

Inatel

Instituto Nacional de Telecomunicações

Avaliação da Internet do Futuro em
Ambientes Virtuais: Uma
Abordagem com a NovaGenesis

VICTOR HUGO DOMINGUES D'ÁVILA

JULHO / 2021



AVALIAÇÃO DA INTERNET DO FUTURO EM AMBIENTES VIRTUAIS: UMA ABORDAGEM COM A NOVA-GENESIS

VICTOR HUGO DOMINGUES D'ÁVILA

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.

D'Ávila, Victor Hugo Domingues

D259a

Avaliação da Internet do Futuro em Ambientes Virtuais: Uma Abordagem com a NovaGenesis. / Victor Hugo Domingues D'Ávila. – Santa Rita do Sapucaí, 2021.

167p.

Orientador: Prof. Dr. Antônio Marcos Alberti.

Dissertação de Mestrado em Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.

Inclui bibliografia e anexo.

1. NovaGenesis 2. Arquitetura de Internet do Futuro 3. Virtualização 4. Redes 5. Sistema operacional para embarcados. 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 16/07/2021,
pela comissão julgadora:

Prof. Dr. Antônio Marcos Alberti
INSTITUTO NACIONAL DE TELECOMUNICAÇÕES (INATEL)

Prof. Dr. Antônio Jorge Gomes Abelém
UNIVERSIDADE FEDERAL DO PARÁ (UFPA)

Prof. Dr. Guilherme Augusto Barucke Marcondes
INSTITUTO NACIONAL DE TELECOMUNICAÇÕES (INATEL)

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

*“Nada é mais poderoso do que uma
ideia cujo tempo chegou”*

Victor Marie Hugo

*Aos meus pais, Luzia e Sebastião,
exemplos de perseverança e integridade.*

Agradecimentos

Agradeço a Deus pela graça da vida e condições para desenvolver este trabalho, um projeto pessoal e profissional. A Ele toda a Glória.

Agradeço aos meus pais, Luzia e Sebastião, pelo apoio ilimitado, pelas palavras de motivação e carinho. Agradeço também a todos os familiares que não deixaram de conceder palavras de incentivo e orações, em especial às minhas avós, Leandra (*in memoriam*) e Maria, e meus avôs, Manoel e Benedito (*in memoriam*).

Agradeço ao professor e orientador Antônio Marcos Alberti pelas tutorias e visões que expandiram minha percepção da ciência e de novas tecnologias. Agradeço a todos os colegas do ICT-LAB pela colaboração nos diversos desafios que se pronunciaram. Nominalmente cito o recorrente apoio recebido do José Rodrigo dos Santos, Élcio Carlos do Rosário, Jorge Roberto Carneiro, Thiago Bueno da Silva, Tibério Tavares Rezende, Karine da Costa, Epper Bonomo, Monalisa Silva, Amanda Mendes, Vinícius Mendonça e Rodrigo Hilário. Agradeço aos colegas dos demais laboratórios do Inatel que também compartilharam propósitos e iniciativas, nominalmente cito o Lucas dos Santos Costa pelo tempo dedicado a ministrar cursos e compartilhar dicas de utilização do Latex.

Agradeço ao CRR por possibilitar a minha participação em projetos de primeira linha na área de telecomunicações. Reconheço a competência do Inatel e empenho de todos os seus funcionários que cuidam diariamente para disponibilizar os melhores recursos, pessoais e tecnológicos, aos seus alunos. Em especial cito a atuação da equipe do CTIC, por não medirem esforços para prover condições tecnológicas à realização deste trabalho. Agradeço às instituições RNP, FIBRE e GENI que receberam e hospedaram propostas de experimentação aqui citadas;

Agradeço ao IFMG campus Ouro Branco e aos colegas de trabalho.

Victor Hugo Domingues D'Ávila

Sumário

Sumário	xii
Lista de Figuras	xiv
Lista de Quadros	xv
Lista de Abreviaturas e Siglas	xvii
Lista de Símbolos	xxi
Resumo	xxiii
Abstract	xxv
1 Introdução	1
1.1 Problemas da Internet Atual	3
1.2 Propostas de Melhoria da Internet	5
1.3 NovaGenesis	6
1.4 Objetivos	7
1.5 Principais Contribuições	7
1.6 Publicações	8
1.6.1 Artigos Publicados em Conferências	8
1.6.2 Apresentações de Trabalhos Técnicos - <i>Workshops</i>	9
1.7 Entendimentos do Capítulo	9
1.8 Estrutura da Dissertação	9
2 Fundamentação Teórica e Revisão Bibliográfica	11
2.1 Busca Pelo Conhecimento	11
2.2 Virtualização	13
2.3 Experimentação em Redes e na Internet	15
2.3.1 PlanetLab	17
2.3.2 GENI	17
2.3.3 FIRE	18
2.3.4 FIBRE	18
2.3.5 Comparativo <i>Testbeds</i>	19
2.4 Internet do Futuro	20
2.4.1 Orientação a Conteúdos	21

2.4.2	Nomeação e Resolução de Nomes	22
2.4.3	CURLING	23
2.4.4	MobilityFirst	25
2.4.5	Named Data Networking	28
2.4.6	Comparativo entre FIAs e Oportunidades de Pesquisa	32
2.5	Entendimentos do Capítulo	35
3	NovaGenesis	37
3.1	O Início	37
3.2	Pilares Conceituais	38
3.2.1	1º Pilar: Linguagem	38
3.2.2	2º Pilar: Ciclo de Vida	39
3.2.3	3º Pilar: Virtualização e Representação	40
3.2.4	4º Pilar: Orientação a Serviços	41
3.3	Princípios do Projeto e Implementação	41
3.3.1	1º Princípio: Integração Coesa e Sinérgica	41
3.3.2	2º Princípio: Generalização ao Máximo	42
3.3.3	3º Princípio: Nomeação e Resolução de Nomes	42
3.3.4	4º Princípio: Roteamento por Nome e Armazenamento de Con- teúdos	42
3.3.5	5º Princípio: Auto-organização	42
3.3.6	6º Princípio: Processamento de Informação como Serviço	43
3.3.7	7º Princípio: Programabilidade do Físico	43
3.3.8	8º Princípio: Ciência de Contextos	43
3.3.9	9º Princípio: Generalização do Meio Físico	43
3.3.10	10º Princípio: Representação e Exposição do Físico	43
3.3.11	11º Princípio: Segurança Intrínseca	44
3.4	Fundamentações e Padrões de Comunicação	44
3.4.1	Ligação de Nomes - <i>Name Bind</i>	44
3.4.2	Publicação/Assinatura	45
3.4.3	Termos e Nomenclaturas da NovaGenesis	46
3.4.4	Detalhes de Implementação	46
3.5	Serviços NG e de Resolução de Nomes	48
3.5.1	Proxy Gateway Controller Service (PGCS)	48
3.5.2	Name Resolution and Networking Cache Service (NRNCS)	49
3.5.3	Aplicação NBSimpleTestAPP	50
3.6	Aplicação NBSimpletestAPP para Avaliação da NG	52
3.6.1	Publicação e Assinatura com o Name Bind Simple Test Appli- cation	52
3.6.2	Fluxo de Comunicação Entre Blocos da NG: Publicação e Assinatura de <i>Name Binds</i>	55
3.7	Entendimentos do Capítulo	57
4	Proposta de Metodologia de Teste e Ambiente de Avaliação em Escala para a NovaGenesis	59
4.1	NovaGenesis em <i>Testbeds</i>	59
4.2	Ambientes de Avaliação e Ferramentas	61

4.2.1	Ambiente Físico	61
4.2.2	Ambiente Virtual	63
4.2.3	Recursos de <i>Software</i> e Conectividade	66
4.3	Topologias de Avaliação e Procedimentos Operacionais	68
4.3.1	Tipos de Nós	68
4.3.2	Cenários com Múltiplos NSNs	70
4.4	Procedimentos Operacionais	74
4.4.1	Preparação dos Ambientes	75
4.4.2	Execução da NG	77
4.4.3	Coleta de Dados	78
4.4.4	Reinicialização e Manutenção	79
4.5	Entendimentos do Capítulo	80
5	Resultados e Análises	81
5.1	Ambiente Físico	81
5.1.1	Publicação de 1000 NBs por Mensagem	82
5.1.2	Assinatura de um Único NB Aleatório por Segundo	84
5.1.3	Comparação de Cenários com Máquinas Físicas	86
5.2	Ambiente Virtual	88
5.2.1	Publicação de 1000 NBs por Mensagem	88
5.2.2	Assinatura de um Único NB Aleatório por Segundo	90
5.2.3	Comparação de Cenários	92
5.3	Comparações entre os Ambientes Físico e Virtual	94
5.4	Entendimentos do Capítulo	96
6	Considerações Finais	97
6.1	Conclusões e Contribuições	97
6.2	Lições Aprendidas	98
6.3	Trabalhos Futuros	99
	Referências Bibliográficas	101
A	Ansible	1
A.1	Playbooks do Ansible - Scripts de execução	1
A.1.1	Playbook para Execução dos Cenários	1
A.1.2	Script para Execução do Nó Core (NCN)	2
A.1.3	Script para Execução do Nó de Armazenamento (NSN)	3
A.1.4	Script para execução do Nó Gerador de Tráfego TGN	4
A.1.5	Clean Script	5
A.1.6	Playbook para Reboot	6
A.1.7	Script para Copiar a NG	6
A.1.8	Script para Copiar os Logs de Execução do NBSimpletestAPP	6
A.1.9	Dependências e programas complementares	7
A.1.10	Definição e agrupamento de Nós	8
B	Configurações NG	13
B.1	PGCS	13

B.2	GIRS	13
B.3	PSS	13
B.4	HTS	14
B.5	NBSimpleTestAPP	14
C	Infraestrutura de Tecnologia, Informação e Comunicação	15
C.1	Firwall	15
C.1.1	Controle e acesso de VM's	15
D	Dados das experimentações	19
D.1	Dados de Publicação e Assinatura de NBs	19
	Glossário	21

Lista de Figuras

2.1	Diagrama representativo das relações do KVM com o hardware e sistemas operacionais hospedeiro e visitante.	15
2.2	Diagrama representativo do ciclo virtuoso do desenvolvimento tecnológico através do investimento em ambientes de experimentação e ciência.	15
2.3	Diagrama representativo das comunicações e topologia do CURLING. Fluxos de comunicação I, II e III.	25
2.4	Representação dos fluxos de comunicação <i>upstream</i> e <i>downstream</i> do NDN.	30
3.1	Fluxo de comunicação NG e a interação entra as camadas 1 e 2 do modelo de referência híbrido.	48
3.2	Sequência do fluxo de informação da publicação de NBs para os serviços da NG.	53
3.3	Sequência do fluxo de informação da assinatura de NBs para os serviços da NG.	55
3.4	Fluxo de comunicação para armazenamento de <i>name binds</i> em rede.	57
4.1	Diagrama representativo dos computadores dos laboratórios de informática.	62
4.2	Equipamento servidor de virtualização utilizado nos experimentos.	64
4.3	Tela de gerência do VMM que exhibe as máquinas virtuais e a utilização individual de CPU.	65
4.4	Ilustração dos fluxos de comunicação interno e externo de VMs em um <i>hypervisor</i> com <i>interface</i> em <i>bridge</i>	66
4.5	Topologia de comunicação, em camada 3, do <i>firewall</i> com os ambientes de experimentação virtual e físico.	67
4.6	Representação dos serviços executados pelo <i>Network Core Node</i> (NCN).	69
4.7	Representação dos serviços executados pelo <i>Network Storage Node</i> (NSN).	69
4.8	Representação dos serviços executados pelo <i>Traffic Generator Node</i> (TGN).	70
4.9	Representação do cenário contendo um HTS. A VM 201 acumula a função do NCN e do NSN, de forma excepcional.	72
4.10	Diagrama representativo para cenário contendo dois HTSes. Primeira abordagem em que é possível observar a distribuição de cargas do TGN a outros NCNs.	72

4.11	Diagrama representativo para cenário contendo três HTSes. Incremento no número de nós aptos a armazenar os NBs.	73
4.12	Diagrama representativo para cenário contendo quatro HTSes. Número máximo de instâncias avaliadas. A distribuição de elementos de armazenamento também é a maior entre todos os demais cenários.	74
4.13	Diagrama representativo dos procedimentos macros executados para avaliação da NG.	75
4.14	Diagrama representativo dos procedimentos de preparação dos ambientes de experimentação.	76
4.15	Diagrama representativo dos procedimentos de execução da NG.	77
4.16	Diagrama representativo dos procedimentos de coleta de dados da experimentação.	79
4.17	Diagrama representativo dos procedimentos de reinicialização e manutenção do ambiente de experimentação.	80
5.1	Publicação de NBs no ambiente físico para o Cenário 1 NSN.	82
5.2	Publicação de NBs no ambiente físico para o Cenário 2 NSNs.	83
5.3	Publicação de NBs no ambiente físico para o Cenário 3 NSNs.	83
5.4	Publicação de NBs no ambiente físico para o Cenário 4 NSNs.	84
5.5	Assinatura de NBs no ambiente físico para o Cenário 1 NSN.	84
5.6	Assinatura de NBs no ambiente físico para o Cenário 2 NSNs.	85
5.7	Assinatura de NBs no ambiente físico para o Cenário 3 NSNs.	85
5.8	Assinatura de NBs no ambiente físico para o Cenário 4 NSNs.	86
5.9	Comparação da publicação nos 4 cenários de laboratório.	87
5.10	Comparação da assinatura nos 4 cenários de laboratório físico.	88
5.11	Publicação de NBs no ambiente virtual para o Cenário 1 NSN.	89
5.12	Publicação de NBs no ambiente virtual para o Cenário 2 NSNs.	89
5.13	Publicação de NBs no ambiente virtual para o Cenário 3 NSNs.	90
5.14	Publicação de NBs no ambiente virtual para o Cenário 4 NSNs.	90
5.15	Assinatura de NBs no ambiente virtual para o Cenário 1 NSN.	91
5.16	Assinatura de NBs no ambiente virtual para o Cenário 2 NSNs.	91
5.17	Assinatura de NBs no ambiente virtual para o Cenário 3 NSNs.	92
5.18	Assinatura de NBs no ambiente virtual para o Cenário 4 NSNs.	92
5.19	Comparação dos resultados de publicação nos quatro cenários virtuais.	93
5.20	Comparação dos resultados de assinatura nos quatro cenários virtuais.	94
5.21	Comparação dos resultados de publicação de NBs para os quatro cenários dos ambientes virtual e físico (laboratórios).	95
5.22	Comparação dos resultados de assinaturas de NBs para os quatro cenários dos ambientes virtual e físico (laboratórios).	95
C.1	Topologia Física da Rede ICTLab	16
C.2	Tela de monitoramento do Zabbix.	17

Lista de Quadros

1.1	Representação das camadas de rede dos modelos de referência Híbrido e TCP/IP.	3
2.1	Comparativo entre <i>testbeds</i>	20
2.2	Representação das camadas de rede do modelo de referência híbrido e da NDN.	29
2.3	Comparativo FIAs	34
3.1	Definição de termos da NovaGenesis.	46
3.2	Parâmetros de inicialização do NBSimpleTestAPP	51
4.1	Especificações da configuração mínima de <i>hardware</i> dos computadores dos laboratórios.	62
4.2	Especificações da configuração de <i>hardware</i> do servidor de experimentação.	63
4.3	Especificações dos recursos das VMs.	63
4.4	Duração aproximada do fluxo de Execução da NG para diferentes quantitativos de NBs.	78
C.1	Detalhe dos <i>alias</i> de endereços e portas do <i>firewall</i>	15
C.2	Interface WAN - Regras de Firewall.	16
C.3	Interface LAN - Regras de Firewall	16
C.4	Interface LAN180ICT - Regras de Firewall	17
C.5	Interface LAN Smart Campus - Regras de Firewall	17
C.6	Interface OpenVPN - Regras de Firewall	18
C.7	VM's ICT-LAB	18

Lista de Abreviaturas e Siglas

4WARD	Architecture and Design for the Future Internet
5G	Redes de quinta geração
5G PPP	5G Infrastructure Public Private Partnership
AKARI	AKARI Architecture Design Project
API	Application programming interface
ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous system
ATM	Asynchronous Transfer Mode
CA	Certificate Authority
CaR	Content-aware Router
CCN	Content Centric Network
CCNx	Content-Centric Networking
CDN	Content Delivery Network
CLNP	Connectionless Network Protocol
CRC	Content resolution controller
CS	Content Store
CURLING	Content-Ubiquitous Resolution and deLivery Infrastructure for Next-Generation Services
DARPA	Defense Advanced Research Projects Agency
DHT	Distributed Hash Table
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DONA	Data-Oriented Network Architecture
DSL	Digital Subscriber Line
ETArch	Entity Title Architecture
Fed4FIRE	Federation for FIRE
Fed4FIRE+	Federation for FIRE Plus
FI	Future Internet
FIA	Future Internet Architecture
FIB	Forwarding Information Base
FIBRE	Future Internet Brazilian Environment for Experimentation
FIRE	Future Internet Research and Experimentation
FITS	Future Internet Testbed with Security
FTP	File Transfer Protocol
GENI	Global Environment for Network Innovations

GIRS	Generic Indirection Resolution Service
GNS	Global Name Service
GUID	Globally Unique Identifier
HTS	Hash Table Service
HTTPS	Hyper Text Transfer Protocol Secure
ICMP	Internet Control Message Protocol
ICN	Information Centric Networks
ICT	Information and Communications Technologies
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IoT	Internet das Coisas - <i>Internet of Things</i>
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
JGN	Japan Gigabit Network
JGN2plus	Japan Gigabit Network 2 plus
JGN-X	Japan Gigabit Network eXtreme
LXC	LinuX Containers
MF	MobilityFirst
M-Lab	Measurement Lab
NA	Network Address
NAT	Network Address Translation
NB	Name binding
NBSimpletestAPP	Name Bind Simple Test Application
NBs	Name bindings
NCN	Network Core Node
NDN	Named Data Networking
NFV	Network Functions Virtualization
NG	NovaGenesis
NIC	Network Interface Controller
NRNCS	Name Resolution and Networking Cache Service
NSF	National Science Foundation
NSFNET	National Science Foundation Network
NSN	Network Storage Node
OMF	Orbit Management Framework
ORBIT	Open access Research testBed for next-generation wireless networks
ORCA	Open Resource Control Architecture
OSI	Open Systems Interconnection
PGCS	Proxy Gateway Controller Service
PIT	Pending Interest Table
PSS	Publish Subscribe Services
PT	Principal Type
PUB/SUB	Publish/Subscribe
PUB/SUB	Publish/Subscribe

QoS	Quality of Service
RAM	Random-access memory
RC	Resource Controller
RFC	Request for Comments
RINA	Recursive InterNetwork Architecture
RNP	Rede Nacional de Ensino e Pesquisa
RTT	Round Trip Time
SCPI	Standard Commands for Programmable Instruments
SDN	Software-Defined Networking
SNMP	Simple Network Management Protocol
SO	Sistema Operacional
SOA	Service-Oriented Architecture
SSH	Secure Shell
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
<i>testbeds</i>	Testbeds
TGN	Traffic Generator Node
UDP	User Datagram Protocol
UFRGS	Universidade Federal do Rio Grande do Sul
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMM	Virtual Machine Manager
VMs	Virtual Machines
XCP	EXplicit Control Protocol
XIA	EXpressive Internet Architecture

Lista de Símbolos

μ	Tempo médio
μ_{low}	Média menos a margem de erro
μ_{up}	Média mais a margem de erro
N	Número de eventos
σ	Desvio padrão
σ_{μ}	Erro padrão a partir da média
x	Tempo instantâneo

Resumo

D'Ávila, V. H. D. Avaliação da Internet do Futuro em Ambientes Virtuais: Uma Abordagem com a NovaGenesis [dissertação de mestrado]. Santa Rita do Sapucaí: Instituto Nacional de Telecomunicações; 2021.

O aumento pela demanda de conectividade é algo recorrente a todas esferas da sociedade. Iniciativas globais buscam interiorizar e democratizar o acesso a este bem público. Buscando atender necessidades de melhoria da Internet, tais como mobilidade, segurança e identificação de objetos na rede, várias iniciativas propõem alternativas à arquitetura da Internet como é conhecida atualmente, as chamadas propostas de Internet do Futuro. Nesse contexto, uma questão importante é: como poderíamos construir um novo sistema de resolução de nomes para a Internet se não houvesse o *Domain Name System* (DNS)? Este trabalho visa conceituar algumas dessas iniciativas e avaliar uma proposta brasileira: a NovaGenesis. Essa proposta tem pilares inovadores, definindo por *software* elementos de rede. Emprega também conceitos como distribuição nativa de conteúdos, separação de identificadores e localizadores, auto nomeação e certificação de nomes. O presente trabalho busca avaliar o ambiente NovaGenesis virtual e comparar com implementações em ambientes físicos. Para isso, o trabalho apresenta uma proposta de ambiente para experimentação e avaliação de desempenho de cenários de resolução nomes em um domínio local, bem como propõe uma metodologia de testes a ser utilizada. Os resultados comprovam as vantagens do método de testes no ambiente virtual desenvolvido, bem como as vantagens da NovaGenesis quanto à resolução de nomes com *cache* de rede distribuído e auto-organizável. Conclui-se que a NovaGenesis é uma arquitetura alternativa, com desempenho satisfatório em um domínio local, para ampliar o suporte à resolução de nomes na Internet, com armazenamento de registros em *cache* de rede e acesso via modelo de comunicação publica/assina.

Palavras-Chave: NovaGenesis, Arquitetura de Internet do Futuro, Virtualização, Redes

Abstract

D'Ávila, V. H. D. Assessing the Future Internet in Virtual Environments: An Approach with NovaGenesis [MSc dissertation]. Santa Rita do Sapucaí: Instituto Nacional de Telecomunicações; 2021.

The increase in the demand for connectivity is something that recurs in all spheres of society. Global initiatives seek to internalize and democratize access to this public good. Seeking to meet the needs of Internet improvement, such as mobility, security and identification of objects on the network, several initiatives propose alternatives to the Internet architecture as it is currently known, the so-called Internet of the Future proposals. In this context, an important question is: how could we build a new name resolution system for the Internet if there were no Domain Name System (DNS) ? This work aims to conceptualize some of these initiatives and evaluate a Brazilian proposal: NovaGenesis. This proposal has innovative pillars, defining network elements by software. It also employs concepts such as native content distribution, separation of identifiers and locators, self-naming and name certification. The present work seeks to evaluate the NovaGenesis virtual environment and compare it with implementations in physical environments. For this, the work presents an environment proposal for experimentation and performance evaluation of name resolution scenarios in a local domain and proposes a test methodology to be used. The results prove the advantages of the test method in a developed virtual environment, as well as the advantages of NovaGenesis in name resolution with distributed and self-organizing network cache. It is concluded that NovaGenesis is an alternative with satisfactory performance in a local domain to expand the support for name resolution on the Internet, with storage of records in network cache and access via publish/subscribe communication model.

Keywords: NovaGenesis, Future Internet Architecture, Virtualization, Network

Capítulo 1

Introdução

A Internet é uma criação humana complexa e de grande sucesso [1]. Suas características de adaptabilidade e flexibilidade contribuem para isso. Esse sucesso é tamanho que em apenas uma geração ela deixou de ser exclusividade dos laboratórios de pesquisa e passou a ser *commodity* e um serviço essencial à Humanidade [2]. Sua importância é tão marcante que a maioria das mídias e sistemas de comunicação, em um momento ou outro, convergem para se integrar a ela [3].

A abrangência de seu alcance e o rápido desenvolvimento passam por suas definições de projeto, que de forma adaptativa [4], estabelecem soluções incrementais às demandas que surgem. Muito desse sucesso se deve à pilha de protocolos de comunicação que predomina em suas implementações. O modelo, inclusive, recebeu o nome dos dois protocolos de maior destaque, o *Transmission Control Protocol / Internet Protocol* (TCP/IP), que conquistavam importante espaço entre pesquisadores e agências de pesquisas, como a *Defense Advanced Research Projects Agency* (DARPA) [5]. Em momentos do passado o TCP/IP precisou ganhar destaque frente ao modelo de referência existente, o *Open Systems Interconnection* (OSI) [6]. Assim, a iniciativa oriunda do mundo científico ganhou o mercado primeiro e fez frente ao modelo com poucas implementações do seu concorrente. A arquitetura da Internet deriva de alguns projetos de conectividade [7] com objetivos de promoção científica, como a *Advanced Research Projects Agency Network* (ARPANET), e uma outra iniciativa com foco na expansão de conectividade em instituições de ensino, a *National Science Foundation Network* (NSFNET) [8]. Esse viés acadêmico contribuiu para que seus conceitos fossem sempre difundidos de forma aberta, via *Request for Comments* (RFC), e discutidos de forma técnica em grupos de trabalhos, como o *Internet Engineering Task Force* (IETF). Essas e outras políticas, igualmente colaborativas, contribuíram para a expansão e desenvolvimento das redes de computadores e a criação de serviços digitais.

A NSFNET foi uma parceria público-privada dos Estados Unidos que, em um curto período de tempo (1987 a 1995), conseguiu unir empresas, universidades e o governo em uma iniciativa de levar rápido acesso à informação para todo o território americano. Em 1988 existiam 210 redes integradas, em 1995, ano do seu encerramento, mais de 50.000 pontos já estavam interconectados [8].

O encerramento dessa parceria estabelecia a transferência de tecnologia para as empresas. O legado disso foi um *backbone* de alta capacidade para a época, capaz de

interligar instituições de ensino e expandir o seu alcance a clientes particulares. Dada essa oferta de interligações, começaram a surgir provedores locais de conectividade e a cobertura de clientes comerciais e residenciais aumentou, fomentando ainda mais o desenvolvimento de redes de computadores e formas de se conectar à Internet.

Antes de apresentar outros conceitos relevantes para o entendimento do objetivo deste trabalho, algumas definições precisam ser feitas [9, 10]:

- **Protocolo:** Processo que estabelece formato, ordem e ações de comunicação entre uma ou mais entidades.
- **Camada:** Agrupamento de funcionalidades e protocolos de rede que atuam com objetivo de realizar um conjunto específico de serviços de rede. Cada camada encapsula os dados (conteúdo útil) da camada anterior/posterior de forma a não precisar analisá-los para realizar suas funções.
- **Interface:** Ponto de contato entre duas camadas, meio que estabelece comunicação e integração entre diferentes camadas.
- **Serviço de rede:** São ações realizadas por uma camada com os dados que lhe são entregues. Na maioria das vezes tem como função enviar, receber, processar ou encaminhar informações.
- **Modelo de referência:** Representação das camadas de uma arquitetura de rede. Para efeitos de comparação, será considerado o modelo de referência híbrido, exposto no Quadro 1.1 e detalhado por Tanenbaum [10].
- **Pilha de protocolos:** Conjunto de protocolos implementados em uma rede. Normalmente se referencia um protocolo por camada. No modelo TCP/IP, o *Transmission Control Protocol* (TCP) corresponde à camada de transporte e o *Internet Protocol* (IP) à camada Internet, responsável pelo endereçamento de rede e o roteamento.

Complementando as informações acima, são descritas as camadas e funcionalidades do modelo de referência Híbrido e do modelo TCP/IP puro no Quadro 1.1. O que muda entre eles é a nomenclatura das camadas e a ausência da camada física para o modelo TCP/IP.

- **Camada de Aplicação:** está relacionada às aplicações de rede ou dos usuários como: *Domain Name System* (DNS), *Secure Shell* (SSH), *Hyper Text Transfer Protocol Secure* (HTTPS), *Simple Network Management Protocol* (SNMP).
- **Camada de Transporte:** está relacionada com as técnicas de controle e envio de informações: *User Datagram Protocol* (UDP) e TCP.
- **Camada de Rede:** está relacionada à identificação de nós e roteamento de informação para os dispositivos de rede: IP, *Internet Control Message Protocol* (ICMP), *Internet Group Management Protocol* (IGMP) e *Internet Protocol Security* (IPsec).
- **Camada de Enlace de Dados:** estabelece condições de controle e otimização do fluxo de informações do meio físico: *Asynchronous Transfer Mode* (ATM), *Virtual Local Area Network* (VLAN) e *Address Resolution Protocol* (ARP).
- **Camada Física:** responsável pelas funcionalidades e definições do meio físico em que são transmitidas as informações, alguns protocolos dessa camada são:

IEEE 802.3, IEEE 802.11, *Integrated Services Digital Network* (ISDN) e *Digital Subscriber Line* (DSL).

Quadro 1.1: *Representação das camadas de rede dos modelos de referência Híbrido e TCP/IP.*

Modelo Híbrido		Modelo TCP/IP	
Nível	Denominação	Nível	Denominação
5	Aplicação	4	Aplicação
4	Transporte	3	Transporte
3	Rede	2	Internet
2	Enlace de dados	1	Enlace de dados
1	Física		

Fonte: Adaptado de Tanenbaum, 2010 [10].

A estruturação dos protocolos TCP/IP foi substancialmente importante para a evolução das camadas de aplicações e dos serviços da Internet. Mesmo assim, não é possível deixar de citar que os mesmos alicerces que compõem sua arquitetura, de certa forma, a prendem a essas definições um tanto quanto inflexíveis para a atualidade.

1.1 Problemas da Internet Atual

Uma das características estruturantes dos modelos de referência é o encapsulamento da pilha de protocolos [11]. Esse encapsulamento se resume em isolar as várias etapas de processamento das informações de forma que a informação enviada/recebida da camada anterior/posterior não influencie a execução das funcionalidades de outras camadas.

Essa característica tem pontos positivos e negativos. Dados os diversos cenários em que ela é eficiente, não houve condições de prever que as interações sociais migrariam de uma forma tão intensa para o mundo virtual e novas funcionalidades seriam exigidas da rede. A Internet atual encontra dificuldades em resolver demandas da sociedade em relação às técnicas e processos do sistema de redes, como: nomeação de entidades [12], identificação [13], segurança [14] e mobilidade [15]. Essas funcionalidades não são implementadas de forma natural em suas camadas. Na maioria das vezes, é necessário utilizar aplicações específicas para obtê-las.

A nomeação de entidades, como dispositivos, pessoas, objetos e tudo mais que precise ser representado virtualmente, não é algo dinâmico e fácil de ser alcançado. Limitações quanto ao esgotamento de endereços disponíveis já são realidade e estão sendo tratadas com a implementação paralela do *Internet Protocol version 6* (IPv6) [16].

Alia-se a isso a dificuldade de estabelecer relação de confiança entre os endereços dos dispositivos e os conteúdos que estes disponibilizam/acessam. Não existe um mecanismo em nível da camada de rede que permita o estabelecimento de confiança entre os dispositivos. Os endereços IPs não são unicamente definidos e/ou certificáveis.

Quando se fala em segurança é necessário pensar em quatro requisitos [17]:

- **Disponibilidade:** Capacidade do conteúdo ou do nó de permanecer disponível na rede.

- **Integridade:** Capacidade de controlar alteração em informações e conteúdos.
- **Confidencialidade:** Capacidade de criptografar comunicações e conteúdos para acesso apenas dos comunicantes.
- **Autenticidade:** Garantia de autoria da produção ou modificação de informação/conteúdo.

Esses requisitos são condições mínimas para se obter uma comunicação segura. Eles não são alcançáveis de forma integral pela camada de rede, mas apenas através de aplicações das camadas superiores.

A mobilidade é outro fator que afeta usuários em constante movimento. Apesar de imperceptível, a cada mudança de rede, durante uma movimentação diária (e.g. casa-escritório-casa), o dispositivo de um usuário tem seus endereços de rede renovados. Com isso, perdem-se as conexões estabelecidas e que dependem de uma comunicação permanente, via *socket*, por exemplo.

Outro problema bem crítico está relacionado ao DNS. Ele é um serviço de destaque e a usabilidade da Internet é dependente dele. Com ele os usuários não precisam lembrar dos endereços IPs dos *website*, aplicações e/ou *hosts* de conteúdos que desejam acessar. De forma simplificada, ele funciona da seguinte forma:

- O usuário que deseja acessar um conteúdo busca a URL do *website* que provê tal conteúdo na Internet. A URL está em uma linguagem de fácil memorização e utilização pelo usuário (e.g. www.inatel.br).
- Para converter a URL em endereço IP é necessário consultar o serviço de registro de nomes da Internet, o DNS.
- Feita a consulta, o usuário recebe o endereço IP do *website* que quer acessar (e.g. 203.0.113.2). A partir de então tem-se o endereço IP do provedor do conteúdo e é possível direcionar requisições de acesso ao *website* e receber o conteúdo desejado.

Observando-se o funcionamento do DNS, percebe-se que para identificar o endereço IP de um *host* é necessário que se percorra todas as camadas de rede até encontrar a aplicação DNS em um servidor remoto. A partir disso, recebe-se o endereço IP do *host* solicitado e então é possível acessar o conteúdo desejado. Essa forma de navegação não é objetiva aos conteúdos que se quer acessar, mas sim aos *sites* que hospedam tais conteúdos. Ela também deposita significativa importância no DNS, de forma que um serviço em nível da camada de aplicação passa a ser exigido para que se alcance os *websites*.

Assim, a arquitetura atual da Internet não consegue atuar sem ter dependência de outras aplicações. Sendo uma arquitetura de propósito genérico, a implementação de um serviço, como a resolução de nome, é possível apenas em suas camadas superiores, especificamente, na camada de aplicação. Percebe-se também uma orientação de conectividade ao provedor *host* do conteúdo e não ao conteúdo por si só. Isso indica que, apesar de adaptável, a Internet atual depende de camadas superiores para permitir a navegabilidade dos usuários. A dependência de aplicações em diferentes camadas pode ser um ponto de limitação da arquitetura, dificultando a manutenção e operação da Internet, como no caso em que associações de nomes (DNS) e endereços IPs passam

a ser definidas como elementos de segurança, mesmo sem que suas camadas possuam qualquer relação [18].

1.2 Propostas de Melhoria da Internet

As demandas da atualidade desafiam a Internet de se manter da forma como foi iniciada. Até o momento, adaptações e aplicações incrementais estão suportando mudanças, porém, existem discussões mais profundas a esse respeito [19]. Algumas propostas indicam que é necessário uma remodelação estrutural profunda da arquitetura, denominada *clean slate* - folha em branco, uma reconstrução a partir do zero [20]. Outras propostas indicam que as melhorias devem ser incrementais e adaptativas, mantendo o núcleo da rede, que se mostra robusto em suas implementações até os dias de hoje.

O modelo *clean slate* se justifica mediante a sua proposta de adequação integral às demandas da sociedade atual. A abordagem dessa reconstrução incentiva pesquisadores a entenderem profundamente as carências estruturais atuais e a idealizar uma implementação capaz de atendê-las. O desafio está em propor soluções a questões não resolvidas da arquitetura atual, tais como: escalabilidade independente da nomeação hierárquica [21], roteamento de tráfego baseado em nome (sem dependência com serviços terciários) [22] e capacidade de estabelecer identidades que não possam ser falsificadas [23].

Em uma outra abordagem, o modelo evolucionário aproveita as implementações de sucesso da Internet e propõe melhorias [24]. Desse ponto de vista, reduz o impacto financeiro de substituir toda a infraestrutura existente e permite alterações dinâmicas de protocolos. Defensores dessa visão indicam que a abordagem *clean slate* não é nova, tendo sido implementada em outras ocasiões sem sucesso nos protocolos de controle de congestionamento de tráfego *eXplicit Control Protocol* (XCP) [25] e *Connectionless Network Protocol* (CLNP) [26]. Mesmo propostas incrementais que podem ser executadas em paralelo enfrentam resistência, vide a lenta adoção ao IPv6 [27]. Uma justificativa que pode ser encontrada para esses fatos é a explosão de serviços e aplicações [19] que acabam priorizando esforços de desenvolvimento à camada de aplicação, assim, as camadas mais baixas ficam presas em lentos processos de atualização, não por falta de iniciativas a mudanças, mas pela necessidade de tempo para que os provedores de conectividade se atualizem sem gerar indisponibilidades nas redes já estabelecidas.

Com intuito de investigar esse tema, pesquisadores se unem em torno de soluções para a denominada Internet do Futuro - *Future Internet* (FI) [28–30]. Nesse ramo as pesquisas de Arquiteturas de Internet do Futuro - *Future Internet Architecture* (FIA) [31–33] se intensificam, sendo alvo de estudos de grupos de cientistas da área da computação e engenharia.

Algumas propostas como *Content-Ubiquitous Resolution and deLivery Infrastructure for Next-Generation Services* (CURLING) [34], que associa implementações de *Software-Defined Networking* (SDN) [35] para entrega de conteúdos; *Named Data Networking* (NDN) [12] e *MobilityFirst* (MF) [31], que oferecem soluções nativas de mobilidade, baseadas em verificação intrínseca de endereços nomeados e localizado-

res, endereçam as limitações da Internet atual e possuem suas formas de resolução de nomes/conteúdos. Assim, essas propostas serão detalhadas em suas definições e experimentações no restante deste trabalho.

Essas e outras pesquisas em FIAs buscam analisar formas de estabelecer melhorias para a Internet e específico na metodologia de resolução de nomes, que na arquitetura atual é majoritariamente executada pelo DNS. As avaliações das FIAs são abertas e receptivas a críticas. As propostas de melhorias podem ser incrementais, sugerir alterações de protocolos, mudanças estruturantes e até a substituição da Internet por uma nova arquitetura. Elas também fomentam a discussão de questões impactantes para a arquitetura atual, idealizam aplicações de conceitos nativos de segurança, mobilidade, nomeação e outras funcionalidades baseadas nas demandas atuais.

A construção dos conceitos de cada uma dessas propostas não é apenas teórica. Para estabelecer e viabilizar seus projetos os autores buscam parcerias em iniciativas locais e internacionais, seja para fomento [27] ou para utilizar ambientes de experimentação compartilhados [36]. Tais ambientes são recorrentes na literatura e se baseiam no modelo inicial da ARPANET [37], comumente denominados *testbeds*.

1.3 NovaGenesis

A NovaGenesis (NG) [38–41] é uma FIA que se conceitua como uma proposta de arquitetura de Internet implementada a partir do zero (*clean slate*). Seus pilares básicos são: linguagem que permite nomear, identificar e estabelecer relações entre as entidades da rede; definição de ciclos de vida para todas as entidades da arquitetura; máxima virtualização de entidades físicas, o que permite lidar com diferentes equipamentos e com a heterogeneidade de uma rede; orientação a serviços: de forma que todas entidades, funções, objetos e a troca de informações possam prever condições de interação na rede com base em acordos de serviços. Associados a esses pilares fundamentais estão previstas regras de implementações que orientam a codificação de seus serviços. Sua implementação atual opera acima da camada de enlace de dados. Isso significa que o protótipo atual da NG possui implementações funcionais de seus conceitos no que pode ser traduzido para camada de rede, transporte e aplicação [42]. Entretanto, a NG permite a definição dinâmica de camadas e seus pontos de integração.

A NG propõe alternativa ao atual modelo de nomeação e resolução de nomes da Internet. A sua implementação permite nomear dispositivos, conteúdos e serviços, sempre de forma autocertificável. Em consequência disso é possível localizar e resolver nomes de rede, conteúdos e serviços de forma distribuída e segura. Portanto, trata-se de uma alternativa à atual solução de nomeação de nós, serviços, domínios, páginas *web* etc. Assim sendo, o objetivo deste trabalho é avaliar a solução que a NG oferece para as limitações da Internet atual no que tange à nomeação, representação de relacionamento entre entidades usando ligações entre nomes, armazenamento e resolução de nomes de forma distribuída e alternativa ao DNS.

Uma dificuldade que se tem em relação à NG é a sua avaliação em diferentes topologias, cenários e ambientes. Apesar de propor a resolução de algumas deficiências da Internet, ela não possui maturidade para ser implementada em total substituição à

arquitetura atual. Não obstante, pesquisadores, empresas e estudantes, que possuem acesso ao seu código, buscam formas de avaliá-la e compará-la das mais diversas formas, executando implementações locais, em *testbeds*, em conjunto com tecnologias de Internet das Coisas - *Internet of Things* (IoT) [43], rádio cognitivo [44] e até em soluções de interconexão de múltiplas FIAs [45]. Essas formas de avaliação contribuem para o desenvolvimento e amadurecimento dos princípios e implementações da arquitetura.

1.4 Objetivos

Dentre diversas propostas de FIAs, a NG é um projeto de pesquisa que apresenta uma arquitetura *clean slate* como solução para alguns problemas da Internet. A necessidade de avaliá-la em suas variações de topologias de serviços e em quantidade de nós em um domínio local foi o que motivou este trabalho. Estabelecendo-se então os principais objetivos deste trabalho são:

- Avaliar a arquitetura de forma automatizada em ambientes físicos e virtuais. Assim, buscar identificar os seus pontos fortes e fracos, a fim de melhorar e ampliar ainda mais a sua base de testes, em número de nós e em formas de estudá-la;
- Definir metodologias, avaliar a NG e suas topologias no que tange ao serviço de resolução de nomes;
- Avaliar a NG em cenários com diferentes composições do seu serviço de armazenamento temporário de conteúdos.
- Avaliar o desempenho da NG com relação ao armazenamento temporário de conteúdos em *cache* local e a resolução de nomes, pois até o momento os testes realizados foram primordialmente executados em ambientes pouco escalonáveis e que demandavam monitoramento e acompanhamento manual por uma equipe no local.
- Obter métricas para avaliar o serviço de nomeação da NG em outros ambientes de experimentação, como *testbeds* e em virtualização baseada em contêineres.

1.5 Principais Contribuições

Dado o contínuo desenvolvimento da NG, este trabalho conseguiu alcançar alguns objetivos que servirão para análise, melhorias de sua implementação e validação das suas propostas. Como principais contribuições obtidas, é possível citar:

- (i) Definição de metodologias para avaliação da NG: foram desenhados métodos e criadas rotinas para automatizar e instanciar a arquitetura em múltiplos nós, máquinas virtuais ou computadores.
- (ii) Avaliação do serviço de resolução de nomes da NG: foi utilizado um nó independente para executar um serviço de publicação e assinatura de *name bindings* (NBs) e avaliar o serviço de resolução de nomes com base nos dados de desempenho obtidos para cada uma das ações. Tais dados são os tempos *Round Trip Time* (RTT) de publicação e assinatura, medidos a partir do nó de aplicação.

- (iii) Avaliação dos serviços da NG executados em diferentes ambientes de experimentação, topologias de serviços e cenários: foi possível variar o ambiente de experimentação, entre físico e virtual; variar as topologias dos serviços da NG, definindo nós específicos para os diferentes serviços e alternar os cenários com diferentes números de nós em execução. Dessa forma, foi possível observar o desempenho e comportamento da resolução de nomes, assinatura e publicação de NBs.
- (iv) Obtenção de dados de desempenho da NG para comparação com outras abordagens práticas e/ou plataformas de experimentação: foi possível padronizar a execução das avaliações da NG e sua resolução de nomes, de forma a permitir que se possa fazer comparações com outros ambientes (e.g. máquinas virtuais, máquinas físicas, contêineres e *testbeds*). Os principais dados obtidos são os tempos RTT de publicação e assinatura de NBs medidos a partir do nó de aplicação.

1.6 Publicações

No decorrer do desenvolvimento deste trabalho alguns resultados foram obtidos e publicados em cinco eventos científicos, dois deles internacionais e outros três nacionais. Outras duas apresentações técnicas foram realizadas em *workshops* voltados para implementações de *testbeds* e *cloud networks*.

1.6.1 Artigos Publicados em Conferências

- (i) REZENDE, Tibério; D'ÁVILA, Victor; COSTA, Karine; ALBERTI, Antônio. Distribuição de Conteúdo via Conexão 5G Híbrida Terrestre/Satélite. **Anais do XII Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP)**, Cuiabá, p. (151-160), dez. 2020. ISSN 2595-6183. Disponível em: <https://doi.org/10.5753/sbcup.2020.11221>.
- (ii) ROSÁRIO, Élcio; D'ÁVILA, Victor; SILVA, Thiago; ALBERTI, Antônio. A Docker-Based Platform for Future Internet Experimentation: Testing NovaGenesis Name Resolution. **IEEE Latin-American Conference on Communications (LATINCOM)**, Salvador, p. (1-5), 2019. Disponível em: <https://ieeexplore.ieee.org/document/8937898>.
- (iii) D'ÁVILA, Victor; ROSÁRIO, Élcio; SANTOS, José; ALBERTI, Antônio. NovaGenesis NameBindings: Avaliação de Virtualização em Escala. **Anais do X Workshop de Pesquisa Experimental da Internet do Futuro**, Gramado, p. (7-12), mai., 2019. Disponível em: <https://sol.sbc.org.br/index.php/wpeif/article/view/7692>.
- (iv) ALBERTI, Antônio; ROSÁRIO, Élcio; CASSIANO, Giovani; D'ÁVILA, Victor; CARNEIRO, Jorge. Performance Evaluation of NovaGenesis Information-Centric Network. **II International Multidisciplinary Conference on Computer and Energy Science (SpliTech)**, Split, Croácia, p. (1-6), jul., 2017. Disponível em: <https://ieeexplore.ieee.org/document/8019276>.

- (v) FERREIRA, Fábio; MIRANDA, Felipe; ROSÁRIO, Élcio; D'ÁVILA, Victor; ALBERTI, Antônio. NovaGenesis no Ambiente FIBRE: Primeiras Impressões. **Anais do VIII Workshop de Pesquisa Experimental da Internet do Futuro**, Belém, p. (6–9), mai., 2017. Disponível em: <https://sol.sbc.org.br/index.php/wpeif/article/view/2602>.

1.6.2 Apresentações de Trabalhos Técnicos - *Workshops*

- (i) D'ÁVILA, Victor; ROSÁRIO, Élcio; MIRANDA, Felipe; ALBERTI, Antônio. NovaGenesis no Ambiente FIBRE: Desempenho da Troca de Conteúdos Nomeados. **2º Workshop do Testbed FIBRE**, São Paulo, jul., 2017. Disponível em: <http://indico-memoria.rnp.br/conferenceDisplay.py?confId=239> e <http://csbc20-17.mackenzie.br/anais/eventos/2-workshop-do-testbed-fibre>.
- (ii) D'ÁVILA, Victor; ALBERTI, Antônio; SCARPIONI, Gabriel; MAGALHÃES, Vaner; CERQUEIRA, Arismar; RODRIGUES, Joel; RIGHI, Rodrigo. Demonstration of Future Internet of Things with NovaGenesis. **2º Cloudscape Brazil and Workshop on Cloud Networks**, São Paulo, jul., 2017. Disponível em: <https://eubrasilcloudforum.eu/en/service-demo/demonstration-future-internet-things-novagenesis>.

1.7 Entendimentos do Capítulo

Neste capítulo é retomado o início do desenvolvimento da Internet e seu estabelecimento através de diversos órgãos da sociedade, principalmente aqueles que possuem viés científico. Em seguida é feita uma explanação de conceitos e termos de arquiteturas de redes. Uma visão geral sobre os principais problemas da Internet também é apresentada, bem como as propostas que surgem para resolver tais problemas. A NovaGenesis é inserida como uma proposta de Arquitetura de Internet do Futuro e então são apresentados os objetivos, contribuições e publicações resultantes deste estudo.

1.8 Estrutura da Dissertação

Esta dissertação é composta de seis capítulos. O primeiro capítulo faz uma introdução às motivações e objetivos do estudo. No Capítulo 2 são apresentadas as fundamentações teóricas e conceituais necessárias para o entendimento das principais contribuições desta dissertação. São explicitadas no Capítulo 3 as ideias e conceitos fundamentais da arquitetura convergente de informação NovaGenesis. Uma apresentação prática é apresentada no Capítulo 4, em que são expostas as metodologias e ferramentas utilizadas na avaliação da NG. O quinto capítulo compreende os resultados obtidos durante o trabalho de pesquisa e suas análises, que comprovam com resultados numéricos que a NG é uma alternativa que merece ser estudada quanto às limitações de nomeação, resolução de nomes, publicação e assinatura de conteúdos da Internet atual. Por fim, as considerações do autor são apresentadas no sexto e último capítulo.

Capítulo 2

Fundamentação Teórica e Revisão Bibliográfica

Este Capítulo detalha a metodologia de revisão bibliográfica utilizada, define tecnologias chaves, faz um breve histórico sobre a importância da experimentação para o surgimento da Internet, aborda os conceitos e arquiteturas relevantes para este trabalho, bem como apresenta uma descrição de como elas são avaliadas. Em seguida, é feita uma avaliação das oportunidades de pesquisa que surgem em relação ao tema, à NovaGenesis (NG), às técnicas de nomeação e resolução de nomes, e também são definidas algumas tecnologias utilizadas como ferramentas neste trabalho.

2.1 Busca Pelo Conhecimento

Os pilares desse estudo são baseados em conceitos de FIA, NG, ambientes de experimentação, nomeação, publicação e assinatura de conteúdos em rede. Alguns dos tópicos que são abordados a seguir possuem interseção de conteúdos. Entretanto, busca-se deixar claro que o trabalho desenvolvido é inédito em sua execução, principalmente por avaliar um conceito muito específico da NG em ambientes que foram criados e personalizados para que se pudesse alcançar tais objetivos.

Ao todo foram abordadas 85 referências, das quais, contemplam-se: artigos científicos, livros acadêmicos, sites oficiais (relacionados a projetos, iniciativas de pesquisa ou tecnologias empregadas), *Request for Comments* (RFC), *white papers* e documentos de especificações técnicas. As principais plataformas de busca científica utilizadas foram: Google Scholar¹, Elsevier², IEEE Xplorer³, *Association for Computing Machinery* (ACM)⁴ e Springer Nature⁵.

Para orientar a escolha por trabalhos de maior impacto foram priorizados aqueles que já possuíam *peer reviews*. Também tiveram preferência publicações das revistas de maior destaque científico, de acordo com a classificação Qualis da CAPES para

¹<https://scholar.google.com>

²<https://www.sciencedirect.com>

³<https://ieeexplore.ieee.org>

⁴<https://dl.acm.org>

⁵<https://link.springer.com>

a área Engenharias IV (Engenharia Elétrica e Engenharia Biomédica). Ainda, foram apurados os trabalhos que melhor detalhavam suas propostas e continham códigos e experimentos.

Dentre os temas mais específicos como virtualização, NG, FIA e resolução de nomes, foram elencadas 40 referências que subsidiam as informações aqui expostas. A busca pelas referências considerou a proximidade e profundidade dada às palavras chaves: “*Virtualization*”, “*Future Internet Architecture*”, “*testbed*”, “*Future Internet*”, “*NovaGenesis*”, “*Information Centric Networks*”, “*Internet*”, “*Name Resolution*”, “*Publish/Subscribe*”, “*Content Centric Network*” e “*Service Oriented Architecture*”.

A avaliação de novas arquiteturas de Internet passa pela história de como foi estabelecida a Internet atual. Para a contextualização histórica do surgimento da Internet, e de como os seus protocolos evoluíram, foram utilizadas fontes majoritariamente referentes ao desenvolvimento de redes de computadores nos Estados Unidos. Através das quais, entende-se o porquê de estabelecer conectividade entre centros de pesquisa, expandir o acesso a unidades acadêmicas e depois habilitar a comercialização de enlaces de conectividades por empresas privadas. Essas questões foram acompanhadas pelo desenvolvimento de *softwares* e protocolos, que aplicaram funcionalidades aos enlaces e dispositivos físicos conectados, incluindo a arquitetura e conjunto de protocolos TCP/IP. Essas ações foram complementares umas às outras e se estruturaram através das atuações de agências de pesquisa e instituições acadêmicas.

A NG, sendo uma proposta de arquitetura de Internet, tem o entendimento de seus conceitos definidos neste trabalho através de inúmeras abordagens, tais como: pesquisas bibliográficas, artigos, documentos descritivos e referenciais técnicos, além de várias interações com o idealizador da proposta: Antônio Marcos Alberti, orientador deste trabalho. Para fins de reprodutibilidade, os conceitos aqui abordados são referenciados em arquivos com proposições científicas públicas ou por experimentações registradas em eventos científicos, sempre acessíveis pelos canais digitais desses meios.

A avaliação experimental da NG se baseou em definir ambientes e técnicas de experimentação. A busca por ferramentas que poderiam ser utilizadas na avaliação experimental da NG, principalmente para a solução de nomeação e resolução de nomes, concentrou-se em iniciativas *open source*. Isso porque soluções proprietárias não são facilmente personalizáveis, possuem custos de aquisição, em alguns casos elevados e poderiam implicar em restrições à publicação dos resultados, com estabelecimento de *copyright* e/ou *royalties*.

A seguir são detalhados os entendimentos sobre temas importantes relacionados a este trabalho: virtualização, experimentação em redes e na Internet, Arquiteturas de Internet do Futuro e oportunidades de pesquisa. Um breve relato das experimentações da NG é feito ainda neste capítulo para amparar a discussão sobre oportunidades de pesquisa e a relação com outras FIAs, apresentado na Seção 2.4.6. No capítulo seguinte, um detalhamento completo da NG é apresentado, especificando de forma integral suas definições de arquitetura e conceitos relacionados à nomeação e resolução de nome.

2.2 Virtualização

A história da computação e das redes de computadores remonta a um passado recente em que recursos computacionais e interligações de redes eram bem mais escassas. Computadores não tinham suas funções bem definidas ainda, era preciso até estimular o uso doméstico desses equipamentos [46]. Sendo ainda restritos, em sua maior parte, a universidades e centros de pesquisa, o acesso a esses recursos era otimizado ao máximo, sempre na tentativa de oferecer condições de uso ao maior número de pesquisadores, professores e projetos.

Uma forma simples de compartilhar o uso de um computador é dividir o tempo de uso dos seus recursos primários, processador e memória, entre vários usuários [47]. Para isso, cria-se um algoritmo de agendamento de recursos que permite o uso compartilhado. Se a tarefa de um usuário não é totalmente concluída em um ciclo de agendamento/processamento, ela é armazenada em disco parcialmente processada. Quando o agendador retoma o trabalho desse usuário ele continuará de onde parou até terminar a execução da tarefa. Para os usuários isso é transparente, eles conseguem enxergar uma máquina dedicada e disponível para uso, o que varia é apenas o tempo de resposta e carregamento das suas solicitações, se comparada com uma estação não compartilhada.

O aumento da capacidade de processamento dos computadores permitiu que o tempo de resposta e o carregamento das solicitações diminuíssem a cada novo avanço tecnológico. Hoje, esse tempo é imperceptível ao usuário, porém, não foi uma mudança apenas no tempo de execução. Os conceitos de compartilhamento de recursos avançaram para questões mais técnicas como: o completo isolamento entre ambientes de cada usuário, a otimização de comunicação com periféricos e o compartilhamento justo de recursos [48]. Essas condições formam a base para o desenvolvimento da virtualização.

A virtualização é uma forma de executar vários Sistemas Operacionais (SOs), serviços ou aplicações em um mesmo equipamento físico, seja um servidor, computador, *switch*, roteador, *modem*, etc.

Dessa forma, ela se torna uma ferramenta que auxilia a experimentação de inúmeras maneiras, principalmente em relação ao maior aproveitamento energético, à melhor utilização dos recursos, à flexibilidade quanto a oferta de serviços [49] e à acessibilidade de configuração dos equipamentos. Através dessa técnica também é possível aumentar a escala de experimentações, simplificar a estrutura física, flexibilizar e agilizar as modificações de topologias. Um fator que também merece destaque é a possibilidade de integração a *frameworks*, sistemas de gerência e sistemas de monitoramento. Essas vantagens são parâmetros importantes para que ambientes dedicados à experimentação, *testbeds*, façam intenso uso dessas funcionalidades.

Quando se expande a virtualização para vários equipamentos de comunicação pode-se criar o fatiamento de rede, ou em inglês, *slicing* [50]. Essa técnica é a virtualização no âmbito da rede, similar à virtualização de SO. No entanto, objetiva-se a utilização compartilhada de dispositivos de rede e enlaces. Para as interfaces físicas de rede é como se elas lidassem de forma segregada com cada instância virtual da rede. Para a camada de enlace isso é estabelecido definindo diferentes *tags* identificadoras para os fluxos de dados. Se aliada ao conceito de virtualização de SO, além de se ter múltiplos

SOs virtuais, também é possível ter distintos planos de dados e canais de comunicação. Dessa forma, pode-se criar redes totalmente virtuais, com canais de comunicações seguros, isolados entre si e desassociados de equipamentos físicos.

A virtualização pode ser caracterizada pela forma com que o *hardware* é abstraído pelo sistema *hypervisor*, ou gerenciador de máquinas virtuais - *Virtual Machine Manager* (VMM) [51]. Os três tipos mais comuns de virtualização são: virtualização completa, paravirtualização e virtualização em nível de *software*, também conhecida como virtualização baseada em contêiner.

Em uma virtualização completa - *Full Virtualization* - ou emulação, o VMM emula todos os componentes de *hardware* necessários ao sistema visitante (*guest*) e o mesmo não precisa passar por nenhuma alteração de *drivers* ou *softwares*. Esse método é recorrente em implementações que utilizam o VirtualBox para criar um ambiente local de grande utilidade em processos de homologação/avaliação de SOs e aplicações.

Na paravirtualização, o *hypervisor* emula uma máquina com *hardware* parecido ao que está fisicamente instalado. Isso permite que os sistemas *guests* tenham acesso direto aos periféricos e consigam execução otimizada de tarefas mais complexas. Para ter benefício total dessa implementação os sistemas *guests* devem ser compatíveis com *drivers* específicos do *hypervisor*. Essa solução é implementada em cenários que exigem alto desempenho e confiabilidade das VMs. Os principais *hypervisors* que atuam dessa forma, de forma parcial ou integral, são: Xen [52, 53], VMware ESXi [54] e o *Kernel-based Virtual Machine* (KVM) [55, 56].

A virtualização em nível de *software*, ou virtualização baseada em contêiner, utiliza o *kernel* do sistema hospedeiro para criar instâncias isoladas de *userspaces* e dedicá-las aos contêineres. Por compartilhar um único *kernel*, essa técnica é a mais leve, porém, não permite que sistemas com núcleos e arquiteturas diferentes do hospedeiro sejam virtualizados. Algumas limitações quanto à segurança e isolamento entre os contêineres devem ser observadas na operação de sistemas [57]. Exemplos de soluções para esse tipo de virtualização são OpenVZ [48, 58] e Docker [57].

Uma escolha comum para criação de ambiente virtuais é o *hypervisor* KVM. O KVM oferece funcionalidades exclusivas ao implementar uma virtualização em nível de *kernel*, é flexível a ponto de oferecer implementações de paravirtualização [59], garante desempenho em comunicações com periféricos e é uma implementação *open source*. Majoritariamente, por essas características, ele é cotado como uma boa escolha de *hypervisor* para infraestruturas de experimentação. A Figura 2.1 ilustra as relações que o KVM estabelece com o *hardware* e os componentes de *software*. Na camada mais baixa está representado o *hardware* do servidor (*host*), logo acima o *kernel* do SO do *host*. O KVM é apresentado como parte desse *kernel*, sendo esse justamente o diferencial em relação a outros *hypervisores*. A sua implementação direta no *kernel* do sistema hospedeiro permite otimizar a comunicação de entrada/saída das VMs com o *hardware*. Na camada superior é possível observar o SO das máquinas virtuais, cada SO possui seu respectivo *kernel*. As VMs podem ser replicadas para atender a quantidade de nós necessários. Os *softwares* e aplicações são executados dentro do SO de cada uma das VMs.

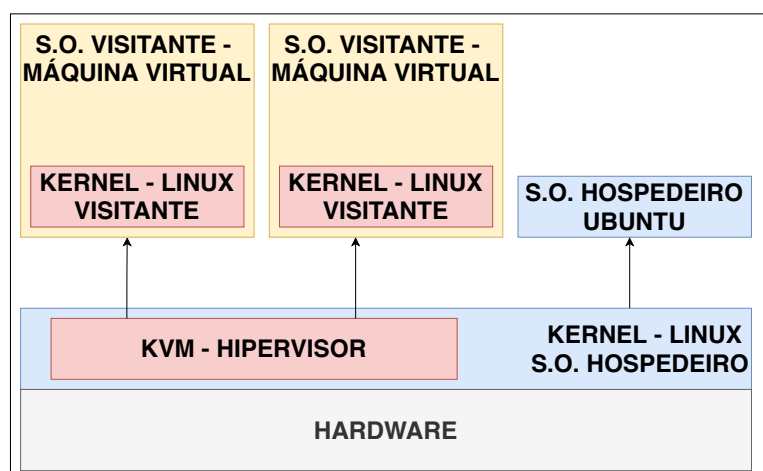


Figura 2.1: Diagrama representativo das relações do KVM com o hardware e sistemas operacionais hospedeiro e visitante.

2.3 Experimentação em Redes e na Internet

A experimentação de protocolos e redes de computadores remonta à época do surgimento da Internet. Alguns autores defendem que nesse momento foram desenvolvidas interconexões para compartilhamento de ambientes de pesquisa, trabalhos científicos e comunicação em rede. Boa parte desses investimentos foram advindos de departamentos dos governos e fundos de pesquisa. Esses ambientes propiciaram o desenvolvimento das primeiras redes de comunicação e, graças ao êxito logrado, foram mantidos e atualizados. Assim, eles seguem por integrar um importante papel no ciclo virtuoso do desenvolvimento tecnológico e científico das redes de comunicação, sendo base para o fomento de experimentações e conseqüente incentivo aos avanços tecnológicos. Esse sucesso retroalimenta e justifica a renovação do investimento inicial, formando um ciclo virtuoso: *Testbed* → Experimentação → Desenvolvimento, como demonstrado na Figura 2.2.

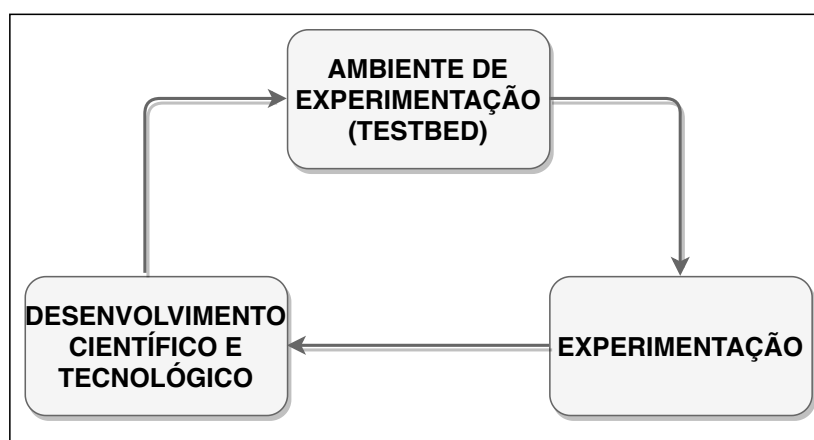


Figura 2.2: Diagrama representativo do ciclo virtuoso do desenvolvimento tecnológico através do investimento em ambientes de experimentação e ciência.

O desenvolvimento e a avaliação de novas arquiteturas de redes e Internet são

possíveis graças aos projetos que criam redes e ambientes de experimentação, denominados *testbeds*. Alguns dos principais, segundo [60], são: *Global Environment for Network Innovations* (GENI), PlanetLab, *Future Internet Research and Experimentation* (FIRE), *Japan Gigabit Network* (JGN) e suas atualizações: *Japan Gigabit Network 2 plus* (JGN2plus) e *Japan Gigabit Network eXtreme* (JGN-X). De forma a complementar essa lista e citar também a plataforma para essa finalidade existente no Brasil, o *Future Internet Brazilian Environment for Experimentation* (FIBRE). Dentre os principais objetivos desses ambientes, é possível destacar que eles buscam agregar vários *campi*, com recursos computacionais e de rede, a um único ambiente de experimentação com gerência centralizada. Essa abordagem permite criar um ambiente de rede controlado e extenso, com condições de oferecer escala para avaliação e experimentação de protocolos e arquiteturas de redes.

Alguns fatores motivam o surgimento e a implementação desses ambientes. A Internet, nos seus primórdios (1969), teve como berço a ARPANET [7], rede de comunicação financiada pelo departamento de defesa americano. Em 1986 a NSFNET se uniu à ARPANET, por meio de uma política de expansão e financiamento de instituições de ensino, de forma que em 1995 os serviços comerciais de intercomunicação já tinham se desenvolvido a ponto de serem independentes e atender as demandas de conectividade da sociedade e dos pesquisadores. Chegando ao ponto da NSFNET poder ser descomissionada.

Os *testbeds* atuais buscam desempenhar o papel que a ARPANET desempenhou no desenvolvimento da Internet. Oferecem escalabilidade e flexibilidade para a comunidade científica avaliar conceitos, novas formas de comunicação, arquiteturas de redes e protocolos. A demanda por esses ambientes é recorrente em projetos de Future Internet (FI), IoT e Redes Definidas por Software - *Software-Defined Networking* (SDN). A oferta desses ambientes de experimentação incentivam a evolução de conceitos e protótipos.

Dentre as características fundamentais dos *testbeds* está a oferta de ambientes de teste e avaliação de desempenho escalonáveis, abertos e flexíveis. Destaque deve ser dado à flexibilidade, que é fator importante quando se precisa avaliar propostas de arquiteturas que podem não se encaixar a equipamentos e tecnologias correntes. A disponibilidade dessas condições permite aos pesquisadores entenderem a dinâmica de novas arquiteturas, avaliar o seu funcionamento prático, escalonar implementações a múltiplos equipamentos e em diversas topologias, avaliar o desempenho de suas teses. Normalmente o acesso a essas plataformas é aberto bastando apenas a identificação do pesquisador e indicação de que o uso será para fins científicos/acadêmicos.

A seguir serão apresentados alguns *testbeds* [61]: PlanetLab, GENI, FIRE e FIBRE que possuem relevância para este trabalho. O GENI e o FIBRE foram plataformas nas quais foram realizadas experimentações que orientaram o desenvolvimento do ambiente virtual local deste estudo. O PlanetLab é um dos precursores do seu tipo, sendo o GENI muito baseado em suas ideias. O FIRE se apresenta como um grande agregador de ambientes de experimentação e se perpetua por diversas renovações de contratos e mudanças de tecnologias. Então, para esse destaque são consideradas plataformas em que a NG foi executada, ambientes que possuem características desejáveis para avaliações de uma FIA e também aqueles que hospedaram alguma das FIAs que serão

citadas nas seções seguintes.

2.3.1 PlanetLab

O PlanetLab [62, 63] é uma iniciativa que remete ao ano de 2003 [64] e ao início de uma plataforma global de pesquisa voltada para o desenvolvimento de serviços de rede, sendo um celeiro [65] para hospedar avanços em soluções de princípios de arquitetura de rede, virtualização, *Content Delivery Network* (CDN), serviços de nomeação, *Publish/Subscribe* (PUB/SUB), localização, *streaming*, medições, armazenamento distribuído, mapeamento, *peer-to-peer*, distribuição de tabelas *hash - Distributed Hash Table* (DHT) [66] e *query processing*. A idealização do *testbed* teve objetivos de longo e curto prazos, chegou a ter mais de setecentas localidades federadas, todas compartilhando ou utilizando algum tipo de recurso para o projeto. Em maio de 2020 o projeto foi encerrado, migrando as funcionalidades e a comunidade ativa para outro ambiente, o *Measurement Lab* (M-Lab) [67].

Enquanto ativa, toda a infraestrutura foi mantida por um consórcio entre departamentos do governo americano: NSF e DARPA, empresas privadas e universidades. Foi uma das primeiras propostas a permitir o fatiamento (*slicing*) dos recursos de rede, permitindo a utilização dos equipamentos para fins de pesquisa e para o tráfego de dados dos usuários de suas instituições. Esse fatiamento da rede foi possível porque o projeto oferecia recursos de computação e rede de forma virtualizada. No seu início foi utilizado o Linux VServer [48, 63], porém, depois de um tempo ocorreu a migração para contêineres, com a adoção da tecnologia *Linux Containers* (LXC) [68]. O PlanetLab contou com uma longa lista de trabalhos desenvolvidos sob sua infraestrutura, dentre os que mais se conectam a este estudo estão os que se referem a PUB/SUB [69], nomeação [70] e DHT [66].

Encerrando suas atividades em maio de 2020 [71] percebeu-se que os principais desafios foram as responsabilizações sobre conteúdos e serviços executados na plataforma em conjunto com as questões burocráticas relacionadas à operação e manutenção de um provedor de Internet/serviços. O *testbed* ganhou tanta escala que esses fatores começaram atarefar as equipes de pesquisadores, optando-se então pela migração para o M-Lab, um projeto dimensionado para manter as flexibilidades do ambiente de experimentação mas com capacidade de lidar com questões de gestão e administração de um grande provedor de serviços de Internet.

2.3.2 GENI

O *Global Environment for Network Innovations* (GENI) [36, 72] é uma plataforma de experimentação que teve início em 2008. Surgiu para atender demandas clamadas pela sua comunidade, dentre elas a capacidade de escalonar experimentações. O GENI é um projeto financiado pela NSF e herda alguns avanços do PlanetLab, dentre eles o *slicing* de recursos, acrescenta ainda em sua arquitetura soluções de *Network Functions Virtualization* (NFV) [73] e (SDN) [74]. Apesar de ser hospedado em diversas universidades americanas, ele é aberto a pesquisadores do mundo todo, independentemente de filiação acadêmica. Dentre suas propostas estão: avaliar arquiteturas, desempenho de rede, protocolos e serviços de rede em escala.

As facilidades desse *testbed* incluem: instanciação de recursos computacionais distribuídos e expansíveis, controle de *switchs* e do tráfego de dados, execução de protocolos de camadas 2 e 3, monitoramento e aquisição dos dados de experimentação em todo a infraestrutura.

A integração dos recursos computacionais e enlaces de rede é feita através de *frameworks*, em específico o *Open Resource Control Architecture* (ORCA) [75] e o *Orbit Management Framework* (OMF). Essa capacidade de integrar recursos através de *softwares* permite a federação de recursos e experimentos de instituições parceiras, aumentando ainda mais a oferta de computação e conectividade disponíveis aos pesquisadores.

Alguns estudos que foram desenvolvidos através do GENI possuem ligação direta com esse trabalho, é o caso do MF [76] e eXpressive Internet Architecture (XIA) [77]. Os dois projetos executaram, respectivamente, aplicações de entrega de conteúdo e de execução de uma arquitetura de rede *clean slate* através de uma rede programável.

2.3.3 FIRE

O *Future Internet Research and Experimentation* (FIRE) [78] é um projeto europeu que oferece um ecossistema de experimentação para diversas áreas da computação. Teve início no ano de 2007 ao propor a evolução e federação de ambientes de experimentações europeus. Com financiamento da comunidade europeia, ele unificou e expandiu o número de *testbeds*, passando a oferecer um ambiente compartilhado maior e mais otimizado para experimentações.

Um dos seus focos é disponibilizar infraestrutura como serviço para desenvolvimento de protocolos, arquiteturas avançadas de rede e FI, incentivando a evolução da conectividade de serviços, conteúdos, pessoas e coisas. Atualmente possui foco em pesquisas de redes de quinta geração (5G)⁶ e IoT. As tecnologias utilizadas para fazer a integração de todas essas tecnologias e recursos são principalmente, o *framework* de controle OMF. Algumas bases de experimentação são integradas ao OMF através de um *Resource Controller* (RC), que operam através de comandos de instrumentação, no padrão *Standard Commands for Programmable Instruments* (SCPI) [79].

Com objetivo de expandir, agregar novos recursos e oferecer interoperabilidade entre os experimentos, foi estabelecida a Federação do FIRE - *Federation for FIRE* (Fed4FIRE) [80]. Em sua versão atual: *Federation for FIRE Plus* (Fed4FIRE+)⁷ que mantém previsão de funcionamento até setembro de 2021.

Alguns dos trabalhos que possuem forte ligação com esse estudo e foram desenvolvidos no FIRE são: avaliações de serviços de rede, como o DNS [81]; nomeação em rede [82, 83]; virtualização e orquestração de recursos [84] e avaliação de FIAs [85].

2.3.4 FIBRE

O *Future Internet Brazilian Environment for Experimentation* (FIBRE) [86–89] é um ambiente de experimentação resultante de uma parceria Brasil-Europa que foi viabi-

⁶<https://5g-ppp.eu>

⁷<https://www.fed4fire.eu/the-project/>

lizada em uma chamada coordenada de *Information and Communications Technologies* (ICT) ⁸. Entrou em operação em 2011, sendo que sua função primária é a de estabelecer uma infraestrutura de pesquisa focada em experimentação de redes. Ele é estruturado em forma de federação, com isso abrange várias unidades menores, denominadas ilhas. Cada ilha possui um conjunto mínimo de recursos de computação e conectividade, podendo até oferecer experimentos adicionais exclusivos à toda federação. Esse é o caso da ilha da Universidade Federal do Rio Grande do Sul (UFRGS) que oferece a possibilidade de desenvolver experimentos com dispositivos IoT.

Atualmente o FIBRE é mantido pela Rede Nacional de Ensino e Pesquisa (RNP), bem como toda a sua infraestrutura de conexão entre ilhas. Essa rede dedicada é denominada (FIBRENet) [90]. É ela que mantém conectadas as instituições, os ambientes de experimentação e uma central de controle. Em 2018, o FIBRE passou por uma atualização em seu *framework* de controle, agora ele conta com o OMF 6.0 [89] para fazer o gerenciamento e controle de todo o *testbed* [91], mesmo sistema utilizado em alguns dos *testbeds* citados anteriormente.

Alguns projetos de FIAs foram avaliados no FIBRE, dentre eles o Protótipo da *Recursive InterNetwork Architecture* (RINA) - (Proto-RINA) [92] e a NovaGenesis [93]. Esses projetos estão diretamente ligados ao tema deste trabalho e à proposta do *testbed*. A avaliação desses projetos também tem relevância no entendimento das capacidades do ambiente e podem servir de critério para o aprofundamento de avaliações ou adaptações. Um *testbed* que se mostra alvo de experimentação de várias FIAs é um indicativo de que ele é flexível e possui uma infraestrutura validada.

Os *testbeds* estão fortemente ligados a esse trabalho porque são plataformas de experimentação e avaliação de protocolos e arquiteturas de rede. A NG foi avaliada em alguns desses ambientes e isso serviu para o amadurecimento da arquitetura, identificação de novas formas de avaliá-la e o entendimento de limites operacionais que precisavam ser superados. A visão da importância dos *testbeds* e a afinidade com FIAs é tanta que se estabeleceu uma ilha do FIBRE no ambiente do Inatel, permitindo assim a continuidade e avanços da experimentação, sejam relacionadas à NG ou a outros projetos de pesquisa.

2.3.5 Comparativo *Testbeds*

De forma a facilitar o entendimento das diferentes características dos *testbeds* o Quadro 2.1 destaca as principais informações das plataformas citadas. O PlanetLab foi encerrado recentemente, em maio de 2020. Ele teve início em 2002 como ambiente de experimentação em rede, mas evoluiu para o modelo de federação e acabou servindo de exemplo para as iniciativas futuras. O GENI teve início em 2008 e permanece ativo. Foi concebido como uma plataforma de federação para experimentação de diversas tecnologias. Ele possui soluções para integração de laboratórios e equipamentos. O FIRE iniciou em 2007 e, desde então, tem passado por sucessivas renovações de contrato, sendo que no momento visa apoiar projetos de 5G e FIA. O FIBRE teve início em 2011 com foco em experimentação de redes e em SDN. Seu foco de atuação foi

⁸<https://ec.europa.eu/research/participants/portal/desktop/en/opportunities/fp7/calls/fp7-ict-2011-eu-brazil.html>

expandido para receber ações de ensino e disseminação de conhecimento, um caminho que alia ações de divulgação científica e pode fomentar pesquisas na área.

Quadro 2.1: *Comparativo entre testbeds*

<i>Testbed</i>	Ano de início	Status	Localização	Principais propostas
PlanetLab	2002	Encerrado (maio/2020)	Estados Unidos	Virtualização por contêineres
GENI	2008	Ativo	Estados Unidos	Federação de <i>testbeds</i> voltados a múltiplas tecnologias: redes sem fio, protocolos e conceitos de FI.
FIRE	2007	Atualizou para Fed4FIRE+	Europa	Federação de projetos voltados para redes de nova geração: SDN, 5G e FIA
FIBRE	2011	Ativo	Brasil-Europa	SDN, FI, Educação

Fonte: Elaborado pelo autor, 2021.

Os *testbeds* contribuem para que as novas arquiteturas de Internet avaliem os principais problemas da arquitetura atual. Questões como mobilidade, entrega de conteúdos, computação em nuvem e segurança não eram conhecidas nos princípios da Internet. Tais ambientes servem para que soluções de FI e FIAs sejam avaliadas em escala sem impactar a Internet atual. Alguns dos projetos que usam bastante os *testbeds* são: *Named Data Networking* (NDN) [94], *MobilityFirst* [31], *eXpressive Internet Architecture* (XIA) [95], *Architecture and Design for the Future Internet* (4WARD) [96], *AKARI Architecture Design Project* (AKARI) [97]. Alguns desses projetos serão detalhados na próxima subseção.

2.4 Internet do Futuro

Um tópico que gera diversidade de opiniões na comunidade científica é quanto à evolução da Internet: se ela deve se dar de forma incremental, resolvendo demandas pontuais e aproveitando o que está estabelecido; ou se uma nova implementação deve ser idealizada, para atender as atuais necessidades de comunicação sem aproveitar o que já existe, iniciando toda a construção do zero, um modelo que é denominado: *clean slate*.

Alguns autores defendem a evolução das redes valendo-se até de argumentos análogos à Teoria da Evolução [98]. Argumentações evolucionárias defendem que a arquitetura e o estado atual da Internet não devem ser totalmente substituídos, mas aprimorados através de melhores implementações e aplicações. A comparação com o modelo evolutivo de Darwin se dá ao ponto em que a Internet é considerada uma entidade que precisa se adaptar, passar por um processo de mutação ou alteração, para atender novas especificações, mantendo implementações que obtiveram sucesso frente aos desafios que já apareceram.

O modelo oposto ao evolucionário é comumente definido em Inglês pelo termo: *clean slate*. Referindo-se a uma reconstrução de arquitetura a partir de novas definições [28]. Essa abordagem considera que as demandas de conectividade da sociedade mudaram tanto que apenas uma mudança completa seria capaz de atendê-las. Essas principais demandas estão relacionadas a melhores definições de nomeação de rede, identificação, mobilidade, entrega de conteúdos e segurança [31, 94, 95]. Justifica-se que Internet atual não suporta mais adaptações e como não possui soluções nativas para

todas essas novas exigências, deveria ser substituída por outra arquitetura que tenha essas implementações em seu *core*.

A autora [19] defende que modelos de FIA não podem ser definidos de forma tão detalhada para representar soluções a todas as demandas. Com isso questiona-se até a capacidade dos *testbeds* em garantir representabilidade do mundo real. Até mesmo grandes iniciativas como o GENI podem ter dificuldade de representar interações que ocorrem em ambientes de produção, tais como: grande número de usuários, múltiplos provedores de Internet, provedores de serviços distribuídos, fabricantes de equipamentos e suas diferentes interpretações de normas técnicas, governantes e legisladores que visam regulamentar a Internet com pouco embasamento técnico. Diante disso, a autora defende que uma abordagem mais apropriada para observar e estudar a Internet é aumentar o número de pontos de monitoramento, para que se possa entender o comportamento do todo e criar soluções correspondentes às deficiências detectadas.

2.4.1 Orientação a Conteúdos

Independentemente de como a próxima geração da Internet se desenvolverá, alguns desafios que se mostram latentes possuem soluções em avaliação e até implementações de provas de conceito como as apresentadas em [99, 100], pesquisas em redes de computadores que se voltam para soluções que entreguem conteúdos de forma mais eficiente. Esse conceito é definido como Redes Centradas em Informação - *Information Centric Networks* (ICN) [15, 101, 102], e se estabelece alterando a orientação de conectividade da rede, de *hosts* para conteúdos. Tal abordagem é recorrente nas propostas de FIAs.

A arquitetura atual é voltada para *hosts*. Nessa visão de projeto toda comunicação originada em um nó cliente objetiva alcançar um nó servidor, ou vice-versa. Mesmo que o objetivo seja apenas acessar o conteúdo de um *site*, como uma notícia por exemplo, o nó cliente precisa encaminhar sua solicitação para a rede, inserir informações detalhadas do endereço de destino, para então encaminhar uma mensagem que alcance o servidor do portal de notícias desejado, no endereço fornecido. O usuário “comum” não tem ciência dessas operações, mas o acesso ao conteúdo desejado é realizado por meio de uma conexão com o servidor que hospeda o conteúdo e está sempre acessível na Internet. Essa forma de comunicação e orientação de conexões não é eficiente em todas as situações. Algumas mudanças têm sido implementadas e aparecem soluções promissoras, principalmente para atender cenários que envolvem dispositivos IoT [103–105].

Quando os usuários buscam por conteúdos eles não se importam onde ou como esses dados estão armazenados. Eles desejam simplesmente ter acesso a um filme, imagem, música, notícia ou informação. Não importando se para acessar tal conteúdo é preciso trafegar informações por diversos servidores ao redor do mundo ou apenas acessar o próximo salto até os servidores do Provedor de Serviço de Internet - *Internet Service Provider* (ISP) - local.

Um modelo de comunicação que chega para atender essa orientação é a publicação e assinatura de conteúdos - *Publish/Subscribe* (PUB/SUB) [106]. Nesse modelo, o requisitante, que pode ser um usuário, uma aplicação, um dispositivo ou serviço [107], informa detalhadamente aos provedores de conteúdos (*brokers*) quais informações quer

ter acesso. Os *brokers* então enviam aos usuários apenas os conteúdos, ou atualizações dos mesmos, que lhes foram solicitadas, de forma personalizada. Essa forma de comunicação é parecida com o que observamos em algumas redes sociais - Twitter⁹, Facebook¹⁰ e Instagram¹¹ - se não houvesse filtros, inserções de publicidades e manipulações dos *feeds*. Em uma implementação em nível de rede seria possível assinar conteúdos de usuários, assuntos, serviços ou páginas e então receber apenas os conteúdos desejados.

A mudança para a orientação a conteúdos pode eliminar a necessidade de utilização dos endereços IPs como os conhecemos. Não seria mais obrigatório buscar pelo endereço do servidor que hospeda um conteúdo para então ter acesso ao que se deseja. Porém, por outro lado, criaria a necessidade de nomear e acessar tais conteúdos de forma distribuída. Nas subseções seguintes são consolidados conceitos de Nomeação e Resolução de nome e apresentadas FIAs que fazem implementações dos dois principais conceitos apresentados: Orientação a Conteúdos e Resolução de Nomes em redes.

2.4.2 Nomeação e Resolução de Nomes

A possibilidade de buscar por conteúdos e não mais pelos hospedeiros de tais conteúdos é uma mudança de paradigma para a Internet [108]. O serviço de nomeação atual da Internet, o DNS, é uma implementação que foi criada especificamente para resolver nomes de *hosts* e máquinas, seu foco não é indexar conteúdos para os usuários. Por isso, ele não permitiria obter vantagens em alguns dos princípios que as iniciativas ICN trazem, como: nomeação independentes da localização, *cache* de conteúdos em rede e roteamento de rede baseado em nomes, conceitos que buscam melhorar a distribuição e acesso aos dados.

Em algumas implementações práticas a mudança de paradigma já é realidade [109]. A desassociação dos conteúdos de seus provedores e a nomeação, endereçamento e localização de informações, como funções da rede, permitem que um usuário que precise acessar um conteúdo possa requisitá-lo apenas pelo nome. De forma que a rede se encarrega de localizar, rotear e entregar o conteúdo através do melhor caminho, sem necessidade de intervenções do requisitante, do provedor da informação ou de serviços intermediários.

Um exemplo de resolução de nome poderia ser o armazenamento de um conteúdo em rede que é acessado através de uma URL [104], com formato similar a: `/inatel/ictlab/artigos/novagenesis`. Esse caminho poderia remeter a um repositório, de forma que toda a URL representaria o nome do repositório e este seria unicamente definido em uma rede.

Uma rede ICN é naturalmente otimizada para a entrega de conteúdos uma vez que lida com a entrega de informações já nomeadas e identificadas [110]. Graças a isso é facilitada a transmissão de dados, uma vez que toda informação trafegada já é devidamente identificada e sabe-se a origem e o destino desse conteúdo. Em contraponto com a arquitetura atual, a solicitação de um arquivo de vídeo a um provedor

⁹<https://www.twitter.com>

¹⁰<https://www.facebook.com>

¹¹<https://www.instagram.com>

de *streaming* gera várias trocas de mensagens por parte do cliente que são paralelas ao tráfego do conteúdo desejado. São feitas consultas de DNS e redirecionamentos via CDNs, só depois que se concretizam a comunicação de entrega de dados entre o cliente e o servidor.

Diante dessas demandas, desafios de otimização e até mudança de paradigmas, algumas FIAs se estabelecem para avaliar de forma parcial ou integral algumas dessas melhorias. Nas subseções seguintes são apresentadas as FIAs: CURLING, MF e NDN, com objetivo de referenciar implementações que apresentam soluções para alguns dos desafios citados, bem como detalhar as afinidades que possuem com a NG. Apesar de existirem outras FIAs, que também poderiam se encaixar nessas descrições, como RINA [111], XIA, Entity Title Architecture (ETArch) [112] e outras mais, foi percebida uma assimilação mais linear dos princípios da NG ao detalhar, na ordem exposta, os projetos escolhidos. Alia-se a isso o entendimento das metodologias adotadas nas experimentações dos respectivos grupos de pesquisa. Tais justificativas são unificadas e apresentadas na forma de avaliação de afinidade com este trabalho ao final da apresentação de cada FIA.

2.4.3 CURLING

O *Content-Ubiquitous Resolution and deLivery Infrastructure for Next-Generation Services* (CURLING) [113] é uma FIA que estabelece um sistema de controle e entrega de conteúdos baseado em ICN e, assim, propõe otimizar tráfego de conteúdos entre redes. Ele utiliza conceitos de SDN, principalmente o desacoplamento das camadas de dados e controle, com a finalidade de criar uma instância dedicada ao controle de conteúdos. A proposta é lidar com a resolução e entrega de qualquer tipo de conteúdo. Para isso ele gerencia, via *software*, a troca de conteúdos entre diferentes redes. Suas funcionalidades se estendem para além da rede local, de forma que o modelo pode ser utilizado entre diferentes domínios e ainda utilizar a atual tecnologia TCP/IP para efetuar a entrega de conteúdos. Seu desenho é focado em tecnologias evolucionárias, de forma que permite a retrocompatibilidade e coexistência com a arquitetura atual.

A arquitetura do CURLING define um Registrador e Controlador de Conteúdos - *content resolution controller* (CRC) - para a rede local. Quando se tem múltiplos domínios, várias redes ou *autonomous systems* (ASs), é definido que os CRCs de cada rede se comuniquem de forma a permitir a integração e a troca de informações, principalmente sobre os conteúdos e registros existentes em cada rede.

O elemento de conectividade que é responsável por estabelecer rotas entre diferentes domínios é o *Content-aware Router* (CaR). Ele está localizado dentro de cada domínio e tem a função de estabelecer a interseção entre diferentes redes, inclusive estabelecer caminhos fim a fim entre publicadores e assinantes de conteúdos, sempre coordenado pelo CRC.

A comunicação entre tais elementos, os publicadores e assinantes de conteúdos, pode ser entendida através do ciclo de vida de conteúdos no CURLING [34] :

1. **Registro de conteúdo:** Publicador (*Publisher*) registra o nome, a localização e o conteúdo a ser publicado no CRC do seu domínio.
2. **Publicação de conteúdo:** O CRC propaga o conteúdo registrado aos CRCs de

outros domínios, neste momento é possível definir regras de acesso ao conteúdo (Ex.: zonas geográficas que podem acessar tal conteúdo).

3. **Resolução de conteúdo:** É a descoberta de conteúdos por parte dos assinantes, acontece quando o assinante solicita ao CRC do seu domínio o conteúdo desejado. O CRC local então verifica se existem registros para tal objeto, localmente ou em outro domínio. A comunicação que é feita com outros CRCs utiliza um canal de comunicação perene.
4. **Entrega de conteúdo e configuração do plano de dados:** A entrega do conteúdo ocorre após o CRC localizar o conteúdo solicitado pelo assinante. Para efetiva entrega do conteúdo é configurada uma rota através dos CaRs, que recebem instruções específicas dos CRCs para que seja configurada uma rota que interligue o publicador e o assinante. Como premissas para estabelecimento dessa rota os CRCs podem avaliar o desempenho atual da rede, restrições e permissões de acessos ao conteúdo.

Para execução de todo esse ciclo, três fluxos principais de comunicação podem ser destacados: (I) A comunicação entre o plano de controle e de dados estabelecida entre o CRC e CaR; (II) A comunicação para descoberta e publicação de conteúdos, iniciada pelo Publicador ou Assinante; (III) O tráfego dos dados que se dá pela criação de uma rota fim a fim entre o Publicador e o Assinante, mas com fluxo através dos CaRs.

A Figura 2.3 ilustra os elementos de um topologia básica do CURLING e esses três fluxos de comunicação. Considerando que o Publicador se encontra no ISP A e o Assinante no ISP B, a comunicação entre o plano de controle e dados é perene, e é demonstrada pela ligações entre os CRCs e CaRs. A solicitação e descoberta de conteúdos acontecem no momento em que o assinante busca por um conteúdo. Ela é representada pelo fluxo de comunicação que se inicia no Assinante, segue para o CRC do ISP B, é recebida pelo CRC do ISP A e alcança o Publicador passando pelo CaR mais próximo. O tráfego de conteúdos é representado em destaque pelo fluxo de dados que segue do Publicador, no ISP A, ao Assinante, no ISP B, sempre passando pelos CaRs que existem entre os dois domínios.

A arquitetura do CURLING se mostra peculiar por definir um plano de controle exclusivo para conteúdos e permitir a troca de estados e dados entre redes distintas. A abordagem da rede como meio para entrega de conteúdos é uma evolução [109] e o CURLING aproveita a sua retrocompatibilidade com as redes atuais para desenvolver sua proposta.

Avaliações Práticas

O desenvolvimento de protótipos e estudo dos conceitos do CURLING é feito por simulações, análises de modelos e *dataset* de topologias de ASs [114]. Em uma dessas simulações ele é comparado com a *Data-Oriented Network Architecture* (DONA) [115] [116], uma outra FIA também com foco em ICN. Esse comparativo tomou como base algumas características comuns das arquiteturas, entre elas: sistema de resolução de nome entre domínios baseado em roteamento por nome, semântica livre para nomeação e baixa latência em roteamento e encaminhamento [116]. Como resultado os autores identificaram viabilidade do CURLING de atuar na resolução de nomes em um AS *Tier*

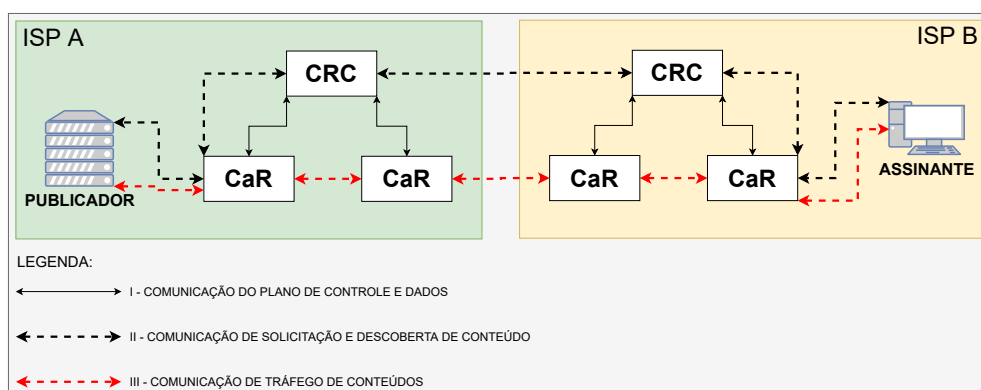


Figura 2.3: Diagrama representativo das comunicações e topologia do CURLING. Fluxos de comunicação I, II e III.

1.

Em outra avaliação, também através dados de provedores de Internet *Tier 1*, os resultados publicados demonstraram viabilidade de operação quando comparado com o serviço de DNS atual da Internet, infraestrutura que sustenta o serviço e os tempos de respostas [113] [117]. Essa avaliação reforça a tese de que a arquitetura atual, voltada para *hosts*, possui alternativas. Os autores deixam ponderações quanto à simulação, porém os dados indicam consistência na resolução de nomes e conteúdos utilizando o CURLING.

Avaliação de Afinidade

O CURLING possui relevância para esse estudo porque consegue demonstrar a viabilidade de uma FIA evolucionária que funciona em paralelo com a arquitetura atual. Suas avaliações de uma rede ICN possuem forte ligação com o tema que se estuda, bem como suas comparações com o DNS.

Quanto à sua proposta de resolução de nomes, observa-se que é possível, através de um controlador de domínio que atua orientando os elementos do plano de dados, trafegar conteúdos de forma otimizada [116]. Além do mais, a sua implementação via *software* se mostra vantajosa, uma vez que é agnóstica ao *hardware*, um possível limitador para futuras arquiteturas de rede.

2.4.4 MobilityFirst

O projeto MobilityFirst (MF) [118, 119] é uma FIA que tem como princípios a mobilidade e a confiabilidade. Sua construção segue uma proposta *clean slate* e é projetada para prover soluções de mobilidade com características adicionais de resiliência e segurança, imperceptíveis aos seus usuários. A motivação para essa construção *clean slate* é porque implementações atuais de mobilidade não oferecem uma transição sem interrupções de conectividade aos usuários. Isso se torna relevante quando se tem um aumento e predominância no número de dispositivos móveis na rede. As fragilidades do IP quanto à mobilidade [120] deixam lacunas para implementações terciárias, além do fato que a sobreposição dos campos de endereçamento e localização do IP não

contribuem para o desenvolvimento de soluções autênticas de mobilidade.

A arquitetura do MF propõe aplicações de mobilidade que separam os nomes/identificadores (ID), dos endereços de rede/localizadores (LOC). Em seguida, faz-se a associação de ambos de forma segura e verificável, através de uma estrutura de nomeação global denominada *Global Name Service* (GNS). Toda essa estrutura é construída por serviços escalonáveis, com abrangência global e distribuídos.

A nomeação e a localização são especificadas de formas distintas. Para atender seus requisitos o MF estabelece uma nomeação que passa a ser definida pelo identificador, *Globally Unique Identifier* (GUID), e o localizador, *Network Address* (NA). O GUID é genérico e pode ser atribuído a uma variedade de *principals*, como: interfaces, serviços, usuários, dispositivos, conteúdos ou outros GUIDs. A adoção de nomeação pelo GUID é autocertificável e estabelecida utilizando chaves públicas e funções *hash*. A principal vantagem dessa implementação é a garantia de que um GUID pode certificar qualquer outro GUID, sem precisar de consultar outros serviços, diferentemente do que se tem na Internet atual, em que para uma simples consulta de domínios é necessário acessar serviços externos como o DNS. O elemento responsável pela localização e endereçamento de rede é o NA. Entende-se como rede todo conjunto de dispositivos que se interconectam e encaminham dados de/para outro conjunto de dispositivos. Assim como no GUID, o NA é autocertificável e implementa a identificação de redes locais, domínios ou ASs. Dessa forma, obtém-se a clara separação do ID e LOC através do GUID e do NA.

O serviço que permite a certificação de nomes, resolução de nomes e a comunicação fim a fim no MF é o GNS. Em uma de suas funções ele é responsável por certificar e resolver qualquer nome da arquitetura para a semântica humana e *vice-versa*. Os nomes podem ser os GUIDs, sendo traduzidos para um formato legível, ou uma solicitação de tradução de nome legível para um GUID. Uma outra função do GNS é manter um mapeamento de todos os GUIDs e seus respectivos NAs, dessa forma, uma solicitação de acesso a um GUID não precisa obrigatoriamente conter o NA. A arquitetura é capaz de interpretar que o GNS deverá ser consultado sobre a localização do identificador que se quer alcançar. Essa facilidade deriva do fato de que a localização atualizada do GUID fica mantida atualizada e disponível no GNS. O que garante a mobilidade na rede, mesmo que não se conheça a localização do *host*/conteúdo/informação que se busca a MF possui mecanismos para alcançá-la.

Em uma comparação com o DNS, ao ser consultado, o GNS consegue certificar e localizar GUIDs na rede, porém, em vez de retornar um endereço IP será retornada a tupla [GUID, NA]. Essa tupla é unicamente identificada e localizada, de forma que contém informações suficientes para alcançar o *principal* desejado em seu destino.

Avaliações Práticas

O MF implementa técnicas para avaliação de seus conceitos de diversas formas, dentre elas [118]: análises teóricas, simulações, medições práticas, experimentações em testbeds e em ambientes em nuvem. Cada modalidade permite uma escala e tipo de avaliação, os modelos teóricos podem sugerir bom desempenho em certas situações, mas precisam ser avaliados na prática. Para fazer as avaliações práticas são utilizados

dois *testbeds* [119], o GENI e o *Open access Research testBed for next-generation wireless networks* (ORBIT) [121]. Nas duas plataformas o objetivo é aprofundar o entendimento da arquitetura e avaliar o seu comportamento em ambientes escalonáveis.

Por possuir financiador comum do GENI, o MF tem condições de estabelecer recursos computacionais e *slices* de forma permanente, além de facilitar o acesso a um número maior de recursos e em maior escala. Desta forma, eles expandem as avaliações para sete *sites*, e contam com conectividade em camada 2, requisitos importantes para avaliação de uma FIA, além do acesso a rede sem fio WiMAX. O resultado da experimentação no GENI foi exitoso, principalmente na execução de comunicação em escala para o tráfego de conteúdo de vídeo entre um servidor e clientes conectados via rede sem fio [119].

O ORBIT é um *testbed* que possui como principal recurso uma malha de equipamentos de redes sem fio. Como isso ele permite avaliações de topologias físicas e configuráveis de acordo com o que se demanda. A avaliação do MF no ORBIT teve como foco gerar resultados relacionados a custo computacional, *throughput* de servidores, latência de solicitações e distribuição de carga [119]. Pelos resultados concluíram que os dados práticos se distanciavam das simulações, em específico a latência, em uma ordem de duas vezes maior na avaliação prática.

Todas essas avaliações e experimentações são rotineiras na ciência, uma hora ou outra, as teses que se fundamentam em predições e simulações precisam ser avaliadas em ambientes reais. Vez ou outras as simulações acabam não correspondendo às expectativas, mas, mesmo assim, geram grandes aprendizados e lições para melhorias das arquiteturas em desenvolvimento.

Avaliação de Afinidade

O MF possui relação com esse trabalho, pois se baseia em uma arquitetura *clean slate*. Sua estrutura de nomeação tem por objetivo prover mobilidade à rede através da dissociação dos parâmetros de identificação (ID) e localização (LOC). Sua definição para os IDs (GUID) é flexível a ponto de poder receber diversos *principals* como atributos de identificação, desta forma, ela também se destaca como uma arquitetura ICN, capaz de resolver e entregar nomes e conteúdos. Aliado ao NA eles formam uma tupla [GUID,NA] capaz de identificar e localizar os *principals* da rede de forma inequívoca.

O grupo de pesquisa responsável pela arquitetura teve sucesso em avaliar o MF de diversas formas e em diversos ambientes de experimentação. Mais uma vez essas implementações se interconectam ao tema desse estudo, pois remontam a uma série de passos que uma proposta de arquitetura de Internet precisa passar para se mostrar viável. Dentre as avaliações teóricas, simuladas, práticas e em ambientes reais, destacam-se as avaliações em *testbeds* e em ambientes controlados, que muitas vezes contrapõem as simulações e cálculos teóricos, mas que superam desafios não previstos e incorporam aprendizados para melhoria da arquitetura. A NG segue esses caminhos, pois tem seus protótipos fortemente baseados em modelos de teoria de filas, mas ainda demanda mais avaliações em ambientes escalonáveis e parecidos com a Internet, tais como os *testbeds*.

2.4.5 Named Data Networking

A *Named Data Networking* (NDN) [12], também conhecida como *Content-Centric Networking* (CCNx) [122] até 2013, é uma proposta de ICN que propõe a nomeação dos conteúdos da Internet e não mais dos *hosts* que armazenam esses conteúdos. Para alcançar esse objetivo ela implementa um robusto sistema de nomeação e resolução de nomes que permite o encaminhamento *stateful* de conteúdos, de forma que qualquer nó possa acessá-los apenas pelo nome. No modelo atual da Internet os conteúdos estão atrelados aos endereços IPs dos seus *hosts*. A NDN faz a nomeação de conteúdos de forma verificável e acessível, extinguindo a necessidade de definir o local de acesso, algo que seria o IP do *host*. Atuando dessa forma, a nomeação, que antes era feita para os *hosts*, passa a ser feita para os conteúdos disponíveis na rede. Esse conceito pode ser melhor entendido lembrando que para acessar o conteúdo de um filme de uma plataforma de *streaming* é preciso acessar primeiro os servidores dessa plataforma, para então ter acesso ao conteúdo desejado. Atualmente, as aplicações da Internet já lidam com requisições de conteúdos por nomes (HTTPS, FTP, etc), o que a NDN propõe é levar esse conceito para a camada de rede, estabelecendo um modelo de comunicação, *request-reply* [123], totalmente estabelecido através da nomeação de conteúdos. Isso altera a perspectiva de nomeação da rede, implementa conceitos de ICN [104] e consequentemente alguns benefícios, tais como: maior disponibilidade, segurança e mobilidade, para usuários e conteúdos.

As características comuns da NDN e do TCP/IP são utilização de datagramas e operação em conjunto com qualquer tecnologia das camadas inferiores que trabalhe com essa orientação: fibra óptica, conexões sem fio, par trançado, etc. As principais mudanças propostas pela NDN são:

1. Substituição da camada de endereçamento de *hosts* por um *namespace* definido via aplicação;
2. Utilização dos nomes NDN ao invés de localizadores de dados (endereço IP ou semelhante);
3. Mudança de paradigma de comunicação: o usuário passa a coletar - *fetch* - conteúdos, ao invés de empurrar pacotes com solicitações de respostas aos seus destinatários.

Essas mudanças representam uma alteração na pilha de protocolos do jeito que havia sido descrito para o Modelo Híbrido no Quadro 1.1. Na pilha da NDN, as camadas 1 e 2 se mantêm, a camada 3 é representada pela nova arquitetura de nomeação e as camadas 4 e 5 se unem em uma única camada, responsável pelo transporte e aplicações da rede. O Quadro 2.2 exhibe esse comparativo entre as duas pilhas.

Fazendo uma analogia a consumidores e produtores de conteúdos, a NDN implementa uma comunicação baseada na manifestação de interesse em receber um conteúdo, por parte de um consumidor, e a entrega desse conteúdo, por parte de um produtor. Em nomenclaturas da arquitetura, o primeiro corresponde ao NDN *Interest packet* e o segundo ao NDN *Data packet*. No âmbito dessa seção ambos serão referenciados apenas por *Interest* e *Data*. O *Interest* [124] é um pacote enviado à rede por um *host* que solicita acesso a algum conteúdo. O único parâmetro obrigatório do *Interest* é o nome do conteúdo, não sendo necessária nenhuma informação referente aos remetentes

Quadro 2.2: Representação das camadas de rede do modelo de referência híbrido e da NDN.

Modelo Híbrido		Modelo NDN	
Nível	Denominação	Nível	Denominação
5	Aplicação	4	Aplicação e Transporte
4	Transporte		
3	Rede	3	NDN
2	Enlace de dados	2	Enlace de dados
1	Física	1	Física

Fonte: Adaptado de Jacobson, 2012 [12].

e/ou destinatários. Outros parâmetros opcionais podem ser inseridos de forma a complementar funções avançadas da rede. O pacote *Data* [125] é composto pelo nome do conteúdo, o conteúdo em si, a sequência de *bytes* e uma assinatura criptográfica [126].

Por segurança, todos os conteúdos são assinados pelos produtores no instante de sua criação. Isso permite a consulta da autenticidade dos dados através de uma consulta às chaves públicas. Uma autoridade certificadora - *Certificate Authority* (CA) - é responsável por distribuir chaves privadas e públicas às entidades da rede (i.e. host, aplicação, usuário). A confiabilidade dos certificados utilizados é baseada em âncoras de confiabilidade [123] que são definidas pelas autoridades de cada rede. Dessa forma, não existe mais uma confiança integral em autoridades certificadoras externas, cada rede pode definir quais serão suas referências de confiabilidade. Uma outra vantagem da assinatura dos conteúdos com base na sua data de criação é a possibilidade de identificar divergências entre versões com menor esforço.

O encaminhamento *stateful* da NDN é o que permite o tráfego dos pacotes *Interest* e *Data* pelos nós. Diferentemente do IP, o encaminhamento não é feito por localizadores, mas exclusivamente pelos nomes. Os próprios protocolos de roteamento são aplicações NDN e as atualizações de rotas são pacotes *Data*, seguramente assinados e distribuídos na rede. Os nós que efetuam encaminhamento da rede possuem a seguinte estrutura mínima: *Content Store* (CS), responsável pelo armazenamento de conteúdos nomeados, podem ser conteúdos próprios ou hospedados (*cache*); *Pending Interest Table* (PIT), tabela que registra o fluxo dos pacotes *Interests* e evita *loops* na rede, identificando e descartando solicitações duplicadas; *Forwarding Information Base* (FIB), responsável pela estratégia de encaminhamento, implementa protocolos e regras de roteamento, define quando encaminhar ou recusar os pacotes *Interests* recebidos e as *interfaces* a serem utilizadas para comunicação.

O processo de encaminhamento e recebimento de informações pode ser separado em dois fluxos: o envio de um pacote *Interest* que busca um conteúdo no próximo nó - fluxo *upstream* - e o retorno de um pacote *Data* ao solicitante do conteúdo - fluxo *downstream*. A Figura 2.4 e os próximos parágrafos exibem e descrevem esses dois fluxos da NDN, de forma que as etapas I a VI, descrevem o caminho percorrido pelo *Interest* e a etapas VII a X descrevem o caminho percorrido pelo pacote *Data*. Na representação, o conteúdo que se busca é nomeado da seguinte forma: */inatel.com/ictlab/novagenesis.pdf*.

A comunicação *upstream* se inicia ao enviar um *Interest* ao nó seguinte, consultando-o se possui o conteúdo desejado (I). Caso o nó possua esse conteúdo, que pode

estar armazenado em seu CS, ele retornará um pacote *Data* (X) com o conteúdo ao solicitante. Caso contrário (II), será feito um registro do *Interest* no PIT e enviado para a FIB (IV) avaliar se a estratégia de encaminhamento permite o repasse do *Interest* para o próximo nó. Caso a PIT e/ou FIB identifiquem inconsistências na solicitação, ela é descartada (III e V). O PIT faz uma verificação em sua tabela para evitar possíveis *loops* e o FIB verifica a estratégia de encaminhamento e rotas para o conteúdo desejado. Se nenhum dos dois recusar a solicitação, o *Interest* é encaminhado para o próximo nó (VI), repetindo o fluxo (I-VI), até que se encontre o conteúdo desejado em algum outro nó.

A comunicação *downstream* se dá quando o conteúdo solicitado é encontrado em um nó (VII ou X), momento em que um pacote *Data* é criado. No caso do conteúdo não ser encontrado no primeiro nó consultado, o retorno desse pacote se dá pelo caminho reverso que o *Interest* percorreu, saltando-se apenas a passagem pelos FIBs. Quando o *Interest* passa pelos PITs dos nós anteriores é registrado o nome do conteúdo procurado e qual interface recebeu essa comunicação, assim, em um retorno nó a nó, é possível fazer o caminho reverso associando o nome do conteúdo especificado no *Data* e por quais interfaces o *Interest* que o buscava passou. Assim, o retorno se dá passando-se pelo PIT (VIII), CS (IX) e chegando ao nó solicitante (X). Na passagem de um *Data* pelo CS (IX) o nó pode decidir se mantém uma cópia local do conteúdo, assim poderá atuar como *cache* em futuras solicitações desse conteúdo.

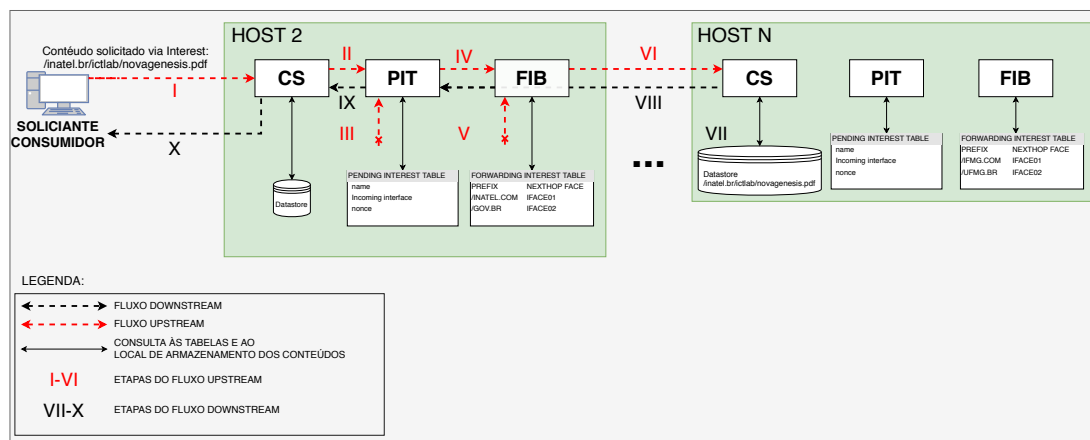


Figura 2.4: Representação dos fluxos de comunicação upstream e downstream do NDN.

A NDN possui estrutura flexível de definição de *Principals* e números de interfaces. O exemplo considerou uma comunicação entre *hosts* através de um caminho único, porém a arquitetura suporta múltiplos caminhos, múltiplos *hosts* e indefinidos *principals*. As interfaces também podem representar canais de comunicação intraprocessos.

Avaliações Práticas

A NDN possui uma estrutura dedicada de experimentação financiada pela NSF. Foi criado um *testbed*¹² [127] exclusivo para o desenvolvimento e avaliação dos seus conceitos. Também possuem maturidade de desenvolvimento para integrar serviços de nuvem, como o Amazon AWS, em suas demonstrações [128].

¹²<https://named-data.net/ndn-testbed/>

A NDN possui códigos bem difundidos, suas avaliações e experimentações não são mais realizadas apenas pelo grupo de pesquisa inicial. Para validação dos protótipos as experimentações são realizadas, principalmente, no GENI [36]. Com o surgimento de aplicações para a arquitetura, outros autores começaram a avaliá-la, desenvolvendo simulações de rede em ambientes privados, como o QualNet¹³ [129], simuladores próprios [32] e até utilizando a arquitetura para validar um *testbed* virtual, o *Future Internet Testbed with Security* (FITS) [130].

No trabalho [32], o autor apresenta uma solução para utilizar as implementações de mobilidade da NDN em benefício dos produtores de conteúdos, é feita uma avaliação da replicação de conteúdos em múltiplos nós e do aumento da disponibilidade dos dados na rede. Em um primeiro protótipo, o foco foi prover mobilidade, não necessariamente melhorar o desempenho da rede, exclusivamente, para os produtores de conteúdos. Como consequência, não se obtiveram bons resultados de mobilidade. Ao alterar a abordagem e utilizar a estrutura de *cache*, roteamento e replicação de conteúdos, da NDN, observou-se melhoria da mobilidade de rede para os produtores de conteúdos e maior disponibilidade dos dados (conteúdos).

No trabalho [131], o autor avalia um serviço sempre *online* de *lookup* para a NDN. Tal serviço serve para prover conteúdos estratégicos, certificados criptográficos e protocolos de roteamento. É idealizado então o DNS para a NDN, ou NDN DNS (NDNS), uma aplicação que permite estabelecer um serviço na rede que permanece sempre disponível e provê acesso a conteúdos críticos, como: certificados criptográficos, segmentação dos prefixos de nomes das rotas e armazenamento persistente para os usuários. Outras vantagens detectadas são relacionadas aos conceitos de ICN, em que, por lidar com os nomes dos conteúdos no roteamento, consegue-se maior eficiência na transmissão da dados *multicast*, *cache* intrarede e seleção automática dos servidores de conteúdos. O autor destaca que a construção intrínseca de assinaturas e verificação de pacotes da NDN se compara com a funcionalidade do *Domain Name System Security Extensions* (DNSSEC), porém, como em qualquer sistema de segurança em desenvolvimento, sugere melhor avaliação do modelo a fim de verificar formas de exigir e automatizar os esquemas de segurança da arquitetura.

Dentre alguns trabalhos que utilizam o *testbed* da NDN para avaliar conceitos da arquitetura estão:

- NDN RTC [132]: Nesse estudo os autores apresentam resultados de experimentação para avaliação de um serviço de videoconferência utilizando a arquitetura NDN. A disponibilidade de um ambiente distribuído em larga escala permitiu que se avaliassem, de forma mais detalhada, condições de uso, tráfego entre redes, distribuição e armazenamento de conteúdos.
- *Violating Consumer Anonymity: Geo-Locating Nodes in Named Data Networking* [133]: os autores fazem uma avaliação se a nomeação de rede apresenta melhores resultados em relação a segurança do que o endereçamento IP. Eles utilizam o *testbed* e um serviço de nuvem para validar simulações. Ao fim, percebem que a localização dos nós e redes é algo sensível e pouco protegido por arquiteturas que se propõem a trabalhar com *cache* de rede.

¹³www.scalable-networks.com

- *Experiments with the Emulated NDN Testbed in ONL* [134]: A experimentação no *testbed* global é algo que necessita de considerável suporte operacional. Para evitar novas implementações de testes, diretamente no cenário global, utiliza-se um emulador que adiciona vantagem de permitir uma avaliação da experimentação sem impactos em outros projetos. Dessa forma, diminui-se as possíveis indisponibilidades e operações de suporte por parte dos mantenedores da infraestrutura. Nesse estudo, os desafios operacionais foram pontuados e serviram de base para evoluir a experimentação da NDN.

Avaliação de Afinidade

A NDN tem forte relação com o trabalho que se apresenta, pois possui um sólido desenvolvimento em suas formas de avaliação e experimentação. Ela partiu de modelos teóricos, simulações, desenvolvimento de simuladores, *testbeds* compartilhados, *testbeds* exclusivos, avaliação em ambiente de nuvem e por último a disponibilização do seu código para avaliação externa e desenvolvimento de produtos. Esse fluxo de desenvolvimento é característico de implementações científicas e implementações *open source*, porém, pode não ser o suficiente para alcançar o mercado de forma integral. Mesmo assim o projeto segue com seu desenvolvimento, em busca de estabelecer uma alternativa ao padrão atual de Internet [135].

Observa-se que para alcançar tal maturidade e desenvolvimento, os líderes do projeto sempre documentam as experiências e os pontos de falha, o que contribui com implementações futuras [123].

2.4.6 Comparativo entre FIAs e Oportunidades de Pesquisa

Como abordado nas subseções anteriores, é indissociável o desenvolvimento das propostas de FIAs, os ambientes de experimentação e as avaliações de desempenho. Sejam as propostas incrementais ou as *clean slate*, quando é preciso atingir um nível de escala maior elas fazem uso desses recursos, compartilhados ou próprios. Como foi mencionado, essa característica é marcante na experimentação do NDN [134], em que, mesmo possuindo um *testbed*, o grupo de pesquisa se prepara e certifica que a proposta de experimentação será executada com o menor impacto operacional em um *testbed* de escala global. Essa característica demonstra a preocupação em estabelecer práticas otimizadas em todas modalidades de experimentação. Assim como cada FIA possui seus pontos de destaque, faz-se necessário apresentar um pequeno comparativo do que as diferenciam, bem como considerar lacunas abertas e apresentar quais são potenciais pontos de investigação, em relação à experimentação de arquiteturas de Internet do futuro, que podem continuar gerando melhorias conceituais ou práticas.

O Quadro 2.3 discrimina as FIAs citadas e algumas de suas principais características, tais como tipo de proposta, se é uma construção com bases incrementais, logo evolutiva; ou se é uma proposta *clean slate*, construída totalmente do zero. A orientação da arquitetura, qual é a característica base para os identificadores de rede: conteúdos, nomes, ou localizadores. A tecnologia utilizada, quais os conceitos ou linguagens de programação adotados para desenvolvimento dos protótipos. Quais as melhorias alcançadas nos trabalhos citados e por último, os tipos de experimentações utilizadas. Por ela observa-

se que o CURLING é uma proposta de evolução incremental, orientada a conteúdos, desenvolvida com implementações de SDN em que as principais melhorias alcançadas são em relação a publicação e assinatura de conteúdos, obtidas por experimentações em simulações e análises de modelos. O MF se apresenta como uma proposta *clean slate*, com orientação a nomes de localizadores de rede, desenvolvido, principalmente, em C/C++ e Java, com principal foco de melhoria em mobilidade e cujas principais experimentações foram baseadas em simulações e utilização do *testbed* GENI. A NDN também é uma proposta *clean slate* cuja orientação é voltada para a troca de conteúdos em rede, sua implementação é baseada em implementações em linguagem C, Java e Javascript, as principais melhorias obtidas pela proposta estão relacionadas a um maior eficiência na resolução de conteúdos e *cache* de rede, sendo que tais resultados foram obtidos principalmente de experimentações nos *testbeds* NDN e GENI.

Oportunidades de Pesquisa

Tendo em vista as formas de experimentação consolidadas das FIAs apresentadas, busca-se avaliar uma arquitetura alternativa que seja *clean slate*, possua orientação diversa, incluindo conteúdos, possua também implementação promissora para atender requisitos de mobilidade, publicação, assinatura e resolução de nomes e conteúdos, algo que não é atendido de forma integral pelas FIAs citadas. A NG é uma arquitetura que possui potencial para atender tais requisitos. De forma a complementar as oportunidades citadas, também existe espaço para pesquisa dos métodos, ambientes e cenários em que se avaliam a NG. Pois sendo uma arquitetura em desenvolvimento, múltiplas frentes de avaliação podem contribuir para a sua evolução e avaliação dos conceitos que são propostos.

A avaliação de uma arquitetura de Internet demanda um ambiente de experimentação flexível, controlado e escalonável. Para atender esses requisitos foram idealizadas algumas ações que objetivaram avaliar a NG em diferentes ambientes, cenários e topologias de serviços, sendo elas:

- Avaliação da execução da NG em ambiente físico controlado.
- Avaliação da execução da NG em ambiente virtual controlado.
- Definição de cenários e topologias de avaliação para a NG.
- Automação de testes em escala.
- Avaliação do serviço de resolução de nomes da arquitetura.
- Avaliação do serviço de resolução de nomes em diferentes topologias.
- Avaliação dos ambientes de experimentação utilizados.

Quadro 2.3: Comparativo FIAs

FIA/Característica	CURLING	MobilityFirst	NDN
Tipo de proposta	Incremental	<i>Clean slate</i>	<i>Clean slate</i>
Orientação	Conteúdo	Nomes e localizadores	Conteúdo
Tecnologia	SDN	Implementações em C/C++ e Java	Implementações em C, Java e Javascript
Melhorias	Publicação e assinatura de conteúdos	Mobilidade	Resolução de conteúdos e <i>cache</i>
Experimentações utilizadas	Simulações e análises de modelos	Simulações e <i>testbed</i> GENI	<i>Testbed</i> NDN e <i>testbed</i> GENI

Fonte: Elaborado pelo autor, 2021.

2.5 Entendimentos do Capítulo

Neste capítulo são expostos os resultados da pesquisa bibliográfica relacionada ao tema em estudo. É feito um detalhamento de como os conceitos de virtualização, experimentação de redes e Internet do futuro se interconectam e se tornam essenciais ao desenvolvimento de novas arquiteturas. Em destaque, são relacionados ambientes de experimentação que fomentam a experimentações de redes e algumas das arquiteturas em desenvolvimento que possuem proximidade com os conceitos da NovaGenesis ou com as metodologias aqui utilizadas. De forma sintética, são comparadas as principais características dos ambientes de experimentação e das arquiteturas citadas. Identificadas algumas lacunas e oportunidades de evolução no desenvolvimento de FIAs, a NG é elencada como arquitetura com potencial para ser avaliada e atender tais requisitos.

Capítulo 3

NovaGenesis

Este capítulo faz uma detalhada apresentação da NovaGenesis. Para entendimento dos seus conceitos é feito um breve histórico do seu início, são apresentados os Pilares Conceituais e os Princípios que norteiam a sua implementação. O padrão de comunicação, publica/assina - Publish/Subscribe (PUB/SUB), que será base para toda a avaliação da arquitetura, também é descrito, bem como as formas de comunicação da arquitetura e os seus principais serviços: PGCS, NRNCS e NBSimpletestAPP. Para exemplificar o funcionamento da arquitetura e os conceitos que se quer priorizar, são detalhados fluxos de publicação e assinatura de NBs, exemplificando como tais comunicações ocorrem entre os nós e entre os processos da NG.

3.1 O Início

O projeto NovaGenesis [136] foi iniciado em 2008 [137] como um estudo de quatro anos (2008-2011) sobre o estado da arte da Internet e paradigmas emergentes de redes de comunicação. Em 2012 [138], foi concluída uma análise baseada nos conceitos de requisitos para uma proposta de *Future Internet*, suas tecnologias estruturantes e desafios. Com base no conhecimento adquirido, foi selecionado um conjunto de ingredientes promissores para resolver tais desafios e atender requisitos da atualidade. Além disso, trabalhou-se na busca de sinergia entre os ingredientes adotados na NG [138]. Em 2012 [139] e 2013 [137], as primeiras proposições e implementações iniciais foram feitas.

Os resultados deste estudo serviram de base para o projeto da arquitetura, definindo um modelo em que o processamento e a troca de informações são tratados como um serviço em que todas as entidades são nomeadas e as ligações entre nomes são armazenadas de forma distribuída, visando atender principalmente aos requisitos de escalabilidade. A ideia central da NG pode ser resumida na seguinte sentença: serviços (incluindo as implementações dos protocolos) que se organizam utilizando nomes, *name bindings* (NBs) e contratos, visando atender objetivos e políticas semanticamente ricas.

Essa breve introdução objetiva apenas posicionar o entendimento temporal quanto ao desenvolvimento da arquitetura. Nas seções seguintes são detalhados os pilares conceituais e norteadores da arquitetura, as fundamentações e princípios de implemen-

tação, com apresentação de detalhes dos serviços e como eles se interagem. Essas informações são essenciais ao desenvolvimento deste trabalho, em destaque o serviço de resolução de nomes desenvolvido pelo Prof. Antônio M. Alberti, que é um dos pontos principais da proposta a ser apresentada para avaliação e experimentação da arquitetura.

3.2 Pilares Conceituais

A implementação prática da NG foi iniciada em 2012, por Antônio Marcos Alberti, idealizador da proposta. Para a codificação de um protótipo em linguagem de programação C/C++ para o GNU-Linux foram estabelecidos alguns conceitos estruturantes, que em uma alusão figurativa podem ser denominados Pilares, no sentido de serem conceitos bases dessa arquitetura. No início de 2013, foram estabelecidos os primeiros testes práticos. Nos anos seguintes algumas experimentações foram publicadas provando a viabilidade da implementação desses conceitos [40].

Sinteticamente, são quatro conceitos que motivam o desenvolvimento da proposta [137]. Eles se estabeleceram através do desenvolvimento de uma ampla pesquisa bibliográfica e avaliação de outras propostas de *Future Internet Architecture*. Em suas definições é utilizado o termo “entidade”, que serve para generalizar a aplicação da NG a equipamentos, serviços, aplicações, pessoas e/ou qualquer outro elemento que se deseje inserir na arquitetura. Os conceitos que permeiam os quatro pilares estão relacionados à nomeação e identificação de entidades; ciclo de vida com etapas bem definidas de criação; gerência e encerramento das entidades; representação de entidades do mundo físico no mundo digital, se apropriando de conceitos de virtualização e implementações via *software*; e por último, a definição de que a arquitetura e todas as suas entidades devem se comportar como um serviço e que possam compor estruturas maiores (macro) de maneira fractal [140]. A seguir são apresentados, em detalhes, cada um desses Pilares, começando pela Linguagem, em seguida o Ciclo de Vida, depois a Virtualização e por fim a Orientação a Serviços.

3.2.1 1º Pilar: Linguagem

O primeiro pilar recebe destaque dada a sua importância no restante da arquitetura. Ele é base até para outros pilares, sendo que a priorização pela linguagem acontece porque as definições semânticas e de sintaxe podem estruturar e organizar os elementos da arquitetura. Nessa esfera se definem como essenciais a nomeação das entidades e as ligações que existem entre elas. Características como localização e identificação também são possíveis através de uma linguagem bem especificada. A NG pode lidar com nomes e operadores semânticos, isso lhe dá condições de representar relações entre entidades que estejam compreendidas umas nas outras ou que sejam equivalentes. Em uma aplicação prática, uma rede passa a ser capaz de identificar quais dispositivos estão compreendidos em seu domínio, ou ainda se existem serviços equivalentes a uma função de rede que se busca.

Em uma abordagem explicativa é possível observar que, se bem definida, a linguagem, nomeação e semântica, há fácil associação e identificação de entidades em um

dado escopo. Ao se apresentar a frase “O celular do Victor está no bolso esquerdo”, os seguintes entendimentos podem ser inferidos: 1) A identificação de entidades; 2) A relação entre elas; 3) Localização parcial de uma das entidades.

Esses entendimentos são possíveis graças aos conceitos prévios da língua portuguesa e experiências do leitor. As entidades relacionadas são: “celular” e “Victor”; entende-se que “celular” é um objeto e “Victor” uma pessoa. A relação entre as entidades também está definida na frase ao dizer “O celular do Victor...”; por isso, entende-se que o objeto “celular” pertence ao “Victor”. Uma outra observação sobre a localização do celular também pode ser feita pelo trecho “...no bolso esquerdo”. Assim, a indicação de localização para o contexto pode ser suficiente, mas para a localização global, tanto do celular quanto do Victor, dependerá de informações adicionais atreladas a essas entidades. Mesmo assim, a frase apresentada, mesmo que com poucas palavras, identifica um conjunto de entidades, estabelece uma relação que delimita uma posse entre elas e permite localizar uma entidade em relação à outra. No entanto, pode ocorrer um problema caso não se saiba previamente que “celular” é um objeto e “Victor” um nome próprio que identifica o sujeito na frase. Assim, se esse conhecimento prévio não existir, ou não puder ser alcançado, a frase perde algumas de suas interpretações.

A NG aplica esses conceitos em sua arquitetura, para isso ela também define em sua linguagem mecanismos de identificação inequívoca, perene e autocertificável. Essas características lhe possibilitam utilizar tecnologias e sistemas implementando a auto-descoberta de recursos, oferta de serviços e descoberta de relacionamentos de forma dinâmica.

Resguardadas as proporções, quando são feitas as codificações e as implementações a linguagem perde um pouco a tangibilidade da linguagem natural exemplificada. É nesse momento que são tratadas as múltiplas relações que as entidades podem ter, para isso são utilizados algoritmos que agregam segurança e padronização na estrutura de nomes. Esse processamento por algoritmos, especificamente baseado em funções *hash*, é o responsável por garantir a certificação dos nomes, aumentar a segurança e confiabilidade da arquitetura. A função de *hash* utilizada é a MurmurHash3 [141] [142].

3.2.2 2º Pilar: Ciclo de Vida

O segundo pilar define ciclos de vida para entidades da arquitetura. O estabelecimento de um ciclo de vida garante uniformidade e funcionalidades para a arquitetura, definindo regras claras para a criação, operação e encerramento de entidades e serviços. É uma definição importante pois é a forma que a NG estabelece para as entidades regras voltadas para a integração, flexibilização, compartilhamento e otimização dos serviços e recursos da rede. Como exemplo, é possível citar a utilização de um serviço de armazenamento compartilhado em rede. Se não for definido um ciclo de vida para a instanciação, utilização e encerramento do serviço, ele pode ser alocado de forma ineficiente a entidades que não façam uso otimizado dessa função, comprometendo um recurso que poderia ter uma utilização muito mais eficiente por outras entidades. Para evitar essa situação toda entidade implementa um ciclo de vida que possui regras bem definidas de exposição, descoberta, oferta, contratação, monitoramento, gerenciamento e encerramento de serviços ou entidades.

- (i) Exposição de características e capacidades: É o momento em que uma entidade é instanciada e apresenta suas funcionalidades ao expor quais recursos pode compartilhar com a rede;
- (ii) Descoberta de serviços: Após feita a apresentação, a entidade busca por outros membros da rede, bem como os recursos públicos que estão disponíveis;
- (iii) Oferta de serviços: É a disponibilização de serviços em rede, pode ser uma função de rede, aplicação ou recurso que uma entidade tenha condições de ofertar a outras entidades;
- (iv) Contratação de serviços: Estabelecidos os contatos e descobertos os recursos, as entidades estabelecem acordos de utilização de um recurso da rede, definindo parâmetros como qualidade do serviço ofertado, duração da interação entre tais entidades e quais recursos serão trocados entre elas;
- (v) Monitoramento de qualidade: A continuidade da utilização dos recursos pode ser condicionada ao tempo ou parâmetros de qualidade do serviço. Desta forma, o ciclo de vida indica que os acordos fechados entre as entidades devem ser constantemente monitorados a fim de serem mantidos;
- (vi) Avaliação de reputação: Esse conceito está relacionado à gerência e permanência do nó na rede. Se é identificado que um nó representa ameaça aos demais, ele pode ser excluído das contratações de serviços e da própria rede;
- (vii) Liberação de recursos: Também associada ao encerramento de um contrato ou remoção de uma entidade, pode ser visto como a finalização de um processo.

3.2.3 3º Pilar: Virtualização e Representação

Neste pilar, o conceito de virtualização e representação de entidades define como lidar com recursos do mundo físico e virtual. O qual tem como objetivo estabelecer condições de levar as entidades do mundo físico para dentro da arquitetura. Isso é feito através de uma representação direta da entidade física em uma entidade virtual, com parâmetros e especificações próximas às reais, conceito também conhecido como Gêmeo Digital - *Digital Twin* [143]. Essa abstração transforma em *software* uma entidade do mundo físico, para que possa, então, integrar a arquitetura de rede e informação. Isso representa uma mudança de paradigma quanto à forma como se lida com roteadores, *switches*, *access points* e qualquer outro dispositivo que atue na rede. Essa lista também se estende a equipamentos fora do núcleo da rede, responsáveis por aplicações, como: sensores e atuadores.

As aplicações que advêm desse conceito são inúmeras. A possibilidade de representar qualquer elemento físico como *software* e dentro da arquitetura complementa a abordagem do ciclo de vida. Com isso, a arquitetura passa a permitir que entidades do mundo físico se insiram à rede, e passem a oferecer e compartilhar os seus recursos de forma contratual. Entidades virtuais também podem se beneficiar dessa representação (e.g., um protocolo de roteamento inteligente pode estabelecer contratos com roteadores que lhe concedam maior prioridade ou abrangência). Em um avançado estado de implementação, espera-se que este pilar fomente maneiras de habilitar a “economia das coisas” [144].

3.2.4 4º Pilar: Orientação a Serviços

O quarto pilar define que toda a arquitetura deve ser orientada a serviços. Assim, a NG deve ser implementada como se fosse uma grande loja. Todas entidades devem possuir condições de estabelecer relações entre si. Até as entidades físicas, quando representadas de forma virtual, devem ter condições de estabelecer relações através de seus representantes. Dessa forma, permite-se que os protocolos e processamentos de informações interajam com as entidades de forma contratual. Por exemplo, pode-se estabelecer que o acesso entre dois *hosts* só aconteça se houver a contratação do meio de comunicação entre eles, exemplos de contratos padronizados e a não obrigatoriedade de contratos também podem ser configurados. Essa abordagem pode parecer burocrática, porém permite que se estabeleçam contratos para cada funcionalidade ou recurso que se deseje usar da rede. Essa é uma metodologia denominada de auto-organização *bottom-up* [145], que define que quanto mais segmentadas e básicas forem as ofertas de serviços e recursos, melhores condições as entidades terão para elencar os serviços necessários para realização de suas tarefas, sejam elas simples ou complexas.

A NG se estabelece como uma arquitetura que é orientada a serviços - *Service-Oriented Architecture* (SOA). O pilares anteriores reforçam esse fato, como a representação de entidades físicas via *software*, definição de ciclos de vida e até mesmo a definição de uma linguagem estruturante. Em conjunto, essas características contribuem para a independência de soluções baseadas em *hardware* proprietários, protocolos fechados e autoridades centralizadoras.

3.3 Princípios do Projeto e Implementação

Nas subseções anteriores foram definidos os pilares estruturantes da NG. A seguir, serão elencados onze princípios que contribuem para que as definições conceituais orientem implementações e evoluções da arquitetura, inclusive a sua codificação [146] [147]. Para definir esses conceitos, o projeto original observou outras propostas de FIAs, implementações que ainda não são padronizados na Internet atual e seus alinhamentos com os pilares já citados. Ao se basear em uma ampla pesquisa e avaliação de arquiteturas contemporâneas, o idealizador da proposta estabeleceu as diretrizes de uma arquitetura convergente de informação. Por exemplo, em um de seus princípios define-se que a arquitetura deve generalizar a utilização de seus princípios ao máximo. Dessa forma, os serviços se dispõem de forma fractal, ou autosimilar, na sua implementação e codificação. Em outras palavras, essa característica é fundamental, pois permite reutilizar códigos similares para compor estruturas distintas em diversos níveis. Com isso, obtém-se escalabilidade e uniformidade em todo código da arquitetura. A seguir, serão detalhados os onze princípios que guiaram o processamento, armazenamento e visualização de informações da NG.

3.3.1 1º Princípio: Integração Coesa e Sinérgica

O primeiro princípio estabelece que todos os habilitadores da arquitetura devem operar de forma colaborativa e integrada entre si. Dessa forma, uma definição que se faça para um princípio de nomeação, segurança ou roteamento não pode ser auto-

excludente, mesmo que sejam definidos como princípios distintos, eles devem operar de forma harmoniosa e cooperativa.

3.3.2 2º Princípio: Generalização ao Máximo

Pelo segundo princípio da arquitetura se estabelece que os conceitos e implementações devem ser utilizados de forma recorrente no código, do maior ao menor escopo. Isso sintetiza a ideia de que se for definida uma forma de nomeação para um tipo de entidade, ela deve ser replicada em toda a arquitetura, não sendo indicado que se estabeleçam múltiplas implementações para a mesma finalidade.

3.3.3 3º Princípio: Nomeação e Resolução de Nomes

O terceiro princípio define que a nomeação e resolução de nomes das entidades devem garantir que qualquer formato e linguagem possam ser utilizados. Os operadores semânticos devem agregar funcionalidades e identificar relações entre entidades. É estabelecido também que sejam ilimitadas as possibilidades de nomeação e ligações (*bind*) de nomes de entidades.

3.3.4 4º Princípio: Roteamento por Nome e Armazenamento de Conteúdos

O quarto princípio define que se estabeleça o roteamento através dos nomes utilizados. Dessa forma, deve ser possível definir rotas e localizar conteúdos através de seus nomes. De forma complementar, define-se que os conteúdos possam ser armazenados na rede, ou pelo menos aqueles que tenham maior demanda de tráfego, para que quando armazenados também possam ser acessíveis pelos seus nomes. Mesmo a NG generalizando conteúdos e informações, esse princípio mantém sinergia com os paradigmas de arquiteturas ICN, pois garante a localização e armazenamento de conteúdos em rede através de nomes. Um diferencial importante da NG é que um único *cache* é utilizado para armazenar conteúdos, bem como as ligações entre esses nomes.

3.3.5 5º Princípio: Auto-organização

O quinto princípio toma como base a constante expansão das redes de dados e define que a auto-organização das entidades é exigência de uma arquitetura convergente. Isso é proposto tendo em vista que as interferências humanas devem ser minimizadas, uma vez que podem trazer pontos de falhas e também não serem exequíveis em todos os níveis de escalas. Os pilares um e dois, já citados, também contribuem com isso, pois estruturam uma nomeação eficiente e definem ciclos de vida para todas as entidades. É idealizada também a flexibilidade quanto à gestão [148] e automação das entidades, seja por *machine learning*, inteligência artificial ou outras tecnologias que se estabeleçam.

3.3.6 6º Princípio: Processamento de Informação como Serviço

O sexto princípio define que o processamento de informações da arquitetura deve ser definido como um serviço. Essa característica tem relação com o quarto pilar, que define a NG como uma SOA. Por esse motivo, implementações rotineiras de protocolos de comunicação, armazenamento de dados e até troca de conteúdos devem ser realizados através de serviços. Essa é uma mudança que pode aumentar a complexidade para uma simples troca de mensagens, pois ao estabelecer um serviço para a troca dessas mensagens, também é estabelecido um ciclo de vida desse serviço, bem como passa a ser possível acompanhar a sua qualidade através das definições de contratos. Algo que poderia ser desnecessário em pequena escala, mas em grande escala traz o benefício de ser mais simples o monitoramento e gerenciamento do processamento de informações, vislumbrando até implementações autônomas.

3.3.7 7º Princípio: Programabilidade do Físico

O sétimo princípio tem relação com o terceiro Pilar, pois define que os recursos físicos sejam controlados por serviços da arquitetura. Isso se justifica dada a tendência no aumento no número de dispositivos e a necessidade de estabelecer uma forma de configurá-los através de serviços da NG.

3.3.8 8º Princípio: Ciência de Contextos

Essa definição orienta a construção da arquitetura de forma que sejam estabelecidas condições para que todos os elementos e entidades da arquitetura tenham ciência do contexto em que estão inseridos. Essa abordagem é fundamentada no primeiro pilar, que determina a utilização de elementos semânticos de linguagem na definição de entidades e suas ligações. Porém, essa semântica pode ser extrapolada para qualquer contexto da arquitetura, tais como: rede, contratos, ambiente, recursos, entidades, etc. Dessa forma, um serviço passa a ser capaz de se orientar pelo *status* de um *link* de dados antes de começar a utilizá-lo.

3.3.9 9º Princípio: Generalização do Meio Físico

O nono princípio objetiva virtualizar e definir por serviços o máximo de entidades físicas possíveis. Alinhado com o terceiro pilar, ele define a criação de métodos que permitam a interação irrestrita entre entidades e seus representantes. A partir dessa definição é possível estabelecer ubiquidade entre o que existe no mundo físico e o que está representado no mundo virtual. Objetos como: bicicletas, carros, edifícios e cidades poderiam ser representados virtualmente, bem como caracterizadas as relações que possuem entre si. Esse princípio também prevê a generalização do físico para o número máximo de casos de uso possível.

3.3.10 10º Princípio: Representação e Exposição do Físico

Complementando o nono princípio, a exposição das entidades físicas e seus recursos contribuem com o estabelecimento dos serviços da NG, principalmente ao abstrair para

o *software* as capacidades do meio físico. Essa premissa é base para que se vislumbre o compartilhamento de recursos físicos em meio virtual e ampare propostas de economia das coisas, de dados, de espectro de radiofrequência e de funções virtuais de rede. Essa é justamente a definição do décimo princípio, que se relaciona também com o conceito trazido anteriormente, o dos Gêmeos Digitais.

3.3.11 11º Princípio: Segurança Intrínseca

A segurança e a privacidade são preceitos de qualquer sistema da atualidade. Esses princípios também são norteadores das implementações da NG e, assim como os demais, devem ser devidamente generalizados em todas as implementações. A segurança é uma característica capaz de garantir identidade, privacidade, responsabilidade e autenticidade para qualquer serviço que se estabeleça. O primeiro pilar é o que tem maior afinidade com esse princípio, pois através de metodologias genéricas, reversíveis e de ligação de nomes, se estabelecem condições para garantir as qualidades mencionadas.

3.4 Fundamentações e Padrões de Comunicação

Nesta seção, são apresentados padrões de comunicação e conceitos que são primordiais aos serviços e implementações práticas da NG, bem como para o entendimento geral da arquitetura. Tais conceitos são relacionados aos paradigmas de *name binding*, *Publish/Subscribe* e a terminologia/nomenclatura adotada na arquitetura. Ademais, explora-se os serviços empregados para constituir os processos NG. Por fim, toma-se como exemplo uma aplicação prática e funcional de distribuição de conteúdo nomeados se valendo da arquitetura proposta, focando no modelo de comunicação e nas métricas que podem ser obtidas nesse cenário.

3.4.1 Ligação de Nomes - *Name Bind*

O ambiente computacional é majoritariamente binário em suas instâncias mais baixas. Os dados são armazenados, processados e trafegam em sua forma digital (0 ou 1). As informações que são exibidas aos usuários são processadas inúmeras vezes até serem exibidas em formato inteligível por humanos. Essa situação se repete quando é preciso lidar com nomes de objetos em sistemas, arquivos, endereços de rede e tantos outros. Existe uma constante tradução de referências inteligíveis por humanos e referências processadas por máquinas. Esse processo de conversão de nomes inteligíveis a humanos e processados por máquinas é denominado resolução de nome (*name resolution*) [149].

A premissa que justifica essa abordagem é o fato de haver diferença entre a forma com que os seres humanos e máquinas lidam com as informações. Para lidar com informações, os computadores são programados para superar desafios de estruturação de dados, tipos definidos de informação, ordenação, sequenciamento e limitação de espaço. Na percepção humana, as conexões entre elementos devem seguir apenas a lógica: ser inteligível e preferencialmente memorizável.

Um exemplo de tradução de nomes que acontece a todo momento na Internet é a conversão de endereços URL para endereços IP. O site *www.example.com* pode ser

associado ao endereço IP: 203.0.113.1 e vice versa. Essa associação é considerada um NB [38]. De um lado existe uma referência que é de fácil utilização por pessoas e do outro um número que é de fácil processamento por máquinas. Esse exemplo é algo usual, porém a resolução de nomes se estabelece em vários níveis, até para ligações entre objetos e componentes exclusivos de *softwares*.

Uma associação de elementos pode ser idealizada de forma diferente do que a associação de nomes e IPs. Por exemplo, um NB poderia ser feito ligando o nome de uma pessoa ao endereço de sua casa ou ligando a placa de um veículo ao seu número de chassi. Diversas são as aplicações que demandam esse tipo de associação. Um exemplo mais específico para redes de computadores poderia ser a associação de conteúdos, dados ou mídias a identificadores únicos, o que proporcionaria uma identificação única e inequívoca dessas entidades.

Na NG os nomes são definidos para cada entidade. Esses nomes podem representar redes, *hosts*, serviços, sistemas, processos e até mesmo conteúdos. O armazenamento desses nomes em rede é idealizado de forma ilimitada. As relações entre entidades são estabelecidas por ligações de nomes autocertificáveis (*self-certified name-based bindings*) [150], isso permite que se estabeleçam qualquer tipo de relação entre as entidades. No entanto, é necessário que os objetos das entidades armazenem informações sobre as relações estabelecidas. As associações de nomes são armazenadas de forma distribuída na rede, permitindo estabelecer um serviço de identificação de nomes escalonável e indiferente aos nós.

3.4.2 Publicação/Assinatura

A crescente discussão [151] [152] pelo paradigma de publicação e assinatura está relacionada, principalmente, com a capacidade de desacoplar as variáveis de fluxo permanente de conexão, tempo e espaço, das entidades envolvidas. Em alguns casos, esse tipo de abordagem permite diminuir o tráfego da rede, pois apenas conteúdos publicados e assinados serão trocados entre as entidades, deixando de existir a troca de dados referente às buscas e estabelecimento de conexões.

- **Publicação:** É o processo de enviar conteúdos para a rede. Na NG, quando conteúdos são enviados, os assinantes podem ser notificados ou não, dependendo de como for definida a publicação. No caso de ocorrer uma notificação, de postagem ou atualização de conteúdos, ela é feita para a rede no nó mais próximo do publicador, o restante da arquitetura se encarrega que ela chegue aos assinantes. Do ponto de vista de quem publica uma informação esse fluxo pode ser muito mais simples, principalmente por estabelecer identificadores únicos aos conteúdos produzidos e não ter que se preocupar com a divulgação nó a nó desses dados. A proposta do NB da NG visa justamente facilitar essas ações, pois oferece também uma infraestrutura distribuída para armazenamento de conteúdos.
- **Assinatura:** É a sinalização do cliente quanto a sua intenção de receber informações ou atualizações sobre determinado tema já publicado ou não na rede. Através dessa abordagem, um nó que deseja receber dados de determinado conteúdo deve se explicito em sua intenção, fazendo isso através da assinatura de tais conteúdos, ou de uma maneira menos abstrata, registrando-se na rede como assinante do

identificador único do conteúdo em questão. Dessa forma, apenas materiais publicados, e que tenham sido assinados, serão encaminhados ao nó. Esse comportamento pode ser comparado a um *feed* de notícias RSS.

Esse conceito de *PUB/SUB* é central para o restante deste trabalho, porém algumas definições adicionais sobre a NG são necessárias antes de seguir com a explicação de como esse tipo de comunicação é estabelecido, de forma prática, na arquitetura.

3.4.3 Termos e Nomenclaturas da NovaGenesis

Para facilitar o entendimento da NG são apresentadas, no Quadro 3.1, as definições dos termos que já foram mencionados nas duas últimas seções e alguns outros que ainda serão apresentados, elas complementam o entendimento da arquitetura, seus serviços e processos.

Quadro 3.1: *Definição de termos da NovaGenesis.*

Termo	Definição
<i>Name</i>	Símbolo utilizado em linguagem natural
<i>Identifier</i>	Nome único para identificação inequívoca em um domínio
<i>Locator</i>	Um nome que relaciona posição espacial ou de distância relacional entre entidades
<i>Name Binding (NB)</i>	Entidade que relaciona nomes
<i>Process</i>	Instância de um programa de computador que possui Blocos e Ações internas
<i>Block</i>	Componente interno de um Processo que pode conter múltiplas Ações
<i>Action</i>	Componente interno de um Bloco que implementa funcionalidades
<i>Message</i>	O <i>Protocol Data Unit (PDU)</i> para troca de informações da NovaGenesis
<i>CommandLine</i>	Descreve uma ação a ser executada no destinatário, bem como seus parâmetros
<i>Service</i>	O mesmo que Processo
<i>Hash Table (HT)</i>	Um bloco que implementa a estrutura de uma tabela <i>hash</i>
<i>Gateway (GW)</i>	Bloco responsável pela troca de mensagens internas do processo
<i>Proxy/Gateway (PG)</i>	Bloco responsável pela troca de mensagens externas aos processos
<i>Hash Table Service (HTS)</i>	Tabela de <i>hash</i> distribuída construída por blocos HT
<i>Generic Indirection Resolution Service (GIRS)</i>	Serviço responsável pela indicação do HTS responsável por armazenar NBs ou conteúdos
<i>Publish/Subscribe Service (PSS)</i>	Serviço responsável pela integração entre publicadores e assinantes
<i>Proxy/Gateway/Controller Service (PGCS)</i>	Serviço responsável pelo encapsulamento de mensagens para envio através da camada de transporte, intermediário de dispositivos IoT e para configurações de controle por software

Fonte: Traduzido de Alberti, 2014-2017 [38–40].

3.4.4 Detalhes de Implementação

Nesta subseção são apresentadas visões práticas da NG, começando pela definição do formato da mensagem que é utilizada, a conceituação de seus termos e a exemplificação de um fluxo de comunicação na arquitetura.

As mensagens da NG são divididas em duas partes, linha de comando (*command line*) e carga útil (*payload*), ambas utilizam caracteres codificados no formato ASCII [153]. A *command line* corresponde às orientações trocadas entre objetos, o *payload* corresponde às informações trocadas com o sistema de arquivo do SO. De forma abstrata, pode-se dizer que os conteúdos da linha de comando são as trocas de informações entre as entidades da NG sobre como processar, armazenar ou encaminhar o conteúdo do *payload*. E a carga útil é justamente a informação/dado/conteúdo que faz uso da arquitetura, seja para ser processada, armazenada ou encaminhada. Um exemplo é dado a seguir, através de uma linha de comando e seus parâmetros, que são detalhados abaixo:

ng -command –alternative version [< n type B1 B2 B3 B4 ... Bn >]

em que:

- command*: indica a ação a ser executada
- alternative*: indica a ação alternativa a ser executada
- version*: define a versão da implementação
- []*: aponta um ou mais argumentos vetoriais
- n*: define o número de elementos do argumento
- type*: define o tipo dos elementos do argumento
- B1 B2 B3 B4 ... Bn*: são os elementos de um argumento

As trocas de mensagens são implementadas na camada NG, posicionadas logo acima da camada 2 do modelo de referência híbrido, referenciado no Quadro 1.1. Uma mensagem é uma linha de comando composta por vários comandos que definem a execução de Tarefas/Ações na NG. Os processamentos feitos pelas camadas 2 e 1 obedecem requisitos dessas camadas, sendo que as mensagens da NG são apenas encapsuladas em quadros da camada de enlace. A Figura 3.1 ilustra como a pilha de comunicação da NG interage com as camadas 1 e 2 do modelo de referência híbrido. Como exemplos, são considerados os fluxos de comunicação do serviço PGCS, em que ele transmite e recebe uma Tarefa/Ação NG através de Mensagens NG. O fluxo de comunicação do serviço PGCS, através das camadas 2 e 1, quando transmite mensagens para a rede é:

1. Uma Tarefa/Ação NG que precisa alcançar nós externos é inserida em um comando NG.
2. O Comando NG é inserido em uma Mensagem NG. Que também será composta por parâmetros e *payload*.
3. O PGCS recebe essa Mensagem, que ainda está na camada NG, e a avalia.
4. O PGCS, ao detectar que a Mensagem deve ser encaminhada a um nó externo, encapsula a Mensagem NG em um quadro *Ethernet* e o transmite para a rede. Nesse ponto são inseridas informações do *MAC Address* do nó que se quer alcançar.
5. O processamento do quadro *Ethernet* pela camada 2 e 1 é algo que não depende da NG. Ele pode seguir por meios compatíveis com o quadro *Ethernet* até que chegue ao nó receptor.

O fluxo inverso se estabelece quando essa comunicação chega ao receptor:

1. O nó receptor recebe um fluxo de dados em sua camada 1, que é encaminhado para a camada 2.
2. Na camada 2, o quadro *Ethernet* é desencapsulado e identifica-se que se trata de uma Mensagem NG.
3. A Mensagem NG é enviada para ser processada na Camada NG, especificamente no PGCS do nó.
4. O PGCS processa a Mensagem NG, distinguindo os comandos e parâmetros.

5. As Tarefas/Ações, definidas pelos comandos, respectivas ao PGCS são executadas.

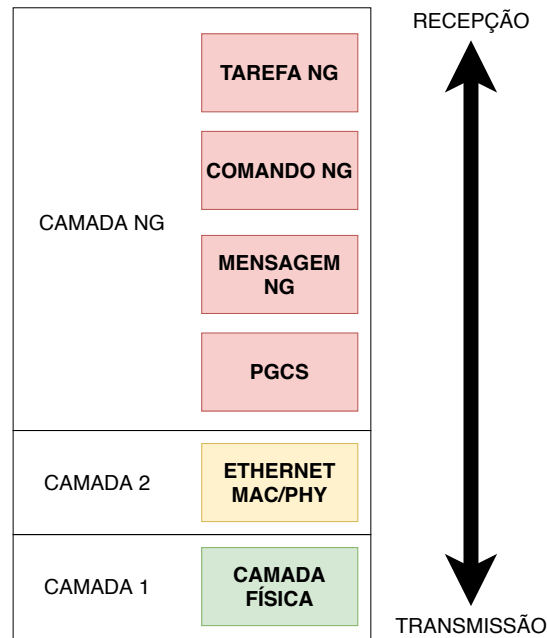


Figura 3.1: Fluxo de comunicação NG e a interação entre as camadas 1 e 2 do modelo de referência híbrido.

3.5 Serviços NG e de Resolução de Nomes

A NG é modular e seus componentes podem ser organizados em diferentes topologias, de acordo com a necessidade e/ou função que se deseja atribuir a um nó. Essa flexibilidade permite coordenar a atuação dos nós de forma a distribuir, balancear ou separar serviços. A seguir são descritos os serviços da arquitetura original que são utilizados nesta dissertação [154]: o *Proxy Gateway Controller Service* (PGCS) é um serviço essencial da rede para encapsulamento de mensagens sobre camada de enlace, representação do físico e tradução de protocolos; o *Name Resolution and Networking Cache Service* (NRNCS) é uma composição de três outros serviços: *Publish Subscribe Services* (PSS), *Generic Indirection Resolution Service* (GIRS) e *Hash Table Service* (HTS), que compõem o *cache* de rede e o sistema de resolução de nomes; e o NBSimpletestAPP, uma aplicação com função específica de gerar tráfego PUB/SUB de NBs, cujo resolução em escala é o principal alvo deste trabalho.

3.5.1 Proxy Gateway Controller Service (PGCS)

O PGCS é o serviço principal da NG. Ele é um *software-router*, ou seja, realiza a função de um roteador na rede, porém suas funções extrapolam as de um roteador normal. Ele também é responsável por iniciar a exposição de recursos, serviços, representação de entidades físicas, nós de armazenamento em *cache*, resolução de nome, encaminhamentos e roteamentos na rede. Ele faz comunicação direta com outros

serviços PGCS, de forma que se houver outro nó na rede eles se comunicam e atualizam todas essas informações um do outro. É uma forma de operacionalizar a descoberta de recursos e entidades da rede, como o PGCS reúne informações sobre todos os serviços e informações do nó, a comunicação com outros pares permite o compartilhamento desses dados.

3.5.2 Name Resolution and Networking Cache Service (NRNCS)

O NRNCS é um serviço da NG que realiza a resolução de nomes e armazenamento de conteúdos em *cache*. Essas tarefas não são executadas por um só serviço da NG, mas sim por três: PSS, GIRS e HTS. Atuando de forma conjunta eles são capazes de: publicar e assinar conteúdos; resolver indireções de nomes definindo uma tabela *hash* adequada ao armazenamento do conteúdo; armazenar o conteúdo nomeado no *cache* de rede, que inclui uma tabela *hash*, além de um sistema de arquivos do Linux para armazenamento de objetos de informação. De forma detalhada, são descritas as funcionalidades de cada um desses serviços:

- O PSS é o serviço de publicação da NG. Pode ser comparado a uma API que possui as implementações dos seguintes métodos: (i) publicar NB, com ou sem dados; (ii) publicar NB e notificar outros serviços sobre a publicação; (iii) assinar NB e/ou conteúdos; (iv) assinar o NB/conteúdo e notificar outros serviços sobre esta ação; (v) entregar NB e/ou conteúdos assinados; (vi) revogar publicação de NB e/ou conteúdo.

- O GIRS é o serviço responsável por receber as mensagens de publicação e assinatura do PSS e encaminhá-las para uma tabela *hash* que armazenará os NBs. É um serviço intermediário entre HTS e PSS, que tem função essencial quando existe mais de um HTS na rede. O direcionamento dos NBs entre múltiplos HTSes é definido pelo GIRS. Para isso ele indica o HTS de destino realizando um cálculo matemático. Não há necessidade de estabelecer qualquer tabela adicional de mapeamento ou *Forwarding Information Base* (FIB). O cálculo se dá pela divisão da chave (*key*) do NB pelo número de instâncias HTSes detectadas, o resto dessa divisão indicará para qual HTS irá aquele NB. Exemplo: se a rede possui 4 HTSes, e 4 NBs possuem as seguintes chaves: 60, 61, 62 e 63. Como resto da divisão de cada uma das quatro chaves tem-se, respectivamente: 0, 1, 2 e 3. Desta forma: 0, 1, 2 e 3 são os identificadores dos HTSes que receberão tais conteúdos. Essa abordagem simplifica o encaminhamento de conteúdos sem a necessidade de manter tabelas adicionais de mapeamento.

- O HTS é o serviço que armazena os NBs e seus conteúdos de forma distribuída em um domínio. A estrutura de dados utilizada é a de uma tabela *hash* distribuída (DHT) baseada em nomes autoverificáveis. A principal vantagem desse modelo é o seu desempenho para buscar registros de NBs. O HTS também é o responsável por responder solicitações de assinaturas dos conteúdos que são armazenados no sistema de arquivos do SO, GNU-Linux no caso.

Esses três serviços são essenciais em um domínio NG. Não precisam estar necessariamente no mesmo nó, mas sem eles a publicação e assinatura de conteúdos não é possível.

3.5.3 Aplicação NBSimpleTestAPP

O serviço NBSimpletestAPP é uma aplicação da NG. Diferentemente dos serviços anteriores, ele não possui papel estruturante na arquitetura. Sua função é ser uma aplicação que faz uso da rede, para gerar publicações e assinaturas de NBs. Essa função tem forte semelhança com a de um servidor DNS, que publica e recebe informações de endereços IPs e nomes de domínios da Internet. Considerando que a associação de endereços IPs e nomes de domínios pode ser definida como um *name binding*, o que o NBSimpletestAPP faz é algo parecido. Ele publica e solicita assinaturas de NBs, gerando um tráfego de dados que é processado pelos outros serviços da NG.

De forma a resumir os métodos de publicação e assinatura de NBs, a publicação acontece primeiro. Ela consiste em gerar NBs dentro de um intervalo determinístico, pré-definido e sequencial, enviando-os a outros nós do domínio. Já a assinatura é o processo inverso, pois o serviço inicia a solicitação de assinatura dos conteúdos publicados através de seus nomes ao domínio, de forma aleatória e que esteja compreendida no intervalo da publicação anterior. Após isso, recebe de volta a assinatura da informação publicada. Estabelecendo tais métodos, tem-se um exemplo de publicação e assinatura de NBs na NG.

A característica de gerar tráfego de rede não é a única do NBSimpletestAPP. Em sua implementação foi idealizada a necessidade de avaliar a NG de uma forma mais rigorosa, por isso, alguns registros dos tempos de comunicação e métricas estatísticas estão presentes em seus *logs* de execução, sendo eles:

- Instante de tempo: tempo em que aconteceu o evento de publicação ou assinatura de NB.
- Valor instantâneo: tempo de ida e volta (*Round Trip Time* - RTT) da publicação ou assinatura.
- Média dos tempos instantâneos

$$\mu = \frac{\sum x}{N} \quad (3.1)$$

em que:

- μ representa o tempo médio.
- x representa o tempo instantâneo.
- N representa o número de eventos.

- Desvio padrão: Medida de variação entre os valores instantâneos:

$$\sigma = \sqrt{\frac{\sum (x - \mu)^2}{N}} \quad (3.2)$$

em que:

- σ representa o desvio padrão.
- x representa o tempo instantâneo.
- μ representa o tempo médio.
- N representa o número de eventos.

- Erro padrão da média

$$\sigma_{\mu} = \frac{\sigma}{\sqrt{N}} \quad (3.3)$$

em que:

- σ_{μ} representa o erro padrão a partir da média μ .
- N representa o número de eventos.
- σ representa o desvio padrão.

- Intervalo de confiança ou margem de erro

$$IC = 1,96 \cdot \sigma_{\mu} \quad (3.4)$$

em que:

- 1,96 é uma constante de aproximação para distribuições normais
- σ_{μ} representa o erro padrão a partir da média μ .

- Média menos a margem de erro

$$\mu_{low} = \mu - IC \quad (3.5)$$

em que:

- μ_{low} representa a média menos a margem de erro.
- μ representa o tempo médio.
- IC representa a margem de erro/intervalo de confiança.

- Média mais a margem de erro

$$\mu_{up} = \mu + IC \quad (3.6)$$

em que:

- μ_{up} representa a média mais a margem de erro.
- μ representa o tempo médio.
- IC representa a margem de erro.

Além de fazer esses registros estatísticos, a aplicação ainda permite personalizar seus parâmetros de operação, através da alteração do seu arquivo de inicialização, explicado no quadro abaixo e também contemplado no Apêndice B.5.

Quadro 3.2: *Parâmetros de inicialização do NBSimpleTestAPP*

Parâmetro	Descrição	Valor Usual
DelayBeforeDiscovery	Tempo de aguardo para descoberta de outros nós após inicialização do serviço	Definido de forma constante em: 3 segundos
DelayBeforeRunPeriodic	Tempo de aguardo para repetição de envio de mensagens	Definido de forma constante em: 60 segundos
NumberOfPublications	Número de publicações que se deseja executar	Definido de forma variável de 1 a 32 milhões
NumberOfSubscriptions	Número de assinaturas que se deseja executar	Definido de forma constante em: 360
NumberOfMessagesPerBurst	Número de mensagens por rajada de comunicação	Definido de forma constante em: 50
NumberOfPubsPerMessage	Número de NBs presentes em cada mensagem	Definido de forma constante em: 100

Fonte: Valores personalizados pelo autor para atender os cenários em avaliação, 2021.

3.6 Aplicação NBSimpletestAPP para Avaliação da NG

Diante da possibilidade de utilizar essa aplicação¹ da NG, para gerar um fluxo de dados e obter informações de desempenho da arquitetura, são apresentados detalhes sobre o fluxo de comunicação do NBSimpletestAPP e suas especificidades. As minúcias desse serviço e como ocorre o fluxo de dados na arquitetura são apresentados em três exemplos, dois deles que detalham o fluxo de informação, para publicação e assinatura de NBs, a nível dos nós e serviços da NG, e um que detalha o fluxo de comunicação para armazenamento de NBs a nível dos blocos intrasserviços da NG.

3.6.1 Publicação e Assinatura com o Name Bind Simple Test Application

Para a aplicação NBSimpletestAPP, uma publicação de NB consiste em enviar um número sequencial, de 1 a 32 milhões, para que ela seja registrada e armazenada em *cache* pela rede, especificamente em um bloco *hash table*. Uma assinatura de NB consiste em solicitar à rede um número aleatório, que esteja contido no intervalo das publicações, e receber esse NB, após o processamento e consulta ao *cache* citado.

A Figura 3.2 ilustra a sequência do fluxo de informação para a publicação de NBs através dos serviços da NG. São ilustrados três nós, sendo: *Node 1*, o responsável pela publicação de NBs; *Node 2*, o responsável pelo processamento das publicações de NBs e encaminhamento para local de armazenamento; *Node 3*, o responsável pelo armazenamento dos NBs. Ainda na Figura 3.2, é possível observar que os serviços não se comunicam diretamente com o serviços de outros nós, essa comunicação é realizada sempre através de um intermediário, o PGCS local do *Node*. Por esse motivo é possível observar que os serviços possuem livre comunicação interna, mas quando precisam acessar um *Node* externo se direcionam ao PGCS local. No exemplo, a publicação de um NB se inicia no *Node 1*, na aplicação geradora do tráfego (NBSimpletestAPP), segue para o *Node 2*, onde é publicada, e encaminhada para armazenamento no *Node 3*. O *Node 3* recebe o NB e o armazena, enviando uma confirmação de publicação ao *Node 1*. De forma detalhada, tem-se o seguinte fluxo:

1. No *Node 1*, a aplicação NBSimpletestAPP gera a publicação do NB e a encaminha para o PGCS local;
2. O PGCS recebe tal informação e como ele possui informações dos recursos e serviços estabelecidos na rede, encaminha para um *Node* que possua o serviço de publicação, no caso o *Node 2*;
3. O PGCS do *Node 2* recebe essa informação e a trata. Percebendo que se trata de uma publicação, encaminha para o serviço interno responsável por isso, o PSS;
4. Quando essa informação chega ao PSS, ele realiza o processamento da publicação e depois a encaminha para o GIRS, para que este possa decidir onde armazenar o NB;
5. O GIRS recebe a informação e a processa quanto ao local de armazenamento.

¹A NG não distingue serviços de aplicações, embora uma aplicação possa ser vista como um serviço que se baseia em outros serviços.

Como nesse exemplo só o *Node 3* possui um serviço compatível de armazenamento, HTS, ele retorna a informação ao PGCS local para que este a envie ao *Node 3*;

6. O PGCS do *Node 2* recebe informação do GIRS e a encaminha para o PGCS do *Node 3*, já sabendo pela etapa anterior que o GIRS havia decidido armazenar o NB no HTS do *Node 3*;
7. O PGCS do *Node 3* recebe a informação do PGCS do *Node 2* e a processa, percebendo que se trata de um armazenamento de NB, o encaminha para o HTS interno;
8. O HTS recebe o NB e o armazena. Ao executar essa ação retorna uma mensagem de “ok” com destino ao serviço que originou a publicação, o NBSimpletestAPP do *Node 1*. Como essa informação se dará para outro *Node*, a mensagem é encaminhada, primeiramente, ao PGCS local;
9. O PGCS do *Node 3* recebe a mensagem de confirmação e a destina para o PGCS do *Node 1*;
10. O PGCS do *Node 1* recebe a informação, a trata e encaminha para o seu serviço local, que é o destinatário da mensagem, o NBSimpletestAPP;
11. O NBSimpletestAPP recebe e processa a confirmação, encerrando então a sequência de publicação de um NB.

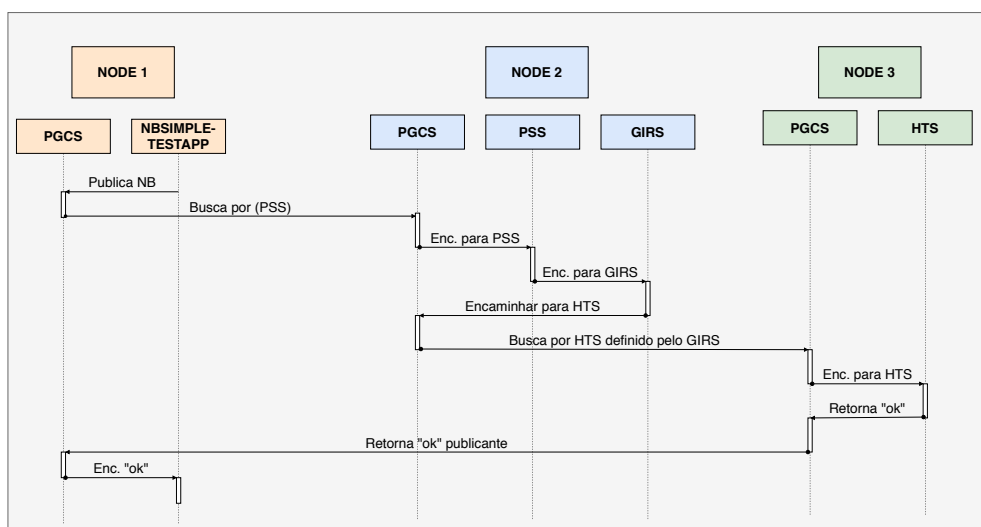


Figura 3.2: Sequência do fluxo de informação da publicação de NBs para os serviços da NG.

A Figura 3.3 ilustra a sequência do fluxo de informação para a assinatura de NBs através dos serviços da NG. São ilustrados três nós, sendo: *Node 1*, o responsável pela assinatura de NBs; *Node 2*, o responsável pelo processamento das assinaturas de NBs recebidas e o encaminhamento da solicitação ao nó que armazena o conteúdo solicitado; *Node 3*, o responsável pelo armazenamento resposta às solicitação de assinaturas de NBs. Assim como no exemplo anterior, as comunicações entre os nós são sempre intermediadas pelo PGCS local. Neste exemplo, a assinatura de um NB se inicia no *Node 1*, na aplicação geradora do tráfego (NBSimpletestAPP), segue para o *Node 2*, onde é feito o processamento da assinatura e identificação de qual nó da rede contém o

conteúdo solicitado, e então segue para o *Node 3*, que recebe a solicitação de assinatura de um conteúdo e envia o NB, correspondente à assinatura e que está armazenado em seu HTS, ao *Node 1*. De forma mais detalhada, tem-se o seguinte fluxo:

1. No *Node 1*, a aplicação NBSimpletestAPP gera a assinatura do NB e a encaminha para o PGCS local;
2. O PGCS recebe tal informação e, como ele possui informações dos recursos e serviços estabelecidos na rede, encaminha para um nó que possua o serviço de publicação e assinatura de conteúdos, no caso, o *Node 2*;
3. O PGCS do *Node 2* recebe essa informação e a trata, percebendo que se trata de uma assinatura, encaminha para o serviço interno responsável por isso, o PSS;
4. Quando essa informação chega ao PSS, ele realiza o processamento da assinatura e depois a encaminha para o GIRS, para que este possa decidir onde foi armazenado o NB solicitado;
5. O GIRS recebe a informação e a processa quanto ao seu local de armazenamento. Como nesse exemplo só o *Node 3* possui um serviço compatível de armazenamento (HTS) ele retorna a informação ao PGCS local para que este a envie ao *Node 3*;
6. O PGCS do *Node 2* recebe a informação do GIRS e a encaminha para o PGCS do *Node 3*, já sabendo pela etapa anterior que o NB está armazenado no HTS do *Node 3*;
7. O PGCS do *Node 3* recebe a informação do PGCS do *Node 2* e a processa, sabendo que se trata de uma assinatura de NB, o encaminha para o HTS interno;
8. O HTS recebe a solicitação de assinatura de NB e a processa, ao executar essa ação consulta e encontra, em sua *hash table* um conteúdo correspondente. Assim, ele retorna a assinatura recebida com o NB solicitado ao remetente inicial, o NBSimpletestAPP do *Node 1*. Antes, claro, seguindo o fluxo através do seu PGCS local;
9. O PGCS do *Node 3* recebe a mensagem de confirmação e a destina para o PGCS do *Node 1*;
10. O PGCS do *Node 1* recebe a informação, a trata e encaminha para o seu serviço local, que é o destinatário da mensagem, o NBSimpletestAPP;
11. O NBSimpletestAPP recebe e processa a assinatura do NB solicitado.

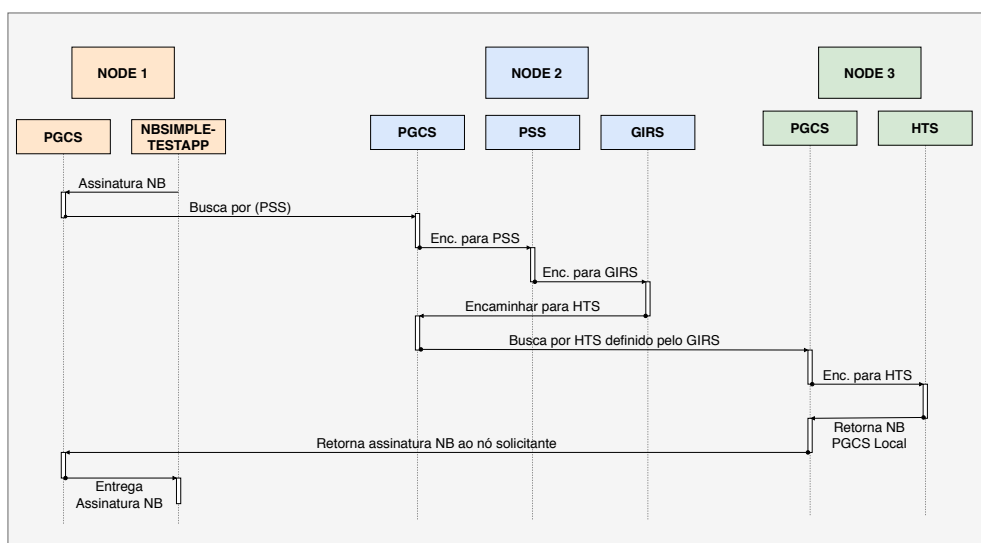


Figura 3.3: Sequência do fluxo de informação da assinatura de NBs para os serviços da NG.

3.6.2 Fluxo de Comunicação Entre Blocos da NG: Publicação e Assinatura de *Name Binds*

A Figura 3.4 ilustra a comunicação que é feita pela aplicação NBSimpletestAPP para realizar o armazenamento de um NB em um nó HTS distribuído, tal qual será realizado nos experimentos dessa dissertação. O fluxo de comunicação está enumerado de 1 a 24 conforme abaixo:

1. O início da comunicação se dá com a criação de uma mensagem de publicação por parte do bloco da aplicação NBSimpleTestAPP. A mensagem é criada no formato adequado para envio ao PSS e é encaminhada para o *gateway* local da aplicação NBSimpleTestAPP.
2. O *gateway* do NBSimpleTestAPP identifica que a mensagem é para um serviço externo e a envia, através da memória compartilhada, para o PGCS do nó APP.
3. O PGCS do nó APP então recebe a mensagem, através da memória compartilhada, em seu bloco do *gateway* local.
4. O *gateway* do nó APP identifica que a mensagem se destina ao nó CORE e que é uma mensagem para um serviço externo, envia a mensagem para o *proxy gateway* local.
5. O *proxy gateway* do PGCS utiliza um *raw socket* e o encapsulamento da NG sobre Ethernet para enviar a mensagem ao nó CORE.
6. Já no CORE, o PGCS, através do seu *proxy gateway*, que monitora os dados trafegados no *raw socket*, recebe a mensagem.
7. A mensagem do nó APP possui o identificador do PSS *identifier* em seu cabeçalho. Com isso o PGCS determina que se trata de um direcionamento a um serviço local e a encaminha para o *gateway*.
8. Para alcançar o PSS do nó CORE, o *gateway* do PGCS direciona a mensagem à memória compartilhada.

9. O PSS recebe a solicitação de publicação, através da comunicação do seu *gateway* com a memória compartilhada.
10. O bloco pub/sub recebe a publicação, a registra e retorna para a o *gateway* do PSS, com destino ao GIRS.
11. O *gateway* do PSS recebe o retorno do PUB/SUB.
12. O *gateway* do PSS retorna a mensagem à memória compartilhada com destino ao GIRS.
13. O GIRS recebe a mensagem, através do seu *gateway*.
14. Em seguida, envia para o bloco *Indirection Resolution (IR)*. No IR é feita a tratativa de definir para qual HTS será enviada a publicação. Como já foi exemplificada na subseção 3.5.2.
15. O IR então retorna a publicação ao *gateway* do GIRS . Isso é feito porque não existe nenhum HTS nesta implementação, o HTS está em um nó externo que só pode ser acessado através do *proxy gateway* do PGCS.
16. É enviada novamente a mensagem à memória compartilhada.
17. O PGCS volta a receber a mensagem, através do seu *gateway*.
18. É encaminhada a mensagem externa ao *proxy gateway*.
19. Novamente, o *proxy gateway* identifica que é uma tratativa com um nó externo, então faz uso do *raw socket* e o encapsulamento da NG sobre *Ethernet* para enviar a mensagem ao nó HTS.
20. O nó HTS possui um PGCS, que através do seu *proxy gateway*, monitora a comunicação do *raw socket*.
21. Assim que o *proxy gateway* recebe a publicação, ele a envia ao *gateway* do PGCS.
22. O *gateway* então envia a mensagem à memória compartilhada para alcançar o HTS.
23. O *gateway* do HTS recebe a mensagem que lhe destinava e a envia para armazenamento.
24. O local de armazenamento definitivo será a *distributed hash table (DHT)*, baseado no sistema de arquivos Linux do nó HTS.

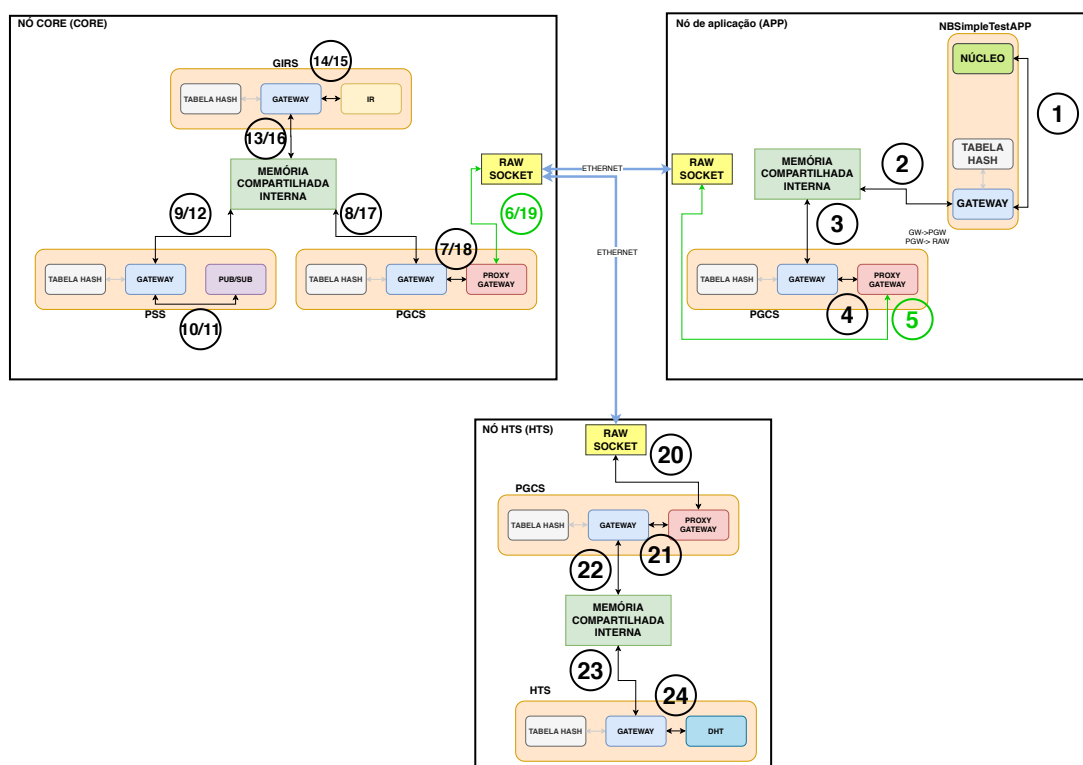


Figura 3.4: Fluxo de comunicação para armazenamento de name binds em rede.

3.7 Entendimentos do Capítulo

Neste capítulo são expostos os princípios estruturantes da NG, sendo elencados seus pilares conceituais, seus princípios de implementação e seus principais serviços. É dado destaque à função de Nomeação e Resolução de Nomes da arquitetura, bem como a descrição do serviço NBSimpletestAPP, que é fundamental para realização das avaliações de rede nos capítulos seguintes. As exemplificações dos fluxos de comunicação para publicação e assinatura de NBs são essenciais às avaliações que se deseja propor no capítulo seguinte, em conjunto com o NBSimpletestAPP e suas métricas, será possível definir uma metodologia para avaliar o serviço de resolução de nomes da NG.

Capítulo 4

Proposta de Metodologia de Teste e Ambiente de Avaliação em Escala para a NovaGenesis

Como destacado no Capítulo 2, observa-se um importante papel dos ambientes de experimentação no desenvolvimento de propostas de Internet do Futuro e novas arquiteturas. As FIAs apresentadas neste trabalho adotam metodologias diversas como avaliação de modelos teóricos, simulações, experimentação em ambientes controlados, em ambientes elásticos, em *testbeds*, em nuvens computacionais e até em ambientes mistos, de forma que eventualmente são comparadas com a arquitetura atual da Internet e os protocolos TCP/IP. Este capítulo tem como objetivo propor uma metodologia de teste e um ambiente de avaliação escalonável para o cenário de resolução de nomes em um domínio local da NG. O objetivo da metodologia é avaliar como a NG resolve alguns dos problemas elencados nos capítulos anteriores ao utilizar nomeação auto verificável, *cache* de rede, auto-organização dos serviços de resolução de nomes, desacoplamento ID/Loc, dentre outras novidades. São citadas as experimentações realizadas da NG em *testbeds* e que justificaram a necessidade de se estabelecer um ambiente de avaliação virtual local.

Em seguida, será descrita uma nova metodologia para esse fim, composta por ambientes de experimentação específicos, procedimentos operacionais, topologias de nós e serviços da NG. Tudo isso para avaliar o serviço de resolução de nomes da arquitetura de diferentes formas, contribuindo para a comprovação científica da NG.

4.1 NovaGenesis em *Testbeds*

As avaliações em *testbeds* da NG que precederam esse trabalho foram publicadas em [93] e apresentadas em [155]. Elas tiveram significativa importância para a evolução das metodologias de experimentação abordadas nas próximas subseções. O detalhamento da NG em *testbeds* gerou importantes percepções quanto à utilização dessas infraestruturas de teste, porém, não é feita uma comparação com o atual trabalho, pois elas não possuem metodologias e execuções comuns, apenas amparam as conclusões qualitativas. A falta de tais comparativos não impactou as avaliações e conclusões deste trabalho. A

seguir são citadas as interações da NG em *testbeds*.

Até o ano de 2016 a NG tinha pouco histórico de experimentações escalonáveis [38]. A maioria dos testes foram feitos em laboratórios de informática e com pouca automação de procedimentos. Questões intrínsecas às execuções manuais podiam comprometer a uniformidade dos experimentos, aumentar o risco de falhas humanas e o tempo de execução. Por volta do segundo semestre de 2015, foi proposta a execução da NG em *testbeds*. A primeira plataforma utilizada foi a GENI [36]. Nesse primeiro momento, a execução foi bem simples. Apenas dois *hosts* executando a NG e trocando mensagens de “HELLO”. Apesar de simples, essa visão trouxe ideias de ajustes e em pouco tempo foi possível executar pela primeira vez a NG em um *testbed*, fato ocorrido em setembro de 2015. Esse evento teve importante significado para o grupo de pesquisa, pois abriu espaço para intensificar a experimentação da NG em ambientes virtuais, em plataformas com maior disponibilidade de recursos e fomentar a utilização de *testbeds*, principalmente aqueles voltados para desenvolvimento de arquiteturas de rede. Isso contribuiu para o estabelecimento de múltiplas fontes de comparação e avaliação da NG. Porém, após algumas rodadas de testes, o GENI se mostrou com algumas limitações de execução em relação aos requisitos do projeto, principalmente quanto à limitação de recursos computacionais (memória e CPU), flexibilidade para o aumento no número de instâncias virtuais e personalização do sistema operacional a ser executado.

Seguindo a busca por outros ambientes de avaliação, surgiu a oportunidade de executar a NG em um *testbed* sediado no Brasil, o FIBRE [86]. Dessa interação vários avanços foram obtidos, destacando-se a possibilidade de gerar um primeiro comparativo entre as execuções da NG em ambientes reais e virtuais [155] o que foi facilitado pela agilidade de contato, interesse e disponibilidade da equipe do FIBRE em fomentar esse tipo de experimentação. Nesse ambiente, também foi possível ajustar necessidades por parte da NG e ter retorno em personalizações de ambientes para execuções dos testes, que são configurações dedicadas de interconexões de rede, expansão de capacidades computacionais e instalações de SOs personalizados.

Pouco a pouco as execuções em ambientes escalonáveis permitiram evoluir a implementação da NG. Os resultados obtidos retroalimentavam as frentes de desenvolvimento e novas possibilidades de pesquisa surgiam [93, 155]. O código que era utilizado na maioria das avaliações se mostrava estável nas condições em que era testado. Isso motivou a busca por formas de extrapolar ainda mais as capacidades da NG. As principais frentes de atuação envolviam experimentações em laboratórios de informática e no FIBRE. A primeira se mostrava difícil de lidar, exigia conciliação de agendas de disponibilidade, dedicação de tempo na preparação física, mobilização e desmobilização de recursos. A segunda, apesar de não exigir tantas ações manuais, e mesmo possuindo facilidade de acesso, não dispunha de recursos para alocação de múltiplas *Virtual Machines* (VMs), em uma única ilha, em condições de permitir controle total dos recursos de *hardware*. Aliava-se a isso a condição de ser uma plataforma compartilhada com outros projetos, o que poderia interferir na padronização da experimentação, na medida em que poderia haver variação na utilização dos recursos primários dos servidores.

Em meados de 2018, o *ICT Laboratory* do Inatel recebeu investimentos e pôde contar com equipamentos e servidores exclusivos para a experimentação da NG. Com isso, surgiram novas oportunidades e as metodologias de avaliação foram repensadas, sempre

com objetivo de extrapolar os conceitos da NG e das suas capacidades. Agora, em condições de controlar e ajustar todos os parâmetros e recursos que uma experimentação em escala exige. Em consequência disso, foi elaborado um plano de ação cujo objetivo era comparar o desempenho da NG e suas topologias em ambientes virtual e físico (laboratórios de informática). Foram estabelecidas metodologias que avaliassem as topologias e serviços, dentre eles publicação, assinatura e distribuição de *name bindings* (NBs) da NG.

4.2 Ambientes de Avaliação e Ferramentas

Dadas as práticas com *testbeds* e a necessidade de um ambiente mais controlado e expansível, entendeu-se que a virtualização poderia ser uma opção. Observou-se que outras FIAs também utilizam de tal recurso quando promovem a expansão de suas experimentações, principalmente quando precisam manter o controle mais fino dos recursos e do ambiente. Com intuito de definir uma nova metodologia de avaliação em um ambiente pouco explorado pela NG, foi necessário manter também os ambientes que eram utilizados. Dessa forma, o ambiente físico (laboratórios de informática) foram mantidos como parâmetro de controle e comparação. Apesar da maior dificuldade operacional, eles possuem um histórico de confiabilidade em experimentações e operação da arquitetura pelo grupo de pesquisa. Para executar e automatizar as funções que se propôs, foram utilizados vários programas externos, que são descritos também como recursos de *softwares*, pois formam um conjunto de soluções que proveem o compartilhamento de recursos, virtualização, sistemas operacionais, interconexão de rede, controle de acesso e automação de procedimentos.

As plataformas de experimentação utilizadas foram laboratórios de informática e um servidor de virtualização. Os equipamentos, *softwares* e metodologia de automação são associações de *commodities*, *softwares open source* e desenvolvimento de aplicações. Com exceção da metodologia, que pode ser considerada uma contribuição científica, o que descreveremos em seguida, são recursos estruturantes desenvolvidos para avaliação da arquitetura proposta.

4.2.1 Ambiente Físico

Para este trabalho, foram utilizados um conjunto de laboratórios de informática que possuem especificações e infraestrutura padronizada. Foram reservados cinco laboratórios com vinte e oito computadores cada. A Figura 4.1 exhibe o diagrama desses laboratórios. O único espaço que não foi utilizado foi o LTI 1.1, todos os demais: LTI 1.2, LTI 1.3, LTI 1.4, LTI 1.5 e LTI 1.6 foram configurados e estavam à disposição para uso. A utilização das máquinas variava de acordo com a disponibilidade dos espaços, pelas suas características similares. Todas máquinas estavam aptas à execução das avaliações. O quantitativo de máquinas utilizado era definido de acordo com a demanda da bateria de testes do momento. Os procedimentos operacionais são descritos na Seção 4.4, mas um cenário de testes pode conter no mínimo dois computadores e no máximo seis, sendo que a execução em paralelo desse cenário era limitada apenas pelo número de equipamentos disponíveis, nos laboratórios, no momento. Cada ambiente

possui um *switch* dedicado interligando suas máquinas em um mesmo enlace de camada 2. Dessa forma, foram feitos agrupamentos de máquinas entre laboratórios para garantir que toda comunicação se estabelecesse sob o mesmo *switch*.

Entre as máquinas existiam pequenas variações de especificações, principalmente em relação à quantidade de memória e disco rígido. Porém, tais especificações não eram inferiores às descritas no Quadro 4.1. O computadores eram da marca e modelo: Dell Optiplex 7010, com processador Intel Core I5 de 4 cores, 4 GB de memória RAM, placa de rede de 1 Gbps e disco rígido de 1 TB.

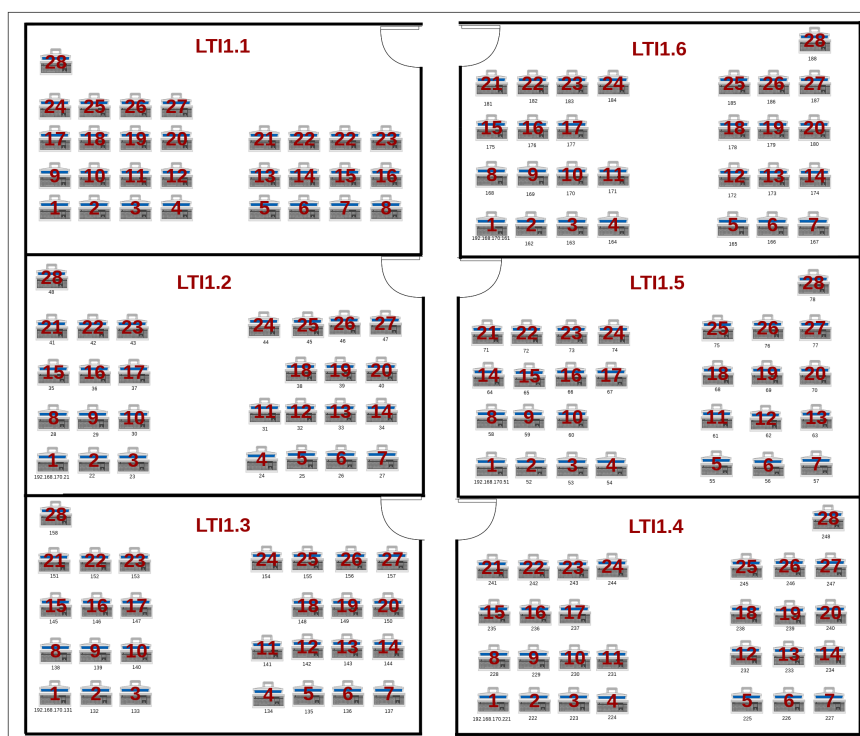


Figura 4.1: Diagrama representativo dos computadores dos laboratórios de informática.

Quadro 4.1: Especificações da configuração mínima de hardware dos computadores dos laboratórios.

Característica	Descrição
Marca/Modelo	Dell Optiplex 7010
Processador	Intel Core i5 3470 4 cores (3.60GHz, 6MB Cache, 5 GT/s)
Memória	4 GB (1x4GB) SDRAM DDR3 1600 MHz
Placa de rede	1GbE: Intel (1000Base-T)
Armazenamento	1x HDD SATA 1TB, 7.2K RPM

Fonte: Elaborado pelo autor, 2021.

O sistema operacional escolhido foi comum aos dois ambientes. Para a sua definição foi preciso conciliar a plataforma em que a NG está compilada que é o GNU-Linux. Com apenas essa restrição, foi escolhida a versão 16.04 LTS do Ubuntu, uma vez que era totalmente compatível com esse requisito, garantindo compatibilidade com todas dependências e adicionando flexibilidade para suportar as rotinas de automação e procedimentos desenvolvidos. Reforçaram essa escolha o longo suporte oferecido

para a versão Ubuntu LTS e a facilidade de personalização das configurações de placas rede da versão 16.04. Os *softwares* instalados no SO são listados no Apêndice A.1.9 e compreendem, principalmente, pré-requisitos de compiladores, programas para realizar a comunicação dos *sockets*, dependências para execução dos códigos de automação e monitoramento.

4.2.2 Ambiente Virtual

O ambiente de virtualização definido para este trabalho é baseado no *hypervisor* KVM. Como explicitado no Capítulo 2.2, a solução possui o código livre, é flexível quanto à integração com outras plataformas de virtualização, permite a paravirtualização e, principalmente, executa a virtualização a partir do *kernel* do sistema hospedeiro, como ilustrado na Figura 2.1. O servidor utilizado para receber o sistema de virtualização é um equipamento com grande capacidade de processamento e pode ser considerado uma *commodity*. As especificações do equipamento são: Dell PowerEdge T640, 2 processadores Intel R Xeon de 10 núcleos, 256 GB de memória, 2 interfaces de rede 10 Gbps, 2 interfaces de rede 1 Gbps, 2 discos sólidos de 480 GB e 3 discos rígidos de 4 TB, como descritos no Quadro 4.2. O equipamento é mostrado na Figura 4.2.

Quadro 4.2: Especificações da configuração de hardware do servidor de experimentação.

Característica	Descrição
Marca/Modelo	Dell PowerEdge T640 BCC
Processador	2x Intel R Xeon TM Silver 4114 10 cores (2.20GHz, 13.75MB Cache, 9.6GT/s)
Memória	256GB (8x32GB) RDIMM DDR4 2667 MT/s
Placa de rede	2x 10GbE: Intel I350 (10GBase-T); 2x 1GbE: Intel I350 (1000Base-T)
Armazenamento	2x SSD SATA 480GB 6 Gbps; 3x HDD SATA 4TB, 7.2K RPM

Fonte: Elaborado pelo autor, 2021.

Detalhadas as especificações de *hardware* do servidor, as máquinas virtuais - *Virtual Machines* (VMs) - foram configuradas da seguinte forma: dezessete VMs foram criadas, todas com as mesmas reservas de recursos: quatro cores de processamento, uma interface de rede em modo *bridge* e uma unidade de armazenamento de 20 GB, como detalhado no Quadro 4.3. Na Figura 4.3 é exibida a tela de console do VMM destacando as VMs utilizadas e suas respectivas utilizações de CPU, em um instante que todas estão em espera.

Quadro 4.3: Especificações dos recursos das VMs.

Característica	Descrição
Processamento	4 cores
Memória	4 GB
Conexão de rede	1 conexão de rede em modo <i>bridge</i>
Armazenamento	20 GB

Fonte: Elaborado pelo autor, 2021.

Uma das VMs foi definida como instância de controle para que pudesse executar as rotinas de automação e gerência das demais VMs. Ela é a única VM que possui conexão com todas as demais, sendo responsável pela execução das rotinas de automação em todo o ambiente. A VM de controle não impacta na experimentação, ela comanda as demais VMs e os computadores dos laboratórios nas rotinas de automação da experimentação,



Figura 4.2: Equipamento servidor de virtualização utilizado nos experimentos.

que podem ser: instalação de dependências e programas; envio de dados e códigos da NG; inicialização da experimentação da NG nos respectivos nós em ordem predefinida; reinicialização dos sistemas; e transferência de resultados.

As VMs possuem conectividade de rede em modo *bridge* [156]. Nessa configuração, as interfaces das VMs têm acesso direto à rede física a que o servidor está conectado. Ou seja, é uma forma que permite o acesso direto às redes físicas sem criar regras complexas, passagem por *Network Address Translation* (NAT) ou caminhos por redes intermediárias. Para os dispositivos externos, as interfaces das VMs são vistas como um nó comum da rede. Já para as comunicações internas, entre VMs, existe um *switch* virtual que efetua a troca de dados entre as instâncias [157]. Dessa forma, pode-se dizer que a interação entre as máquinas virtuais é estabelecida através de uma rede virtual.

A Figura 4.4 ilustra os fluxos de comunicação, interno e externo, em um *hypervisor* configurado com *interface bridge*. O fluxo de comunicação interno é estabelecido através de um *switch* virtual (*vSwitch*), ao qual as VMs estão conectadas. O fluxo de comunicação externa também passa pelo *vSwitch*, porém segue para a placa de rede física - *Network Interface Controller* (NIC) - e alcança o *switch* da rede que se conecta fisicamente a outros computadores (PC1 e PC2). Os dispositivos externos detectam as VMs como sendo nós diretamente conectados ao *switch* físico. A comunicação interna se beneficia do *switch* virtual por não ter que trafegar esses dados através do *switch* externo.

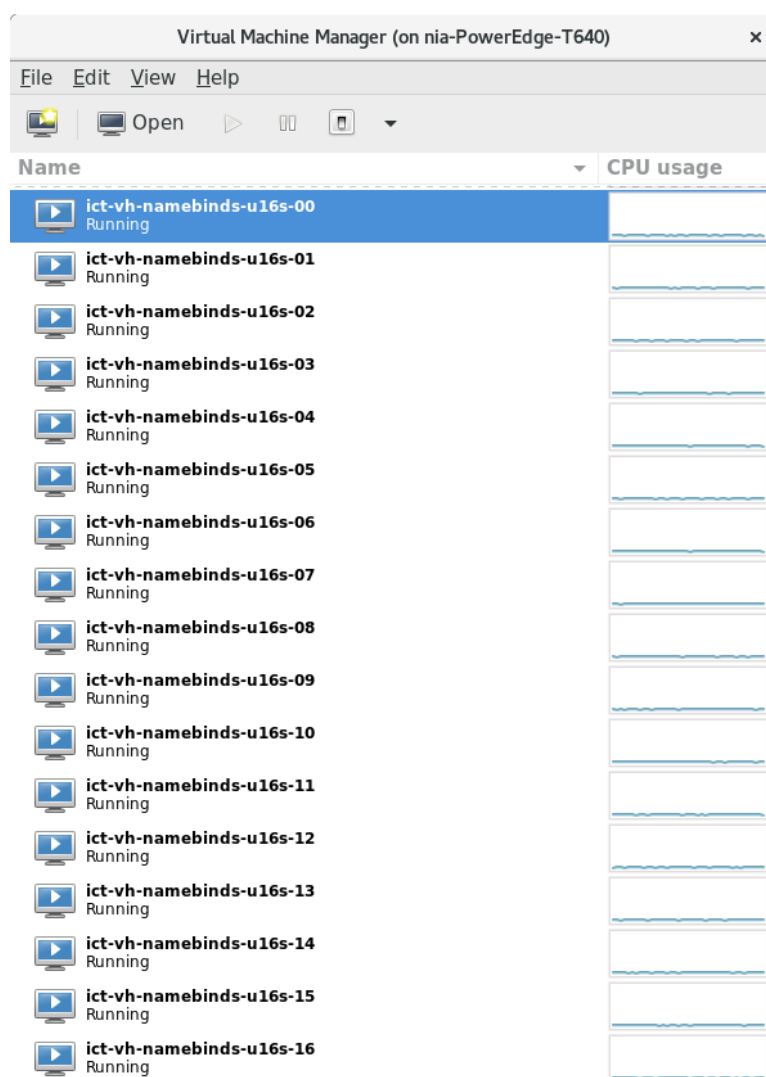


Figura 4.3: Tela de gerência do VMM que exibe as máquinas virtuais e a utilização individual de CPU.

Percebe-se que as abstrações do ambiente virtual unem os recursos de *hardware* através de soluções de *software*, ofertando escalabilidade e gerência otimizadas ao ambiente.

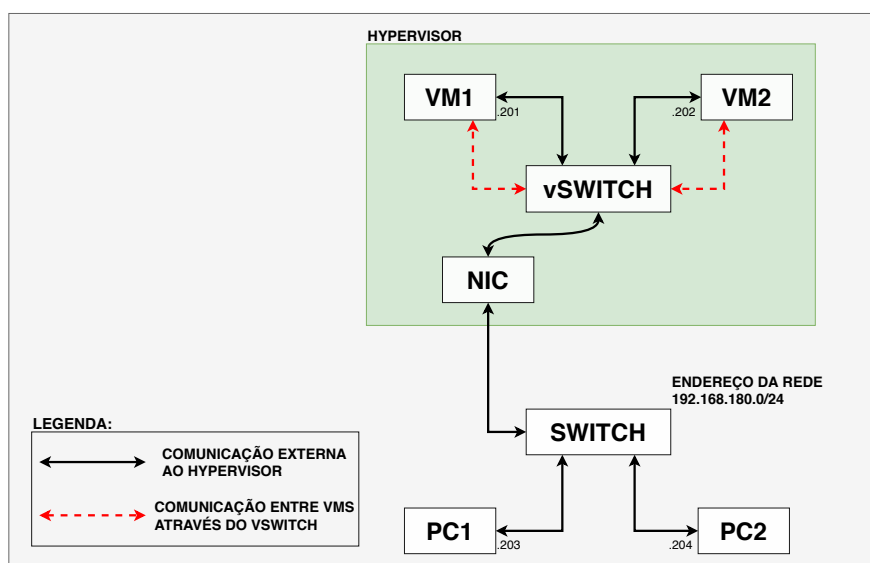


Figura 4.4: Ilustração dos fluxos de comunicação interno e externo de VMs em um hipervisor com interface em bridge.

4.2.3 Recursos de *Software* e Conectividade

Atuando em conjunto com os recursos físicos e virtuais estão os *softwares* selecionados para este trabalho. Essas aplicações foram importantes durante todo o tempo, desde a escolha por uma solução de virtualização, a definição de um sistema operacional, a escolha de ferramentas de automação e controle de configurações, a gerência e conectividade de rede. A motivação para a escolha do sistema de virtualização e SOs foram explicitadas nas Seções 4.2.1 e 4.2.2. A seguir são detalhadas as demais soluções, começando pela ferramenta responsável pelo controle de configurações e automação dos nós, o Ansible. Em seguida é detalhada a solução que garantiu conectividade interna e externa aos nós, o *firewall* virtual PFSense.

O Ansible [158] [159] é um *software* que facilita o controle de múltiplos nós, pois controla a infraestrutura computacional via definições de código [160], sendo capaz de automatizar processos, gerenciar configurações, executar aplicações e *scripts* em múltiplas estações. Essas ações são feitas seguindo um livro de execuções, ou em sua terminologia um *playbook*, que define os nós que devem ser acessados e as rotinas a serem executadas. A execução das rotinas em cada nó se dá de forma distribuída e pode ser ordenada como se desejar. Em uma execução de rotinas paralelas (i.e. atualização de SO) o nó de gerência pode executar um *playbook* de atualização em todos os nós ao mesmo tempo, independente do tempo de execução e a finalização de cada um. Já para uma rotina que deve seguir uma ordem de execução, como é o caso da inicialização da NG, também é possível definir que os nós atuem apenas após a finalização do passo anterior, atuando de forma síncrona. Como exemplos de algumas rotinas de operação de rede sistemas, em que o Ansible é útil, pode-se citar: envio e recebimento de arquivos de múltiplas fontes, troca de dados de configuração, configuração de dependências e execução de *scripts*.

O Ansible também é uma implementação *open source* e possui vasto material de orientação voltado para automação do ciclo de vida de aplicações em ambientes

virtuais, sendo até uma ferramenta de uso comum para essa finalidade [160]. Com a sua utilização neste trabalho intentou-se obter melhorias quanto à diminuição no tempo da execução dos testes, otimização da disponibilidade dos recursos e garantia de homogeneidade da execução. Os *playbooks* e *scripts* utilizados para instanciar os serviços da NG são referenciados no Apêndice A.1.

Para estabelecimento de conexões entre estações de controle, servidores de virtualização, máquinas virtuais e laboratórios de informática, foi necessário estabelecer alguns serviços e configurações avançadas de rede. Vale ressaltar que essas configurações se deram em âmbito de uma rede convencional, pois interligava os equipamentos de controle da experimentação aos ambientes de testes, bem como dispositivos externos. Diagramas e arquivos de configuração estão listados no Apêndice C.1.1.

Os ambientes de experimentação físico e virtual, descritos nas Seções 4.2.1 e 4.2.2, possuem conectividade em camada dois, requisito suficiente para executar a NG. Porém, para implementar as rotinas de automação e operar os ambientes de forma integrada foi necessário estabelecer um controle para a comunicação em camada três (via IP). A solução encontrada para satisfazer essa demanda foi a instalação de um *firewall* que pudesse ser gerenciado e realizasse a função de *gateway* para os ambientes, fazendo principalmente a interface entre a rede isolada da experimentação e a Internet. Decidiu-se então pela utilização do *firewall* pfSense [161], na versão 2.4.4-RELEASE-p3. Como princípio para essa escolha seguiu-se a preferência por adotar sistemas *open source*, que são abertos em sua implementação e flexíveis para receberem modificações. A Figura 4.5 ilustra o *firewall* como elemento central entre os ambientes de experimentação virtual e físico. O endereço da rede privada utilizado é 192.168.180.0/24. A comunicação intraredes não passa pelo *firewall*, ele assume função na topologia quando há a necessidade de encaminhamento de dados entre a rede interna e a Internet. A escolha da sua instalação se justifica pela flexibilidade e controle adicional que ele implementa à camada três dos ambientes, principalmente porque os sistemas de automação e controle se baseiam no endereçamento IP.

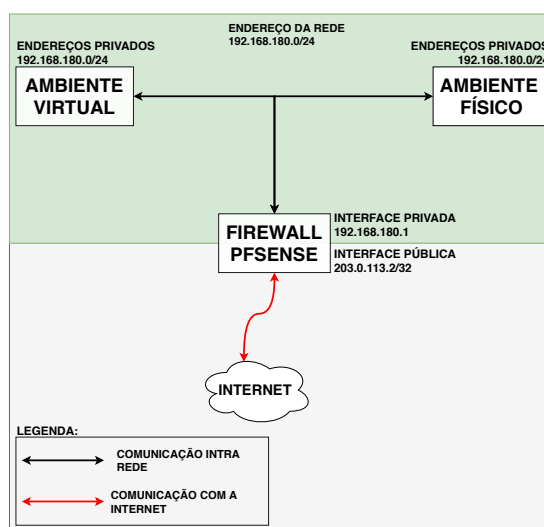


Figura 4.5: Topologia de comunicação, em camada 3, do firewall com os ambientes de experimentação virtual e físico.

4.3 Topologias de Avaliação e Procedimentos Operacionais

Neste trabalho, são utilizados *name bindings* para avaliar a NG. Eles são gerados pela aplicação NBSimpletestAPP, publicados e, então, assinados. Sua função é realizar a troca de NBs com os serviços do núcleo da NG, promovendo a publicação e assinatura de conteúdos nomeados (NBs), garantindo um fluxo de dados para avaliações de desempenho da arquitetura. Os números de publicações e assinaturas podem ser personalizados, conforme arquivo de configuração que é descrito no Apêndice B.5.

A NG possui uma arquitetura voltada a serviços que permite flexibilizar as funções instanciadas de acordo com a finalidade da rede. O objetivo desse trabalho é avaliar o comportamento da arquitetura em sua capacidade de resolução e distribuição de *Name Binds* em domínio local da rede. Para isso, são detalhados quatro cenários/topologias que alternam sua composição através de três tipos de nós. Esses três tipos de nós, variam entre si pelos serviços da NG que executam e são denominados pela função principal que possuem na rede: *Network Core Node* (NCN), *Network Storage Node* (NSN) e *Traffic Generator Node* (TGN). Os cenários são denominados pelo número de nós NSNs que os compõem, sendo que podem ser: 1 NSN, 2 NSNs, 3 NSNs ou 4 NSNs. No decorrer desta seção serão apresentadas as descrições dos tipos de nós e dos cenários de avaliação.

4.3.1 Tipos de Nós

Nesta seção, o termo “nó” é utilizado para denominar uma VM ou um computador que execute a NG. Como já dito, o nó poderá assumir funções diferentes, de acordo com a finalidade que se almeja. Porém, essa definição é exclusiva para entendimento dessa experimentação. A NG possui definições muito mais amplas e flexíveis do que as limitações aqui impostas, em nome da didática e do objetivo experimental.

A inicialização dos serviços da NG em um nó se dá pela execução de seus arquivos executáveis, que coincidem em nome com o serviços que executam: PGCS, GIRS, PSS, HTS e NBSimpletestAPP. Cada executável possui um arquivo de inicialização que contém parâmetros de configuração. Os arquivos utilizados e as explicações desses parâmetros são detalhados no Apêndice A.1.10. Um desses serviços, que é essencial e por isso deve ser executado em todos nós e de forma prioritária, é o PGCS que foi detalhado na Seção 3.5. É importante lembrar que ele é responsável pela identificação dos nós, serviços da NG e suas ligações, seja no nó que está instanciado, através da comunicação intraprocessos, ou no meio, através das comunicações intradomínio. Os tipos de nós que serão detalhados a seguir variam na instanciação dos serviços GIRS, PSS, HTS e NBSimpletestAPP e por isso exercem funções diferentes na rede.

Network Core Node (NCN)

O NCN executa funções de *Core* da rede e é o responsável pela descoberta de outros nós e serviços na rede. Ele é composto pelos serviços: PGCS, GIRS e PSS. Conhecendo as definições desses serviços, sabe-se que as principais funções realizadas pelo nó será a descoberta de outros nós e serviços NG na rede (PGCS), bem como estará apto a

receber publicações e assinaturas de NBs (PSS) e encaminhá-las (GIRS) a um local de armazenamento, externo ao nó. Isso porque nele não é executado nenhum serviço de armazenamento (HTS). Cabe lembrar que a função que se está atribuindo ao nó é em relação ao cenário de avaliação que se quer construir. A opção por instanciar os serviços de publicação, assinatura e resolução de indireções, bem como não instanciar um serviço de armazenamento na rede, fará com que os fluxos de NBs cheguem a esse nó e sejam direcionados para armazenamento em outro nó. Tem-se, portanto, uma concentração de fluxos entrantes e saíntes de dados no nó, o que lhe confere uma característica de centro/núcleo - *Core* - da rede. A Figura 4.6 ilustra uma VM que instancia os serviços do NCN: PGCS, GIRS e PSS. A disposição dos serviços não possui relação com a ordem de comunicação realizada interprocessos, apenas ilustra a presença de tais serviços.

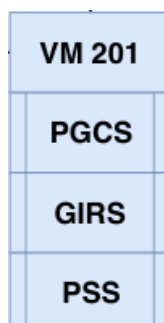


Figura 4.6: Representação dos serviços executados pelo Network Core Node (NCN).

Network Storage Node (NSN)

O NSN possui função de armazenamento na rede, sendo o responsável pelo armazenamento dos NBs, conteúdos que são publicados, assinados e trafegados nesta avaliação. A composição dos serviços nesse nó se dá apenas pelo PGCS e HTS. Assim, percebe-se a manutenção do serviço PGCS, de descoberta de nós e serviços da NG, acompanhado de um serviço HTS, que tem função exclusiva de armazenar os NBs e responder as solicitações de assinaturas destes. Graças a esse serviço, percebe-se então uma forte característica de armazenamento do nó. Merecendo destaque o fato que o HTS é uma tabela baseada em *hash*. Seu espaço de escrita para armazenamento dos NBs é a memória RAM. A Figura 4.7 ilustra uma VM, especificamente um NSN, que instancia os serviços PGCS e HTS.

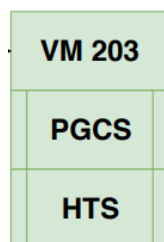


Figura 4.7: Representação dos serviços executados pelo Network Storage Node (NSN).

Traffic Generator Node (TGN)

O TGN tem função de geração de tráfego na rede sendo o responsável pela execução da aplicação que gera o tráfego de dados que serve de base para as avaliações pretendidas. Os serviços NG executados nesse nó são apenas o PGCS e o NBSimpletestAPP. Como nos demais nós, o PGCS é o responsável por detectar outros serviços e nós no domínio. O NBSimpletestAPP foi descrito na Subseção 3.5.3 e na Seção 3.6, mas merece destaque pois sua atuação define a principal função do TGN: O NBSimpletestAPP cria associações de nomes (NBs) aleatórios e publica tais conteúdos na rede. Tais conteúdos são recebidos pela rede até chegarem ao seu local de armazenamento, no HTS do NSN. Em um segundo momento, são solicitadas as assinaturas desses conteúdos para a rede, os HTSes que possuem tal conteúdo se manifestam e enviam então os NBs ao NBSimpletestAPP. Essa é uma síntese do fluxo de publicação e assinatura e refletem a função desse nó nos cenários que serão propostos. A Figura 4.8 ilustra uma VM que representa o TGN, por contemplar os serviços PGCS e o NBSimpletestAPP, na figura abreviado para “NB-APP”.

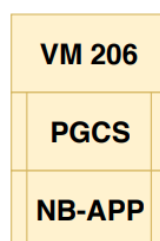


Figura 4.8: Representação dos serviços executados pelo Traffic Generator Node (TGN).

4.3.2 Cenários com Múltiplos NSNs

A investigação dos elementos da NG, como eles se estruturam e das funções que eles executam pode variar de acordo com a composição dos serviços. Uma máquina virtual não precisa executar todos serviços da NG a todo momento. O único que deve ser recorrente é o PGCS, os demais podem ser alterados para atender distintos objetivos. Dada essa possibilidade, os serviços NG utilizados em cada nó foram categorizados em tipos de nós, como visto na Seção 4.3.1. Desta forma, cada nó passa a ser identificado por: NCN, NSN, TGN, que corresponde ao principal serviço executado. A composição de cenários de testes com vários tipos de nós também possibilita avaliações da arquitetura, de forma que outras capacidades e funcionalidades da arquitetura também passam a ser avaliadas. Nesse sentido, chega-se a um dos objetivos deste trabalho, que é avaliar a distribuição do serviço de armazenamento em rede, HTS, de distintas formas e em diferentes quantidades. Para isso foram feitas avaliações com um, dois, três e quatro NSNs.

A variação de NSNs que se propõe não é simplesmente teórica, ela pode significar uma implementação de resolução de nomes que demande: (I) distribuir NBs em regiões geográficas distintas; (II) distribuir a carga de armazenamento ou processamento do sistema de resolução de nomes em vários nós; (III) balancear o acesso ao serviço, seja para publicação ou assinatura de NBs. Alguns objetivos de operação e desenvolvimento da NG também podem ser avaliados, como: (IV) identificar o melhor método de

armazenamento distribuído de NBs; (V) como é o funcionamento da distribuição de NBs entre os serviços da NG e a relação com o número de HTSes presentes; (VI) obtenção de dados para melhoria da arquitetura; (VII) avaliar se os tempos de publicação e assinatura são escalares ou exponenciais; (VIII) avaliar o comportamento da arquitetura com diferentes serviços habilitados e em número variável de instâncias.

Para todos os cenários, a aplicação utilizada para teste é o NBSimpletestAPP. Como explicado anteriormente, na Subseção 3.5.3 e na Seção 3.6, ela permite que a arquitetura publique e assine NBs. Como o objetivo é entender a capacidade da NG em atuar de forma distribuída, a aplicação que gera e recebe o tráfego é base para a observação desejada. O tráfego gerado é baseado em NBs, como abordado na Seção 3.4.1. Para variar os cenários de experimentação são utilizados os três tipos de nós descritos na Seção 4.3.1, de forma que são definidos quatro cenários, que variam entre si o número de NSNs. O número total de nós varia de dois a seis, isso porque que não foi possível expandir o número de nós no mesmo domínio. Haja vista que, na época, existia uma limitação de faltar um controle de transmissão de pacotes no atual protótipo da NG, algo que seria parecido com o TCP e que permitiria fazer retransmissões das comunicações que, porventura, fossem perdidas pelas camadas inferiores da rede.

Outros serviços da NG têm potencial para serem avaliados e melhor entendidos na prática, mas o objetivo deste trabalho é alcançado de uma forma mais objetiva sem múltiplas variações de número de nós e serviços.

Cenário 1: 1 NSN

O primeiro desses cenários é composto por um nó que realiza a função de *core* e armazenamento de forma mista, contendo apenas um HTS e possuindo uma especificidade: o serviço HTS, função de armazenamento característica do NSN, é executado dentro do NCN, de forma que esse nó executa os seguintes serviços: PGCS, GIRS, PSS e HTS. O outro nó do cenário é o TGN. Este é executado sem excepcionalidades: seus serviços são o PGCS e o NBSimpletestAPP. O intuito de definir essa topologia é poder estabelecer avaliações de publicação e assinatura de NBs com apenas dois nós e poder gerar dados para avaliar aplicações que demandem uma topologia reduzida. A Figura 4.9 ilustra duas VMs e os serviços que executam. Como dito, a VM 201 acumula a função do NCN e do NSN. A VM 206 é responsável apenas pela função de geração de tráfego (TGN). A representação de interligação entre as duas é para ilustrar que estão no mesmo domínio e se comunicam, uma explicação sobre o fluxo de publicação e assinatura de conteúdos na NG foi feita na Subseção 3.6.1.

Cenário 2: 2 NSNs

Expandindo o primeiro cenário, cogitou-se distribuir NSNs na rede local, em diferentes VMs ou máquinas físicas. Essa distribuição implicaria em maior resiliência e redundância do serviço HTS, podendo atender situações em que apenas uma instância não pudesse suprir a capacidade de armazenamento e distribuição de conteúdos de um domínio. De forma a realizar essa avaliação, definiu-se um cenário que é composto por quatro nós, que executam as funções de *core* (NCN), geração de tráfego (TGN) e armazenamento (NSN). Diferentemente do cenário anterior, não existe nenhum HTS

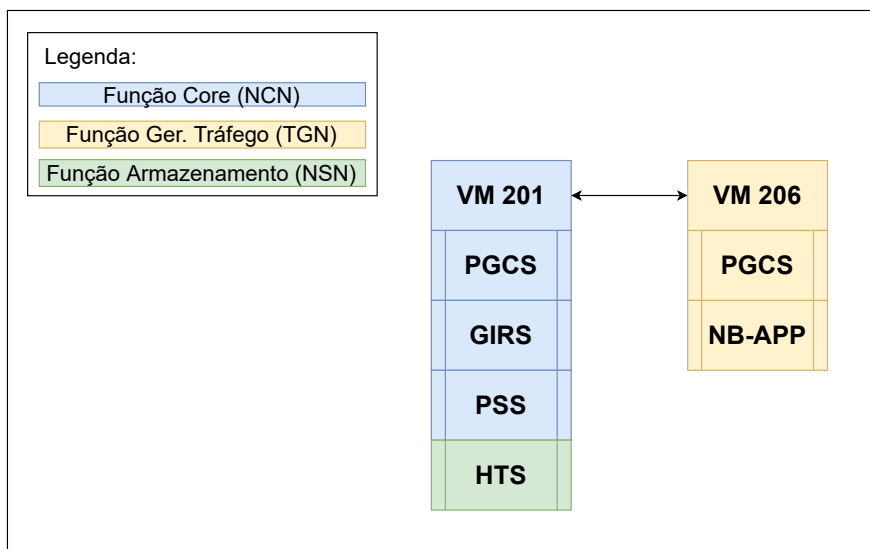


Figura 4.9: Representação do cenário contendo um HTS. A VM 201 acumula a função do NCN e do NSN, de forma excepcional.

atrelado à instância que realiza a função *core*. As duas VMs que hospedam os NSNs executam, adicionalmente, apenas o PGCS, que é o serviço de conectividade da NG. A Figura 4.10 ilustra os nós que compõem o cenário. Eles são quatro, sendo que um executa a função de *core* (NCN), dois executam a função de armazenamento (NSN) e um executa a função de geração de tráfego (TGN).

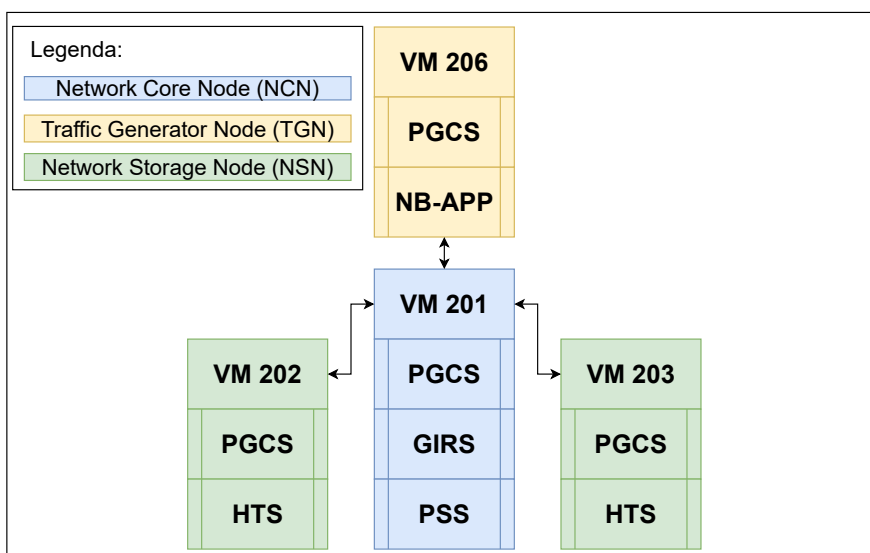


Figura 4.10: Diagrama representativo para cenário contendo dois HTSes. Primeira abordagem em que é possível observar a distribuição de cargas do TGN a outros NCNs.

Cenário 3: 3 NSNs

Graças à auto-organização distribuída dos serviços da NG é possível continuar adicionando e variando o número de instâncias virtuais e serviços presentes na rede. Como um dos focos do estudo é avaliar o armazenamento de NBs, realizado pelo serviço

HTS, presente no NSN, foram adicionadas mais instâncias deste, de forma a possibilitar a expansão e análise do serviço de armazenamento e *cache* da arquitetura. A Figura 4.11 ilustra o cenário 3 NSNs que é composto por três NSNs, um NCN e outro nó que realiza a função de geração de tráfego, o (TGN). O principal objetivo de expandir a distribuição dos conteúdos nomeados em múltiplos nós é avaliar a escalabilidade dos serviços NG e possíveis pontos que demandem melhorias.

Cenário 4: 4 NSNs

Como último cenário, um quarto nó de armazenamento é adicionado à rede, com intuito de avaliar a escalabilidade e se os tempos de publicação e assinatura de NBs seriam escalares ou exponenciais. Na Figura 4.12 é possível ver o número máximo de nós que foram utilizados no experimento. Ao todo são seis instâncias, sendo quatro para a função de armazenamento e cache (NSN), um para coordenar a comunicação entre os nós (NCN) e um executando a aplicação geradora de tráfego (TGN).

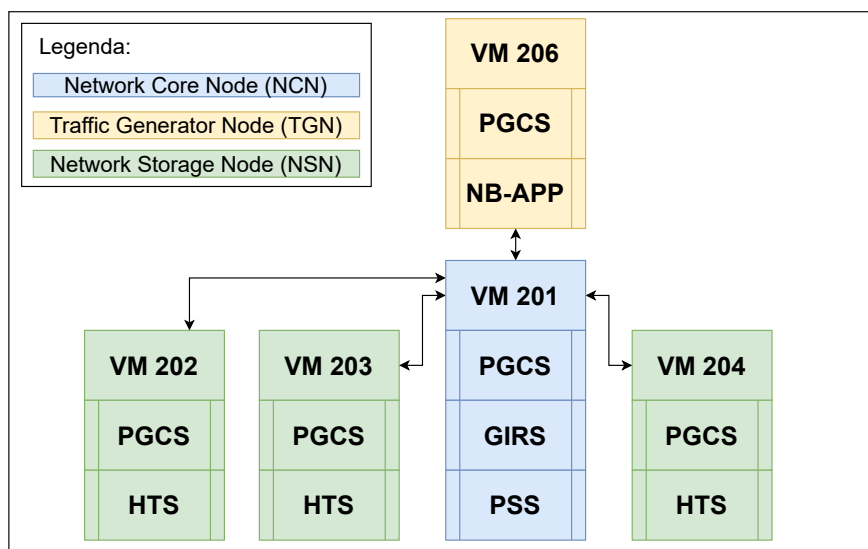


Figura 4.11: Diagrama representativo para cenário contendo três HTSes. Incremento no número de nós aptos a armazenar os NBs.

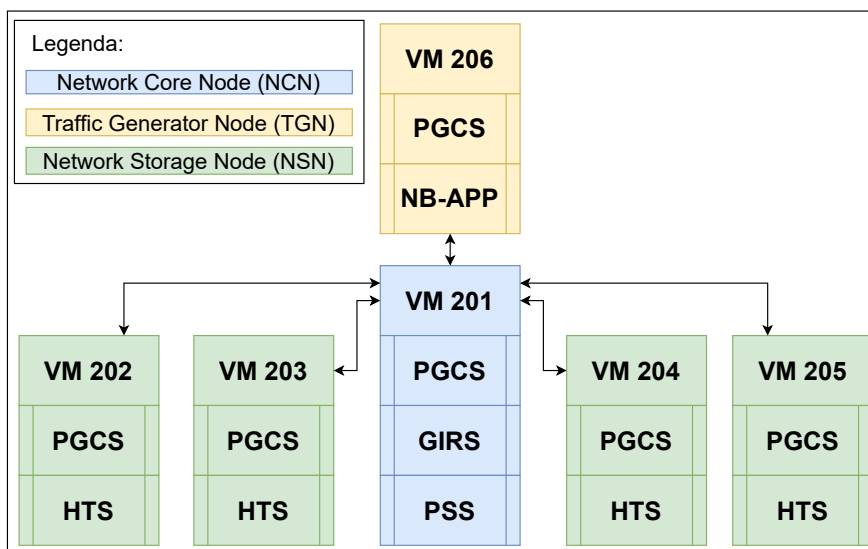


Figura 4.12: Diagrama representativo para cenário contendo quatro HTSes. Número máximo de instâncias avaliadas. A distribuição de elementos de armazenamento também é a maior entre todos os demais cenários.

Variação de Publicação e Assinatura de NBs

De forma a avaliar os cenários em diferentes circunstâncias, a aplicação teve o parâmetro de número de publicações de NBs variando de 1 a 32 milhões, com incremento em escala quadrática. O número de assinaturas obtido foi sempre constante e definido para 360. Tais parâmetros são configurados no arquivo de inicialização do NBSimpletestAPP e foram detalhados no Quadro 3.2.

4.4 Procedimentos Operacionais

Os procedimentos operacionais desta experimentação estão fortemente ligados à execução de comandos GNU/LINUX, configuração de conectividade via redes e gerência de sistemas. Como abordado anteriormente, algumas ferramentas de automação, como o Ansible, foram utilizadas. Elas tiveram importante destaque na padronização das execuções dos testes. Aliadas aos *scripts* e *playbooks*, proveram condições de instanciar os serviços da NG de uma forma controlada e distribuída.

De uma forma macro, os principais procedimentos operacionais foram: Preparação do Ambiente, Execução da NG, Coleta de Dados, Reinicialização e Manutenção.

- A Preparação do Ambiente é um procedimento que, em condições normais, é executado apenas uma vez para cada nó. Nessa etapa é feita a instalação dos sistemas operacionais, são instaladas as dependências de softwares e é feita a configuração de conectividade.
- A Execução da NG é o procedimento que instancia cada um dos serviços da arquitetura, na ordem e em quantidades que se deseja avaliar.
- A Coleta de Dados é realizada após cada teste, sendo ela a etapa que arquiva, em local seguro e fora dos nós, os *logs* de execução da sessão.

- A Reinicialização e Manutenção é o procedimento em que é feita uma avaliação do estado dos nós. Caso algum nó não retorne ao estado funcional, uma manutenção individual deve ser realizada. Caso esteja tudo normal, eles são apenas reiniciados.

Os últimos três procedimentos fazem parte de um *loop* que permite a execução de vários testes consecutivos da arquitetura. A Figura 4.13 ilustra esse fluxo de procedimentos macros. Como cada uma dessas etapas é composta por atividades mais específicas e únicas, uma visão individual é feita para cada uma delas nas subseções seguintes.

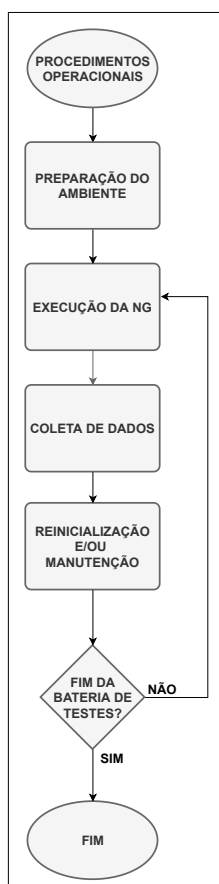


Figura 4.13: Diagrama representativo dos procedimentos macros executados para avaliação da NG.

4.4.1 Preparação dos Ambientes

Os ambientes de experimentação são dois: físico e virtual. Para os dois casos as ações que preparam os computadores ou máquinas virtuais são comuns, sendo elas: Instalação do Sistema Operacional, Instalação de Dependências e Aplicações de Suporte e Cópia da NG para o nó, como ilustra a Figura 4.14 e é explicitado a seguir.

A instalação do sistema operacional difere um pouco de como é feita para cada um dos ambientes, porém o SO de ambos é o Ubuntu 16.04 LTS. Na instalação do SO nos computadores físicos foi necessário configurar o *dual boot* para funcionar em paralelo

com o Windows. Para as máquinas virtuais o procedimento de configuração do SO foi mais simples, pois foi possível fazer a configuração de apenas uma VM e cloná-la para criar as demais instâncias. Um cuidado adicional que se teve nesse caso foi o de garantir que o *Media Access Control address* (MAC address) e o nome de cada um dos nós fossem diferentes.

A instalação de dependências é o procedimento que visa garantir funcionalidades de execução, monitoramento e controle do ambiente. Isso é possível através da instalação de bibliotecas necessárias à NG e dos programas que permitem o suporte e gerência de toda a experimentação. O Apêndice A.1.9 exibe o *playbook* do Ansible respectivo a essa ação, nele são discriminadas as bibliotecas e os programas complementares utilizados nos ambientes de experimentação.

O procedimento de copiar a NG também é estabelecido através de uma rotina do Ansible. O *playbook* que consta no Apêndice A.1.7 é o responsável por tal rotina. Essa rotina executa, de uma só vez, a transferência dos executáveis da NG aos nós que executarão os testes. A vantagem de se realizar esse procedimento pelo Ansible é a garantia de que ele será executado de forma homogênea e sem a necessidade de acompanhar a execução em cada nó.

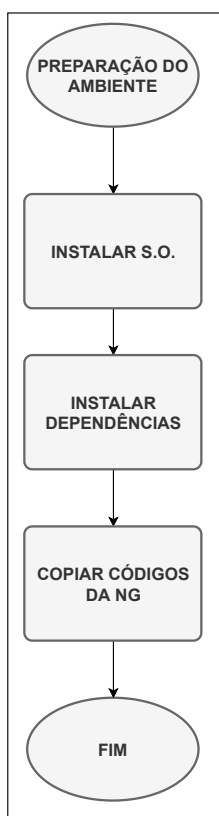


Figura 4.14: Diagrama representativo dos procedimentos de preparação dos ambientes de experimentação.

4.4.2 Execução da NG

A execução da NG é o procedimento que mais se beneficia da automação do Ansible, isso porque a inicialização dos serviços e dos nós deve seguir uma ordem. Quando existem múltiplos nós e diferentes cenários de avaliação, a execução da NG de forma manual pode se tornar um ponto de falha. De forma prática, os procedimentos que iniciam a avaliação dos cenários são: Inicialização do NCN, Inicialização dos NSNs, Inicialização do TGN, Armazenamento dos Resultados de Publicação e Assinatura de NBs. Esse fluxo de ações é ilustrado na Figura 4.15 e descrito em detalhes a seguir.

O NCN é inicializado primeiro porque ele provê o serviço de interconexão com os demais nós do ambiente. Sua inicialização é feita pelo *playbook* do Ansible, Apêndice A.1.1, nele é identificado o nó e o *script* que será executado. O *script*, que é detalhado no Apêndice A.1.2, corresponde aos comandos e parâmetros que executam os serviços PGCS, GIRS e PSS da NG no nó NCN, seja ele físico ou virtual.

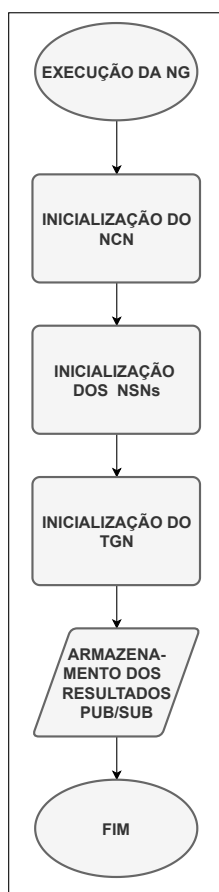


Figura 4.15: Diagrama representativo dos procedimentos de execução da NG.

A inicialização dos NSNs é feita de forma parecida, porém o número de instâncias pode variar de um a quatro. O *playbook*, Apêndice A.1.1, identifica os nós e os *scripts* que serão executados em cada uma dessas instâncias. O *script* de inicialização desse tipo de nó contém os comandos e parâmetros para inicialização dos serviços PGCS e HTS e ele pode ser consultado no Apêndice A.1.3.

O TGN é o último a ser inicializado, devido ao fato dele ser o gerador de tráfego e

elemento de avaliação da arquitetura. Por isso espera-se que todos os demais serviços estejam prontos e operacionais antes da sua inicialização. O mesmo *playbook* que inicia os nós anteriores também inicia esse, porém executando um *script*, tal qual listado no Apêndice A.1.1, que inicia apenas os serviços PGCS e NBSimpletestAPP. De forma a aguardar e garantir que o PGCS de todos os demais nós estejam operacionais, é inserido um tempo mínimo de espera antes do início do NBSimpletestAPP.

Por último, após execução das publicações e assinaturas do NBSimpletestAPP, são mantidos nos TGNs os *logs* de execução dos testes. Eles são compostos dos tempos e estatísticas de publicação e assinatura dos NBs, sendo armazenados no disco, a fim de serem coletados pelo fluxo seguinte, a Coleta de Dados.

A Execução da NG é o fluxo de procedimentos que mais se repete durante a experimentação. Por esse motivo é importante destacar o tempo aproximado que é consumido para os diferentes números de NBs publicados. O Quadro 4.4 registra tal informação, sendo que a publicação de um milhão de NBs é feita em aproximadamente quinze minutos, enquanto a publicação de trinta e dois milhões de NBs é feita em aproximadamente vinte e quatro horas.

Quadro 4.4: *Duração aproximada do fluxo de Execução da NG para diferentes quantitativos de NBs.*

Quantidade de NBs [milhão]	Duração
1	15 minutos
2	30 minutos
4	1 hora e 30 minutos
8	3 horas
16	6 horas
32	12 horas

Fonte: Elaborado pelo autor, 2021.

4.4.3 Coleta de Dados

A Coleta dos Dados de execução do NBSimpletestAPP é uma etapa importante da experimentação, pois são esses dados que possibilitam a análise da NG. Após cada rodada de testes é feito o *download* dos *logs* de publicação e assinatura de NBs dos Nós APP para o Nó de Controle. Na Figura 4.16 é ilustrado o único procedimento que é executado nessa rotina. Ele consiste em executar um *playbook* do Ansible, detalhado no Apêndice A.1.8, que se encarrega de baixar os *logs* dos múltiplos nós de aplicação ao nó de controle. Novamente, a execução de uma rotina distribuída e automatizada contribui para a garantia de automatização de tarefas recursivas e a redução de possíveis erros de execução manual.

Para composição dos resultados que são analisados no próximo capítulo foram utilizadas dez execuções para cada quantitativo de NBs. Consequentemente, para os seis níveis de NBs foram aproveitadas sessenta execuções. Como a análise consiste em avaliar a publicação e assinatura, para os dois cenários, o total de execuções aproveitadas foi de cento e vinte. Um número maior de testes foi realizado, porém, todos esses dados não foram aproveitados por estarem incompletos ou resultarem de execuções interrompidas. Isso aconteceu principalmente nas experimentações mais demoradas executadas nos laboratórios de informática.

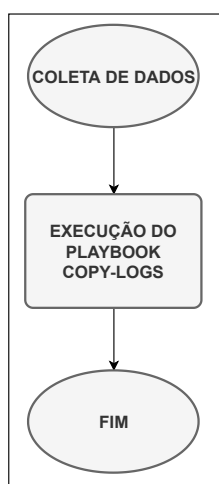


Figura 4.16: Diagrama representativo dos procedimentos de coleta de dados da experimentação.

4.4.4 Reinicialização e Manutenção

A Reinicialização e Manutenção do ambiente de experimentação é um fluxo de procedimentos importantes para a preservação da homogeneidade dos testes. Ela garante que não haja resquícios das execuções dos programas experimentados entre uma sessão de testes e outra.

A Manutenção é a verificação de que os nós se mantenham operacionais e íntegros. Os procedimentos executados nessa rotina são ilustrados na Figura 4.17, eles são iniciados com uma verificação de erros de execução dos nós e possuem dois procedimentos: Verificação e correção dos nós e Execução do *playbook* de reinicialização. O primeiro só é executado caso tenha sido detectado algum erro durante a execução da NG. O segundo é executado a cada finalização de bateria de testes.

O fluxo de Reinicialização e Manutenção se inicia fazendo uma avaliação, não automatizada, quanto a possíveis erros de execução dos nós. Tais erros podem ser: indisponibilidade do nó, não execução dos serviços NG e/ou não execução dos comandos de gerência do Ansible. Apesar de não serem indisponibilidades detectadas de forma automatizada, elas são registradas pelos *logs* de execução dos *playbooks* do Ansible. A correção de tais indisponibilidades é feita de forma pontual, acessando nó a nó e fazendo os ajustes que forem necessários. A primeira ação que normalmente é realizada é a execução do *playbook* de Instalação de Dependências, citado na Subseção 4.4.1 e detalhado no Apêndice A.1.9. Caso não seja suficiente para restabelecer o nó às experimentações, acessos individuais são feitos e a manutenção da instância segue até que o nó seja restabelecido, seja uma máquina virtual ou um computador físico. Após o restabelecimento das funções do nó, ele estará apto a seguir operando no ambiente.

O procedimento de reinicialização dos nós é executado por um *playbook* que é detalhado no Apêndice A.1.6. A execução desse procedimento garante atuação ampla e distribuída, sendo possível economizar tempo e garantir que os nós desejados sejam reinicializados de forma uniforme.



Figura 4.17: Diagrama representativo dos procedimentos de reinicialização e manutenção do ambiente de experimentação.

4.5 Entendimentos do Capítulo

Neste Capítulo são apresentadas as técnicas para avaliação da NG em diferentes ambientes. A motivação que justificou o estabelecimento de tais ambientes em infraestrutura próxima aos pesquisadores foi a necessidade de, ainda, manter muitas personalizações nos nós que executam a arquitetura, além do fato de manter parâmetros de comparação confiáveis e próximos às avaliações já realizadas. São apresentados também os ambientes, físico e virtual, que receberam tais experimentações além de serem explicitadas as personalizações realizadas para que eles suportassem as experimentações. O foco é na confiabilidade e reprodutibilidade dos testes e seus resultados. Quanto às avaliações da NG, são detalhados os tipos de nós e cenários que são estabelecidos para estressar o serviço de publicação e assinatura de conteúdos nomeados da arquitetura (NBSimpletestAPP). Aqui a preocupação é de que o método desenvolvido seja fiel a prova do problema que se quer resolver com a NovaGenesis. O entendimento dessa experimentação não é completo sem as explicações que são feitas sobre os procedimentos operacionais dos testes, sobre os fluxos das rotinas e o detalhamento do processo utilizado para iniciar os serviços da NG.

Capítulo 5

Resultados e Análises

Neste capítulo são apresentados os resultados e avaliações obtidas através das experimentações realizadas da NovaGenesis em ambientes físico e virtual. Os experimentos consistiram em avaliar a publicação e assinatura de *name bindings* nos dois ambientes. Por sua vez, cada ambiente foi avaliado em quatro cenários, diferenciados pelo número de serviços *Hash Table Service* (HTSes) que possuíam distribuídos na rede: 1, 2, 3 e 4 HTSes. São apresentados, individualmente, os gráficos para as respectivas publicações e assinaturas de forma a permitir comparações entre os ambientes e seus cenários. Os gráficos também são compilados por tipo de comunicação, publicação ou assinatura, e por ambiente, físico ou virtual.

Três seções agrupam os resultados e avaliações. A primeira corresponde ao ambiente físico. A segunda apresenta os resultados e avaliações para o ambiente virtual. Por último, uma seção dedicada à avaliação conjunta dos ambientes físico e virtual é apresentada. Para todos os experimentos o arquivo de inicialização padrão foi o citado na Tabela 3.2, sendo que com exceção do parâmetro que varia o número de publicações, *NumberOfPublications*, os demais foram comuns. Assim, o número de publicações era variado de acordo com o experimento, mas o número de assinaturas se manteve comum em 360 assinaturas.

As principais características que diferenciam cada resultado são:

- Tipo de Ambiente: físico ou virtual.
- Cenário de experimentação quanto ao número de NSNs presentes: 1, 2, 3 e 4.
- Tipo de comunicação: publicação ou assinatura de *name bindings*.
- Variação no número de NBS: 1 a 32 milhões.
- Tempo de ida e volta da comunicação - *Round Trip Time* (RTT).
- Margem de erro das medições de RTT, cujo cálculo foi detalhado na Subseção 3.5.3.

5.1 Ambiente Físico

Nesta subseção são apresentados os resultados das avaliações da NG em ambiente físico. O ambiente que recebeu as experimentações foi descrito na Seção 4.2.1. Para

permitir uma avaliação detalhada, os resultados são exibidos de forma individual para cada cenário. Os mesmos que foram definidos na Subseção 4.3.2, e para os métodos de comunicação, descritos na Subseção 3.4.2. Desta forma, os possíveis cenários são: 1 NSN, 2 NSNs, 3 NSNs e 4 NSNs para publicação ou assinatura de NBs. Ao final da apresentação dos gráficos individuais é feita uma unificação dos resultados, distinguindo os métodos de comunicação, publicação e assinatura, de forma a permitir comparações entre os cenários.

5.1.1 Publicação de 1000 NBs por Mensagem

A Figura 5.1 apresenta os resultados de publicações de NBs para o Cenário 1 NSN. O eixo horizontal representa a variação de NBs publicados, o vertical representa o RTT médio. Por esses dados é possível observar um crescente aumento no RTT à medida que se aumentam o número de NBs publicados ao todo, fato que pode ser associado ao aumento no número de NBs que é enviado e processado pelo nó.

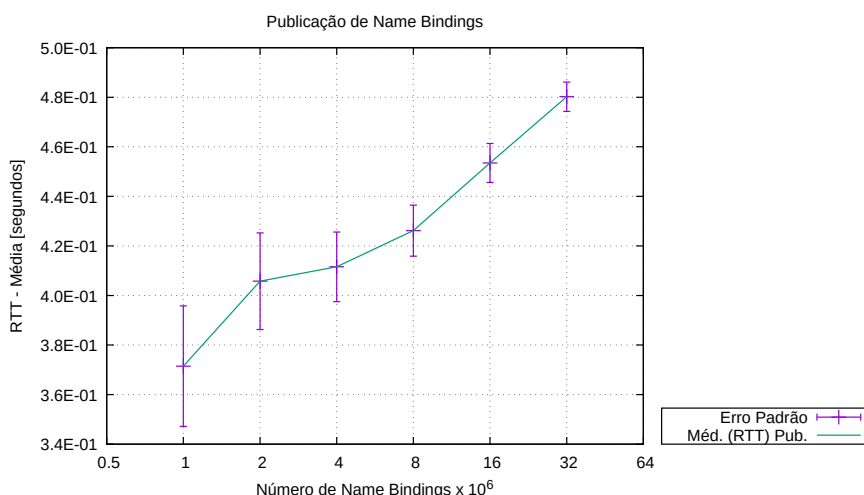


Figura 5.1: *Publicação de NBs no ambiente físico para o Cenário 1 NSN.*

A Figura 5.2 apresenta os resultados de publicações de NBs para o Cenário 2 NSNs. O eixo horizontal representa a variação de NBs publicados ao todo. O vertical representa o RTT médio das mensagens de publicação que levam 1000 NBs cada. Por esses dados é possível observar que mesmo com o aumento no número de NSNs o tempo RTT aumenta com o número de publicações de NBs.

A Figura 5.3 apresenta os resultados de publicações de NBs para o Cenário 3 NSNs. O eixo horizontal representa a variação de NBs publicados ao todo, o vertical representa o RTT médio das mensagens de publicação contendo 1000 NBs cada. Por esses dados é possível observar que existe uma tendência de aumento no Round Trip Time em decorrência do aumento no número de publicação de NBs, porém, para o caso de 32 milhões de NBs há um aumento considerável em relação aos pontos anteriores. Tal fato pode indicar uma possível ocupação dos serviços que distribuem as publicações na rede, como o GIRS.

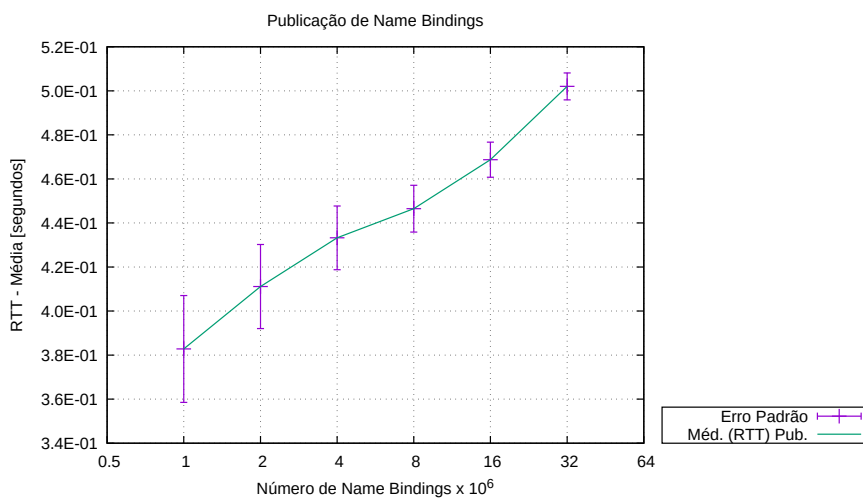


Figura 5.2: Publicação de NBs no ambiente físico para o Cenário 2 NSNs.

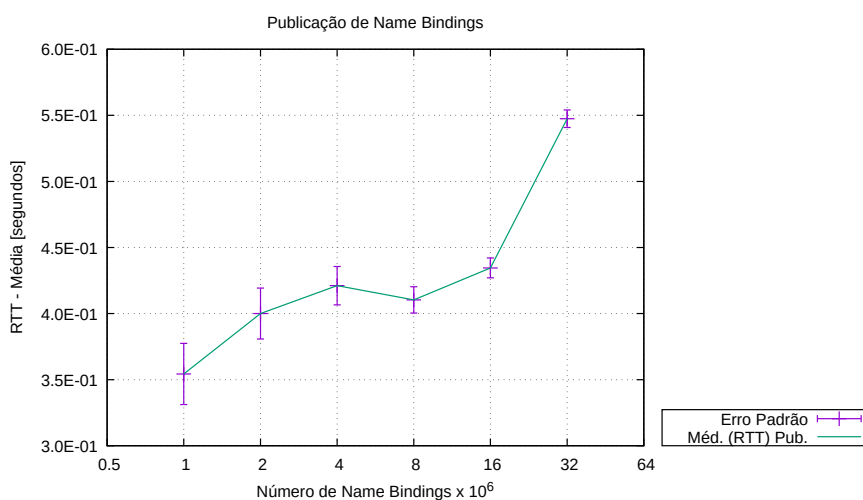


Figura 5.3: Publicação de NBs no ambiente físico para o Cenário 3 NSNs.

A Figura 5.4 apresenta os resultados de publicações de NBs para o Cenário 4 NSNs. O eixo horizontal representa a variação de NBs publicados, o vertical representa o RTT médio. Por esses dados é possível observar que ocorre um pequeno aumento no RTT para o caso de 2 milhões de NBs, porém tal comportamento não se propaga para os outros casos. Atribui-se isso a uma possível sobrecarga dos serviços NG, uma vez que o intervalo de confiança também aumentou nesse ponto.

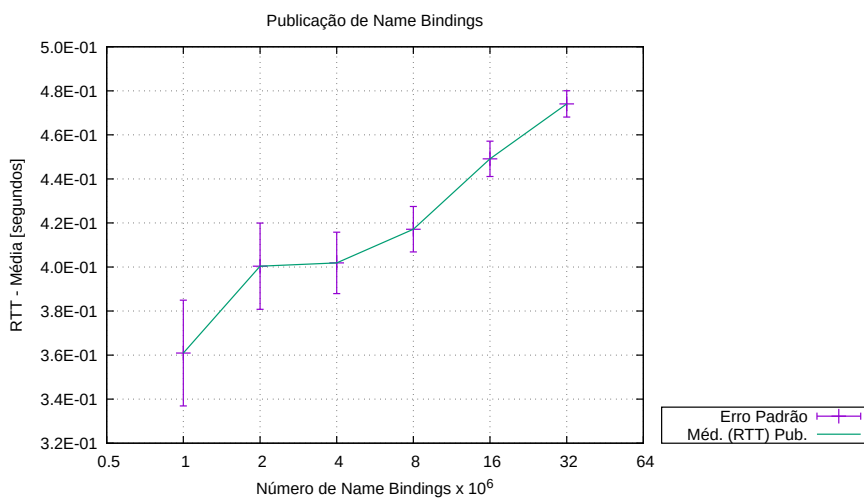


Figura 5.4: *Publicação de NBs no ambiente físico para o Cenário 4 NSNs.*

5.1.2 Assinatura de um Único NB Aleatório por Segundo

A Figura 5.5 apresenta os resultados de assinaturas de NBs para o Cenário 1 NSN. O eixo horizontal representa o número de publicações de NBs realizadas antes de serem solicitadas as assinaturas desses registros de relação entre nomes; o eixo vertical representa o RTT médio de uma única assinatura individual dentro de todo conjunto publicado. Por esses resultados é possível observar que há uma tendência de aumento no RTT médio, mas ela é mínima, ocorrendo inclusive oscilações em que o RTT médio de uma experimentação com mais publicações, 8 milhões de NBs, é menor do que uma com menos publicações, 4 milhões de NBs. Isso mostra que a assinatura de um único NB aleatório dentro de todo o conjunto publicado é bastante escalável, não sofrendo grande impacto pelo aumento na quantidade de NBs publicados ao todo.

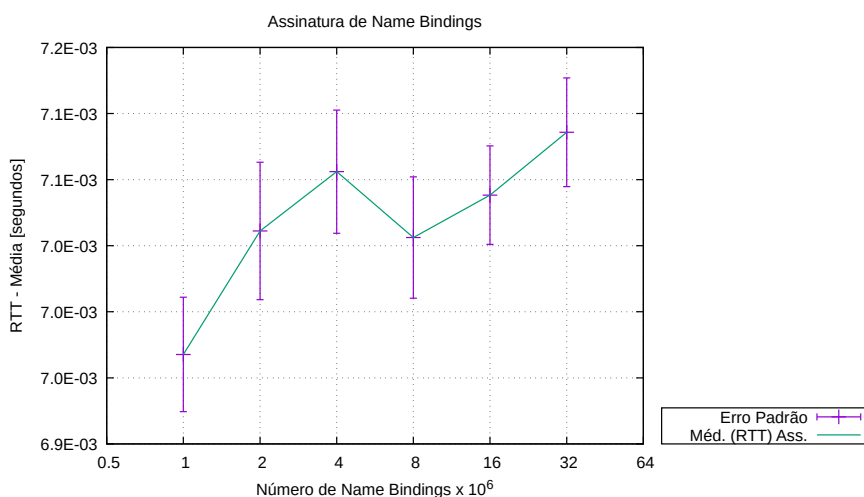


Figura 5.5: *Assinatura de NBs no ambiente físico para o Cenário 1 NSN.*

A Figura 5.6 apresenta os resultados de assinaturas de NBs para o Cenário 2 NSNs. O eixo horizontal representa o número de publicações de NBs realizados antes de serem solicitadas as assinaturas de NBs. O eixo vertical representa o RTT médio das 360

assinaturas. Por esses dados é possível observar que nos níveis de 16 e 32 milhões de NBs parece haver uma estabilização no RTT, possivelmente devido ao número maior de NSNs na rede e a capacidade de distribuição da carga de assinatura entre os diferentes nós.

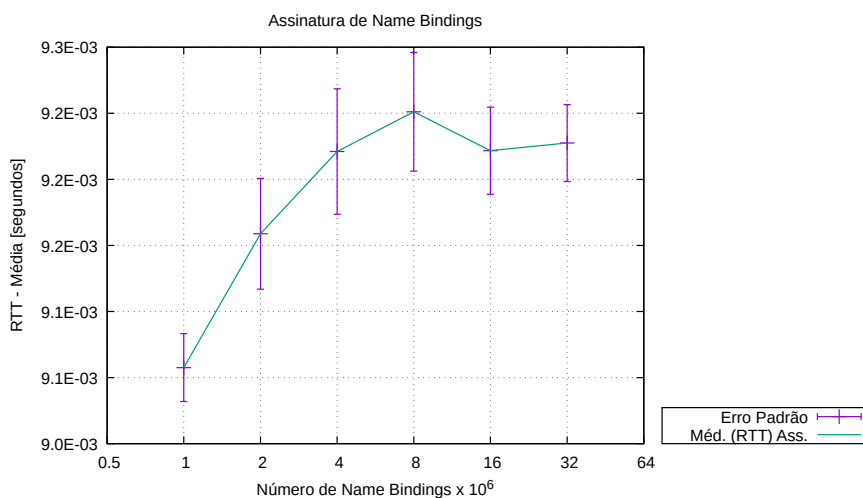


Figura 5.6: Assinatura de NBs no ambiente físico para o Cenário 2 NSNs.

A Figura 5.7 apresenta os resultados de assinaturas de 360 NBs para o Cenário 3 NSNs. O eixo horizontal representa o número de publicações de NBs realizadas antes de serem solicitadas as assinaturas de NBs, o eixo vertical representa o RTT médio das 360 assinaturas. Por esses dados é possível observar uma acentuada redução do RTT médio com o aumento do número de publicações. Tal fato pode estar relacionado ao aumento no número de NSNs disponíveis para responderem as solicitações de assinaturas.

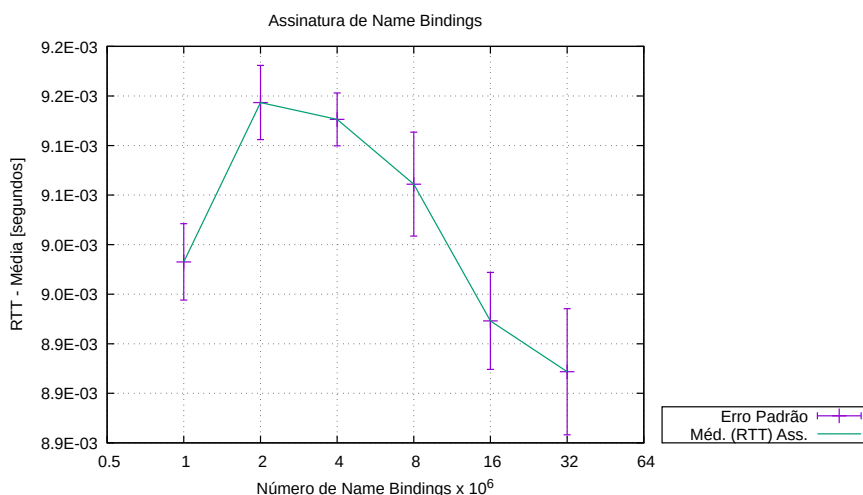


Figura 5.7: Assinatura de NBs no ambiente físico para o Cenário 3 NSNs.

A Figura 5.8 apresenta os resultados de assinaturas de 360 NBs para o Cenário 4 NSNs. O eixo horizontal representa o número de publicações de NBs realizadas antes de serem solicitadas as assinaturas de NBs. O eixo vertical representa o RTT médio das 360 assinaturas. Por esses resultados é possível observar que existem dois picos opostos de RTT: o maior do gráfico, em 2 milhões de NBs, e o menor, em 4 milhões de NBs.

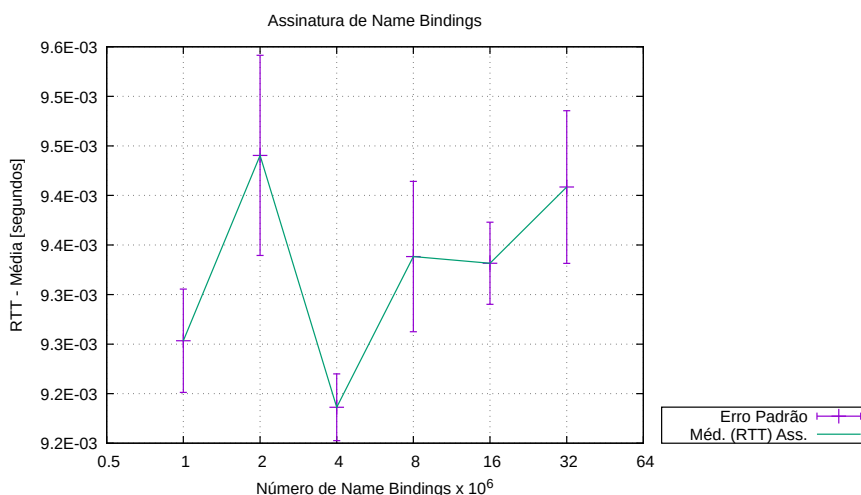


Figura 5.8: Assinatura de NBs no ambiente físico para o Cenário 4 NSNs.

5.1.3 Comparação de Cenários com Máquinas Físicas

A Figura 5.9 apresenta um comparativo dos resultados das publicações nos quatro cenários de experimentação com computadores físicos. As curvas são diferenciadas pelo número de serviços HTSes presentes em cada cenário, resultantes da variação dos cenários de um a quatro NSNs, descritos na subseção 4.3.2. Nela é possível perceber que existe uma variação crescente de tempo ao se aumentar o número de publicações gradativamente. Observa-se que o cenário com maior número de NSNs apresenta tempo de resposta menor. A avaliação com quatro NSNs apresentou resultados de tempo menores do que as implementações de um e dois NSNs para todos os números de publicações. A experimentação com três NSNs apresentou uma variação nos níveis de 4 e 32 milhões de publicações, obtendo resultados até melhores que a abordagem com quatro NSNs nos níveis de 8 e 16 milhões de publicações, porém teve o pior desempenho para 32 milhões de publicações ao todo.

Em uma comparação binária, é possível perceber que a distribuição do serviço HTS, para o cenário de dois NSNs, resultou em um aumento no tempo de publicação, se comparada com a implementação em que o serviço HTS é executado dentro do nó NCN da rede e com apenas uma instância. O cenário com três NSNs oscila entre todos os demais, não apresentando uma curva estável, mesmo considerando as margens de erro. Partindo do comportamento da curva do cenário de um NSN e comparando-a com as demais, de forma individual, é possível identificar que apenas o cenário de quatro NSNs obteve menor tempo de resposta do que ela em todos os níveis de publicações.

Diante dessas informações, o autor conclui que a implementação distribuída dos NSNs só diminui o tempo de resposta, na execução de publicação de conteúdos, para implementações com mais de quatro nós. O autor também conclui que a desassociação do serviço HTS do NCN e a sua distribuição em dois nós independentes apresenta rendimento inferior à execução mais básica, com apenas um serviço HTS em execução dentro do NCN.

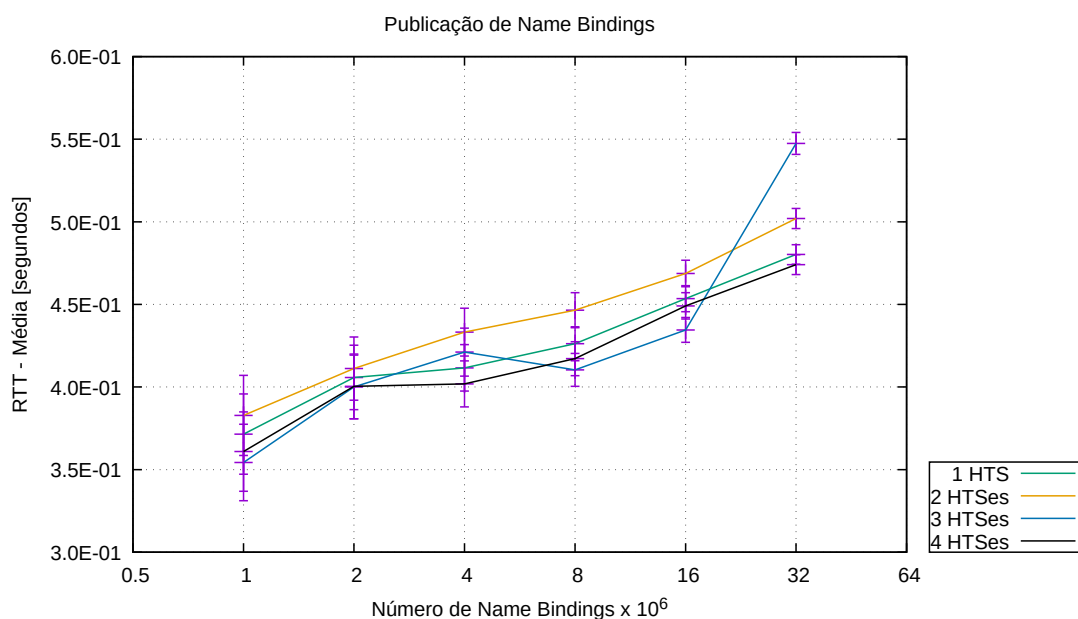


Figura 5.9: Comparação da publicação nos 4 cenários de laboratório.

A Figura 5.10 apresenta um comparativo dos resultados de assinatura nos quatro cenários de experimentação em laboratórios de informática. As curvas são diferenciadas pelo número de serviços HTSes presentes em cada cenário, resultantes da variação dos cenários de um a quatro NSNs, descritos na subseção 4.3.2. Nela é possível perceber uma destacada diferença no desempenho entre a implementação com um NSN e as demais. A curva para o cenário de dois NSNs se estabelece próxima às curvas do cenário de três e quatro NSNs. Ela apresenta resultado ligeiramente melhor que as demais apenas no nível de 2 milhões de assinaturas. O cenário de três NSNs apresenta o menor tempo de resposta nos demais níveis: 1, 4, 8, 16 e 32 milhões de assinaturas. Para o cenário com quatro NSNs, a curva se mostra menos eficiente que as demais, além de apresentar uma convergência com os demais cenários, para o nível de quatro NSNs. Para os cenários de dois, três e quatro NSNs não é observada melhoria nos tempos de respostas quando comparados com o cenário de um HTS executado dentro do nó NCN.

Diante dessas informações, o autor conclui que a distribuição do serviço HTS em rede não contribui para a redução no tempo de resposta, porém o aumento do tempo de resposta se mostra estável para o aumento no número de NSNs, sendo um indicativo de que a arquitetura suporta conexões distribuídas e existem condições para o aumento de escala da NG de forma sustentável.

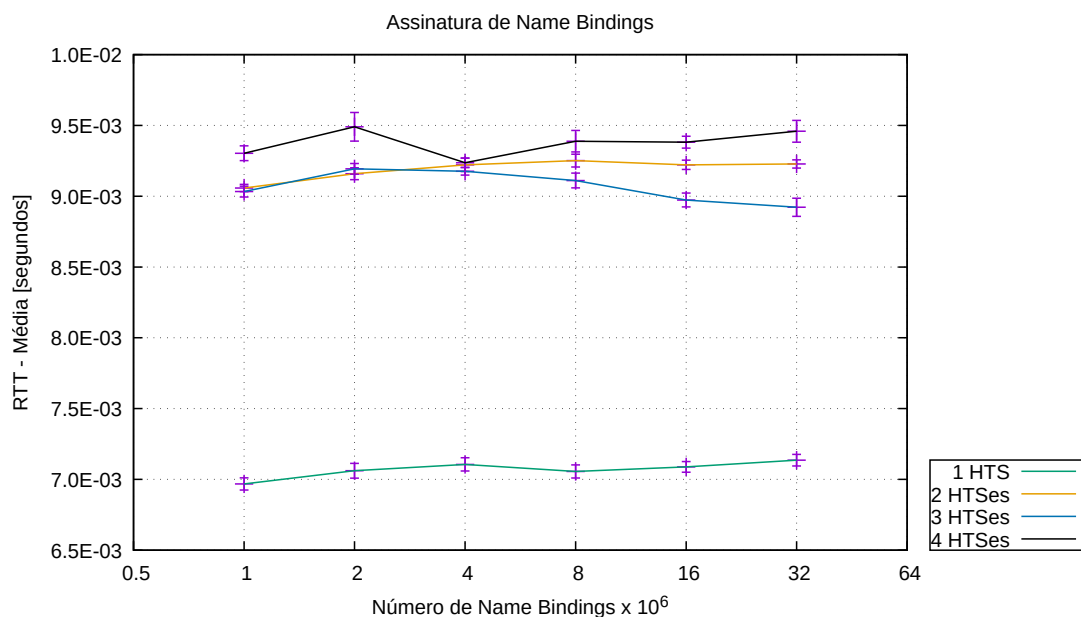


Figura 5.10: Comparação da assinatura nos 4 cenários de laboratório físico.

5.2 Ambiente Virtual

Nesta seção são apresentados os resultados e avaliações da NG em ambiente virtual. O ambiente que recebeu essas experimentações foi descrito na Seção 4.2.2. Para permitir uma avaliação detalhada, os resultados são exibidos primariamente em seu formato individual. Os cenários avaliados foram introduzidos na Seção 4.3.2 e são: 1 NSN, 2 NSNs, 3 NSNs e 4 NSNs. Os métodos de comunicação utilizados, publicação e assinatura de NBs, são os mesmos que foram detalhados na Seção 3.6.1. Ao final desta Subseção são unificados, por orientação de comunicação, os resultados dos cenários, com a finalidade de se fazer comparações adicionais.

5.2.1 Publicação de 1000 NBs por Mensagem

A Figura 5.11 apresenta os resultados de publicações de NBs para o Cenário 1 NSN. O eixo horizontal representa o número de publicações de NBs. O eixo vertical representa o RTT médio para a publicação de 1000 NBs em uma única mensagem. Por esses resultados é possível observar que o tempo RTT quase dobra do nível de 1 a 32 milhões de publicações de NBs. Tal aumento pode ser resultante de uma sobrecarga do nó responsável pela função de armazenamento da rede, no caso, como existe apenas uma instância do serviço HTS, o NCN.

A Figura 5.12 apresenta os resultados de publicações de NBs para o Cenário 2 NSNs. O eixo horizontal representa o número de publicações de NBs. O eixo vertical representa o RTT médio para a publicação de 1000 NBs em uma única mensagem. Por esses resultados é possível observar que há um aumento do RTT médio à medida que se aumentam as publicações, porém não chega a dobrar esse valor. Possivelmente, a distribuição dos dois NSNs em rede contribui para isso.

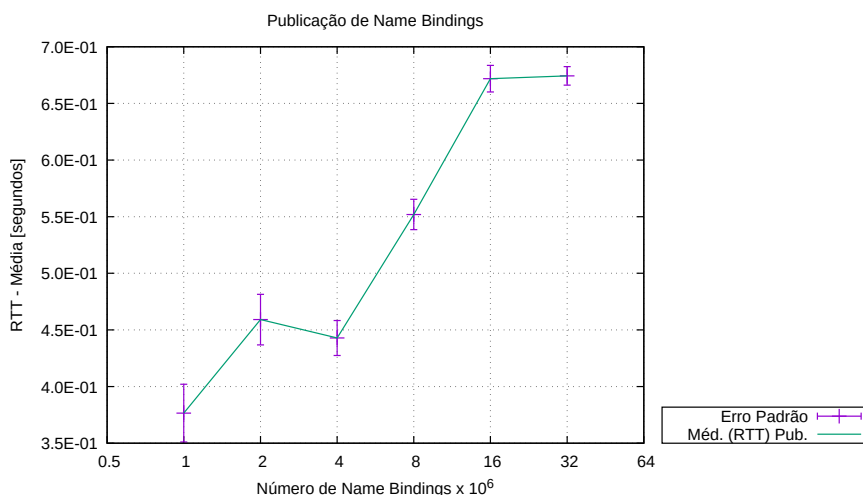


Figura 5.11: Publicação de NBs no ambiente virtual para o Cenário 1 NSN.

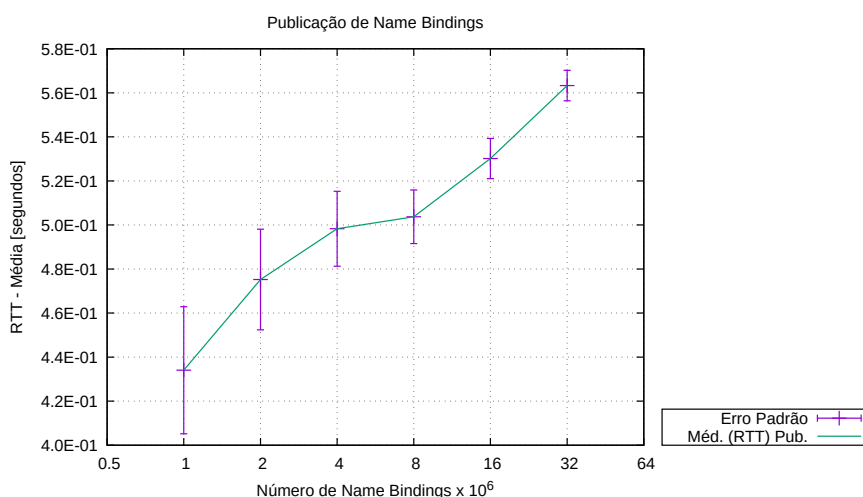


Figura 5.12: Publicação de NBs no ambiente virtual para o Cenário 2 NSNs.

A Figura 5.13 apresenta os resultados de publicações de NBs para o Cenário 3 NSNs. O eixo horizontal representa o número de publicações de NBs. O eixo vertical representa o RTT médio para a publicação de 1000 NBs em uma única mensagem. Por esses resultados é possível observar que há um aumento destacado no RTT médio para o nível de 32 milhões de NBs. Possivelmente, devido a uma sobrecarga dos serviços que fazem a distribuição dos NBs aos NSNs.

A Figura 5.14 apresenta os resultados de publicações de NBs para o Cenário 4 NSNs. O eixo horizontal representa o número de publicações de NBs. O eixo vertical representa o RTT médio para a publicação de 1000 NBs em uma única mensagem. Por esses resultados é possível observar que há um erro padrão maior nos níveis de 1 e 2 milhões de publicações de NBs e observa-se que esse erro diminui com o aumento do número de publicações. O RTT médio aumenta no decorrer da experimentação, mas não é exponencial, algo que pode ser atribuído à distribuição dos NSNs em rede.

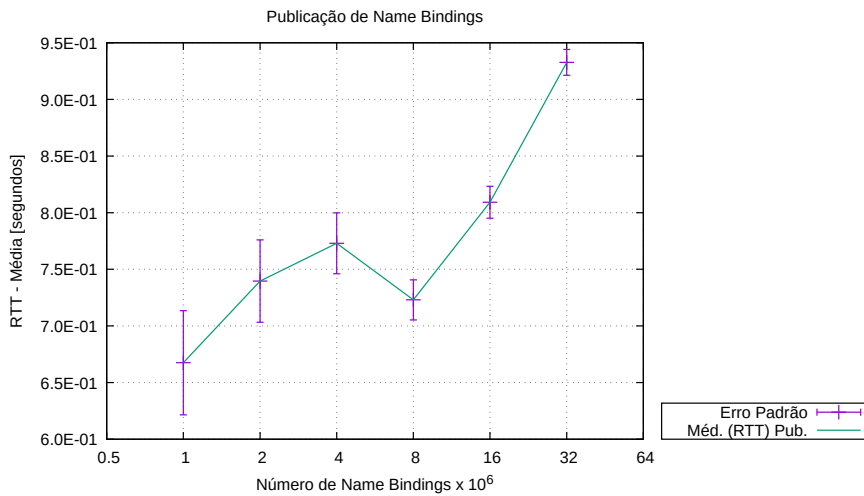


Figura 5.13: *Publicação de NBs no ambiente virtual para o Cenário 3 NSNs.*

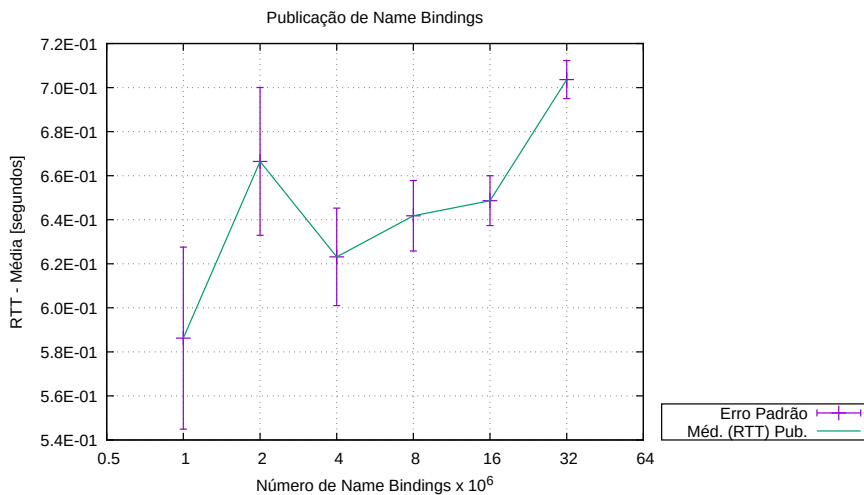


Figura 5.14: *Publicação de NBs no ambiente virtual para o Cenário 4 NSNs.*

5.2.2 Assinatura de um Único NB Aleatório por Segundo

A Figura 5.15 apresenta os resultados de assinaturas de NBs para o Cenário 1 NSN. O eixo horizontal representa o número de publicações de NBs realizadas antes de serem solicitadas as 360 assinaturas de NBs, o eixo vertical representa o RTT médio das assinaturas. Por esses resultados é possível perceber que há uma tendência de aumento no RTT médio, possivelmente por haver apenas um serviço HTS responsável pelo retorno das assinaturas.

A Figura 5.16 apresenta os resultados de assinaturas de NBs para o Cenário 2 NSNs. O eixo horizontal representa o número de publicações de NBs realizadas antes de serem solicitadas as assinaturas de NBs. O eixo vertical representa o RTT médio das 360 assinaturas. Por esses resultados é possível observar que há uma nítida redução no RTT médio à medida que se aumentam as publicações de NBs. Tal comportamento pode estar relacionado com a presença de dois NSNs na rede, o que distribuiria a carga e facilitaria a entrega das assinaturas.

A Figura 5.17 apresenta os resultados de assinaturas de NBs para o Cenário 3 NSNs.

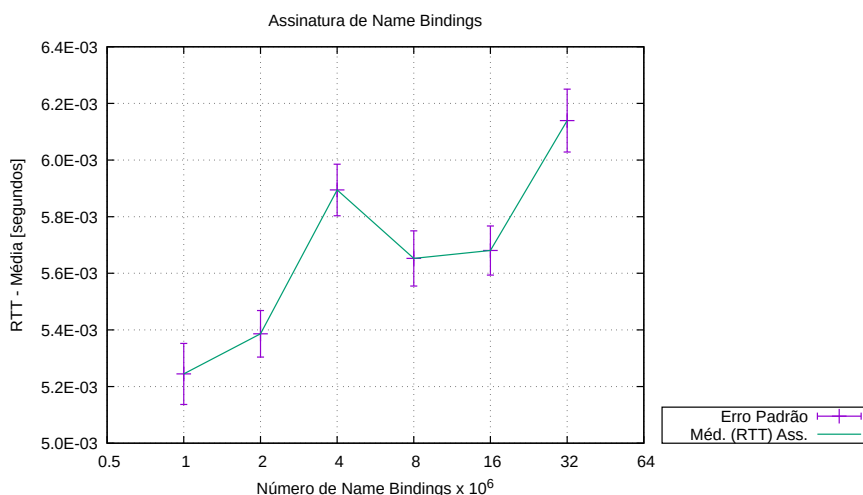


Figura 5.15: Assinatura de NBs no ambiente virtual para o Cenário 1 NSN.

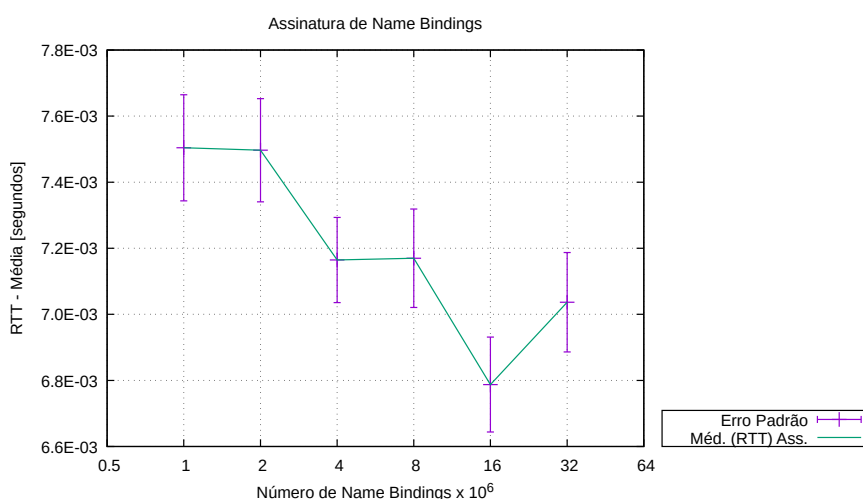


Figura 5.16: Assinatura de NBs no ambiente virtual para o Cenário 2 NSNs.

O eixo horizontal representa o número de publicações de NBs realizadas antes de serem solicitadas as assinaturas de NBs. O eixo vertical representa o RTT médio das 360 assinaturas. Por esses resultados é possível observar que há uma tendência de manutenção do RTT médio próximo ao tempo de 7.4E-3. A distribuição dos serviços HTSes na rede, diante de um número fixo de assinaturas, apresenta efeito de normalização desses resultados.

A Figura 5.18 apresenta os resultados de assinaturas de NBs para o Cenário 4 NSNs. O eixo horizontal representa o número de publicações de NBs realizadas antes de serem solicitadas as assinaturas de NBs. O eixo vertical representa o RTT médio das 360 assinaturas. Por esses resultados é possível observar que há uma tendência de queda no RTT médio com o aumento no número de NBs. Tal redução coaduna com o aumento no número de NSNs da rede.

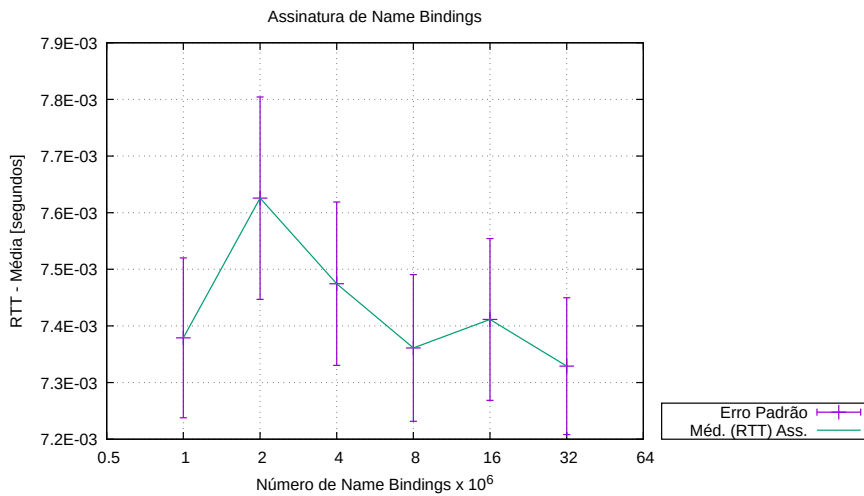


Figura 5.17: Assinatura de NBs no ambiente virtual para o Cenário 3 NSNs.

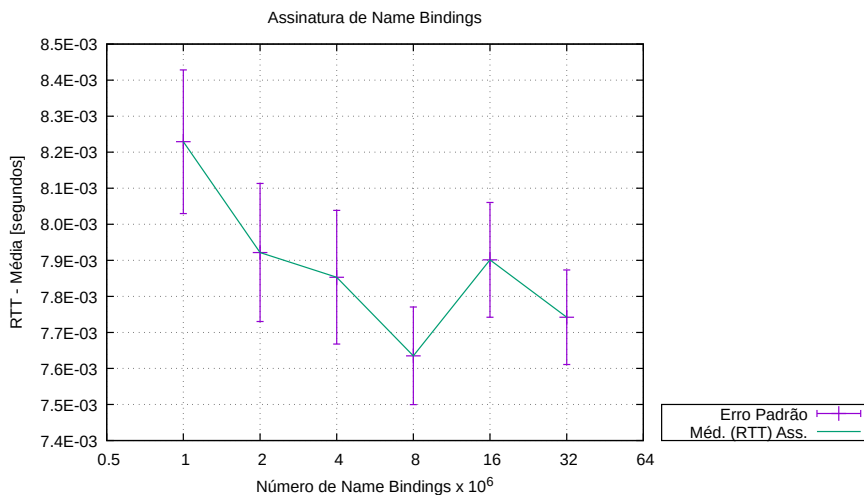


Figura 5.18: Assinatura de NBs no ambiente virtual para o Cenário 4 NSNs.

5.2.3 Comparação de Cenários

A Figura 5.19 apresenta um comparativo dos resultados das publicações de NBs nos quatro cenários de avaliação para o ambiente virtual. As curvas são diferenciadas pelo número de serviços HTSes presentes em cada cenário, resultantes da variação dos cenários de um a quatro NSNs, descritos na subseção 4.3.2. Nela é possível observar que a tendência de aumento do RTT médio na publicação de NBs é predominante e recorrente para todas as curvas. Em uma comparação individual, a curva do cenário de três NSNs apresenta o pior desempenho, não superando nenhuma das demais curvas em nenhum dos níveis de NBs. O cenário com quatro NSNs só garantiu melhor rendimento que a curva de um NSN para o nível de 16 milhões de NBs. A curva do cenário de um NSN é superior em desempenho para os níveis de 1 e 2 milhões de NBs publicados, em todo o comparativo. A distribuição de dois NSNs na rede apresentou o melhor desempenho entre as demais, superando o cenário de um NSN a partir de 4 milhões de NBs, e os demais cenários em todos os casos.

Desta forma, o autor conclui que a distribuição de HTSes na arquitetura da NG pode

propiciar desempenho superior em comparação com abordagem restrita a um HTS. O aumento de NSNs não implicou em uma mudança no comportamento de crescimento da curva padrão para um NSN, de forma que indica capacidade de escalabilidade da arquitetura para esse caso.

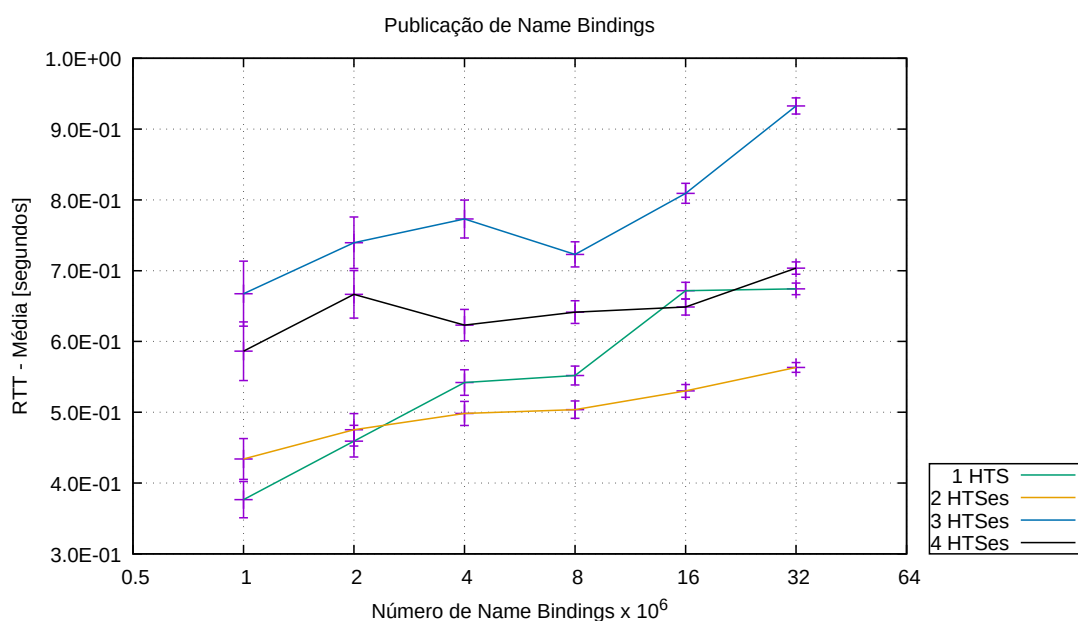


Figura 5.19: Comparação dos resultados de publicação nos quatro cenários virtuais.

A Figura 5.20 apresenta um comparativo dos resultados de assinatura nos quatro cenários de experimentação em ambiente virtual. Foi notado que no ambiente controlado existe uma tendência de aumento no RTT médio com relação ao número de NSNs e redução em função do aumento de NBs. O único cenário que apresentou tendência de crescimento, no tempo RTT médio, foi o que implementou apenas um NSN. As quatro curvas se apresentam bem distintas nesse gráfico, com exceção do cruzamento dos pontos da curva de dois e três NSNs no nível de 1 milhão de NBs, em todos os demais pontos não ocorrem intersecções. Analisando de forma individual, o cenário com um NSN obteve melhor desempenho que todos os demais. O cenário de dois, três e quatro NSNs ficaram um patamar acima do primeiro e tiveram resultados próximos entre si.

Desta forma, o autor conclui que o ambiente virtual controlado propicia maior fidelidade do comportamento esperado e medido, como destaque cita-se a redução no tempo de assinatura de cada cenário ao incremento de publicações de NBs. A observação de que se estabelece um patamar diferente de resultados para o cenário com apenas um NSN sinaliza que o desempenho da implementação do NSN junto a outros serviços do *core* da NG, como GIRS e PSS, supera os demais. Porém, atuando de forma distribuída, o HTS pode ser escalado a múltiplos nós sem que isso represente um aumento desproporcional no tempo de resposta às assinaturas de NBs.

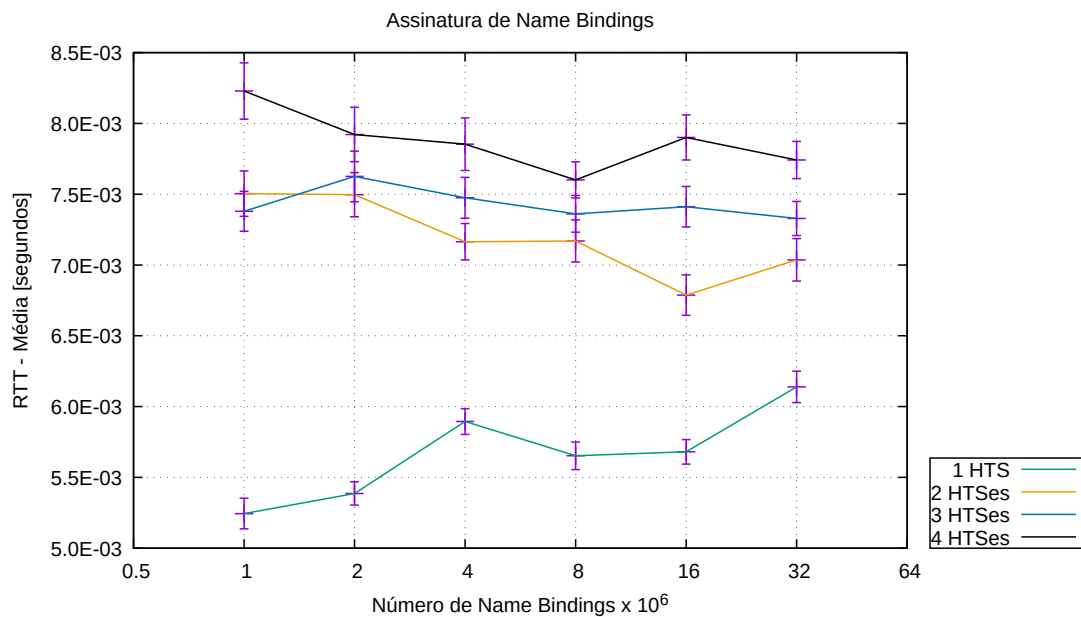


Figura 5.20: Comparação dos resultados de assinatura nos quatro cenários virtuais.

5.3 Comparações entre os Ambientes Físico e Virtual

Nesta subseção são comparados os gráficos dos resultados dos ambientes físico e virtual em seus dados de publicação e assinatura de NBs. Tal comparação é obtida através da composição dos resultados expostos na tabela que consta no Apêndice D.1 e gráficos já apresentados nas Figuras 5.9, 5.10, 5.19 e 5.20. Assim, é possível plotar esses gráficos, agrupados, nas Figuras 5.21 e 5.22. As curvas são diferenciadas pelo número de serviços HTSes presentes em cada cenário, resultantes da variação dos cenários de um a quatro NSNs, descritos na subseção 4.3.2.

A Figura 5.21 é uma composição dos resultados da publicação de NBs para os cenários de 1, 2, 3 e 4 NSNs nos ambientes físico e virtual. Ela é uma composição feita para comparação dos resultados já apresentados nas Figuras 5.9 e 5.19. Analisando os dados que ela apresenta, é possível perceber que o ambiente virtual possui tempos de publicação de NBs maiores que o do ambiente físico para todos os cenários de NSNs. Apesar de maiores, os tempos de publicação do ambiente virtual não apresentam comportamento díspar em relação ao ambiente físico, suas curvas se mantêm próximas a um limite inferior ao dobro dos tempos do ambiente físico.

A Figura 5.22 é uma composição dos resultados da assinatura de NBs para os cenários de 1, 2, 3 e 4 NSNs nos ambientes físico e virtual. Ela é uma composição feita para comparação dos resultados já apresentados nas Figuras 5.10 e 5.20. Analisando tais dados, observa-se que o ambiente virtual possui menor tempo de assinatura se comparado com os mesmos cenários do ambiente físico. Ainda, os resultados das assinaturas do ambiente virtual são melhores do que todos os três cenários: 2, 3 e 4 NSNs do ambiente físico. A exceção é o cenário 1 NSN do ambiente físico, que se mostra, majoritariamente, com um tempo de resposta inferior em relação aos cenários

citados. O cenário e ambiente com menor tempo RTT médio é, claramente, o de 1 NSN no ambiente virtual.

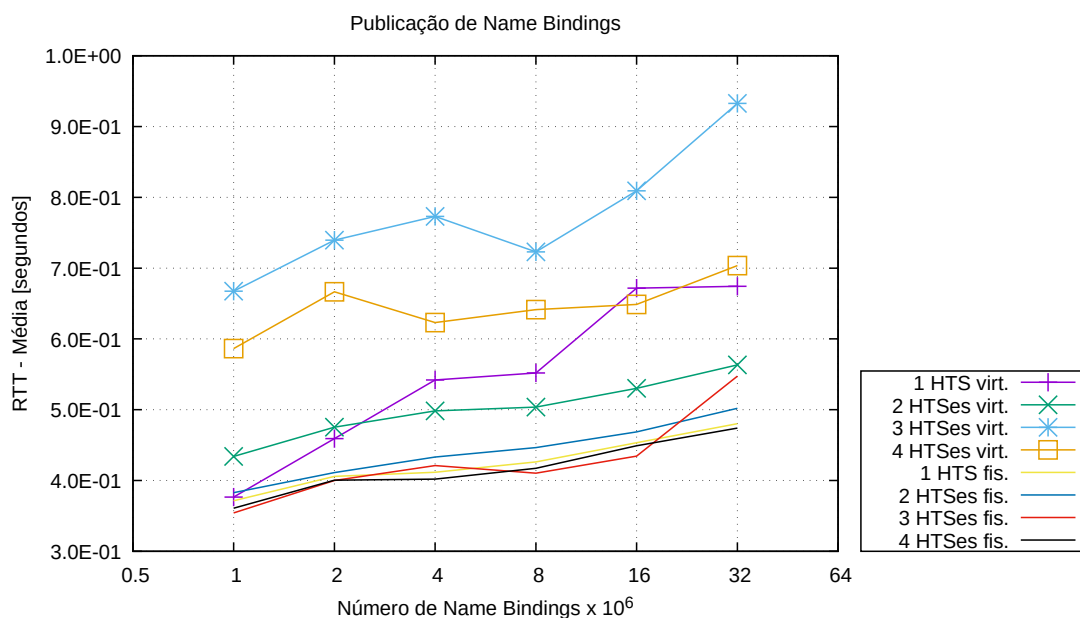


Figura 5.21: Comparação dos resultados de publicação de NBs para os quatro cenários dos ambientes virtual e físico (laboratórios).

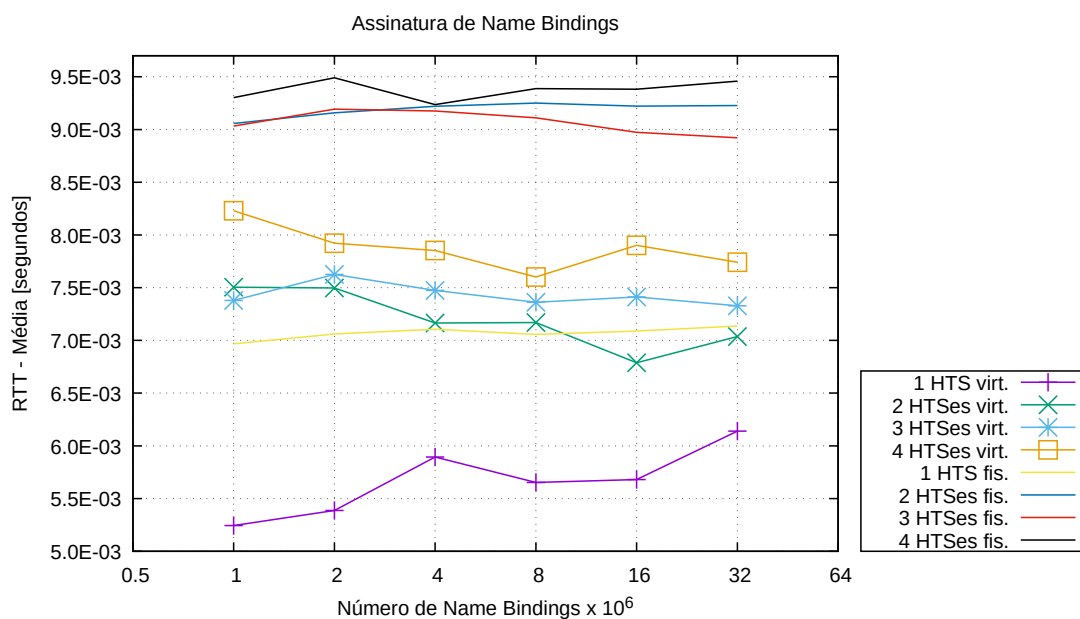


Figura 5.22: Comparação dos resultados de assinaturas de NBs para os quatro cenários dos ambientes virtual e físico (laboratórios).

5.4 Entendimentos do Capítulo

Neste capítulo são apresentados, em formato de gráficos, os resultados obtidos nas avaliações do serviço de publicação e assinatura de nomes da NG, em ambientes físico e virtual, em quatro diferentes cenários. São apresentados gráficos segmentados em orientação de comunicação e ambientes, além de serem compilados comparativos que permitem a avaliação detalhada dos resultados. Tais comparativos, entre os ambientes físico e virtual, suportam alguns dos objetivos iniciais deste estudo, dentre eles o de identificar a viabilidade de utilizar o ambiente virtual para experimentação da NovaGenesis.

Capítulo 6

Considerações Finais

Por fim, neste capítulo são apresentadas as principais conclusões e contribuições obtidas, as lições aprendidas de realizar a avaliação de uma arquitetura de Internet e alguns trabalhos futuros, que poderão ser feitos em continuidade às frentes de investigação descritas.

6.1 Conclusões e Contribuições

O presente trabalho apresentou contribuições diretas para a NG e como destaque é possível citar a consolidação de metodologias mais ágeis e controladas de experimentação, a avaliação da arquitetura em ambientes físicos e virtuais, a investigação do desempenho do serviço de resolução de nomes da arquitetura e a obtenção de dados para comparação da NG em outros ambientes de experimentação, como contêineres.

Dada a necessidade que se teve de estabelecer uma rotina de automação das experimentações, os *scripts shell*, os *playbooks* do Ansible e a configuração dos ambientes contribuíram consideravelmente para que fosse possível a realização de tantas avaliações e de distintas formas, variando em quatro cenários a topologia dos serviços da NG e em dois ambientes, físico e virtual. Dentre as metodologias que se estabeleceram para obter uma experimentação mais ágil, a gerência centralizadas dos testes, bem como a alta capacidade de virtualização de sistemas operacionais se destacam. A primeira pelo fato de que facilita o controle e gerência das rotinas de experimentação. A segunda pela maior agilidade que permitiu alcançar, ao lidar com máquinas virtuais ao invés de computadores físicos.

Para realizar tais avaliações foram configurados ambientes físicos e virtuais. A utilização de computadores físicos já era prática estabelecida pelo grupo de pesquisa da NG, porém a utilização de máquinas virtuais foi idealizada com a finalidade de oferecer um ambiente de experimentação mais flexível, dinâmico, ágil e simples de ser mantido. Os métodos de avaliação desenvolvidos também se mostraram mais aptos a serem empregados em ambientes virtuais, uma vez que as manutenções pontuais que eram necessárias no ambiente físico eram decorrentes de indisponibilidades difíceis de serem tratadas remotamente ou de forma ágil. Tais características contribuíram para uma das interpretações deste trabalho que é justamente a possibilidade de utilizar apenas

ambientes virtuais para futuras avaliações da NG e de seus serviços, desenvolvendo maturidade para fazer uso até de *testbeds* externos.

Quanto à investigação do desempenho dos serviços de resolução de nomes da NG, este trabalho foi pioneiro em uma das formas de avaliar e comparar o serviço NRNCS em ambientes físicos e virtuais. A avaliação que merece destaque é a distribuição dos serviços do NRNCS (GIRS, PSS e HTS) em rede e a execução de um serviço gerador de publicações e assinaturas (NBSimpletestAPP) a partir de outros nós da rede. Tais práticas permitiram a idealização de diversos cenários de experimentação (1 NSN, 2 NSNs, 3 NSNs e 4 NSNs) e a avaliação do desempenho da arquitetura, para publicação e assinatura de NBs, nessas diferentes composições de serviços NG.

Os resultados das avaliações nos diferentes cenários e ambientes consolidam a interpretação do autor de que a distribuição de NBs em distintos NSNs é implementada de forma adequada e escalonável. As curvas de publicação e assinatura de NBs não tiveram comportamento exponencial, o que demonstra a viabilidade de execução da aplicação sem a degradação da comunicação entre múltiplos nós, fato esse importantíssimo para validar a NG como alternativa ao atual sistema de resolução de nomes da Internet. A assinatura de um NB, dentre um conjunto de 32 milhões de NBs publicados, em um domínio foi, no pior caso, de aproximadamente 7,7 ms na avaliação com máquinas virtuais e 9,4 ms nos cenários com máquinas físicas. Esses resultados são promissores e colocam o sistema de resolução de nomes da NG como alternativa ao estabelecido, ou até mesmo a serviços similares de outras FIAs. Ainda, considerando possíveis aplicações distribuídas, seria possível obter benefício na distribuição do armazenamento de conteúdos em múltiplas instâncias de NSNs, agregando ao sistema características de resiliência, redundância e distribuição de carga para acessos aos conteúdos armazenados.

6.2 Lições Aprendidas

Com a realização deste trabalho, o autor conclui que a avaliação de uma arquitetura como a NG possui uma diversidade de desafios tanto operacionais, quanto científicos. A operação da rede de dados, computadores, servidores e aplicações de controle gerou um imenso legado e aprendizado. Os desafios de avaliar uma arquitetura de Internet do Futuro exige que se tenha sólidos conhecimentos sobre as arquiteturas atuais e sobre redes de computadores. Só assim é possível entender os novos conceitos que são propostos e conseguir avaliá-los. Em alguns momentos os conceitos dos diversos estudos podem gerar confusão na operação dos equipamentos, mas não é algo que não possa ser superado com uma revisão detalhada das documentações.

O desenvolvimento e avaliação de uma FIA perpassa por momentos que exigem uma alta demanda de recursos computacionais, de conectividade e de rede. Para que se possa estabelecer uma rotina fluída das execuções é importante deixar os operadores das redes do campus, universidade e/ou empresa cientes das avaliações que estão sendo feitas. Com isso evita-se indisponibilidades não previstas e pode-se até conseguir acesso a recursos exclusivos, que facilitam a operação das redes em teste. Para todas as experimentações realizadas neste trabalho o autor teve a felicidade de receber total apoio do Inatel e da equipe de TIC da instituição.

6.3 Trabalhos Futuros

Durante o desenvolvimento deste estudo e de suas implementações surgiram oportunidades de conhecer novas arquiteturas de Internet, topologias, ambientes de experimentação e tendências tecnológicas. Como não foi possível investigar e adicionar ao escopo deste trabalho todas as visões adquiridas, são elencadas a seguir possibilidades para a continuidade das propostas apresentadas, sendo algumas, inclusive, com escopos até um pouco visionários.

A avaliação feita neste trabalho compreendeu a variação no serviço de armazenamento de NBs e conteúdos da NG, o HTS. Para avaliar outros serviços que também se relacionam com a resolução de nomes, mas não especificamente com o armazenamento de NBs e conteúdos, sugere-se a variação no número de instâncias dos serviços GIRS e PSS na rede.

Caso sejam realizadas novas implementações do serviço NRNCS, a metodologia de testes aqui explicitada pode ser refeita e utilizada como métrica base. Bastando para isso instanciar o serviço de conectividade (PGCS) em cada nó, a nova implementação do serviço NRNCS, nos nós que se deseja avaliar e a aplicação de geração de tráfego em outro nó, o NBSimpletestAPP.

Em um contexto próximo ao que foi sugerido, seria oportuno expandir a avaliação para um número ainda maior de nós e observar o comportamento do NRNCS e como se comporta a divisão dos conteúdos para os HTSes, verificando se existe alguma tendência de oscilação de desempenho das publicações feitas aos HTSes da rede.

Durante a realização dos estudos bibliográficos foi possível perceber que alguns *testbeds*, como o PlanetLab, abordado na subseção 2.3.1, têm que lidar com inúmeros problemas operacionais, seja pela necessidade de manter um sistema de alta disponibilidade, de última geração para os pesquisadores ou pela obrigatoriedade de terem de seguir legislações que normatizam a Internet, o que acaba adicionando burocracia e diminuindo a flexibilidade que se busca. Nesse sentido, o autor incentiva a investigação da execução ininterrupta da NG, podendo ser estabelecida uma aplicação que seja executada através da NG e que seja mantida em operação de forma ininterrupta. No decorrer dessa execução, provavelmente, serão percebidos pontos de melhorias quanto à operação, quanto código, ou pelas interações da arquitetura com outras redes.

Referências Bibliográficas

- [1] L. Kleinrock, “History of the Internet and its flexible future,” *IEEE Wireless Communications*, vol. 15, no. 1, pp. 8–18, 2 2008. [Online]. Available: <http://ieeexplore.ieee.org/document/4454699/>
- [2] K. Salamatian, “Toward a polymorphic future internet: a networking science approach,” *IEEE Communications Magazine*, vol. 49, no. 10, pp. 174–178, 10 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/6035832/>
- [3] M. Conti, S. Chong, S. Fdida, W. Jia, H. Karl, Y.-D. D. Lin, P. Mähönen, M. Maier, R. Molva, S. Uhlig, and M. Zukerman, “Research challenges towards the Future Internet,” *Computer Communications*, vol. 34, no. 18, pp. 2115–2134, 12 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.comcom.2011.09.001><http://linkinghub.elsevier.com/retrieve/pii/S0140366411002714>
- [4] D. Meyer and G. Zobrist, “TCP/IP versus OSI,” *IEEE Potentials*, vol. 9, no. 1, pp. 16–19, 2 1990. [Online]. Available: <http://ieeexplore.ieee.org/document/46812/>
- [5] B. M. Leiner, R. E. Kahn, and J. Postel, “ABSTRACT,” Tech. Rep., 2003. [Online]. Available: <http://www.acm.org>
- [6] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, “A brief history of the internet,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 22–31, 10 2009. [Online]. Available: <https://dl.acm.org/doi/10.1145/1629607.1629613>
- [7] V. G. Cerf, “The day the Internet age began,” *Nature*, vol. 461, no. 7268, pp. 1202–1203, 10 2009. [Online]. Available: <http://www.nature.com/articles/4611202a>
- [8] K. D. Frazer, “NSFNET : A Partnership for High-Speed Networking Final Report,” *Merit Network*, vol. 6, no. 4, p. 44, 1995.
- [9] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. USA: Addison-Wesley Publishing Company, 2009.
- [10] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. USA: Prentice Hall Press, 2010.
- [11] C. Guo and S. Zheng, “Analysis and evaluation of the TCP/IP protocol stack of LINUX,” *International Conference on Communication Technology Proceedings, ICCT*, vol. 1, pp. 444–453, 2000.
- [12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, “Networking named content,” *Communications of the ACM*, vol. 55, no. 1, p. 117, 1 2012. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=>

- 2063176.2063204<https://named-data.net/publications/>
- [13] P. Martinez-julia and A. F. Skarmeta, “Beyond the separation of identifier and locator : Building an identity-based overlay network architecture for the Future Internet,” *Computer Networks*, vol. 57, no. 10, pp. 2280–2300, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2012.11.020>
- [14] D. K. Smetters and V. Jacobson, “Securing network content,” Tech. Rep., 2009. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.366.8858&rep=rep1&type=pdf>
- [15] A. Sharma, X. Tie, H. Uppal, A. Venkataramani, D. Westbrook, and A. Yadav, “A global name service for a highly mobile internet network,” in *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14*, vol. 44, no. 4. New York, New York, USA: ACM Press, 2014, pp. 247–258. [Online]. Available: <http://dl.acm.org/citation.cfm?doi=2619239.2626331>
- [16] M. Sailan, R. Hassan, and A. Patel, “A comparative review of IPv4 and IPv6 for research test bed,” in *2009 International Conference on Electrical Engineering and Informatics*, vol. 2, no. August. IEEE, 8 2009, pp. 427–433. [Online]. Available: <http://ieeexplore.ieee.org/document/5254698/>
- [17] J. Hintzbergen, K. Hintzbergen, A. Smulders, and H. Baars, *Fundamentos de Segurança da Informação: com base na ISO 27001 e na ISO 27002*. Brasport, 2018.
- [18] U. Steinho, A. Wiesmaier, and R. Araújo, “The State of the Art in DNS Spoofing,” Tech. Rep. [Online]. Available: http://wiesisoft.com/publications/reviewed/200606_ACNS06_The_State_of_the_Art_in_DNS_Spoofing.pdf
- [19] J. Rexford and C. Dovrolis, “Future Internet architecture,” *Communications of the ACM*, vol. 53, no. 9, pp. 36–40, 9 2010. [Online]. Available: <https://dl.acm.org/doi/10.1145/1810891.1810906>
- [20] P. Jäppinen, R. Guarneri, and L. M. Correia, “An applications perspective into the Future Internet,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 249–254, 1 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1084804512001853><https://linkinghub.elsevier.com/retrieve/pii/S1084804512001853>
- [21] M. E. M. Campista, M. G. Rubinstein, M. Igor, H. M. K. Costa, and O. C. M. B. Duarte, “Challenges and Research Directions for the Future Internet networking,” vol. 16, no. 2, pp. 1050–1079, 2014.
- [22] M. Gritter and D. R. Cheriton, “An architecture for content routing support in the internet,” in *3rd conference on USENIX Symposium on Internet Technologies and Systems*, vol. 3. San Francisco, California: USENIX Association, 3 2001, pp. 4–4. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1251440.1251444>
- [23] J. Torres, M. Nogueira, and G. Pujolle, “A Survey on Identity Management for the Future Network,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 787–802, 2013. [Online]. Available: <http://ieeexplore.ieee.org/document/6275425/>
- [24] J. Pan, R. Jain, S. Paul, and C. So-in, “MILSA: A New Evolutionary Architecture

- for Scalability, Mobility, and Multihoming in the Future Internet,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1344–1362, 10 2010. [Online]. Available: <http://ieeexplore.ieee.org/document/5586447/>
- [25] Yongguang Zhang and T. Henderson, “An implementation and experimental study of the explicit control protocol (XCP),” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 2. IEEE, 2005, pp. 1037–1048. [Online]. Available: <http://ieeexplore.ieee.org/document/1498332/>
- [26] B. Sugiarto, D. Laidi, N. Rizal, M. Galinium, P. Atmadiputra, H. Fahmi, T. Sampurno, and M. Kisworo, “Design and Implementation of the Connectionless Network Protocol (Clnp) As Loadable Kernel Modules in Linux Kernel 2 . 6,” no. 8, 2012.
- [27] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey, “Measuring IPv6 adoption,” in *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14*, vol. 44, no. 4. New York, New York, USA: ACM Press, 2014, pp. 87–98. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2619239.2626295>
- [28] A. Feldmann, “Internet clean-slate design,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, p. 59, 7 2007. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1273445.1273453>
- [29] M. Sifalakis, A. Louca, G. Bouabene, M. Fry, A. Mauthe, and D. Hutchison, “Functional composition in future networks,” *Computer Networks*, vol. 55, no. 4, pp. 987–998, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2010.12.006>
- [30] A. Rutkowski, “Understanding next generation Internet,” *IEEE Communications Magazine*, vol. 37, no. 9, pp. 99–102, 1999. [Online]. Available: <http://ieeexplore.ieee.org/document/790967/>
- [31] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, “MobilityFirst future internet architecture project,” in *Proceedings of the 7th Asian Internet Engineering Conference on - AINTEC '11*. New York, New York, USA: ACM Press, 2011, pp. 1–3. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2089016.2089017>
- [32] M. B. Lehmann, M. P. Barcellos, and A. Mauthe, “Providing producer mobility support in NDN through proactive data replication,” in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, no. Noms. IEEE, 4 2016, pp. 383–391. [Online]. Available: <http://ieeexplore.ieee.org/document/7502835/>
- [33] A. Anand, J. W. Byers, S. Seshan, P. Steenkiste, F. Dogar, D. Han, B. Li, H. Lim, M. Machado, W. Wu, A. Akella, and D. G. Andersen, “XIA,” in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks - HotNets '11*. New York, New York, USA: ACM Press, 2011, pp. 1–6. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2070562.2070564>
- [34] W. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. Garcia de Blas, F. Ramon-

- Salguero, L. Liang, S. Spirou, A. Beben, and E. Hadjioannou, "Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 112–120, 3 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5723808<http://ieeexplore.ieee.org/document/5723808/>
- [35] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6834762>
- [36] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, no. 2014, pp. 5–23, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.bjp.2013.12.037>
- [37] P. T. Kirstein, "Research on Networks versus Networks for Research : The Need for International , Internet , Testbeds," pp. 1–9, 1999.
- [38] A. M. Alberti, M. A. F. Casaroli, D. Singh, and R. da Rosa Righi, "Naming and name resolution in the future internet: Introducing the NovaGenesis approach," *Future Generation Computer Systems*, vol. 67, no. September, pp. 163–179, 2 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.future.2016.07.015><https://linkinghub.elsevier.com/retrieve/pii/S0167739X16302643>
- [39] A. M. Alberti, G. D. Scarpioni, V. J. Magalh, A. C. S. Jr, V. J. Magalhaes, S. A. Cerqueira, J. J. Rodrigues, and R. da R. Righi, "Advancing NovaGenesis Architecture Towards Future Internet of Things," *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–15, 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/7970111/>
- [40] A. M. Alberti and D. Singh, "Developing a NovaGenesis architecture model for service oriented future Internet and IoT: An advanced transportation system scenario," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, 3 2014, pp. 359–364. [Online]. Available: <http://ieeexplore.ieee.org/document/6803188/>
- [41] A. M. Alberti, E. C. Rosário, G. Cassiano, J. R. Santos, V. H. D. Ávila, and J. R. Carneiro, "Performance Evaluation of NovaGenesis Information-Centric Network," in *2nd International Multidisciplinary Conference on Computer and Energy Science*. Split: 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), 2017, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/8019276/>
- [42] D. L. Fussia, A. M. Alberti, and S. Rita, "Um Modelo para Navegação Web usando Nomeação Auto- Certificável e Publica / Assina," *Simpósio Brasileiro de Redes de Computadores - WORKSHOP DE REDES P2P, DINÂMICAS, SOCIAIS E ORIENTADAS A CONTEÚDO (WP2P+)*, vol. 6, 2016. [Online]. Available: http://sbrc2016.ufba.br/downloads/WP2P/Um_Modelo_para_Navegacao_Web_usando_Nomeacao_AutoCertificavel_e_Publica-Assina.pdf
- [43] A. Marcos Alberti, M. M. Bontempo, J. R. Santos, S. A. Cerqueira, R. da Rosa Righi, A. M. Alberti, M. M. Bontempo, J. Rodrigo, C. S. Jr, and

- R. Righi, “NovaGenesis Applied to Information-Centric, Service-Defined, Trustable IoT/WSAN Control Plane and Spectrum Management,” *MDPI - Sensors*, pp. 1–34, 2018.
- [44] A. M. Alberti, D. Mazzer, M. M. Bontempo, L. H. de Oliveira, R. da Rosa Righi, and A. Cerqueira Sodr e, “Cognitive radio in the context of internet of things using a novel future internet architecture called NovaGenesis,” *Computers and Electrical Engineering*, vol. 57, pp. 147–161, 1 2017.
- [45] J. A. Gavazza, J. C. Melo, T. B. da Silva, A. M. Alberti, P. F. Rosa, F. de Oliveira Silva, F. L. Verdi, and J. A. Suruagy, “Future Internet Exchange Point (FIXP): Enabling Future Internet Architectures Interconnection,” in *Advances in Intelligent Systems and Computing*, vol. 1151 AISC. Springer, 4 2020, pp. 703–714. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-44041-1_62
- [46] S. Lipoff, “Mass Market Potential for Home Terminals,” *IEEE Transactions on Consumer Electronics*, vol. CE-25, no. 2, pp. 169–184, 5 1979. [Online]. Available: <http://ieeexplore.ieee.org/document/4050152/>
- [47] G. Estrin and L. Kleinrock, “Measures, models and measurements for time-shared computer utilities,” in *Proceedings of the 1967 22nd national conference on -*. New York, New York, USA: ACM Press, 1967, pp. 85–96. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=800196.805978>
- [48] B. des Ligneris, “Virtualization of Linux Based Computers: The Linux-VServer Project,” in *19th International Symposium on High Performance Computing Systems and Applications (HPCS’05)*. IEEE, 2005, pp. 340–346. [Online]. Available: <http://ieeexplore.ieee.org/document/1430092/>
- [49] T. Petrovi c and K. Fertalj, “Demystifying desktop virtualization,” *Proceedings of the 9th WSEAS International Conference on Applied Computer Science, ACS ’09*, pp. 241–246, 2009.
- [50] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, “The Algorithmic Aspects of Network Slicing,” *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 8 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/8004165/>
- [51] L. R. Bays, R. R. Oliveira, M. P. Barcellos, L. P. Gaspary, and E. R. Mauro Madeira, “Virtual network security: threats, countermeasures, and challenges,” *Journal of Internet Services and Applications*, vol. 6, no. 1, p. 1, 12 2015. [Online]. Available: <http://www.jisajournal.com/content/6/1/1>
- [52] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” in *Proceedings of the nineteenth ACM symposium on Operating systems principles - SOSP ’03*. New York, New York, USA: ACM Press, 2003, p. 164. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=945445.945462>
- [53] T. Deshane, Z. Shepherd, J. N. Matthews, M. Ben-Yehuda, A. Shah, and B. Rao, “Quantitative Comparison of Xen and KVM,” *Xen Summit*, no. Cli, pp. 23–25, 2008. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=>

- 10.1.1.156.8134&rep=rep1&type=pdf
- [54] “VMware Understanding Full Virtualization, Paravirtualization, and Hardware Assist,” Tech. Rep. [Online]. Available: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf
- [55] L. Abeni and D. Faggioli, “Using Xen and KVM as real-time hypervisors,” *Journal of Systems Architecture*, vol. 106, no. January, p. 101709, 6 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1383762120300035>
- [56] B. Zhang, X. Wang, R. Lai, L. Yang, Z. Wang, Y. Luo, and X. Li, “Evaluating and Optimizing I/O Virtualization in Kernel-based Virtual Machine (KVM),” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6289 LNCS, pp. 220–231. [Online]. Available: http://link.springer.com/10.1007/978-3-642-15672-4_20
- [57] T. Combe, A. Martin, and R. Di Pietro, “To Docker or Not to Docker: A Security Perspective,” *IEEE Cloud Computing*, vol. 3, no. 5, pp. 54–62, 9 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7742298/>
- [58] J. Che, Y. Yu, C. Shi, and W. Lin, “A synthetical performance evaluation of OpenVZ, Xen and KVM,” *Proceedings - 2010 IEEE Asia-Pacific Services Computing Conference, APSCC 2010*, pp. 587–594, 2010.
- [59] J. Hwang, S. Zeng, F. Y. Wu, and T. Wood, “A component-based performance comparison of four hypervisors,” in *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management, IM 2013*, 2013, pp. 269–276.
- [60] J. Pan, S. Paul, and R. Jain, “A survey of the research on future internet architectures,” *IEEE Communications Magazine*, vol. 49, no. 7, pp. 26–36, 7 2011. [Online]. Available: <https://ieeexplore.ieee.org/document/5936152/>
- [61] O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, H. Kobayashi, S. Palazzo, S. Sahni, X. Shen, M. Stan, J. Xiaohua, A. Zomaya, and G. Coulson, “Testbedsand Research Infrastructures - Development of Networks and Communities,” Tech. Rep.
- [62] J. S. Turner, M. Wilson, C. Wiseman, D. Zar, P. Crowley, J. DeHart, A. Freestone, B. Heller, F. Kuhns, S. Kumar, J. Lockwood, and J. Lu, “Supercharging planetlab,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, p. 85, 10 2007. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1282427.1282391>
- [63] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, and S. Muir, “Operating System Support for Planetary-Scale Network Services Andy,” in *Symposium on Networked Systems Design and Implementation*, San Francisco, California, 2004. [Online]. Available: https://www.usenix.org/legacy/events/nsdi04/tech/full_papers/bavier/bavier.html/
- [64] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, “A blueprint for introducing disruptive technology into the Internet,” *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 59–64, 1 2003. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=774763.774772>

- [65] “PlanetLab Bibliography — PlanetLab.” [Online]. Available: <https://www.planet-lab.org/biblio>
- [66] S. Rhea, B. Godfrey, B. Karp, J. Kubiawicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, “OpenDHT,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, p. 73, 10 2005. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1090191.1080102>
- [67] C. Dovrolis, K. Gummadi, A. Kuzmanovic, and S. D. Meinrath, “Measurement Lab: Overview and an invitation to the research community,” in *Computer Communication Review*, vol. 40, no. 3, 6 2010, pp. 53–56. [Online]. Available: <https://dl.acm.org/doi/10.1145/1823844.1823853>
- [68] D. Bernstein, “Containers and Cloud: From LXC to Docker to Kubernetes,” *IEEE Cloud Computing*, vol. 1, no. 3, pp. 81–84, 9 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/7036275/>
- [69] V. Ramasubramanian and E. G. Sirer, “The design and implementation of a next generation name service for the internet,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, p. 331, 10 2004. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1030194.1015504>
- [70] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, p. 181, 10 2007. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1282427.1282402>
- [71] Larry Peterson, “It’s Been a Fun Ride - Systems Approach.” [Online]. Available: <https://www.systemsapproach.org/blog/its-been-a-fun-ride>
- [72] R. McGeer, M. Berman, C. Elliott, and R. Ricci, *The GENI Book*, R. McGeer, M. Berman, C. Elliott, and R. Ricci, Eds. Cham: Springer International Publishing, 2016. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-33769-2>
- [73] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network Function Virtualization: State-of-the-art and Research Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 619–639, 9 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7243304/http://ieeexplore.ieee.org/document/7467400/http://ieeexplore.ieee.org/document/7738442/http://arxiv.org/abs/1509.07675http://dx.doi.org/10.1109/COMST.2015.2477041>
- [74] Yong Li and Min Chen, “Software-Defined Network Function Virtualization: A Survey,” *IEEE Access*, vol. 3, pp. 2542–2553, 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7350211/>
- [75] J. Chase, L. Grit, and D. Irwin, “Beyond virtual data centers: Toward an open resource control architecture,” *Selected Papers from the International Conference on the Virtual Computing Initiative*, no. Vm, pp. 1–10, 2007.
- [76] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, *MobilityFirst Future Internet Architecture Project*, 2011.
- [77] R. Grandl, D. Han, S. B. Lee, H. Lim, M. MacHado, M. Mukerjee, and

- D. Naylor, "Supporting network evolution and incremental deployment with XIA," in *SIGCOMM'12 - Proceedings of the ACM SIGCOMM 2012 Conference Applications, Technologies, Architectures, and Protocols for Computer Communication*. New York, New York, USA: ACM Press, 2012, pp. 281–282. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2342356.2342410>
- [78] M. Serrano, N. Isaris, H. Schaffers, J. Domingue, M. Boniface, and T. Korakis, "Building the Future Internet through FIRE," 2017, pp. 1–794. [Online]. Available: http://www.riverpublishers.com/research_details.php?book_id=427http://riverpublishers.com/dissertations_xml/9788793519114/9788793519114.xmlhttps://nitlab.inf.uth.gr/NITlab/papers/fire_book.pdf
- [79] J. Pieper, "Standard Commands for Programmable Instruments (SCPI) - Volume 1 : Syntax and Style," *Style (DeKalb, IL)*, vol. 1, p. 819, 1999. [Online]. Available: <http://www.jpacsoft.com/links.htm%5Cnhttp://www.scpiconsortium.org>
- [80] T. Wauters, B. Vermeulen, W. Vandenberghe, P. Demeester, S. Taylor, L. Baron, M. Smirnov, Y. Al-Hazmi, A. Willner, and M. Sawyer, "Federation of Internet experimentation facilities: architecture and implementation," *European Conference on Networks and Communications (EuCNC)*, no. June, pp. 1–5, 2014.
- [81] B. Sousa, V. Fonseca, P. Simoes, and L. Cordeiro, "Evaluation of scalable, on-demand DNS-as-a-Service," Tech. Rep.
- [82] A. Ioannou, F. O. Coileain, D. Collins, Y. Zhang, and B. Chen, "CLONE: An NDN Architecture for Content Distribution at Remote Tourist Sites-a TCP/IP and NDN Comparison: An NDN Architecture for Content Distribution at Re-mote Tourist Sites-a TCP/IP and NDN." [Online]. Available: <https://doi.org/10.1145/3267955.3269018>
- [83] A. Willner, M. Giatili, P. Grosso, C. Papagianni, M. Morsey, and I. Baldin, "Using Semantic Web Technologies to Query and Manage Information within Federated Cyber-Infrastructures," Tech. Rep. [Online]. Available: www.mdpi.com/journal/information
- [84] S. Hoque, M. S. D. Brito, A. Willner, O. Keil, and T. Magedanz, "Towards Container Orchestration in Fog Computing Infrastructures," in *Proceedings - International Computer Software and Applications Conference*, vol. 2. IEEE Computer Society, 9 2017, pp. 294–299.
- [85] J. Perelló, A. López, and D. Careglio, "Experimenting with Real Application-specific QoS Guarantees in a Large-scale RINA Demonstrator," Tech. Rep.
- [86] S. Sallent, A. Abelém, I. Machado, L. Bergesio, S. Fdida, J. Rezende, S. Azodolmolky, M. Salvador, L. Ciuffo, and L. Tassiulas, "FIBRE project: Brazil and Europe unite forces and testbeds for the internet of the future," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 44 LNICST, p. 372, 2012.
- [87] A. Abelem, S. Fdida, I. MACHADO, L. CIUFFO, D. MARQUES, T. SALMITO, M. STANTON, A. Abelem, J. F. d. REZENDE, M. SALVADOR, S. SALLENT, L. BERGESIO, S. Fdida, M. CHANNEGOWDA, L. TASSIULAS, and D. GIATSIOS, "Building an infrastructure for experimentation between Brazil

- and Europe to enhance research collaboration in future Internet,” in *TERENA Networking Conference, Dublin, Ireland*, no. April. Dublin: TERENA, 5 2014, pp. 1–4. [Online]. Available: <https://tnc2014.terena.org/getfile/1071>
- [88] T. Salmito, L. Ciuffo, I. Machado, M. Salvador, M. Stanton, N. Rodriguez, A. Abelem, L. Bergesio, S. Sallent, and L. L. Baron, “FIBRE an international testbed for future internet experimentation,” in *Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos - SBRC*, no. October 2011, 2014, pp. p–969.
- [89] D. Caled, T. Salmito, J. F. Santos, L. Ciuffo, J. Rezende, and I. Machado, “The Next Generation of the FIBRE Software Architecture,” in *Anais do VIII WPEIF 2017 – Workshop de Pesquisa Experimental da Internet do Futuro*, Belém, 2017, pp. 3–6.
- [90] L. E. Folly, M. F. Schwarz, G. Araújo, A. S. Moura, and L. N. Ciuffo, “FIBREnet - o backbone do testbed FIBRE,” in *Anais do IX WPEIF 2018 – Workshop de Pesquisa Experimental da Internet do Futuro*, Campos do Jordão, 2018. [Online]. Available: <http://www.sbrc2018.ufscar.br/wp-content/uploads/2018/04/06-181797.pdf>
- [91] B. Oliveira, S. Inf, K. Vieira, C. Inf, S. Kleber, V. Cardoso, S. Inf, K. Vieira, and C. Inf, “Controle e Monitoramento de Experimentos para a Internet do Futuro Usando OMF,” Tech. Rep. [Online]. Available: http://sbrc.org.br/sbrc2012/publicacoes/99308_1.pdf
- [92] C. Luiz and J. A. S. Monteiro, “Investigação do Protótipo ProtoRINA utilizando o Ambiente FIBRE,” *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 1–4, 2016. [Online]. Available: http://www.sbrc2016.ufba.br/downloads/WPEIF/154481_1-Investigacao-do-ProtoRINA-Ambiente-FIBRE.pdf
- [93] F. A. Ferreira, F. S. Miranda, E. C. d. Rosario, V. H. D. Davila, and A. M. Alberti, “NovaGenesis no Ambiente FIBRE: Primeiras Impressões,” in *WORKSHOP DE PESQUISA EXPERIMENTAL DA INTERNET DO FUTURO (WPEIF)*. Belém: Sociedade Brasileira de Computação, 2017, pp. 6–9. [Online]. Available: <https://sbrc2017.ufpa.br/wp-content/uploads/2017/05/proceedingsWPEIF2017.pdf>
- [94] L. Zhang, V. Jacobson, S. Diego, P. Crowley, S. Louis, L. Wang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, “Named data networking,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 7 2014. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2656877.2656887>
- [95] A. Anand, J. W. Byers, S. Seshan, P. Steenkiste, F. Dogar, D. Han, B. Li, H. Lim, M. Machado, W. Wu, A. Akella, and D. G. Andersen, “XIA,” in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks - HotNets '11*. New York, New York, USA: ACM Press, 2011, pp. 1–6. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2070562.2070564>
- [96] L. M. Correia, H. Abramowicz, M. Johnsson, and K. Wüstel, Eds., *Architecture and Design for the Future Internet*, ser. Signals and Communication Technology. Dordrecht: Springer Netherlands, 2011. [Online]. Available:

- <http://link.springer.com/10.1007/978-90-481-9346-2>
- [97] A. Architecture, D. Project, O. Publish, and E. E. May, “New Generation Network Architecture AKARI Conceptual Design (ver2 . 0),” *Architecture*, no. May, 2010.
- [98] C. Dovrolis, “What would darwin think about clean-slate architectures?” *Computer Communication Review*, vol. 38, no. 1, pp. 29–34, 2008.
- [99] Y. Hayamizu, K. Hirata, and M. Yamamoto, “CCAR: Caching and Content-Aware Routing for Content Oriented Networking,” in *2018 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, vol. 1. IEEE, 5 2018, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8445945/referenceshttps://ieeexplore.ieee.org/document/8445945/>
- [100] H. Luo, Y. Xu, W. Xie, Z. Chen, J. Li, H. Zhang, and H.-c. Chao, “A Framework for Integrating Content Characteristics into the Future Internet Architecture,” *IEEE Network*, vol. 31, no. 3, pp. 22–28, 5 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7880516/>
- [101] G. Xylomenos, G. Pavlou, I. Psaras, and I. Karakonstantis, “Named Functions at the Edge,” in *IEEE Symposium on Computers and Communications (ISCC)*, Barcelona, 2019, pp. 1–6.
- [102] D. Smetters and V. Jacobson, “Securing network content,” *Palo Alto Research Center (PARC)*, pp. 1–7, 2009. [Online]. Available: <https://www.parc.com/content/attachments/TR-2009-01.pdf>
- [103] O. Waltari and J. Kangasharju, “Content-Centric Networking in the Internet of Things,” in *2016 13th IEEE Annual Consumer Communications and Networking Conference, CCNC 2016*. Institute of Electrical and Electronics Engineers Inc., 3 2016, pp. 73–78.
- [104] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, “Information-centric networking for the internet of things: Challenges and opportunities,” *IEEE Network*, vol. 30, no. 2, pp. 92–100, 3 2016.
- [105] R. P. S. Chaib and A. M. Alberti, “A Simple Solution for IoT Experimentation in the Context of Future Internet Architectures,” *Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF_SBRC)*, vol. 8, 2017. [Online]. Available: <https://portaldeconteudo.sbc.org.br/index.php/wpeif/article/view/2607/2569>
- [106] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. M. Kermarrec, “The Many Faces of Publish/Subscribe,” *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, 6 2003. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=857076.857078>
- [107] E. Bertrand-martínez, P. Feio, V. Nascimento, B. Pinheiro, and A. Abelém, “A Methodology for Classification and Evaluation of IoT Brokers,” *9th Latin American Network Operations and Management Symposium, LANOMS 2019, Niterói, Rio de Janeiro, Brazil, September 25-27, 2019*, 2019. [Online].

- Available: http://dl.ifip.org/db/conf/lanoms/lanoms2019/195912_1.pdf
- [108] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, “A survey of naming and routing in information-centric networks,” pp. 44–53, 2012.
- [109] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, “A Survey of Information-Centric Networking Research,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6563278/>
- [110] C. Yan, Q. N. Nguyen, I. Benkacem, D. Okabe, A. Nakao, T. Tsuda, C. Safitri, T. Taleb, and T. Sato, “Design and implementation of integrated ICN and CDN as a video streaming service,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11618 LNCS. Springer Verlag, 6 2019, pp. 194–206. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-30523-9_16
- [111] Y. Wang, Y. Wang, F. Esposito, I. Matta, and J. Day, “RINA: An Architecture for Policy-Based Dynamic Service Management.” [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.431.2380>
- [112] “ETArch-SG - A SDN-based Architecture to Support the Communication Requirements of Future Smart Grids — IEEE Conference Publication — IEEE Xplore.” [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8538454>
- [113] W. K. Chai, G. Pavlou, G. Kamel, K. V. Katsaros, and N. Wang, “A Distributed Interdomain Control System for Information-Centric Content Delivery,” *IEEE Systems Journal*, vol. PP, pp. 1–12, 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8430531/>
- [114] “The IPv4 Routed /24 Topology Dataset.” [Online]. Available: https://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml
- [115] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. Hyun Kim, and S. Shenker, *A Data-Oriented (and Beyond) Network Architecture*, 2007.
- [116] K. V. Katsaros, X. Vasilakos, T. Okwii, and G. Xylomenos, “On the Interdomain Scalability of Route-by-Name Information-Centric Network Architectures,” 2015.
- [117] W. K. Chai, G. Pavlou, G. Kamel, K. V. Katsaros, N. Wang, and S. Member, “A Distributed Interdomain Control System for Information-Centric Content Delivery,” *IEEE Systems Journal*, vol. PP, pp. 1–12, 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8430531/>
- [118] A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao, and S. Banerjee, “MobilityFirst,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 74–80, 7 2014. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2656877.2656888>
- [119] F. Bronzino, K. Nagaraja, I. Seskar, and D. Raychaudhuri, “Network service abstractions for a mobility-centric future internet architecture,” in *Proceedings*

- of the eighth ACM international workshop on Mobility in the evolving internet architecture - MobiArch '13*. New York, New York, USA: ACM Press, 2013, p. 5. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2505906.2505908>
- [120] Y. Wang and J. Bi, “A solution for IP mobility support in software defined networks,” in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*. Institute of Electrical and Electronics Engineers Inc., 9 2014.
- [121] “Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols - IEEE Conference Publication.” [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1424763>
- [122] “RFC 8569 - Content-Centric Networking (CCNx) Semantics.” [Online]. Available: <https://tools.ietf.org/html/rfc8569>
- [123] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang, and L. Zhang, “A Brief Introduction to Named Data Networking,” in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, vol. 2019-Octob. IEEE, 10 2018, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/8599682/>
- [124] “Interest Packet — NDN Packet Format Specification version 0.3.” [Online]. Available: <https://named-data.net/doc/NDN-packet-spec/current/interest.html>
- [125] “Data Packet — NDN Packet Format Specification version 0.3.” [Online]. Available: <https://named-data.net/doc/NDN-packet-spec/current/data.html>
- [126] H. Zhang, Y. Li, Z. Zhang, A. Afanasyev, and L. Zhang, “NDN host model,” *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 3, pp. 35–41, 9 2018. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3276799.3276804>
- [127] H. Lim, A. Ni, D. Kim, Y.-B. Ko, S. Shannigrahi, and C. Papadopoulos, “NDN Construction for Big Science: Lessons Learned from Establishing a Testbed,” *IEEE Network*, vol. 32, no. 6, pp. 124–136, 11 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8540804/>
- [128] V. Jacobson, J. Burke, D. Estrin, L. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, K. Shilton, L. Wang, E. Yeh, and P. Crowley, “Named Data Networking (NDN) Project Principal Investigators,” *Computer*, 2013.
- [129] M. Meisel, V. Pappas, and L. Zhang, “Ad hoc networking via named data,” in *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture - MobiArch '10*, vol. 28, no. 2. New York, New York, USA: ACM Press, 1 2010, p. 3. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1859983.1859986>
- [130] P. H. V. Guimaraes, L. H. G. Ferraz, J. V. Torres, D. M. F. Mattos, A. F. Murillo P., M. E. Andreoni L., I. D. Alvarenga, C. S. C. Rodrigues, and O. C. M. B. Duarte, “Experimenting Content-Centric Networks in the future internet testbed environment,” in *2013 IEEE International Conference on Communications Workshops (ICC)*, no. Wcc. IEEE, 6 2013, pp. 1383–1387. [Online]. Available: <http://ieeexplore.ieee.org/document/6649453/>

- [131] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang, “NDNS: A DNS-like name service for NDN,” in *2017 26th International Conference on Computer Communications and Networks, ICCCN 2017*. Institute of Electrical and Electronics Engineers Inc., 9 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/8038461>
- [132] P. Gusev and J. Burke, “NDN-RTC,” in *Proceedings of the 2nd International Conference on Information-Centric Networking - ICN '15*. New York, New York, USA: ACM Press, 2015, pp. 117–126. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2810156.2810176>
- [133] A. Compagno, M. Conti, P. Gasti, L. V. Mancini, and G. Tsudik, “Violating Consumer Anonymity: Geo-Locating Nodes in Named Data Networking,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9092, pp. 243–262. [Online]. Available: http://link.springer.com/10.1007/978-3-319-28166-7_12
- [134] Z. Lailari, H. B. Abraham, B. Aronberg, J. Hudepohl, H. Yuan, J. DeHart, J. Parwatar, and P. Crowley, “Experiments with the Emulated NDN Testbed in ONL,” in *Proceedings of the 2nd International Conference on Information-Centric Networking - ICN '15*. New York, New York, USA: ACM Press, 2015, pp. 219–220. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2810156.2812616>
- [135] “NitLab Website.” [Online]. Available: <https://nitlab.inf.uth.gr/NITlab/about/publicationsni>
- [136] A. M. Alberti, D. Mazzer, M. M. Bontempo, L. H. de Oliveira, R. da Rosa Righi, A. Cerqueira Sodr e, L. H. D. Oliveira, R. Righi, A. Cerqueira, and S. Jr, “Cognitive radio in the context of internet of things using a novel future internet architecture called NovaGenesis,” *Computers and Electrical Engineering*, vol. 57, pp. 147–161, 2017.
- [137] A. M. Alberti, “A conceptual-driven survey on future internet requirements, technologies, and challenges,” *Journal of the Brazilian Computer Society*, vol. 19, no. 3, pp. 291–311, 2013.
- [138] —, “Searching for synergies among future internet ingredients,” in *Communications in Computer and Information Science*, vol. 310 CCIS, 2012, pp. 61–68.
- [139] A. M. Vaz, B. M. Martins, R. C. Brand o, A. M. Alberti, A. Manuel Vaz, B. Magalhaes Martins, R. Carneiro Brandao, and A. Marcos Alberti, “Internet of information and services: A conceptual architecture for integrating services and contents on the future internet,” *IEEE Latin America Transactions*, vol. 10, no. 6, pp. 2292–2300, 12 2012. [Online]. Available: <https://ieeexplore.ieee.org/document/6418135/http://ieeexplore.ieee.org/document/6418135/>
- [140] M. Huhns and M. Singh, “Service-oriented computing: key concepts and principles,” *IEEE Internet Computing*, vol. 9, no. 1, pp. 75–81, 1 2005. [Online]. Available: <https://ieeexplore.ieee.org/document/1407782/>
- [141] M. Kido n and K. Kido n, “Evolutionary design of domain specific

- non-cryptographic hash functions,” Tech. Rep. [Online]. Available: <http://excel.fit.vutbr.cz>
- [142] “GitHub · aappleby/smhasher.” [Online]. Available: <https://github.com/aappleby/smhasher/wiki>
- [143] S. Haag and R. Anderl, “Digital twin – Proof of concept,” *Manufacturing Letters*, vol. 15, pp. 64–66, 1 2018.
- [144] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, “Internet of Things, Blockchain and Shared Economy Applications,” *Procedia Computer Science*, vol. 98, pp. 461–466, 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1877050916322190>
- [145] M. Pizka and A. Bauer, “A brief top-down and bottom-up philosophy on software evolution,” in *International Workshop on Principles of Software Evolution (IWPSE)*, 2004, pp. 131–136.
- [146] “Introducing NovaGenesis as a Novel Distributed System-Based Convergent Information Architecture.” [Online]. Available: https://www.researchgate.net/publication/323749626_Introducing_NovaGenesis_as_a_Novel_Distributed_System-Based_Convergent_Information_Architecture
- [147] A. M. Alberti, “Searching for Synergies among Future Internet Ingredients Searching for Synergies among Future Internet Ingredients,” no. August 2012, 2016.
- [148] I. V. D. C. Motta, A. M. Alberti, I. Jo, D. Camargo, and S. Rita, “Gerência e Controle como um Serviço,” *SBRC 2016 - VII WORKSHOP DE PESQUISA EXPERIMENTAL DA INTERNET DO FUTURO (WPEIF)*, 2016. [Online]. Available: http://www.sbrc2016.ufba.br/downloads/WPEIF/154499_1-Gerencia-Controle-Como-Servico.pdf
- [149] M. Hitchens and J. Rosenberg, “Bindings between names and objects in a persistent system,” in *[1992] Proceedings of the Second International Workshop on Object Orientation in Operating Systems*. IEEE Comput. Soc. Press, 1992, pp. 26–37. [Online]. Available: <http://ieeexplore.ieee.org/document/253001/>
- [150] A. M. Alberti, A. Vaz, R. Brandão, and B. Martins, “Internet of information and services (IoIS),” in *Proceedings of the 7th International Conference on Future Internet Technologies - CFI '12*. New York, New York, USA: ACM Press, 2012, p. 45. [Online]. Available: <http://doi.acm.org/10.1145/2377310.2377325http://dl.acm.org/citation.cfm?doid=2377310.2377325>
- [151] Y. Liu and B. Plale, “Survey of publish subscribe event systems,” *Indiana University Department of Computer Science*, no. TR574, pp. 1–19, 2003. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.3753&rep=rep1&type=pdf>
- [152] S. Tarkoma, M. Ain, and K. Visala, “The publish/subscribe internet routing paradigm (PSIRP): Designing the future internet architecture,” *Towards the Future Internet: A European Research Perspective*, pp. 102–111, 2009.
- [153] “RFC 20 - ASCII format for Network Interchange.” [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc20>

- [154] A. M. Alberti, L. H. D. Oliveira, and M. A. F. Casaroli, “NovaGenesis : Convergent Information Architecture,” *SBRC - WPEIF*, pp. 13–16, 2015. [Online]. Available: <http://sbrc2015.ufes.br/wp-content/uploads/proceedingsWPEIF2015.pdf>
- [155] V. H. D. Davila, E. C. Rosario, F. Simoes Miranda, and A. M. Alberti, “Nova-Genesis no Ambiente FIBRE: Desempenho da Troca de Conteúdos Nomeados,” CSBC, São Paulo, Tech. Rep., 2017.
- [156] S. Ali and S. Ali, “Virtualization with KVM,” in *Practical Linux Infrastructure*. Apress, 2015, pp. 53–80. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4842-0511-2_3
- [157] H. M. Tseng, H. L. Lee, J. W. Hu, T. L. Liu, J. G. Chang, and W. C. Huang, “Network virtualization with cloud virtual switch,” in *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*. IEEE Computer Society, 2011, pp. 222–227.
- [158] “Ansible: Up and Running: Automating Configuration Management and Deployment ... - Lorin Hochstein, Rene Moser.” [Online]. Available: https://books.google.com.br/books?hl=en&lr=&id=h5YtDwAAQBAJ&oi=fnd&pg=PP1&dq=ansible&ots=MKlf-kUPYU&sig=bZ-2GIfpKIgA8qZvThm8i6wHDo0&redir_esc=y#v=onepage&q&f=false
- [159] “Ansible Configuration Management - Daniel Hall.” [Online]. Available: https://books.google.com.br/books?hl=en&lr=&id=ETQmAgAAQBAJ&oi=fnd&pg=PT5&dq=ansible&ots=CKqWxc5OqI&sig=23Qw9dEZQ56OxmZJwYVW99Zi6qc&redir_esc=y#v=onepage&q=ansible&f=false
- [160] Nishant Kumar Singh, S. Thakur, H. Chaurasiya, and H. Nagdev, “Automated provisioning of application in IAAS cloud using Ansible configuration management,” in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, no. September. IEEE, 9 2015, pp. 81–85. [Online]. Available: <http://ieeexplore.ieee.org/document/7375087/>
- [161] “pfSense® - World’s Most Trusted Open Source Firewall.” [Online]. Available: <https://www.pfsense.org/>

Apêndice A

Ansible

Os apêndices registram alguns *scripts*, configurações técnicas e dados das experimentações. Informações que não são primárias neste trabalho mas contribuem para o entendimento das implementações técnicas realizadas. Este apêndice registra os *scripts* utilizados nas automatizações feitas com o Ansible.

A.1 Playbooks do Ansible - Scripts de execução

A.1.1 Playbook para Execução dos Cenários

Os *playbooks* executados pelo Ansible podem ser generalizados a partir do exemplo a seguir, para o cenário de 3 HTSes, o que muda em cada cenário são apenas os números de *hosts* inicializados.

```
1 FROM Playbook do Ansible para Cenário de 3 HTSes
  ↪ 01-3hts.yml
2 ---
3 - hosts: 201
4   become: true
5   tasks:
6     - name: exec clean script
7       script: clean.sh
8     - name: exec script remote
9       script: 01-3hts-core.sh
10
11 - hosts: 202, 203, 204
12   become: true
13   tasks:
14     - name: exec clean script
15       script: clean.sh
16     - name: exec script remote
17       script: 02-3hts-hts.sh
18
19 - hosts: 205
```

```
20   become: true
21   tasks:
22     - name: exec clean script
23       script: clean.sh
24     - name: exec script remote
25       script: 03-3hts-nbapp.sh
```

A.1.2 Script para Execução do Nó Core (NCN)

A inicialização do NCN é feita utilizando o *script* abaixo.

```
1 FROM Script inicialização VM Core:
   ↪ 01-3hts-core.sh
2 #!/bin/bash
3 #Script to execute NG' core in multiples xterm's
4 #BASE=`cd ../; pwd`;
5 BASE=/home/ng/workspace/novagenesis;
6 #screen -a;
7 sudo killall PGCS HTS GIRS PSS;
8 sleep 4;
9 sudo sh $BASE/Scripts/clean.sh
10 sleep 3;
11
12 sudo killall PGCS HTS GIRS PSS;
13 sleep 4;
14 sudo sh $BASE/Scripts/clean.sh
15 sleep 3;
16
17 sudo killall PGCS HTS GIRS PSS;
18 sleep 4;
19 sudo sh $BASE/Scripts/clean.sh
20 sleep 3;
21
22 sudo killall PGCS HTS GIRS PSS;
23 sleep 4;
24
25 sudo sh $BASE/Scripts/clean.sh
26 sleep 3;
27
28 sudo killall PGCS HTS GIRS PSS;
29 sleep 4;
30 sudo sh $BASE/Scripts/clean.sh
31 sleep 3;
32
33
34
```

```
35 sudo nohup $BASE/PGCS/Debug/PGCS $BASE/IO/PGCS/ 8888
   ↪ Intra_Domain -p Ethernet Intra_Domain ens3
   ↪ 52:54:00:6e:47:1e 1200 Ethernet Intra_Domain ens3
   ↪ 52:54:00:03:68:8d 1200 Ethernet Intra_Domain ens3
   ↪ 52:54:00:26:cc:4b 1200 Ethernet Intra_Domain ens3
   ↪ 52:54:00:04:d9:83 1200 &> /dev/null &
36
37 sleep 10;
38
39 sudo nohup $BASE/GIRS/Debug/GIRS $BASE/IO/GIRS/ &>
   ↪ /dev/null &
40
41 sleep 10;
42
43 #sudo nohup $BASE/HTS/Debug/HTS $BASE/IO/HTS/ &>
   ↪ /dev/null &
44
45 sudo nohup $BASE/PSS/Debug/PSS $BASE/IO/PSS/ &>
   ↪ /dev/null &
46
47 sleep 4;
```

A.1.3 Script para Execução do Nó de Armazenamento (NSN)

A inicialização do NSN é feita pelo *script* abaixo.

```
1 FROM Script inicialização VMs HTS: 02-3hts-hts.sh
2
3 #!/bin/bash
4 #Script to execute NG's core in multiple xterm's
5 #BASE=`cd ../; pwd`;
6 BASE=/home/ng/workspace/novagenesis;
7 #screen -a;
8
9 #kill and clear shared mem.
10 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
11 sleep 3;
12 sudo sh $BASE/Scripts/clean.sh
13 sleep 3;
14
15 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
16 sleep 3;
17 sudo sh $BASE/Scripts/clean.sh
18 sleep 3;
19
20 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
21 sleep 3;
```

```
22 sudo sh $BASE/Scripts/clean.sh
23 sleep 3;
24
25 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
26 sleep 3;
27 sudo sh $BASE/Scripts/clean.sh
28 sleep 3;
29
30 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
31 sleep 3;
32 sudo sh $BASE/Scripts/clean.sh
33
34 sleep 10;
35
36 sudo nohup $BASE/HTS/Debug/HTS $BASE/IO/HTS/ &>
  ↪ /dev/null &
37
38 sleep 5;
```

A.1.4 Script para execução do Nó Gerador de Tráfego TGN

O *script* que inicia a aplicação no nó gerador de tráfego (TGN) segue abaixo.

```
1 FROM Script inicialização VM Aplicação:
  ↪ 03-3hts-nbapp.sh
2
3 #!/bin/bash
4 #Script to execute NG's core in multiple xterm's
5 #BASE=`cd ../; pwd`;
6 BASE=/home/ng/workspace/novagenesis;
7 #screen -a;
8
9 #kill and clear shared mem.
10 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
11 sleep 3;
12 sudo sh $BASE/Scripts/clean.sh
13 sleep 3;
14
15 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
16 sleep 3;
17 sudo sh $BASE/Scripts/clean.sh
18 sleep 3;
19
20 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
21 sleep 3;
22 sudo sh $BASE/Scripts/clean.sh
23 sleep 3;
```

```
24
25 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
26 sleep 3;
27 sudo sh $BASE/Scripts/clean.sh
28 sleep 3;
29
30 sudo killall PGCS HTS GIRS PSS NBSimpleTestApp;
31 sleep 3;
32 sudo sh $BASE/Scripts/clean.sh
33
34 sleep 3;
35 #execute PGCS
36
37 sudo nohup $BASE/PGCS/Debug/PGCS $BASE/IO/PGCS/ 8888
   ↪ Intra_Domain -p Ethernet Intra_Domain ens3
   ↪ 52:54:00:9e:3c:25 1200 &> /dev/null
38
39 sleep 60;
40
41 sudo nohup
   ↪ $BASE/NBSimpleTestApp/Debug/NBSimpleTestApp
   ↪ $BASE/IO/NBSimpleTestApp/ &> /dev/null &
42
43 sleep 5;
```

A.1.5 Clean Script

Para execução correta de seguidas experimentações é necessário certificar que não há resquícios de execução dos códigos de outra sessão de testes. Para isso o *script* abaixo é referenciado dentro dos *scripts* de inicialização.

```
1 FROM Script de limpeza: clean.sh
2
3 #!/bin/bash
4
5 ME=`whoami`
6
7 IPCS_S=`ipcs -s | egrep "0x[0-9a-f]+ [0-9]+" | grep
   ↪ $ME | cut -f2 -d" "`
8 IPCS_M=`ipcs -m | egrep "0x[0-9a-f]+ [0-9]+" | grep
   ↪ $ME | cut -f2 -d" "`
9 IPCS_Q=`ipcs -q | egrep "0x[0-9a-f]+ [0-9]+" | grep
   ↪ $ME | cut -f2 -d" "`
10
11
12 for id in $IPCS_M; do
13     ipcrm -m $id;
```

```
14 done
15
16 for id in $IPCS_S; do
17     ipcrm -s $id;
18 done
19
20 for id in $IPCS_Q; do
21     ipcrm -q $id;
22 done
23
24 rm -rfv /dev/shm/sem.*
25
```

A.1.6 Playbook para Reboot

Durante a experimentação, diversas reinicializações dos nós são necessárias. Para executar isso de forma automatizada é utilizado o *playbook* do Ansible abaixo.

```
1 FROM Script playbook reboot.yml
2
3 ---
4 - name: Reboot all hosts except control
5   hosts: 201, 202, 203, 204, 205
6   become: true
7   tasks:
8     # Send the reboot command
9     - shell: reboot
```

A.1.7 Script para Copiar a NG

Para copiar os códigos da NG para os nós é utilizado o *playbook* do Ansible a seguir.

```
1 FROM Script playbook copy-ng.yml
2
3 ---
4 - hosts: all:!control
5   become: true
6   tasks:
7
8     - name: copy ng to all hosts
9     copy: src=/home/ng/workspace/novagenesis/
10         ↪ dest=/home/ng/workspace/novagenesis
11         ↪ mode=0777
```

A.1.8 Script para Copiar os Logs de Execução do NBSimpletestAPP

Após execução dos testes, é necessário copiar os dados obtidos pelas execuções. Para isso é utilizado o *playbook* do Ansible abaixo.


```
1 ---
2
3 - hosts: all:!control
4   become: true
5
6   tasks:
7     - name: Copy logs from nodes
8       tags: sync-pull
9       synchronize:
10        src: /home/ng/workspace/
11         → novagenesis/IO/NBSimpleTestApp/
12        dest: /home/ng/workspace/logs mode=0777
13        mode: pull          # From remote host to
14         → control
```

A.1.9 Dependências e programas complementares

O *playbook* a seguir é o responsável pela instalação das dependências do sistema operacional necessárias à execução da NG e dos códigos de automação e monitoramento.

```
1 FROM Script playbook apt-all.yml
2
3 ---
4 - hosts: lti3, lti4, lti5
5   become: true
6   tasks:
7     - name: install control components
8       apt: name={{item}} state=present
9         → update_cache=yes
10      with_items:
11        - traceroute
12        - mtr
13        - curl
14        - htop
15        - zabbix-agent
16        - nmap
17        - libzmq1
18        - libzmq-dev
19        - libzmq5
20        - wireshark
21        - xterm
22        - software-properties-common
23        - build-essential
24        - python-software-properties
25        - screen
26        - aptitude
```

26 - wakeonlan

A.1.10 Definição e agrupamento de Nós

O arquivo abaixo é o responsável por definir os nós e agrupá-los para facilitar a referência nos *playbooks* do Ansible.

```
1 FROM ansible.cfg
2 [defaults]
3 inventory = ./dev
4
5 [core]
6 192.168.180.[201:216]
7 [201]
8 192.168.180.201
9 [202]
10 192.168.180.202
11 [203]
12 192.168.180.203
13 [204]
14 192.168.180.204
15 [205]
16 192.168.180.205
17 [206]
18 192.168.180.206
19 [207]
20 192.168.180.207
21 [208]
22 192.168.180.208
23 [209]
24 192.168.180.209
25 [210]
26 192.168.180.210
27 [211]
28 192.168.180.211
29 [212]
30 192.168.180.212
31 [213]
32 192.168.180.213
33 [214]
34 192.168.180.214
35 [215]
36 192.168.180.215
37 [216]
38 192.168.180.216
39
40 [control]
```

```
41 control ansible_connection=local
42
43 [lti6]
44 192.168.170.[161:187]
45
46 [lti5]
47 192.168.170.[51:77]
48
49 [lti4]
50 192.168.170.[221:248]
51
52 [lti3]
53 192.168.170.[131:158]
54
55 [lti2]
56 192.168.170.[21:47]
57 [52]
58 192.168.170.52
59 [53]
60 192.168.170.53
61 [54]
62 192.168.170.54
63 [55]
64 192.168.170.55
65 [56]
66 192.168.170.56
67 [57]
68 192.168.170.57
69 [58]
70 192.168.170.58
71 [59]
72 192.168.170.59
73 [60]
74 192.168.170.60
75 [61]
76 192.168.170.61
77 [62]
78 192.168.170.62
79 [63]
80 192.168.170.63
81 [64]
82 192.168.170.64
83 [65]
84 192.168.170.65
85 [66]
86 192.168.170.66
```

87 [67]
88 192.168.170.67
89 [68]
90 192.168.170.68
91 [69]
92 192.168.170.69
93 [70]
94 192.168.170.70
95 [71]
96 192.168.170.71
97 [72]
98 192.168.170.72
99 [73]
100 192.168.170.73
101 [74]
102 192.168.170.74
103 [75]
104 192.168.170.75
105 [76]
106 192.168.170.76
107 [77]
108 192.168.170.77
109 [131]
110 192.168.170.131
111 [132]
112 192.168.170.132
113 [133]
114 192.168.170.133
115 [134]
116 192.168.170.134
117 [135]
118 192.168.170.135
119 [136]
120 192.168.170.136
121 [137]
122 192.168.170.137
123 [138]
124 192.168.170.138
125 [139]
126 192.168.170.139
127 [140]
128 192.168.170.140
129 [141]
130 192.168.170.141
131 [142]
132 192.168.170.142

```
133 [143]
134 192.168.170.143
135 [144]
136 192.168.170.144
137 [145]
138 192.168.170.145
139 [146]
140 192.168.170.146
141 [147]
142 192.168.170.147
143 [148]
144 192.168.170.148
145 [149]
146 192.168.170.149
147 [150]
148 192.168.170.150
149 [151]
150 192.168.170.151
151 [152]
152 192.168.170.152
153 [153]
154 192.168.170.153
155 [154]
156 192.168.170.154
157 [155]
158 192.168.170.155
159 [156]
160 192.168.170.156
161 [157]
162 192.168.170.157
163 [158]
164 192.168.170.158
165
166 [221]
167 192.168.170.221
168 [222]
169 192.168.170.222
170 [223]
171 192.168.170.223
172 [224]
173 192.168.170.224
174 [225]
175 192.168.170.225
176 [226]
177 192.168.170.226
178 [227]
```

179 192.168.170.227
180 [228]
181 192.168.170.228
182 [229]
183 192.168.170.229
184 [230]
185 192.168.170.230
186 [231]
187 192.168.170.231
188 [232]
189 192.168.170.232
190 [233]
191 192.168.170.233
192 [234]
193 192.168.170.234
194 [235]
195 192.168.170.235
196 [236]
197 192.168.170.236
198 [237]
199 192.168.170.237
200 [238]
201 192.168.170.238
202 [239]
203 192.168.170.239
204 [240]
205 192.168.170.240
206 [241]
207 192.168.170.241
208 [242]
209 192.168.170.242
210 [243]
211 192.168.170.243
212 [244]
213 192.168.170.244
214 [245]
215 192.168.170.245
216 [246]
217 192.168.170.246
218 [247]
219 192.168.170.247
220 [248]
221 192.168.170.248

Apêndice B

Configurações NG

Este apêndice registra os parâmetros utilizados na inicialização dos serviços da NG.

B.1 PGCS

```
1 FROM Script inicialização PGCS PGCS.ini
2
3 DelayBeforePublishingServiceOffer 6
4 DelayBeforeDiscovery 9
5 DelayBeforeRunPeriodic 5
6 DelayBeforeANewPeerEvaluation 6
7 DelayBeforeANewPublish 7
8 DelayBeforeFirstPeriodic 120
9 DelayBetweenMessageEmissions 10
```

B.2 GIRS

```
1 FROM Script inicialização GIRS GIRS.ini
2
3 DelayBeforeRunInitiatilization 3
4 DelayBeforeDHTPSSDiscovery 60
5 DelayBeforePIDPublishing 20
6 DelayBeforeRunExposition 5
7 DelayBeforeRunPeriodic 5
```

B.3 PSS

```
1 FROM Script inicialização PSS PSS.ini
2
3 DelayBeforeRunInitiatilization 3
4 DelayBeforeGIRSDiscovery 60
5 DelayBeforePIDPublishing 20
```

```
6 DelayBeforeRunExposition 5
7 DelayBeforeSendingToGIRS 0.000001
8 DelayBeforeSendingANotification 0.0001
9 DelayBeforeRunPeriodic 5
```

B.4 HTS

```
1 FROM Script inicialização HTS HTS.ini
2
3 DelayBeforeRunInitiatilization 3
4 DelayBeforeGIRSPSSDiscovery 60
5 DelayBeforePIDPublishing 20
6 DelayBeforeRunExposition 5
7 DelayBeforeRunPeriodic 10
```

B.5 NBSimpleTestAPP

```
1 FROM Script inicialização NBSimpleTestAPP APP.ini
2
3 DelayBeforeDiscovery 3
4 DelayBeforeRunPeriodic 60
5 NumberOfPublications 1000000 * Valor que variou de
  ↳ acordo com o cenário em avaliação: [1000000,
  ↳ 2000000, 4000000, 8000000, 16000000 ou 32000000]
6 NumberOfSubscriptions 360
7 NumberOfMessagesPerBurst 50
8 NumberOfPubsPerMessage 1000
9 NumberOfHTSs 1hts
10 Trial trial1
```


Apêndice C

Infraestrutura de Tecnologia, Informação e Comunicação

Este apêndice registra as configurações de rede e seus dispositivos utilizados para viabilização dos cenários de teste

C.1 Firwall

Regras de Firewall

É importante que qualquer experimento, serviços e/ou aplicações do ICT-LAB estejam protegidas pelo *firewall*. No Quadro C.1 são descritos os *alias* que agrupam portas, endereços IP's ou de rede para criação de tais regras.

Quadro C.1: *Detalhe dos alias de endereços e portas do firewall.*

Alias	Tipo	Informação
5G_APP	IP	192.168.180.15
AP_5G	IP	203.0.113.2, 203.0.113.3
RFC1918	NETWORK	10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
5G_APP_PORTS	PORTS	4200, 4000, 8000, 80, 443, 14000, 14001, 14002
AP_5G_PORTS	PORTS	14000, 4100, 14001, 14002, 80, 443

Fonte: Elaborado pelo autor, 2021.

As regras criadas para as interfaces: WAN, LAN, Lan180ict, Lan-scampus e OpenVPN são detalhadas nos Quadros: C.2, C.3, C.4, C.5 e C.6, respectivamente.

C.1.1 Controle e acesso de VM's

As diversas VM's configuradas no ambiente do ICT-LAB atendem a projetos de pesquisa, monitoramento, controle e seguranças das aplicações. Nesta seção elas são listadas e descritas, no Quadro C.7 é compilada as instruções para acessar cada VM.

Quadro C.2: Interface WAN - Regras de Firewall.

Protocol	Source	Port	Dest.	Port	Rule	DIRECTION
TCP/UDP	*	*	AP_5G	AP_5G_PORTS	ALLOW	INPUT
ANY	AP_5G	*	*	*	ALLOW	OUTPUT
ICMP	*	*	*	*	ALLOW	BI-DIRECTIONAL
ANY	*	*	5G_APP	*	ALLOW	INPUT
TCP	*	*	This Firewall	443, 80, 1194	ALLOW	INPUT

Fonte: Elaborado pelo autor, 2021.

Quadro C.3: Interface LAN - Regras de Firewall

Protocol	Source	Port	Dest.	Port	Rule	DIRECTION
ANY	*	*	LAN Address	443, 80	ALLOW	INPUT
ANY	LAN NET	*	*	*	ALLOW	OUTPUT

Fonte: Elaborado pelo autor, 2021.

Inteligação de switches

Na Figura C.1 é ilustrada a topologia proposta das interligações de rede para o ICT-Lab, proporcionando separação entre VLANs de cada projeto em andamento no laboratório, melhorando o desempenho da rede e seu gerenciamento. Os equipamentos são: um *firewall*/roteador virtual que é responsável por realizar o roteamento entre as VLANs e permite ter mais de uma WAN de entrada. Atualmente o laboratório possui duas WANs (Inatel e RNP). Um *switch* gerenciável o qual conecta todos os dispositivos e proporciona a segmentação da rede.

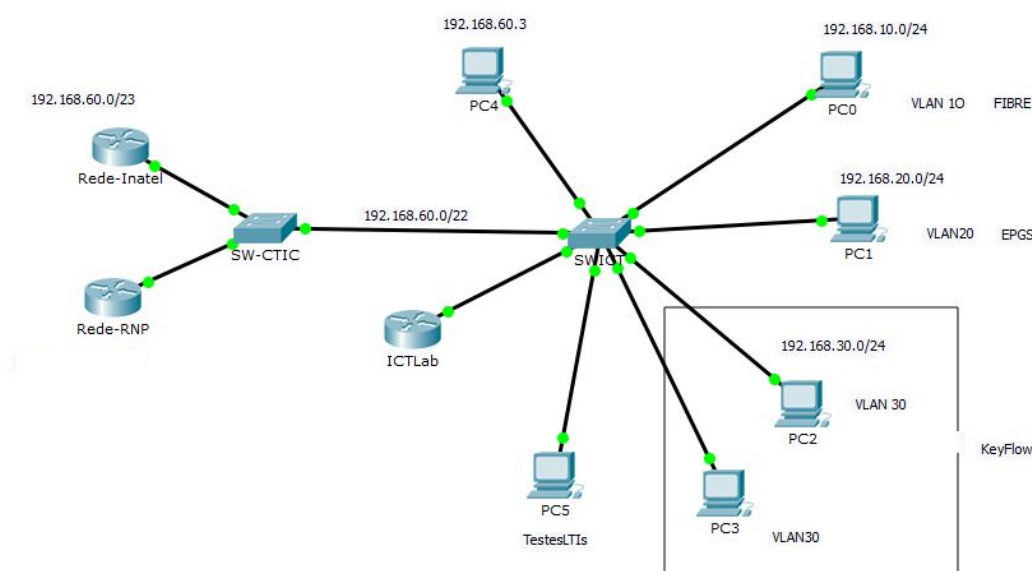


Figura C.1: Topologia Física da Rede ICTLab

Quadro C.4: Interface LAN180ICT - Regras de Firewall

Protocol	Source	Port	Dest.	Port	Rule	DIRECTION
ANY	*	*	*	*	ALLOW	OUTPUT

Fonte: Elaborado pelo autor, 2021.

Quadro C.5: Interface LAN Smart Campus - Regras de Firewall

Protocol	Source	Port	Dest.	Port	Rule	DIRECTION
ANY	*	*	*	*	DENY	BI-DIRECTIONAL

Fonte: Elaborado pelo autor, 2021.

Zabbix

Com objetivo de monitorar todo o ambiente de rede do ICT-LAB foi instalado uma VM que executa uma versão em docker do sistema de monitoramento de rede Zabbix.



Figura C.2: Tela de monitoramento do Zabbix.

Quadro C.6: Interface OpenVPN - Regras de Firewall

<i>Protocol</i>	<i>Source</i>	<i>Port</i>	<i>Dest.</i>	<i>Port</i>	<i>Rule</i>	<i>DIRECTION</i>
<i>ANY</i>	*	*	*	*	<i>ALLOW</i>	<i>BI-DIRECTIONAL</i>

Fonte: Elaborado pelo autor, 2021.

Quadro C.7: VM's ICT-LAB

VM	Host	Memória (GB)	CPU (cores)	Disco (GB)	Acesso	IP	Usuário	Senha	Aplicação	Observações
00-zabbix	NIA	1	1	20	WEB/SSH	192.168.180.3	user	*****	Zabbix	Usuário web zabbix: Admin Senha web zabbix: *****
03-5G	NIA	4	4	500	APP.WEB/SSH	192.168.180.15	user	*****	Dashboard e APP Smart Garden 5G	Regra de NAT 1:1 configurada no firewall 203.0.113.2:192.168.180.15
ict-vh-namebinds	NIA	4	4	20	SSH	192.168.180.200-216	user	*****	Testes Name Bindings	17 VM's que executam testes da NG Name Bindings.
firewall	ICT1950	4	8	20	WEB/SSH	192.168.180.1	user	*****	Firewall PFSense do ICT-LAB	Possui acesso múltiplas redes e vlans.

Apêndice D

Dados das experimentações

Este apêndice registra os resultados de publicação de NBs, assinatura de NBs e comparações entre os ambientes virtual e físico para os cenários de 1 a 4 HTSes.

D.1 Dados de Publicação e Assinatura de NBs

A Tabela D.1 exibe os resultados de publicação de NBs, assinatura de NBs e comparações entre os ambientes virtual e físico para os cenários de 1 a 4 HTSes.

Tabela D.1: *Resultados de publicação de NBs, assinatura de NBs e comparações entre os ambientes virtual e físico para os cenários de 1 a 4 HTSes.*

Cenários, ambientes e comparações	1M	2M	4M	8M	16M	32M
1 HTS Pub. Virt	0.376589639948	0.459143293875	0.542011943447	0.55189434432	0.671794429396	0.674274314826
1 HTS Pub. Fis.	0.371483914696	0.405785643282	0.4115695243	0.42617405678	0.453481003564	0.480213064839
1 HTS Pub. Virt / 1 HTS Pub. Fis.	1.0137441354794	1.13149220894422	1.31693896521823	1.29499751460681	1.48141691518769	1.40411489023536
1 HTS Sub. Virt.	0.00524465641584	0.00538652157805	0.00589444886833	0.00565255488608	0.00568058535559	0.00613924322547
1 HTS Sub. Fis.	0.0069676953486	0.00706118896667	0.00710601093305	0.00705619114584	0.00708825470223	0.00713583159362
1 HTS Pub. Virt / 1 HTS Pub. Fis.	0.752710351622046	0.762834928150951	0.829501801202552	0.801077347431621	0.801408187801555	0.860340262368155
2 HTS Pub. Virt	0.434048082705	0.475218096254	0.498294519437	0.50372302935	0.530159438002	0.563291369129
2 HTS Pub. Fis.	0.382788896941	0.411142910793	0.4332425612	0.44649816018	0.46871585276	0.502015415335
2 HTS Pub. Virt / 2 HTS Pub. Fis.	1.13390980295831	1.15584650441233	1.15015135645219	1.12816372893212	1.13108919802945	1.12205990478023
2 HTS Sub. Virt.	0.00750416749944	0.00749672180527	0.0071644664025	0.00716993701168	0.00678752057197	0.00703656662734
2 HTS Sub. Fis.	0.00905756506667	0.00915881555196	0.00922107932833	0.00925113435972	0.00922169341999	0.0092275083167
2 HTS Pub. Virt / 2 HTS Pub. Fis.	0.828497222399628	0.81852525173582	0.776966138930022	0.775033280556203	0.736038411042444	0.762564105697437
3 HTS Pub. Virt	0.667506364421	0.73956578375	0.772957409104	0.72299715827	0.80925115319	0.93266303621
3 HTS Pub. Fis.	0.354302250975	0.400047877299	0.42111869071	0.41039333864	0.434542397854	0.547430820413
3 HTS Pub. Virt / 3 HTS Pub. Fis.	1.88400260676893	1.84869318328426	1.83548587644211	1.7617175772539	1.86230654865097	1.70370940296414
3 HTS Sub. Virt.	0.00737902044584	0.00762564456974	0.00747456171138	0.00736119738055	0.007411147502693	0.0073289717789
3 HTS Sub. Fis.	0.00903260342056	0.00919336767888	0.00917637044086	0.00911100734507	0.00897308103793	0.00892175926058
3 HTS Pub. Virt / 3 HTS Pub. Fis.	0.816931741854611	0.829472380100543	0.814544460639657	0.807945499520772	0.825967691097522	0.821471591514737
4 HTS Pub. Virt	0.586237377515	0.666479370619	0.623118006186	0.64144917396	0.648666201339	0.703642089263
4 HTS Pub. Fis.	0.360915945248	0.400361096784	0.40188442402	0.41715006926	0.449107468992	0.474070003567
4 HTS Pub. Virt / 4 HTS Pub. Fis.	1.62430445435757	1.6646956359463	1.55049056132365	1.53769403681964	1.44434516485531	1.48425777621164
4 HTS Sub. Virt.	0.00822916322501	0.00792174839356	0.00785313001	0.00760113393336	0.00790133732949	0.00774208666202
4 HTS Sub. Fis.	0.00930339684613	0.00949040219277	0.00923629029583	0.00938828821363	0.00938154814398	0.00945847203492
4 HTS Pub. Virt / 4 HTS Pub. Fis.	0.884533182998976	0.834711557282047	0.850247205151784	0.809640028128302	0.842221050111028	0.818534604049869

Glossário

ASCII	Abreviação de: American Standard Code for Information Interchange. Padrão de codificação para de caracteres em meios eletrônicos (p. 46)
backbone	Estrutura central de interconexão entre diversas redes. Estabelece caminhos entre redes de diferentes tamanhos e distâncias. (p. 1)
C/C++	Linguagens de programação. (p. 38)
clean slate	An absence of existing restraints or commitments (pp. 7, 20)
CPU	Central Processing Unit - Unidade Central de Processamento ou processador (pp. xiii, 60, 63)
endpoint	Computador, <i>notebook</i> , <i>desktop</i> ou qualquer dispositivo que dê acesso à rede para os usuários.
framework	Abstração de software baseada em bibliotecas e funções genéricas que servem de base para construção da solução desejada. (pp. 13, 18)
GNU-Linux	Sistema operacional baseado no kernel linux e ferramentas GNU. (pp. 38, 62)
guest	Sistema operacional visitante, hospedado por um hypervisor ou gerenciador de máquinas virtuais (p. 14)
host	Computador ou dispositivo conectado à uma rede. (p. 4)
hypervisor	Equipamento de <i>hardware</i> ou <i>software</i> que cria, executa e gerencia máquinas virtuais (p. 14)
IEEE 802.11	Grupo de trabalho do Institute of Electrical and Electronics Engineers (IEEE) que define o padrão tecnológico para rede sem fio local (wireless local area network - WLAN). (p. 2)
IEEE 802.3	Grupo de trabalho do Institute of Electrical and Electronics Engineers (IEEE) que define o padrão tecnológico para redes cabeadas Ethernet. (p. 2)
ISP	Internet Service Provider (pp. 21, 24)
kernel	Núcleo de um sistema operacional (p. 14)
KVM	O <i>Kernel-based Virtual Machine</i> é um módulo de virtualização presente no <i>kernel</i> do Linux que permite sua utilização como <i>hypervisor</i> (pp. xiii, 14, 63)
KVM	Paravirtualização implementada através da API virtio https://www.linux-kvm.org/page/Main_Page (p. 14)
MAC	Media Access Control address (p. 75)

NSF	National Science Foundation é uma organização americana de teste, inspeção e certificação de produtos.
RSS	Really Simple Syndication: formato para distribuição de conteúdos em tempo real através de agregadores de conteúdos. (p. 45)
slicing	Virtualização de equipamentos de rede (p. 13)
socket	Software responsável pelo envio e recebimento de dados através de um rede. (p. 4)
testbed	Plataforma científica para condução de experimentos de computação e rede. (pp. 6, 8, 12, 13, 16–21, 26, 27, 30–32, 59, 60, 99)
URL	Endereço de uma página ou recurso disponível em rede/Internet (pp. 4, 22, 44)
userspace	área do usuário (p. 14)
VirtualBox	VirtualBox is a general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use. https://www.virtualbox.org (p. 14)
VMware ESXi	Discover a robust, bare-metal hypervisor that installs directly onto your physical server https://www.vmware.com/products/esxi-and-esx.html (p. 14)
WiMAX	Worldwide Interoperability for Microwave Access/Padrão da família IEEE 802.16 (p. 27)
Xen	Xen project software is extremely versatile and customizable due to its unique architecture bringing the power of virtualization everywhere. https://xenproject.org (p. 14)