Dissertação de Mestrado

Inatel
Instituto Nacional de Telecomunicações

ESPECIFICAÇÃO DE ARQUITETURA DE GERENCIAMENTO E CONTROLE PARA INTERNET DO FUTURO: UMA ABORDAGEM NO ESCOPO DA NOVAGENESIS

Isabela Vasconcelos de Carvalho Motta

Outubro de 2016

Especificação de Arquitetura de Gerenciamento e Controle para Internet do Futuro: Uma Abordagem no Escopo da NovaGenesis

Isabela Vasconcelos de Carvalho Motta

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

2016

Motta, Isabela Vasconcelos de Carvalho
M921e

Especificação de arquitetura de gerenciamento e controle para internet do futuro: uma abordagem no escopo da novagenesis. / Isabela Vasconcelos de Carvalho Motta. – Santa Rita do Sapucaí, 2016.
100 p.

Orientador: Prof. Dr. MSc. Antônio Marcos Alberti.
Dissertação de Mestrado em Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.
Inclui bibliografia e anexo.

1. Internet do Futuro 2. NovaGenesis 3. Gerência de Redes 4. Controle de Rede 5. Novas Arquiteturas 6. Mestrado em 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. Carlos Roberto dos Santos

INATEL

_____

Dr. Mateus Augusto Silva Santos

Ericsson

_____

Prof. Dr. José Marcos Câmara Brito

Coordenador do Curso de Mestrado

Ao Fernando,

que nasceu durante este trabalho e dividiu seu colo com os livros.

Agradecimentos

Ao professor Antônio Marcos Alberti pela excelente orientação fornecida durante a elaboração deste trabalho.

Aos meus pais Ronaldo e Raquel, Tatiana, Bernardo, Daltinho, Marcela e Fernando por tudo.

Ao INATEL, a FAPEMIG e ao Centro de Referência em Radiocomunicações (CRR) pelo apoio concedido.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AAA	*Active Antenna Agent*
ANA	Agência Nacional de Águas
APA	*Access Point Agent*
API	*Application Program Interfaces*
ASCII	*American Standard Code for Information Interchange*
BLE	*Bluetooth Low Energy*
CaaS	*Controller-as-a-Service*
CCN	*Content Centric Networking*
CDAP	*Common Application Protocol*
CEP-ID	*Connection End-Point Identifier*
CKAN	*Comprehensive Knowledge Network*
CLI	*Command Line Interface*
CMIP	*Common Management Information Protocol*
CONGEP	*COmmom NovaGEnesis Protocol*
CSCF	*Call Section Control Function*
DAF	*Distributed Application Facility*
DHT	*Distributed Hash Table*
DIF	*Distributed IPC Facility*
DiffServ	*Differentiated services*
DNS	*Domain Name System*
DoS	*Deny of Service*
D-SDN	*Decentralize*-SDN
ETSI	*European Telecommunications Standards Institute*
FAI	Faculdade de Administração e Informática
FCAPS	*Fault, Configuration, Accounting, Performance and Security*
FI	*Future Internet*
FIA	*Future Internet Architecture*
FIA	*Future Internet Assembly*
FIB	*Forwarding Information Base*
FIND	*Future Internet Design*

GE	*Generic Enabler*
GENI	*Global Environment for Network Innovations*
GIRS	*Generic Indirection Resolution Service*
GUI	*Graphical User Interface*
HetNet	*Heterogeneous Network*
HTS	*Hash Table Service*
HTTP	*Hypertext Transfer Protocol*
IaaS	*Infrastructure-as-a-Service*
ICANN	*Internet Corporation for Assigned Names and Numbers*
ICMP	*Internet Control Message Protocol*
ICN	*Information Centric Networking*
ICT Lab	*Information and Communication Technologies Laboratory*
IETF	*Internet Engineering Task* Force
IEEE	*Institute of Electrical and Electronics Engineers*
IMS	*IP Multimedia Subsystem*
INATEL	Instituto Nacional de Telecomunicações
INM	*In-Network Management*
IoT	*Internet of Things*
IP	*Internet Protocol*
IPC	*Inter Process Communication*
IPFIX	*IP Flow Information eXport*
IPTV	*IP Television*
ISO	*International Organization for Standardization*
ITU	*International Telecommunication Union*
MANO	*Management and Orchestrator*
MeS	*Mediator Service*
MG	*Media Gateways*
MGC	*Media Gateway Controller*
MGCP	*Media Gateway Control Protocol*
MIB	*Management Information Base*
MIT	*Massachusetts Institute of Technology*
M2M	Máquina para Máquina

NAT	*network address translation*
NB	*Name Binding*
NDN	*Named Data Networking*
NDO	*Named Data Objects*
NetInf	*Network of Information*
NFV	*Network Function Virtualization*
NFVO	*Network Function Virtualization Orchestration*
NG	*NovaGenesis*
NGN	*Next Generation Network*
NGSI	*Next Generation Standard Interface*
NLN	*Natural Language Name*
NMS	*Network Management Systems*
NONM	*Name-Oriented Network Management*
NRS	*Name Resolution Service*
OAM	*Operation Administration and Maintenance*
OFNIC	*OpenFlow Network Information and Control*
OPEX	*Operational Expenditure*
OSI	*Open Systems Interconnection*
OSPF	*Open Shortest Path First*
PaaS	*Platform-as-a-Service*
PAN	*Personal Area Networks*
PARC	*Palo Alto Research Center*
PDF	*Policy Decision Function*
PGC	*Proxy-Gateway Controller*
PGCMA	*Proxy-Gateway Controller and Management Agent*
PIaaS	*Protocol Implemented as a Service*
PIT	*Pending Interest Table*
PKI	*Public Key Infraestructure*
POXA	POX *Agent*
PSS	*Publish/Subscribe Service*
QoE	*Quality of Experience*
QoS	*Quality of Service*

RACF	*Resource Admission Control Function*
RFC	*Request for Comments*
RFID	*Radio-Frequency IDentification*
RIB	*Resource Information Base*
RINA	*Recursive InterNetwork Architecture*
RIP	*Routing Information Protocol*
RMA	*Resource Management Agent*
RTCP	*Real-Time Control Protocol*
RTP	*Real-Time Protocol*
SaaS	*Software-as-a-Service*
SDN	*Software Defined Networking*
SIP	*Session Initiation Protocol*
SLA	*Service Level Agreement*
SMI	*Structure of Management Information*
SNMP	*Simple Network Management Protocol*
SOA	*Service Oriented Architecture*
SSA	*Spectrum Sensing Agent*
SSL	*Secure Socket Layer*
SVN	*Self-Verifying Names*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
URI	*Uniform Resource Identifier*
USB	*Universal Serial Bus*
VIM	*Virtualized Infrastructure Manager*
VLAN	*Virtual Local Area Network*
VNF	*Virtual Network Functions*
VNFM	*Virtual Network Function Manager*
VPN	*Virtual Private Network*
WOCA	*Wireless and Optical Convergent Access*
XML	*eXtensible Markup Language*

Publicações

MOTTA, I.V.C.; ALBERTI, A.M.. Gerência e Controle como um Serviço **XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (WPEIF)**. Junho de 2016, Salvador, Bahia, Brasil.

MOTTA, I.V.C.; ALBERTI, A.M.. Integrated Management and Control Architecture For New Generation Networks **XXXIV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais**. Agosto de 2016, Santarém, Pará, Brasil.

Resumo

As mudanças de paradigmas que estão surgindo para a Internet do Futuro (*Future Internet* - FI) juntamente com as escalas cada vez maiores no número de dispositivos, conectividade e interatividade estão desafiando o modelo tradicional de gestão e controle de equipamentos de rede (roteadores, comutadores, *gateways*, sensores, etc.) e de computação (servidores, *proxies*, controladores, funções virtuais, etc.).

Assim, é necessário reexaminar os modelos atuais de controle e gerenciamento de redes e computação em nuvem sob a ótica das propostas que emergem em FI, tais como *Internet of Things* (IoT), *Software Defined Networking* (SDN), *Network Function Virtualization* (NFV), *Content Centric Networking* (CCN) e outras arquiteturas para Internet do Futuro.

Este trabalho revisa as atuais práticas de controle e gerenciamento e discute suas limitações em cenários emergentes. Também propõe um novo modelo de controle e gerenciamento no contexto da NovaGenesis, uma proposta de FI. É fornecida uma comparação qualitativa correlacionando os modelos já estabelecidos com o modelo proposto para a iniciativa NovaGenesis.

Palavras-chave: Internet do Futuro, NovaGenesis, Gerência de Rede, Controle de Rede, Novas Arquiteturas

Abstract

The paradigms shifts that are emerging in the Future Internet (FI), coupled with the increasingly larger scales on number of devices, connectivity, and interactivity are challenging the traditional control and management model for devices (routers, switches, gateways, sensors, etc) and for computation (servers, proxies, controllers, virtual functions, etc).

Thus, it is necessary to re-examine the current control/management models under the optics of the FI proposals that relay in some emerging technologies, including Internet of Things (IoT), Software Defined Networking (SDN), Network Function Virtualization (NFV), Content Centric Networking (CCN) and other FI concepts.

This work reviews the current control/management practices and discusses their limitations on emerging scenarios. It also proposes a new control/management model in the context of a FI proposal called NovaGenesis. A qualitative comparison is provided, correlating the already established models with our initiative.

Keywords: Future Internet, NovaGenesis, Network Management, Network Control, New Architectures

1 Introdução

1.1 Contextualização

A operação de redes de telecomunicações e de nuvens computacionais (*clouds*) engloba todo o controle e gerenciamento de equipamentos dessas infraestruturas. Ações de controle são realizadas intensamente para manter a rede funcionando de forma adequada às necessidades dos usuários. Ações de gerenciamento são executadas para preservar a operação estável da rede, mantendo sua saúde em um longo prazo. Portanto, a principal diferença entre as duas funções (controle e gerência) pode ser vista como a escala de tempo dos eventos (ou ações) realizados junto à rede conforme ilustra a Figura 1.

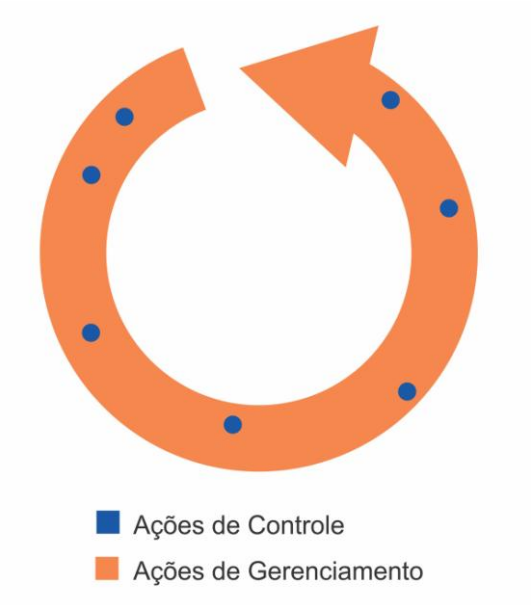

Figura 1 – *Frequências das ações de controle e gerenciamento de rede.*

O controle e o gerenciamento de dispositivos são duas das grandes dificuldades na atual arquitetura da Internet (1). Algumas das preocupações mais importantes são:

- **Implantação incremental e heterogênea de protocolos de gerenciamento e controle**: A Internet tornou-se popular muito rapidamente quando comparado às outras redes de comunicações, como por exemplo, Rádio e Televisão, exigindo incrementos constantes à medida que novas tecnologias foram surgindo. Esse rápido crescimento causou o aparecimento de um enorme número de protocolos adicionados à arquitetura. Com o surgimento de vários *softwares* proprietários e desenvolvimentos fechados de dispositivos de rede, tornou-se extremamente difícil desenvolver e implantar abordagens inovadoras no âmbito de gerência e controle de rede. Neste contexto, diversos modelos foram criados para gerenciar essas redes multi-protocolos. A integração das principais funções de gerenciamento é importante e não deve ser vista como uma simples adição de recursos extras na rede.

- **Escalabilidade e interoperabilidade dos protocolos de gerenciamento e controle:** O grande número de equipamentos de rede tornou a tarefa de controlar ou gerir a infraestrutura de comunicações cada vez mais complexa. Conectividade universal é um objetivo comum entre as redes de computadores. É necessário tornar as redes mais fáceis de serem gerenciadas e interoperadas de forma satisfatória. A falta de interoperabilidade entre os modelos atuais de gerenciamento e controle cria uma barreira que impede a gestão eficaz e eficiente das redes. O problema da interoperabilidade pode resultar em incompatibilidades entre modelos diferentes de gerência. Problemas de interoperabilidade estão intimamente relacionados à falta de transparência na rede, as diferentes definições para *Management Information Base* (MIB) e *Structure of Management Information* (SMI) e mapeamentos incompletos para traduções de protocolos (2).

- **Exposição de *hardware* para *software*:** Fornecedores de equipamentos de rede e nuvem normalmente expõem seus dispositivos para *software* proprietários, dificultando o controle e gerência integrados. A orquestração conjunta de *hardware* e *software* para diferentes fornecedores ainda está em fase inicial. A exposição aberta e homogênea de *hardware* pode facilitar a descoberta da rede e permitir o controle e gerenciamento simultâneos para múltiplas redes e para diferentes aplicações.

- **Excessiva interferência humana:** O atual gerenciamento de rede tem sido dependente de seres humanos, tornando-se demorado, caro e suscetível a falhas. Redes heterogêneas (*Heterogeneous Networks* - HetNets) demandam excessiva interferência humana. Cada nova tecnologia de rede tem um requisito de gerência específica que exige maior conhecimento devido ao aumento da diversidade e escala. Redes com múltiplas tecnologias tornam o gerenciamento e controle mais complexos e diminuem a eficiência nas atividades de Operação, Administração e Manutenção (OAM).

Motivados por essas deficiências e muitas outras, por exemplo: nomeação; mobilidade; segurança; privacidade; distribuição de conteúdo; etc.; dezenas de projetos em todo o mundo buscam redesenhar os protocolos de comunicação e a estrutura da Internet. Essas iniciativas são coletivamente chamadas de *Future Internet* (FI) (3). Como a Internet foi aberta ao público em geral e começou a ser usada para uma crescente diversidade de aplicações, um complexo aglomerado de soluções remendadas foi desenvolvido para alargar o seu âmbito, criando algumas inconsistências que agora começaram a serem questionadas, incluindo aspectos de controle e gerência. Assim, as funções e escalas de Internet mudaram consideravelmente de seus propósitos originais (4). Muitos pesquisadores começaram a pensar se a pilha atual da Internet pode suportar ou não os multifacetados crescimentos exponenciais que estamos vivenciando no número de dispositivos, mobilidade, interatividade, conteúdo e tráfego.

O termo *Future Internet Architecture* (FIA) (5) (6) foi cunhado por algumas iniciativas pioneiras, que incluem o projeto 4D (7) *Future Internet Design* (FIND) (8) *Global Environment for Network Innovations* (GENI) (9) e o Europeu *Future Internet Assembly* (FIA) (10).

Em 2008, o Instituto Nacional de Telecomunicações (Inatel) começou sua própria FIA, com o projeto chamado NovaGenesis (www.inatel.br/novagenesis/), que é uma arquitetura de informação convergente (folha em branco ou em Inglês *clean slate*) que pode ser aplicada universalmente. O projeto NovaGenesis não engloba somente troca e distribuição de conteúdo, Internet, transporte de dados, tecnologias como Internet das Coisas (*Internet of Things* – IoT) (11) e Redes Centradas em Informação (*Information Centric Networking* - ICN (12)), mas também processamento e armazenamento de conteúdo, serviços e aplicações de computação em nuvem (13).

Neste contexto, é imperativo identificar e compreender as angústias e limitações dos modelos de gerenciamento e controle na rede atual e em propostas FIAs. Este trabalho compromete-se a tal tarefa, dando origem aos requisitos e especificações para uma arquitetura de gerenciamento e controle na NovaGenesis, abrangendo computação e infraestrutura de rede.

1.2 Motivação

Como deve ser o gerenciamento e controle em uma FIA como a NovaGenesis? Este trabalho analisa as arquiteturas existentes de controle e gerenciamento atuais, fazendo um paralelo entre: (i) a Internet atual; (ii) as abordagens emergentes de gerenciamento e controle, tais como IoT, Redes Centradas em Conteúdos (*Content Centric Networking* – CCN) (12), Redes Definidas por *Software* (*Software Defined Networking* – SDN) (14), Virtualização das Funções da Rede (*Network Function Virtualization* – NFV) (15) (16) e computação em nuvem; (iii) a proposta de gerenciamento e controle para a NovaGenesis. Esta dissertação apresenta um modelo de referência para controle e gerenciamento de redes emergentes que aborda com eficácia requisitos tais como escalabilidade, interoperabilidade, heterogeneidade dos recursos, ciclo de vida de objetos de

informação e serviços e orientação a serviços. Esse modelo é a principal contribuição do trabalho.

1.3 Metodologia Científica

Pode-se classificar este trabalho como de natureza aplicada quanto ao interesse prático na aplicação de seus resultados na resolução de problemas reais. Em relação aos seus objetivos, a presente pesquisa é exploratória, pois visou aprofundar o entendimento de um problema por meio de levantamentos bibliográficos para estabelecer relações entre as tecnologias estudadas e a arquitetura proposta. Sua abordagem é qualitativa, pois os resultados da pesquisa não podem ser mensurados numericamente.

Este trabalho foi concebido em 3 fases: (i) na primeira fase foi escolhido o tema, delimitado o problema e hipóteses da pesquisa; (ii) a segunda fase é referente à execução da pesquisa. Foram realizadas revisões literárias por meio de minucioso levantamento e estudo dos principais artigos científicos que abordam o tema deste trabalho. (iii) e a ultima fase foi dedicada à análise dos dados e informações obtidas na etapa anterior. Nesta fase as ideias foram organizadas de forma sistemática visando à elaboração da proposta final e a análise de viabilidade da solução por meio de estudos de caso.

1.4 Estrutura da Dissertação

Este trabalho começa com uma visão geral dos principais problemas de gerenciamento e controle nas redes atuais. Em seguida, no Capítulo 2, é discutida a arquitetura de gerenciamento e controle utilizada na maior parte das implementações em uso, confrontando a utilização em tecnologias para FI. Também são abordadas as tecnologias emergentes para FI. No Capítulo 3 os conceitos da NovaGenesis são apresentados. Os principais requisitos e desafios para gerenciamento e controle em redes do futuro são discutidos no Capítulo 4. Foi realizado um estudo dos trabalhos relacionados de gerência e controle para FI. Em seguida, no Capítulo 5, é apresentado um modelo de gerenciamento e controle no escopo da NovaGenesis,

listando cenários e casos de uso. A ideia é fornecer os argumentos necessários para utilizar a NovaGenesis como uma arquitetura integrada para IoT, SDN, NFV e CCN, cobrindo não só o plano de dados, mas também o plano de gerenciamento e controle. No Capítulo 6 é apresentada uma comparação das abordagens existentes com a iniciativa NovaGenesis. Finalmente, no Capítulo 7, o trabalho é concluído resumindo seus principais resultados e ações futuras.

2 Revisão Tecnológica e Conceitos

Neste capítulo serão apresentados conceitos importantes de gerenciamento e controle, formando o embasamento teórico para o entendimento dos próximos capítulos.

"A rede de dados de hoje é dividida em três planos principais: (i) o plano de dados que encaminha os pacotes de dados individuais; (ii) o plano de controle que implementa os algoritmos de roteamento distribuídos entre os elementos de rede; e (iii) o plano de gerência que monitora a rede e configura os mecanismos do plano de dados e os protocolos do plano de controle" (17).

A seguir são discutidos sobre os planos de gerenciamento e controle do ponto de vista da arquitetura de redes atuais.

2.1 Arquiteturas Atuais de Gerenciamento e Controle

2.1.1 Arquiteturas de Controle

As atuais arquiteturas de rede, como *Open Systems Interconnection* (OSI) (18) ou *Transmission Control Protocol - Internet Protocol* (TCP/IP) (19) incluem um plano de controle para equipamentos de operação de rede. O plano de controle foi criado inicialmente para executar um único algoritmo de roteamento. Atualmente, o plano de controle exerce funções extremamente complexas que incluem a configuração simultânea de várias funções ligadas ao roteamento, desempenho, qualidade, de forma a garantir que os recursos disponíveis sejam usados de forma harmoniosa (17).

Arquiteturas IP convergentes como *Next Generation Network* (NGN) (20), *IP Multimedia Subsystem* (IMS) (21) e *IP Television* (IPTV) (22) também fornecem modelo de controle de recursos de rede. Neste contexto, uma arquitetura de controle engloba como os diversos componentes arquiteturais são controlados. O TCP/IP e muitas outras arquiteturas adotam um plano de controle distribuído. As ações de controle são um reflexo da interação entre o controle distribuído implementado usando protocolos padrão.

Roteadores de Internet são controlados por protocolos distribuídos. O plano de controle é implementado dentro do equipamento de roteamento, incluindo funções de configuração de tabela de roteamento (ex.: *Routing Information Protocol* - RIP (23) e *Open Shortest Path First* – OSPF (24)), relatório de erros (*Internet Control Message Protocol* – ICMP (25)), classificação de tráfego, priorização de pacotes (*Differentiated Services – DiffServ* (26)), *firewall*, *multicast* e muitas outras operações de controle.

A maior parte do tráfego na Internet atual é baseado em TCP (www.caida.org). O TCP implementa mecanismos de controle de fluxo baseado em janelas deslizantes. Existem muitas pesquisas sobre como melhorar a eficiência do controle TCP, porém poucos testes desses novos algoritmos foram simulados em escala (27). Assim, o modelo TCP/IP usado na Internet pode ser inadequado em relação à estabilidade e resposta rápida às variações que os cenários futuros exigem, tais como: (i) capacidade para satisfazer as necessidades de tráfego; (ii) escalabilidade; (iii) abertura; (iv) robustez; (v) segurança; (vi) diversidade; (vii) ubiquidade; (viii) integração e simplificação e (ix) extensibilidade (28);

Ainda no contexto da Internet, o objetivo da NGN é trazer tráfego de telefonia em escala para redes TCP/IP. *Media Gateways* (MGs) são usados para traduzir o tráfego telefônico de canais E1 para aplicativos em tempo real usando o *Real-Time Protocol* (RTP) (29). O responsável por controlar e monitorar as sessões multimídia em curso é do *Real Time Control Protocol* (RTCP) (30). O RTCP é baseado na transmissão periódica de pacotes de controle para os participantes de uma sessão. Esses pacotes carregam informações para que as aplicações controlem e monitorem o desempenho de fluxos de dados em tempo real. Mensagens e relatórios são enviados pela origem e pelo destino para realizar o diagnóstico comportamental da rede, incluindo métricas tais como *jitter* entre a chegada de pacotes RTP consecutivos, número de pacotes perdidos, número de pacotes transmitidos, etc. Os equipamentos da camada de controle *Media Gateways Controllers* (MGCs) ou *SoftSwitches* utilizam-se dos protocolos mestre/escravo *Media Gateway Control Protocol* (MGCP / *Internet Engineering Task* Force - IETF) (31) e H.248 (International *Telecommunication Union* - ITU) (32), que permitem controlar todos os MG da camada de transporte de uma rede IP.

A arquitetura IMS também implementa MGC semelhante a NGN. No entanto, muitas outras funções de controle são implementadas na camada IMS. Exemplos são: (i) *Resource Admission Control Function* (RACF); (ii) *Policy Decision Function* (PDF); e (iii) *Call Section Control Function* (CSCF). Esses *softwares* implementam funções de controle que fornecem admissão de fluxos de tráfego, *Quality of Service* (QoS), implantação de políticas específicas e estabelecimento de sessões do *Session Initiation Protocol* (SIP) (33). O CSCF solicita os recursos de transporte necessários para uma nova sessão SIP enquanto que o RACF configura os fluxos admitidos na rede.

Arquiteturas de IPTV normalmente adicionam funções de controle e distribuição de conteúdo, incluindo troca de canal de vídeo, QoS, *Quality of Experience* (QoE), controle de propriedade intelectual, seleção de *middleware*, configuração de IP *multicast*, etc. O modelo de controle IPTV é muito semelhante ao NGN e IMS. Essas arquiteturas podem ser vistas como o "estado da arte" do controle de funções em redes TCP/IP antes do surgimento da SDN.

2.1.2 Arquiteturas de Gerência

A abordagem mais comum para gerenciar redes hoje é combinar as informações necessárias para o gerenciamento em um único computador que interage com os diversos componentes da rede. Estas informações apoiariam o diagnóstico e solução de problemas de rede. Diversos padrões de gerenciamento têm sido propostos, entre eles o modelo OSI, que usa *Common Management Information Protocol* (CMIP) (34), e o modelo da Internet, que usa *Simple Network Management Protocol* (SNMP) (35). Devido à sua simplicidade de implementação, o SNMP tornou-se um padrão da indústria para gerenciamento de rede, uma prova de que modelos de gerenciamento e controle emergentes devem ser simples e eficazes para serem considerados pela indústria.

A arquitetura de gerenciamento de redes usando o SNMP tem quatro componentes básicos. Será discutido brevemente cada um deles.

- **Agente:** É o elemento gerenciado da rede. Estes elementos devem fornecer ou tornar disponíveis dados contextualizados (informações) para o Gerenciador de rede. Em FI, agentes devem manter ontologias para aumentar o "entendimento" de elementos gerenciados e fornecer informação confiável, ordenada, precisa e contextualizada. Em redes autonômicas, o agente é o mesmo encontrado em sistemas legados, embora possa ser adaptado para permitir o autogerenciamento.

- **Gerente:** ou estações de gerenciamento são responsáveis por recolher ou receber dados do equipamento gerenciado. Gerentes geralmente têm uma interface homem-máquina que permitem a gestão amigável por um operador de sistema. FIAs devem ter gerentes autonômicos inteligentes para diminuir a interação com os seres humanos (36). Gerentes e controladores do futuro serão obrigados a operar como pilotos automáticos que regem, controlam e gerenciam os elementos autonomicamente e de acordo com as condições de funcionamento, objetivos, regras, regulamentos e processos de negócios estipulados por operadores humanos e/ou de inteligência artificial. Segundo (37), um gerente autonômico deve ter capacidade de monitoramento, análise, planejamento, tomada de decisão, execução e aprendizado para alcançar o objetivo de gerenciar a rede de forma autônoma, ou seja, com o mínimo de intervenção de operadores humanos. Segundo (36), a computação autonômica tem quatro propriedades: autoconfiguração, auto-otimização, autocura e autoproteção. Para alcançar essas propriedades autonômicas, o gerente deve estar ciente do seu estado (*self-awareness*), do meio externo (*self-situation*), preparado para detectar mudanças (*self-monitoring*) e se autoajustar (*self-adjustment*) (38).

- **Protocolo de Gerenciamento:** As redes atuais são complexas demais para terem equipamentos gerenciados individualmente pelos operadores. O protocolo de gerência foi concebido com o objetivo de ampliar e simplificar o gerenciamento dos recursos de rede de longa distância. O SNMP, como o nome indica, tem arquitetura simples e de fácil implementação. O SNMP usa

MIB para mapear os elementos gerenciados em objetos, utilizando o modelo cliente-servidor para envio de mensagens espontâneas (*traps)* e varredura de dados. No modo varredura, o gerente (cliente) examina os agentes (servidores) para coletar os dados. Enquanto que para envio das *traps* o agente envia uma mensagem espontânea para o gerente (cliente).

- **Informações de Gerência:** SNMP usa uma estrutura de árvore para identificar as variáveis de gerenciamento. Devido ao grande número de variáveis, vários ramos da árvore foram definidos como uma coleção de objetos relacionados entre si. Cada objeto é representado por um nó (texto descritivo e um nome da variável associada com um valor). Uma MIB é um grupo de objetos relacionados com um assunto (por exemplo: uma variável de desempenho de rede ou um determinado protocolo de comunicação). Eles podem ser definidos por uma *Request for Comments* (RFC) ou por uma especificação proprietária, definida por uma empresa. As *communities* implementam política de acesso que autenticam o tráfego SNMP entre agentes e gerente com o objetivo de aumentar a relação de confiança entre eles, porém elas não evitam o uso mal intencionado, pois são enviadas sem criptografia.

2.2.1 FCAPS

Nesta seção serão discutidas as cinco áreas funcionais FCAPS (39) introduzidas pelo modelo OSI da *International Organization for Standardization* (ISO) para o sistema de gerenciamento de rede (*Network Management Systems* - NMS). FCAPS são as iniciais de *Fault*, *Configuration*, *Accounting*, *Performance* e *Security*. Para cada uma das cinco áreas funcionais serão debatidos requisitos de gerenciamento, operação e administração em FIAs e para implementação de serviços e recursos em arquiteturas emergentes.

- **Falhas:** As falhas são motivo de preocupação no mundo da tecnologia da informação. Com o advento da gestão pró ativa da rede/*cloud*, ou seja, encontrar as falhas antes que elas aconteçam, visualizadores de alarmes tornaram-se requisito essencial para um bom gerente de rede/infraestrutura. O gerenciamento de falhas pró ativo é alcançado analisando a causa raiz de uma falha. Uma nova arquitetura de gestão deve encontrar um ponto de falha que pode culminar em uma falha de sistema e atuar sobre este ponto antes que todo o sistema esteja comprometido, de forma a aumentar a confiabilidade da rede. Autocura é uma propriedade emergente nessa área. Tecnologias como NFV e SDN dependem de componentes implementados em *software*. Portanto, em FI, falhas em elementos computacionais podem ter um efeito devastador sobre a rede de comunicação, desativando controladores ou funções de rede implementadas em *software*.

- **Configuração:** As diretivas de configuração atuais não estão preparadas para reagir continuamente às condições de rede (autoconfiguração) e operadores devem ajustar manualmente a configuração de rede através de um conjunto de comandos de configuração de baixo nível chamado de *Comand Line Interface* (CLI). A configuração em redes IP tradicionais é um problema maior quando há necessidade de manter a consistência das bases de dados em dois ou mais elementos de rede. Nesses casos, é necessário verificar se os comandos foram enviados com sucesso em todos os elementos envolvidos, antes de alterar o banco de dados destes elementos. Enviar dados de configuração para elementos de rede requer o conhecimento da informação do elemento gerenciado como endereços e formato da CLI. Devido a essa limitação, FIAs devem ter métodos para reconfigurar dinamicamente os dispositivos de rede quando um evento ocorre, a chamada propriedade de autoconfiguração (36). Para isto é necessário um banco de dados que armazena a informação raiz e as manipula conforme a demanda, mostrando o seu estado em uma interface amigável e fácil de usar. Tecnologias autonômicas requerem a definição de objetivos de alto nível, plano de

execução de ações, monitoramento de resultados, análise, correlacionamento e tomada de decisão. É o chamado ciclo de decisão autonômico (36).

- **Contabilização:** A finalidade desta área funcional da gestão é entender o comportamento da rede para que seus recursos não ultrapassem cotas pré-estabelecidas. Faz medições contínuas relacionadas ao faturamento. FIAs devem implementar algoritmos que garantam a utilização justa e ideal dos recursos. A exposição de recursos físicos existentes e o gerenciamento dinâmico de recursos são necessários. Expor significa representar em objetos de informação as capacidades, limitações, estados e fatias de uso, etc. Nas FIAs, tecnologias de exposição e composição dinâmica permitem a virtualização (uso compartilhado de fatias de um recurso físico). A contabilização e gerência de recursos incluem o inventário de elementos virtuais e seus respectivos substratos físicos.

- **Desempenho:** O princípio da gestão de desempenho é analisar o comportamento do sistema com o objetivo de introduzir melhorias na rede. FIAs não devem tolerar um elemento (físico ou virtual) com baixo desempenho que provoca atrasos na evolução da rede e insatisfação ao usuário. A análise de desempenho de funções virtuais em NFV está em sua infância. A convergência rede/*cloud* exige computação de alto desempenho, elástica, com suporte em *hardware* à virtualização de recursos de rede e computação. Outra questão importante diz respeito onde devem rodar as funções virtuais.

- **Segurança:** Uma das maiores preocupações da área de segurança é restringir o acesso a redes e sistemas, impedindo sua utilização abusiva, intencionalmente ou não, e proteger o funcionamento dos recursos de rede. FIAs precisam de um ambiente seguro, privado e confiável para onde os dados possam ser processados e trocados em mensagens confidenciais. Há uma discussão sobre quanta informação pode ser extraída de dispositivos de rede identificáveis e se esta informação deve ser regulada pela lei. Novas

arquiteturas usam os chamados nomes autocertificáveis (*Self-Verifying Names* – SVN) (40) para nomear entidades (nós, serviços, domínios, conteúdos). Esses nomes podem ser usados como identificadores e localizadores. São calculados utilizando funções *hash* a partir de padrões digitais de entrada, como por exemplo, um arquivo digital. Esses novos esquemas e espaços de nomeação utilizando SVNs trazem avanços significativos ao suporte a segurança e privacidade, e precisam ser incorporados aos modelos de controle e gerência de rede. Outro avanço importante a ser incorporado é a chamada operação/composição dinâmica de serviços baseada em contratos. Nela, os programas de computador estabelecem contratos com acordos de nível de serviço (*Service Level Agreements* – SLAs) antes de enviar dados. Os contratos visam estabelecer com clareza as metas e os relacionamentos entre os serviços. Esse modelo favorece a formação de redes de confiança entre programas de gerência e controle, e pode ser aplicado aos modelos atuais e futuros (SDN, NFV, etc.).

2.3 Tecnologias Emergentes para Internet do Futuro

Ao longo dos anos, a Internet tornou-se essencial para a vida das pessoas. Mundialmente, pessoas estão conectadas, interagindo, publicando conteúdos ou somente navegando. A distância geográfica não é mais um limitante para a interação entre pessoas, seja para fins comerciais ou sociais, contribuindo em vários fatores para o desenvolvimento do ser humano. Ocorre que a Internet tornou-se extremamente importante para a sociedade atual, porém ela carece de melhorias em vários aspectos de sua arquitetura. Preocupados com uma possível ossificação da rede mundialmente famosa, diversos países têm buscado a cura para os atuais problemas da Internet. Um alto valor em investimentos tem sido aplicado em pesquisas cientificas de FI com o objetivo de solucionar seus problemas e adequá-la para os novos milhões de pessoas e dispositivos (principalmente) que deverão ser conectados a ela à medida que a inclusão digital cresce e a chamada Internet das Coisas floresce.

Além das dificuldades encontradas na Internet no âmbito do controle e gerenciamento de rede já discutidas neste capítulo, pode-se citar ainda fatores adicionais limitantes que corroboram com a importância em pesquisas científicas na área de FI. São eles:

- **Nomeação, Mobilidade e Multihoming:** A semântica do endereço IP é usada para identificar e localizar o dispositivo na rede (41). Quando um dispositivo é movido há perda da sua identidade, pois o endereço IP é alterado. Como consequência, a rede perde a rastreabilidade do dispositivo e arruína-se a relação entre identificador e localizador. Para manter a rastreabilidade é necessário que se tenha um identificador persistente como um documento de identidade (um SVN encaixa perfeitamente nesse papel). Da mesma forma, uma identidade única para todos os dispositivos permite mobilidade generalizada de coisas e conteúdos.

  A pilha de protocolos TCP/IP não foi desenhada para suportar dispositivos móveis. Manter ativa a comunicação de um dispositivo conectado à Internet sem degradação ou perda da qualidade do serviço é um grande desafio no *handover,* ou seja, durante a troca da estação remota responsável.

  No sistema de resolução de nomes da Internet, o *Domain Name System* (DNS), um domínio é vinculado a um endereço IP estático. Assim, a ligação será perdida se houver uma mudança dinâmica de endereços IP em dispositivos móveis.

  Já o *Multihoming* é o suporte a múltiplas presenças na Internet. Visa melhorar a confiança e desempenho da rede, porém os múltiplos caminhos de comunicação e os vários endereços IP com diferentes cenários de mobilidade aumentam muito a complexidade de roteamento e a demanda por recursos da rede.

A escassez de endereços IP válidos pode ser um obstáculo para a consistência de identificadores na Internet. Muitos acreditam que uma grande parte das coisas do mundo real possa estar interligada à rede e possuir representantes no mundo virtual. Para atender aos requisitos de rastreabilidade de fontes e alcançar a escalabilidade necessária, a rede do futuro deve se interligar a um número inimaginável de coisas, maior ainda (e porque não?) que os $3{,}4e^{+38}$ endereços do IP versão 6 (IPv6). Em muitas situações, se faz necessário determinar exatamente que entidade tem um determinado identificador. Por exemplo, um dispositivo vestível que acompanha uma pessoa cardíaca na rua. Na hora que se detecta alguma anomalia, é necessário se ter certeza da identidade do nó e da relação desta com a identidade do paciente. Isso pode ser comprometido se a identidade do nó não for sólida e rastreável. O IPv6 permite que nós tenham endereços únicos globais, atendendo a esse cenário de escassez de endereços. Entretanto, há uma grande barreira na indústria para a utilização do IPv6, onde grande parte dos *software* e *hardware* permanecem utilizando o sistema de endereçamento IPv4.

- **Segurança, Privacidade, Transparência e Anonímia:** Identificadores não persistentes como aqueles utilizados na Internet atual contribuem para o aumento da insegurança da rede pelo simples fato de que todas as coisas não possuem uma identidade, trazendo dificuldade para rastrear o comportamento de qualquer dispositivo na rede e indefinição de responsáveis pela utilização de recursos.

  Não há segurança sobre a identidade de um dispositivo na rede, porque atualmente qualquer dispositivo pode se conectar a Internet fazendo uso de um endereço IP. A rede carece de autenticação de equipamentos e qualquer dispositivo pode se passar por outro se usar o mesmo endereço IP.

  Por outro lado é necessário manter a privacidade das características de navegação dos dispositivos da rede.

Os usuários estão vulneráveis quanto ao uso indiscriminado de envio de dados tipo *spams*, disseminação de vírus, ataques de negação de serviço (*Deny of Service* – DoS), *trojans, spywares*, etc. As preocupações aumentam com o uso do comércio virtual onde altos valores têm sido trocados mundialmente e o impacto na economia é enorme. A utilização de *firewalls* filtra pacotes indesejados, mas tem ação limitada contra certos tipos de ataques. A Internet das Coisas vem para desafiar os modelos já estabelecidos e a gerência de segurança deve englobar esses desafios.

2.3.1 Internet das Coisas

Internet das Coisas, em inglês *Internet of Things* (IoT) é o termo usado para designar a conexão de tudo que existe no mundo real ao mundo virtual por meio da Internet (ou formando-se novas Internets, como originalmente preconizado pelo Auto-ID center (42) do *Massachusetts Institute of Technology* – MIT). Desta forma, "coisa" pode ser um dispositivo celular, um computador, ou ainda uma cadeira, uma roupa, um documento ou até mesmo um ser humano, enfim, qualquer objeto que o leitor desta dissertação possa imaginar. Posto isto, com o advento da IoT, a escala dos problemas atuais da Internet tende a aumentar exponencialmente.

As mudanças para que se faça possível a Internet das Coisas não se resumem à parte do *software*, mas também aos objetos que serão conectados, ou seja, haverá grande impacto em *hardware* e *firmware*. Os dispositivos para IoT devem prever redução de tamanho, custo e principalmente de gasto energético. Uma das principais tecnologias sem fio utilizadas para apoiar a IoT é a *Radio-Frequency IDentification* (RFID), que permite identificar um grande número de pequenos nós a um baixo custo. Essa tecnologia e várias outras (*Personal Area Networks* - PANs) possuem alcance da comunicação entre os nós de aproximadamente 10 metros. Neste caso, a escassez de endereços IP é uma dificuldade a mais, tornando opaca a identificação desses nós. Além disso, alguns nós RFID não possuem bateria, e para economizar energia precisam entrar em modo *sleep*. Durante este tempo, haverá uma lacuna na comunicação, o que caracteriza a chamada comunicação intermitente.

No contexto de *Application Programming Interfaces* (APIs) e *software*, um *middleware* para IoT deve ser invisível do ponto de vista dos detalhes tecnológicos. Para isto, tem-se utilizado o desenvolvimento de arquiteturas baseadas em serviço (*Service Oriented Architecture* – SOA (43)) que permite o reuso de *software* e *hardware* e torna o processo de engenharia mais eficiente. Na arquitetura SOA utilizam-se blocos de construção baseados em *WebServices*. *WebService* é um tipo de serviço que se comunica utilizando protocolos da Internet (*Hypertext Transfer Protocol* - HTTP) e envia e recebe dados usando o formato *eXtensible Markup Language* (XML).

A abordagem *Big Data* (44) visa processar e correlacionar a grande quantidade de dados sensoriados no ambiente IoT. A análise destes dados, de forma estruturada ou não estruturada, vem sendo empregada pela indústria com o objetivo de trazer as mais diversas formas de interatividade das "coisas" com o mundo real. Entender a necessidade do ser humano, baseado em análises estatísticas e ferramentas de Inteligência de Negócios (*Business Intelligence*) e entregar o que se precisa no tempo certo, é o que deseja conquistar com o *Big Data*. Usualmente, *Big Data* e IoT são tecnologias ligadas. Um excelente exemplo de que infraestruturas de rede e computação estão convergindo, e sua gerência e controle devem ser integradas.

A computação em nuvem – *cloud computing* é a infraestrutura que suporta o *Big Data*, provendo o armazenamento dos "megadados" provenientes da IoT. A computação em nuvem não se limita à necessidade de armazenamento de dados em um *hardware* específico, na nuvem ele pode ser alcançado de qualquer lugar. A possibilidade de ter acesso à informação não importa onde o usuário esteja, vai de encontro ao requisito de ubiquidade que a IoT necessita para permitir sua intensa evolução.

2.3.2 Redes Definidas por *Software*

As Redes Definidas por *Software* (*Software Defined Networking* – SDN) foram motivadas em função da computação em nuvem e da virtualização da rede. A

SDN é baseada em código aberto, com isto é possível que qualquer *hardware* habilitado para SDN possa ser utilizado independente do *software*, que contribui para a integração de equipamentos de diferentes fabricantes. Esta característica permitiu a rápida evolução do paradigma SDN.

A SDN desacopla o plano de dados com o plano de controle, movendo o plano de controle para fora dos equipamentos. O plano de controle visa popular a tabela de tráfego e o Plano de Dados executa o encaminhamento dos pacotes. O Plano de Controle centraliza a inteligência de toda a rede e toma as decisões quanto à atualização das tabelas de fluxo. O controlador possui uma base de dados única que contém os resultados das observações da rede. Ele utiliza destas informações para controlar o tráfego da rede. Como o Plano de Controle é implementado logicamente na arquitetura SDN, a configuração e implementação de novos componentes de controle é mais simples, aberta, ágil e adaptável se compararmos com as redes legadas onde o controle está descentralizado em milhares de elementos de rede físicos ou virtuais.

O OpenFlow (45) é o protocolo mais conhecido para implementar SDN. Ele é o responsável pela comunicação entre o *software*, que é representado pelo controlador, e o *hardware*, que são os elementos da camada física. Os elementos da camada física, tais como *switches* e roteadores, estão no Plano de Dados. Posto que a implementação de uma rede SDN deva ser feita em código aberto é necessário que se utilize um canal seguro para estabelecer e finalizar as conexões. O protocolo mais usado para alcançar tal finalidade é o *Secure Socket Layer* (SSL) que confere segurança na comunicação fornecendo uma chave privada. O protocolo TCP também pode ser utilizado sozinho para o estabelecimento da conexão entre o *switch* e o controlador em aplicações onde a simplicidade é mais importante que a segurança.

Os dispositivos de rede devem ter uma tabela de encaminhamento de tráfego para suportar o padrão OpenFlow. Nesta tabela estão as regras necessárias para encaminhar o tráfego de uma porta de entrada para uma porta de saída. O elemento de rede pode ser habilitado para trabalhar simultaneamente em redes SDN ou legadas. Quando ele se comporta deste modo é chamado de elemento híbrido.

O protocolo OpenFlow pode atuar na forma pró-ativa ou reativa. No modo pró-ativo, o controlador centralizado mantém pré-configurado uma grande

quantidade de regras de encaminhamento de pacotes nos dispositivos de rede. Enquanto que no modo reativo a configuração do fluxo é gerada a partir de eventos assíncronos notificados ao controlador.

Exemplos de controladores SDN para OpenFlow são o NOX (46) (47) e o POX (46). O controlador NOX faz parte da primeira geração de controladores e é amplamente utilizado em aplicações que necessitam de alto desempenho. O POX é um controlador NOX implementado na linguagem de programação Python (o NOX foi escrito em linguagem C++). O POX possui baixo desempenho, mas possui maior simplicidade de implementação.

2.3.3 Virtualização de Funções de Rede

A Virtualização de Funções de Rede (*Network Function Virtualization* – NFV) permite que funcionalidades de rede possam ser virtualizadas substituindo *hardware* especializado, como servidores de grande porte, por máquinas virtuais que comportam alto desempenho e escalabilidade. Os operadores de serviços de telecomunicações têm abraçado o NFV como forma de atender à enorme demanda de tráfego sem que se aumentem os custos operacionais atrelados ao aumento da infraestrutura, uma vez que as máquinas virtuais são capazes de cumprir esta função.

Exemplos de elementos de rede cuja função pode ser virtualizada são os *Firewall*, *Proxy*, *Network Address Translation* (NAT) e *Virtual Private Network* (VPN). Esses elementos são conhecidos como *middleboxes*. Os *middleboxes* são servidores dedicados que executam uma determinada função nas redes legadas. Com o advento do ecossistema NFV, esses servidores são virtualizados e compartilhados garantindo a versatilidade e elasticidade dos recursos da rede, além da enorme otimização da infraestrutura.

Além dos benefícios citados acima, o NFV fornece subsídios para prover os serviços de nuvem tais como, *Infraestructure-as-a-Service* (IaaS), *Software-as-a-Service* (SaaS) e *Platform-as-a-Service* (PaaS), que possibilitam agilidade no consumo sob demanda. O conceito IaaS permite a contratação de uso de servidores virtuais ao invés de comprá-los. De forma análoga ao conceito IaaS, no SaaS é

possível utilizar um *software* sem que seja necessário adquirir a licença. O modelo PaaS é um merge entre os conceitos IaaS e SaaS, onde o usuário cria sua própria plataforma adequando *hardware* e *software* à aplicação. Em todos os modelos o pagamento é baseado na utilização efetiva dos recursos contratados.

A combinação das tecnologias SDN e NFV revolucionarão as arquiteturas de rede tradicionais atendendo aos abstrusos requisitos de orçamento, combinados à agilidade na adição de ferramentas *hardware* e *software* que demandam o mercado atual.

2.3.4 Redes Centradas em Conteúdo

Content Centric Networking (CCN) é uma rede que desacopla a localização do conteúdo da sua identidade, focando no conteúdo em si e não na sua localização física. Desta forma, o nome dos conteúdos é utilizado para identificá-los e posteriormente localizá-los na rede. Segundo Jacobson (12), uma vez que os usuários da Internet focam no conteúdo da rede, é natural que a rede seja orientada em conteúdos e não na localização dos conteúdos como foi desenhada a arquitetura atual da Internet.

O termo ICN é também usado para designar o mesmo conceito das Redes Centradas em Conteúdo. Neste trabalho é usada a sigla CCN para se referir às pesquisas baseadas em dados nomeados (*Named Data Objects* – NDO) que têm a finalidade de prover conteúdos na rede. Segundo Ahlgren (48) *"A principal abstração da abordagem ICN é o NDO. O NDO é independente da localização e dos métodos de armazenamento e transporte"*.

A abordagem CCN visa melhorar a eficiência na transferência de conteúdos, permitindo que eles sejam roteados sem que a rede precise solicitar aos seus produtores, ou seja, não é necessário recorrer à fonte inicial que gerou o conteúdo para que se tenha acesso a ele. Todo conteúdo possui um identificador único que pode ser replicado ao longo da rede, por meio de roteadores de conteúdos que armazenam os dados em *cache (content store)*, permitindo assim um melhor desempenho no acesso à informação.

Os usuários de uma rede CCN ou consumidores da informação, não precisam conhecer o endereço do servidor para procurar o que necessitam, bastam apenas buscar pelo nome que desejam, uma vez que qualquer nó da rede tem autonomia para disponibilizar ao usuário o conteúdo solicitado desde que se tenha uma cópia válida deste conteúdo. Neste conceito, o usuário solicita os pacotes à rede sem identificar o destinatário. Desta forma, são facilmente evitados ataques por inundação (DoS) cuja finalidade é esgotar o processamento de um determinado nó da rede.

A proteção dos conteúdos que trafegam pela rede é outro ponto alcançado pela abordagem CCN, que visa garantir a segurança e autenticidade dos dados através da própria infraestrutura da rede que obriga que todos os dados sejam assinados. Para alcançar tal vantagem, criou-se na pilha de protocolos CCN uma camada específica de segurança (49). Esta camada garante a integridade durante a troca de conteúdos, uma vez que todo pacote na rede CCN deve conter a assinatura de seu emissor no próprio nome do pacote. A assinatura é gerada por meio de uma PKI (*Public Key Infrastructure*) que pode ser verificada no momento da entrega, garantindo assim a relação de confiança entre produtores e consumidores. Outra forma para garantir integridade aos pacotes é a utilização de nomes autocertificáveis que incorporam diretamente o *hash* do conteúdo no nome do objeto.

3 NovaGenesis: Concepção e Conceitos

NovaGenesis (50) é uma nova arquitetura convergente de informação que integra vários ingredientes considerados chaves para a Internet do Futuro, tais como IoT, SDN, NFV, CCN, dentre outros. É uma iniciativa para criar uma nova arquitetura de processamento, armazenamento e troca de informações. A NovaGenesis pode ser vista como uma arquitetura sobreposta a Internet atual, bem como uma nova arquitetura de Internet que pode integrar as redes de comunicação em nuvem (50). O objetivo é superar os problemas e limitações da Internet, sendo suficientemente flexível para se adaptar às novas tecnologias que vão surgir.

Existem alguns pontos a considerar ao construir uma FIA: (i) a forma como as informações, serviços, sistemas operacionais e entidades físicas são nomeadas, identificadas e localizadas; (ii) o ciclo de vida de informações e serviços, ou seja, como eles são expostos, descobertos e acessados de forma autorizada; (iii) como coordenar serviços, negociar, contratar, alocar recursos, acompanhar a qualidade, avaliar a reputação e liberar os contratos; (iv) a maneira como os protocolos são implementados e organizados em camadas, no sentido de definir uma arquitetura; (v) como coordenar sinergicamente os recursos físicos e virtuais; (vi) como controlar e gerenciar entidades físicas para otimizar recursos físicos; (vii) como persistentemente identificar conteúdos e serviços independentemente de suas localizações; (viii) como incluir trilhões de "coisas" na Internet.

A construção de uma arquitetura mais flexível baseia-se nestes pontos e os seguintes conceitos e opções de *design*:

3.1 Todas as Existências são Nomeadas

Mais e mais coisas estão sendo conectadas à Internet e uma nova arquitetura deve estar preparada para reconhecê-las pelos seus nomes. A arquitetura de rede atual normalmente limita os espaços para nome disponíveis. Por exemplo, a Internet emprega pelo menos quatro espaços para nomes: nomes de domínio, nomes de rede e *host*, soquetes e nomes de recursos uniformes (*Uniform Resource Identifiers* – URIs).

Eles são geridos pela ICANN, *Internet Corporation for Assigned Names and Numbers* e resolvidos através do sistema de nome de domínio (DNS) e outros sistemas não padronizados. A NovaGenesis suporta a nomeação de todas as coisas, utilizando não apenas *Natural Language Names* (NLNes), mas também *Self-Verifying Names* (SVNes), que são preferíveis para linguagem de máquina. Todas as coisas podem ser nomeadas através de NLNes e SVNes, o que garante que uma existência individual pode ser denotada de muitas formas, permitindo a criação de um enorme grafo de nomes. Esses nomes podem ser usados para identificar e localizar "coisas", *software*, qualquer outra existência, que pode ser um serviço, um usuário ou qualquer equipamento de rede, por exemplo.

A NovaGenesis incrementa o espaço de nomes usados no TCP/IP com uma abordagem de nomenclatura genérica, que permite qualquer tipo de esquema de nomeação, inclusive SVNes gerados a partir de atributos perenes de entidades, como por exemplo o padrão binário de um arquivo ou um conjunto de características de fabricação de um computador. Os nomes estão vinculados uns aos outros para formar um grafo distribuído de nomes, representado por pares <chave, valor>. Não só as chaves, mas também os valores podem ser NLNes ou SVNes. Estes pares são definidos como um *Name Binding* (NBs) ou em Português, ligação entre nomes.

A Figura 2 ilustra o procedimento de nomeação e armazenamento de NBs na NovaGenesis onde dois *hash codes* estão relacionados por uma ligação de nomes. O primeiro deles é o resultado *hash* de um nome codificado em *American Standard Code for Information Interchange* (ASCII), enquanto que o segundo representa um *hash* gerado a partir de um padrão binário de um arquivo.

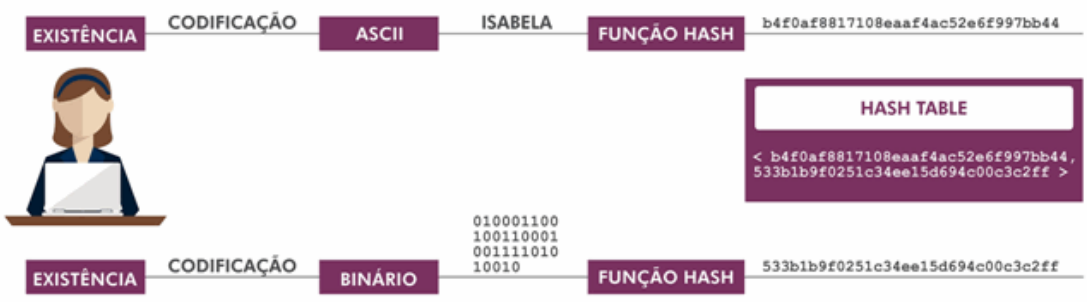

Figura 2 – *Processo de Nomeação e Armazenamento de NB na NovaGenesis.*

3.2 Modelo Publica/Assina para Acesso a Conteúdos e Serviços

Normalmente, SVNes são a saída de funções *hash* criptografadas, que são armazenadas como chaves ou valores em um NB. A NovaGenesis armazena as ligações entre nomes em uma tabela *hash* distribuída, que é operada por um serviço denominado *Hash Table Service* (HTS). Esses NBs podem ser vistos como identificadores e/ou localizadores, dependendo da maneira que estão ligados a outros nomes.

A Figura 3 ilustra um grafo que ajuda a visualizar a forma como a NovaGenesis usa NBs para identificar e/ou localizar existências.

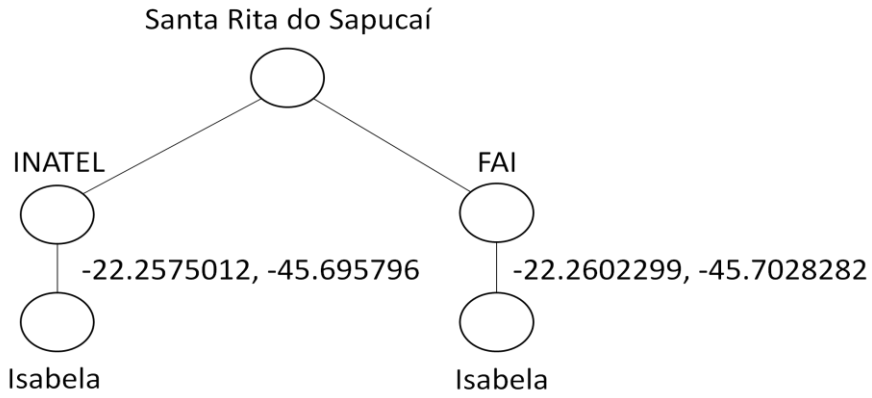

Figura 3 – *Name Bindings Representados por um Grafo.*

O NB < *INATEL; Isabela* > representa a ligação da aluna Isabela ao Inatel. Enquanto que o NB < *FAI; Isabela* > representa a ligação da aluna Isabela à Faculdade de Administração e Informática (FAI). Desta forma, o NB < *INATEL; Isabela* > é usado para identificar a aluna Isabela dentro do escopo INATEL. Ainda na Figura 3, o nome INATEL pode ser localizado pelo NB que se liga ao nome latitude/longitude < *INATEL; -22.2575012,-45.695796* >.

No entanto, como são baixados os NBs de alguma HTS para outros serviços? Como eles são descobertos e transportados? Ou, como eles são enviados para serem armazenados no HTS e depois lidos por outros serviços? Ligações entre nomes e até mesmo entre nomes e conteúdos são publicados por qualquer serviço através de um

serviço publica/assina (*Publish/Subscribe Service* – PSS). Portanto, o PSS é um serviço bem conhecido em qualquer domínio da NovaGenesis. Quando um serviço publica ou assina um NB para a rede em uma nuvem NovaGenesis, ele faz isso por meio de uma instância PSS. Este serviço pub/sub também é usado para disponibilizar dados ou informações de outros serviços.

O modelo NovaGenesis ainda tem a última peça desse quebra-cabeça, que é o chamado serviço genérico de resolução de Indireções (*Generic Indirection Resolution Service* – GIRS). O GIRS seleciona uma instância adequada de HTS para armazenar conteúdos e NBs, fazendo o balanceamento de carga entre vários HTSes. Ele recebe os NBs e conteúdos que estão sendo publicados no PSS e os encaminha para um ou mais HTSes.

3.3 Toda Entidade de *Software* é Vista como um Serviço

Este paradigma inclui o chamado protocolo implementado como um serviço (*Protocol-Implemented-as-a-Service* – PIaaS) (50). Toda entidade de *software* é vista como um serviço que pode trocar informações, negociar informações processadas, armazenar dados ou delegar tarefas. O ciclo de vida de serviços é realizado por meio da interface fornecida pelo PSS. Portanto, serviços publicam palavras-chave que ajudam na descoberta de outros serviços e conteúdos, tais como descritores, configurações, etc. Eles também negociam publicando e subscrevendo serviços. Finalmente, eles contratam uns aos outros, criando uma rede de serviços confiáveis. Assim, dados são trocados baseados em contratos já estabelecidos que permitem que diferentes serviços compartilhem SVNes para ajudar na comunicação. Assim, escopos, espaços, identificadores e localizadores são descobertos e assinados, criando a base para o modelo de comunicação autoverificável e distribuído.

Os contratos são assinados através do PSS que oferece a dissociação entre produtores e consumidores de conteúdo, dando mais flexibilidade para aplicações e reduzindo o tráfego de dados. Dessa forma, o assinante recebe o conteúdo desejado do HTS local, ao invés de buscar em algum domínio distante. Na NovaGenesis, nomes e conteúdos podem ser guardados no domínio local. Assim, o HTS funciona

como um *cache* de rede, armazenando localmente conteúdos autorizados de outros domínios.

3.4 Toda Existência Física é Representadas por Serviços

Um *Proxy/Gateway/Controller* (PGC) é um serviço que representa as "coisas" do mundo físico no ecossistema NovaGenesis, ou seja, ele representa uma existência para os serviços de descoberta, negociação, contratação, acompanhamento, liberação, troca de dados, etc. Eles expõem características, atributos, nomes e o estado das "coisas" que eles representam. O PGC busca o serviço requerido dinamicamente de acordo com as características expostas e estabelecem contratos. Os contratos vão estabelecer os acordos entre os serviços com o objetivo de formar uma rede de relacionamentos de confiança. Estes contratos possuem as metas de nível de desempenho, disponibilidade e tempo de resposta, dentre outros aspectos relevantes e específicos para cada tipo de serviço que visam garantir uma relação produtiva e transparente entre os serviços. A ideia do PCG foi demonstrada em (50).

Um PGC provê três funcionalidades: (i) atuar como um *proxy*, representando as "coisas" no ecossistema de serviço NovaGenesis; (ii) agir como um *gateway* para encapsular as mensagens NovaGenesis sobre padrões de comunicação de baixo nível, tais como *ZigBee, Institute of Electrical and Electronics Engineers* (IEEE) 802.15.4, ou *Bluetooth Low Energy* (BLE); (iii) atuar como um controlador, configurando dispositivos (gerência de configuração) de acordo com os contratos estabelecidos.

As "coisas" não podem estabelecer contratos sozinhas. Elas exigem um representante de *software* para ajudá-las. A ideia é que as "coisas" tenham seus atributos, capacidades, estados e outros dados contextualizados disponíveis para outros serviços que possam estar interessados no que uma "coisa" pode fazer. Este modelo pode ser aplicado a qualquer dispositivo do mundo físico, incluindo rede e infraestrutura em nuvem ou equipamentos *ad-hoc*. Em outras palavras, os equipamentos da infraestrutura de rede e de computação podem ser vistos como "coisas", seguindo o mesmo modelo. A heterogeneidade dos dispositivos do mundo

físico é enorme. Portanto, o suporte para gerenciamento e controle em HetNets é obrigatório em FIAs. *Gateways* deverão interoperar com uma infinidade de tecnologias através da troca de mensagens com o mínimo de gasto energético.

Finalmente, controles definidos por *software* devem ser aplicados num sentido mais amplo do que no *OpenFlow*. PGCs podem controlar um ou mais dispositivos ("coisas") de acordo com suas funcionalidades, interfaces, etc. Um PGC segue o paradigma controlador como um serviço (*Controller-as-a-Service* – CaaS), onde os recursos de rede em nuvem são controlados por uma rede de confiança de mesmo nível do CaaS. A configuração de dispositivos ou de qualquer outra "coisa" emerge como o resultado de um comportamento social entre PGCs e outros serviços NovaGenesis. Em outras palavras, certa configuração em uma "coisa" é o resultado de uma ação baseada em um contrato, dependendo do comportamento social de todo o ecossistema. Desta forma, migra-se de um modelo pré-estabelecido de gerenciamento e controle para um modelo dinâmico regido por contratos entre serviços, incluindo os controladores.

3.5 Formato das Mensagens NovaGenesis

Uma mensagem NovaGenesis apresenta a seguinte estrutura genérica:

```
ng -name --alternative version[< Y TIPO NNNNNN.....NN> < Y
                    TIPO NNN NNN NNN>]
```

Onde:

ng: campo inicial. Todas as mensagens NovaGenesis devem começar com "ng" na versão atual;

name: comando. Pode assumir os valores: p, s, *notify*, rvk e d. Que significam:

p, para publish

s, para subscribe

notify,

rvk, para revoke

d, para delivery

alternative: pode assumir os valores: b (bind), s (status) e *notify*

version: versão do protocolo. Exemplo: 0.1;

[< >]: *container* da mensagem. É um conjunto de vetores na forma de argumentos. Onde:

Y: quantidade de elementos do vetor;

TIPO: assume qualquer valor no formato de uma *string*;

N: Nome autocertificável. Argumentos e elementos do *container*;

As mensagens NovaGenesis podem publicar, assinar, notificar ou revogar eventos. Cada primitiva está descrita com mais detalhes nos itens que seguem.

3.5.1 Publish

Uma mensagem tipo *pub* é estruturada da seguinte forma:

```
ng -p --b VERSÃO [ < TAMANHO TIPO CATEGORIA > < TAMANHO TIPO CHAVE > <
                        TAMANHO TIPO VALORES > ]
```

O campo categoria pode indicar se a mensagem contém carga útil. Para publicar e notificar outros serviços usa-se o formato abaixo:

```
ng -p --notify VERSÃO [ < TAMANHO TIPO CATEGORIA > < TAMANHO TIPO CHAVE > <
      TAMANHO TIPO VALORES > < _TAMANHO TIPO pub HID OSID PID BID > ]
```

Para cada serviço parceiro a ser informado, deve ser usado um vetor adicional contendo a palavra pub, que indica que a notificação se refere a uma publicação. Demais valores são, respectivamente, (i) o identificador do *Host* onde o parceiro se encontra; (ii) o identificador do sistema operacional; (iii) o identificador do próprio serviço parceiro; e (iv) o identificador do bloco interno ao serviço e que vai tratar a notificação.

HID = *Host* ID

OSID = *Operating System* ID

PID = *Process* ID

BID = *Internal Block* ID

3.5.2 Subscribe

Para assinar um *binding* (com ou sem conteúdo) se usa o formato abaixo:

```
ng -s --b VERSÃO [ < TAMANHO TIPO CATEGORIA > < TAMANHO TIPO CHAVES > ]
```

3.5.3 Notify

O comando *notify* é formatado da seguinte maneira:

```
ng -notify --s VERSÃO [ < TAMANHO TIPO CATEGORIA > < TAMANHO TIPO CHAVE > <
      TAMANHO TIPO VALORES > < _TAMANHO TIPO pub HID OSID PID BID > ]
```

3.5.4 Revoke

Para revogar um *binding* (com ou sem conteúdo) se usa a chamada abaixo do PSS:

```
ng -rvk --b VERSÃO [ < TAMANHO TIPO CATEGORIA > < TAMANHO TIPO CHAVES > ]
```

3.5.5 Delivery

Por fim, tem-se o *delivery* que é gerado a partir do HTS para entregar nomes e conteúdos.

```
ng -d --b VERSÃO [ < TAMANHO TIPO CATEGORIA > < TAMANHO TIPO CHAVE > <
                        TAMANHO TIPO VALORES > ]
```

O *delivery* possui formato igual ao pub. É como se um serviço estivesse publicando para o outro. O *delivery* também deve ser tratado no serviço que recebe a

resposta do HTS. O conteúdo associado ao *delivery* vem na carga útil da mensagem, se houver. Nesse caso, existe ainda um comando chamado *Info* que grava o conteúdo da carga útil na forma de um arquivo no sistema operacional com o nome recebido no campo VALORES abaixo.

```
ng -info --_Alternative VERSÃO [ < TAMANHO TIPO VALORES > ]
```

4 Análise de Requisitos e Desafios para a Nova Geração de Sistema de Gerenciamento e Controle de Redes

As expectativas para abordagens em torno do gerenciamento e controle para a próxima geração de redes estão crescendo vertiginosamente. Os objetivos são oferecer suporte às novas tecnologias, novos serviços, HetNets, IoT, bem como reduzir *Operational Expenditure* (OPEX), aumentar a escalabilidade e permitir a introdução do paradigma de *software-as-a-service*. Em geral, os seguintes paradigmas devem tornar-se parte das exigências e desafios para a operação da próxima geração de arquiteturas.

4.1 Novos Requisitos de Gerência e Controle de Redes sob a ótica daInternet das Coisas

Dados produzidos através de sensores e atuadores aumentam a ritmo exponencial. Sistemas de gerenciamento e controle são esperados para suportar o aumento imprevisível dos volumes de tráfego de gerenciamento e controle devido a escalabilidade de aplicações de IoT, que incluem descobrir, configurar e reconfigurar um enorme número de novos elementos que emergem na rede de "coisas". Da mesma forma como ocorre nas redes atuais, mas em maior escala, tecnologias de gerenciamento e controle para IoT devem interoperar com uma variedade de *softwares* e *middlewares*, sendo capazes de melhorar o suporte à segurança, desempenho e confiabilidade (51).

Problemas provenientes do grande volume de dados impedem o progresso de abordagens IoT em todas as fases do ciclo de vida. A enorme quantidade de dados requer arquiteturas consolidadas para gerenciamento, com o objetivo de maximizar e organizar os recursos da rede, porém não existe um padrão amplamente aceito para IoT (52). Há uma necessidade de gerenciamento de *clouds* para hospedagem e fornecimento de aplicações IoT, compartilhando infraestrutura e recursos de tal forma que sejam satisfeitos os requisitos de QoS dos usuários. Detecção de eventos

em tempo real é um grande desafio para estruturas sensíveis ao contexto no paradigma IoT (52).

Segundo Namiot (53), aplicações IoT devem ser desenvolvidas como micro-serviços e o seu gerenciamento deve ser centralizado e completamente separado destes serviços. Ainda segundo Namiot (53), sistemas IoT orquestram vários dispositivos que individualmente não possuem conhecimento globalizado, fazendo com que o sistema logicamente centralizado seja capaz de gerenciar serviços de cada dispositivo individualmente.

Muitos modelos emergentes de gerenciamento para IoT têm surgido ultimamente. A maioria destes trabalhos são relacionados ao gerenciamento dos dados massivos provenientes da IoT, mas poucos deles abrangem as cinco áreas de gerenciamento FCAPS. FIWARE (54) propõe um conjunto de serviços para gerenciamento e controle de dispositivos IoT. Ele solicita um agente para um ou mais dispositivos IoT serem controlados. O agente traduz protocolos específicos de IoT para o modelo *publish/subscribe* do FIWARE. Isto permite que os serviços FIWARE controlem as alterações de valores ou configurações nos dispositivos IoT. Uma ideia semelhante é proposta pelo *SmartSantander* (55), onde um recente padrão criado para IoT, chamado oneM2M, lançou um modelo para o gerenciamento de dispositivos.

A *Technical Report* 0005 (56) aborda modelos de uso para comunicação M2M (máquina para máquina) que incluem aplicações que devem informar o status do equipamento e avisar imediatamente se houver uma falha ou se o sistema atingir limiares previamente configurados. Gerenciamento e controle remoto para M2M são usados em diversas áreas de atuação como em petróleo e gás, saneamento, energia, automação industrial, saúde e meio ambiente. Estes dispositivos devem fornecer informações de forma oportuna para sistemas de gerenciamento. Um exemplo de atuação de controle/gerência em M2M é a operação da rede hidrométrica brasileira que prevê a instalação de plataformas coletora de dados que transmitem periodicamente informações meteorológicas para a Agência Nacional de Águas (ANA). Os dados pluviométricos e fluviométricos são enviados automaticamente para o sistema de gerência da ANA via *WebServices*, o que ajuda as autoridades brasileiras a evitar problemas como deslizamentos e enchentes (57).

4.1.1 FIWARE

Para o controle de rede, o FIWARE usa um controlador SDN chamado *OpenFlow Network Information and Control* (OFNIC) que monitora o status da rede e fornece quase em tempo real estatísticas com diferentes níveis de granularidade. Também é capaz de controlar os recursos de encaminhamento da rede.

Para o gerenciamento da rede IoT, FIWARE usa um componente chamado IoT *Agent* que translada o agente de um elemento legado em um agente IoT com suporte à plataforma FIWARE. Todo equipamento deve estar ligado ao *Context Broker* que é um ponto de acesso para dados de forma contextualizada. O *Context Broker* recebe e contextualiza os dados provenientes de diferentes fontes.

O IoT *Agent* é um servidor HTTP onde é possível configurar URLs que estão associadas aos protocolos de diversos equipamentos gerenciados e que são implementados como módulos ao IoT *Agent*. Se o elemento possuir uma API *Next Generation Standard Interface* (NGSI-9/NGSI-10) integrada ao *Context Broker* (Orion), não será necessário o componente IoT *Agent* conforme ilustra a Figura 4. O *Orion Context Broker* é o componente FIWARE chamado de *Generic Enablers* - GE que provê o gerenciamento de contexto dos dados provenientes dos dispositivos gerenciados por meio de mensagens *pub/sub*. Para armazenar e publicar as informações históricas dos dados de contexto gerados ao longo do tempo, FIWARE utiliza o *Comprehensive Knowledge Network* (CKAN) que é uma plataforma web de código aberto que provê interface gráfica e ferramentas com recursos para gerenciamento dos dados.

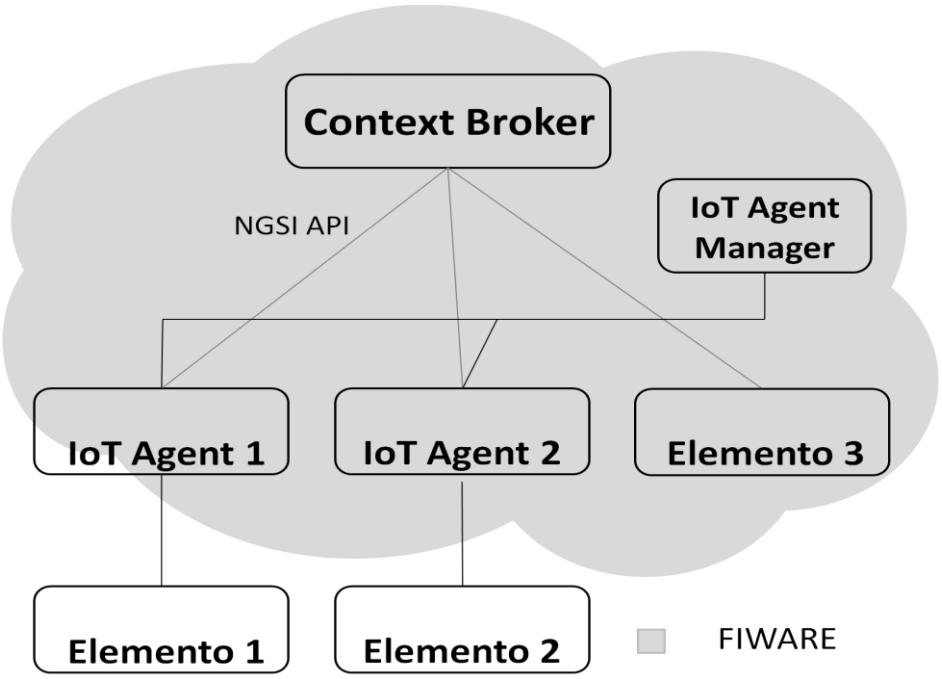

Figura 4 – *Arquitetura de Gerenciamento FIWARE.*

4.2 Novos Requisitos de Gerência e Controle de Redes sob a ótica das Redes Definidas por *Software*

O paradigma SDN pode fornecer melhores mecanismos de gerenciamento em comparação com métodos legados (58). SDN permite que várias tarefas de gerenciamento sejam executadas com alto grau de flexibilidade porque é mais fácil introduzir características adicionais sobre o controlador de rede, da mesma forma que é feito com aplicativos em um sistema operacional. A abordagem de controle logicamente centralizado pode favorecer a eficácia do controle da rede (58).

Em redes tradicionais, os planos de encaminhamento e controle estão confinados dentro de cada dispositivo de rede em um conjunto bem definido de protocolos padronizados. Com o desacoplamento dos planos, como promove a SDN, configurá-los pode ser particularmente complexo, considerando que ambos os planos

podem operar com protocolos SDN. Segundo Wickboldt (14), protocolos SDN devem incluir interfaces de gerenciamento para otimizar a configuração nestas redes.

Os controladores podem ser instalados de forma centralizada ou distribuída. O controlador centralizado, mais comumente implementado, tem a desvantagem de apresentar um ponto único de falha. Ele também pode não ser suficiente para escalar o gerenciamento no plano de dados em redes com grande quantidade de nós (59). Enquanto que o controlador distribuído, que é implementado de forma modularizada, permite escalabilidade e maior resiliência às falhas. No entanto, o gerenciamento da consistência dos estados de todos os seus componentes é um obstáculo.

O OpenFlow é um protocolo *southbound*, ou seja especializado em se comunicar com os elementos de rede, focado apenas no encaminhamento de pacotes e em configuração de dispositivos. Outras áreas de gestão, como desempenho, contabilidade e segurança são ainda pouco exploradas. Um desafio aberto é expandir a gerência para outras funcionalidades. A forma como controladores se comunicam com outros controladores também é pouco explorado. A questão do âmbito limitado do Openflow emerge em outras abordagens SDN como o Procera (58) que é um *framework* que melhora a gerência e controle da configuração e usa OpenFlow para se comunicar com o equipamento. SNMP pode ser usado como uma fonte de eventos para o controlador Procera. O Opflex (60) da Cisco difere do OpenFlow porque centraliza somente as definições das políticas de controle no chamado *policy manager*. A inteligência quanto às decisões de controle/gerência fica distribuída nos nós da rede.

A abordagem *Decentralize*-SDN (D-SDN) (61) propõe um modelo descentralizado no plano de controle por meio de controladores principais e secundários. Neste cenário, os controladores são hierarquicamente distribuídos. Os principais delegam atividades de controle para elementos secundários, de modo a aumentar a disponibilidade do plano de controle e melhorar a tolerância às falhas.

4.3 Novos Requisitos de Gerência e Controle de Redes sob a ótica das Redes Centradas em Conteúdo

FIAs devem reconhecer a estrutura das informações (ex.: arquivos, contratos, base de dados, etc) e como acessar essa informação de forma pontual, confiável, segura e privada e ainda que seja independente da localização. O mais difícil é contextualizar e correlacionar informações de forma correta em meio a uma grande quantidade de dados sendo detectados no mundo físico. O modelo *host-centric* da Internet atual dá lugar ao *information-centric* que suporta o acesso à informação independente da localização, uma das limitações da Internet atual. Arquiteturas CCN emergentes, como *Named Data Networking* (NDN) (62) e *Network of Information* (NetInf) (63) estão adotando um modelo de controle e gerenciamento distribuído, ou seja, cada nó CCN gerencia seu próprio conteúdo, armazenado na *content store* que é um *cache* dos conteúdos. Pesquisas CCN com foco no monitoramento de rede não têm sido amplamente desenvolvidas (64).

O maior desafio para as iniciativas CCN em relação à gerência e controle é como gerenciar os dados armazenados em *cache* (65). Deve ser necessário um mecanismo de gerenciamento colaborativo entre os nós uma vez que todos potencialmente devem possuir recursos para armazenamento. O armazenamento em *cache* é necessário para distribuir conteúdo, e a forma como eles são gerenciados está diretamente relacionada ao aumento da disponibilidade dos dados. Em (66), os autores propõem uma arquitetura de gerência orientada ao conteúdo apresentada como *Name-Oriented Network Management* (NONM) que permite o gerenciamento de nós CCN por meio do protocolo SNMP.

As abordagens citadas acima permitem que sistemas de gerência legados como SNMP e *IP Flow Information eXport* (IPFIX) interoperem com arquiteturas emergentes CCN.

4.3.1 CCNx

CCNx é um protótipo CCN de código aberto implementado pelo *Palo Alto Research Center* (PARC) do grupo *Xerox*. CCNx usa nomes hierárquicos com estrutura similar ao URL usado na Internet e suporta o protocolo IP.

No CCNx, o conteúdo é requisitado por meio do pacote de interesses e pode ser enviado por qualquer nó que possua uma cópia em *cache*. Os pacotes de interesse e de dados são encapsulados em *User Datagram Protocol* (UDP). Os nós possuem uma tabela de interesses pendentes (*Pending Interest Table* – PIT) que armazenam os pacotes de interesse. Ao longo do caminho entre o requisitante e a fonte, as mensagens de interesse podem ser comparadas com os *caches* dos nós e assim não é necessário que se faça todo o percurso até a fonte do conteúdo. Uma vez encontrado o conteúdo desejado, ele é enviado seguindo o caminho inverso percorrido pelo pacote *interest packet*. CCNx possui um elemento chamado *Forwarding Information Base* (FIB) para fazer roteamento de dados por meio de roteadores de conteúdos.

Em (64), foi proposto um modelo para o monitoramento do estado dos nós CCNx e estatísticas de tráfego usando SNMP e IPFIX. O IPFIX foi usado para capturar e analisar a informação do fluxo durante a distribuição dos conteúdos e o SNMP para identificar o estado dos nós CCN por meio da definição de uma MIB II para CCN. Cada nó CCN possui um agente SNMP com as variáveis mapeadas das tabelas *Content Store*, PIT e FIB. Além do agente SNMP, os nós também possuem um agente IPFIX para detecção do fluxo de chegada dos pacotes. Os dados são enviados para o servidor/coletor que fará a análise de desempenho do fluxo.

4.3.2 Network of Information (NetInf)

Network of Information (NetInf) (63) é uma arquitetura desenvolvida pela iniciativa europeia 4WARD onde os conteúdos são nomeados por nomes planos autocertificáveis para roteamento e resolução de nomes. NetInf possui diferentes protocolos para o sistema de resolução de nomes, inclusive aqueles baseados em tabelas *hash* (*Distributed Hash Tables* – DHTs). Geralmente, é o serviço de

resolução de nomes que descobre onde o conteúdo está armazenado, mas ele pode usar um protocolo que faz uso da ciência de *cache* (*cache-aware NetInf transport protocol*) que também informa o caminho onde o conteúdo está localizado.

Em (67) são explorados aspectos de autogerenciamento para NetInf uma vez que o projeto inicial não cobria aspectos de gerência. Desta forma, foi introduzido o conceito de *In-Network Management* (INM) que cobre as cinco áreas funcionais FCAPS. A arquitetura INM é baseada em entidades que têm propriedades autonômicas (36) que se autoconfiguram conforme a demanda, analisando as informações recebidas da rede, políticas de alto nível e SLA. As funcionalidades de gerência FCAPS são delegadas aos nós NetInf com componentes INM que possuem processos de autogerenciamento e interagem ponto a ponto com outros nós colaborando e/ou consumindo serviços deles.

4.4 Novos Requisitos de Gerência e Controle de Redes sob a ótica da Virtualização das Funções de Rede

Novas arquiteturas de rede certamente implementarão muitos serviços na forma de *software*, virtualizando (emulando) componentes físicos. Sistemas de gerenciamento e controle devem ser operacionais em tempo integral. Recursos de alta disponibilidade, combinados com dados de alta confiabilidade são fundamentais para alcançar este objetivo.

Virtualização de rede corrobora com tecnologias como SDN, uma vez que a virtualização otimiza os recursos de *hardware* e melhora as atividades de controle/gerência, tal como o controle de latência e desempenho da rede. As abordagens SDN e NFV, apesar de complementares, não possuem soluções de gerenciamento que combinem ambas as tecnologias de forma integrada (68). Ainda segundo (68), o suporte à Contabilização (*Accounting*) é uma área funcional FCAPS esquecida nas atuais soluções de gerenciamento em NFV. Prover soluções de gerenciamento e controle que atendam integralmente às 5 áreas funcionais FCAPS é um desafio em aberto. Se considerarmos que o NFV é uma realidade no mundo das

redes de telecomunicações, torna-se imperativo o suporte à contabilização como funcionalidade para as soluções de gerenciamento.

No ecossistema NFV, sistemas de gerenciamento devem ser capazes de orquestrar um grande número de funções virtuais de rede atendendo requisitos como escalabilidade, segurança, autocorreção em caso de falha, bem como portabilidade de *software* de vários fabricantes. Além disto, o sistema de gerenciamento em NFV deve suportar mecanismos que permitam monitorar recursos dinamicamente.

Soluções têm sido propostas para gerenciar redes virtuais. OpenStack e OpenNebula são plataformas capazes de gerenciar recursos na nuvem usando IaaS.

Redes virtuais para FI devem estar preparadas para gerenciar ambientes com um grande número de nós físicos. Em função disto, técnicas de gerenciamento autonômico (36) vêm sendo usadas para gerenciar o substrato da rede. Substrato é o conjunto de recursos físicos de uma rede virtual. A maioria das iniciativas de FI não propõem técnicas de autogerenciamento (67). Em (69), os autores apresentam uma gestão auto-organizada em NFV com base no monitoramento de controle de *loops* e medidas de nós locais e vizinhos através de um gerenciador virtual distribuído em cada nó.

4.4.1 Management and Orchestration (MANO)

Management and Orchestration (MANO) é um *framework* proposto pelo *European Telecommunications Standards Institute* (ETSI) que prove funcionalidades para o aprovisionamento das Funções de Rede Virtuais (*Virtual Network Functions* – VNF) que incluem a orquestração e o gerenciamento de recursos físicos e virtuais que suportam NFV.

MANO possui 2 gerentes funcionais e 1 orquestrador: (i) O VNF *Manager* (VNFM) é responsável pelo ciclo de vida de gerenciamento das VNFs; (ii) o *Virtualized Infraestructure Manager* (VIM) que controla e gerencia os recursos de uma dada infraestrutura NFV e (iii) o NFV *Orchestrator* (NFVO) que lida com a orquestração de recursos e serviços da infraestrutura NFV (68). São responsabilidades do NFVO as tarefas relacionadas ao ciclo de vida de

gerenciamento dos serviços de rede, tais como: análise de medidas de desempenho, correlação de eventos, validação e autorização de recursos da infraestrutura NFV.

A Figura 5 ilustra os blocos funcionais do *framework* MANO.

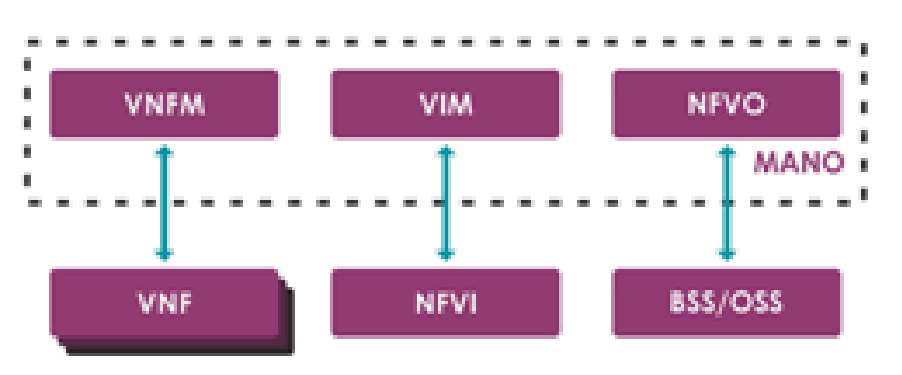

Figura 5 – *Framework ETSI NFV MANO.*

A Operadora de Telecomunicações Telefonica implementou o *framework* MANO em código aberto chamado de OpenMANO (70). O OpenMANO possui, além dos blocos funcionais VIM e NFVO, uma interface gráfica amigável (*Graphical User Interface* - GUI) e integra ao gerenciamento um controlador SDN. Com relação ao atendimento FCAPS, OpenMANO possui as funcionalidades de configuração, contabilização e desempenho. As áreas funcionais Falhas e Segurança não foram implementadas no OpenMano, o que preocupa no contexto da Internet das Coisas.

As principais iniciativas baseadas no *framework* MANO tais como OpenMano, OpenNFV (implementada pela americana *Hewlett-Packard*) e CloudBand (da francesa Alcatel-Lucent) focam em soluções de gerenciamento centralizadas e portanto apresentam limitações de escalabilidade (68). A adoção de um gerenciamento distribuído facilitaria o monitoramento para melhor reagir às mudanças inerentes à abordagem NFV.

4.5 Recursive InterNetwork Architecture (RINA)

A arquitetura de inter-rede recursiva (RINA –*Recursive InterNetwork Architecture*) é uma proposta de FIA originada a partir do livro "*Patterns in Network Architecture: A Return to Fundamentals*" escrito pelo professor John Day da Universidade de Boston em 2008. RINA (71) defende que as redes de comunicação servem apenas para comunicação entre processos (IPC – *Inter Process Communication*). Assim sendo, camadas recursivas são formadas para permitir IPC. RINA separa as funções de controle e gerenciamento, implementado-as em IPC *Control* e IPC *Manager*, respectivamente. O IPC *Manager* realiza a alocação de fluxos e de recursos para as camadas da rede, implementando autenticação, *parser*/gerador de mensagens do protocolo *Common Application Protocol* (CDAP), e mantendo uma *Resource Information Base* (RIB) *Daemon*, que se assemelha a uma MIB. Já o componente IPC *Control* realiza o controle de transmissão e retransmissão e controle de fluxo, com suporte à qualidade de serviço.

Um conjunto de processos IPC forma um *Distributed IPC Facility* (DIF) que trabalha como uma camada de rede. Em RINA, os planos de gerência e controle são implementados juntos aos processos IPC. Além do DIF, a RINA também emprega o conceito de *Distributed Application Facility* (DAF). Um DAF é um conjunto de processos que se unem para executar uma determinada aplicação distribuída. O DAF pode executar as seguintes funções de gerenciamento (72):

1) Gerenciamento de um único DIF: Monitora o estado entre os processos IPC de um DIF. Os processos IPC publicam periodicamente o seu estado e assinam a publicação do estado dos *links* de outros processos IPCs com o objetivo de tomar ciência do estado do DIF e se autoconfigurar para responder às mudanças de estado da rede;
2) Gerenciamento de um conjunto de DIFs: Neste caso, o DAF desempenha a função de gerenciamento centralizado de rede.
3) Gerenciamento de Aplicações: É feita hierarquicamente, distribuindo solicitações para os DIFs.

A arquitetura RINA utiliza o protocolo CDAP para coordenar o conjunto de operações nos objetos gerenciados e usa a RIB, que funciona como uma MIB, reunindo todos os objetos gerenciados e seus atributos. Os eventos de gerência são assinados baseados no modelo pub/sub. As assinaturas do RIB *Daemon* são análogas às notificações de eventos do modelo OSI e às *traps* do SNMP.

4.6 Resumo do Capítulo

Este item apresenta reflexões com os principais requisitos e desafios de gerência e controle relatados neste capítulo. Está claro que para suportar as contínuas mudanças associadas às necessidades da Internet, os requisitos para gerenciamento e controle de redes devem ser adaptáveis de forma que garanta a interoperabilidade com diversos elementos de rede e ainda com novas arquiteturas que possam surgir.

A nova geração de arquiteturas de redes, como as que estudamos neste capítulo (FIWARE, SDN, CCN, NetInf, NFV e RINA), demandam recursos virtuais onipresentes. Para gerenciar e controlar todos esses recursos, as redes de nova geração requerem que as aplicações de gerenciamento e controle assegurem níveis de qualidade de serviço que incluem disponibilidade, confiabilidade, usabilidade, desempenho, segurança, configuração e custo, Os níveis de qualidade serão conseguidos por meio de políticas bem definidas para gerenciamento e controle de rede implementadas através da colaboração entre recursos, tornando-os mais gerenciáveis e abrangendo todas as cinco áreas de gerenciamento FCAPS introduzidas pelo TMN.

Semelhanças e diferenças entre os modelos de controle e gerência estudados neste capítulo serão resumidas nas tabelas comparativas do capítulo 6.

5 Proposta para Controle e Gerenciamento na NovaGenesis

Neste capítulo é apresentada uma nova arquitetura de gerenciamento e controle capaz de operar sobre os paradigmas IoT, SDN, CCN e NFV em particular com foco nos requisitos e desafios de gerência e controle relatados no Capítulo 4. A arquitetura permite o controle e gerenciamento de elementos implementados em *software* e *hardware* de forma elástica, segura, escalável, utilizando redes de confiança e que acomode novos requisitos para gerenciamento e controle. Para isso, a arquitetura deve ser flexível e adaptável o suficiente para lidar com o ambiente dinâmico e convergente encontrado nas FIAs. Novas funcionalidades podem ser adicionadas sem alterar os componentes já existentes. A arquitetura prevê uma estrutura organizada e genérica de operação capaz de se integrar facilmente a um grande número de equipamentos heterogêneos.

O gerenciamento e controle da NovaGenesis é híbrido, ou seja, compreende linguagens e tecnologias diferentes para adaptar as constantes mudanças e as novas arquiteturas. Também é construído em blocos para suportar a demanda do mercado que não permite implementações longas. Assim sendo, esta proposta descreve a maneira que o gerente e o controlador devem compartilhar as informações com os elementos gerenciados ou agentes no ambiente NovaGenesis.

Existem vários padrões para implementação da arquitetura de gerenciamento e controle. É possível que FIAs tenham que conviver com várias propostas de novas arquiteturas em um ambiente híbrido. É descrito o cenário de gerenciamento e controle NovaGenesis para assegurar a interoperabilidade adaptável às mudanças inerentes à FI. A arquitetura é simples, escalável e elástica. A escalabilidade é garantida porque é utilizada uma arquitetura distribuída no *Southbound*. A simplicidade é conseguida com gerenciamento e controle logicamente centralizado no *Northbound* e a elasticidade é requerida para lidar com múltiplas solicitações simultâneas, de modo que todos os elementos da arquitetura consumam recursos de maneira eficiente.

A arquitetura de Gerenciamento e Controle NovaGenesis é baseada no modelo *Everything-as-a-Service* (73), onde o *Northbound* é composto por serviços

gerente e um controlador visando formarem juntos um plano de orquestração que usa o serviço pub/sub comum da NovaGenesis. Elementos do *Southbound* não podem acessar diretamente o *Northbound* por razões de segurança e também para garantir o desempenho destes elementos que são a inteligência da rede. O elemento responsável por mediar a comunicação entre os planos *Northbound* e *Southbound* é o *Mediator Service*. Neste contexto, *Mediators Services* devem expor as características de diferentes protocolos de gerenciamento e controle para a NovaGenesis, estabelecendo contratos e controlando agentes a fim de otimizar os recursos de rede de forma perfeita e automática, usando todos os novos recursos de uma rede convergente programável. A comunicação entre os elementos gerenciados (agentes) e *Mediator Services* será iniciada após a assinatura dos contratos entre o agente e o mediador.

O plano de orquestração trabalha conferindo privilégios de orientação aos serviços da NovaGenesis com o objetivo de organizar e estruturar a rede do ponto de vista macro. A Figura 6 ilustra uma visão holística da rede com todos os elementos que compõem de forma hierárquica a Gerência e Controle da NovaGenesis. Os elementos que compõem a arquitetura de Gerência e Controle NovaGenesis serão descritos nos próximos itens.

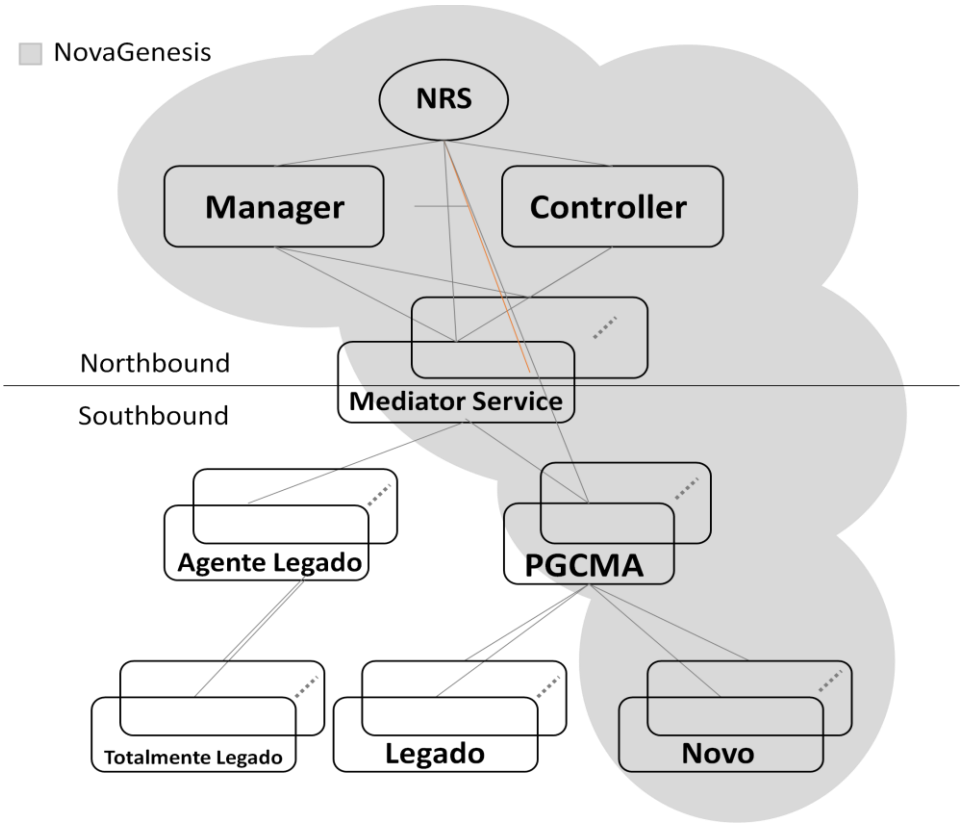

Figura 6 – *Modelo de Gerenciamento e Controle NovaGenesis.*

5.1 Elementos da Arquitetura

5.1.1 *Manager* - Gerente NovaGenesis

A gerência é feita por uma lógica centralizada no serviço gerenciador que integra todas as cinco áreas funcionais de gestão FCAPS e deve ter conhecimento do estado atual da rede/nuvem, analisando e detectando a causa raiz das falhas, tomando decisões e agindo em elementos gerenciados, com o objetivo de otimizar constantemente a rede/nuvem. O gerente NovaGenesis inclui aspectos de gerência para FIAs, abrangendo os requisitos discutidos no Capítulo 4, que incluem alta robustez, segurança, heterogeneidade e escalabilidade dos elementos gerenciados e possui capacidade de adaptação às inovações inerentes às redes de nova geração estudadas neste trabalho.

O gerenciador, assim como o controlador (descrito adiante), segue o modelo orientado a serviços onde a gerência e controle é vista como um serviço que pode executar ou delegar tarefas para administrar e operar os recursos da rede, oferecendo a dissociação entre produtores e consumidores por meio do *Name Resolution Service* (NRS).

O gerente de rede centralizado terá uma visão global da rede, mas não significa que seja um ponto único de falha. Ele será modularizado de modo que se uma das instâncias falhar perde-se um pouco da capacidade, mas não se perde toda a função do gerente, ou seja, o Gerente NovaGenesis será resiliente à falhas. A informação de gerência e controle fica distribuída de forma coerente no NRS. Assim, todos os gerentes/controladores usam a mesma base de dados para as suas ações.

5.1.2 *Controller* - Controlador NovaGenesis

O Controlador de rede NovaGenesis segue o ciclo de vida dos serviços, colaborando com a composição de serviços, exposição, descoberta e contratação de recursos, QoS e segurança. Possui o objetivo de analisar informações de tráfego, encaminhamento, roteamento, controle de congestionamento, implementação de políticas e mecanismos de controle de acesso, conflitos de recursos e governança, configuração de parâmetros e funcionalidades, etc.

Este componente compreende integralmente aspectos de controlador IoT, SDN, CCN e NFV, o que inclui análise do estado da rede, publicação de informações sobre a topologia, descoberta de dispositivos conectados à rede, distribuição de configurações de encaminhamento da rede e mecanismos de segurança entre os serviços.

5.1.3 Mediator Service (MeS)

A mediação é feita por um serviço especial responsável pela tradução de informações e protocolos. O *Mediator Service* (MeS) implementa um serviço para

enviar e receber dados de elementos gerenciados e um agente para se conectar ao Gerente NovaGenesis. A interoperabilidade é percebida na inteligência do *Mediator Service* que compreende vários protocolos e linguagens, simultaneamente. Estes elementos representarão os protocolos de gerenciamento no ambiente NovaGenesis. Ele irá gerenciar os elementos puramente NovaGenesis através de protocolo único implementado sobre o NRS, bem como gerenciar elementos legados de forma transparente.

O *Mediator Service* delega a responsabilidade de gerenciar e controlar a rede para o Plano de Gerenciamento e Controle. O *Mediator Service* deve realizar as seguintes tarefas:

- **Agregação:** O *Mediator Service* concentra os dados de gerenciamento e gera informação para enviar ao Gerenciador e Controlador que podem ser completamente diferente da informação original (exemplo: informação agregada, médias estatísticas, etc);
- **Filtragem:** O *Mediator Service* filtra os dados de gerência e envia os dados mais importantes para o Gerente NovaGenesis;
- **Correlação:** O *Mediator Service* correlaciona dados, determina padrões e os publica para o Gerenciador e Controlador.
- **Inteligência:** Combina todas as ferramentas de dados para prover informação contextualizada ao Gerente e Controlador.

5.1.4 Proxy/Gateway/Controller and Management Agent (PGCMA)

A função do PGC foi descrita no capítulo anterior. Na perspectiva do controle de rede, o PGCMA é um controlador especializado para equipamentos novos e legados. Ele representa esses equipamentos mantendo contrato com um ou mais serviços de controle de rede. Ele pode ser definido como um *Proxy*, representando equipamentos para o plano de controle. Desta forma, a coerência das ações de controle segue uma hierarquia definida pelos controlados. Ele possui também a

função de *gateway* que é capaz de se conectar a um elemento legado. Quando o PGCMA representa/conecta a um elemento NovaGenesis, ele não usa as funções de *gateway*, pois elas não são necessárias.

5.1.5 Name Resolution Service (NRS)

O NRS é um macro sistema formado por três componentes NovaGenesis para nomeação, armazenamento e entrega de conteúdos e ligações entre nomes armazenados no *cache* de rede distribuída. São eles: HTS, GIRS e PSS. Eles proveem nomeação, resolução de nomes, armazenamento em *cache* e entrega de nomes e conteúdos para o PGCMA. O NRS fornece o acesso autenticado/autorizado a conteúdos e nomes.

5.1.6 Agente Legado

É um agente implementado usando algum gerenciamento ou controle legado, como o SNMP.

5.1.7 Elemento Totalmente Legado

É um equipamento tradicional de gerência e controle, como elementos SNMP ou componentes de roteamento TCP/IP, controladores Ethernet, *access point* Wi-Fi, *no-breaks, modens, switches,* etc.

5.1.8 Elemento Legado

O elemento legado é um equipamento tradicional que pode ser controlado pelo PGCMA, pois ele implementa a função de *gateway* capaz de interoperar com

elementos do plano de controle e gerenciamento legado. Neste cenário, o PGCMA irá transladar os objetos da MIB do elemento legado e os traduzem para uma mensagem NovaGenesis conforme protocolo definido adiante.

5.1.9 Elemento Novo

É um elemento puramente NovaGenesis que publica e notifica o PGCMA quando algum evento ocorre. O elemento NovaGenesis interage diretamente com o PGCMA por meio de mensagens NovaGenesis usando protocolo definido adiante.

O elemento NovaGenesis deve ter processos que gravam no PGCMA as propriedades do elemento/objeto.

5.1.10 Modelo de Interação

Os elementos da arquitetura NovaGenesis descritos neste capítulo interagem entre si formando um ecossistema de gerenciamento e controle ilustrado na Figura 7.

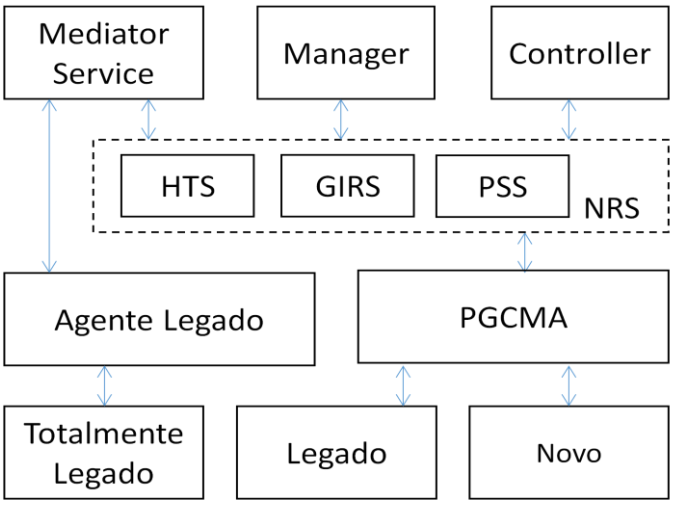

Figura 7 – *Modelo de Interação entre novos serviços para gerenciamento e controle*.

5.2 Protocolo

O *COmmon NovaGenesis Protocol* (CONGEP) é o protocolo único de aplicação da NovaGenesis que especifica como será realizada a troca de informações na rede com o objetivo de facilitar a interoperabilidade dos elementos e módulos do sistema da forma mais genérica possível. O CONGEP é usado em toda comunicação da NovaGenesis e não somente para comunicação de gerenciamento e controle. O CONGEP foi baseado nos protocolos tradicionais de gerenciamento de rede, SNMP e CMIP, mas seguindo os paradigmas da NovaGenesis. O CONGEP é simples como o SNMP e seguro como o CMIP.

As mensagens do protocolo usam a interface já existente com as primitivas descritas na seção 3.5. As ações que deverão ser executadas nos elementos gerenciados são anexadas às mensagens NovaGenesis. Quando a mensagem possui uma ação, ela deverá vir com uma linha em branco e um *payload* que é a carga útil da mensagem conforme ilustra a Figura 8. A carga útil será um arquivo com extensão XML, JSON ou TXT que conterá as ações a serem executada nos elementos. Quando um bloco funcional recebe uma mensagem NovaGenesis ele fará o *parser* desta mensagem executando cada linha de comando do arquivo do *payload.*

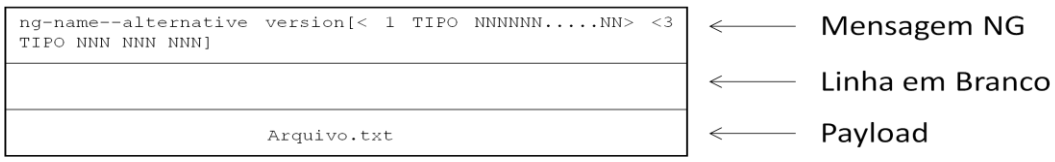

Figura 8 – *Estrutura da Mensagem NovaGenesis.*

Todos os elementos NovaGenesis que fazem parte da arquitetura proposta para gerenciamento e controle devem estar aptos a interoperar com as mensagens do protocolo CONGEP. O PGCMA publicará as informações de gerenciamento e controle para o *Mediator Service* por meio de mensagens CONGEP. O *Mediator Service*, por sua vez, notificará as informações coletadas para o *Manager* ou *Controller* também por meio de mensagens CONGEP. Desta forma, no âmbito da

arquitetura de gerência e controle as mensagens NovaGenesis possuem as seguintes funções:

- *Publish*: Os elementos gerenciados publicam os dados de gerenciamento para o gerente. O pub também é utilizado para avisar ao gerente um determinado comportamento ocorrido no sistema. Funciona como uma *trap* SNMP, no contexto de disseminador de dados ou eventos.
- *Subscribe:* O gerente assina os dados publicados pelos elementos gerenciados e toma ciência deles.
- *Notify*: A mensagem *notify* é utilizada para avisar eventuais clientes quando um determinado dado de interesse é publicado.
- *Revoke*: Um evento de gerência e controle já publicado anteriormente pode ser cancelado pelo elemento que publicou o evento por meio de uma mensagem *revoke*.

Para informar aos elementos de gerência e controle que um parceiro tem interesse em aceitar a oferta de serviço, a mensagem *subscribe* virá acompanhada de uma notificação. Desta forma, o elemento notificado saberá quando poderá dar início à formulação do contrato, dando prosseguimento ao ciclo de vida do serviço. A estrutura de uma mensagem *subscribe* com notificação será a mesma da mensagem *publish* com notificação conforme abaixo:

ng -s --notify VERSÃO [< TAMANHO TIPO CATEGORIA > < TAMANHO TIPO CHAVES > < _TAMANHO TIPO sub HID OSID PID BID >]

5.2.1 Mecanismos para Gerência de Falhas

O comando *notify* da NovaGenesis virá com o nível de criticidade da mensagem. Neste cenário, as mensagens *notify* irão se comportar como uma *trap* SNMP para notificação de eventos. Os níveis de severidade das mensagens ou alarmes seguirão a definição da recomendação X.733 (74) da ITU amplamente utilizada na gerência de redes que define 6 níveis. São eles:

Cleared: Indica quando um alarme já reportado deixou de estar em falha;

Indeterminate: Quando não foi possível determinar o nível do alarme;

Critical: Indica que o alarme deve ser tratado imediatamente;

Major: Indica que o alarme é urgente;

Minor: Indica a existência de um alarme que não afeta a condição do serviço, mas que deve ser corrigido;

Warning: Indica a detecção de uma potencial condição de falha.

Um vetor de indicação de prioridade de notificação virá no inicio da mensagem *notify* estruturada da seguinte forma:

```
ng -notify --s VERSÃO [< TAMANHO TIPO PRIORIDADE > < TAMANHO TIPO CATEGORIA
> < TAMANHO TIPO CHAVE > < TAMANHO TIPO VALORES > < _TAMANHO TIPO pub HID
OSID PID BID > ]
```

A ação para tratar o comando *notify* é de responsabilidade dos serviços de gerência e controle (e.g. MeS, *Manager* e *Controller*). O nível de severidade de alarmes na mensagem *notify* é utilizado para dar prioridade ao tratamento da mensagem, ou seja, as mensagens não serão tratadas por ordem de chegada, mas sim por ordem de prioridade nos serviços de gerência e controle prestados. Todavia, se houver duas mensagens de mesma prioridade, o tratamento é por ordem de chegada.

Considera-se importante fornecer as notificações de alarme (mensagens *notify*) com um estilo padronizado, usando um conjunto comum de notificações, com parâmetros padronizados e as definições de parâmetro. As notificações podem ser importadas para qualquer tipo de objeto gerenciado. Desta forma, o arquivo de *payload* para notificação de falhas deverá conter no mínimo os seguintes parâmetros:

Parâmetro 1: Nome

Atributo: *string* contendo o nome do objeto em que ocorreu o alarme.

Parâmetro 2: Tipo
Atributos: *Communications alarm, Quality of service alarm, Processing error alarm, Equipment alarm, Environmental alarm.*
Comentário: As causas prováveis para cada tipo de alarme descrito acima podem ser vistas na recomendação X.733.

Parâmetro 3: Valor
Atributo: *string* contendo o valor medido.

5.2.2 Mecanismos para Gerência de Desempenho

Para fins de gerenciamento de desempenho, quando uma medida for alterada, o elemento gerenciado publicará as novas medidas e notificará o gerente. O *payload* para fins de gerência de desempenho conterá os seguintes parâmetros:

Parâmetro 1: Nome

Atributo: *string* contendo o nome do objeto medido.

Parâmetro 2: Tipo
Atributos: *em branco*

Parâmetro 3: Valor
Atributo: *string* contendo o valor medido.

Pode ocorrer do elemento gerenciado se comportar de forma passiva. Neste caso, o gerente enviará uma publicação com o comando tipo *get* conforme definido abaixo nos mecanismos de gerência de configuração.

5.2.3 Mecanismos para Gerência de Contabilização

No gerenciamento de contabilização pode ser usado o mesmo mecanismo proposto para o gerenciamento de desempenho. Ou pode-se adotar um sistema de geração de bilhetes. Este sistema será responsável pelo armazenamento dos dados da aplicação/serviço para efeito de processamento destas informações para bilhetagem e/ou contabilização de uso dos recursos da rede.

Para a geração de bilhetes, o *payload* deverá conter um arquivo com os campos que permitam monitorar recursos e quanto desses recursos está sendo utilizado por uma entidade.

5.2.4 Mecanismos para Gerência de Configuração

A gerência de configuração também deve ser implementada por meio do protocolo CONGEP que seguirá por uma mensagem *publish* informando os parâmetros atuais de configuração do elemento gerenciado. Estes parâmetros são definidos abaixo e devem ser publicados para conhecimento do módulo responsável.

As operações são usadas para ler ou escrever valores na tabela *hash* no elemento gerenciado. São definidas 02 (duas) operações no protocolo:

Parâmetro 1: Nome

Atributo: *string* contendo o nome do objeto a ser lido/alterado.

Parâmetro 2: Tipo

Atributo: *string* contendo o nome do comando.

*Get:*Esta operação é usada para ler o valor no objeto gerenciado.

Set: Esta operação é usada para atualizar um valor no objeto gerenciado.

Parâmetro 3: Valor
Atributo: valor a ser sobrescrito.

Se a intenção for apagar um valor, a mensagem será pub com *payload set* e o parâmetro valor vazio. Para ler um objeto, deverá ser enviada uma mensagem *pub* com *payload get* e parâmetro valor vazio. Desta forma, com apenas 02 operações é possível criar, apagar, ler, escrever, iniciar e parar qualquer atributo, mantendo a premissa inicial da simplicidade.

A gerência de configuração será realizada por meio do protocolo CONGEP se os elementos gerenciados forem puramente NovaGenesis ou elementos legados cuja MIB seja conhecida pelo PGCMA. Elementos totalmente legados serão gerenciados pelos protocolos de gerência tradicionais.

5.2.5 Mecanismos para Gerência de Segurança: Autenticação, Integridade, Privacidade, Persistência e Proveniência

O protocolo CONGEP possui na mensagem um nome autocertificável (SVN) para fins de teste de integridade no objeto de quem assinou a mensagem. O módulo responsável pela garantia da entrega da mensagem na ordem correta dos dados é o PGCMA, ou seja, ele garante que as mensagens sejam processadas na ordem correta.

O tempo de tratamento de resposta a uma publicação é configurável. Depois de alcançado este tempo será considerado "*timeout*" e a publicação é revogada por meio da mensagem *revoke*. Os serviços retransmitem a publicação indefinidamente ou até que seja assinada. Este mecanismo evita que publicações não importantes e que não devem ser assinadas após certo período de tempo permaneçam na rede e interrompa o bom andamento do ciclo de vida do serviço. De forma diferente das publicações, as subscrições são duráveis e ficam armazenadas no HTS. A mensagem *revoke* é permitida somente para quem gerou a mensagem *pub*, porém publicações de objetos obsoletos de gerência e controle podem ser revogadas via interface NRS.

Além do mecanismo de persistência descrito acima, o PGCMA envia sistematicamente mensagens de *alive* ao PSS para que o *Manager* tome ciência do ambiente e certifique que não houve desconexão do elemento.

5.3 Cenários

Abaixo é descrita a sequência de ações necessárias para completar o fluxo de gerenciamento de um equipamento na NovaGenesis. A Figura 9 propõe um diagrama de sequência para o cenário de controle e gerenciamento orientado a serviços especificado anteriormente. A comunicação entre os serviços de controle e gerenciamento deve ser feita de forma assíncrona pelo PSS localizado no NRS. Todos os conteúdos e nomes publicados são armazenados no NRS. O NRS é responsável por divulgar os eventos gerados por quaisquer serviços de controle e gerenciamento. As entidades de serviço de controle e gerenciamento podem ser produtores ou assinantes de eventos.

1. Inicialização: É o ponto de partida do ciclo de vida de um serviço. Todos os elementos da arquitetura de gerência e controle são executados e instanciados. A etapa de inicialização não é mostrada no diagrama.
2. Exposição: Todos os serviços de controle e gerenciamento expõem (publicando) seus nomes, funcionalidades, limitações, capacidades e condições para a contração do serviço. As palavras-chave são publicadas para o PSS que está localizado no sistema NRS. As ligações entre nomes em linguagem natural e autocertificáveis são armazenadas na HTS;
3. Descoberta: Os serviços clientes assinam as palavras-chave e descrições para selecionar o parceiro apropriado. O serviço cliente assina novamente buscando mais detalhes do serviço;
4. Oferta: Depois de selecionar candidatos a parceiros, os serviços de controle e gerenciamento podem publicar ofertas de acordo com o nível de serviço para os pares recém-descobertos. Os pares convidados analisam e decidem se aceitam ou declinam a parceria. Se a decisão for pelo prosseguimento ao serviço, o par assina a oferta e notifica o parceiro.
5. Contratação: O contrato é formulado. O contrato descreve as responsabilidades de cada parte e estabelece níveis de serviço e qualidade. O parceiro é notificado. O parceiro analisa e assina o contrato. O parceiro acrescenta uma contraoferta à proposta e notifica o cliente. A contraproposta é assinada e analisada. Se for aceita, será publicado um aceite para ser assinado. Se o contrato inicial estiver adequado para o parceiro, não haverá a etapa de contraoferta.
6. Admissão e Operação: Serviços de controle e gerenciamento trocam dados e monitoram os processos que deverão estar de acordo com os níveis pré-estabelecidos em contrato.

Após a carga do sistema, o *Mediator Service* entra em estado de espera (*stand by*) a fim de aguardar as informações de *status* e desempenho de todos os blocos funcionais, tornando-se assim um processo orientado a dados (*data-driven*).

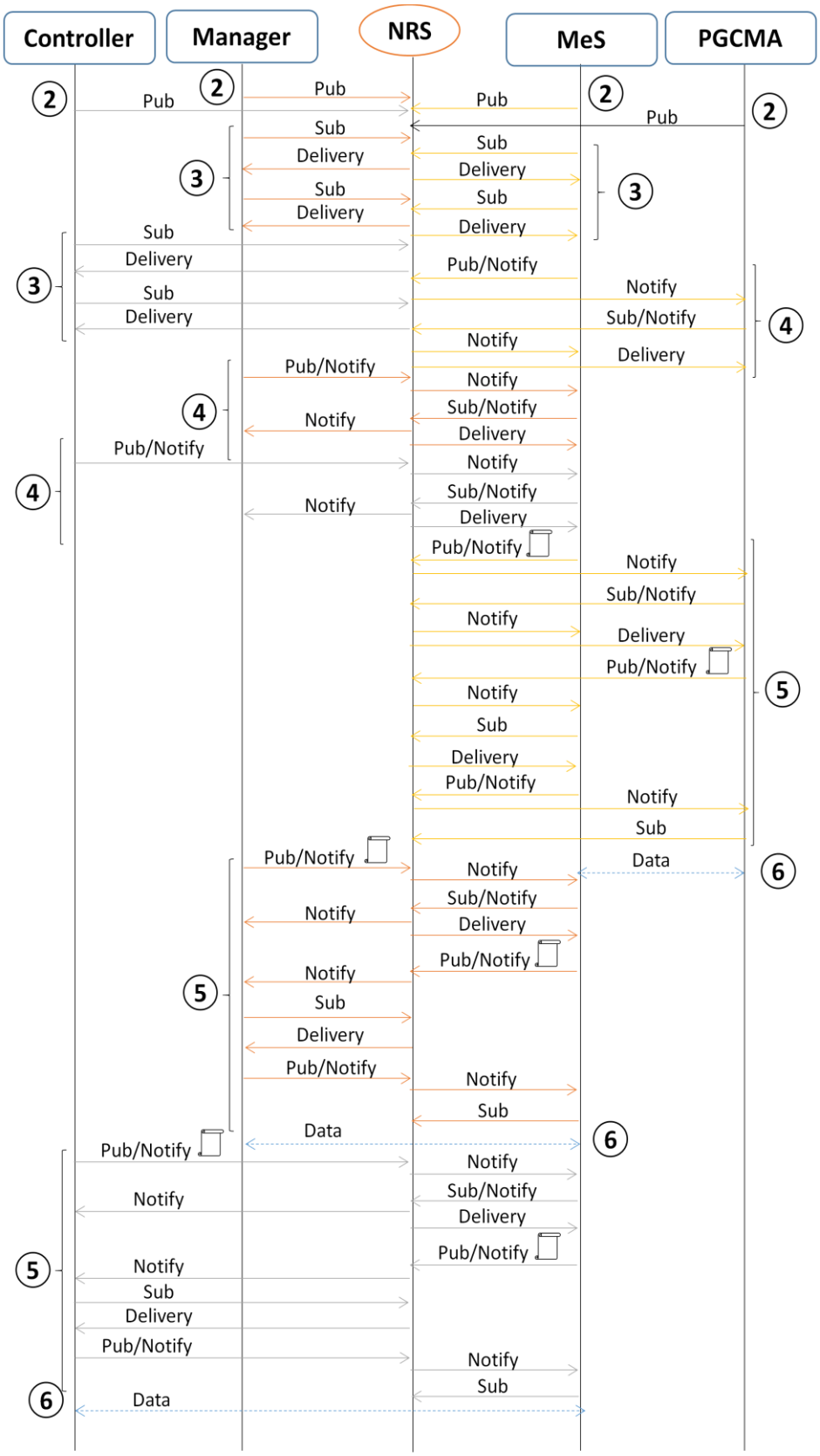

Figura 9 – *Diagrama de Sequencia de Interações de Controle e Gerenciamento.*

O *Manager* e o *Controller* delegam ao PGCMA o conhecimento das atividades de nomeação, inserção, criação, modificação, supressão, armazenamento e entrega de nomes no NRS, uma vez que toda comunicação de gerência e controle deve obrigatoriamente acessar o PGCMA. Deste modo, o PGCMA monitora a integridade, a disponibilidade, o controle de concorrência (*logging*) e segurança (acessibilidade) do NRS. Sobre uma nova ótica, veja o fluxo da Figura 10 para armazenamento dos dados gerenciados no HTS, onde toda comunicação de controle e gerência deve passar pelo PGCMA:

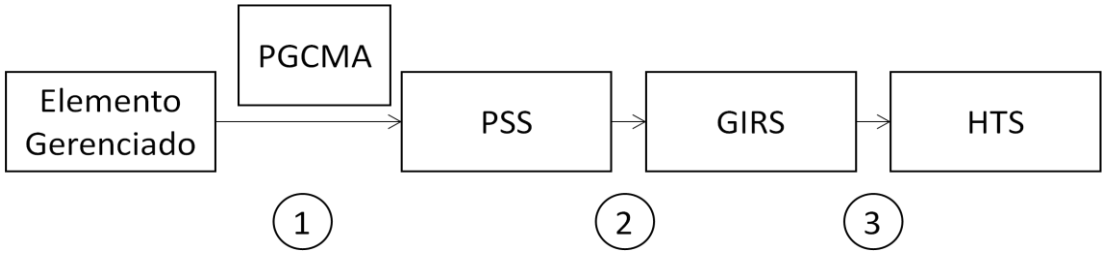

Figura 10 – *Diagrama de Sequencia de Armazenamento de Dados de Controle e Gerência*.

1. Elemento gerenciado envia mapeamento ao PGCMA para ser publicado;
2. PSS faz uma cópia do mapeamento e envia mensagem com mapeamento e conteúdo para o GIRS;
3. GIRS escolhe qual HTS deverá armazenar o mapeamento.

5.4 Estudos de Caso

Nesta seção é feita a análise da implementação realizada em conjunto nos laboratórios do Inatel: *Information and Communication Technologies Laboratory* (ICT Lab) e *Wireless and Optical Convergent Access* (WOCA), coordenado pelo Prof. Dr. Arismar Cerqueira Sodré Júnior. O desenvolvimento baseou-se em um cenário real inserido no ecossistema NovaGenesis para orquestrar o sensoriamento e a gerência espectral de Wi-Fi usando a plataforma de controle de rádio sobre fibra

(75). O objetivo desta implementação foi o de criar uma aplicação que transfere dados entre pontos remotos utilizando os recursos de Wi-Fi, fazendo a troca dinâmica de canais e controlando o rádio através de sinais ópticos (rádio sobre fibra). O objetivo neste estudo de caso é modelar esse cenário real de acordo com o modelo de referência que está sendo proposto nessa dissertação.

A solução implementada pelo ICT Lab/WOCA foi composta por um analisador de espectro que faz o sensoriamento do meio e um *access point* Wi-Fi. A Figura 11 ilustra este cenário (76). O agente (ou serviço) que representa o analisador de espectro é o SSA (*Spectrum Sensing Agent*), enquanto que o *access point* é representado pelo APA (*Access Point Agent*). A RMA (*Resource Management Agent*) faz o papel do gerente de recursos de rádio. Posteriormente, dois outros serviços foram implementados: AAA (*Active Antenna Agent*) e POXA (POX *Agent*) (50). É importante frisar que esses serviços foram implementados antes do modelo de referência para controle e gerência da NovaGenesis tivesse sido concebido. Portanto, esse estudo de caso é aplicado para fins de validação em um cenário já testado na prática.

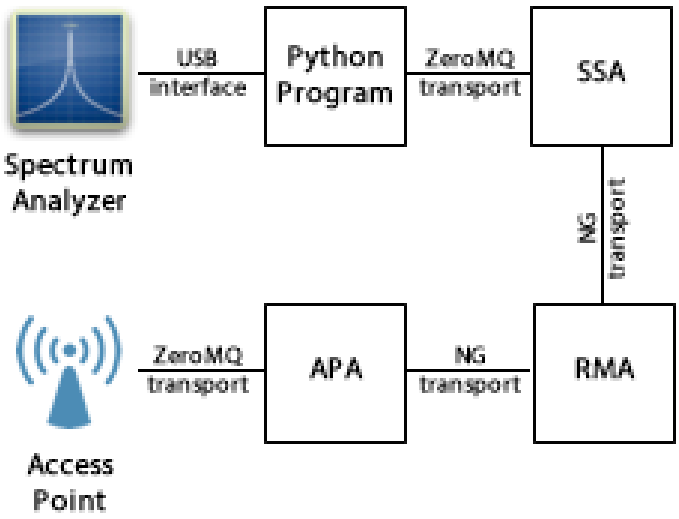

Figura 11 – *Cenário implementado pelo ICT Lab/WOCA.*

A Figura 11 exemplifica os desafios da gerência em redes convergentes NovaGenesis. O SSA, por exemplo, utiliza uma interface ZeroMQ sobre TCP/IP

para troca de mensagens com um programa em Python conectado ao analisador de espectro na porta *Universal Serial Bus* (USB) de um *laptop*. Já o APA utiliza ZeroMQ sobre TCP/IP para alterar a configuração do *access point*. Observe que a comunicação entre os serviços APA, SSA e RMA é feita usando-se mensagens NovaGenesis sobre Ethernet diretamente, sem TCP/IP.

A aplicação do modelo de referência para gerenciamento/controle NovaGenesis no cenário da Figura 11 deve englobar a análise dos dados coletados pelos serviços representantes do *hardware*, visando otimizar os recursos da rede e utilizar o melhor canal possível para os clientes conectados. Neste panorama, a NovaGenesis deve gerenciar dispositivos físicos (analisador de espectro, *access point*, controlador de antena) e virtuais (controlador SDN) conforme ilustra a Figura 12. O *Controller* deve realizar a alocação dinâmica dos recursos com base nas informações do estado de todos os processos envolvidos, utilizando a propriedade *situation-aware*. Essa propriedade consiste em contextualizar dados com o objetivo de tomar ciência de todos os processos que o envolvem, de modo que o elemento tome decisões da forma mais inteligente possível. Na aplicação do modelo de gerência e controle proposto nessa dissertação, cada equipamento (físico ou virtual) possui um agente que expõe as capacidades, protocolos, informações de gerência e controle disponíveis. Na Figura 12 esses agentes são agregados em um único PGCMA. De fato, essa solução tem a vantagem de concentrar em um único serviço a representação/controle/gerência de vários equipamentos da rede, escalando esse modelo de 1:N. Dependendo das tecnologias e particularidades dos equipamentos sendo operados, os agentes mostrados na Figura 12 poderiam ser implementados em diferentes PGCMAs.

A solução proposta na Figura 12 permite que o serviço mediador MeS agregue informações relativas aos vários contratos que ele mantém com os PGCMAs da rede, trocando essas informações agregadas com o *Controller* da rede. Por exemplo, ele poderia indicar que opera X dispositivos de sensoriamento espectral nas bandas 2.4 GHz e/ou 5.8 GHz. Poderia ainda indicar que opera Y *access points* de Wi-Fi e suas respectivas bandas. Essas informações alimentam o ciclo de decisão no *Controller* da rede e podem alimentar outros serviços do ecossistema, sempre usando

contratos, nomeação autocertificável, modelo publica/assina com sigilo, autenticação, integridade e autorização.

O serviço de gerenciamento deve expor para os demais serviços as suas características, representando o mundo físico e o virtual. Após a descoberta da aplicação que transfere dados via Wi-Fi, um contrato é publicado para garantir a integridade da relação de confiança entre elas, o contrato é negociado conforme interesse de cada parceiro que tem a opção de emitir uma contraproposta até que se chegue a um acordo entre os componentes deste cenário (*Controller* e MeS; *Manager* e MeS; PGCMA e MeS). Somente após assinatura do contrato tem-se a troca de informações para a aplicação do serviço. Esta medida otimiza o desempenho da rede que é utilizada somente quando o contrato for assinado.

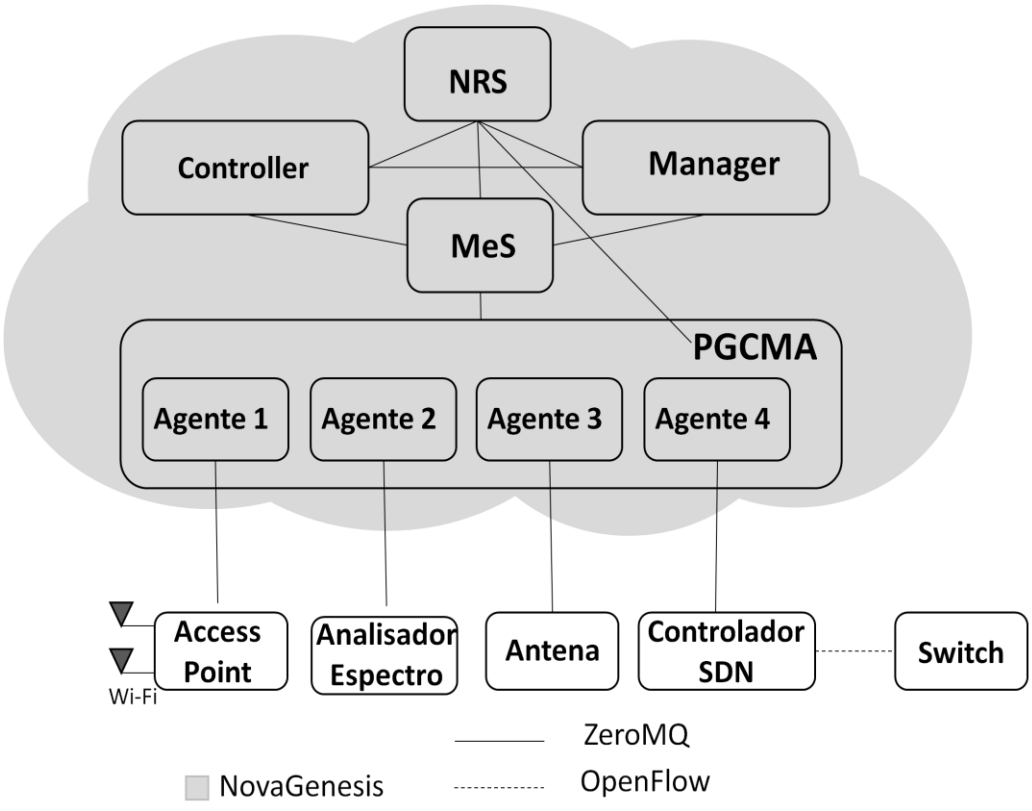

Figura 12 – *Diagrama Esquemático do Estudo de Caso.*

Todos os dispositivos do mundo físico podem possuir PGCMAs que os representem, fornecendo informações e estados, e acessando-os diretamente. Abaixo é analisado como seria o fluxo de ações do serviço em questão usando a arquitetura de gerência e controle NovaGenesis proposta nessa dissertação.

5.4.1 Estudo de Caso 1: Sistema de Gerenciamento e Controle NovaGenesis Interoperando com Equipamentos Legados

1. O PGCMA, por meio de um agente do analisador de espectro, coleta informações de faixa de frequência e taxa de amostragem disponíveis no *hardware*;
2. O PGCMA, por meio de um agente do *Access Point*, coleta informações de banda e canais de Wi-Fi disponíveis nos *Access Points* que ele representa;
3. O PGCMA expõe (publica e notifica) para o MeS as capacidades do analisador de espectro e *Access Points* usando as primitivas *publish* e *notify*. As capacidades fazem parte de uma oferta de contrato que o PGCMA envia para o MeS.
4. O *Mediator Service* assina a oferta de contrato publicada pelo PGCMA e notificada pelo NRS.
5. O MeS analisa a oferta e aceita se adequada ou seja, se os níveis de SLA estão de acordo com o estabelecido para o serviço.
6. O MeS publica um objeto de aceitação da oferta, notificando o PGCMA;
7. O PGCMA recebe o objeto de aceitação de oferta e inicia a operação conjunta com o MeS;
8. Supondo que o MeS já tenha um contrato com o *Controller*, ele filtra as informações relevantes para as ações do *Controller*, publicando-as e notificando-as;
9. O *Controller* assina os dados publicados pelo MeS, analisando-os;
10. O PGCMA publica amostras de energia na faixa de frequência sendo monitorada para o MeS. O MeS pode fazer uma média temporal e publicar para o *Controller* o valor médio em uma certa janela de tempo. Ou existe

ainda a possibilidade de que o PGCMA publique dados não só para o MeS, mas também para o *Controller*. Essas opções ainda terão que ser avaliadas em testes futuros;

11. O *Controller*, de forma autônoma, analisa os dados publicados pelo MeS e avalia o canal de operação Wi-Fi mais favorável, decidindo se deve efetuar a troca de canal ou não. Se o *Controller* decidir por efetuar a troca do canal, ele publica e notifica o *Mediator Service* sobre a configuração que se faz necessária. Aqui também o *Controller* poderia acionar diretamente o PGCMA relacionado ao *hardware* cuja troca de canal se faz necessária;
12. O *Mediator Service* por sua vez publica e notifica o *Access Point*, por meio do PGCMA;
13. O PGCMA assina para receber a configuração de troca de canal;
14. O PGCMA configura o *Access Point* com o novo canal. No cenário da **Figura *12***, o *Access Point* teve o *middleware* substituído pelo *OpenWrt* (77), que permite que ele se torne totalmente configurável pelo PGCMA.

5.4.2 Estudo de Caso 2: Sistema de Gerenciamento e Controle NovaGenesis usando Abordagem SDN

1. O PGCMA, por meio de um agente do controlador SDN, determina as funcionalidades de SDN disponíveis no controlador, incluindo versão de protocolos, regras, aplicações disponíveis, tais como descoberta de rede, por exemplo;
2. O PGCMA expõe (publica e notifica) para o MeS as capacidades do controlador SDN usando as primitivas *publish* e *notify*. As capacidades fazem parte de uma oferta de contrato que o PGCMA envia para o MeS;
3. O *Mediator Service* assina a oferta de contrato publicada pelo PGCMA e notificada pelo NRS;
4. O MeS analisa a oferta e aceita se adequada;
5. O MeS publica um objeto de aceitação da oferta, notificando o PGCMA;

6. O PGCMA assina o objeto de aceitação e inicia a operação conjunta com o MeS e o *Controller*;
7. Supondo que o MeS já tenha um contrato com o *Controller*, ele filtra as informações relevantes para as ações do *Controller*, publicando-as e notificando-as. Agora o *Controller* sabe que existe um controlador SDN operacional na rede e pode explorar suas aplicações de rede via MeS;
8. Inicia-se a troca de mensagens operacionais entre *Mediator Service* e o *Controller*;
9. O PGCMA monitora o *log* de eventos do controlador SDN que ele representa. Ele determina então, de forma autônoma, que não existe uma regra de encaminhamento correta para uma *Virtual Local Area Network* (VLAN) configurada no controlador SDN. Ou seja, o comutador está descartando quadros, pois o controlador SDN está aplicando erroneamente uma regra;
10. O PGCMA publica uma consulta para o MeS, que correlaciona essa situação com as informações provenientes de outros PGCMAs. O MeS então descobre que realmente existe uma falta de regra implementada em vários controladores;
11. O MeS publica uma consulta para o controlador de rede (*Controller*), que de forma autônoma verifica que a regra instalada foge de um contrato estabelecido com uma aplicação cliente da rede (não mostrada na **Figura *12***). Então, ele cria uma regra coerente ao que a aplicação deseja e a publica para o MeS, notificando-o;
12. O MeS assina a nova regra e publica para o PGCMA que representa o controlador SDN;
13. O PGCMA envia as informações de configuração para o controlador SDN, que por sua vez atua sobre o comutador usando um protocolo legado de SDN. Observe que o PGCMA pode realizar o mesmo serviço que um controlador OpenFlow, mas não estando limitado as abstrações do OpenFlow;

5.4.3 Estudo de Caso 3: Sistema de Gerenciamento e Controle NovaGenesis Gerenciando Falhas em Elementos Legados

1. O PGCMA, por meio de um agente do analisador de espectro, coleta informações de alarmes do *Access Point* que ele representa;
2. O PGCMA expõe (publica e notifica) para o MeS os alarmes ativos do *Access Point* usando as primitivas *publish* e *notify* e inclui o nível de severidade na mensagem. Os alarmes do *Access Point* fazem parte de uma oferta de contrato que o PGCMA envia para o MeS.
3. O *Mediator Service* assina a oferta de contrato publicada pelo PGCMA e notificada pelo NRS.
4. O MeS analisa a oferta e aceita se adequada.
5. O MeS publica um objeto de aceitação da oferta, notificando o PGCMA;
6. O PGCMA recebe o objeto de aceitação de oferta e inicia a operação conjunta com o MeS;
7. Supondo que o MeS já tenha um contrato com o *Manager*, ele correlaciona as falhas, filtra os alarmes de maior severidade e publica e notifica ao *Manager*;
8. O *Manager* assina por ordem de prioridade os alarmes publicados pelo MeS;
9. O *Manager*, de forma autônoma, analisa os alarmes publicados pelo MeS, decidindo se deve atuar no *Access Point*. Se o *Manager* decidir por agir, ele publica e notifica o *Mediator Service* sobre qual configuração que se faz necessária.
10. O *Mediator Service* por sua vez publica e notifica o *Access Point*, por meio do PGCMA que o controla o hardware cuja atuação se faz necessária.
11. O PGCMA assina para receber a nova configuração;
12. O PGCMA configura o *Access Point* com o objetivo de sanar os alarmes ativos. Neste cenário o *Access Point* teve o *middleware* substituído pelo OpenWrt, que permite que ele se torne totalmente configurável.

5.4.4 Estudo de Caso 4: Sistema de Gerenciamento e Controle NovaGenesis Garantindo a Integridade dos Dados na Rede

1. Imagine o cenário da Figura 12 onde se deseja expandir a área de cobertura *wireless* usando vários *Access Point* Wi-Fi *dualband* e a premissa de que todos devem trabalhar na mesma rede.
2. O Sistema de Gerenciamento e Controle da NovaGenesis, por meio dos passos 1 a 10 descritos no Casos de Uso 1, decide por trocar o canal de operação Wi-Fi dos *Access Point* (de 2,4GHz para banda 5,8GHz);
3. O *Controller* publica e notifica o *Mediator Service* sobre as configurações dos *Access Point* que se fazem necessárias.
4. O *Mediator Service* por sua vez publica e notifica os *Access Point* por meio do PGCMA;
5. O PGCMA assina para receber a configuração de troca de canal dos *Access Point*;
6. O PGCMA inicia a configuração dos *Access Point* no novo modo.
7. O PGCMA não consegue finalizar a configuração em um dos *Access Point* devido à falta de comunicação (exemplo: falta de energia). Obs.: Neste cenário todas as tentativas de configuração foram realizadas por um período de tempo pré-definido.
8. O PGCMA avisa o MeS que não foi possível realizar a configuração em um determinado *Access Point.*
9. MeS publica e notifica o *Manager* sobre a falha de configuração em um determinado *Access Point*;
10. *Manager* identifica inconsistência de dados na rede e solicita ao *Controller* que reconfigure todos os *Access Point* que tiveram seu modo alterado, ou seja, que volte ao estado inicial para que mantenha a integridade dos dados na rede.

6 Comparação das Abordagens Existentes com a NovaGenesis.

Várias iniciativas têm sido propostas para gerência e controle de redes de próxima geração. Foram analisadas nesta dissertação as arquiteturas de gerenciamento e controle de diversas abordagens emergentes descritas no Capítulo 4 tais como FIWARE, RINA e CCNx.

Nesta seção, é apresentada uma análise comparativa das arquiteturas atuais, bem como das arquiteturas emergentes descritas ao longo deste trabalho. O objetivo é contrastar as características de cada proposta com o modelo de referência para controle e gerência da NovaGenesis. A Tabela 1, no Anexo A, resume os resultados desta comparação considerando os diversos aspectos elencados nas arquiteturas de controle e gerenciamento estudadas nesta dissertação.

O SNMP é o protocolo mais utilizado na Internet por se tratar de um protocolo de comunicação simples. Sendo assim, ele não utiliza muitos comandos e define duas operações básicas: *set e get*, bem como suas derivações. Utiliza-se do método *polling* para monitoração nos elementos gerenciados e executado em intervalos regulares ou sob demanda do operador. As *traps* são usadas para comunicar de forma espontânea um evento de gerência que por sua natureza não pode ser monitorado efetivamente através do *polling*. Para suportar o comando GET-NEXT, que é usado para acessar o atributo do próximo objeto na hierarquia da MIB, usa-se o conceito de tabelas que permite que se faça a varredura por todos os objetos da MIB. Por sua simplicidade, o SNMP não é orientado à conexão e por isto não confiável, enquanto que o CMIP segue o modelo orientado à conexão.

O CMIP é um protocolo suportado pela pilha OSI e com maior número de operações básicas. Ele geralmente é usado para gerenciamento de redes mais complexas como as redes telefônicas. A MIB do protocolo CMIP utiliza o conceito de classes de objetos que herdam propriedades similares de outros objetos que estão em uma classe de ordem superior. O CMIP é orientado a eventos, e por isto o uso da rede é otimizado devido ao menor número de *pollings* se comparado ao SNMP.

A arquitetura de gerência do projeto RINA é similar ao modelo de gerência OSI e também suporta o modelo gerente-agente. Todo processo IPC possui uma

camada de gerenciamento com funções de alocação de fluxo, de recursos e geração da tabela de encaminhamento. Possui um módulo para autenticação e outro para atualizar e/ou manter o estado dos objetos gerenciados por meio do protocolo CDAP, da RIB e da RIB *Daemon* que trabalham baseados no modelo *pub/sub*. O endereçamento RINA é utilizado nos processos IPC por meio do *Connection End-Point Identifier* (CEP-ID). Esse esquema de endereçamento é visto somente dentro de cada DIF.

Em CCN pode-se utilizar a proposta de gerenciamento de (66) utilizando-se de arquitetura interoperável com sistemas legados por meio do protocolo SNMP e uma MIB CCN definida a partir da MIB II. O *Content Store* armazena os conteúdos e é gerenciado por cada nó CCN. A FIB é uma base de informações usada para encaminhar os pacotes de interesse para a fonte de dados correspondente de forma análoga à tabela de roteamento IP.

FIWARE usa os protocolos NGSI9 e NGSI10. NGSI9 é o protocolo para gerenciamento da disponibilidade do contexto (gerencia o provedor da informação) enquanto que o NGSI10 é usado para gerenciamento de contexto (gerencia as entidades). As operações do protocolo são: GET para receber a informação, POST para criar nova informação, PUT para atualizar e DELETE para cancelar/apagar a informação. Seguindo a linha de plataforma aberta, FIWARE usa a base de dados MongoDB para armazenamento dos dados de contexto no Orion.

O CONGEP é o protocolo único da NovaGenesis que define a troca de mensagens entre os elementos da rede. Somente duas operações são usadas para executar todas as tarefas de gerência, reduzindo a complexidade das operações na arquitetura NovaGenesis. Todas as mensagens são baseadas no paradigma pub/sub, inclusive as notificações de alarmes que são mensagens do tipo *publish notify* que se comportam de forma similar às *traps* SNMP.

A arquitetura de Gerência e Controle proposta para a NovaGenesis cobre diversas limitações relacionadas às pesquisas das tecnologias de gerência e controle atuais e abordagens emergentes como IoT, SDN, CCN e NFV. A Tabela 2, no Anexo A, apresenta um resumo de como a arquitetura proposta atinge os desafios de gerência e controle apresentados para as iniciativas atuais e para IoT, SDN, CCN e NFV.

No sistema de gerência da Internet, o SNMP geralmente é usado sobre UDP onde os dados são transmitidos somente uma vez. Na NovaGenesis é possível implementar serviços que sejam sensíveis à garantia de entrega e necessitem de retransmissão. Ainda na Internet atual existe a falta de padronização para o gerenciamento de redes heterogêneas. Na NovaGenesis os elementos mediadores MeS e PGCMA são capazes de suportar quaisquer novos elementos de rede e intermedia a comunicação entre o *Manager* e *Controller* apoiando para a formação do ciclo de vida de serviços e para a diminuição da interação humano-máquina.

Quanto ao suporte a IoT, o modelo proposto na NovaGenesis permite que várias aplicações sejam construídas e inseridas na plataforma utilizando-se de todos os recursos oferecidos por ela, inclusive o suporte à um grande número de conjunto de elementos.

A NovaGenesis atende ao paradigma CCN uma vez que utiliza do esquema de nomeação para o encaminhamento dos pacotes, porém ela não possui as limitações de gerência e controle das iniciativas CCN atuais, como o gerenciamento de conteúdos e *cache*. A NovaGenesis oferece para qualquer serviço, incluindo os de gerenciamento e controle, acesso ao sistema de resolução de nomes (NRS) para prover os dados em *cache* disponibilizados na tabela *hash* distribuída.

Com relação às limitações do SDN relacionadas ao controle centralizado, na NovaGenesis estes desafios foram todos endereçados. A informação de gerência e controle está concentrada em um único ponto, que facilita a localização de erros, porém a inteligência na tomada de decisões está distribuída logicamente no *Manager* e no *Controller*. O MeS atua como cliente do *Manager* e *Controller*, realizando algumas tarefas de controle e gerenciamento que contribui para balancear o tráfego no *Northbound.*

A NovaGenesis permite que funções de rede sejam virtualizadas, porém é um desafio implementa-las sem que aumente a complexidade dos sistemas de gerenciamento e controle. Para isto, o *Manager* e o *Controller* devem manter ciência do estado de todos os processos da rede (propriedade *self-awereness*), capacidade de detectar mudanças de estado (*self-monitoring*), e de ser adaptativo para gerência e controle de novos elementos de rede (*self-adjustment*) além de ter ciência do ambiente (*self-situation*).

Em suma, o modelo proposto reúne de forma sinérgica diversos aspectos inovadores para sistemas de gerência e controle em FI, de forma bastante flexível para permitir a sua evolução e integração com novas iniciativas. Tal característica é bastante interessante no cenário tecnológico atual, onde as tecnologias para FI existentes atendem às limitações descritas na Tabela 2, porém não de forma global, onde uma tecnologia abrange um determinado requisito e não trata do outro. Surge então a necessidade de uma nova proposta que supra as limitações atuais de forma única, integrada, coesa e harmônica. A plataforma de gerenciamento e controle apresentada neste trabalho éa solução para tais necessidades, pois ela combina as vantagens de diversas tecnologias tais como IoT, SDN, CCN e NFV e ainda permite que elas sejam integradas em uma única solução NovaGenesis.

7 Conclusões e Trabalhos Futuros

Neste trabalho, foi criado um cenário para gerenciamento e controle em função das necessidades da NovaGenesis. Inicialmente, foram revisitadas as discussões sobre a arquitetura atual de gerência, sobre conceitos como FCAPS e protocolo SNMP. Foram discutidas algumas das mais importantes iniciativas FIA sob a perspectiva de gerenciamento e de controle de rede.

Foram elencados diversos requisitos para gerenciamento e controle da próxima geração de redes em que se discutiu uma solução para redes sem IP, essencialmente a rede NovaGenesis. Em seguida, foi apresentada para a arquitetura NovaGenesis uma proposta conceitual para gerenciamento e controle da rede. Foi especificado um protocolo específico de controle e gerência, que inclui a comunicação entre os elementos do ecossistema NovaGenesis. Este trabalho foi projetado genérico o suficiente para atender a qualquer tecnologia FIA de forma completa e integrada, pois abrangem todas as cinco áreas funcionais de gerenciamento FCAPS. Por fim, foi realizado um comparativo das iniciativas emergentes de FI com a NovaGenesis sob a ótica do gerenciamento e controle de rede.

Como proposta para trabalhos futuros está a implementação na NovaGenesis da arquitetura de controle e gerenciamento proposta nesta dissertação. Outra possibilidade de trabalho futuro é aprofundar a análise do modelo sugerido, explorando o potencial da arquitetura proposta para suportar os desafios identificados neste trabalho e propor novos algoritmos que incorporam propriedades de autogerência, como autoconfiguração, auto-otimização, autorreparo e autoproteção. Neste contexto, a relação com técnicas de Inteligência Artificial deverá ser explorada.

8 Anexo A

Tabela 1 – *Comparação das Arquiteturas de Gerenciamento com a NovaGenesis*

COMPARATIVO	Internet	OSI	RINA	CCN	FIWARE	NovaGenesis
PROTOCOLO	SNMP	CMIP	CDAP	SNMP	NGSI	CONGEP
OPERAÇÕES	GET-Request SET Request GET-NEXT-Request GET-Response Trap	EventReport EventReport-Confirmed Get Linked-Reply Set Set-Confirmed Action Action-Confirmed Create Detete Cancel-Get-Confirmed	CREATE DELETE READ WRITE START STOP	GET SET GET-NEXT GET-BULK INFORM	GET POST PUT DELETE	GET SET
ARMAZENAMENTO	MIB Tabela Base de Dados	MIB Classe Base de Dados	RIB	MIB CCN Content Store	Base de Dados MongoDB	HTS
EVENTOS	Traps	Notificações	Subscrições	Traps	Notificações	Notify
MODELO COMUNICAÇÃO	Polling	Eventos	PUB/SUB	PUB/SUB	PUB/SUB	PUB/SUB
ENDEREÇAMENTO	UDP/IP	TCP/IP	CEP-ID	FIB	RESTful API TCP/IP	Hash

Tabela 2 – *Comparação das Arquiteturas de Controle e Gerenciamento com a NovaGenesis*

Abordagem	Problemas/Limitações	Modelo de Controle/Gerência NovaGenesis
Internet	Perda de pacotes de gerenciamento devido ao uso do UDP. Implementações já existentes que não permitem o uso do TCP.	No protótipo corrente não existe retransmissão. Devido à orquestração via representantes e contratos, informações de controle/gerência críticas podem automaticamente ser transportadas por serviços (ou implementações em *hardware*) que suportem retransmissão.
	Heterogeneidade de protocolos de gerenciamento. A solução padronizada de gerência atual não atende a todos os casos que estão emergindo.	Permite a customização de PGCMA/MeS para controle/gerência das mais diversas tecnologias, com ou sem TCP/IP. Implementa de forma homogênea (usando a mesma API do NRS) representantes de recursos heterogêneos, transladando protocolos quando necessário, controlando elementos de *software* conforme instruções providenciadas pelo *Controller* ou *Manager*.
	Falta de interoperabilidade de controle e gerenciamento	O modelo proposto é genérico, podendo ser usado com as mais diversas soluções de gerência e controle existentes.
	Excessiva interferência humana	O ciclo de vida dos serviços de gerência e controle favorece a operação autonômica. Soma-se a isso a estrutura hierárquica de serviços que favorece a divisão de responsabilidades.
IoT	Suporte a novas tecnologias	PGCMA atua como representante dos mais diversos elementos gerenciados. Novas tecnologias podem ser integradas as legadas.
	Escalabilidade e elasticidade	Pode haver diversos PGCMA na rede, dependendo da heterogeneidade da rede e do número de dispositivos em uso. Os serviços NovaGenesis podem ser instanciados conforme a necessidade. Pode-se inclusive reduzir o número de serviços quando uma parte significativa dos nós está desligada. Todos os serviços podem ser espelhados, dividindo a carga de trabalho. O NRS naturalmente mantém a coerência e versionamento dos dados

		publicados usando função *hash*.
	Proveniência e integridade dos dados	A NovaGenesis opera com nomeação autocertificável para todas entidades, incluindo objetos de informação, serviços, sistemas operacionais, interfaces de rede, terminais, equipamentos de rede, comutadores, etc. Através dos mapeamentos entre nomes, todas as relações entre entidades podem ser obtidas por quem está autorizado a ver. Todos os dados, controles, contratos, comandos de gerência tem a sua integridade verificada quando assinadas.
	Diversidade de interfaces de programação	O NRS fornece um conjunto único de primitivas para todas as ações, sejam de controle ou de gerência. Dentro da NovaGenesis uma única interface de programação é usada tanto pelas implementações de protocolos, como serviços de gerência e controle.
CCN	Gerenciamento de conteúdos e *caches*	A NovaGenesis possui um serviço de tabela *hash* distribuída, que pode executar em diversas instâncias. Tanto o *Manager*, quanto o *Controller* possuem livre acesso ao sistema de resolução de nomes. Os sistemas de gerência e controle podem ser usados para gerenciar/controlar os serviços chave da própria NovaGenesis.
	Gerência e controle não tiram proveito do *cache* de rede.	Na NovaGenesis, os componentes do NRS são usados por qualquer serviço, incluindo os serviços de controle e gerência. Assim, os objetos de informação de controle e gerência tiram vantagem dos paradigmas CCN.
SDN	Interfaces de gerenciamento independentes para plano de dados e controle	A NovaGenesis possui uma interface única para os serviços de gerenciamento e controle.
	Ponto único de falha no controlador centralizado	A centralização do controle é uma característica inovadora do SDN, mas um ponto controverso se ela for implementada em um único controlador. Na NovaGenesis o *Manager* e *Controller* são logicamente centralizados, mas implementados de forma distribuída.
	Insuficiente escalabilidade no plano de Dados para IoT	Controle e Gerenciamento em IoT requerem soluções em escala. O modelo de Controle e Gerenciamento da NovaGenesis fornece uma

		abordagem distribuída e auto-organizada para IoT. Na NovaGenesis, o PGCMA e o MeS realizam a mediação de informações de controle e gerência e corroboram para balancear o tráfego na rede.
NFV	Complexidade nos procedimentos de operação e manutenção das redes virtuais	O modelo proposto para controle e gerenciamento na NovaGenesis utiliza a mesma abordagem para as entidades físicas e virtuais. É necessário que o sistema de gerência e controle seja simples, como no modelo proposto para a NovaGenesis. Ainda assim, certo grau de autogerência é requerido. O *Manager* e *Controller* são capazes de acomodar propriedades *self*-*.
	Abordagem integrada de gerenciamento das arquiteturas complementares NFV e SDN.	A arquitetura de Gerência e Controle NovaGenesis permite monitorar as mais diversas tecnologias emergentes de forma completa e integrada, pois abrange todas as cinco áreas funcionais FCAPS.

Referências Bibliográficas

1. **Rao, Umesh Hodeghatta.** Challenges of Implementing Network Management Solution. *International Journal of Distributed and Parallel Systems.* 2011, Vol. 2, 5, p. 67.

2. **Kalyanasundaram, Pramod and Sethi, Adarshpal S.** Interoperability issues in heterogeneous network management. *Journal of Network and Systems Management.* 1994, Vol. 2, 2, pp. 169-193.

3. **Shenker, Scott.** Fundamental design issues for the future Internet. *Selected Areas in Communications, IEEE Journal on.* 1995, Vol. 13, 7, pp. 1176-1188.

4. **Leighton, T.** Improving Performance on the internet. *Commun. ACM.* Fevereiro 2009, Vol. 52, 2, pp. 44-51. [Online]. Available: http://doi.acm.org/10.1145/1461928.1461944.

5. **Alberti, Antônio Marcos.** A conceitual-driven survey on future internet requirements, technologies, and challenges. [ed.] Springer. *Journal of the Brazilian Computer Society.* 2013, Vol. 19, 3, pp. 291-311.

6. **Alberti, Antonio Marcos, et al.** Internet of information and services (iois): a conceptual integrative architecture for the future internet. *Proceedings of the 7th International Conference on Future Internet Technologies.* 2012, pp. 45-45.

7. **Iqbal, Hammad and Znati, Taieb.** Overcoming failures: fault-tolerance and logical centralization in clean-slate network management. *INFOCOM.* Março 2010, pp. 1-5.

8. **Paul, Subharthi, Pan, Jianli and Jain, Raj.** Architectures for the future networks and the next generation Internet: A survey. *Computer Communications.* 2011, Vol. 34, 1, pp. 2-42.

9. **G. group.** Geni design principles. *Computer.* setembro 2006, Vol. 39, 9, pp. 102-105.

10. **Stuckmann, Peter and Zimmermann, Rainer.** European research on future Internet design. *Wireless Communications.* 5, 2009, Vol. 16, pp. 14-22.

11. **Qin, Zhijing and Denker, Grit, et al.** A software defined networking architecture for the internet-of-things. *Network Operations and Management Symposium (NOMS), IEEE.* 2014, pp. 1-9.

12. **Jacobson, V.** CCN routing and forwarding. *Stanford NetSeminar*. 2011. Tech. Rep..

13. **Lin, Geng, et al.** Cloud computing: IT as a service. *IT Professional Magazine*. 2, 2009, Vol. 11, p. 10.

14. **Wickboldt, Juliano, et al.** Software-defined networking: management requirements and challenges. *Communications Magazine, IEEE*. 1, 2015, Vol. 53, pp. 278-285.

15. **Esteves, Rafael P, Granville, Lisandro Z and Boutaba, Raouf.** On the management of virtual networks. *Communications Magazine, IEEE*. 7, 2013, Vol. 51, pp. 80-88.

16. **Sun, Songlin, et al.** Integrating network function virtualization with SDR and SDN for 4G/5G networks. *Network, IEEE*. 3, 2015, Vol. 29, pp. 54-59.

17. **Greenberg, Albert, et al.** A clean slate 4D approach to network control and management. *ACM SIGCOMM Computer Communication Review*. 2005, Vol. 34, 5, pp. 41-54.

18. **Klerer, S Mark.** The OSI management architecture: an overview. *Network, IEEE*. 1988, Vol. 2, 2, pp. 20-29.

19. **Postel, Jon.** Transmission control protocol. 1981.

20. **Lee, Chae-Sub and Knight, Dick.** Realization of the next-generation network. *Communications Magazine, IEEE*. 2005, Vol. 43, 10, pp. 34-41.

21. **Poikselkä, Miikka and Mayer, Georg.** The IMS: IP multimedia concepts and services. 2013.

22. **Xiao, Yang, Du, Xiaojiang and Zhang, Jingyuan.** Internet protocol television (IPTV): the killer application for the next-generation internet. *Institute of Electrical and Electronics Engineers*. maio 2007.

23. **Hedrick, Charles L.** Routing information protocol. 1988.

24. **Moy, John.** Open shortest path first (ospf) version 2. *IETF: The Internet Engineering Taskforce RFC*. 1998, Vol. 2328.

25. **Postel, Jon.** Internet control message protocol. 1981.

26. **Blake, Steven, et al.** An architecture for differentiated services. rfc 2475, Dezembro 1998.

27. **Low, Steven H, Paganini, Fernando and Doyle, John C.** Internet congestion control. *Control Systems, IEEE.* 2002, Vol. 22, 1, pp. 28-43.

28. **Harai, H.** Designing New-Generation Network: Overview of AKARI Architecture Design. *Asia Communications and Photonics Conference and Exhibition.* 2009, p. FL2.

29. **Jacobson, Van, et al.** RTP: A transport protocol for real-time applications. 2003.

30. **Huitema, Christian.** Real time control protocol (RTCP) attribute in session description protocol (SDP). 2003.

31. **Arango, Mauricio, et al.** Media gateway control protocol (MGCP) version 1.0. 1999.

32. **Taylor, Tom.** Megaco/H. 248: a new standard for media gateway control. *Communications Magazine, IEEE.* 2000, Vol. 38, 10, pp. 124-132.

33. **Rosenberg, Jonathan, et al.** *SIP: session initiation protocol.* RFC 3261, Internet Engineering Task Force. 2002.

34. **Warrier, Unnikrishnan S, et al.** *Common management information services and protocols for the internet (CMOT and CMIP).* 1990. RFC 1189.

35. **Case, Jeffery, et al.** A simple network management protocol (SNMP). 1989.

36. **Kephart, Jeffrey O and Chess, David M.** The vision of autonomic computing. *IEEE Computer Magazine.* 2003, Vol. 36, 1, pp. 41-50.

37. **Pereira, Getúlio Emílio Oliveira.** Especificação de Uma Plataforma Referencial Convergente para a Nova Geração de Redes sem Fio. *Instituto Nacional de Telecomunicações.* 2013.

38. **Berns, Andrew and Ghosh, Sukumar.** Dissecting self-* properties. *Self-Adaptive and Self-Organizing Systems, 2009. SASO'09. Third IEEE International Conference on.* 2009, pp. 10-19.

39. **Rec, I.T. U. T.** M.3400. *TMN management functions.* 2000.

40. **Ghodsi, Ali, et al.** Naming in content-oriented architectures. *Proceedings of the ACM SIGCOMM workshop on Information-centric networking.* 2011, pp. 1-6.

41. **Ramírez, Wilson, et al.** A survey and taxonomy of ID/Locator Split Architectures. *Computer networks.* 2014, Vol. 60, pp. 13-33.

42. **MIT.** Auto-ID Center. 1999. http://autoidlabs.org/wordpress_website/.

43. **Papazoglou, Michael P, et al.** Service-Oriented Computing: State of the Art and Research Challenges. *Computer.* 11, 2007, pp. 38-45.

44. **Dobre, Ciprian and Xhafa, Fatos.** Parallel programming paradigms and frameworks in big data era. *International Journal of Parallel Programming.* 5, Vol. 42, pp. 710-738.

45. **McKeown, Nick, et al.** OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review.* 2008, Vol. 38, 2, pp. 69-74.

46. NOX. An open-source OpenFlow controller. 2011. [Online]. Available: http://www.noxrepo.org/.

47. **Gude, Natasha, et al.** NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review.* 2008, Vol. 38, 3, pp. 105-110.

48. **Ahlgren, Bengt, et al.** A survey of information-centric networking. *Communications Magazine, IEEE.* 2012, Vol. 50, 7, pp. 26-36.

49. **Jacobson, Van, et al.** Networking named content. *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. 2009, pp. 1-2.

50. **Alberti, Antonio M, et al.** A NovaGenesis proxy/gateway/controller for OpenFlow software defined networks. *Network and Service Management (CNSM), 2014 10th International Conference on.* Novembro 2014, pp. 394-399.

51. **Sundmaeker, Harald, et al.** Vision and challenges for realising the Internet of Things. 2010, Vol. 20, 10.

52. **Perera, Charith, et al.** Context aware computing for the internet of things: a survey. *IEEE Communications Surveys \& Tutorials.* 1, 2014, Vol. 16, pp. 414-454.

53. **Namiot, Dmitry and Sneps-Sneppe, Manfred.** On IoT Programming. *International Journal of Open Information Technologies.* 2014, Vol. 2, 10.

54. FIWARE. www.fiware.org.

55. SmartSantander. http://www.smartsantander.eu/.

56. Machine to Machine (M2M) Communications Technical Report. *IEEE 802.16 Broadband Wireless Access Working Group.* 2010.

57. **Cholez, Thibault.** Introduction to Content-Centric Networking and the CCNx framework. *6th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2012).*

58. **Kim, Hyojoon and Feamster, N.** Improving network management with software defined networking. *Communications Magazine, IEEE*. 2, 2013, Vol. 51, pp. 114-119.

59. **Rothenberg, Christian Esteve, et al.** OpenFlow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. *Cad. CPqD Tecnologia, Campinas.* 2010, Vol. 7, 1, pp. 65-76.

60. **Smith, M, et al.** OpFlex control protocol. *IETF*. Abril 2014.

61. **Santos, Mateus AS, et al.** Decentralizing SDN's control plane. *39th Annual IEEE Conference on Local Computer Networks.* 2014, pp. 402-405.

62. **Zhang, Lixia, et al.** Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC.* 2010.

63. **Dannewitz, Christian, et al.** Network of Information (NetInf)--An information-centric networking architecture. *Computer Communications.* 2013, Vol. 36, 7, pp. 721-735.

64. **Kang, W., et al.** A Network Monitoring Tool for CCN. *World Telecommunications Congress (WTC), 2012.* 2012, pp. 1-3.

65. **Ahlgren, Bengt, et al.** Content, connectivity, and cloud: ingredients for the network of the future. *Communications Magazine*. 2011, pp. 62-70.

66. **de Lima Oliveira, Marciel and Rothenberg, Christian Esteve.** SNMP Proxy CCN: Uma proposta de arquitetura para gerência de redes orientadas a conteúdo interoperável com sistemas legados. *SBRC - IX Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo - Wp2p+*. 2014.

67. **Pentikousis, Kostas, et al.** Self-management for a network of information. *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on.* 2009, pp. 1-5.

68. **Mijumbi, Rashid, et al.** Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*. 1, 2016, Vol. 54, pp. 98-105.

69. **Marquezan, Clarissa C, et al.** Distributed autonomic resource management for network virtualization. *Network Operations and Management Symposium (NOMS), IEEE*. 2010, pp. 463-470.

70. **Telefonica.** OpenMano. http://www.tid.es/long-term-innovation/network-innovation/telefonica-nfv-reference-lab/openmano.

71. **Vrijders, Sander, et al.** Prototyping the recursive internet architecture: the IRATI project approach. *Network, IEEE.* 2014, Vol. 28, 2, pp. 20-25.

72. **Wang, Yuefeng, et al.** RINA: an architecture for policy-based dynamic service management. 2013.

73. **Banerjee, Prith, et al.** Everything as a service: Powering the new information economy. *IEEE Computer Society.* 3, 2011, Vol. 44, pp. 36-43.

74. **ITU, T.** Recommendation X.733 : Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function. 1992.

75. **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.* 2014.

76. **Alberti, A.M., et al.** Service-oriented, name-based, and software-defined spectrum sensing and dynamic resource allocation for wi-fi networks using novagenesis. *VI International Workshop on Telecomunication.* 2015.

77. **Fainelli, Florian.** The OpenWrt embedded development framework. *Proceedings of the free and open source software developers european meeting.* 2008.