

Inatel

Instituto Nacional de Telecomunicações

CONTRIBUIÇÕES NO DESENVOLVIMENTO
DE SERVIÇOS DE REPRESENTAÇÃO,
CONTROLE E ALOCAÇÃO DINÂMICA PARA
A NOVAGENESIS

Lúcio Henrique de Oliveira

Julho de 2015

**Contribuições no desenvolvimento de serviços de representação,
controle e alocação dinâmica para a NovaGenesis**

Lúcio Henrique de Oliveira

Dissertação apresentada ao Instituto Nacional de
Telecomunicações, como parte dos requisitos para
obtenção do Título de Mestre em Telecomunicações

Orientador: Prof. Dr. Antônio Marcos Alberti

Santa Rita do Sapucaí

2015

Oliveira, Lúcio Henrique de
O482c Contribuições no desenvolvimento de serviços de representação, controle e alocação dinâmica para a NovaGenesis. / Lúcio Henrique de Oliveira. – Santa Rita do Sapucaí, 2015. 65p.

Orientador: Prof. Dr. Antônio Marcos Alberti.
Dissertação de Mestrado – Engenharia de Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.
Inclui bibliografia.

1. Internet do futuro 2. Nova Genesis 3. Redes definidas por software 4. Sensoriamento espectral 5. Alocação dinâmica 6. Engenharia de Telecomunicações. I. Alberti, Antônio Marcos II. Instituto Nacional de Telecomunicações – INATEL. III. Título.

CDU 621.39

FOLHA DE APROVAÇÃO

Dissertação defendida e aprovada em ___/___/____, pela comissão julgadora:

Prof. Dr. Antônio Marcos Alberti
INATEL

Prof. Dr. Arismar Cerqueira Sodré Junior
INATEL

Prof. Dr. Rodrigo da Rosa Righi
Unisinos

Prof. Dr. José Marcos Câmara Brito
Coordenador do Curso de Mestrado

*“A mente que se abre a uma nova ideia jamais
voltará ao seu tamanho original.”*

-- Albert Einstein

Agradecimentos

Agradeço a Deus por ter me abençoado a cada instante de minha existência, à toda minha família, especialmente a minha companheira e amiga Nina e aos combustíveis da minha vida, Valentina e Miguel.

Agradeço também aos amigos e colegas e ao Prof. Dr. Antônio Marcos Alberti por orientar e partilhar de seu sonho conosco. À todos os professores do mestrado e à Gisele pela atenção, dedicação e carinho comigo durante todos os anos.

Ao Inatel, ICT Lab, ao WOCA e a todos aqueles que me apoiaram incondicionalmente, o meu muito obrigado.

Índice

Lista de Figuras.....	viii
Lista de Quadros	ix
Lista de Abreviaturas e Siglas.....	x
Publicações	xii
Resumo	xiii
Abstract.....	xiv
Capítulo 1. Introdução	15
Capítulo 2. Tecnologias e conceitos	21
2.1 Internet do Futuro	21
2.2 Ethernet e Wi-Fi.....	23
2.3 Redes Definidas por Software	24
2.4 Transporte de mensagens.....	25
Capítulo 3. NovaGenesis	26
Capítulo 4. Desenvolvimento de novos serviços para NovaGenesis.....	31
4.1 <i>Spectrum Sensing Agent (SSA)</i>	31
4.2 <i>Resource Management Agent (RMA)</i>	34
4.3 <i>Access Point Agent (APA)</i>	38
4.4 <i>Python OpenFlow Controller Agent (POXA)</i>	41
Capítulo 5. Implementação dos 4 novos serviços NovaGenesis.....	44
5.1 <i>Spectrum Sensing Agent (SSA)</i>	44
5.2 <i>Resource Management Agent (RMA)</i>	48
5.3 <i>Access Point Agent (APA)</i>	51
5.4 <i>Python OpenFlow Controller Agent (POXA)</i>	52
5.5 <i>Hash Table Service (HTS)</i> do domínio	56
Capítulo 6. Conclusões	58
Referências.....	61

Lista de Figuras

Figura 1 - Previsão de crescimento de tráfego [2].....	15
Figura 2 - Cenário de inovação no NovaGenesis para uma rede óptica sem-fio com sensoramento espectral e alocação dinâmica de recursos [4].....	17
Figura 3 - Contribuições dos quatro serviços NovaGenesis desenvolvidos: RMA, APA, SSA e POXA.....	20
Figura 4 - Arquitetura NovaGenesis [4]	26
Figura 5 - Ecossistema NovaGenesis formado por blocos [4].....	30
Figura 6 - Diagrama de sequência de funcionamento do SSA.	34
Figura 7 - Diagrama de sequência de funcionamento do RMA.....	37
Figura 8 - Diagrama de sequência de funcionamento do APA.....	39
Figura 9 - Interdependência entre SSA, RMA e APA.....	40
Figura 10 - Diagrama de sequência de funcionamento do POXA.....	42
Figura 11 - Cenário de implementação da arquitetura NG proposta em [6].....	60

Lista de Quadros

Quadro 1 - Inicialização do SSA	44
Quadro 2 - Descoberta do RMA pelo SSA	45
Quadro 3 - SSA consultando via ZeroMQ.....	45
Quadro 4 - Script Python enviando dados	45
Quadro 5 - SSA recebendo dados via ZeroMQ	46
Quadro 6 - Mensagem do SSA para o RMA	46
Quadro 7 - Recebimento das informações do SSA pelo RMA.....	47
Quadro 8 - Inicialização do RMA.....	48
Quadro 9 - Publicação para o RMA.....	48
Quadro 10 - Mensagem enviada para agente específico.....	49
Quadro 11 - RMA enviando mensagem para AAA.....	50
Quadro 12 - Inicialização do APA.....	51
Quadro 13 - Processamento da mensagem recebida pelo APA.....	52
Quadro 14 - Inicialização do POXA.....	53
Quadro 15 - RMA enviando mensagem para POXA específico	54
Quadro 16 - Tratamento da mensagem recebida pelo POXA.....	55
Quadro 17 - Verificação da não existência de fluxos no OpenFlow	55
Quadro 18 - Registro de criação do fluxo pelo OpenFlow	55
Quadro 19 - Verificação do fluxo instalado no OpenFlow.....	56
Quadro 20 - Extrato da HTS do domínio.....	57

Lista de Abreviaturas e Siglas

AAA	<i>Active Antenna Agent</i>
AP	<i>Access point</i>
APA	<i>Access Point Agent</i>
ARPANET	<i>Advanced Research Projects Agency Network</i>
CERN	<i>Conseil Européen pour la Recherche Nucléaire</i>
CIA	<i>Convergent Information Architecture</i>
CLI	<i>Command Line Interface</i>
FDDI	<i>Fiber Distributed Data Interface</i>
GIRS	<i>Generic Indirection Resolution Service</i>
GW	<i>Gateway</i>
HT	<i>Hash Table</i>
HTS	<i>Hash Table Service</i>
IaaS	<i>Infrastructure-as-a-Service</i>
IBM SNA	<i>IBM Systems Network Architecture</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
LAN	<i>Local Area Network</i>
MAC	<i>Media Access Control</i>
NAT	<i>Network Address Translation</i>
NB	<i>Name Binding</i>
NG	<i>NovaGenesis</i>
OF	<i>OpenFlow</i>
OSI	<i>Open Systems Interconnection</i>
PG	<i>Proxy-Gateway</i>
PGS	<i>Proxy-Gateway Service</i>
POXA	<i>Python OpenFlow Controller Agent</i>
PSS	<i>Publish/Subscribe Service</i>
RMA	<i>Resource Management Agent</i>
SAIL	<i>Scalable and Adaptive Internet Solutions</i>
SCN	<i>Self-Certifying Name</i>

SDN	<i>Software Defined Networking</i>
SOA	<i>Service Oriented Architecture</i>
SSA	<i>Spectrum Sensing Agent</i>
SSH	<i>Secure Shell</i>
TCP	<i>Transmission Control Protocol</i>
ULN	<i>Unique Legible Name</i>
USB	<i>Universal Serial Bus</i>
WOCA	<i>Wireless and Optical Convergent Access</i>
XIA	<i>eXpressive Internet Architecture</i>

Publicações

ALBERTI, A.M.; FERNANDES, V.H.O.; CASAROLI, M.A.F.; OLIVEIRA, L.H.; PEDROSO, F.M.; SINGH, D.. A NovaGenesis proxy/gateway/controller for OpenFlow software defined networks. In: **Network and Service Management (CNSM), 2014 10th International Conference on**. IEEE, 2014. p. 394-399. Rio de Janeiro, Brasil.

ALBERTI, A.M.; CASAROLI, M.A.F.; CERQUEIRA, S.A.; COSTA, I.F.; OLIVEIRA, L.H.; RAIMUNDO-NETO, E.. Service-oriented, Name-based, and Software-defined Spectrum Sensing and Dynamic Resource Allocation for Wi-Fi Networks using NovaGenesis. In: **VI International Workshop on Telecommunications**. INATEL, 2015. Santa Rita do Sapucaí, Brasil.

ALBERTI, A.M.; CASAROLI, M.A.F.; OLIVEIRA, L.H.. NovaGenesis: Convergent Information Architecture. In: **VI Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF2015)**. XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. 2015. Vitória, Brasil.

Resumo

Com o aumento da capacidade de processamento computacional, crescimento no número de dispositivos conectados e os avanços das tecnologias de comunicação, a Internet se tornou protagonista de uma era onde tudo gira em torno da informação. A informação é o centro de nossa era, a Era Digital. Porém, com o aumento das ofertas de serviços e tecnologias disponíveis, a arquitetura atual da Internet, desenvolvida a mais de quarenta anos tem mostrado sinais de sua idade. Com isso, problemas que antes eram impensáveis à época de sua criação começaram a emergir.

Este trabalho faz o uso de uma nova arquitetura de rede idealizada para criar serviços que resolvem problemas de alocação dinâmica de recursos de substrato físico - equipamentos, hardware e elementos de rede - representação e controle de dispositivos utilizando a NovaGenesis. A NovaGenesis é uma proposta de Internet do Futuro iniciada em 2008 no Inatel que propõe uma nova arquitetura de Internet “do zero”, baseada principalmente em nomeação e resolução de nomes, auto-organização de serviços, redes programáveis e serviços representantes.

Foram desenvolvidos quatro novos serviços para a NovaGenesis, para lidar com os problemas apresentados que representam dispositivos do mundo real, como um analisador de espectro e um *access point*, bem como elementos de redes definidas por *software* que podem ser controlados e gerenciados pela NovaGenesis a partir de um serviço de tomada de decisões - um gerente de alocação dinâmica de recursos. São os serviços de sensoriamento espectral, que faz a captura de informações de um analisador de espectro e envia para a NovaGenesis, de gerenciamento de recursos que toma decisões e dispara, através da NovaGenesis, comandos de operações em *access points* e redes geridas por software.

Uma prova de conceito é apresentada, bem como a relação com a Internet das Coisas e uma proposta de implementação desses serviços em um cenário real em conjunto com o Laboratório WOCA (*Wireless and Optical Convergent Access*) do Inatel.

Palavras-chave: Internet do Futuro, NovaGenesis, Redes definidas por software, Sensoriamento espectral, Alocação dinâmica, Engenharia de Telecomunicações.

Abstract

The continuous growth of computational processing capacity, the number of connected devices and advances in communications, the Internet has become a protagonist of an era, in which everything is concerning information. However, with the of available services and technologies, the current Internet architecture, developed more than 40 years ago, has been shown signs of its age. As a consequence, problems that were unthinkable at the beginning have been increasingly showing up in the last years.

This work makes use of a new network architecture to create services that solve problems of dynamic allocation of physical substrate resources - hardware and network elements - representing and controlling them using NovaGenesis architecture. The NovaGenesis is a future Internet project initiated in 2008 at Intel, which proposes a new Internet architecture from scratch, based primarily on naming, name resolution, self-organization, programmable networks and representation of physical world devices.

Four new NovaGenesis services to deal with dynamic resource allocation on wireless environment were developed and implemented in this work. They represent real-world devices like a spectrum analyzer or an Wi-Fi access point. Also, network elements can be defined by software – they are controlled and managed by NovaGenesis through a decision service that dynamically configure data flow forwarding using OpenFlow protocol. Are services of spectral sensing, which takes data from spectrum analyzer and sends it to NovaGenesis, resource management that make decisions and triggers operational commands to access points and software-defined networks.

A proof of concept of these services working together is presented, and the relationship with the Internet of Things is also explored. Moreover, it is presented a proposal of their implementation in real telecom networks in collaboration with the Wireless and Optical Convergent Access (WOCA) Laboratory.

Keywords: Future Internet, NovaGenesis, software defined networks, spectrum sensing, dynamic allocation, Telecommunications Engineering.

Capítulo 1. Introdução

Com o avanço da tecnologia e o aumento do uso das redes de telecomunicações, vislumbrar um futuro para a Internet passou a ser tema dos mais diversos simpósios, congressos, fóruns de discussão e encontros de pesquisadores mundo a fora. A estrutura e tecnologia que utilizamos hoje é antiga e desenvolvida quando não se tinha à disposição o potencial tecnológico disponível atualmente. Alguns conceitos se mostraram eficientes, enquanto outros se tornaram extremamente defasados e ineficientes perante o crescimento observado [1]. Dentre os problemas da arquitetura atual, podemos citar a escalabilidade, mobilidade, flexibilidade, confiabilidade e gerenciamento. A área de telecomunicações vem sofrendo mutações e implementações de novas tecnologias, equipamentos, conceitos e mudanças estruturais, porém tais evoluções devem acontecer a um passo mais acelerado para que a tecnologia acompanhe a necessidade dos usuários e esteja livre de adaptações em tecnologias ultrapassadas. Mesmo assim, nem sempre a evolução de tecnologias existentes consegue suprir as demandas que já existem e que despontam no futuro próximo, como a Internet das coisas e a crescimento do consumo de tráfego nos últimos anos, sendo esperado para 2019 um tráfego 24,3 *Exabytes* por mês [2].



Figura 1 - Previsão de crescimento de tráfego [2]

O fato é que um futuro sem provimento dos serviços de telecomunicações com qualidade é impensável e a sociedade consumidora já cobra isso das grandes corporações. As evoluções precisam acontecer neste cenário, portanto as soluções antigas já avançaram significativamente dentro do “espaço de desenho” disponível. Os desenvolvedores e

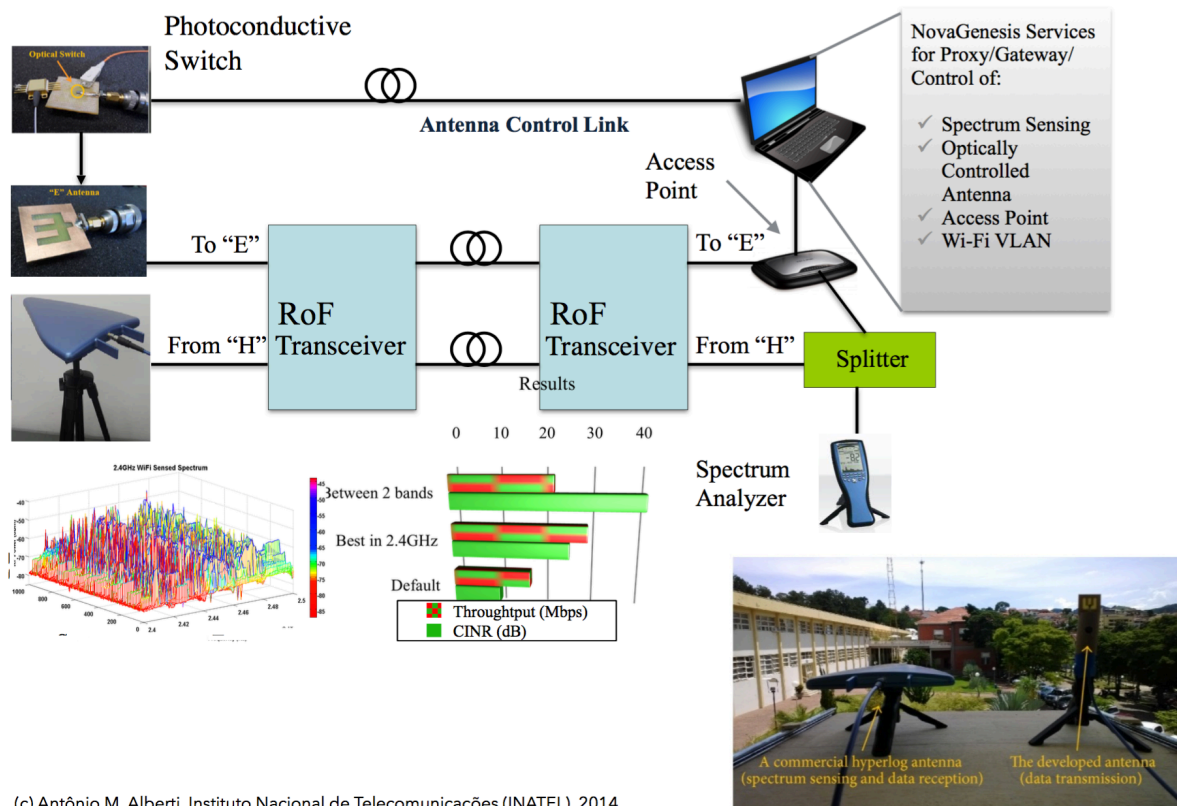
engenheiros estão encontrando dificuldades para manter a qualidade da arquitetura atual no provimento de serviços de telecomunicações e tecnologias para a Internet. Soluções disruptivas são necessárias para conciliar tecnologias de ponta, serviços de qualidade e viabilidade econômica e, um novo futuro já começou a ser desenhado e novas arquiteturas de informação surgiram para resolver estas demandas, aproveitando ou evoluindo tecnologias que deram certo e criando novas, mais eficientes. É nesse ponto que o cenário tecnológico atual permite que novas tecnologias ou ideias surjam a partir do zero [3] ou aproveitando o que há de melhor no ecossistema das redes de telecomunicações. Um novo espaço de desenho está sendo definido.

É neste contexto que essa dissertação se insere, explorando o uso de uma nova arquitetura de Internet [4] para desenvolver serviços que podem resolver problemas comuns encontrados na arquitetura atual. Esta arquitetura é chamada NovaGenesis (NG) e está em desenvolvimento no Inatel desde 2008.

Entre as dificuldades da arquitetura atual está a incapacidade de representar dispositivos do mundo real em aplicações, serviços e camadas de rede. Com cada vez mais equipamentos de todas as formas conectados à Internet, é latente a necessidade de permitir que qualquer dispositivo seja representado e localizado, integrando à Internet. O conceito de dispositivos sociais [5] emerge para colaborar com a ideia de que há necessidade de permitir que comportamentos sociais como troca de informação e conhecimento sejam transportados para a rede criando, portanto, comportamentos sociais baseados na Internet para dispositivos inteligentes.

Com o avanço das tecnologias para solucionar problemas atuais, novas demandas de desenvolvimento se tornam necessárias. Uma das possibilidades é utilizar dispositivos para auxiliar na oferta de serviços de qualidade, fazendo o uso de analisadores de espectro para realizar o sensoriamento espectral, melhorando a tomada de decisões sobre o canal ou tecnologia recomendados para utilização [6]. Com isso, devem existir componentes que auxiliem nas tomadas de decisões para alocar recursos de substrato e permitir que os serviços sejam oferecidos de acordo com o que se deseja. Em um cenário de inovação de uma rede óptica e sem-fio com sensoriamento espectral e alocação dinâmica de recursos como o desenvolvido no laboratório WOCA (*Wireless and Optical Convergent Access*) do Inatel, é necessário que as decisões e ações sejam tomadas por agentes inteligentes que tenham a capacidade computacional de processar a informação e se comunicar com os elementos da rede, sejam elas redes tradicionais ou redes definidas por software (*software defined networks*

ou SDN) [7]. É fundamental que as redes atuais e futuras permitam a gerência de seus recursos.



(c) Antônio M. Alberti, Instituto Nacional de Telecomunicações (INATEL), 2014.

Figura 2 - Cenário de inovação no NovaGenesis para uma rede óptica sem-fio com sensoriamento espectral e alocação dinâmica de recursos [4]

E são nessas áreas que o trabalho contribui, desenvolvendo agentes e serviços para representar socialmente na arquitetura NovaGenesis elementos de rede. Os elementos podem ser: um *access point* (AP), um controlador de antena, um analisador de espectro. Ainda há possibilidade de representar *softwares* e aplicativos dentro da arquitetura da NovaGenesis, como um controlador para redes definidas por *software* [7]. A ideia é permitir a orquestração do gerenciamento de recursos para melhoria de redes sem-fio com sensoriamento e alocação dinâmica de recursos, bem como um agente para controlar redes definidas por *software*. Contribui também na codificação e implementação da nova arquitetura de rede NovaGenesis. Por orquestração de recursos podemos entender como a união de diversos elementos e serviços de um sistema para, juntos, produzir resultados, permitindo que haja um serviço capaz de entender as necessidades e interoperar com os demais serviços produzindo o resultado desejado.

Nos dois cenários de contribuição, o requerimento fundamental é fazer com que toda a arquitetura seja baseada em uma arquitetura orientada a serviços (*service oriented*

architecture ou SOA) [8]. O SOA preza que as funcionalidades implementadas pelas aplicações sejam disponibilizadas como serviços, que podem ser facilmente interoperados, reutilizáveis e compartilhados entre as aplicações, da mesma forma que facilitam o processo de encontrar, definir e gerenciar os serviços ofertados. Para isso, na primeira contribuição, uma rede sem-fio tem a sua disposição um serviço que fornece o melhor canal Wi-Fi a ser utilizado. É um serviço que não está embarcado em *hardware* nem mesmo é obrigatório para todos os *access points*, mas sim um serviço que está à disposição dos elementos de rede para que se possa melhorar a forma como outros serviços (da rede) passam a ser ofertados.

A segunda abordagem utiliza a ideia de redes definidas por *software* [9], proposta criada para lidar com a complexidade das novas gerações de rede. A ideia central é “extrair a simplicidade” a partir do que os engenheiros de rede aprenderam durante décadas para lidar com a complexidade de desenhar e implantar redes de comunicação baseadas em *software* e não mais dependentes apenas do *hardware*, sendo este apenas um substrato para prover serviços. O OpenFlow (OF) [7] surgiu como a primeira concepção desta ideia e possivelmente é a iniciativa mais conhecida até o momento. Um serviço de convergência [10] foi desenvolvido para utilizar o melhor do OpenFlow em uma arquitetura também orientada a serviços.

Para que estas abordagens sejam possíveis, torna-se necessário que estes serviços sejam expostos para a rede e seus elementos. Esta exposição utiliza o conceito de exposição de nomes [11] por meio de “*name bindings*” (ligações entre nomes), tal estratégia permite relacionar vários nomes com vários outros nomes e objetos, representando os relacionamentos de tudo o que existe no mundo virtual ou físico. Desta maneira, cada serviço pode ser identificado por um nome único em um dado escopo. Expor os serviços com seus respectivos nomes dá ao ambiente a habilidade de descobri-los. Essa abordagem é conhecida como rede centrada em serviços ou em inglês *service-centric networking* [12]. Encontrar serviços baseados em seu nome e saber o que eles podem oferecer muda o cenário de como nós utilizamos a tecnologia a nosso favor.

Tendo descoberto esses serviços, um tipo de contrato deve ser estabelecido para manter, por exemplo, a qualidade do serviço e a integridade de todo o sistema. Um contrato deve possuir alguns requisitos e premissas entre os elementos de rede, e isso deve ser respeitado para garantir uma relação de confiança entre eles.

A perspectiva básica dos serviços definidos é fazer com que todos os elementos trabalhem bem em conjunto, criando uma rede segura, confiável e dinâmica, visando a melhor

maneira de utilizar os recursos disponíveis. Isto é fundamental para a Internet do futuro [13], ideia em ascensão nos últimos anos que prevê o estabelecimento de novas arquiteturas e conceitos que nortearão a Internet a ser utilizada pelas próximas gerações. Tal demanda deve-se ao fato de a arquitetura atual ter sido desenvolvida a quase 40 anos e não acompanha as evoluções das tecnologias atuais. O desafio é inovar utilizando novas abordagens, compatíveis ou “sobre” a arquitetura atual, mas fornecendo um novo *design*, melhorado e mais eficiente [14]. Utilizou-se a NovaGenesis, que é uma arquitetura híbrida, baseada e centrada em nomes, serviços, informações e *hosts*, definida por *software*, auto-organizável e *mobile friendly* que pretende ser uma nova arquitetura “*clean-slate*” para a nova geração convergente de redes e computação em nuvem.

O objetivo principal deste trabalho é demonstrar quatro novos serviços desenvolvidos para fazer com que uma rede de *access points* Wi-Fi possa utilizar o melhor canal para os clientes conectados e também controlar os recursos de uma rede definida por *software*, utilizando a NovaGenesis como arquitetura. São eles, o *Resource Management Agent* (RMA), responsável por definir qual recurso deve ser alocado a um determinado serviço de acordo com um algoritmo de decisão; o *Access Point Agent* (APA) que representa as habilidades e recursos de um *access point* Wi-Fi real, interagindo e controlando o dispositivo; o *Spectrum Sensing Agent* (SSA) que atua como um representante de um dispositivo de sensoriamento espectral na NovaGenesis; e o Python *OpenFlow Controller Agent* (POXA) que representa, controla e se comunica com um controlador OpenFlow. A dissertação também contribui no desenvolvimento de arquiteturas orientadas a serviço, redes definidas por *software* e baseada em serviços, e tem por objetivo secundário servir de referência para pesquisadores que queiram evoluir na integração desses paradigmas, principalmente utilizando a NovaGenesis como base.

Os quatro novos serviços desenvolvidos contribuem fundamentalmente em quatro áreas de interesse da NovaGenesis. Como pode se ver na Figura 1, o RMA auxilia na tomada de decisões para alocação dinâmica de recursos. O impacto da funcionalidade deste serviço na codificação da NovaGenesis também é significativo. O APA contribui para a codificação da NovaGenesis criando uma interface para comunicação e gerenciamento com um dispositivo externo utilizando um controlador para o OpenFlow e de uma biblioteca de comunicação entre *hosts*. O SSA fornece significativa contribuição no aspecto de representação de um dispositivo de sensoriamento espectral, além estabelecer padrão de desenvolvimento para este tipo de comportamento dentro do ecossistema da NovaGenesis. O POXA tem funcionalidade

similar ao APA, porém atua no controle de fluxos de tráfego de dados em uma rede legada, contribuindo para ampliar a comunicação de componentes NG sobre redes programáveis OpenFlow.

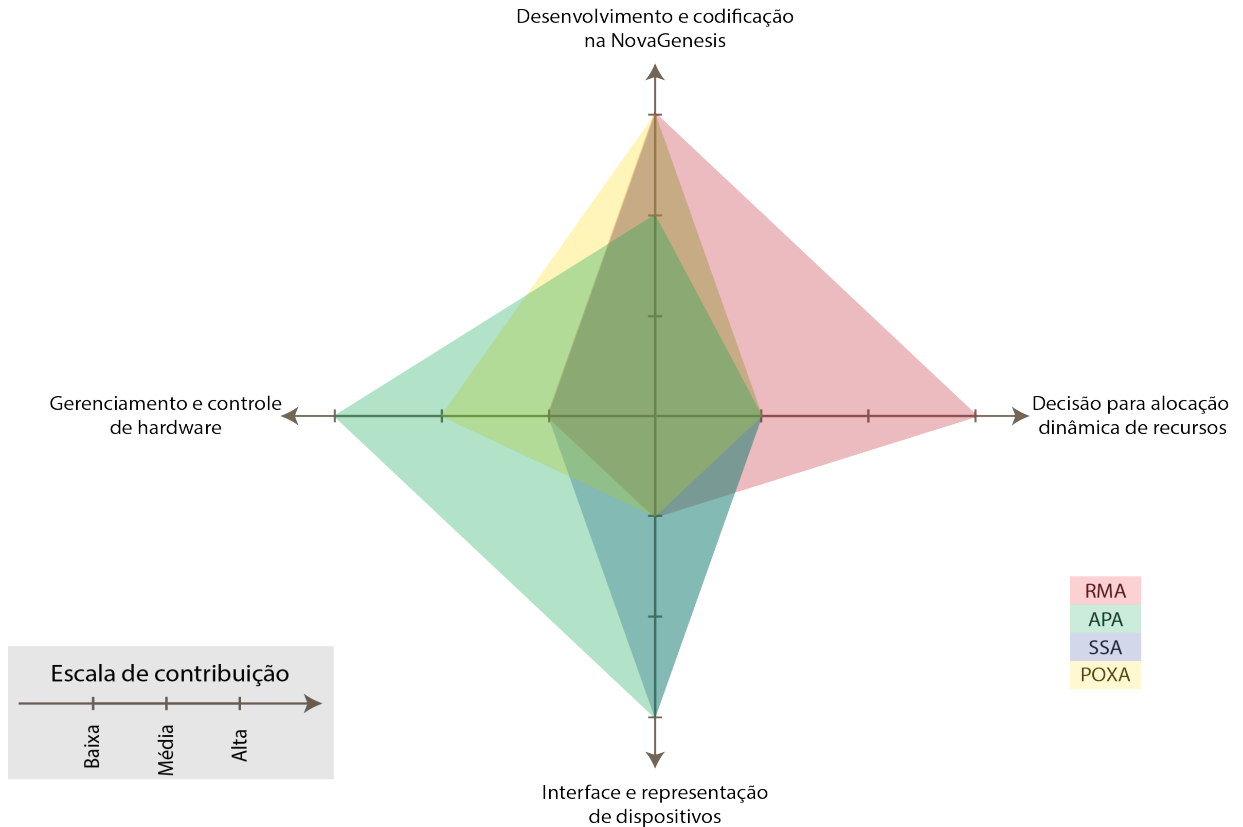


Figura 3 - Contribuições dos quatro serviços NovaGenesis desenvolvidos: RMA, APA, SSA e POXA.

O Capítulo 2 apresenta as tecnologias e conceitos utilizados ao longo da dissertação, os quais são importantes para contextualização, referência e aplicação dos serviços desenvolvidos. O Capítulo 3 apresenta as premissas básicas e os serviços já existentes na NovaGenesis, fundamentais para os novos serviços implementados e base teórica para o uso de alguns conceitos a serem explorados ao longo de todo o texto, como os conceitos de nome e nomeação. O Capítulo 4 apresenta os quatro novos serviços desenvolvidos para permitir o sensoriamento espectral, a alocação dinâmica de recursos e a configuração dos *access points* e do controlador OpenFlow. No Capítulo 5 são apresentados registros de mensagem e informações trocadas pela NG para demonstrar uma prova de conceito de que a implementação dos serviços é funcional na nova arquitetura. Por fim, o Capítulo 6 conclui a dissertação e fornece subsídios para os trabalhos futuros que podem ser desenvolvidos a partir dos serviços definidos.

Capítulo 2. Tecnologias e conceitos

Neste capítulo serão apresentadas algumas tecnologias e conceitos importantes para o entendimento dos serviços desenvolvidos, bem como determinar as tecnologias que foram utilizadas, apresentando embasamento teórico para a escolha realizada.

2.1 Internet do Futuro

É fato que a Internet modificou o cenário econômico mundial, principalmente depois do chamado “estouro da bolha”. A Internet utilizada atualmente tem origem no projeto da ARPANET (*Advanced Research Projects Agency Network*) desenvolvida nas décadas de 60 e 70. Embora por motivos acadêmicos e militares, a Internet transcendeu o campo da pesquisa e foi absorvida pela sociedade, tornando-se uma ferramenta fundamental de sobrevivência. É muito raro, hoje em dia, encontrar regiões e populações desconectadas. Todo tipo de serviço é baseado na Internet e está conectado em rede. De uma conta bancária a uma cirurgia complexa executada à distância, tudo converge para o mundo digital. Nossas casas, dispositivos pessoais e relacionamentos estão conectados e no mundo virtual.

O uso da mesma tecnologia de Internet há décadas apresenta vantagens e desvantagens. Ela não foi concebida para suportar a demanda e as necessidades de hoje [15] e consome muita energia [16]. Energia é uma questão em evidência em um mundo cada vez mais preocupado com a escassez dos recursos naturais. Chegou-se a um ponto em que uma nova Internet deve emergir para resolver estas e outras demandas existentes a um bom tempo na esfera global. Uma nova arquitetura de Internet é necessária. A arquitetura atual apresenta sinais de sua idade: é quase impossível controlar os *spams*, sofre para suportar dispositivos móveis, não se relaciona tecnicamente com tecnologias mais evoluídas e muito menos é nativa no suporte a Internet das Coisas (*Internet of Things*, ou IoT)[5][13][15].

Um nova Internet, ou uma Internet do futuro é necessária. Duas linhas de pesquisa se destacaram, a linha do *clean-slate* e a linha evolucionária. Na linha do *clean-slate* [14], a ideia é desenvolver uma nova arquitetura de Internet sem se preocupar com o *design* ou os limites da atual. É como desenvolver tudo novamente, sem interferência e sem normas ou imposições da proposta atual. Resolvendo problemas conhecidos sem utilizar adaptações e arquitetando novas funcionalidades para a rede. Isso parece ser a solução ideal para um futuro digital mais seguro e confiável. A promessa de uma Internet do futuro segura e confiável é atraente, mas

não é convincente. Uma nova arquitetura de Internet não basta ser melhor para prevalecer. A própria Internet atual teve concorrentes, como a proposta IBM *Systems Network Architecture* (IBM SNA) ou a DEC-NET [17].

Desenvolver uma arquitetura de Internet *clean-slate* significa ter mais erros que na proposta atual. Uma proposta nova, além de ser convincente precisa se sobressair em aspectos e ser mais viável economicamente. As propostas atuais de Internet *clean-slate* prevêm a resolução de alguns problemas da atual, como o endereçamento, a mobilidade, a segurança, a escalabilidade, a confiabilidade e a disponibilidade [15][18].

Por outro lado, utilizar a Internet atual e melhorá-la por meio da compreensão e resolução dos problemas existentes tem sido o caminho percorrido até o momento. O modelo utilizado até então tem sido evoluído por meio de adaptações [13]. Tecnologias evolutivas também têm sido criadas para lidar com os problemas não imaginados na concepção da ideia original. O NAT (*Network Address Translation*) [19] é um exemplo de tecnologia evolutiva desenvolvida para resolver um problema existente e se adaptou muito bem ao modelo da Internet atual. É admissível pensar que a Internet atual foi pensada como *clean-slate* na visão dos engenheiros da década de 60-70, portanto não se pode rebaixar ou invalidar a tecnologia que utilizamos até hoje.

Entretanto, a Internet de hoje está caminhando para o seu “limite de evoluções”, estando com um gargalo em inovações e criando novos problemas na tentativa de resolver os antigos. Cientes dessa realidade, os pesquisadores iniciaram nos últimos 15 anos a tentativa de esboçar uma nova arquitetura de Internet e, buscando justificativas para evoluir a atual chegaram na necessidade de ter que desconstruir a arquitetura existente para estabelecer uma nova. Os projetos que seguem essa linha têm se mostrado vantajosos em vários requisitos. Entre eles o *Scalable and Adaptive Internet Solutions* (SAIL) [21], *eXpressive Internet Architecture* (XIA) [22], o Akari [23] e a NovaGenesis, que é base do trabalho proposto nesta dissertação.

2.2 Ethernet e Wi-Fi

A *Ethernet* é uma tecnologia que surgiu em 1972 para transmissão de dados e interconexão de redes locais (*local area network* ou LAN) baseada no uso de quadros [24]. Esta tecnologia define o padrão de cabeamento, sinais elétricos, formato de quadros e protocolos para a subcamada de acesso ao meio (*media access control* ou MAC) do modelo OSI (*Open Systems Interconnection*). O *Institute of Electrical and Electronics Engineers* (IEEE) padronizou a *Ethernet* como 802.3 e desde os anos 90 tem sido a tecnologia para este fim mais amplamente utilizada, tomando o espaço dos padrões *Token Ring* e *Fiber Distributed Data Interface* (FDDI) [25].

Os padrões Ethernet evoluíram e os protocolos sofreram modificações para se tornarem mais eficientes e viabilizam altas taxas de transmissão. Os padrões atuais do protocolo *Ethernet* são [24]:

- 10 megabits/seg: 10Base-T Ethernet (IEEE 802.3)
- 100 megabits/seg: Fast Ethernet (IEEE 802.3u)
- 1 gigabits/seg: Gigabit Ethernet (IEEE 802.3z)
- 10 gigabits/seg: 10 Gigabit Ethernet (IEEE 802.3ae)

Com o surgimento dos *notebooks*, a necessidade dos consumidores de que a rede estivesse disponível sem o uso de cabos fez surgir o Wi-Fi [26]. O padrão para comunicação sem-fio homologado pelo IEEE é o 802.11 e em sua primeira versão possuía uma taxa teórica máxima de transmissão de até 2 Mbps. A partir daí outros padrões foram homologados, visando aumento de desempenho e taxa de transmissão. Os padrões, taxas e frequência dos padrões atuais são [27]:

- 802.11 – 2 Mbps e 2,4 GHz
- 802.11a – 54 Mbps e 5,0 GHz
- 802.11b – 11 Mbps e 2,4 GHz
- 802.11g – 54 Mbps e 2,4 GHz
- 802.11n – 600 Mbps e 2,4 e 5,0 GHz

Comercialmente, na maioria dos equipamentos domésticos utiliza-se o padrão 802.11g que oferece um desempenho bom para uso doméstico. Em redes corporativas e em provedores de serviços de Internet, o uso é do padrão 802.11n.

2.3 Redes Definidas por Software

Rede definida por *software*, ou *Software Defined Radio (SDN)* [28] é um termo que abrange o desenvolvimento de redes de computadores que permitem o controle de dispositivos de rede a partir da abstração das funcionalidades do nível mais baixo de uma infraestrutura de rede. Essencialmente o conceito de SDN pretende desacoplar a parte de controle da parte de dados. Com essa ideia, os *switches* e roteadores são capazes de trocar mensagens obedecendo regras definidas por um controlador de rede implementado em *software* fora dos equipamentos.

Com o crescimento exponencial do número de dispositivos conectados na rede, o controle e gerenciamento é atualmente um problema para os operadores de rede, sendo necessária muitos ajustes manuais e alto custo para a manutenção e suporte dos elementos de rede. Outro ponto crucial a se observar é a evolução da tecnologia, que provoca um desenvolvimento acelerado das tecnologias implementadas em *software* e a dificuldade de atualização ou *upgrade* dos itens de *hardware*. Por isso, permitir que o *software* implemente as funcionalidades de controle e gerência de uma rede garante que todo o ecossistema possa evoluir rapidamente acompanhando assim, o avanço tecnológico. O uso de *hardware* que permita configuração e modificação em seu *firmware* faz com que o uso de redes definidas por *software* seja mais interessante para um ambiente autônomo, reduzindo a intervenção humana e permitindo escalabilidade, qualidade e confiabilidade.

SDNs permitem a exposição, compartilhamento, controle e virtualização das funcionalidades de rede. Uma das primeiras implementações de SDN é o OpenFlow (OF) [7].

Desenvolvido pela Universidade de Stanford, foi concebido para suprir a demanda de novas arquiteturas e protocolos de rede, e também para possibilitar logicamente o controle centralizado de redes distribuídas. O OF implementa a abstração de encaminhamento, estados distribuídos, configurações e especificações. Em uma rede que faz o uso do OF, os *switches* são programáveis e simples. Todo o controle e a complexidade é movida para o *software* situado em elementos OF fora dos nós de encaminhamento.

Uma rede programável OF é composta por equipamentos de rede que suportam a abstração de encaminhamento OF. Utilizando um canal seguro com os protocolos TCP/IP e OF, realizam as mudanças nas tabelas de encaminhamento dos equipamentos e configuram como um fluxo de dados deverá ser tratado, bem como as possíveis ações para cada tipo de quadro em um fluxo específico.

2.4 Transporte de mensagens

Quando se faz necessária a comunicação entre processos em um sistema operacional (*threads* [29]) é importante considerar que nem sempre o remetente e o destinatário poderão lidar com a mensagem ao mesmo tempo e que eles não podem controlar fatores externos como perda de conexão, por exemplo. Um protocolo ou *framework* de transporte de mensagens faz com que o *software* em si não precise se preocupar com a transmissão e o gerenciamento de uma fila de mensagens.

O ZeroMQ [30] é um *framework* criado em 2007 para o transporte de mensagens entre vários ambientes e vários tipos, como intra-processos, inter-processos, TCP e *multicast*. É possível conectar *sockets* entre terminais utilizando diversos comportamentos como *pub-sub* e *request-reply*. Este *framework* funciona independentemente do cenário a ser utilizado e da linguagem implementada. Uma vantagem do ZeroMQ é que ele pode trabalhar sem a necessidade de *message broker*, ou seja, sem um intermediário que faz a conversão da mensagem do remetente para o destinatário.

Em 2011, o *Conseil Européen pour la Recherche Nucléaire* (CERN) [31] investigou formas de unificar as soluções intermediárias que utilizavam para operar os aceleradores de partículas. Um estudo realizado por eles comparou algumas soluções de transporte de mensagem e optou pelo ZeroMQ, principalmente por sua versatilidade.

Capítulo 3. NovaGenesis

NovaGenesis (NG) [4] é um projeto que foi iniciado em 2008, pelo Prof. Dr. Antônio Marcos Alberti, para lidar com uma questão latente: imagine se não houvesse uma arquitetura de Internet hoje, como poderíamos desenhar tal arquitetura, do zero, utilizando os princípios e tecnologias atuais? A partir dessa pergunta, foram selecionadas algumas tecnologias para fazer parte dos princípios de *design* da NovaGenesis, visando uma perfeita sinergia entre elas. Uma prova de conceito foi implementada e testada em 2012 na Coréia do Sul. Outros avanços têm sido feitos desde então no Inatel, Brasil.

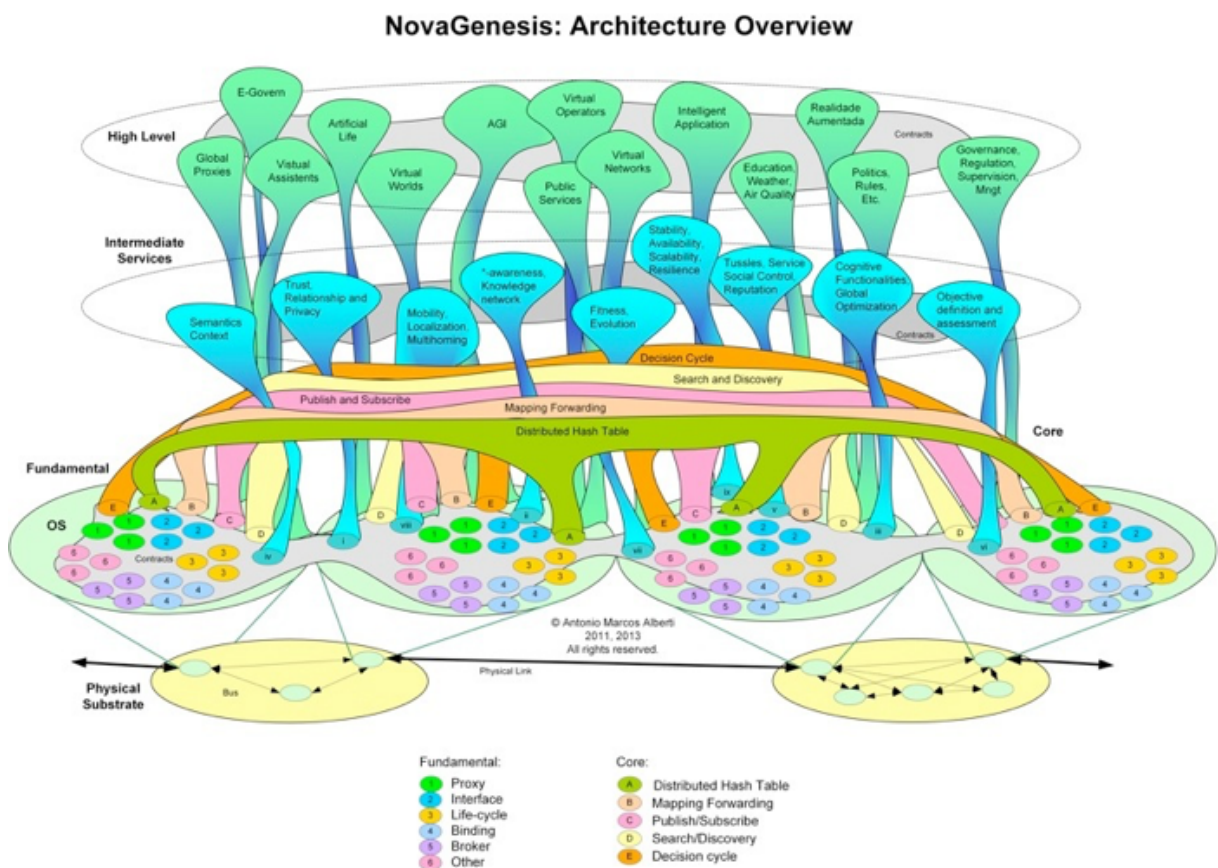


Figura 4 - Arquitetura NovaGenesis [4]

A NovaGenesis pode ser definida como uma arquitetura convergente de informação; A sua ideia central é a nomeação. Um nome é um conjunto de símbolos que caracteriza alguma coisa, algum objeto, alguma existência. É uma arquitetura profundamente inspirada e baseada em linguagem. Neste contexto, uma importante decisão que teve que ser tomada foi: quais existências precisam ser nomeadas em uma arquitetura convergente? Por existência, pode-se entender simplesmente tudo que existe. As pessoas, tradicionalmente e

evolutivamente, precisam nomear tudo: de um inseto a um avião, de uma cor a um arquivo. Um importante fato a se notar a respeito da evolução das arquiteturas atuais é que elas precisam ser capazes de entender o que as pessoas querem dizer, ou seja, entender o significado da linguagem utilizada – definida como semântica [32].

A nomeação deve ser flexível e ampla, mas nem todos os nomes naturalmente utilizados são eficientes, precisos e seguros para uso. Portanto, a NovaGenesis adota uma metodologia de nomeação chamada *Self-Certifying Naming* (SCN), ou nomes auto-certificáveis [33]. SCN é uma técnica de obtenção de um nome seguro, preciso, eficiente e possível de ser verificado de qualquer objeto. Um SCN é obtido a partir da passagem do objeto (uma foto digital, uma representação digital da íris humana, um arquivo, um pacote de rede, etc.) em uma função *hash* [34]. Toda informação que será processada, trocada ou armazenada depende de nomes e suas relações. O alvo de uma comunicação, independentemente do tipo que for, é algo que pode ser nomeado e identificado em um determinado escopo. Outros nomes relacionados ao destinatário podem indicar a distância ou a localização até o alvo da comunicação.

Um *Name Binding* é um relacionamento ou mapeamento entre dois ou mais nomes. Um nome pode ser relacionado com diversos outros nomes ou objetos. Por exemplo, o padrão de uma impressão digital de uma pessoa pode ser relacionado ao nome dessa pessoa, que por sua vez está relacionada ao nome de uma rua em um bairro de uma cidade. Um mapeamento desses nomes permitirá a localização do sujeito a partir do cruzamento dessas informações. Esse é o comportamento natural das pessoas, é o processo de relacionamento semântico. A proposta da NovaGenesis é fazer com que um NB possa representar os relacionamentos entre existências nomeadas. A arquitetura deve ser genérica o suficiente para permitir a criação de qualquer estrutura de nomeação. Uma estrutura de nomeação é um esquema que nomeia existências a partir de estratégias definidas e não há limites para a quantidade de espaços de nomeação existentes, portanto, a resolução de nomes dentro e entre esses espaços de nomeação é generalizada.

A NovaGenesis utiliza nomes para identificar e localizar alvos de comunicação. Um alvo pode ser um conteúdo, um programa de computador, o próprio computador ou qualquer outra existência. Para que isso aconteça, a arquitetura prevê uma tabela distribuída, um grafo, com os relacionamentos mapeados. Nomes que são únicos em algum escopo são também identificadores naquele escopo.

Localizadores são nomes que apontam para a posição de alguma existência em um determinado espaço. Qualquer elemento de uma arquitetura de informação convergente (*Convergent Information Architecture* ou CIA) é definido por existências físicas nomeadas, são os chamados recursos de substrato. Antenas, fibras ópticas, microprocessadores, memórias e discos rígidos são exemplos desses substratos.

A NovaGenesis define como serviço, uma existência que é capaz de processar, trocar e armazenar informação. Baseada nessa definição, um programa de computador, ou até mesmo um processo, pode ser considerado um serviço. Qualquer recurso de substrato pode ser representado por serviços nomeados, por exemplo *Infrastructure-as-a-service* (IaaS) [35]. Mesmo a implementação de um protocolo pode ser considerada um serviço, a partir do momento em que ele expõe seus recursos para outros protocolos como um serviço.

Protocolos e serviços são regidos por diferentes regras e conceitos. O modelo OSI [36] define um serviço como os recursos oferecidos por outras camadas, enquanto o protocolo é definido como um conjunto de regras para troca de mensagens entre as entidades em uma mesma camada. Na NovaGenesis não há distinção entre implementações de protocolos e serviços. De acordo com a definição de serviço, um protocolo é implementado como um serviço que processa, armazena e troca informação para construir redes. Portanto, serviços utilizam outros serviços indefinidamente iniciando a partir daqueles necessários para implementar *networking*.

A NovaGenesis prevê um ciclo de vida de um serviço que inclui exposição de recursos, descoberta de pares, negociação, contrato, monitoramento, avaliação e liberação. Essas etapas utilizam os *name bindings* descritores de serviços e contratos são objetos de informações baseados em SCNs. Um contrato é definido como parte da informação que especifica limites, responsabilidades e cláusulas a serem respeitadas, bem como os critérios e punições para um serviço que não é executado com qualidade.

As implementações de novos serviços a serem demonstrados nesta dissertação utilizam um conjunto de serviços básicos já implementados na NovaGenesis. Uma breve descrição dos serviços que compõem este núcleo base da arquitetura é feita a seguir.

O *hash table service* (HTS) é um serviço da NovaGenesis responsável por armazenar dentro de um escopo, ou domínio, o mapeamento de nomes, NBs, tanto para conteúdos (objetos de informação) quanto para serviços e agentes que estão em execução. Cada serviço tem sua própria *Hash Table* (HT) que armazena suas definições em nomes naturais e/ou auto-verificáveis, relacionamentos, informações e mapeamentos que são importantes. Essas

informações, do próprio serviço, são armazenadas nas HTs do domínio. A partir desse serviço é possível descobrir os parceiros e as funcionalidades que eles provêm. Uma HTS de um domínio pode se comunicar com outras de diferentes domínios e a partir daí criar um ambiente distribuído com todos os mapeamentos necessários para que uma rede de comunicações possa ser formada, habilitando a troca de informações entre todos os seus elementos, perfeitamente.

O *proxy-gateway service* (PGS) é um serviço que possui a funcionalidade de representante e tradutor ao mesmo tempo. Como *proxy*, o PGS é responsável por representar recursos que não são NovaGenesis dentro da arquitetura, como por exemplo o *socket* TCP/IPv6 [37] ou Ethernet, caso se deseje usar algumas dessas tecnologias junto com a NovaGenesis. É um serviço fundamental para interconectar protocolos NovaGenesis através de tecnologias legadas, através do recurso de *gateway*, fazendo uma ponte entre componentes NG e componentes do sistema operacional em computadores, bem como com elementos da Internet atual e outras redes locais de pacotes. O PGS ainda tem a responsabilidade de realizar algumas operações iniciais como o mapeamento de alguns NBs relacionados ao domínio, aos *hosts* e seus processos. A comunicação *inter-host* é responsabilidade do PGS através de um componente interno (bloco interno) chamado de *proxy-gateway* (PG).

O *publish/subscribe service* (PSS) realiza o *rendezvous* entre publicadores e subscritores. O PSS é responsável por permitir que uma entidade NG anuncie relacionamentos (NBs) a serem publicados a outras entidades. Também permite que entidades sejam notificadas sobre mapeamentos publicados por outras entidades e possam subscrevê-las. O serviço Pub/Sub é eficaz na arquitetura NovaGenesis por permitir que os recursos e informações sejam publicadas aos interessados, notificando-os, eliminando a comunicação direta e utilizando um padrão de troca de informações eficiente para redes de telecomunicações [38]. Atualmente, o PSS não implementa em seu *core* conceitos de segurança e privacidade, mas já permite em sua estrutura atual implementação destes mecanismos.

Generic indirection resolution system (GIRS) é um processo usado para decidir qual a HTS é a mais apropriada para armazenar os mapeamentos SCN a partir da quantidade de HTS disponíveis no domínio. O GIRS realiza um balanceamento de carga de informações e NBs, distribuindo igualmente entre as *hash tables* disponíveis. Além de auxiliar na seleção, o GIRS possui mecanismos para recuperar e encaminhar mapeamentos publicados.

Os serviços definidos acima podem ser referenciados também como processos no ecossistema NovaGenesis e estes processos são formados por blocos, conforme podem ser vistos na figura abaixo. Cada processo listado acima tem um bloco especializado em prover o serviço a ser ofertado, porém cada serviço (ou processo) tem alguns blocos básicos (comuns) desenhados na estrutura da NovaGenesis. Em outras palavras, a NovaGenesis implementa a ideia de *software fractal*, em que componentes de uma escala do projeto são reutilizados em vários componentes dessa e outras escalas. São eles: o *gateway* (GW), *hash table* (HT) e *command line interface* (CLI). O GW é responsável pela comunicação intra-processo, o HT armazena mapeamentos locais SCN e o CLI é uma interface para um operador NG humano.

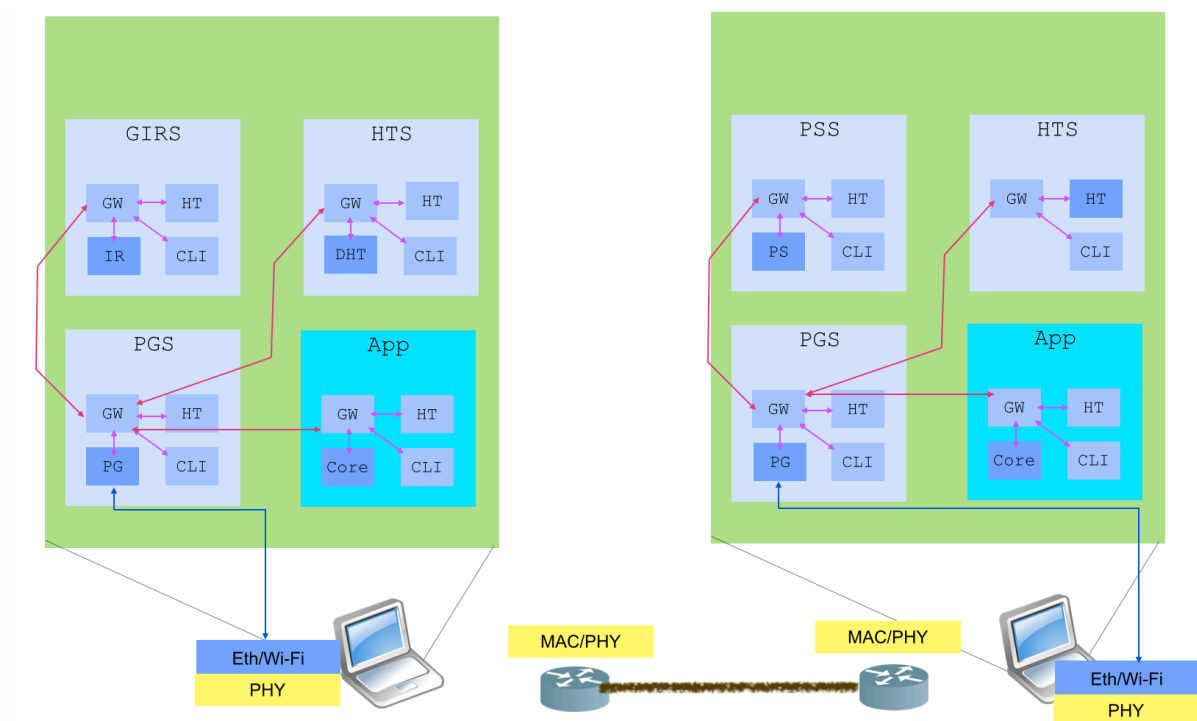


Figura 5 - Ecossistema NovaGenesis formado por blocos [4]

Capítulo 4. Desenvolvimento de novos serviços para NovaGenesis

Como contribuição ao projeto NovaGenesis, foram desenvolvidos quatro novos serviços que permitem o uso das informações de sensoriamento espectral, a tomada de decisões de uso de canais Wi-Fi e o controle de *access points* Wi-Fi e de controladores OpenFlow. Os novos serviços desenvolvidos foram denominados: *Spectrum Sensing Agent (SSA)*; *Resource Management Agent (RMA)*; *Access Point Agent (APA)* e *Python OpenFlow Controller Agent (POXA)*.

4.1 *Spectrum Sensing Agent (SSA)*

O conceito da NovaGenesis determina que um equipamento do mundo real tenha um serviço que o represente e exponha seus recursos e suas habilidades para outros agentes do ecossistema. Com isso, o *Spectrum Sensing Agent* é um serviço NG desenvolvido para permitir a exposição dos dados de um analisador de espectro eletromagnético para os agentes interessados.

O analisador de espectro é um instrumento eletrônico utilizado para amostrar as componentes harmônicas de sinais eletromagnéticos. Em um ambiente de rede, como a NG, um analisador de espectro pode ser útil para fornecer informações relevantes sobre o ambiente de rádio em uma dada região. Por exemplo, pode-se optar por utilizar um canal menos poluído em uma conexão sem-fio ou verificar se a informação propagada por um *access point* está de fato chegando a região sob observação.

Um analisador de espectro comercial normalmente possui uma interface serial que pode ser conectada a um computador e um *software* do equipamento faz a leitura de tais informações. O objetivo do SSA é capturar as informações fornecidas pelo equipamento e disponibilizá-las para a NG.

Um *script* desenvolvido em Python [39] configura o analisador de espectro para realizar a captura das informações, publicando as informações em um formato vetorial por meio do *ZeroMQ*. É amostrada a informação de energia presente nos 11 canais Wi-Fi na frequência de 2,4 GHz. Os dados são formatados em um vetor com 11 posições, onde a posição 0 corresponde ao canal 1, e assim sucessivamente. Os dados são publicados pelo *ZeroMQ* utilizando o padrão *Push/Pull*.

Neste padrão, o servidor (o computador com a interface direta com o analisador de espectro) obtém os dados do equipamento a partir da porta serial e publica para os clientes. Por sua vez os clientes, possuindo o endereço do servidor podem receber a informação, também utilizando o ZeroMQ. São configurados, no *script* de envio os parâmetros da função da biblioteca do ZeroMQ. Para envio (no servidor) é utilizado o padrão *PUSH*, que irá disponibilizar em uma porta TCP (*Transmission Control Protocol*) específica (também definida no *script*) a informação desejada. No SSA, a biblioteca do ZeroMQ é configurada para “escutar” na mesma porta TCP/IP com o padrão *PUSH*, que irá receber todas as informações que foram publicadas pelo servidor. Neste padrão, cada cliente coleta uma vez a informação enviada pelo servidor e cada cliente consome a mesma informação do servidor, sendo, portanto, satisfatório ao cenário pretendido para implantação.

O SSA, em sua inicialização, recebe como parâmetro o endereço IP do computador que possui o servidor que fornece as informações do analisador de espectro, bem como a porta em que o ZeroMQ está operando. Neste ponto, a NG interopera com tecnologia legada (TCP/IP) para fazer a comunicação entre um dispositivo do mundo real e a arquitetura NovaGenesis. A comunicação entre o SSA e os demais serviços NG utiliza o *raw socket* Ethernet/Wi-Fi (sem TCP/IP). A mesma implementação pode ser feita utilizando puramente elementos NG conectados diretamente ao analisador de espectro, mas considerando que o mesmo pode estar fora de alcance de estações que tenham o substrato NG e até mesmo possam estar remotamente conectados ao *core* da aplicação, é importante demonstrar que a NovaGenesis suportará as tecnologias atuais da arquitetura da Internet.

Na inicialização, o SSA já estabelece uma conexão com o servidor, não sendo necessário que este esteja transmitindo as informações, mas apenas que o *host* esteja ativo. O SSA expõe aos outros elementos da rede suas capacidades, através de palavras-chave como: *SSA*, *Core*, *Spectrum*, *Sensing*, *Agent*, *Wi-Fi* e o *ULN* (*unique legible name*). O *ULN* é um nome único em linguagem natural dado ao serviço pelo usuário, similar ao nome de *host* de um computador, que permite que se estabeleça uma conexão direta com o agente, identificando-o em relação aos outros que desempenham a mesma função em um dado escopo. Por exemplo, o *ULN* de um SSA pode ser “1_AMB_EXTERNO”, indicando que aquele SSA disponibiliza os dados do analisador de espectro do ambiente externo.

Na NG, o SSA expõe suas funcionalidades a partir das palavras-chave, descobre os parceiros que lhe interessam, como os serviços básicos de rede NG e outros que podem se interessar nos dados fornecidos por ele e firma contrato entre eles. Uma função de publicação

dos dados obtidos é agendada para acontecer periodicamente e é ela que faz a busca das informações.

Essas informações são trocadas na NG a partir de mensagens. Uma mensagem NG pode conter linhas de comando e dados (*payload*). Uma mensagem pode ser formatada de acordo com a regra a seguir:

ng -command --alternative version [vectorial arguments]

onde:

command é a ação a ser realizada.

alternative seleciona entre as alternativas que podem ser executadas em um comando.

version é a versão desejada do comando.

vectorial arguments são os argumentos (opcionais) do comando.

Cada comando pode ter um ou mais vetores de argumentos, sendo estruturados da seguinte forma:

< n type E1 E2 E3 E4 ... En >

onde:

n é o número de elementos no vetor.

type é o tipo de argumento presente no vetor.

E1 E2 E3 E4 ... En são os elementos do vetor.

Após a coleta das informações, o SSA irá preparar e publicar os dados no formato de uma mensagem NG. Os parceiros que se inscreveram para receber tal informação serão notificados da existência da mensagem e estarão aptos a recebê-la.

O cenário descrito nos parágrafos anteriores pode ser visualizado na Figura 3, que mostra o relacionamento do SSA com os outros elementos do sistema. Embora o cenário demonstrado atenda a um propósito específico, a metodologia de desenvolvimento, funções e dificuldades técnicas sanadas no desenvolvimento do SSA contribuem para o ecossistema da NG, podendo essa implementação ser generalizada para qualquer outro serviço ou agente que necessite se comunicar com periféricos através de tecnologia legada, utilizando uma biblioteca de comunicação (ZeroMQ) que suporte diferentes sistemas operacionais, linguagens de programação e modelos de transmissão das informações.

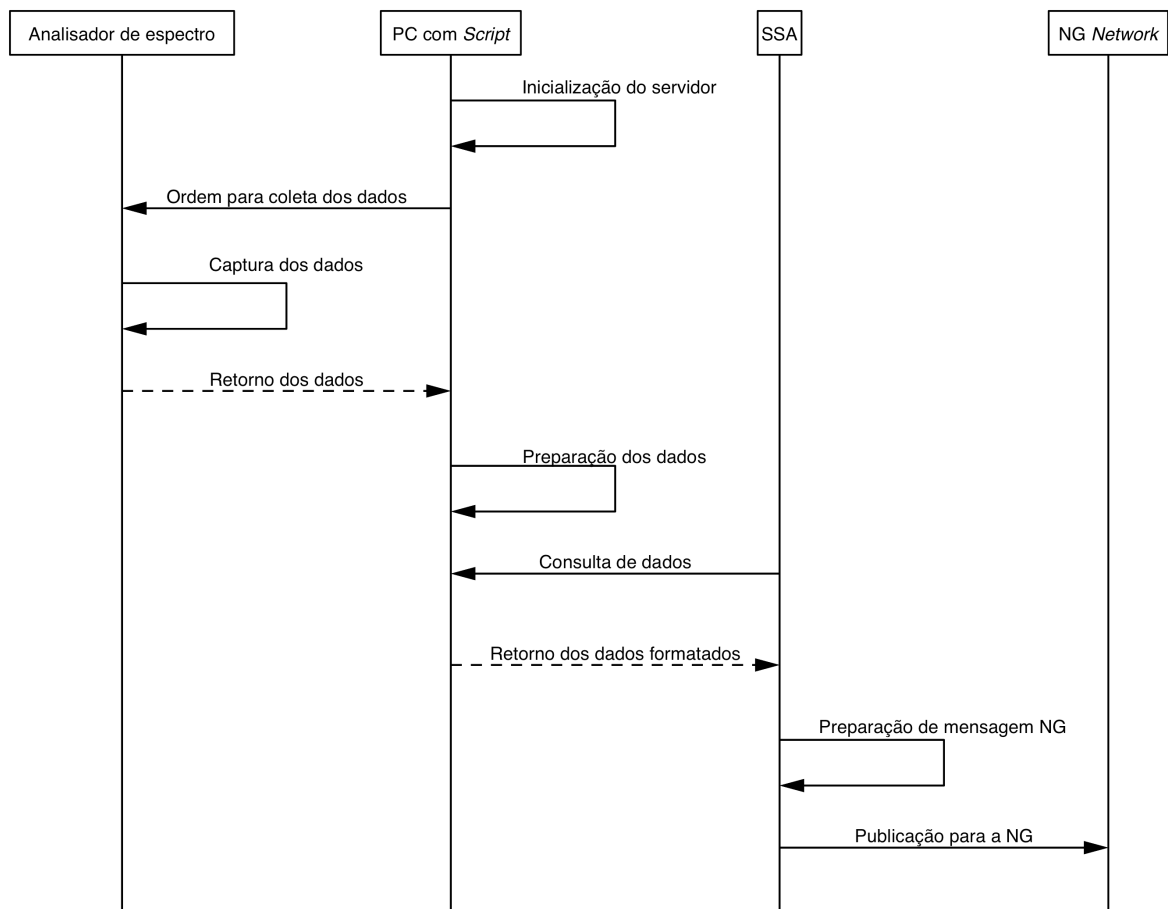


Figura 6 - Diagrama de sequência de funcionamento do SSA.

4.2 Resource Management Agent (RMA)

Na NovaGenesis, da mesma forma que há representantes do mundo real em seu ecossistema que tem a capacidade de obter informações e enviar comandos ou executar tarefas neles, existe a possibilidade de definir qual tarefa ou sob qual forma os dispositivos devem operar. Por exemplo, considerando a NG como arquitetura, um sistema de segurança residencial pode ter uma janela representada e um serviço que decida ou ajude a decidir se ela deve ser fechada ou não em caso, por exemplo, de chuva. É este o papel do *Resource Management Agent*, um serviço que permite que os controles e as decisões sejam tomadas a partir da entrada de informação de diversos outros serviços e a comunicação com tantos outros serviços “atuadores”.

Em um ambiente IoT, o RMA se torna um serviço útil para auxiliar a tomada de decisões, pois pode de forma autônoma, integrada, e com uma base de informações concisa,

auxiliar na operação de um dispositivo e até mesmo de uma rede inteira de diversos tipos de equipamentos, nas mais diversas áreas. O objetivo do RMA é assinar as informações disponíveis na NG e que servem ao propósito de tomar alguma decisão baseada nessas informações. Dessa forma, ele deve se comunicar tanto com os agentes que representam socialmente os dispositivos na NG quanto com os agentes que controlam e tem interface com esses mesmos dispositivos ou outras arquiteturas. Portanto, adota-se que o RMA, por ser um serviço que irá operar no *core* da NovaGenesis deve ser “puramente NG”, não possuindo interface direta com qualquer artifício ou elemento que não faça parte da arquitetura.

O RMA consiste em um serviço que essencialmente pode receber quaisquer informações dos serviços dos quais ele descobriu e tenha se tornado parceiro (através do estabelecimento de um acordo de nível de serviço ou contrato). Uma função executada no momento em que recebe as informações de seus parceiros deve verificar os dados recebidos, catalogar e armazenar em sua própria HT, na HT do domínio (HTS), ou em uma estrutura de dados específica chamada *DataStore*. Ela foi definida dentro do *core* da NG para servir a todo e qualquer serviço. Basicamente ela armazena um identificador de categoria, o nome da chave, o valor do dado em forma de texto e em forma numérica. Seguindo a definição de categorias [4] a estrutura permite que os dados sejam agrupados por categorias que possam ser futuramente implementadas. A chave é a identificação em si do dado, como por exemplo a descrição do canal de Wi-Fi ao qual pertence uma certa medida e as duas opções de armazenamento de valores são para facilitar as conversões e a codificação em diversos pontos que exigem tal manipulação. Há ainda a possibilidade de armazenar um *timestamp* [40] para indicar o momento em que o dado foi recebido ou armazenado, sendo útil para algoritmos de decisão que queiram tratar a “idade” do dado.

Ao receber a informação, de acordo com cada parceiro, um vetor é criado para armazenar e catalogar tais informações e garante que elas estejam a disposição no momento em que se desejar tomar as decisões.

A função do RMA que toma a decisão pode ser chamada periodicamente ou a partir de uma solicitação de outro agente ou serviço. O algoritmo de decisão pode e deve ser implementado de acordo com a necessidade de cada serviço requisitante ou participante da rede. No cenário proposto nesta dissertação, o RMA possui um algoritmo de decisão que analisa os dados recebidos de um SSA e estima o canal que está mais favorável a uma operação sem ruídos ou interferências.

Neste caso, um algoritmo pode ser desenvolvido, para por exemplo, percorrer todos os valores armazenados de cada canal e realizar uma média aritmética simples para definir qual canal está, historicamente, mais aceitável para a utilização. A informação com o canal escolhido, sendo resultado da função do algoritmo de decisão é encapsulada em uma mensagem NG e preparada para ser enviada ao serviço que irá executar a alteração do dispositivo de rede.

Para o RMA foi criada uma nova função de publicação que permite que os dados sejam publicados para um parceiro em específico, permitindo que um serviço possa se comunicar com outro através do seu ULN, como visto no tópico relacionado ao SSA. Com isso, a partir de uma solicitação ou definição, o RMA pode disparar a mensagem contendo a informação ou comando que deve ser repassado para um serviço exclusivamente. Ainda é possível publicar a informação para todos os elementos da NG ou apenas para aqueles que o RMA tenha descoberto.

Na Figura 4, pode-se observar a ideia de funcionamento do RMA no cenário proposto em [6], foco de implementação de tais serviços. O APA, representante e atuador de um *access point* será apresentado a seguir.

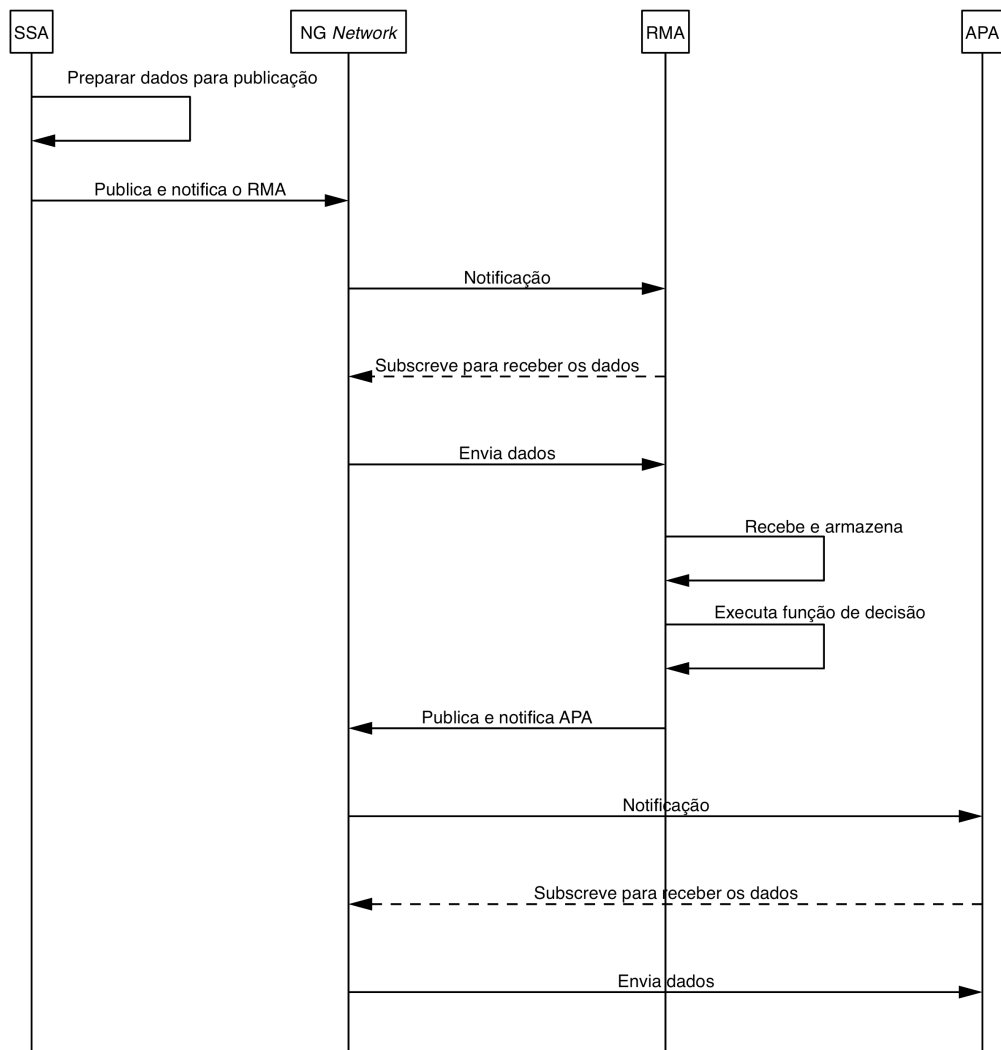


Figura 7 - Diagrama de seqüência de funcionamento do RMA.

O RMA tem como proposta ser um serviço fundamental da NovaGenesis, auxiliando também na orquestração da própria rede, incorporando um elemento abstrato que podemos definir como o “tomador de decisões”, tarefa que, nas redes e arquiteturas atuais, normalmente são desenvolvidas por operadores humanos. Considerando que o RMA está dotado de capacidade de armazenar informação e tomar decisão a partir delas, pode ajudar a definir serviços, capacidades, premissas e recursos necessários para a execução de tarefas e, a partir disso, orquestrar com outros serviços a criação e disponibilização dos mesmos.

4.3 Access Point Agent (APA)

O *Access Point Agent* é um representante de um *access point*, equipamento utilizado para prover acesso a uma rede Wi-Fi e, os equipamentos domésticos, normalmente operam nos canais 2,4 e/ou 5,8 GHz. Este equipamento muitas vezes opera de forma individual, com suas configurações *default* e quase nunca é otimizado para o uso em decorrência da região em que se situa. A NovaGenesis prevê que os dispositivos devem se comunicar e interagir para juntos oferecerem os melhores recursos aos seus utilizadores, de acordo com as premissas e limitações definidas.

O serviço APA foi desenvolvido com o objetivo de ser um representante “social” de um AP na rede NG, fornecendo informações e executando tarefas diretas no equipamento. Alguns equipamentos permitem interação e controle diretamente através de uma interface serial, *Universal Serial Bus* (USB) ou mesmo biblioteca que permita a manipulação das configurações. Uma parcela dos *access points* disponíveis permitem que o seu *firmware* seja alterado, podendo por exemplo, substituir o seu *software* de gerenciamento pelo OpenWrt [41], baseado em Linux e de código aberto. Sua vantagem é permitir que todas as configurações do equipamento sejam acessadas e modificadas, tornando-o totalmente gerenciável e configurável.

Com um *hardware* com esta habilidade, o APA pode acessar diretamente o equipamento, controlando e alterando suas configurações, como o canal em que opera, o nome que identifica a rede disponível, entre outros modos de operação conforme determinação do próprio APA ou qualquer outro agente da NG, como o RMA. Portanto, o APA foi desenvolvido para ser um representante que se comunica diretamente com o equipamento e ao receber uma mensagem com dados de controle e operação, pode acessar ou se comunicar sem *scripts* ou programas intermediários com o dispositivo, observando a necessidade de permitir que tal dispositivo possua tal funcionalidade, seja através de seu *firmware* adaptado ou original. Na Figura 5, pode-se observar o funcionamento do APA no cenário proposto em [6], interagindo e controlando diretamente o dispositivo, além de representá-lo no ecossistema NG.

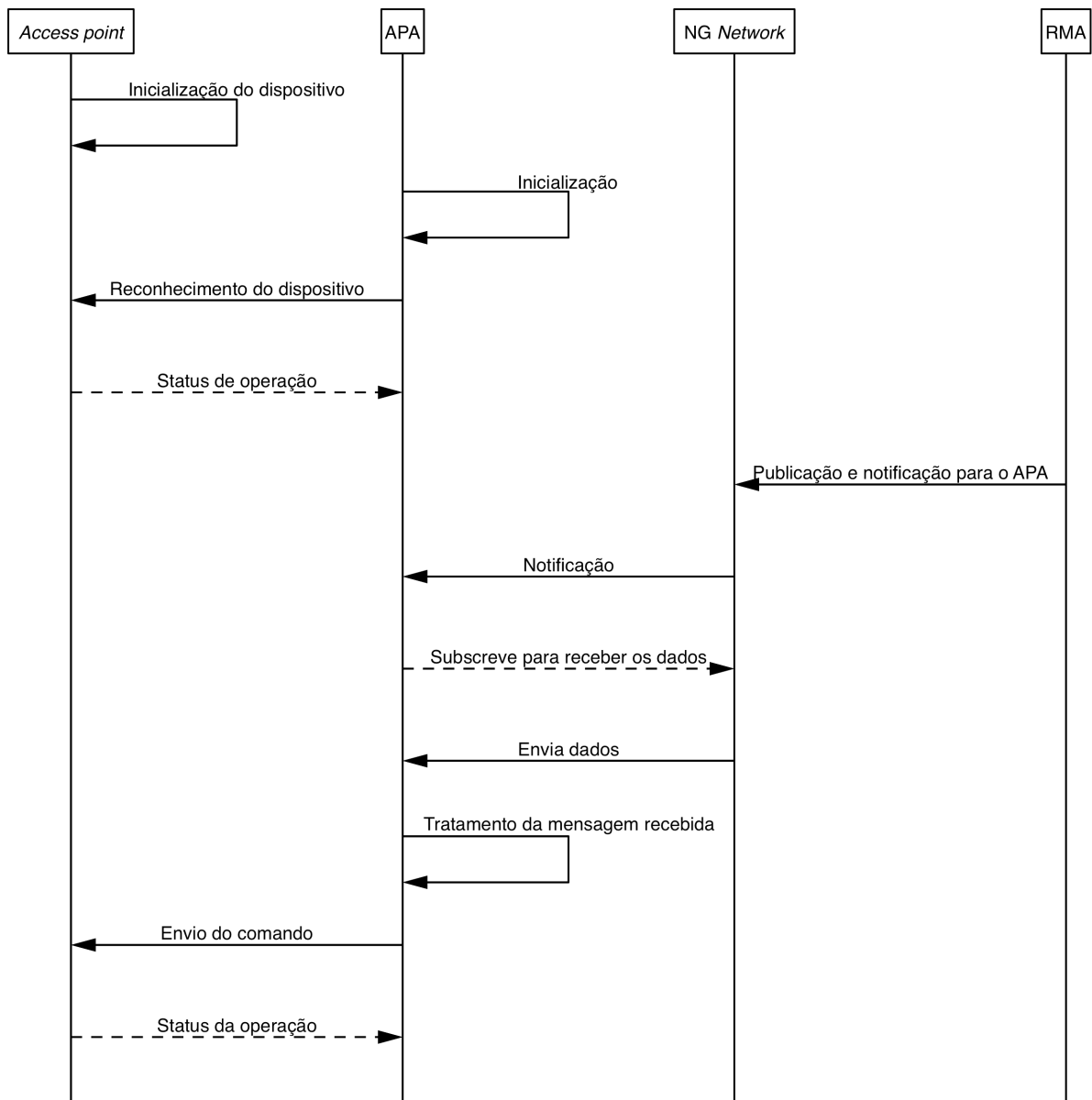


Figura 8 - Diagrama de sequência de funcionamento do APA.

O APA, em sua inicialização, recebe como parâmetro o endereço de hardware de um AP e também um ULN para o identificar diretamente. O algoritmo de decisão pode utilizar desse nome único legível para enviar ordens especificamente para o dispositivo, e também dentro da filosofia NG, propagar a informação para todos os representantes de *access points* na rede. Logo no início de seus procedimentos, o APA estabelece a conexão com o equipamento e garante que ele esteja respondendo e tenha controle sobre ele.

Ao receber uma mensagem NG solicitando a alteração de alguma configuração, tal como o RMA solicitando a alteração do canal de operação do equipamento Wi-Fi, o APA trata a mensagem recebida e envia o comando diretamente ao equipamento que o executa, retornando o resultado da operação.

A proposta de um serviço como o APA é representar e expor as habilidades de um dispositivo para a NovaGenesis, mas também permitir que se utilize tecnologia legada para comunicação direta. O protocolo mais comum e recomendado é o SSH (*Secure Shell*) [42] que permite, através da rede atual, conectar de forma segura um *host* a outro. Com isso, não é necessária a existência de um terceiro serviço, seja no mesmo *host* ou em um intermediário, que receba as informações do elemento NG e transfira o comando para o dispositivo. O objetivo é simplificar a ligação entre dispositivos que não possuem NovaGenesis em seu *core*, mas que se comunicam através de protocolos e interfaces que possuem os dois elementos nas pontas.

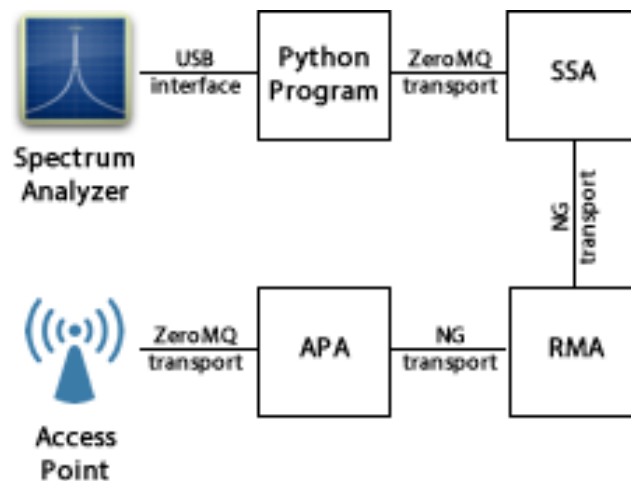


Figura 9 - Interdependência entre SSA, RMA e APA

Como pode-se observar nos tópicos anteriores, o SSA, APA e RMA formam, de certa forma, um trio interdependente para permitir que seus objetivos sejam alcançados, ilustrado na figura acima. O SSA representa um analisador de espectro na NG e é responsável por coletar as informações geradas por ele a partir de um *script* que interpreta os dados e os disponibiliza para serem utilizados. Com as informações acumuladas, o RMA pode executar um algoritmo de decisão que determina qual o melhor canal para uso no momento, de acordo com premissas, padrões e comportamentos desejados e envia a informação para o APA, que se encarrega de enviar os comandos de alteração diretamente ao *access point*. Este modelo traz, além do aspecto prático implementado para este cenário em questão, a lógica de

funcionamento de um ecossistema baseado em Internet das Coisas, onde dispositivos reagem de acordo com a informação, comportamento e decisão de outros elementos da rede, mantendo contratos entre si.

4.4 Python OpenFlow Controller Agent (POXA)

Assim como o APA, o *Python OpenFlow Controller Agent* é um representante de um equipamento ou elemento fora do ecossistema da NovaGenesis. O POXA representa, portanto, um controlador *OpenFlow*, neste caso, o POX [43]. O POX é um controlador desenvolvido em Python que visa permitir o desenvolvimento rápido de aplicativos de rede para controlar equipamentos via OF.

O POXA foi desenvolvido com o intuito de representar uma instância do POX que esteja rodando em uma rede legada ou definida por *software*, e pode se comunicar utilizando as mesmas estratégias adotadas no SSA, através do ZeroMQ, utilizando o também o modelo de operação *PUSH/PULL*. O *ZeroMQ* faz com que o POXA possa se comunicar com o POX e o mesmo realizar as alterações dentro do contexto exigido em um ou mais equipamentos de uma rede definida por *software*.

Uma rede OF programável é composta por equipamentos de rede que suportam os conceitos e abstrações de uma rede SDN. Utilizando SSH e protocolos OF, pode-se realizar mudanças nas tabelas de fluxo dos equipamentos, bem como definir ações exclusivas para cada quadro de um tráfego específico que compõem esta rede.

Um *switch* OF pode ser controlado por um ou mais controladores ao mesmo tempo. É neste ponto que o POX age como controlador de dispositivos OF. Assim, a NG ativa a possibilidade de expor os recursos de um controlador e até mesmo de uma rede OF, auxiliando na orquestração e administração de uma rede definida por *software*, principalmente por poder se basear nas políticas de contrato existentes na filosofia da nova arquitetura.

Na Figura 6 pode-se observar como é o funcionamento do POXA frente ao ecossistema NovaGenesis e ao cenário de implementação de um controlador OF. Para ilustrar, será considerado, assim como no APA, que o RMA após execução de seu ciclo de decisão, deseje alterar alguma configuração de fluxo em um *switch* com o OpenFlow.

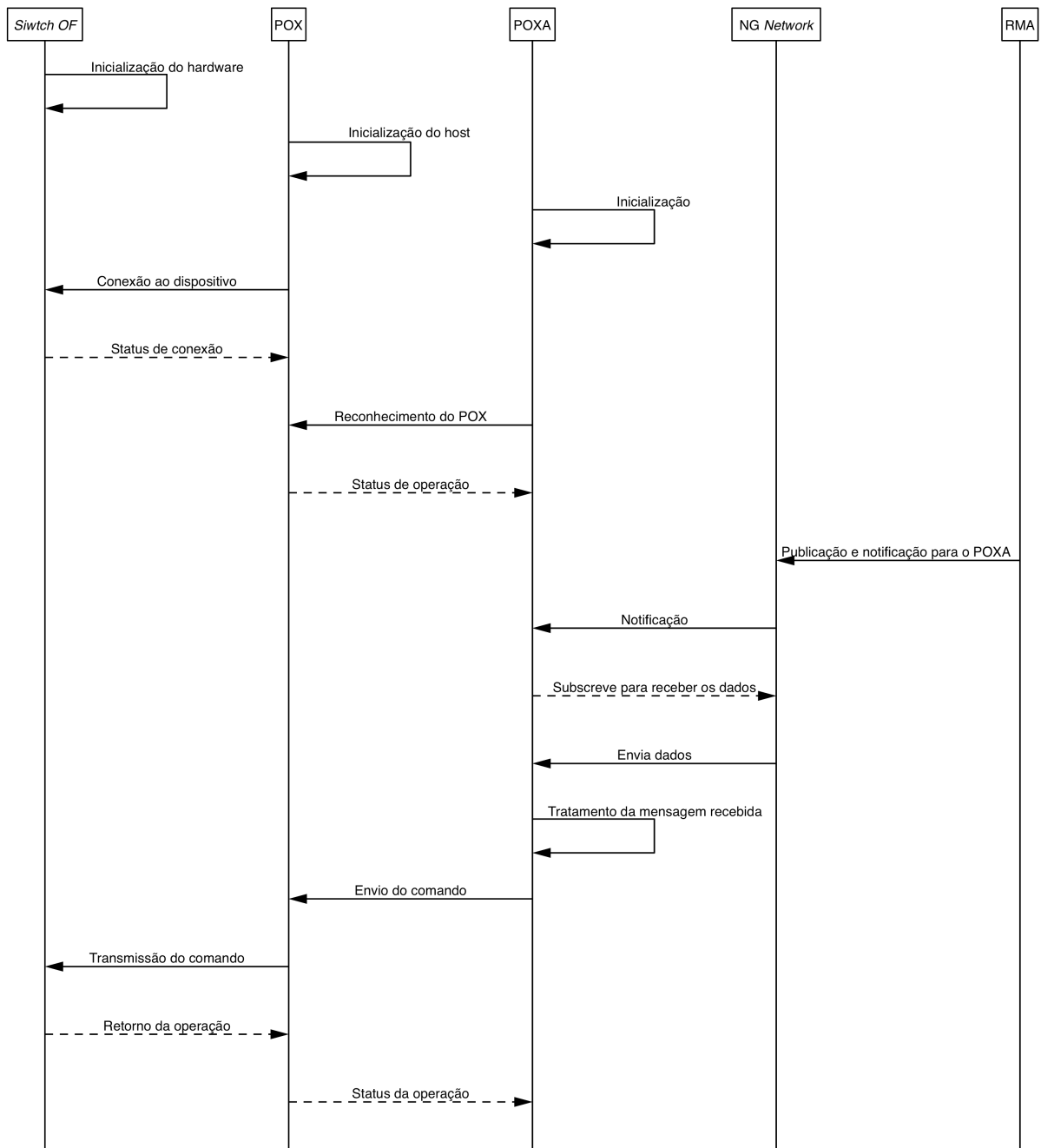


Figura 10 - Diagrama de sequência de funcionamento do POXA.

Neste cenário, o POXA opera com 2 dispositivos que não fazem parte do *core* da NovaGenesis. Um *switch* que tem seu *firmware* adaptado para operar com o OpenFlow e o POX que é o controlador disponível para a configuração da rede. É importante ressaltar que os dois agentes não precisam estar fisicamente separados. Se o equipamento suportar a instalação do POX ou mesmo se for virtualizado, é possível que estejam no mesmo *hardware*.

Na inicialização, o POXA recebe o endereço IP do POX e a porta em que o ZeroMQ está operando para que seja possível a comunicação entre eles. Da mesma forma, o POX realiza uma conexão inicial ao *switch* para garantir que o controle possa ser enviado.

O RMA, por exemplo, ao definir que uma alteração deve ser realizada na tabela de fluxo do *switch* ou na sua configuração, realiza a publicação das informações para que o POXA possa recebê-las, trata-las e enviá-las ao POX.

O POX também irá fazer um tratamento da mensagem recebida, conectar-se ao dispositivo e transmitir o comando que deve ser executado. Tanto o dispositivo quanto o POX retornarão uma mensagem de resultado da operação, que pode estar disponível ao POXA para que, por sua vez, comunique aos interessados, como por exemplo o RMA.

Capítulo 5. Implementação dos 4 novos serviços NovaGenesis

Neste capítulo, são apresentados *logs* e registros de mensagens trocadas na arquitetura NovaGenesis que demonstram o funcionamento dos serviços propostos e implementados.

5.1 *Spectrum Sensing Agent (SSA)*

No quadro abaixo pode-se observar a inicialização do SSA, já realizando a conexão ao ZeroMQ, que neste caso está rodando na mesma máquina (*localhost*) na porta TCP 5551.

```
(The unique legible name is SSA01)
(The I/O path is /home/lucio/workspace/novagenesis/IO/SSA/)
(The Socket is tcp://127.0.0.1:5551)
(The process legible name is SSA)
(The process self-certifying name is 2664FD52A4EF04471DBB9B971FD5C257_SSA_PID)
(The operating system legible name is Linux 3.13.0-24-generic #47-Ubuntu SMP Fri
May 2 23:30:00 UTC 2014)
(The host legible name is lucio-VGN-FW350J x86_64)
(The      operating      system      self-certifying      name      is      =
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
(The host self-certifying name is = 6336892BE50F9DBF08EB39AB7F7C704F_HID)

      (Moving to operational state)

      (NumberOfIPCInputKeys = 1)
      (Adding the InputKey = 6879)
      (NumberOfIPCOutputKeys = 1)
      (Adding the OutputKeys = 6880)
[1]      (IQ Size = 0)
[1]      (IQ Size = 1)
(DSCN = 72C8AE5CAFC8CC6E3E43C5E9A47DE733_DID)
(Connecting to the controlled device with the socket tcp://127.0.0.1:5551)
```

Quadro 1 - Inicialização do SSA

Como o SSA e todos os outros agentes a serem apresentados são serviços NovaGenesis, todas as etapas de descoberta, exposição e troca de contratos (ciclo de vida dos serviços) são executadas antes da troca efetiva de amostras de espectro e por razão de simplicidade das explicações serão suprimidas deste texto.

No quadro abaixo, observa-se que o SSA já descobriu o RMA, trocou contratos e está verificando se há alguma subscrição dos mesmos. Os valores em hexadecimal mostrados são os nomes auto-certificáveis (SCN) usados como identificadores/localizadores das existências na NG.

```
(Time = 4552.36)
(NextPeerEvaluationTime = 4544.27)
(1. Check for PSS awareness.)
  (Aware of a PSS on Categories 2 and 9)
  (The SSA is already aware of this PSS)
(2. Check for new peer application tuples)
  (Testing the candidate = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
  (The SSA is already aware of this candidate)
  (The SSA is already aware of this candidate)
(3. Show the discovered server App(s))
  (Aware of the Application 0)
  (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
  (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
  (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
  (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
(4. Check subscriptions)
  (Testing subscription 0)
  (Subscription status is Scheduling required)
  (The publisher is already known and have the index 0)
  (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
  (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
  (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
  (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
```

Quadro 2 - Descoberta do RMA pelo SSA

Periodicamente, em um tempo definido para cada programa, uma função de consulta ao *bus* do ZeroMQ é realizada para verificar se há dados disponíveis. No exemplo no quadro abaixo, o registro de que não há informação é mostrado.

```
(Preparing SSData Publish.)
Waiting for SSData...
(Nothing received from Spectrum Sense, skipping...)
```

Quadro 3 - SSA consultando via ZeroMQ

O *script* desenvolvido em Python que faz a leitura das informações do analisador de espectro é executado e os dados são enviados para o *bus* do ZeroMQ. Neste exemplo, foram utilizados dados fictícios, gerados a partir de uma função randômica, no intervalo de 0 a 100.

```
Connecting via PUSH..
Sending Data...
[90, 87, 48, 76, 18, 65, 26, 36, 50, 79, 4]
```

Quadro 4 - Script Python enviando dados

Abaixo, a confirmação do recebimento dos dados pelo SSA e a preparação da mensagem para ser publicada. Cabe ressaltar que este processo não precisa ser sincronizado, pois o ZeroMQ irá “aguardar” o recebimento das informações pelo subscritor.

```

(Preparing SSData Publish.)
Waiting for SSData...
(Message received from Spectrum Sense: [90, 87, 48, 76, 18, 65, 26, 36, 50, 79,
4])
(Aware of channel data values. Preparing to publish.)
(1. Setting limiter.)
(2. Setting source.)
(3. Setting Destinations.)
(4. Creating message.)
(5. Charging payload.)
(6. Generating SCN.)
(7. Pushing to GW input queue.)
[1]      (IQ Size = 3)
(Channel data published -----.)

```

Quadro 5 - SSA recebendo dados via ZeroMQ

Logo, a mensagem NG é enviada ao RMA, com as informações em formato de texto, dentro do arquivo (*payload*) SSAFile1.txt. Em toda troca de informações na NG é garantida a integridade dos dados.

```

[3]      (Finished)
          (t = 5006.921437)
[2]      (Processing)
          (File = SSAFile1.txt)
          (Peer index 0)
          (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
          (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
          (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
          (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
          (The payload hash is ACA135D81BAD0A4CB4E53C3BB7807B45)

```

Quadro 6 - Mensagem do SSA para o RMA

Por fim, um quadro mostrando que o RMA conhece o SSA, é parceiro dele e recebeu as informações publicadas pelo SSA.

```

(Time = 5007.754655)
(NextPeerEvaluationTime = 5012.12104)
(1. Check for PSS awareness.)
    (Aware of a PSS on Categories 2 and 9)
    (The RMA is already aware of this PSS)
(2. Check for new peer application tuples)
    (Too early for that. Wait next ng -run --evaluate)
(3. Show the discovered server App(s))
    (Aware of the Application 0 )
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 02E70FC90F0D8D78563C00FE8F0A7ACD_APA_PID)
    (BID = A920FC691CDE8F15F1AC9CA9561640F8_Core_BID)
    (Aware of the Application 1 )
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 9486105B8A39B1E3C03F0ADCDED80BFB_APA_PID)
    (BID = 47A9A33CCC7D9B42D417DE724EFFF9B3_Core_BID)
    (Aware of the Application 2 )
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 2664FD52A4EF04471DBB9B971FD5C257_SSA_PID)
    (BID = D8C80A763A2881133A5C86A3975D7BAF_Core_BID)
(4. Check subscriptions)
    (Testing subscription 0)
    (Subscription status is Waiting delivery)
    (The publisher is already known and have the index 1)
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 9486105B8A39B1E3C03F0ADCDED80BFB_APA_PID)
    (BID = 47A9A33CCC7D9B42D417DE724EFFF9B3_Core_BID)
    (Testing subscription 1)
    (Subscription status is Processing required)
    (The publisher is already known and have the index 3)
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 2664FD52A4EF04471DBB9B971FD5C257_SSA_PID)
    (BID = D8C80A763A2881133A5C86A3975D7BAF_Core_BID)
    (Checking the file received from the peer with name SSAFile1.txt)
    (File extension = txt)
    Store (DATASTORE)
    Channel = 1
    Value = 90
    Channel = 2
    Value = 87
    Channel = 3
    Value = 48
    Channel = 4
    Value = 76
    Channel = 5
    Value = 18
    Channel = 6
    Value = 65
    Channel = 7
    Value = 26
    Channel = 8
    Value = 36
    Channel = 9
    Value = 50
    Channel = 10
    Value = 79
    Channel = 11
    Value = 4
    Check stored DATASTORE
    Tamanho do vetor: 11
    (Going to parse the file SSAFile1.txt)
    (Deleting the subscription with index = 2)

[3] (Finished)

```

Quadro 7 - Recebimento das informações do SSA pelo RMA

5.2 Resource Management Agent (RMA)

Do mesmo modo que o SSA, podemos observar no quadro abaixo a inicialização do RMA.

```
(The unique legible name is RMA01)
(The I/O path is /home/lucio/workspace/novagenesis/IO/RMA/)
(The Socket is tcp://10.0.185.163:5555)
(The process legible name is RMA)
(The process self-certifying name is 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
(The operating system legible name is Linux 3.13.0-24-generic #47-Ubuntu SMP Fri
May 2 23:30:00 UTC 2014)
(The host legible name is lucio-VGN-FW350J x86_64)
(The      operating      system      self-certifying      name      is      =
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
(The host self-certifying name is = 6336892BE50F9DBF08EB39AB7F7C704F_HID)

      (Moving to operational state)

      (NumberOfIPCInputKeys = 1)
      (Adding the InputKey = 3060)
      (NumberOfIPCOutputKeys = 1)
      (Adding the OutputKeys = 3061)
[1]      (IQ Size = 0)
[1]      (IQ Size = 1)
(DSCN = 72C8AE5CAFC8CC6E3E43C5E9A47DE733_DID)
```

Quadro 8 - Inicialização do RMA

Os registros de publicação de uma informação para o RMA podem ser confirmados no item anterior, na demonstração da publicação de informações pelo SSA. Portanto, podemos verificar a preparação do RMA para publicação de uma mensagem para o APA. Lembrando que toda publicação feita por um serviço passa pelo núcleo da NovaGenesis (PSS para GIRS para HTS).

```
(Preparing APAData Publish.)
(Aware of APA data values. Preparing to publish.)
(1. Setting limiter.)
(2. Setting source.)
(3. Setting Destinations.)
(4. Creating message.)
(5. Charging payload.)
(6. Generating SCN.)
(7. Pushing to GW input queue.)
[1]      (IQ Size = 4)
(Channel data published -----.)
```

Quadro 9 - Publicação para o RMA

No quadro abaixo, há uma mensagem que deve ir para um agente específico. É mostrado neste ponto que o destinatário é o agente com nome único legível “APA01”. Com isso, apenas o agente em questão será notificado e receberá a informação. Neste caso, o nome em Português é usado para identificar o serviço alvo e traduzido para o nome auto-certificável relacionado.


```

(_ULName = APA01)
(Peer = 02E70FC90F0D8D78563C00FE8F0A7ACD_APA_PID)
(PeerIndex = 0)
(HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
(OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
(PID = 02E70FC90F0D8D78563C00FE8F0A7ACD_APA_PID)
(BID = A920FC691CDE8F15F1AC9CA9561640F8_Core_BID)
(File = APAFile1.txt)
(Path = /home/lucio/workspace/novagenesis/IO/RMA/)

(The payload hash is F249E533E5E7CF78E0AB857F7B7E5F8F)
(The following message was published to the peer)
(
ng -msg --cl 0.1 [ < 1 string 72C8AE5CAFC8CC6E3E43C5E9A47DE733_DID > < 4 string
6336892BE50F9DBF08EB39AB7F7C704F_HID 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID
89F30CF99F75D2BB209B74D489D28661_RMA_PID 2A5D623904EFA29288214E33EC865B5B_Core_BID
> < 4 string 6336892BE50F9DBF08EB39AB7F7C704F_HID
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID 1A13C48D59444355A920FC69D417DE72_PSS_PID
CB897B7E118CFABBD55A93C37F14E7D6_PS_BID > ]
ng -pub --notify 0.1 [ < 1 string 18 > < 1 string F249E533E5E7CF78E0AB857F7B7E5F8F
> < 1 string APAFile1.txt > < 5 string pub 6336892BE50F9DBF08EB39AB7F7C704F_HID
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID 02E70FC90F0D8D78563C00FE8F0A7ACD_APA_PID
A920FC691CDE8F15F1AC9CA9561640F8_Core_BID > ]
ng -info --payload 0.1 [ < 1 string APAFile1.txt > ]
ng -pub --bind 0.1 [ < 1 string 2 > < 1 string F249E533E5E7CF78E0AB857F7B7E5F8F > <
1 string 2A5D623904EFA29288214E33EC865B5B_Core_BID > ]
ng -pub --bind 0.1 [ < 1 string 2 > < 1 string F249E533E5E7CF78E0AB857F7B7E5F8F > <
1 string 89F30CF99F75D2BB209B74D489D28661_RMA_PID > ]
ng -pub --bind 0.1 [ < 1 string 2 > < 1 string F249E533E5E7CF78E0AB857F7B7E5F8F > <
1 string 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID > ]
ng -pub --bind 0.1 [ < 1 string 9 > < 1 string F249E533E5E7CF78E0AB857F7B7E5F8F > <
1 string 6336892BE50F9DBF08EB39AB7F7C704F_HID > ]
ng -message --type 0.1 [ < 1 string 1 > ]
ng -message --seq 0.1 [ < 1 string 319 > ]
ng -scn --seq 0.1 [ < 1 string D7E623FDF46BB3AC750D7B6BFAD40D69 > ]

There is a payload of 19 bytes

```

Quadro 10 - Mensagem enviada para agente específico

O RMA também é capaz de enviar informações para todos os parceiros, utilizando função específica para isso. Neste caso, para exemplo, o RMA prepara uma mensagem e envia para todos os seus parceiros. A mensagem é endereçada ao *Active Agent Antenna* (AAA) que tem seu *core* já desenvolvido, porém sem funcionalidade na NG e será alvo de trabalhos futuros.

```
(Preparing AAAData Publish.)
(Aware of AAA data values. Preparing to publish.)
(1. Setting limiter.)
(2. Setting source.)
(3. Setting Destinations.)
(4. Creating message.)
(5. Charging payload.)
(6. Generating SCN.)
(7. Pushing to GW input queue.)
[1]      (IQ Size = 4)
(Channel data published -----.)

[3]      (Finished)

          (t = 4850.759952)

[2]      (Processing)

          (File = AAAFile1.txt)
          (Peer index 0)
          (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
          (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
          (PID = 02E70FC90F0D8D78563C00FE8F0A7ACD_APA_PID)
          (BID = A920FC691CDE8F15F1AC9CA9561640F8_Core_BID)
          (Peer index 1)
          (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
          (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
          (PID = 9486105B8A39B1E3C03F0ADCDED80BFB_APA_PID)
          (BID = 47A9A33CCC7D9B42D417DE724EFFF9B3_Core_BID)
          (Peer index 2)
          (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
          (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
          (PID = B7F45C0ACC01EE301F4D6B576CB2CDB2_AAA_PID)
          (BID = 8223165BEDE3ABE79D51404A3D25ED96_Core_BID)
          (Peer index 3)
          (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
          (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
          (PID = 2664FD52A4EF04471DBB9B971FD5C257_SSA_PID)
          (BID = D8C80A763A2881133A5C86A3975D7BAF_Core_BID)
          (Peer index 4)
          (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
          (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
          (PID = 8B48F85F8F0A7ACD189BABAA5A2B1D53_POXA_PID)
          (BID = A6DB57CD7F0005560FAC3C6A8EA9DE13_Core_BID)

          (The payload hash is B8FE9A4EFE381A059022C003813A682A)
```

Quadro 11 - RMA enviando mensagem para AAA

5.3 Access Point Agent (APA)

Podemos verificar também a inicialização do APA na arquitetura NovaGenesis conforme mostrado abaixo.

```
(The unique legible name is APA01)
(The I/O path is /home/lucio/workspace/novagenesis/IO/APA01/)
(The Socket is tcp://10.0.185.162:5555)
(The process legible name is APA)
(The process self-certifying name is 02E70FC90F0D8D78563C00FE8F0A7ACD_APA_PID)
(The operating system legible name is Linux 3.13.0-24-generic #47-Ubuntu SMP Fri
May 2 23:30:00 UTC 2014)
(The host legible name is lucio-VGN-FW350J x86_64)
(The      operating      system      self-certifying      name      is      =
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
(The host self-certifying name is = 6336892BE50F9DBF08EB39AB7F7C704F_HID)

      (Moving to operational state)

      (NumberOfIPCInputKeys = 1)
      (Adding the InputKey = 10155)
      (NumberOfIPCOutputKeys = 1)
      (Adding the OutputKeys = 10156)
[1]      (IQ Size = 0)
[1]      (IQ Size = 1)
(DSCN = 72C8AE5CAFC8CC6E3E43C5E9A47DE733_DID)
```

Quadro 12 - Inicialização do APA

Considerando que todo o processo de publicação das informações para o APA foi exibido nos registros do RMA, é importante destacar apenas o recebimento da informação pelo APA e consequente envio da informação para *access point*.

```

(Time = 4556.37)
  (NextPeerEvaluationTime = 4558.81)
  (1. Check for PSS awareness.)
    (The APA is already aware of this PSS)
  (2. Check for new peer application tuples)
    (Too early for that. Wait next ng -run --evaluate)
  (3. Show the discovered server App(s))
    (Aware of the Application 0 )
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
    (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
  (4. Check subscriptions)
    (Testing subscription 0)
    (Subscription status is Waiting delivery)
    (The publisher is already known and have the index 0)
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
    (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
    (Testing subscription 1)
    (Subscription status is Processing required)
    (The publisher is already known and have the index 0)
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
    (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
    (Checking the file received from the peer with name APAFile1.txt)
    (File extension = txt)
    (Going to parse the file APAFile1.txt)
    (File extension = txt)
    (Command sent to AP: CHANGE_CHANNEL_3)
    (Status received from AP: OK)

  (Deleting the subscription with index = 1)

```

Quadro 13 - Processamento da mensagem recebida pelo APA

5.4 Python OpenFlow Controller Agent (POXA)

Abaixo, pode-se observar a inicialização do POXA, que ocorre de forma similar aos outros agentes apresentados anteriormente. Ele também se conectará via ZeroMQ ao POX que está na porta 5555 para este exemplo, e em outro *host*.

```

(The unique legible name is POXA01)
(The I/O path is /home/lucio/workspace/novagenesis/IO/POXA01/)
(The Socket is tcp://192.168.60.218:5555)
(The process legible name is POXA)
(The process self-certifying name is 8B48F85F8F0A7ACD189BABAA5A2B1D53_POXA_PID)
(The operating system legible name is Linux 3.13.0-24-generic #47-Ubuntu SMP Fri
May 2 23:30:00 UTC 2014)
(The host legible name is lucio-VGN-FW350J x86_64)
(The operating system self-certifying name is =
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
(The host self-certifying name is = 6336892BE50F9DBF08EB39AB7F7C704F_HID)

(Moving to operational state)

(NumberOfIPCInputKeys = 1)
(Adding the InputKey = 10958)
(NumberOfIPCOutputKeys = 1)
(Adding the OutputKeys = 10959)
[1] (IQ Size = 0)
[1] (IQ Size = 1)
(DSCN = 72C8AE5CAFC8CC6E3E43C5E9A47DE733_DID)
(Connecting to the controlled device with the socket tcp://192.168.60.218:5555)

```

Quadro 14 - Inicialização do POXA

O quadro abaixo demonstra registros do RMA, mas exibe processo de publicação e notificação do comando para o “POXA01”, que neste caso é o nome único legível do POXA na rede testada.

```

[2]      (Processing)

      (_ULName = POXA01)
      (Peer = 8B48F85F8F0A7ACD189BABAA5A2B1D53_POXA_PID)
      (PeerIndex = 4)
      (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
      (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
      (PID = 8B48F85F8F0A7ACD189BABAA5A2B1D53_POXA_PID)
      (BID = A6DB57CD7F0005560FAC3C6A8EA9DE13_Core_BID)
      (File = POXAfile1.txt)
      (Path = /home/lucio/workspace/novagenesis/IO/RMA/)

      (The payload hash is F3D7692F6D4418D45B458E54083574FC)
      (The following message was published to the peer)
(
ng -msg --cl 0.1 [ < 1 string 72C8AE5CAFC8CC6E3E43C5E9A47DE733_DID > < 4 string
6336892BE50F9DBF08EB39AB7F7C704F_HID          5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID
89F30CF99F75D2BB209B74D489D28661_RMA_PID    2A5D623904EFA29288214E33EC865B5B_Core_BID
> < 4 string 6336892BE50F9DBF08EB39AB7F7C704F_HID
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID      1A13C48D59444355A920FC69D417DE72_PSS_PID
CB897B7E118CFABBD55A93C37F14E7D6_PS_BID > ]
ng -pub --notify 0.1 [ < 1 string 18 > < 1 string F3D7692F6D4418D45B458E54083574FC
> < 1 string POXAfile1.txt > < 5 string pub 6336892BE50F9DBF08EB39AB7F7C704F_HID
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID      8B48F85F8F0A7ACD189BABAA5A2B1D53_POXA_PID
A6DB57CD7F0005560FAC3C6A8EA9DE13_Core_BID > ]
ng -info --payload 0.1 [ < 1 string POXAfile1.txt > ]
ng -pub --bind 0.1 [ < 1 string 2 > < 1 string F3D7692F6D4418D45B458E54083574FC > <
1 string 2A5D623904EFA29288214E33EC865B5B_Core_BID > ]
ng -pub --bind 0.1 [ < 1 string 2 > < 1 string F3D7692F6D4418D45B458E54083574FC > <
1 string 89F30CF99F75D2BB209B74D489D28661_RMA_PID > ]
ng -pub --bind 0.1 [ < 1 string 2 > < 1 string F3D7692F6D4418D45B458E54083574FC > <
1 string 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID > ]
ng -pub --bind 0.1 [ < 1 string 9 > < 1 string F3D7692F6D4418D45B458E54083574FC > <
1 string 6336892BE50F9DBF08EB39AB7F7C704F_HID > ]
ng -message --type 0.1 [ < 1 string 1 > ]
ng -message --seq 0.1 [ < 1 string 344 > ]
ng -scn --seq 0.1 [ < 1 string 6097DB94307E59C261BAC9C2852677B3 > ]

There is a payload of 14 bytes

```

Quadro 15 - RMA enviando mensagem para POXA específico

Por último, podemos constatar que a mensagem é recebida pelo POXA, tratada e enviada ao POX.


```

[2]      (Processing)

(Time = 5034.59)
(NextPeerEvaluationTime = 5034.89)
(1. Check for PSS awareness.)
    (Aware of a PSS on Categories 2 and 9)
    (The POXA is already aware of this PSS)
(2. Check for new peer application tuples)
    (Too early for that. Wait next ng -run --evaluate)
(3. Show the discovered server App(s))
    (Aware of the Application 0 )
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
    (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
(4. Check subscriptions)
    (Testing subscription 0)
    (Subscription status is Processing required)
    (The publisher is already known and have the index 0)
    (HID = 6336892BE50F9DBF08EB39AB7F7C704F_HID)
    (OSID = 5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB_OSID)
    (PID = 89F30CF99F75D2BB209B74D489D28661_RMA_PID)
    (BID = 2A5D623904EFA29288214E33EC865B5B_Core_BID)
    (Checking the file received from the peer with name POXAFile1.txt)
    (File extension = txt)
    (Command sent to OpenFlow:

◆wf
-----
1)      (Status received from OpenFlow: )
        (Deleting the subscription with index = 0)

[3]      (Finished)

```

Quadro 16 - Tratamento da mensagem recebida pelo POXA

Para confirmação, podemos testar no equipamento com OpenFlow a existência dos fluxos. Neste primeiro quadro, observa-se que não há fluxos criados.

```

# dpctl dump-flows tcp:127.0.0.1
stats_reply (xid=0x25156283): flags=none type=1(flow)

```

Quadro 17 - Verificação da não existência de fluxos no OpenFlow

Aqui, o recebimento pelo POX do comando enviado pelo POXA e conseqüentemente execução do mesmo.

```

no message
Wait...
\x01\x88\x77\x66\x03\x04\x05\x00\x01\x02\x03\x04\x06\x31
DEBUG:ng.pox_agent:installing flow

```

Quadro 18 - Registro de criação do fluxo pelo OpenFlow

Por último, a execução do comando novamente para verificar a existência do fluxo instalado. Ou seja, a NovaGenesis através do POXA cria fluxos de dados no equipamento OF. Esse recurso permite que os serviços NovaGenesis contratem fluxos de dados nos *access*

points da rede. Esses fluxos podem ser priorizados ou agrupados de acordo com as necessidades dos serviços NovaGenesis. Tudo isso é orquestrado via contratos.

```
# dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0xc5da1d53): flags=none type=1(flow)
cookie=0, duration_sec=125s,
duration_nsec=649000000s,table_id=1,priority=32768,
n_packets=0,n_bytes=0,idle_timeout=0,
hard_timeout=0,d1_src=00:01:02:03:04:06,
d1_dst=88:77:66:03:04:05,actions=output:1
```

Quadro 19 - Verificação do fluxo instalado no OpenFlow

5.5 Hash Table Service (HTS) do domínio

No próximo quadro, é exibido o conteúdo da tabela *hash* (HT) do serviço HTS. A saída diz respeito a execução do teste de prova de conceito, onde pode-se observar os relacionamentos através de NBs propostos na arquitetura NovaGenesis, bem como os nomes únicos registrados pelos agentes.


```

[2]      (Processing)

          -msg --cl 0.1
          -list --bind 0.1

Category 0:

DHT      7B867C00390982245BC8F0DA684AB103
GW       57C6DB0F19D24E3D4DF1B86EEDEBA60D
OS       FEEDEF59032422153095D4A6D9B735D2
lucio-VGN-FW350J x86_64      6336892BE50F9DBF08EB39AB7F7C704F
HTS     5E7919DC83A058583D59A38ADD3AB859
Domain   7856D6FFD2A83BCDA8A2B7297CB3FC2D

Category 1:

F608F9D7C83883CE40753B719D916E68      Wireless
30FDD8F23F3C9C1412F20A1D82283324      RMA
B3B19327CC030529D4DA48DC5522C60C      Proxy
B2E6B38C7DD37F775A807447D9057D33      Point
649D188CFC82D91083A85E6DD7D6CC4B      APA
3154407283A02267BDC456C2FAFCE29E      POXA01
6336892BE50F9DBF08EB39AB7F7C704F      lucio-VGN-FW350J x86_64
8163DB8CEBF9D263E6041739EF602C6E      SSA
3B2090FDE82BF20EC74D536473CF3EF8      HT
57C6DB0F19D24E3D4DF1B86EEDEBA60D      GW
90C821C655A04503DAB857CBF05E1117      Wi-Fi
B743DE8A492DE389BDAA6AAFC049D9A3      Active
5E7919DC83A058583D59A38ADD3AB859      HTS
7B867C00390982245BC8F0DA684AB103      DHT
31B2E529273E77B588479D46D88DF0FD      APA01
5BE4B27DBEDF6F0EB6B9F3C1FE84A7AB      Linux 3.13.0-24-generic #47-Ubuntu SMP Fri
May 2 23:30:00 UTC 2014
2EA279D1897E057432FF262FDB6F2AF5      Resource
7856D6FFD2A83BCDA8A2B7297CB3FC2D      Domain
C46EBE682BE0FA90209B1C0DF204AB27      APA02
1B0ED435EC540B8B417CD5799E20DA7A      Controller
C60DEFEB77FB393B0CDB2CDCB0A0338B      AAA
5FC18DA8275D35020E62B85F028619EC      Spectrum
DD757C7DCC5BF228FC8C39C1F1C92CF6      Manager
FEEDEF59032422153095D4A6D9B735D2      OS
C94C13D84F41E49C768347D4E11E28F4      SSA01
56BA903896310C76076A6C6D26E61785      Management
5D609337542D4845409DD081A2317870      Sensing
BD56B036546A5EA50B8B89F6D077BF55      Agent
79F3434AB6F2D2CA28DFB604F73D4773      Core
5F157CF476FD3C9DDB653F6B8654E617      RMA01
425ABBE33677A3E7F270A9CCC18786F1      Process
6849EF1606556D9CF32893626881C37C      OpenFlow
9FD3AC3BBFEC994AB22C61EEE3376D7D      POX
586E349F22E8038F582D2FFD21C45071      POXA
02039DF41DCE7E440684F72F347446FB      Access
885A037B996A8FDC978FE3B05896648C      Antenna

Category 18:

ED0F2DFFEEA29597C54C0E9E18FDC7C0      SSAFile1.txt
F3D7692F6D4418D45B458E54083574FC      POXAFile1.txt
B8FE9A4EFE381A059022C003813A682A      AAAFile1.txt
F249E533E5E7CF78E0AB857F7B7E5F8F      APAFile1.txt
-scen --seq 0.1

[3]      (Finished)

```

Quadro 20 - Extrato da HTS do domínio

Na categoria 1 são armazenados todos os relacionamentos entre os SCNs e o nomes em linguagem natural (Português). Observe que estão armazenados os nomes dos serviços, ULN de alguns deles e funções ou característica que eles possuem. Da mesma forma, na categoria 18 ficam as informações publicadas no domínio NovaGenesis.

Capítulo 6. Conclusões

Este trabalho teve como objetivo principal desenvolver um conjunto de quatro novos serviços NovaGenesis que são capazes de realizar a exposição de equipamentos de substrato (físicos) à NG, tais como *access points* e analisadores de espectro, além de controladores de rede definida por *software* OpenFlow. Estes serviços também permitem operar ou se comunicar com os dispositivos para realizar alterações de configuração ou controle, tais como a mudança de canais de operação em redes Wi-Fi e a criação ou mudança de fluxo de dados nos *switches*.

Foi desenvolvido um serviço capaz de tomar decisões de alocação dinâmica de recursos, o RMA, fundamental para o próprio *core* da NovaGenesis, orquestrando as tomadas de decisão para os serviços que compõem a nova arquitetura.

No Capítulo 1 foi apresentada uma introdução sobre os problemas e deficiências da arquitetura atual de Internet e as buscas que têm sido realizadas para resolver essas demandas. No Capítulo 2 foram apresentadas as tecnologias e conceitos utilizados, em especial, o conceito de Internet do Futuro, que é a motivação fundamental para este trabalho e a NovaGenesis. Também foram apresentados os conceitos de Ethernet, Wi-Fi, SDN e de transporte de mensagem que são fundamentais para compreender as inovações realizadas nos serviços desenvolvidos.

No Capítulo 3 foi descrita a NovaGenesis, como proposta para nova arquitetura de Internet, apresentando de forma sucinta cada serviço constante da implementação atual e a sua filosofia básica. No Capítulo 4, foram apresentados os novos agentes desenvolvidos para compor a arquitetura NovaGenesis. Todos os conceitos e serviços apresentados no Capítulo 3 estão presentes e incorporados em cada serviço desenvolvido.

O Capítulo 5 demonstrou a implementação dos agentes, com logs de mensagens trocadas entre os elementos implementados, permitindo a confirmação do funcionamento dos processos e troca de informações definidas. Com isso, os serviços desenvolvidos tornam-se aptos a serem utilizados para futuras implementações da NovaGenesis, respondendo bem ao objetivo que lhes foi proposto. O SSA é um representante de um analisador de espectro na NovaGenesis, de tal forma que as informações do espectro eletromagnético fornecidas por ele podem ser utilizadas para tomar decisões de uso de canais de comunicação. Tarefa desenvolvida pelo RMA, que cataloga e armazena informações de uso dos canais Wi-Fi (embora outras informações possam ser correlacionadas) para utilizar em seus algoritmos de

decisão, a fim de estabelecer melhorias ou adequar a prestação de serviços de acordo com os contratos estabelecidos. Também foram implementados o APA e o POXA, que além de representar os dispositivos e elementos, são capazes de interagir com eles para executar tarefas e funções de controle e configuração de rede.

O trabalho também contribuiu para o estabelecimento, incorporação e implementação do protocolo de transporte de mensagens *ZeroMQ* na NovaGenesis. Dessa forma, é possível manter comunicação com sistemas operacionais, programas e *hardwares* que operam utilizando inúmeros modos de funcionamento disponíveis no *ZeroMQ*. Com essa contribuição, o *core* da NG é capaz de se comunicar via *ZeroMQ* com qualquer elemento externo à arquitetura.

Outra contribuição foi o desenvolvimento do conceito de nomes únicos legíveis que permitem que várias instâncias de um serviço NovaGenesis possam ser executadas em um mesmo ambiente (ou *host*) e possam ser localizadas por um identificador em linguagem natural.

Uma possibilidade de trabalho futuro é a implementação de um algoritmo inteligente, baseado possivelmente em redes neurais para a tomada de decisões do cenário apresentado neste trabalho e em outros futuros. Desta forma, o algoritmo pode se tornar eficaz e eficiente a cada nova execução, pois poderá fazer uso dos aprendizados e históricos da rede.

Seguindo a ideia já descrita de permitir que o RMA orquestre elementos da rede NG, é objetivo permitir que o RMA seja o executor da rede, ou seja, um *start* da NovaGenesis, fazendo com que a rede emerja a partir dele, sendo responsável pela inicialização do *core* e dos serviços básicos que compõem a arquitetura.

Como proposta de trabalhos futuros a serem desenvolvidos com base nos serviços desenvolvidos nesta dissertação, estão a implementação totalmente do cenário proposto em [6] em conjunto com o laboratório WOCA, do Inatel, na arquitetura NovaGenesis, permitindo que a mesma controle e gerencie todos os aspectos presentes na proposta. Abaixo, uma proposta de cenário utilizando várias tecnologias de comunicação, como rádio frequência, rádio sobre fibra, fibra e ethernet, além de redes sem-fio utilizando os agentes da NovaGenesis como elementos da rede.

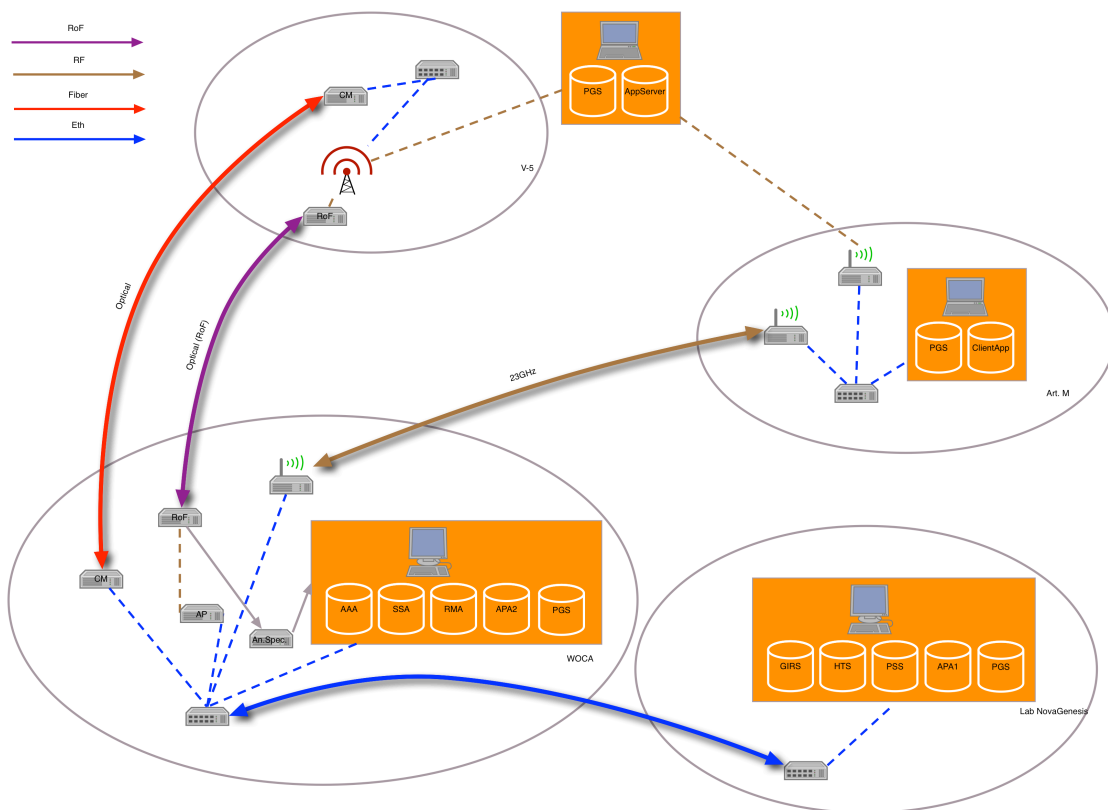


Figura 11 - Cenário de implementação da arquitetura NG proposta em [6]

Todos os serviços básicos NG que suportarão tal cenário foram desenvolvidos, com exceção de um serviço que controle a alteração da banda de operação de uma antena, cujo nome proposto é *Active Antenna Agent* (AAA), embora a aplicação básica com o *core* já tenha sido desenvolvida.

O APA, SSA e POXA possuem muitas características e códigos similares que podem ser unificados e condensados em um representante genérico, capaz de identificar o tipo de dispositivo e as funcionalidades que ele possui e controles que pode executar. Dessa forma, um mesmo serviço pode se adaptar para representar diferentes tipos de equipamentos.

Além das possibilidades descritas acima, uma implementação futura no ZeroMQ pode permitir que ele se comunique diretamente com a NG, eliminando assim a necessidade do TCP/IP neste processo, e habilita a NovaGenesis em um *framework* estável, reconhecido e utilizado em diversas implementações em sistemas atuais.

Referências

- [1] JAIN, Raj. Internet 3.0: ten problems with current internet architecture and solutions for the next generation. In: **Military Communications Conference, 2006. MILCOM 2006. IEEE**. IEEE, 2006. p. 1-9.
- [2] Cisco Visual Networking Index – White Paper. [Online] http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html. Acessado em 20 de julho de 2015.
- [3] FELDMANN, Anja. Internet clean-slate design: what and why?. **ACM SIGCOMM Computer Communication Review**, v. 37, n. 3, p. 59-64, 2007.
- [4] NovaGenesis Project. [Online] <http://www.inatel.br/novagenesis>. Acessado em 20 de maio de 2015.
- [5] ORTIZ, Antonio M. et al. The cluster between internet of things and social networks: Review and research challenges. **Internet of Things Journal, IEEE**, v. 1, n. 3, p. 206-215, 2014.
- [6] RAIMUNDO-NETO, E. et al. Implementation of an Optical-Wireless Network with Spectrum Sensing and Dynamic Resource Allocation Using Optically Controlled Reconfigurable Antennas. **International Journal of Antennas and Propagation**, v. 2014, 2014.
- [7] BRAUN, Wolfgang; MENTH, Michael. Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices. **Future Internet**, v. 6, n. 2, p. 302-336, 2014.
- [8] ROSEN, Michael et al. **Applied SOA: service-oriented architecture and design strategies**. John Wiley & Sons, 2012.
- [9] MONSANTO, Christopher et al. Composing Software Defined Networks. In: **NSDI**. 2013. p. 1-13.

- [10] ALBERTI, Antonio M. et al. A NovaGenesis proxy/gateway/controller for OpenFlow software defined networks. In: **Network and Service Management (CNSM), 2014 10th International Conference on**. IEEE, 2014. p. 394-399.
- [11] ALBERTI, Antonio M. et al. Towards An Opportunistic, Socially-Driven, Self-Organizing, Cloud Networking Architecture with NovaGenesis.
- [12] GRIFFIN, David et al. Service-Centric Networking. **Handbook of Research on Redesigning the Future of Internet Architectures**, p. 1-26, 2015.
- [13] MORENO-VOZMEDIANO, Rafael; MONTERO, Rubén S.; LLORENTE, Ignacio M. Key challenges in cloud computing: Enabling the future internet of services. **Internet Computing, IEEE**, v. 17, n. 4, p. 18-25, 2013.
- [14] REXFORD, Jennifer; DOVROLIS, Constantine. Future Internet architecture: clean-slate versus evolutionary research. **Communications of the ACM**, v. 53, n. 9, p. 36-40, 2010.
- [15] SHENKER, Scott. Fundamental design issues for the future Internet. **Selected Areas in Communications, IEEE Journal on**, v. 13, n. 7, p. 1176-1188, 1995.
- [16] BOLLA, Raffaele et al. Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures. **Communications Surveys & Tutorials, IEEE**, v. 13, n. 2, p. 223-244, 2011.
- [17] DOROGOVTSEV, Sergei N.; MENDES, José FF. **Evolution of networks: From biological nets to the Internet and WWW**. Oxford University Press, 2013.
- [18] ALBERTI, Antonio M.; SINGH, Dhananjay. Developing a NovaGenesis architecture model for service oriented future Internet and IoT: An advanced transportation system scenario. In: **Internet of Things (WF-IoT), 2014 IEEE World Forum on**. IEEE, 2014. p. 359-364.

- [19] SUNSHINE, Carl A. (Ed.). **Computer Network Architectures and Protocols**. Springer Science & Business Media, 2013.
- [20] FIBRE Project. [Online] <http://fibre-ict.eu/>. Acessado em 12 de fevereiro de 2015.
- [21] SAIL Project. [Online] <http://www.sail-project.eu/>. Acessado em 12 de fevereiro de 2015.
- [22] XIA Project. [Online] <https://www.cs.cmu.edu/~xia/>. Acessado em 12 de fevereiro de 2015.
- [23] HARAI, H. et al. **New generation network architecture akari conceptual design**. Tech. Rep. v1, NICT, 2010.
- [24] METCALFE, Robert M.; BOGGS, David R. Ethernet: distributed packet switching for local computer networks. **Communications of the ACM**, v. 19, n. 7, p. 395-404, 1976.
- [25] FOROUZAN, Behrouz A.; MOSHARRAF, Firouz. **Redes de computadores: uma abordagem top-down**. AMGH, 2013.
- [26] TANENBAUM, Andrew S. **Redes de computadores**. Pearson Educação, 2003.
- [27] PERAHIA, Eldad; STACEY, Robert. **Next Generation Wireless LANS: 802.11 n and 802.11 ac**. Cambridge university press, 2013.
- [28] SEZER, Sakir et al. Are we ready for SDN? Implementation challenges for software-defined networks. **Communications Magazine, IEEE**, v. 51, n. 7, p. 36-43, 2013.
- [29] TANENBAUM, Andrew S.; VAN STEEN, Maarten. **Distributed systems**. Prentice-Hall, 2007.
- [30] Documentação do ZeroMQ. [Online] <http://zguide.zeromq.org/page:all>. Acessado em 17 de março de 2015.

- [31] DWORAK, Andrzej et al. Middleware trends and market leaders 2011. In: **Conf. Proc.** 2011. p. FRBHMULT05.
- [32] BERNERS-LEE, Tim et al. The semantic web. **Scientific american**, v. 284, n. 5, p. 28-37, 2001.
- [33] DANNEWITZ, Christian et al. Network of Information (NetInf)—An information-centric networking architecture. **Computer Communications**, v. 36, n. 7, p. 721-735, 2013.
- [34] CARROLL, John M. **Computer security**. Butterworth-Heinemann, 2014.
- [35] HWANG, Kai; DONGARRA, Jack; FOX, Geoffrey C. **Distributed and cloud computing: from parallel processing to the internet of things**. Morgan Kaufmann, 2013.
- [36] ZIMMERMANN, Hubert. OSI reference model--The ISO model of architecture for open systems interconnection. **Communications, IEEE Transactions on**, v. 28, n. 4, p. 425-432, 1980.
- [37] SOUZA, MSc Hugo Vieira L. Redes de computadores. **Editora Campus**, 2014.
- [38] ESPOSITO, Christian; COTRONEO, Domenico; RUSSO, Stefano. On reliability in publish/subscribe services. **Computer Networks**, v. 57, n. 5, p. 1318-1343, 2013.
- [39] Documentação Python. [Online] <https://www.python.org/doc/>. Acessado em 7 de abril de 2015.
- [40] Unix timestamp. [Online] <http://cm.bell-labs.com/cm/cs/who/dmr/pdfs/man22.pdf>. Acessado em 7 de abril de 2015.
- [41] Documentação OpenWrt. [Online] <https://openwrt.org/>. Acessado em 7 de abril de 2015.

[42] Especificação do SSH. [Online] <https://tools.ietf.org/html/rfc4252>. Acessado em 7 de abril de 2015.

[43] POX [Online] <https://github.com/noxrepo/pox> . Acessado em 7 de abril de 2015.