Brasil 6G

Projeto Brasil 6G

Relatório Técnico da Atividade 5.3 - Integração de Componentes, Ferramentas, Plataformas e Novas Implementações

01/12/2023

Histórico de Atualizações:

Versão	Data	Autor(es)	Notas
1	01/12/2023	Antonio Marcos Alberti (Inatel) Cristiano Bonatto Both (Unisinos) Ciro J. Almeida Macedo (UFG/IFG) Diego Gabriel Soares Pivoto (Inatel) Flávio de Oliveira Silva (UFU) Gustavo Araújo (RNP) Kleber Vieira Cardoso (UFG) Luciano Leonel Mendes (Inatel) Maurício Amaral Gonçalves (UFU) Michelle Soares Pereira Facina (CPQD) Natal Vieira de Souza Neto (UFU) Rodrigo Moreira (UFU) Rogério S. Silva (UFG/IFG) Sand Luz Correa (UFG) Tibério Tavares Rezende (Inatel)	Elaboração de conteúdo
2	01/12/2023	Luciano Leonel Mendes	Revisão de texto

Lista de Tabelas

Acrônimos

3GPP *3rd Generation Partnership Project*
5G Quinta Geração de Rede Móvel Celular
5GC 5G Core
6G Sexta Geração de Rede Móvel Celular
AD *Analog-to-Digital*
ADRF *Analytics Data Repository Function*
DA *Digital-to-Analog*
AI *Artificial Intelligence*
AMF *Access and Mobility Management Function*
AnLF *Analytics Logical Function*
API *Application Programming Interfaces*
AUSF *Authentication Server Function*
BS *Base Station*
CDN *Content Distribution Network*
CRR *Centro de Referência em Radiocomunicações*
DCCF *Data Collection Coordination Function*
DLT *Distributed Ledger Technologies*
DN *Data Network*
DNN *Data Network Name*
DNS *Domain Name System*
FL *Federated Learning*
FSO *Free-space Optical Communication*
GFDM *Generalized Frequency Division Multiplexing*
GRE *Generic Routing Encapsulation*
GTP-U *GPRS Tunnelling Protocol for User Data*
HNI *Home Network Identity*
HTTP *Hypertext Transfer Protocol*
IA Inteligencia Artificial
IKE *Internet Key Exchange*
IMSI *International Mobile Subscriber Identity*
IDU *Indoor Unit*
IP *Internet Protocol*
IPsec *Internet Protocol Security*
IPSec *Internet Protocol Security*

MAC *Media Access Control*

MCC *Mobile Country Code*

MFAF *Message Framework Adaptation Function*

MNC *Mobile Network Code*

MSIN *Mobile Subscriber Identification Number*

MTLF *Model Training Logical Function*

N3A *Non-3GPP Access*

N3IWF *Non-3GPP Interworking Function*

NTN *Non-Terrestrial Networks*

NSSF *Networking Slice Selection Function*

NAS *Non-Access Stratum*

NF *Network Function*

NGAP *NG Application Protocol*

non-3GPP *Non-Third Generation Partnership Project*

NRF *Network Repository Function*

NTN *Non-Terrestrial Network*

eNWDAF *Evolved Network Data Analytics Function*

NWDAF *Network Data Analytics Function*

OAM *Network Operations, Administration and Maintenance*

PCF *Policy Control Function*

PDU *Packet Data Unit*

PHY *Physical Layer*

PLMN *Public Land Mobile Network*

QoS *Quality of Service*

RF *Radio Frequency*

SA *Security Association*

SDN *Software Defined Networking*

SDR *Software Defined Radio*

SMF *Session Management Function*

SMO *Service Management and Orchestration*

SNR *Signal-to-Noise Ratio*

S-NSSAI *Network Slice Selection Assistance Information*

TCP *Transmission Control Protocol*

UDM *Unified Data Management*

UDP *User Datagram Protocol*

UDR *Unified Data Repository*

UE *User Equipment*

UE-non3GPP *User Equipment Non-Third Generation Partnership Project*

UPF *User Plane Function*

VLC *Visible Light Communications*

Wi-Fi *Wireless Fidelity*

WebUI *Web User Interface*

Sumário

1 Introdução

Nessa seção tem-se uma apresentação inicial dos resultados já obtidos na FASE 2 do projeto Brasil 6G no contexto da META 5, bem como um panorama das atividades realizadas.

1.1 Objetivos Gerais

O objetivo da META 5 do Projeto Brasil 6G FASE 2 é projetar soluções técnicas (habilitadores tecnológicos) integradas a partir do estado da arte de 5G/6G, incluindo alguns novos desenvolvimentos quando possível e necessário. Ao final das atividades, espera-se experimentar com elementos do núcleo, acesso e aplicações que façam sentido para uma arquitetura orientada ao 6G, testando-os e avaliando-os. O trabalho visa tirar proveito da flexibilidade inerente dos componentes softwarizados, aproveitando dos habilitadores orientados a serviços. Espera-se avaliar a interoperabilidade mínima com as redes de acesso e de transporte. Vale ressaltar que não teremos uma arquitetura completa de 6G, mas sim orientada ao 6G. Adicionalmente, poderão ser desenvolvidos componentes que permitam a integração dos elementos selecionados para experimentação, caso se mostre necessário.

1.2 Atividades Previstas

A META 5 se divide nas 3 atividades comentadas a seguir. Esse relatório cobre a Atividade 5.3 em sua totalidade. A Atividade 5.1 visou refinar o projeto do 6G que desejamos demonstrar. O entregável da Atividade 5.1 foi o relatório técnico entregue em 19/12/2022. A Atividade 5.2 visou definir como seria o 6G em termos dos componentes a serem implantados, suas interações, componentes adicionais, etc. Os resultados dessa atividade foram integrados ao relatório da Atividade 5.1

1.2.1 Atividade 5.3 - Integração de Componentes, Ferramentas, Plataformas e Novas Implementações

Essa atividade visa integrar os componentes do 6G em laboratório. Nesse relatório, reportamos todas as ações realizadas no escopo da Atividade 5.3. Essa atividade se embasa na plataforma alvo da META 2. Ela visa terminar as integrações em campo necessárias, preparar cadernos de experimentação, realizar testes e demonstrações. Formalmente, a Atividade 5.3 cobre:

- 5.3.A - Integrar e configurar os componentes, ferramentas e plataformas conforme selecionado na Atividade 5.2;
- 5.3.B - Implementar novos componentes especificados (se houverem).

1.3 Objetivos Específicos

Os seguintes objetivos específicos foram definidos para a META 5 da FASE 2 do Projeto Brasil 6G:

- Trazer componentes inovadores para o núcleo do 6G. Foram integrados componentes para fatiamento de rede, IA, satélite e acesso não-3GPP;

- Mostrar a integração de diversos acessos. Estamos trabalhando com o acesso não-3GPP via Wi-Fi, LoRa, Sigfox, Transceptor 6G MIMO-GFDM e satélite GEO;
- Integrar o transceptor MIMO-GFDM e seus componentes de *software*. Estamos integrando esse transceptor em mais de um ambiente de teste, como núcleo Proto 6G e diversas aplicações;
- Integrar *Open Radio Access Network* Open RAN com outros componentes. A integração com Open RAN não avançou como o esperado no contexto da META 5. Outras metas do projeto estão investigando tal integração;
- Identificar e resolver problemas do 5G. Alguns problemas foram identificados e estamos trabalhando para saná-los;
- Suportar diversas aplicações inovadoras de 6G. Ao todo 11 aplicações foram previstas para teste em 2023. Um subconjunto desse número foi efetivamente testado nos ambientes do projeto;
- Desenvolver componentes adicionais, se necessário. Estamos trabalhando no desenvolvimento e integração de dois novos componentes ao Proto 6G: a N3IWF e a NWDAF;
- Configurar e prover fatiamento de recursos. Duas soluções de fatiamento foram desenvolvidas: fatiamento fim a fim (entregue da Atividade 5.2 em 2022) e fatiamento como um serviço (entregue no escopo desse relatório);
- Integração de redes e componentes com satélite. Um esforço adicional foi realizado para integrar um satélite real da empresa Viasat ao ambiente do Proto 6G. Isso foi realizado em parceria com o SINDISAT e ABRASAT;
- Prover suporte à Inteligência Artificial (IA). A especificação de um novo componente para o core (NWDAF) foi realizada.

1.4 Conjunto de Ações

A partir desses objetivos específicos, um conjunto de ações foi criado para cumprir as Atividades 5.1, 5.2 e 5.3 da META 5. A Figura 1 ilustra como cada ação contribui nas Atividades da META 5. As Atividades 5.1 e 5.2 dependem somente das Ações 1 até 9. Tendo em vista que a Ação 10, prevista para a Atividade 5.3, refere-se à integração completa em laboratório, ela está diretamente relacionada com a Ação 5, responsável pelo suporte às aplicações da META 6, pois a infraestrutura criada em laboratório tem como objetivo, consequentemente, suportar as aplicações consolidadas para o projeto Brasil 6G. Sendo assim, as Ações 5 e 10 serão unificadas e discutidas em uma única seção neste relatório. Além disso, a Ação 11, que estava

Já a Ação 11, também prevista como parte da Atividade 5.3, cujo escopo e prazo de entrega estão no escopo desse relatório, será movida para o Relatório 6.2 do Projeto Brasil 6G. Como o Relatório 6.2 tem como foco discutir a integração das aplicações desenvolvidas no projeto, seu objetivo e resultados são alcançados a partir da Ação 11, relacionado à testes unitários, integrados, e demonstrações das aplicações.. A seguir detalhamos novamente todas as ações da META 5, além de citar a Ação 11, que será movida para o Relatório 6.2 do Projeto Brasil 6G:

- **Ação 1** - Arquitetura do 6G e Problemáticas do 5G: cuida dos aspectos macro do 6G;

- **Ação 2** - Núcleo do 6G: cuida do núcleo e possibilidades de integração, componentes, interfaces, relação com RAN, etc.;
- **Ação 3** - Acesso do 6G: cuida da rede de acesso e possibilidades de integração, componentes, interfaces, relação com núcleo, etc. Cobre RAN, *Open RAN* e transceptor MIMO-GFDM;
- **Ação 4** - Plataforma da META 2: cuida do relacionamento com a META 2. A plataforma do 6G é definida na META 2, mas os requisitos da META 5 são *inputs* importantes;
- **Ação 5** - Aplicações da META 6: cuida do relacionamento com a META 6. Os casos de uso são definidos na META 6, mas a META 5 deve suportá-los;
- **Ação 6** - Fatiamento fim-a-fim: cuida do fatiamento de recursos no 6G;
- **Ação 7** - Rede satelital: cuida do relacionamento com a rede satelital e sua integração com as outras redes e serviços;
- **Ação 8** - Inteligência Artificial: cuida da relação do núcleo, acesso e demais redes e serviços com IA;
- **Ação 9** - *Distributed Ledger Technologies* (DLT): cuida da relação dos componentes com *Blockchain*, IOTA, etc.;
- **Ação 10** - Integração completa em laboratório: cuida da integração final dos componentes em um ou mais *setups* de experimentação. Devido ao alinhamento com a Ação 5, esta ação será abordada em conjunto com a mesma, de forma unificada em uma única seção;
- **Ação 11** - Testes Unitários, Integrados e Demonstrações: cuida dos cenários de experimentação/testes e demostrações. Conforme previamente discutido, esta seção será movida para o Relatório 6.2 do Projeto Brasil 6G, dado seu alinhamento direto com o objetivo do mesmo.

Ação	Nome	5.1.A	5.1.B	5.1.C	5.1.D	5.2.A	5.2.B	5.2.C	5.2.D	5.2.E	5.3.A	5.3.B
1	Arquitetura do 6G e Problemáticas do 5G	x	x	x	x	x			x			
2	Núcleo do 6G			x	x	x	x	x	x		x	
3	Acesso do 6G			x	x	x	x	x	x		x	
4	Plataforma da META 2				x	x	x	x		x	x	
5	Aplicações da META 6				x		x				x	
6	Fatiamento fim a fim			x	x	x	x	x	x		x	
7	Rede Satelital			x	x	x	x	x	x		x	
8	Inteligência Artificial			x	x	x	x	x	x		x	
9	Distributed Ledger			x	x	x	x	x	x		x	
10	Integração Completa em Laboratório										x	x
11	Testes Unitários e Integrados											x

Figura 1: Divisão para realização das atividades da META 5 através das Ações 1 a 11.

1.5 Divisão de Responsabilidades

A seguinte divisão de responsabilidades foi democraticamente acordada para efetivar os trabalhos no contexto desse relatório:

- Alex Vidigal Bastos (UFSJ) - Responsável pela Ação 9;
- Antonio Marcos Alberti (Inatel) - Responsável pela META 5 e Ações 1, 4, 7, e 11;
- Cristiano Bonatto Both (Unisinos) - Responsável pela Ação 6;
- Diego Gabriel Soares Pivoto (Inatel) - Responsável pelas Ações 5 e 10;
- Flávio de Oliveira Silva (UFU) - Responsável pelas Ações 2 e 8;
- Kleber Cardoso (UFG) - Responsável pela Ação 3.

1.6 Estrutura do Restante do Documento

As Seções de 1 a 9 apresentam os resultados de cada Ação prevista no contexto da Atividade 5.3 e sua relação com outras metas do projeto, incluindo **novas implementações ou integrações** realizadas em 2023 em cada área da META 5. Em resumo, tem-se o seguinte:

- A Seção 2 descreve as atividades da Ação 1. Apresenta uma consolidação final de toda a arquitetura de 6G do projeto tal qual avançamos de 2022 até 2024. Os trabalhos realizados em 2022 no contexto de integração de toda arquitetura do Proto 6G se mostram bastante assertivos. Logo, a Seção 2 faz um apanhando dos resultados anteriores e atualiza para consolidar os resultados de 2023.
- Na Seção 3 são descritos os estudos e as atividades da Ação 2 que abordam o núcleo da arquitetura Proto 6G; nessa Seção 3 são destacadas os adendos e melhorias no núcleo realizadas no contexto da Atividade 5.3. A arquitetura Proto 6G adota o núcleo Free5GC, conforme relatórios anteriores. Na Seção 3 os avanços relacionados exclusivamente a instalação e execução desse núcleo no ano de 2023/2024 são reportados.
- A Seção 4 aborda a Ação 3 em relação à rede de acesso para o protótipo da arquitetura do 6G; essa Seção 4 descreve o suporte a múltiplas redes com a integração de diversas tecnologias de acesso 5G e os avanços realizados rumo ao Proto 6G. No ano de 2023, um grande esforço foi feito para integrar o acesso Não-3GPP ao núcleo Free5GC. A Seção 4 traz os detalhes desse esforço feito.
- A Seção 5 deste documento descreve os avanços realizados na META 5 em 2023 visando suporte às atividades da META 2 do projeto Brasil 6G FASE 2. Em resumo, descreve os ambientes de experimentação em uso de fato e a relação com a rede FIBRE da RNP, com enfase a ilha do Inatel.
- Na Seção 6 são relatados os trabalhos da Ações 5 e 10 no suporte de aplicações da META 6. Em 2022 foi realizado um trabalho de mapeamento de todas as aplicações a serem testadas no Proto 6G em 2023. A Seção 6 traz os avanços das Ações 5 e 10 através de trabalhos realizados para o desenvolvimento de uma infraestrutura de integração completa em trabalho que permite o suporte às aplicações previstas para a META 6. Os detalhes

de experimentação, demostração e avaliação de desempenho das aplicações serão apresentados no Relatório 6.2 do Projeto Brasil 6G, tendo em vista que a Ação 11, previamente prevista para este relatório, foi movida para o documento referente à META 6.

- A descrição e os trabalhos executados em fatiamento fim-a-fim de recursos físicos e cibernéticos em 2023 são apresentados na Seção 7. Em 2022, foram feitos experimentos de fatiamento de recursos com o núcleo Free5GC. Em 2023, o cenário de fatiamento foi expandido para suportar fatiamento como um serviço. Essa Seção 7 detalha esses avanços.

- A Seção 8 descreve os avanços da integração da rede satelital com o Proto 6G em 2023/2024. Os resultados da integração do Proto 6G a um enlace satelital físico oferecido pela Viasat no contexto do projeto SINDISAT C (em parceira com ABRASAT e Inatel) são reportados. Esse cenário é uma das aplicações do Proto 6G.

- Na Seção 9 são traçados os avanços relativos a integração de IA com os componentes do Proto 6G. Um levantamento de como a IA pode ser integrada ao Proto 6G foi realizado. A partir desse resultado, a implementação da função NWDAF do padrão 3GPP foi especificada.

- A Seção 10 apresenta as considerações finais desse relatório. Vale destacar que a Seção que tinha como finalidade descrever os avanços no suporte a DLTs na arquitetura 6G foi descontinuada devido a falta de atuação dos parceiros do projeto.

2 Ação 1 - Arquitetura do Proto 6G

Nesta seção, apresentamos uma consolidação final de toda a arquitetura Proto 6G. Iniciamos fazendo uma recapitulação dos principais avanços obtidos em 2022. Em seguida, apresentamos uma visão consolidada atual da arquitetura Proto 6G.

2.1 Principais Avanços Obtidos em 2022 nas Atividades 5.1 e 5.2

- **Ação 1** - Os princípios de projeto do Proto 6G foram revisitados, bem como os escopos de projeto. A partir destes, foi feita uma revisão da arquitetura evolucionária para o 6G proposta na primeira fase do Brasil 6G. Uma duzia de decisões de projeto foram tomadas em grupo democraticamente e reportadas neste relatório. Essas decisões guiaram os trabalhos do grupo em 2022. Por fim, uma visão que consolida as relações macro entre núcleo, acesso e aplicações foi feita para sumarizar o Proto 6G como um todo.

- **Ação 2** - Uma revisão do estado da arte em núcleos para o Proto 6G foi realizada. As alternativas Free5GC, Open5GS e CN5G (OAI) foram comparadas e analisadas considerando aspectos funcionais, técnicos, suporte a inteligência artificial, fatiamento de recursos, múltiplas redes de acesso, conformidade ao padrão 3GPP e robustez da implementação. A escolha do núcleo Free5GC como solução para evoluir o 5G na direção do 6G foi justificada. O principal motivo foi o suporte já implementando ao acesso não-3GPP, aspecto fundamental para acomodar os transceptores 6G do Inatel, bem como conectividade Wi-Fi, SigFox e LoRaWAN, importantes para os cenários de demonstração do projeto.

- **Ação 3** - Um estudo minucioso foi realizado para determinar as melhores opções de acesso para o 6G do Brasil. Caminhos preferenciais para integração do acesso via transceptor 6G de longa distância, acesso via Wi-Fi, e acesso de dispositivos IoT de baixo consumo de energia foram determinados e analisados. A escolha do acesso não-*3rd Generation Partnership Project* (3GPP) não-confiável foi o escolhido para integrar os transceptores 6G ao núcleo 3GPP. O enlace formado por esses transceptores irá fornecer um *backhaul* sem fio via *gateway* para transportar o tráfego das demais tecnologias, e.g., Wi-Fi, LoRa e Sigfox. Vale lembrar que esse caminho preferencial escolhido inicialmente não torna inviável a adoção futura de outros caminhos discutidos na Ação 3. Também é interessante observar que é possível conectar simultaneamente múltiplas tecnologias a um único transceptor 6G, assim como também é possível ter tecnologias diferentes conectadas a computadores diferentes, cada uma controlando um transceptor 6G do lado *User Equipment* (UE) diferente, ou seja, usar múltiplos computadores atuando como equipamento de usuário (UE). Todas as escolhas de caminhos foram justificadas e comparadas no texto.

- **Ação 4** - Essa ação é fundamental ao projeto, pois visa definir em conjunto com os integrantes da META 2 do projeto Brasil 6G quais são as demandas de núcleo, acesso, software, aplicações, redes, IoT, etc. para a plataforma de experimentação em 6G do projeto. Decidimos apoiar a plataforma do projeto Brasil 6G na rede FIBRE da RNP. Tal solução se justifica pela facilidade de interconexão, orquestração e operação já existentes na rede FIBRE e nas diversas ilhas das instituições participantes do projeto. Com isso em mente, adotamos o Kubernetes como principal ambiente para orquestração de funções virtuais de rede do Proto 6G. Analisamos, desenhamos e implementamos a plataforma que

une as instituições participantes, bem como os laboratórios internos do Inatel. O ICT Lab do Inatel foi escolhido como ponto de convergência da plataforma, facilitando integrações e demostrações do Proto 6G. O principal data center do projeto fica no *rack* do ICT Lab. Os laboratórios do Inatel foram interconectados por uma malha redundante de fibras ópticas. Um total de 6 servidores foram preparados conforme especificação da RNP na rede FIBRE. Esta é hoje a maior ilha da rede FIBRE no Brasil. Testes de funcionamento do core Free5GC executando distribuído em três ilhas foram realizados pela Unisinos, UFG e Inatel. A integração com a nuvem do projeto LaMCAD da UFG foi realizada. Além do acesso via FIBRE, o acesso por VPN ao ICT Lab para fins de configuração de cenários foi também implementado. O servidor principal do projeto Brasil 6G foi especificado através de um levantamento democrático de requisitos e a compra realizada. Entretanto, até o momento o mesmo ainda não foi entregue pelo fornecedor. Usando recursos de outros projetos, todos os servidores do ICT Lab foram protegidos por *nobreak*, condição fundamental para estabilidade da plataforma do Brasil 6G. Ainda, a empresa Vivavox de Santa Rita do Sapucaí oferece acesso a Internet de 100 Mbps necessários ao ICT Lab. A equipe da META 5 ainda ajudou na escrita dos relatórios da META 2.

- **Ação 5** - Essa ação também é fundamental ao projeto, pois visa definir quais são as demandas das aplicações da META 6 com relação as tecnologias existentes na META 5 e 2. Foi realizado um mapeamento de demandas para 10 aplicações da rede Proto 6G em relação ao núcleo de acesso. Além disso, é importante ressaltar que a rede não está limitada a essas aplicações, dado que outras aplicações serão implementadas futuramente. Com base nesse levantamento, uma visão detalhada de possíveis caminhos ótimos para integração de aplicações, rede de acesso e núcleo foi realizada. As demais ações da META 5 foram alinhadas e estão sendo desenvolvidas para suportar não apenas os requisitos das aplicações do 6G, como também tirar proveito dos recursos instalados na META 5 (sinalização 5G, novos componentes do núcleo, virtualização e orquestração de recursos, fatiamento fim a fim, integração com transceptores 6G, enlace de satélite e rede FIBRE da RNP). Nosso objetivo é maximizar as possíveis integrações de cenários e resultados do projeto. Vale ressaltar que, conforme previamente discutido, a Ação 10 será unificada com a Ação 5 neste relatório, dado que ambas as ações possuem pontos convergentes capazes de serem discutidos em uma única seção. Também é importante destacar que novas aplicações foram apontadas para serem desenvolvidas e consequentemente o mapeamento das demandas de aplicações foi atualizado, conforme será visto na Seção 2.2 deste relatório, e descrito com mais detalhes no relatório 6.1 da META 6 referente à Ação 11.

- **Ação 6** - O fatiamento da infraestrutura física e virtual de forma dinâmica e como um serviço é o alvo dessa ação. Em 2022, foi realizado um levantamento dos possíveis caminhos para introduzir o fatiamento fim a fim de recursos no Proto 6G. Embora o fatiamento fim a fim esteja padronizado pelo 3GPP, poucas são as soluções que o implementam de forma dinâmica. Essa é uma contribuição do projeto Brasil 6G. Realizamos um estudo das relações macro entre núcleo, acesso e aplicações para o suporte ao fatiamento dinâmico. Implementamos um cenário experimental de fatiamento dinâmico no LaMCAD da UFG. O protótipo testado integra um controlador, o núcleo do Proto 6G e o Kubernetes adotado na plataforma da META 2, provendo uma orquestração dinâmica total de fatiamento de rede. A ferramenta Terraform foi utilizada para a padronização do ambiente, provisionamento e replicação das máquinas virtuais. O gerenciamento de pacotes e atualização dos módulos de softwares utilizados foram realizados pela ferramenta Ansible,

que ficou responsável, por exemplo, pelas configurações de rede, Kubernetes, *Global System for Mobile Communications Tunnelling Protocol* (GTP) para 5G (GTP5G), etc. O Kubernetes foi configurado com um *Master* e dois *Nodes*. O protótipo suporta uma *Container Network Interface* (CNI) extra para gerenciar redes e sub-redes necessárias para a comunicação interna nos protocolos NGAP e *Non-Access Stratum* (NAS) com o núcleo, utilizando o projeto Free5GC. Por fim, um emulador de UEs e RAN para a geração de pedidos de conexão com o núcleo e tráfego de rede foi integrado no protótipo. O testador chamado my5G-RAN-tester foi escolhido para essa emulação. O sucesso dessa ação nos levou a adotar e amadurecer esse cenário como base para todo o projeto Brasil 6G. Estamos alinhando as demais ações com esse cenário, de forma a suportar fatiamento de recursos para algumas das aplicações que serão demonstradas em 2023. Ainda, estamos aprimorando esse cenário para oferecer fatiamento como um serviço (Slicing-as-a-Service), avançando o estado da arte da orquestração de recursos no 6G.

- **Ação 7** - A integração com satélite é fundamental para o Brasil, pais continental e carente de conectividade no campo. Nesse contexto, na Ação 7 realizamos diversos levantamentos de possíveis caminhos para integrar a rede satelital oferecida pelo Projeto C (parceria do Inatel com o SINDISAT) ao Projeto Brasil 6G. As demandas de componentes no acesso, núcleo e aplicações do Proto 6G foram estudadas para suportar integração com enlace satelital físico a ser oferecido pelo SINDISAT ao Inatel. Um ranqueamento de vantagens e desvantagens de cada caminho de integração foi fornecido. As possibilidades de caminho preferencial foram determinadas e estão em implementação, agregando valor a todo o projeto Brasil 6G. Dois caminhos foram estabelecidos: (i) *Untrusted* N3A com Satélite em Backhaul e Transceptor 6G; e (ii) *Untrusted* N3A com Satélite em Fronthaul e Transceptor 6G. Embora não seja mandatória, a integração com o Transceptor 6G do CRR Inatel é amplamente desejável para aumentar a área de cobertura da solução. Por solicitação do SINDISAT pretendemos implantar os dois caminhos. Entretanto, o caminho 1 está sendo implementado por primeiro. O caminho 2 será implementado havendo tempo hábil para tal. Nesse momento, estamos aguardando a instalação do enlace satelital físico no Inatel para dar continuidade nos caminhos 1 e 2. Essa Ação 7 resultou na Aplicação 11 (App 11), que se junta as outras dez na descrição da Ação 5.

- **Ação 8** - O estado da arte para integração da inteligência artificial no Proto 6G foi estudado nessa ação. Foi feita uma revisão sobre como o 3GPP pretende avançar com a IA a partir das *Releases* existentes e em preparação. Esse levantamento explorou em quais porções da arquitetura 5G/6G a IA tem sido empregada. As topologias lógicas de implantação da IA no plano de controle foram determinadas (centralizada, distribuída e híbrida). Essa topologias nos levaram aos possíveis caminhos para implantação da IA no Proto 6G. Escolhemos um caminho híbrido, ou seja, distribuído com alguns componentes centralizados. Esse caminho preferencial se relaciona com a função NWDAF que se encontra em padronização no 3GPP. Em 2023 pretendemos especificar uma arquitetura de IA para suporte ao plano de controle do Proto 6G.

- **Ação 9** - Por fim, na Ação 9 o foco foi na introdução das tecnologias de registro imutável descentralizado de informações e computação determinística através de contratos inteligentes em Blockchain e/ou Tangle. O estado da arte para integração de DLTs no Proto 6G foi estudado nessa ação em 2022. Foi feita uma revisão sobre como e se o 3GPP pretende avançar com DLTs nas *Releases* existentes e em preparação. Detectamos que existe

interesse do 3GPP nessas tecnologias, mas o atual estágio de desenvolvimento é anterior a qualquer especificação. A partir de uma revisão da literatura na área, determinamos os principais uso de DLT em 5G/6G, bem como um possível cenário de integração dessas tecnologias através de contratos inteligentes com a arquitetura Free5GC. Infelizmente, essa ação foi descontinuada em 2023.

2.2 Visão Consolidada do Proto 6G

A Figura 2 apresenta uma visão consolidada da arquitetura 6G proposta, incluindo aplicações da META 6, plataforma da META 2, e componentes do acesso e núcleo presentes na META 5. À esquerda são apresentadas as aplicações do 6G que serão executadas em uma fazenda na zona rural de Santa Rita do Sapucaí, MG. Elas se conectam por Wi-Fi, LoRa e Sigfox a Transceptores UE-T6G do Inatel. Esses transceptores implementam um emulador de estação de assinante da solução My5G Core, que é um *fork* do projeto *open source* Free5GC. Os UE-T6G fornecem a conectividade necessária da fazenda até a montanha das três torres (*three towers*), onde um Transceptor BS-T6G recebe o tráfego vindo da fazenda. Este transceptor também implementa a função N3IWF do My5G Core, que implementa parte do Proto 6G Core. Consequentemente, isso possibilita receber o tráfego proveniente das aplicações em uma rede de acesso *Untrusted Non-3GPP* e o entregar para o Servidor Brasil 6G que fica no ICT Lab do Inatel, onde por por sua vez está o restante das funções do núcleo My5G Core da rede.

Para tanto, todos os procedimentos de entrada na rede são feitos para as aplicações usando o emulador de UE do My5G Core. Depois de autorizado, o tráfego das aplicações segue para a UPF dentro do Proto 6G Core. A partir daí, três caminhos são possíveis: (i) aplicações locais rodando virtualizadas (App 1, 2, 7, 9, e 17) no mesmo Servidor Brasil 6G; (ii) aplicações rodando na Internet (3, 4, 5, 6, 8, 10, 12, 13, 14, 15, e 16), cujo tráfego escoa pela conexão patrocinada pela Operadora Vivavox no ICT Lab; e (iii) onde a App 11 utiliza a conexão da Internet através de um enlace satelital. No caso da App 11, o tráfego do BS-T6G pode ser enviado via núcleo da rede pelo enlace satelital em implantação no Inatel até um satélite físico GEO em parceria com o SINDISAT. Sendo assim, o parceiro SINDISAT pode, por exemplo, devolver o tráfego recebido no HUB satelital em Santa Rita do Sapucaí via rede cabeada.

Outras aplicações e cenários são possíveis com o WOCA, IoTRG e o próprio CRR do Inatel. A partir da rede FIBRE, componentes do Proto 6G Core podem ser distribuídos em outras ilhas nacionalmente. Até o momento componentes podem ser instanciados na Unisinos e UFG.

A arquitetura contém ainda componentes de *Service Management and Orchestration* (SMO) para orquestração de fatias fim a fim como um serviço, incluindo recursos de gerência de dados, *onboarding*, orquestração de funções virtuais de rede e laço de controle de qualidade. Vale observar que a App 11 inclui o acesso remoto a conteúdos e armazenamento temporário na fazenda, usando servidor NGINX, plataforma de NFV e controlador ONOS.

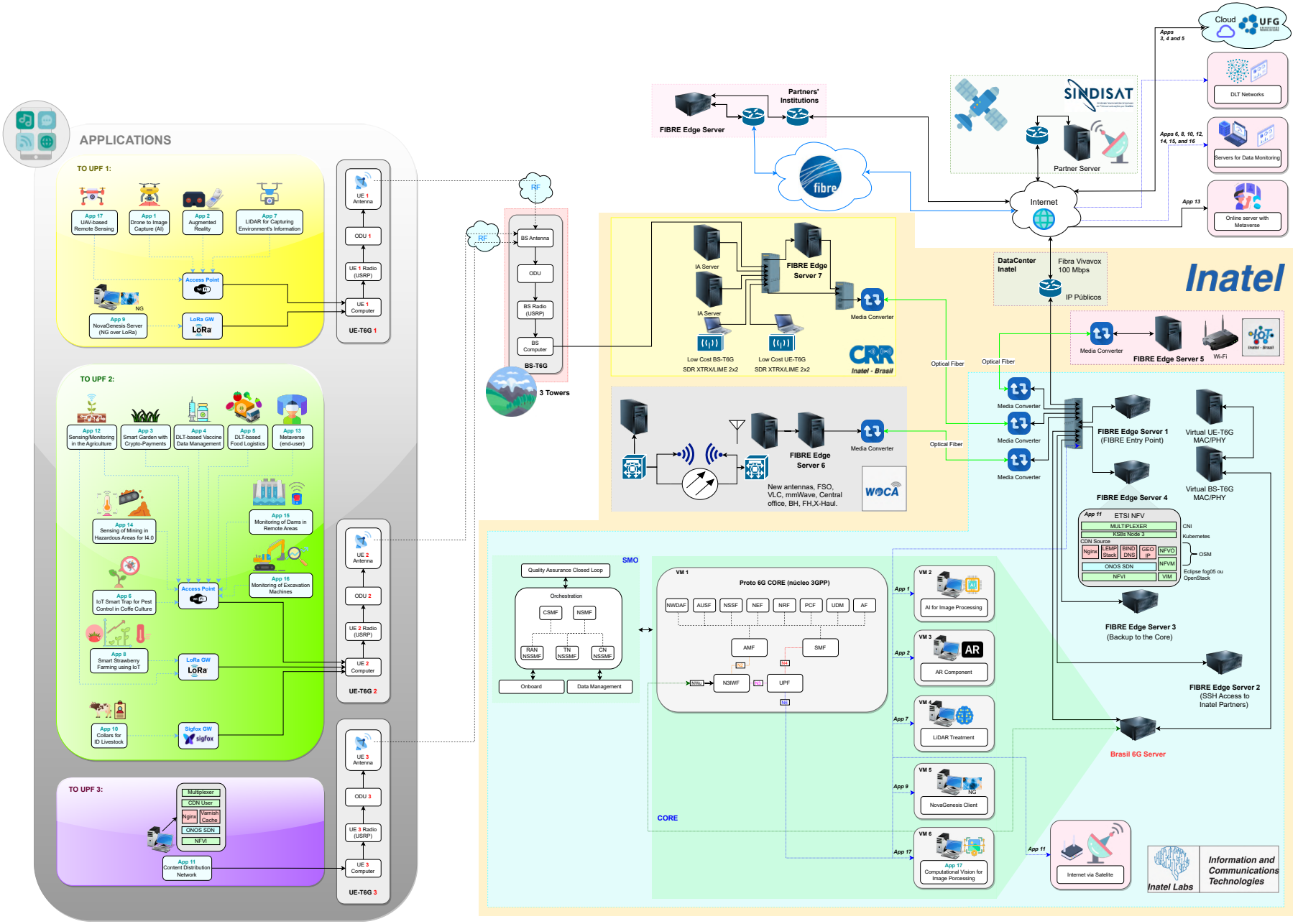


Figura 2: Visão consolidada da arquitetura 6G proposta nesse relatório. Autores: Diego Pivoto e Antônio Alberti.

3 Ação 2 - Núcleo do Proto 6G

Na Seção 3 são descritos os estudos e as atividades da Ação 2 que abordam o núcleo e um protótipo de arquitetura para 6G. São destacados os adendos e melhorias no núcleo realizadas no contexto da Atividade 5.3 (em 2023).

3.1 Visão Geral do Núcleo Proto 6G

O objetivo geral da **Ação 2** é ter como resultado o núcleo da Sexta Geração de Rede Móvel Celular (6G) levando em conta a visão do projeto Brasil 6G, aqui chamado *Proto 6G*. Nesse contexto, a **Ação 2** parte do pressuposto que o núcleo do 6G será construído tendo como base o núcleo da Quinta Geração de Rede Móvel Celular (5G). Este pressuposto está alinhado com o que se observa no 3GPP. O *Release 17* do 5G introduziu funcionalidades para o suporte para *Non-Terrestrial Network* (NTN) [1] e este suporte é ampliado no *Release 18* do 5G. No que diz respeito ao uso de *Artificial Intelligence* (AI) esta evolução do 5G também é observada com a *Network Data Analytics Function* (NWDAF) ao longo dos *Release 16*, *Release 17* e *Release 18* do 5G [2].

A **Ação 2** possui como ponto de partida uma implementação do 5G Core (5GC). Sendo assim, como resultado da avaliação do estado da arte referente às implementações disponíveis e com código aberto do 5GC a ação utiliza como base o core Free5GC que será modificado e ampliado ao longo do desenvolvimento do projeto Brasil 6G. A Figura 3 mostra como a **Ação 2** está articulada com as demais ações realizadas na META 5 do projeto Brasil 6G.

A **Ação 3** e **Ação 7** estão relacionadas com a capacidade do 6G de suportar diferentes tecnologias de acesso e neste caso o foco são as bandas não-licenciadas e também a integração com satélites. Por outro lado a **Ação 6** está relacionada com o capacidade de fatiamento suportada pelo Proto 6G. Por fim, a **Ação 8** propõe evoluir a visão atual de como a AI é utilizada no núcleo e distribuída ao longo da rede, chamada eNWDAF.

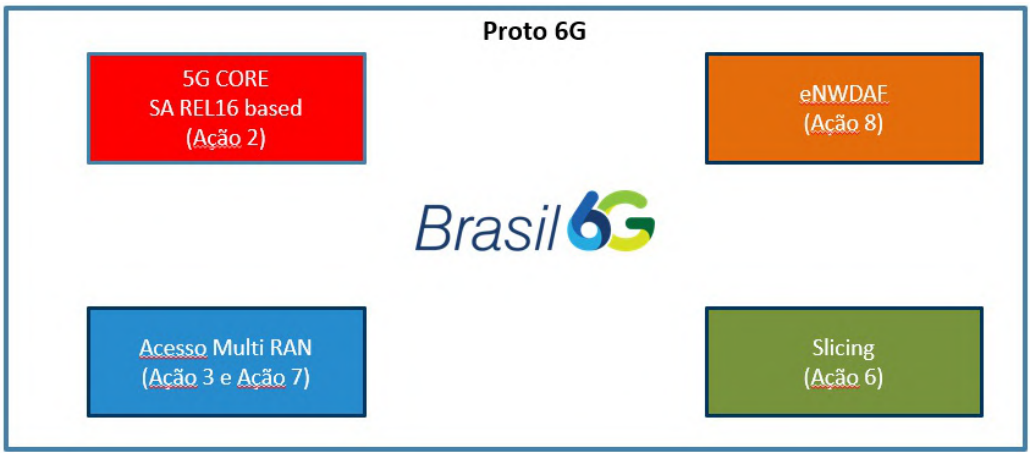


Figura 3: Correlação entre as Ações Relacionadas ao Núcleo do Proto 6G.

A realização da **Ação 2** está relacionada com os desenvolvimentos executados em cada uma destas ações e sua posterior integração ao longo do Projeto Brasil 6G. Logo, ao longo deste documento as seções seguintes apresentam estes desenvolvimentos em maior detalhe, principalmente no que diz respeito à Ação 3 (Seção 4), Ação 6 (Seção 7) e na Ação 7 (Seção 8) e na Ação 8 (Seção 9).

3.2 Contribuições para Cloudificação das Funções do Núcleo

As redes celulares móveis tradicionais consistem em poucos componentes que implementam funções de rede monolíticas e utilizam interfaces padronizadas para comunicação com funções externas. Esta abordagem garantiu a confiabilidade e a resiliência exigidas por essas funções, consumindo diretamente os recursos fornecidos pelos servidores físicos. Entretanto, os operadores e fornecedores de telecomunicações encontram dificuldades em atualizar os sistemas celulares já existentes, devido à natureza das funções monolíticas que compõem o seu sistema. Por outro lado, a computação nativa em nuvem é uma solução consolidada para o desenvolvimento de software no mundo da Tecnologia de Informação. De acordo, com a *Cloud Native Computing Foundation* (CNCF), computação nativa em nuvem é uma abordagem que serve para construir e executar aplicações escaláveis, resilientes e observáveis em ambiente de nuvem. Esses conceitos e tecnologias são também desejáveis em redes celulares móveis para prover aplicações baseada em:

- Funções de rede escaláveis por meio da decomposição em micros serviços, considerando escalabilidade horizontal e vertical.
- Cloud native Network Functions (CNFs) resilientes, através da simplicidade de substituir e restaurar micros serviço com falha ao invés de uma aplicação inteira.
- Encapsulamento dos micros serviços que compõem uma função de rede e suas dependências em contêineres, para oferece melhor portabilidade na infraestrutura de rede, permitindo um consumo eficiente de recursos.

Os projetos de código aberto para núcleos de redes de celulares móveis, tais como open5GS[1] e free5GC[2], já são baseados em arquiteturas de micros serviços (Service-Based Architecture - SBA). Entretanto, suas implementações não são projetadas para serem executadas em ambientes completamente voltados para computação nativa em nuvem. Por exemplo, para as funções do núcleo baseadas em micros serviços poderem ser instanciadas em qualquer nodo de um cluster Kubernetes, é preciso escrever todas as metas informações de configurações (redes, sistemas operacionais, orquestração, etc.) em arquivos específicos como por exemplo JavaScript Object Notation (JSON) e portar para ferramentas de automatização, como Helm[3]. Desta forma, no projeto Brasil 6G, foi projetado e desenvolvido uma solução para suportar funções de núcleo SBA em computação nativa em nuvem.

5G-all-in-one-helm é um projeto de código aberto implementado em Helm charts para implantar automaticamente um sistema composto por um núcleo SBA e uma RAN para redes celulares móveis, voltado para computação nativa em nuvem. Nesse contexto, foi utilizado exclusivamente *plugins* de rede nativos do kubernetes, como por exemplo Calico[4]. O sistema suporta qualquer núcleo e RAN, desde que baseados no modelo SBA. Atualmente, o projeto Brasil 6G optou por utilizar o projeto free5GC para o núcleo e os projetos UERANSIM e my5G-RANTester para emular a RAN. Informações sobre o projeto *5G-all-in-one-helm* estão disponíveis no GitHub[5].

[1] https://open5gs.org/
[2] https://free5gc.org/
[3] https://helm.sh/
[4] https://docs.tigera.io/calico/latest/about
[5] https://github.com/zanattabruno/5G-all-in-one-helm/tree/main

4 Ação 3 - Acesso do Proto 6G

A Seção 4 aborda a Ação 3 em relação à rede de acesso para o protótipo da arquitetura do 6G; essa Seção 4 descreve o suporte a múltiplas redes com a integração de diversas tecnologias de acesso 5G e os avanços realizados rumo ao Proto 6G. Conforme descrito na Seção 4 do Relatório Técnico 5.1 e 5.2 [3], a integração de redes de acesso sem fio com o núcleo pode ser caracterizada como confiável ou não confiável. A principal diferença entre o que é acesso confiável e acesso não-confiável, está associada à relação que existe entre o ponto de acesso não-3GPP e o núcleo. O componente responsável por prover suporte a uma rede de acesso não-3GPP não-confiável e o núcleo é a função *Non-3GPP Interworking Function* (N3IWF). Essa função atua como um intermediário entre os dispositivos que utilizam tecnologias *Non-Third Generation Partnership Project* (non-3GPP), (ex. dispositivos que operam em redes *Wireless Fidelity* (Wi-Fi) ou outras tecnologias sem fio), e a rede principal.

No contexto do Proto 6G, o acesso ao núcleo descrito na Seção 3 é realizado através da função N3IWF. Neste sentido, o que chamamos de *acesso*, trata-se de um artefato de software [4], desenvolvido no âmbito da Ação 3 intitulado *User Equipment Non-Third Generation Partnership Project* (UE-non3GPP). Conforme ilustrado na Figura 4, este artefato é capaz de se conectar ao núcleo da rede através da função N3IWF, provendo uma integração segura e eficiente entre as aplicações descritas na Ação 5 desse relatório.

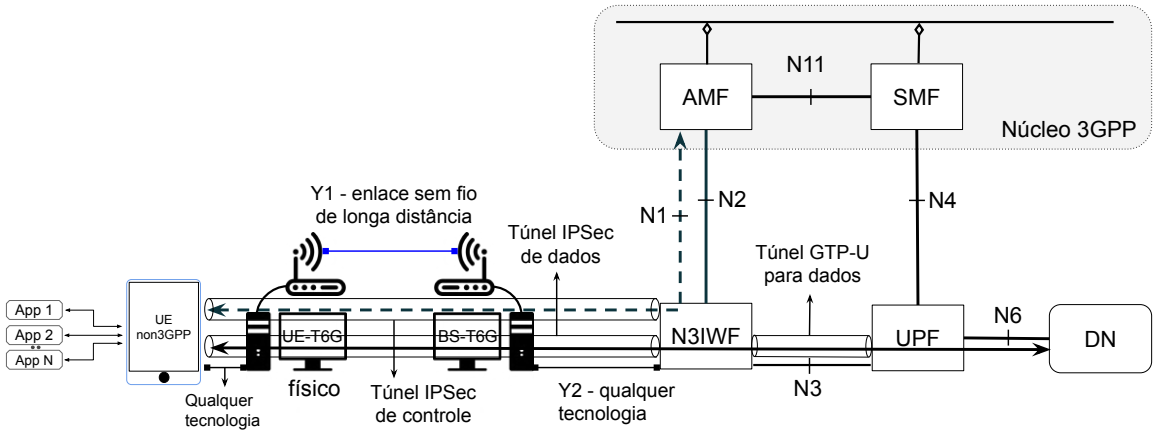


Figura 4: Comunicação vinda das aplicações através do acesso não-3GPP não-confiável com UE (lógico) em dispositivo extra.

4.1 Experimentação Exclusiva com o Acesso do Proto 6G

Essa subseção apresenta um detalhamento sobre como experimentar na prática com o acesso do Proto 6G no ambiente do projeto Brasil.

4.1.1 Pré-requisitos, Instalação do Cenário e Configuração

A seguir serão descritos elementos que são pré-requisitos necessários à execução do UE-non3GPP, os procedimentos de instalação e configuração e as etapas de teste, para validar a instalação.

Pré-requisitos

Para utilizar o UE-non3GPP, os seguintes pré-requisitos devem ser atendidos:

- A máquina onde será instalado o UE-non3GPP deve estar devidamente configurada com o Sistema Operacional Ubuntu 20.04 (LTS) x64, possuir no mínimo de 1G de memória RAM, possuir 2GB de espaço em disco, e ter a linguagem de programação *Go* versão *1.14.4* devidamente instalada.

- O núcleo free5GC [5] deve estar devidamente configurado e com as funções *Unified Data Repository* (UDR), *Unified Data Management* (UDM), *User Plane Function* (UPF), *Session Management Function* (SMF), *Policy Control Function* (PCF), *Networking Slice Selection Function* (NSSF), *Network Repository Function* (NRF), *Authentication Server Function* (AUSF) e *Access and Mobility Management Function* (AMF) em execução.

- A função N3IWF deve estar devidamente configurada e conectada ao núcleo free5GC [5] da seguinte forma: (i) à SMF através da interface N2 e (ii) à UPF através da interface N3. A N3IWF deve possuir ainda, 2 interfaces de rede visíveis ao UE-non3GPP. A primeira destinada à comunicação do tipo *Internet Key Exchange* (IKE) e a segunda destinada à comunicação *Internet Protocol Security* (IPsec), conforme ilustrado na Figura 4.

- As credenciais de acesso do UE-non3GPP devem estar devidamente registradas na base de dados do free5GC. O registro pode ser realizado através de *Web User Interface* (WebUI) provida pelo próprio free5GC, ou através da execução de script disponível no repositório do projeto [4] (*include_ue.sh*).

Instalação e configuração UE-non3GPP

Para instalar o UE-non3GPP, deve-se clonar o repositório do projeto [4] e em seguida informar os parâmetros de configuração que estão localizados no arquivo `config.yml`, localizado no diretório `config`, conforme descrito a seguir:

- **permanentKeyValue** - Valor-chave permanente usado nas operações de autenticação e identificação do UE-non3GPP nas operações das funções do plano de controle no núcleo da rede. Deve ser configurado com um valor hexadecimal equivalente ao valor-chave registrado na base de dados do free5GC [5].

- **opValue** - Parâmetro utilizado para calcular valores temporários, gerados pelo plano de controle, durante o processo de autenticação do UE-non3GPP com o núcleo da rede. Deve ser configurado com o valor hexadecimal equivalente ao valor registrado na base de dados do free5GC [5].

- **opcValue** - Parâmetro complementar ao **opValue**. Também é utilizado para calcular valores temporários gerados pelo plano de controle, durante o processo de autenticação do UE-non3GPP com o núcleo da rede. Deve ser configurado com o valor hexadecimal, equivalente ao valor registrado na base de dados do free5GC [5].

- **sequenceNumber** - Número sequencial que atua como contador individual para cada UE-non3GPP. Deve ser configurado com o valor hexadecimal, equivalente ao valor registrado na base de dados do free5GC [5].

- **msin** - Parâmetro de identificação que juntamente com *Mobile Country Code* (MCC) e *Mobile Network Code* (MNC) compõem o *International Mobile Subscriber Identity* (IMSI), responsável por identificar o UE-non3GPP no núcleo da rede. Deve ser configurado com

um valor numérico, equivalente ao valor registrado na base de dados do free5GC [5]. Além de estar devidamente cadastrado na base de dados do free5GC [5], o IMSI do UE-non3GPP deverá estar contido em algum bloco de configuração UEROUTINGINFO do arquivo UEROUTING.YAML contido na SMF. O UEROUTING.YAML é um arquivo que especifica a faixa de endereços *Internet Protocol*s (IPs), a qual será atribuída ao endereço do UE-non3GPP; sendo a UPF responsável por atender as requisições do UE-non3GPP após o construção do túnel de dados.

- **mcc** - Parâmetro utilizado para identificar o país ao qual o UE-non3GPP pertence. Juntamente com *msin* e MNC compõem o IMSI que identifica o UE-non3GPP no núcleo da rede. O valor atribuído ao MNC deve ser equivalente aos MCCs contidos nos arquivos de configuração das funções N3IWF, SMF, AMF, NSSF, AUSF, NRF.
- **mnc** - Parâmetro utilizado para identificar a identidade da rede móvel. A combinação entre MCC e MNC gera o *Home Network Identity* (HNI).
- **ranuengapid** - Identificador do UE na rede de acesso sobre a interface N2 que liga N3IWF e AMF [6].
- **amfuengapid** - Identificador de associação entre UE e AMF sobre a interface N2 que liga N3IWF e AMF [6].
- **authenticationmanagementfield** - Valor numérico utilizado no gerenciamento de autenticação. Deve ser configurado com o valor numérico equivalente ao valor registrado na base de dados do free5GC [5].
- **localpublicipaddr** - Deve ser configurado com o endereço IP local da máquina onde está sendo executado o UE-non3GPP. O valor informado será utilizado para a construção de um túnel de conexão *User Datagram Protocol* (UDP) com a N3IWF responsável por comunicação do tipo IKE. É através desta conexão que acorre as trocas de chaves entre as funções do plano de controle e o UE-non3GPP, para a que os túneis ilustrados na Figura 4 possam ser estabelecidos.
- **localpublicportudpconnection** - Parâmetro complementar ao *localpublicipaddr*. Deve ser configurado com um valor numérico que representará a porta para conexão UDP entre UE-non3GPP e N3IWF.
- **linkgre.name** - Este parâmetro será utilizado para definir o nome da interface de rede do túnel responsável pelo plano de dados no UE-non3GPP. O nomenclatura da interface se dará através do valor atribuído ao parâmetro *name* concatenado a um valor numérico.
- **ipsecinterfacename** - Este parâmetro será utilizado para definir o nome da interface de rede do túnel *Internet Protocol Security* (IPSec) responsável por trafegar os dados do plano de controle no UE-non3GPP.
- **ipsecinterfacemark** - Valor numérico que representa a marcação da interface virtual IPSec no túnel responsável por trafegar os dados do plano de controle entre UE-non3GPP e N3IWF. O valor atribuído ao parâmetro deve ser equivalente ao valor definido no parâmetro *IPSecInterfaceMark* contido no arquivo de configuração da N3IWF `n3iwfcfg.yaml`. Este valor será utilizado pela N3IWF para atribuir segurança aos pacotes IPs trafegados com o UE-non3GPP.

- **sst** - Este parâmetro será utilizado pelas funções N3IWF, NSSF e SMF para caracterizar o tipo de serviço do *Slice* ao qual a comunicação do UE-non3GPP será atribuída. O valor atribuído ao parâmetro deverá ser equivalente ao valor contido no arquivo de configuração da N3IWF `n3iwfcfg.yaml`.

- **sd** - Este parâmetro será utilizado pelas funções N3IWF, NSSF e SMF em conjunto com o parâmetro *sst*, para diferenciar o tráfego de um mesmo tipo de serviço dentro de um *Slice*. O valor atribuído ao parâmetro deverá ser equivalente ao valor contido no arquivo de configuração da N3IWF `n3iwfcfg.yaml`.

- **pdusessionid** - Este parâmetro será encaminhado pelo UE-non3GPP à AMF através da N3IWF com objetivo de solicitar uma seção inicial de *Packet Data Unit* (PDU). O valor da parâmetro não deve ser alterado.

- **ikebindaddress** - Este parâmetro deve ser configurado com o endereço IP da máquina onde está sendo executada a N3IWF. O valor deverá ser equivalente ao endereço configurado no atributo *IKEBindAddress* contido no arquivo de configuração da N3IWF `n3iwfcfg.yaml`.

- **ikebindport** -Este parâmetro representa a porta através da qual a N3IWF irá responder as comunicações do tipo IKE do UE-non3GPP. É uma porta padrão especificada pelo 3GPP e o valor 500 não deve ser modificado.

Após configurar cada um dos parâmetros necessários para conectar o UE-non3GPP à N3IWF, basta inicializar a execução do agente responsável por construir os túneis de comunicação dos planos de dados e de controle, ilustrados na Figura 4. A inicialização deve ser realizada a partir do diretório base utilizado na etapa de clonagem do repositório do UE-non3GPP com privilégios de administrador (sudo su). A inicialização pode ser realizada com o seguinte comando: `go run cmd/main.go ue`. Após a execução, duas interfaces de rede deverão ser criadas no UE-non3GPP, conforme ilustrado na Figura 5.

```
gretun1: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 1438
        inet 10.60.0.1  netmask 255.255.255.255  destination 10.60.0.1
        inet6 fe80::a00:32  prefixlen 64  scopeid 0x20<link>
        unspec 0A-00-00-32-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 0  (UNSPEC)
        RX packets 59  bytes 8644 (8.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 70  bytes 5595 (5.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ipsec0-default: flags=193<UP,RUNNING,NOARP>  mtu 1478
        inet 10.0.0.50  netmask 255.255.255.0
        inet6 fe80::4380:a3ab:2952:fc4c  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 0  (UNSPEC)
        RX packets 117  bytes 13463 (13.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 129  bytes 10699 (10.6 KB)
        TX errors 8  dropped 8 overruns 0  carrier 8  collisions 0
```

Figura 5: Interfaces de rede responsáveis pelo tráfego de comunicação dos planos de dados e de controle, geradas no UE-non3GPP após a execução do comando de inicialização.

4.1.2 Caderno de Experimentação

Conforme ilustrado na Figura 4 e descrito na Seção 4 do Relatório Técnico 5.1 e 5.2 [3], o UE-non3GPP é um artefato de software responsável por prover acesso a uma determinada rede de dados através de um núcleo de 5G, por intermédio da função N3IWF. Neste sentido, após a execução das etapas descritas nas seções anteriores, dois experimentos principais podem ser realizados visando aferir o correto funcionamento do UE-non3GPP, conforme descrito a seguir.

Acesso à rede de dados

O primeiro experimento tem como objetivo verificar a existência de conectividade entre o UE-non3GPP e algum outro dispositivo, posicionado na rede de dados interligada pela UPF, por intermédio da N3IWF. Supondo que a UPF em questão, forneça acesso à Internet, o resultado de um teste `PING` ao endereço IP 8.8.8.8 (google.com) utilizando a interface de rede `gretun1`, pode indicar a existência ou não existência de conectividade. O teste pode ser executado através do seguinte comando: `ping -I gretun1 8.8.8.8`. O resultado esperado é o equivalente ao ilustrado na Figura 6, que indica o *reply*, executado por parte do equipamento de destino. Essa resposta demonstra a existência de conectividade entre UE-non3GPP e o dispositivo posicionado na rede de dados, interligada pela UPF.

```
root@ictlab-OptiPlex-960:/home/uecore/brasil6g/UE-non3GPP-v1# ping -I gretun1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 10.60.0.1 gretun1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=19.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=18.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=21.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=20.3 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=116 time=19.1 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=116 time=18.2 ms
```

Figura 6: Resultado esperado de um teste de PING executado na interface de rede `gretun1` do UE-non3GPP, direcionado a um endereço IP posicionado na rede de dados.

Rastreamento da rota dos pacotes IP

O experimento que demonstra o acesso à rede de dados, possibilita verificar a existência de conectividade entre UE-non3GPP e o dispositivo interligado pela UPF. Conforme ilustra a Figura 4, o cenário de utilização do Proto 6G envolve um conjunto de elementos de software, distribuídos em diferentes máquinas. Neste sentido, para aferir o correto funcionamento do plano de dados do UE-non3GPP, faz-se necessário verificar as rotas utilizadas pelos pacotes IP que trafegam na interface de rede `gretun1`. Supondo que N3IWF está executando em uma máquina dedicada, e que as demais funções do núcleo 3GPP estejam executando e uma outra máquina distinta, um rastreamento de rota realizado nos pacotes que trafegam pela interface `gretun1` deve indicar a existência de um único salto entre o endereço IP de onde está sendo executado o UE-non3GPP e os demais saltos existentes na Internet. E este único salto, presente no rastreamento das rotas, deve possuir endereço IP equivalente ao endereço da interface N6, ilustrada na Figura 4, que conecta a UPF à rede de dados.

O teste pode ser executado através do comando `traceroute`, parametrizado com o endereço IP atribuído à interface de rede `gretun1`. O comando completo ficaria da seguinte forma:

`traceroute -s «ip-gretun1» 8.8.8.8`, onde o marcador `«ip-gretun1»` deverá ser substituído pelo endereço de IP atribuído pelo UE-non3GPP à interface `gretun1`. Supondo que o endereço IP da interface `gretun1` do UE-non3GPP, seja equivalente ao IP ilustrado na Figura 5, o rastreamento da rota dos pacotes IP até o host `8.8.8.8` pode ser realizado da seguinte forma: `traceroute -s 10.60.0.1 8.8.8.8`. O resultado esperado é o equivalente ao ilustrado na Figura 7. O endereço IP com final 146, destacado na imagem é o endereço de IP da interface N6 que faz a interligação entre UPF e a rede de dados. Todos os demais elementos computacionais intermediários, com suas respectivas interfaces de rede foram suprimidos, o que demonstra um funcionamento coerente do plano de dados resultante da ligação UE-non3GPP e núcleo 3GPP através da N3IWF.

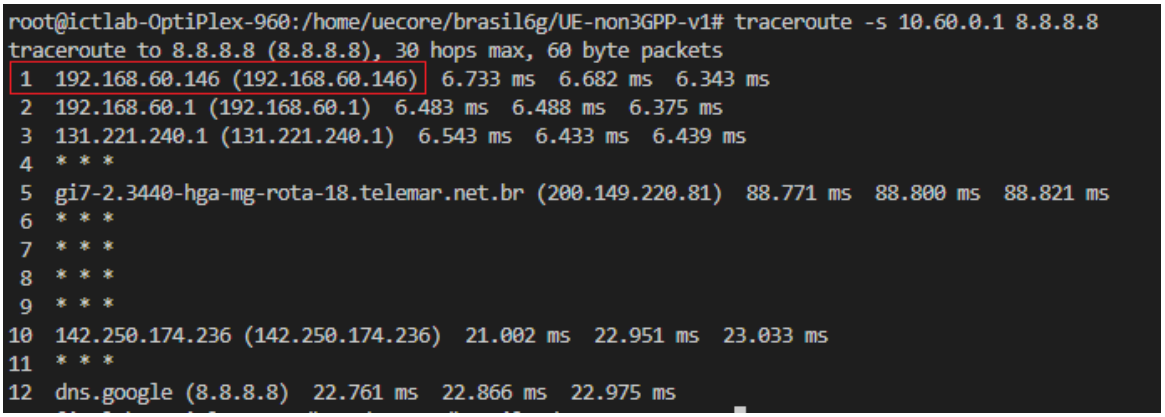


Figura 7: Resultado esperado de um teste de rastreamento de pacotes IP executado na interface de rede `gretun1` do UE-non3GPP.

Como elemento adicional às informações detalhadas neste relatório, construímos um repositório [6] contendo uma documentação detalhada, voltado para a execução dos experimentos do acesso ao Proto 6G. Seguindo os passos descritos no repositório é possível implantar um núcleo 3GPP, configurar uma função N3IWF devidamente conectada ao núcleo, e finalmente implantar um UE-non3GPP e executar todos os testes descritos neste relatório.

4.2 Lições Aprendidas e Ações Futuras

A estrutura do free5GC [5] tem com base uma arquitetura de microsserviços. Isso significa que cada uma das funções de rede pode ser instalada em um computador, máquina virtual ou contêiner de forma independente. Esse tipo de arquitetura tem se destacado como uma abordagem eficaz para o desenvolvimento e manutenção de sistemas de *software*, oferecendo diversas vantagens, dentre as quais se destacam a (i) escalabilidade vertical, i.e., a capacidade de aumentar o desempenho de um componente específico do sistema, adicionando recursos computacionais como processamento e memória; e a (ii) escalabilidade horizontal, i.e., a capacidade de aumentar o desempenho do sistema através da distribuição da carga entre instâncias de serviços distintos.

Considerando as premissas de uma arquitetura de microsserviços e a utilização do free5GC [5] como núcleo 5G no escopo deste projeto, uma das atividades executadas foi a de *conteinerizar* suas funções de rede e prepará-las para uso no Kubernetes [7]. Assim, as funções (UDR, UDM,

[6] https://github.com/LABORA-INF-UFG/Proto 6G-Install

UPF, SMF, PCF, NSSF, NRF, AUSF, AMF e N3IWF) foram convertidas em pequenos artefatos de *software*, implantáveis e gerenciáveis através de Kubernetes, conhecidas como *pods*. No entanto, durante o desenvolvimento do UE-non3GPP, foram identificadas algumas limitações técnicas, que exigiram o uso do free5GC fora do Kubernetes [7], conforme descrito a seguir.

Túneis IPSec entre UE-non3GPP e N3IWF

No contexto do Proto 6G, o acesso ao núcleo é realizado através da função N3IWF por intermédio do UE-non3GPP. Conforme descrito nas seções anteriores, o principal objetivo é o de prover acesso ao núcleo a dispositivos que utilizam tecnologias non-3GPP. Neste contexto, o enlace entre UE-non3GPP e N3IWF é o responsável por garantir todos os requisitos de segurança no processo de comunicação de dados. Conforme ilustrado na Figura 4, isso ocorre através da construção de 2 túneis IPSec. O primeiro para o tráfego da comunicação de dados e o segundo para o tráfego da comunicação de controle. O processo de construção desses túneis envolve um conjunto de rotinas a serem executadas, tando do lado do UE-non3GPP quanto do lado da N3IWF. Algumas dessas rotinas envolvem diretamente o núcleo do sistema operacional, por exemplo, é necessário utilizar o arcabouço XFRM [8] em ambos os lados, visando definir corretamente as políticas para realizar a criptografia/decriptografia dos pacotes trafegados pelos túneis de comunicação.

Durante o desenvolvimento do UE-non3GPP constatamos que ao implantar a N3IWF como uma Kubernetes, as rotinas envolvendo as políticas XFRM, do lado da N3IWF, não são devidamente aplicadas pelo sistema operacional. Os testes realizados sugerem que o fluxo para construção dos túneis IPSec tenta definir as políticas XFRM diretamente no sistema operacional onde o Kubernetes está instalado. No entanto, a N3IWF não está implantada diretamente no sistema operacional, ela está dentro de uma *pod* do Kubernetes e, portanto, esse deveria ser o responsável por executar as rotinas de criptografia da N3IWF.

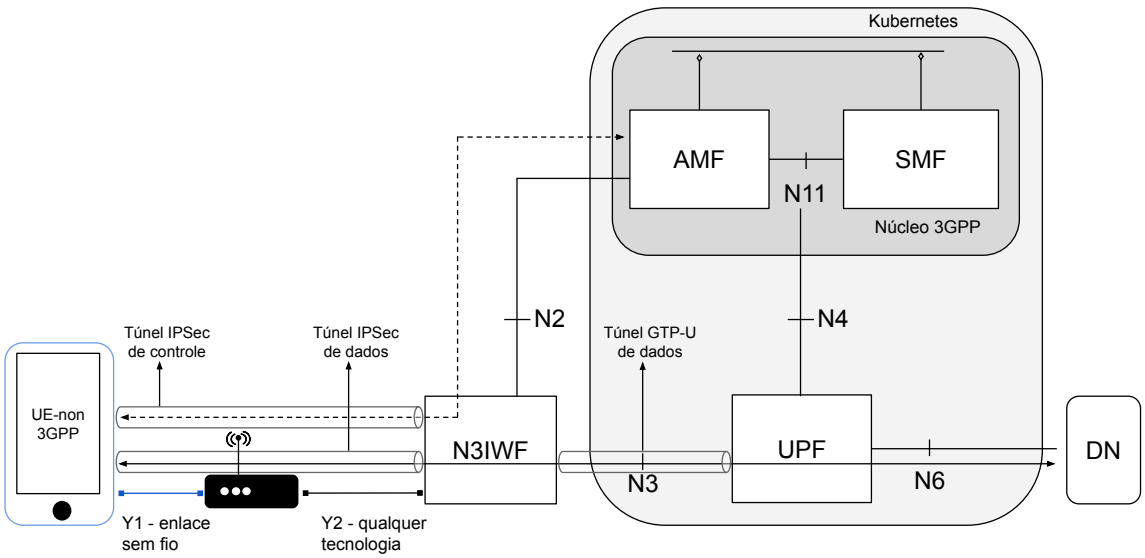


Figura 8: Acesso não-3GPP não-confiável com UE (lógico) em dispositivo extra e N3IWF implantada fora do Kubernetes.

Atualmente, estão sendo investigadas abordagens para solucionar o problema através de configurações avançadas do Kubernetes. No entanto, até o momento, não foi encontrada uma solução que possibilite a utilização da N3IWF dentro de uma *pod*. Temporariamente, para

contornar o problema, a N3IWF está fora Kubernetes, com uma topologia similar à ilustrada na Figura 8.

Túnel N3 entre N3IWF e UPF

Com a remoção da N3IWF de dentro do Kubernetes, é possível construir os túneis para o tráfego de dados e de controle, entre UE-non3GPP e N3IWF. No entanto, existe uma etapa adicional envolvendo a construção do túnel N3, entre N3IWF e UPF. Esse túnel é o elo de ligação entre UE-non3GPP e a rede de dados externa (e.g., a Internet), e ocorre através da porta *Transmission Control Protocol* (TCP) 38412, de acordo com o padrão 3GPP. Embora a N3IWF tenha conhecimento acerca da porta de comunicação, em um estágio inicial ela não conhece a localização da UPF à qual deverá se conectar. O processo de *descoberta* da UPF, ocorre de forma indireta através da interface N2, por intermédio da AMF, SMF e demais funções do plano de controle. Ou seja, as demais funções do plano de controle, que estão em *pods* em uma rede interna do Kubernetes, devolvem à N3IWF o endereço IP da UPF na qual a N3IWF deverá se conectar. No entanto, como a UPF está dentro de uma *pod*, o endereço IP devolvido à N3IWF é o endereço IP da *pod*, ou seja, um endereço IP que somente é visível para as funções que estão conteinerizadas. Quando a N3IWF tenta estabelecer conexão, ocorre um estouro de temporização, pois o endereço IP da UPF não é visível à N3IWF que não está dentro da *pod* pelo motivo descrito na seção anterior. Para contornar o problema, todas as funções de rede do free5GC foram retiradas do Kubernetes, i.e., todo o free5GC foi implantado de forma direta no sistema operacional. Apesar das funcionalidades e desempenho do free5GC não serem afetadas, essa decisão é considerada temporária e tem objetivo possibilitar o início de testes envolvendo algumas aplicações previstas no escopo do projeto.

As dificuldades relatadas nas seções anteriores demonstram que, mesmo diante da abordagem inovadora por parte dos órgãos de padronização, com especificações detalhadas e direcionadas para a *softwarização* das redes de comunicação, na prática, alguns elementos das redes de telecomunicações demonstram não estarem completamente preparados para esse novo cenário. As dificuldades descritas nas seções anteriores estão devidamente registradas e estão sendo endereçadas. Integrantes do projeto estão atuando na identificação de abordagens que permitam reintegrar os recursos do Kubernetes, gerando um cenário de Proto 6G com uma arquitetura de microsserviços que ofereça uma abordagem dinâmica e eficiente para a escalabilidade. Uma opção seria o 3GPP descrever suas funções de forma compatível com o Kubernetes ou outro ambiente similar, de forma a evitar tais dificuldades.

5 Ação 4 - Suporte à Plataforma da META 2

Em 2022, foram realizadas diversas ações para especificar, integrar e configurar a plataforma de experimentação do projeto Brasil 6G FASE 2 na rede FIBRE da RNP. Uma parte importante dos trabalhos foi a preparação da ilha Inatel nessa rede. Entretanto, os trabalhos de 2023 não ficaram restritos à rede FIBRE, pois devido à questões de interoperabilidade e atualização de software, outros ambientes de experimentação foram preparados e utilizados em paralelo. Isso se fez necessário para a realização de testes unitários no Inatel e outras instituições parceiras.

No Inatel, ambientes de teste fora da rede FIBRE RNP têm sido utilizados no CRR e no ICT Lab. Alguns ambientes implementaram seus próprios *clusters* Kubernetes em versões adequadas para o que se queria experimentar. Outros não utilizaram o Kubernetes. Diversos motivos levaram a diversidade de ambientes de experimentação e avaliação. Eles são apresentados individualmente em cada seção desse relatório, pois variam de ação para ação da META 5. Apesar dessa diversidade de ambientes, tanto no Inatel, quanto em outros parceiros, o objetivo foi sempre trazer os avanços obtidos para a rede FIBRE. Isso foi feito ao longo de 2023 com certo sucesso. Entretanto, diversos esforços ainda estão em andamento, e uma vez concluídos, serão incorporados as ilhas FIBRE da RNP.

5.1 Situação da Ilha FIBRE no Inatel

A Figura 9 ilustra a atual situação da ilha FIBRE no Inatel, indicando os sistemas integrados em vários laboratórios (CRR, ICT Lab, WOCA e IoTRG).

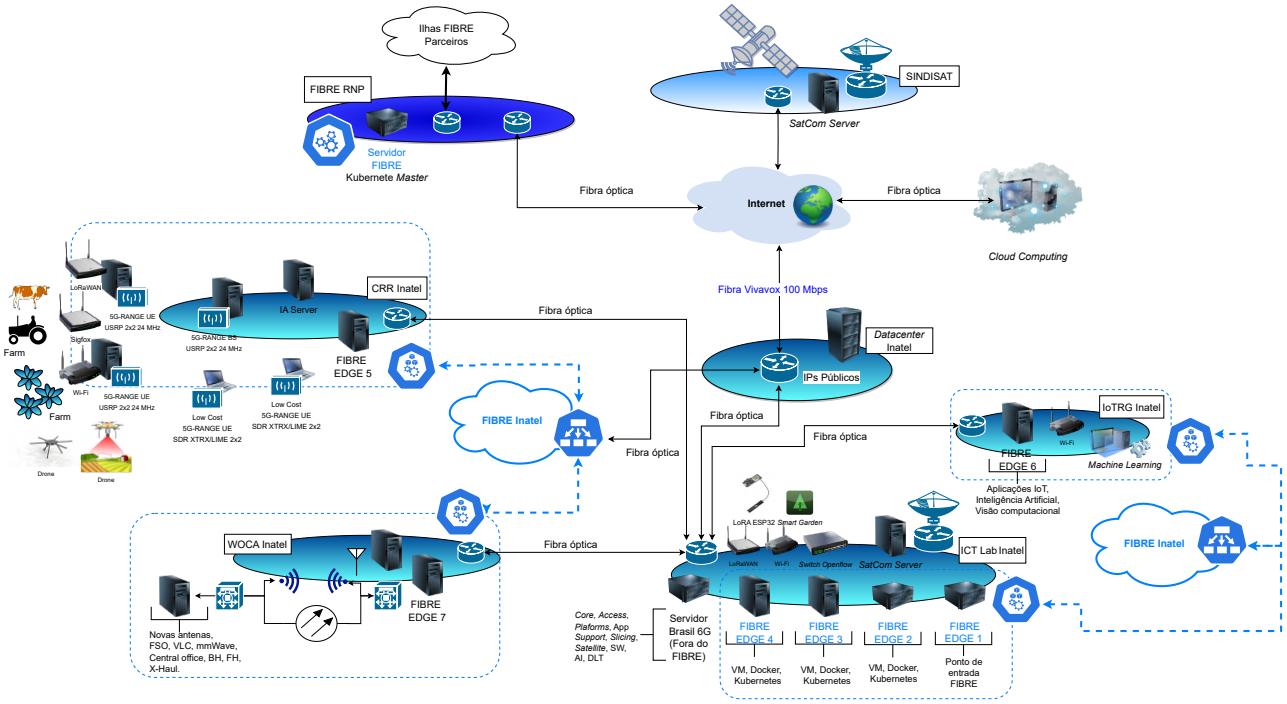


Figura 9: Cenário de conectividade integrando os componentes da ilha da rede FIBRE RNP do Inatel.

Após a realização do trabalho de integração interna dos laboratórios do Inatel foi possível realizar testes de conectividade entre todos os componentes da ilha FIBRE. A Figura 9 ilustra o cenário final com todos os componentes, desenvolvido para a Plataforma da META 2 do Projeto

Brasil 6G - Fase 2. O laboratório ICT Lab gerencia os servidores *edge* 1-4 da rede FIBRE. O servidor *edge* FIBRE 1 é ponto de entrada da ilha no Inatel e é constantemente monitorado pela RNP. Dessa forma esse servidor ficou definido como gerenciados dos outros *nodes* da ilha e não receberá aplicações do projeto. Os servidores *edge* FIBRE 2, 3 e 4 estão disponíveis para suportar aplicações SDN/NFV, enlaces conteinerizados e emulados do transceptor do Inatel ou demais aplicações que possam vir a ser exploradas. Como definido anteriormente, o servidor Brasil 6G irá hospedar a maioria dos *softwares* da META 5, incluindo o núcleo Proto 6G, acesso, plataformas, suporte a aplicações da META 6, fatiamento fim a fim, *software* dos cenários com satélite, Inteligencia Artificial (IA) e *Distributed Ledger Technologies* (DLT). Este servidor no momento atual está fora da ilha FIBRE, sendo cascateado no servidor *edge* FIBRE 1. O servidor Brasil 6G pode ser inserido como um ponto de conexão da rede FIBRE caso seja necessário no futuro para expensão dos cenários de experimentação. No ICT Lab também está instalado o terminal de satélite em parceira com o Projeto SINDISAT C, em execução com o Inatel. O objetivo é atuar nos cenários de *fronthaul* e *backhaul* possíveis de serem construídos com o transceptor *Generalized Frequency Division Multiplexing* (GFDM) e satélite SGDC.

O servidor *edge* FIBRE 5 está instalado no CRR conforme mostrado na Figura 9. O CRR também possui disponível transceptores 6G com *Software Defined Radio* (SDR) USRP MIMO 2x2 com largura de banda de 24 MHz. Dois drones também já encontram disponíveis. Em termos de conectividade *Non-3GPP Access* (N3A) tem-se LoRaWAN, Wi-Fi e Sigfox (que demanda ativação de licença). A introdução de outro SDR, o XTRX/LIME MIMO 2x2 é prevista a acontecer.

O laboratório IoTRG possui participação focada nas aplicações da META 6 e para isso hospeda o servidor *edge* FIBRE 6 da Figura 9. Já o laboratório WOCA gerencia o servidor edge FIBRE 7 e possui diversos cenários *Free-space Optical Communication* (FSO), *Visible Light Communications* (VLC), ondas milimétricas e *central office* desenvolvidos com o estado da arte para 6G. Esses cenários sem somam aos demais e poderão ser combinados com enlaces de fibra óptica para emular redes metropolitanas, incluindo novos tipos de antenas e sistemas rádio sobre fibra.

Ainda na Figura 9 o *Datacenter Inatel* oferece acesso do ICT Lab a Internet, bem como aos IPs públicos usados na ilha do FIBRE para acesso externo pela RNP. O componente FIBRE RNP interconecta as ilhas das instituições UFG, Unisinos e Inatel, dentre outras participantes do projeto Brasil 6G. O SINDISAT é parceiro do Inatel no Projeto C e oferece enlace de satélite através de um *gateway* (GW) até o satélite GEO SGDC. Através da Internet, o ICT Lab poderá configurar componentes, experimentos, etc. no *SatCom Server* satelital do(s) parceiro(s) SINDISAT. O enlace entrou em instalação em outubro de 2022 e ficou disponível em março de 2023 até a presente data.

6 Ações 5 e 10 - Suporte às Aplicações da META 6, e Integração Completa em Laboratório

Na Seção 6, são relatados os trabalhos das Ações 5 e 10 para o desenvolvimento de uma infraestrutura de integração completa em laboratório para o suporte às aplicações da META 6 do projeto Brasil 6G Fase 2 junto ao núcleo e à rede de acesso Proto 6G. No ano de 2023, foram feitas inúmeras ações para a integração completa dos esforços das diversas ações em laboratório. Até o momento de entrega desse relatório os seguintes componentes da arquitetura completa foram integrados e estão executando com sucesso no Inatel:

- Núcleo - Conforme Seção 3, um núcleo funcional Free5GC foi instalado tanto fora da rede FIBRE RNP, quanto internamente a nós da ilha do Inatel nessa rede. Instalações externas ao FIBRE foram feitas no CRR e no ICT Lab do Inatel. A instalação no FIBRE foi feita exclusivamente no ICT Lab.

- Acesso - Conforme Seção 4, uma solução de acesso não-3GPP untrusted foi implementada a partir do desenvolvimento da função N3IWF. Essa solução se encontra funcional. Entretanto, a função N3IWF possui algumas características especificadas pelo 3GPP que não são compatíveis com o ambiente Kubernetes. Portanto, tal função executa em máquina física a parte e fora da rede FIBRE RNP. Os trabalhos continuam para encontrar uma forma de resolver esse problema complexo de configuração de sistemas. A instalação no CRR utiliza transceptores BS-T6G e UE-T6G físicos, enquanto que as instalações no ICT Lab usam uma versão virtual da MAC desses transceptores BS-T6G e UE-T6G.

- Integração ao FIBRE RNP - As atividades de integração a rede FIBRE RNP são descritas na Seção 5. A falta de compatibilidade entre versões do Kubernetes e demais componentes do Proto 6G (incluindo o Free5GC e a função N3IWF) dificultou o andamento das ações junto a rede FIBRE RNP. Soma-se a isso a atualizações frequentes existentes nessa rede. Espera-se resolver essas dificuldades em eventual continuação desse trabalho, permitindo assim ofertar no FIBRE todos os avanços realizados no Brasil 6G.

- Fatiamento Fim a Fim - Um cenário de fatiamento fim a fim foi desenvolvido para teste no Kubernetes fora da rede FIBRE no ICT Lab do Inatel em 2022. A equipe da Ação 6 apresenta na Seção 7 os avanços de 2023, que incluem um cenário ainda mais completo de fatiamento como um serviço. A integração desse cenário com aquele que possui a função N3IWF se mostrou extramente complexa e não haverá tempo hábil na FASE 2 do projeto para tal.

- Rede Satelital - De acordo com a Seção 8, o setup do Free5GC fora do FIBRE foi testado com um enlace de satélite real da empresa Viasat em parceria com o SINDSAT/ABRASAT.

- Inteligência Artificial - O cenário que especifica a função NWDAF utiliza o núcleo Free5GC. Até o momento da escrita desse relatório a implementação da função NWDAF ainda não realizada, o que leva essa contribuição a ser integrada em uma eventual continuação do projeto Brasil 6G.

- Distributed Ledger Technologies - As ações relativas a blockchain foram descontinuadas em 2023 devido a falta de participação do pesquisador parceiro. Em resumo, realizamos

em 2023 integrações relevantes tanto no CRR do Inatel, quanto no ICT Lab (dentro e fora do FIBRE RNP).

- Aplicações - Dadas as aplicações previstas em 2022 para a FASE 2 do Brasil 6G, foram integradas novas aplicações e feito um novo remanejamento de ordem e conexão com as UEs, cujas principais informações estão sumarizadas na Figura 2 e discutidas e forma detalhada e minuciosa nos Relatórios 6.1 e 6.2 do Projeto Brasil 6G, referentes à META 6, responsável pelas demandas de aplicações. Assim sendo, esta seção compreende as Ações 5 e 10 de forma a propôr a integração em laboratório de uma infraestrutura de rede capaz de suportar as aplicações previstas no Projeto Brasil 6G. A infraestrutura desenvolvida em laboratório para atender à essas aplicações será descrita com mais detalhes à seguir.

Para atender às demandas das aplicações previstas na META 6 do Projeto Brasil 6G, é necessária a elaboração de um ambiente capaz de possibilitar a realização de testes iniciais de comunicação e conectividade entre uma UE e o núcleo, considerando uma rede de acesso *Untrusted Non-3GPP*, que por sua vez, é composta pelos transceptores do Projeto Brasil 6G. A partir destes testes, é possível analisar algumas características do enlace de comunicação, como por exemplo, vazão e latência, e finalmente especificar os requisitos mínimos das aplicações a serem executadas na rede. Desta forma, a Figura 10 ilustra o ambiente de testes iniciais de integração entre UE, rede de acesso e núcleo da rede Proto 6G desenvolvido no laboratório *Centro de Referência em Radiocomunicações* (CRR), considerando um cenário real com o uso dos transceptores UE-T6G e BS-T6G.

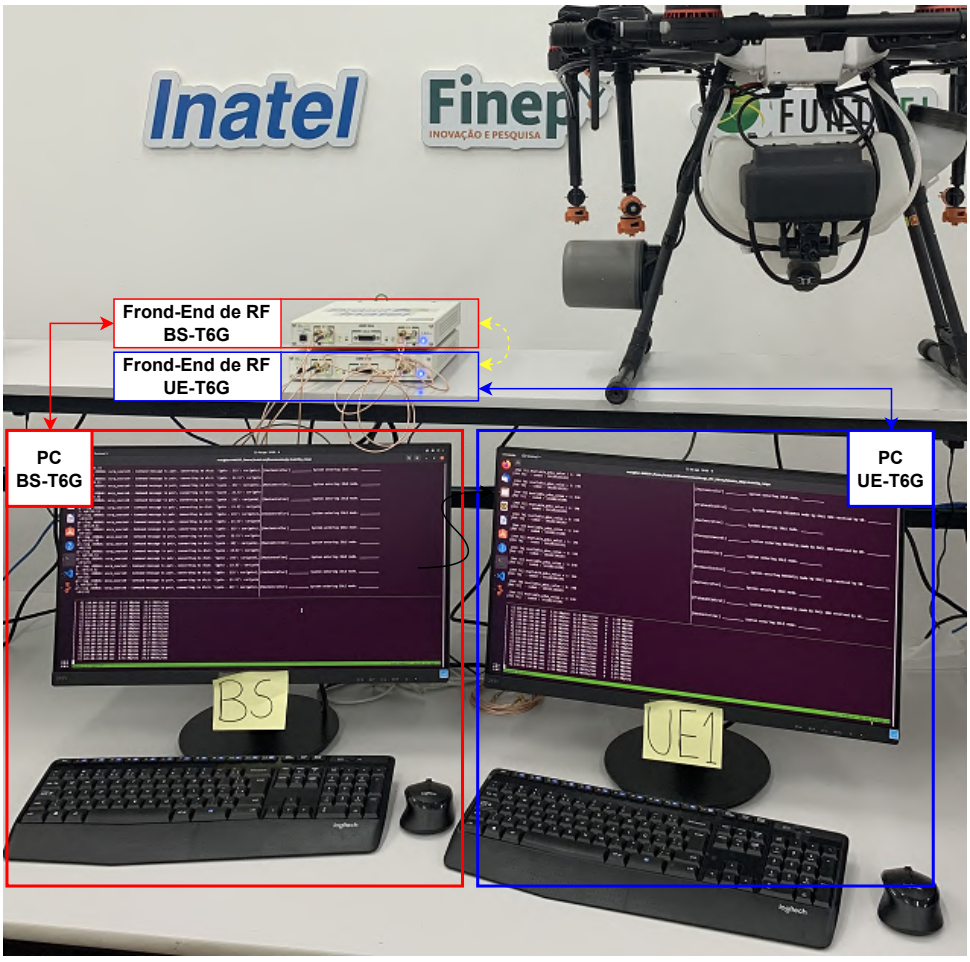


Figura 10: Configuração do ambiente de testes iniciais de integração entre UE, rede de acesso e núcleo da rede Proto 6G considerando cenário real com uso dos transceptores UE-T6G e BS-T6G.

Conforme ilustrado na Figura 10, o ambiente de testes inicial consiste de uma rede de acesso *Untrusted Non-3GPP* formada pelos transceptores BS-T6G e UE-T6G. Cada transceptor consiste de um *front-end* de *Radio Frequency* (RF) integrado à um computador. Um dos transceptores foi devidamente configurado para operar como uma *Base Station* (BS), enquanto o outro como um terminal UE da rede de acesso Brasil 6G. Os *front-ends* de RF são SDRs compostos por conversores *Analog-to-Digital* (AD) e *Digital-to-Analog* (DA), além de dispositivos de RF para conversão do sinal de banda-base para canal e vice-versa. Esses SDRs podem ser programados através de seus respectivos computadores (PC BS-T6G e PC UE-T6G), conforme ilustrado na Figura 10, cuja função principal é executar os códigos das camadas *Physical Layer* (PHY) e *Media Access Control* (MAC), desenvolvidos neste projeto em linguagem C++. A camada PHY é responsável pela realização das etapas de processamento de sinais e implementada via *software* GNU Radio para programar os SDRs, enquanto a camada MAC está relacionada ao controle de acesso ao meio físico de comunicação, gerenciando como os dispositivos compartilham e transmitem dados na rede.

É válido ressaltar que este experimento foi feito inicialmente com uma única UE conectada à rede, e a comunicação entre os SDRs foi realizada de forma cabeada à uma curta distância, considerando apenas a *Indoor Unit* (IDU) dos transceptores, além de minimizar eventuais

limitações relacionadas à *Signal-to-Noise Ratio* (SNR) e outras características do canal. Desta forma, o objetivo deste teste é exclusivamente validar a comunicação entre um equipamento de usuário e o núcleo através de uma rede de acesso *Untrusted Non-3GPP* usando os transceptores do Projeto Brasil 6G. Finalmente, os resultados esperados são: o registro e autenticação de uma UE não-3GPP ao núcleo; e possibilitar que o núcleo seja capaz de fornecer à UE conectividade à Internet ou à uma rede privada qualquer, de acordo com o endereço cadastrado à UPF direcionada especificamente para a mesma.

As funções (também conhecidas como componentes) do núcleo da rede Proto 6G, detalhadas na Seção 4, foram divididas para serem executadas em dois equipamentos distintos. Assim, o componente N3IWF foi integrado ao PC BS-T6G, ilustrado na Figura 10. Os demais componentes do núcleo, por sua vez, foram executados em um computador à parte, conectado ao PC BS-T6G via cabo Ethernet. A Figura 11 e a Tabela 1 detalham, respectivamente, a ilustração do diagrama geral da rede e a descrição dos modelos dos equipamentos usados durante os testes.

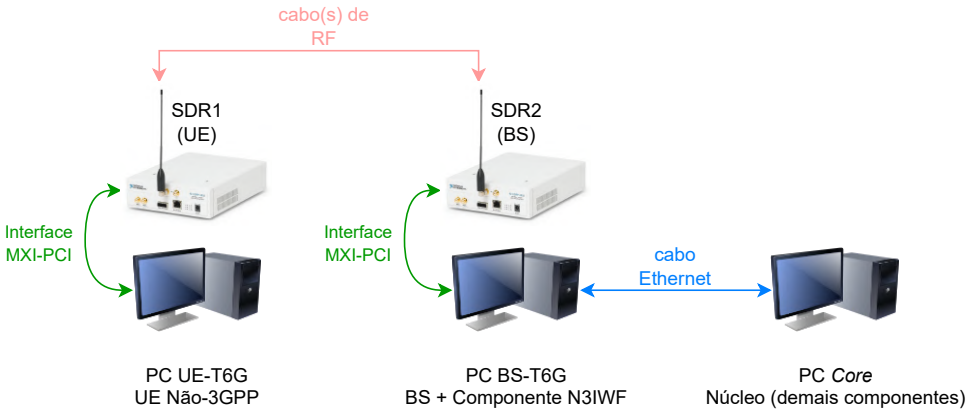


Figura 11: Diagrama geral dos equipamentos usados nos testes e demostração.

Tabela 1: Descrição dos equipamentos usados para os testes iniciais de integração entre UE, BS e núcleo, considerando a implementação dos *front-ends* de RF.

Equip.	**Descrição**	**Modelo**	**Características de Hardware**	**Características de Software**
PC UE-T6G	Computador para UE Não-3GPP.	Personalizado.	- Processador Intel(R) Core(TM) i9-13900K; - Memória RAM de 32GB; - HD SSD de 1TB; - Placa de Vídeo GF 12GB RTX 3060.	- OS Ubuntu 20.04 LTS x64; - kernel 5.15.0-79-generic; - GNU Radio 3.9; - go version 1.14.4; - software para UE não-3GPP.
PC BS-T6G	Computador para BS e N3IWF.	Personalizado.	- Processador Intel(R) Core(TM) i9-13900K; - Memória RAM de 32GB; - HD SSD de 1TB; - Placa de Vídeo GF 12GB RTX 3060.	- OS Ubuntu 20.04 LTS x64; - kernel 5.15.0-79-generic; - GNU Radio 3.9; - go version 1.14.4; - componente N3IWF do núcleo free5gc instalado.
PC *Core*	Computador para demais componentes do núcleo 3GPP.	Dell XPS 8930.	- Processador Intel(R) Core(TM) i7-8700; - Memória RAM de 16GB; - HD de 1TB; - Placa de Vídeo Intel(R) UHD Graphics 630.	- OS Ubuntu 20.04 LTS x64; - kernel 5.15.0-79-generic; - GNU Radio 3.9; - go version 1.14.4; - free5gc instalado (funções: UDR, UDM, UPF, SMF, PCF, NSSF, NRF, AUSF, e AMF).
SDR1	SDR do transceptor da UE.	Ettus USRP X310.	- Xilinx Kintex-7 XC7K410T FPGA; - 14 bit 200 MS/s ADC; - 16 bit 800 MS/s DAC; - Faixa de frequência: DC - 6 GHz com *daughterboard* adequado; - Até 160MHz de *bandwidth* por canal; - 2 *wide-bandwidth* RF *daughterboard slots*; - GPSDO opcional; - Múltiplas interfaces de alta velocidade (Dual 10G, PCIe Express, ExpressCard, Dual 1G).	- Compatível com arquitetura USRP Hardware Driver (UHD); - UHD fornece API C++ API para controle e *stream* da USRP; - Compatível com *frameworks* de *software*, incluindo: GNU Radio, LabVIEW, e Matlab; - Imagem FPGA fornecida com UHD personalizável para integrar seu próprio sinal de processamento.
SDR2	SDR do transceptor da BS.	Ettus USRP X310.	- Xilinx Kintex-7 XC7K410T FPGA; - 14 bit 200 MS/s ADC; - 16 bit 800 MS/s DAC; - Faixa de frequência: DC - 6 GHz com *daughterboard* adequado; - Até 160MHz de *bandwidth* por canal; - 2 *wide-bandwidth* RF *daughterboard slots*; - GPSDO opcional; - Múltiplas interfaces de alta velocidade (Dual 10G, PCIe Express, ExpressCard, Dual 1G).	- Compatível com arquitetura USRP Hardware Driver (UHD); - UHD fornece API C++ API para controle e *stream* da USRP; - Compatível com *frameworks* de *software*, incluindo: GNU Radio, LabVIEW, e Matlab; - Imagem FPGA fornecida com UHD personalizável para integrar seu próprio sinal de processamento.

Após a configuração dos equipamentos mapeados na Figura 11 e na Tabela 1, o primeiro passo é executar o *script* do núcleo, disponível no computador "PC *Core*", que consiste em inicializar seus respectivos componentes: UDR, UDM, UPF, SMF, PCF, NSSF, NRF, AUSF, e AMF. A Figura 12 ilustra a execução do *script* do núcleo e consequentemente algumas de suas funções sendo inicializadas.

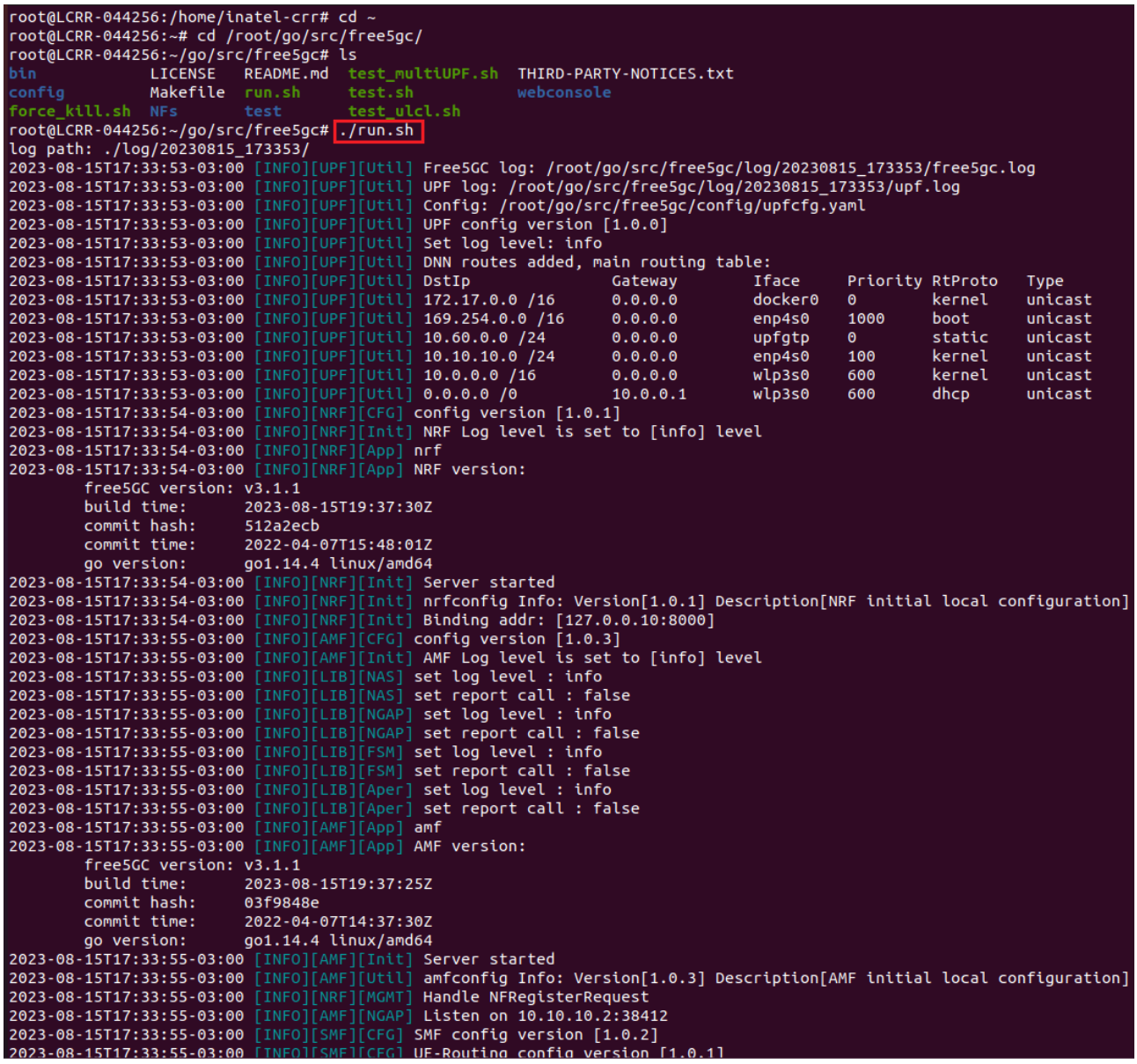


Figura 12: Execução do *script* para inicialização das funções do núcleo.

Além disso, ainda no "PC *Core*", é necessário executar o *script* de inicialização do servidor de banco de dados responsável pelo registro das credenciais de acesso de uma UE Não-3GPP. A Figura 13 mostra a inicialização do servidor de base de dados do free5GC.

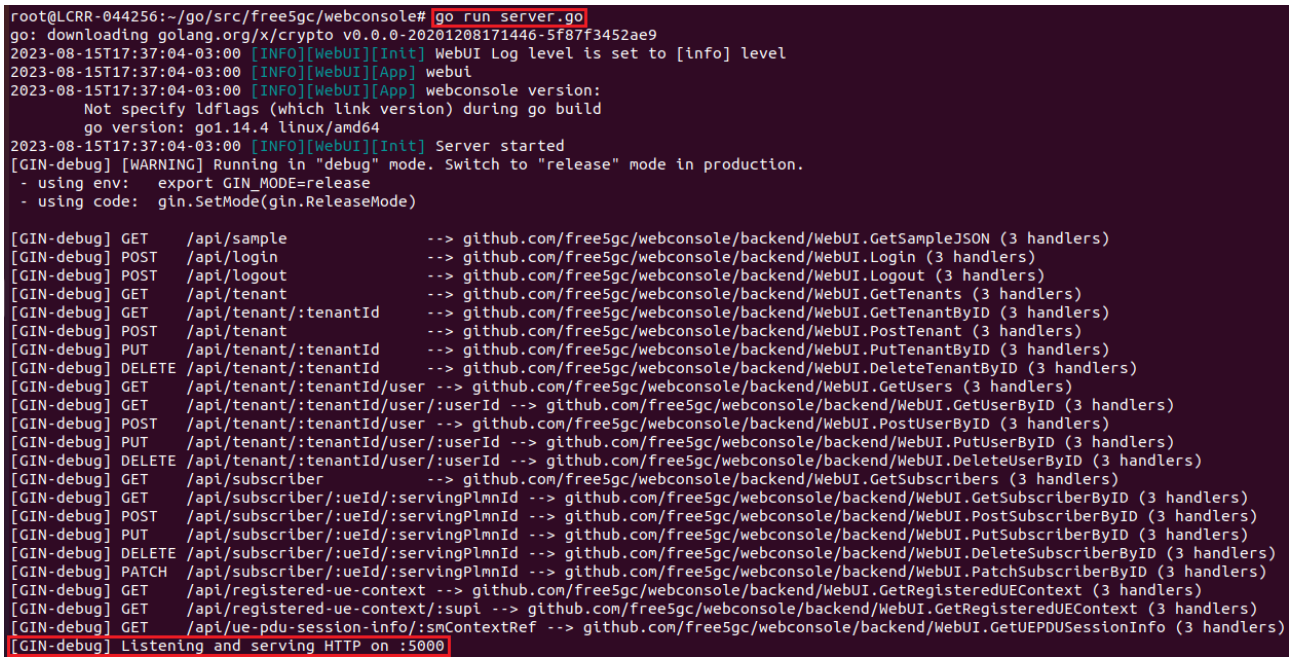


Figura 13: Execução do *script* para inicialização do servidor de banco de dados para registro das credenciais de acesso da UE Não-3GPP.

Conforme destacado na Figura 13, após a inicialização do servidor, o mesmo apresenta a seguinte mensagem: "*listening* and *serving Hypertext Transfer Protocol* (HTTP) on :5000". Em resumo, esta mensagem significa que o servidor está ativo e pronto para receber requisições HTTP na porta 5000. Isso é fundamental para o funcionamento da aplicação *WebUI* que gerencia o registro de credenciais de acesso de uma UE no Proto 6G. O servidor irá aguardar o recebimento de uma mensagem de requisição de uma UE para essa porta, de modo a tratá-la e executar as operações necessárias para registrar e gerenciar as credenciais de acesso dessa UE no seu banco de dados.

Já no computador responsável por executar os códigos das camadas PHY e MAC da BS (PC BS-T6G), conforme detalhado na Figura 11, também é executada uma função específica do núcleo, denominada N3IWF (veja a Seção 4). Desta forma, a BS atua como um ponto de interconexão entre o núcleo e a rede *Untrusted Non-3GPP*, além de realizar o gerenciamento de mobilidade, tráfego e *Quality of Service* (QoS). Esta função também possui como principal função a segurança e autenticação dos dispositivos e conexões que entram na rede Proto 6G por meio de uma rede não-3GPP. Dado que os dados provenientes da UE Não-3GPP são recebidos pela BS através do enlace de comunicação dos transceptores da rede de acesso *Untrusted Non-3GPP*, este componente é crucial para manter a integridade da rede e das informações transmitidas. A Figura 14 ilustra a execução do *script* de inicialização do componente N3IWF no computador "PC BS-T6G".

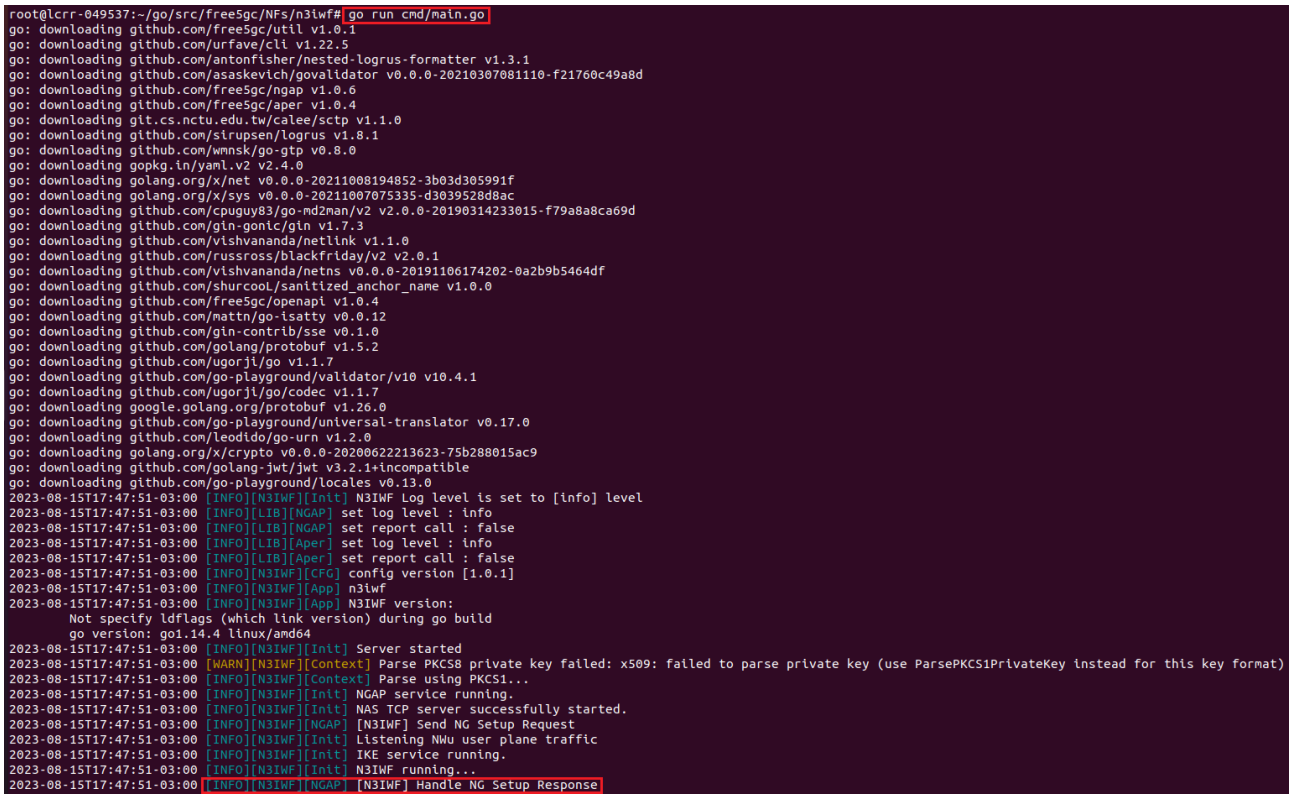


Figura 14: Execução do *script* para inicialização do componente N3IWF no computador "PC BS-T6G".

Por fim, no computador cuja função principal é executar os códigos das camadas PHY e MAC da UE (PC UE-T6G), como ilustrado na Figura 11, também é necessária a execução de um *script* responsável por inicializar o processo de configuração da UE Não-3GPP, permitindo assim a autenticação e registro da UE no núcleo da rede. Tal ação consequentemente irá prover acesso à uma UPF, que para este teste em específico possa enviar dados à Internet. A Figura 15 apresenta o processo de inicialização do *script* de configuração da UE para se comunicar com o núcleo da rede.

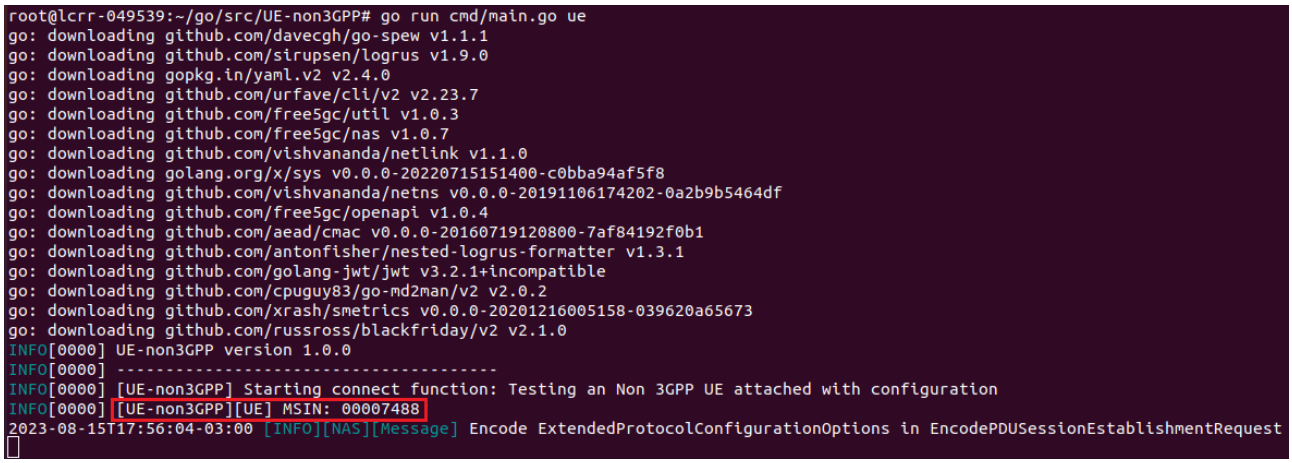


Figura 15: Inicialização do *script* de configuração da UE para comunicação com o núcleo, executado no computador "PC UE-T6G".

O fluxo básico para um usuário acessar serviços em uma rede de dados (por exemplo, a Internet) consiste na criação da conexão de sinalização com o plano de controle (N1), registro e estabelecimento de uma sessão de transporte de PDU. Como pode ser visto na Figura 15, ao iniciar o processo de requisição de registro e estabelecimento de sessões PDU pela UE Não-3GPP, alguns parâmetros devem ser previamente configurados para serem enviados. Essas configurações referem-se, por exemplo, a parâmetros da rede de acesso, parâmetros de segurança, código do *slice* desejado, e até mesmo o identificador de usuário, que neste exemplo trata-se do *Mobile Subscriber Identification Number* (MSIN) da UE configurado com o valor "00007488", também destacado na Figura 15.

Durante o registro, é realizado o estabelecimento da conexão de sinalização entre UE e rede de acesso via interface de referência N1 através de túneis IPSec para acessos não-3GPP, além da conexão entre rede de acesso e AMF via interface de referência N2, através da associação *NG Application Protocol* (NGAP) de UE. Além disso, são executados processos de criação de contexto de usuário, autenticação, e aplicação de políticas de UE e de gerenciamento de acesso.

No "PC *Core*", onde o núcleo é executado, é possível verificar no terminal do servidor para registro das credenciais de acesso da UE Não-3GPP, previamente inicializado conforme ilustra a Figura 13, que a UE foi registrada dentro do seu banco de dados. A Figura 16 apresenta o momento em que a UE é inscrita no banco de dados do servidor.

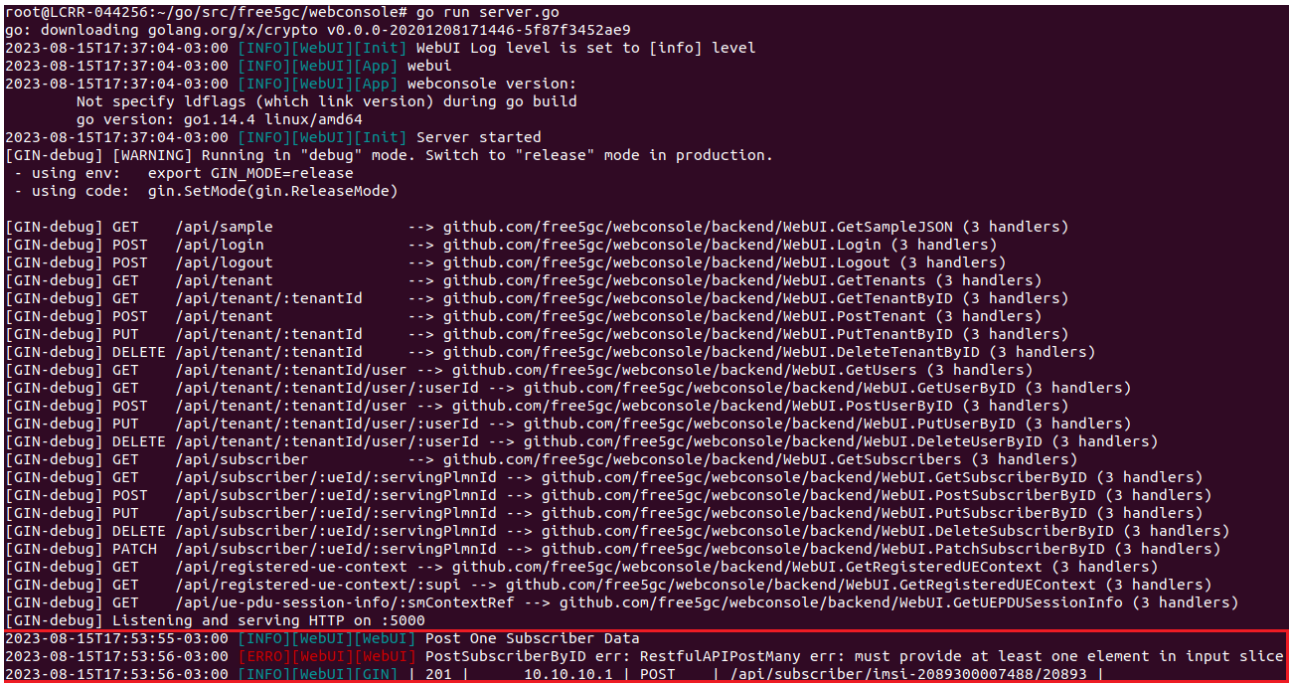


Figura 16: Registro da UE Não-3GPP no banco de dados do servidor de credenciais de acesso, executado no computador "PC *Core*".

Após o estabelecimento da conexão de sinalização de plano de controle N1 e o registro completo da UE, é realizado o procedimento de requisição de serviço para estabelecer a conexão de sinalização entre UE e AMF; e finalmente o estabelecimento de sessões de transporte de PDU para transmissão e recepção de dados através de uma UPF.

Uma sessão PDU representa uma conexão lógica entre a UE e a rede de dados, ou *Data Network* (DN). Para conectar com essa rede de dados, a UE deve primeiro submeter uma requisição de estabelecimento de sessão PDU para o núcleo. Durante o processo de estabelecimento da sessão, alguns parâmetros fundamentais são definidos para descrever as propriedades do PDU, como por exemplo: o identificador único da sessão PDU (*PDUSessionID*), o identificador de *slicing* (*Network Slice Selection Assistance Information* (S-NSSAI)), *Data Network Name* (DNN), entre outros. Visto que o plano de usuário representa a conexão da sessão PDU com a rede de dados, o mesmo é construído sob tunelamento dos dados pela rede de transporte. Desta forma, o objetivo é garantir criptografia fim-a-fim entre o UE e a DN. Assim, é criada uma interface de rede na UE utilizando o túnel *Generic Routing Encapsulation* (GRE) como técnica de encapsulamento de pacotes de rede para o tráfego de dados entre a UE Não-3GPP e a N3IWF. Posteriormente, esses dados são encaminhados para a UPF via túneis *GPRS Tunnelling Protocol for User Data* (GTP-U). A Figura 17 ilustra a execução do comando *ifconfig* no computador "PC UE-T6G", mostrando as interfaces de rede criadas para estabelecer os planos de controle e usuário.

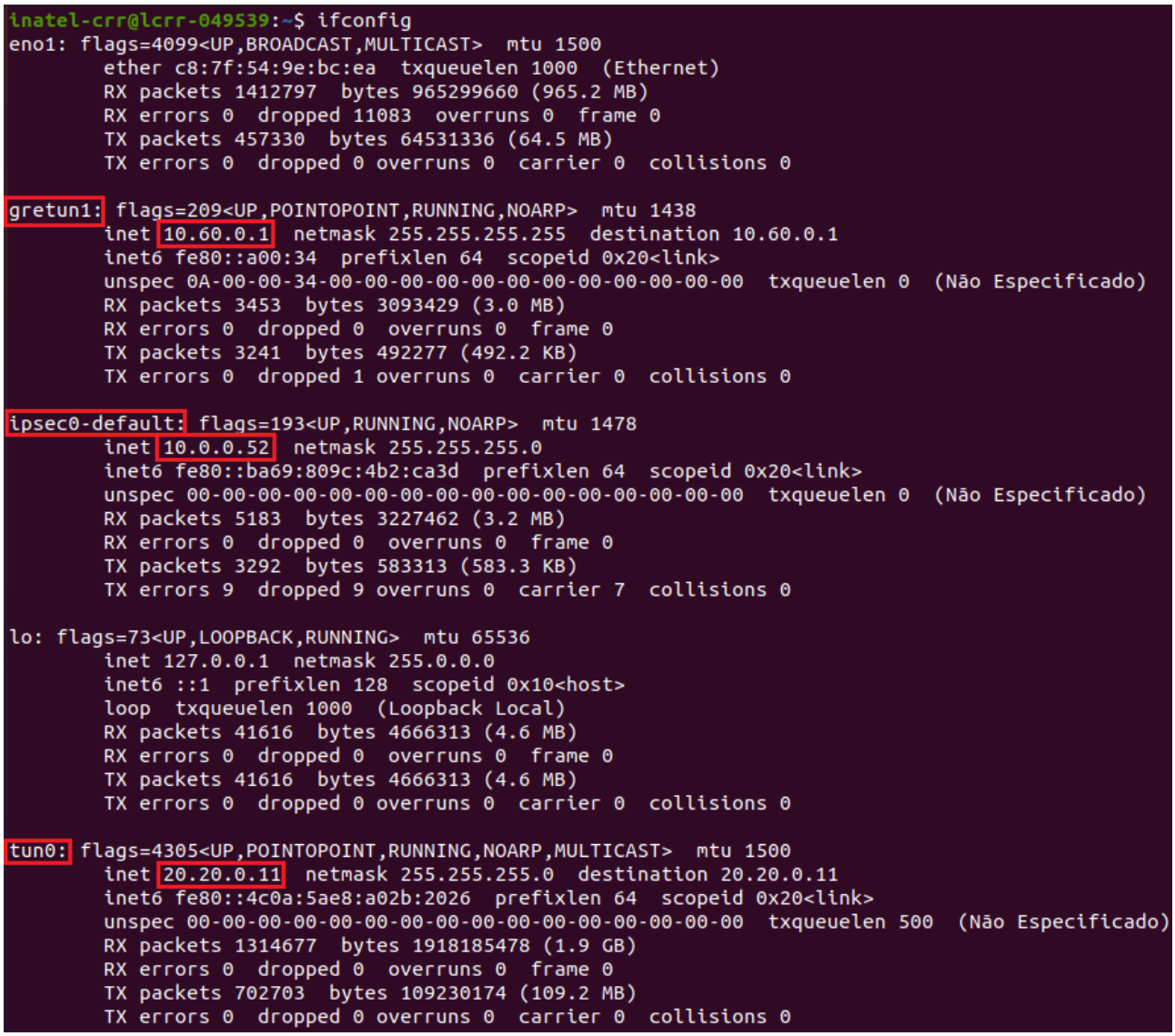


Figura 17: Comando *ifconfig* executado no computador "PC UE-T6G".

Conforme mostrado na Figura 17, três interfaces de rede podem ser destacadas: (i) *tun0*, *ipsec0-default* e *gretun*. O *tun0* é referente ao tunelamento responsável pelo tráfego de pacotes entre UE e BS usando os *front-ends* de RF. Esta interface de rede é usada para estabelecer o procedimento de *IKE*v2 *Security Association* (SA) com o componente N3IWF, de modo a criar a sinalização IPSec. A criação da sinalização IPSec é estabelecida através da interface *ipsec0-default* e usada para transferir tráfego de sinalização *Non-Access Stratum* (NAS) no plano de controle. Já a interface *gretun1* é criada no processo de estabelecimento de plano de usuário, e é por meio dela que serão trafegados pacotes de dados à N3IWF; e consequentemente para a UPF do núcleo da rede. A Figura 18 apresenta um *log* do núcleo onde parte da etapa de estabelecimento de sessão PDU está sendo executada, definindo alguns dos parâmetros fundamentais previamente mencionados (*PDUSessionID*, S-NSSAI, DNN), além da seleção da UPF para a UE registrada (IMSI com valor MSIN configurado para "00007488").

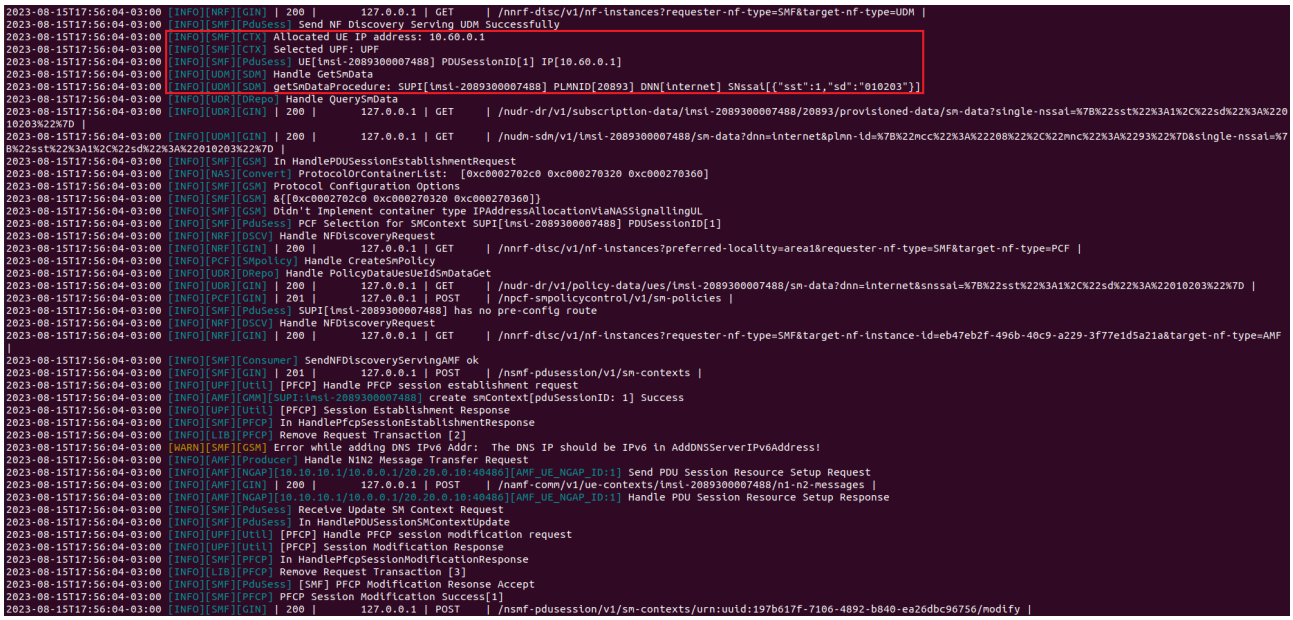
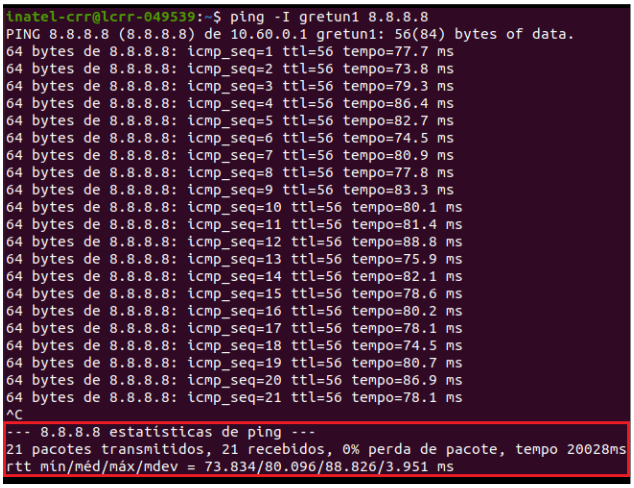


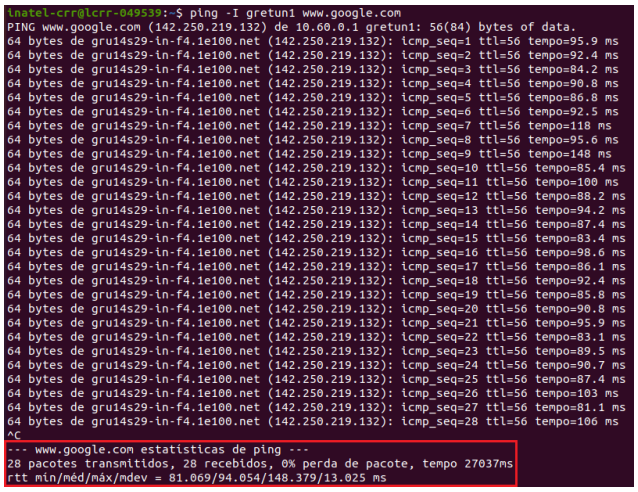
Figura 18: *Log* do núcleo da rede no "PC *Core*", com etapas de estabelecimento de sessão de transporte de PDU.

De acordo com a Figura 18, percebe-se que o estabelecimento da sessão de PDU no núcleo ("PC *Core*") utiliza o endereço IP 10.60.0.1 para a UPF selecionada, o que corresponde, consequentemente, ao endereço alocado para a interface de rede *gretun1* no "PC UE-T6G", destinada para o tráfego de dados do plano de usuário.

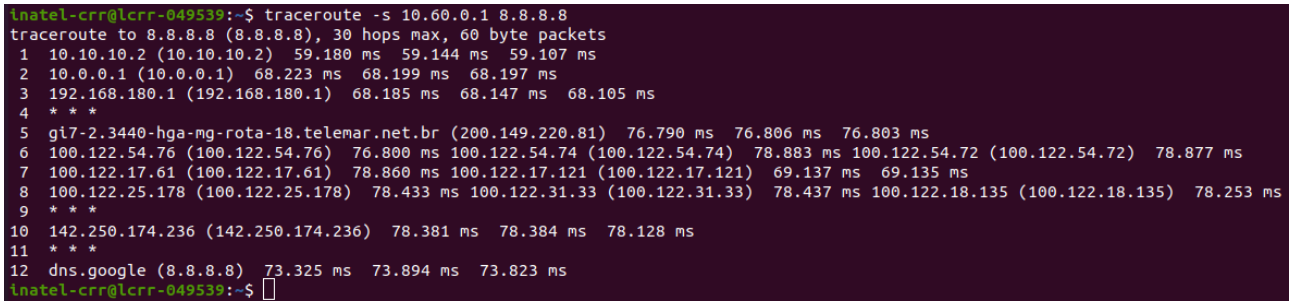
Uma vez estabelecidos os planos de controle e usuário, testes iniciais foram realizados para verificar se o fluxo de dados estava sendo trafegado pela interface de rede *gretun1* alocada com o endereço IP 10.60.0.1. Para validação, foram executados comandos de *ping* e *traceroute* no "PC UE-T6G"com o objetivo de analisar o tráfego e a rota de acesso à Internet. A Figura 19 ilustra os comandos de *ping* e *traceroute* realizados no cenário de testes proposto.



(a) Teste de *ping* com interface *gretun1* para endereço IP 8.8.8.8.



(b) Teste de *ping* com interface *gretun1* para *Domain Name System* (DNS) do *www.google.com*.



(c) Teste de *traceroute* com origem em 10.60.0.1 e destino no endereço IP 8.8.8.8.

Figura 19: Testes de *ping* e *traceroute* para análise do tráfego de dados entre UE e DN Internet.

Na Sub-Figura 19a é mostrado o resultado de um teste de *ping* ao destino 8.8.8.8, correspondente a um endereço de servidor DNS público mantido pela Google. Também foi especificado através do parâmetro *-I gretun1* a interface de rede que deve ser usada para enviar os pacotes, e os resultados mostraram que a UE Não-3GPP possui conectividade à Internet com uma latência média de 80,086 ms. Já na Sub-Figura 19b, foi executado o mesmo teste, porém usando no comando de *ping* o nome do endereço de destino (www.google.com). Os resultados obtidos mostraram que a resolução de nomes está respondendo corretamente com uma latência média de aproximadamente 94.054 ms. Por fim, através da Sub-Figura 19c, é possível observar com a ajuda do comando de *traceroute* o rastreamento da rota que os pacotes IP seguem de um ponto de origem para o destino na rede. Para este comando em específico, foi definido o endereço da interface de rede *gretun1* como origem (10.60.0.1), enquanto que o endereço de destino adotado foi 8.8.8.8. Percebe-se que o primeiro salto (*hop*) refere-se ao endereço IP 10.10.10.2, presente no computador "PC *Core*", que por sua vez possui uma interface de rede que provê acesso à Internet. Para este teste, a interface de rede com conectividade à Internet está atrelada à um adaptador de rede Wi-Fi cujo *gateway* padrão é definido como 10.0.0.1, sendo este endereço IP o segundo salto mapeado pelo comando *traceroute*. Ao final do rastreamento, pode-se analisar que após 12 *hops* o pacote chegou ao destino esperado. Para facilitar o entendimento das configurações usadas no teste, a Figura 20 apresenta as principais definições de endereços de rede usadas para a avaliação de desempenho da rede com relação ao acesso da UE Não-3GPP para

a rede de dados da Internet.

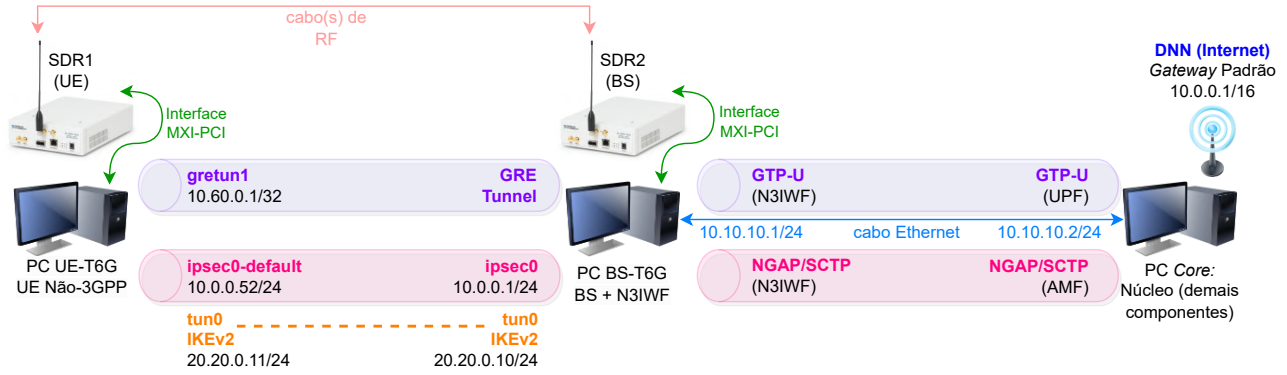


Figura 20: Diagrama geral da rede com principais endereços IP definidos para a execução dos testes.

Por fim, a Figura 21 ilustra os resultados obtidos com testes via *Web*. Nota-se que o usuário consegue assistir à vídeos no *site* do YouTube, mostrando novamente a conectividade à Internet da UE Não-3GPP fornecida pelo núcleo da rede. Testes de *throughput* também foram realizados com o serviço *Speedtest* para medir a velocidade da conexão de Internet, e resultados mostraram uma vazão de *Uplink* aproximada de 19,81 Mbps, o que corresponde à uma taxa próxima do valor esperado, dadas as configurações de alocação de banda da UE previamente estabelecidas nos SDRs.

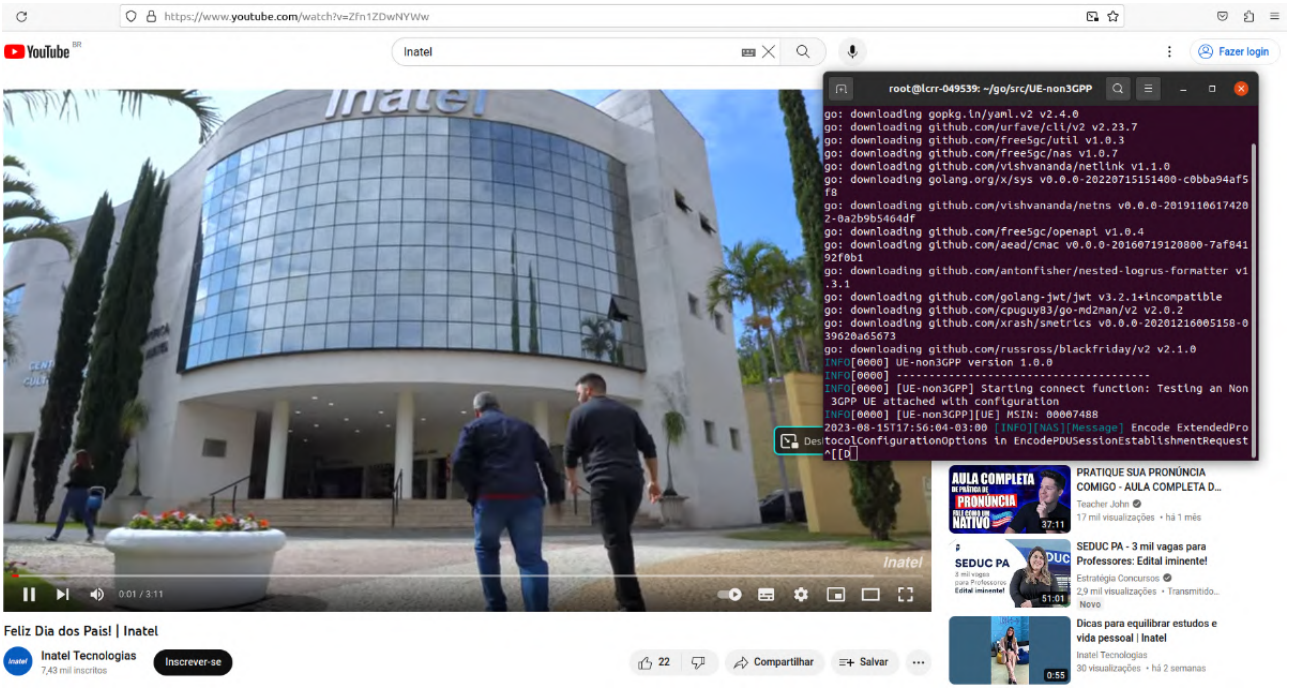


Figura 21: Teste de conectividade à Internet via acesso *web* ao YouTube para visualização de vídeos *online*.

7 Ação 6 - Fatiamento Fim a Fim

A Seção 7 apresenta uma descrição para a execução de um fatiamento fim-a-fim de recursos físicos e cibernéticos em 2023. Inicialmente, a Ação 6 descreve as etapas de instalação da infraestrutura e os elementos necessários para compor a plataforma de fatiamento de rede 6G como serviço, chamada NASP - *Network Slice as a Service Plataform*. Posteriormente, as funcionalidades da NSAP são apresentadas e analisadas. É importante destacar que a instalação e execução da NSAP é realizada em um **ambiente de computação em nuvem**. Finalmente, uma bateria de experimentos é apresentados para validar e analisar os resultados obtidos com a NASP.

7.1 Pré-requisitos

Os pré-requisitos de recursos computacionais virtuais e sistema operacional necessários para a realização da Ação 6 são listados abaixo.

- Sistema Operacional: Ubuntu 22.04 (LTS) x64. Por exemplo, kernel 5.4.0-122-genérico.
- Memória: no mínimo 8 GB de RAM.
- Disco: 80 GB.

Os pré-requisitos de software necessários para a configuração do ambiente de computação em nuvem é discutido na seguinte subseção.

7.2 Instalação dos Componentes de Fatia de Rede 6G

A instalação e execução da NASP será baseada na topologia de rede representada na Figura 22, i.e., uma rede não 3GPP se conectando em um núcleo SBA 3GPP. Neste contexto, para o correto funcionamento da NASP, é necessária uma infraestrutura Kubernetes, Núcleo SBA com a função N3IWF e um testador com suporte a UEs que visam validar todas as funcionalidades da plataforma. O processo de instalação envolve quatro etapas principais com pelo menos duas máquinas Linux. Além disso, a ilustração apresenta os túneis necessários para o estabelecimento do plano de controle e plano de dados, utilizando o protocolo IPSec entre UE e N3IWF e túnel GTP entre UE e UPF. Para cada plano de controle é estabelecido uma rede IP.

A etapa de instalação de cada componente é descrita a seguir. É importante destacar que todas as etapas de instalação descritas devem ser executadas com privilégios de *root* (*SUDO SU*). Além disso, todos *scripts* e códigos-fontes utilizados para o desenvolvimento da NASP estão disponíveis no GitHub[7].

Infraestrutura Kubernetes

Cada máquina deve ter instalado o sistema operacional Linux com Ubuntu 22.04. Além disso, as máquinas precisam ter os recursos computacionais especificados, com ambiente para instalação das ferramentas make, gcc e curl.

```
sudo apt install -y dwarves make gcc curl
```

[7]https://github.com/fhgrings/6G_Network_Slice_Install

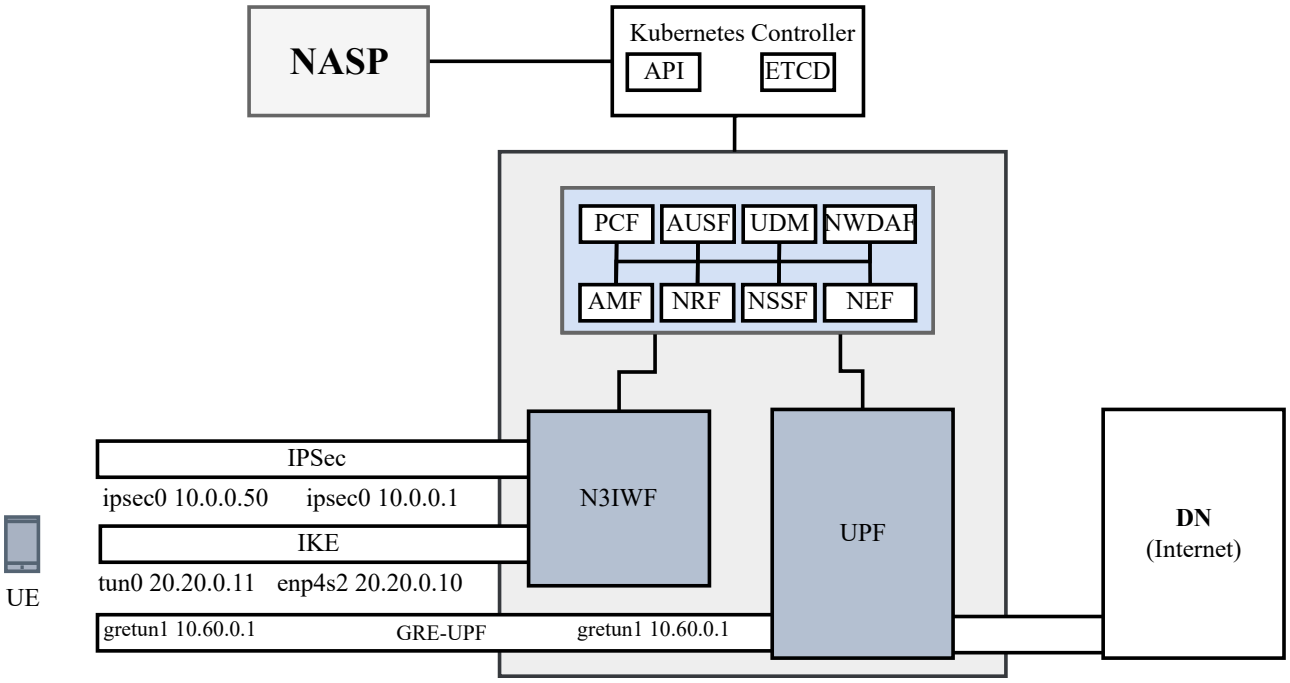


Figura 22: Infraestrutura virtual do NASP.

Inicialmente, é necessário copiar os módulos para o kernel Linux BPF, utilizando o seguinte comando:

```
cp /sys/kernel/btf/vmlinux /usr/lib/modules/`uname -r`/build/
```

Além disso, o módulo GTP5G deve ser instalado, com o seguinte comando:

```
git clone -b v0.8.2 https://github.com/free5gc/gtp5g.git
cd gtp5g && make clean && make && make install
```

Configurando a máquina Master Kubernetes (ETCD instalado)

Para configurar o *Cluster* Kubernetes é necessário configurar a máquina *Master*, i.e., o componente *etcd*, obtendo acesso a um Sistema Operacional, por exemplo Ubuntu 22.04 com os recursos computacionais especificados anteriormente. Desta forma, acesse via terminal a máquina em que será instalado o núcleo SBA com o *software* free5GC e execute o seguinte comando para instalar as bibliotecas do Python, python3-pip e python3-venv.

```
apt install -y python3-pip python3-venv
```

Em seguida, é necessário instalar o Kubernetes com o software Kubespray disponível no GitHub. Kubespray é uma ferramentas de provisionamento e gerenciamento de configuração de *clusters* Kubernetes.

```
git clone https://github.com/kubernetes-sigs/kubespray.git
```

Para o correto funcionamento do Kubernetes é necessário gerar a chave pública RSA e copiá-la para o arquivo .autozired_keys em cada máquina do cluster Kubernetes. Desta forma, a comunicação entre os nodos do *cluster* Kubernetes utiliza uma transmissão segura de dados.

```
ssh-keygen -t rsa -C "$HOSTNAME" -f "$HOME/.ssh/id_rsa" -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

cat ~/.ssh/id_rsa.pub
```

Além disso, é importante configurar os pacotes do Kubespray, como apresentado abaixo:

```
VENVDIR=kubespray-venv
KUBESPRAYDIR=kubespray
python3 -m venv $VENVDIR
source $VENVDIR/bin/activate
cd $KUBESPRAYDIR
pip install -U -r requirements.txt
```

A partir desse ponto, já é possível criar o inventário do Kubespray, com o seguinte comando:

```
cp -rfp inventory/sample inventory/mycluster
```

O próximo passo é declarar os IPs do cluster Kubernetes. Nessa configuração, é importante ter atenção nos espaço em branco entre os IPs.

```
declare -a IPS=(IP_1 IP_2 IP_3)
# should look like
# declare -a IPS=(10.116.0.3 10.116.0.2)
```

A configuração do inventário deve ser realizada com o seguinte comando:

```
CONFIG_FILE=inventory/mycluster/hosts.yaml python3
   ↪ contrib/inventory_builder/inventory.py ${IPS[@]}
```

A rede do *cluster* Kubernetes deve ser configurada utilizando o *software* Multus. Esse *software* é um *plugin* de interface de rede de contêiner para Kubernetes que permite adicionar várias interfaces de rede a *pods*.

```
sed -e 's/kube_network_plugin_multus:
   ↪ false/kube_network_plugin_multus: true/' -i
   ↪ roles/kubespray-defaults/defaults/main.yaml; grep
   ↪ plugin_multus roles/kubespray-defaults/defaults/main.yaml
```

Posteriormente, a configuração do Multus, é interessante habilitar o redirecionamento dos IPs, utilizando o *software* Calico. Esse *software* é uma solução de rede que permite que cargas de trabalho do Kubernetes se comuniquem de maneira contínua e segura.

```
sed -e 's/calico_allow_ip_forwarding:
   ↪ false/calico_allow_ip_forwarding: true/' -i
   ↪ roles/network_plugin/calico/defaults/main.yml; grep forw
   ↪ roles/network_plugin/calico/defaults/main.yml
```

A partir deste ponto, deve-se executar a ferramenta Ansible para configurar e automatizar o cluster Kubernetes e deixá-lo preparado para a instalação do núcleo SBA.

```
ansible-playbook -i inventory/mycluster/hosts.yaml  --become
   ↪ --become-user=root cluster.yml
```

A saída esperada do *playbook* do Ansible pode ser observada na Figura 23. Através do *playbook* pode-se analisar que a instalação e configuração do ambiente do cluster Kubernetes é automatizado.

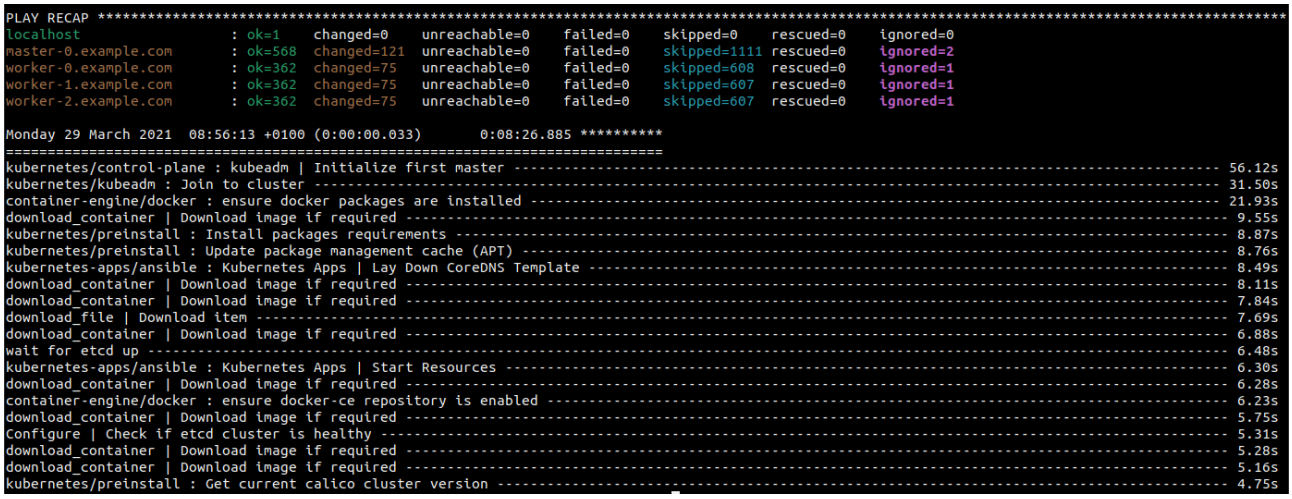


Figura 23: Saída do playbook Ansible.

Após a instalação do *playbook*, é possível instalar a ferramenta de linha de comando do Kubernetes, chamada *kubectl*. Com essa ferramenta é possível visualizar o estado de todos os nodos do *cluster*.

```
curl -fsSL
    ↪ https://packages.cloud.google.com/apt/doc/apt-key.gpg |
    ↪ sudo gpg --dearmor -o
    ↪ /etc/apt/keyrings/kubernetes-archive-keyring.gpg

echo "deb [signed-by=/etc/apt/keyrings/kubernetes-
    ↪ archive-keyring.gpg] https://apt.kubernetes.io/
    ↪ kubernetes-xenial main" | sudo tee
    ↪ /etc/apt/sources.list.d/kubernetes.list

sudo apt update

sudo apt install kubectl
```

Finalmente, é possível visualizar os nodos do *cluster* Kubernetes com o seguinte comando:

```
kubectl get nodes
```

O resultado do comando acima pode ser visualizado na Figura 24.

Instalação de Ferramentas de Observabilidade de Componentes Kubernetes

Uma das principais características de um ambiente de computação em nuvem é o suporte ao ferramental de observabilidade e de automação do ambiente. Nesse contexto, Kubernetes possui algumas ferramentas que auxiliam na observabilidade do sistema. Para a instalação dessas ferramentas, é necessário acessar via terminal uma máquina do Kubernetes *Cluster*. O primeiro passo é verificar a versão do Kubernetes, com o seguinte comando.

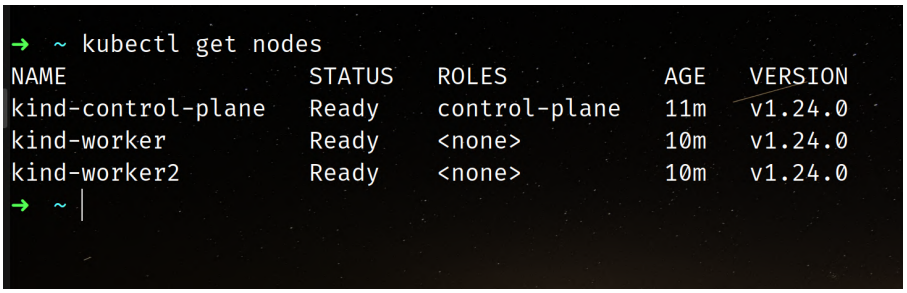


Figura 24: Nodos disponíveis no *Cluster* Kubernetes.

```
sudo kubectl --version
```

Com a versão do Kubernetes, é possível instalar a ferramenta Helm, que auxilia no gerenciamento de aplicações para o Kubernetes. Para iniciar a instalação, use o seguinte comando.

```
curl -fsSL https://raw.githubusercontent.com/helm/helm/master/
   ↪ scripts/get-helm-3 | bash
```

Além disso, é possível instalar a ferramenta Grafana via Helm para obter um *Dashboard* de gráficos que podem ser customizados para a observabilidade do *cluster* Kubernetes.

```
helm repo add grafana https://grafana.github.io/helm-charts
helm create ns grafana
helm install grafana -n grafana
```

Após a instalação da ferramenta Grafana, é interessante utilizar a ferramenta Loki, que é um sistema de agregação de logs multi-tenant, horizontalmente escalável e altamente disponível. Para a instalação do Loki, o comando abaixo é necessário.

```
helm repo add loki https://grafana.github.io/helm-charts
helm create ns loki
helm install loki -n grafana
```

Por fim, recomenda-se a utilização da ferramenta Linkerd, para auxiliar na observabilidade, *logging* e controle de tráfego. Essa ferramenta foi projetada para Kubernetes, nativamente voltada para computação em nuvem. Sua instalação necessita dos seguintes comandos.

```
# To add the repo for Linkerd stable releases:
helm repo add linkerd https://helm.linkerd.io/stable

# To add the repo for Linkerd edge releases:
helm repo add linkerd-edge https://helm.linkerd.io/edge

helm install linkerd-crds linkerd/linkerd-crds -n linkerd
   ↪ --create-namespace

helm install linkerd-control-plane -n linkerd --set-file
   ↪ identityTrustAnchorsPEM=ca.crt --set-file
   ↪ identity.issuer.tls.crtPEM=issuer.crt --set-file
   ↪ identity.issuer.tls.keyPEM=issuer.key
   ↪ linkerd/linkerd-control-plane
```

Chegando nesse ponto, o ambiente da infraestrutura do Kubernetes e as ferramentas de observabilidade estão instaladas. O próximo passo é instalar a plataforma NASP para prover um fatiamento de rede como serviço para redes 6G.

Instalação da NASP na infraestrutura do Cluster Kubernetes

Na máquina local é necessário instalar python3, pip e Poetry. É importante destacar que pip e Poetry são ferramentas para gerenciamento de dependências e empacotamento do Python. Por exemplo, Poetry permite declarar as bibliotecas que o projeto NASP depende e que irá gerenciá-las (instalá-las/atualizá-las).

```
sudo apt install -y python3 python3-pip poetry
```

O próximo passo, é clonar o repositório NASP disponível no GitHub.

```
git clone https://github.com/fhgrings/NASP.git
```

Acesse o diretório NASP em:

```
cd NASP/nasp
```

Instale e atualize os pacotes *Python* com a ferramenta *Poetry*.

```
poetry install --all
```

Após a instalação, utilize o seguinte comando para executar a NASP no ambiente *poetry*.

```
poetry run flask run --debug
```

Finalmente, copie o arquivo .kube/config de uma das máquinas do *cluster* kubernetes e cole em:

```
./NASP/config/kubeconfig
```

7.3 Experimentos de Validação, Resultados e Análise

Após a instalação e configuração das quatro etapas para prover o ambiente baseado em computação em nuvem, a plataforma NASP está disponível em http://localhost:5000/nasp. Dessa forma, é possível realizar experimentos de validação da NASP e obter resultados para análise. O primeiro passo para utilizar a NASP é definir uma infraestrutura para as redes de Acesso, Transporte e Núcleo. É importante destacar que a rede de Transporte é opcional na atual versão da NASP. Um exemplo dessa infraestrutura pode ser obtido no repositório https://github.com/fhgrings/NASP/tree/main/helm_charts. Para utilizar esse exemplo, abra o arquivo values.yaml no diretório raiz do repositório Helm e atualize os valores de entrada necessários para a infraestrutura escolhida e, posteriormente, salve o arquivo.

```
1 global:
2   name: "free5gc"
3   userPlaneArchitecture: "single"  # possible values are "single" and
      ↪ "ulcl"
4   nrf:
5     service:
6       name: "nrf-nnrf"
```

```
      type: "ClusterIP"
      port: "8000"
      nodePort: "30800"
    nodeSelector:
  sbi:
    scheme: "http"
  amf:
    service:
      name: "amf-n2"
      port: "38412"
      targetport: "31412"
      protocol: "SCTP"
      type: "NodePort"
      n2:
```

Após atualizar e salvar as configurações, compacte a pasta inteira em um arquivo ZIP e carregue na plataforma http://localhost:5000/nsst, usando o botão "Add Subnet Slice Template", como pode ser observado na Figura 25.

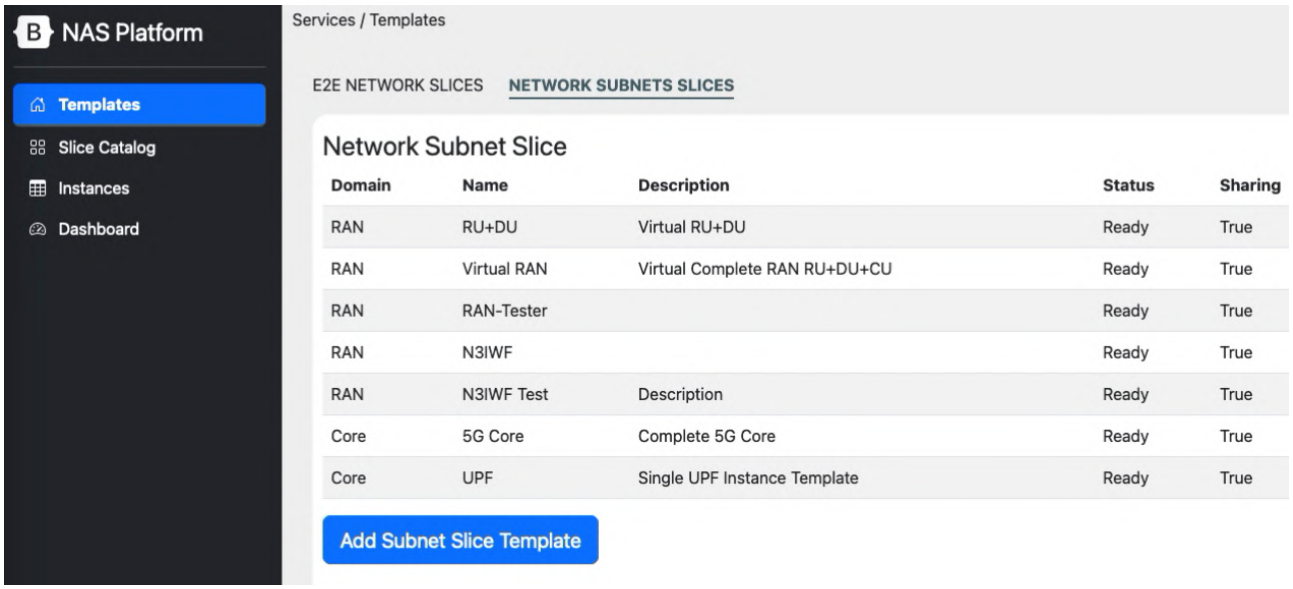


Figura 25: Adicionar Modelo de Fatiamento de Sub-rede.

Para adicionar um UE na rede, copie o arquivo de https://github.com/fhgrings/NASP/tree/main/helm_charts/rantester e novamente compacte a pasta em um arquivo ZIP. Posteriormente, importe na aba NSST da plataforma NASP, como apresentado na Figura 26.

Após a criação do modelo, é necessário implantá-lo na plataforma NASP. Para isso, crie um NST usando o núcleo SBA com a ferramenta free5GC e a RAN com um UE através da ferramenta RAN_Tester. Essa ferramenta emula uma rede de acesso com a possibilidade de instanciar vários UEs. Após a criação, pressione "Deploy Slice", como apresentado na Figura 27.

Após alguns segundos, a fatia de rede aparecerá como uma instância na guia "Network Slice Instances", como pode ser observado na Figura 28.

Testes de disponibilidade, conectividade e desempenho

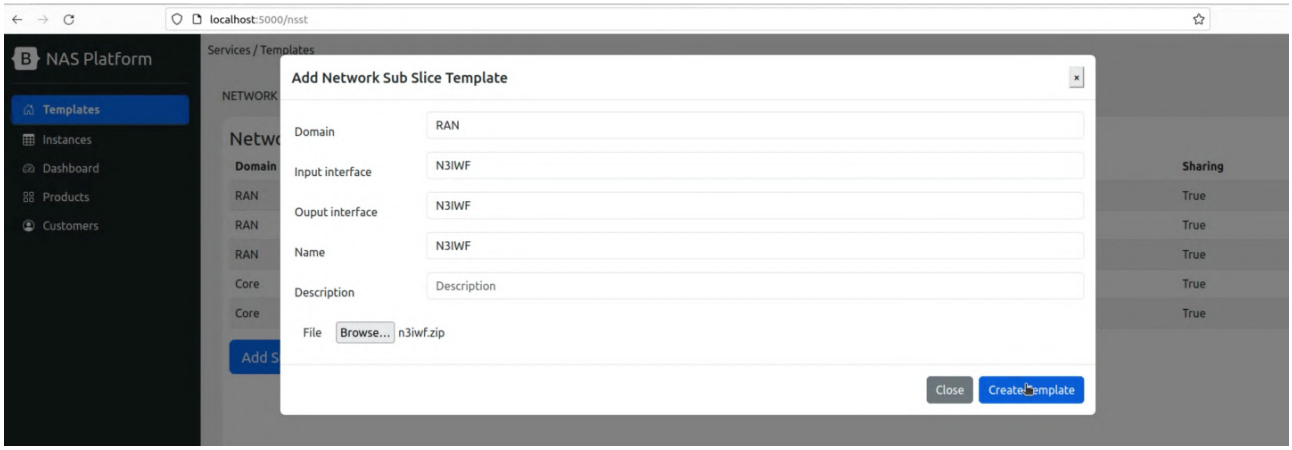


Figura 26: Criar Modelo de Fatiamento de Sub-rede.

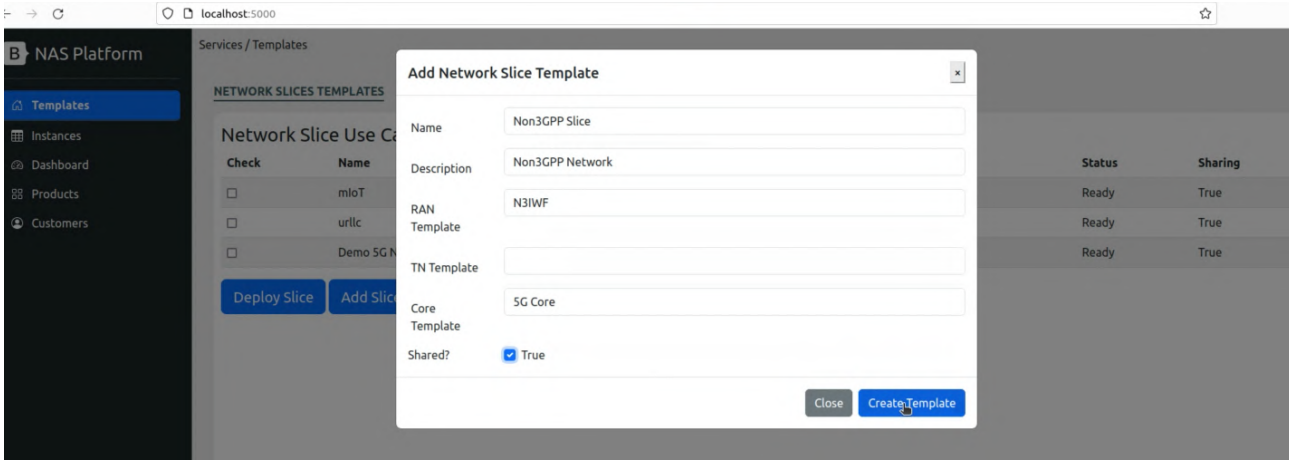


Figura 27: *Deploy Slice* na plataforma NASP.

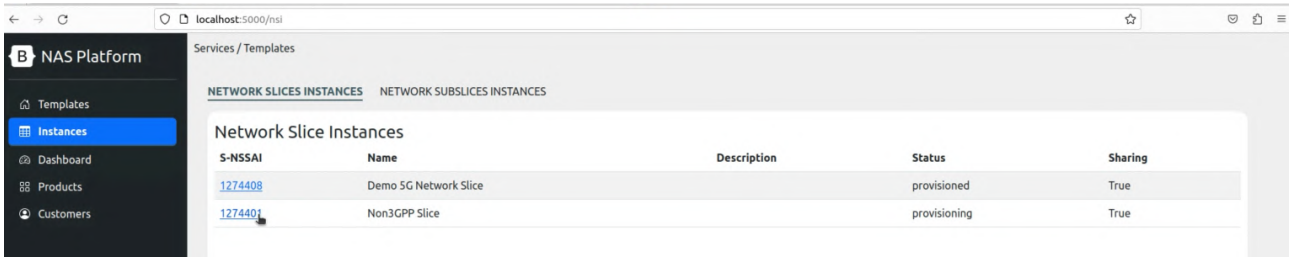


Figura 28: Instâncias da Plataforma NASP.

Inicialmente, para verificar onde o UE está localizado no *cluster* Kubernetes, use o comando abaixo:

```
kubectl get pods -A -o wide | grep -e rantester -e NAME
```

O retorno do comando acima pode ser visto na Figura 29.

Para conectar-se a UE e executar os testes de carga de rede, basta utilizar o comando abaixo:

```
kubectl exec -it ran-rantester-0 -n {GIVEN_NAMESPACE} -- bash
```

```
C02FJ1DRMD6Q :: ~/Documents/mestrado » kubectl get pods -A -o wide | grep -e rantester -e NAME
NAMESPACE     NAME                                READY   STATUS    RESTARTS        AGE   IP              NODE    NOMINATED NODE   READINESS GATES
default       ran-rantester-0                     1/1     Running   0               25h   10.233.102.162  node1   <none>           <none>
```

Figura 29: Disponibilidade do *Slice* da RAN e UE.

Dentro do PON, é possível executar um teste de conectividade com o *ping*, por exemplo.

```
kubectl exec -it ran-rantester-0 -n {GIVEN_NAMESPACE} -- bash
```

A Figura 30 apresenta o resultado do *ping* com uma latência média de 1.883 ms.

```
root@ran-rantester-0:/workspace/my5G-RANTester/cmd# ping -I uetun1 www.google.com
PING www.google.com (142.251.40.132) from 10.1.0.1 uetun1: 56(84) bytes of data.
64 bytes from lga25s80-in-f4.1e100.net (142.251.40.132): icmp_seq=1 ttl=116 time=1.80 ms
64 bytes from lga25s80-in-f4.1e100.net (142.251.40.132): icmp_seq=2 ttl=116 time=1.90 ms
64 bytes from lga25s80-in-f4.1e100.net (142.251.40.132): icmp_seq=3 ttl=116 time=1.94 ms
^C
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.800/1.883/1.942/0.070 ms
root@ran-rantester-0:/workspace/my5G-RANTester/cmd#
```

Figura 30: Teste de disponibilidade.

Com o tráfego de dados ativado com o *ping*, é possível analisar os dados detalhadamente com a ferramenta Grafana. O Dashboard dessa ferramenta deve começar a mostrar os dados de métricas relacionados à instância do fatiamento de rede em execução, como pode ser observado na Figura 31.

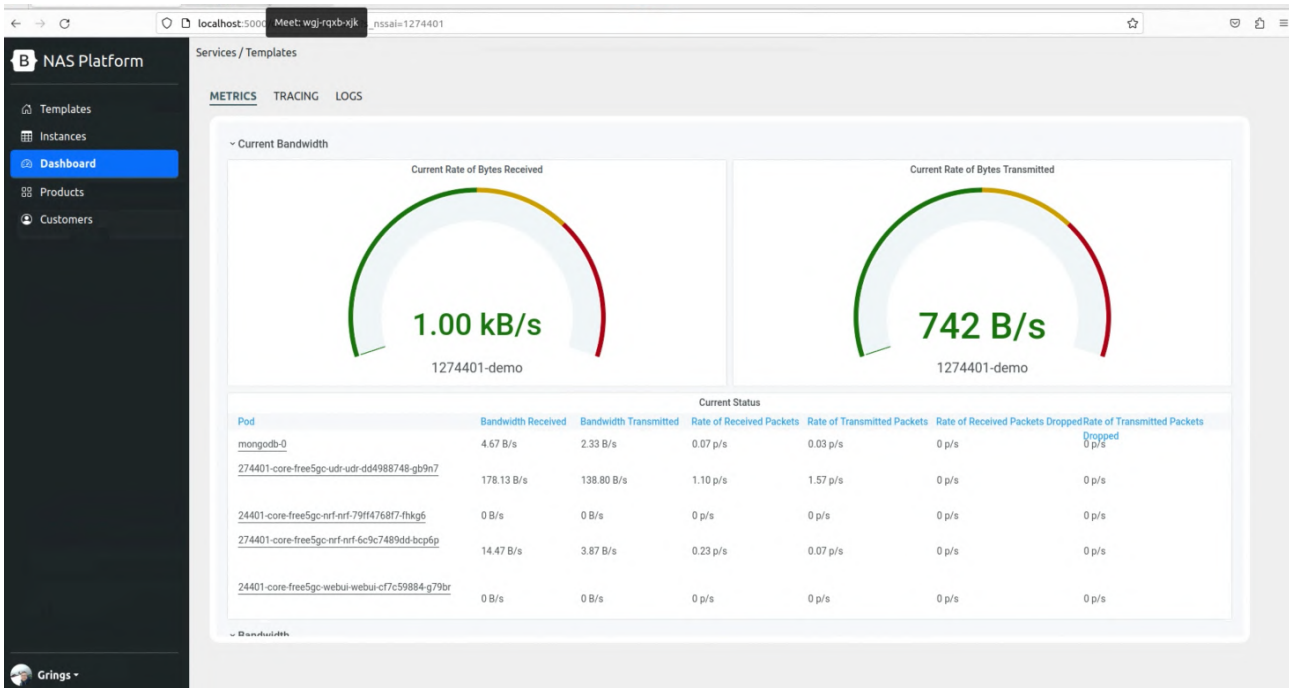


Figura 31: Dashboard do Grafana.

Finalmente, testes de observabilidade podem ser vistos na Figura 32. Com essa visualização é possível analisar se todos os serviços disponibilizados pela plataforma NASP estão ativos e sem erros. Observabilidade vem se tornando uma habilidade essencial para manter sistemas nativos de nuvem. Oriundo da teoria de controle, o termo observabilidade enfatiza a necessidade de trazer maior visibilidade à compreensão do comportamento de um sistema complexo usando telemetria coletada do sistema em tempo de execução. Nesse contexto, diferentemente de um monitoramento caixa preta, o objetivo da observabilidade é minimizar o tempo necessário para que operadores de sistemas obtenham uma visão contextual precisa sobre a corretude e o desempenho do sistema. Dessa forma, os resultados de observabilidade são fundamentais para à compreensão do comportamento dos diferentes fatiamentos em 6G. Por exemplo, o trabalho do projeto Brasil 6G de Bruno et al. [9], publicado no Journal of Internet Services and Applications (JISA), investigou a detecção de anomalias em sistemas nativos de computação em nuvem usando soluções *Commercial Off-The-Shelf* (COTS) de observabilidade e aprendizado de máquina.

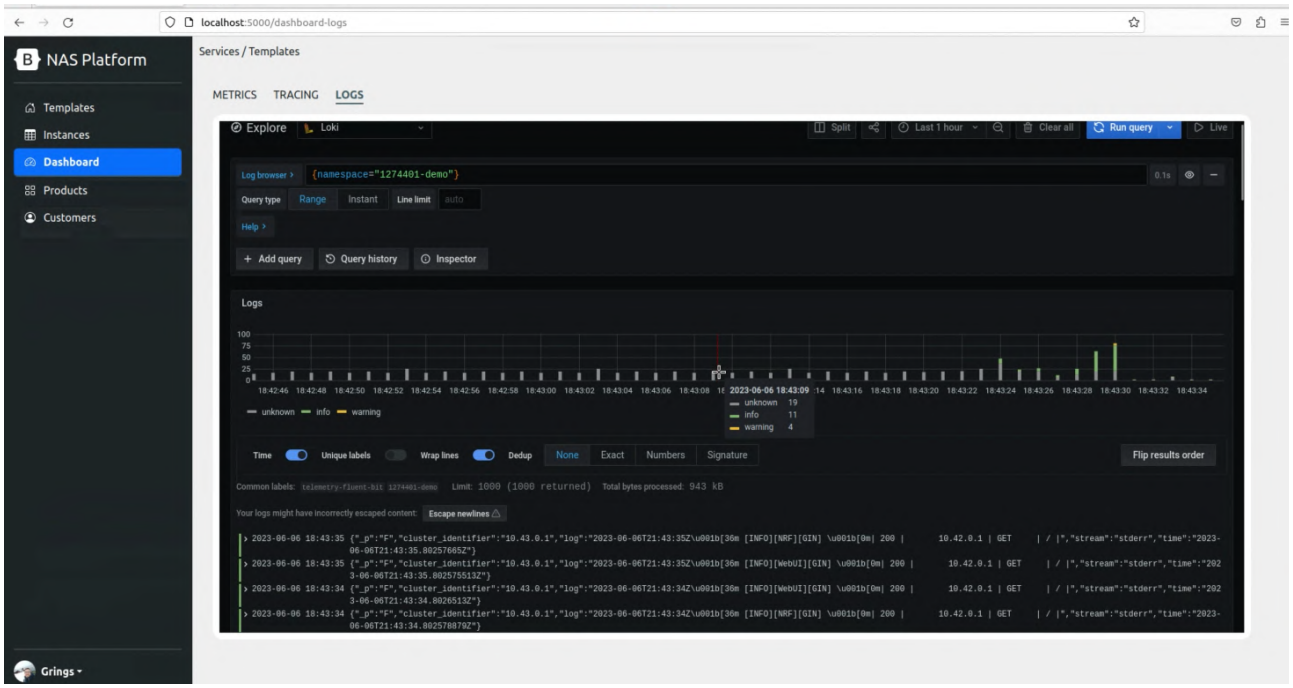


Figura 32: Teste de observabilidade.

8 Ação 7 - Rede Satelital

A Figura 33 mostra um corte vertical do Caminho 1 original proposto para a integração dos componentes satelitais advindos do projeto SINDISAT ao projeto Brasil 6G. Nessa figura, no plano de controle, o laptop cliente de *Content Distribution Network* (CDN) (ou seja, que fica do lado remoto da rede) implementaria a pilha NAS/TCP/Inner-IP/IPSec/IP/Ethernet. Em seguida, a UE 6G receberia os quadros Ethernet gerados pelo laptop (contendo IP) e geraria um sinal de RF. A BS 6G receberia então os pacotes IP do outro lado da rede e entregaria para um terminal satelital (*Terminal Server*). O *Terminal Server* implementaria um *gateway* TCP que receberia os segmentos transportados nesse enlace desde o *laptop*, retirando o conteúdo (NAS) desses segmentos e entregando ao TCP *Optimizer* (implementado em *hardware* ou *software*). Do outro lado do enlace satelital, os segmentos TCP seriam recebidos e a carga útil entraria em outra conexão TCP. Logo, no *Hub Server* existiria um *gateway* reverso. A carga útil (NAS) seria então entregue a função N3IWF. Finalmente, ainda no plano de controle, a N3IWF entregaria os pacotes NAS a AMF usando a pilha de protocolos NGAP/SCTP/IP/Ethernet.

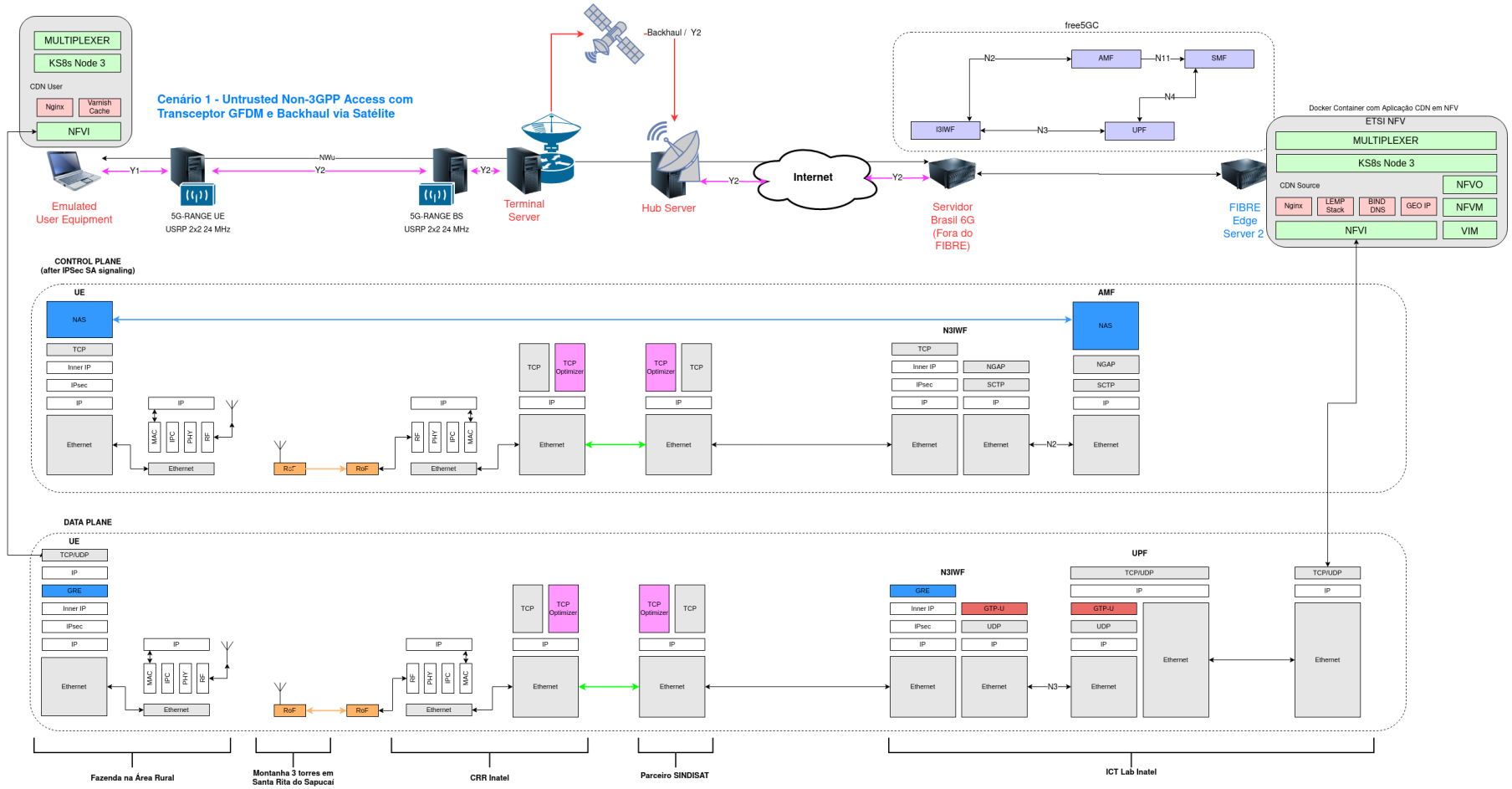


Figura 33: Caminho 1 original para a integração da rede satelital como rede de acesso à Internet no Projeto Brasil 6.

Com a finalização do desenvolvimento e testes da função N3IWF adicionada para completar o núcleo do Proto6G no período desse relatório, pudemos agora trabalhar em uma variação do Caminho 1 originalmente proposto no Relatório Final das Atividades 5.1 e 5.2. Essa variação utiliza o satélite para conectar o núcleo Free5GC a Internet. Ou seja, todo a solução 5G é servida pelo link de satélite. A proposta original de usar o satélite como backhaul para a rádio base 5G/6G continua válida, mas dependente de um caminho de retorno configurável a partir do hub satelital até o ICT Lab, no Inatel. Para essa condição, o uso de IPs públicos no ICT Lab precisa novamente ser habilitado. Muitas mudanças de configuração acontecerão no Inatel desde que o Projeto B foi finalizado.

A Figura 34 ilustra o Caminho 1 adaptado para o contexto atual do projeto Brasil 6G com o núcleo Free5GC e sua nova função N3IWF. No ICT Lab, uma dupla de PCs (Desk03 e Desk08) emulam o enlace criado com a MAC virtual dos transceptores BR6G. Ou seja, esses dois computadores executam uma versão virtual idêntica da MAC (BR6G_MAC) dos transceptores físicos do CRR. O PC a esquerda na figura (Desk03) deve conter ainda os programas que emulam um terminal de usuário 5G e os pacotes de software necessários para o lado cliente da CDN. Esses pacotes serão os mesmos usados na dissertação de mestrado do aluno Tibério Tavares Rezende, nomeadamente: Nginx e Vanish cache. O PC Desk08 também conterá uma instância da BR6G_MAC, bem como o software que emula uma rádio base 5G. Possui ainda a implementação da função N3IWF para acesso não confiável não 3GPP ao núcleo Free5GC.

O Servidor do Brasil 6G contém todos os demais componentes do núcleo do Free5GC, incluindo os novos componentes desenvolvidos para o Proto6G. Ele está conectado a um servidor do projeto SINDISAT (Server #44251). Esse servidor contém diversos pacotes de software para preparar a comunicação via satélite SGDC-1 da Telebras. Após o tráfego subir para o satélite, ele desce no *hub* satelital em Brasília/DF onde é levado à Internet global. Pela Internet global é possível se acessar o último servidor, o PC Desk04, que conterá diversos pacotes de software do lado servidor da CDN de teste.

Em Dezembro de 2023, um teste de *ping* entre o Desk03 e o DNS do Google 8.8.8.8 foi realizado, de forma a demostrar essa configuração. Estamos agora instalando os pacotes de CDN nos PCs Desk03 e Desk04. Também pretendemos inserir nesse cenário um switch OpenFlow disponível no Inatel para avaliar a participação de *Software Defined Networking* (SDN) na rede.

8.1 Configuração do Ambiente de Testes por Conexão via Satélite

De acordo com as informações disponibilizadas pelos parceiros do SINDISAT, existiu a possibilidade em análise de dois ambientes de teste via satélite a serem exploradas para o projeto Brasil 6G. Em relação a solução GEO em banda Ka, o Inatel representado pelos laboratórios de pesquisa CRR e ICT Lab, possui experiência em projetos anteriores em configuração em banda Ku, utilizando apenas um satélite. Considerando as experiências anteriores foram propostas algumas soluções entre os parceiros e a opção escolhida foi a Viasat/Telebras com o Satélite Geoestacionário de Defesa e Comunicações Estratégicas (SGDC). Nesse caso de uso foi fornecido ao projeto o satélite da Telebras em banda Ka, SGDC 1, com equipamentos de rede da Viasat. A vsat remota foi instalada e configurada em Santa Rita do Sapucaí/MG no campus Inatel e o *Gateway* (ou *hub*) foi configurado na estação de Brasília/DF. Em seguida um canal de comunicação entre Brasília/DF e o Inatel foi configurado e testado com sucesso.

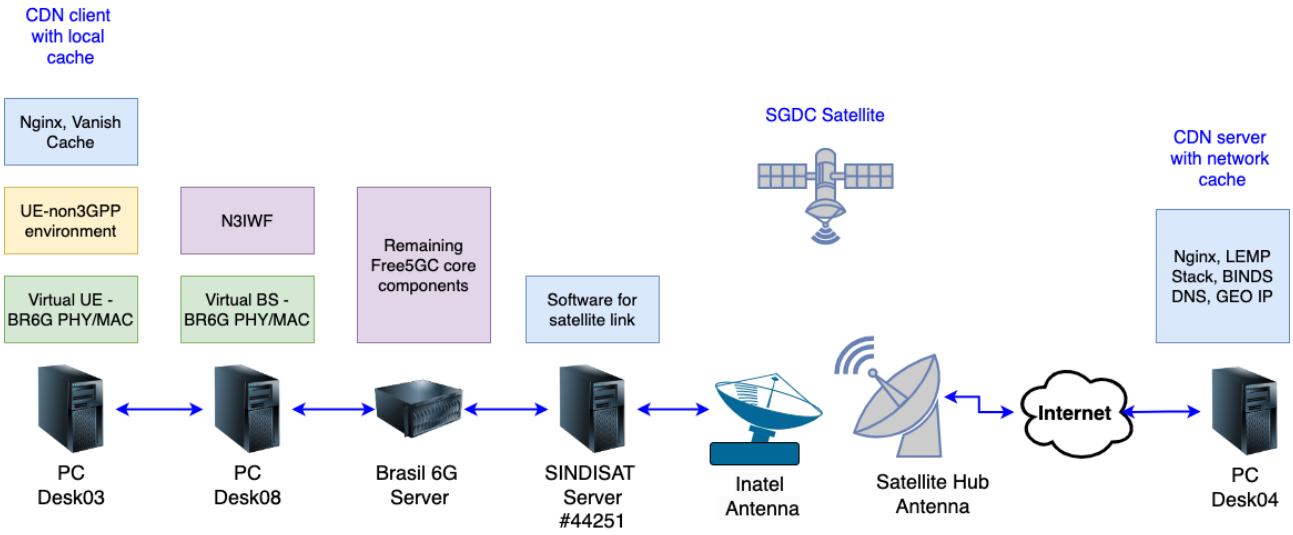


Figura 34: Caminho 1 adaptado para a integração da rede satelital como rede de acesso à Internet no Projeto Brasil 6.

8.2 Satélite Geoestacionário de Defesa e Comunicações Estratégicas

O SINDISAT sendo representado pelas empresas Viasat/Telebras, e em parceria com o Inatel, disponibilizou o satélite SGDC para realização de pesquisas e experimentação. O SGDC é o primeiro satélite do governo brasileiro e o primeiro da operadora Visiona Tecnologia Espacial a ser lançado pela empresa francesa Arianespace em contrato conjunto com a Thales Alenia Space. A Visiona Tecnologia Espacial é formada por um consórcio entre as empresas de telecomunicações e aviação Telebras e a Embraer. O SGDC é dedicado às comunicações estratégicas para o Ministério da Defesa, bem como à implantação de serviços de banda larga para o Ministério das Telecomunicações brasileiro. Em particular, o principal objetivo e caso de uso do SGDC é proporcionar às populações de moradias isoladas dos grandes centros urbanos acesso a serviços de banda larga a Internet. O SGDC foi lançado ao espaço no dia 04 de maio de 2017 e a altura aproximada de sua órbita é de 35.786 km e velocidade em órbita de 3,07 km/s. O SGDC está posicionado a 75° na longitude oeste e oferece cobertura de banda X (faixa de frequência para comunicação por satélite privativa) em toda a América do Sul e rotas marítimas vizinhas, e também possui carga útil em banda Ka (faixa de frequência compreendida entre 27 e 40 GHz) com cobertura completa de norte a sul do Brasil. A duração de vida útil planejada para o satélite é de 18 anos. A Tabela 2 detalha as especificações do SGDC.

Tabela 2: Especificações do satélite SGDC.

Equipamento:	Satélite Geoestacionário de Defesa e Comunicações Estratégicas (SGDC)
Órbita:	Geoestacionária
Lançador:	Ariane 5 ECA
Base de Lançamento:	Kourou, Guiana Francesa
Posição:	75° *West* (oeste)
Plataforma modelo:	Spacebus 4000 C4
Segmentos de solo:	Centros de TT&C: 1 e Gateways: 4
Envergadura:	37 metros
Altura:	7 metros
Peso:	5,8 tons
Potência:	11 kW
Vida útil:	18 anos

A estrutura do enlace via satélite SGDC no Inatel é feita com a instalação da antena parabólica receptora de sinal foi feita no campus da faculdade no prédio 2, e o modem no ICT Lab. Inicialmente fez-se um estudo do melhor local de instalação em conjunto com setores de engenharia elétrica e segurança do trabalho do Inatel. Foi constatado que o melhor local de instalação era acima do laboratório de experimentação com conexão de cabo coaxial até o modem no ICT Lab. A antena foi instalada com visão direta ao satélite SGDC geoestacionário localizado na posição orbital de 75º de longitude oeste.

9 Ação 8 - Inteligência Artificial

A Seção 9 traz os avanços relativos a integração de AI com os componentes do Proto 6G tal qual realizados em 2023. De forma macro, as atividades desenvolvidas envolvem:

- Especificação técnica de uma evolução da função NWDAF, aqui chamada eNWDAF, com base na 3GPP TS 29.520 version 16.6.0 [10] e TS 23.288 version 16.6.0 Release 16 [11];
- Implementação de componentes do núcleo do Proto 6G NWDAF (GitHub - https://github.com/open6gc);
- Experimentação com o cenário de uso "Network Performance" *analytics* - especificado na Seção 6.6 da TS 23.288 version 16.6.0 Release 16 [11]:
  - Alteração na função AMF implementada pelo Free5GC para suportar `Namf_Event_Exposure Subscribe` e `Notify`;
  - Implementação do componente/microserviço de "Network Performance" *analytics* no Proto 6G NWDAF;
  - Realização de experimento simulado utilizando Free5GC e UERASIM;
- Adaptação da especificação da eNWDAF para cobrir os novos procedimentos especificados na 3GPP TS 23.288 version 18.4.0 [12] lançada em 19/12/2023.

9.1 Visão em alto-nível do Proto 6G eNWDAF

O Proto 6G NWDAF é uma implementação *open-source* da função NWDAF especificada pelo 3GPP TS 29.520 e TS 123.288 version 16.6.0 Release 16[8], que tem como propósito central fornecer serviços de *analytics* para outras Funções de Rede (NFs - *Network Functions*), permitindo a tomada de decisões embasadas em algoritmos de inteligência artificial e métricas coletadas da rede. Este protótipo parte da NWDAF como especificada e detalha a sua especificação técnica em direção ao eNWDAF vislumbrado como a evolução dos mecanismos e funções associadas à AI no núcleo capaz de suportar os casos de uso e os requisitos do 6G nesta área.

Considerando a ampla gama de possibilidades de utilização de funções de *analytics* e dificuldade de prover uma solução completa para todos os possíveis cenários, optamos por especificar o Proto 6G eNWDAF como uma solução modular baseada em microsserviços que viabiliza a implantação desacoplada de funções de *analytics* de acordo com as necessidades dos operadores e especificidades da rede. Essa abordagem também provê mais liberdade para os operadores, que podem escalar a solução de acordo com a dimensão de suas redes, optar pela utilização de funções de *analytics* de diferentes fabricantes, e decidir por implantar apenas as funções de *analytics* definidas como requisitos para o núcleo 5G/6G implantado.

O Proto 6G eNWDAF também define coletores de métricas como componentes desacoplados pelo mesmo motivo, o que permite fácil adaptação desta solução às nuances da rede e às necessidades do operador.

Na v18.4.0, a especificação TS 23.288 [12] adicionou a *Data Collection Coordination Function* (DCCF), uma função de rede com objetivos análogos aos coletores do Proto 6G eNWDAF. A definição de quais coletores precisam ser implantados tem ligação com a definição de quais

[8] A solução prevê os componentes arquiteturais necessários para implementação da especificação na v18.4.0 que foi lançada posteriormente à concepção. A Subseção 9.1.5 esclarece as diferenças para a v18.4.0.

funções de *analytics* são necessárias. Sendo assim, essa flexibilidade permite a otimização na utilização de recursos, bem como maior escalabilidade. Além disso, considerando que determinados componentes da rede, tais como o OAM, podem ser providos por diferentes fornecedores, acreditamos que o Proto 6G eNWDAF deve suportar a utilização de coletores específicos de acordo com as interfaces de extração de métricas disponíveis.

Apesar de se apresentar como um simples barramento de funções de *analytics* (tal como um barramento de serviços definido pela *Service Oriented Architecture* - SOA), o eNWDAF definido neste trabalho também assume a responsabilidade de armazenar as métricas e garantir que atendam aos critérios de qualidade. Adicionalmente, o NWDAF também incorpora uma camada de *caching* para otimizar a utilização de recursos computacionais na execução das funções de *analytics*(considerando parâmetros e atualização de métricas). Portanto, a função *Analytics Data Repository Function* (ADRF) está prevista nessa implementação através do configdb, analytics-cache e metrics-store, os quais serão especificados a frente.

Além do DCCF e do ADRF, a função *Message Framework Adaptation Function* (MFAF), descrita mais a frente e externa ao NWDAF, foi especificada no Proto 6G para melhor funcionamento da mensageria assíncrona.

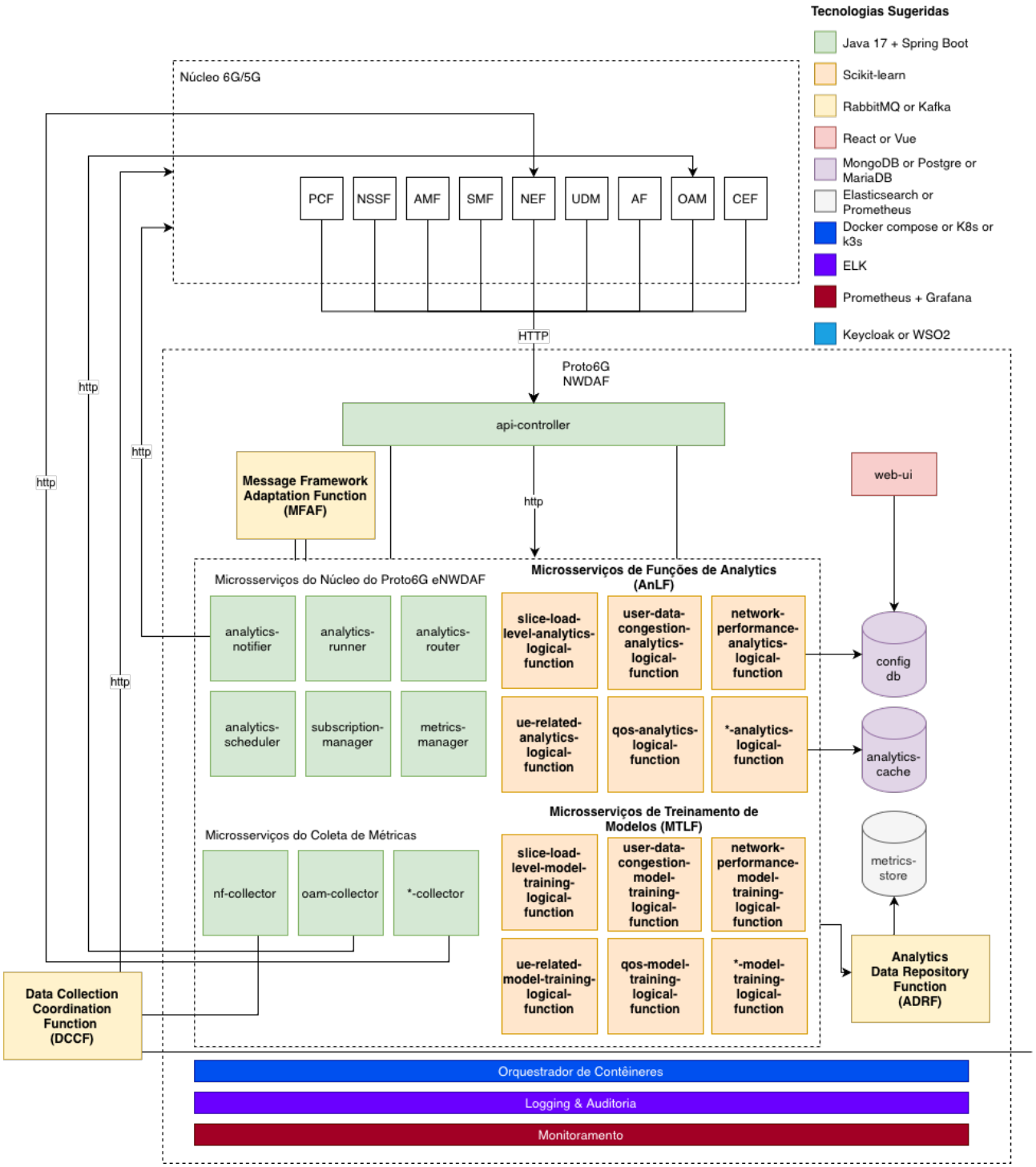


Figura 35: Especificação em alto nível do Proto 6G eNWDAF.

A Figura 35 apresenta, em alto nível, os serviços que compõem o Proto 6G eNWDAF, evidenciando suas interfaces e propondo tecnologias para a implementação. Dentre os componentes, destacam-se os denominados "Componentes do Núcleo do Proto 6G eNWDAF", responsáveis por interfacear a utilização de funções de *analytics* tanto de forma síncrona (*request/reply*) quanto assíncrona (*subscribe/notify*). Esses componentes também têm a função de armazenar e curar as métricas, descobrir funções de rede, balancear requisições e gerenciar o *cache* de respostas. Detalhes adicionais sobre esses componentes são apresentados na Tabela 3:

Tabela 3: Descrição do Componentes Principais do Proto 6G eNWDAF

Componente	**Descrição**	**Detalhes Técnicos**
analytics-notifier	Utilizado para enviar *analytics* para as NFs de maneira assíncrona com base nas subscrições.	Implementado em Java com *Spring Framework*
analytics-runner	Responsável por executar uma função de *analytics* e gerenciar o *cache*	Implementado em Java com *Spring Framework*
analytics-scheduler	Responsável por atender as subscrições de *analytics*, disparando o analytics-runner para a execução e o analytics-notifier para notificação	Implementado em Java com *Spring Framework*
analytics-router	Responsável por distribuir das requisições de *analytics* pelas instâncias de serviços de *analytics* disponíveis de maneira balanceada. Funciona como um serviço de resolução de nomes (e.g. DNS) para as instâncias de serviços de *analytics* da NWDAF	Implementado em Java com *Spring Framework*
subscription-manager	Faz o registro de subscrições em *analytics*. Esse serviço visa apenas a implementação de um CRUD para as subscrições, aplica validações e responde requisições de *analytics*	Implementado em Java com *Spring Framework*
metrics-manager	Responsável por gerenciar a base de métricas, aplicando políticas de controle de retenção e validação. Trata-se de um processo assíncrono que atualiza a metrics-store de maneira transparente para os demais componentes	Implementado em Java com *Spring Framework*

Além dos componentes principais, coletores e funções de *analytics*, o Proto6G eNWDAF define componentes adicionais que possibilitam o armazenamento de métricas, configurações e *cache*, além de possibilitar a interface com componentes externos e a interface administrativa. Esses componentes são detalhados na Tabela 4:

Tabela 4: Descrição do Componentes Adicionais do Proto 6G eNWDAF

Componente	Descrição	Detalhes Técnicos
api-controller	API disponibilizada para as *network functions* (NFs) requisitarem ou subscreverem em *analytics* suportados pelo NWDAF	Implementado em Java com *Spring Framework*
web-ui	Web-UI utilizada para configurar o NWDAF, verificar métricas e executar atividades administrativas, tais como invalidar o armazenamento temporário de dados	Implementado em Java com *Spring Framework*
MFAF	Abstração para acesso ao *broker* de eventos. Permite comunicação assíncrona entre os componentes e a publicação de eventos úteis para *debugging* e monitoramento. Esta função de rede está descrita na TS 23.288 como uma entidade externa ao NWDAF. Porém, fez-se necessário sua construção para que o NWDAF funcione corretamente no modelo assíncrono.	*Off the shelf*: RabbitMQ
metrics-store	Abstração para armazenamento, gerenciamento e busca de métricas utilizadas principalmente pelos serviços de *analytics* e coletores de métricas.	*Off the shelf*: Memcached
config-db	Abstração para armazenar dados contextuais (temporários) tais como subscrições e descoberta de *analytics*. Utilizada pelos componentes de orquestração e agendamento para identificar quais as subscrições precisam ser atendidas e quais instâncias de serviços de *analytics* podem ser utilizadas. Em conjunto com o metrics-store, este componente engloba a implementação do ADRF.	*Off the shelf*: Memcached ou Redis

Conforme mencionado anteriormente, de acordo com as necessidades dos operadores (bem como das Funções de Rede), a eNWDAF pode fazer uso de coletores específicos, completamente desacoplados do núcleo. Por exemplo, é possível ter coletores específicos para um determinado OAM utilizado pelo operador, enquanto os dados obtidos são padronizados e armazenados para análise por meio dos serviços de *analytics*. Exemplos de coletores são apresentados na Tabela 5:

Tabela 5: Exemplos de Coletores do Proto 6G eNWDAF

Componente	Descrição	Detalhes Técnicos
nf-collector	Realiza coleta (*pulling*) de métricas periodicamente implementando as interfaces de exposição de métricas definidas para cada Função de Rede pelo 3GPP.	Implementado em Java com *Spring Framework*
oam-collector	Extrai métricas diretamente do OAM utilizado pelo Operador. Em conjunto com o nf-collector, engloba os serviços previstos no DCCF.	(Fora do escopo deste trabalho)

Além disso, conforme descrito anteriormente, o Proto 6G eNWDAF almeja suportar implementações diversas de serviços de *analytics*, descobrindo essas instâncias automaticamente e distribuindo a carga entre elas por meio do *analytics-router*. A Seção 6 da TS 23.288 define de maneira abrangente todos os tipos de *analytics* que devem ser suportados pelo NWDAF. Portanto, todos os serviços de *analytics* implementados na infraestrutura do operador devem aderir a um dos tipos de *analytics* especificados pelo 3GPP. Exemplos de funções de *analytics* são apresentados na Tabela 6:

Tabela 6: Exemplos de Funções de *analytics* do Proto 6G eNWDAF

Componente	Descrição	Detalhes Técnicos
network-performance-analytics-function	Provê análise sobre a utilização (e.g. largura de banda, processamento, memória) da infraestrutura de rede em uma determinada área e intervalo de tempo	(Fora do escopo deste trabalho)
nf-load-analytics-function	Provê uma análise sobre a utilização de funções de rede em uma determinada área e intervalo de tempo	(Fora do escopo deste trabalho)

9.1.1 Fluxo de Requisição de *Analytics*

A Figura 36 apresenta um Diagrama de Sequência com um fluxo de requisição de *analytics* de forma síncrona:

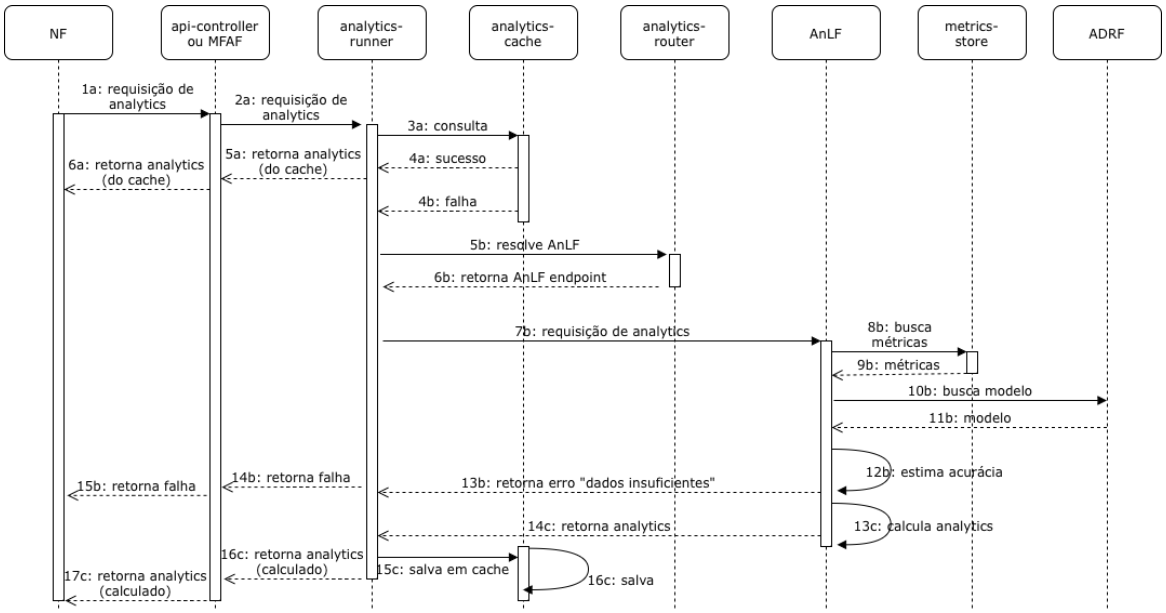


Figura 36: Diagrama de Sequência apresentando cenários possíveis em uma Requisição de *Analytics*.

Como pode ser observado na Figura 36, a requisição normalmente é iniciada por uma *Network Function* (NF) que envia uma requisição de *analytics* através do *api-controller*, componente este implementa a especificação de interface de integração definida pelo 3GPP, ou através do MFAF que permite a abstração da requisição/subscrição de *analytics* através de um barramento de eventos. Independentemente da forma de requisição, o componente de integração aciona o componente *analytics-runner* que é responsável por efetivamente executar a função de *analytics*.

O *analytics-runner* primeiramente verifica o *cache* e, em caso de falha, consulta o *analytics-router* para descobrir qual o *endpoint* de uma *Analytics Logical Function* (AnLF) que pode ser utilizada para obter um relatório de um determinado tipo de *analytics* dentre os especificados pelo 3GPP nas *releases* 16, 17 e 18). Neste ponto, o *analytics-runner* faz uma requisição para a instância de AnLF resolvida pelo *analytics-router*.

A AnLF inicia por buscar um modelo de IA armazenado na ADRF que possa ser utilizado no processamento do *analytics* requisitado. Este modelo é criado de forma assíncrona por um fluxo descrito adiante neste relatório e envolve a utilização de uma *Model Training Logical Function* (MTLF) equivalente que baseado em métricas gera um modelo para ser utilizado de maneira mais reapida pela AnLF. Caso um modelo aplicável não seja localizado, uma falha é retornada para a funçao requisitante.

Caso um modelo de IA esteja disponível, a AnLF busca dados da *metrics-store*, executa o algoritmo de análise de sua implementação com base no modelo criado pela MTLF correpondente e retorna um relatório de analytics. Caso não existam métricas suficientes para executar a função de *analytics* com a acurácia mínima, a AnLF retorna um erro. No percurso de volta, o *analytics-runner* é responsável por armazenar o resultado em *cache* para futuras requisições.

9.1.2 Fluxo de Subscrição em *Analytics*

A Figura 37 apresenta um Diagrama de Sequência com um fluxo de requisição de *analytics* de forma assíncrona (subscrição):

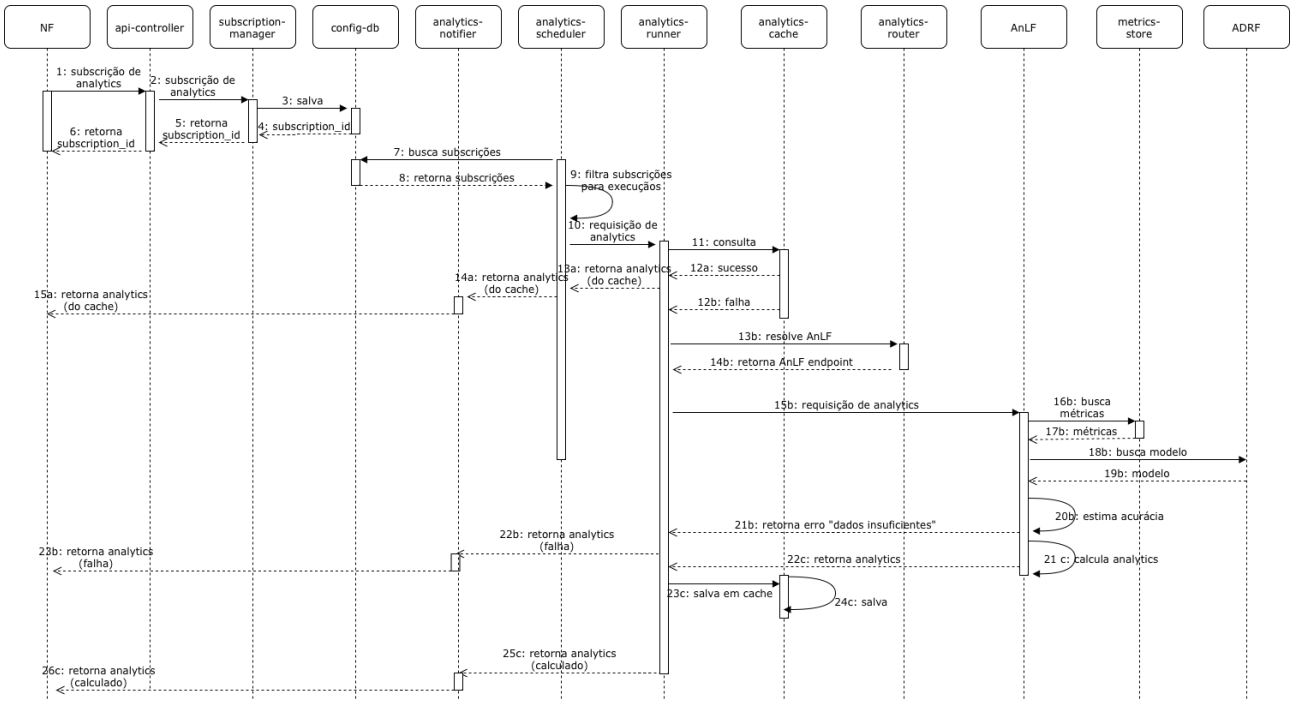


Figura 37: Diagrama de Sequência apresentando desdobramentos possíveis em uma Subscrição de *Analytics*.

Conforme pode ser observado na figura anterior, a subscrição também é normalmente iniciada por uma Função de Rede e processada pelo componente *api-controller* ou MFAF. Ao identificar uma requisição de subscrição, o *api-controller* (ou MFAF) aciona o componente *subscription-manager* para o registro da subscrição, que, por sua vez, gera um *subscription_id* que é retornado à Função de Rede requisitante. Nesse momento, o componente *analytics-scheduler* passa a avaliar a condição de execução da nova subscrição de forma periódica. Quando atendida, faz uma requisição para o componente *analytics-runner*, que segue o fluxo exatamente como descrito na subseção anterior. Ao receber uma resposta do *analytics-runner*, o *analytics-scheduler* envia uma notificação para a Função de Rede subscrita por meio do componente *analytics-notifier*.

9.1.3 Fluxo de Coleta, Gerenciamento e Utilização de Métricas

A Figura 38 apresenta um Diagrama de Sequência com um fluxo de coleta, gerenciamento e utilização de métricas.

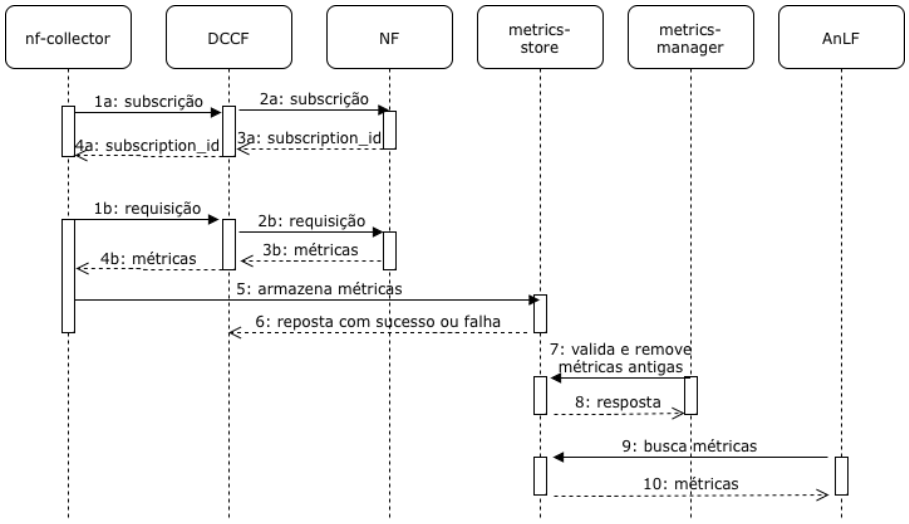


Figura 38: Diagrama de Sequência apresentando o fluxo de coleta, gerenciamento e utilização de métricas.

Como pode ser observado pela Figura 38, os componentes do tipo **-collector* podem coletar métricas de forma passiva ou ativa. Em ambos os casos, caso a coleta de métricas seja em relação a uma NF, deve-se utilizar um DCCF para intermediar a comunicação entre o NWDAF e as NFs, de maneira a evitar consultas repetidas. No exemplo apresentado, o componente *nf-collector* coleta métricas de outras Funções de Rede, e portanto, utiliza um DCCF. Ao obter as métricas o coletor as armazena na *metrics-store* conforme o formato do Proto 6G eNWDAF.

Para garantir a qualidade e evitar métricas antigas, o Proto 6G implementa o *metrics-manager*, executado periodicamente e/ou acionado por eventos do *event-manager*, que realiza a curadoria das métricas, aumentando a precisão das funções de *analytics*.

Ressalta-se que o eNWDAF aqui especificado possui uma arquitetura que permite a implantação distribuída dos seus componentes, de forma que a aplicação eNWDAF possa ser utilizada como um ponto de troca entre *Public Land Mobile Network*s (PLMNs) e, ademais, possa ser integrado com servidores de *Federated Learning* (FL) através do api-controller. O NWDAF como ponto de troca e a sua integração com servidores FL está prevista na TS 23.288 versão 18.4.0.

9.1.4 Fluxo de Treinamento de Modelos

A Figura 39 apresenta um Diagrama de Sequência com um fluxo de treinamento de modelos para a aplicação de algoritmos de inteligência artificial.

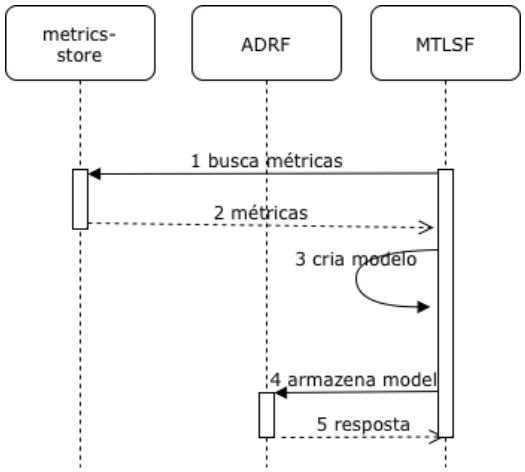


Figura 39: Diagrama de Sequência apresentando o fluxo de treinamento de modelos de IA.

Como pode ser observado pela Figura 39, o fluxo é iniciado por uma MTLF que com base nas métricas acessadas via *metrics-store* gera um modelo e armazena-o na ADRF, para subsequente uso por uma AnLF equivalente.

9.1.5 Considerações referentes às *releases* 16, 17 e 18

O NWDAF especificado para o Proto 6G, aqui chamado eNWDAF, e descrito neste documento foi concebido de acordo com a especificação TS 23.288 versão 16.6.0 *Release* 16 [11]. Esta concepção ocorreu ao longo do ano de 2023 e trata-se de um detalhamento técnico de como o NWDAF pode ser implementado, considerando as tecnologias, padrões e ferramentas. A TS 23.288 traz os requisitos funcionais e interfaces, mas não relata tecnicamente como a solução pode ser desenvolvida.

No fim de 2023, particularmente em 19/12/2023 o 3GPP lançou a versão 18.4.0 da TS 23.288, *Release* 18 [12]. Da *release* 17 até a *release* 18 o 3GPP esmiuçou o NWDAF em maiores detalhes, trazendo inclusive novas funções para AI e *analytics* como um todo, quais sejam: MFAF, ADRF e DCCF.

Como ponto positivo do Proto 6G eNWDAF, pode-se afirmar que essas funções, apesar de terem sido descritas posteriormente em data corrente, estavam previstas na concepção do Proto 6G eNWDAF, com apenas uma diferença de nomenclatura: *event-manager*, *config-db/metrics-store* e microsserviços de coleta de métricas, representando respectivamente MFAF, ADRF e DCCF.

Entendeu-se que essas funções, separadas logicamente do NWDAF, agregam valor ao funcionamento de AI e *analytics* e portanto foram consideradas no Proto 6G, conforme mostra a Figura 35. Tais funções, assim como o NWDAF, também não estão disponíveis nos principais projetos de código aberto disponíveis. Como ponto positivo, entende-se que o Proto 6G está seguindo na mesma direção do 3GPP visto que, precedentemente a *release* 18, os pesquisadores do Proto 6G entendiam que deveria haver componentes para troca de eventos, repositório de métricas e coletores específicos.

9.2 Experimentação com Análise de Desempenho de Rede

A Tabela 8, apresentada no Relatório Técnico das Atividades 5.1 e 5.2 do Projeto Brasil6G [3], mostra que no momento os principais projetos de código aberto com as funções de um CORE 5G não possuem a função NWDAF disponível. Assim, a primeira atividade relacionada com o desenvolvimento dos mecanismos relacionados à AI no núcleo Proto 6G envolve o desenvolvimento de uma função NWDAF. Considerando a premissa da evolução do Core 5G rumo ao Core 6G, o ponto de partida é a especificação disponível da NWDAF pelo 3GPP.

A NWDAF é parte da arquitetura do 5G [13]. A especificação TS 23.501 Release 16 [13] descreve a função NWDAF na Seção 6.2.18. Apresenta as diversas interações entre a NWDAF e outras funções do núcleo, descrevendo fatores que devem ser considerados para descoberta e seleção de funções na Seção 6.3.13. Apresenta também os serviços básicos oferecidos pela NWDAF na Seção 7.2.12 de especificação técnica do 3GPP.

No geral, os serviços oferecidos pela NWDAF permitem que diferentes NF do núcleo possam subscrever, através do envio da primitiva (*nwdaf_AnalyticsSubscription*), em diferentes tipos de análises resultantes da NWDAF e também requisitar estes serviços a qualquer momento através do envio da primitiva (*Nnwdaf_AnalyticsInfo*).

De forma complementar, a especificação técnica TS 23.588 [11] descreve os procedimentos para suportar as funções de análise de dados suportados pela NWDAF. A Figura 40 descreve o procedimento para que uma NF possa subscrever na NWDAF afim de obter informações sobre o desempenho da rede. A NWDAF interage com duas outras funções de rede, neste caso a AMF e a NRF. Neste procedimento é previsto ainda a interação da NWDAF com sistemas de *Network Operations, Administration and Maintenance* (OAM).

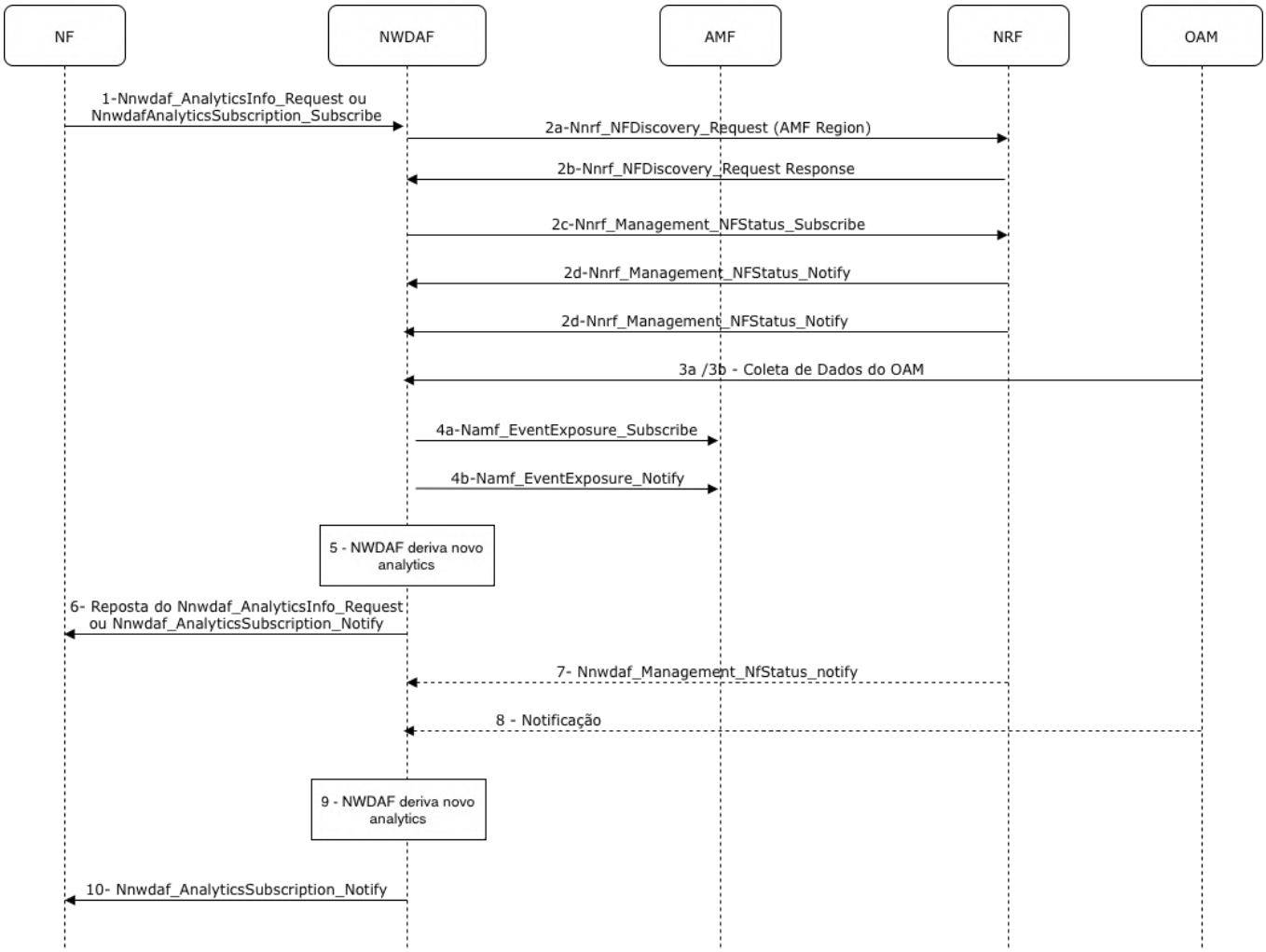


Figura 40: Procedimento para Subscrição na Análise de Performance da Rede [11].

Para exemplificar o procedimento supramencionado, desenvolveu-se um protótipo no Proto 6G com o intuito de experimentar o procedimento de análise de desempenho de rede na NWDAF. Este procedimento está descrito na Seção 6.6 (*Network Performance Analytics*) da especificação técnica TS 23.288 [11] *release* 16. Neste relatório, apresenta-se as alterações realizadas no projeto Free5GC, bem como o passo a passo do experimento realizado.

9.2.1 Intervenções no Free5GC

De acordo com a TS 23.288 [11], o procedimento de Análise de Desempenho de Rede realizado pela NWDAF envolve: qualquer NF que precise de informações dessa análise, a NWDAF, a AMF, a NRF e o OAM. A NF requisitante envia uma das seguintes mensagens para a NWDAF: *Nnwdaf_AnalyticsInfo_Request* ou *Nnwdaf_AnalyticsSubscription_Subscribe* para comunicação síncrona ou assíncrona, respectivamente. A NWDAF requisita informações para a AMF e a NRF apenas no modelo assíncrono.

A NWDAF envia a mensagem *Nnrf_Management_NfStatus_Subscribe* em direção à NRF e, eventualmente, a NRF envia as informações de *status* para a NWDAF através da mensagem *Nnrf_Management_NfStatus_Notify*. Similarmente, a NWDAF envia a mensagem *Namf_EventExposure_Subscribe* em direção à AMF e aguarda um eventual retorno através

da mensagem *Namf_EventExposure_Notify*.

Apesar do projeto Free5GC possuir as NFs necessárias para execução deste experimento, exceto a NWDAF, as mensagens de notificação previstas na TS 23.288 [11] não foram implementadas no Free5GC. Portanto, nenhum experimento com a NWDAF pode ser realizado com o Free5GC da distribuição oficial, dado que a NWDAF requisita notificações síncronas. Para tratar este contratempo, implementou-se algumas modificações no código fonte da função AMF no Free5GC, para enviar notificações do tipo REGISTRATION_STATE_REPORT.

```
func HandleRegistrationRequest
(ue *context.AmfUe, anType models.AccessType, procedureCode int64,
registrationRequest *nasMessage.RegistrationRequest,
) error {

...

notify.SendAMFEventNotification(
 ue, models.AmfEventType_REGISTRATION_STATE_REPORT)
return nil
}
```

O código acima mostra a função *HandleRegistrationRequest* da AMF. O código original está sendo abstraído pelas reticências (...). O código da chamada *notify.SendAMFEventNotification* foi a intervenção necessária no Free5GC. Ao realizar o registro de qualquer UE, a AMF passa agora a enviar uma notificação do tipo REGISTRATION_STATE_REPORT. O código fonte da função *SendAMFEventNotification* foi desenvolvido em uma nova classe na AMF, no novo arquivo `internal/sbi/producer/notify/event_exposure.go`. Este arquivo não está sendo apresentado neste relatório e está versionado em https://github.com/open6gc/amf. Basicamente, a *notify.SendAMFEventNotification* busca o ID da subscrição realizada pela NWDAF na memória da AMF, concatena com informações do UE que está sendo registrado na AMF e envia as informações do registro para a NWDAF através de uma requisição POST na URL de *callback* fornecida pela NWDAF no momento da subscrição.

9.2.2 Apresentação do Experimento

A NWDAF foi desenvolvida à luz dos conceitos descritos na Seção 9.1. Este desenvolvimento trata-se de um protótipo que implementa as *Application Programming Interfaces* (APIs) especificadas no *openapi* arquivos `TS29510_Nnrf_Management.yaml` e `TS29510_Nnrf_NFDiscovery.yaml`. Estas APIs permitem a exposição da mensagem *Nnwdaf_AnalyticsInfo_Request* para que qualquer NF possa requisitar.

O experimento consistiu em executar todas as NFs disponíveis no Free5GC através do utilitário `run.sh` disponibilizado pelo mesmo, bem como executar o *webconsole* para cadastro dos IMSIs simulados. Após a execução das NFs, utilizou-se o projeto my5G-RANTester para registrar UEs no núcleo Free5GC. O intuito deste experimento é demonstrar a interação entre NWDAF e AMF através das primitivas *Namf_EventExposure_Subscribe* e *Namf_EventExposure_Notify*.

A Figura 41 apresenta a configuração de UE e **GNB!** (**GNB!**) utilizadas no my5G-RANTester. Basicamente, um UE é registrado na AMF conforme funcionamento padrão do Free5GC. Previamente, uma NF requisitou à NWDAF subscrição em todas as mensagens de registro.

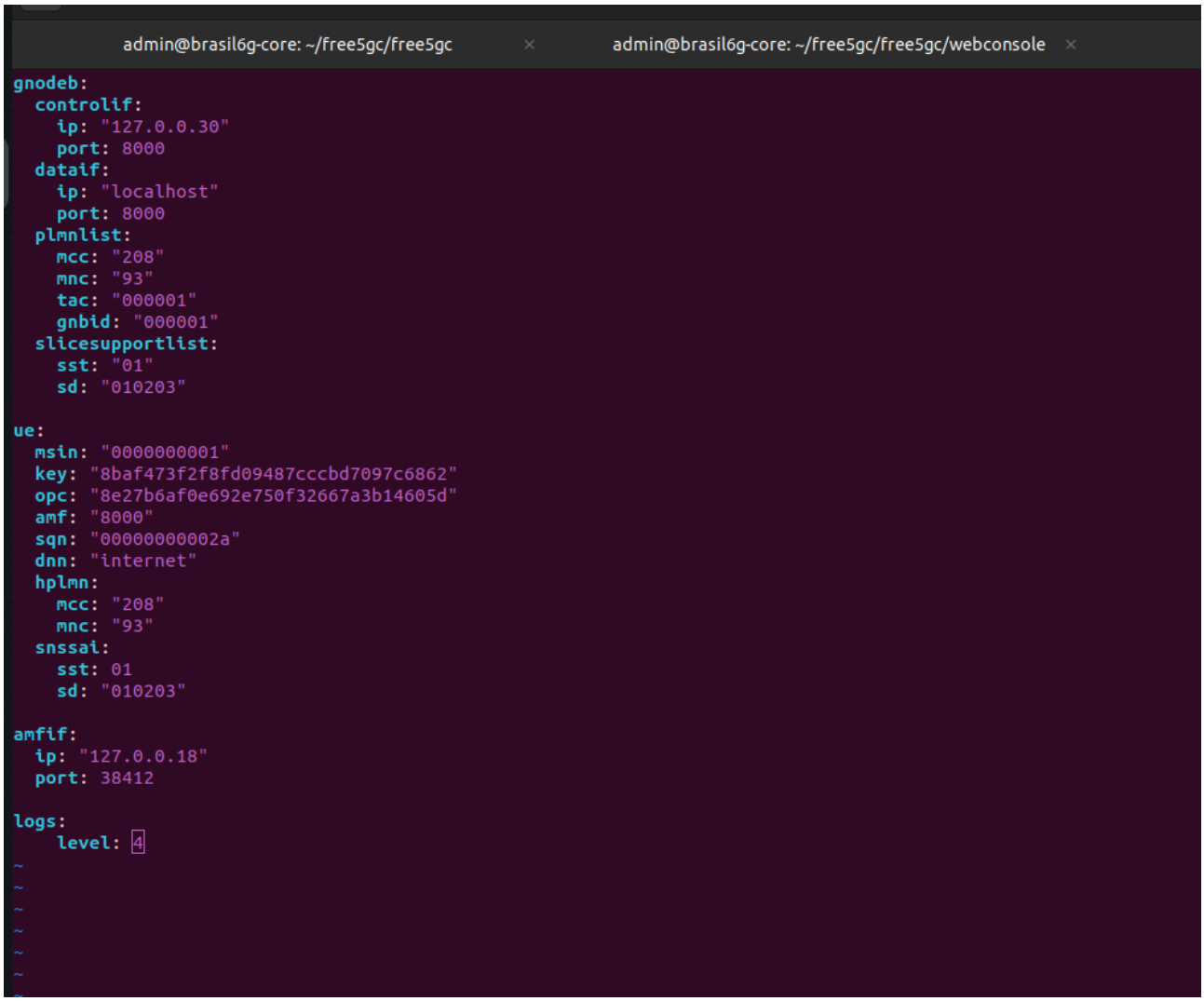


Figura 41: Arquivo de configuração utilizado no my5G-RANTester para o experimento.

A Figura 42 apresenta a saída da execução do my5G-RANTester. Observa-se que o UE é registrado com sucesso após autenticação. Este registro é básico e já foi demonstrado em relatórios anteriores do Brasil 6G.

Por fim, a Figura 43 mostra a saída da execução das NFs do Free5GC. Em destaque há a intervenção realizada no experimento aqui relatado. Após receber uma requisição de registro, a AMF passa a interagir com a NWDAF. Inicialmente, a AMF pega o código da subscrição da NWDAF, verifica para quais tipos de eventos há subscrição e envia as informações de registro para a AMF através do *endpoint* `http://localhost:8080/notification/amf` exposto pela NWDAF.

O experimento basicamente demonstra a nova integração entre AMF e NWDAF. Todavia, a partir dessa integração é possível conduzir experimentos mais elaborados visto que a mensagem de notificação foi implementada no Free5GC para a AMF. Ainda, dado que as NFs do Free5GC seguem um modelo de implementação padronizado, o esforço para inserir mensagens de notificação em outras NFs é mínimo. Com essas notificações, quaisquer funções de análise da NWDAF podem ser implementadas, não somente a análise de registro de UEs como demonstrado nesta Ação 8 desse relatório.

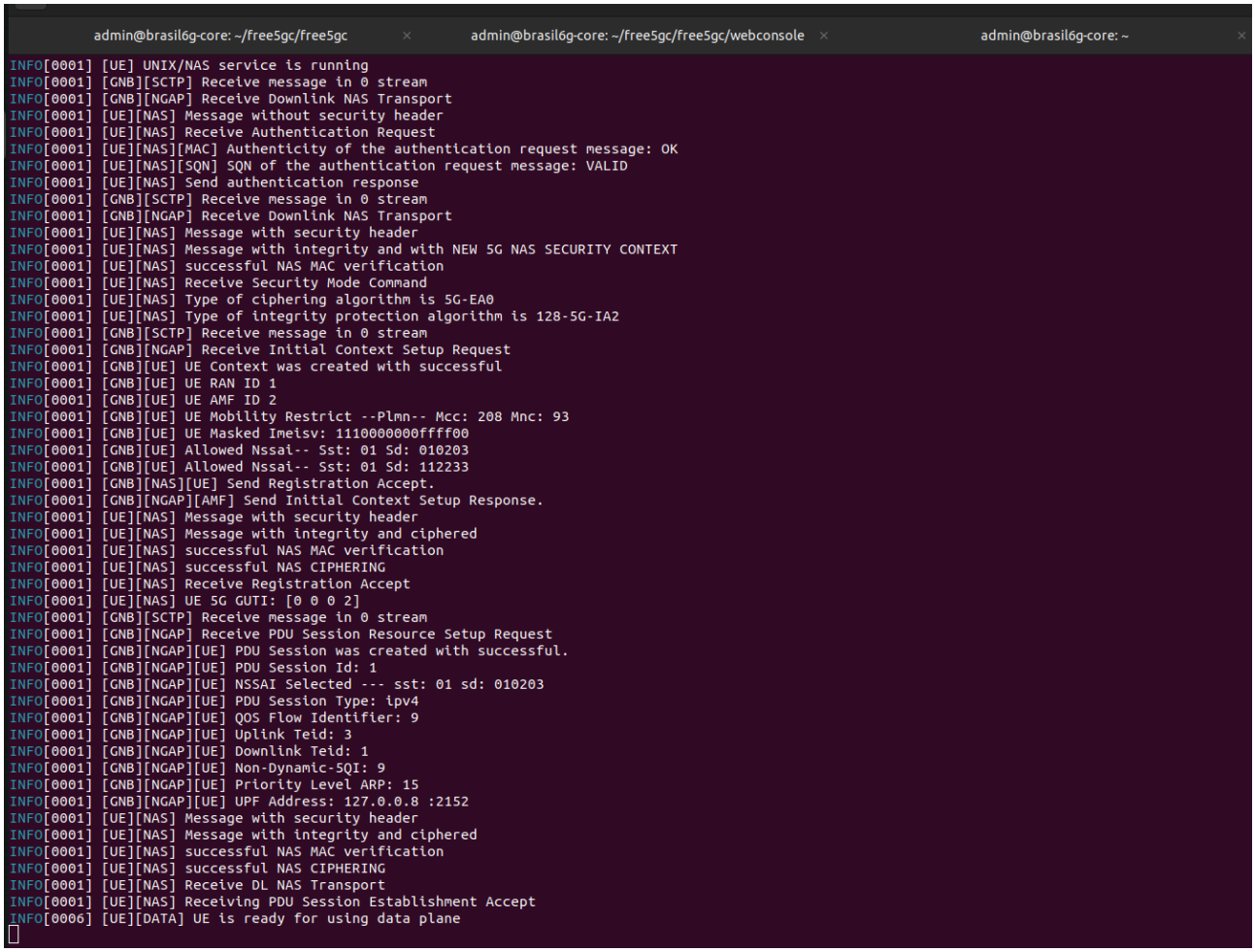


Figura 42: Saída da execução do my5G-RANTester.

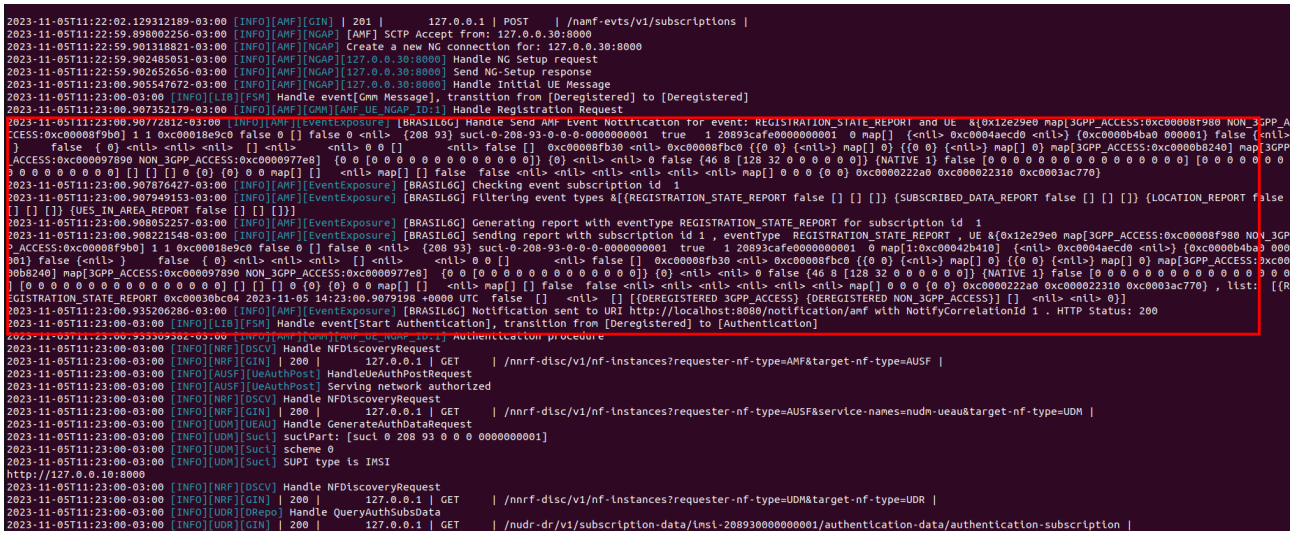


Figura 43: Saída de execução da AMF destacando a integração com a NWDAF desenvolvida para o Proto 6G.

10 Considerações Finais

Este relatório apresenta o resultado da Atividade 5.3 realizada durante o ano de 2023/2024. No inicio de 2022, as Atividades 5.1, 5.2 e 5.3 da META 5 foram dividas em 11 ações. As Ações de 1-9 foram executadas em paralelo, e como tal, tiveram seus resultados reportados em 2022 e agora em 2023. Em 2023, essas ações continuaram sendo executadas e aprimoradas, juntando-se as Ações 10 e 11, que abraçam a Atividade 5.3. Para facilitar o entendimento, a seguir apresentamos as principais considerações finais para cada Ação realizada:

- **Ação 1** - A consolidação da arquitetura do Proto 6G foi alcançada, incluindo o desenvolvimento e integração de novos componentes voltados para IA, acesso e fatiamento. Sabemos agora que dependendo do ambiente em uso para experimentação, limitações de pacotes e versionamento de software, mais de uma arquitetura é necessária. Logo, a arquitetura do Proto 6G se materializa de diferentes formas, com diferentes recursos dependendo do ambiente em uso. Por exemplo, a materialização com Kubernetes é diferente daquela que usa somente contêineres Docker. Pela mesma razão, a materialização no FIBRE da RNP é particular e limitada aos pacotes e softwares compatíveis com esse ambiente. O trabalho de migração de uma materialização para outra é constante e demandante. Ressalta-se ainda a existência de incompatibilidades entre os pacotes *open source* adotados no Proto 6G e o padrão 3GPP. Tais incompatibilidades causam transtorno e retrabalho em algumas materializações.

- **Ação 2** - Diversas atividades ao longo do ano 2023 contribuíram com o objetivo de especificar e implementar um núcleo de rede 6G conforme a visão do projeto Brasil 6G, o chamado *Proto 6G*. Estas ações estão distribuídas ao longo deste documento onde são apresentadas as especificações e implementações em aspectos como acesso multi RAN, englobando a Ação 3 e a Ação 7, ainda aspectos de slicing na Ação 6 e finalmente sobre o uso de AI no Proto 6G. Concretamente foram realizadas a implementação de novas funções como o NWDAF integrado com o core do Free5GC e ainda a cloudificação das funcões do Free5GC para suporte nativo de implantação em ambientes de nuvem ampliando as possibilidades futuras de experimentação e implantação para os componentes do *Proto 6G*.

- **Ação 3** - A Ação 3 do projeto Brasil 6G, focada no acesso do Proto 6G, integra múltiplas redes de acesso 5G com avanços rumo ao Proto6G. Nesse relatório descrevemos a configuração e operação do UE-non3GPP, um artefato de software que conecta dispositivos não-3GPP ao núcleo da rede através da função N3IWF, garantindo uma integração segura e eficiente. Detalhamos os pré-requisitos, instalação, configuração, e experimentos práticos para validar a conectividade e funcionamento deste acesso Proto6G, destacando também as dificuldades enfrentadas, como a compatibilidade com Kubernetes, e as ações futuras para solucionar esses desafios.

- **Ação 4** - Diversos ambientes foram preparados e disponibilizados para materializar a arquitetura Proto 6G, incluindo ambientes com e sem contêineres Docker, com e sem Kubernetes, dentro e fora do FIBRE da RNP. O grupo evolui para executar cenários práticos onde era possível, com as plataformas compatíveis, migrando sempre que possível para as materializações mais sofisticadas do Proto 6G. O aprendizado com a plataforma da META 2 foi enorme.

- **Ações 5 e 10** - Em relação às Ações 5 e 10, foi desenvolvida uma infraestrutura de rede que consiste em: (i) transceptor UE-T6G; (ii) transceptor BS-T6G juntamente com componente N3IWF do núcleo; e (iii) Servidor Brasil 6G executando as demais funções do núcleo da rede. Para a realização dos testes, a UE foi registrada no núcleo com o auxílio da função N3IWF executada no computador do transceptor BS-T6G, dado que o acesso é feito através de uma rede *Untrusted Non-3GPP*. Além do registro e autenticação, foi criada uma UPF capaz de encaminhar os dados da UE *Non-3GPP* para a rede de dados da Internet através de um tunelamento seguro e confiável. Por fim, foram realizados testes através de comandos de *ping* e *iperf3* para monitorar a latência e vazão da rede. Através dos resultados obtidos, pôde-se observar que o ambiente criado de forma integrada em laboratório é capaz de suportar as aplicações demandadas para o Projeto Brasil 6G mediante limitações de vazão e latência mínimas da rede. Como trabalhos futuros, espera-se testar as aplicações dentro da infraestrutura de rede desenvolvida, através das seguintes variantes: (i) testes das aplicações de forma isolada; (ii) testes com aplicações sendo executadas simultaneamente em uma mesma UE; (iii) testes com aplicações sendo executadas em diferentes UEs, sendo elas registradas no núcleo da rede e funcionais ao mesmo tempo; e (iv) testes com diferentes UEs sendo atendidas com diferentes UPFs, como por exemplo, Internet, rede privada, e rede satelital.

- **Ação 6** - A Ação 6 cobre a execução do fatiamento fim-a-fim de recursos físicos e cibernéticos utilizando a plataforma NASP (Network Slice as a Service Platform) em um ambiente de computação em nuvem. A instalação e configuração da NASP envolvem a criação de uma infraestrutura baseada em Kubernetes, integrada ao núcleo SBA 3GPP com a função N3IWF, permitindo a validação das funcionalidades de fatiamento de rede 6G. Foram realizados testes de disponibilidade, conectividade e desempenho, utilizando ferramentas como Grafana e Linkerd para monitoramento e observabilidade, com resultados que indicam a eficácia do sistema em fornecer serviços de fatiamento de rede de forma automatizada e escalável.

- **Ação 7** - A Ação 7 do relatório técnico aborda a integração de redes satelitais ao projeto Brasil 6G, destacando o desenvolvimento e teste da função N3IWF para acesso do núcleo do Proto6G à Internet via satélite. O relatório descreve a adaptação do Caminho 1, utilizando um satélite para conectar o núcleo Free5GC à Internet, com infraestrutura emuladora e servidores específicos no ICT Lab e no *hub* satelital de Brasília/DF. Um teste de *ping* bem-sucedido foi realizado, e novos testes com pacotes CDN e um switch OpenFlow estão previstos. O SGDC, satélite utilizado, é detalhado como parte fundamental da infraestrutura, com destaque para sua cobertura e especificações técnicas.

- **Ação 8** - Nesta ação o trabalho foi desenvolvido em três eixos. O primeiro eixo levou em conta a especificação técnica da função chamada eNWDAF para o núcleo do *Proto 6G*. Esta função está alinhada com a premissa da evolução funções e mecanismos existentes no núcleo do core 5G para suportar os requisitos e casos de uso relativos ao uso de AI pelo 6G. É importante destacar que esta proposição de especificação técnica levou em conta a especificação técnica funcional proposta pelo 3GPP no *Release 16* nas a 3GPP TS 29.520 version 16.6.0 [10] e TS 23.288 version 16.6.0 Release 16 [10] e também suas evoluções propostas ao longo de 2023, incluindo até a TS 23.288 version 18.4.0 [12]. Um aspecto importante aqui é que houve uma convergência entre a visão de evolução dentro do projeto com o proposto pelo 3GPP no final de 2023 para a NWDAF no Release 18

[12]. A especificação técnica da eNWDAF abre caminho para a implementação e evolução dos componentes de AI do *Proto 6G* como o próximo passo do trabalho. O segundo eixo envolveu intervenções no código do Free5GC para desenvolver uma NWDAF. Além de desenvolver o código da NWDAF levando em conta a especificação TS 23.288 version 16.6.0 [10], a implementação também modificou o código de funções já existentes no código do Free5GC, como AMF, para suportar as mensagens de requisição e reposta e também de subscrição e notificação preconizadas pela especificação para a comunicação entre as funções do núcleo e a NWDAF. A existência da NWDAF no Free5GC é uma condição para a sua evolução, ou seja, implementar a eNWDAF. Além disso, como um resultado concreto, há agora uma NWDAF disponível para experimentação na sequência do projeto. Finalmente, no terceiro eixo foi realizada uma avaliação experimental do uso da NWDAF criada para realizar uma análise do desempenho da rede, conforme cenário descrito na TS 23.288 version 16.6.0 Release 16 [10].

- **Ação 9** - Essa ação foi descontinuada devido a falta de participação do parceiro do projeto.

Referências

[1] X. Lin, "An Overview of 5G Advanced Evolution in 3GPP Release 18," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 77–83, Sep. 2022, conference Name: IEEE Communications Standards Magazine. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9927255

[2] M. A. Garcia-Martin, M. Gramaglia, and P. Serrano, "Network Automation and Data Analytics in 3GPP 5G Systems," *IEEE Network*, pp. 1–1, 2023, conference Name: IEEE Network. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10273383/authors#authors

[3] Centro de Referência em Rádiocomunicação (CRR) - Inatel, "Relatório Técnico das Atividades 5.1 e 5.2 - Projeto e Seleção de Componentes, Plataformas, Ferramentas e Especificação," *Projeto Brasil 6G*, Dezembro 2022.

[4] "UE-non3GPP," 2023. [Online]. Available: https://github.com/LABORA-INF-UFG/UE-non3GPP

[5] free5GC, "free5GC," 2022. [Online]. Available: https://www.free5gc.org/

[6] ETSI, "Telecommunication management; Charging management; 5G system, charging service," European Telecommunications Standards Institute, Technical Specification (TS) 132.291, 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/132200_132299/132291/17.07.00_60/ts_132291v170700p.pdf

[7] kubernetes, "kubernetes," 2023. [Online]. Available: https://kubernetes.io/

[8] L. system programming training, "ip-xfrm," 2023. [Online]. Available: https://man7.org/linux/man-pages/man8/ip-xfrm.8.html

[9] G. Z. Bruno *et al.*, "Anomaly Detection in Cloud-native B5G Systems using Observability and Machine Learning COTS Solutions," *Journal of Internet Services and Applications (JISA)*, pp. 1–11, 2023.

[10] 3GPP, "5G; 5G System; Network Data Analytics Services; Stage 3 (3GPP TS 29.520 version 16.6.0 Release 16)," ETSI, Sophia Antipolis, Technical Specification TS 23.288, Jan. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/129500_129599/129520/16.06.00_60/ts_129520v160600p.pdf

[11] ——, "5G; Architecture enhancements for 5G System (5GS) to support network data analytics services (3GPP TS 23.288 version 16.6.0 Release 16)," ETSI, Sophia Antipolis, Technical Specification TS 23.288, Jan. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf

[12] ——, "5G; Architecture enhancements for 5G System (5GS) to support network data analytics services (3GPP TS 23.288 version 18.4.0 Release 18)," 3GPP, Sophia Antipolis, Technical Specification TS 23.288, Dec. 2023. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579

[13] ——, “5G; System architecture for the 5G System (5GS) (3GPP TS 23.501 version 16.6.0 Release 16),” ETSI, Sophia Antipolis, Technical Specification TS 23.501, Oct. 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf