Unit 5. Adaptive Modulation and Coding

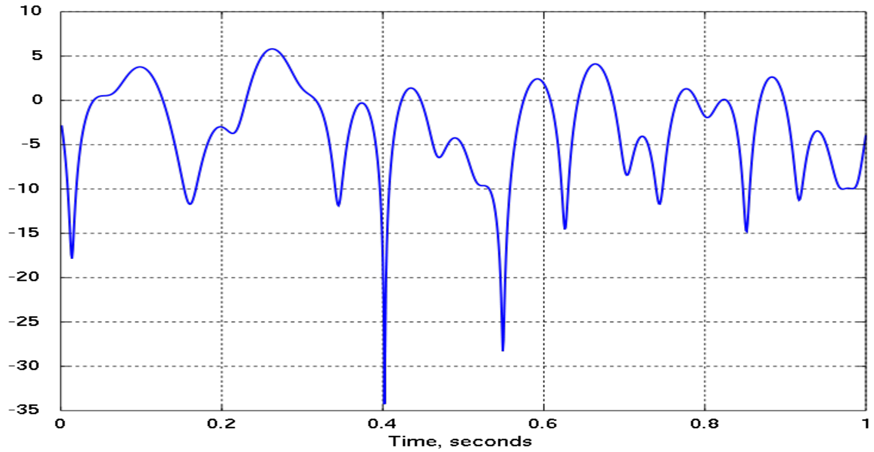
ECE-GY 6023. WIRELESS COMMUNICATIONS

PROF. SUNDEEP RANGAN

Questions

- Fading causes variations in SNR
  - Large scale and small-scale variations
- TX must adapt the MCS
- Basic questions:
  - How does the TX select the MCS?
  - How does it send the selected MCS to the RX?
  - What happens if it selects the MCS incorrectly?

High rate



Low rate

Learning Objectives

- ❑ Describe ARQ protocol in 802.11 WiFi systems
- ❑ Implement and simulate a simple trial-and-error MCS selection system
- ❑ Describe the key components in scheduling in 5G NR
  - Data and control channels, timeline
- ❑ Compute delays and packet ordering with errors in a multiple ARQ processes
- ❑ Describe the CSI-RS channel
- ❑ Implement and simulate simple CSI measurements from the CSI-RS measurements
- ❑ Describe and analyze Hybrid ARQ
- ❑ Implement and simulate multi-process HARQ process over a fading channel

Outline

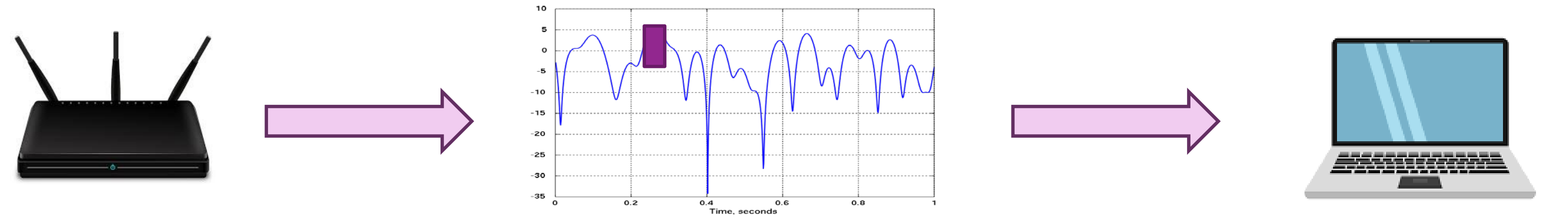
- MCS Selection and ARQ in 802.11 Wireless LAN
- MCS Selection, Scheduling and ARQ in 5G NR
- CSI Feedback and Link Adaptation in 5G NR
- Hybrid ARQ

802.11 Family of Wireless LAN Standards

Frequency range, or type	PHY	Protocol	Release date[13]	Frequency	Bandwidth	Stream data rate[14]	Allowable MIMO streams
				(GHz)	(MHz)	(Mbit/s)	
1–6 GHz	DSSS/FHSS[15]	802.11-1997	Jun 1997	2.4	22	1, 2	N/A
	HR-DSSS[15]	802.11b	Sep 1999	2.4	22	1, 2, 5.5, 11	N/A
	OFDM	802.11a	Sep 1999	5	5/10/20	6, 9, 12, 18, 24, 36, 48, 54 (for 20 MHz bandwidth, divide by 2 and 4 for 10 and 5 MHz)	N/A
		802.11j	Nov 2004	4.9/5.0[D][16][failed verification]			
		802.11p	Jul 2010	5.9			
		802.11y	Nov 2008	3.7[A]			
	ERP-OFDM	802.11g	Jun 2003	2.4			
	HT-OFDM[18]	802.11n	Oct 2009	2.4/5	20	Up to 288.8[B]	4
					40	Up to 600[B]	
	VHT-OFDM[18]	802.11ac	Dec 2013	5	20	Up to 346.8[B]	8
					40	Up to 800[B]	
					80	Up to 1733.2[B]	
					160	Up to 3466.8[B]	
	HE-OFDMA	802.11ax	Est. Feb 2021	2.4/5/6	20	Up to 1147[F]	8
					40	Up to 2294[F]	
					80	Up to 4804[F]	
					80+80	Up to 9608[F]	
mmWave	DMG[21]	802.11ad	Dec 2012	60	2,160	Up to 6,757[22] (6.7 Gbit/s)	N/A
		802.11aj	Apr 2018	45/60[C]	540/1,080[24]	Up to 15,000[25] (15 Gbit/s)	4[26]
	EDMG[27]	802.11ay	Est. March 2021	60	8000	Up to 20,000 (20 Gbit/s)[28]	4

Full details on Wikipedia

MCS Selection in 802.11

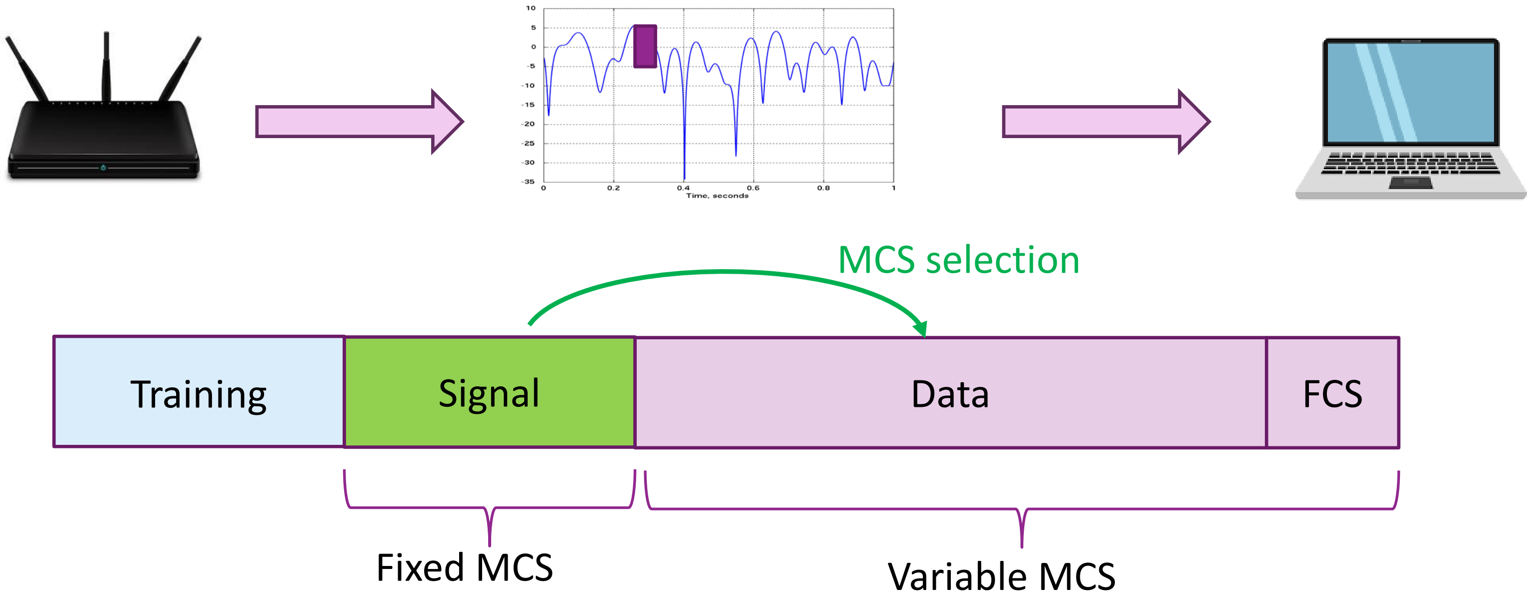


MCS Options in 802.11n (HT) and 802.11ac (VHT)

- ❑ Each packet is transmitted in some MCS
- ❑ Ex: 802.11n (see full table on Wikipedia)
  - Eight possible MCS options
  - Up to four possible spatial streams (More on this in MIMO)
- ❑ Trade off SNR requirement with data rate

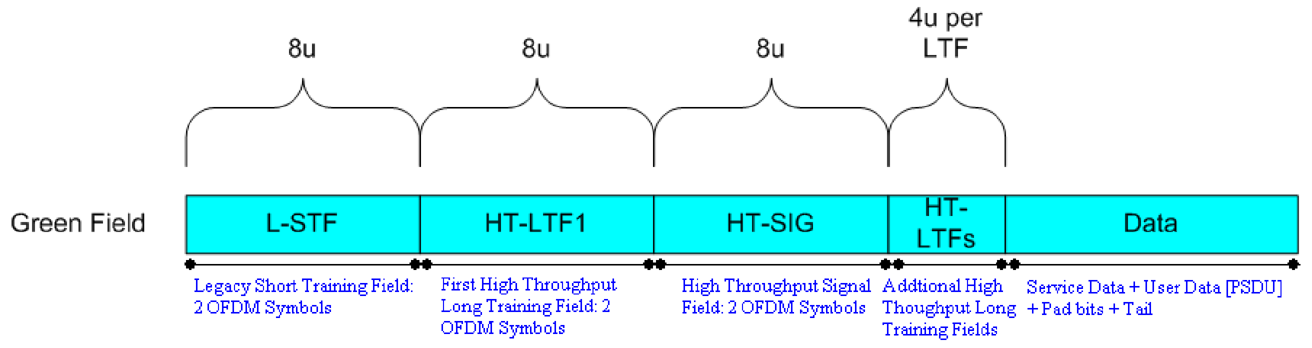
HT MCS	VHT MCS	Modulation	Coding	20MHz Data Rate 800ns	20MHz Data Rate 400ns	20MHz Min. SNR	20MHz RSSI	40MHz Data Rate 800ns	40MHz Data Rate 400ns	40MHz Min. SNR	40MHz RSSI
1 Spatial Stream											
0	0	BPSK	1/2	6.5	7.2	2	-82	13.5	15	5	-79
1	1	QPSK	1/2	13	14.4	5	-79	27	30	8	-76
2	2	QPSK	3/4	19.5	21.7	9	-77	40.5	45	12	-74
3	3	16-QAM	1/2	26	28.9	11	-74	54	60	14	-71
4	4	16-QAM	3/4	39	43.3	15	-70	81	90	18	-67
5	5	64-QAM	2/3	52	57.8	18	-66	108	120	21	-63
6	6	64-QAM	3/4	58.5	65	20	-65	121.5	135	23	-62
7	7	64-QAM	5/6	65	72.2	25	-64	135	150	28	-61
	8	256-QAM	3/4	78	86.7	29	-59	162	180	32	-56
	9	256-QAM	5/6			31	-57	180	200	34	-54
2 Spatial Streams											

Sending the MCS Selection



- Training field: To detect presence of packet, synchronize RX and train equalizer
- Signal field: Packet format including MCS and PDU length
- Data: User data plus MAC information such as address info
- FCS: Frame check sum to detect if data is received correctly

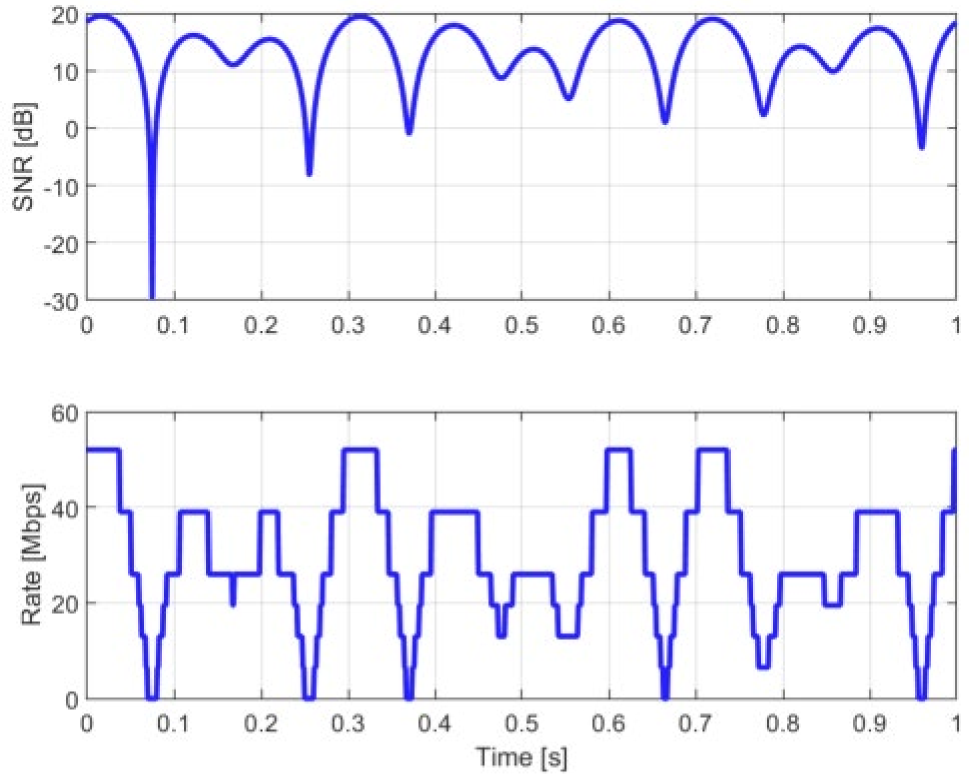
Example Details in 802.11n



From http://rfmw.em.keysight.com/wireless/helpfiles/n7617/Content/Main/MIMO_OFDM_Signal_Structure.htm

- Packet components:
  - L-STF and HT-LTF1: Training fields for synchronization and equalization training
  - HT-SIG: Encodes the MCS and PDU length
  - Data: The PDU encoded at the MCS

Example: MCS Selection on a Fading Channel

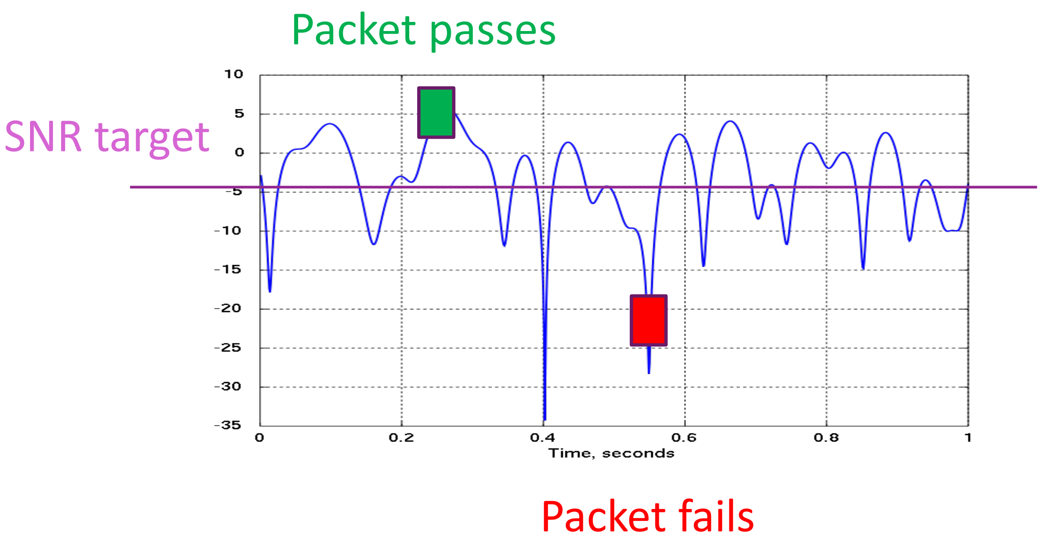


- Simple simulation:
  - 3 path channel
  - Doppler spread 10 Hz
  - Narrowband fading
  - Avg SNR = 15 dB
  - Time is divided into 1ms time slots
- At each time slot:
  - Select MCS based on SNR
  - Observe discrete rate options
- See demo

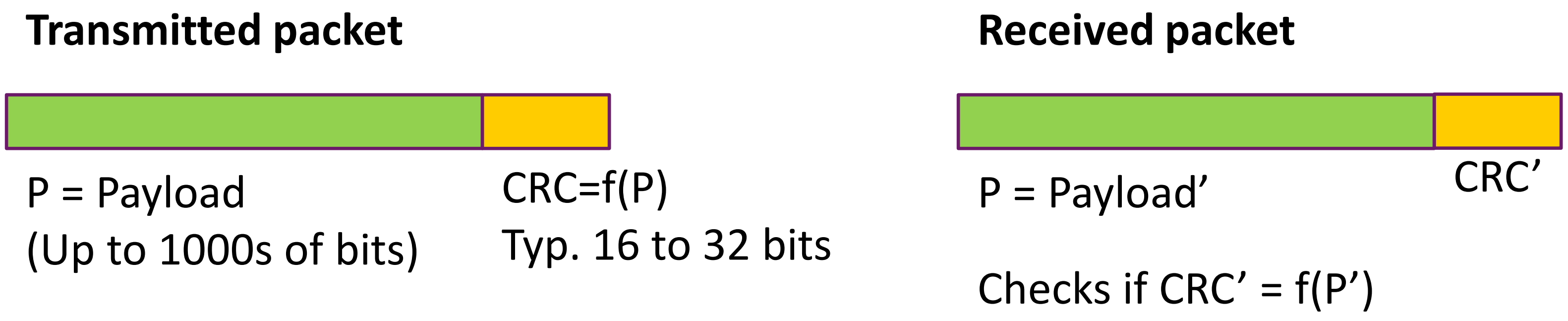
Packet Errors

- Packets may experience errors
  - MCS may not be selected correctly

- Questions:
  - How does the TX select the MCS optimally?
  - How does the RX detect the errors?
  - What can the TX do?

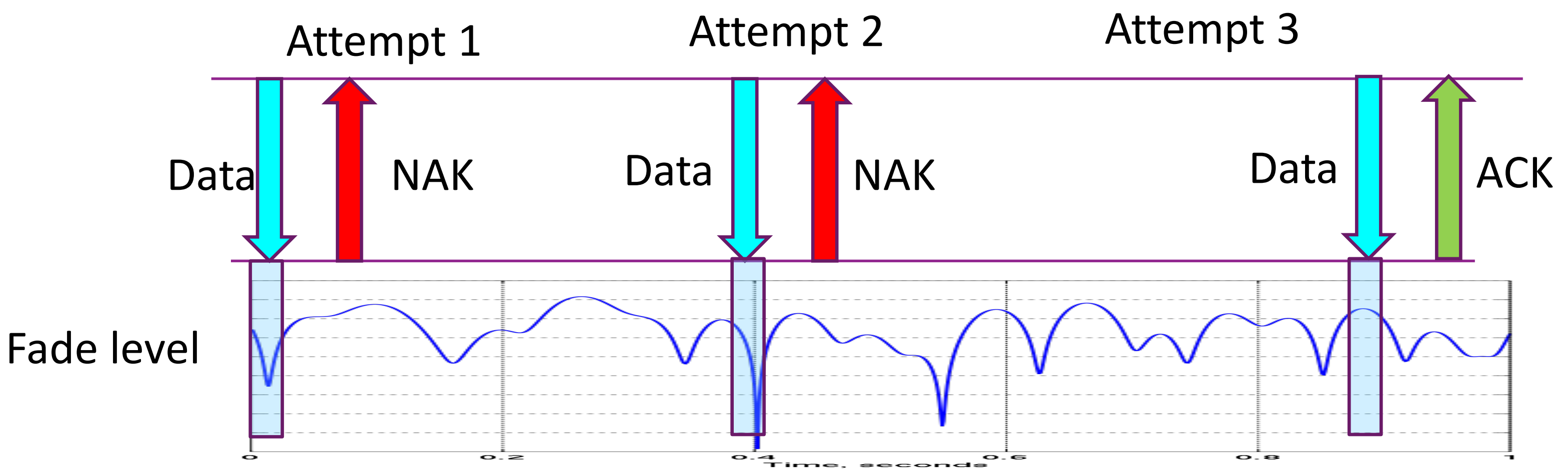


Detecting Errors with CRCs



- How does the receiver know that a data packet received successfully?
- Each packet contains a CRC (cyclic redundancy check)
  - CRC = f(payload)
- Receiver checks if CRC=f(payload) for decoded packet
- If CRC fails, receiver knows there is an error
- If CRC passes, very high probability that packet is correct
  - Chance that CRC passes for a random packet is 2^{-n}, n = num bits in CRC

Auto-Repeat Request (ARQ)



- Can overcome errors with multiple attempts
  - RX success is checked with a CRC on packet
  - RX then replies with ACK (success) or NACK (fail)
  - TX resends on NAK.
  - Continues until passes

Error Probability with ARQ

- Let
  - γ_k = SNR in attempt k (assume flat fading for now)
  - γ_{min} = minimum SNR to pass
- Assume independent fades in each transmission
- Probability of failure after K trials:

$$P_{out}(K) = P(\gamma < \gamma_{min})^K$$

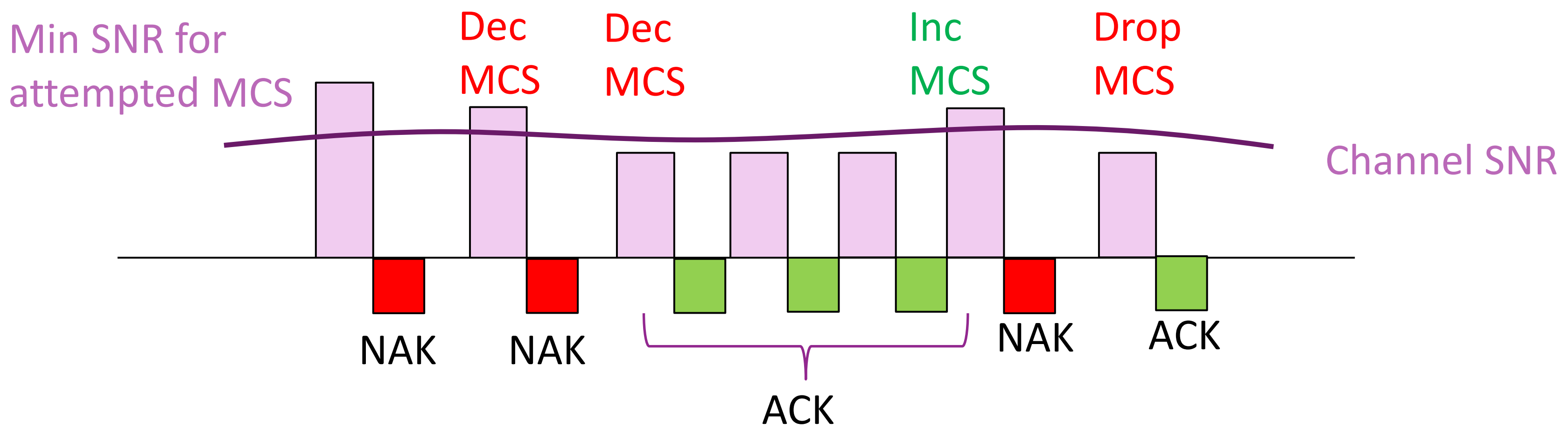
  - Decreases exponentially
  - No loss: with infinite attempts, every packet passes eventually
- Essentially exploits time diversity

Problems with ARQ

- Delay
  - Time between transmissions $\gg$ coherence time
  - Example: $f_d = 10$ Hz.
  - Say we separate transmissions by $1/(2f_d) = 0.05$ sec.
  - Four transmissions takes > 200 ms.

- Still need a reasonable estimate of SNR
  - Suppose average SNR is $\mathrm{E}(\gamma)$
  - Select an MCS with some minimum SNR γ_{min}
  - If we target MCS with $\gamma_{min} \gg E(\gamma)$: High probability of error, many wasted attempts
  - If we target MCS with $\gamma_{min} \ll E(\gamma)$: Under-utilize the channel

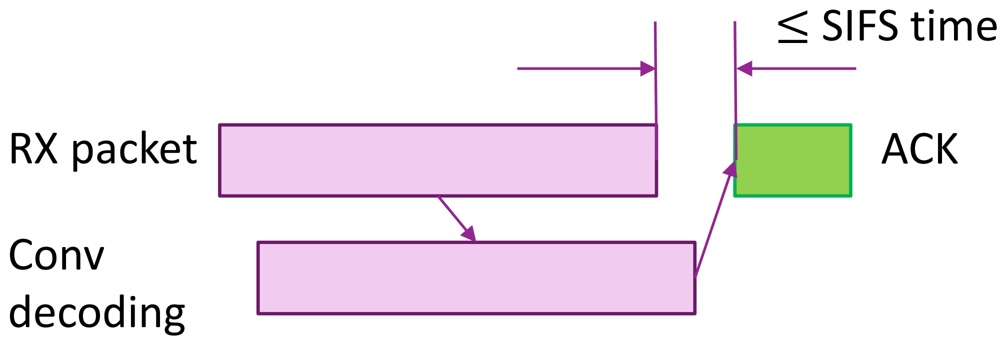
Trial and Error Rate Adaptation



- How do we set MCS in absence of channel state information?
- Typically use simple "trial and error" method:
  - No ACK => Try lower rate for retransmission
  - ACK => Occasionally try higher rate.
  - "Probe" to see if channel has improved
- Does not require any channel quality feedback.
- Widely used in 802.11 transmitters

Decoding Time in 802.11

- In 802.11 systems, ACK is sent very rapidly
- ACK sent within Short Interframe Space
  - Typically, around 10 us
  - Very challenging engineering effort
- Start decoding the packet while it is being received
  - Cannot interleave across OFDM symbols



Standard	SIFS (μs)[1]
IEEE 802.11-1997 (FHSS)	28
IEEE 802.11-1997 (DSSS)	10
IEEE 802.11b	10
IEEE 802.11a	16
IEEE 802.11g	10
IEEE 802.11n (2.4 GHz)	10
IEEE 802.11n, IEEE 802.11ac (5 GHz), IEEE 802.11ax	16
IEEE 802.11ah (900 MHz)	160
IEEE 802.11ad (60 GHz)	3

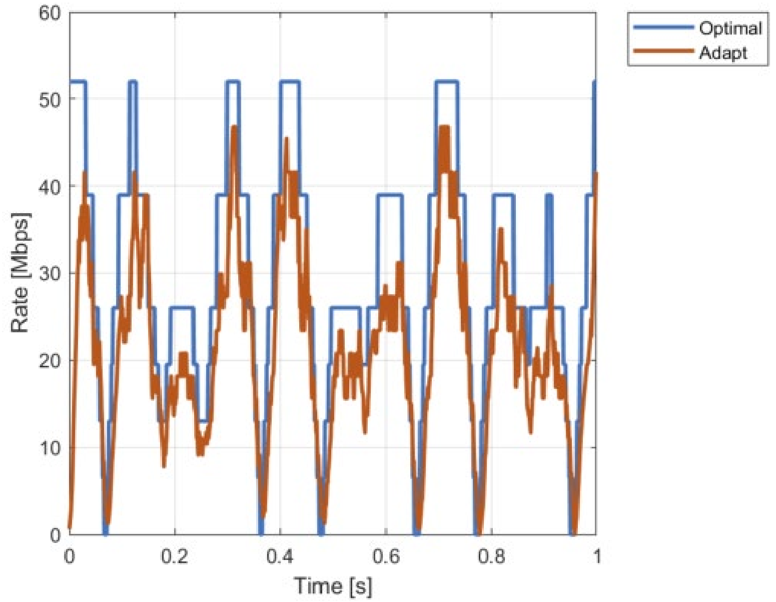
In-Class Problem

In-Class Problem

We will now try to implement a trial-and-error: Let `mcsInd(k)` be the MCS index attempted in time slot `k` and let `rateAdapt(k)` be the goodput in time slot `k`. We will adapt this MCS index as follows:

- Start with an initial MCS, `mcsInd(1)=1`.
- At each time `k`, check if `snr(k) >= minSnr(mcsInd(k))`, to see if the packet at time `k` passes.
- If the packet fails, set `rateAdapt(k)=0` since no data was transmitted. Also, decrement the MCS index for the next slot.
- If the packet passes, set `rateAdapt(k)` based on the rate achieved in that time slot. Also, with probability pup try increase the MCS index for the next slot.

Plot the `rateAdapt` vs. time.

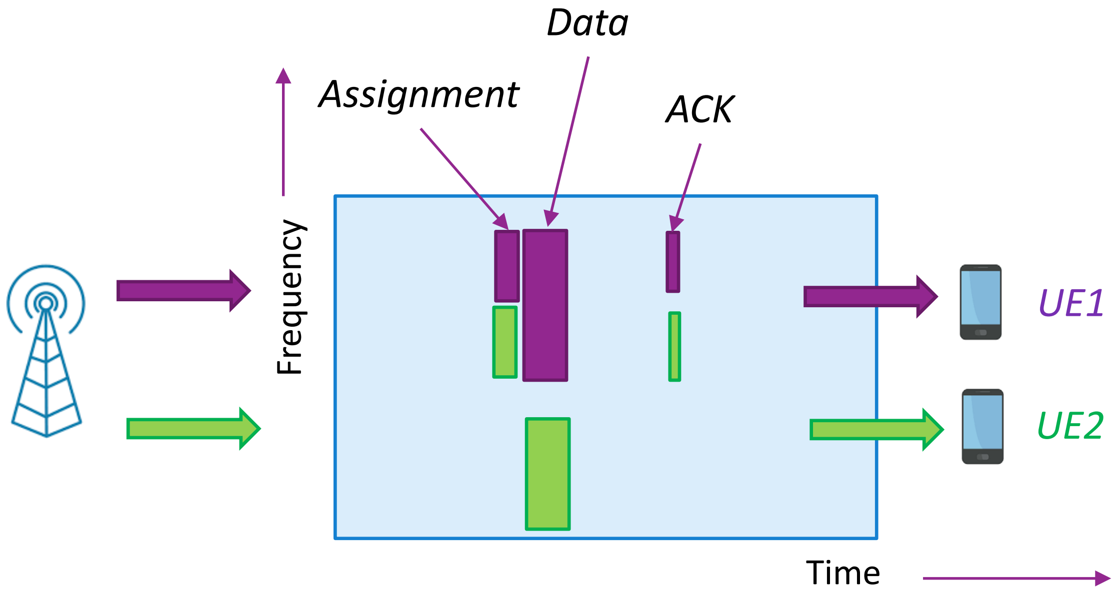


Outline

- MCS Selection and ARQ in 802.11 Wireless LAN
- MCS Selection, Scheduling and ARQ in 5G NR
- CSI Feedback and Link Adaptation in 5G NR
- Hybrid ARQ

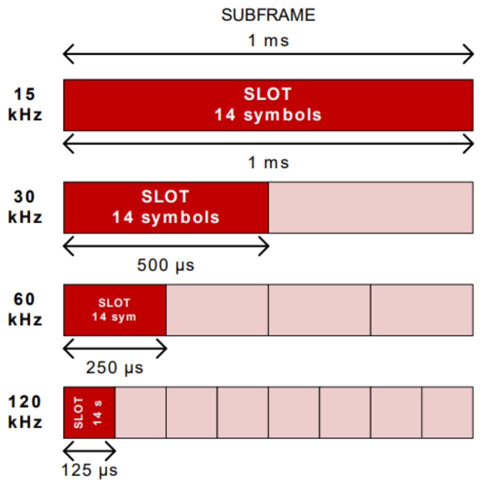
Scheduling in Cellular OFDM Systems

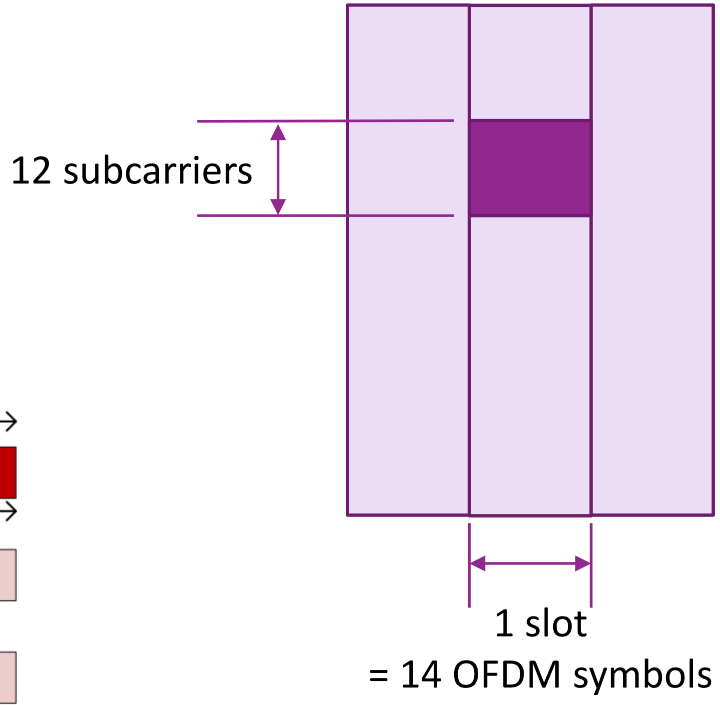
- Transmissions are scheduled by the base station
- Each assignment from the base station contains:
  - A UE identity
  - Direction (uplink or downlink)
  - Time and frequency resources for data
  - MCS for the data
  - Time and frequency resources for the ACK
  - Other formatting information
- Scheduling with OFDM enables:
  - Very flexible resource allocation among multiple users
  - Rapid adaptation to channel conditions and traffic requirements
  - Used in 4G LTE and 5G NR



Resource Blocks in 5G NR

- Assignments are in units of resource blocks
  - One slot in time = 14 OFDM symbols (normal mode)
  - 12 sub-carriers in frequency
- Each assignment has one or more RBs in a slot
- Slot time depends on sub-carrier spacing

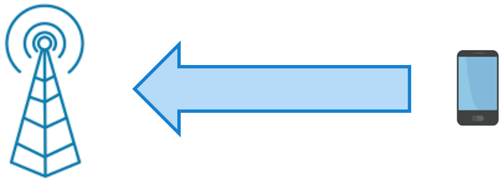




Key Channels in 5G NR Data

Downlink: gNB to UE

- ❑PDSCH: Physical downlink shared channel
  - DL data
- ❑PDCCH: Physical downlink control channel:
  - Assignments for both the uplink and downlink
  - Schedules resources on PDSCH and PUSCH



Uplink: UE to gNB

- ❑PUSCH: Physical uplink shared channel
  - UL data
  - May also contain UL ACKs
- ❑PUCCH: Physical uplink control channel:
  - Scheduling requests
  - UL ACKs (if PUSCH is not available)

PDCCH Scheduling

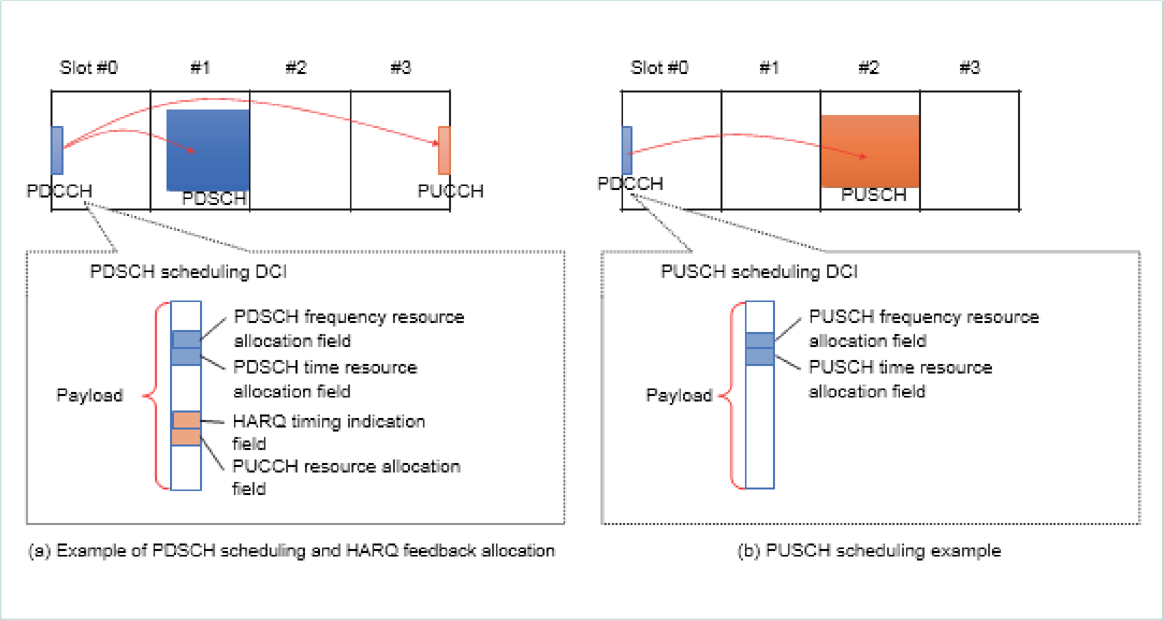


Figure 5 Example of resource allocation for data channel and HARQ-ACK

- UL and DL grants sent on PDCCH
  - DCI: Downlink control information
  - Many different DCI formats
- All DCI contain
  - Time allocation: Slot number
  - Frequency allocation: Set of RBs
  - MCS
  - ARQ location (DL grant only)
  - Many other formatting details…

5G NR MCS Options

- Transport block (TB):
  - Transmission associated with one assignment
  - Occurs in one or more RBs in a slot
- Each TB is transmitted with some MCS
  - Modulation and coding scheme
  - Currently there are 28 options
- Set by the base station depending on SNR

< 38.214 - Table 5.1.3.1-1: MCS index table 1 for PDSCH >

MCS Index I_{MCS}	Modulation Order Q_m	Target code Rate x [1024] R	Spectral efficiency
0	2	120	0.2344
1	2	157	0.3066
2	2	193	0.3770
3	2	251	0.4902
4	2	308	0.6016
5	2	379	0.7402
6	2	449	0.8770
7	2	526	1.0273
8	2	602	1.1758
9	2	679	1.3262
10	4	340	1.3281
11	4	378	1.4766
12	4	434	1.6953
13	4	490	1.9141
14	4	553	2.1602
15	4	616	2.4063
16	4	658	2.5703
17	6	438	2.5664
18	6	466	2.7305
19	6	517	3.0293
20	6	567	3.3223
21	6	616	3.6094
22	6	666	3.9023
23	6	719	4.2129
24	6	772	4.5234
25	6	822	4.8164
26	6	873	5.1152
27	6	910	5.3320
28	6	948	5.5547
29	2	reserved	
30	4	reserved	
31	6	reserved	

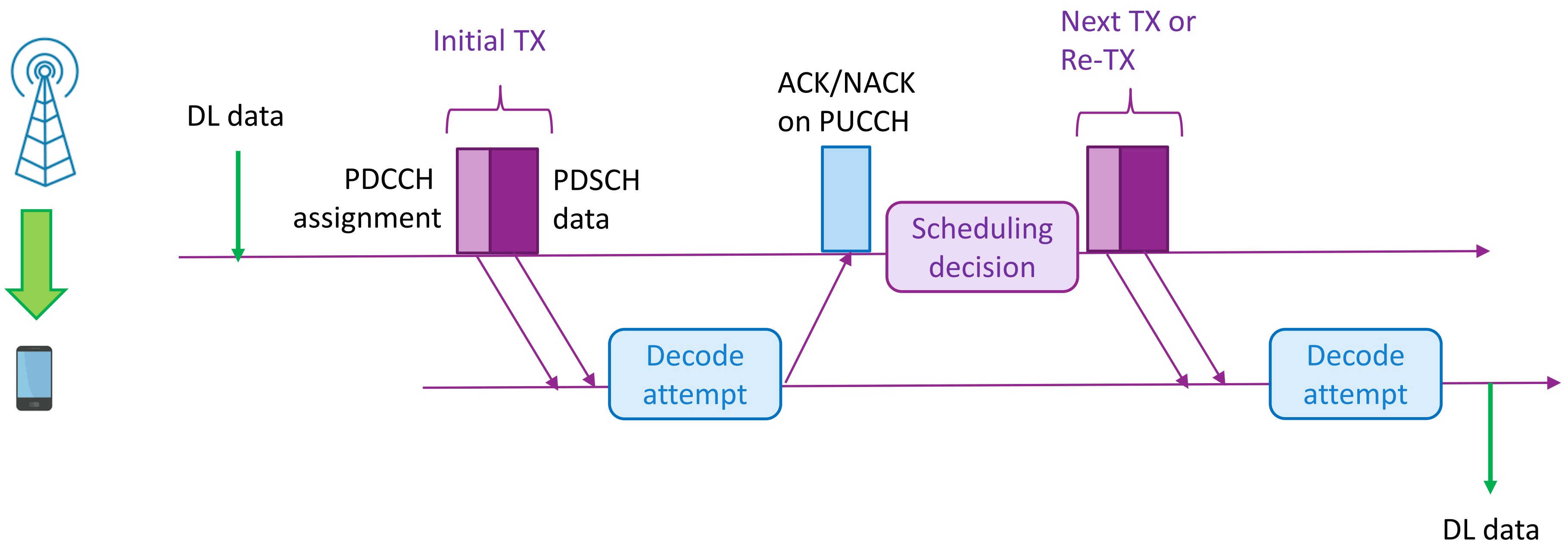
Example: Computing TB Size

- Suppose that in a system with a slot size T=125 μs, a downlink grant:
  - Spans 10 RBs
  - Uses MCS 12 (16-QAM, Code rate =434/1024)
  - On average 10 REs in each RB are used for overhead (e.g., PDCCH, reference signals, …)
- What is the transport block size and instantaneous data rate?
- Solution:
  - Each RB has 12 x 14=168 total REs.
  - There are 168-10=158 for data
  - The TB size will be:
    $b \approx 158(10)(1.6953) = 26.8(10)^3$ bits
  - The instantaneous data rate is $R = \frac{b}{T} = \frac{26.8(10)^3}{125(10)^{-6}} = 21.4$ Mbps

< 38.214 - Table 5.1.3.1-1: MCS index table 1 for PDSCH >

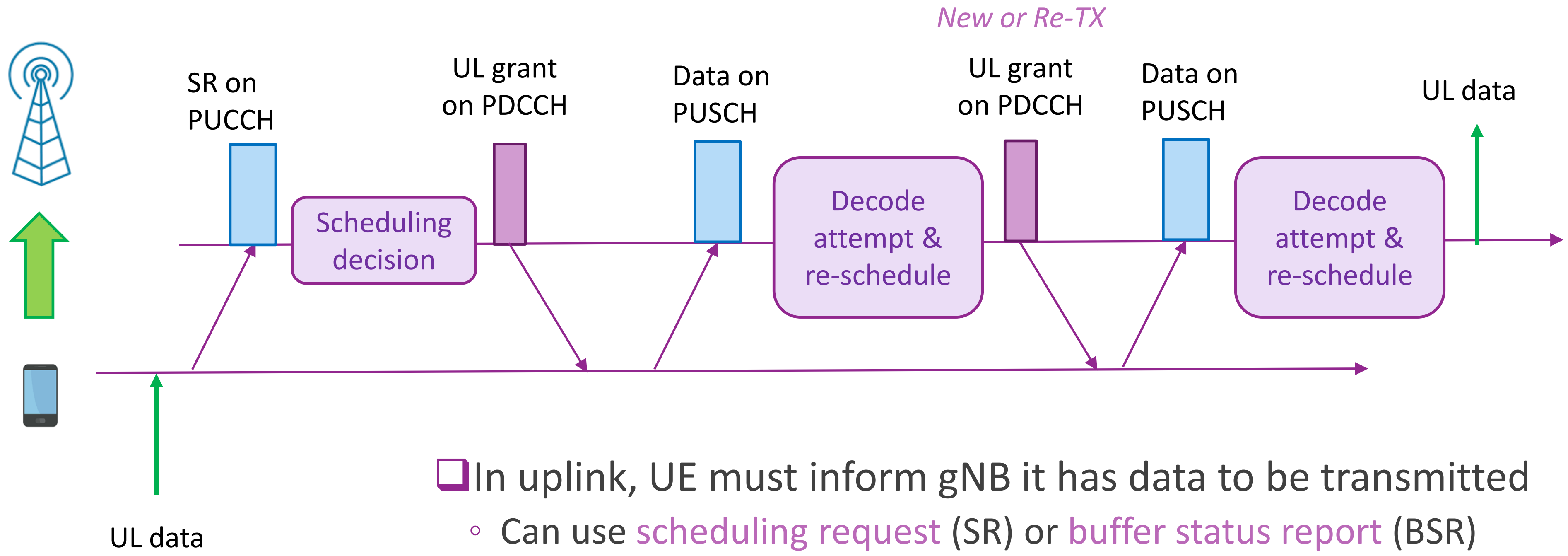
MCS Index I_{MCS}	Modulation Order Q_m	Target code Rate x [1024] R	Spectral efficiency
10	4	340	1.3281
11	4	378	1.4766
12	4	434	1.6953
13	4	490	1.9141
14	4	553	2.1602

Downlink Scheduling Timeline



- ACKs for downlink data transmitted on PUCCH
  - Physical uplink control channel
  - Can also be transmitted on the uplink data channel if it is available

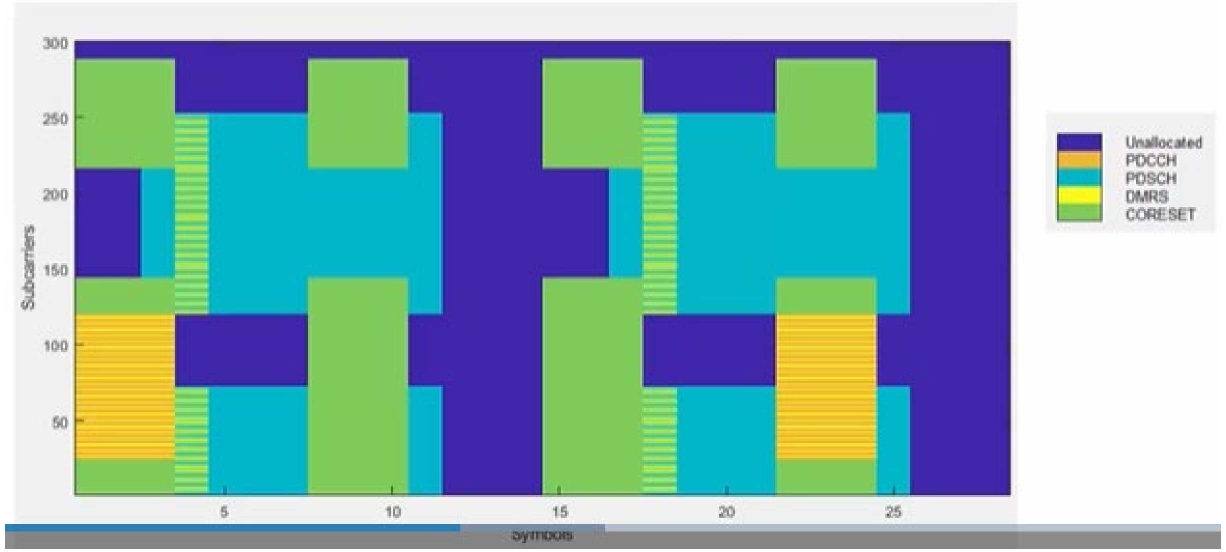
Uplink Scheduling Timeline



- In uplink, UE must inform gNB it has data to be transmitted
  - Can use scheduling request (SR) or buffer status report (BSR)
- No explicit ACK needs to be transmitted
  - gNB can grant the re-transmission or not

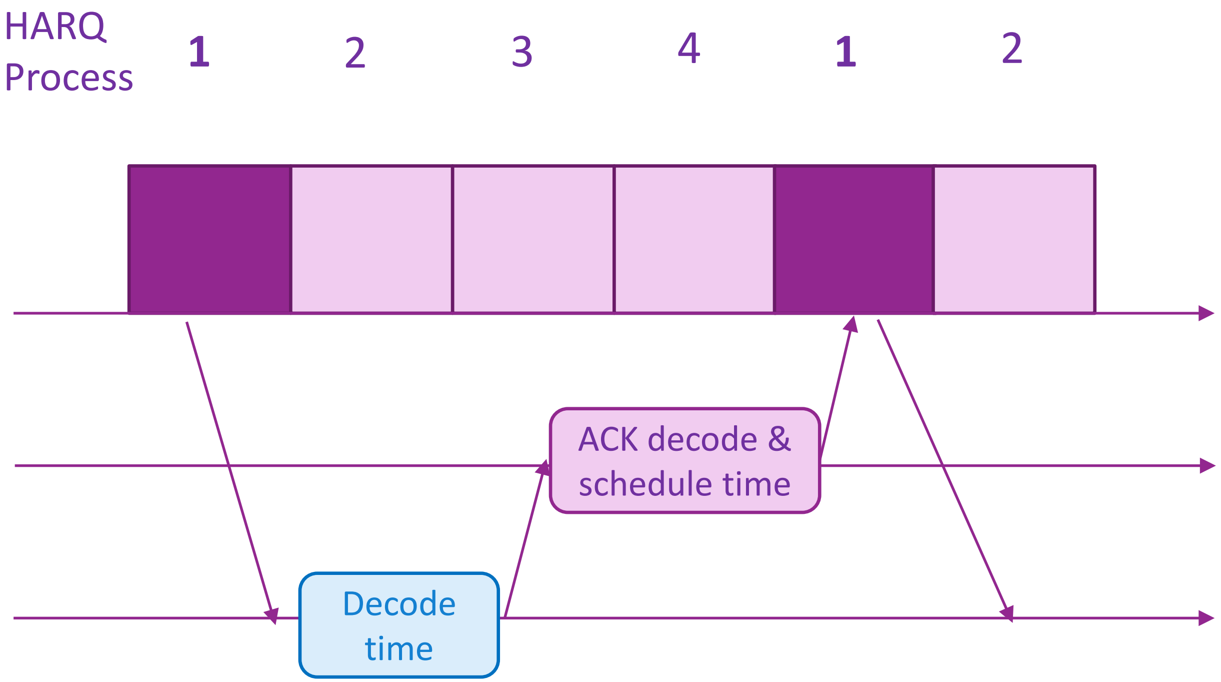
PDCCH Channel Configuration

- 5G NR provides very flexible allocation of resources to the PDCCH
  - Allocated to control resource sets (CORESETs)
  - CORESETs provide possible locations of the PDCCH assignments
  - UEs search CORESETs for PDCCH allocated to it
- Configurability can tradeoff delay, overhead, …
- See MATLAB excellent 5G Toolbox video, PDCCH mapping to Coreset, for all the details



Multiple HARQ Processes

- Transmissions involves several delays:
  - Transmission & decode of data
  - Transmission & decode of ACK
  - Decision for scheduling
- To enable continuous transmissions:
  - Use multiple HARQ processes in parallel
  - 5G NR supports up to 16
- HARQ process scheduling
  - HARQ processes indicated in DCI
  - Can be in any order
  - Very flexible



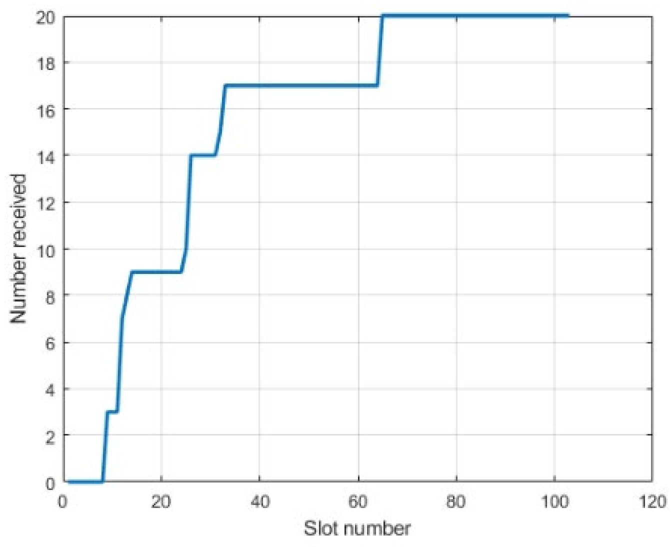
In-Class Exercise

Problem 1: Simple HARQ Simulation

Write MATLAB code to simulate the following simple HARQ process:

- The gNB transmits `ntbs` transport blocks (TBs) in slots `t=1,...,nslot`
- In slot `t`, the gNB uses a HARQ process with index `iharq = mod(t-1,nharq)+1=1,...,nharq`.
- If there was no previous transmission on that HARQ process or the previous transmission has been successfully transmitted, the gNB start transmitting the TB with the lowest index that has not previous started transmission.
- Each TB transmission fails with a probability `pfail`.
- If it fails, the TB will be retransmitted in the next slot for that HARQ process (i.e., `nharq` slots later)
- If it passes, assume it arrives at the UE a time `t+dly`.

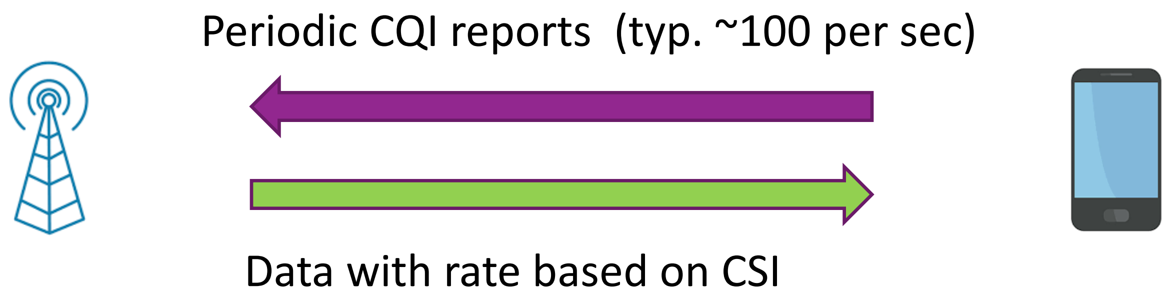
Simulate this system with the parameters below. For each `t` compute `numRx(t)`= the maximum `k` such that the first `k` TBs have been received by time `t`. Plot `numRx(t)` vs. `t`.



Outline

- MCS Selection and ARQ in 802.11 Wireless LAN
- MCS Selection, Scheduling and ARQ in 5G NR
- CSI Feedback and Link Adaptation in 5G NR
- Hybrid ARQ

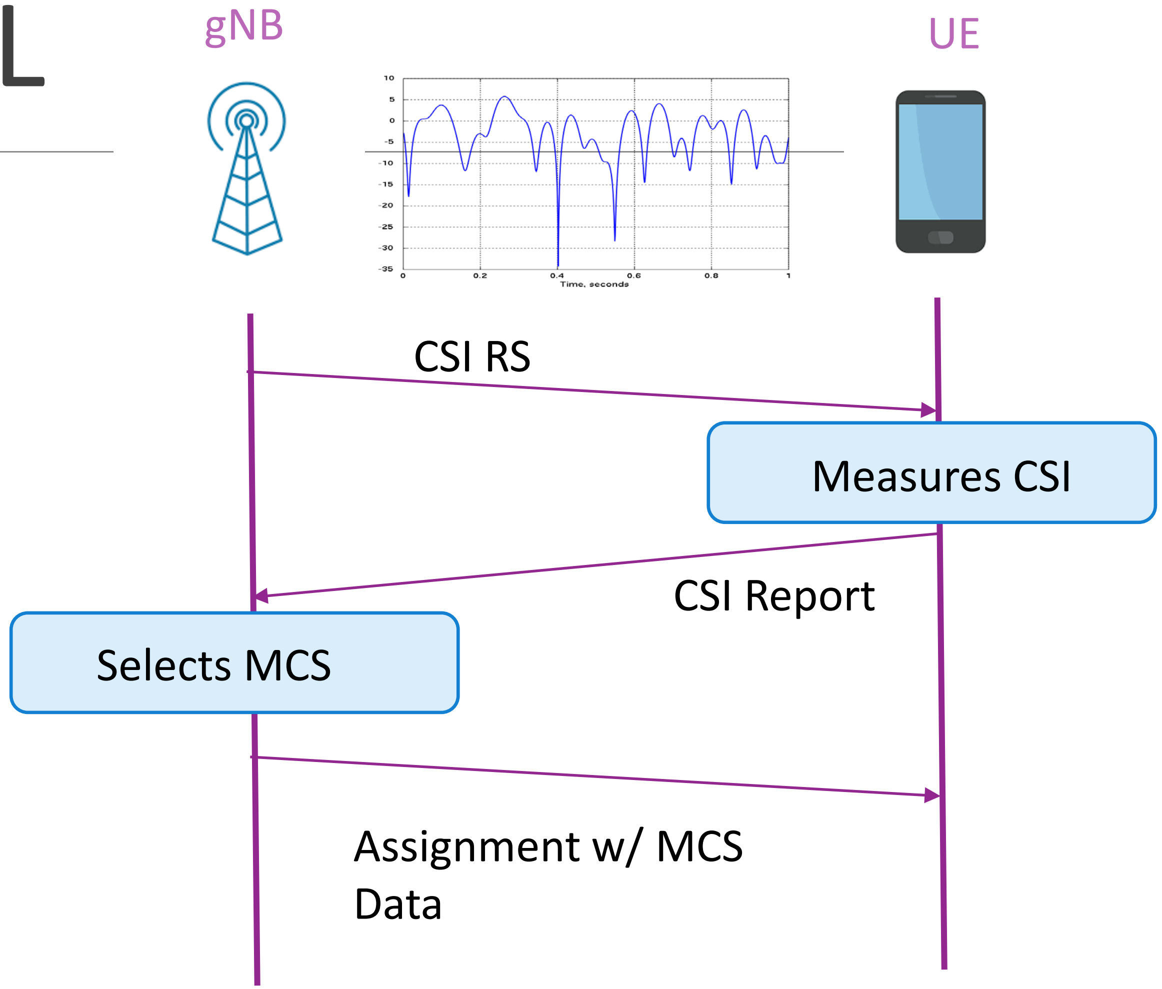
CQI Feedback



- Trial and error rate adaptation is inexact
  - Fading can result in many wasted attempts
- Ideally, TX needs to needs to know the channel quality information (CQI)
- In cellular systems, CQI is often performed via feedback
  - CQI is measured at the UE
  - Fed back to the base station in a CQI report

CSI Feedback in NR DL

- gNB sends periodic CSI RS:
  - CSI reference signals
  - Typically, 1 CSI-RS per 5 to 10ms
- UE measures CSI from the CSI-RS
- Sends back report to the gNB
- gNB can correctly select MCS



CSI-RS Configuration in MATLAB

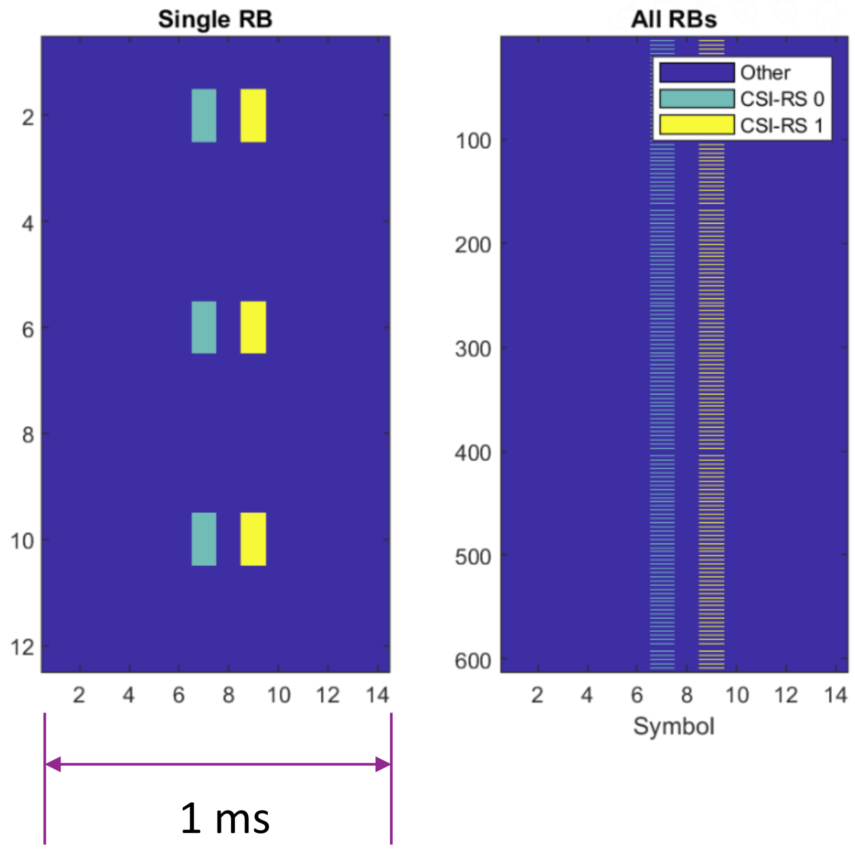
- Ex to the right:
  - One zero power (ZP) CSI-RS: Can measure noise
  - One non-zero power (NZP): Can measure signal strength
- Configurable structure
  - Periodicity
  - Density
  - Position in RB
- Trades off CSI estimation accuracy and overhead

```
csirs = nrCSIRSConfig();
% CSI-RS resource                #0        #1
csirs.CSIRSType            = {'zp', 'nzp'};

% Periodicity
% [period offset] for each CSI-RS resource.  The period is in slots.
% So, in this case, we are allocating these resources once per 10ms
csirs.CSIRSPeriod          = {[10,0], [10,0]};

% Row from TS 38.211 Section 7.4.1.5.
% Row 1 is for single port
csirs.RowNumber            = [1, 1];

% Density = 3 number of REs per RB
csirs.Density              = {'three', 'three'};

% Symbol location within the RB
csirs.SymbolLocations      = {6, 8};

% Subcarrier offset
csirs.SubcarrierLocations = {0, 0};

% Measurement bandwidth in terms of number of resource blocks
csirs.NumRB                = [NRB, NRB];
```

Visualizing the CSI-RS Configuration

- Example to the right:
  - Two CSI-RS configurations sent
  - CSI-RS0: Zero power
  - CSI-RS1: Non-zero power
  - 1 slot = 1 subframe = 1 ms
  - 15 kHz sub-carrier spacing
- Pattern repeats over RBs
- Sent once every 10 ms
- Total CSI-RS overhead in this case:

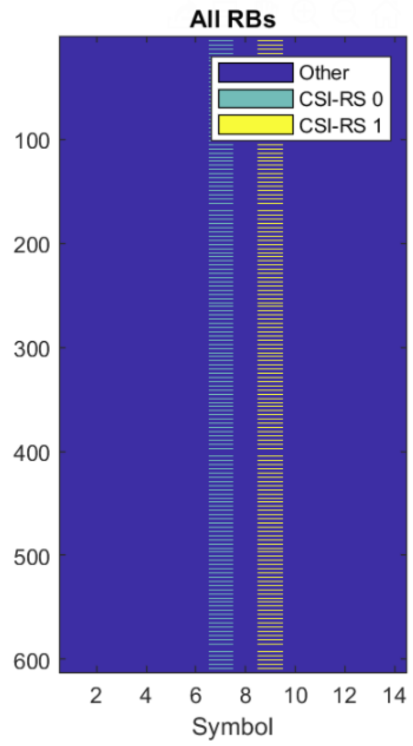
$$\frac{1}{10}\,\frac{3(2)}{(14)(12)} = 0.0036$$



Estimating the CSI using ZP and NZP

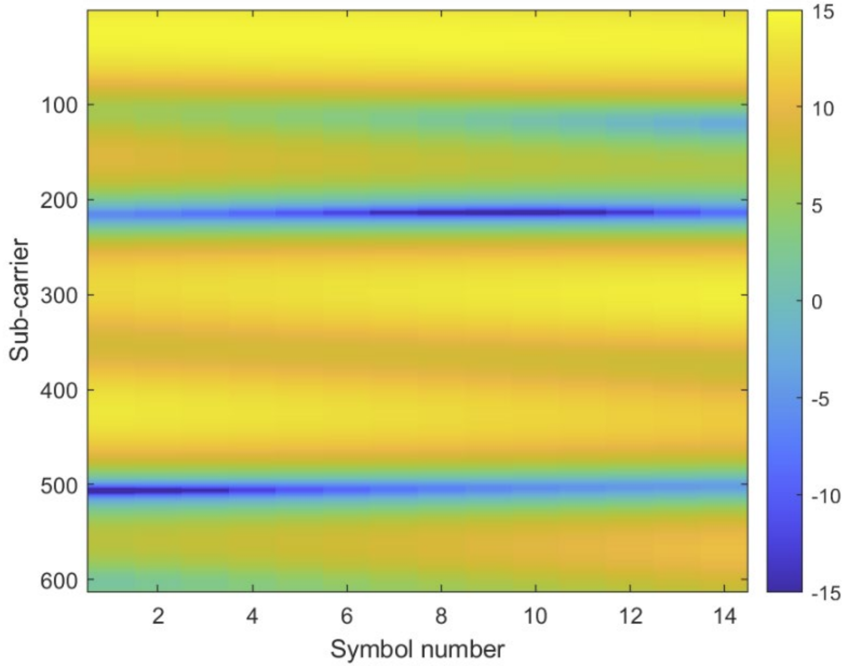
- On ZP CSI-RS:
  - There was no transmitted signal
  - Receive $r_k = w_k$ noise on the signal
  - Noise estimate $\hat{\sigma}^2 = \frac{1}{N}\sum_k |r_k|^2$
- On NZP CSI-RS:
  - Receive $r_k + h_k x_k + w_k$
  - $x_k =$ TX symbol, $h_k =$channel gain
  - Average power: $E|r_k|^2 = |h_k|^2 |x_k|^2 + \sigma^2$
  - SNR on that RE: $\gamma_k = \frac{|h_k|^2 |x_k|^2}{\sigma^2}$
  - Estimate SNR on that symbol:

$$\hat{\gamma}_k = \frac{1}{\hat{\sigma}^2} \max\{0, |r_k|^2 - \hat{\sigma}^2\}$$



Example Simulation

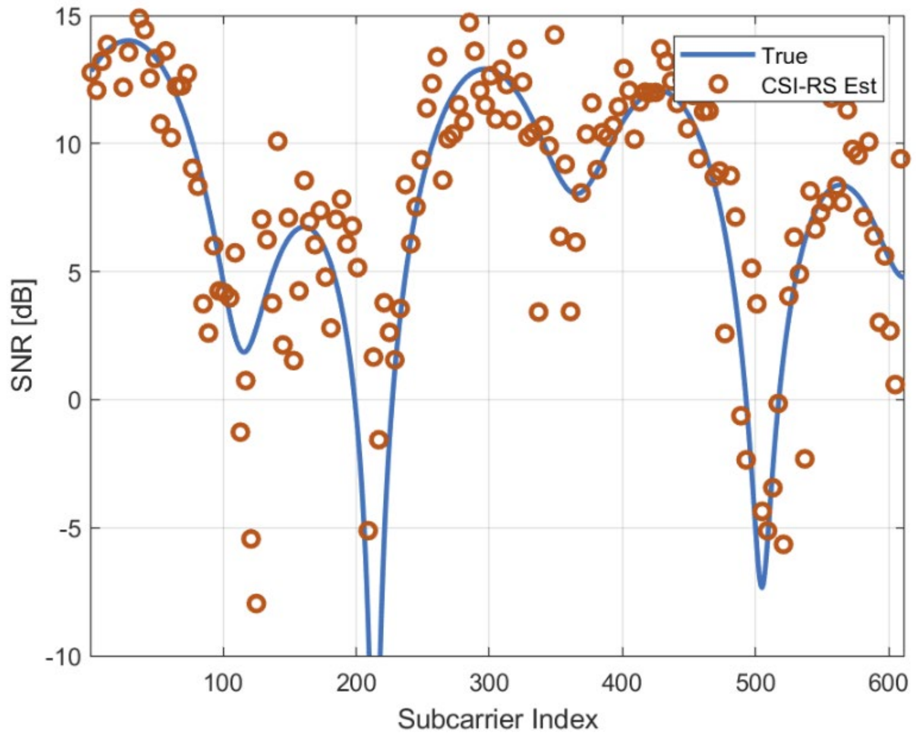
SNR [dB] per RE



- Three path channel, average SNR = 10 dB
- Simulated in frequency-domain
- Significant frequency selective fading
  - Bandwidth $W > \frac{1}{2\delta}$
  - But coherence time $>$ 1ms.

```
% Channel tap parameters
gain = [0, -3, -5]';              % Relative gain in dB
fd = [0, 100, -70]';              % Doppler shift in Hz
dly = [0, 200e-9, 500e-9]';       % Delay in seconds

% Create the channel
EsN0Avg = 10;
fdchan = FDChan(carrierConfig, 'gain', gain, ...
    'fd', fd, 'dly', dly, 'EsN0Avg', EsN0Avg);
```

Example Estimate



- Plotted:
  - True channel on symbol for the CSI-RS
  - Red: Estimates on the CSI-RS sub-carriers
- Individual CSI-RS estimates are noisy
- But averaged values can be accurate

Estimate the CSI with NZP CSI-RS Only

- Often can estimate CSI with NZP CSI-RS only
- Suppose channel is approximately constant over K REs used by CSI-RS
- Then we can model received symbols on the REs as:

$$r_k = hx_k + w_k, \qquad k = 1, \dots, K$$

  - h =Channel gain
  - x_k =known TX symbol, $w_k = CN(0, \sigma^2)$

- Using linear estimation, we can estimate channel and noise in the group

$$\hat{h} = \frac{\sum_k x_k^* r_k}{\sum_k |x_k|^2}, \qquad \hat{\sigma}^2 = \frac{1}{K-1} \sum_k |r_k - \hat{h}|^2$$

  - Factor $K-1$ is to eliminate bias in variance estimate

- Estimate for the SNR on group: $\hat{\gamma}_k = \max\left\{\frac{|\hat{h}|^2 - \hat{\sigma}^2}{\hat{\sigma}^2}, 0\right\}$
  - Subtract $\hat{\sigma}^2$ to remove noise on the transmitted signal dimension.

Implementation in MATLAB

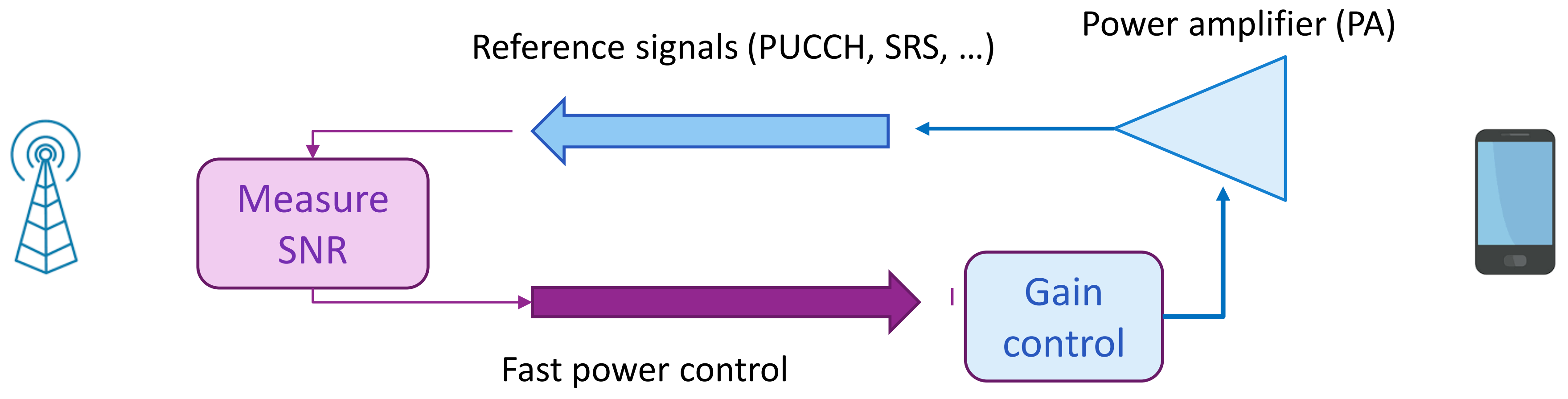
- Typically perform average in one RB
  - Assuming channel is roughly constant over this bandwidth
  - Depends on coherence bandwidth
- For each RB j get:
  - Noise estimate $\hat{\sigma}_j^2$ and channel estimate $\hat{h}_j$
- Compute wideband SNR estimate:

$$\hat{\gamma} = \max\left\{\frac{E_s - E_n}{E_n}, 0\right\}$$

  - $E_{sig} =$ average of $|h_j|^2$ in RBs
  - $E_n =$ average of $\hat{\sigma}_j^2$ in RBs
- See demo

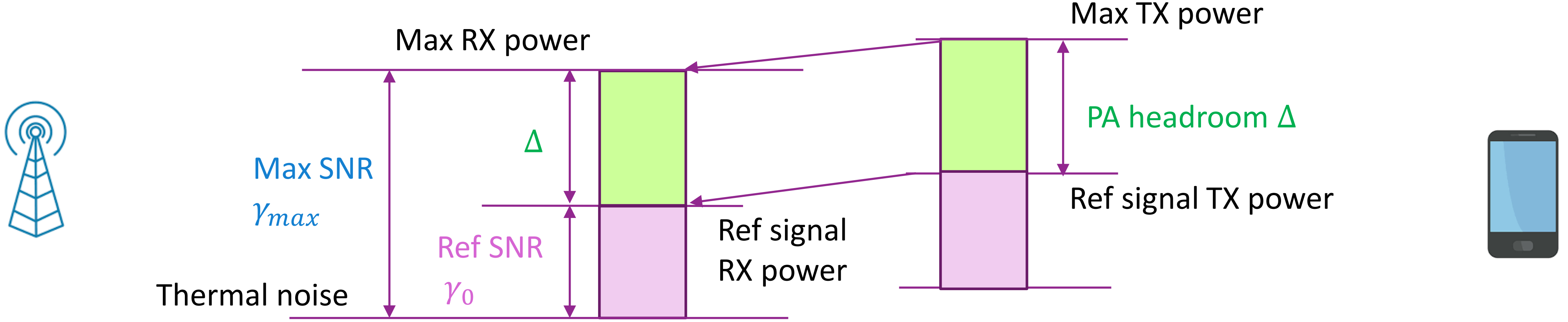
```
% Number of REs to average over
navg = 3;
nblks = floor(length(hestCsi)/navg);
hestMat = reshape(hestCsi(1:navg*nblks), navg, nblks);

% Average the results in each group
havg = mean(hestMat,1);
noiseEst = navg/(navg-1)*mean(abs(hestMat - havg).^2, 'all');

% Signal power esimate removing the noise in the band.
sigEst = max(0, mean(abs(havg).^2) - noiseEst/navg);

% Get the aveage SNR. There is a minimum to prevent very low numbers.
snrAvgEst = 10*log10(max(1e-3, sigEst / noiseEst));
```

Uplink Power Control



- ❑ Uplink is power controlled (more on this later)
  - Keep transmit RX power at a target
  - Save UE battery, reduces interference
  - Minimizes dynamic range and near-far effect in multi-user systems
- ❑ Cannot use RX SNR as a measure of channel quality

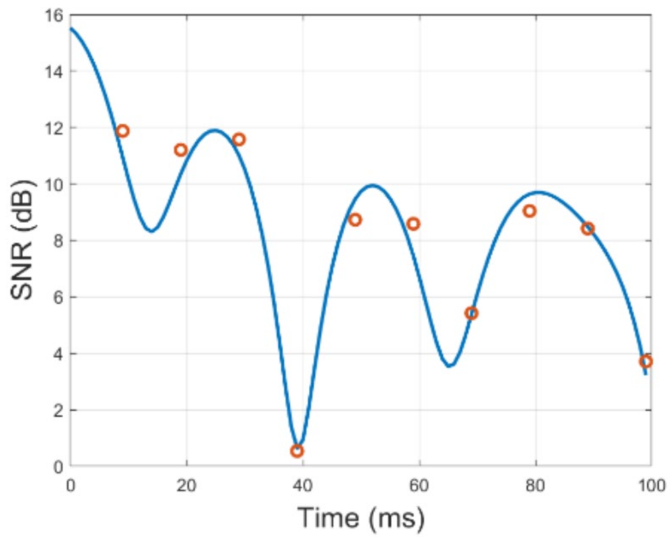
Uplink CSI



- Suppose SNR is measured on some reference signal, γ_0
  - Can be sounding reference signal (SRS), PUCCH, PUSCH, …
- UE periodically sends a power headroom (PAH), Δ
  - Difference between maximum available power and current reference signal powr
- Then, maximum SNR can be computed from $\gamma_{max} = \gamma_0 + \Delta$
- Provides basis for scheduling the UE

Errors in CSI

- In general, there will be errors in the CSI at the TX
- The RX does not measure the CSI exactly
- Time delay between:
  - The CSI measurement report and TX based on the CSI



Further Topics

- The 5G NR standard provides many powerful features for CSI reporting
  - Flexible configuration of the CSI-RS and CSI reporting
  - Rich CSI reporting: wideband, per sub-band, multiple layers, …

- Optimizing the use of the CSI-RS is a complex and interesting design problem
  - Channel quality estimation algorithms
  - Tradeoffs in overhead and accuracy
  - Channel quality tracking and prediction
  - Particularly challenging when we get to multi-antenna systems

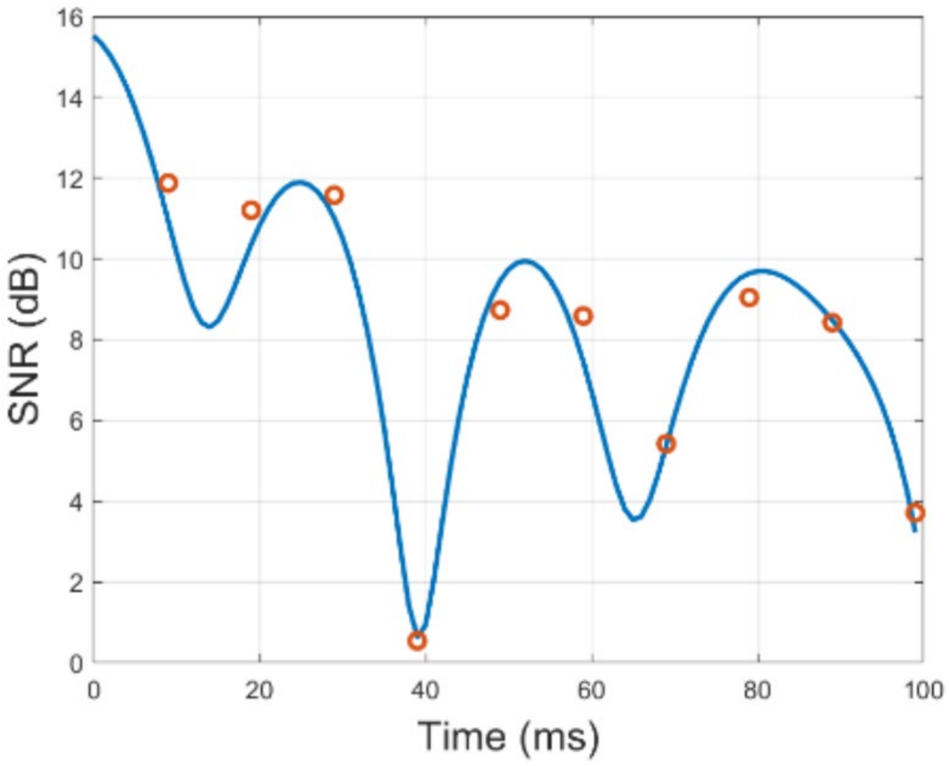
In Class Exercise: CSI Tracking

In-Class Exercise: Tracking CSI over Time

We conclude by showing how to track the channel CSI over time.

First, we configure the CSI-RS to use only the NZP CSI-RS with the same parameters as

```
csirs = nrCSIRSConfig();
csirs.CSIRSType = {'nzp'};

% TODO:  Complete the remaining parameters as above
%   csirs.CSIRSPeriod = ...
```



Outline

- MCS Selection and ARQ in 802.11 Wireless LAN
- MCS Selection, Scheduling and ARQ in 5G NR
- CSI Feedback and Link Adaptation in 5G NR
- Hybrid ARQ

Hybrid ARQ

- Standard ARQ wastes transmissions on losses
  - No data sent in each failed transmission
- Hybrid ARQ concept:
  - On each failure, store received symbols
  - Combine new symbols with old until successfully decoded
- “Failed” transmissions are not wasted
- Two methods:
  - Chase combining
  - Incremental redundancy

Combing Multiple Symbols

- Suppose we receive the same symbol x , K times:

$$r_k = h_k x + w_k, \qquad w_k \sim CN(0, N_0), \qquad E|x|^2 = E_x$$

- SNR on each transmission is: $\gamma_k = \frac{|h_k|^2 E_x}{N_0}$

- Use match filter to estimate x:

$$z = \sum_{k=1}^{K} h_k^* r_k = Gx + v, \qquad G = \sum_{k=1}^{K} |h_k|^2, \qquad v = \sum_{k=1}^{K} h_k w_k$$

  - Signal energy = $G^2|x|^2$
  - Noise energy = GN_0
  - SNR after combining: $\gamma_{tot} = \frac{G^2|x|^2}{GN_0} = \frac{G|x|^2}{N_0} = \sum_{k=1}^{K} \gamma_k$

SNR with combining is the sum of the individual SNRs

Combing Symbols Using LLRs

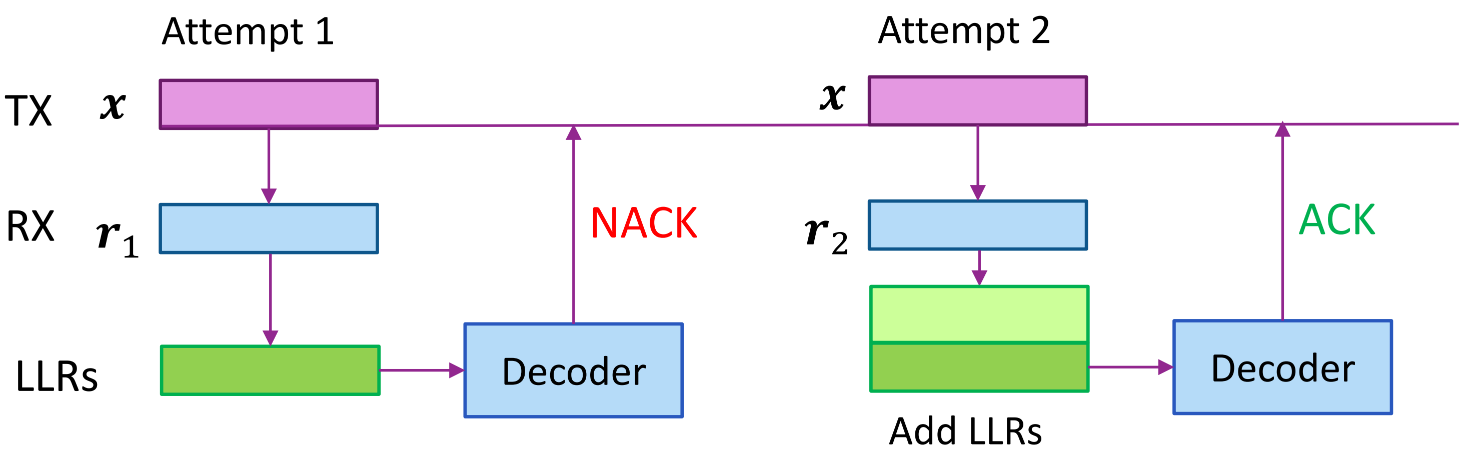
- Combining across symbols can be done by adding LLRs
- Suppose we receive the same symbol x , K times: $r_k = h_k x + w_k, \quad w_k \sim CN(0, N_0)$
- Say some coded bit c_i appears in this symbol
- The LLR using all K symbols is: $LLR_i = \ln \frac{P(r_1, \ldots, r_K | c_i = 1)}{P(r_1, \ldots, r_K | c_i = 0)}$
- Assuming independence of noise in difference symbols:

$$LLR_i = \ln \frac{P(r_1|c_i = 1) \cdots P(r_K|c_i = 1)}{P(r_1|c_i = 0) \cdots P(r_K|c_i = 0)} = \ln \frac{P(r_1|c_i = 1)}{P(r_1|c_i = 0)} + \cdots + \ln \frac{P(r_K|c_i = 1)}{P(r_K|c_i = 0)}$$

$$= \quad LLR_{i1} + \cdots + LLR_{iK}$$

- We can simply add the LLRs obtained from K transmissions

HARQ Chase Combining



- ❑Transmit symbols $x = (x_1, \dots, x_n)$ in each attempt
- ❑RX computes the LLRs for the block
- ❑Adds to previous LLRs
- ❑Continues to add LLRs until decoder passes

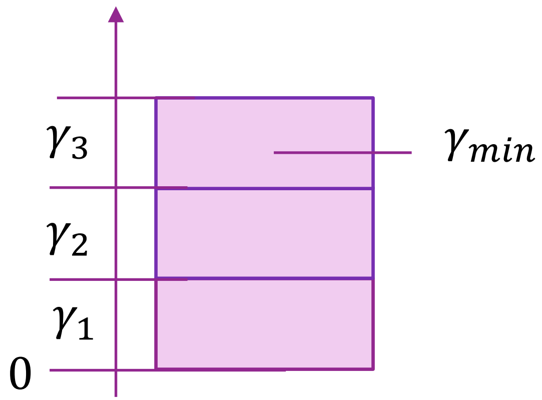
Chase Combining and SNR Accumulation

- Suppose that we modulate a packet that needs γ_{min} SNR
  - Achieves a rate per symbol $R = \alpha \log_2(1 + \gamma_{min})$
  - That is, achieves Shannon capacity within a fraction α
- We transmit K times
  - In each transmission, all symbols experiences some SNR γ_k
- Packet will pass on transmission K when:

$$\gamma_1 + \cdots + \gamma_{K-1} \le \gamma_{min} \le \gamma_1 + \cdots + \gamma_K$$

- If increments are small, $\gamma_{min} \approx \gamma_1 + \cdots + \gamma_K$
- Effective Rate $R_{Chase} = \frac{R}{K} = \frac{\alpha}{K} \log_2(1 + \gamma_{min}) \approx \frac{\alpha}{K} \log_2(1 + \sum_{k=1}^{K} \gamma_k)$

Accumulated SNR



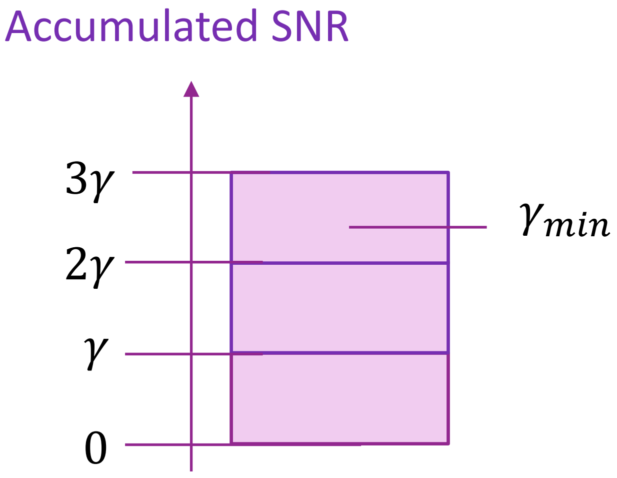
Delay-Accuracy Tradeoff

- Suppose that SNR in each transmission is a constant γ
- Will terminate after K transmissions when:

$$\gamma(K-1) \le \gamma_{min} \le \gamma K$$

- Worst case delay: $K \le \gamma_{min}/\gamma$
- Worst case SNR error:

$$\frac{\text{RX SNR}}{\text{Reqd SNR}} = \frac{\gamma K}{\gamma_{min}} \ge \frac{K}{K-1}$$



- Delay-accuracy tradeoff: With larger delay K, received SNR matches the required SNR better

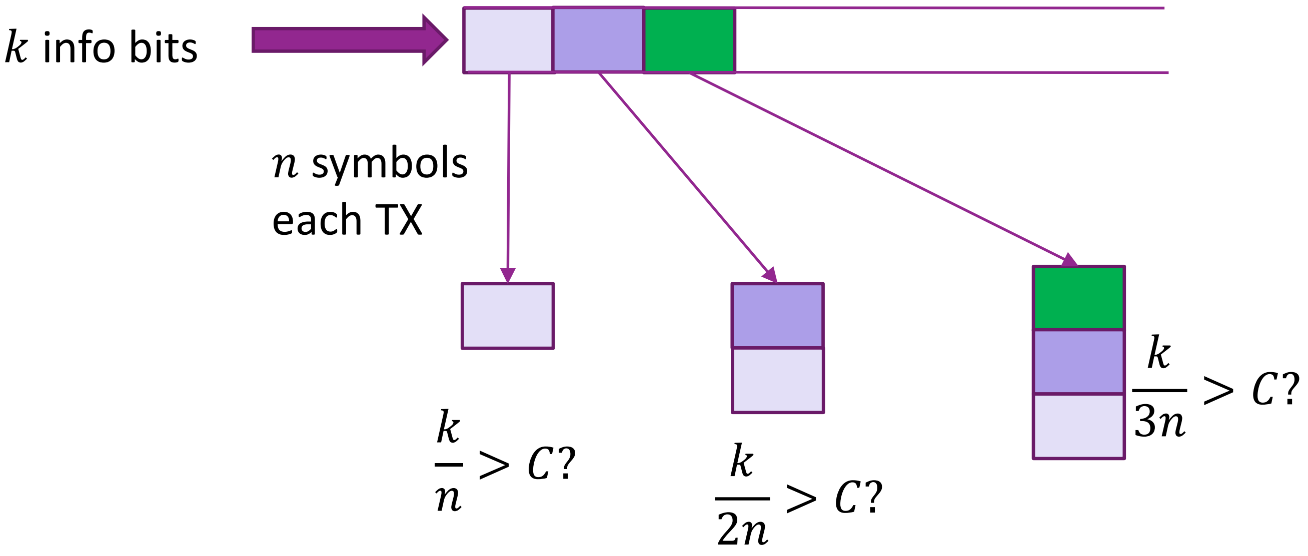
Rateless Gaussian Random Code

- Chase combining effectively creates a repetition code
  - Not optimal
- Alternate method: "infinite" random code:
  - k information bits, $M = 2^k$ messages
  - Each code word is an infinite iid sequence

$$s_m = (s_{m1}, s_{m2}, \dots), \qquad s_{mi} \sim CN(0, E_s)$$

- If we take first n symbols where n is large, we get
  - Code that is optimal for AWGN or fading channel
  - Code can be reliably received if $\frac{k}{n} > C$,
  - $C =$ avg capacity per symbol.

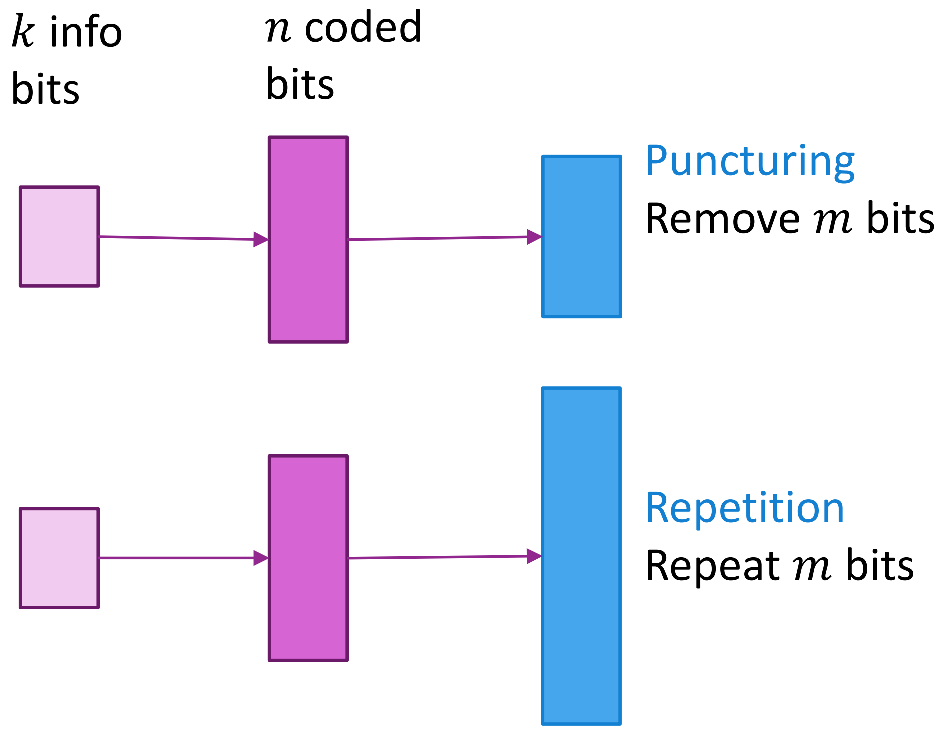
Using a Rateless Code



- Suppose capacity C not known
- Terminate when $\frac{k}{n(d-1)} < C < \frac{k}{nd}$
- Can achieve arbitrarily close to capacity without knowing anything!
  - Works for fading or constant channels

Rate Matching via Puncturing and Repetition

- Many codes have limited rate options
  - Ex: Convolution codes, $\frac{1}{2}, \frac{1}{3}$
- Two options to get different rates
- Puncturing
  - Take (n, k) "base code" or "mother code"
  - Remove m of the n code bits
  - Effective rate = $\frac{k}{n-m}$
  - At decoder, set LLRs to zero for punctured bits
- Repetition
  - Repeat m of the n code bits
  - Effective rate = $\frac{k}{n+m}$
  - At decoder, add LLRs for repeated bits



Practical Rateless Codes

- Incremental redundancy
  - Code data with low rate “mother code” (Usually rate 1/3 or 1/5)
  - In first TX send systematic bits + some parity
  - Decoder treats missing bits as erasures.
  - If fail, TX sends more parity bits
  - Continue until it passes
- In 3GPP codes, each TX is called a redundancy version
- First proposed by Mandelbaum 1974

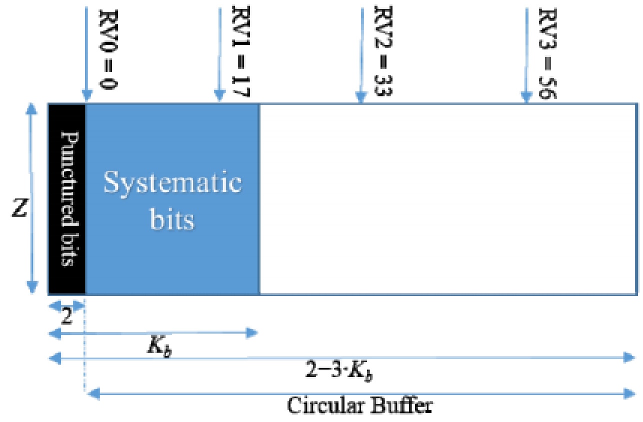
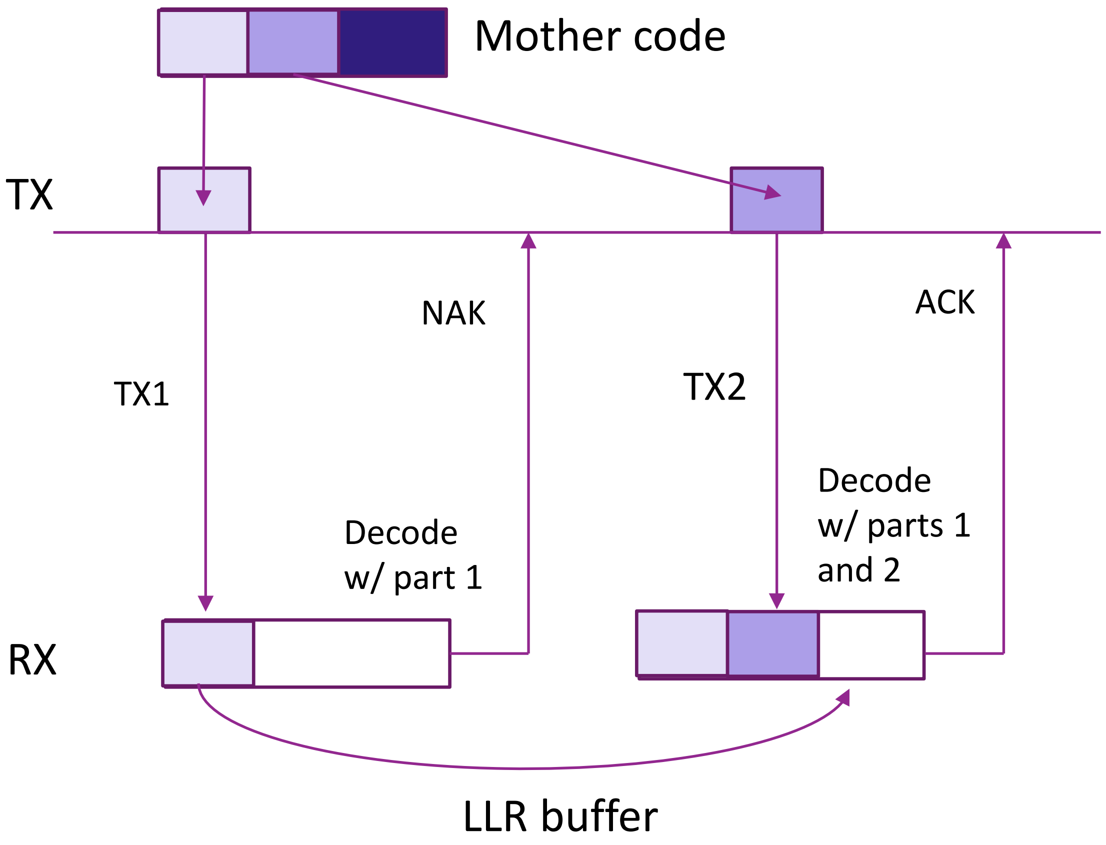


Fig. 2. NR LDPC circular buffer and RV placements, for BG1.

Hamidi-Sepeh et al, Analysis of 5G LDPC Codes Rate-matching Design, 2018

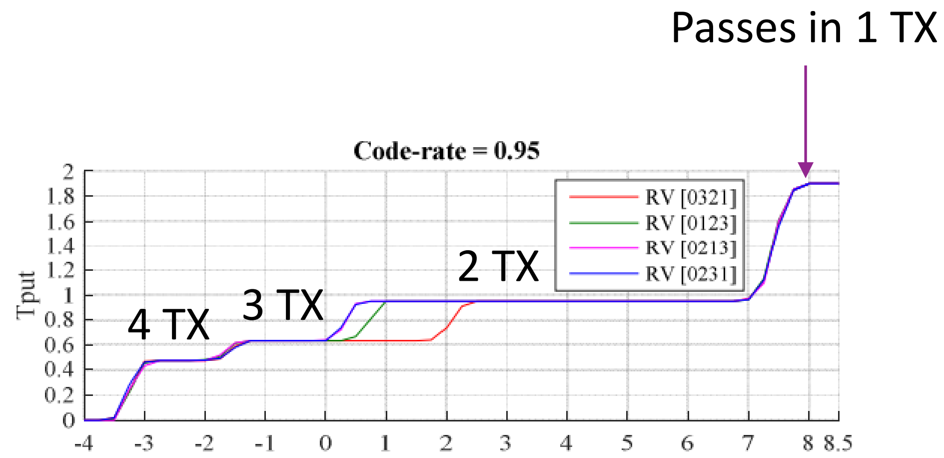
IR Illustrated

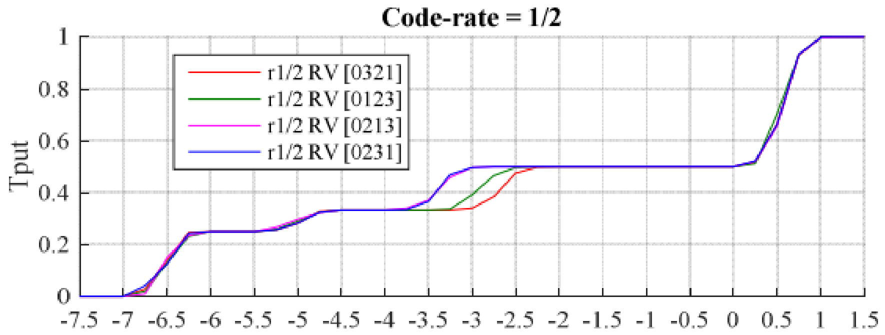


- Decode 1:
  - Use part 1
  - Treat parts 2, 3 as erasures
- Decode 2:
  - Use parts 1,2
  - Treat part 3 as erasure

5G NR HARQ Performance

- Simulation with QPSK
- With multiple transmissions:
  - Can obtain reasonable over wide range of SNRs
  - Staircase throughput curve
- Selecting wrong MCS may not lead to poor rate:
- Ex: At SNR=1 dB
  - Get ~0.95 bps/Hz with option 1 (QPSK, rate=0.95)
  - Get ~1 bps/Hz with option 1 (QPSK, rate=0.5)





Hamidi-Sepeh et al, Analysis of 5G LDPC Codes Rate-matching Design, 2018

Analysis

- Exact analysis complex
  - Depends on puncturing and precise fading pattern
  - Must use simulation

- Effective SNR per symbol
  - Computes an “effective SNR”
  - Accounts for fading and HARQ
  - Many simulations show this model predicts actual performance on real codes well
  - Used widely in network simulations to avoid simulating the LDPC codes

Effective SNR

- Suppose
  - Mother code b_0 information bits, modulated over N_0 symbols
  - Transmit N symbols. May have $N \neq N_0$ due to puncturing or repetition
  - Each transmitted symbol experiences SNR $\gamma_i, i = 1, \dots, N$
- Effective SNR per symbol: Value of $\hat{\gamma}_s$ such that:

$$\log_2(1 + \hat{\gamma}_s) = \frac{1}{N_0} \sum_{i=1}^{N} \log_2(1 + \gamma_i).$$

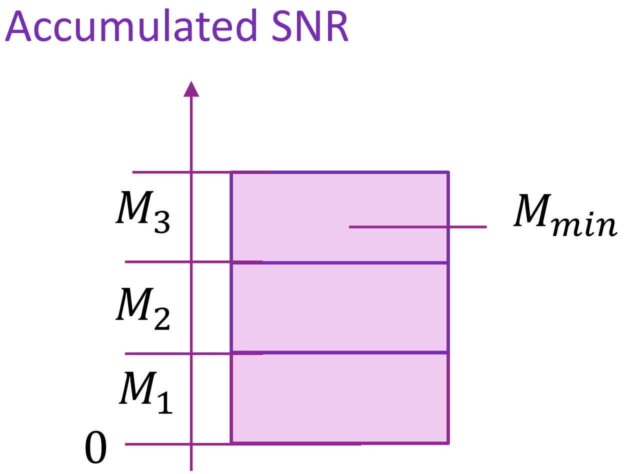
  - Represents effective mutual information per symbol in the original codeword
- Key result: For most practical codes:
  - Performance on a fading channel with rate matching ≈ Performance on a constant channel with $\hat{\gamma}_s$

IR and Mutual Information Accumulation

- Suppose that we modulate a packet that needs γ_{min} SNR
  - Achieves a rate per symbol $R = \alpha \log_2(1+\gamma_{min})$
  - That is, achieves Shannon capacity within a fraction α
- We transmit K times with $1/K$ symbols in each transmission
  - In each transmission, all symbols experiences some SNR γ_k
- Packet will pass on transmission K when:

$$\frac{1}{K}(M_1 + \cdots + M_{K-1}) \le \log_2(1+\gamma_{min}) \le \frac{1}{K}(M_1 + \cdots + M_K)$$

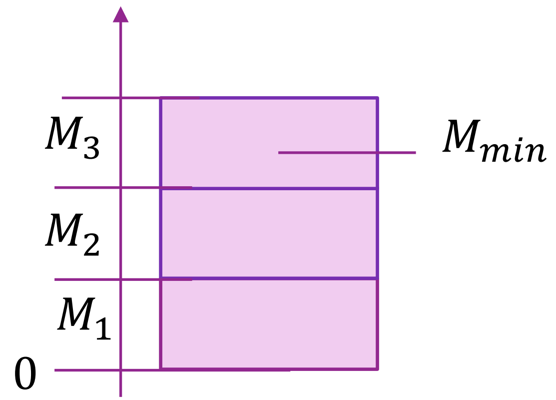
  - $M_k = \log_2(1+\gamma_k)$= mutual information per symbol in transmission k



Consequences of MI Accumulation

- Approximately capacity achieving:
  - Suppose that increments are small
  - Then, $\log_2(1+\gamma_{min}) \approx \frac{1}{K}(M_1 + \cdots + M_K)$
  - Rate will be: $R_{IR} = R = \alpha \log_2(1+\gamma_{min}) \approx \frac{\alpha}{K}\sum_{k=1}^{K} \log_2(1+\gamma_k)$
  - Ergodic capacity, $C = \frac{1}{K}\sum_{k=1}^{K} \log_2(1+\gamma_k)$
  - Hence, $R_{IR} = \alpha C$
- Outperforms Chase combining
  - In identical situation, we saw $R_{Chase} = \frac{\alpha}{K}\log_2(1+\sum_{k=1}^{K}\gamma_k)$
  - By Jensen's inequality: $R_{IR} \geq R_{Chase}$
  - IR adds degrees of freedom. Chase adds power.

Accumulated SNR



Practical Considerations

- Cellular standards use very powerful coding methods
- Provides excellent performance
- But presents several implementation challenges
- Often need very large LLR buffer
  - Must be stored on chip
  - LLR memory often larger than decoder or even the entire modem!
- Decoding delay:
  - Typical turnaround time is several slots (100s of us to milliseconds)
  - Much longer than WiFi (Recall SIFS time is 10 us)

Lab: Multi-Process NR HARQ Simulation

Simulating the 5G NR PDSCH with Multi-Process HARQ

In the previous lab, we created a simple frequency-domain channel along with a gNB T
multi-process Hybrid ARQ. The MATLAB's 5G Toolbox has excellent tools to simulate

In going through the lab, you will learn to:

- Create PDSCH allocations for a subset of the RBs
- Measure the time-varying wideband SNR and the SNR on a subset of the reso
- Encode and decode data with HARQ using MATLAB 5G Toolbox tools
- Create a simple multi-process TX and RX
- Evaluate the performance of the link with HARQ on a fading channel

Some of the code in this lab is based on MATLAB's PDSCH throuhput demo. You will
used some functions that are not available in R2020A. Also, along with this file, you wi

- `FDChan.m`: A class for frequency-domain channel simulation
- `NRgNBTxFD.m`: A class for NR gNB TX frequency-domain simulation
- `NRUERxFD.m`: A class for NR UE RX frequency-domain simulation