Akademia Górniczo-Hutnicza
Wydział Informatyki, Elektroniki i Telekomunikacji
Kierunek Teleinformatyka

Metody Symulacyjne

Projekt 1

Analiza symulacyjna scenariusza agregacji ramek standardu 802.11n

Jan Ściga, Kamil Sobolak, Hubert Talar

Kraków, 26 kwietnia 2023

Spis treści

1 Analiza problemu **3**
1.1 Analiza implementacji scenariusza . . . 3
1.2 Cel symulacji . . . 6
1.3 Topologia badanej sieci . . . 7

2 Model systemu **7**
2.1 Ustawienia symulacji . . . 7
2.2 Parametry wejściowe symulacji . . . 8
2.3 Parametry wyjściowe symulacji . . . 8
2.4 Uproszczenia symulacji . . . 8

3 Symulacja zdarzeń dyskretnych **8**
3.1 Sekwencja zdarzeń symulacji . . . 8
3.2 Analiza czasu wykonania symulacji . . . 9

4 Parametry symulacji **10**
4.1 Ustawienia czasu symulacji . . . 10
4.2 Liczba niezależnych symulacji . . . 10
4.3 Czas rozruchu . . . 10
4.4 Wybrane parametry pomiarowe . . . 10

5 Wyniki symulacji **11**

6 Wnioski **13**

7 Podsumowanie **13**

1 Analiza problemu

W tym rozdziale zaprezentowano analizę problemu będącego przedmiotem raportu, która składa się z następujących części:

- Analiza implementacji scenariusza
- Cel symulacji
- Topologia badanej sieci

1.1 Analiza implementacji scenariusza

Po pobraniu przez użytkownika danych wejściowych zaprezentowanych w podrozdziale 2.2, program tworzy obiekty na podstawie abstrakcyjnych modeli dostępnych w ns-3. Są to stacje oraz access pointy (*Nodes*), kanał radiowy (*Channel*), oraz interfejsy (*NetDevices*). Dodatkowo następuje konfiguracja warstwy fizycznej oraz MAC transmitujących węzłów. Wymienione działania zaprezentowane zostało w Listingu 1.

```
NodeContainer wifiStaNodes;
 wifiStaNodes.Create (4);
 NodeContainer wifiApNodes;
 wifiApNodes.Create (4);

 YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
 YansWifiPhyHelper phy;
 phy.SetPcapDataLinkType (WifiPhyHelper::DLT_IEEE802_11_RADIO);
 phy.SetChannel (channel.Create ());

 WifiHelper wifi;
 wifi.SetStandard (WIFI_STANDARD_80211n);
 wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode", StringValue ("HtMcs7"), "
   ControlMode", StringValue ("HtMcs0"));
 WifiMacHelper mac;

 NetDeviceContainer staDeviceA, staDeviceB, staDeviceC, staDeviceD, apDeviceA, apDeviceB, apDeviceC,
   apDeviceD;
 Ssid ssid;
```

Listing 1: Modelowanie abstrakcji ns-3

Kolejnym etapem wykonania programu jest konfiguracja wszystkich czterech sieci wykorzystywanych w symulacji. Każda z sieci operuje w paśmie 5 GHz jednak z wykorzystaniem innych kanałów co zapewnia separacje transmisji. W przypadku sieci A (*Network A*) testowany jest przypadek domyślnej agregacji 802.11n , co widoczne jest w Listingu 2.

```
// Network A
 ssid = Ssid ("network-A");
 phy.Set ("ChannelSettings", StringValue ("{36, 0, BAND_5GHZ, 0}"));

 mac.SetType ("ns3::StaWifiMac",
              "Ssid", SsidValue (ssid));
 staDeviceA = wifi.Install (phy, mac, wifiStaNodes.Get (0));

 mac.SetType ("ns3::ApWifiMac",
              "Ssid", SsidValue (ssid),
              "EnableBeaconJitter", BooleanValue (false));
 apDeviceA = wifi.Install (phy, mac, wifiApNodes.Get (0));
```

Listing 2: Konfiguracja sieci A

Sieć B (*Network B*) przedstawia przypadek całkowicie wyłączonej agregacji co widoczne jest w sekcjach *Disable A-MPDU* w Listingu 3.

```
 // Network B
 ssid = Ssid ("network-B");
 phy.Set ("ChannelSettings", StringValue ("{40, 0, BAND_5GHZ, 0}"));
 mac.SetType ("ns3::StaWifiMac",
              "Ssid", SsidValue (ssid));

 staDeviceB = wifi.Install (phy, mac, wifiStaNodes.Get (1));

 // Disable A-MPDU
 Ptr<NetDevice> dev = wifiStaNodes.Get (1)->GetDevice (0);
 Ptr<WifiNetDevice> wifi_dev = DynamicCast<WifiNetDevice> (dev);
 wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmpduSize", UintegerValue (0));

 mac.SetType ("ns3::ApWifiMac",
              "Ssid", SsidValue (ssid),
              "EnableBeaconJitter", BooleanValue (false));
 apDeviceB = wifi.Install (phy, mac, wifiApNodes.Get (1));
```

```
18
19  // Disable A-MPDU
20  dev = wifiApNodes.Get (1)->GetDevice (0);
21  wifi_dev = DynamicCast<WifiNetDevice> (dev);
22  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmpduSize", UintegerValue (0));
```

Listing 3: Konfiguracja sieci B

Kolejną konfigurowaną siecią jest Sieć C (*Network C*). Ten przypadek obejmuje wyłączenie agregacji A-MPDU oraz włączenie A-MSDU co widoczne jest w Listingu 4.

```
1   // Network C
2   ssid = Ssid ("network-C");
3   phy.Set ("ChannelSettings", StringValue ("{44, 0, BAND_5GHZ, 0}"));
4   mac.SetType ("ns3::StaWifiMac",
5                "Ssid", SsidValue (ssid));
6
7   staDeviceC = wifi.Install (phy, mac, wifiStaNodes.Get (2));
8
9   // Disable A-MPDU and enable A-MSDU with the highest maximum size allowed by the standard (7935 bytes)
10  dev = wifiStaNodes.Get (2)->GetDevice (0);
11  wifi_dev = DynamicCast<WifiNetDevice> (dev);
12  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmpduSize", UintegerValue (0));
13  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmsduSize", UintegerValue (7935));
14
15  mac.SetType ("ns3::ApWifiMac",
16               "Ssid", SsidValue (ssid),
17               "EnableBeaconJitter", BooleanValue (false));
18  apDeviceC = wifi.Install (phy, mac, wifiApNodes.Get (2));
19
20  // Disable A-MPDU and enable A-MSDU with the highest maximum size allowed by the standard (7935 bytes)
21  dev = wifiApNodes.Get (2)->GetDevice (0);
22  wifi_dev = DynamicCast<WifiNetDevice> (dev);
23  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmpduSize", UintegerValue (0));
24  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmsduSize", UintegerValue (7935));
```

Listing 4: Konfiguracja sieci C

Ostatnią konfigurowaną siecią jest Sieć D (*Network D*) widoczna w Listingu 5, która w przeciwieństwie do poprzedniego przykładu obejmuje włączenie agregacji A-MPDU oraz wyłączenie A-MSDU.

```
1   // Network D
2   ssid = Ssid ("network-D");
3   phy.Set ("ChannelSettings", StringValue ("{48, 0, BAND_5GHZ, 0}"));
4   mac.SetType ("ns3::StaWifiMac",
5                "Ssid", SsidValue (ssid));
6
7   staDeviceD = wifi.Install (phy, mac, wifiStaNodes.Get (3));
8
9   // Enable A-MPDU with a smaller size than the default one and
10  // enable A-MSDU with the smallest maximum size allowed by the standard (3839 bytes)
11  dev = wifiStaNodes.Get (3)->GetDevice (0);
12  wifi_dev = DynamicCast<WifiNetDevice> (dev);
13  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmpduSize", UintegerValue (32768));
14  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmsduSize", UintegerValue (3839));
15
16  mac.SetType ("ns3::ApWifiMac",
17               "Ssid", SsidValue (ssid),
18               "EnableBeaconJitter", BooleanValue (false));
19  apDeviceD = wifi.Install (phy, mac, wifiApNodes.Get (3));
20
21  // Enable A-MPDU with a smaller size than the default one and
22  // enable A-MSDU with the smallest maximum size allowed by the standard (3839 bytes)
23  dev = wifiApNodes.Get (3)->GetDevice (0);
24  wifi_dev = DynamicCast<WifiNetDevice> (dev);
25  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmpduSize", UintegerValue (32768));
26  wifi_dev->GetMac ()->SetAttribute ("BE_MaxAmsduSize", UintegerValue (3839));
```

Listing 5: Konfiguracja sieci D

Po ustawieniu adresacji, ns-3 ustawia we fragmencie widocznym w Listingu 6 model mobilności jako *ConstantPositionMobilityLevel*, który nie zakłada poruszania się transmitujących węzłów. Ustawia także rozmieszczenie stacji w jednej płaszczyźnie.

```
1  // Setting mobility model
2   MobilityHelper mobility;
3   Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
4   mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
5
6   // Set position for APs
7   positionAlloc->Add (Vector (0.0, 0.0, 0.0));
```

```
  positionAlloc->Add (Vector (10.0, 0.0, 0.0));
  positionAlloc->Add (Vector (20.0, 0.0, 0.0));
  positionAlloc->Add (Vector (30.0, 0.0, 0.0));
  // Set position for STAs
  positionAlloc->Add (Vector (distance, 0.0, 0.0));
  positionAlloc->Add (Vector (10 + distance, 0.0, 0.0));
  positionAlloc->Add (Vector (20 + distance, 0.0, 0.0));
  positionAlloc->Add (Vector (30 + distance, 0.0, 0.0));
```

Listing 6: Ustawianie modelu mobilności oraz rozmieszczenia stacji

Symulator ns-3 konfiguruje również sposób generowania pakietów w symulacji. Konfigurowane parametry to częstotliwość nadawania pakietów, protokół transportowy (UDP), przedział czasu w którym nadawane są pakiety oraz maksymalną liczbę wysłanych pakietów. Konfigurację dwóch (*Network A* oraz *Network B*) z czterech sieci prezentowanych w symulacji przedstawiono w Listingu 7.

```
// Setting applications
 uint16_t port = 9;
 UdpServerHelper serverA (port);
 ApplicationContainer serverAppA = serverA.Install (wifiStaNodes.Get (0));
 serverAppA.Start (Seconds (0.0));
 serverAppA.Stop (Seconds (simulationTime + 1));

 UdpClientHelper clientA (StaInterfaceA.GetAddress (0), port);
 clientA.SetAttribute ("MaxPackets", UintegerValue (4294967295u));
 clientA.SetAttribute ("Interval", TimeValue (Time ("0.0001"))); //packets/s
 clientA.SetAttribute ("PacketSize", UintegerValue (payloadSize));

 ApplicationContainer clientAppA = clientA.Install (wifiApNodes.Get (0));
 clientAppA.Start (Seconds (1.0));
 clientAppA.Stop (Seconds (simulationTime + 1));

 UdpServerHelper serverB (port);
 ApplicationContainer serverAppB = serverB.Install (wifiStaNodes.Get (1));
 serverAppB.Start (Seconds (0.0));
 serverAppB.Stop (Seconds (simulationTime + 1));

 UdpClientHelper clientB (StaInterfaceB.GetAddress (0), port);
 clientB.SetAttribute ("MaxPackets", UintegerValue (4294967295u));
 clientB.SetAttribute ("Interval", TimeValue (Time ("0.0001"))); //packets/s
 clientB.SetAttribute ("PacketSize", UintegerValue (payloadSize));
```

Listing 7: Konfiguracja ustawień generowanego ruchu

Fragment programu zaprezentowany w Listingu 9 prezentuje sposób wyliczenia przepływności dla każdego z Access Pointów jako liczbę odebranych bajtów pomnożoną przez rozmiar pakietu w bitach podzielonąy przez czas symulacji. Wyniki prezentowane są w jednostce [Mb/s].

```
double throughput = totalPacketsThroughA * payloadSize * 8 / (simulationTime * 1000000.0);
 std::cout << "Throughput with default configuration (A-MPDU aggregation enabled, 65kB): " << throughput
   << " Mbit/s" << '\n';
 if (verifyResults && (throughput < 58.5 || throughput > 59.5))
   {
     NS_LOG_ERROR ("Obtained throughput " << throughput << " is not in the expected boundaries!");
     exit (1);
   }
 throughput = totalPacketsThroughB * payloadSize * 8 / (simulationTime * 1000000.0);
 std::cout << "Throughput with aggregation disabled: " << throughput << " Mbit/s" << '\n';
 if (verifyResults && (throughput < 30 || throughput > 31))
   {
     NS_LOG_ERROR ("Obtained throughput " << throughput << " is not in the expected boundaries!");
     exit (1);
   }
 throughput = totalPacketsThroughC * payloadSize * 8 / (simulationTime * 1000000.0);
 std::cout << "Throughput with A-MPDU disabled and A-MSDU enabled (8kB): " << throughput << " Mbit/s" <<
   '\n';
 if (verifyResults && (throughput < 51 || throughput > 52))
   {
     NS_LOG_ERROR ("Obtained throughput " << throughput << " is not in the expected boundaries!");
     exit (1);
   }
 throughput = totalPacketsThroughD * payloadSize * 8 / (simulationTime * 1000000.0);
 std::cout << "Throughput with A-MPDU enabled (32kB) and A-MSDU enabled (4kB): " << throughput << " Mbit/
   s" << '\n';
 if (verifyResults && (throughput < 58 || throughput > 59))
   {
     NS_LOG_ERROR ("Obtained throughput " << throughput << " is not in the expected boundaries!");
     exit (1);
   }
```

Listing 8: Obliczenie poziomu przepływności oraz prezentacja wyników symulacji

1.2 Cel symulacji

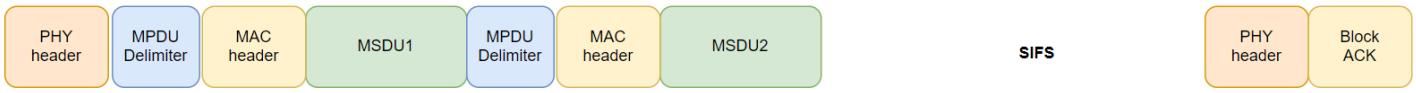
Program *wifi-aggregator.cc* zawiera kod źródłowy realizujący symulację bezprzewodowego połączenie pomiędzy stacją bazowymi oraz punktem dostępowym, które zapewnia funkcjonalność agregacji ramek. Do analizy będą użyte dwa poniższe sposoby agregacji

- A-MPDU (Aggregate MAC Protocol Data Uni) – sposób agregacji ramek otrzymanych przez warstwę fizyczną od warstwy łącza danych, wszystkie MPDU posiadają nagłówki MAC oraz są łączone w jeden dłuższy MPDU dopiero na warstwie fizycznej
- A-MSDU (Aggregate MAC Service Data Unit) – sposób agregacji ramek otrzymanych przez warstwę łącza danych od warstwy sieciowej, wszystkie MSDU są łączony w jeden MPDU posiadający nagłówek MAC

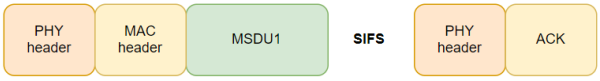
Uruchomiona symulacja pozwala na przeanalizowanie 4 scenariuszy z zastosowanymi różnymi opcjami agregacji ramek

1. A-MSDU wyłączone, A-MPDU włączone z maksymalnym rozmiarem 56 kB
2. A-MSDU wyłączone, A-MPDU wyłączone
3. A-MSDU włączone z maksymalnym rozmiarem 8 kB, A-MPDU wyłączone
4. A-MSDU włączone z maksymalnym rozmiarem 4 kB, A-MPDU włączone z maksymalnym rozmiarem 32 kB

Uruchomiona symulacja pozwala na przeanalizowanie 4 scenariuszy z zastosowanymi różnymi opcjami agregacji ramek, których schematy przedstawiono na poniższych rysunkach:

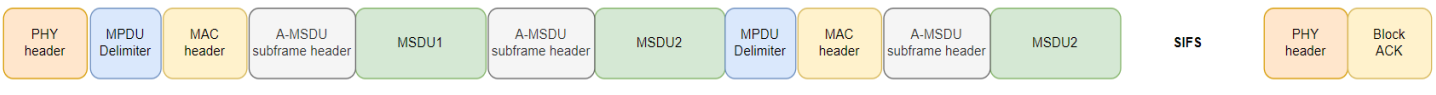


Rysunek 1: Włączona agregacja A-MPDU



Rysunek 2: Wyłączona agregacja A-MPDU oraz A-MSDU

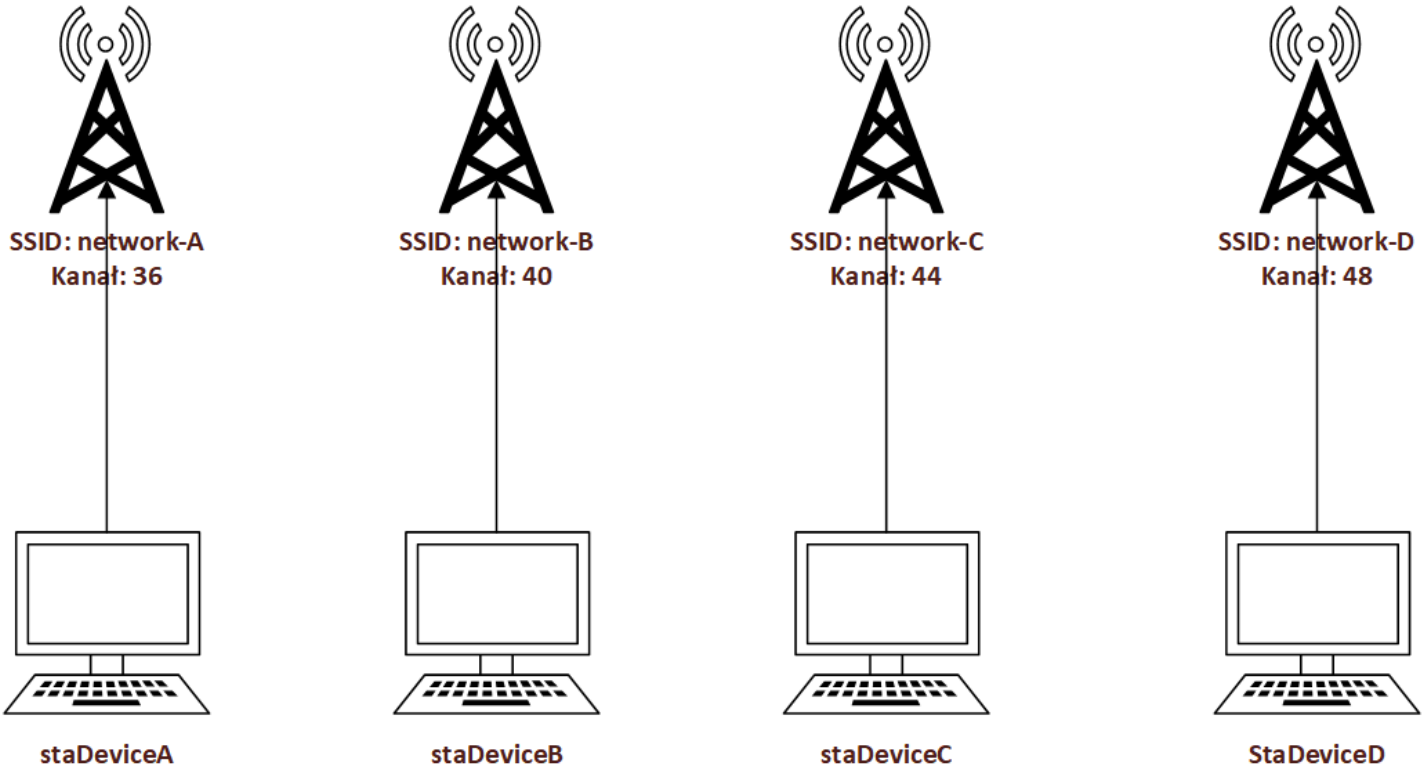
Rysunek 3: Włączona agregacja A-MSDU



Rysunek 4: Włączona agregacja A-MPDU oraz A-MSDU

1.3 Topologia badanej sieci

Symulowana architektura posiada cztery działające niezależnie sieci, z których każda realizuje jeden ze wcześniej przypadków. Połączenia są zestawiane pomiędzy jednym punktem dostępowym i jedną stacją bazową. Cała architektura wraz z adresacją jest widoczna na rysunku 5



Rysunek 5: Topologia badanej sieci

2 Model systemu

W tym rozdziale zaprezentowano model systemu uwzględniając ustawienia symulacji, jego parametry wejściowe oraz wyjściowe, a także uproszczenia zastosowane w symulacji.

2.1 Ustawienia symulacji

Oprócz parametrów agregacji ramek dla każdej sieci przedstawionych w wyjaśnieniu celu symulacji istnieje jeszcze wiele innych, które mają wpływ na symulację. Wszystkie transmisje realizowane są przy pomocy poniższych parametrów:

- Standard IEEE 802.11n
- Liczba sieci 4
- Liczba węzłów dla każdej sieci (Nodes) 2
- Pasmo 5 GHz
- Użyte kanały odpowiednie kolejno dla sieci to 36, 40, 44, 48
- Dane wysyłane są protokołem UDP
- Interwał pomiędzy nadawanymi pakietami to 0.0001 sekundy
- Domyślne wartości modyfikowalnych parametrów:
  - Czas symulacji 10 sekund
  - Wyłączony mechanizm RTS
  - Wyłączony mechanizm generowania pliku .pcap
  - Wyłączona weryfikacja wyników symulacji

2.2 Parametry wejściowe symulacji

- `payloadSize` - ilość danych wysłanych w sieci wyrażona w bajtach
- `enableRts` - opcja umozliwiająca użycie mechanizmu RTS/CTS bazującego na użyciu dwóch ramek odpowiednio *request to send* i *clear to send* dzięki którym dochodzi do komunikacji między stacją i punktem dostępowym
- `simulationTime` - czas jaki trwa symulacja, wartość domyślna 10 sekund.
- `distance` - odległość między stacją i punktem dostępowym w metach, wartość domyślna 5
- `enablePcap` - opcja zapisująca pliki w rozszerzeniu .pcap (możliwe do analizy w wiresharku)
- `verifyResults` - opcja umożliwiająca sprawdzenie czy przepustowość mieści się w spodziewanych granicach

2.3 Parametry wyjściowe symulacji

Parametrami wyjściowymi symulacji są całkowite przepustowości dla czterech sieci odpowiednio: `throughputA`, `throughputB`, `throughputC`, `throughputD`. Przepustowość jest wynikiem pomnożenie przesalanych pakietów przez rozmiar danych i podzielone przez czas symulacji (aby uzyskać wynik w jednostce Mbit/s).

```
double throughput = totalPacketsThroughA * payloadSize * 8 / (simulationTime * 1000000.0);
throughput = totalPacketsThroughB * payloadSize * 8 / (simulationTime * 1000000.0);
throughput = totalPacketsThroughC * payloadSize * 8 / (simulationTime * 1000000.0);
throughput = totalPacketsThroughD * payloadSize * 8 / (simulationTime * 1000000.0);
```

Listing 9: Sposób obliczania przepustowości dla czterech sieci

2.4 Uproszczenia symulacji

Głównym uproszczeniem zawartym w powyższej symulacji jest wprowadzenie czterech niezależnie działających sieci co zakłada brak interferencji pojawiających się z innych sygnałów. W dodatku nie jest brane pod uwagę środowisko takie jak ukształtowanie tereny, grubość i występowanie ścian. Stacje się nie zmieniają położenia w wyniku czego sygnał stabilny oraz nie wprowadzają opóźniania spowodowanego obliczeniami na sprzęcie. Kolejnym uproszczeniem jest użycie tylko jednej stacji w sieci.

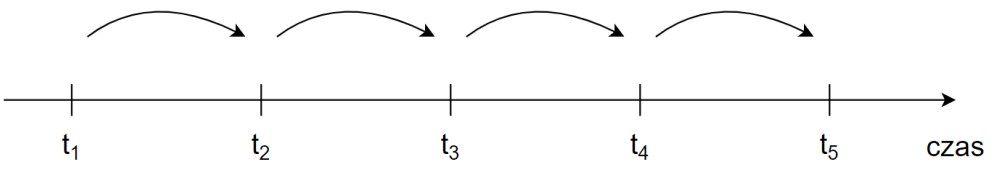
3 Symulacja zdarzeń dyskretnych

Symulacja zdarzeń dyskretnych określa sposób przeprowadzania symulacji przez ns-3, w którym zakładamy, że przebieg symulacji modeluje się w postaci sekwencji następujących po sobie zdarzeń dyskretnych, pomiędzy którymi nie zachodzi żadna zmiana w systemie. W tym rozdziale przedstawiono sekwencję pięć zdarzeń dyskretnych zachodzących w symulacji oraz porównano rzeczywisty czas symulacji z czasem potrzebnym na wykonanie całej symulacji.

3.1 Sekwencja zdarzeń symulacji

W ramach badanej symulacji zachodzą sekwencje zdarzeń dyskretnych, które mogą być przedstawione w sposób zaprezentowany na Rysunku 6. Wyszczególnionym na rysunku chwilach symulacji możemy przypisać następujące zdarzenia:

t_1 - oczekiwanie backoff, t_2 - wysłanie ramki danych, t_3 - odebranie ramki danych , t_4 - oczekiwanie SIFS, t_5 - wysłanie ACK



Rysunek 6: Symulacja zdarzeń dyskretnych

3.2 Analiza czasu wykonania symulacji

Analizę czasu wykonania symulacji przeprowadzono z wykorzystaniem biblioteki chrono mierząc czas przed oraz po wykonaniu symulacji, co przedstawiono w Listingu 10.

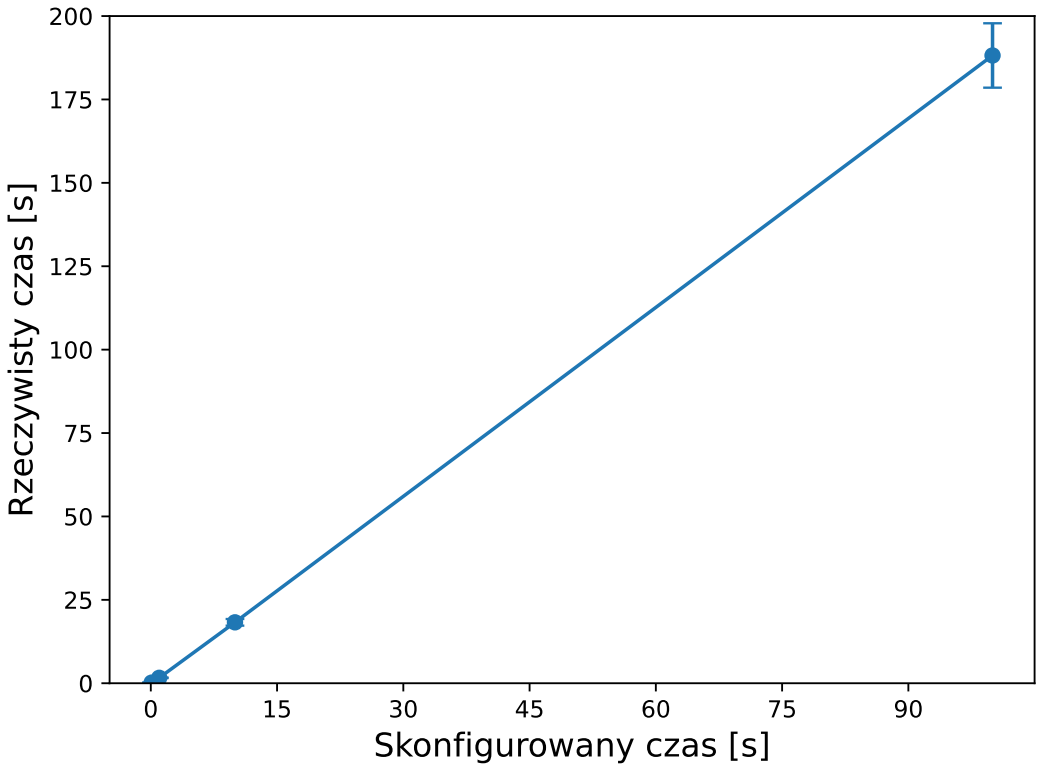
```
#include <chrono>
...
auto start = std::chrono::high_resolution_clock::now();

Simulator::Run ();

auto finish = std::chrono::high_resolution_clock::now();
std::chrono::duration<double> elapsed = finish - start;
std::cout << "Elapsed time: " << elapsed.count() << " s\n\n";
```

Listing 10: Konfiguracja pomiaru rzeczywistego czasu symulacji

Symulacje powtórzono 10 razy dla każdego z badanych czasów, a wyniki przedstawiono z uwzględnieniem poziomu ufności równym 95 procent zgodnie z rozkładem t-Studenta. Wyniki podsumowano na Wykresie 7.



Wykres 7: Analiza wpływu skonfigurowanego czasu symulacji na rzeczywisty czas wykonania

Przeprowadzona analiza wskazuje na to, że realny czas wykonania symulacji jest w każdym przypadku większy od ustawionego czasu symulacji ponieważ ns-3 wykorzystuje czas również na konfigurację symulacji (np. zaplanowanie zdarzeń, konfigurację transmitujących stacji), a ustawiony czas dotyczy wyłącznie czasu zdarzeń dyskretnych związanych z symulacją (np. wysłanie ramki danych, oczekiwanie w procedurze backoff). Symulacja widoczna na wykresie 7 wskazuje również na zależność liniową pomiędzy badanymi wielkościami (czasem skonfigurowanym oraz czasem rzeczywistym). Pozwala to stwierdzić, że nadmiarowy czas związany z przeprowadzeniem symulacji rośnie wraz z czasem ustawianym jako czas symulacji.

4 Parametry symulacji

Aby móc wyciągnąć odpowiednie wnioski z symulacji należy poprawnie dobrać jej parametry. Dodatkowo im więcej razy zostanie ona wykonana tym pozwoli to uzyskać wyniki bardziej zbliżone do rzeczywistych wartości.

4.1 Ustawienia czasu symulacji

Program symuluje przekazaną mu konfigurację przez określony czas i z jego pomocą oblicza wyniki końcowe. Ważne jest zatem aby był on dostosowany do otrzymania jak najbardziej wiarygodnych wyników i dodatkowo pozwolił zakończyć symulację w niedługim okresie. Dla domyślnych parametrów wejściowych, zmiennej wartości czasu oraz RngRun równego 596 symulacja zwróciła poniższe wyniki

simulationTime [s]	A-MPDU (65kB) [Mbit/s]	Brak agregacji [Mbit/s]	A-MSDU (8kB) [Mbit/s]	A-MPDU (32kB) i A-MSDU (4kB) [Mbit/s]
10	59,5736	30,7095	51,7414	58,9848
15	59,5873	30,7071	51,7618	59,0056
20	59,5907	30,7089	51,7691	59,0166
25	59,5974	30,7076	51,7805	59,0241
30	59,6034	30,6985	51,7842	59,0299
35	59,6037	30,697	51,7902	59,0314
40	59,6069	30,6989	51,7873	59,0313
45	59,6075	30,7063	51,7864	59,0289
50	59,6068	30,7118	51,7892	59,0298
55	59,6075	30,7118	51,7872	59,0299
60	59,6064	30,7138	51,7865	59,0303

Tabela 8: Wyniki porównania wydajności zależnej od czasu symulacji

Kiedy zostaną one umieszczone na wykresach to dla każdego przypadku wykażą, że im dłuższy czas symulacji tym mniejsze wahania pojawiają się w wartościach wydajności. Dodatkowo na poniższych wykresach, w których są włączone mechanizmy agregacji widać, że im dłuższy czas symulacji tym wydajność jest większa co znaczy o poprawie ich skuteczności. Od 50 sekundy można zauważyć, że wahania są znacząco mniejsze, więc najlepszym czasem do przeprowadzenia analizy jest 60 sekund.

4.2 Liczba niezależnych symulacji

Do uzyskania jak najlepszych wyników należy przeprowadzić wiele symulacji, żeby później wyciągnąć średnią. Do każdej z nich należy wprowadzić inne ziarno, które zapewni, że symulacje będą niezależne. Ich liczba powinna być jak największa, ale łączny czas ich wykonania powinien pozostać racjonalny. Wykonanie 3 równolegle działających symulacji zajmuje komputerowi około 2,5 minuty, co przy dużej ilości powtórzeń staje się kłopotliwe. Z poniższej tabeli można odczytać przedział ufności na poziomie 95%, który pokazuje, że wartości większe bądź mniejsze od średniej o 0,005 dla 3 symulacji są akceptowalne.

Run	Throughput	Average	Deviation	Confidence interval
1	59,7026	59,7026	0	0
2	59,7040	59,7033	0,0010	0,0089
3	59,7066	59,7044	0,0020	0,0050

Tabela 9: Przedziały ufności zależne od liczby symulacji.

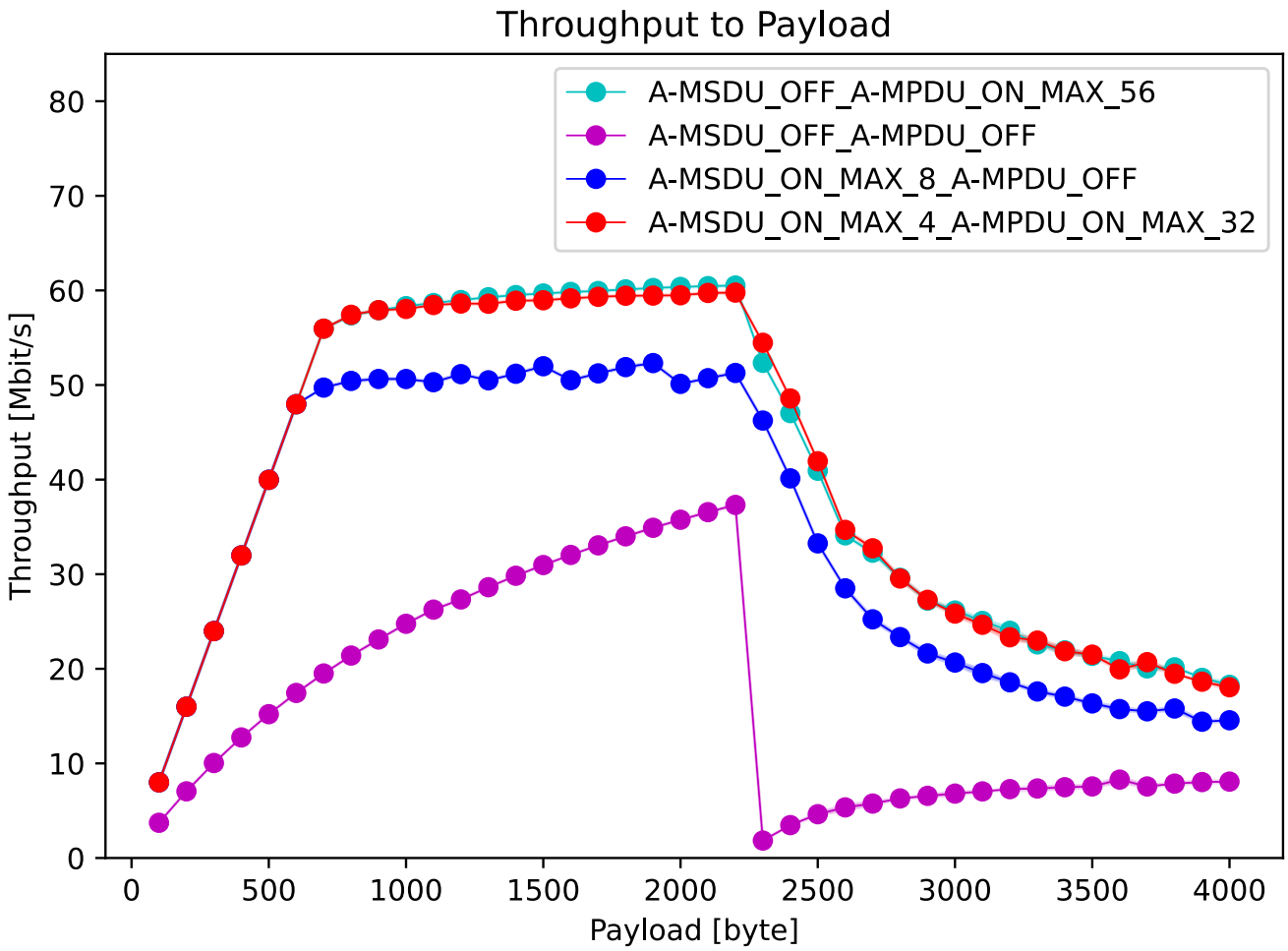
4.3 Czas rozruchu

Jest to czas, w którym stacje się asocjują do ich punktów dostępowych. Trwa on 1 sekundę i w jej trakcie nie są wysyłane żadne dane pomiarowe. Po jej upłynięciu, kiedy wszystkie połączenia są ustanowione, rozpoczyna się transfer danych wraz z ich pomiarem.

4.4 Wybrane parametry pomiarowe

Najważniejszym parametrem pomiarowym jest Throughput, który przekłada się na wydajność sieci. Będzie on porównywany zależnie od payloadSize oraz distance. Został on wybrany ze względu na badanie mechanizmów agregacji ramek, których efektywność przekłada się bezpośrednio na niego.

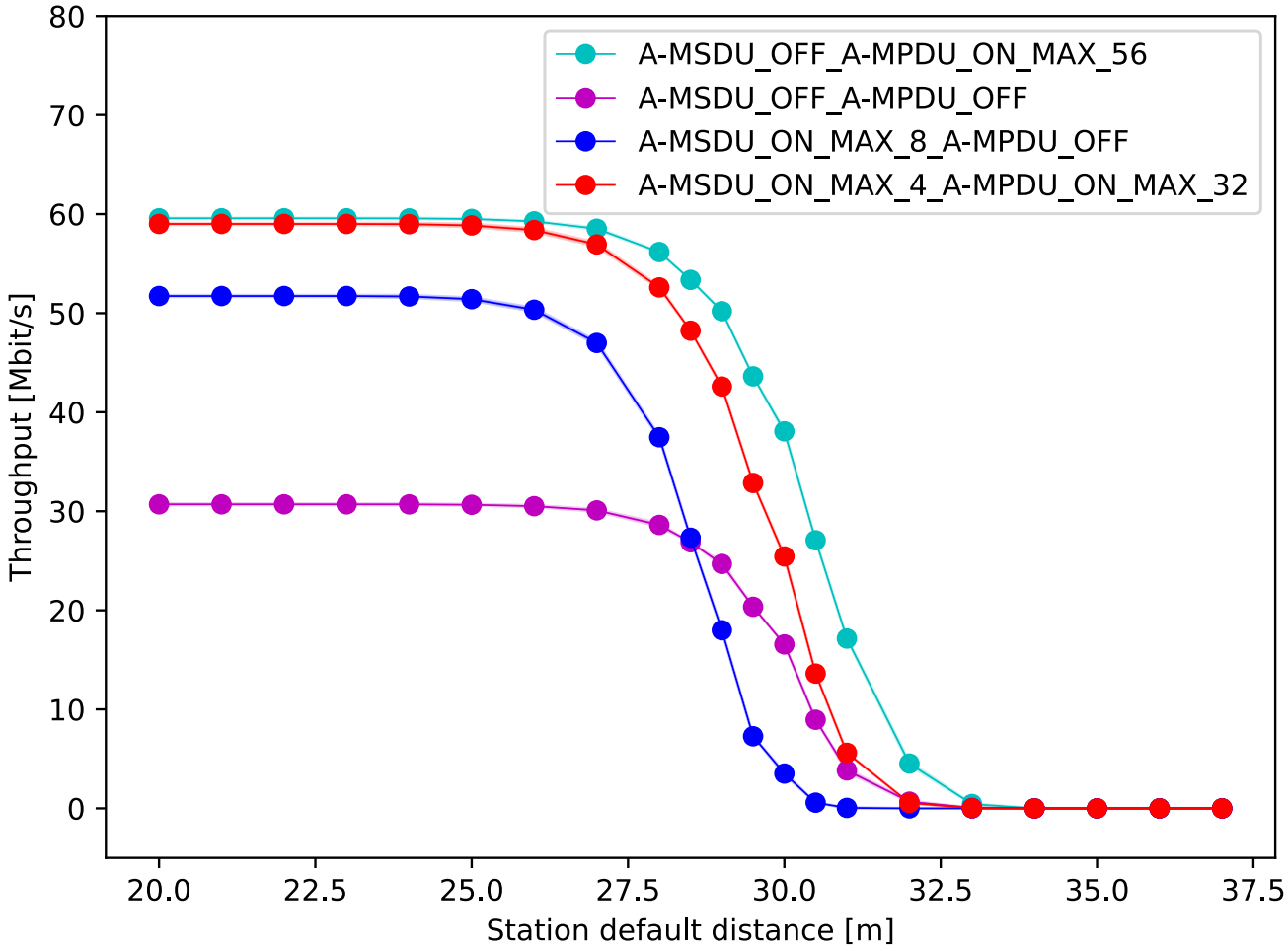
5 Wyniki symulacji



Wykres 10: Przepustowość w zależności od ilości danych.

Payload size	A_MSDU_OFF_A_MPDU_ON_MAX_56	A_MSDU_OFF_A_MPDU_ON_MAX_56	A_MSDU_ON_MAX_8_A_MPDU_OFF	A_MSDU_ON_MAX_4_A_MPDU_ON_MAX_32
100	0.005943	0.010452	0.003036	0.004958
200	0.012092	0.020905	0.006066	0.010425
300	0.018432	0.027014	0.008892	0.016023
400	0.023804	0.026866	0.012471	0.020609
500	0.029925	0.034626	0.015991	0.02689
600	0.03657	0.038868	0.016823	0.031083
700	0.039546	0.043512	0.084089	0.038697
800	0.059505	0.054176	0.080395	0.013095
900	0.05655	0.065842	0.096169	0.009953
1000	0.060843	0.054726	0.091531	0.018465
1100	0.058733	0.05689	0.087151	0.017556
1200	0.059016	0.057241	0.087071	0.016705
1300	0.060623	0.066945	0.07917	0.018203
1400	0.060984	0.059449	0.084837	0.017076
1500	0.062054	0.055476	0.076602	0.020938
1600	0.061256	0.048571	0.083456	0.019352
1700	0.057467	0.049265	0.087903	0.014101
1800	0.062872	0.06098	0.093121	0.016452
1900	0.062064	0.062815	0.084124	0.018879
2000	0.063095	0.070692	0.08375	0.017923
2100	0.062265	0.072536	0.088878	0.014701
2200	0.061234	0.068628	0.086089	0.015847
2300	0.057891	0.05337	0.105213	0.012112
2400	0.059764	0.026556	0.097882	0.054176
2500	0.067819	0.139319	0.092014	0.07766
2600	0.036144	0.278843	0.089084	0.026496
2700	0.186859	0.171144	0.204933	0.245457
2800	0.186049	0.061963	0.131607	0.139114
2900	0.160125	0.118718	0.252581	0.345987
3000	0.225269	0.198593	0.204159	0.29673
3100	0.255315	0.031008	0.361983	0.137687
3200	0.36762	0.088118	0.118144	0.105261
3300	0.230431	0.170301	0.176056	0.001678
3400	0.332185	0.185332	0.15863	0.007139
3500	0.129579	0.121744	0.138123	0.027174
3600	0.159212	0.26988	0.220849	0.057711
3700	0.329678	0.141683	0.041347	0.121599
3800	0.076022	0.07391	0.241204	0.119669
3900	0.055078	0.152996	0.102455	0.055117
4000	0.25692	0.053319	0.118905	0.13781

Tabela 11: Przedziały ufności przepustowości w zależności od ilości danych.



Wykres 12: Przepustowość w zależności od odległości.

Distance	A_MSDU_OFF_A_MPDU_ON_MAX_56	A_MSDU_OFF_A_MPDU_ON_MAX_56	A_MSDU_ON_MAX_8_A_MPDU_OFF	A_MSDU_ON_MAX_4_A_MPDU_ON_MAX_32
20	0.035669	0.070523	0.075479	0.020222
21	0.035669	0.070523	0.075479	0.020222
22	0.035669	0.070523	0.075479	0.020222
23	0.037791	0.069415	0.083654	0.021117
24	0.04349	0.07971	0.125164	0.036138
25	0.047572	0.062456	0.149801	0.03474
26	0.048314	0.081088	0.183157	0.107871
27	0.097084	0.112042	0.178695	0.117399
28	0.059923	0.171691	0.137647	0.126745
28.5	0.01511	0.001434	0.014342	0.005737
29	0.064787	0.174437	0.105221	0.122971
29.5	0.005171	0.037946	0.002484	0.014342
30	0.186509	0.025978	0.136767	0.176869
30.5	0.014342	0.013682	0.001511	0.014342
31	0.110715	0.169338	0.042223	0.117252
32	0.195614	0.073482	0	0.035749
33	0.052683	0.008776	0	0.012751
34	0	0	0	0
35	0	0	0	0
36	0	0	0	0
37	0	0	0	0

Tabela 13: Przedziały ufności przepustowości w zależności od odległości.

6 Wnioski

Dla wykresu 10 można zauważyć, że najmniejsza przepustowość osiągana jest dla scenariusza, który nie posiada włączonych żadnych mechanizmów agregacji. Natomiast dla pozostałych scenariuszy do 500 bajtów wzrost jej był bardzo podobny. Zwiększając dalej przepustowość widać, że A-MSDU dla 8 kB radzi sobie najgorzej ze wszystkich kombinacji, a scenariusze A-MPDU dla 56 kB oraz A-MSDU dla 4 KB z A-MPDU dla 32 kB są bardzo zbliżone do siebie i górują ponad resztą.

Na wykresie 12 widać kolejno spadek przepustowości w scenariuszach: A-MSDU dla 8 kB, brak mechanizmów agregacji, A-MSDU dla 4 KB z A-MPDU dla 32 kB, A-MPDU dla 56 kB. Przedstawie je to w odwrotnej skuteczności względem zwiększającego się dystansu. Warto zaznaczyć, że po przekroczeniu 34 m wszystkie scenariusze uzyskują przepustowość zbliżoną do 0 co uniemożliwia ich dalszą analizę.

Podsumowując, konfiguracją, która uzyskuje jednocześnie najlepszy stosunek przepustowość do rozmiaru ramek i dystansu jest A-MPDU dla 56 kB. Jest to spowodowane maksymalnym rozmiarem ramki, która w porównaniu do pozostałych scenariuszy w tym jest największa.

7 Podsumowanie

Projekt dotyczący analizy scenariuszy agregacji ramek w sieciach standardu 802.11n pozwolił na przeprowadzenie szeregu symulacji, umożliwiajacych przyjrzeniu się zarówno ogólnie procesowi symulacji w sieciach teleinformatycznych jak i szczegółowo mechanizmowi agregacji ramek w standardzie WiFi.

Wartościowym punktem wyjścia do podobnych badań w przyszłości mogłaby być symulacja badająca sprawiedliwość w dostępie do kanału dla stacji działających w sieci 802.11n, korzystających z różnych schematów agregacji. Innym aspektem badawczym wydaje się być symulacja uwzględniająca współistnienie stacji implementujących różne rodzaje agregacji z dodatkowym uwzględnieniem wpływu zakłóceń powstających w kanale radiowym.