signal energy ($\boldsymbol{m_n}$)over a window of length $\boldsymbol{L}$

$$m_n = \sum_{k=0}^{L-1} r_{n-k} r_{n-k}^{*} = \sum_{k=0}^{L-1} \left| r_{n-k} \right|^2$$

Received Signal Energy Detection (cont'd)

Algorithm evaluation

- Pros.
  - Simple algorithm
- Cons.
  - The value of the threshold depends on the received signal energy
  - It is difficult to set a threshold to decide when a new packet is coming

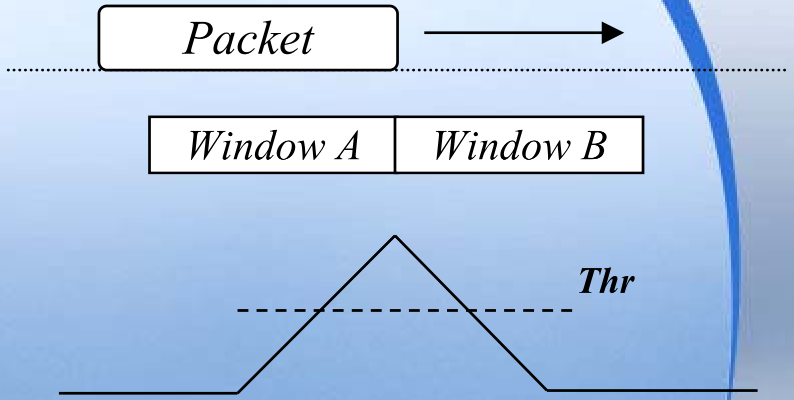


Algorithm response

IEEE 802.11a packet with SNR = 10 dB L = 32,

Double Sliding Window Detection (DSW)

- The algorithm
  - Use two consecutive sliding windows to calculate the received signal energy
  - Form the decision variable $\boldsymbol{m_n}$ as a ratio of the total energy contained inside the two windows
- Windows A and B are considered stationary relative to the sliding packets
- When only noise is received, the output is flat
- The response of $\boldsymbol{m_n}$ can be though of as a differentiator
  - The response is large when the input level changes rapidly



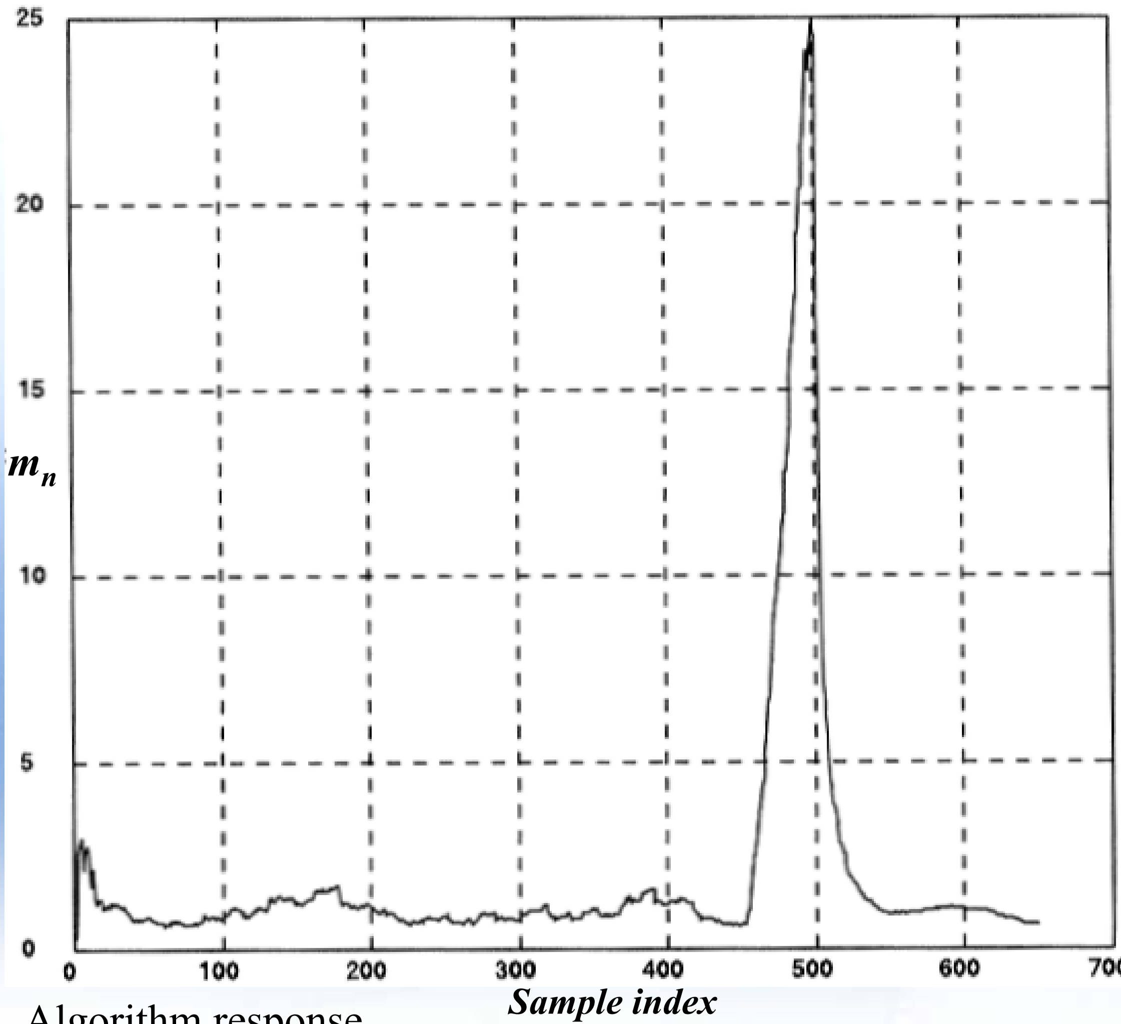
$$a_n = \sum_{m=0}^{M-1} r_{n-m} r_{n-m}^* = \sum_{m=0}^{M-1} |r_{n-m}|^2$$

$$b_n = \sum_{l=0}^{L-1} r_{n-l} r_{n-l}^* = \sum_{l=0}^{L-1} |r_{n-l}|^2$$

$$m_n = \frac{a_n}{b_n}$$

DSW (cont'd)

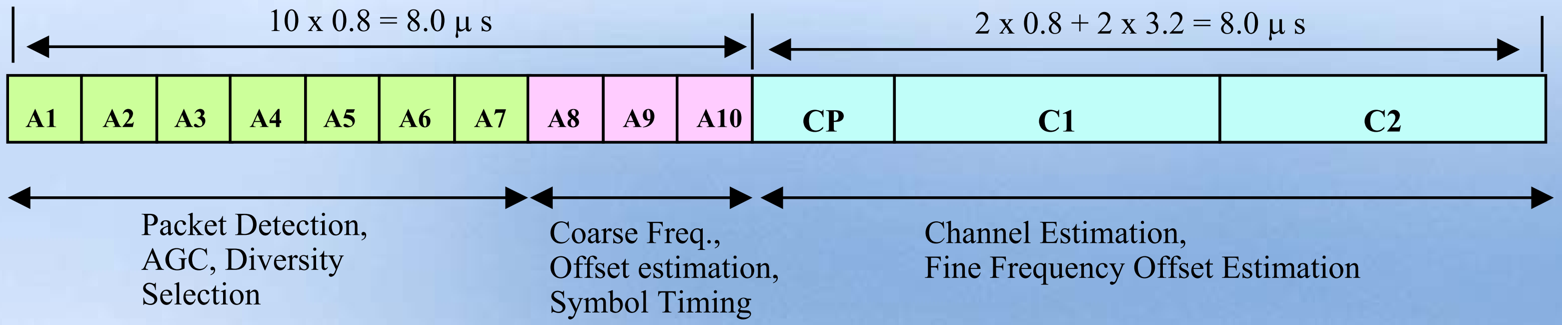
- The result of the DSW *does not* depend on the total received power
- The DSW is a good approach



Algorithm response
IEEE 802.11 preamble, SNR = 10 dB

Preamble Aided Packet Detection

- In this approach, the known IEEE 802.11 preamble structure is incorporated into the synchronization algorithm
- The IEEE 802.11 was designed to aid the detection of the incoming *packet edge*


10 x 0.8 = 8.0 μ s
2 x 0.8 + 2 x 3.2 = 8.0 μ s
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
CP
C1
C2
Packet Detection, AGC, Diversity Selection
Coarse Freq., Offset estimation, Symbol Timing
Channel Estimation, Fine Frequency Offset Estimation

Preamble parts:
A1 → A10 : short training symbols (16 samples each)
CP : cyclic prefix (32 samples) that protects C1 and C2 from ISI
C1 & C2 : Long training symbols (64 samples)

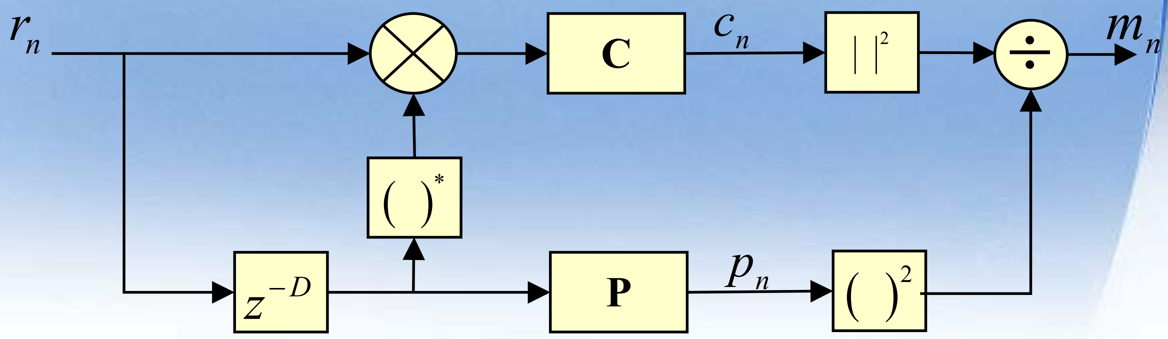
Preamble Aided Detection (cont'd)

- Delay and Correlate Algorithm
  - This algorithm utilizes the same DSW algorithm, but with taking advantage of the periodicity of the short training symbols at start of the preamble

$$c_n = \sum_{k=0}^{L-1} r_{n+k} r^*_{n+k+D}$$

$$p_n = \sum_{k=0}^{L-1} r_{n+k+D} r^*_{n+k+D} = \sum_{k=0}^{L-1} \left| r_{n+k+D} \right|^2$$

$$\Rightarrow m_n = \frac{\left| c_n \right|^2}{\left(p_n \right)^2}$$

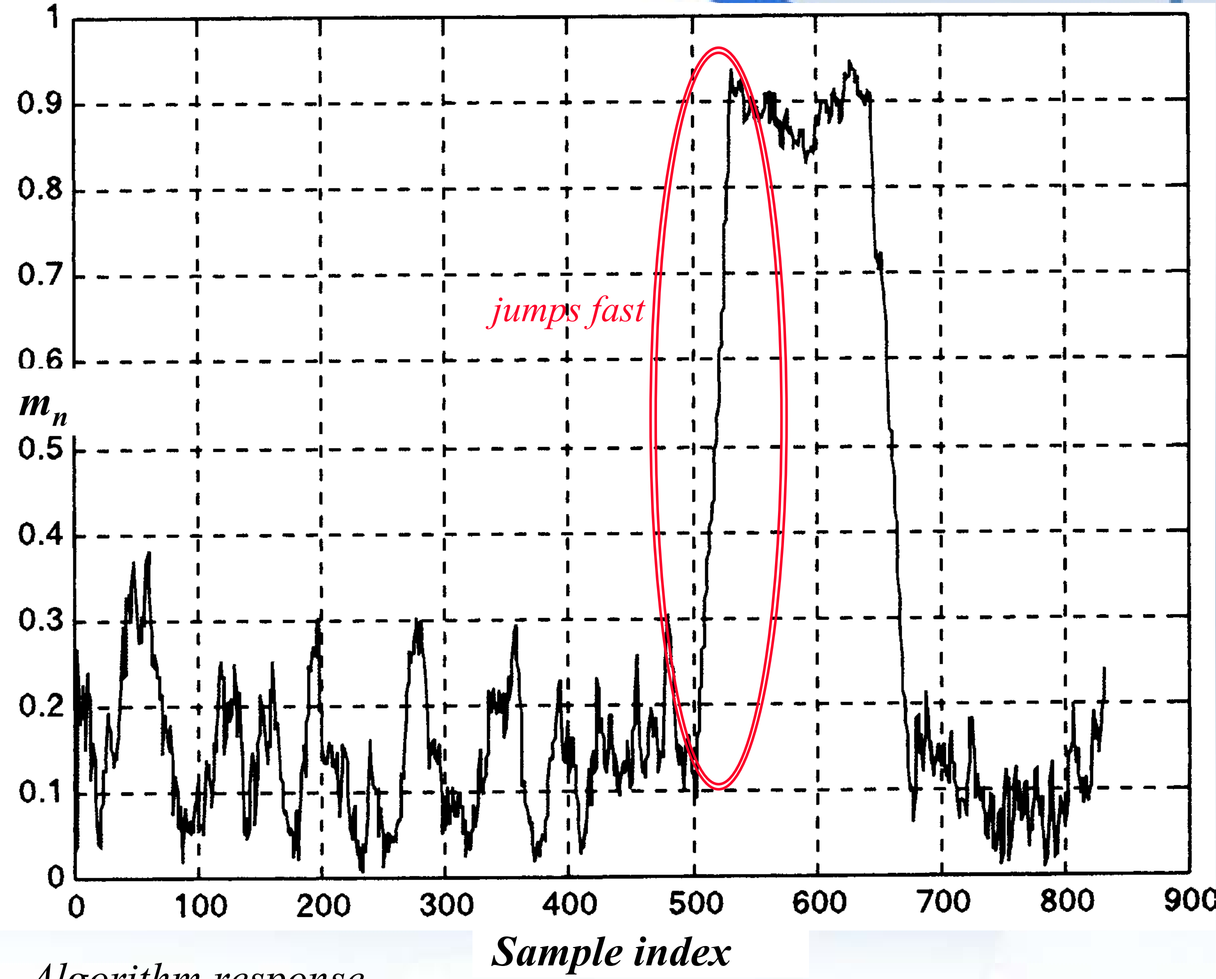


Delay and Correlate Algorithm
Two sliding windows C & P

The delay D = the period of the start of the preamble (e.g. D = 16 for IEEE 802.11)

Preamble Aided Detection (cont'd)

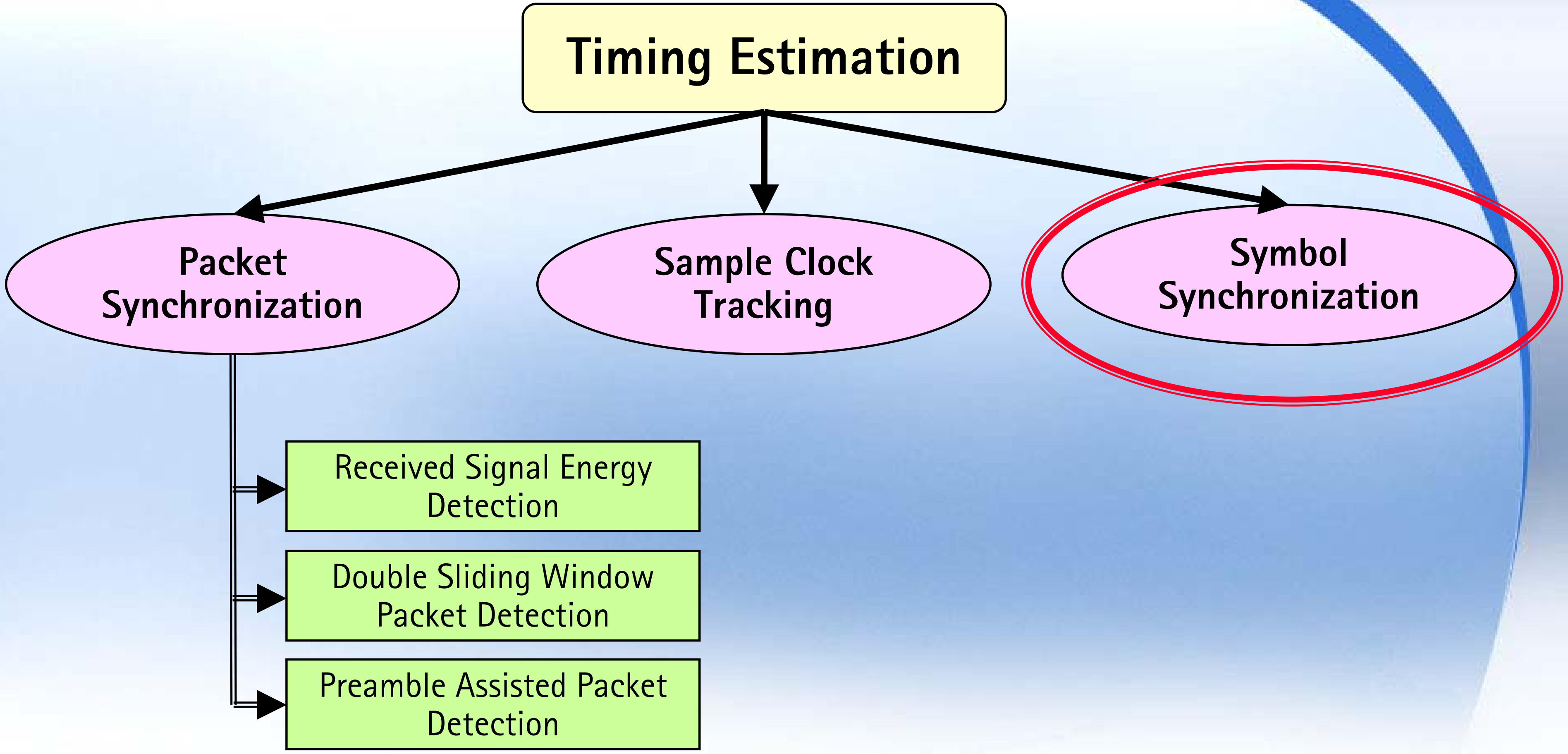
- When the received signal consists of only noise, the output c_n of the delayed crosscorrelation is zero-mean r.v.
- Once the start of the packet is received, c_n is a crosscorrelation of the identical short training symbols, which causes m_n to jump quickly to its maximum value
- This jump gives a good estimate of the packet edge



Algorithm response

IEEE 802.11a packet with SNR = 10 dB

Packet Detection Methods


Timing Estimation
Packet Synchronization
Sample Clock Tracking
Symbol Synchronization
Received Signal Energy Detection
Double Sliding Window Packet Detection
Preamble Assisted Packet Detection

Symbol Timing

- **Symbol Timing** is the task of finding the precise moment of when individual OFDM symbols start and end

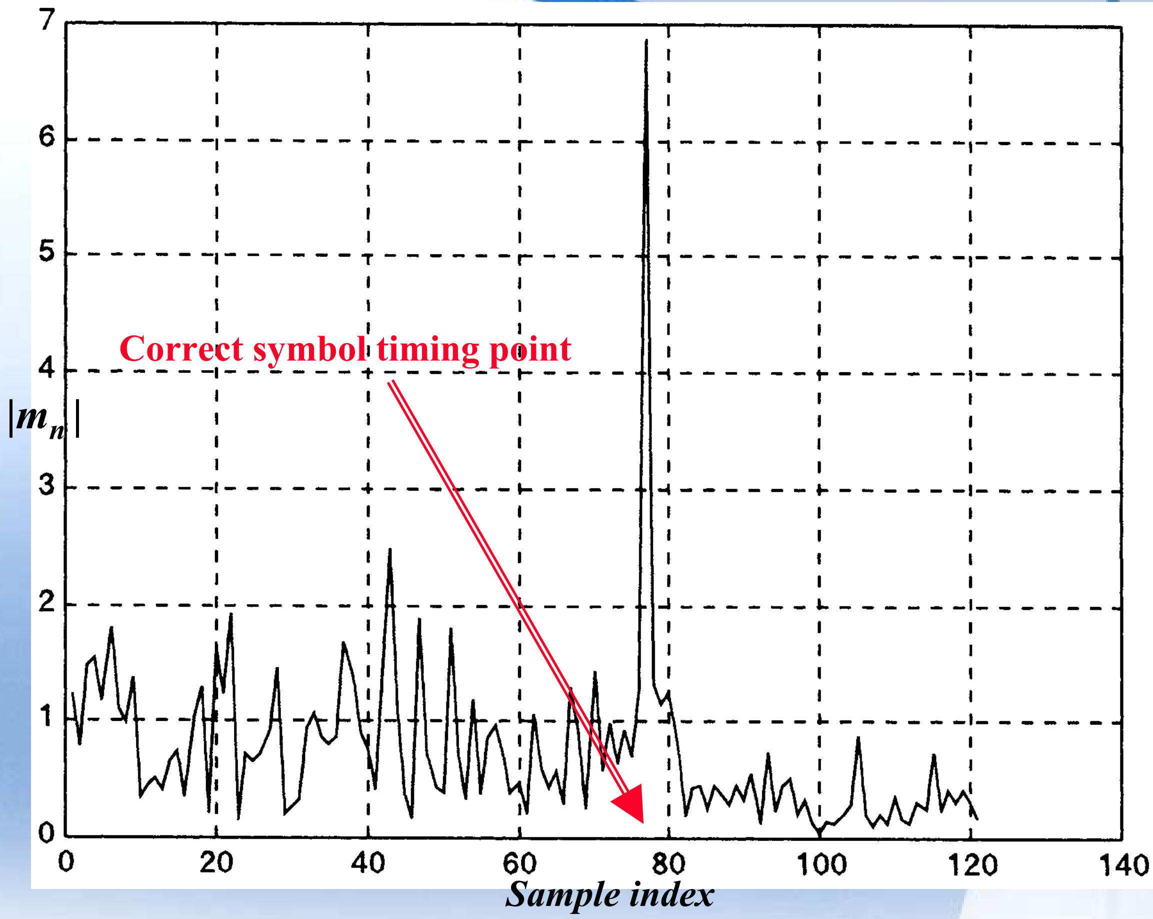
- Packet Detection provides an estimate of the packet edge
- Symbol timing refines the estimate to the symbol level
- Symbol timing is essential in OFDM
  - The symbol timing results defines the DFT window, i.e., the set of samples used to calculate DFT of each received OFDM symbol
  - The DFT result is then used to demodulate the subcarriers of the symbol

Symbol Timing Algorithm

- Symbol timing can be performed by calculating the crosscorrelation of the received signal $\boldsymbol{r_n}$ and a known reference $\boldsymbol{t_k}$

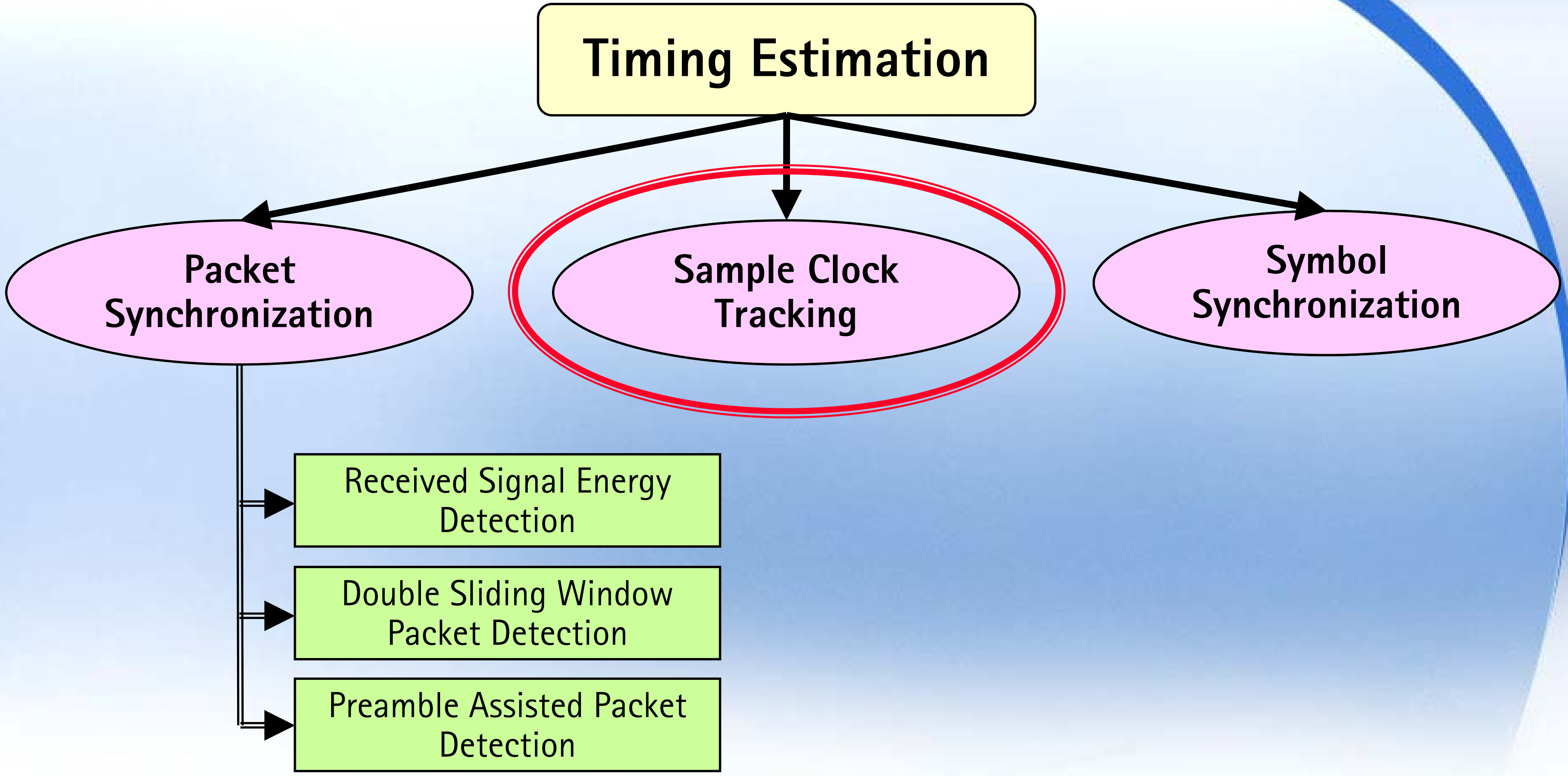
$$\hat{t}_s = \arg\max_n \left| \sum_{k=0}^{L-1} r_{n+k} t_k^* \right|^2$$

In the equation, the value of ***n*** that corresponds to maximum absolute value of the crosscorrelation is the symbol timing estimate



The output of the crosscorrelator that uses the first 64 samples of the long training symbols of IEEE 802.11 preamble
SNR = 10 dB

Packet Detection Methods


Timing Estimation
Packet Synchronization
Sample Clock Tracking
Symbol Synchronization
Received Signal Energy Detection
Double Sliding Window Packet Detection
Preamble Assisted Packet Detection

Sample Clock Synchronization

- The task of tracking the sampling clock frequency
- The sampling clock error has two main negative impacts:
  - A slow shift of the symbol timing point
    - Results in a subcarrier rotation
  - Loss of SNR due to intercarrier interference (ICI)
    - Results in a loss of subcarriers orthogonalilty
- The normalized sampling error is given by

$$t_\Delta = \frac{T' - T}{T}$$

T' Receiver sampling period

T Transmitter sampling period

- The degradation in SNR was proved to be

$$D_n \approx 10\log\left(1 + \frac{\pi^2}{3}\frac{E_s}{N_0}\left(kt_\Delta\right)^2\right)$$

where k: Subcarrier index

Sample Clock Synchronization (cont'd)

- The amount of rotation angle experienced by different subcarriers is given by

$$e^{j2\pi k t_\Delta l \frac{T_s}{T_u}}$$

where k: Subcarrier index
l : OFDM symbol index
T_s: Duration of the total OFDM symbol
T_u: Useful data portion of the symbol

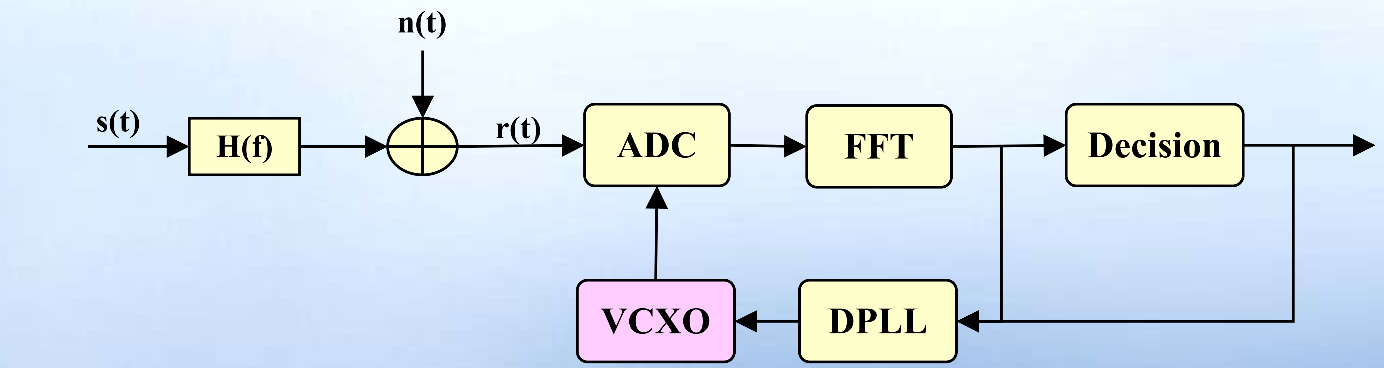
- The angle depends on both subcarrier index and the OFDM symbol index
  - Hence, the impact is larger on the outermost subcarriers and increase with consecutive OFDM symbols
- Large subcarrier rotation prevents the correct demodulation

Estimating the Sampling Frequency Error

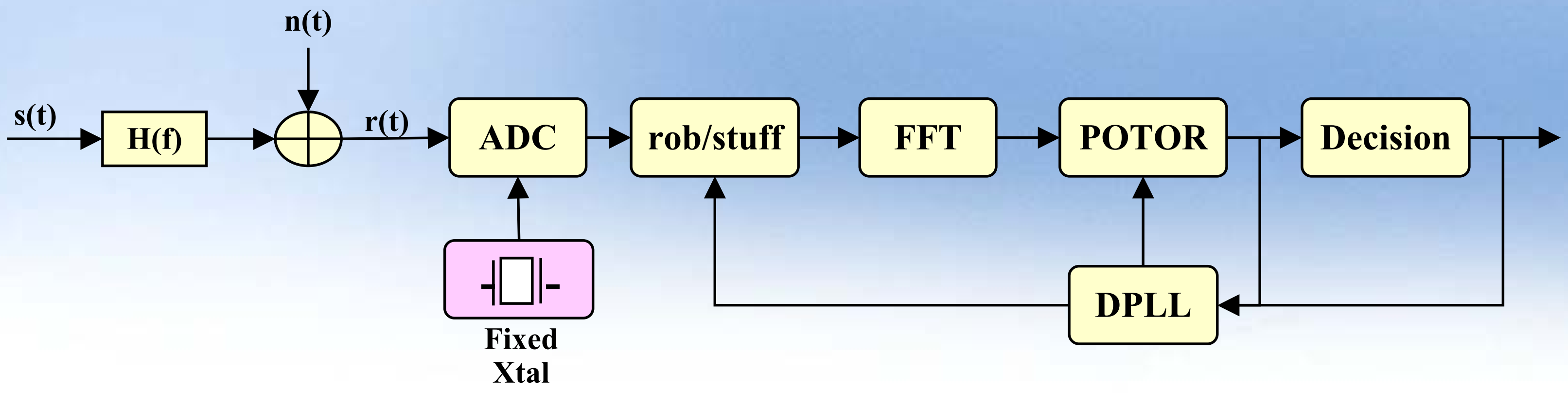
- The approach of estimating the sampling frequency error rely on the ***pilot subcarriers***
- The pilot subcarriers are used to transmit known data called ***pilot symbols***
- The receiver can utilize the pilot symbols to perform synchronization functions
- The sampling frequency offset is estimated by using the knowledge of the linear relationship between the phase rotation caused by the offset and the pilot subcarrier index

Correcting the Sampling Frequency Error

- The rotation caused by the sampling frequency offset can be corrected with two main approaches
  - By adjusting the sampling frequency for the receiver ADC



  - By de-rotating the subcarrier after the DFT processing



Next . . .

- Overview
- Timing Estimation
- **Frequency Synchronization**
- Channel Estimation
- Summary

Frequency Synchronization

- OFDM is very sensitive to the carrier frequency offset
- The carrier frequency error causes a significant SNR degradation in OFDM
- The degradation is caused by two main phenomena:
  - Subcarrier amplitude reduction
    - This is because the desired subcarrier is no longer sampled at the peak of the sinc-function of DFT
  - ICI caused by neighboring subcarriers
- The SNR degradation is approximated by

$$SNR_{loss} = \frac{10}{3\ln 10}\left(\pi T f_{\Delta}\right)\frac{E_s}{N_0}\ dB$$

Frequency Synchronization Approaches

- The estimation algorithms of the carrier frequency offset in OFDM can be divided into three types:
  - Data-aided algorithms
    - Based on special training information embedded into the transmitted signal
  - Non data-aided algorithms
    - The received signal is analyzed in the frequency domain
  - Cyclic prefix based algorithms

- In WLAN applications, the first type is the most important (Data-aided)

Next . . .

- Overview
- Timing Estimation
- Frequency Synchronization
- **Channel Estimation**
- Summery

Channel Estimation

- Channel Estimation is the task of estimating the frequency response of the radio channel
- The impulse response of a time varying channel

$$h(\tau;t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n(t))$$

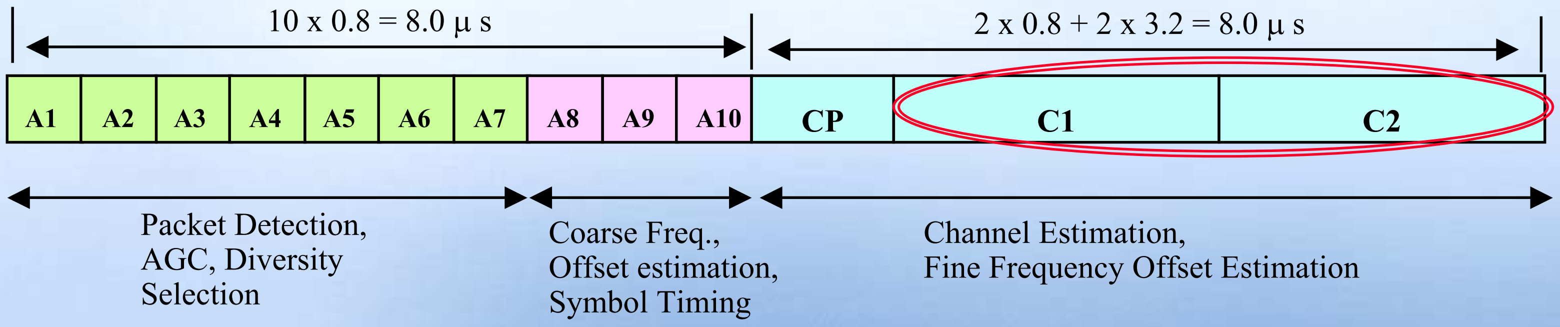
- The WLAN applications generally assume a quasistationary channel

$$h(\tau) = \sum_n \alpha_n e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n)$$

- There are two approaches for handling the channel estimation
  - Frequency Domain
  - Time Domain

Frequency Domain Channel Estimation

- The channel estimation is based on the training data transmitted on every subcarrier
  - The training symbols in the WLAN preamble are used for this purpose



10 x 0.8 = 8.0 μ s
2 x 0.8 + 2 x 3.2 = 8.0 μ s
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
CP
C1
C2
Packet Detection, AGC, Diversity Selection
Coarse Freq., Offset estimation, Symbol Timing
Channel Estimation, Fine Frequency Offset Estimation

Preamble parts:
A1 → A10 : short training symbols (16 samples each)
CP : cyclic prefix (32 samples) that protects C1 and C2 from ISI
C1 & C2 : Long training symbols (64 samples)

- The contents of C1 and C2 are identical
  - Averaging over C1 and C2 can be used to improve the quality of estimate

Frequency Domain Channel Estimation (cont'd)

- After DFT processing, the received training symbols are

$$R_{l,k} = H_k X_k + W_{l,k}$$

Where $R_{l,k}$: the received training symbols
X_k : the training symbol
H_k : channel response
W_k : additive noise

- The channel estimate can be calculated as the following:

$$\hat{H}_k = \frac{1}{2}(R_{1,k} + R_{2,k})X_k^*$$

$$= \frac{1}{2}(H_k X_k + W_{1,k} + H_k X_k + W_{2,k})X_k^*$$

$$= H_k \left|X_k\right|^2 + \frac{1}{2}(W_{1,k} + W_{2,k})X_k^*$$

Selecting data training amplitude to be equal to one

$$= H_k + \frac{1}{2}(W_{1,k} + W_{2,k})X_k^*$$

The noise samples $W_{1,k}$ and $W_{2,k}$ are statistically independent

Time Domain Channel Estimation

- The channel impulse response is estimated

$r_{1,n} = h * x_n + w_{1,n}$ The received time domain signal during the two long training symbols

$$X = \begin{bmatrix} x_1 & x_{64} & x_{63} & \cdots & x_{64-L+2} \\ x_2 & x_1 & x_{64} & & x_{64-L+2} \\ \vdots & & \vdots & & \vdots \\ x_{63} & x_{62} & & & x_{64-L} \\ x_{64} & x_{63} & & \cdots & x_{64-L+1} \end{bmatrix}$$

The circular convolution matrix of the training symbols, L the max length of impulse response that can be estimated

$$h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_L \end{bmatrix}$$

The channel impulse response vector

$$\begin{aligned} \hat{h} &= \frac{1}{2} X^{-1}(r_{1,n} + r_{2,n}) \\ &= \frac{1}{2} X^{-1}(Xh + w_1 + Xh + w_2) \\ &= X^{-1}Xh + \frac{1}{2} X^{-1}(w_1 + w_2) \\ &= h + \frac{1}{2} X^{-1}(w_1 + w_2) \end{aligned}$$

Next . . .

- Overview
- Timing Estimation
- Frequency Synchronization
- Channel Estimation
- **Summary**

Summary

- Synchronization is an essential task in WLAN
- WLAN synchronization can be decomposed into four main task:
  - Timing estimation
  - Sampling frequency estimation
  - Frequency Synchronization
  - Channel Estimation
- Timing estimation consists of two main task
  - The packet synchronization
  - The symbol synchronization

References

- OFDM Wireless LANs: A Theorethical and Practical Guide, Juha Heiskala, John Terry, Sams Publishing 2002

Thank You !

Homework

- Q1. In that equation in slide 12, we have accumulated the signal energy over a window of size L. Why ?
- Q2. In the double sliding window algorithm (presented in slide 14), the output of the receiver is flat when only noise is received. Why? And how does this algorithm solve the problems encountered when using single window?
- Q3. In block diagram of the Preamble Assisted Detection algorithm (slide 17); what is the purpose of using window P?
- Q4. The most important frequency synchronization algorithm for WLAN applications is the "Data-aided algorithms" (slide 29). Why? Where did we get such data to "aid" the frequency synchronization?