



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Elettronica
Curriculum: Sistemi Elettronici e Nanotecnologie

**IMPLEMENTAZIONE DI OPEN5GS E SRSRAN SU
PIATTAFORMA SDR: STUDIO DELLE FUNZIONALITA' E
VERIFICA DELLE PRESTAZIONI**

Implementation of Open5GS and srsRAN on the SDR platform:
Functional study and performance verification

Relatore:
Prof. Ennio Gambi

Candidato:
Lorenzo Pollonara
Mat. 1108622

Correlatori:
Ing. Adelmo De Santis
ATLC Wireless team manager
Rodolfo Magnaterra

Anno Accademico 2023-2024

Indice

Introduzione	2
1 Stato dell'arte del 5G	5
1.1 Introduzione	5
1.2 Evoluzione delle Tecnologie Mobili	6
1.3 Architettura e Componenti del 5G	8
1.3.1 Radio Access Network (RAN)	8
1.3.2 Core Network	9
1.3.3 User Equipment (UE)	12
1.4 5G Standalone e Non-Standalone	12
2 Network Slicing	14
2.1 Introduzione al Network Slicing	14
2.2 Funzioni di rete	16
2.2.1 UPF	17
2.3 DNN e 5QI	18
2.3.1 Differenze tra DNN e APN	18
2.3.2 QoS	19
2.3.3 5QI	22
2.4 SST e SD	25
2.4.1 Valori standard dell'SST	26

3	Software utilizzati	28
3.1	Open5GS	28
3.2	srsRAN	30
3.3	Installazione Open5GS	32
3.4	Installazione srsRAN	33
4	Realizzazione del test set ed implementazione del gNb	34
4.1	Configurazione rete 5G per test latenza	34
4.1.1	Configurazione della SIM	37
4.1.2	Configurazione del TestCenter	39
4.1.3	Configurazione del PC	43
4.1.4	Configurazione ATOS	51
4.2	Configurazione rete 5G per slicing	53
4.2.1	Configurazione delle SIM	54
4.2.2	Configurazione del TestCenter	57
4.2.3	Configurazione del PC	59
4.2.4	Configurazioni ATOS	70
5	Test latenza	72
5.1	Modalità TDD	72
5.2	Modalità FDD	78
5.3	Aumento della banda	83
5.4	Sintesi	85
6	Test slicing	88
6.1	Risultati	89
6.1.1	Carico voce fisso	89
6.1.2	Carico voce variabile	93
6.2	Sintesi	94
7	Implementazione traffico voce reale	96

7.1	Test set	97
7.2	Risultati ottenuti	100
	Conclusioni	104

Elenco delle figure

1.1	Core Network [8].	9
1.2	5G SA e NSA [10].	12
2.1	Esempio di slicing [14].	15
2.2	Tabella valori 5QI [8].	24
2.3	SST e SD [8].	25
2.4	Valori standard SST [8].	26
3.1	Architettura di rete di Open5GS [19].	29
4.1	Interfaccia WebUI per configurare la SIM.	39
4.2	Banco test di latenza.	39
4.3	Assegnazione IP.	40
4.4	Assegnazione dimensione frame.	41
4.5	Assegnazione carico.	42
4.6	Impostazione IPv4 Modifier.	43
4.7	Interfaccia WebUI per configurare SIM internet.	56
4.8	Interfaccia WebUI per configurare SIM voce.	57
4.9	Banco test slicing.	58
4.10	Modifiche al file 99-open5gs.netdev.	63
4.11	Modifiche al file 99-open5gs.network.	64
4.12	Modifiche al file 98-open5gs.netdev.	64
4.13	Modifiche al file 98-open5gs.network.	65

5.1	Confronto latenza in banda n78, MCS PDSCH 1, MCS PUSCH	
	0.	74
5.2	Statistiche connessione downlink e uplink	74
5.3	Confronto latenza in banda n78, MCS PDSCH 9, MCS PUSCH	
	8.	75
5.4	Statistiche connessione downlink e uplink	75
5.5	Confronto latenza in banda n78, MCS PDSCH 21, MCS PUSCH	
	20.	76
5.6	Statistiche connessione downlink e uplink	76
5.7	Confronto latenza in banda n78, MCS PDSCH 27, MCS PUSCH	
	26.	77
5.8	Statistiche connessione downlink e uplink	77
5.9	Confronto latenza in banda n7, MCS PDSCH 1, MCS PUSCH 0.	79
5.10	Statistiche connessione downlink e uplink	79
5.11	Confronto latenza in banda n7, MCS PDSCH 9, MCS PUSCH 8.	80
5.12	Statistiche connessione downlink e uplink	80
5.13	Confronto latenza in banda n7, MCS PDSCH 21, MCS PUSCH	
	20.	81
5.14	Statistiche connessione downlink e uplink	81
5.15	Confronto latenza in banda n7, MCS PDSCH 27, MCS PUSCH	
	26.	82
5.16	Statistiche connessione downlink e uplink	82
5.17	Confronto latenza in downlink	84
5.18	Confronto latenza in uplink	84
6.1	Test latenza e throughput con slicing in banda 7, carico dati	
	1Mbps	90
6.2	Test latenza e throughput con slicing in banda 7, carico dati	
	1Mbps, MCS libero	91

6.3	Test latenza e throughput con slicing in banda 78, carico dati 1Mbps, MCS libero	92
6.4	Test latenza e throughput con slicing in banda 7, carico dati variabile, MCS libero	93
7.1	Test set traffico voce reale	100
7.2	Traffico RTP analizzato con Wireshark	102
7.3	Traffico UDP analizzato con Wireshark	102
7.4	Traffico UDP analizzato con Wireshark	102

Sommario

Questa tesi si propone di condurre un'analisi dettagliata delle prestazioni della radio X310 tramite l'implementazione dei due *software* Open5GS e srsRAN. Si sono confrontate le prestazioni con una radio con hardware meno performante, la B210. L'obiettivo principale è identificare i vantaggi e le limitazioni di ciascuna radio in vari scenari operativi, con particolare attenzione alla latenza, al *throughput* e all'affidabilità delle comunicazioni.

La seconda parte della tesi si concentra sullo studio approfondito e l'implementazione dello *slicing*, una tecnologia fondamentale per il 5G che permette la creazione di sottoreti virtuali indipendenti all'interno della stessa infrastruttura fisica. Lo *slicing* consente una gestione ottimizzata del traffico di rete, assegnando priorità differenti ai vari tipi di traffico. In questo caso si è andati a esplorare l'efficacia della tecnologia, implementandola e misurandone le prestazioni.

Introduzione

Nel panorama delle telecomunicazioni moderne il 5G promette di rivoluzionare il modo in cui le reti mobili operano e supportano una vasta gamma di applicazioni e servizi. Questa quinta generazione di reti mobili non solo aumenta significativamente la velocità di trasmissione dei dati, ma riduce anche drasticamente la latenza e aumenta la capacità di connessione, consentendo di gestire un numero maggiore di dispositivi connessi simultaneamente. Queste caratteristiche rendono il 5G una piattaforma abilitante per tecnologie emergenti come l'*Internet of Things* (IoT), la realtà aumentata (AR) e virtuale (VR), i veicoli autonomi, le *smart cities* e molto altro ancora.

Una delle tecnologie chiave che permettono al 5G di raggiungere questi obiettivi ambiziosi è lo *slicing* di rete. Lo *slicing* di rete è un'innovativa tecnica che consente di suddividere una singola infrastruttura fisica di rete in più sottoreti virtuali indipendenti, ognuna delle quali può essere ottimizzata per requisiti specifici di prestazione e servizio. Questa capacità di creare "slice" di rete distinti permette di garantire livelli di servizio (QoS) personalizzati per diversi tipi di traffico, migliorando l'efficienza e la flessibilità della gestione delle risorse di rete.

L'obiettivo del lavoro di tesi consiste quindi nell'implementazione di alcune funzionalità previste dal 5G, allo scopo di verificare il reale raggiungimento di questi obiettivi.

La tesi è organizzata come segue:

- **Capitolo 1:** Questo capitolo fornisce un'introduzione generale al 5G, esplorando le sue caratteristiche principali, l'evoluzione dalle generazioni precedenti di reti mobili, l'architettura e i vantaggi.
- **Capitolo 2:** In questo capitolo viene approfondito il concetto di network *slicing*, una tecnologia fondamentale per il 5G che permette la creazione di sottoreti virtuali indipendenti. Verranno esaminati la sua architettura, i benefici, i casi d'uso pratici e le principali sfide tecniche.
- **Capitolo 3:** Questo capitolo descrive i vari software impiegati durante la ricerca e la sperimentazione. Vengono illustrate le funzionalità di ciascun software e le ragioni della loro scelta.
- **Capitolo 4:** In questo capitolo viene descritta la configurazione del banco di prova utilizzato per i test. Include dettagli sull'hardware, la topologia di rete e le specifiche configurazioni necessarie per eseguire i test sul 5G e sul network *slicing*.
- **Capitolo 5:** Questo capitolo presenta e analizza i risultati dei test sulla latenza delle comunicazioni 5G. Verranno discussi i tempi di risposta in diversi scenari operativi e le prestazioni relative delle radio X310 e B210.
- **Capitolo 6:** In questo capitolo vengono presentati i risultati della sperimentazione sul network *slicing*, analizzando le prestazioni in termini di latenza, *throughput* e affidabilità delle diverse *slice* di rete.
- **Capitolo 7:** Questo capitolo esplora i risultati ottenuti dal test del traffico voce reale, utilizzando telefoni e applicazioni di comunicazione per verificare l'efficacia del network *slicing* nel gestire il traffico vocale.

- **Capitolo 8:** Il capitolo finale riassume i risultati chiave della ricerca, presenta le conclusioni principali.

Questa ricerca contribuisce in modo significativo alla comprensione delle potenzialità e delle sfide associate all'implementazione dello *slicing* di rete nel contesto del 5G. I risultati ottenuti forniscono preziose indicazioni per la progettazione e la gestione delle reti 5G, evidenziando le aree in cui lo *slicing* di rete può offrire i maggiori benefici e le strategie per superare le eventuali limitazioni. Questi risultati sono cruciali per l'adozione e l'ottimizzazione delle tecnologie di *slicing* di rete, migliorando così l'efficienza e l'affidabilità delle reti di comunicazione del futuro.

Capitolo 1

Stato dell'arte del 5G

1.1 Introduzione

Il 5G, la quinta generazione delle tecnologie di comunicazione mobile, rappresenta una rivoluzione nel campo delle telecomunicazioni, promettendo di trasformare radicalmente non solo il modo in cui si comunica, ma anche il modo in cui si interagisce con il mondo digitale. Dal suo concetto iniziale, il 5G ha attirato un'enorme attenzione sia da parte dell'industria che della comunità accademica, grazie alle sue potenziali applicazioni in settori come l'*Internet of Things* (IoT), la realtà aumentata e virtuale, l'Industria 4.0, e molte altre [1].

L'avvento del 5G è stato guidato dalla necessità di rispondere alle crescenti esigenze di velocità di trasmissione dati, latenza ridotta, maggiore capacità di connessione e miglior efficienza energetica. Rispetto alle generazioni precedenti, il 5G introduce nuove tecnologie come il *Massive MIMO*, il *Beamforming*, l'uso delle onde millimetriche e il *Network Slicing*, che insieme permettono di raggiungere prestazioni senza precedenti.

In questo capitolo, si fornisce una panoramica dello stato dell'arte del 5G, esaminando l'evoluzione delle tecnologie mobili che hanno portato alla sua

nascita, le architetture di rete e le componenti principali, i differenti tipi di implementazione.

Questo capitolo servirà quindi come base per comprendere le fondamenta su cui si costruisce il 5G e come questa tecnologia stia plasmando il futuro delle comunicazioni e dell'interazione digitale.

1.2 Evoluzione delle Tecnologie Mobili

La tecnologia mobile ha visto una rapida evoluzione, partendo dall'1G, che offriva solo servizi vocali analogici, al 4G, che ha introdotto la banda larga mobile con velocità di dati molto elevate. Ogni generazione ha migliorato significativamente le capacità di comunicazione, portando all'esigenza di una nuova generazione, il 5G, per affrontare le crescenti richieste di velocità, capacità e latenza.

Il passaggio al 5G è stato guidato dalla necessità di una maggiore efficienza spettrale, capacità di rete e supporto per nuove applicazioni che richiedono prestazioni di rete avanzate. Il 5G promette di rivoluzionare vari settori industriali e migliorare significativamente la qualità della vita quotidiana.

L'architettura del 5G è composta da diverse componenti chiave, tra cui la *Radio Access Network* (RAN), la *Core Network* e la gestione delle risorse. La RAN 5G utilizza tecnologie avanzate come il *Massive MIMO* (*Multiple Input Multiple Output*) e il *Beamforming* per migliorare la copertura e la capacità. Mentre le tecnologie chiave usate dal 5G sono:

- **Massive MIMO:** Questa tecnologia consente la comunicazione simultanea con un gran numero di utenti, riducendo significativamente il tempo necessario per accedere alle risorse e diminuendo così la latenza nella trasmissione delle informazioni. In altre parole, l'uso di *array* di antenne

avanzate permette una gestione più efficiente del traffico, ottimizzando i tempi di accesso e riducendo la latenza complessiva delle comunicazioni. Inoltre, la presenza di numerose antenne crea una sorta di ridondanza, permettendo al sistema di gestire meglio gli effetti negativi come interferenze e attenuazioni del segnale [2].

- **Beamforming:** Questa tecnologia permette di concentrare il segnale in direzioni specifiche, aumentando la potenza del segnale ricevuto dagli utenti. Grazie alla capacità di dirigere i segnali in modo preciso, il *Beamforming* riduce le interferenze tra diversi utenti e dispositivi. Il 5G sfrutta bande di frequenza più alte, che tendono ad avere una portata più limitata e sono più suscettibili alle interferenze. Il *Beamforming* consente di utilizzare queste bande in modo più efficiente, migliorando la capacità complessiva del sistema e permettendo una gestione ottimale delle risorse dello spettro. Inoltre con il *Beamforming*, è possibile servire più utenti contemporaneamente senza compromettere la qualità del servizio. La capacità di fornire segnali più forti e meno soggetti a interferenze contribuisce a rendere le connessioni più stabili e affidabili [3].
- **Millimeter Waves:** Le onde millimetriche operano in una gamma di frequenze molto elevata, tipicamente tra i 24 GHz e i 100 GHz. Questa ampia disponibilità di spettro consente di raggiungere velocità di trasmissione dati estremamente elevate. La maggiore larghezza di banda delle *mmWave* permette di gestire un numero molto elevato di dispositivi connessi contemporaneamente. Inoltre grazie alla loro capacità di trasportare grandi quantità di dati in modo rapido ed efficiente, contribuiscono a ridurre la latenza [4].
- **Network Slicing:** Questa tecnologia permette agli operatori di rete di creare reti virtuali dedicate per diverse applicazioni. Ogni segmento può avere caratteristiche specifiche di latenza, larghezza di banda e affida-

bilità, adattandosi perfettamente ai requisiti dei vari servizi. Grazie al *Network Slicing*, le risorse della rete possono essere allocate in modo dinamico e flessibile, ottimizzando l'utilizzo dell'infrastruttura e riducendo i costi operativi. Questo approccio consente di gestire in modo più efficiente i picchi di traffico e di garantire la qualità del servizio (QoS) per ogni *slice* [5].

- **Edge Computing:** Consente l'elaborazione dei dati più vicino agli utenti finali, ovvero vicino alla fonte di generazione, anziché inviarli a *data center* lontani. Questo riduce significativamente la latenza e quindi si migliorano le prestazioni delle applicazioni. Gli utenti possono sperimentare una maggiore velocità e una riduzione dei tempi di attesa, rendendo i servizi più efficienti e reattivi. Infatti con l'*Edge Computing*, solo i dati rilevanti vengono trasmessi alla rete centrale, riducendo la quantità di dati che devono essere trasportati su lunghe distanze. Questo allevia il carico sulla rete e migliora l'efficienza complessiva della banda. Inoltre questa tecnologia consente di distribuire la capacità di elaborazione su una vasta rete di nodi periferici. Questo approccio permette di gestire meglio il crescente numero di dispositivi connessi e la crescente domanda di dati [6].

1.3 Architettura e Componenti del 5G

L'architettura del 5G è progettata per essere altamente flessibile, modulare e scalabile, al fine di soddisfare le tre segmenti: la *Radio Access Network* (RAN) e la *Core Network* e l'*User Equipment* (UE).

1.3.1 Radio Access Network (RAN)

La parte RAN del 5G include le stazioni base (denominate gNodeB o gNB), che comunicano direttamente con i dispositivi mobili. Queste stazioni base

utilizzano tecnologie avanzate come *Massive MIMO* e *Beamforming* per migliorare la capacità di rete e la copertura del segnale. Inoltre la RAN gestisce la trasmissione dei dati tra i dispositivi mobili e il core network, assicurando che i dati siano trasmessi in modo efficiente e con bassa latenza, e supporta anche il *Network Slicing*, che consente di creare reti virtuali su misura per diversi tipi di servizi e applicazioni. Per fare ciò la RAN del 5G adotta un'architettura flessibile e modulare, che consente di separare le funzioni di controllo e gestione dalle funzioni di trasmissione dei dati. Questa separazione permette una maggiore agilità nella gestione della rete e facilita l'implementazione di nuove funzionalità.

1.3.2 Core Network

La *Core Network* del 5G è costituita da varie funzioni di rete, queste collaborano per gestire la mobilità, le sessioni di dati e il traffico utente in modo efficiente [7]. La *Core Network* è responsabile della gestione delle connessioni, della sicurezza, dell'autenticazione degli utenti e dell'instradamento del traffico.

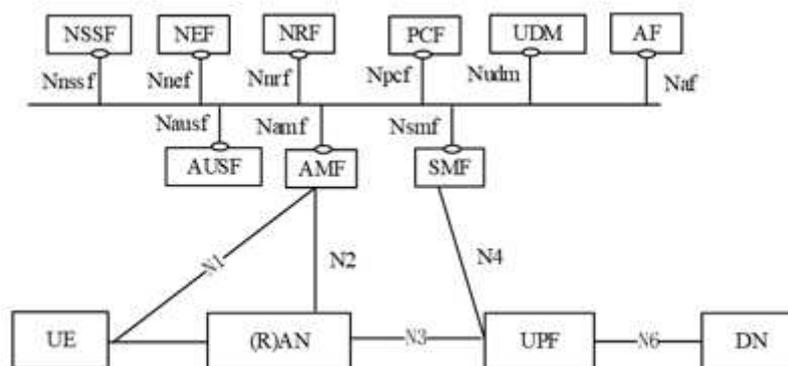


Figura 1.1: Core Network [8].

Le funzioni di rete principali sono:

- **AMF (Access and Mobility Management Function):** È un componente centrale e cruciale nella rete *core* 5G. Funziona come una sorta

di *hub* di controllo centrale per la gestione del piano di controllo nella rete 5G. L'AMF è responsabile di varie funzioni critiche che permettono la gestione dell'accesso, della mobilità e della sicurezza degli utenti (UE, User Equipment) nella rete 5G [9]. In particolare gestisce la registrazione degli utenti; stabilisce e rilascia connessioni di segnalazione del piano di controllo con l'utente; mantiene la conoscenza della posizione dell'utente all'interno della rete e gestisce gli *handover*; autentica gli utenti durante la registrazione iniziale e ne verifica l'identità e la validità delle credenziali SIM.

- **NRF (Network Repository Function):** Tutte le funzioni di rete 5G (NFs) nella rete dell'operatore sono memorizzate centralmente nella *Network Repository Function* (NRF). L'NRF fornisce un'API basata su standard che consente alle NFs 5G di registrarsi e trovarsi reciprocamente. Un elemento cruciale necessario per eseguire la nuova architettura basata sui servizi (SBA) nel core 5G è l'NRF.
- **PCF (Policy Control Function):** È una funzione essenziale nella rete core 5G che gestisce e applica le politiche di controllo della rete [9]. È progettato per supportare la gestione delle risorse di rete, la qualità del servizio (QoS), la gestione della larghezza di banda e altre politiche di controllo necessarie per garantire un funzionamento ottimale della rete 5G.
- **NSSF (Network Slicing Selection Function):** È una funzione fondamentale nella rete core 5G che gestisce la selezione delle *slice* di rete. Le *slice* di rete sono segmenti virtualizzati della rete 5G, ciascuno configurato per soddisfare requisiti specifici di servizio o di applicazione. L'NSSF assicura che ogni dispositivo o servizio sia associato alla *slice* di rete più appropriata, ottimizzando così l'efficienza e le prestazioni della rete.

- **UDM (Unified Data Management):** È una funzione chiave nella rete core 5G che gestisce i dati degli utenti e le informazioni di abbonamento [9]. In sostanza in base agli abbonamenti degli utenti, approva l'accesso alla rete e il *roaming* per accedere ai servizi 5G in modo sicuro e conforme alle politiche di rete. È progettato per essere *cloud-native*, offrendo flessibilità e scalabilità necessarie per supportare i servizi 5G. L'UDM è simile al *Home Subscriber Server* (HSS) utilizzato nelle reti LTE, ma con funzionalità avanzate per il contesto 5G.
- **AUSF (Authentication Server Function):** È una funzione cruciale nella rete core 5G che gestisce i processi di autenticazione degli utenti [9]. L'autenticazione è un aspetto fondamentale per garantire che solo gli utenti autorizzati possano accedere ai servizi della rete 5G. L'AUSF lavora a stretto contatto con altre funzioni di rete per verificare l'identità degli utenti e assicurare la sicurezza della rete.
- **UPF (User Plane Function):** È una componente chiave della rete core 5G che gestisce il piano utente. Il piano utente si occupa della trasmissione dei dati tra l'utente finale e la rete. L'UPF è responsabile dell'instradamento, della gestione e dell'elaborazione del traffico dati, garantendo che i pacchetti di dati viaggino tra i dispositivi degli utenti e le risorse di rete in modo efficiente e sicuro.
- **SMF (Session Management Function):** È una componente cruciale della rete core 5G che si occupa della gestione delle sessioni utente e del piano dati. È responsabile della configurazione, del controllo e della gestione delle sessioni di dati per gli utenti, assicurando che le connessioni siano stabilite, mantenute e terminate in modo efficiente e sicuro [9].

1.3.3 User Equipment (UE)

L'UE 5G (*User Equipment*) è qualsiasi dispositivo che può connettersi e comunicare con la rete 5G. Questi dispositivi includono *smartphone*, *tablet*, *laptop*, dispositivi IoT (*Internet of Things*), dispositivi *wearable*, e altri terminali che possono accedere ai servizi e alle applicazioni offerti dalla rete 5G. L'UE 5G è equipaggiato con la tecnologia necessaria per supportare le velocità elevate, la bassa latenza e le capacità avanzate della rete 5G.

In particolare l'UE 5G interagisce con vari componenti della rete 5G. Con il gNB per avere una connessione alla stazione base 5G per l'accesso alla rete; con l'AMF per gestire la registrazione, autenticazione e mobilità dell'utente; con l'SMF per la gestione delle sessioni di dati e delle politiche di qualità del servizio; con l'UPF per l'instradamento e gestione del traffico dati tra l'UE e la rete; con l'UDM per la fornitura delle informazioni di abbonamento e profilo dell'utente.

1.4 5G Standalone e Non-Standalone

Una delle distinzioni chiave all'interno della tecnologia 5G è tra le modalità *Standalone* (SA) e *Non-Standalone* (NSA).

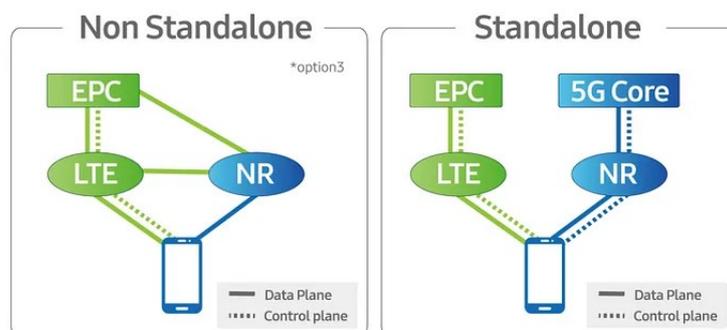


Figura 1.2: 5G SA e NSA [10].

Non-Standalone

Il 5G NSA è una modalità di implementazione iniziale del 5G che utilizza l'infrastruttura esistente del 4G LTE. In questa configurazione, la rete 4G LTE funge da ancoraggio per la rete 5G, con il 5G che fornisce velocità più elevate e capacità aggiuntive. Questa modalità permette un'adozione rapida del 5G, riducendo i costi e il tempo necessario per l'implementazione. Tuttavia, il 5G NSA non sfrutta appieno tutte le potenzialità del 5G, come la bassa latenza e le funzionalità avanzate di *slicing* della rete. In particolare la modalità NSA utilizza l'infrastruttura esistente del 4G LTE come base per la gestione del controllo e del segnale, aggiungendo la capacità del 5G per la trasmissione dei dati [11]. La rete di controllo (*Core Network*) rimane quella del 4G. Può offrire velocità di *download* e *upload* elevate grazie all'utilizzo della banda larga del 5G, ma le prestazioni possono essere limitate dall'infrastruttura 4G sottostante.

Standalone

Il 5G SA è una modalità di implementazione completamente indipendente che utilizza un'infrastruttura di rete 5G dedicata. Questo approccio consente di sfruttare tutte le funzionalità avanzate del 5G, inclusi la bassa latenza, la maggiore efficienza della rete e il *network slicing*, che permette di creare segmenti di rete virtualizzati personalizzati per diversi tipi di servizi o applicazioni. Il 5G SA rappresenta la vera essenza della visione del 5G, offrendo una piattaforma robusta per l'innovazione tecnologica futura. In particolare l'infrastruttura è completamente nuova, con un core network 5G e una rete di accesso radio (RAN) 5G [11]. La latenza è significativamente ridotta rispetto al NSA, permettendo tempi di risposta molto rapidi. Offre velocità di *download* e *upload* massime, grazie alla completa adozione delle tecnologie 5G.

Capitolo 2

Network Slicing

2.1 Introduzione al Network Slicing

Uno dei principali vantaggi delle reti mobili 5G è la capacità di offrire connessioni ad alta velocità e con bassa latenza, anche con un elevato numero di dispositivi connessi. Tuttavia, non sempre sono richieste prestazioni estreme, o comunque non in tutte le situazioni le caratteristiche del 5G vengono sfruttate appieno. Infatti ci sono situazioni in cui la larghezza di banda e la latenza sono cruciali, come nei veicoli autonomi, e altre in cui la larghezza di banda è minima e la latenza non è un problema, come in molti scenari IoT. Assemblare e gestire una rete che supporti sempre tutti questi diversi casi d'uso è costoso e complesso per un operatore 5G, ma soprattutto non è ottimale gestire diverse tipologie di traffico in un'unica maniera. Per questo motivo, il 5G introduce il concetto di *Network Slicing* che consente di dividere il traffico a seconda delle esigenze [12].

Tradizionalmente, le reti mobili sono monolitiche, suddivise solo in una rete di trasporto cablata e *wireless* (la *Core Network*) che veicola il traffico verso e dalla parte *wireless* di accesso (la RAN). Il *Network Slicing*, in estrema

sintesi, applica i concetti di *Software-Defined Networking* a entrambe queste macro-componenti. L'idea è di creare reti virtuali che condividono la stessa infrastruttura fisica di accesso e trasporto [13]. La necessità del *Network Slicing* nasce dal fatto che ogni tipo di applicazione dovrebbe "percepire" una rete configurata in modo ottimale per gestire il suo traffico. Con lo *slicing* questo diventa possibile, anche se la rete percepita dall'applicazione è una "slice", una "fetta" virtuale della rete fisica, non la rete nella sua interezza. Inoltre il *Network Slicing* prevede una virtualizzazione distinta per la parte di accesso e per la parte *core*.

Il *Network Slicing* nella parte di trasporto è più semplice da comprendere perché i principi di base sono gli stessi del SDN applicato alle reti aziendali. L'elemento chiave è la virtualizzazione delle funzioni di rete (NFV, *Network Function Virtualization*). Mentre nella parte RAN è meno intuitivo, poiché nella parte di accesso radio non è possibile separare completamente il *control plane* dallo *user plane*, come previsto dai principi del SDN. Lo *slicing* avviene invece associando a ogni singola "fetta" specifiche regole di gestione del traffico nella parte radio. Il vantaggio del *Network Slicing* non risiede solo nel far percepire a un'applicazione la sua rete ideale, ma anche nel fatto che le varie "fette" della rete fisica sono isolate tra loro. Ciò garantisce una maggiore sicurezza delle comunicazioni e permette di modificare il funzionamento di una *slice* senza influenzare le altre. Questo consente a un operatore di modificare radicalmente un servizio offerto ai clienti intervenendo solo su quello specifico servizio.

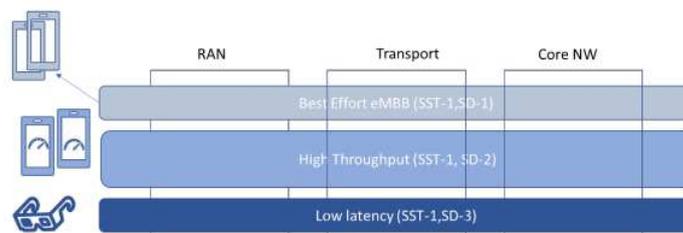


Figura 2.1: Esempio di slicing [14].

2.2 Funzioni di rete

Come detto i diversi scenari applicativi hanno esigenze molto diverse per quanto riguarda le funzioni di rete, le prestazioni del sistema, la sicurezza, l'esperienza dell'utente e così via. Utilizzare la stessa rete per fornire servizi diversi inevitabilmente porta a una rete molto complessa, incapace di soddisfare i requisiti di prestazioni necessari per le applicazioni specifiche, e rende la manutenzione della rete estremamente complessa, aumentando anche i costi operativi. Al contrario, se si forniscono servizi in base alle diverse esigenze dei vari scenari aziendali, attraverso il dispiegamento di una rete proprietaria, che contiene solo le funzioni richieste dagli scenari applicativi, l'efficienza del servizio sarà notevolmente migliorata, in particolare perchè ogni tipologia di traffico viene gestita in maniera autonoma e senza interferire con le altre.

Una *slice* di rete 5G è costituita da un insieme di funzioni di rete, risorse e funzionalità di rete che hanno una configurazione specifica. Queste funzioni di rete e la loro configurazione corrispondente formano una rete logica completa, che contiene le caratteristiche di rete necessarie per un servizio o utente specifico.

In particolare la core network si divide in due: il piano di controllo e il piano dati. Ci sono le funzioni di rete comuni che vengono usate da tutte le *slice*, e le funzioni di rete specifiche che vengono caratterizzate per ogni *slice*. Solitamente tra le funzioni comuni si trovano l'AMF, l'NRF, l'NSSF ed altre, mentre tra quelle specifiche si hanno l'SMF, il PCF, l'UPF e altre. Quindi in generale quelle in comune sono quelle che riguardano il piano di controllo, mentre quelle specifiche sono quelle del piano utente.

2.2.1 UPF

Un ruolo fondamentale nel 5G e in particolare nello *Slicing* è svolto proprio dall'*User Plane Function* (UPF). L'UPF è responsabile della gestione del traffico dati degli utenti, e svolge un ruolo cruciale nella separazione tra il *control plane* e lo *user plane*, permettendo una maggiore flessibilità e scalabilità della rete. In particolare si occupa della trasmissione dei dati effettivi tra gli utenti e la rete [15]. In sostanza fornisce il punto di interconnessione tra l'infrastruttura 5G e la Data Network. Il suo compito è quindi quello di creare e mantenere un tunnel tra un UE e la Data Network per trasportare i flussi di dati delle applicazioni dell'UE.

In generale, nel contesto del 5G, l'UPF svolge diverse funzioni chiave, infatti instrada e inoltra il traffico dati tra l'utente e le reti esterne; applica le politiche di *Quality of Service* (QoS) per garantire che i dati degli utenti ricevano il livello di servizio appropriato in termini di larghezza di banda, latenza e priorità; può eseguire ispezioni approfondite dei pacchetti per applicare politiche di sicurezza o per ottimizzare il traffico; comunica con le funzioni del *control plane*, come il *Session Management Function* (SMF), per ricevere istruzioni su come gestire i flussi di traffico; supporta la continuità del servizio mentre l'utente si sposta tra diverse celle o *slice* di rete [16].

Nel contesto specifico del *Network Slicing*, l'UPF può essere configurato e ottimizzato per soddisfare i requisiti specifici di ciascuna *slice* di rete. Infatti ogni *slice* può avere uno o più UPF dedicati, garantendo che il traffico dati di una *slice* non interferisca con quello di un'altra. Questo isolamento è cruciale per mantenere la qualità del servizio e la sicurezza [17]. Inoltre può essere configurato per applicare politiche di QoS specifiche per ogni *slice*, adattandosi ai requisiti di larghezza di banda, latenza e affidabilità necessari per ciascuna applicazione o servizio. Infine le risorse dell'UPF possono essere allocate e scalate dinamicamente in base alla domanda di traffico, migliorando l'efficienza

e l'utilizzo delle risorse di rete.

2.3 DNN e 5QI

2.3.1 Differenze tra DNN e APN

Il *Data Network Name* (DNN) e l'*Access Point Name* (APN) sono due concetti chiave nelle reti mobili, rispettivamente nel contesto delle reti 5G e 4G LTE. Sebbene abbiano funzioni simili in termini di instradamento del traffico dati, esistono delle differenze fondamentali tra i due termini.

L'APN è un identificatore utilizzato nelle reti 4G LTE per definire le connessioni tra la rete dell'operatore mobile e la rete di destinazione, come Internet o una rete privata. L'APN specifica le impostazioni necessarie per stabilire la connessione, inclusi gli indirizzi IP, i protocolli di sicurezza e le regole di instradamento. Funzioni dell'APN:

- **Identificazione della Rete di Destinazione:** L'APN identifica la rete a cui l'utente si sta connettendo, come Internet o una VPN aziendale.
- **Configurazione delle Connessioni:** L'APN contiene le informazioni necessarie per configurare la connessione dati, inclusi gli indirizzi IP e le impostazioni di sicurezza.
- **Differenziazione del Traffico:** Gli operatori possono utilizzare diversi APN per offrire servizi differenziati, come connessioni Internet standard, connessioni private aziendali o servizi di contenuti specifici.

Il DNN è l'equivalente dell'APN nel contesto delle reti 5G. Analogamente all'APN, il DNN identifica la rete di dati a cui l'utente si connette, ma è progettato per supportare la maggiore flessibilità e le capacità avanzate delle reti 5G. Funzioni del DNN:

- **Identificazione della Rete di Destinazione:** Il DNN identifica la rete di destinazione, che può essere una rete pubblica come Internet o una rete privata specifica.
- **Configurazione delle Connessioni:** Il DNN fornisce le impostazioni necessarie per stabilire e gestire la connessione dati, comprese le politiche di QoS e di sicurezza.
- **Supporto per Servizi:** Uno degli aspetti più innovativi del DNN è la sua integrazione con il *Network Slicing* nel 5G, consentendo la creazione di *slice* di rete personalizzate con configurazioni specifiche per diversi servizi o applicazioni.
- **Flessibilità Avanzata:** Il DNN può supportare una gamma più ampia di servizi e scenari applicativi rispetto all'APN, grazie alla maggiore flessibilità e alle capacità avanzate delle reti 5G.

Oltre a queste, un'altra differenza fondamentale sta nel fatto che l'APN permette di "switchare" il traffico a differenti *gateway* nella *Core Network*, ma controlla le componenti solo dopo il *gateway* (non può avere il controllo sulle componenti prima del *gateway*), mentre il *Network Slicing* può controllare le configurazioni sia per l'APN ma anche per il percorso *Radio Access*.

2.3.2 QoS

Una delle caratteristiche distintive dell'architettura 5G è la capacità di differenziare e gestire i flussi di traffico con diversi requisiti di *Quality of Service* (QoS). La Qualità del Servizio (QoS) si riferisce alle prestazioni di una rete o di un servizio percepite dai suoi utenti. Nel contesto del 5G, la QoS include vari fattori, come affidabilità, disponibilità, latenza, *throughput* e prioritizzazione del traffico. Questi elementi sono cruciali per garantire un'esperienza utente ottimale e soddisfare le esigenze diversificate delle applicazioni.

La QoS nella Core Network 5G è definita dai flussi di QoS, che comprendono il *Guaranteed Bit Rate* (GBR), il *Non-guaranteed Bit Rate* (Non-GBR) e i flussi riflessivi per configurazioni dinamiche. Questi flussi di QoS stabiliscono i parametri di qualità specifici applicabili al traffico dello *user plane*. Ogni flusso di QoS nella *Core Network* 5G è identificato da un *QoS Flow ID* (QFI). Il QFI funge da identificatore univoco per un determinato flusso di QoS e consente una gestione e un controllo precisi del traffico in base ai requisiti di qualità specifici.

All'interno di una *PDU Session*, il traffico dello *user plane* che condivide lo stesso QFI riceve un trattamento coerente per quanto riguarda l'instradamento del traffico. Ciò include aspetti come la schedulazione e le soglie di ammissione. Assegnando lo stesso QFI ai flussi di traffico correlati, la rete assicura che questi flussi siano gestiti in modo uniforme, mantenendo le garanzie di QoS desiderate e la coerenza all'interno della *PDU Session*.

Il controllo e l'istituzione dei flussi di QoS sono affidati alla *Session Management Function* (SMF) all'interno della *Core Network 5G*. L'SMF ha l'autorità di preconfigurare i flussi di QoS o di istituirli dinamicamente tramite procedure come l'istituzione della *PDU Session* o la modifica della *PDU Session*. Questa flessibilità consente un'allocazione efficiente e un adattamento delle risorse di QoS in base ai requisiti specifici del traffico dello *user plane* e alle condizioni della rete.

Due parametri QoS fondamentali inclusi nel profilo QoS per ogni flusso QoS sono l'*Allocation and Retention Priority* (ARP) e il *5G QoS Identifier* (5QI). L'ARP determina il livello di priorità assegnato al flusso QoS, consentendo alla rete di allocare le risorse di conseguenza. Il 5QI rappresenta la specifica classe di QoS assegnata al flusso, definendo le caratteristiche di qualità e l'accordo di livello di servizio associato. Per i flussi QoS Non-GBR, un ulteriore parametro QoS chiamato *Reflective QoS Attribute* (RQA) può essere incluso nel profilo

QoS. L'RQA fornisce ulteriori informazioni sui requisiti di configurazione dinamica per il flusso Non-GBR, permettendo alla rete di adattarsi e regolare il proprio comportamento di conseguenza.

ARP

L'inclusione del parametro *Allocation and Retention Priority* (ARP) nel profilo QoS svolge vari ruoli all'interno della *Core Network 5G*. In primo luogo, permette alla rete di decidere se accettare o rifiutare le richieste di istituzione, modifica o *handover* dei flussi QoS, soprattutto in presenza di limitazioni di risorse. Considerando il livello di ARP del flusso richiesto, la rete può stabilire se è in grado di soddisfare i requisiti di risorse del flusso o se deve rifiutarlo a causa di vincoli di risorse. Inoltre, l'ARP può essere utilizzato per determinare quale flusso QoS rilasciare e liberare risorse durante le limitazioni di risorse, gestendo efficacemente l'allocazione delle risorse in modo dinamico. Il livello di priorità ARP assegnato a un flusso QoS definisce la sua importanza relativa all'interno della rete. La gamma dei livelli di priorità ARP va da 1 a 15, con 1 che rappresenta il livello di priorità più alto [18]. Ciò consente una granularizzazione della priorità, permettendo alla rete di distinguere tra i diversi flussi e allocare le risorse di conseguenza in base ai livelli di priorità assegnati. Assegnando specifici livelli di priorità ARP ai flussi QoS, la rete può garantire che i flussi con livelli di priorità più elevati ricevano un trattamento preferenziale e un'allocazione delle risorse.

La capacità di pre-emption dell'ARP è un altro aspetto determinato dal parametro ARP. Essa definisce se un flusso QoS può ottenere risorse che erano state precedentemente assegnate a un altro flusso QoS con un livello di priorità ARP inferiore. In situazioni in cui le risorse diventano scarse, un flusso QoS con priorità più alta può dover pre-emptare le risorse di un flusso a priorità inferiore per soddisfare i suoi requisiti. La capacità di pre-emption dell'ARP consente alla rete di prendere decisioni informate riguardo alla riallocazione

delle risorse e di garantire che i flussi con livelli di priorità più elevati possano ottenere le risorse necessarie anche se inizialmente allocate a flussi a priorità inferiore.

2.3.3 5QI

Il 5QI è un valore scalare che serve come riferimento per una combinazione standardizzata di caratteristiche QoS del 5G. Rappresenta un insieme di parametri specifici del nodo di accesso che governano il trattamento QoS del traffico per un flusso QoS specifico. Questi parametri includono vari aspetti come i pesi di schedulazione, le soglie di ammissione, le soglie di gestione delle code e la configurazione del protocollo del livello di collegamento, tra gli altri [8]. Parametri del 5QI:

- **Tipo di Risorsa:** Un flusso QoS GBR può utilizzare il tipo di risorsa GBR o il tipo di risorsa GBR critico per il ritardo. Questi tipi di risorse hanno definizioni distinte per parametri come il *Packet Delay Budget*(PDB) e il *Packet Error Rate* (PER). Inoltre, il parametro *Maximum Data Burst Volume* (MDBV) è applicabile solo al tipo di risorsa GBR critico per il ritardo. La scelta del tipo di risorsa determina le caratteristiche e i requisiti specifici associati al flusso QoS.
- **Livello di Priorità:** Il livello di priorità assegnato a un flusso QoS indica la sua importanza relativa nella schedulazione delle risorse di rete. I livelli di priorità sono definiti in modo tale che il livello più basso corrisponda alla priorità più alta. Questo parametro viene utilizzato per distinguere tra i flussi QoS dello stesso UE e quelli provenienti da diversi UE. Assegnando i livelli di priorità, la rete può gestire e prioritizzare efficacemente l’allocazione delle risorse in base alle esigenze specifiche di ciascun flusso.

- **Packet Delay Budget:** Specifica il limite massimo per il ritardo che un pacchetto può subire tra l'UE e il punto di terminazione presso l'UPF. Questo parametro assicura che il ritardo nella trasmissione dei pacchetti rimanga entro limiti accettabili. Lo stesso valore del *Packet Delay Budget* è applicabile sia per la direzione *Uplink* che per la direzione *Downlink*, garantendo coerenza nelle prestazioni di ritardo.

- **Maximum Data Burst Volume (MDBV):** Il parametro MDBV rappresenta la quantità massima di dati che la rete di accesso 5G è obbligata a gestire entro un determinato periodo, in linea con il PDB. Indica la dimensione del *burst* di dati che può essere servito dalla rete senza superare i limiti specificati.

- **Packet Error Rate (PER):** Definisce un limite massimo per il tasso di pacchetti IP che sono stati processati dal mittente di un protocollo del livello di collegamento, ma che non sono stati consegnati con successo al ricevitore corrispondente del livello superiore. Stabilisce una soglia per il livello accettabile di perdita di pacchetti nel processo di trasmissione, garantendo una consegna affidabile dei dati tra gli enti della rete.

- **Default Averaging Window (DAW):** Ogni flusso QoS GBR è associato a una DAW. Questa rappresenta la durata su cui vengono calcolati i parametri GBR e MBR. Questa finestra determina il periodo di tempo su cui vengono misurati i valori GBR e MBR e fornisce una base per un'accurata allocazione e gestione delle risorse.

5QI Value	Resource Type	Default Priority Level	Packet Delay Budget	Packet Error Rate	Default Maximum Data Burst Volume (NOTE 3)	Default Averaging Window	Example Services
1	GBR (NOTE 1)	20	100 ms (NOTE 11, NOTE 12)	10 ⁻²	N/A	2000 ms	Conversational Voice
2		40	150 ms (NOTE 11, NOTE 13)	10 ⁻³	N/A	2000 ms	Conversational Video (Live Streaming)
3		30	80 ms (NOTE 11, NOTE 13)	10 ⁻³	N/A	2000 ms	Real Time Gaming, V2X messages (see TS 23.287), Electricity distribution medium voltage, Process automation monitoring
4		50	300 ms (NOTE 11, NOTE 13)	10 ⁻⁶	N/A	2000 ms	Non-Conversational Video (Buffered Streaming)
65 (NOTE 9, NOTE 12)		7	75 ms (NOTE 7, NOTE 8)	10 ⁻²	N/A	2000 ms	Mission Critical user plane Push To Talk voice (e.g., MCPTT)
66 (NOTE 12)		20	100 ms (NOTE 10, NOTE 13)	10 ⁻²	N/A	2000 ms	Non-Mission-Critical user plane Push To Talk voice
67 (NOTE 12)		15	100 ms (NOTE 10, NOTE 13)	10 ⁻³	N/A	2000 ms	Mission Critical Video user plane
75 (NOTE 14)		N/A	N/A	N/A	N/A	N/A	N/A
71		96	150 ms (NOTE 11, NOTE 15)	10 ⁻⁶	N/A	2000 ms	"Live" Uplink Streaming (e.g. TS 26.238)
72		96	300 ms (NOTE 11, NOTE 15)	10 ⁻⁴	N/A	2000 ms	"Live" Uplink Streaming (e.g. TS 26.238)
73		96	300 ms (NOTE 11, NOTE 15)	10 ⁻⁶	N/A	2000 ms	"Live" Uplink Streaming (e.g. TS 26.238)
74		96	500 ms (NOTE 11, NOTE 15)	10 ⁻⁶	N/A	2000 ms	"Live" Uplink Streaming (e.g. TS 26.238)
76		96	800 ms (NOTE 11, NOTE 15)	10 ⁻⁴	N/A	2000 ms	"Live" Uplink Streaming (e.g. TS 26.238)
5		non-GBR (NOTE 1)	10	100 ms (NOTE 10, NOTE 13)	10 ⁻⁶	N/A	N/A
6 (NOTE 1)	80		300 ms (NOTE 10, NOTE 13)	10 ⁻⁶	N/A	N/A	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7	70		100 ms (NOTE 10, NOTE 13)	10 ⁻³	N/A	N/A	Voice, Video (Live Streaming) Interactive Gaming
8	80		300 ms (NOTE 13)	10 ⁻⁶	N/A	N/A	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9	90		300 ms (NOTE 13)	10 ⁻⁶	N/A	N/A	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
69 (NOTE 9, NOTE 12)	5		60 ms (NOTE 7, NOTE 8)	10 ⁻⁶	N/A	N/A	Mission Critical delay sensitive signalling (e.g., MC-PTT signalling)
70 (NOTE 12)	55		200 ms (NOTE 7, NOTE 10)	10 ⁻⁶	N/A	N/A	Mission Critical Data (e.g. example services are the same as 5QI 6,8,9)
79	65		80 ms (NOTE 10, NOTE 13)	10 ⁻²	N/A	N/A	V2X messages (see TS 23.287)
80	68		10 ms (NOTE 5, NOTE 10)	10 ⁻⁶	N/A	N/A	Low Latency eMBB applications Augmented Reality
82	Delay critical GBR		19	10 ms (NOTE 4)	10 ⁻⁴	255 bytes	2000 ms
83		22	10 ms (NOTE 4)	10 ⁻⁴	1354 bytes (NOTE 3)	2000 ms	Discrete Automation (see TS 22.261); V2X messages (UE-RSU Platooning, Advanced Driving; Cooperative Lane Change with low LoA, See TS 22.185)
84		24	30 ms (NOTE 5)	10 ⁻⁵	1354 bytes (NOTE 3)	2000 ms	Intelligent transport systems (see TS 22.261)
85		21	5 ms (NOTE 5)	10 ⁻⁵	255 bytes	2000 ms	Electricity Distribution-High voltage (see TS 22.261); V2X messages (Remote Driving, See TS 22.185; NOTE 18, see TS 22.261)
86		18	5 ms (NOTE 5)	10 ⁻⁴	1354 bytes	2000 ms	V2X messages (Advanced Driving; Collision Avoidance, Platooning with high LoA, See TS 22.185, TS 23.287)
87		25	5 ms (NOTE 4)	10 ⁻³	500 bytes	2000 ms	Interactive Service - Motion tracking data, (see TS 22.261)
88		25	10 ms (NOTE 4)	10 ⁻³	1125 bytes	2000 ms	Interactive Service - Motion tracking data, (see TS 22.261)
89		25	15 ms (NOTE 4)	10 ⁻⁴	17000 bytes	2000 ms	Visual content for cloud/edge split rendering (see TS 22.261)
90	25	20 ms (NOTE 4)	10 ⁻⁴	63000 bytes	2000 ms	Visual content for cloud/edge split rendering (see TS 22.261)	

Figura 2.2: Tabella valori 5QI [8].

2.4 SST e SD

Il concetto di *Single Network Slice Selection Assistance Information* (S-NSSAI) è cruciale nell'architettura delle reti 5G per supportare il *Network Slicing*. L'S-NSSAI è un identificatore che consente alla rete 5G di selezionare e gestire specifiche *slice* di rete per i vari servizi e applicazioni. Questo identificatore è composto da due componenti principali: il *Slice/Service Type* (SST) e il *Slice Differentiator* (SD).

L'SST è un identificatore che specifica il tipo di servizio che una *slice* di rete 5G intende supportare. L'SST indica la natura della *slice* e i suoi requisiti di servizio, consentendo una configurazione ottimale della rete per diversi scenari d'uso. Funzioni dell'SST:

- **Identificazione del Tipo di Servizio:** L'SST aiuta la rete a identificare il tipo di servizio che la *slice* deve supportare, consentendo una configurazione appropriata delle risorse di rete.
- **Ottimizzazione delle Risorse:** Permette di ottimizzare l'allocazione delle risorse di rete per garantire che le specifiche esigenze di servizio siano soddisfatte.

Il campo SST ha una lunghezza di 8 bit, il che implica che può indicare un totale di 256 diversi tipi di *slice*. Inoltre può avere valori standardizzati e non standardizzati. I valori da 0 a 127 appartengono all'intervallo SST standardizzato. I valori da 128 a 255 appartengono all'intervallo specifico dell'operatore. L'SST è obbligatorio, quindi la *slice* deve avere un valore in questo campo [8].

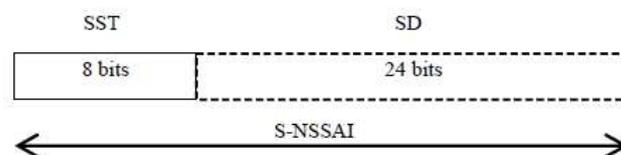


Figura 2.3: SST e SD [8].

L'SD, invece, è un identificatore opzionale che fornisce ulteriori dettagli per distinguere tra *slice* con lo stesso SST. L'SD consente di differenziare ulteriormente le *slice* che, pur appartenendo alla stessa categoria di servizio, hanno requisiti specifici o configurazioni particolari. Funzioni dell'SD:

- **Personalizzazione delle Slice:** L'SD consente una personalizzazione più fine delle *slice*, permettendo di adattare ulteriormente le risorse di rete in base a esigenze specifiche.
- **Differenziazione dei Servizi:** Aiuta a differenziare i servizi forniti da diverse *slice*, anche se appartengono alla stessa categoria SST, garantendo che ogni *slice* possa essere configurata e ottimizzata per requisiti specifici.

Come detto il campo SD è un campo opzionale, inoltre una particolarità dell'SD è che ha un valore riservato al "no SD value associated with the SST" definito come esadecimale FFFFFFFF. In alcuni protocolli, il campo SD non è incluso per indicare che nessun valore SD è associato all'SST. Viene utilizzato per indicare *slice* specifiche nel caso in cui una o più *slice* siano supportate nello stesso SST.

2.4.1 Valori standard dell'SST

I valori SST standardizzati forniscono un modo per stabilire l'interoperabilità globale per lo *slicing*, in modo che i *Public Land Mobile Networks* (PLMN) possano supportare il caso d'uso del *roaming* in modo più efficiente per i tipi di *slice*/servizi più comunemente utilizzati. I valori standard sono:

Slice/Service type	SST value	Characteristics
eMBB	1	Slice suitable for the handling of 5G enhanced Mobile Broadband.
URLLC	2	Slice suitable for the handling of ultra- reliable low latency communications.
MIoT	3	Slice suitable for the handling of massive IoT.
V2X	4	Slice suitable for the handling of V2X services.
HMTC	5	Slice suitable for the handling of High-Performance Machine-Type Communications.
HDLLC	6	Slice suitable for the handling of High Data rate and Low Latency Communications.

Figura 2.4: Valori standard SST [8].

- **eMBB (enhanced Mobile Broadband)**: Ottimizzato per applicazioni che richiedono alta velocità di trasmissione dati e grande capacità di *throughput*.
- **URLLC (Ultra-Reliable Low Latency Communications)**: Progettato per applicazioni che necessitano di bassa latenza e alta affidabilità.
- **MIoT (Massive Internet of Things)**: Destinato a supportare un gran numero di dispositivi IoT che richiedono connessioni stabili ma con bassi requisiti di larghezza di banda.
- **V2X (Vehicle-to-Everything)**: Progettato per supportare la comunicazione tra veicoli e infrastrutture stradali, altri veicoli, pedoni e reti.
- **HMTC (High Mobility Tactical Communications)**: Ottimizzato per comunicazioni tattiche ad alta mobilità, spesso utilizzate in ambienti critici e dinamici.
- **HDLLC (High Definition Low Latency Communication)**: Progettato per comunicazioni ad alta definizione con bassa latenza, ideale per applicazioni che richiedono sia una qualità video elevata sia una risposta rapida.

Ogni tipo di SST standardizzato nel 5G è progettato per soddisfare specifici requisiti di servizio e ottimizzare le risorse di rete per diverse applicazioni. Questi SST permettono alle reti 5G di fornire un'ampia gamma di servizi con prestazioni ottimali, garantendo al contempo l'interoperabilità globale e l'efficienza nel supportare il *roaming* tra diversi PLMN.

Capitolo 3

Software utilizzati

In questo progetto sono stati utilizzati due software per la parte *Core* e per la parte radio, cioè *Open5GS* e *srsRAN*.

3.1 Open5GS

Open5GS è una *suite software open source*, in linguaggio C, che fornisce un'implementazione completa della rete *Core* per le reti mobili 4G (LTE) e 5G. È sviluppato e mantenuto da una comunità di sviluppatori e ricercatori che contribuiscono alla sua evoluzione continua. Progettato per essere altamente modulare e flessibile, Open5GS permette ai ricercatori, sviluppatori e operatori di rete di implementare e gestire reti mobili avanzate senza costi elevati associati alle soluzioni proprietarie.

La *suite* comprende tutti i componenti principali necessari per il funzionamento di una rete *Core*, come l'MME (*Mobility Management Entity*), l'HSS (*Home Subscriber Server*), l'SGW (*Serving Gateway*) e il PGW (*Packet Gateway*) per le reti 4G, oltre ai componenti 5G come l'AMF (*Access and Mobility Manage-*

ment Function), l'SMF (Session Management Function) e l'UPF (User Plane Function).

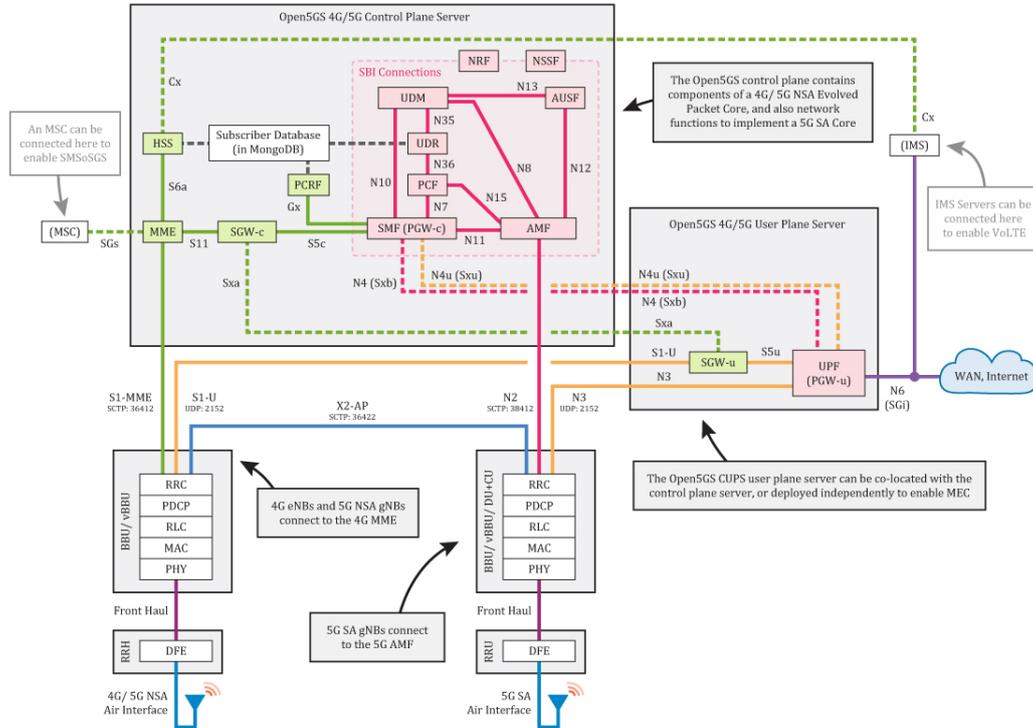


Figura 3.1: Architettura di rete di Open5GS [19].

Questo rende Open5GS una scelta ideale per la sperimentazione, la prototipazione e lo sviluppo di reti mobili private e pubbliche. Inoltre questo software offre una serie di funzionalità che lo rendono adatto per la ricerca e lo sviluppo di reti mobili 5G e LTE:

- **Supporto per 5G NSA e SA:** Open5GS supporta sia la modalità *Non-Standalone* che *Standalone*, permettendo la transizione graduale dalle reti LTE alle reti 5G complete.
- **Compatibilità con varie tecnologie di accesso radio:** Può essere integrato con diverse tecnologie di accesso radio (RAN), come eNB e gNB.

- **Gestione avanzata delle sessioni e della mobilità:** Include funzionalità avanzate per la gestione delle sessioni degli utenti e la loro mobilità all'interno della rete.
- **Scalabilità e flessibilità:** La sua architettura modulare consente di scalare facilmente il sistema e di adattarlo a diverse dimensioni di *deployment*, da piccole reti di test a grandi implementazioni commerciali.

L'architettura di Open5GS è progettata per essere scalabile, consentendo la distribuzione su più server per gestire un alto numero di utenti e traffico. La comunità *open source* attiva fornisce supporto continuo, aggiornamenti e nuove funzionalità, garantendo che il progetto rimanga all'avanguardia della tecnologia mobile.

Open5GS è utilizzato in una varietà di scenari, dalla ricerca accademica allo sviluppo commerciale. È ampiamente utilizzato nelle università e nei centri di ricerca per studiare e sviluppare nuove tecnologie e protocolli per le reti mobili. Le aziende di telecomunicazioni e i fornitori di soluzioni di rete utilizzano Open5GS per la prototipazione di nuove funzionalità e servizi 5G. Fornisce un ambiente flessibile per il testing e la validazione delle reti 5G, permettendo di simulare vari scenari di rete e di verificare le prestazioni e la compatibilità dei componenti. Infine viene anche utilizzato in sperimentazioni sul campo e pilotaggi per valutare le prestazioni delle reti 5G in condizioni reali.

3.2 srsRAN

L'altro software utilizzato è srsRAN. Questo è una *suite* di *software open source* per lo sviluppo e la ricerca sulle reti di accesso radio (RAN) LTE e 5G. Originariamente progettato per supportare la ricerca accademica, srsRAN è diventato uno strumento popolare anche nell'industria delle telecomunicazioni per la prototipazione e il testing di nuove tecnologie. L'architettura di srsRAN è

modulare e flessibile, permettendo l'implementazione di vari componenti della rete LTE e 5G. I principali componenti di srsRAN includono:

- **srsENB**: Implementazione dell'eNodeB (eNB) per la gestione della connessione radio con gli utenti e l'instradamento dei dati tra gli utenti e la rete core.
- **srsUE**: Implementazione del User Equipment (UE) per emulare il comportamento di un dispositivo mobile in una rete LTE/5G.
- **srsEPC**: Implementazione dell'Evolved Packet Core (EPC), che gestisce le funzioni di core network per le reti LTE.
- **srsGNB**: Implementazione del gNodeB (gNB) per la gestione delle connessioni radio in una rete 5G.

SrsRAN offre una serie di funzionalità avanzate che lo rendono adatto per una vasta gamma di applicazioni di ricerca e sviluppo:

- **Supporto per LTE e 5G NR**: Include il supporto per entrambe le tecnologie LTE e 5G New Radio (NR), permettendo la ricerca e lo sviluppo su entrambe le generazioni di reti mobili.
- **Modularità e flessibilità**: La sua architettura modulare permette di utilizzare e combinare diversi componenti in base alle esigenze specifiche del progetto.
- **Compatibilità con hardware SDR**: Può essere utilizzato con hardware di radio definita dal software (SDR), come USRP, per implementare e testare reti reali.
- **Supporto per MIMO e tecnologie avanzate**: Include il supporto per tecnologie avanzate come MIMO e *beamforming*, essenziali per le prestazioni avanzate delle reti 5G.

- **Strumenti di analisi e debugging:** Fornisce una serie di strumenti per l'analisi e il debugging delle reti, facilitando la ricerca e lo sviluppo.

Come Open5GS anche srsRAN è utilizzato in una varietà di contesti, dai laboratori di ricerca accademici alle sperimentazioni sul campo nelle industrie delle telecomunicazioni.

3.3 Installazione Open5GS

Per l'installazione del *software* Open5GS, si è seguito il tutorial disponibile direttamente sul sito ufficiale del software [19]. La procedura inizia con il *download* di MongoDB, un sistema di gestione di database progettato per memorizzare, recuperare e gestire dati in formato BSON (Binary JSON). MongoDB viene utilizzato per gestire le sessioni utente, memorizzare i profili utente, i dati di autenticazione e autorizzazione, le configurazioni di rete e i registri delle attività. Successivamente, si installa GnuPG (GNU Privacy Guard), una *suite open source* utilizzata per la crittografia e la sicurezza dei dati, necessaria per gestire la chiave GPG di MongoDB. Infine si fa in modo che il software si avvii in automatico con il sistema tramite:

```
$ sudo systemctl enable mongod
```

Quindi si procede con l'installazione dei pacchetti necessari per compilare correttamente il codice sorgente di Open5GS. Il codice sorgente è disponibile su GitHub, quindi si clona il *repository* e si segue una serie di comandi per compilare il *software*. Una volta completata la compilazione, si verifica il successo dell'operazione tramite:

```
$ ./build/tests/registration/registration
```

e infine si completa l'installazione con:

```
$ ninja install
```

Successivamente, si seguono le istruzioni del tutorial per installare WebUI, un'interfaccia grafica che semplifica il processo di registrazione delle SIM e consente la connessione alla rete configurata.

3.4 Installazione srsRAN

Anche per la parte della rete di accesso, ovvero per il *software* srsRAN si è seguito il tutorial disponibile sul sito ufficiale [20]. Per prima cosa si scaricano le dipendenze che servono per la corretta compilazione del codice sorgente di SRSRAN. Questo passaggio è cruciale per assicurare che tutte le librerie e i pacchetti richiesti siano presenti e correttamente configurati sul sistema. Successivamente, si clona il *repository* srsRAN direttamente da GitHub, dove è disponibile il codice sorgente aggiornato del *software*.

```
$ git clone https://github.com/srsRAN/srsRAN_Project.git
```

Quindi si esegue la compilazione eseguendo una serie di comandi specifici forniti nel tutorial. Una volta completata la compilazione con successo, si procede con l'installazione del *software* srsRAN sul sistema.

```
$ sudo make install
```

Capitolo 4

Realizzazione del test set ed implementazione del gNB

Dopo aver effettuato il download dei software, è fondamentale aggiornare l'FPGA. Per eseguire questa operazione, è necessario utilizzare il seguente comando:

```
$ uhd_image_loader --args="type=x300 , addr=192.168.30.2 ,  
fpga=XG"
```

Questo comando permette di caricare l'immagine FPGA relativa alla X310 della famiglia X300 con l'indirizzo IP specificato. L'ultimo parametro (fpga=XG) è utilizzato per configurare il comportamento a 10 Gigabit delle porte SFP.

4.1 Configurazione rete 5G per test latenza

Dopo aver configurato la Core Network, si procede con la configurazione della rete di accesso. In primo luogo, è necessaria l'installazione dei driver, che permettono al computer di comunicare con il dispositivo X310. Questo dispositivo funge da gNB per la rete di accesso 5G. Successivamente, si procede con la

modifica dei file di configurazione destinati al gNB. Questi file si trovano nella cartella di installazione di SRSRAN Project, nel percorso ‘srsRAN Project/-configs’. Sebbene siano disponibili numerosi file di configurazione, non ce ne sono di specifici per il dispositivo X310. Pertanto, si è deciso di utilizzare i file relativi ai dispositivi b200 e b210, ovvero `gnb_b210_20MHz_oneplus_8t.yml` e il `gnb_rf_b210_fdd_srsUE.yml`, modificando i parametri specifici. Questi parametri includono informazioni cruciali come la larghezza di banda, l’SCS, l’utilizzo del MIMO e gli MCS utilizzati. I file sono stati poi rinominati `gnb_x310_20MHz_oneplus_8t.yml` e `gnb_rf_x310_fdd_srsUE.yml`. Il primo file è progettato per la trasmissione nella banda n78, utilizzando la modalità TDD con una larghezza di banda di 20 MHz e una separazione tra le sottoportanti di 30 kHz. Le modifiche apportate includono l’adattamento del PLMN ID per includere quello di test, il TAC per consentire alla cella di connettersi alla Core Network, il clock interno e i guadagni di antenna in trasmissione e ricezione per garantire una comunicazione affidabile. Inoltre, sono stati introdotti campi aggiuntivi per specificare il MCS con cui la cella deve trasmettere e il MIMO. Il file di configurazione è riportato di seguito, con i campi corrispondenti ai parametri fondamentali evidenziati [21].

```

*****
amf:
  addr: 127.0.1.100
  bind_addr: 127.0.0.1
ru_sdr:
  device_driver: uhd
  device_args: type=x300, addr=192.168.30.2
  clock: internal
  sync: internal
  srate: 23.04
  tx_gain: 20
  rx_gain: 0
  time_alignment_calibration: -200

cell_cfg:
  dl_arfcn: 627340
  band: 78
  channel_bandwidth_MHz: 20
  common_scs: 30
  plmn: "00101"
  tac: 7
  nof_antennas_ul: 1
  nof_antennas_dl: 1
  pci: 1
  pusch:
    min_ue_mcs: [0-28]
    max_ue_mcs: [0-28]
    mcs_table: qam256, qam64
  pdsch:
    min_ue_mcs: [0-28]
    max_ue_mcs: [0-28]
    mcs_table: qam256, qam64
*****

```

Nel secondo file `gnb_rf_x310_fdd_srsUE.yml` utilizzato durante le misure, la trasmissione è stata impostata nella banda n7, utilizzando la modalità FDD con una larghezza di banda di 10 MHz e una separazione tra le sottoportanti (SCS) di 15 kHz. Le modifiche apportate hanno incluso la personalizzazione del PLMN, del TAC, del clock interno e dei guadagni delle antenne sia in trasmissione che in ricezione. Anche in questo caso, sono stati aggiunti campi per specificare il MCS e il MIMO.

```

*****
amf:
  addr: 127.0.1.100
  bind_addr: 127.0.0.1
ru_sdr:
  device_driver: uhd
  device_args: type=x300, addr=192.168.30.2
  clock: internal
  sync: internal
  srate: 11.52
  tx_gain: 20
  rx_gain: 0

cell_cfg:
  dl_arfcn: 531000
  band: 7
  channel_bandwidth_MHz: 10
  common_scs: 15
  plmn: "00101"
  tac: 7
  nof_antennas_ul: 1
  nof_antennas_dl: 1
  pci: 1
  pusch:
    min_ue_mcs: [0-28]
    max_ue_mcs: [0-28]
    mcs_table: qam256, qam64
  pdsch:
    min_ue_mcs: [0-28]
    max_ue_mcs: [0-28]
    mcs_table: qam256, qam64
*****

```

4.1.1 Configurazione della SIM

Dopodiché si passa alla configurazione della SIM attraverso il software fornito da Open Cells, chiamato *UICC/SIM programming*. La configurazione avviene in maniera molto semplice passando tutti i parametri attraverso il comando:

```
$ sudo ./program_uicc --port /dev/ttyUSB0
--adm 10803136
--opc 504f20634f6320504f50206363500a4f
--iccid 89860061100000000021
--isdn 00000021
--key 6874736969202073796d4b2079650a73
--spn "Open5GS"
--imsi 001010100001121
```

Come valori sono stati inseriti quelli specificati da Open5GS, in particolare l'IMSI (*International Mobile Subscriber Identity*) è un identificativo univoco dell'abbonato ed è composto da una sequenza di 15 cifre, suddivise in tre parti fondamentali: l'MCC, l'MNC e il *Mobile Subscriber Identification Number* (MSIN). Le prime cinque cifre rappresentano il PLMN ID, mentre l'MSIN è il numero di telefono associato alla SIM e serve a identificare in modo univoco l'abbonato all'interno della rete e dell'operatore specifici.

Quindi si passa alla registrazione della SIM tramite WebUI. Si accede a quest'ultima attraverso l'indirizzo `https://localhost:3000`, si deve effettuare l'accesso inserendo come username "admin" e come password 1423. Si preme il tasto + e si inseriscono i valori associati a quelli riportati nella configurazione della SIM. In particolare abbiamo:

```
AMF:8000
SUBSCRIBER KEY:6874736969202073796d4b2079650a73
USIM Type: OPc
Operator Key: 504f20634f6320504f50206363500a4f
APN:internet Type*:ipv4
5QI/QCI*:7
ARP Priority Level (1-15)*:8
```

Seguendo tutti questi passaggi si ottiene questo:

Edit Subscriber

Subscriber Configuration

IMSI*
001010100001121

Subscriber Key (K)*
6674736969202073796d4b2079650a73

Authentication Management Field (AMF)*
B000

USIM Type
OPc

Operator Key (OPc/OP)*
504f20634f6320504f50206363500a4f

UE-AMBR Downlink*
1

Unit
Gbps

UE-AMBR Uplink*
1

Unit
Gbps

Slice Configurations

SST*
1 2 3 4

SD

Default S-NSSAI

Session Configurations

DNN/APN*
internet

Type*
IPv4

SQI/QCI*
7

ARP Priority Level (1-15)*
8

Capability*
Disabled

Vulnerability*
Disabled

Figura 4.1: Interfaccia WebUI per configurare la SIM.

4.1.2 Configurazione del TestCenter

Per quanto riguarda i test di latenza il banco utilizzato è stato così schematizzato:

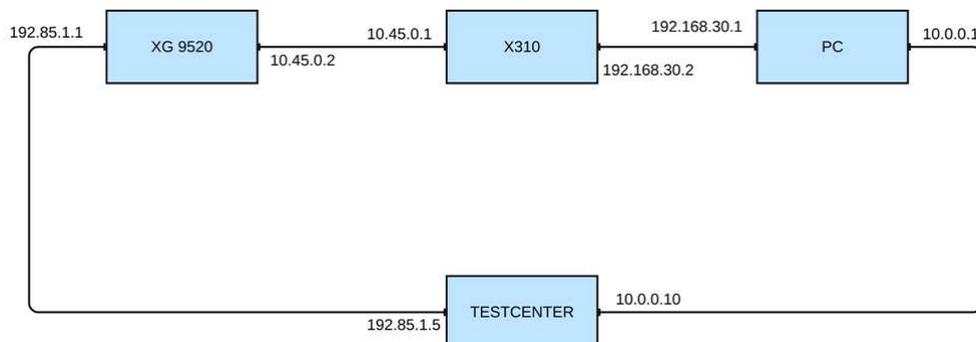


Figura 4.2: Banco test di latenza.

Il setup sperimentale coinvolge il PC Asus ExpertCenter, la radio X310, il router ATLC XG9520-5GWAX e lo Spirent TestCenter. È importante notare che la connessione tra il PC e la radio avviene tramite SFP Ethernet a 10 Gigabit, mentre sia l'XG9520-5GWAX che il computer Asus ExpertCenter vengono collegati tramite due cavi Ethernet al TestCenter alle porte 5 e 6: in questo modo lo Spirent TestCenter può generare il traffico su una porta e misurare la latenza sulla porta di arrivo.

Partendo dal TestCenter, si assegna alla porta 5 l'indirizzo IP 10.0.0.10, la *netmask* 255.255.255.0 e il *gateway* 10.0.0.1. Allo stesso modo per la porta 6 si imposta l'indirizzo IP 192.85.1.5, *netmask* 255.255.255.0 e *gateway* 192.85.1.1, quest'ultimo è l'indirizzo da assegnare all'interfaccia Ethernet dell'XG9520-5GWAX.

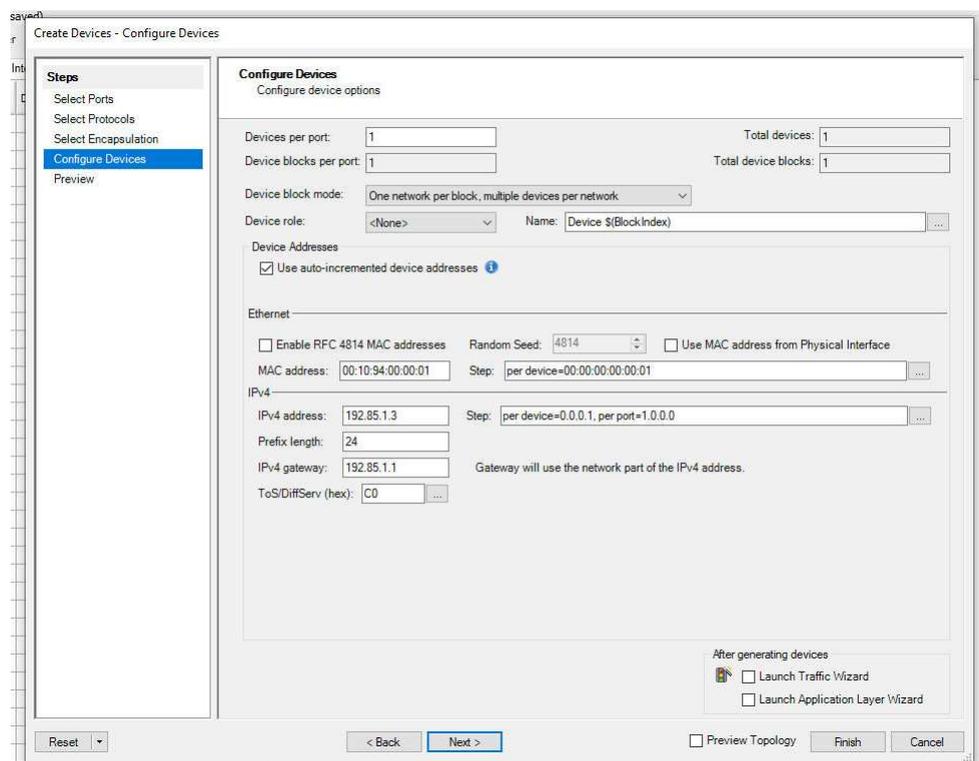


Figura 4.3: Assegnazione IP.

Si imposta la dimensione dei frame che compongono il traffico a 128 byte perché le trasmissioni a bassa latenza sono caratterizzate da pacchetti di dimensioni ridotte.

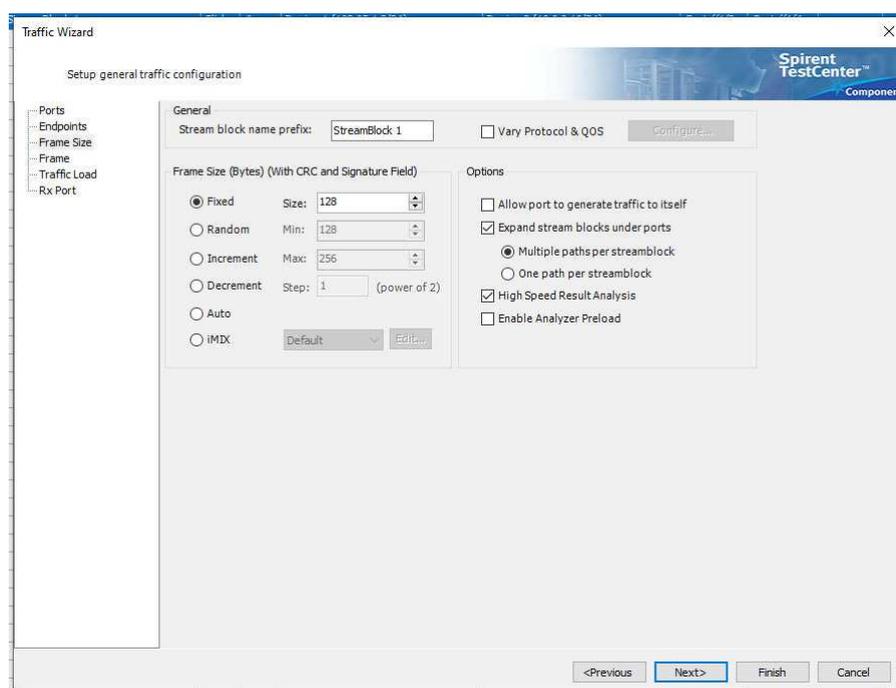


Figura 4.4: Assegnazione dimensione frame.

A questo punto, si selezionano le impostazioni relative al carico che il TestCenter deve generare su entrambe le porte. Si sceglie un carico di 1 Mbps, come si vede dalla Figura 4.5: tale scelta è data dalla volontà di evitare che un carico molto elevato generi code all'interno dei dispositivi che devono trasmettere.

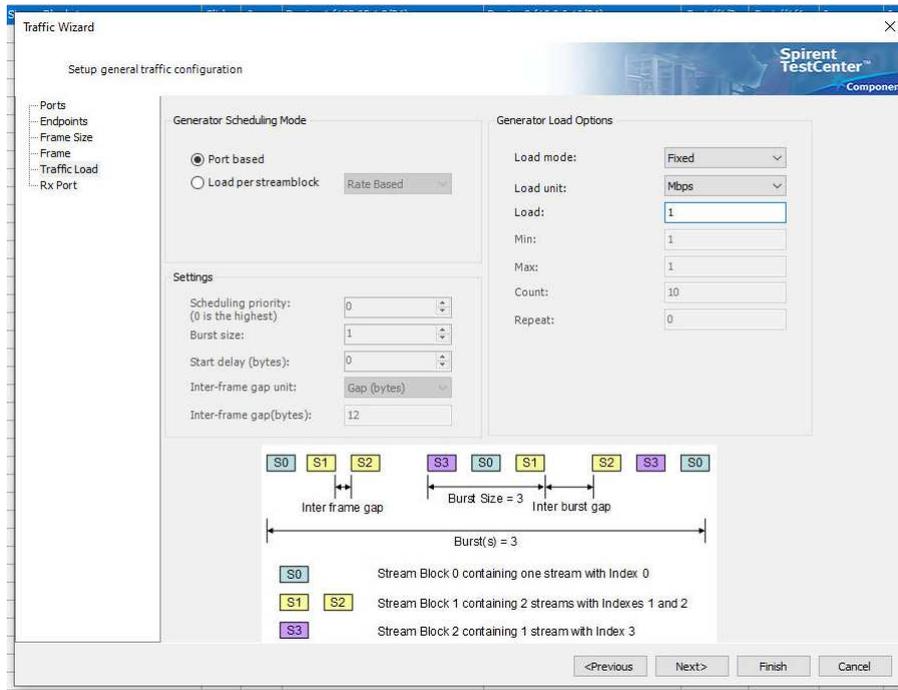


Figura 4.5: Assegnazione carico.

L'ultima operazione da fare nel TestCenter è quella di inserire un IPv4 Modifier. Si preme con il tasto destro sullo *streamblock 1-3*, quello che ha come *source* la porta 10.0.0.10 e come *destination* la porta 192.85.1.5, si va su edit, si va su *Frame* e da qui si preme con il tasto destro e si seleziona *IPv4 Modifier*. Si apre la finestra mostrata in Figura 4.6, e si inserisce alla voce *Value* l'indirizzo 10.45.0.2.

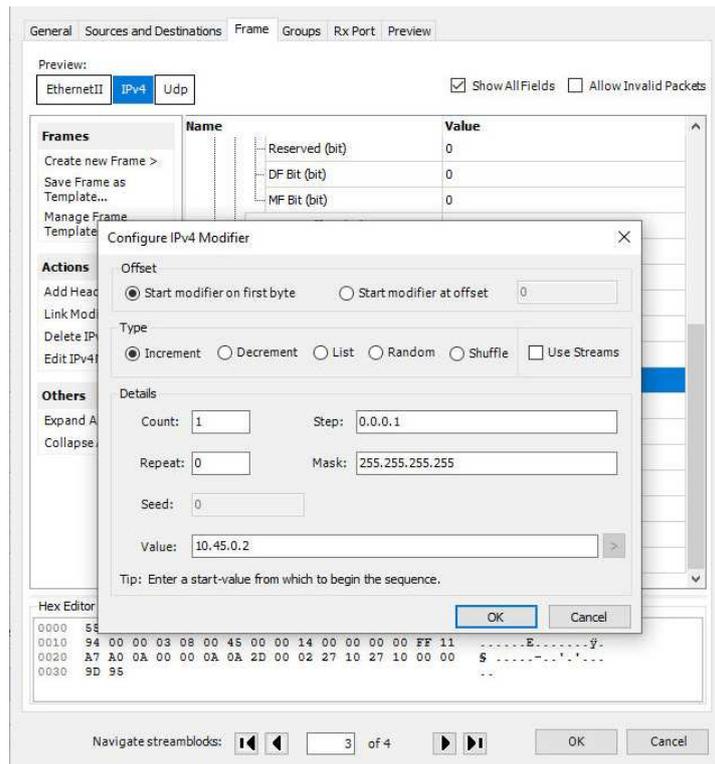


Figura 4.6: Impostazione IPv4 Modifier.

In questo modo tutti i pacchetti che arrivano alla porta 10.45.0.1, che hanno destinazione 192.85.1.5, vengono mandati all'XG9520-5GWAX, ovvero alla porta 10.45.0.2. La configurazione del TestCenter è completata, consentendo la comunicazione attraverso la porta 6 con l'XG9520-5GWAX e tramite la porta 5 con il computer.

4.1.3 Configurazione del PC

Si passa ora alla configurazione del computer, in particolare alle regole di *routing*. Per prima cosa si deve verificare che, una volta accese tutte le interfacce, si abbia questa situazione eseguendo il comando

```
$ ifconfig
```

```
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST>
mtu 1500
    inet 10.0.0.1 netmask 255.255.255.0
    broadcast 10.0.0.255
    inet6 fe81::ba4:75cb:8f6:5bf4 prefixlen 64
    scopeid 0x20<link>
    ether 70:51:5f:04:83:b0 txqueuelen 1000
    (Ethernet)
    RX packets 313728 bytes 2166288620 (2.1 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 294694 bytes 2162304256 (2.1 GB)
    TX errors 0 dropped 0 overruns 0
    carrier 0 collisions 0

enp1s0f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
mtu 9000
    inet 192.168.30.1 netmask 255.255.255.0
    broadcast 192.168.30.255
    inet6 fe80::ba4:74cb:8f6:5bf3 prefixlen 64
    scopeid 0x20<link>
    ether 80:61:5f:04:83:b0 txqueuelen 1000
    (Ethernet)
    RX packets 313728 bytes 2166288620 (2.1 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 294694 bytes 2162304256 (2.1 GB)
    TX errors 0 dropped 0 overruns 0
    carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 9100 bytes 1225050 (1.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9100 bytes 1225050 (1.2 MB)
    TX errors 0 dropped 0 overruns 0
    carrier 0 collisions 0
```

```
ogstun: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>
mtu 1400
    inet 10.45.0.1 netmask 255.255.0.0
    destination 10.45.0.1
    inet6 fe80::2e75:21c9:4530:46aa prefixlen 64
    scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
    txqueuelen 500 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 34 bytes 2384 (2.3 KB)
    TX errors 0 dropped 0 overruns 0
    carrier 0 collisions 0
```

Se non si hanno questi indirizzi associati a queste interfacce si devono andare a modificare dalle impostazioni del PC. Fatto questo, prima di andare a lanciare la rete si devono andare a modificare alcune rotte. Per prima cosa si esegue questo comando:

```
$ sudo ip route del default via 192.168.30.1 dev
enp1s0f0 proto static metric 20100
```

il quale esegue l'operazione di eliminazione di una rotta predefinita (*default route*) dalla tabella di *routing* del sistema. Poi si esegue quest'altro comando:

```
$ sudo ip route add default via 192.168.30.2 dev
enp1s0f0 proto static metric 20100
```

che invece in questo caso aggiunge una nuova rotta predefinita (*default route*) alla tabella di *routing* del sistema. In particolare questa rotta di default specifica che i pacchetti IP in uscita, che non hanno un'altra rotta specifica, devono essere inviati al *gateway* 192.168.30.2, attraverso l'interfaccia *enp1s0f0*. A questo punto eseguendo il comando

```
$ ip route
```

si deve avere questa situazione:

```
default via 192.168.30.2 dev enp1s0f0 proto static
metric 20100
10.0.0.0/24 dev enp0s31f6 proto kernel scope link
src 10.0.0.1 metric 102
10.45.0.0/16 dev ogstun proto kernel scope link
src 10.45.0.1
192.168.30.0/24 dev enp1s0f0 proto kernel scope link
src 192.168.30.1 metric 103
```

Tranne la prima, tutte le altre regole di *routing* sono state configurate in modo automatico dal sistema operativo. La seconda regola specifica che la rete 10.0.0.0/24, rappresentante la connessione tra il computer e il TestCenter, è raggiungibile tramite l'interfaccia `enp0s31f6`. La terza regola specifica che la rete 5G, caratterizzata dagli indirizzi IP della subnet 10.45.0.0/16, è raggiungibile tramite l'interfaccia `ogstun`. Un'ulteriore azione da fare prima di passare al lancio della rete 5G è eseguire il comando:

```
$ sudo iptables -I FORWARD -j ACCEPT
```

Questo comando inserisce una regola nella catena FORWARD di iptables che accetta tutto il traffico di inoltro. Ciò significa che qualsiasi pacchetto che passa attraverso il dispositivo (non destinato direttamente a esso o proveniente da esso) sarà consentito, migliorando la connettività di rete tra altri dispositivi che passano attraverso questo dispositivo. Ora si passa ad analizzare le modifiche effettuate all'eseguibile `run_5g_box_PLNM00101_x310.sh` che avvia la rete 5G. Le prime due istruzioni che vengono eseguite quando si lancia questo *script* sono

```
$ sudo sysctl -w net.ipv4.ip_forward=1
$ sudo sysctl -w net.ipv6.conf.all.forwarding=1
```

i quali abilitano l'inoltro dei pacchetti IP a livello di kernel sul sistema. In questo modo il sistema può inoltrare i pacchetti di rete da una interfaccia di rete all'altra, consentendo la comunicazione tra reti diverse. Questa operazione viene effettuata sia per i pacchetti IPv4 sia per quelli IPv6. Poi si eseguono tre istruzioni relative a tre diverse regole per la tabella NAT. La prima

```
$ sudo iptables -t nat -A PREROUTING -d 192.168.30.1/32  
-i enp1s0f0 -j NETMAP --to 127.0.1.100/32
```

configura una regola nella tabella NAT per la catena PREROUTING che mappa i pacchetti destinati all'indirizzo IP 192.168.30.1 all'indirizzo IP 127.0.1.100 quando arrivano sull'interfaccia di rete `enp1s0f0`. L'indirizzo 127.0.1.100 è quello associato all'ngap, ovvero il protocollo di comunicazione tra gNB e AMF. In altre parole, qualsiasi pacchetto che arriva sull'interfaccia `enp1s0f0` con destinazione 192.168.30.1 verrà modificato in modo che la sua nuova destinazione sia 127.0.1.100. Questo tipo di regola è tipicamente usata per fare DNAT (*Destination Network Address Translation*), che è una forma di NAT in cui l'indirizzo di destinazione dei pacchetti viene cambiato. La seconda regola

```
$ sudo iptables -t mangle -A PREROUTING -d 192.168.30.1/32  
-i enp1s0f0 -j ACCEPT
```

è relativa alla tabella "mangle" di iptables. In particolare il comando configura una regola nella tabella "mangle" per la catena PREROUTING che accetta i pacchetti destinati all'indirizzo IP 192.168.30.1 che arrivano sull'interfaccia `enp1s0f0`. In altre parole, qualsiasi pacchetto che arriva sull'interfaccia `enp1s0f0` con destinazione 192.168.30.1 verrà accettato immediatamente e permesso di continuare il suo percorso senza ulteriori manipolazioni o controlli. L'ultima regola è simile alla prima, solo che questa volta si vanno a mappare i pacchetti destinati all'indirizzo IP 127.0.1.100 verso l'indirizzo IP 192.168.30.1.

```
$ sudo iptables -t nat -A POSTROUTING -s 127.0.1.100/32  
-o enp1s0f0 -j NETMAP --to 192.168.30.1/32
```

In sintesi questo comando configura una regola che traduce tutti i pacchetti con indirizzo sorgente 127.0.1.100 inviati attraverso l'interfaccia `enp1s0f0` in modo che l'indirizzo sorgente venga modificato in 192.168.30.1. Poi si esegue un comando che cancella la rotta nel caso in cui dovesse essere predefinita dalla tabella di instradamento del sistema, che instrada il traffico tramite il *gateway* 10.0.0.1 sulla rete `enp0s31f6`.

```
$ sudo ip route del default via 10.0.0.1 dev enp0s31f6  
proto static metric 20102
```

Successivamente questo *script* contiene i percorsi di tutti i file di configurazione, inclusi quelli del gNB, che in questo caso si trovavano nella *directory* `/home/oai/jobs/launcher_open5gs`.

```
$ AMF_FILE=/home/oai/jobs/launcher_open5gs/amf.yaml  
$ AUSF_FILE=/home/oai/jobs/launcher_open5gs/ausf.yaml  
$ BSF_FILE=/home/oai/jobs/launcher_open5gs/bsf.yaml  
$ HSS_FILE=/home/oai/jobs/launcher_open5gs/hss.yaml  
$ MME_FILE=/home/oai/jobs/launcher_open5gs/mme.yaml  
$ NRF_FILE=/home/oai/jobs/launcher_open5gs/nrf.yaml  
$ NSSF_FILE=/home/oai/jobs/launcher_open5gs/nssf.yaml  
$ PCF_FILE=/home/oai/jobs/launcher_open5gs/pcf.yaml  
$ PCRF_FILE=/home/oai/jobs/launcher_open5gs/pcrf.yaml  
$ SCP_FILE=/home/oai/jobs/launcher_open5gs/scp.yaml  
$ SGWC_FILE=/home/oai/jobs/launcher_open5gs/sgwc.yaml  
$ SGWU_FILE=/home/oai/jobs/launcher_open5gs/sgwu.yaml  
$ SMF_FILE=/home/oai/jobs/launcher_open5gs/smf.yaml  
$ UDM_FILE=/home/oai/jobs/launcher_open5gs/udm.yaml  
$ UDR_FILE=/home/oai/jobs/launcher_open5gs/udr.yaml  
$ UPF_FILE=/home/oai/jobs/launcher_open5gs/upf.yaml  
$ GNB_FILE=/home/oai/jobs/launcher_open5gs/  
gnb_x310_20MHz_oneplus_8t.yaml  
# $ GNB_FILE=/home/oai/jobs/launcher_open5gs/  
gnb_rf_x310_fdd_srsUE.yaml
```

Le istruzioni successive presenti nello *script* servono a fermare, tramite il comando `stop`, e chiudere tramite il comando `pkill` tutti i processi corrispondenti.

```
$ sudo systemctl stop open5gs-mmed &
$ sudo systemctl stop open5gs-sgwcd &
$ sudo systemctl stop open5gs-smfd &
$ sudo systemctl stop open5gs-amfd &
$ sudo systemctl stop open5gs-sgwud &
$ sudo systemctl stop open5gs-upfd &
$ sudo systemctl stop open5gs-hssd &
$ sudo systemctl stop open5gs-pcrfd &
$ sudo systemctl stop open5gs-nrfd &
$ sudo systemctl stop open5gs-scpd &
$ sudo systemctl stop open5gs-seppd &
$ sudo systemctl stop open5gs-ausfd &
$ sudo systemctl stop open5gs-udmd &
$ sudo systemctl stop open5gs-pcfd &
$ sudo systemctl stop open5gs-nssfd &
$ sudo systemctl stop open5gs-bsfd &
$ sudo systemctl stop open5gs-udrd &
$ sudo systemctl stop open5gs-webui &

$ sudo pkill open5gs-nrfd &
$ sudo pkill open5gs-scpd &
$ sudo pkill open5gs-amfd &
$ sudo pkill open5gs-smfd &
$ sudo pkill open5gs-upfd &
$ sudo pkill open5gs-ausfd &
$ sudo pkill open5gs-udmd &
$ sudo pkill open5gs-pcfd &
$ sudo pkill open5gs-nssfd &
$ sudo pkill open5gs-bsfd &
$ sudo pkill open5gs-mmed &
$ sudo pkill open5gs-sgwcd &
$ sudo pkill open5gs-sgwud &
$ sudo pkill open5gs-hssd &
$ sudo pkill open5gs-pcrfd &
```

Successivamente si eseguono gli eseguibili relativi alle funzioni di rete in modo da avviarle contemporaneamente. Per fare ciò si passano i percorsi dei file di configurazione che sono stati indicati precedentemente nello *script*.

```
$ sudo open5gs-nrfd -c $NRF_FILE &
$ sleep 1s
$ sudo open5gs-scpd -c $SCP_FILE &
$ sleep 1s
$ sudo open5gs-amfd -c $AMF_FILE &
$ sleep 1s
$ sudo open5gs-smfd -c $SMF_FILE &
$ sleep 1s
$ sudo open5gs-upfd -c $UPF_FILE &
$ sleep 1s
$ sudo open5gs-ausfd -c $AUSF_FILE &
$ sleep 1s
$ sudo open5gs-udmd -c $UDM_FILE &
$ sleep 1s
$ sudo open5gs-pcfd -c $PCF_FILE &
$ sleep 1s
$ sudo open5gs-nssf -c $NSSF_FILE &
$ sleep 1s
$ sudo open5gs-bsfd -c $BSF_FILE &
$ sleep 1s
$ sudo open5gs-udrd -c $UDR_FILE &
$ sleep 1s
$ sudo open5gs-mmed -c $MME_FILE &
$ sleep 1s
$ sudo open5gs-sgwcd -c $SGWC_FILE &
$ sleep 1s
$ sudo open5gs-sgwud -c $SGWU_FILE &
$ sleep 1s
$ sudo open5gs-hssd -c $HSS_FILE &
$ sleep 1s
$ sudo open5gs-pcrfd -c $PCRF_FILE &
$ sleep 1s
```

Poi viene riavviata l'interfaccia WebUI tramite:

```
$ sudo systemctl stop open5gs-webui
$ sudo systemctl restart open5gs-webui
$ sleep 5s
```

A seguire vengono eseguiti due comandi per impostare la dimensione massima del buffer di memoria in uscita e in ingresso per i *socket* di rete a 24.912.805 byte. Questo può essere utile per ottimizzare le prestazioni di rete in scenari

ad alto utilizzo della larghezza di banda. Questo viene fatto perché aumentare la dimensione massima del buffer di memoria in uscita e in ingresso può ridurre la latenza e migliorare il *throughput*.

```
$ sudo sysctl -w net.core.wmem_max=24912805
$ sudo sysctl -w net.core.rmem_max=24912805
```

In seguito si configura l'interfaccia di rete `enp1s0f0`, ovvero quella relativa alla connessione tra PC e radio, per utilizzare un'unità di trasmissione massima (MTU) di 9000 byte. La MTU è la dimensione massima di un singolo pacchetto di dati che può essere trasmesso sulla rete senza dover essere frammentato. Un MTU di 9000 byte è tipico per i cosiddetti "jumbo frames". Questi sono pacchetti di rete più grandi rispetto alla dimensione standard di 1500 byte utilizzata nella maggior parte delle reti Ethernet. Utilizzare jumbo frames può migliorare le prestazioni di rete riducendo l'overhead di pacchetti, aumentando l'efficienza della trasmissione dei dati e riducendo il carico della CPU.

```
$ sudo ifconfig enp1s0f0 mtu 9000
```

Infine viene avviata la cella tramite l'eseguibile `gNB` fornito da SRSRAN Project, utilizzando il seguente comando

```
$ sudo gnb -c $GNB_FILE
```

In questo modo lo *script* lancia la cella e la rete 5G dovrebbe essere funzionante.

4.1.4 Configurazione ATOS

L'ultima cosa da fare è controllare la configurazione dell'XG9520-5GWAX. Per farlo è necessario collegare il dispositivo al PC tramite seriale, digitare nel terminale

```
$ sudo gtkterm
```

in questo modo si accede al *framework* ATOS. Fatto ciò per controllare la configurazione basterà mandare il comando

```
conf
```

dal terminale e verificare che si abbia questa situazione

```
set system loglevel 5
set system name ATOSNT
set system console-baudrate 115200
set system scroll-line 255
set wlan0 off
set wlan0 RF-mode ax-b-g-n
set wlan0 ssid ATOS-NT:wlan0
set wlan0 nick ATOS-NT:wlan0-NICK
set wlan1 off
set wlan1 RF-mode ax-ac-an
set wlan1 ssid ATOS-NT:wlan1
set wlan1 nick ATOS-NT:wlan1-NICK
set wlan1 chan-width 160
set wwan loglevel 5
set wwan wwan0 loglevel 5
set wwan wwan0 type network-ifc
set wwan wwan0 apn internet
add interfaces IFC eth0-1 eth0-1
add interfaces IFC wwan0 wwan0
add interfaces IFC eth1 eth1
add interfaces IFC eth0-2 eth0-2
add interfaces IFC eth0-3 eth0-3
add interfaces IFC eth0-4 eth0-4
set interfaces eth0-1 ip address 192.168.1.1/24
set interfaces eth1 ip address 192.85.1.1/24
set security off
add ip route 0.0.0.0 0.0.0.0 wwan0 1
add napt IFC wwan0
add napt wwan0 PROXY 10000 192.85.1.5 10000 udp
set napt wwan0 on
set syslog loglevel 5
```

In particolare si deve verificare la presenza di tre comandi.

```
set wwan wwan0 apn internet
```

il quale permette connessioni solo se la SIM ha quell'APN, in questo caso internet. APN che viene impostato nell'interfaccia WebUI. L'altro comando importante è

```
set interfaces eth1 ip address 192.85.1.1/24
```

che assegna l'interfaccia `eth1` l'indirizzo ip `192.85.1.1` con *netmask* `255.255.255.0`. Infine verificare che sia presente

```
add napt wwan0 PROXY 10000 192.85.1.5 10000 udp
```

che configura una regola di *Network Address Port Translation* (NAPT) sull'interfaccia di rete chiamata `wwan0`, ovvero quella relativa alla rete 5G. NAPT è una forma di *Network Address Translation* (NAT) che permette di tradurre sia indirizzi IP che numeri di porta. Ovvero questo comando crea una regola di traduzione degli indirizzi di rete e delle porte (NAPT) che inoltra il traffico UDP in arrivo sulla porta 10000 dell'interfaccia `wwan0` all'indirizzo IP interno `192.85.1.5` sulla porta 10000. Eseguite tutte queste modifiche è possibile tornare al TestCenter e avviare del traffico per poter effettuare le misurazioni di latenza.

4.2 Configurazione rete 5G per slicing

Seguendo lo stesso ordine usato per le modifiche effettuate per i test di latenza, partiamo dai file di configurazione. In entrambi i file i cambiamenti effettuati

sono gli stessi di quelli effettuati per la latenza, quindi ci si può riferire al capitolo 4.1.

4.2.1 Configurazione delle SIM

Per lo *slicing* sono necessarie due SIM, che vanno entrambe configurate tramite *UICC/SIM programming*. Per semplicità si è usata la stessa SIM usata per la latenza, quindi anche in questo caso per la configurazione ci si può riferire al capitolo 1.1.1, mentre la seconda è stata così configurata:

```
$ sudo ./program_uicc --port /dev/ttyUSB0
--adm 10803136
--opc 504f20634f6320504f50206363500a4f
--iccid 89860061100000000021
--isdn 00000021
--key 6874736969202073796d4b2079650a73
--spn "Open5GS"
--imsi 001010100001117
```

Si passa quindi alla registrazione delle SIM sull'interfaccia WebUI. Si accede all'indirizzo <https://localhost:3000>, e per la prima SIM si inseriscono questi valori:

```
IMSI:001010100001121
AMF:8000
SUBSCRIBER KEY:6874736969202073796d4b2079650a73
USIM Type: OPc
Operator Key: 504f20634f6320504f50206363500a4f
SST*:2 SD:000001
APN:internet Type*:ipv4
5QI/QCI*:9
ARP Priority Level (1-15)*:8
```

Per la seconda invece:

```
IMSI:001010100001117
AMF:8000
SUBSCRIBER KEY:6874736969202073796d4b2079650a73
USIM Type: OPc
Operator Key: 504f20634f6320504f50206363500a4f
SST*:1 SD:000001
APN:voce Type*:ipv4
5QI/QCI*:7
ARP Priority Level (1-15)*:1
```

Si può notare che oltre al diverso codice IMSI che serve ad identificare le due diverse SIM, una è stata dedicata per il traffico internet caratterizzato da una priorità minore, mentre l'altra è stata usata per il traffico voce, con una priorità maggiore. Ovviamente le SIM hanno due valori diversi di SST per lavorare su due slice differenti.

Edit Subscriber

Subscriber Configuration

IMSI*
001010100001121

[+](#)

Subscriber Key (K)*: 6874736969202073796d4b2079650a73
Authentication Management Field (AMF)*: 8000

USIM Type: OPc
Operator Key (OPc/OP)*: 504f20634f6320504f50206363500a4f

UE-AMBR Downlink*: 1 Unit: Gbps
UE-AMBR Uplink*: 1 Unit: Gbps

Slice Configurations

SST*
 1 2 3 4

SD: 000001 Default S-NSSAI [×](#)

Session Configurations

DNN/APN*: internet Type*: IPv4 [×](#)

5QI/QCI*: 9

ARP Priority Level (1-15)*: 8

Capability*: Disabled Vulnerability*: Disabled

[CANCEL](#) [SAVE](#)

Figura 4.7: Interfaccia WebUI per configurare SIM internet.

Edit Subscriber

001010100001117

+

Subscriber Key (K)* **Authentication Management Field (AMF)***

6874736969202073796d4b2079650a73 8000

USIM Type **Operator Key (OPc/OP)***

OPc 504f20634f6320504f50206363500a4f

UE-AMBR Downlink* **Unit** **UE-AMBR Uplink*** **Unit**

1 Gbps 1 Gbps

Slice Configurations

SST* **SD** Default S-NSSAI

1 2 3 4 000001 ✕

Session Configurations

DNN/APN* **Type*** ✕

voce IPv4

5QI/QCI*

7

ARP Priority Level (1-15)*

1

Capability* **Vulnerability***

Disabled Disabled

[CANCEL](#) [SAVE](#)

Figura 4.8: Interfaccia WebUI per configurare SIM voce.

4.2.2 Configurazione del TestCenter

Per i test relativi allo *slicing* il banco di prova è stato così impostato:

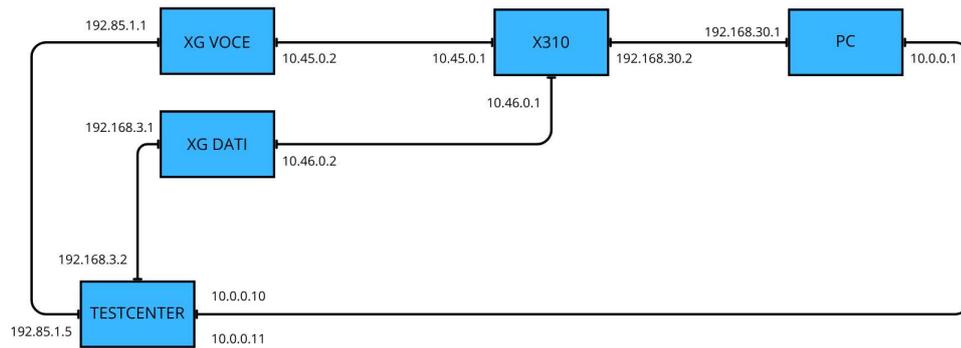


Figura 4.9: Banco test slicing.

I dispositivi utilizzati sono stati gli stessi del banco per i test di latenza, con l'aggiunta di un secondo XG9520-5GWAX. Anche in questo caso la connessione tra il PC e la radio avviene tramite SFP Ethernet a 10 Gigabit. I due XG9520-5GWAX e il PC vengono collegati al TestCenter tramite Ethernet alle porte 3, 5 e 6. Si passa alla configurazione del TestCenter, in particolare delle porte e degli *streamblock*. Si aggiunge un nuovo Device selezionando la porta 5 e si assegnano i parametri: indirizzo IP 10.0.0.10, la *netmask* 255.255.255.0 e il *gateway* 10.0.0.1. Allo stesso modo, selezionando sempre la porta 5, si inserisce un nuovo Device, questa volta con questi parametri: indirizzo IP 10.0.0.11, la *netmask* 255.255.255.0 e il *gateway* 10.0.0.1. Si nota come il *gateway* sia lo stesso in quanto entrambi questi Device fanno riferimento al PC. In sostanza in questo modo il traffico generato dal TestCenter viaggia sulle stesse porte e quindi sullo stesso cavo ma con due indirizzi diversi. Per quanto riguarda le porte 3 e 6 si procede alla stessa maniera assegnando alla porta 3 indirizzo IP 192.168.3.2, *netmask* 255.255.255.0 e *gateway* 192.168.3.1, mentre alla porta 6 indirizzo IP 192.85.1.5, la *netmask* 255.255.255.0 e il *gateway* 192.85.1.1. Per semplicità si è deciso di lasciare la porta 6 con gli stessi indirizzi usati per i test di latenza. Per quanto riguarda la dimensione dei frame che compongono il traffico, si è impostato 82 byte per il traffico voce, mentre per il traffico dati 1000 byte. Il carico è stato inizialmente impostato ad 1 Mbps per il traffico

voce, mentre a 5 Mbps per quello dati. Nel caso dello *slicing* si devono aggiungere due IPv4 Modifier, uno nello *streamblock* 102-4, con *source* l'indirizzo 10.0.0.10 e *destination* 192.85.1.5, impostando come *Value* l'indirizzo 10.45.0.2, esattamente come fatto per la latenza, mentre l'altro nello *streamblock* 97-4, con *source* l'indirizzo 10.0.0.11 e *destination* 192.168.3.2, impostando come *Value* l'indirizzo 10.46.0.2. In questo modo tutti i pacchetti che arrivano alla porta 10.45.0.1, che hanno destinazione 192.85.1.5, vengono mandati al primo XG9520-5GWAX, ovvero alla porta 10.45.0.2. Mentre tutti i pacchetti che arrivano alla porta 10.46.0.1, che hanno destinazione 192.168.3.2, vengono mandati al secondo XG9520-5GWAX, ovvero alla porta 10.46.0.2.

4.2.3 Configurazione del PC

Per quanto riguarda le modifiche effettuate ai vari file di configurazione si è seguito il tutorial [22]. Le prime modifiche che si devono effettuare riguardano i file della radio. Si parte dal file `amf.yaml`, si devono aggiungere le due *slice* configurate nella WebUI:

```
*****
plmn_support:
  - plmn_id:
      mcc: 001
      mnc: 01
      s_nssai:
        - sst: 1
+         sd: 00001
+         - sst: 2
+         sd: 00001
*****
```

Stessa cosa per il file `nssf.yaml`:

```
*****
nssf:
  sbi:
    server:
      - address: 127.0.0.14
        port: 7777
    client:
      scp:
        - uri: http://127.0.0.200:7777
      nsi:
        - uri: http://127.0.0.10:7777
        s_nssai:
          sst: 1
+         sd: 00001
+         sst: 2
+         sd: 00001
*****
```

Dopodiché si deve creare una copia dei file `upf.yaml` e `smf.yaml` che qui sono stati chiamati `upf1.yaml`, `upf2.yaml`, `smf1.yaml` e `smf2.yaml`. Nel file `smf1.yaml` le uniche modifiche da fare sono:

```
*****
session:
  - subnet: 10.45.0.1/16
-   - subnet: 2001:db8:cafe::1/48
+   dnn: voce
+   dev: ogstun1
*****
```

Il file `smf2.yaml` invece subisce diverse modifiche dovute principalmente al fatto che tutti gli indirizzi IP fanno riferimento ad una Core Network virtuale diversa rispetto a quella dell'`smf1`. In particolare si ha:

```

*****
smf:
  sbi:
    server:
-     - address: 127.0.0.4
+     - address: 127.0.0.24
      port: 7777
    client:
      scp:
        - uri: http://127.0.0.200:7777
  pfcp:
    server:
-     - address: 127.0.0.4
+     - address: 127.0.0.24
    client:
      upf:
-     - address: 127.0.0.7
+     - address: 127.0.0.17
  gtpc:
    server:
-     - address: 127.0.0.4
+     - address: 127.0.0.24
  gtpu:
    server:
-     - address: 127.0.0.4
+     - address: 127.0.0.24
  metrics:
    server:
-     - address: 127.0.0.4
+     - address: 127.0.0.24
      port: 9090
  session:
-     - subnet: 10.45.0.1/16
+     - subnet: 10.46.0.1/16
-     - subnet: 2001:db8:cafe::1/48
+     dnn: internet
+     dev: ogstun2
*****

```

Si passa al file `upf1.yaml`, in cui le modifiche da fare, rispetto a quello utilizzato per i test di latenza, sono:

```
*****
gtpu:
  server:
    - address: 127.0.1.100
  session:
    - subnet: 10.45.0.1/16
-   - subnet: 2001:db8:cafe::1/48
+   dnn: voce
+   dev: ogstun1
*****
```

Anche in questo caso le modifiche per il file `upf2.yaml` sono dovute alla Core Network diversa.

```
*****
upf:
  pfcps:
    server:
-     - address: 127.0.0.7
+     - address: 127.0.0.17
    client:
  gtpu:
    server:
-     - address: 127.0.1.100
+     - address: 127.0.1.101
    session:
-     - subnet: 10.45.0.1/16
+     - subnet: 10.46.0.1/16
-     - subnet: 2001:db8:cafe::1/48
+     dnn: internet
+     dev: ogstun2
  metrics:
    server:
-     - address: 127.0.0.7
+     - address: 127.0.0.17
      port: 9090
*****
```

In ultimo si deve creare un nuovo file chiamato `slicing.yaml`, al cui interno si devono andare a scrivere le *slice* usate, in questo caso:

```

*****
+   slicing:
+     - sst: 1
+     sd: 000001
+     - sst: 2
+     sd: 000001
*****

```

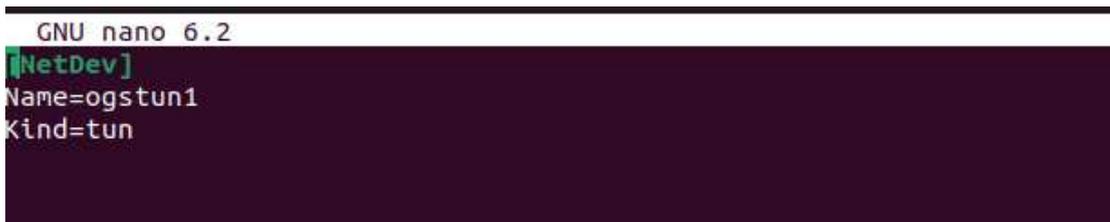
Si nota come una delle modifiche ricorrenti riguarda l'interfaccia `ogstun`. Questa è un'interfaccia virtuale tra pc e radio e si riferisce proprio alla rete 5G. Per prima cosa si devono andare a modificare i file `99-open5gs.netdev` e `99-open5gs.network` all'interno della *directory* `/etc/systemd/network`. Per farlo ci si deve spostare prima nella *directory* e poi aprire il file per modificarlo attraverso i comandi:

```

$ cd /etc/systemd/network
$ sudo nano 99-open5gs.netdev

```

A questo punto si deve andare a modificare il nome dell'interfaccia `tun` inserendo `ogstun1`, in questo modo:



```

GNU nano 6.2
[NetDev]
Name=ogstun1
Kind=tun

```

Figura 4.10: Modifiche al file `99-open5gs.netdev`.

Lo stesso procedimento si deve fare con per l'altro file:

```

$ cd /etc/systemd/network
$ sudo nano 99-open5gs.network

```

Anche qui si deve modificare solamente il nome dell'interfaccia.

```
GNU nano 6.2
[Match]
Name=ogstun1

[Network]
Address=10.45.0.1/16

[Link]
MTUBytes=1400
RequiredForOnline=false
```

Figura 4.11: Modifiche al file 99-open5gs.network.

A questo punto per creare la seconda interfaccia virtuale `ogstun2` si deve andare a creare una copia di questi due file, rinominandoli all'interno della stessa *directory*. In questo caso sono stati rinominati `98-open5gs.netdev` e `98-open5gs.network`.

```
$ cd /etc/systemd/network
$ sudo ./99-open5gs.netdev ./98-open5gs.netdev
$ sudo ./99-open5gs.network ./98-open5gs.network
```

A questo punto si modifica il primo, andando a cambiare il nome dell'interfaccia in `ogstun2`

```
$ cd /etc/systemd/network
$ sudo nano 98-open5gs.netdev
```

```
GNU nano 6.2
[NetDev]
Name=ogstun2
Kind=tun
```

Figura 4.12: Modifiche al file 98-open5gs.netdev.

Il file `98-open5gs.network` va modificato cambiando sia il nome dell'interfaccia sia cambiando l'indirizzo IP.

```
$ cd /etc/systemd/network
$ sudo nano 98-open5gs.network
```

```
GNU nano 6.2
[Match]
Name=ogstun2

[Network]
Address=10.46.0.1/16

[Link]
MTUBytes=1400
RequiredForOnline=false
```

Figura 4.13: Modifiche al file 98-open5gs.network.

Fatte queste modifiche è bene riavviare le interfacce Linux e l'intero sistema operativo.

```
$ sudo systemctl restart NetworkManager.service
$ sudo reboot
```

Per quanto riguarda le modifiche ai file `gnb_x310_20MHz_oneplus_8t.yml` e `gnb_rf_x310_fdd_srsUE.yml` rimangono esattamente le stesse di quelle fatte per i test di latenza, quindi si possono vedere al capitolo 4.1. Anche per le modifiche allo *script* che lancia la rete, `run_5g_box_PLNM00101_x310.sh`, si possono seguire i passaggi fatti precedentemente, con alcune piccole differenze. Infatti quando si vanno a scrivere i percorsi di tutti i file di configurazione, oltre a cambiare il percorso della *directory*, in questo caso `/home/oai/jobs/slicing_open5gs`, si devono andare ad aggiungere i due percorsi relativi all'`upf2` e all'`smf2`, ottenendo:

```
$ AMF_FILE=/home/oai/jobs/slicing_open5gs/amf.yaml
$ AUSF_FILE=/home/oai/jobs/slicing_open5gs/ausf.yaml
$ BSF_FILE=/home/oai/jobs/slicing_open5gs/bsf.yaml
$ HSS_FILE=/home/oai/jobs/slicing_open5gs/hss.yaml
$ MME_FILE=/home/oai/jobs/slicing_open5gs/mme.yaml
$ NRF_FILE=/home/oai/jobs/slicing_open5gs/nrf.yaml
$ NSSF_FILE=/home/oai/jobs/slicing_open5gs/nssf.yaml
$ PCF_FILE=/home/oai/jobs/slicing_open5gs/pcf.yaml
$ PCRF_FILE=/home/oai/jobs/slicing_open5gs/pcrf.yaml
$ SCP_FILE=/home/oai/jobs/slicing_open5gs/scp.yaml
$ SGWC_FILE=/home/oai/jobs/slicing_open5gs/sgwc.yaml
$ SGWU_FILE=/home/oai/jobs/slicing_open5gs/sgwu.yaml
$ SMF1_FILE=/home/oai/jobs/launcher_open5gs/smf1.yaml
$ SMF2_FILE=/home/oai/jobs/launcher_open5gs/smf2.yaml
$ UDM_FILE=/home/oai/jobs/slicing_open5gs/udm.yaml
$ UDR_FILE=/home/oai/jobs/slicing_open5gs/udr.yaml
$ UPF1_FILE=/home/oai/jobs/slicing_open5gs/upf1.yaml
$ UPF2_FILE=/home/oai/jobs/slicing_open5gs/upf2.yaml
$ GNB_FILE=/home/oai/jobs/slicing_open5gs/
gnb_x310_20MHz_oneplus_8t.yaml
# $ GNB_FILE=/home/oai/jobs/slicing_open5gs/
gnb_rf_x310_fdd_srsUE.yaml
```

Allo stesso modo quando si passano i percorsi dei file di configurazione appena indicati per eseguirli, si dovrebbero eseguire due upf e due smf:

```
$ sudo open5gs-nrfd -c $NRF_FILE &
$ sleep 1s
$ sudo open5gs-scpd -c $SCP_FILE &
$ sleep 1s
$ sudo open5gs-amfd -c $AMF_FILE &
$ sleep 1s
$ sudo open5gs-smfd -c $SMF1_FILE &
$ sleep 1s
$ sudo open5gs-smfd -c $SMF2_FILE &
$ sleep 1s
$ sudo open5gs-upfd -c $UPF1_FILE &
$ sleep 1s
$ sudo open5gs-upfd -c $UPF2_FILE &
$ sleep 1s
$ sudo open5gs-ausfd -c $AUSF_FILE &
$ sleep 1s
$ sudo open5gs-udmd -c $UDM_FILE &
$ sleep 1s
$ sudo open5gs-pcfd -c $PCF_FILE &
$ sleep 1s
$ sudo open5gs-nssf -c $NSSF_FILE &
$ sleep 1s
$ sudo open5gs-bsfd -c $BSF_FILE &
$ sleep 1s
$ sudo open5gs-udr -c $UDR_FILE &
$ sleep 1s
$ sudo open5gs-mmed -c $MME_FILE &
$ sleep 1s
$ sudo open5gs-sgwcd -c $SGWC_FILE &
$ sleep 1s
$ sudo open5gs-sgwud -c $SGWU_FILE &
$ sleep 1s
$ sudo open5gs-hssd -c $HSS_FILE &
$ sleep 1s
$ sudo open5gs-pcrfd -c $PCRF_FILE &
$ sleep 1s
```

L'ultima differenza si ha quando si avvia la cella, in quanto oltre al percorso relativo al file gNB, si deve passare anche il percorso del file contenente le slice configurate:

```
$ sudo gnb -c $GNB_FILE -c $SLICING_FILE &
```

Prima di passare alle configurazioni di ATOS, si deve verificare che le rotte siano corrette. Anche qui si verifica che accese tutte le interfacce si abbia questa situazione:

```
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST>
mtu 1500
    inet 10.0.0.1 netmask 255.255.255.0
    broadcast 10.0.0.255
    inet6 fe81::ba4:75cb:8f6:5bf4 prefixlen 64
    scopeid 0x20<link>
    ether 70:51:5f:04:83:b0 txqueuelen 1000
    (Ethernet)
    RX packets 313728 bytes 2166288620 (2.1 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 294694 bytes 2162304256 (2.1 GB)
    TX errors 0 dropped 0 overruns 0
    carrier 0 collisions 0

enp1s0f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
mtu 9000
    inet 192.168.30.1 netmask 255.255.255.0
    broadcast 192.168.30.255
    inet6 fe80::ba4:74cb:8f6:5bf3 prefixlen 64
    scopeid 0x20<link>
    ether 80:61:5f:04:83:b0 txqueuelen 1000
    (Ethernet)
    RX packets 313728 bytes 2166288620 (2.1 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 294694 bytes 2162304256 (2.1 GB)
    TX errors 0 dropped 0 overruns 0
    carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 9100 bytes 1225050 (1.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9100 bytes 1225050 (1.2 MB)
    TX errors 0 dropped 0 overruns 0
    carrier 0 collisions 0
```

```
ogstun1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>
mtu 1400
  inet 10.45.0.1 netmask 255.255.0.0
  destination 10.45.0.1
  inet6 fe80::3330:fcf2:2ee8:d6fd
  prefixlen 64 scopeid 0x20<link>
  unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
  txqueuelen 500 (UNSPEC)
  RX packets 3 bytes 180 (180.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 22 bytes 1484 (1.4 KB)
  TX errors 0 dropped 0 overruns 0
  carrier 0 collisions 0

ogstun2: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>
mtu 1400
  inet 10.46.0.1 netmask 255.255.0.0
  destination 10.46.0.1
  inet6 fe80::e8f2:c56e:c24d:8650
  prefixlen 64 scopeid 0x20<link>
  unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
  txqueuelen 500 (UNSPEC)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 19 bytes 1304 (1.3 KB)
  TX errors 0 dropped 0 overruns 0
  carrier 0 collisions 0
```

Si passa alla configurazione delle rotte. Si può anche in questo caso seguire il tutorial relativo alla latenza capitolo 1.1.3, ottenendo perciò queste rotte:

```
default via 192.168.30.2 dev enp1s0f0 proto static
metric 20100
10.0.0.0/24 dev enp0s31f6 proto kernel scope link
src 10.0.0.1 metric 102
10.45.0.0/16 dev ogstun1 proto kernel scope link
src 10.45.0.1
10.46.0.0/16 dev ogstun2 proto kernel scope link
src 10.46.0.1
192.168.30.0/24 dev enp1s0f0 proto kernel scope link
src 192.168.30.1 metric 101
```

4.2.4 Configurazioni ATOS

Infine i due XG9520-5GWA vanno configurati nel seguente modo. In quello in cui si inserisce la SIM relativa al traffico dati, ovvero la SIM con IMSI: 001010100001121, si deve avere questa configurazione:

```
set system loglevel 5
set system name ATOSNT
set system console-baudrate 115200
set system scroll-line 255
set wlan0 off
set wlan0 RF-mode ax-b-g-n
set wlan0 ssid ATOS-NT:wlan0
set wlan0 nick ATOS-NT:wlan0-NICK
set wlan1 off
set wlan1 RF-mode ax-ac-an
set wlan1 ssid ATOS-NT:wlan1
set wlan1 nick ATOS-NT:wlan1-NICK
set wlan1 chan-width 160
set wwan loglevel 5
set wwan wwan0 loglevel 5
set wwan wwan0 type network-ifc
set wwan wwan0 apn internet
add interfaces IFC eth0-1 eth0-1
add interfaces IFC wwan0 wwan0
add interfaces IFC eth1 eth1
add interfaces IFC eth0-2 eth0-2
add interfaces IFC eth0-3 eth0-3
add interfaces IFC eth0-4 eth0-4
set interfaces eth0-1 ip address 192.168.1.1/24
set interfaces eth1 ip address 192.168.3.1/24
set security off
add ip route 0.0.0.0 0.0.0.0 wwan0 1
add napt IFC wwan0
add napt wwan0 PROXY 10000 192.168.3.2 10000 udp
set napt wwan0 on
set syslog loglevel 5
```

Come per la latenza è importante verificare che l'APN, l'interfaccia eth1 e il PROXY siano settati in modo corretto. Invece il secondo XG9520-5GWA, con la SIM per il traffico voce, ovvero quella con l'IMSI: 001010100001117, deve

avere questa configurazione:

```
set system loglevel 5
set system name ATOSNT
set system console-baudrate 115200
set system scroll-line 255
set wlan0 off
set wlan0 RF-mode ax-b-g-n
set wlan0 ssid ATOS-NT:wlan0
set wlan0 nick ATOS-NT:wlan0-NICK
set wlan1 off
set wlan1 RF-mode ax-ac-an
set wlan1 ssid ATOS-NT:wlan1
set wlan1 nick ATOS-NT:wlan1-NICK
set wlan1 chan-width 160
set wwan loglevel 5
set wwan wwan0 loglevel 5
set wwan wwan0 type network-ifc
set wwan wwan0 apn voce
add interfaces IFC eth0-1 eth0-1
add interfaces IFC wwan0 wwan0
add interfaces IFC eth1 eth1
add interfaces IFC eth0-2 eth0-2
add interfaces IFC eth0-3 eth0-3
add interfaces IFC eth0-4 eth0-4
set interfaces eth0-1 ip address 192.168.1.1/24
set interfaces eth1 ip address 192.85.1.1/24
set security off
add ip route 0.0.0.0 0.0.0.0 wwan0 1
add napt IFC wwan0
add napt wwan0 PROXY 10000 192.85.1.5 10000 udp
set napt wwan0 on
set syslog loglevel 5
```

Come prima verificare che l'APN, l'interfaccia `eth1` e il PROXY siano settati a dovere. Dopo aver effettuato tutte queste operazioni è possibile avviare il traffico dal TestCenter per effettuare i test relativi allo slicing.

Capitolo 5

Test latenza

Dopo aver configurato i setup sperimentali, sono state condotte misurazioni per valutare le prestazioni delle due tecnologie in termini di latenza. Tutti i test sono stati eseguiti nella camera anecoica messa a disposizione da A TLC S.r.l. Ogni prova ha avuto una durata di due minuti per garantire che i valori medi di latenza riflettessero accuratamente i valori istantanei in condizioni stabili. I risultati delle misurazioni saranno presentati separatamente per ciascuna tecnologia: inizialmente i risultati ottenuti in TDD, seguiti da quelli in FDD.

5.1 Modalità TDD

Per primi sono stati effettuati i test in banda n78 (3500MHz), utilizzando il file di configurazione `gnb_x310_20MHz_oneplus_8t.yml`. Questo file specifico è stato scelto per configurare la radio con parametri adatti alla banda n78, una frequenza ampiamente utilizzata per le comunicazioni 5G grazie alla sua capacità di supportare alte velocità di trasmissione dati e una buona copertura.

Durante questi test, i valori della larghezza di canale e della spaziatura tra le sottoportanti sono stati mantenuti costanti, rispettivamente a 20 MHz e

30 KHz. Questo approccio ha permesso di focalizzare l'attenzione sull'analisi dell'influenza del *Modulation and Coding Scheme*(MCS) sulle prestazioni della rete. La larghezza di canale di 20 MHz è stata scelta per rappresentare una configurazione tipica nelle implementazioni 5G, mentre la spaziatura di 30 KHz tra le sottoportanti è stata selezionata per ottimizzare l'efficienza dello spettro e ridurre le interferenze intersottoportanti.

Sono stati condotti numerosi test variando i valori dell'MCS. L'MCS determina la combinazione di modulazione e codifica usata per trasmettere i dati, influenzando direttamente la velocità di trasmissione e la robustezza della comunicazione. Testare diversi schemi di modulazione e codifica è cruciale per identificare le configurazioni che offrono il miglior compromesso tra velocità e affidabilità, soprattutto in un ambiente di rete 5G dove le condizioni possono variare rapidamente.

Per ogni valore di MCS testato, sono stati anche variati i guadagni del *front end* sia in trasmissione sia in ricezione. Questo è stato fatto per garantire una comunicazione pulita e priva di errori, minimizzando la perdita di pacchetti. I guadagni *front end* influenzano la potenza del segnale trasmesso e ricevuto, e la loro regolazione è fondamentale per ottimizzare la qualità del collegamento radio.

I risultati ottenuti sono stati confrontati con quelli della radio B210. La radio B210, un dispositivo SDR (Software Defined Radio) popolare per test e sperimentazioni, è stata utilizzata come termine di paragone per valutare le prestazioni relative della configurazione testata con la radio X310. Questo confronto ha permesso di mettere in evidenza eventuali differenze di prestazioni, sia in termini di latenza che di qualità della comunicazione.

MCS 0

La prima prova è stata eseguita selezionando gli MCS 1 per il PDSCH e 0 per il PUSCH, corrispondenti a una modulazione 64-QAM. I valori del guadagno front end sono stati messi in ricezioni a 0 e in trasmissione 20.

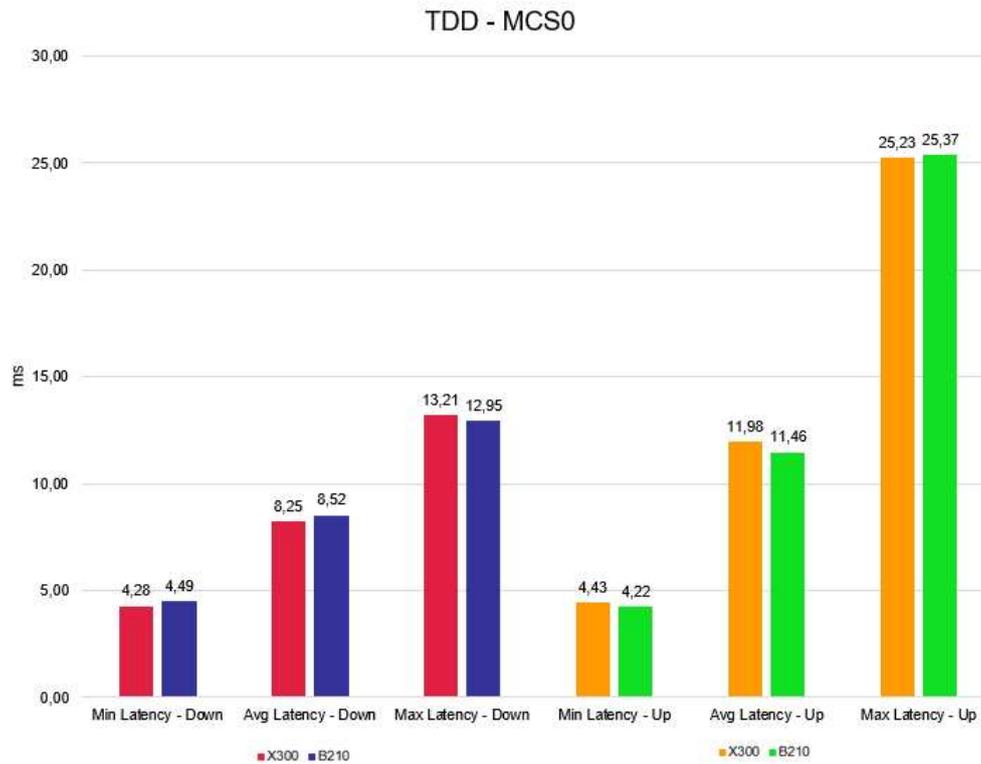


Figura 5.1: Confronto latenza in banda n78, MCS PDSCH 1, MCS PUSCH 0.

pci	rnti	DL								UL								
		cqi	ri	mcs	brate	ok	nok	(%)	dL_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr
1	4604	15	1	1	841k	933	0	0%	114	22.2	1	806k	480	0	0%	2.01k	0us	20
1	4604	15	1	1	835k	933	0	0%	0	22.3	1	811k	505	0	0%	1.45k	0us	21
1	4604	15	1	1	838k	938	0	0%	114	22.3	1	815k	488	0	0%	535	0us	21
1	4604	15	1	1	838k	934	0	0%	0	22.0	1	808k	494	0	0%	1.45k	0us	20
1	4604	15	1	1	840k	930	0	0%	114	22.7	1	817k	493	0	0%	1.04k	n/a	20
1	4604	15	1	1	839k	937	0	0%	0	22.2	1	814k	502	0	0%	1.04k	0us	20
1	4604	15	1	1	839k	935	0	0%	0	22.3	1	811k	489	0	0%	1.45k	0us	26
1	4604	15	1	1	838k	942	0	0%	114	22.2	1	807k	483	0	0%	2.01k	0us	21
1	4604	15	1	1	840k	932	0	0%	0	22.3	1	814k	486	0	0%	1.04k	0us	20
1	4604	15	1	1	839k	939	0	0%	114	22.6	1	812k	493	0	0%	745	0us	20
1	4604	15	1	1	840k	935	0	0%	0	22.6	1	812k	489	0	0%	1.04k	0us	20

Figura 5.2: Statistiche connessione downlink e uplink

Da questo primo confronto si può notare che le prestazioni delle due radio siano molto simili in queste condizioni.

MCS 8

Poi si è passati a dei valori di MCS pari a 9 per il PDSCH e 8 per il PUSCH, corrispondenti a una modulazione 64-QAM. I valori del guadagno front end sono stati lasciati a 0 in ricezioni e 20 in trasmissione.

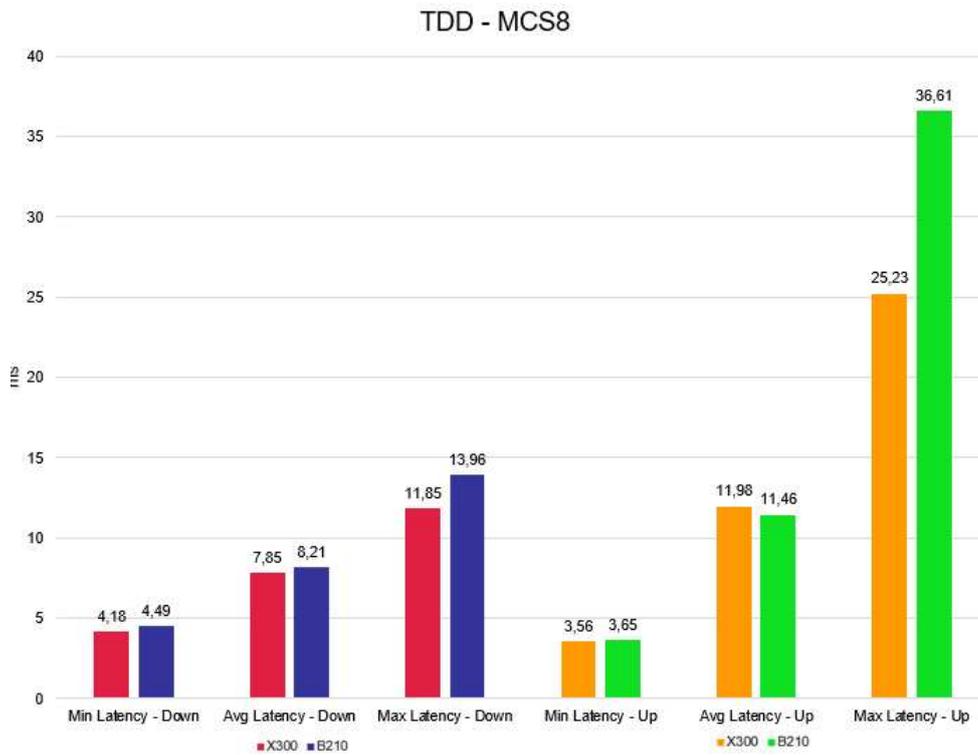


Figura 5.3: Confronto latenza in banda n78, MCS PDSCH 9, MCS PUSCH 8.

-DL-										-UL-									
pci	rnti	cqi	ri	mcs	brate	ok	nok	(%)	dl_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr	
1	4605	15	1	9	879k	845	0	0%	0	23.7	9	794k	137	0	0%	535	n/a	21	
1	4605	15	1	9	879k	845	0	0%	114	23.8	9	794k	136	1	0%	2.01k	0us	25	
1	4605	15	1	9	878k	844	0	0%	0	23.6	9	811k	139	4	2%	384	0us	21	
1	4605	15	1	9	879k	845	0	0%	7	23.2	9	818k	138	2	1%	535	0us	21	
1	4605	15	1	9	879k	844	0	0%	0	23.9	9	792k	135	1	0%	1.45k	0us	21	
1	4605	15	1	9	879k	845	0	0%	0	23.5	9	810k	135	2	1%	1.04k	0us	21	
1	4605	15	1	9	879k	845	0	0%	114	23.8	9	806k	141	3	2%	745	0us	21	
1	4605	15	1	9	878k	844	0	0%	0	23.5	9	797k	136	3	2%	745	0us	21	
1	4605	15	1	9	879k	845	0	0%	114	23.4	9	794k	135	5	3%	2.01k	0us	25	
1	4605	15	1	9	878k	844	0	0%	0	23.7	9	838k	144	1	0%	535	0us	21	
1	4605	15	1	9	879k	845	0	0%	7	23.5	9	787k	130	2	1%	535	0us	21	

Figura 5.4: Statistiche connessione downlink e uplink

In questo caso si nota già un lieve miglioramento delle prestazioni a favore della radio X300 in *downlink*.

MCS 20

In questo caso si sono impostati i valori dell'MCS a 21 per il PDSCH e 20 per il PUSCH, corrispondenti a una modulazione 256-QAM. I valori del guadagno front end sono stati messi in ricezioni a 10 e in trasmissione 20.

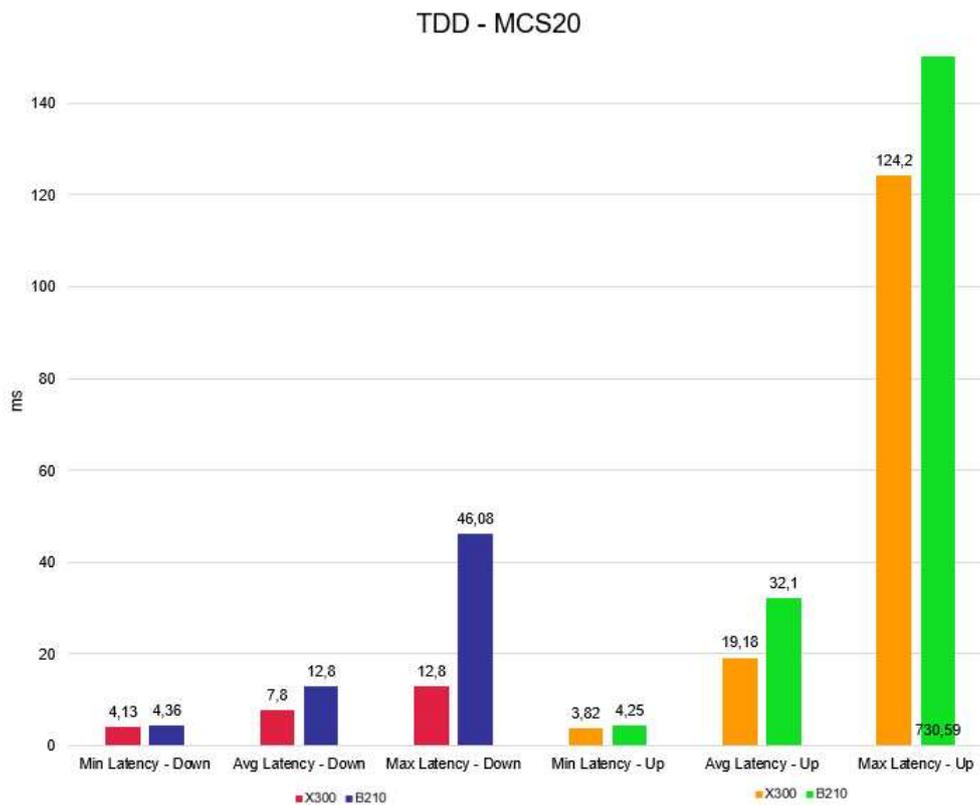


Figura 5.5: Confronto latenza in banda n78, MCS PDSCH 21, MCS PUSCH 20.

pci	rnti	-----DL-----							-----UL-----									
		cqi	ri	mcs	brate	ok	nok	(%)	dL_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr
1	4603	15	1	21	1.2M	845	0	0%	114	28.3	21	791k	100	0	0%	1.04k	0us	24
1	4603	15	1	21	1.2M	845	0	0%	228	28.2	21	794k	100	0	0%	1.04k	0us	23
1	4603	15	1	21	1.2M	844	0	0%	114	28.2	21	794k	100	0	0%	1.04k	0us	23
1	4603	15	1	21	1.2M	845	0	0%	228	28.2	21	795k	100	0	0%	745	0us	23
1	4603	15	1	21	1.2M	844	0	0%	228	28.3	21	792k	100	0	0%	745	0us	24
1	4603	15	1	21	1.2M	845	0	0%	114	28.3	21	793k	100	0	0%	1.04k	0us	23
1	4603	15	1	21	1.2M	844	0	0%	228	28.2	21	793k	100	0	0%	1.04k	0us	24
1	4603	15	1	21	1.2M	845	0	0%	114	28.2	21	795k	100	0	0%	745	0us	23
1	4603	15	1	21	1.2M	845	0	0%	228	28.3	21	794k	100	0	0%	745	0us	23
1	4603	15	1	21	1.2M	844	0	0%	228	28.3	21	792k	100	0	0%	745	0us	24
1	4603	15	1	21	1.2M	845	0	0%	228	28.2	21	793k	100	0	0%	1.04k	0us	23

Figura 5.6: Statistiche connessione downlink e uplink

Con questi valori di MCS si nota un netto miglioramento delle prestazioni di latenza a favore della X300 sia in *downlink* sia in *uplink*.

MCS 26

L'ultimo test effettuato con questi valori di larghezza di banda e spaziatura fra sottoportanti è stato con MCS a 27 per il PDSCH e 26 per il PUSCH, corrispondenti a una modulazione 256-QAM. I valori del guadagno front end sono stati messi in ricezioni a 0 e in trasmissione 20.

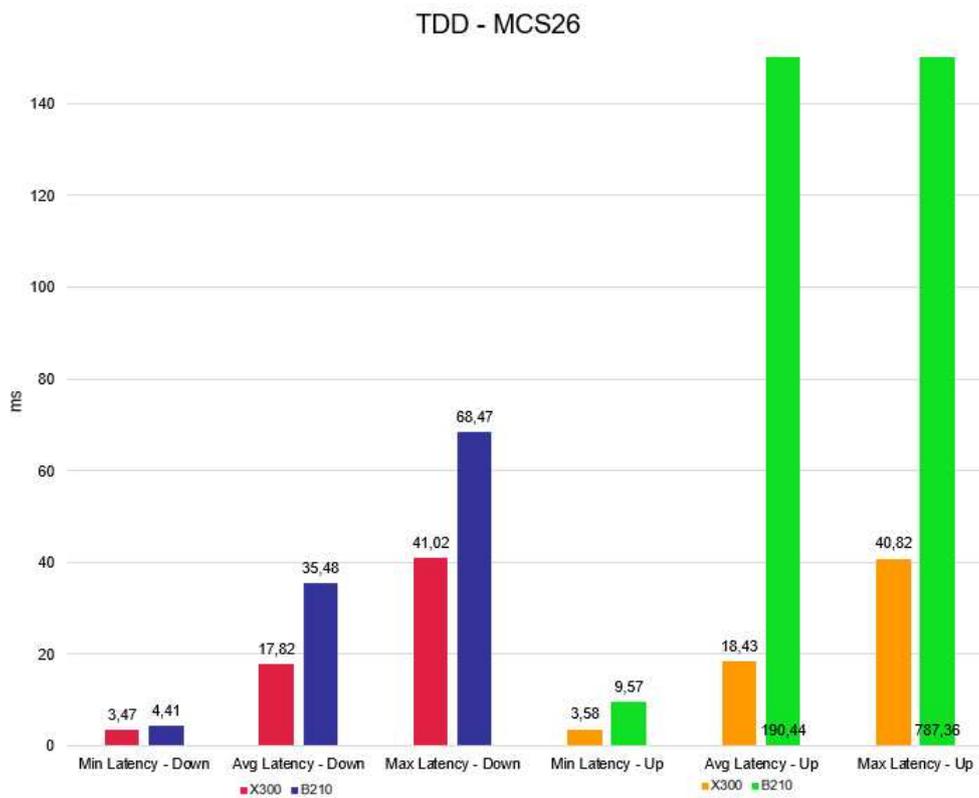


Figura 5.7: Confronto latenza in banda n78, MCS PDSCH 27, MCS PUSCH 26.

pci	rnti	--DL--								--UL--								
		cqi	ri	mcs	brate	ok	nok	(%)	dL_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr
1	4603	13	1	26	1.0M	889	52	5%	0	27.5	27	794k	100	0	0%	745	0us	24
1	4603	13	1	26	1.0M	886	73	7%	0	27.6	27	793k	100	0	0%	745	0us	24
1	4603	13	1	26	1.0M	895	53	5%	114	27.5	27	794k	100	0	0%	745	n/a	24
1	4603	13	1	26	1.0M	889	68	7%	0	27.1	27	792k	100	0	0%	1.04k	0us	25
1	4603	13	1	26	1.0M	887	79	8%	114	27.3	27	796k	100	0	0%	745	0us	24
1	4603	13	1	26	1.0M	876	84	8%	0	27.2	27	791k	100	0	0%	1.04k	0us	25
1	4603	13	1	26	1.1M	894	62	6%	114	27.3	27	793k	100	0	0%	1.04k	0us	24
1	4603	13	1	26	1.1M	902	39	4%	0	27.3	27	793k	100	0	0%	1.45k	0us	25
1	4603	13	1	26	1.0M	890	48	5%	0	27.2	27	796k	100	0	0%	745	0us	24
1	4603	14	1	26	1.0M	896	43	4%	114	27.3	27	793k	100	0	0%	745	0us	24
1	4603	13	1	26	1.0M	890	57	6%	0	27.2	27	792k	100	0	0%	1.04k	0us	25

Figura 5.8: Statistiche connessione downlink e uplink

Come prima, anche qui si nota una prestazione decisamente migliore della X300 rispetto alla B210 sia nella tratta in *downlink* sia nella tratta in *uplink*.

5.2 Modalità FDD

Successivamente, sono stati avviati i test in banda n7 (2600MHz), utilizzando il file di configurazione specifico `gnb_rf_x310_fdd_srsUE.yml`. Questo passo ha permesso di valutare le prestazioni della rete in una banda di frequenza diversa e di analizzare l'effetto di vari parametri di configurazione sulla qualità della comunicazione.

Durante questi test, sono stati mantenuti costanti i valori della larghezza di canale e della spaziatura tra le sottoportanti, rispettivamente 10 MHz e 15 KHz. Questo approccio ha permesso di focalizzare l'attenzione sull'analisi delle variazioni dell'MCS, senza introdurre ulteriori variabili che avrebbero potuto complicare l'interpretazione dei risultati.

Inoltre, per ogni MCS testato, sono stati variati anche i guadagni del *front end* sia in trasmissione sia in ricezione. L'obiettivo di questa variazione era ottenere una comunicazione pulita, senza perdita di pacchetti, ottimizzando così le condizioni di trasmissione e ricezione. È stato osservato che, indipendentemente dal MCS utilizzato, i valori di guadagno ottimali erano 0 dB in ricezione e 20 dB in trasmissione. Questi valori hanno garantito la miglior qualità di trasmissione, minimizzando la perdita di pacchetti e assicurando una comunicazione stabile e affidabile.

Infine, anche in questo caso i risultati ottenuti sono stati confrontati con quelli della radio B210. L'analisi comparativa ha fornito ulteriori *insight* sulla robustezza e l'efficienza della configurazione testata, evidenziando eventuali vantaggi o svantaggi nell'utilizzo di una radio rispetto all'altra.

MCS 0

La prima prova è stata eseguita selezionando gli MCS 1 per il PDSCH e 0 per il PUSCH, corrispondenti a una modulazione 64-QAM.

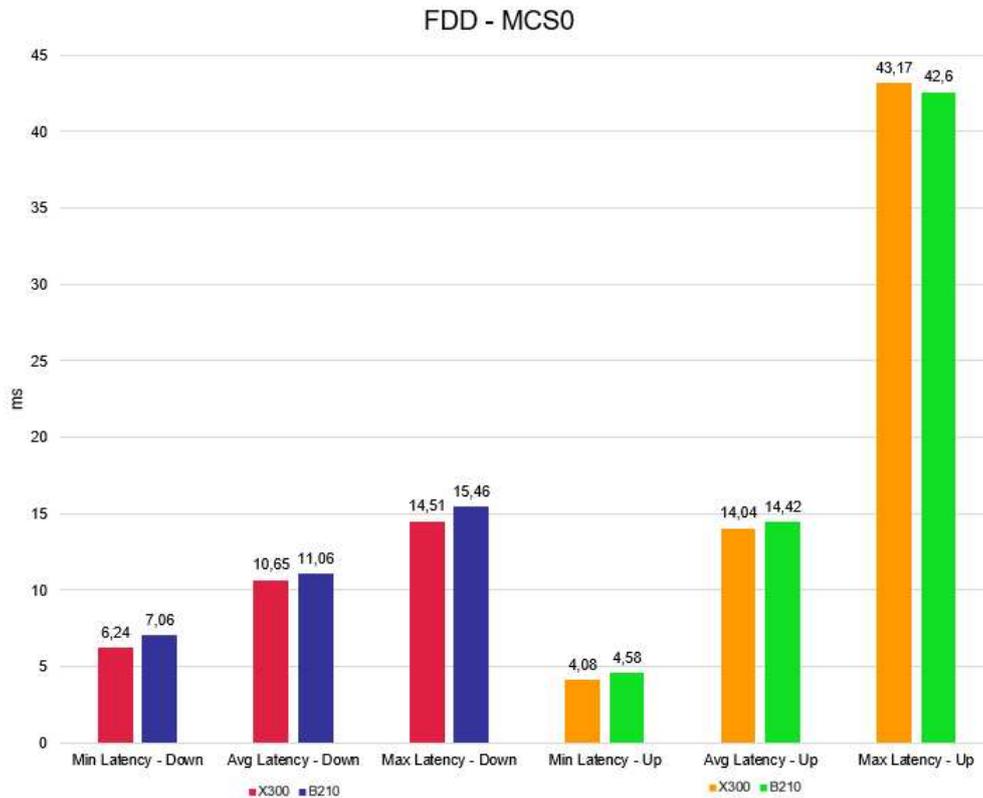


Figura 5.9: Confronto latenza in banda n7, MCS PDSCH 1, MCS PUSCH 0.

pci	rnti	DL							UL									
		cqi	ri	mcs	brate	ok	nok	(%)	dl_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr
1	4608	15	1	1	885k	939	0	0%	114	25.9	1	827k	512	0	0%	1.45k	n/a	19
1	4608	15	1	1	889k	936	0	0%	114	26.3	1	815k	531	0	0%	2.01k	0us	19
1	4608	15	1	1	888k	935	0	0%	0	25.9	1	826k	517	0	0%	1.04k	n/a	19
1	4608	15	1	1	893k	937	0	0%	114	25.8	1	819k	509	0	0%	1.04k	0us	28
1	4608	15	1	1	886k	935	0	0%	114	25.8	1	827k	519	0	0%	1.04k	n/a	19
1	4608	15	1	1	889k	937	0	0%	114	25.8	1	825k	523	0	0%	597	0us	19
1	4608	15	1	1	886k	938	0	0%	114	25.7	1	830k	524	0	0%	0	0us	19
1	4608	15	1	1	887k	939	0	0%	0	25.9	1	820k	524	0	0%	1.45k	n/a	27
1	4608	15	1	1	887k	939	0	0%	114	25.8	1	823k	522	0	0%	1.45k	n/a	27
1	4608	15	1	1	887k	935	0	0%	114	26.3	1	825k	548	0	0%	1.04k	0us	24
1	4608	15	1	1	888k	938	0	0%	114	25.8	1	817k	514	0	0%	1.45k	0us	19

Figura 5.10: Statistiche connessione downlink e uplink

Come per la trasmissione in TDD si nota che con questi valori di MCS le due radio si comportano in maniera molto simile.

MCS 8

Anche in questo caso il secondo test stato eseguito impostando gli MCS a 9 per il PDSCH e a 8 per il PUSCH, corrispondenti a una modulazione 64-QAM.

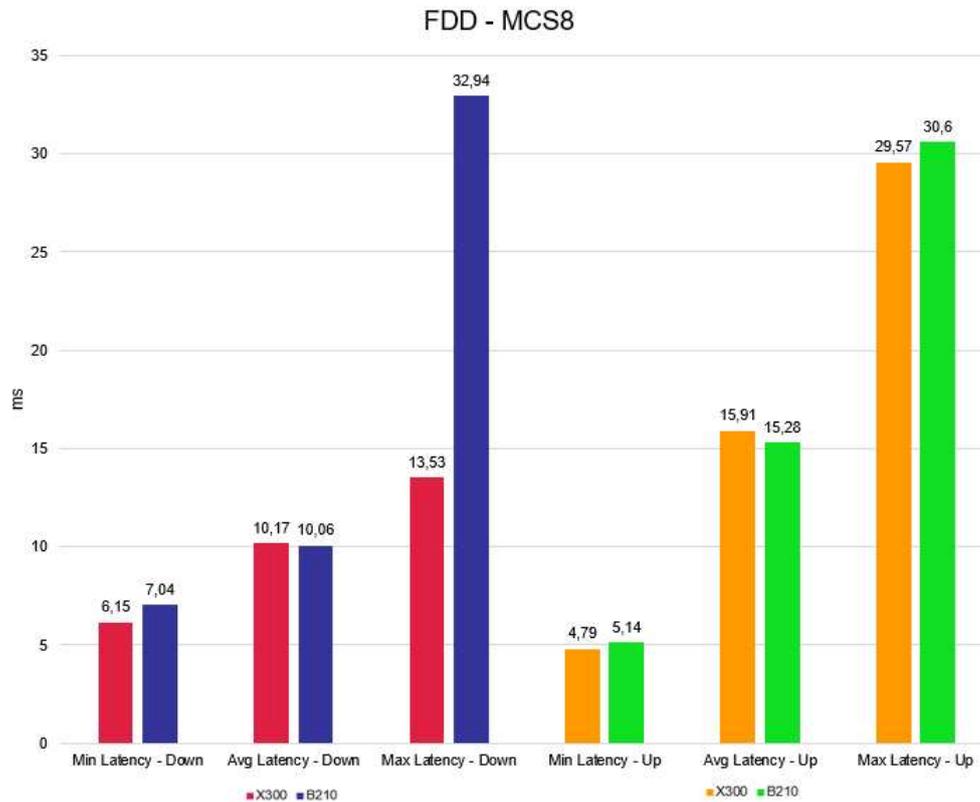


Figura 5.11: Confronto latenza in banda n7, MCS PDSCH 9, MCS PUSCH 8.

pci	rnti	-DL-							-UL-									
		cqi	ri	mcs	brate	ok	nok	(%)	dL_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr
1	4608	14	1	9	161k	101	0	0%	114	27.7	9	235k	33	0	0%	1.45k	0us	20
1	4608	14	1	9	882k	845	0	0%	114	29.6	9	797k	149	0	0%	1.45k	0us	20
1	4608	14	1	9	882k	845	0	0%	114	29.4	9	792k	150	0	0%	1.45k	0us	24
1	4608	14	1	9	880k	844	0	0%	114	29.4	9	799k	147	0	0%	535	0us	20
1	4608	15	1	9	882k	845	0	0%	0	28.6	9	795k	134	0	0%	535	0us	20
1	4608	15	1	9	880k	844	0	0%	114	28.7	9	796k	136	0	0%	535	0us	20
1	4608	15	1	9	882k	845	0	0%	114	29.4	9	795k	148	0	0%	384	0us	20
1	4608	14	1	9	882k	845	0	0%	114	29.5	9	797k	154	0	0%	535	0us	20
1	4608	15	1	9	880k	843	0	0%	114	28.4	9	796k	137	0	0%	384	0us	20
1	4608	14	1	9	883k	846	0	0%	114	28.9	9	793k	142	0	0%	1.04k	0us	20
1	4608	15	1	9	881k	844	0	0%	114	28.8	9	798k	138	0	0%	384	0us	20

Figura 5.12: Statistiche connessione downlink e uplink

Anche in questo caso non si notano grandi differenze prestazionali tra le due radio.

MCS 20

La terza prova è stata effettuata con valori di MCS a 21 per il PDSCH e a 20 per il PUSCH, corrispondenti a una modulazione 256-QAM.

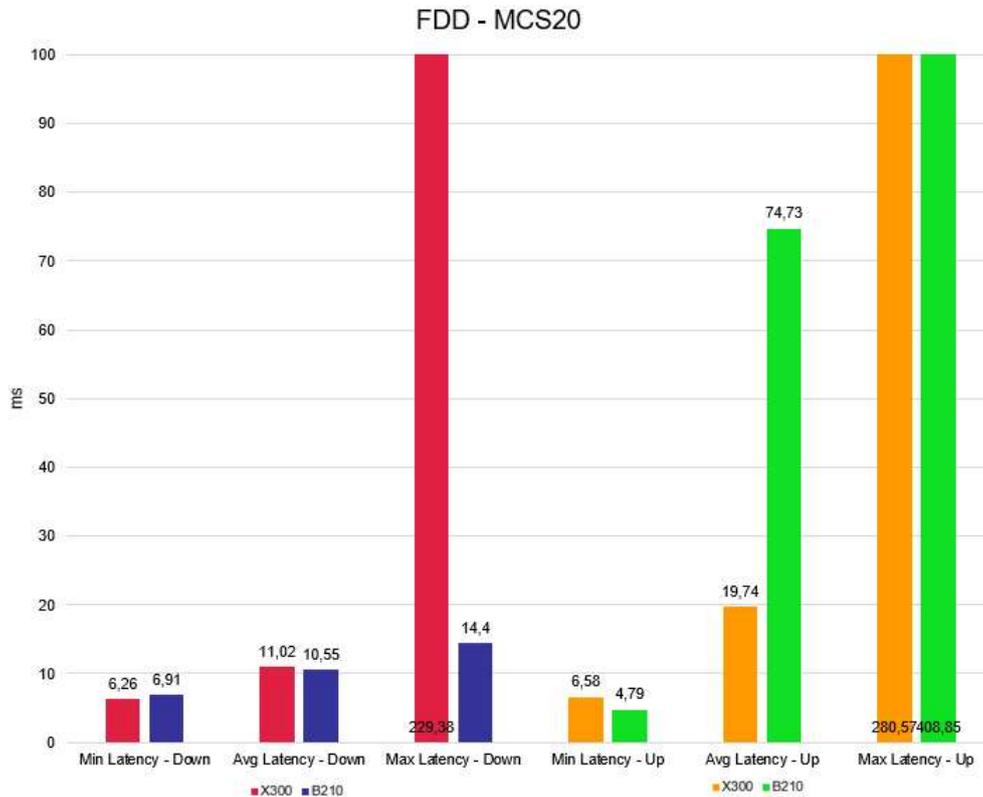


Figura 5.13: Confronto latenza in banda n7, MCS PDSCH 21, MCS PUSCH 20.

pci	rnti	--DL--							--UL--									
		cqi	ri	mcs	brate	ok	nok	(%)	dL_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr
1	4604	15	1	21	438k	261	0	0%	114	33.3	21	427k	40	0	0%	1.04k	0us	24
1	4604	15	1	21	1.2M	845	0	0%	114	33.3	21	794k	100	0	0%	745	0us	23
1	4604	15	1	21	1.2M	844	0	0%	114	33.2	21	792k	100	0	0%	1.04k	0us	23
1	4604	15	1	21	1.2M	844	0	0%	0	33.1	21	792k	107	0	0%	1.04k	0us	23
1	4604	15	1	21	1.2M	845	0	0%	114	32.8	21	794k	100	0	0%	745	0us	23
1	4604	15	1	21	1.2M	844	0	0%	114	32.8	21	792k	100	0	0%	1.45k	0us	24
1	4604	15	1	21	1.2M	846	0	0%	114	33.0	21	795k	100	0	0%	1.04k	0us	23
1	4604	15	1	21	1.2M	844	0	0%	114	32.7	21	793k	100	0	0%	1.04k	0us	23
1	4604	15	1	21	1.2M	845	0	0%	0	32.7	21	794k	100	0	0%	745	0us	23
1	4604	15	1	21	1.2M	844	0	0%	114	32.8	21	793k	100	0	0%	745	0us	23
1	4604	15	1	21	1.2M	845	0	0%	114	32.6	21	792k	100	0	0%	1.45k	0us	23

Figura 5.14: Statistiche connessione downlink e uplink

In questo caso si vede come la latenza media in *downlink* sia molto simile, mentre in *uplink* sia molto più bassa nel caso della X300.

MCS 26

Infine si è impostato l'MCS a 27 per il PDSCH e a 26 per il PUSCH, corrispondenti a una modulazione 256-QAM.

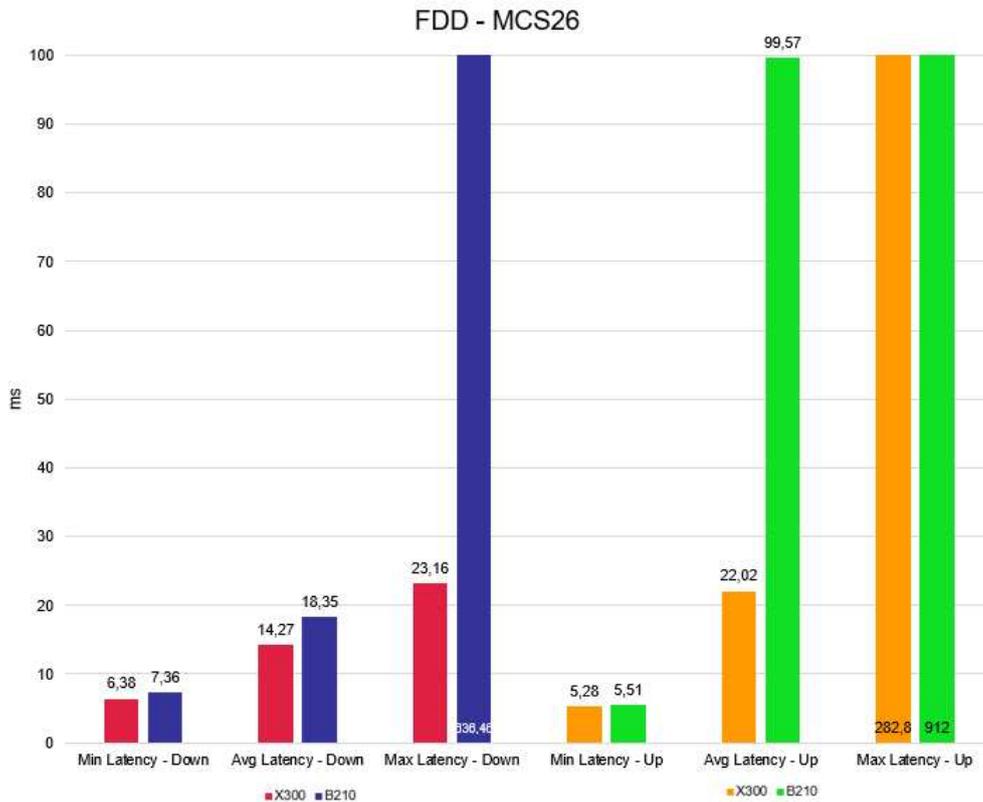


Figura 5.15: Confronto latenza in banda n7, MCS PDSCH 27, MCS PUSCH 26.

pci	rnti	DL								UL								
		cqi	ri	mcs	brate	ok	nok	(%)	dL_bs	pusch	mcs	brate	ok	nok	(%)	bsr	ta	phr
1	4603	15	1	27	982k	938	0	0%	114	30.5	27	784k	70	26	27%	2.01k	0us	21
1	4603	15	1	27	975k	935	0	0%	121	32.7	27	808k	94	6	6%	1.04k	0us	24
1	4603	15	1	27	981k	938	0	0%	114	33.9	27	794k	100	0	0%	1.04k	0us	24
1	4603	15	1	27	979k	937	0	0%	121	34.0	27	794k	100	0	0%	745	0us	24
1	4603	15	1	27	975k	935	0	0%	114	33.8	27	792k	100	0	0%	1.04k	0us	24
1	4603	15	1	27	976k	937	0	0%	121	33.6	27	794k	100	0	0%	1.04k	0us	24
1	4603	15	1	27	977k	936	0	0%	7	33.7	27	794k	100	0	0%	745	0us	24
1	4603	15	1	27	979k	938	0	0%	114	33.9	27	792k	100	0	0%	1.04k	0us	24
1	4603	15	1	27	982k	940	0	0%	121	33.8	27	794k	100	0	0%	1.04k	0us	24
1	4603	15	1	27	344k	329	0	0%	0	33.3	27	270k	35	0	0%	0	0us	29
1	4603	15	1	0	0	0	0	0%	0	n/a	0	0	0	0	0%	0	0us	29

Figura 5.16: Statistiche connessione downlink e uplink

Con questi valori di MCS la X300 si comporta meglio sia nella tratta *downlink*, ma soprattutto nella tratta in *uplink*

5.3 Aumento della banda

Le ultime verifiche condotte si sono concentrate sulla larghezza di banda e sulla spaziatura tra le sottoportanti. L'obiettivo di queste analisi era determinare se le modifiche a questi parametri potessero influenzare positivamente i valori di latenza, abbassandoli ulteriormente, o se i valori ottenuti nelle misurazioni precedenti rappresentassero già il limite massimo delle prestazioni radio.

In dettaglio, la larghezza di banda e la spaziatura tra le sottoportanti sono elementi cruciali nel determinare la capacità e l'efficienza di una rete di comunicazione wireless. La larghezza di banda definisce il range di frequenze utilizzate per la trasmissione dei dati, mentre la spaziatura tra le sottoportanti riguarda la distanza in frequenza tra i singoli portanti utilizzati nella modulazione OFDM (Orthogonal Frequency Division Multiplexing).

Per le verifiche, sono stati configurati diversi scenari di test variando sistematicamente la larghezza di banda e la spaziatura tra le sottoportanti e i valori di MCS. Ogni scenario è stato analizzato attentamente per valutare l'impatto di queste variazioni sulla latenza di rete. In particolare sono stati effettuati test in TDD, in banda n78, con valori di MCS pari a 0, 8, 20 per il PUSCH e pari a 1, 9, 21 per il PDSCH con larghezza di banda aumentata a 80MHz. Mentre in FDD, banda n7, si è effettuato un test a MCS 0 con larghezza di banda aumentata a 40MHz.

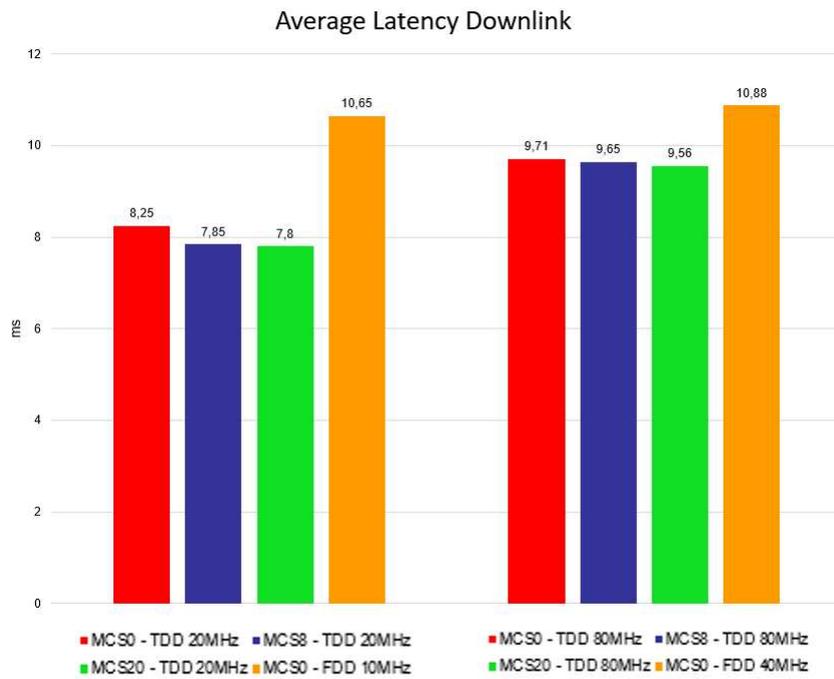


Figura 5.17: Confronto latenza in downlink

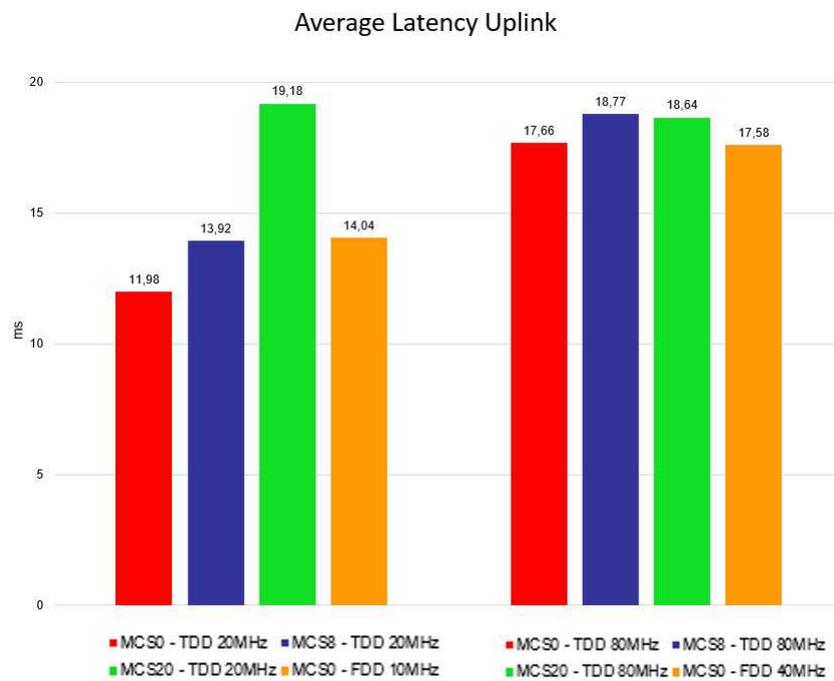


Figura 5.18: Confronto latenza in uplink

Le misurazioni hanno rivelato che le modifiche alla larghezza di banda non

hanno mostrato miglioramenti significativi, suggerendo che i parametri di rete originali erano già ottimizzati per le condizioni operative specifiche. Questo risultato implica che, per ottenere ulteriori riduzioni della latenza, potrebbero essere necessarie modifiche più sostanziali all'architettura di rete o l'adozione di nuove tecnologie di trasmissione.

5.4 Sintesi

I test di latenza hanno rivelato diversi elementi significativi riguardo il comportamento delle due radio utilizzate, la X310 e la B210. Inizialmente, è stato osservato che le due radio si comportano in modo simile quando vengono utilizzati bassi valori di MCS. In questa fase, i valori di latenza rilevati sono pressoché identici, indicando che entrambi i dispositivi sono in grado di gestire efficacemente le comunicazioni a schemi di modulazione e codifica meno complessi.

Con l'aumento del valore dell'MCS, la radio X310 ha dimostrato prestazioni superiori rispetto alla B210, specialmente in *uplink*. I valori di latenza media della X310 erano notevolmente più bassi, suggerendo una maggiore efficienza nella gestione di schemi di modulazione e codifica più complessi. Questo comportamento può essere attribuito alle capacità hardware avanzate della X310, che le permettono di elaborare i dati più rapidamente e con maggiore precisione rispetto alla B210.

È stato cruciale studiare e impostare correttamente i valori del guadagno *front end*. La variazione di questi valori può influenzare drasticamente le prestazioni della radio. Se i guadagni non sono configurati correttamente, la radio potrebbe non funzionare correttamente o si potrebbero ottenere valori elevati di pacchetti ricevuti errati. Questo fenomeno è noto come "*blinding*", dove la potenza di trasmissione troppo elevata può accecare la ricezione, causando un

alto tasso di errori.

Un'altra variabile considerata nei test è stata la larghezza di banda. Tuttavia, l'aumento della larghezza di banda non ha portato a un miglioramento dei risultati di latenza. Questo indica che, oltre una certa soglia, la larghezza di banda non è il fattore limitante per la latenza, e che altre variabili, come l'efficienza della modulazione e codifica e la gestione del traffico, potrebbero avere un impatto maggiore.

Uno degli aspetti più rilevanti emersi dai test è la differenza nella percentuale di pacchetti ricevuti errati tra le due radio. La X310 ha mostrato una percentuale di pacchetti ricevuti errati pari a zero in tutti i test, a differenza della B210, che ha registrato errori di ricezione. Ciò è probabilmente dovuto al fatto che B210 non riesce a reggere schemi di modulazione più avanzati e rate di codifica più stringenti. Questo risultato evidenzia la maggiore affidabilità della X310, attribuibile alla sua architettura hardware più avanzata.

Nonostante queste differenze, è importante notare che i risultati complessivi di latenza tra la X310 e la B210 non erano drasticamente diversi. Questo porta a riflettere sul fatto che il limite delle prestazioni potrebbe non essere dovuto all'hardware, ma piuttosto al software utilizzato per gestire le comunicazioni. Le limitazioni software potrebbero includere inefficienze nel protocollo di comunicazione, nella gestione del traffico o nella configurazione dei parametri di rete.

In sintesi, i test di latenza hanno dimostrato che, sebbene la radio X310 possieda capacità hardware superiori rispetto alla B210, le differenze nei risultati di latenza non sono così marcate come ci si potrebbe aspettare. La gestione accurata dei guadagni front end e la configurazione appropriata dei parametri di rete sono fondamentali per ottimizzare le prestazioni. Tuttavia, l'analisi suggerisce che per ottenere miglioramenti significativi nelle prestazioni di la-

tenza, potrebbe essere necessario un miglioramento del software di gestione della rete, piuttosto che un semplice aggiornamento dell'hardware.

Capitolo 6

Test slicing

In questo capitolo, si procederà all'analisi dei risultati ottenuti dai test di *Network Slicing*. L'obiettivo principale era verificare che le due reti virtuali create operassero correttamente, assicurando che una gestisse i pacchetti relativi al traffico voce e l'altra i pacchetti relativi al traffico dati. Come descritto nel capitolo 4.2, sono stati impostati due livelli diversi di priorità per queste reti virtuali.

Per la rete dedicata al traffico dati, la SIM è stata configurata con un 5QI pari a 9. Questo valore è tipicamente utilizzato per servizi come video, *streaming*, *chat* e *file sharing*. Inoltre, il livello di priorità ARP è stato impostato a 8, che rappresenta il livello di priorità più basso tra quelli dedicati alle risorse per i servizi autorizzati a ricevere un trattamento prioritario all'interno di un dominio dell'operatore.

Per la rete destinata al traffico voce, la SIM è stata configurata con un 5QI pari a 7, che secondo lo standard è il livello relativo alle comunicazioni vocali. Il livello ARP è stato impostato a 1, garantendo la priorità massima per questo tipo di traffico.

Come discusso nel capitolo 4.2, i pacchetti relativi al traffico voce hanno una dimensione di 82 byte, mentre i pacchetti relativi al traffico dati hanno una dimensione di 1000 byte. Questa differenza nella dimensione dei pacchetti riflette le diverse esigenze di larghezza di banda e latenza dei due tipi di traffico.

Durante i test, sono state monitorate attentamente le prestazioni di entrambe le reti virtuali per assicurarsi che rispettassero le configurazioni impostate. Sono stati eseguiti vari scenari per simulare condizioni di traffico realistiche e verificare il comportamento delle reti sotto carico.

6.1 Risultati

Sono stati condotti numerosi test per valutare le prestazioni della rete, variando una serie di parametri chiave. Questi test hanno avuto l'obiettivo di analizzare in dettaglio l'impatto delle diverse configurazioni sui valori di latenza e *throughput*, due metriche fondamentali per la qualità del servizio di una rete di comunicazione.

6.1.1 Carico voce fisso

I test analizzati in questa sezione sono stati condotti mantenendo il carico del traffico dati costante e fisso a 1 Mbps per garantire un riferimento stabile durante le misurazioni. Le prime serie di test sono state effettuate in banda n7, operando in modalità FDD, con una larghezza di canale di 40 MHz. Per queste misurazioni, l'MCS è stato fissato a 21 per il PDSCH e a 20 per il PUSCH. Il carico del traffico dati è stato variato attraverso quattro scenari distinti: 5 Mbps, 50 Mbps, 100 Mbps e 200 Mbps. Questo approccio ha permesso di osservare come la rete gestisce differenti volumi di dati e come queste variazioni influenzano le prestazioni in termini di latenza e *throughput* sia per il traffico voce che per il traffico dati.

	Throughput dati	Throughput voce	Latency Downlink dati (ms)	Latency Uplink dati (ms)	Latency Downlink voce (ms)	Latency Uplink voce (ms)
5Mbps	OK	OK	10.72	19.35	11.04	16.94
50Mbps	OK	OK	9.93	30.06	11.08	26.12
100Mbps	25% Uplink	OK	9.51	184.94	11.61	99.62
200Mbps	60% Uplink 20% Downlink	OK	215.71	193.59	11.01	51.62

Figura 6.1: Test latenza e throughput con slicing in banda 7, carico dati 1Mbps

I test hanno mostrato che la latenza in *downlink* per il traffico voce rimane molto costante indipendentemente dal carico di traffico dati. Questo è indicativo della capacità della rete di gestire il traffico voce con alta priorità, garantendo una comunicazione stabile e senza ritardi significativi. La costanza della latenza è essenziale per mantenere la qualità delle chiamate vocali, un requisito critico per molte applicazioni di comunicazione. Il *throughput* del traffico voce ha raggiunto valori massimi in tutti gli scenari testati. Questo conferma l'efficacia delle configurazioni di priorità impostate per il traffico voce, assicurando che i pacchetti vocali vengano trasmessi con la massima efficienza possibile. La rete è riuscita a mantenere elevate prestazioni per il traffico voce anche con l'aumento del carico del traffico dati. Invece il *throughput* del traffico dati è stato massimo fino a un carico di 50 Mbps. Tuttavia, oltre questo carico, è iniziato a degradare progressivamente. Questo degrado del *throughput* indica che la rete inizia a saturarsi, riducendo la capacità di trasmettere dati aggiuntivi senza perdere prestazioni. La degradazione del *throughput* del traffico dati è un segnale che la rete ha raggiunto il suo limite di capacità efficiente per quella configurazione. Di conseguenza, la latenza per il traffico dati ha mostrato un peggioramento con carichi superiori a 50 Mbps. L'aumento della latenza è particolarmente evidente a carichi di 100 Mbps e 200 Mbps, indicando che la rete non è in grado di mantenere la stessa efficienza di trasmissione sotto carichi elevati.

Il secondo test è stato condotto mantenendo le stesse condizioni del primo,

ma con una differenza chiave: questa volta, l'MCS è stato lasciato libero di adattarsi dinamicamente alle condizioni del canale. Questa strategia consente alla trasmissione di adeguarsi automaticamente al *Channel Quality Indicator* (CQI), ottimizzando così le prestazioni della rete in tempo reale. In questo caso il carico del traffico dati è variato tra 100 Mbps, 150 Mbps e 200 Mbps.

	Throughput dati	Throughput voce	Latency Downlink dati (ms)	Latency Uplink dati (ms)	Latency Downlink voce (ms)	Latency Uplink voce (ms)
100Mbps	OK	OK	9.34	13.49	12.65	21.08
150Mbps	25% Uplink	OK	8.77	151.75	11.41	39.7
200Mbps	50% Uplink 25% Downlink	OK	217.14	147.57	11.21	35.29

Figura 6.2: Test latenza e throughput con slicing in banda 7, carico dati 1Mbps, MCS libero

Anche con l'MCS adattivo, la latenza in *downlink* per il traffico voce è rimasta molto costante. Questo dimostra che la priorità assegnata al traffico voce è sufficientemente alta da garantire una latenza stabile, indipendentemente dalle variazioni del MCS e dalle condizioni del canale. Il *throughput* del traffico voce ha continuato a essere massimo in tutti gli scenari testati. L'MCS adattivo ha permesso di mantenere elevate prestazioni per il traffico voce, confermando che la rete può efficacemente gestire le comunicazioni vocali anche con condizioni di canale variabili. Per quanto riguarda il traffico dati la latenza in *downlink* è rimasta molto costante, mantenendo valori bassi anche con carichi di traffico variabili, fino a 200 Mbps. A questi valori infatti il *throughput* del traffico dati in *downlink* ha iniziato a degradare. Invece in *uplink*, il *throughput* del traffico dati ha iniziato a degradare a partire da 150 Mbps. Questo degrado è stato accompagnato da un aumento della latenza in *uplink*, suggerendo che la rete sta raggiungendo i suoi limiti di capacità per il traffico in uscita.

Il terzo ed ultimo test effettuato mantenendo il carico del traffico voce fisso a 1 Mbps, è stato condotto in banda n78, utilizzando la modalità TDD, con una larghezza del canale pari a 80 MHz e lasciando l'MCS libero di adattarsi

dinamicamente alle condizioni del canale. In questo scenario, il carico del traffico dati è stato variato attraverso cinque diversi valori: 50 Mbps, 100 Mbps, 150 Mbps, 200 Mbps e 250 Mbps.

	Throughput dati	Throughput voce	Latency Downlink dati (ms)	Latency Uplink dati (ms)	Latency Downlink voce (ms)	Latency Uplink voce (ms)
50Mbps	OK	OK	8.85	38.64	9.61	18.94
100Mbps	40% Uplink	OK	8.43	279.64	9.64	17.46
150Mbps	55% Uplink	OK	7.94	258.67	10.26	20.19
200Mbps	65% Uplink	OK	8.17	254.17	10.03	17.36
250Mbps	12% Downlink 75% Uplink	OK	149.31	274.58	10.07	17.46

Figura 6.3: Test latenza e throughput con slicing in banda 78, carico dati 1Mbps, MCS libero

In questo caso si può notare come la latenza, sia in *downlink* che in *uplink*, per il traffico voce è rimasta costante e bassa in tutti gli scenari testati. Questo dimostra l'efficacia della priorità assegnata al traffico voce, che garantisce una qualità stabile delle comunicazioni vocali, indipendentemente dal carico di traffico dati. Il *throughput* del traffico voce ha raggiunto il massimo in tutti gli scenari testati. L'adattamento dinamico del MCS ha permesso di mantenere elevate prestazioni per il traffico voce, confermando che la rete può gestire efficacemente le comunicazioni vocali anche in condizioni di canale variabili e con alti carichi di traffico dati. Il *throughput* del traffico dati in *downlink* è rimasto massimo fino a un carico di 200 Mbps. Tuttavia, a 250 Mbps, il *throughput* ha iniziato a degradare, indicando che la capacità di trasmissione della rete viene saturata a questo livello di carico. Di pari passo, la latenza in *downlink* è peggiorata significativamente a 250 Mbps, confermando che la rete sta raggiungendo i suoi limiti operativi. Invece il *throughput* del traffico dati in *uplink* ha iniziato a degradare già a partire da 100 Mbps. Questo peggioramento è stato accompagnato da un aumento della latenza in *uplink*, suggerendo che la capacità *uplink* della rete è inferiore a quella *downlink*.

6.1.2 Carico voce variabile

Gli ultimi test riguardanti il *Network Slicing* sono stati effettuati variando anche il carico del traffico voce. Questi test sono stati eseguiti esclusivamente in banda n7, utilizzando la modalità FDD, con una larghezza di canale di 40 MHz e un MCS libero di adattarsi alle condizioni del canale. In questa serie di test, il carico del traffico voce è stato variato tra 10 Mbps, 50 Mbps, 75 Mbps e 100 Mbps, mentre il carico del traffico dati è stato variato tra 50 Mbps, 100 Mbps, 150 Mbps e 200 Mbps.

	Throughput dati	Throughput voce	Latency Downlink dati (ms)	Latency Uplink dati (ms)	Latency Downlink voce (ms)	Latency Uplink voce (ms)
50Mbps – 10Mbps	OK	OK	9.88	15.37	9.32	19.4
100Mbps – 50Mbps	5% Uplink	OK	9.61	150.19	7.18	42.38
150Mbps – 75Mbps	45% Downlink 55% Uplink	OK	400.74	265.39	7.53	62.48
200Mbps – 100Mbps	55% Downlink 75% Uplink	30% Uplink	410.41	295.4	8.55	82.7

Figura 6.4: Test latenza e throughput con slicing in banda 7, carico dati variabile, MCS libero

Si nota che la latenza per il traffico voce, sia in *downlink* che in *uplink*, è rimasta costante e bassa per tutti i valori di carico testati. Questo risultato conferma che la rete è in grado di mantenere una qualità stabile delle comunicazioni vocali, indipendentemente dal carico di traffico dati. In particolare, la latenza in *downlink* ha mostrato una notevole stabilità, garantendo una trasmissione fluida e continua delle comunicazioni vocali. Il *throughput* del traffico voce ha raggiunto valori massimi in quasi tutti gli scenari testati. Tuttavia, in *uplink* ha mostrato segni di degradazione solo quando il carico del traffico voce ha raggiunto 100 Mbps e quello del traffico dati 200 Mbps. Questo indica che la rete può gestire elevati volumi di traffico voce senza compromettere significativamente le prestazioni, ma che esiste un limite oltre il quale le risorse diventano insufficienti per mantenere il *throughput* massimo. Il *throughput* del traffico dati in *downlink* è rimasto massimo fino a un carico di 150 Mbps. A

partire da questo valore ha iniziato a degradare, con un aumento significativo della latenza. Questo suggerisce che la capacità di trasmissione della rete viene saturata a carichi di traffico dati molto elevati. Invece il *throughput* del traffico dati in *uplink* ha iniziato a degradare già a partire da 100 Mbps, con un corrispondente aumento della latenza. Questo comportamento indica che la capacità *uplink* della rete è inferiore a quella *downlink*, un fenomeno comune nelle configurazioni FDD.

6.2 Sintesi

Questi test hanno fornito preziose indicazioni sull'efficacia del *Network Slicing* nel creare due reti virtuali separate, ciascuna dedicata a diversi tipi di traffico, in questo caso voce e dati. Le misurazioni effettuate hanno confermato che lo *slicing* di rete permette una gestione efficiente del traffico, garantendo prestazioni ottimali per entrambi i flussi, anche in condizioni di carico variabile.

Lo *slicing* di rete ha dimostrato la sua capacità di creare due reti distinte, ciascuna delle quali può gestire specifici tipi di traffico. Nei test, il traffico voce e dati sono stati trasportati su reti virtuali separate, permettendo un'allocazione dedicata delle risorse di rete. Questo isolamento ha garantito che le prestazioni di una rete non fossero influenzate dal carico sull'altra.

Durante i test, il *throughput* del traffico voce è rimasto sempre massimo, indipendentemente dal carico del traffico dati. La latenza dello *stream* voce è stata costantemente bassa e stabile, dimostrando che la priorità assegnata al traffico voce è stata rispettata. Questo è particolarmente importante per garantire la qualità delle chiamate vocali, che sono sensibili alla latenza e alla perdita di pacchetti.

Con l'MCS libero di adattarsi alle condizioni del canale, la rete ha potuto ottimizzare le prestazioni dinamicamente. Questo adattamento ha permesso

di mantenere buone performance per lo *stream* voce anche quando il carico del traffico dati aumentava, confermando l'efficacia delle configurazioni adattive in un ambiente di rete variabile.

Nonostante la degradazione del traffico dati, le prestazioni del traffico voce sono rimaste stabili. Questo dimostra che la separazione delle reti attraverso lo *slicing* consente di mantenere elevate prestazioni per uno *stream* anche se l'altro inizia a degradare. Questo è un vantaggio significativo del network *slicing*, poiché permette di isolare e proteggere i flussi di traffico prioritari.

I risultati dei test hanno confermato che le due reti virtuali create funzionavano correttamente, rispettando le priorità impostate e garantendo la qualità del servizio per entrambi i tipi di traffico. La rete per il traffico voce ha mantenuto una latenza bassa e una stabilità elevata, grazie all'alto livello di priorità ARP. La rete per il traffico dati ha gestito efficacemente volumi elevati di traffico, con una qualità di servizio adeguata per applicazioni meno sensibili alla latenza.

Capitolo 7

Implementazione traffico voce reale

Questo capitolo si concentrerà sull'analisi dettagliata del test set e dei risultati ottenuti tramite il *Network Slicing* e l'applicazione di traffico voce reale. Questo ha permesso di valutare l'efficacia e le prestazioni dello *slicing* in un contesto pratico.

Questo approccio è stato reso possibile grazie all'utilizzo del dispositivo router A TLC BG8520E. Il router A TLC BG8520E rappresenta una soluzione avanzata per la gestione e l'ottimizzazione delle reti di telecomunicazione, offrendo una piattaforma robusta e versatile per l'implementazione di servizi di rete.

Sul router A TLC BG8520E, sono stati installati due server distinti: un server voce e un server dati. Il server voce è stato configurato per gestire il traffico voce, permettendo di simulare chiamate reali e monitorare la qualità del servizio in condizioni variabili. Parallelamente, il server dati è stato utilizzato per il traffico dati, consentendo di analizzare l'interazione e la coesistenza di diverse tipologie di traffico sulla stessa infrastruttura di rete.

7.1 Test set

Il test set utilizzato in questa fase dell'analisi è molto simile a quello impiegato per i test del *Network Slicing*, descritto nel capitolo 4.2 ma presenta alcune differenze fondamentali che meritano una descrizione dettagliata. Prima di tutto, come già accennato, il router A TLC BG8520E è stato utilizzato come server voce e dati, sostituendo quindi il TestCenter che precedentemente simulava il traffico. Questa modifica ha permesso di lavorare con traffico reale anziché simulato, fornendo un'analisi più accurata e realistica delle prestazioni del sistema.

Sul router A TLC BG8520E sono stati configurati due server distinti:

- **Server Iperf per il traffico dati:** Questo server è stato impostato per gestire pacchetti UDP, consentendo di misurare la larghezza di banda, la latenza, il *jitter* e la perdita di pacchetti nel traffico dati.
- **Server Voce per il traffico RTP:** Questo server è stato configurato per gestire pacchetti RTP, utilizzati nelle comunicazioni vocali in tempo reale.

Il router A TLC BG8520E è stato collegato all'indirizzo IP 10.0.0.1, lo stesso indirizzo utilizzato dal TestCenter nei test precedenti. Questa scelta ha semplificato la configurazione, permettendo di mantenere una coerenza nelle impostazioni di rete e di ridurre il rischio di errori dovuti a cambiamenti di indirizzo IP.

Per quanto riguarda la configurazione interna del PC, non sono state apportate modifiche significative rispetto ai test dello *slicing*. Di conseguenza, tutti i file di configurazione della radio, le configurazioni delle rotte e delle SIM sono rimasti invariati. Per i dettagli su queste configurazioni, si può fare riferimento direttamente al capitolo dedicato allo *slicing* 4.2.

Anche la configurazione della radio è rimasta invariata. La radio è configurata con due interfacce virtuali, ogstun1 e ogstun2, collegate rispettivamente a due XG9520-5GWAX. Questa configurazione permette di gestire separatamente il traffico voce e il traffico dati, ottimizzando le prestazioni di ciascun tipo di traffico.

Per quanto riguarda gli XG9520-5GWAX, vi sono alcune specifiche configurazioni. L'XG per il traffico dati è stato collegato alla radio tramite l'indirizzo IP 10.46.0.2 e al router A TLC BG8520E tramite l'indirizzo IP 192.168.3.1. La sua configurazione non ha subito modifiche rispetto ai test precedenti. Invece l'XG per il traffico voce presenta alcune novità rispetto alla configurazione precedente. Innanzitutto due telefoni sono stati collegati a questo XG tramite classici cavi telefonici, permettendo di simulare chiamate vocali reali. Poi si è aggiornata la configurazione in questo modo:

```
set system loglevel 5
set system name ATOSNT
set system console-baudrate 115200
set system scroll-line 255
set wlan0 off
set wlan0 RF-mode ax-b-g-n
set wlan0 ssid ATOS-NT:wlan0
set wlan0 nick ATOS-NT:wlan0-NICK
set wlan1 off
set wlan1 RF-mode ax-ac-an
set wlan1 ssid ATOS-NT:wlan1
set wlan1 nick ATOS-NT:wlan1-NICK
set wlan1 chan-width 160
set wwan loglevel 5
set wwan wwan0 loglevel 5
set wwan wwan0 type network-ifc
set wwan wwan0 apn voce
add interfaces IFC eth0-1 eth0-1
add interfaces IFC wwan0 wwan0
add interfaces IFC eth1 eth1
add interfaces IFC eth0-2 eth0-2
add interfaces IFC eth0-3 eth0-3
add interfaces IFC eth0-4 eth0-4
set interfaces eth0-1 ip address 192.168.1.1/24
set interfaces eth1 ip address 192.85.1.1/24
set security off
add ip route 0.0.0.0 0.0.0.0 wwan0 1
add napt IFC wwan0
add napt wwan0 PROXY 10000 192.85.1.5 10000 udp
set napt wwan0 on
set syslog loglevel 5
set voip sip loglevel 4
add voip user-terminal POTS pots1
add voip user-terminal POTS pots2
add voip trunk SIP 2
add voip trunk SIP 1
set voip trunk sip-2 on
set voip trunk sip-2 user-name 8765
set voip trunk sip-2 proxy-host 10.0.0.10
set voip trunk sip-2 sip-registration off
set voip trunk sip-1 on
set voip trunk sip-1 user-name 4321
set voip trunk sip-1 proxy-host 10.0.0.10
set voip trunk sip-1 sip-registration off
```

```

add voip call-mng OUTBOUND pots-line-2
ALL-NUMBER 0 NO-PREPEND sip-2 default
add voip call-mng OUTBOUND pots-line-1
ALL-NUMBER 0 NO-PREPEND sip-1 default
add voip call-mng INBOUND sip-2
ALL-NUMBER pots-line-2
add voip call-mng INBOUND sip-1
ALL-NUMBER pots-line-1

```

Le modifiche e le configurazioni descritte sopra rappresentano un passo significativo verso l'implementazione di un ambiente di test più realistico e rappresentativo delle condizioni operative effettive. L'utilizzo del router A TLC BG8520E ha permesso di superare le limitazioni del TestCenter, fornendo una piattaforma più versatile e potente per l'analisi delle prestazioni del network slicing. Si ottiene questa configurazione:

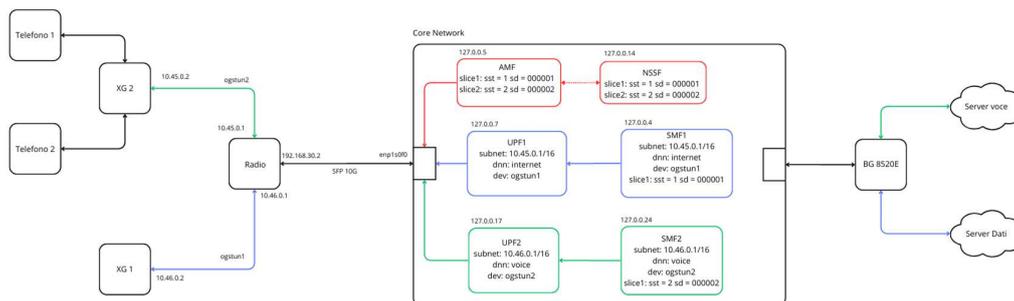


Figura 7.1: Test set traffico voce reale

7.2 Risultati ottenuti

Durante l'esecuzione di vari test, sono stati ottenuti risultati interessanti che confermano l'efficacia della configurazione adottata. I test di latenza hanno mostrato valori in linea con le aspettative e con quelli registrati con il TestCenter. Questo indica che la sostituzione del TestCenter con il router A TLC BG8520E non ha introdotto variazioni significative in termini di ritardi nella trasmissione dei pacchetti. La latenza costante è un segno positivo, poiché ga-

rantisce che le comunicazioni siano rapide e senza ritardi percettibili, essenziale soprattutto per le applicazioni in tempo reale come le chiamate vocali.

Un altro risultato notevole è stato il raggiungimento di una capacità di traffico dati massimo pari a 35 Mbps. Questo valore rappresenta una significativa quantità di dati gestiti dalla rete, dimostrando la capacità del sistema di supportare carichi di traffico elevati senza compromettere le prestazioni.

La parte più interessante dei test riguarda la qualità delle chiamate vocali. Le telefonate effettuate sono risultate pulite, senza disturbi o interruzioni, e senza latenza percepibile. Questo risultato è una chiara dimostrazione del fatto che la priorità assegnata al traffico voce è stata rispettata in maniera efficace. La capacità di garantire comunicazioni vocali di alta qualità è cruciale in molte applicazioni pratiche, e questi test confermano che la configurazione attuale è adeguata per questo scopo.

Utilizzando il software *Wireshark*, è stato possibile analizzare in dettaglio i pacchetti di dati in transito sulla rete. L'analisi ha rivelato che i due tipi di traffico, voce e dati, viaggiavano su reti separate. In particolare, il traffico RTP, relativo alle chiamate vocali, proveniva ed era indirizzato all'indirizzo IP 10.45.0.2, mentre il traffico dati utilizzava l'indirizzo IP 10.46.0.2. Questa distinzione è una prova evidente della creazione di due reti virtuali distinte, ciascuna dedicata a un tipo specifico di traffico. La separazione del traffico voce e dati su reti diverse è fondamentale per garantire che il traffico voce, che richiede maggiore priorità e minore latenza, non venga interferito dal traffico dati, che può essere più tollerante ai ritardi.

No.	Time	Source	Destination	Protocol	Length
3634	39.329802599	10.45.0.2	10.0.0.10	RTP	74
3635	39.329812395	10.45.0.2	10.0.0.10	RTP	74
3636	39.334547443	10.0.0.10	10.45.0.2	RTP	74
3637	39.354009687	10.0.0.10	10.45.0.2	RTP	74
3638	39.369836659	10.45.0.2	10.0.0.10	RTP	74
3639	39.369845621	10.45.0.2	10.0.0.10	RTP	74
3640	39.374014991	10.0.0.10	10.45.0.2	RTP	74
3641	39.394006539	10.0.0.10	10.45.0.2	RTP	74
3642	39.409934084	10.45.0.2	10.0.0.10	RTP	74
3643	39.409951274	10.45.0.2	10.0.0.10	RTP	74
3644	39.414016199	10.0.0.10	10.45.0.2	RTP	74

Figura 7.2: Traffico RTP analizzato con Wireshark

No.	Time	Source	Destination	Protocol	Length
2967...	2297.3844079...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3844745...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3845391...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3846046...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3846742...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3847396...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3848047...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3848740...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3849386...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3850033...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3850808...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3851445...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3852090...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3860720...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3861421...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3862067...	10.0.0.10	10.46.0.2	UDP	1390
2967...	2297.3862714...	10.0.0.10	10.46.0.2	UDP	1390

Figura 7.3: Traffico UDP analizzato con Wireshark

No.	Time	Source	Destination	Protocol	Length
2443...	2284.3967478...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3967546...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3978417...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3978495...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3978564...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3986433...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3986517...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3986586...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3997249...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3997325...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.3997395...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.4007168...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.4007276...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.4007547...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.4018328...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.4018475...	10.46.0.2	10.0.0.10	UDP	1390
2443...	2284.4018549...	10.46.0.2	10.0.0.10	UDP	1390

Figura 7.4: Traffico UDP analizzato con Wireshark

Si può concludere che i test effettuati con il dispositivo BG-850E hanno confermato l'efficacia del *Network Slicing* nel gestire traffico voce e dati in modo

separato e prioritario. La latenza stabile, la capacità di gestire un traffico dati massimo di 35 Mbps e la qualità impeccabile delle chiamate vocali sono risultati che dimostrano la robustezza e l'affidabilità della configurazione adottata. L'analisi con *Wireshark* ha ulteriormente validato la creazione di reti virtuali distinte, garantendo che il traffico voce e dati venga trattato in modo ottimale secondo le loro specifiche esigenze.

Conclusioni

I test condotti per confrontare le prestazioni delle radio X310 e B210 hanno rivelato diverse osservazioni significative. Nonostante la X310 sia dotata di un hardware superiore e una connessione più veloce tramite SFP, i risultati ottenuti non hanno mostrato i miglioramenti attesi rispetto alla B210, che utilizza un hardware meno prestante e una connessione USB.

In particolare a bassi valori di MCS, i comportamenti delle due radio sono praticamente identici. Questo suggerisce che, a livelli di modulazione e codifica meno complessi, le differenze nell'hardware e nelle connessioni delle due radio non influenzano significativamente le prestazioni. A valori elevati di MCS, si osservano differenze di latenza, in particolare in *uplink*, con la X310 che presenta prestazioni leggermente migliori. Questo indica che, sotto carichi di dati più intensi e complessi, l'hardware superiore e la connessione più veloce della X310 possono offrire un vantaggio in termini di latenza. Tuttavia, queste differenze non sono sufficientemente significative da giustificare l'aspettativa di un miglioramento sostanziale delle prestazioni complessive. Si è anche notato che aumentando la banda disponibile, i valori delle prestazioni rimangono invariati per entrambe le radio. Questo suggerisce che l'incremento della larghezza di banda non porta a miglioramenti significativi delle prestazioni, probabilmente a causa di altre limitazioni sistemiche o di rete. È possibile che il software utilizzato sia il vero collo di bottiglia, impedendo di sfruttare appieno le capacità hardware della X310 e di ridurre ulteriormente i valori di latenza. Un risultato

fondamentale è che, in tutte le condizioni testate, il numero di pacchetti ricevuti errati dalla radio X310 è pari a 0. Questo indica condizioni di comunicazione ottimali per la X310, contrastando con la B210, che mostrava percentuali di errori più elevate, specialmente ad alti valori di MCS. La maggiore affidabilità della X310 in termini di integrità dei pacchetti ricevuti rappresenta un notevole vantaggio per applicazioni che richiedono un alto grado di precisione e affidabilità.

La sperimentazione relativa allo *slicing* ha prodotto risultati estremamente positivi. Dopo uno studio approfondito del funzionamento dello *slicing*, è stato possibile allestire un banco di prova che ha dimostrato vari aspetti significativi della tecnologia.

Infatti lo *slicing* consente di gestire contemporaneamente due tipi di traffico differenti, permettendo una separazione logica e una gestione dedicata per ciascun tipo di traffico. Questo è particolarmente utile in scenari dove diverse tipologie di dati necessitano di trattamenti prioritari differenti. Inoltre impostando una priorità più alta per il traffico voce, si è verificato che questa viene rispettata. In particolare, la latenza e il *throughput* del traffico voce risultano sempre migliori rispetto a quelli del traffico dati, che ha una priorità inferiore. Questo dimostra che lo *slicing* permette una gestione efficace della qualità del servizio (QoS), garantendo che le comunicazioni vocali, che sono sensibili alla latenza, ricevano il trattamento prioritario necessario. Attraverso test eseguiti con due telefoni generanti traffico voce reale, si è potuto confermare che il traffico viene instradato in due sottoreti virtuali separate. Questo evidenzia l'efficacia dello *slicing* nel mantenere l'isolamento del traffico e nel garantire che i diversi tipi di dati non interferiscano tra loro, migliorando così la qualità complessiva del servizio e la possibilità di gestire i due flussi separatamente.

I test hanno dimostrato che, sebbene la radio X310 disponga di un hardware superiore e di una connessione più veloce, le prestazioni complessive non mi-

gliorano sensibilmente rispetto alla B210, ad eccezione di una gestione della latenza leggermente migliore in *uplink* a valori elevati di MCS. Tuttavia, la capacità della X310 di mantenere un numero di pacchetti errati pari a 0 rappresenta un vantaggio significativo in termini di affidabilità delle comunicazioni. Questo risultato suggerisce che, per applicazioni critiche in cui l'integrità dei dati è fondamentale, la X310 offre una maggiore affidabilità rispetto alla B210.

Inoltre, i test di *slicing* hanno confermato l'efficacia di questa tecnologia nella gestione di traffici diversi con differenti priorità, migliorando significativamente la qualità del servizio per il traffico voce rispetto al traffico dati. Questi risultati suggeriscono che lo *slicing* può essere una soluzione valida per ottimizzare le risorse di rete e migliorare l'esperienza utente nei sistemi di comunicazione avanzati.

In conclusione, la combinazione di hardware avanzato e tecnologie di *slicing* rappresenta un potenziale significativo per le reti del futuro, garantendo affidabilità, efficienza e qualità del servizio superiori. Questi aspetti sono fondamentali per lo sviluppo e l'implementazione di reti di comunicazione moderne e resilienti. Pertanto, ulteriori ricerche e sviluppi nel campo delle tecnologie di *slicing* e dell'ottimizzazione del software per sfruttare appieno le capacità hardware potrebbero portare a ulteriori miglioramenti nelle prestazioni delle reti di comunicazione.

Bibliografia

- [1] Salahdine et al. “5G, 6G, and Beyond: Recent advances and future challenges”. In: *Ann. Telecommun* 78 (2023), pp. 525–549. DOI: 10.1007/s12243-022-00938-3.
- [2] G. Manivasagam e M. Shukla. “Massive MIMO in 5G Networks: Trends and Challenges”. In: *2023 International Conference on Power Energy, Environment & Intelligent Control (PEEIC)* (2024). DOI: 10.1109/PEEIC59336.2023.10450543.
- [3] W. T. Chishiri e K. Kumar. “Analysis of Beamforming Performance in 5G Communication”. In: *2023 3rd International Conference on Intelligent Technologies (CONIT)* (2023). DOI: 10.1109/CONIT59222.2023.10205906.
- [4] Y. Niu et al. “A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges”. In: (2015). DOI: 10.1007/s11276-015-0942-z.
- [5] K. Carlberg, E. W. Burger e R. P. Jover. “Dynamic 5G Network Slicing for First Responders”. In: *2019 Principles, Systems and Applications of IP Telecommunications (IPTComm)* (2019). DOI: 10.1109/IPTCOMM.2019.8921240.
- [6] S. Lanka, T. A. Win e S. Eshan. “A review on Edge computing and 5G in IOT: Architecture & Applications”. In: *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (2021). DOI: 10.1109/ICECA52323.2021.9675934.
- [7] S. Jing e H. Wang. “Design and Implementation of a 5G Network Architecture Based on Software Defined Netw”. In: *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)* (2023), pp. 1444–1449. DOI: 10.1109/ICETCI57876.2023.10176812.
- [8] 3. G. P. P. (3GPP). “Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS); Stage 2 (Release 18)”. In: (2022).
- [9] G. BROWN. “Service-Oriented 5G Core Networks”. In: 2019.

- [10] A. Muhumed. “What are NSA and SA standards in 5G?” In: 2023. URL: <https://medium.com/@ahmedmuhumed/what-are-nsa-and-sa-standards-in-5g-e5403b415c49>.
- [11] H. Fehmi et al. “5G Network: Analysis and Compare 5G NSA /5G SA”. In: *Procedia Computer Science* 203 (2022), pp. 594–598. DOI: 10.1016/j.procs.2022.07.085.
- [12] D. Sattar e A. Matrawy. “Optimal Slice Allocation in 5G Core Networks”. In: *IEEE Networking Letters* 1.2 (2019), pp. 48–51. DOI: 10.1109/LNET.2019.2908351.
- [13] K. Carlberg, E. W. Burger e R. P. Jover. “Dynamic 5G Network Slicing for First Responders”. In: *2019 Principles, Systems and Applications of IP Telecommunications (IPTComm)* (2019), pp. 1–4. DOI: 10.1109/IPTCOMM.2019.8921240.
- [14] 5. Americas. “Commercializing 5G Network Slicing”. In: 2022. URL: <https://www.5gamericas.org/commercializing-5g-network-slicing/>.
- [15] H. T. Nguyen, T. V. Do e C. Rotter. “Scaling UPF Instances in 5G/6G Core With Deep Reinforcement Learning”. In: *IEEE Access* 9 (2021). DOI: 10.1109/ACCESS.2021.3135315.
- [16] L. Mamushiane. “Towards Stress Testing Open5GS Core (UPF Node) On A 5G Standalone Testbed”. In: *2023 IEEE AFRICON* (2023), pp. 1–6. DOI: 10.1109/AFRICON55910.2023.10293284.
- [17] K.-H. Phung et al. “User Plane Management Function: A Solution for Automatic Deployment of UPF on cloud-native 5G Core Network Architecture”. In: *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)* (2023), pp. 773–778. DOI: 10.1109/ICTC58733.2023.10392651.
- [18] “QoS parameters”. In: 2021. URL: <https://forum.huawei.com/enterprise/en/introduction-to-5g-qos-parameters/thread/667248098726002688-667213872962088960>.
- [21] A. Lupo. *NPN 5G PER INDUSTRIA 4.0: VERIFICA DELLE PRESTAZIONI E CONFRONTO CON WI-FI 6*. 2023.

Sitografia

- [19] *Open5GS Building Open5GS from Sources*. URL: <https://open5gs.org/open5gs/docs/guide/02-building-open5gs-from-sources/>.
- [20] *SRSRAN 5G Installation Guide*. URL: https://docs.srsran.com/projects/project/en/latest/user_manuals/source/installation.html.
- [22] *Slice configuration*. URL: https://github.com/s5uishida/open5gs_5gc_ueransim_snssai_upf_sample_config.