


Program Współpraca Polska-RPA	RAPORT CZĄSTKOWY z realizacji projektu w ramach programu międzynarodowego Polska-RPA			 Narodowe Centrum Badań i Rozwoju
Nr raportu	IR-RATfor5G-05			
Data aktualizacji raportu:	2021.11.17	Wersja	4	
Numer umowy	PL-RPA2/02/RATfor5G+/2019	Akronim	RATfor5G+	
Okres realizacji projektu	Od	2019.01.01	do	2022.6.30
Tytuł projektu	Technologie dostępu radiowego dla standardu 5G i przyszłych generacji sieci bezprzewodowych			
Tytuł raportu	Wyniki symulacji – cz.1			

Poniżej przedstawiono wyniki symulacji w pierwszej części projektu. Zaimplementowano tzw. podejście „dwu-fazowe” dołączania funkcji waterfilling dodanej przez zespół badawczy UTP/PBS. Porównanie „single-phased” (kod oryginalny) vs „two-phased” podejścia do symulacji, z uwzględnieniem modyfikacji zespołu RATfor5G+ oraz kodu oryginalnego. Kolejno realizowane testy mają na celu pokazać, że modyfikacje oryginalnego kodu symulatora 5G LL Vienna, poczynione przez zespół nie wpływają negatywnie na poprawność jego działania, oraz przedstawiają wydajność symulatora rozbudowanego o algorytm waterfilling.

Cel:

- Potwierdzić czy implementacja obu podejść nie zmienia sposobu działania symulatora w trybie „domyślnym” (tj bez wterfilling)
- Na maszynie ratfor5g → „two-phased”
- Na maszynie ratfor5g-second → „single-phased”

Założenia:

- Scenariusz (NOMA_F3K_PdA_xxxms_YYY)
 - BS1: UE1, UE2 / 36 subcarriers per user
 - BS2: UE3:36, UE4:36 (OFDM), UE5, UE6 (NOMA)
 - Xxx – prędkości: 0,10,20ms
 - YYY – SinglePhased, TwoPhased
- Ustawienia główne
 - `scStr.simulation.pilotSpacingFrequency` = 2;
 - `scStr.simulation.pilotPatternUplink` = 'Rectangular';
 - `scStr.simulation.pilotPatternDownlink` = 'Rectangular';
 - `scStr.simulation.nFrames` = 3000;
- Waterfilling
 - **WYŁĄCZONY** – dla obu przypadków żeby nie wprowadzać modyfikacji do wyników

Wyniki

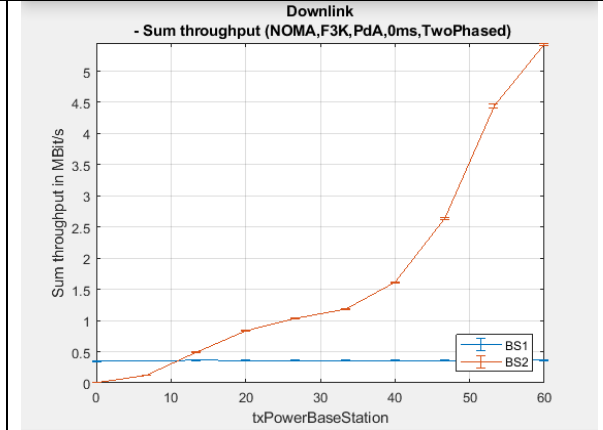
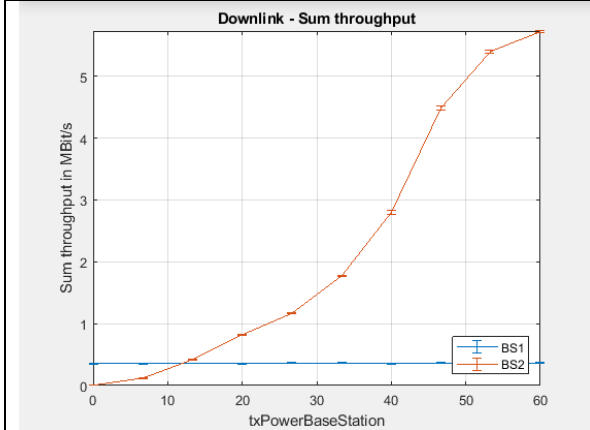
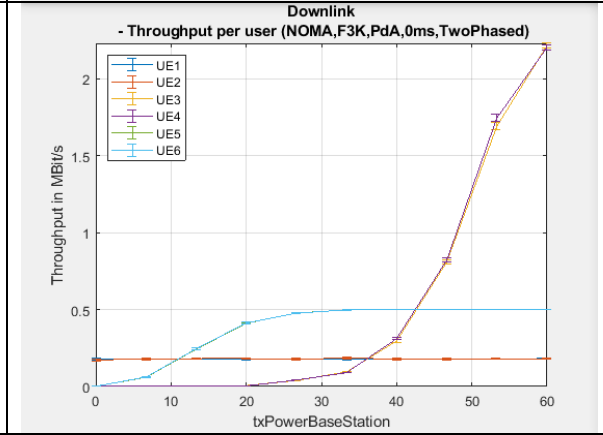
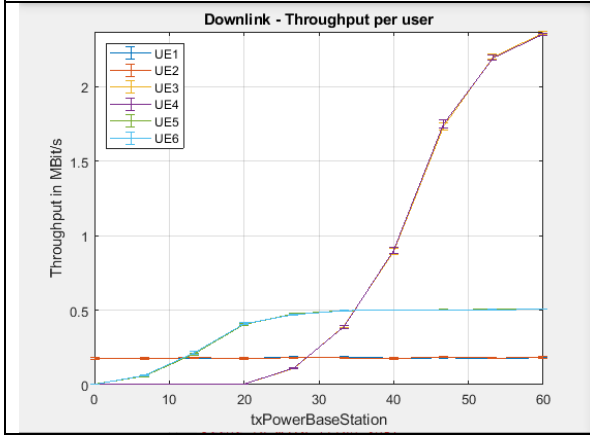
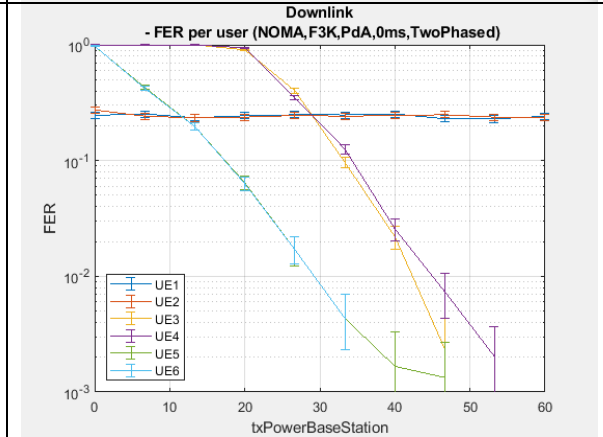
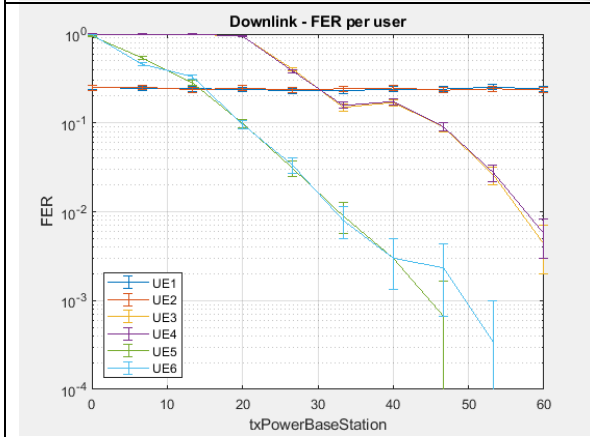
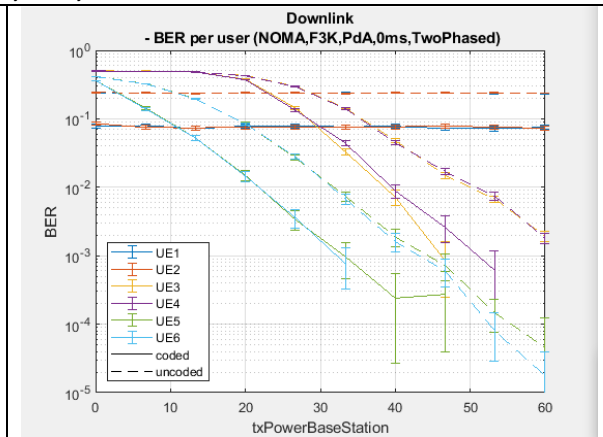
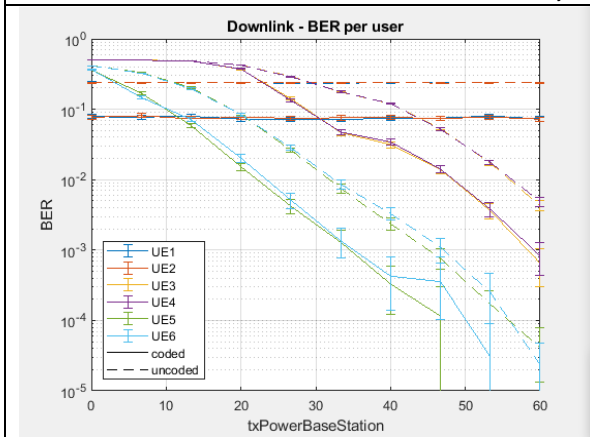
- czas uruchomienia

- 8:39 – SinglePhased
- 8:49 – TwoPhased

SinglePhased (aka "user-level waterfilling")

TwoPhased (aka „system level waterfilling")

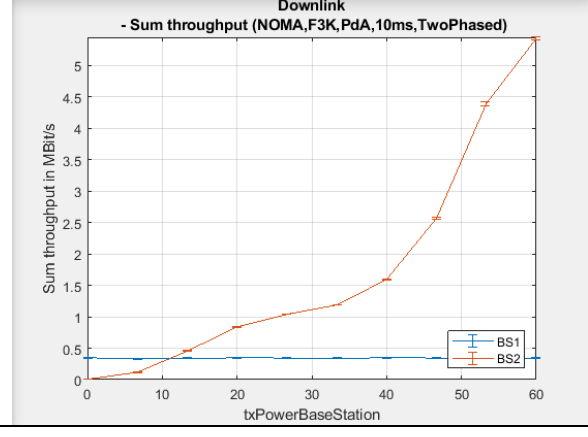
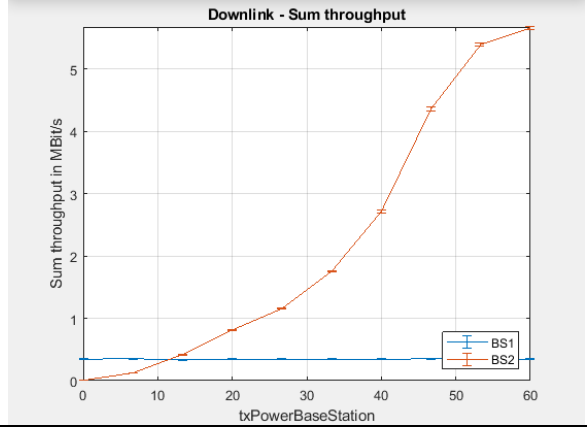
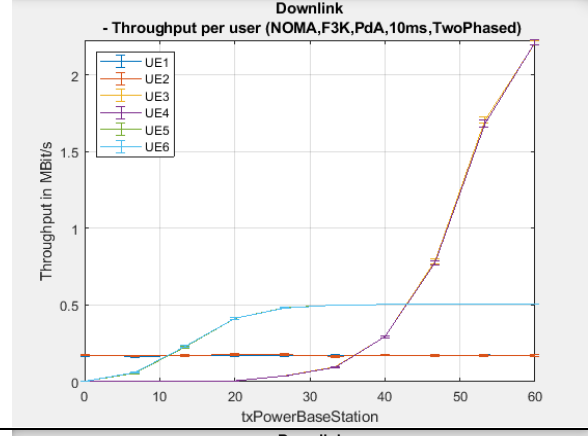
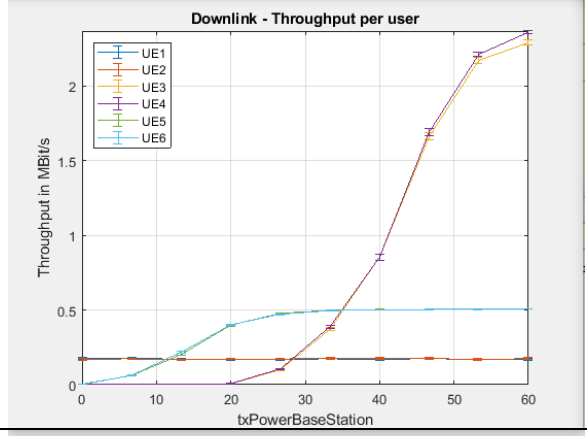
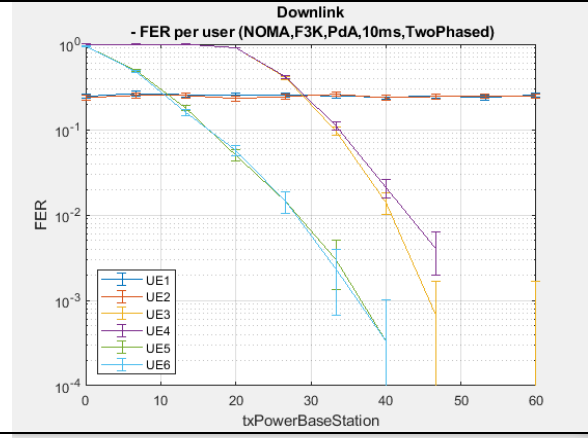
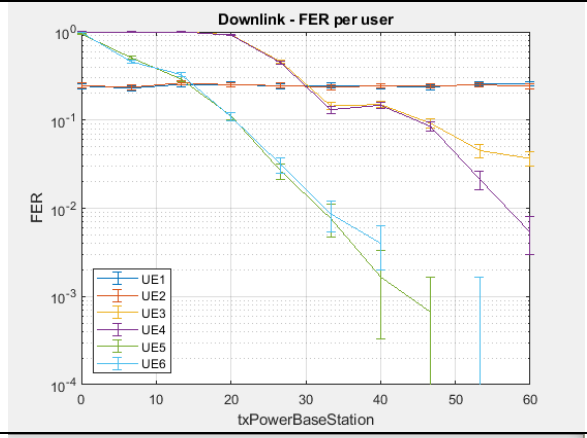
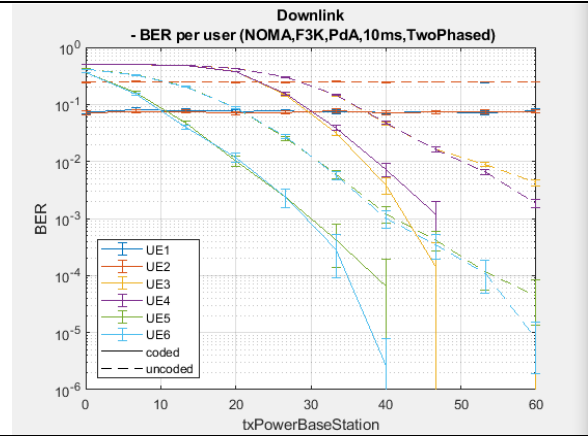
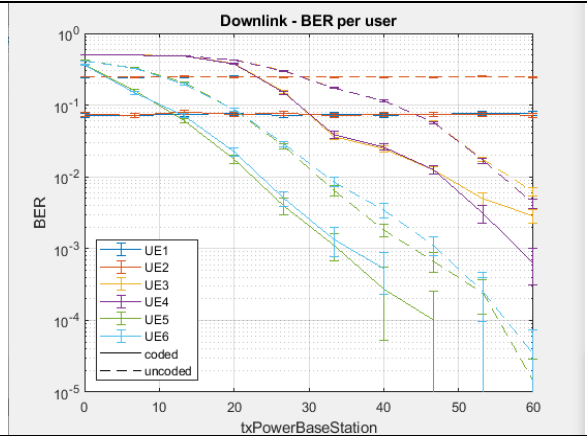
NOMA, F3K, PdA, 0ms



SinglePhased (aka "user-level waterfilling")

TwoPhased (aka „system level waterfilling")

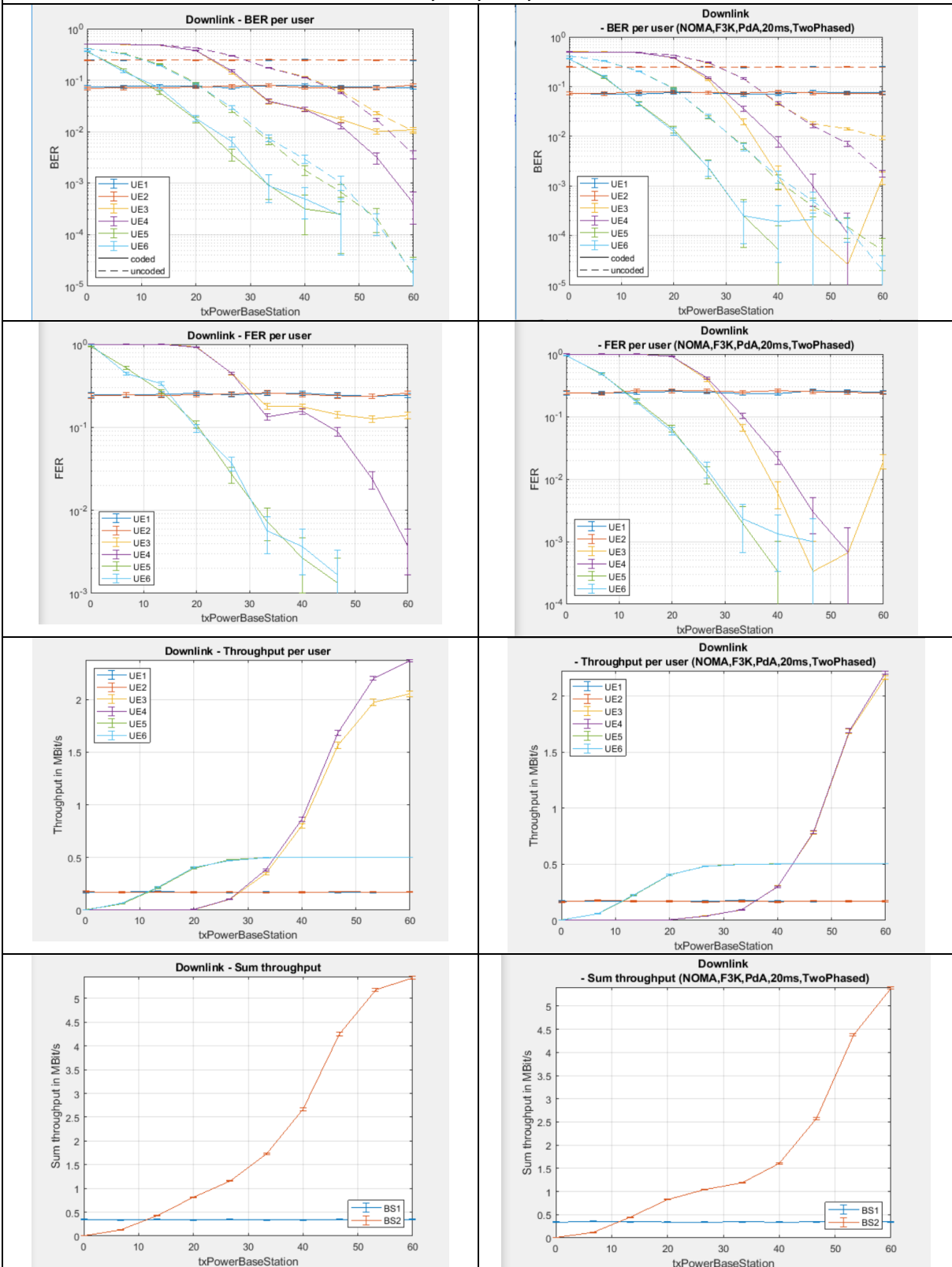
NOMA, F3K, PdA, 10ms



SinglePhased (aka "user-level waterfilling")

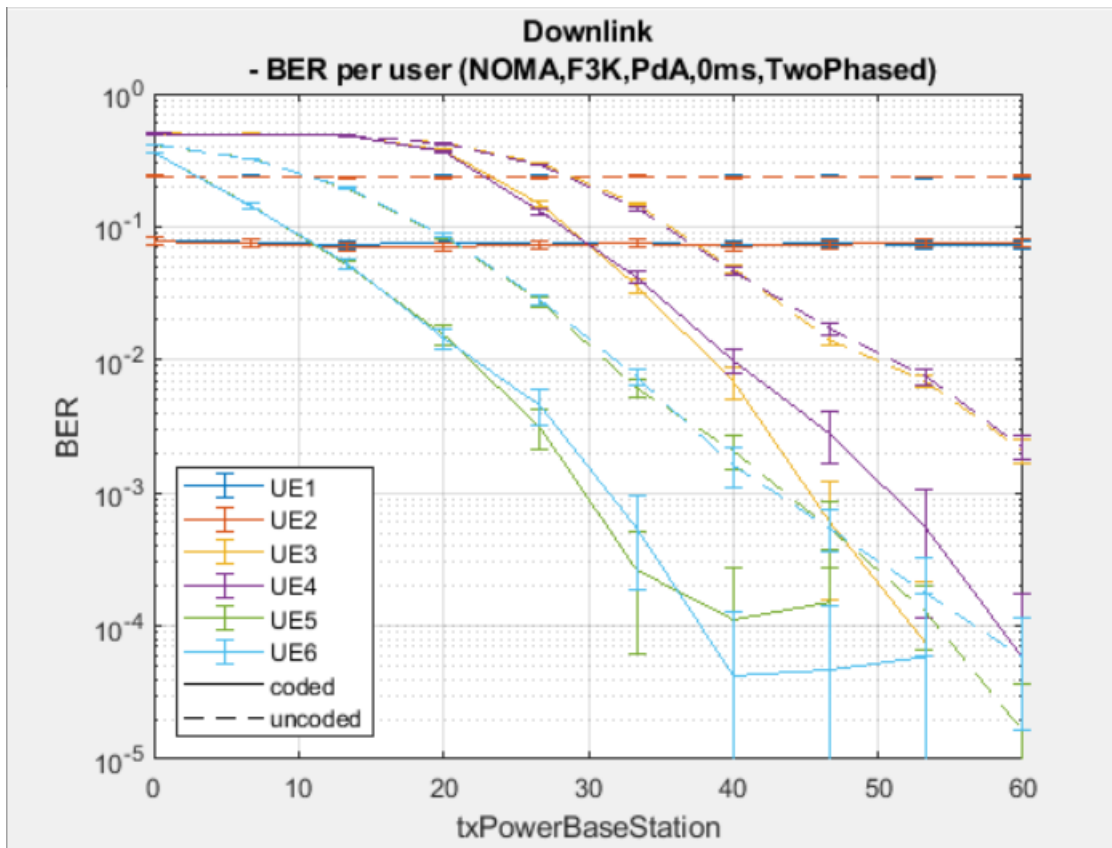
TwoPhased (aka „system level waterfilling“)

NOMA, F3K, PdA, 20ms

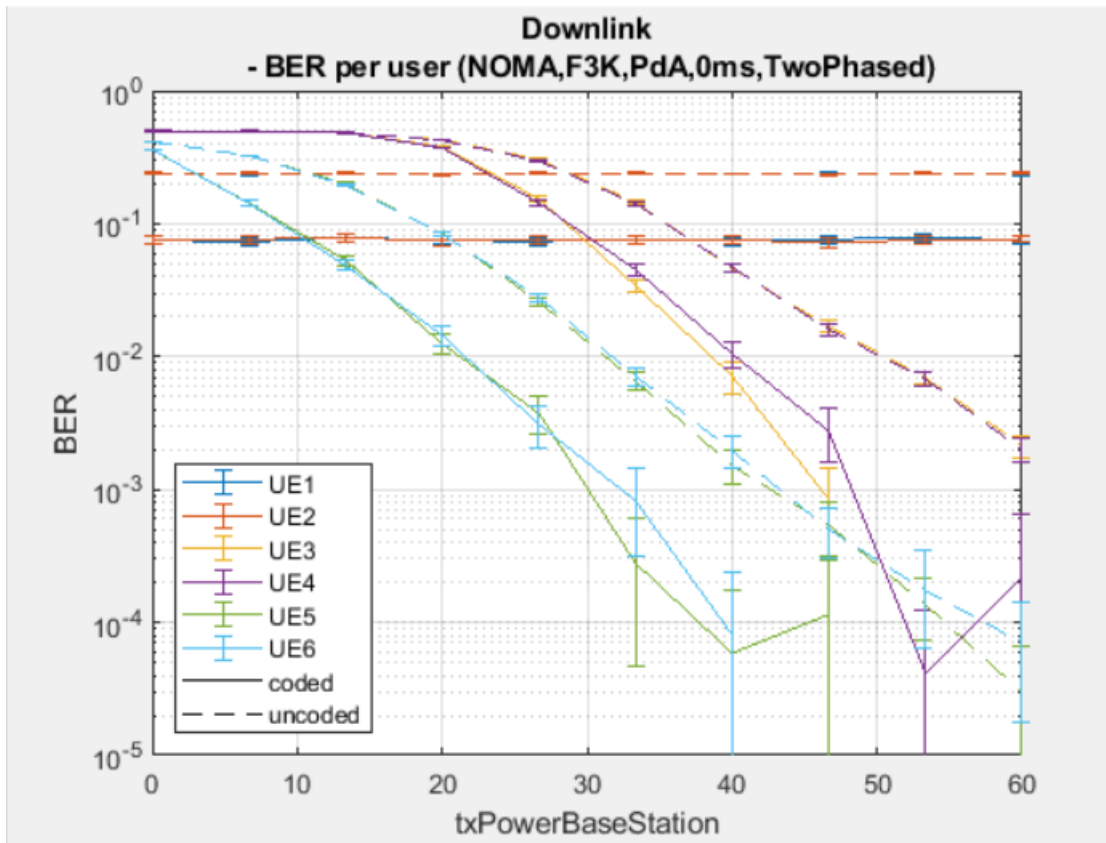


Test powtarzalności dla jednego symulatora:

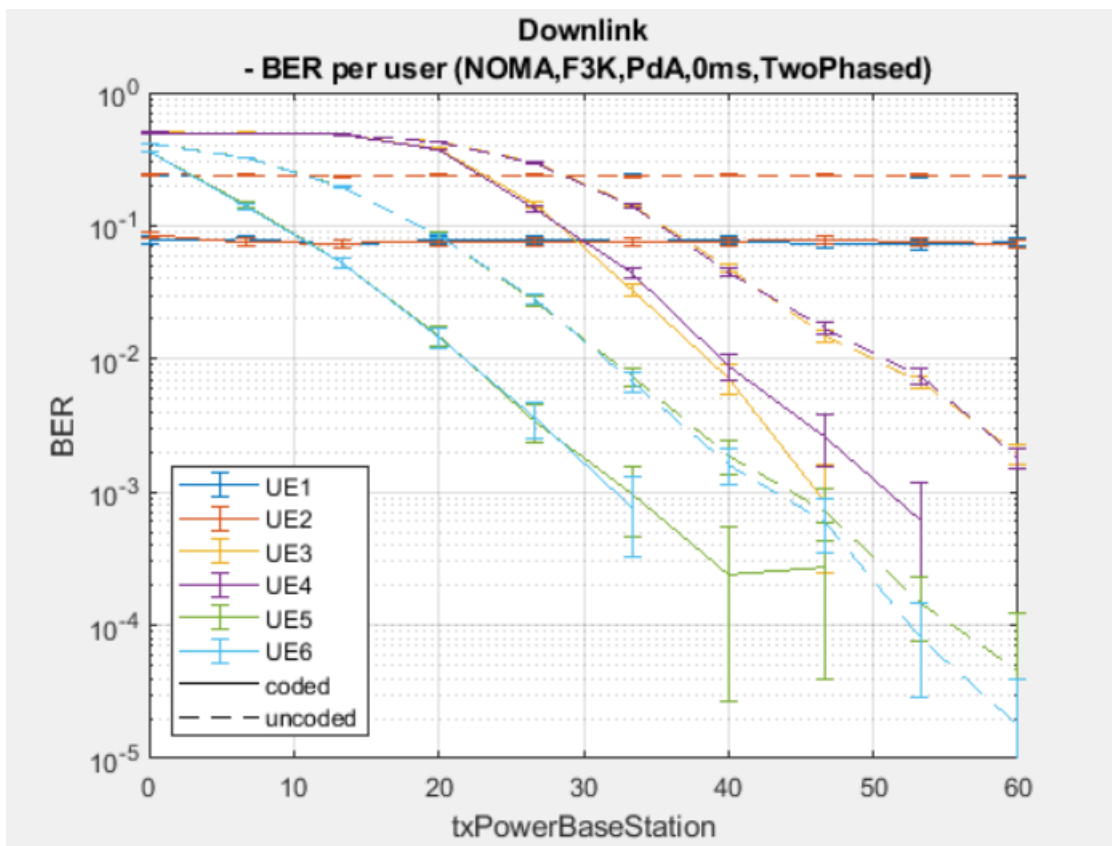
[Uruchomienie PIERWSZE]



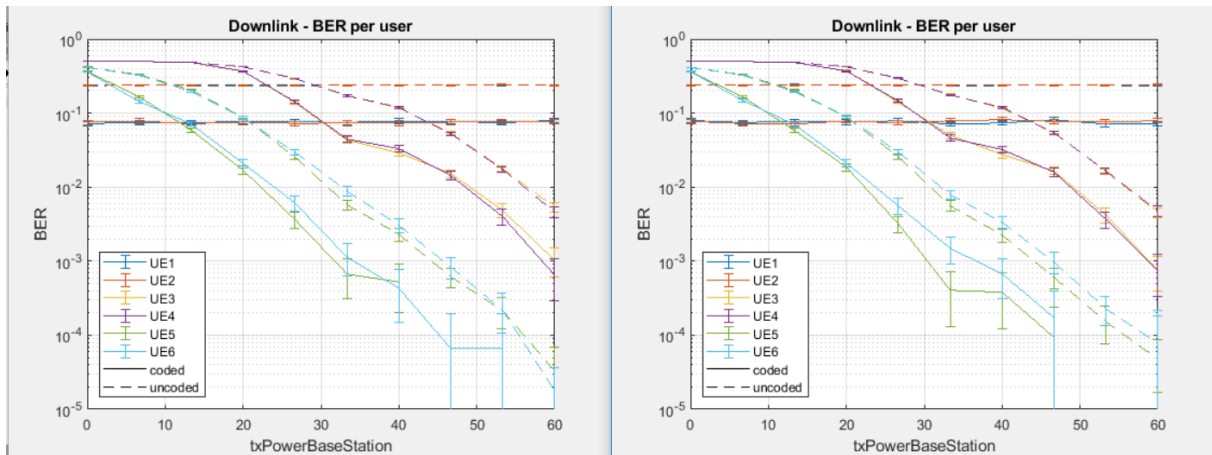
[Uruchomienie DRUGIE]



[Uruchomienie TRZECIE]



Test powtarzalności (single phased)



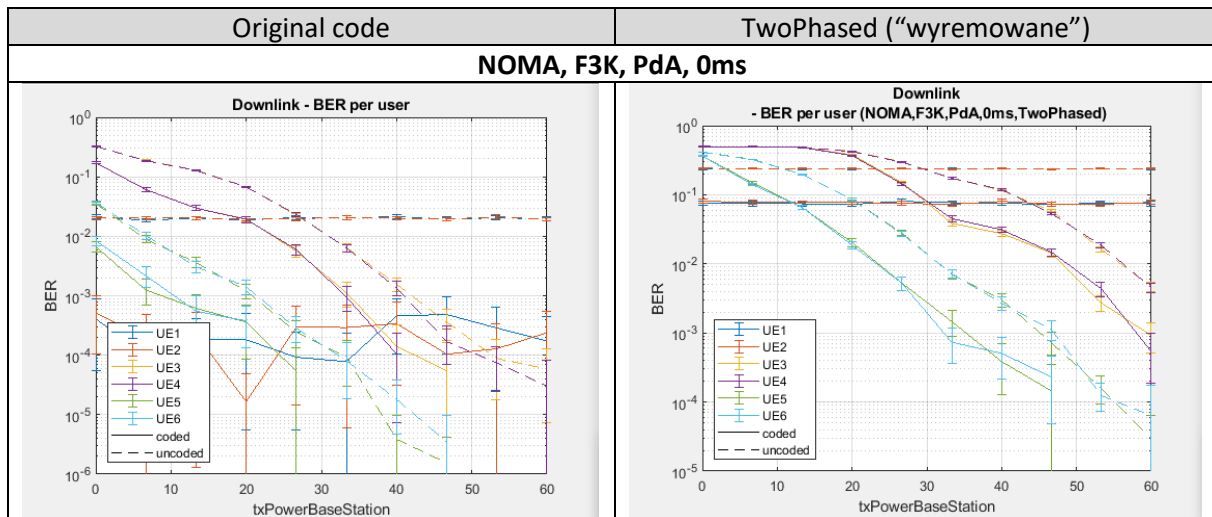
[04.10.2020]

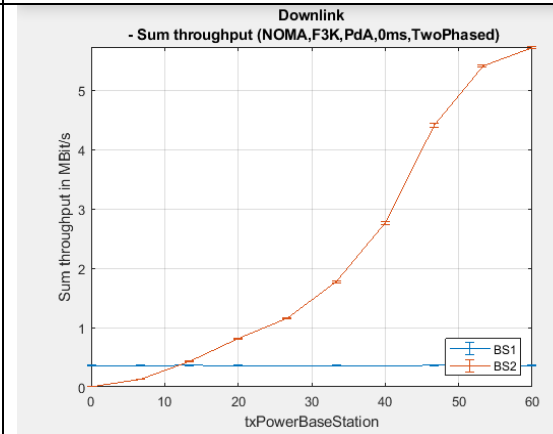
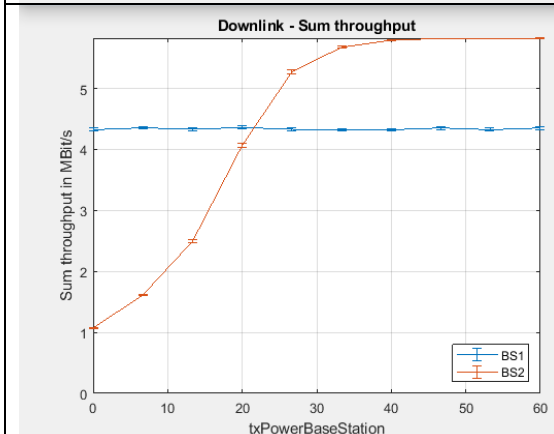
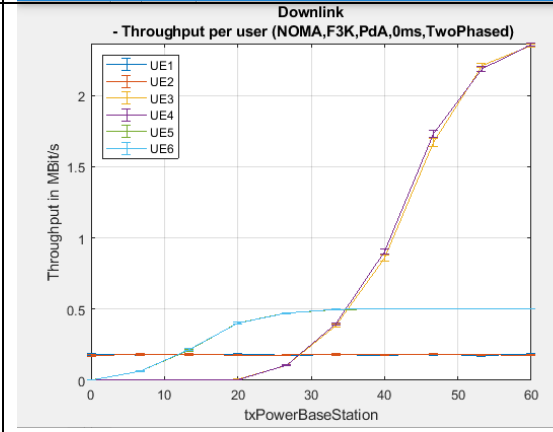
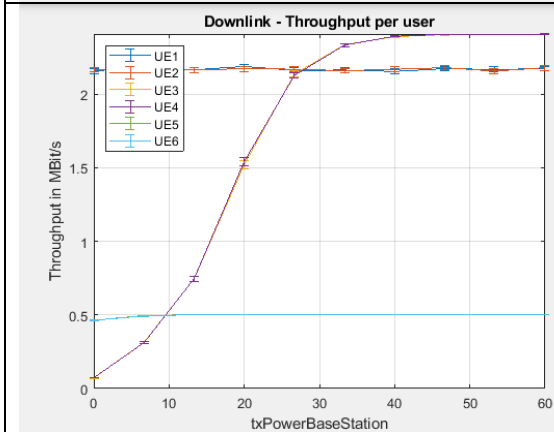
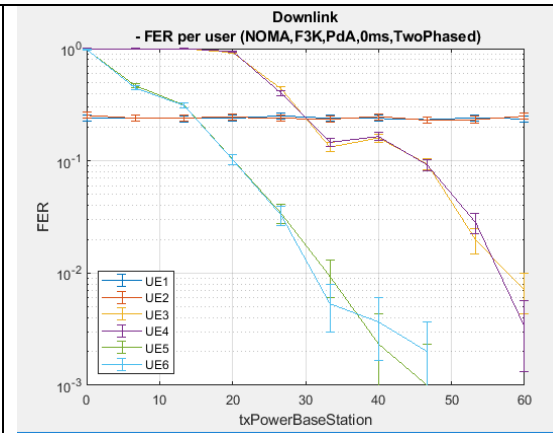
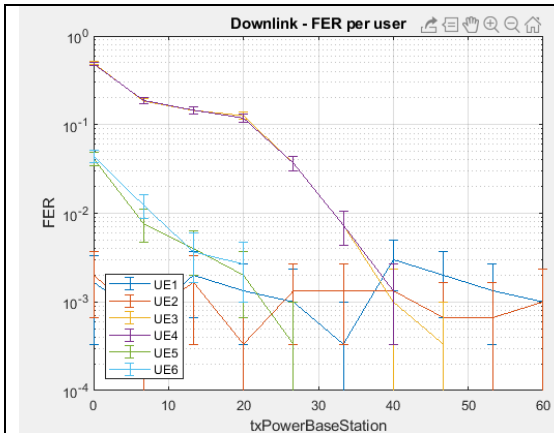
Kolejne porównanie

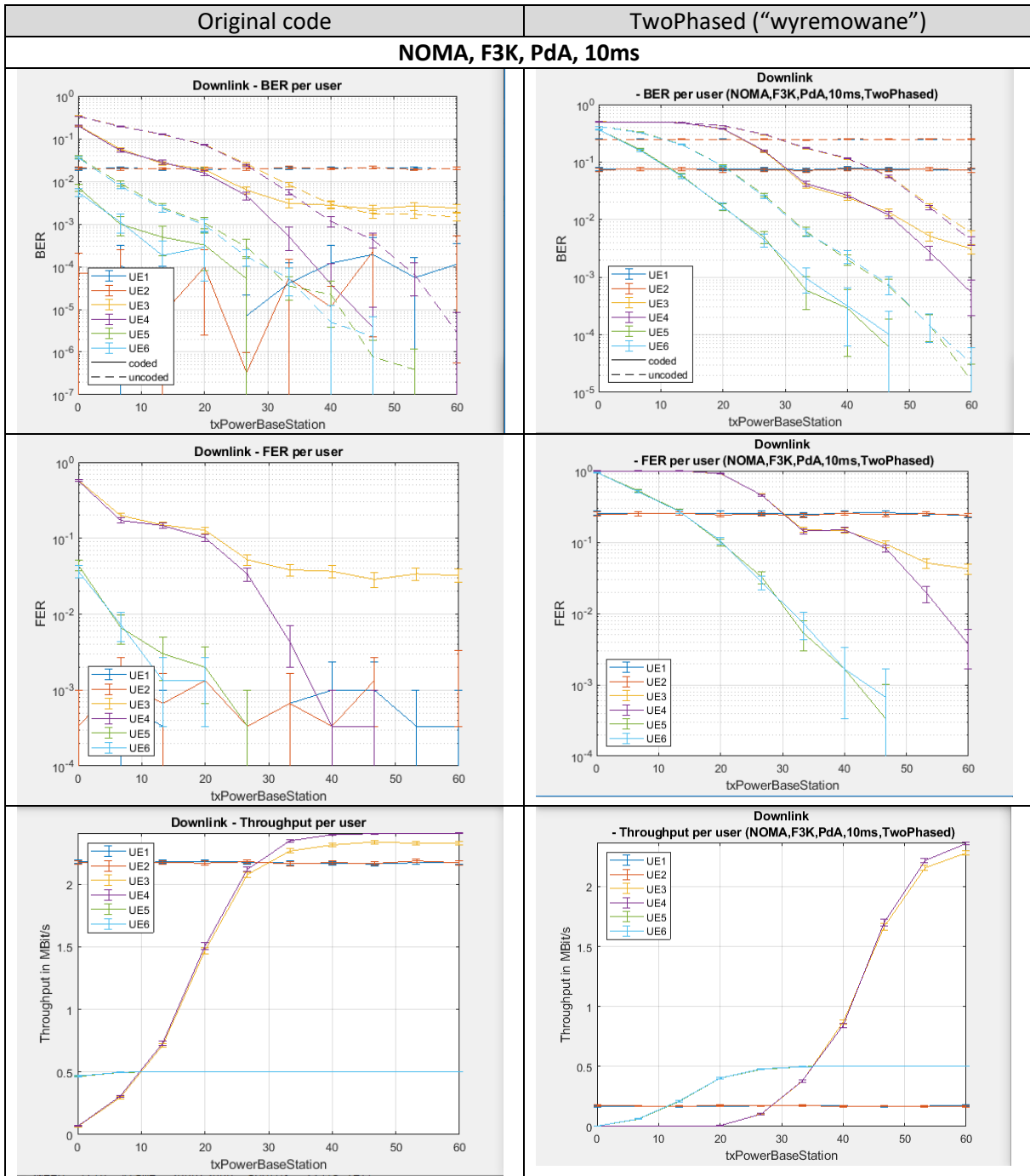
1. Komputer1 (RATfor5G)
 - a. "Two-phased" – wyłączyłem działanie „dwufazowe” i z tego wyszło „SinglePhased”
2. Komputer2 (RATfor5G-second)
 - a. Original code 5G LL

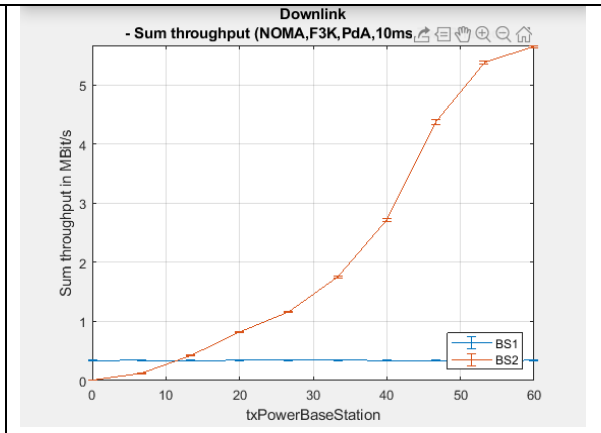
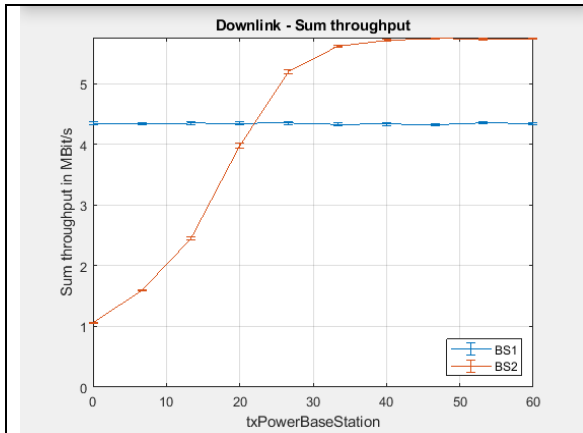
Oba testy dla tych samych scenarios co powyżej.

Wszystkie poniższe testy są **BEZ WATERFILLING**



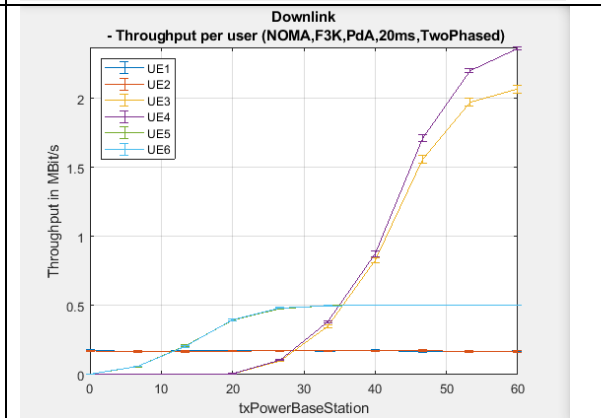
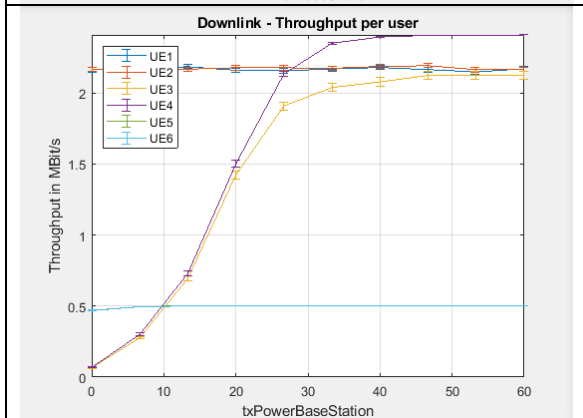
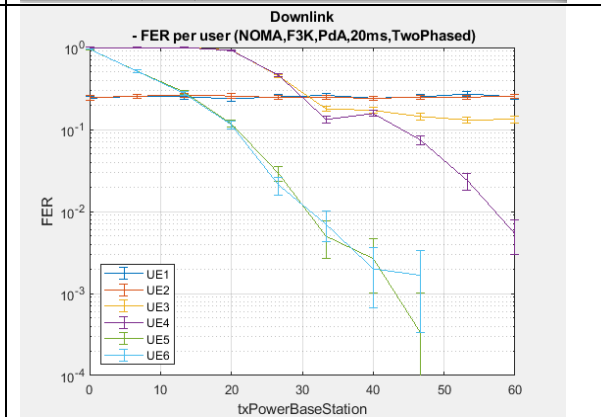
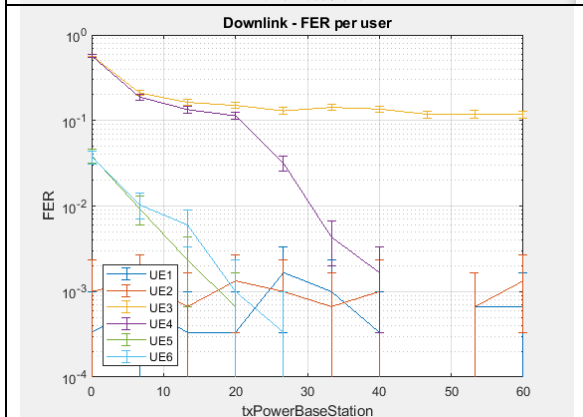
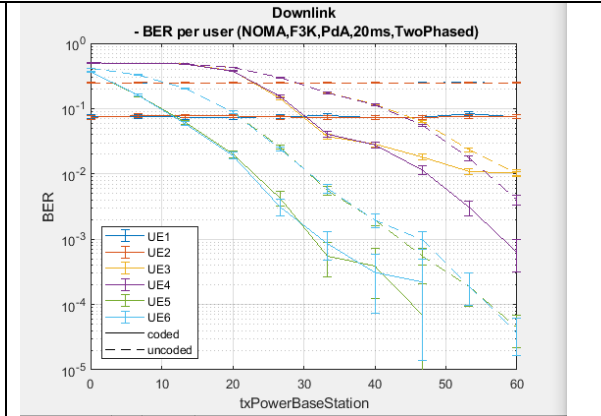
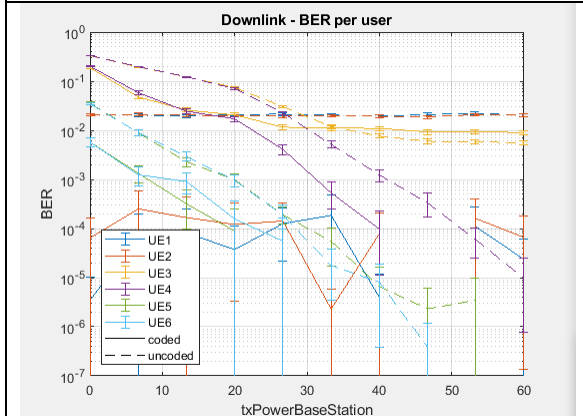


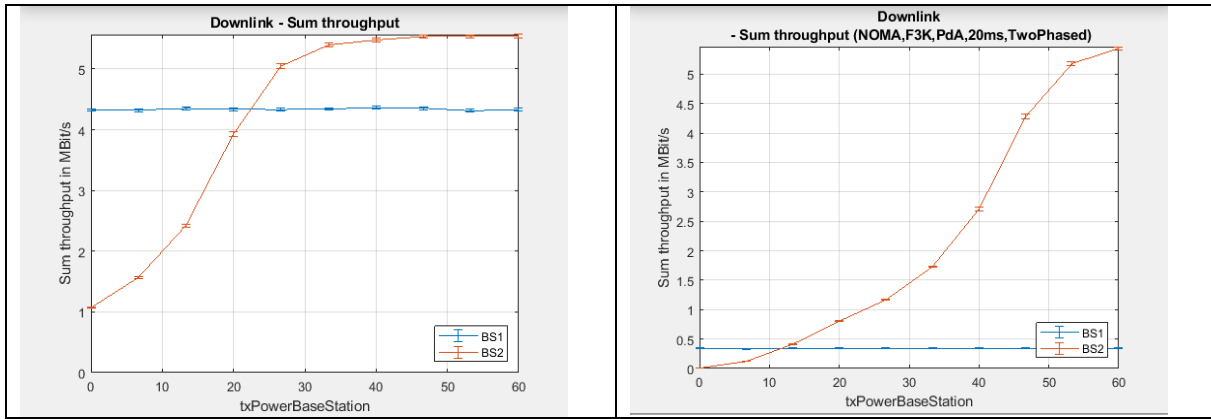




Original code **TwoPhased ("wyremowane")**

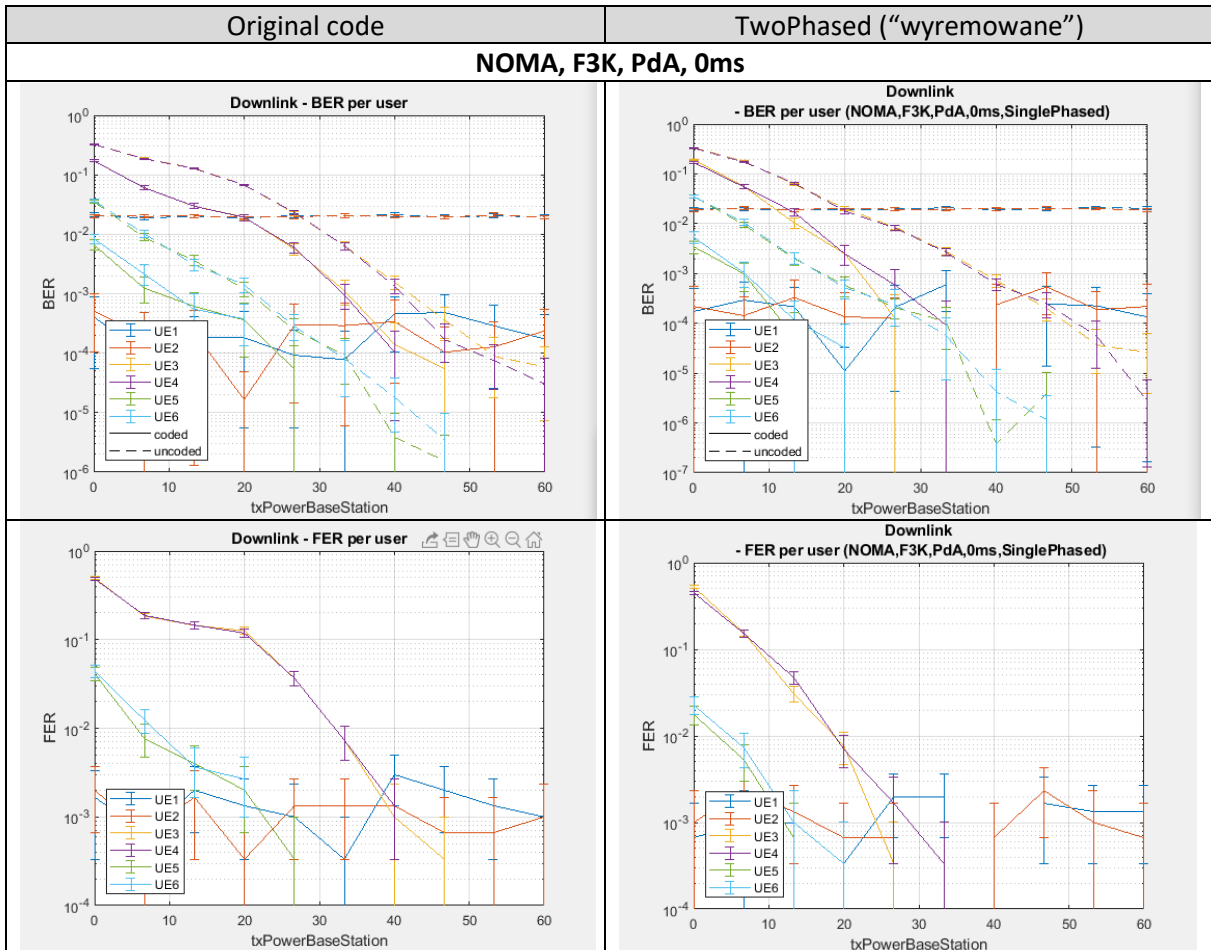
NOMA, F3K, PdA, 20ms

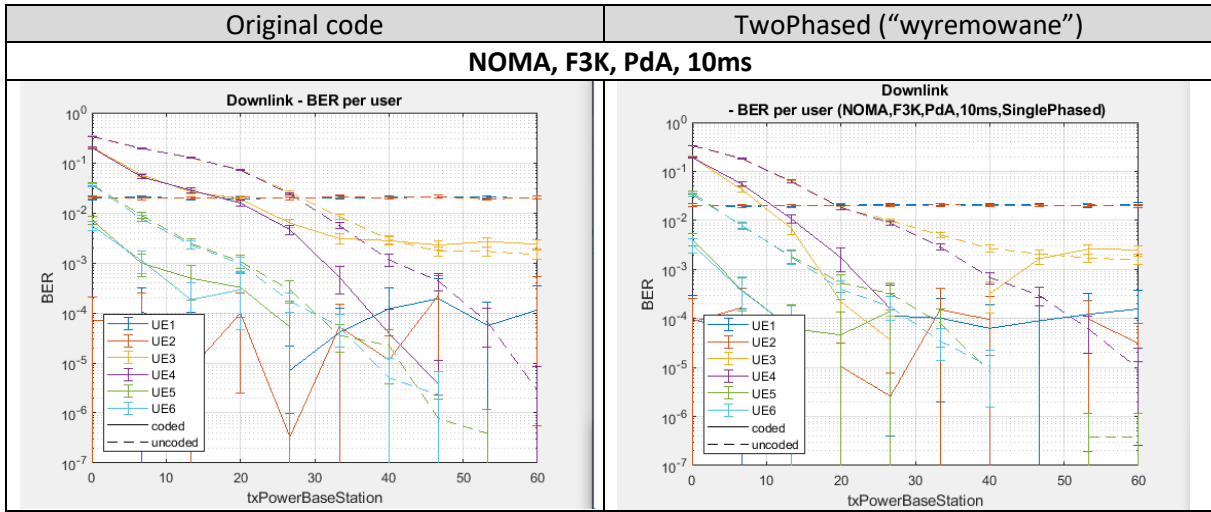
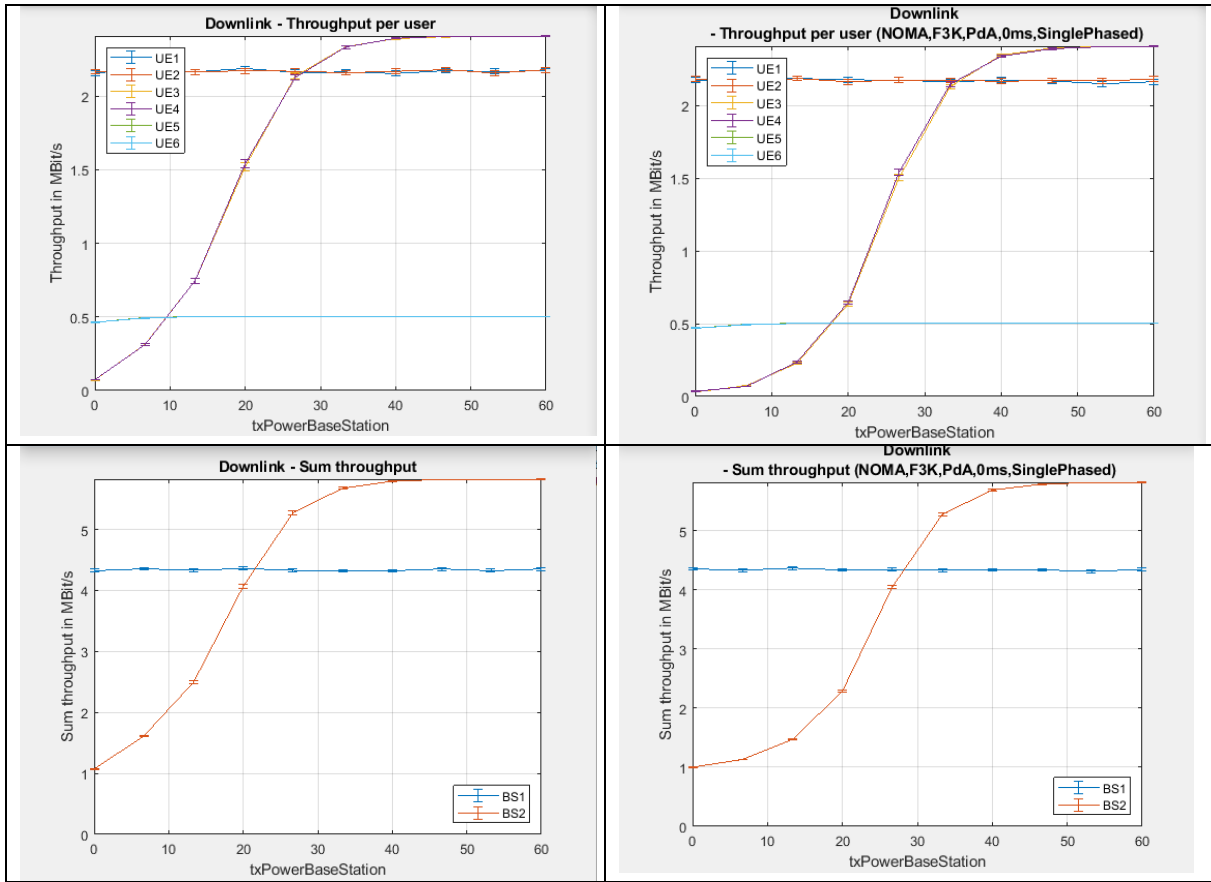


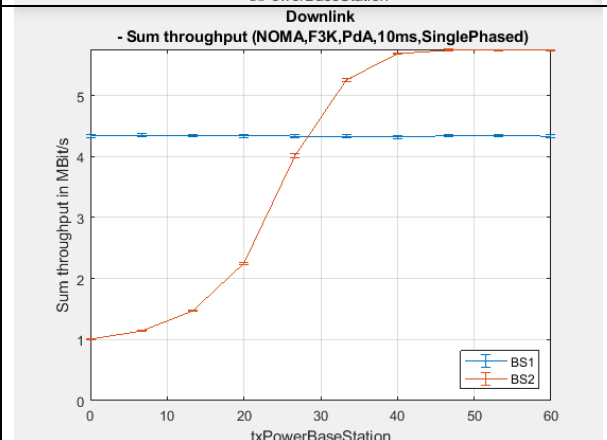
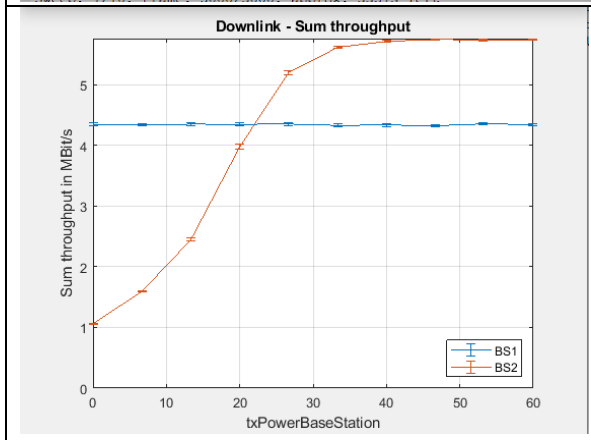
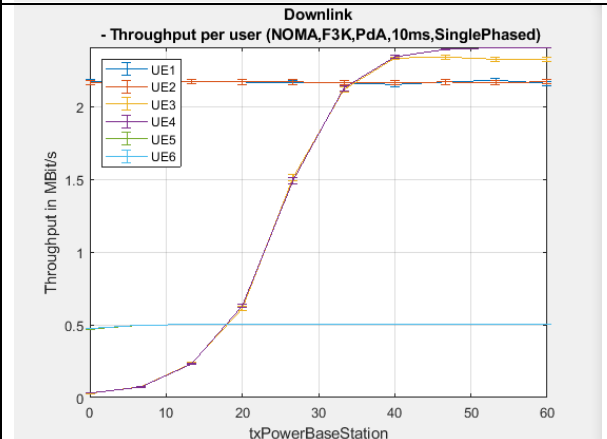
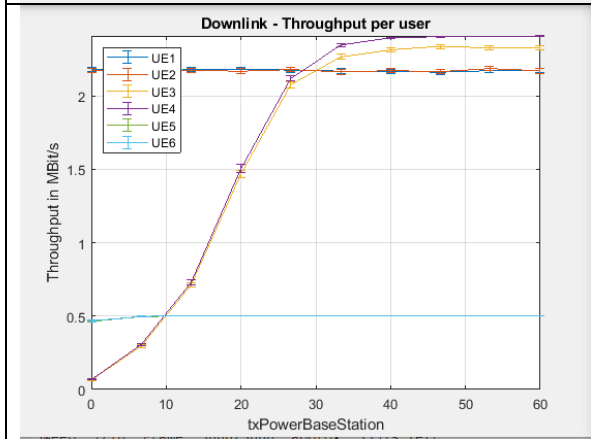
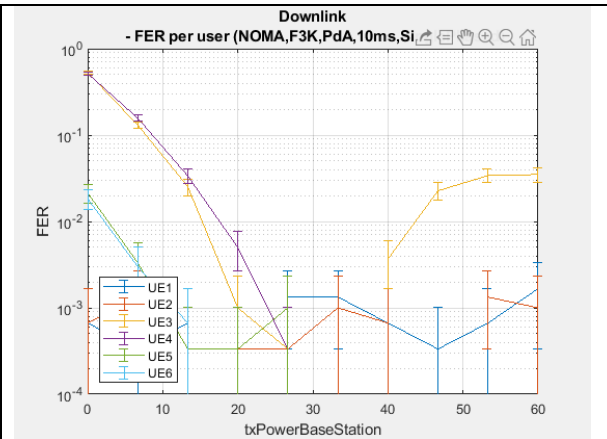
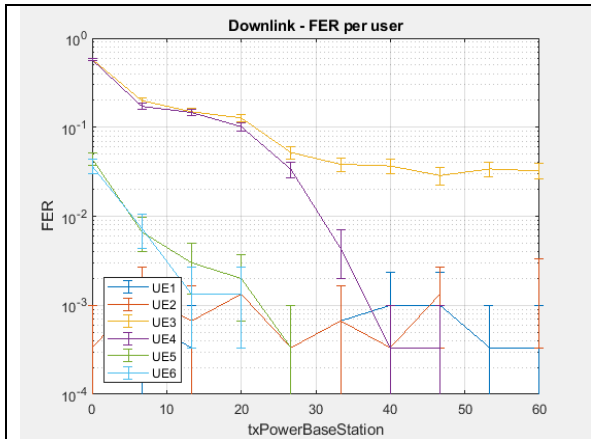


[05.10.2020]

Elementy dodane w projekcie RATfor5G+ zostały dodane (tzw. podejście „two phased”) do czystego kodu. Wygląda że po tej operacji (ale jeszcze nie sprawdzałem czy waterfilling nadal działa) jest dużo lepiej.

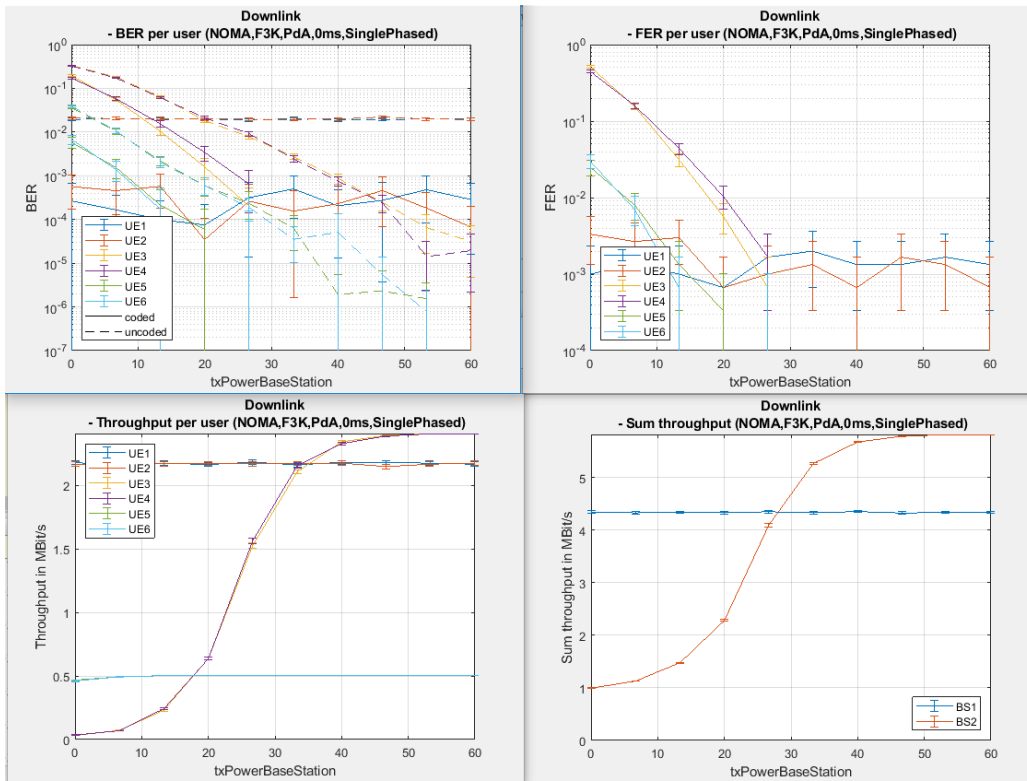




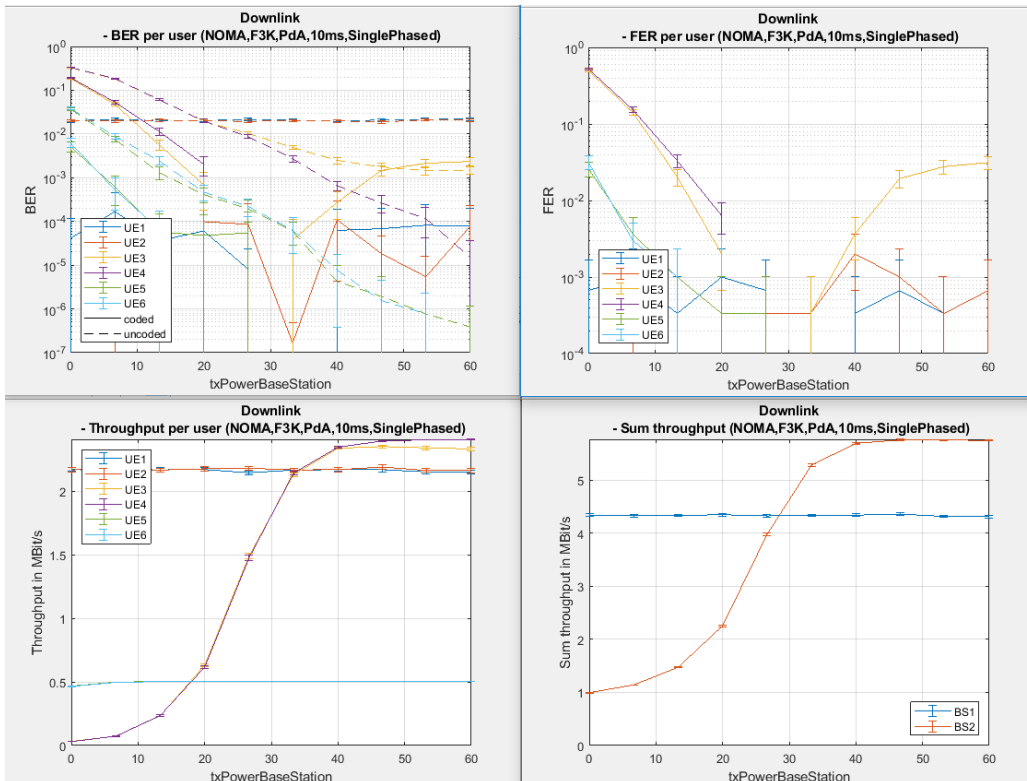


Na koniec testy „z waterfilling” włączonym (per system).

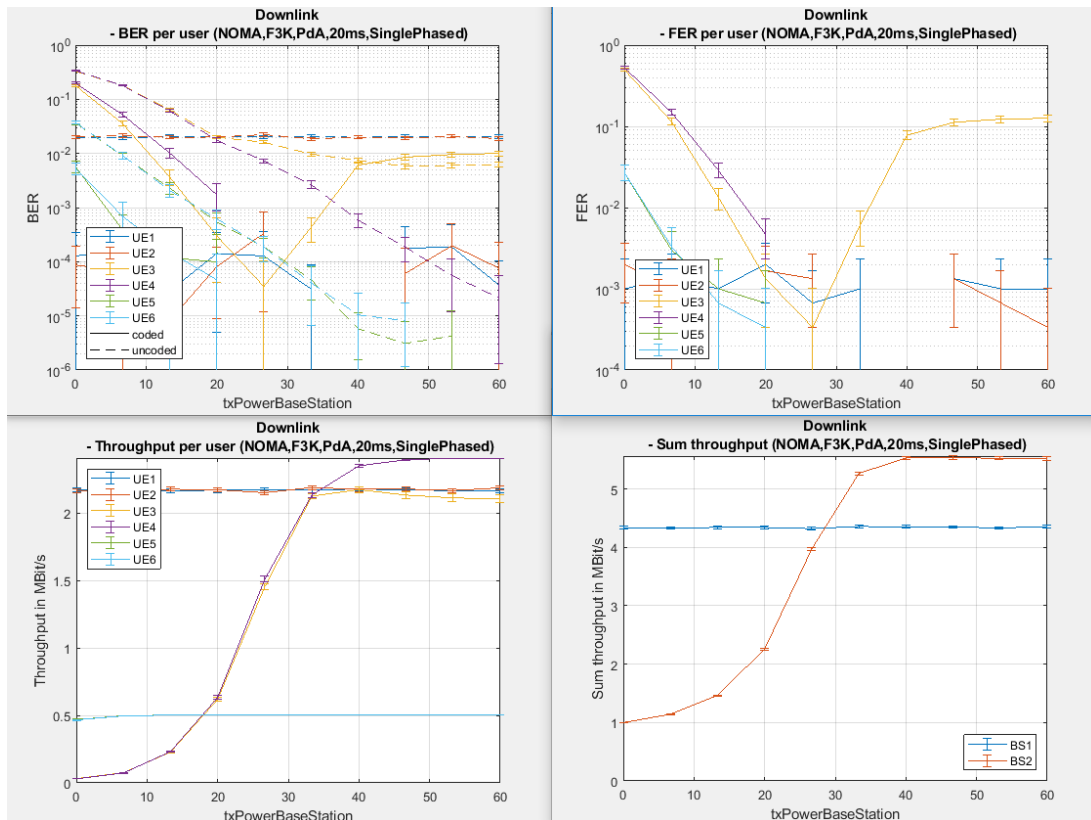
UWAGA ---- to są testy „Two phases” ale nazwy plików ze scenariuszami były złe, stąd na wykresach pojawiło się „single phased”!!!!



UWAGA ---- to są testy „Two phases” ale nazwy plików ze scenariuszami były złe, stąd na wykresach pojawiło się „single phased”!!!!



UWAGA ---- to są testy „Two phases” ale nazwy plików ze scenariuszami były złe, stąd na wykresach pojawiło się „single phased”!!!!

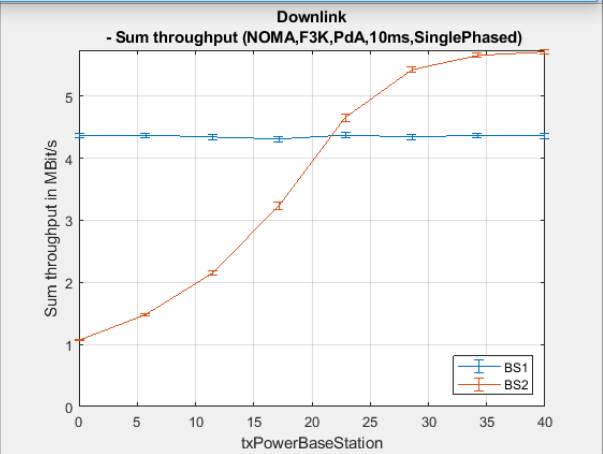
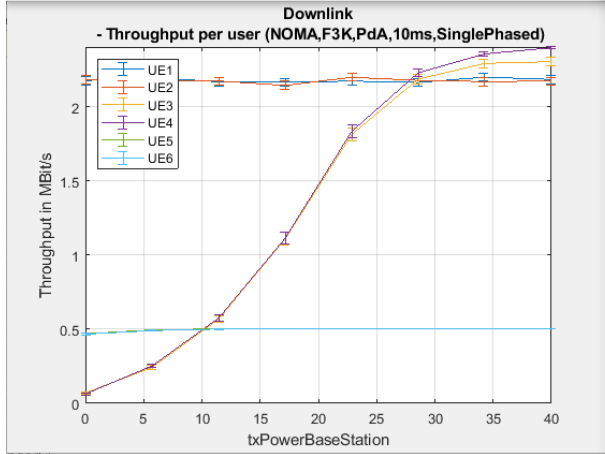
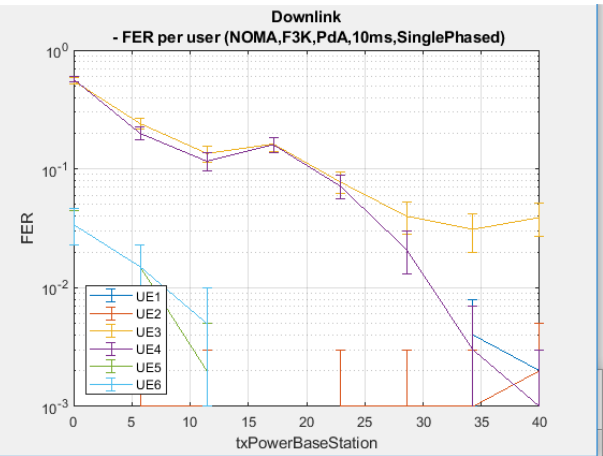
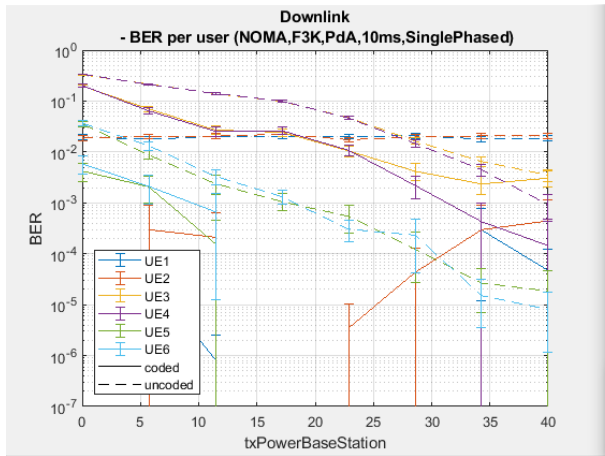
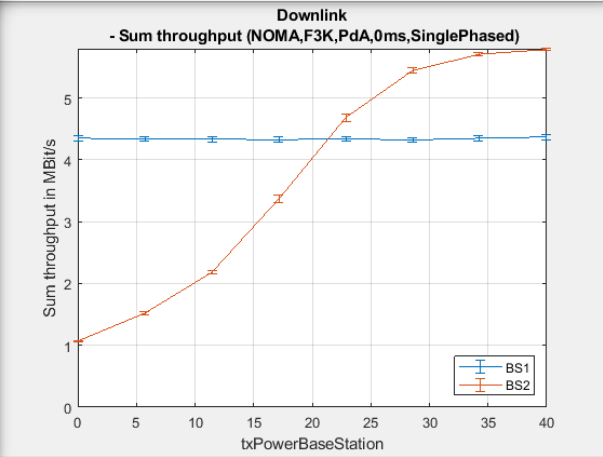
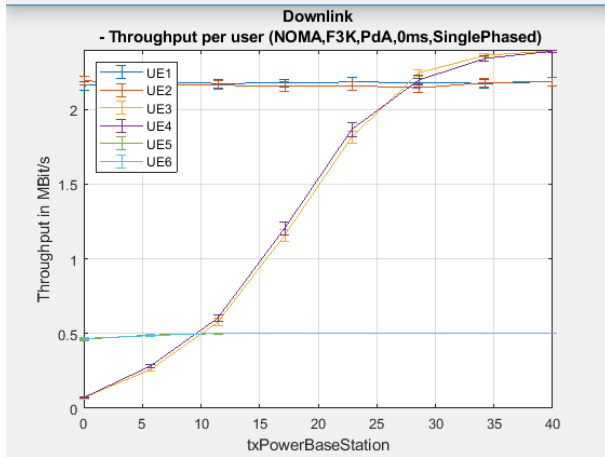
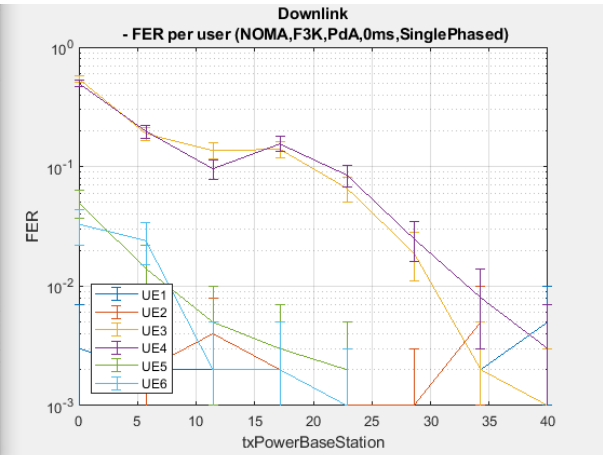
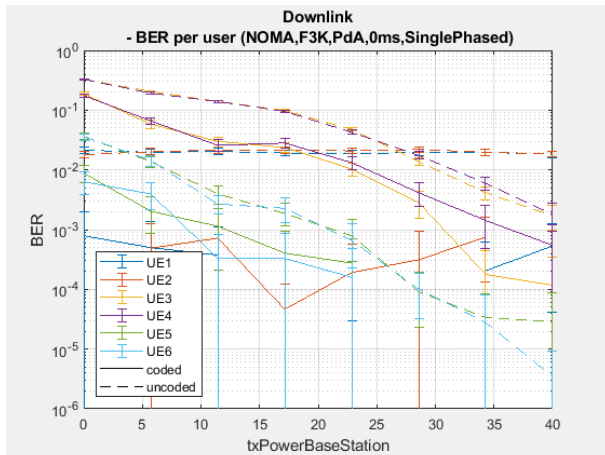


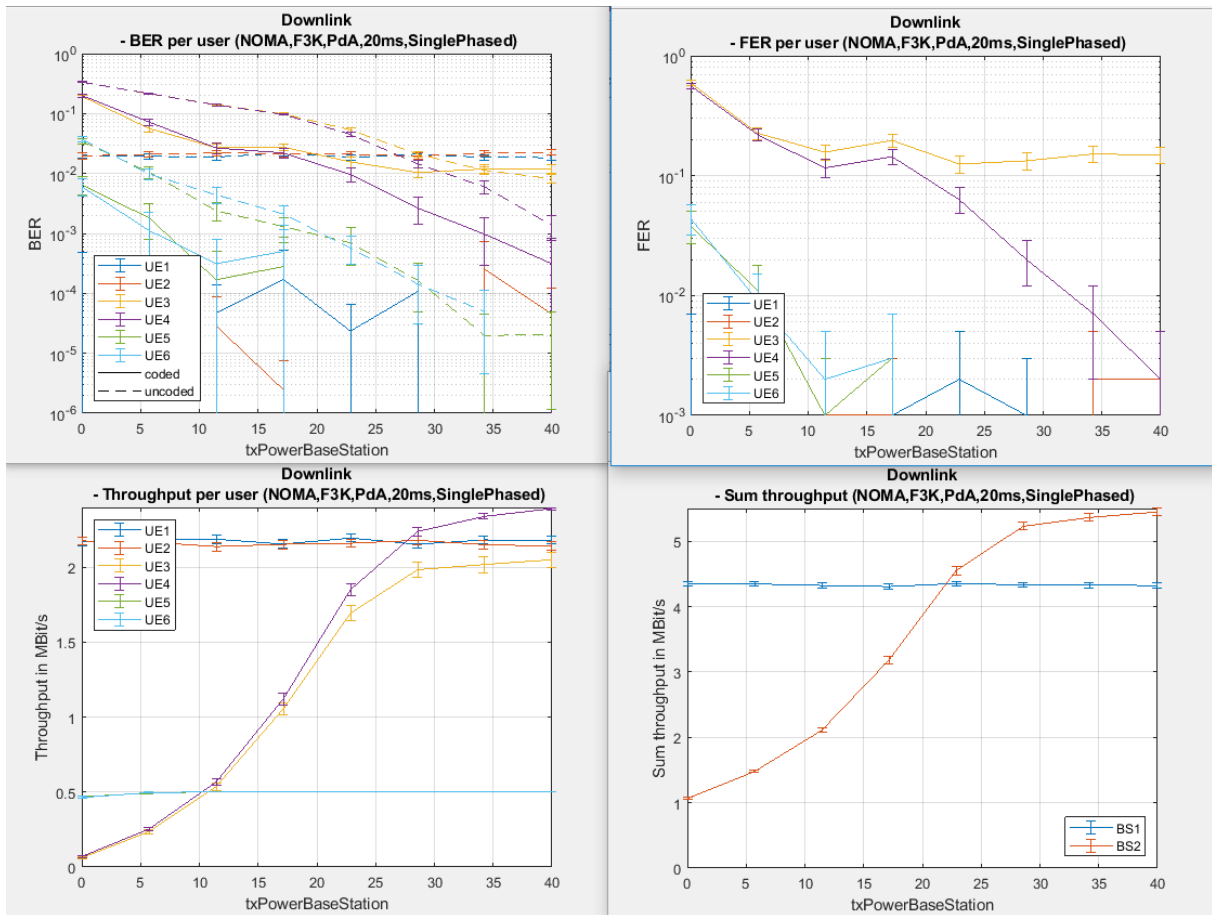
Kolejne podejście do kalibracji (06.10)

1. KROK1:
 - a. Wyłączam obj.channel.k = 0 dla usera „UE3” → to nie był dobry pomysł
 - b. Te same scenariusze (1000 ramek, moc od 0-40dbm)
2. KROK2: (porównanie „5GLLSimulator” vs „5GLLSimulatorBIS”)
 - a. Wyłączam linie zbędne w uruchomieniu „Two phased”
 - i. Main – wersja podstawowa + dodane raportowanie MCS
 - ii. Link – wyłączona faza 2, wyłączone updateFeedback(channelAF)
 - iii. BaseStation – wyłączone „waterfilling” po stronie transmitSignal i RX
 - b. `Poniżej dane do analizy MCS



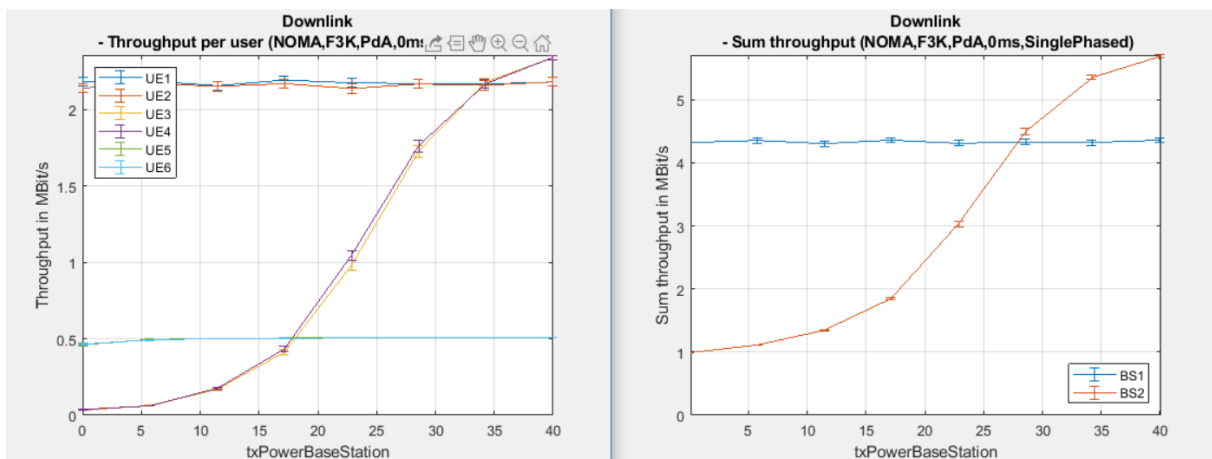
Original with MCS reporting.zip



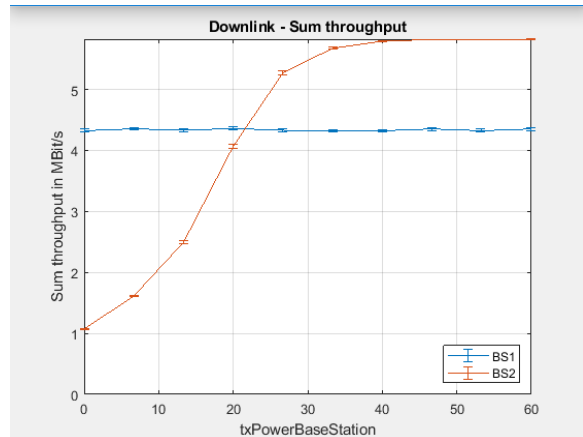


[13.10.2020]

Poniżej wersja dla „two-phased” z powyższymi wszelkimi „mnożeniami, dzieleniami i podstawieniami” wynikającymi z kodu „waterfilling”.



Trzeba to porównywać z wersją oryginalną dla takiego samego ustawienia/scenariusza:



KALIBRACJA symulatora z dodatkami RATfor5G+ [14.10.2020]

- Udało się zidentyfikować wymagane poprawki:
 - W link.m – było podwójne mnożenie zmiennej „channel” razy pewna wartość
 - Niestety na skutek odpalania dwóch faz, to mnożenie w drugiej fazie realizowane było błędnie po raz drugi w efekcie kanał był „skalowany” * 1/x2
 - Bez dodania dodatkowego „if” w linii 182 kod niestety dwukrotnie wykonywał mnożenie z linii 184 oraz 186. Zatem po przejściu przez kolejną fazę kanał był „dwukrotnie zeskalowany”.
 - uruchomił testy dla naszych scenariuszy (Frames=1000). Wykresy poniżej są zrobione już dla praktycznie całego „odkomentowanego” (bez komentarzy) kodu „two phased”. Wszystkie mnożenia * sqrt(waterfilling) i dzielenia / sqrt(waterfilling) zostały przywrócone.
 - Nazwy w tytułach wykresów „singlePhased” są błędne, bo wykresy są zrobione w oparciu o podejście dwu-fazowe (twophased), należy mieć to na uwadze.

```

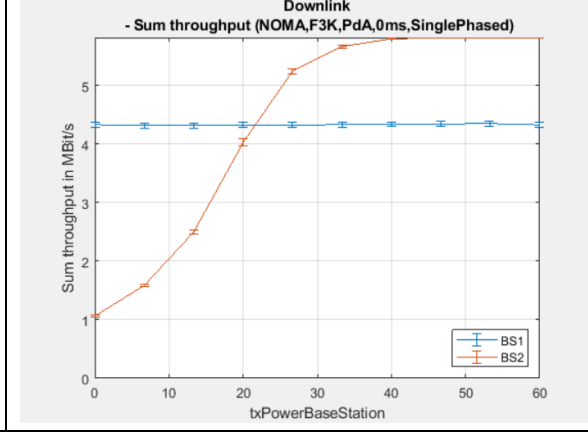
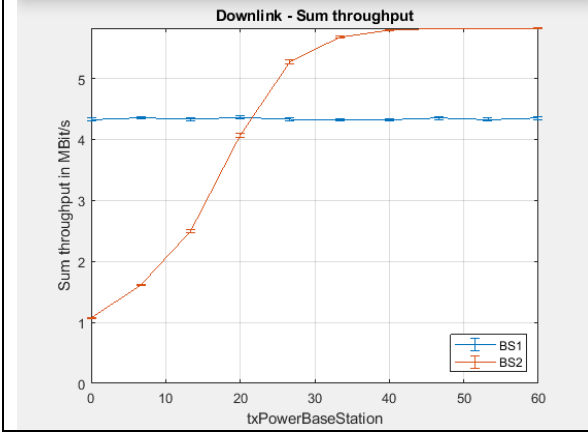
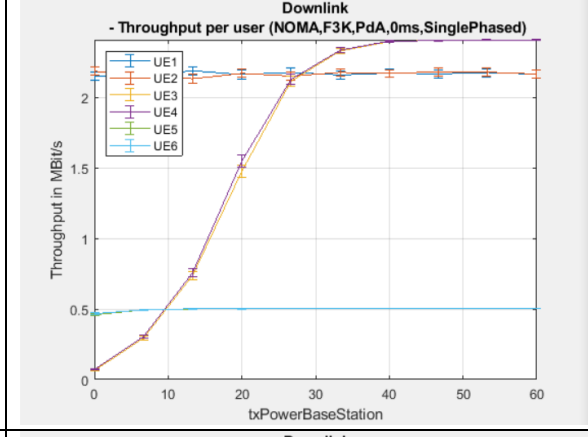
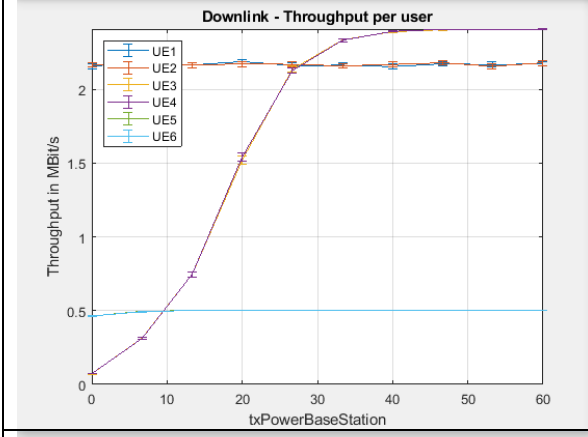
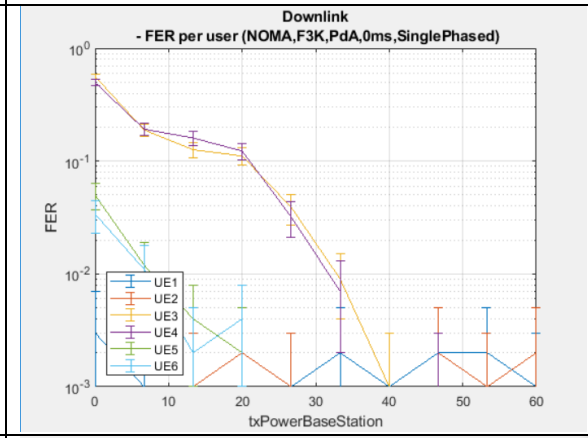
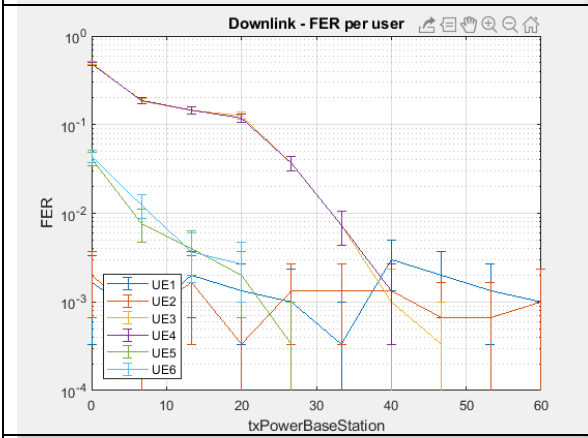
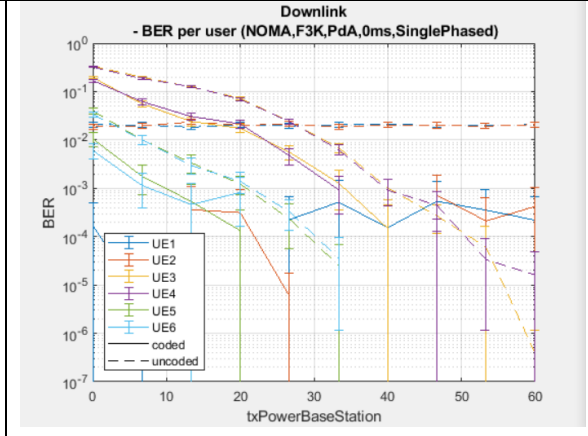
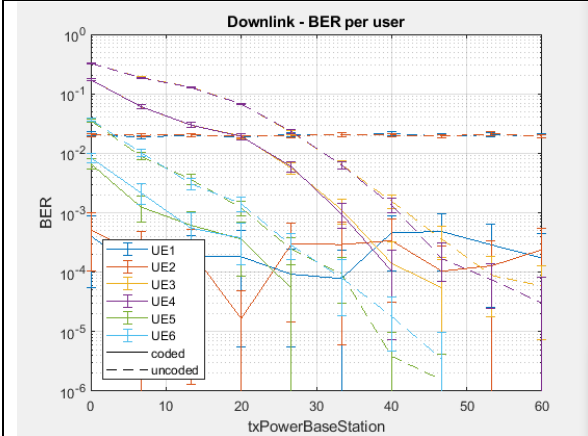
164 - switch obj.transmissionMode
165 -     case 'CLSM'
166 -         % if the feedback delay is 0, the channel for the next transmission is used for the feedback calculation otherwise
167 -         % the current channel is used
168 -         if obj.Feedback.delay == 0
169 -             if (obj.updatePhase == 2)
170 -                 % if obj.Transmitter == 1
171 -                 % return; %% no need to do anything else as the updateLink for this link is already DONE in Phase==1
172 -                 % else
173 -                 channel = obj.kkk; %% RATfor5G for the "transmitter2" second Phase of "updateLink" only
174 -                 % end
175 -             else
176 -                 channel = 10^((obj.transmitPower-30-obj.pathloss)/20) * obj.powerScale *obj.Channel.GetTransferFunction(obj.Modulator.WaveformObj);
177 -             end
178 -         else
179 -             channel = obj.Modulator.Channel;
180 -         end
181 -         %WK added if
182 -         if obj.updatePhase == 1 % run this code ONLY ONCE, in Phase1, otherwise channel will be multiplied TWICE in Phase2
183 -         if obj.IsMUSTNearUE
184 -             channel = channel * obj.Modulator.SignalConstellation(1).MUSTNearUEScale;
185 -         elseif obj.IsMUSTFarUE
186 -             channel = channel * sqrt(1 - Links(obj.Transmitter, obj.MUSTNearUE).Modulator.SignalConstellation(1).MUSTNearUEScale^2);
187 -         end
188 -     end
189 - end

```

Original code

TwoPhased ("wyremowane")

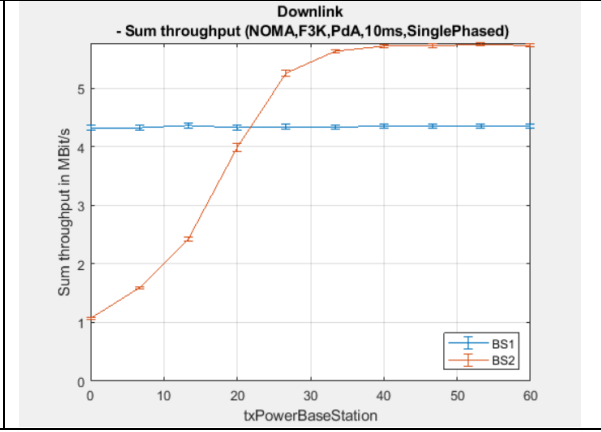
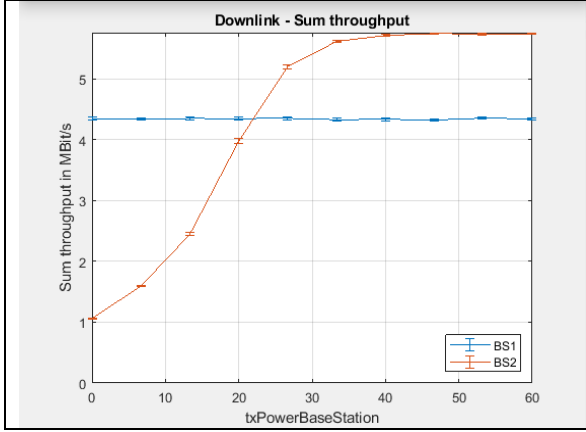
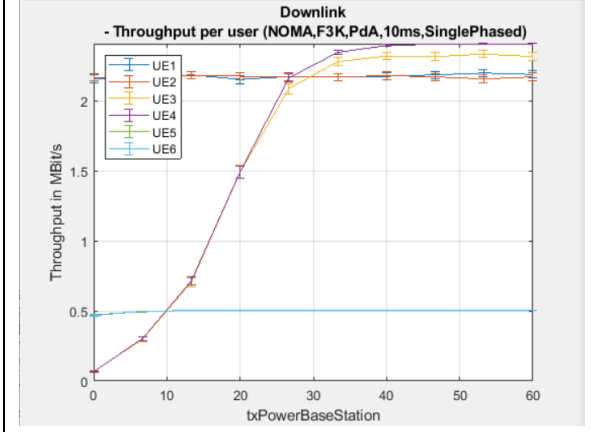
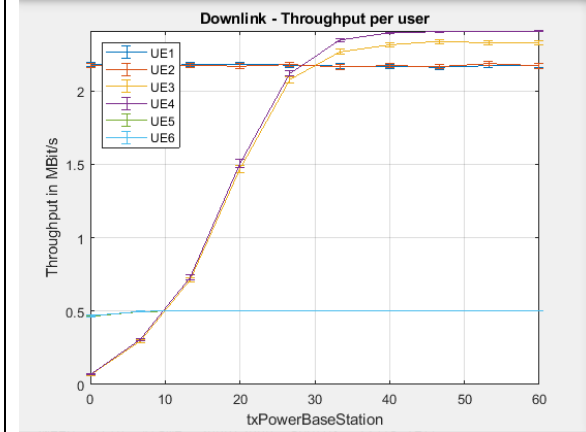
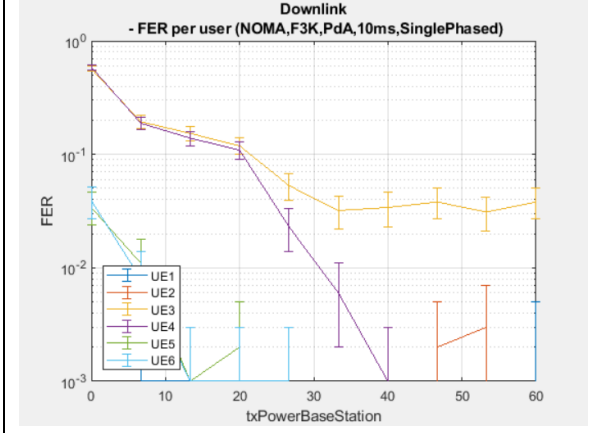
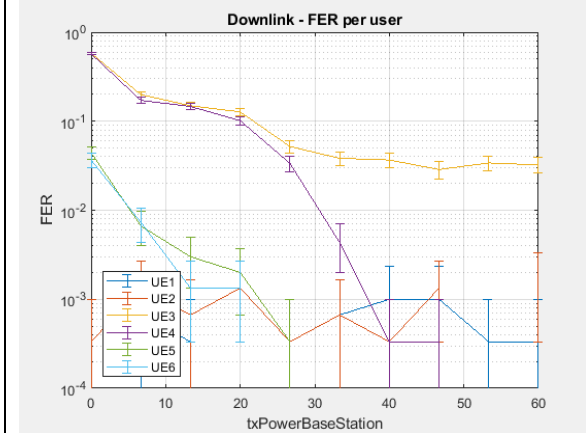
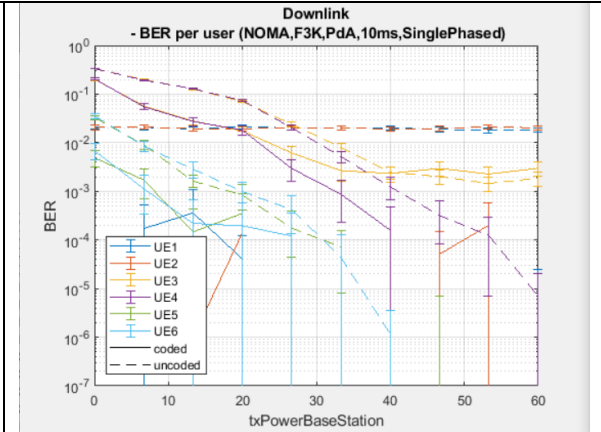
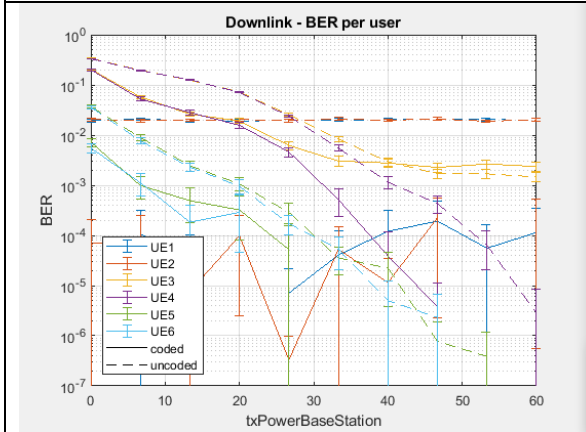
NOMA, F1K, PdA, 0ms



Original code

TwoPhased (naprawione, bez waterfilling)

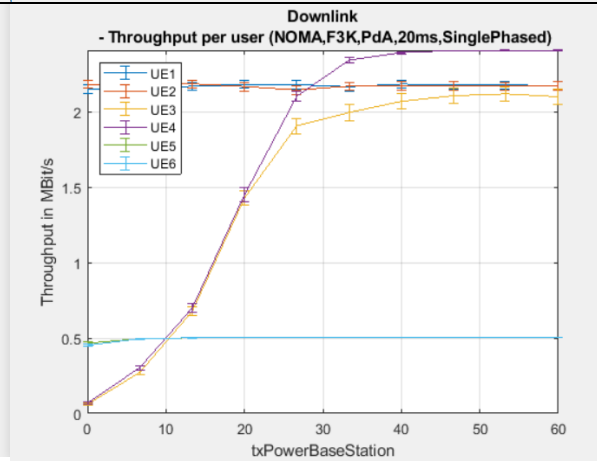
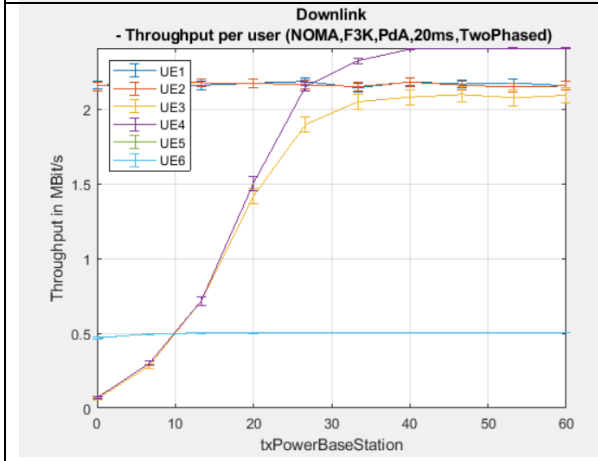
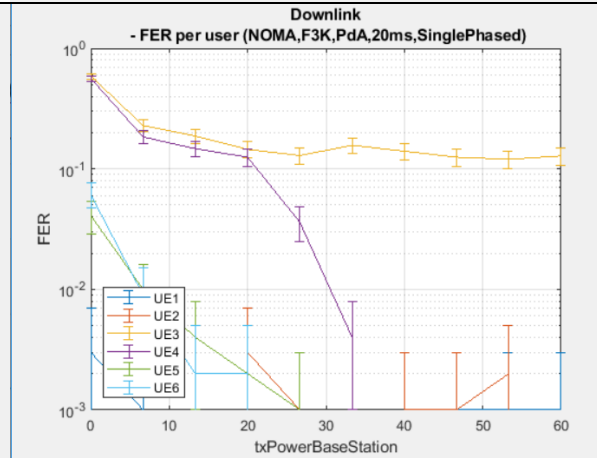
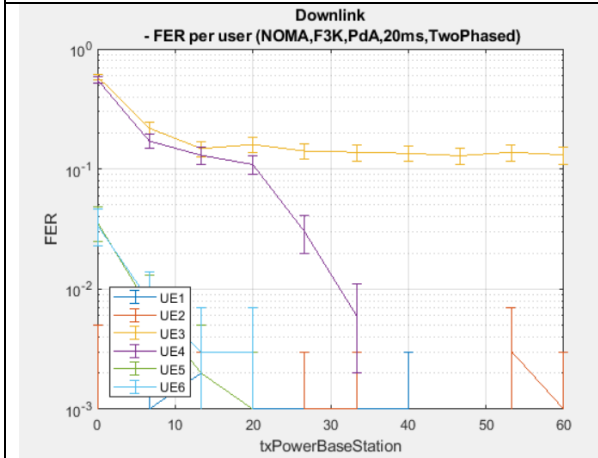
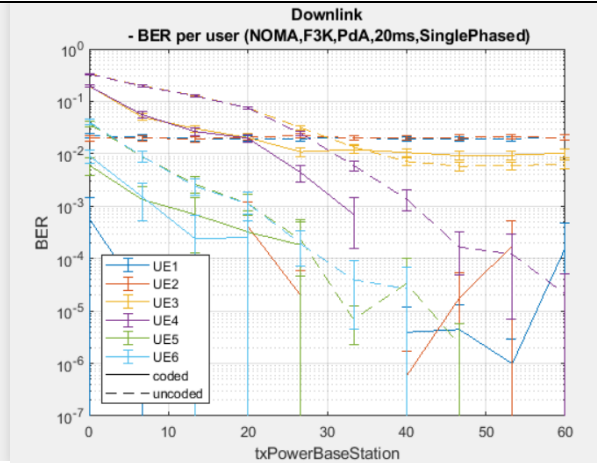
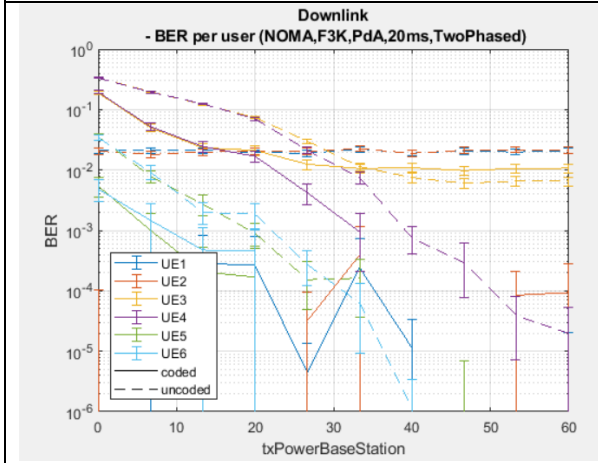
NOMA, F1K, PdA, 10ms

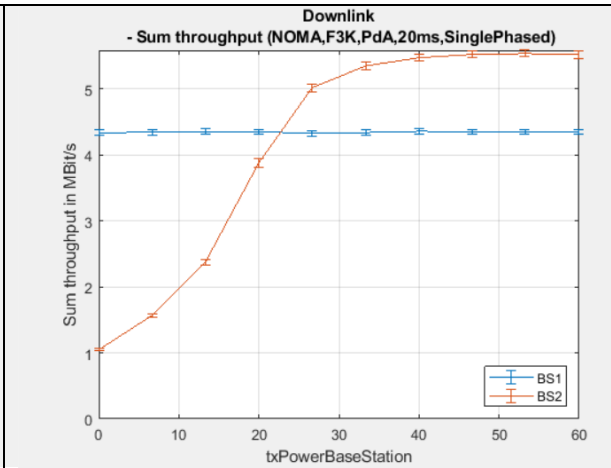
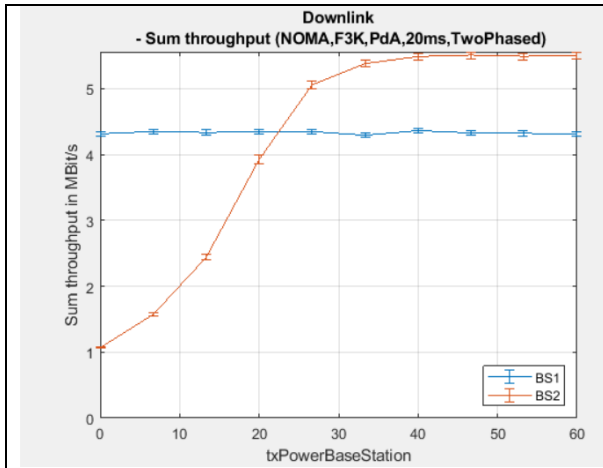


Original code

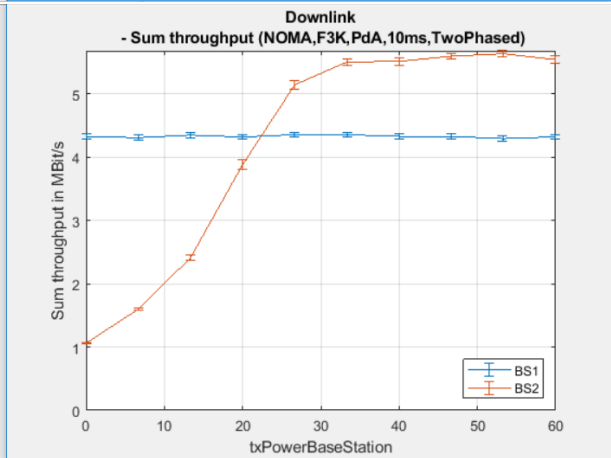
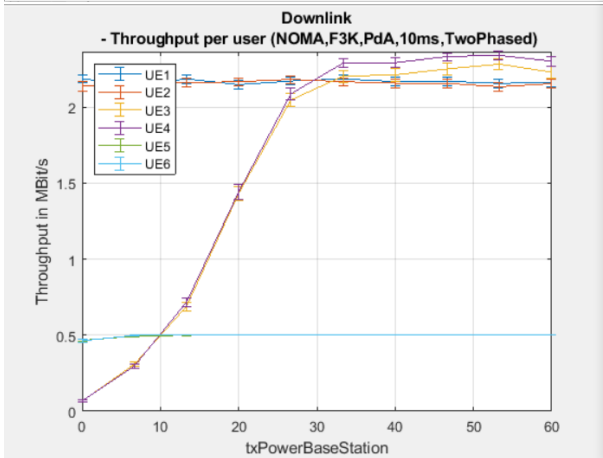
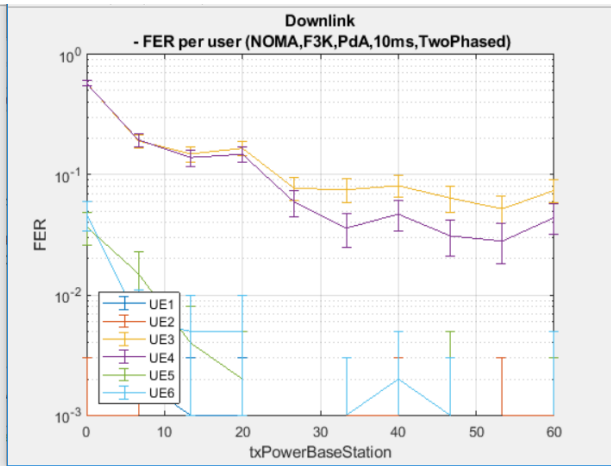
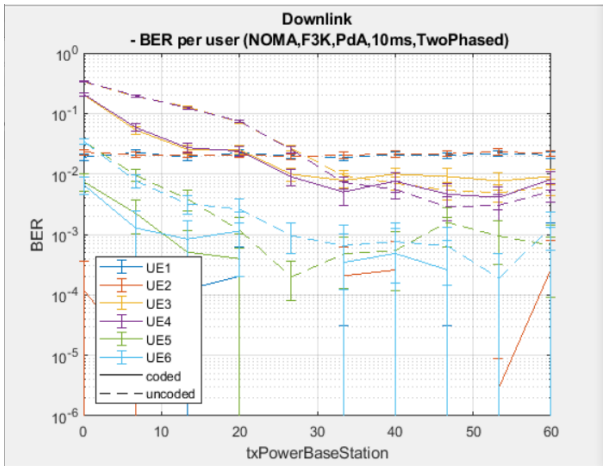
TwoPhased (naprawione, bez waterfilling)

NOMA, F1K, PdA, 20ms





Pomiary z waterfilling włączonym (1000 ramek)



15.10.2020

Testy mechanizmu **waterfilling** dla powyższych 3-ech scenariuszy. (Frames=1000)

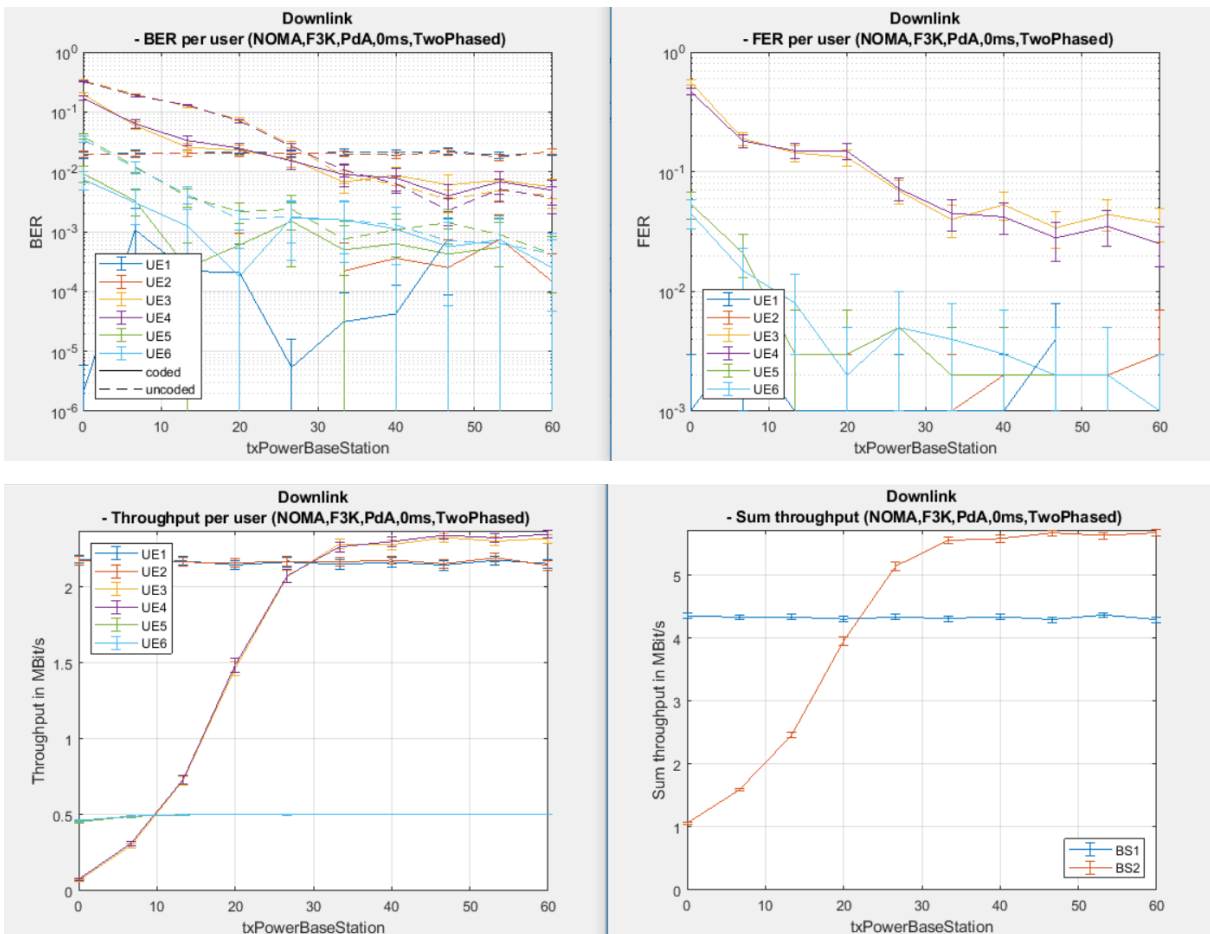
PC: ratfor5g-second

Version: 5GLLSimulator (folder)

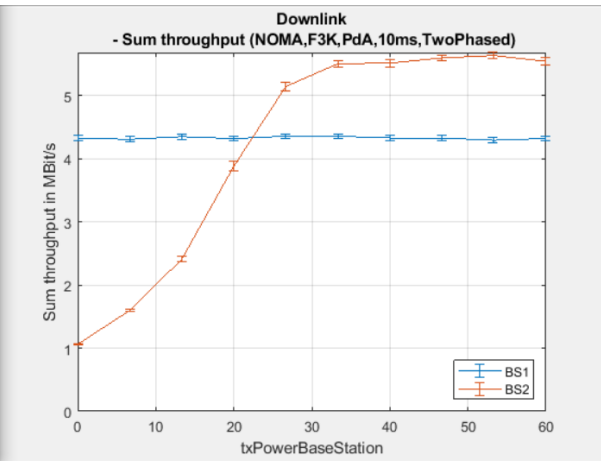
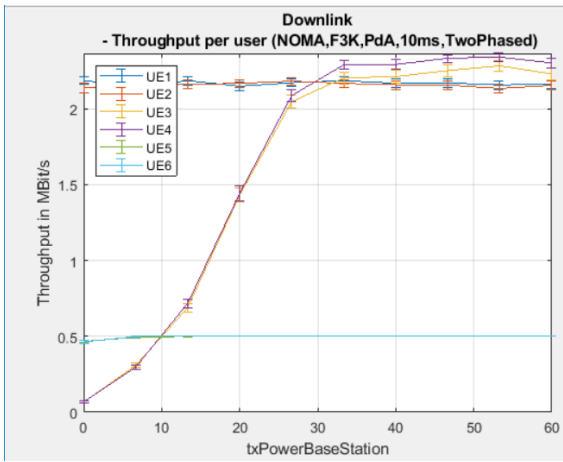
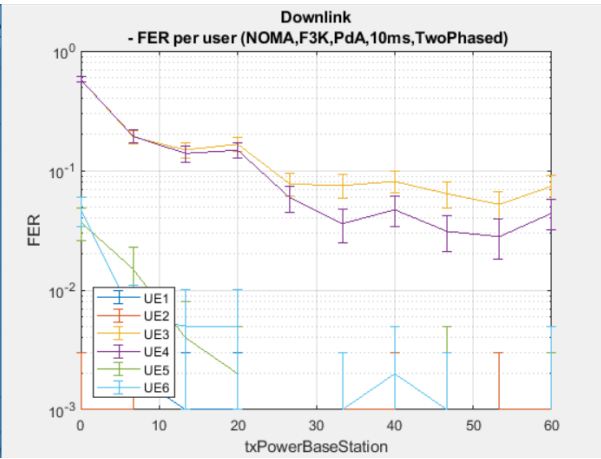
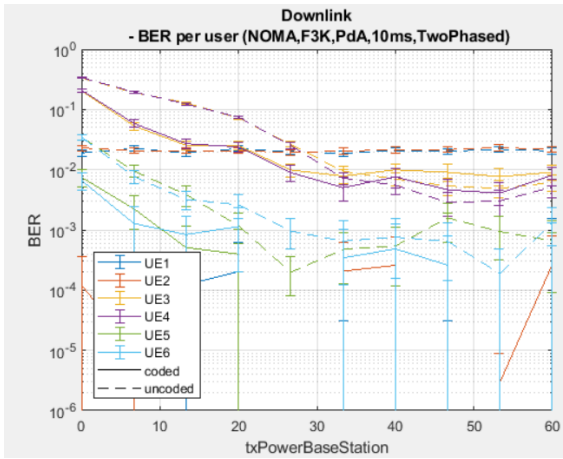
```
Editor - C:\Users\User\Documents\5GLLSimulator\TWO_PHASED_main.m
214 for iBS = 1:nBS
215     % Downlink
216     if simParams.simulation.simulateDownlink && ~isempty(Links(BS{iBS}.ID, iLink)) && strcmp(Links(BS{iBS}.ID, iLink)
217         test = Links(BS{iBS}.ID, iLink);
218         Links(BS{iBS}.ID, iLink).updateLink( simParams, Links, iFrame );
219     elseif simParams.simulation.simulateDownlink && ~isempty(Links(BS{iBS}.ID, iLink)) && strcmp(Links(BS{iBS}.ID, iL
220         Links(BS{iBS}.ID, iLink).Channel.NewRealization(iFrame, simParams.simulation.frameStructure, 'Downlink');
221     end
222 end
223 end
224 end
225 end
226 end
227
228
229 %% RATfor5G waterfilling here
230 simParams.schedule.fixedScheduleDL(2) = getSchedule4All(iSweep,iFrame,simParams,Links, BS);
231
232 %% RATfor5G - SECOND PHASE of updateLink for the Transmitter==2 (as the primary links of Transmitter1 are already fully u
233
234 %iBS = 2; % only BS=2 is in our interest for the moment
```

Rysunek 1 TWO_PHASED_main function

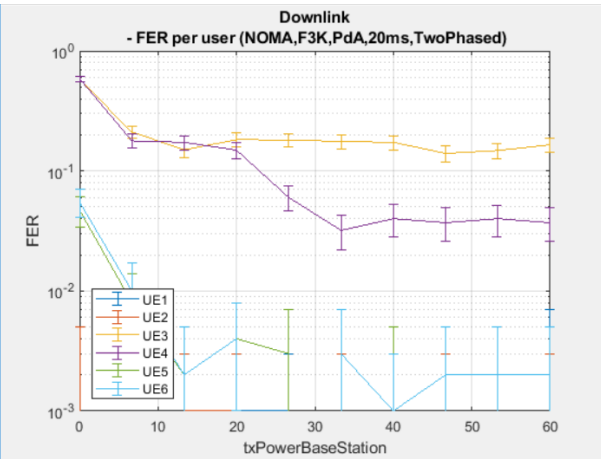
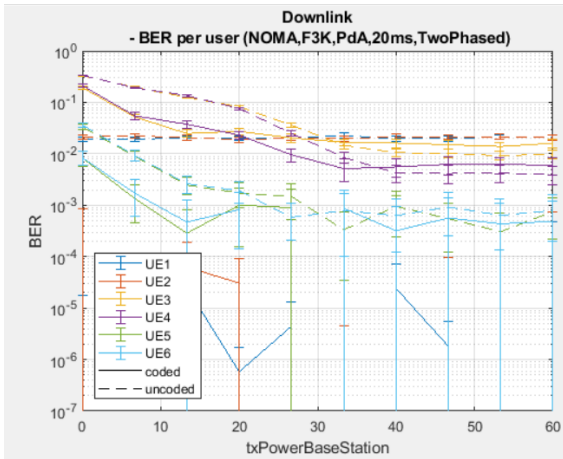
[0ms]

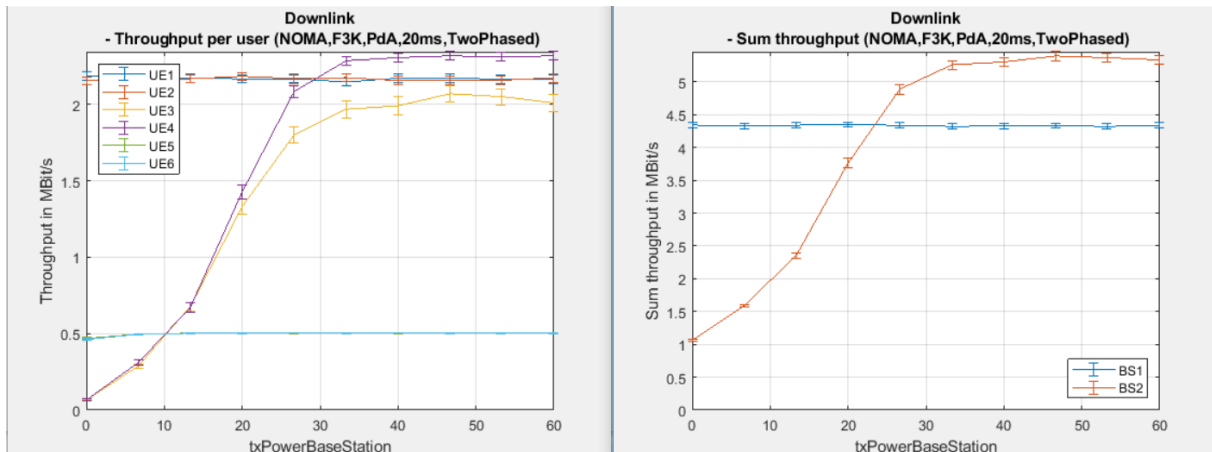


[10ms]



[20ms]



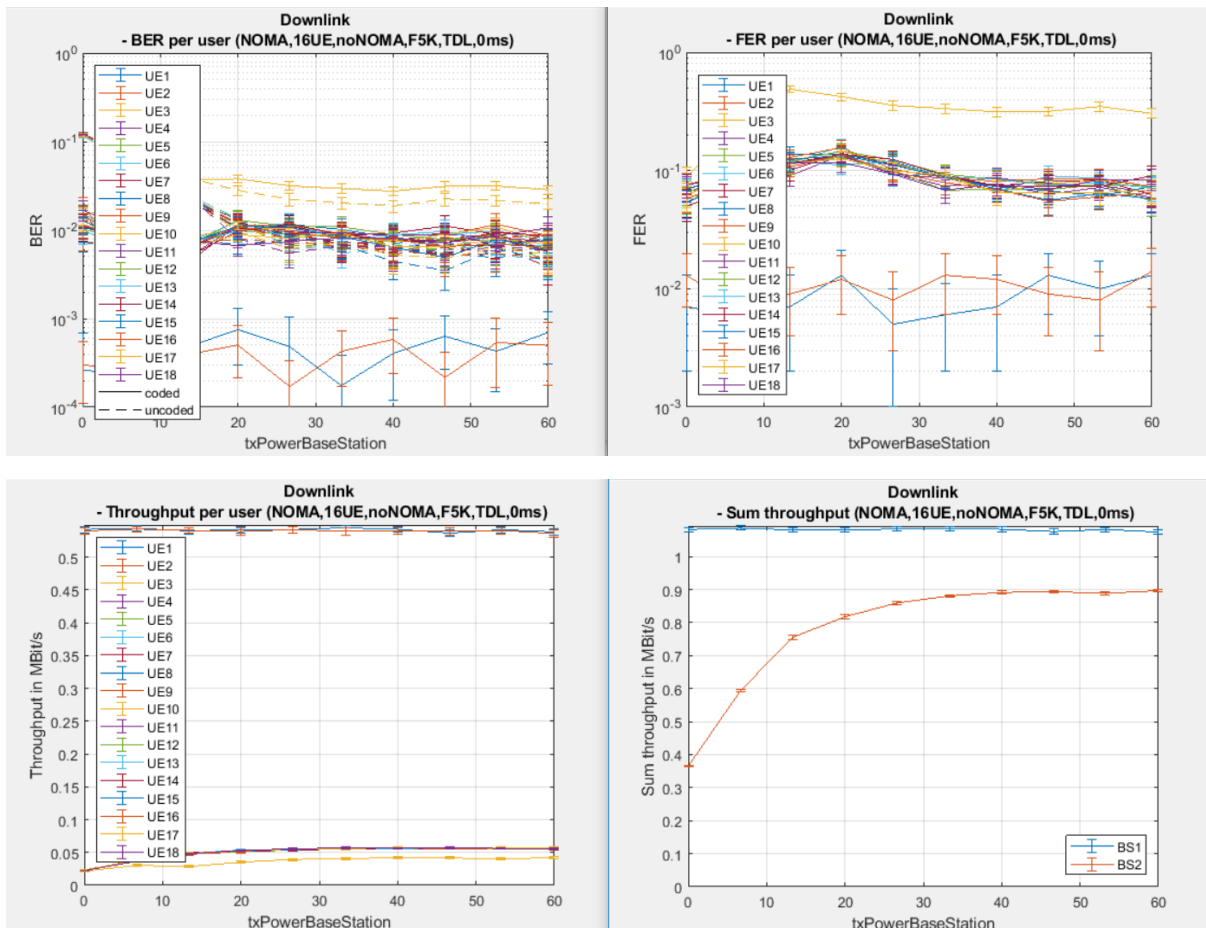


16.10.2020 – pierwsze wyniki dla **dużej liczby userów** (bez NOMA jeszcze)

- SYM1:
 - NOMA_16UE_noNOMA_F5K_TDL_0ms (ale dla 1000 ramek)
 - Resources
 - UE1, UE2 (BS1) = 8 podnosnych
 - UE3-UE18 = 1 podnośna każdy
 - Waterfilling = YES (per system)
 - `scStr.modulation.nSymbolsTotal = [14*18+1];`
 - PathLoss = 110 (all UE)
 - KOMENTARZ: *symulacja idzie bardzo wolno. Na koniec symulacji, już po wyświetleniu wykresów a przed zapisem wyników pojawił się komunikat 'out of memory' i że plik do zapisu 'might be corrupted'*
- SYM2:
 - j.w. ale:
 - `scStr.modulation.nSymbolsTotal = [15];`
 - żadne inne parametry się NIE ZMIENIAJĄ
 - KRYTYCZNE: Wyniki pokazują że to jest ZA MAŁĄ wartością żeby uzyskać rozsądne wyniki
- SYM3:
 - Ustawienia tak jak dla SYM1 ale **BEZ waterfilling**

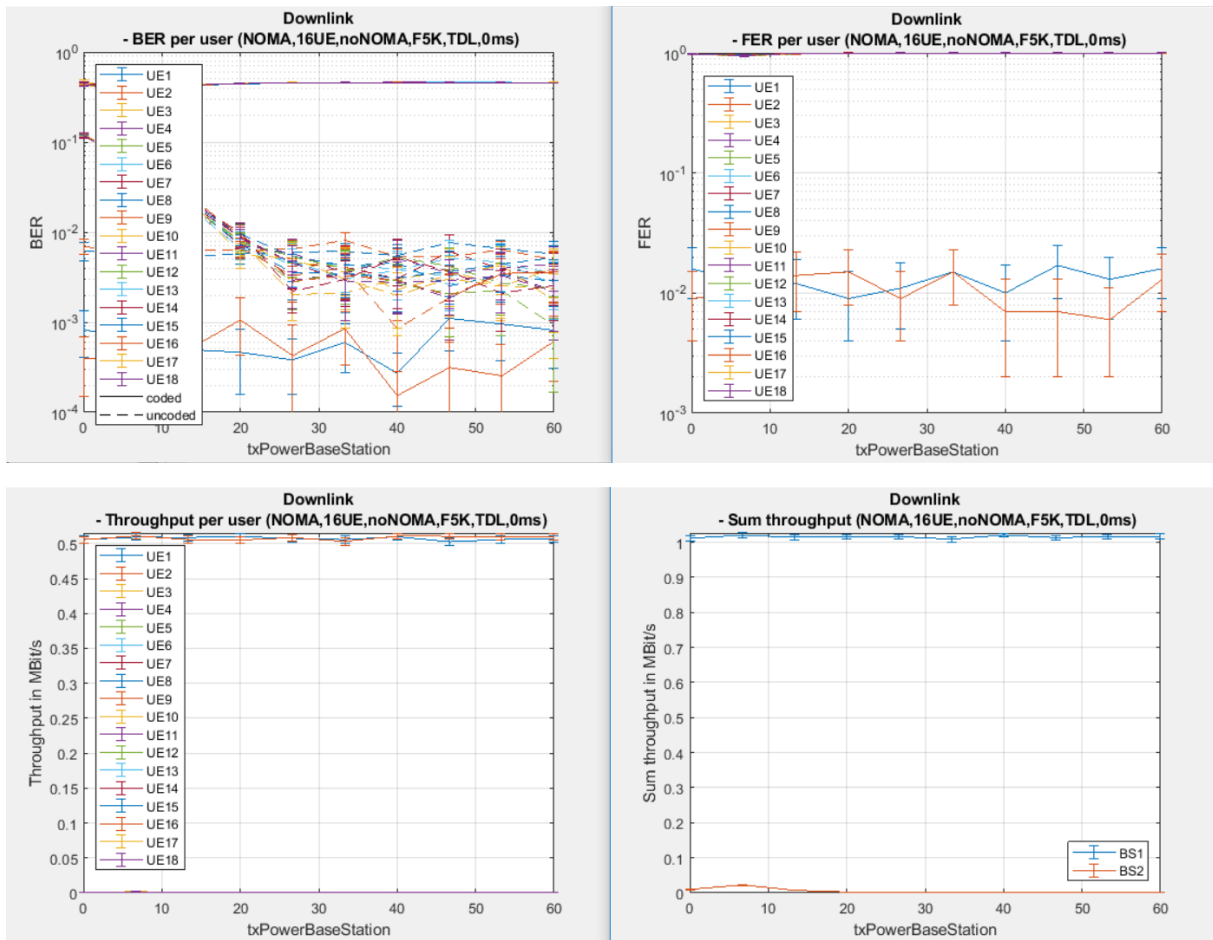
SYM1 (około 15h trwała symulacja; zużycie CPU około 30-40%, RAM około 50-60%)

- Różnica dla usera UE3 (żółty) może wynikać z tego że ma on inną wartość współczynnika PHY.K = 0. Other users PHY.K = 8. Inne parametry UE3 są takie same jak pozostałych.
- Ważne jest to że poniższe wyniki (a) w ogóle się policzyły/nie wysypały (b) to są wyniki dla velocity=0 więc jak się doda prędkości obliczenia mogą być dużo bardziej złożone
- Wyniki są około 4-razy mniejsze niż dla sytuacji bazowej „72 podnośnych” (np. na stronach 24-25). Na 72 podnośnych osiągamy około 5.5Mb/s max a tutaj uzyskujemy około 1Mb/s.
-



SYM2 (około 45min trwała symulacja; zużycie CPU około 20%, RAM około 25%)

WAŻNE: widać że liczba time symbols musi być odpowiednio duża żeby udało się wygenerować rozsądne wyniki.



SYM3 (CPU: 33-53%, RAM: 64%)

Zużycie pamięci na poziomie 62-64% jest stabilne w trakcie obliczeń danego sweep'a.

The screenshot shows a MATLAB simulation environment with the following components:

- Editor:** Displays MATLAB code for a 5G simulation. Key lines include:

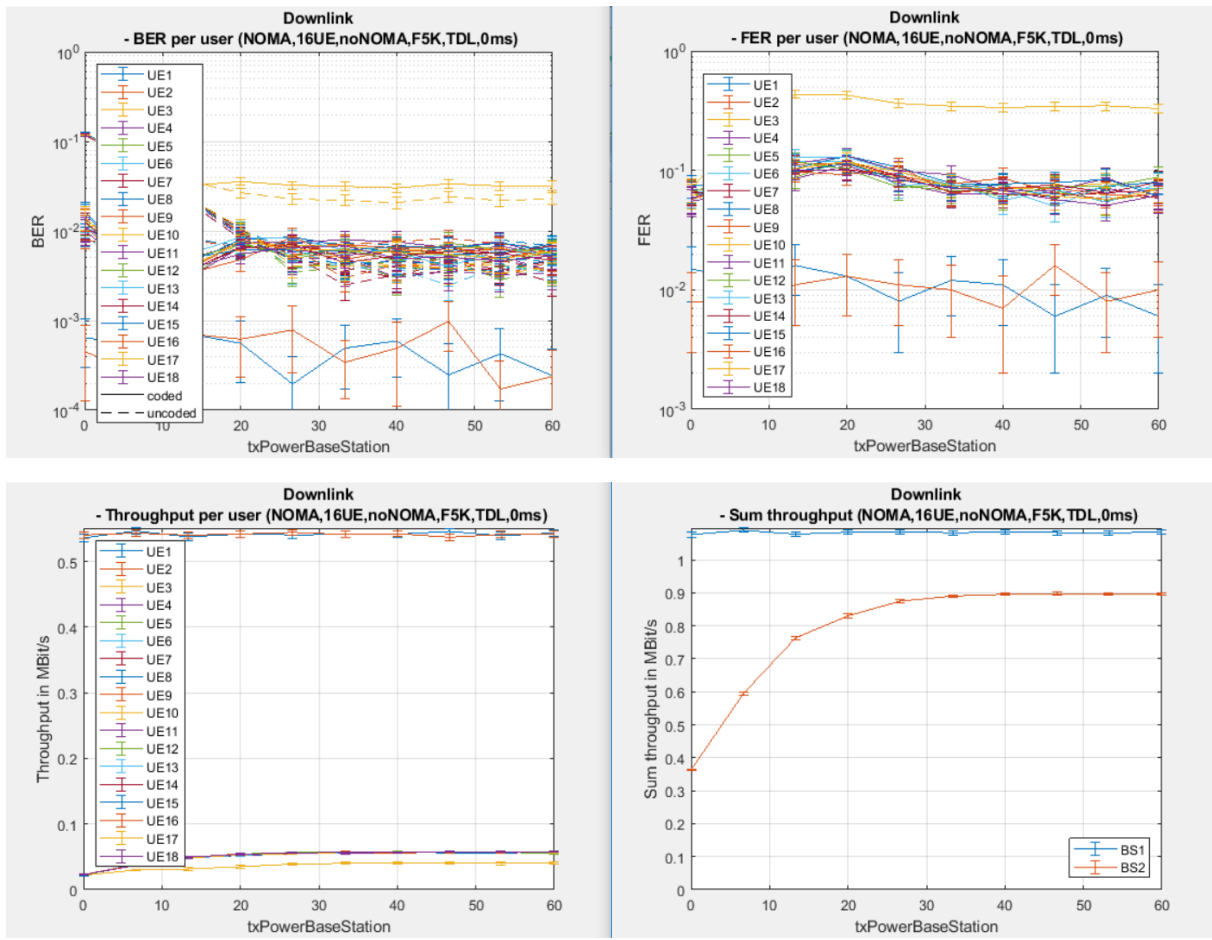

```

      %% RATfor5G waterfilling here
      %%simParams.schedule.fixedSchedule
      %% RATfor5G - SECOND PHASE of update
      %iBS = 2; % only BS=2 is in our in
      for iBS = 1:NBS
        for iLink = 1:L
          % Downlink
          if simParams.simulation.si
            test = Links(BS(iBS).I
            Links(BS(iBS).ID, iLin
            test.updatePhase = 1;
          elseif simParams.simulatio
            continue; %% the updat
          end
        end
      end
      end
      end
      
```
- Command Window:** Shows the output of the simulation, with the first line circled in red:


```

      Sweep=1, Frame = 1, BS = 2
      Sweep=1, Frame = 2, BS = 1
      Sweep=1, Frame = 2, BS = 2
      
```
- Resource Monitor:** Shows system resource usage:
 - CPU: 48% (2.19 GHz)
 - Pamięć: 19,8/31,0 GB (64%)
 - Dysk 0 (C:): 0%
 - Ethernet: Wysł: 200 Odebr.: 0 Kb/s

..po pierwszym Sweep zeszło 50tys (czyli 13h „spadło” w ciągu 1.5h)



Cel:

- Przyspieszenie realizacji obliczeń poprzez realizację docelowych testów na wielu maszynach dla dużej liczby ramek

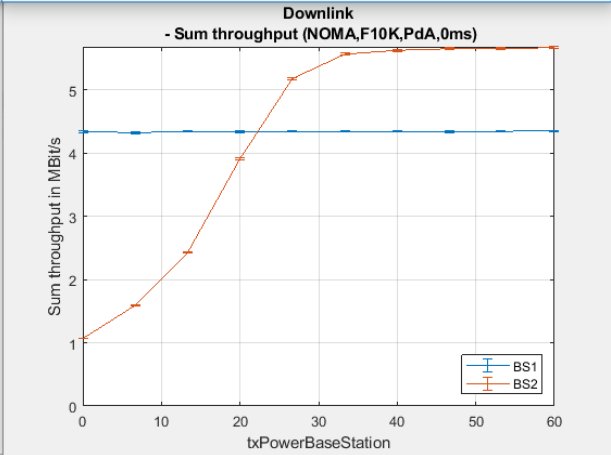
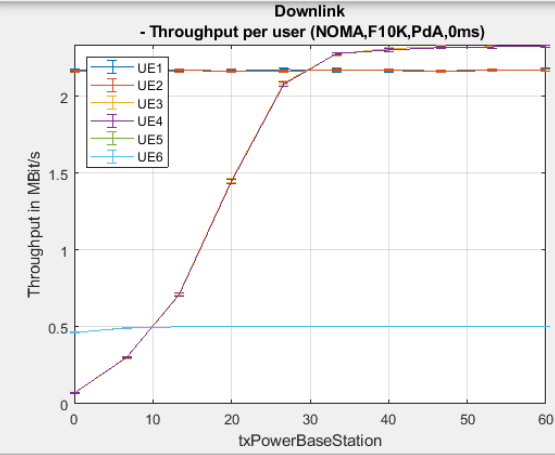
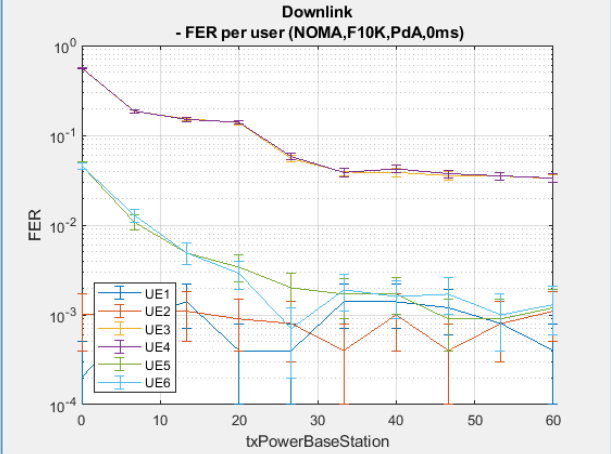
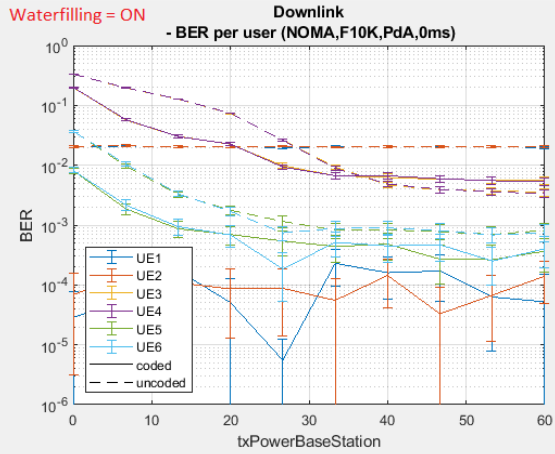
Założenia:

- Ustawienia główne
 - `scStr.simulation.pilotSpacingFrequency = 2;`
 - `scStr.simulation.pilotPatternUplink = 'Rectangular';`
 - `scStr.simulation.pilotPatternDownlink = 'Rectangular';`
 - `scStr.simulation.nFrames = 10000;`
 - `Waterfilling = ON (per system)`
- Waterfilling
 - WŁĄCZONY – dla obu przypadków żeby nie wprowadzać modyfikacji do wyników

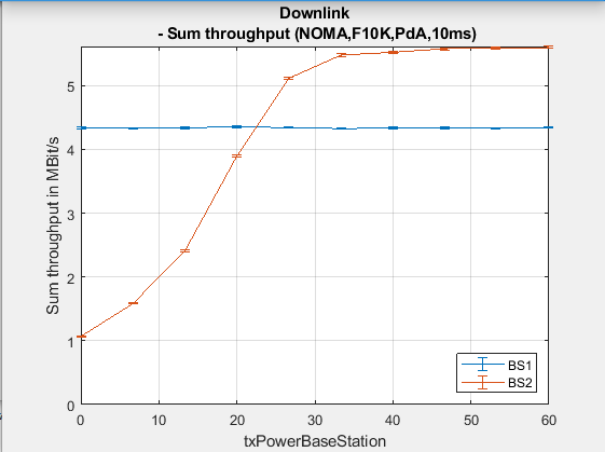
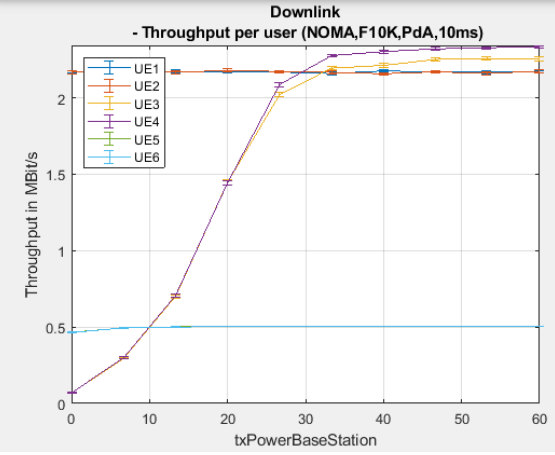
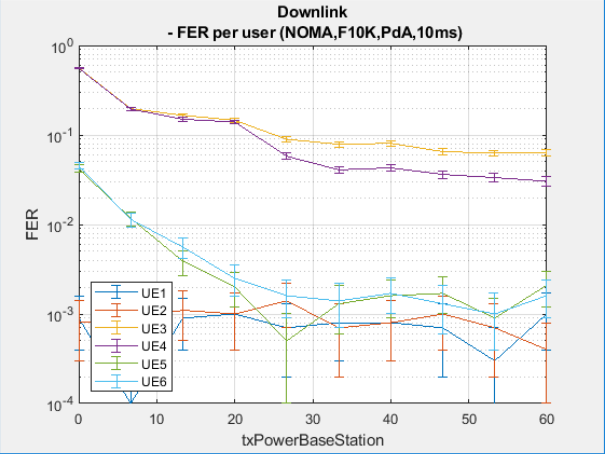
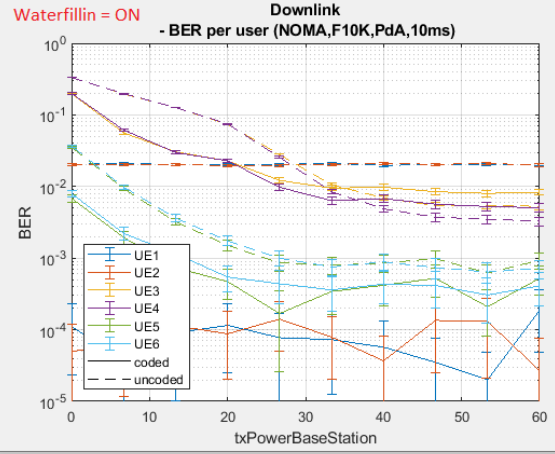
Wyniki

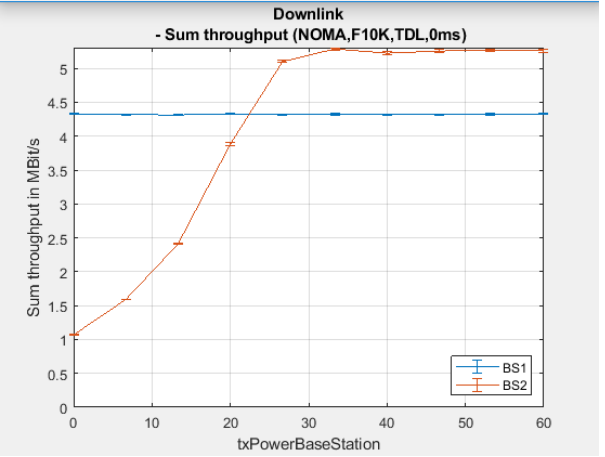
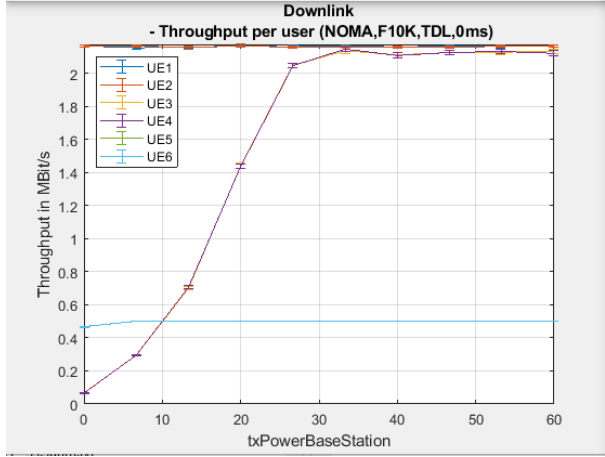
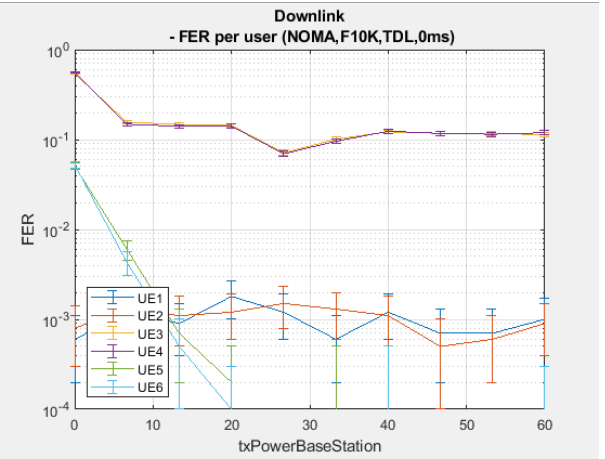
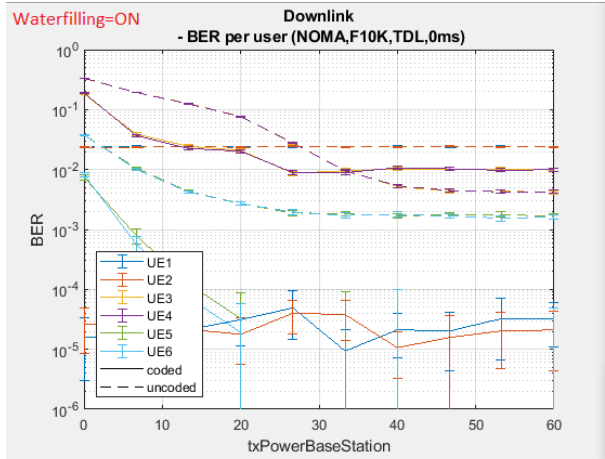
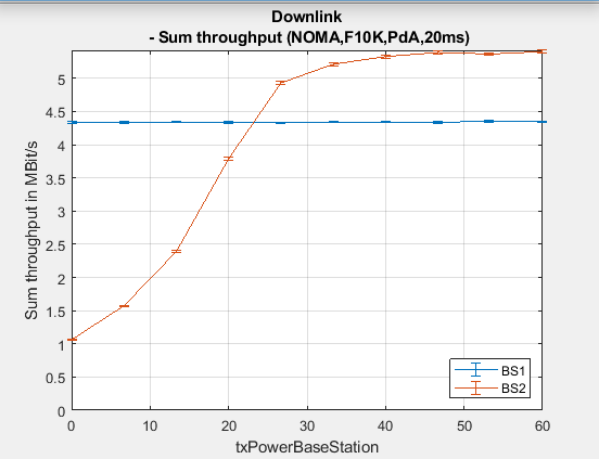
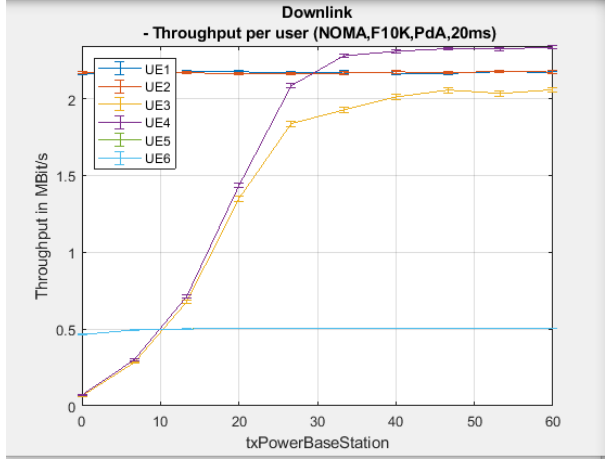
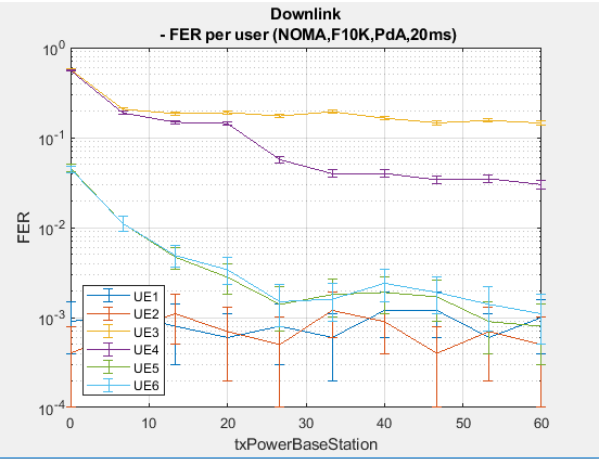
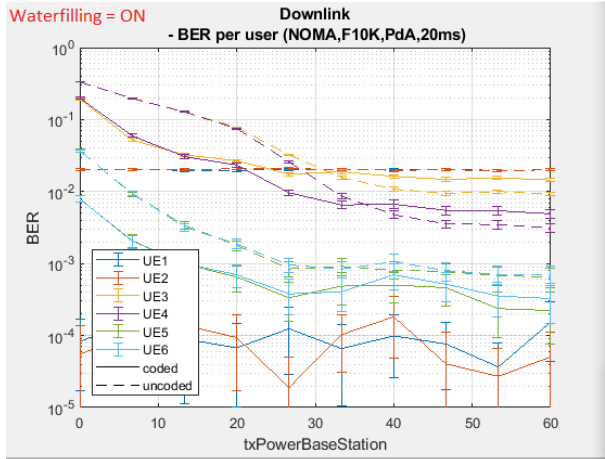
Komputer1 (z waterfilling)

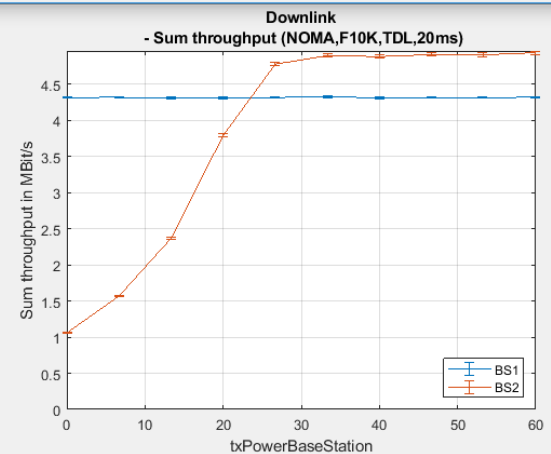
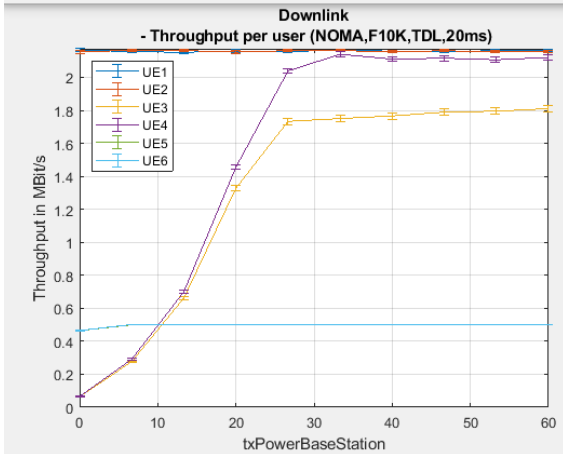
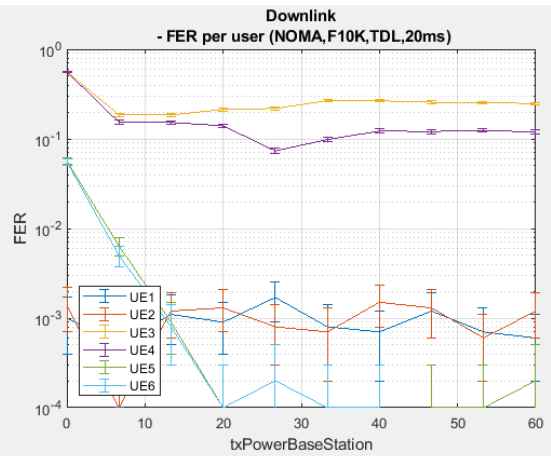
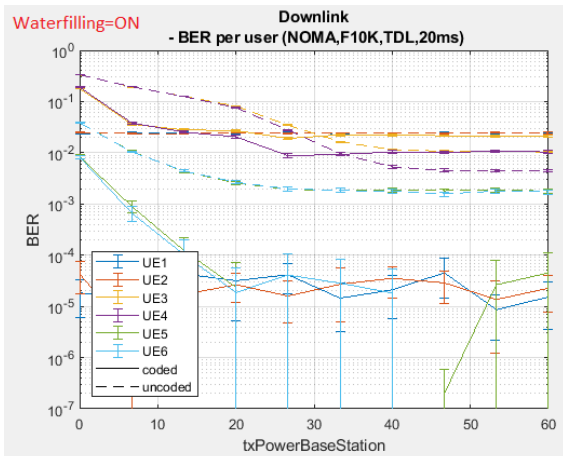
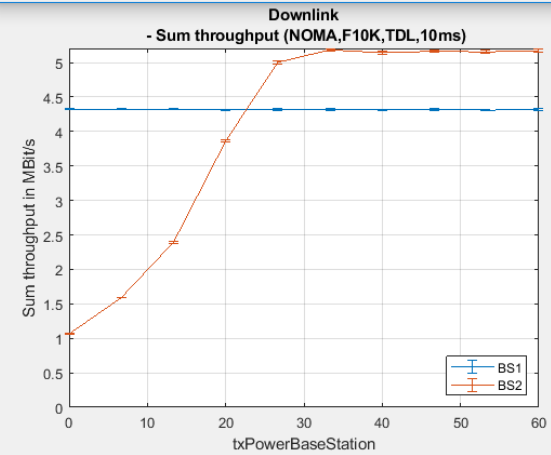
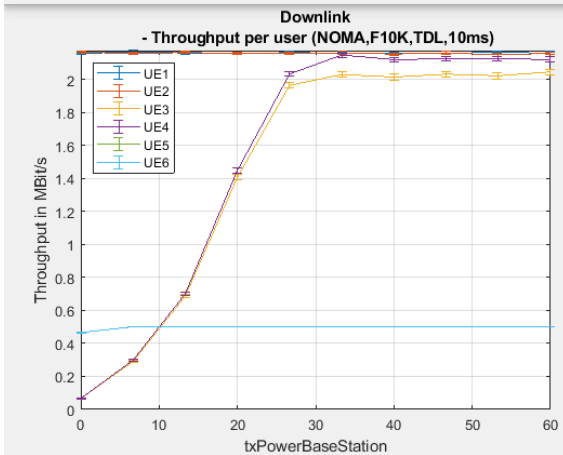
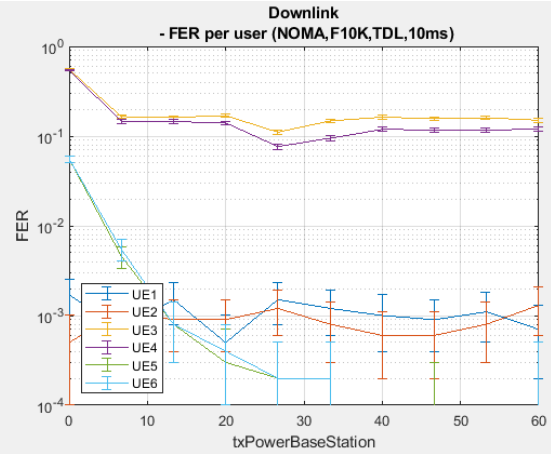
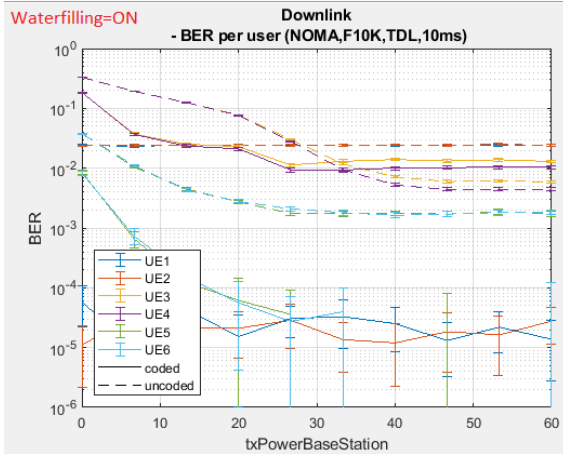
Waterfilling = ON



Waterfillin = ON

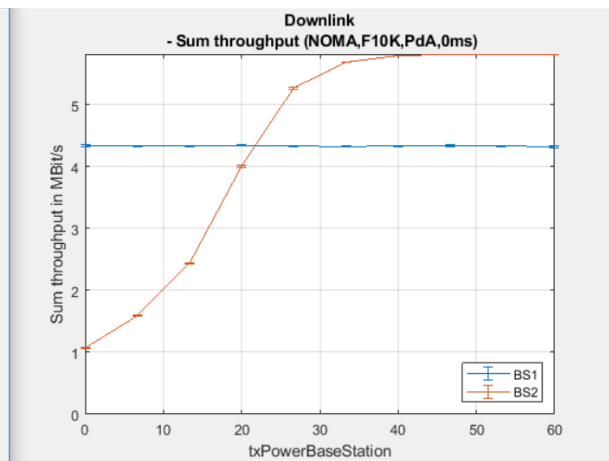
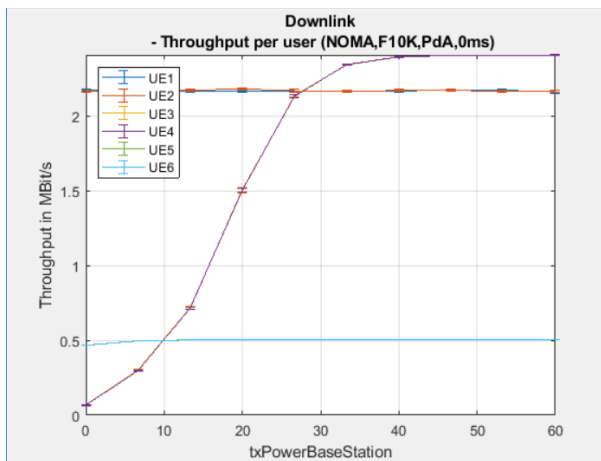
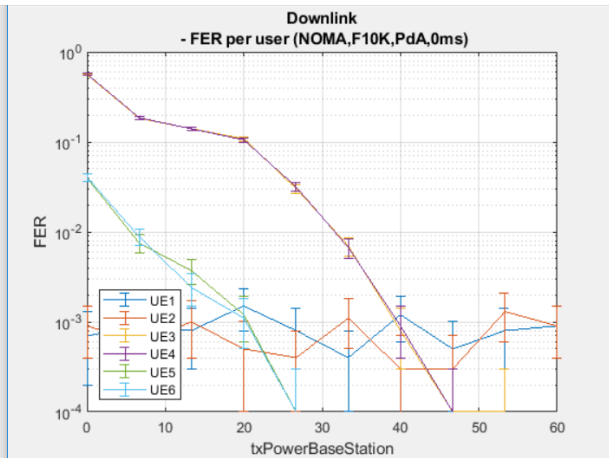
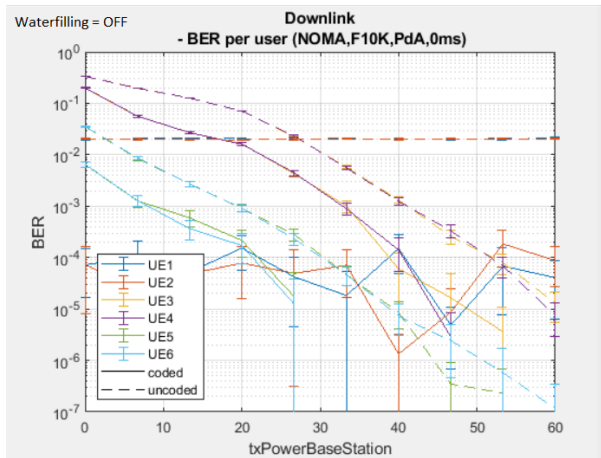




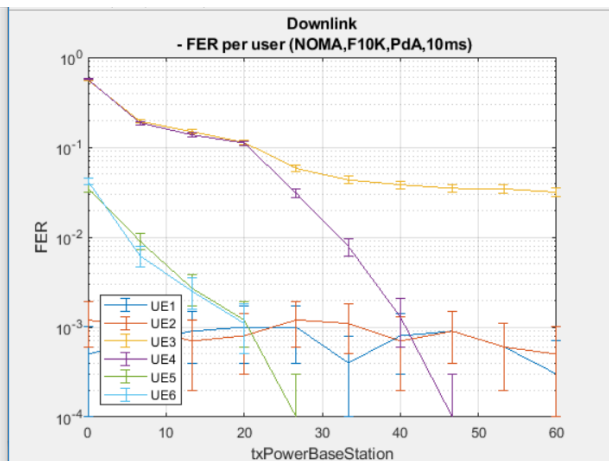
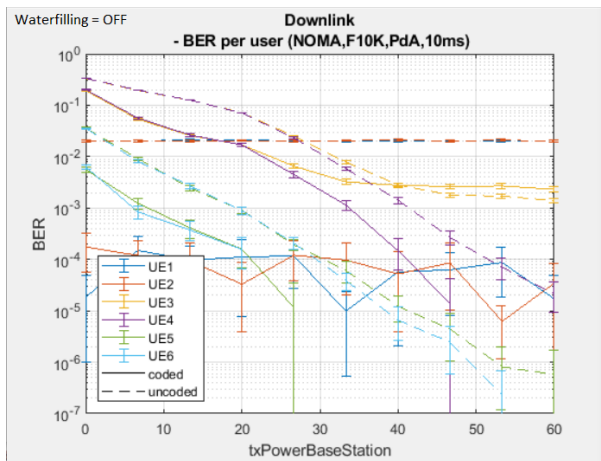


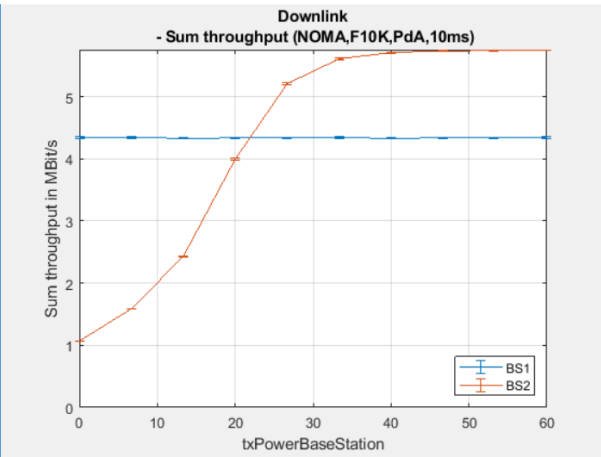
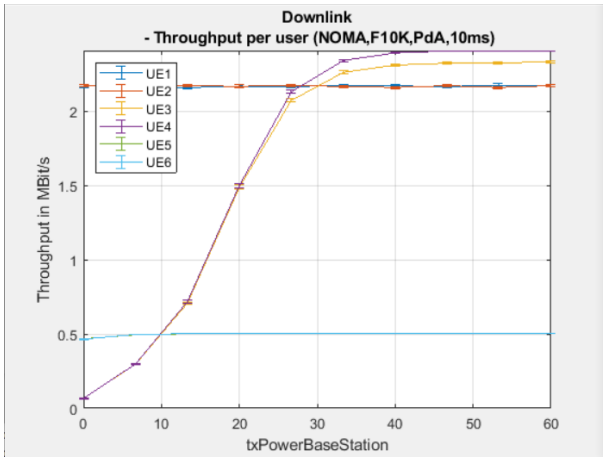
Komputer1 (bez waterfilling)

[Sim1]

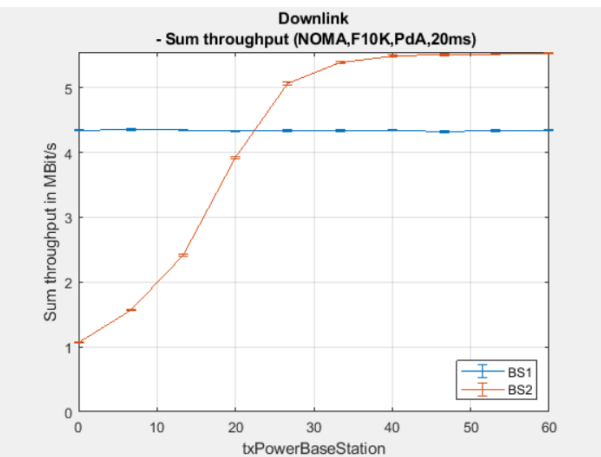
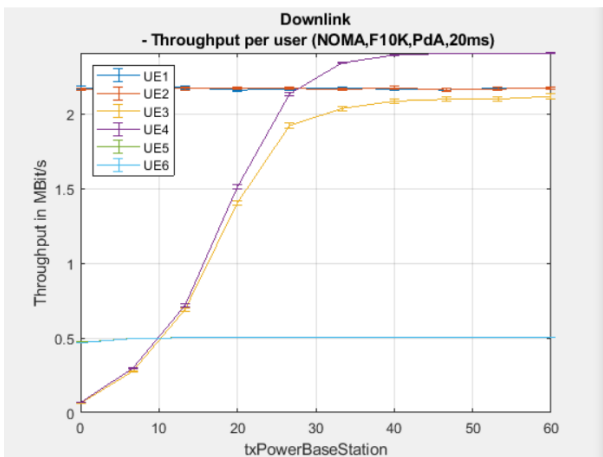
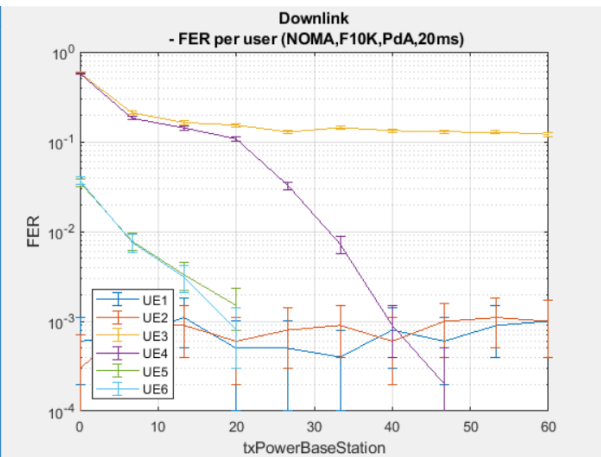
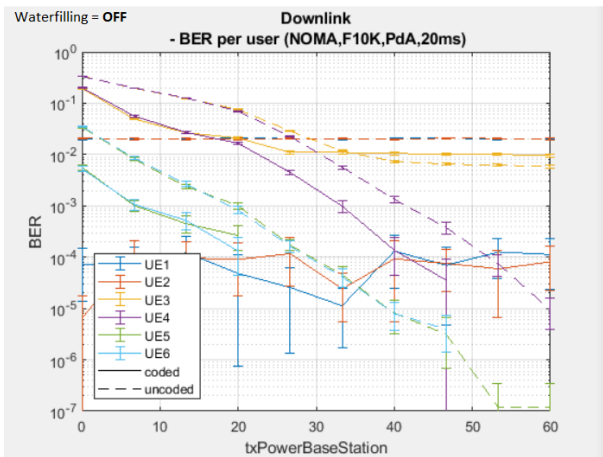


[Sim2]

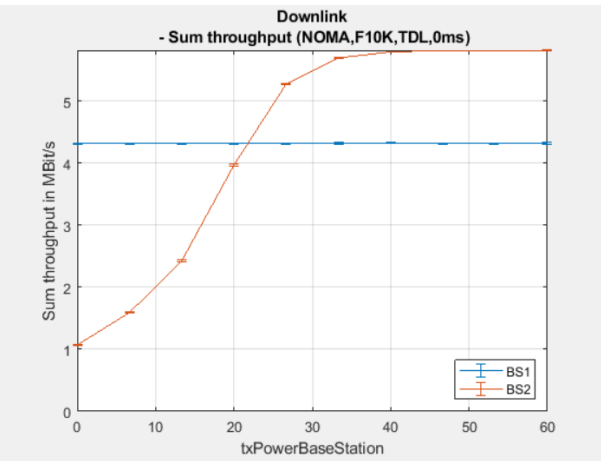
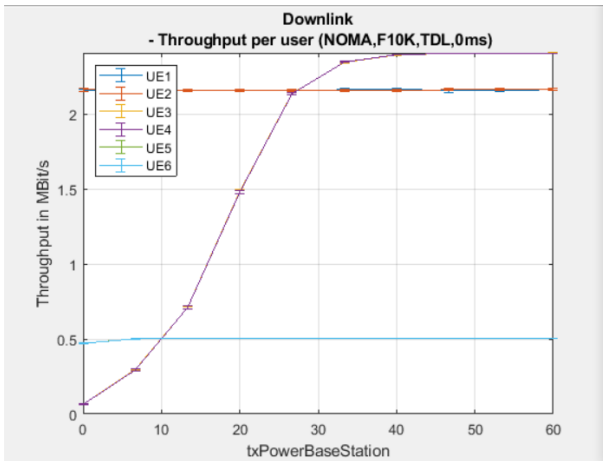
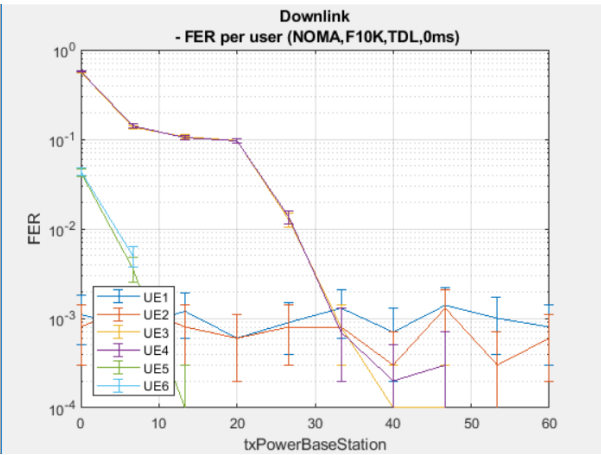
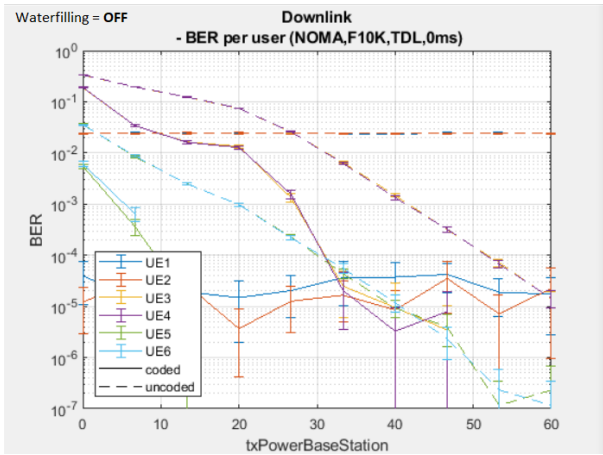




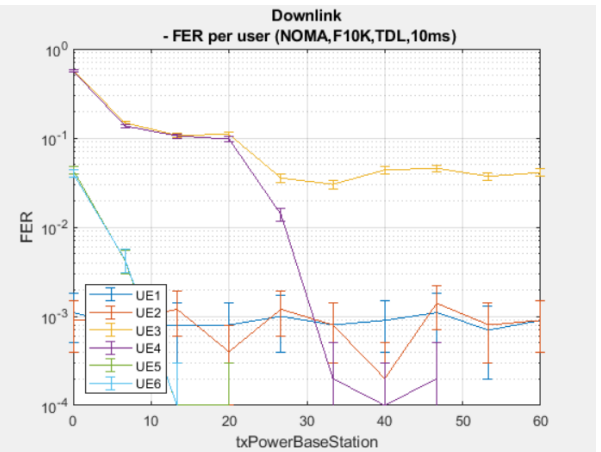
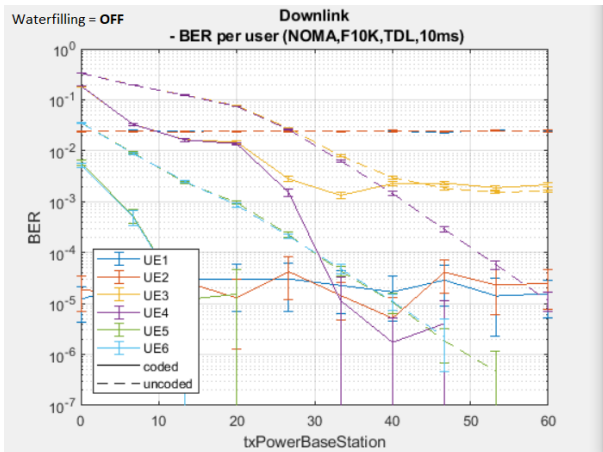
[Sim3]

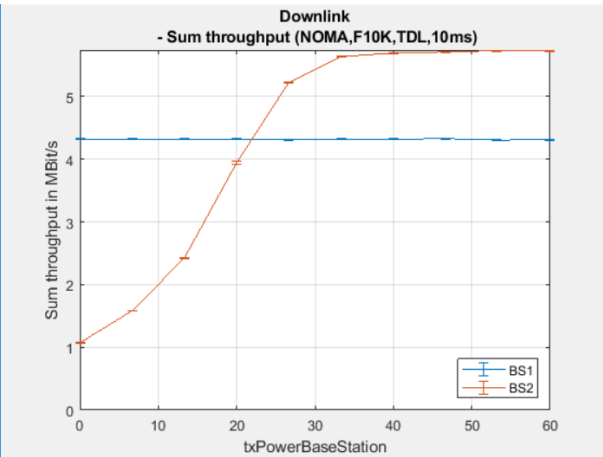
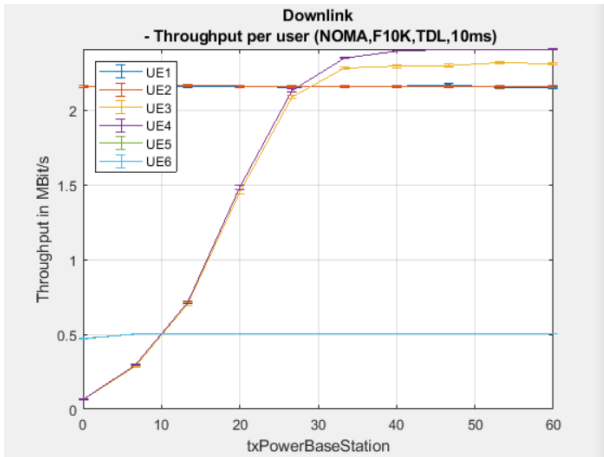


[Sim4]

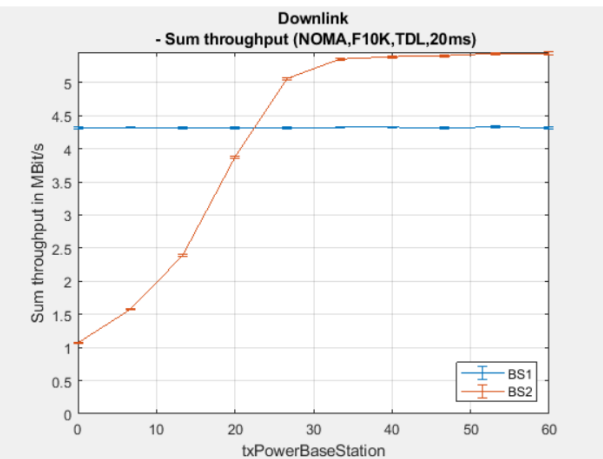
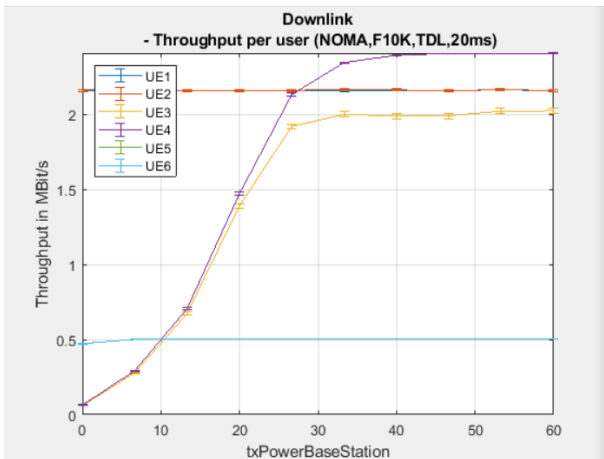
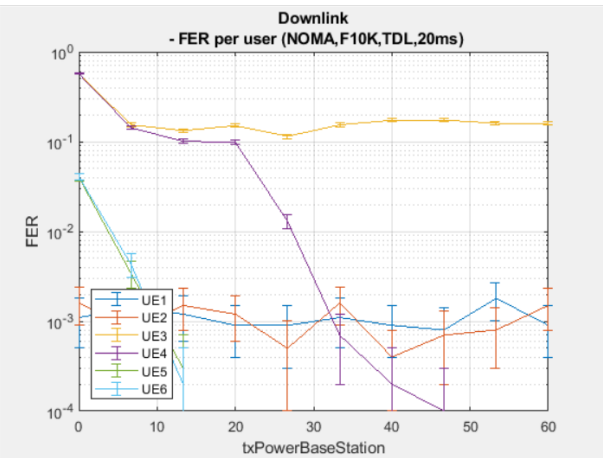
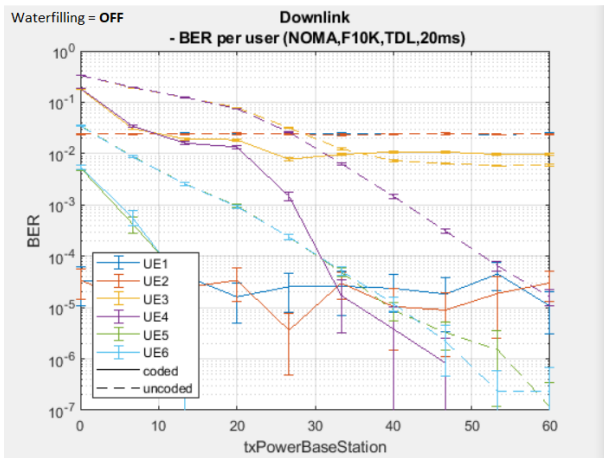


[Sim5]



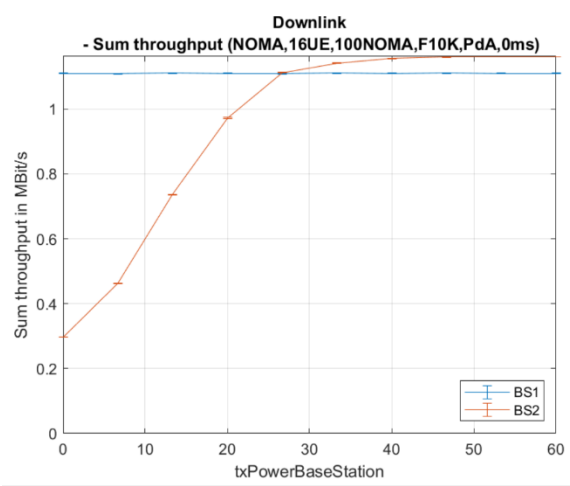
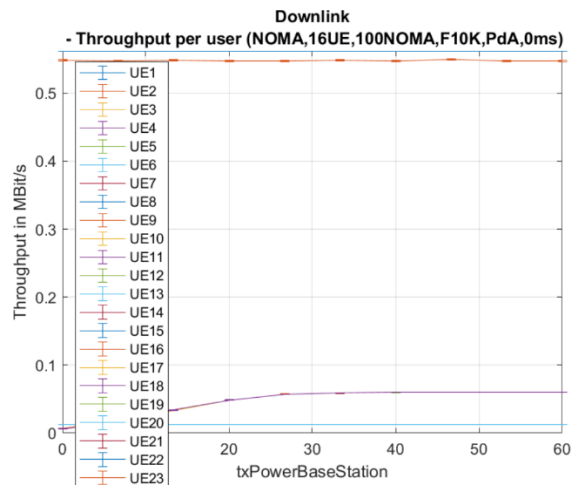
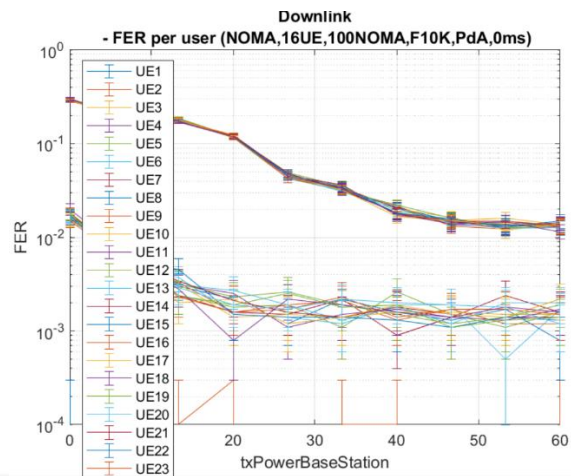
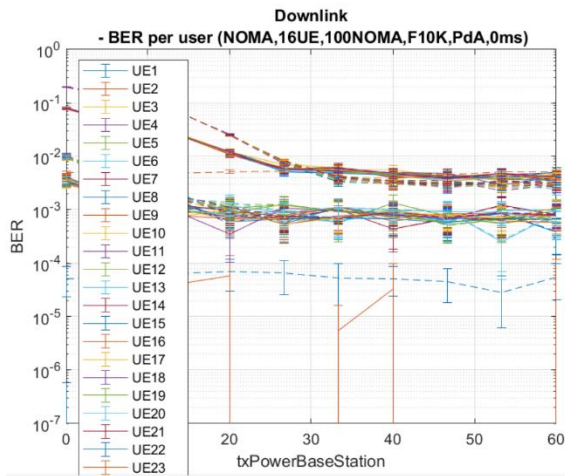


[Sim6]

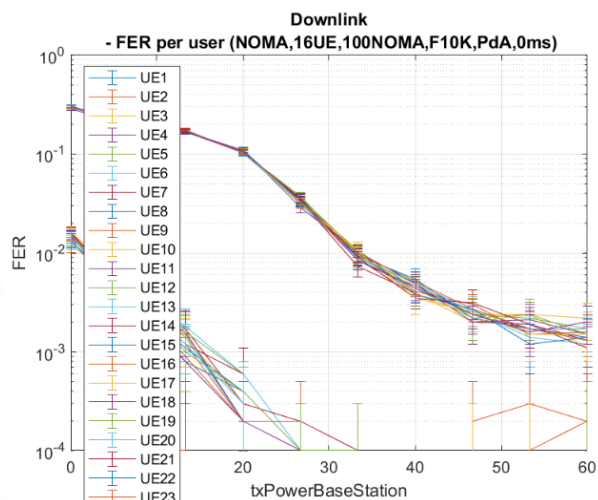
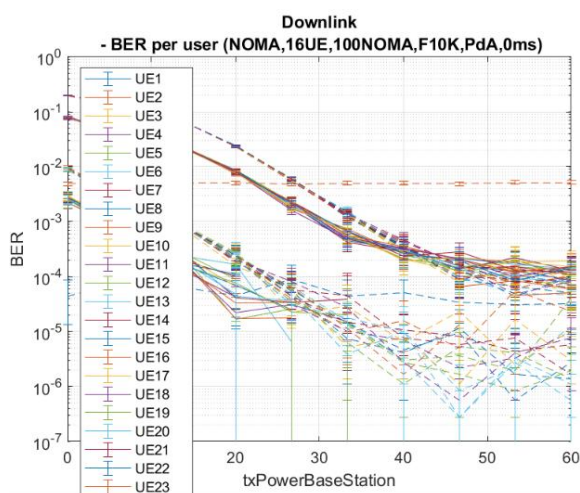


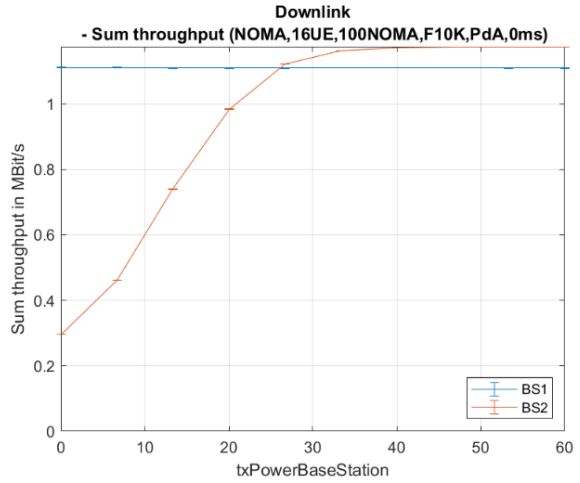
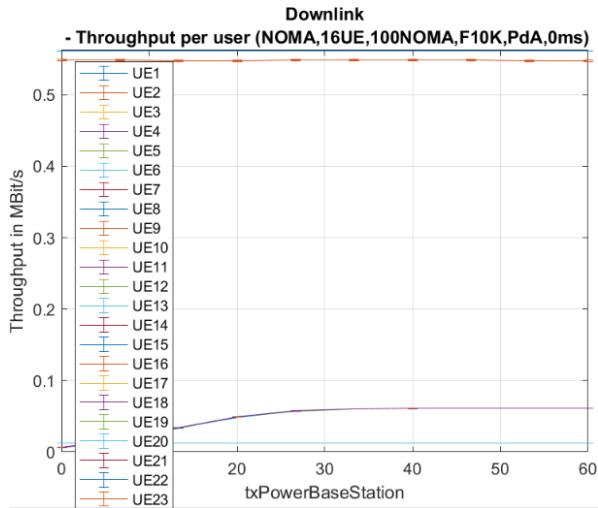
Komputer2 – wyniki

Tura1 (z waterfilling)

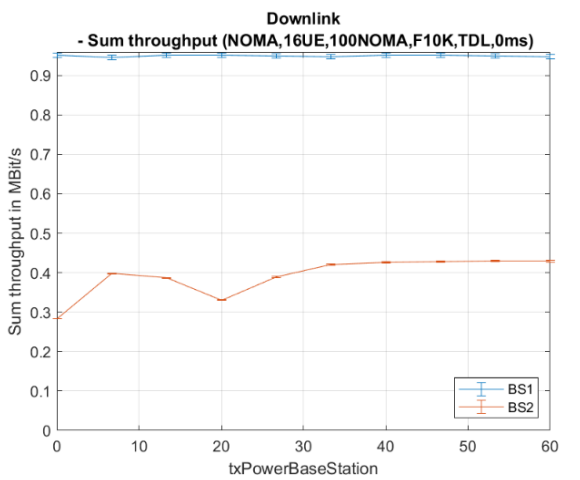
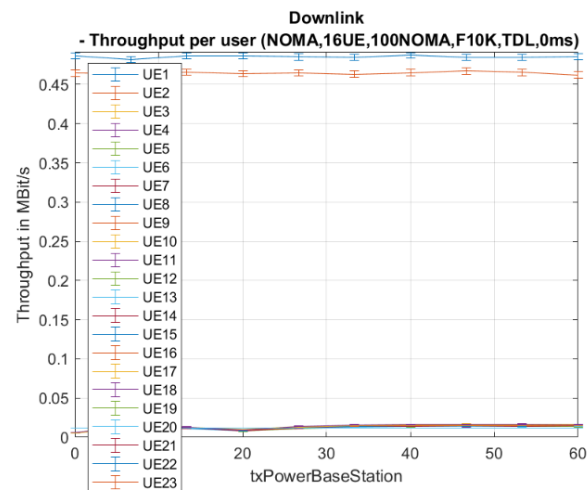
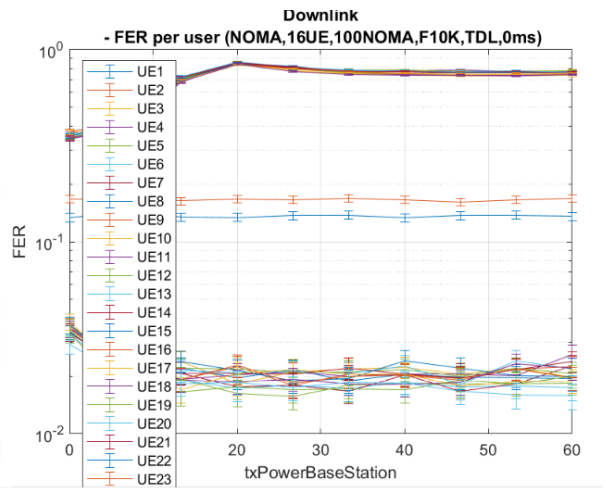
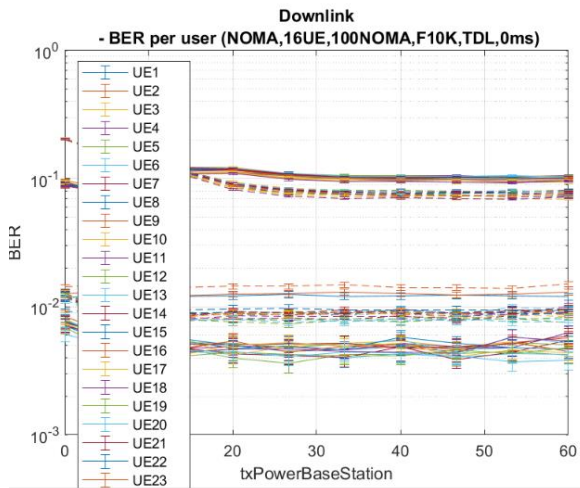


Tura2 (PdA – bez waterfilling)



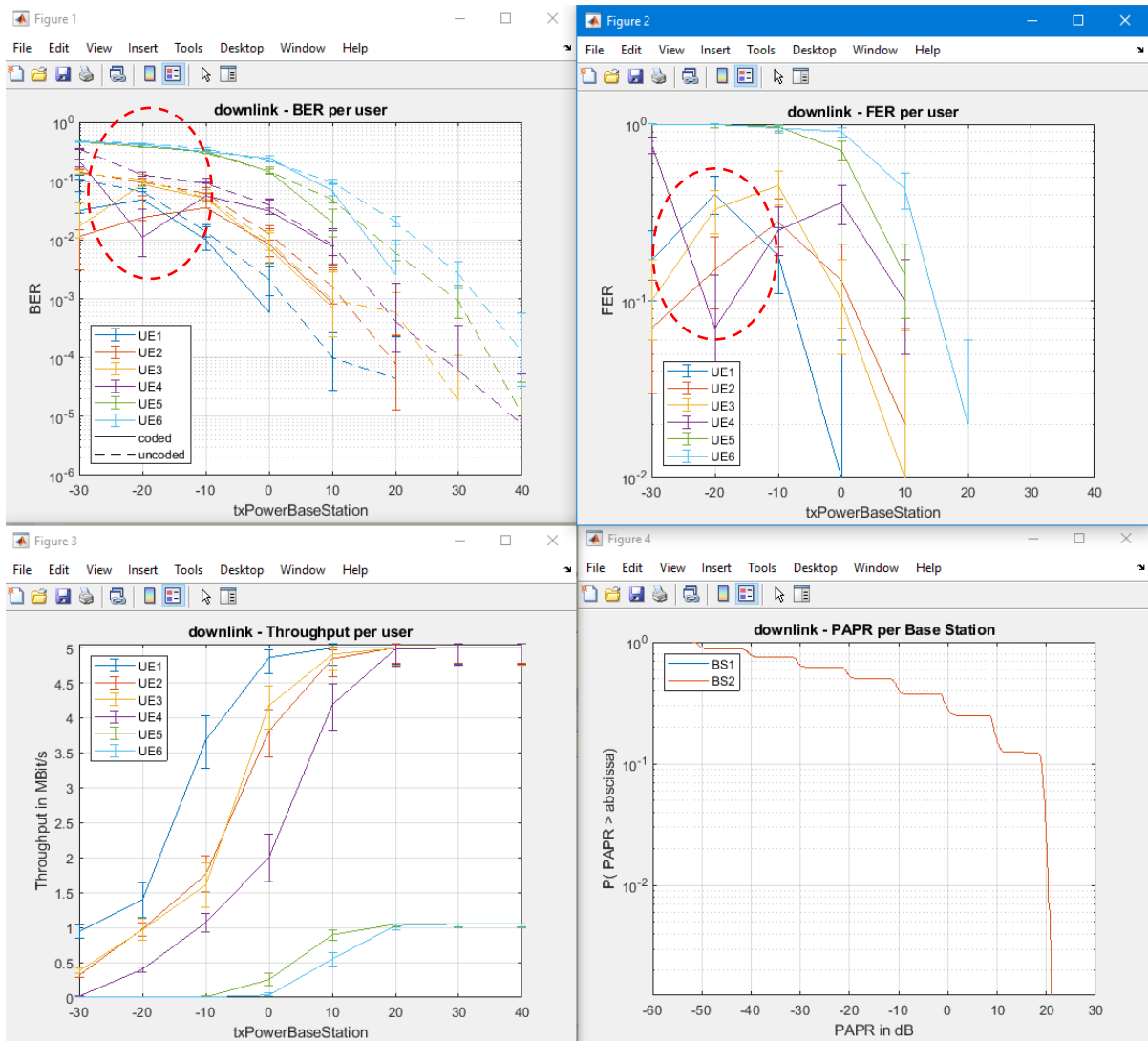


Tura3 (TDL – bez waterfilling)

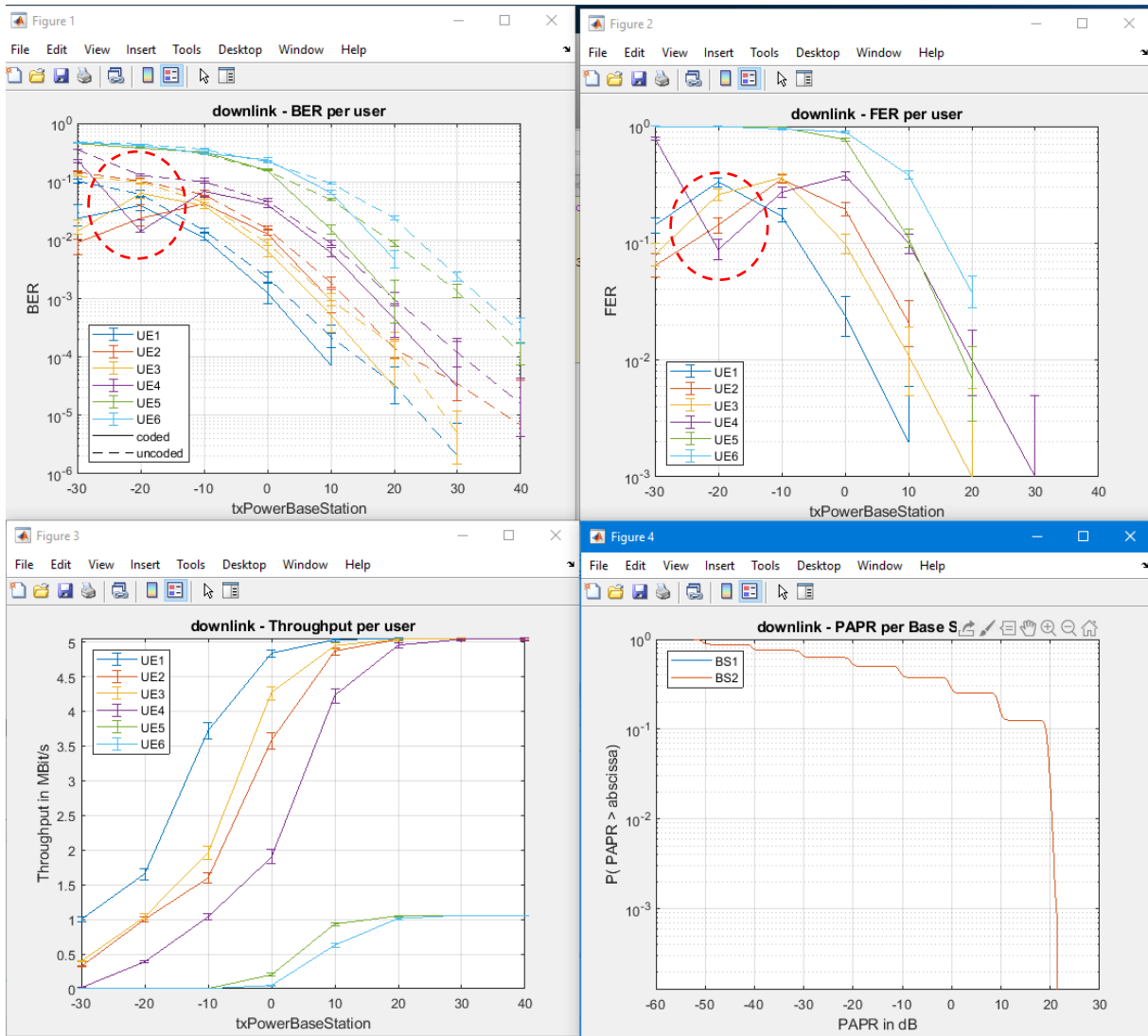


Tura4 (z waterfilling)

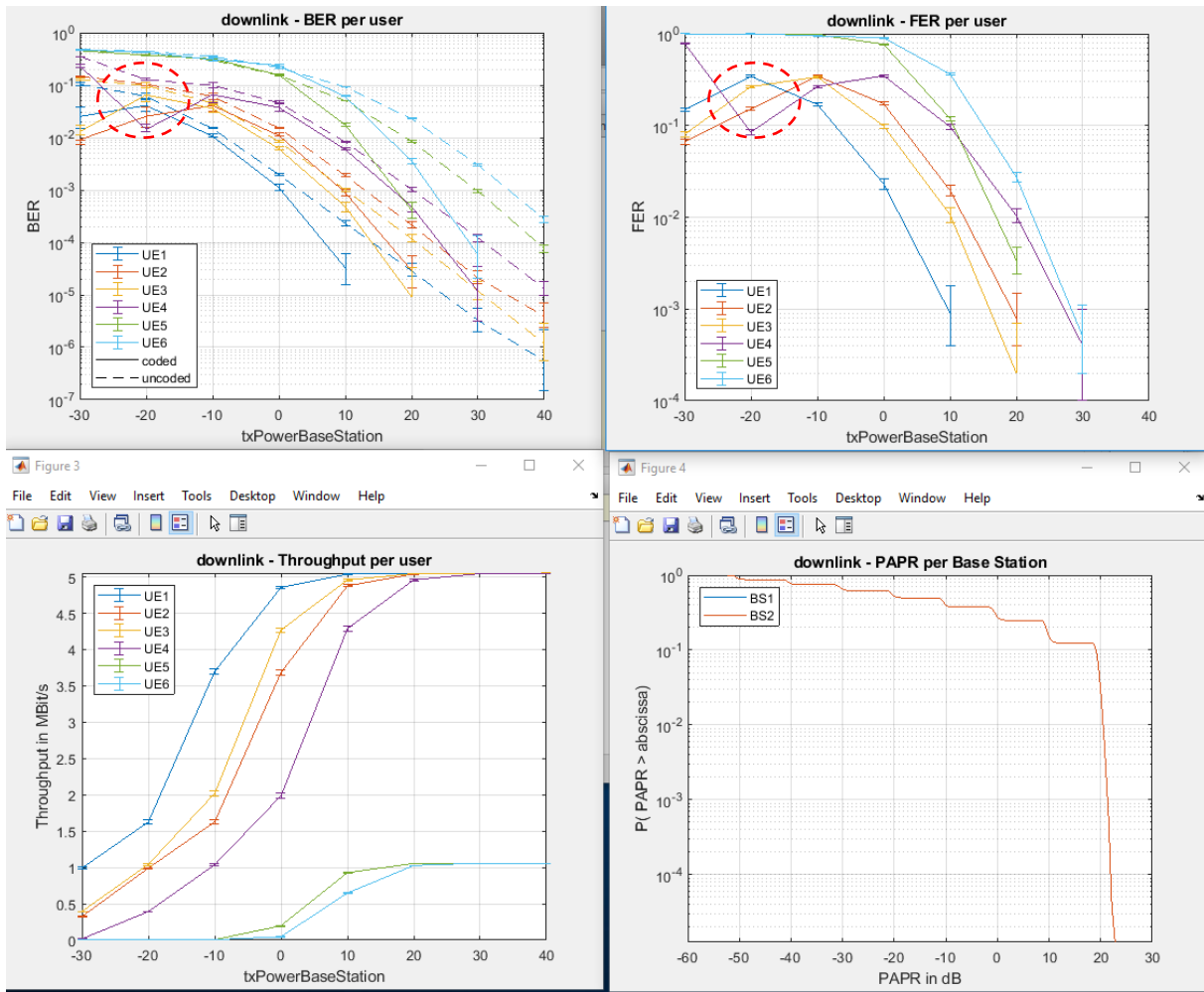
Frames = 100; (other params – Appendix1)



Frames = 1000 (other params – Appendix1)



Frames = 10000 (other params – Appendix1)



Podsumowanie

W ramach **przebadania algorytmów alokacji zasobów radiowych NOMA świadomych poziomem interferencji, w połączeniu z mechanizmami kontroli błędów** pozyskano licencję na symulator "Vienna 5G Link Level simulator". **Dokonano implementacji techniki waterfilling w symulatorze dla użytkowników OMA/NOMA.** Główna metoda realizująca sterowanie mocą zgodnie z podejściem waterfilling jest wywoływana w oparciu o parametry kanału radiowego generowanego każdorazowo przy tworzeniu nowej ramki OFDM. Aby zapewnić brak negatywnego wpływu zastosowanych zmian w kodzie na oryginalny kod, zespół wykonał niezbędne testy integracyjne i kalibracyjne. W wyniku testów potwierdzono poprawne i nie zaburzone działanie zmodyfikowanego kodu. Kontrola mocy na wyjściu w nadajniku pokazała, że zarówno przed jak i po załączeniu algorytmu waterfilling poziom mocy całkowitej pozostaje niezmienny, co potwierdza poprawność generacji sygnału.

Dotychczasowe wnioski i obserwacje z badań nad metodami ograniczania interferencji:

- uzyskane wyniki pokazują efekt zastosowania sterowania mocą użytkowników OMA i wpływ takich zmian na użytkowników OMA i NOMA w komórce (zgodnie z opisanymi wcześniej scenariuszami)
- zespół przebadał scenariusze z różnymi metodami kodowania (TC, LDPC, polarne). Osiągnięte rezultaty pokazują, że turbo kody są najszybsze, jeśli chodzi o symulacje (minimalny computing power) natomiast LDPC dawały lepsze rezultaty, ale symulacje wymagały więcej czasu (ze względu na większą złożoność obliczeniową)
- duża liczba użytkowników (Scenariusz1) w symulatorze prowadzi do istotnego wzrostu czasu obliczeń (ponadto wyniki zbierano dla różnej liczby symboli) – około 15h dla symbolu o długości

253 z aktywną funkcją waterfilling. Zmniejszenie długości symbolu do 15 skraca symulację do 45min. Zużycie CPU w trakcie symulacji to około 40%, a RAM do 60%.

- w trakcie prac nad implementacją metod sterowania mocą w symulatorze zidentyfikowano następujące ograniczenia po stronie środowiska symulacyjnego (a) brak możliwości sterowania mocą "per podnośna", (b) brak implementacji NOMA dla kierunku uplink
- żeby zapewnić poprawną weryfikację implementacji waterfilling, przy braku możliwości sterowania mocą "per podnośna" w oryginalnym symulatorze, postanowiono o przygotowaniu symulacji, w których użytkownicy będą korzystali z pojedynczych podnośnych. Dodatkowo zaplanowano także symulacje w których liczba podnośnych na użytkownika była >1 .
- Uzyskane wyniki pokazują, że zastosowanie funkcji „adaptive loading” w systemach OFDM w połączeniu z waterfilling nie daje dużej poprawy.

Appendix1

```
% parameter file for the NOMA scenario from "Versatile Mobile
Communications Simulation: The Vienna 5G Link Level Simulator"
%
% The setup is as follows: We have two cells (BSs) with 72 subcarriers
each.
% The first cell supports two users only and therefore each one gets 36
% subcarriers, while the second cell supports MUST operation and therefore
% can have two groups of two superimposed users (in the power-domain). In
% each group, one of the superimposed users has a much higher pathloss than
% the other one (i.e., a cell edge user). The results are obtained over the
% transmit power of the BSs. This scenario then shows the gain/impact of
the
% MUST operation compared to the case when it is not enabled.

% What to expect: UE1, UE2 and will be performing unaffected (orthogonal).
% UE3 and UE4 will suffer a performance loss at the low SNR regime due to
% the extra interference from UE5 and UE6, respectively. Overall, cell 2
will
% perform better since it can squeeze in UE5 and UE6 and therefore will
have
% a higher downlink throughput when the transmit power is sufficiently
high.

%% Topology
% Specify all the nodes in ascending order with starting
% index of 1 (BS0 or UE0 is not allowed).
scStr.topology.nodes = ['BS1,BS2,UE1,UE2,UE3,UE4,UE5,UE6'];

% Primary (desired) links
scStr.topology.primaryLinks = [ 'BS1:UE1','...
                                'BS1:UE2','...
                                'BS2:UE3','...
                                'BS2:UE4','...
                                'BS2:UE5','...
                                'BS2:UE6'
                                ];

Kierunek: DL
Sweep param: txPowerBS (-30,40,8)
Frames: 1000 (per sweep value)
Freq: 2,5GHz
P(BS): 30 dBm
P(UE): 30 dBm
BS(antennas): 2x2
User antennas: 2
User speed: 0 m/s
Pathloss:
    UE1, UE3 80dB
    UE2, UE4 90dB
    UE5, UE6 110dB // cell edge
amplifierOBO = 1dB // Amplifier output back-off, per BS, in dB
channel.DopplerModel: Jakes-discrete
Channel.TimeCorrelation: false
Channel.spatialCorrelation: none
channel.nPaths: 50;
channel.powerDelayProfile = 'PedestrianA';
channel.K = 0; //Rayleigh
scStr.channel.delta = 1;
```

```

%% Channel Estimation and Equalization Parameters
channelEstimationMethod = 'Approximate-Perfect';
noisePowerEstimation    = false;
pilotPattern            = 'LTE Downlink';
equalizerType           = 'One-Tap';
receiverTypeMIMO        = 'MMSE';

%% MIMO Parameters
% Layer mapping
layerMapping.mode       = '5G';
layerMapping.table.Uplink = {1;2;[1,2]};
layerMapping.table.Downlink = {1;2;[1,2]};

% MIMO mode
modulation.transmissionMode = 'CLSM';
modulation.delayDiversity  = 1;

%% Feedback Parameters
feedback.delay           = 0;
feedback.averager.Type  = 'miesm';
modulation.cqiTable     = 0;

feedback.cqi            = true;

scStr.modulation.nStreams = [2]; % per Link; number of active spatial
streams
scStr.modulation.mcs = [15]; % parameter is unused
% Per link, precoder selection (used when feedback is disabled)
scStr.modulation.precodingMatrix{1} = 1/sqrt(2)*ones(2,2); % per Link;
employed precoding matrix

%% Modulation Parameters
% waveform
scStr.modulation.waveform = {'OFDM'};

% time and bandwidth setup (number of subcarriers, frame duration, CP
% length, sampling rate)
modulation.numberOfSubcarriers = [72]; % per BS; number of used subcarriers
modulation.subcarrierSpacing = [15e3]; % per BS; per base station in Hz
modulation.nSymbolsTotal = [15]; % per BS; total number of time-symbols
per frame, the frame duration will be nSymbolsTotal/subcarrierSpacing
modulation.nGuardSymbols = [1]; % per BS; select how many of the total
time-symbols will be used as guard symbols (cyclic prefix in OFDM)
scStr.modulation.samplingRate = 'Automatic'; % sampling rate has to be the
same for all nodes (across all base stations):

%% Channel Coding Parameters
% All links are operating with the same coding parameters, enter it only
once.
coding.code = {'Turbo'};
coding.decoding = {'MAX-Log-MAP'};
coding.decodingIterations = [8];

%% Schedule
% static schedule per base station

% BS1 does Orthogonal Multiple Access

```

```
scStr.schedule.fixedScheduleDL{1}           = ['UE1:36,UE2:36'];           %
schedule for BS1 Downlink
scStr.schedule.fixedScheduleUL{1}           = [];                           %
No uplink for BS1.

% BS2 does MUST operation
scStr.schedule.fixedScheduleDL{2}           =
['UE3:36,UE4:36,UE5:UE3,UE6:UE4'];       % schedule for BS2 Downlink
scStr.schedule.fixedScheduleUL{2}           = [];
% No uplink for BS2.
```