



Pós-Graduação em Ciência da Computação

Ênio César Silva de Lima

UM MIDDLEWARE PARA ARQUITETURAS DE INTERNET DO FUTURO: um estudo de caso com RINA



UNIVERSIDADE FEDERAL DE PERNAMBUCO

posgraduacao@cin.ufpe.br

www.cin.ufpe.br/~posgraduacao

Recife
2019

Ênio César Silva de Lima

**UM MIDDLEWARE PARA ARQUITETURAS DE INTERNET
DO FUTURO: um estudo de caso com RINA**

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de concentração: Redes de Computadores e Sistemas Distribuídos

Orientador: Prof. Dr. José Augusto Suruagy Monteiro.

Recife
2019

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

L732m Lima, Ênio César Silva de
 Um middleware para arquiteturas de internet do futuro: um estudo de caso
 com RINA / Ênio César Silva de Lima. – 2019.
 77 f.: il., fig., tab.

 Orientador: José Augusto Suruagy Monteiro.
 Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn,
 Ciência da Computação, Recife, 2019.
 Inclui referências.

 1. Redes de computadores. 2. Internet do futuro. 3. RINA. I. Monteiro, José
 Augusto Suruagy (orientador). II. Título.

 004.6 CDD (23. ed.) UFPE- MEI 2019-103

Ênio César Silva de Lima

“Um Middleware para Arquiteturas de Internet do Futuro: Um Estudo de Caso com RINA”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 12/03/2019.

BANCA EXAMINADORA

Prof. Dr. Kelvin Lopes Dias
Centro de Informática/UFPE

Prof. Dr. Antônio Marcos Alberti
Instituto Nacional de Telecomunicações

Prof. Dr. José Augusto Suruagy Monteiro
Centro de Informática/UFPE

(Orientador)

Dedico essa dissertação à minha família, base de meu crescimento profissional, em especial: mãe.

AGRADECIMENTOS

À minha mãe, Maria de Lourdes Silva de Lima, pela dedicação na tarefa de educar e pelo carinho e amor intenso sempre.

Ao meu orientador, Dr. José Augusto Suruagy Monteiro, que confiou em minha capacidade, me incentivou e me acolheu.

Ao programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco.

Aos amigos da Secretaria de Educação do Estado de Pernambuco que me incentivaram durante o mestrado.

Por fim, agradeço a todos aqueles, que direta ou indireta contribuíram para a concretização deste trabalho, que me impulsionaram e apoiaram nesse período tão importante para a minha formação acadêmica.

“ Basicamente, um sistema de computação é a coleção de sistemas de processamento sob um regime de gerenciamento comum. ” (DAY, 2008, p. 240).

RESUMO

O cenário da Internet mudou completamente da sua origem onde poucas aplicações ocupavam a rede e, hoje em dia, em que centenas de milhões de usuários transmitem cada vez mais informações através de aplicações. Todavia a Internet não foi preconcebida para esse desígnio e as aplicações que surgiram ao longo das últimas décadas foram “adaptadas” à rede de maneira desordenada. Consequentemente, representam um conjunto significativo de problemas e desafios, haja vista que a arquitetura atual da Internet já apresenta inúmeros problemas ainda não solucionados, impedindo o atendimento dos requisitos de novas aplicações e serviços. Diante desse cenário, surgiram propostas, intituladas arquiteturas de Internet do Futuro, que preconizam a reengenharia da Internet. No entanto, o desenvolvimento de aplicações para fazerem uso dessas novas arquiteturas de rede ainda é uma atividade pouco explorada, complexa, demorada e desafiadora. Portanto, esta dissertação propõe um *Middleware* Orientado a Mensagens que visa promover o desenvolvimento de aplicações para novas arquiteturas de Internet. A finalidade é tornar menos complexa e mais produtiva a construção de aplicações para a chamada Internet do Futuro, de modo que sejam transparentes os detalhes de programação em redes ao desenvolvedor da aplicação. A proposta aborda um aspecto polimórfico, o qual acredita-se que sua arquitetura poderá suportar APIs específicas para estabelecer comunicação entre aplicações utilizando protocolos e paradigmas exclusivos de distintas arquiteturas de rede. Foi implementada uma prova de conceito do *Middleware* com a RINA, arquitetura de Internet do Futuro que se baseia no paradigma de comunicação entre processos (IPC). Na experimentação, uma aplicação teste foi desenvolvida para transmitir mensagens usando o modelo *publish/subscribe* e uma avaliação experimental do protótipo foi realizada, o que validou os componentes e as funcionalidades. Com base nos resultados obtidos, acredita-se que o *Middleware* poderá abarcar demais arquiteturas de Internet do Futuro, pois considera-se que novas aplicações poderão utilizar mais de uma proposta de rede, sendo capaz de avaliar qual arquitetura melhor lhe atende.

Palavras-chave: Internet do Futuro. *Middleware* Orientado a Mensagens (MOM). RINA.

ABSTRACT

The Internet landscape has completely changed since its origin where few applications were in the network, and today, where hundreds of millions of users are transmitting more and more information through applications. However, the Internet has not been preconceived for this design and the applications that have emerged over the last few decades have been "adapted" to the network in a disorderly fashion. Consequently, they represent a significant set of problems and challenges, given that the current Internet architecture already presents many problems that have not yet been solved, preventing the fulfillment of the requirements of new applications and services. Given this scenario, proposals have appeared, named Future Internet Architectures (FIAs), that advocate the reengineering of the Internet. However, developing applications to make use of these new network architectures is still an under-exploited, complex, time-consuming, and challenging activity. Therefore, this dissertation presents a Message Oriented Middleware that aims at promoting the development of applications for new Internet architectures. The purpose is to make the construction of applications for the so-called Internet of the Future less complex and more productive so that the details of network programming are transparent to the application developer. The proposal addresses a polymorphic aspect, which is believed to be able to support specific APIs to communicate between applications using protocols and paradigms unique to different network architectures. As a starting point, a proof of concept of the Middleware was implemented with RINA, a FIA that is based on the inter-process communication (IPC) paradigm. In the experiment, a test application was developed to transmit messages using the publish / subscribe model and an experimental evaluation of the prototype was performed, which validated its components and functionalities. Based on the results obtained, it is believed that the Middleware will be able to embrace other Future Internet architectures, since it is considered that new applications can use more than one network proposal, being able to choose the architecture which best suits their needs.

Keywords: *Future Internet. Message Oriented Middleware (MOM). RINA.*

LISTA DE FIGURAS

Figura 1 - Visão geral da proposta	18
Figura 2 - RINA: visão geral	23
Figura 3 - Estrutura da arquitetura RINA	24
Figura 4 - Exemplo de adoção da RINA com a Internet atual	27
Figura 5 - Arquitetura de <i>Middleware</i>	30
Figura 6 - Arquitetura de <i>Middleware</i> Orientado a Mensagens	31
Figura 7 - Modelo de Mensagem Point-to-Point	32
Figura 8 - Modelo de Mensagem <i>Publish/Subscriber</i>	33
Figura 9 - Arquitetura do <i>Middleware</i> PolimorFI	38
Figura 10 - Arquitetura do PolimorFI e todos os seus componentes	39
Figura 11 - Aplicação acessando diferentes arquiteturas de rede por meio do PolimorFI	41
Figura 12 - Cenário de uso: Produção e consumo de informações no modelo <i>publish/subscribe</i>	49
Figura 13 - Cenário de Uso com RINA.....	50
Figura 14 - Cenário de experimento	51
Figura 15 - Arquitetura do PolimorFI com a instância da RINA.	52
Figura 16 - Diagrama de sequência - anunciar tópico.....	54
Figura 17 - Diagrama de sequência - publicar mensagem (<i>Publish</i>)	55
Figura 18 - Diagrama de sequência - assinar tópico (<i>Subscribe</i>)	56
Figura 19 - Latência X Número de nós (100 mensagens).....	66
Figura 20 - Latência X Número de nós (500 mensagens).....	66
Figura 21 - Latência X Número de nós (1000 mensagens).....	67
Figura 22 - Comparativo geral.....	67

LISTA DE TABELAS

Tabela 1 - Aspectos chaves das arquiteturas de Internet RINA e TCP/IP	27
Tabela 2 - Consolidação da Análise comparativa RINA vs TCP/IP	29
Tabela 3 - Pacotes de <i>Software</i> do ProtoRINA acoplados ao PolimorFI.....	57
Tabela 4 - Fatores para os experimentos	62
Tabela 5 - Tempo de envio de mensagens (ms).....	64

PRINCIPAIS ABREVIACÕES

API	<i>Application Programming Interface</i>
CCN	<i>Content-Centric Networking</i>
CDAP	<i>Common Distributed Application Protocol</i>
DAF	<i>Distributed Application Facility</i>
DIF	<i>Distributed IPC Facilities</i>
DTP	<i>Data Transfer Protocol</i>
EFCP	<i>Error and Flow Control Protocol</i>
FIA	<i>Future Internet Architecture</i>
FIFO	<i>First-In, First-Out</i>
ICN	<i>Information-Centric Networking</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
IPC	<i>Inter-Process Communication</i>
MOM	<i>Middleware Orientado a Mensagem</i>
QoS	<i>Quality of Service</i>
RINA	<i>Recursive InterNetwork Architecture</i>
RPC	<i>Remote Procedure Call</i>
SCN	<i>Service-Centric networking</i>
SDN	<i>Software Defined Networking</i>
SO	<i>Sistema Operacional</i>
VM	<i>Virtual Machine</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CONTEXTO	13
1.2	MOTIVAÇÃO E JUSTIFICATIVA	15
1.3	OBJETIVOS	19
1.3.1	<i>Objetivo Geral</i>	19
1.3.2	<i>Objetivos Específicos</i>	19
1.4	METODOLOGIA.....	19
1.5	ESTRUTURA DA DISSERTAÇÃO	20
2	FUNDAMENTAÇÃO TEÓRICA.....	21
1.6	ARQUITETURAS PARA INTERNET DO FUTURO	21
1.7	UMA VISÃO GERAL SOBRE RINA.....	22
1.7.1	<i>Protocolos RINA</i>	24
1.7.2	<i>Protótipos da RINA</i>	25
1.8	COMPARAÇÃO DE ASPECTOS CHAVE: RINA VS INTERNET ATUAL	26
1.9	MIDDLEWARE	29
1.9.1	<i>Middleware Orientado a Mensagens</i>	31
1.9.2	<i>Modelos de Mensagens</i>	32
1.10	CONSIDERAÇÕES FINAIS	34
3	O MIDDLEWARE POLIMORFI.....	35
3.1	VISÃO GERAL DA PROPOSTA.....	35
3.1.1	<i>Requisitos</i>	36
3.1.2	<i>Arquitetura</i>	37
3.2	TRABALHOS RELACIONADOS	42
4	PROVA DE CONCEITO	48
4.1	CENÁRIO DE USO	48
4.2	CENÁRIO DE EXPERIMENTO	50
4.3	ARQUITETURA DO POLIMORFI.....	52
4.4	DIAGRAMAS DE SEQUÊNCIA.....	53
4.5	IMPLEMENTAÇÃO	56
4.5.1	<i>Instância da RINA</i>	56
4.5.2	<i>Implementação do PolimorFI</i>	58
4.6	CONCLUSÕES	59
5	AVALIAÇÃO EXPERIMENTAL.....	61
5.1	DEFINIÇÃO DOS EXPERIMENTOS	61
5.1.1	<i>Objetivo</i>	61
5.1.2	<i>Serviço</i>	61
5.1.3	<i>Métrica</i>	61
5.1.4	<i>Parâmetros de Carga de Trabalho</i>	62
5.1.5	<i>Preparações dos Experimentos</i>	63
5.1.6	<i>Técnica de Avaliação</i>	63
5.2	AMBIENTE DE MEDIÇÕES	63
5.3	APRESENTAÇÃO DOS RESULTADOS.....	64
5.4	CONSIDERAÇÕES FINAIS	68
6	CONCLUSÕES E TRABALHOS FUTUROS	70
6.1	CONSIDERAÇÕES FINAIS	70
6.2	CONTRIBUIÇÕES	71
6.3	TRABALHOS FUTUROS	71
	REFERÊNCIAS	72

1 INTRODUÇÃO

Esta seção relata as principais motivações para a realização deste trabalho, sua justificativa, questão de pesquisa, lista os objetivos de pesquisa almejados e, finalmente, mostra como está estruturado o restante da presente dissertação.

1.1 CONTEXTO

Os sistemas sobre IP (*Internet Protocol*) se popularizam cada vez mais, uma multiplicidade de aplicações são disponibilizadas aos usuários finais - também conhecidos como aplicações ou serviços *Over the Top* (GRZECH et al., 2012) - para os mais distintos fins. A título de exemplo pode-se citar redes sociais; Internet *banking*; ferramentas de mensagens instantâneas e troca de conteúdo multimídia, serviços de transporte privado urbano, jogos, *streaming* de mídia (vídeo e áudio), dentre outras, fazendo da Internet – e toda a sua infraestrutura – artifício indissociável da sociedade, instituições e governos.

A Internet original foi projetada em um ambiente operacional confiável de universidades e laboratórios de pesquisa. Todavia, essa suposição há muito tempo foi invalidada com a chamada comercialização da Internet. A pilha de protocolos TCP/IP, certamente, viabilizou a evolução da computação conectada dentre outros desenvolvimentos desde a sua invenção em meados da década de 70. No entanto, foi projetada para uma era na qual a comunicação era entre computadores e terminais em locais fixos, e na qual a interface do usuário era texto em vez de mídia dinâmica, como áudio e vídeo. Além disso, não se tinha a presunção que sistemas governamentais, empresas e comércios funcionariam exclusivamente por meio de aplicações conectadas à Internet. Assim, pode-se dizer que a Internet hoje é utilizada para “mover dinheiro”, mover pacotes é só uma mera consequência.

Para ilustrar algumas das limitações da Internet, as aplicações de multimídia exigem qualidade de serviço (QoS) de rede, mobilidade contínua em diferentes redes de acesso, segurança e suporte a *multicast* para entrega eficiente de conteúdo; o

esgotamento dos endereços de rede (IPv4), que inibe o desenvolvimento da denominada Internet das coisas (do inglês *Internet of Things*¹); a deficiência de suporte à mobilidade, que impede a entrega de conteúdo sensível à localização dos usuários; a incapacidade de priorizar tráfegos cruciais, que faz com que aplicações essenciais não funcionem de maneira adequada (RNP, 2011).

Ao conectar “objetos” com diferentes recursos a uma rede, potencializa-se o surgimento de novas aplicações. Deste modo, as demandas novas aparecem mostrando que os serviços da arquitetura da Internet atual não atendem adequadamente algumas aplicações. Tem-se a previsão que o mercado de aplicações movimente US\$ 139 bilhões em 2021 (CANALTECH, 2017). Com uma infinidade de novas aplicações encontrando novos usos da Internet, certamente os protocolos do núcleo da rede serão uma grande preocupação para a próxima geração. Isso motivou várias iniciativas para propor a substituição da pilha de protocolos TCP/IP. Em tal caso, surgem novos paradigmas de comunicação para aplicações emergentes (LI, 2018).

A fim de solucionar essas e outras limitações, surgiram várias soluções apresentadas como extensão/adição/modificação ao protocolo TCP/IP, mas suas implementações são bastantes desafiadoras e levantam outro problema em relação ao gerenciamento de um plano de controle complexo. Desde a década de noventa, novas arquiteturas de rede para uma Internet do Futuro surgiram e este tema ainda tem chamado atenção da comunidade de pesquisa.

Propostas como NewArch², *Future Internet Design* (FIND)³, Arquiteturas de Internet do Futuro (AMBROSIN et al., 2016; BOUK et al., 2017; PAN; PAUL; JAIN, 2011) e *Future Internet Architecture Next Fase* (FISHER, 2014); alavancaram múltiplos projetos que se concentraram em novas arquiteturas de rede.

Concernente às arquiteturas de Internet do Futuro, vêm sendo investigadas e polemizadas duas abordagens: a primeira, alcunhada de evolucionária (REXFORD; DOVROLIS, 2010), (PAN; PAUL; JAIN, 2011b) pretende efetuar modificações incrementais na arquitetura da Internet atual, sem perder a compatibilidade com ela. Em contrapartida, tem-se a segunda abordagem – denominada de revolucionária ou *clean*

¹ *Internet of Things* - termo que designa a futura geração de eletroeletrônicos e qualquer outro tipo de objeto capaz de se comunicar via internet, compartilhando informações e interagindo com outros dispositivos.

² <https://www.isi.edu/newarch/>

³ <http://www.nets-find.net/>

slate – que visa trocar a arquitetura atual da Internet por uma nova, integralmente reconstruída (BRAUN; MONTI, 2013).

Nesse sentido, entende-se que, no mínimo, as pesquisas em Internet do Futuro irão ter forte influência na evolução da Internet atual. De outro lado, FERNANDES et al. (2011) apresentam uma abordagem pluralista que se baseia na ideia de que a Internet deve suportar várias pilhas de protocolos simultaneamente, onde um conjunto de propostas para novas arquiteturas coexistam entre si e com a Internet atual podendo dar origem a novas infraestruturas globais de informação, potencializando novos negócios, aplicações inovadoras, novas WWWs (*World Wide Webs*) integradas à atual, possibilitando que seja desenvolvido um conjunto pluralista de aplicações.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

Será que a internet atual está pronta para os desafios impostos pela grande quantidade de aplicações para “coisas” que estarão interoperando e compartilhando informações na rede? Será que a estrutura TCP/IP sustentará adequadamente a evolução das aplicações nos próximos decênios? Qual a principal razão de grandes pesquisas avançarem na missão de querer mudar a arquitetura da Internet? Por que mudar para uma nova Internet? Qual a real importância das propostas de novas arquiteturas para a Internet? Como elas podem impactar, no cenário atual das Tecnologias da Informação e Comunicação (TIC) emergentes e futuras no mundo? Embora as respostas para essas indagações sejam escopo de grande debate, acredita-se que a reengenharia da Internet é estratégica para o futuro das redes e de suas aplicações, cada vez mais pervasivas, e almeja beneficiar uma ampla gama de serviços e aplicações. Entretanto, compreende-se que os esforços em Internet do Futuro devem evoluir também para a camada de aplicações e implantar futuras aplicações de Internet sobre novas arquiteturas de rede.

A Internet é utilizada mundialmente por centenas de milhões de usuários que consultam, transmitem e recebem cada vez mais informações através de aplicações. Todavia, como esta rede não foi preconcebida para esse propósito, as aplicações que surgiram ao longo das décadas foram sendo “adaptados” à rede de maneira desordenada. Além disso, a arquitetura atual da Internet já apresenta inúmeros problemas ainda não solucionados, impedindo o atendimento dos requisitos de novas aplicações e serviços. Dessa maneira, tecnologias futuras devem lidar com a

complicação, limitações e imperfeições das tecnologias precedentes (FERNANDES et al., 2011), (JÄPPINEN; GUARNERI; CORREIA, 2013), (ICIN, 2019).

O cenário de rede mudou consideravelmente das raízes da Internet, onde apenas algumas aplicações distintas ocuparam a rede. Atualmente, a rede precisa fornecer conectividade entre bilhões de usuários finais, executando uma grande variedade de aplicações distribuídas, permitindo também a mobilidade deles na maioria dos casos (LEON et al., 2017). O modelo de serviço de melhor esforço da Internet é uma má combinação para muitas aplicações em tempo real, como por exemplo IPTV e videoconferência (REXFORD; DOVROLIS, 2010).

Outrossim, espera-se que a Internet do Futuro seja mais ágil, escalável, segura e confiável. Enquanto isso, testemunha-se o desenvolvimento e o crescimento sem precedentes de novas aplicações e serviços nos últimos anos, desde serviços baseados em localização, redes sociais, computação em nuvem e aplicações baseados em P2P (ponto-a-ponto). Essas aplicações que estão surgindo rapidamente, com diferentes requisitos e implicações para o *design* da Internet do futuro, representam um conjunto significativo de problemas e desafios (LI et al., 2010).

Os campos de novas aplicações são muitos e abrangem desde as cidades inteligentes até os sistemas de transporte inteligente, e de sistemas para a saúde e proteção ambiental para *Smart Grids* (ITT, [s.d.]). Todas essas áreas de aplicação são exemplos concretos da chamada convergência Ciber-Física (RITCHEY DIANE, 2018), na qual componentes do mundo real interagem com o ciberespaço via sensores, computação e elementos de comunicação.

De acordo com CASTA; JAIN; HAUSI (2013), as abordagens pesquisadas da Internet do Futuro não apresentam quaisquer aplicações como uma experiência final com resultados convincentes. Para os autores o nível de maturidade para o futuro da Internet mundial está em um foco de estágio na resolução de problemas de arquitetura e de rede antes de passar para a camada de aplicações e implantar futuros aplicações de Internet.

Apesar de haver progresso em pesquisas sobre arquiteturas de Internet do Futuro, o desenvolvimento de aplicações para fazerem uso dessas novas redes ainda é uma atividade pouco explorada, complexa, demorada e desafiadora. Isso ocorre porque as

propostas envolvem uma ampla variedade de conceitos de *design*, paradigma de comunicação, protocolos, identificação dos nós, segurança, dentre outros aspectos.

Determinadas abordagens para projetar e construir redes são abordadas na literatura, alguns exemplos são: comunicação Inter-processos - *Inter-Process Communication* - IPC (SMALL; DAY; CHITKUSHEV, 2011) ; redes centradas em serviços - *Service-Centric networking* – SCN (BRAUN et al., 2011); redes centradas em conteúdo - *Content-Centric Networking* – CCN (BRAUN; MONTI, 2013); e redes centradas em informações - *Information-Centric Networking* – ICN (SIRIS et al., 2012), (ALBERTI et al., 2017).

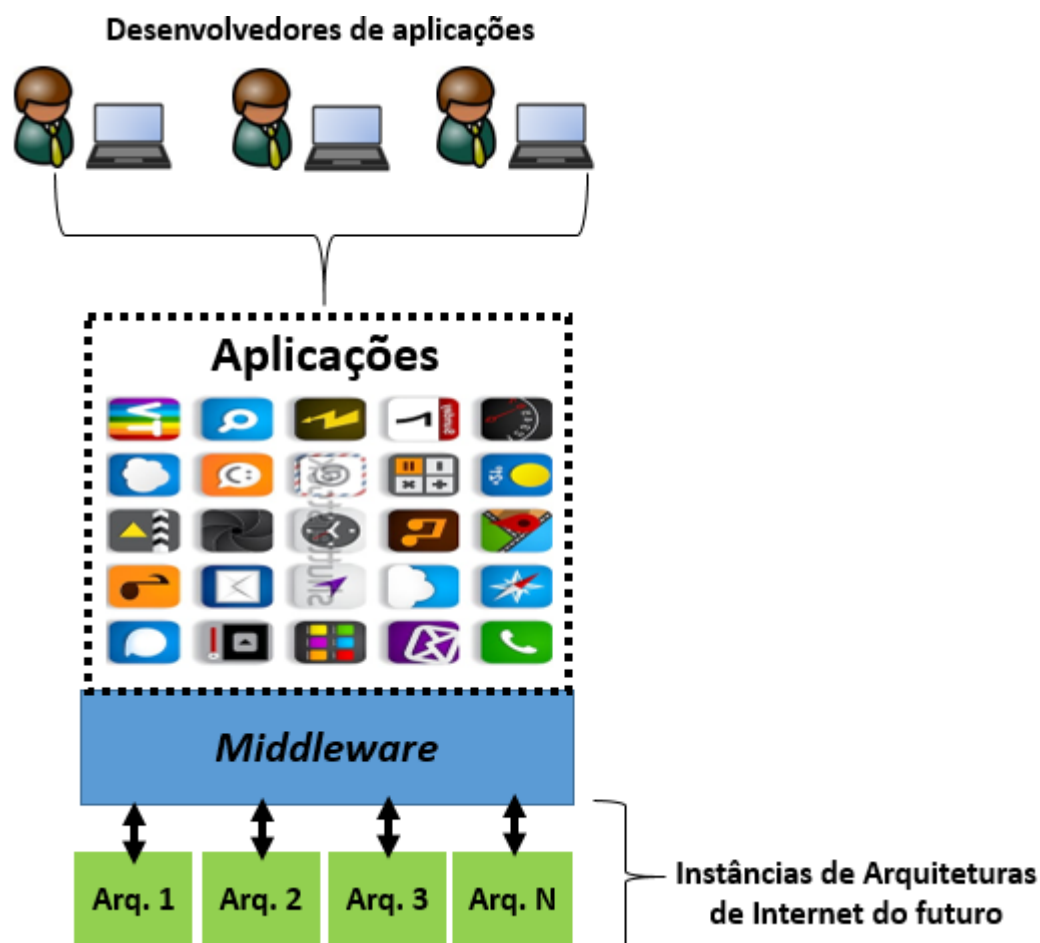
Face ao exposto, este trabalho considera que a melhor abordagem é criar as condições para uma Internet do Futuro onde aplicações desenvolvidas utilizam-se de serviços disponíveis na arquitetura de Internet do Futuro que melhor lhe atendam. A partir disso, propõe-se um *Middleware* Orientado a Mensagens que visa oferecer aos desenvolvedores de aplicações, pesquisadores da academia e da indústria, o projeto e testes de aplicações distribuídas para arquiteturas de Internet do Futuro, permitindo que aplicações multiarquitetura sejam criadas e avaliadas. Considera-se ainda que, possivelmente, novas aplicações podem utilizar mais de uma proposta ao mesmo tempo (utilizando mais de uma API), podendo avaliar em tempo real qual arquitetura melhor lhe atendem.

A

Figura 1 apresenta uma visão geral da proposta. Nesta representação, pode-se observar a presença de alguns atores, os quais são definidos a seguir:

- **Desenvolvedor de Aplicações:** indivíduo responsável por desenvolver as aplicações que utilizarão as arquiteturas de rede abarcadas no PolimorFI.
 - **Aplicações:** Qualquer aplicação desenvolvida para ser executada nas arquiteturas de rede suportadas pelo PolimorFI.
 - **Instâncias de Arquiteturas de Internet do Futuro:** Correspondem à existência de componentes da arquitetura de Internet do Futuro propriamente dita. Cada arquitetura oferece serviços específicos.
-

Figura 1 - Visão geral da proposta



Fonte: Elaborada pelo autor

Nesse contexto, os projetistas devem receber abstrações de modelagem e programação adequadas para desenvolver aplicações inovadoras, capazes de se beneficiar das arquiteturas de Internet do Futuro.

A implantação de um *Middleware* direcionado às aplicações da Internet do Futuro promoverá uma estrutura de aplicações de modo que elas aproveitem o que cada arquitetura de rede propicia de melhor, reduz a quantidade de codificação que o desenvolvedor necessita escrever. Isto gera uma produtividade e a confiabilidade das aplicações, garantindo uma padronização nos projetos a serem desenvolvidos.

A partir da implementação é plausível reutilizar a codificação da aplicação para novas aplicações, tendo como benefício a produtividade e a confiabilidade do funcionamento de aplicações a serem desenvolvidas. Desta forma garante-se um desenvolvimento com segurança e um produto final padronizado e confiável.

1.3 OBJETIVOS

Os objetivos desta dissertação são apresentados abaixo.

1.3.1 *Objetivo Geral*

Diante da sensibilidade da Internet atual, quanto ao desenvolvimento e o aumento sem precedentes de novas aplicações e serviços nos últimos anos, que pode impactar, por exemplo, o tráfego da rede, são crescentes os desafios e a preocupação com diferentes requisitos e implicações para o futuro *design* da Internet.

Neste contexto, o objetivo geral do presente trabalho é projetar um *Middleware* Orientado a Mensagens, denominado PolimorFI, para facilitar o desenvolvimento de aplicações distribuídas para arquiteturas de Internet do Futuro.

1.3.2 *Objetivos Específicos*

- Definir uma arquitetura que permita o desenvolvimento de aplicações de forma que seja transparente ao desenvolvedor detalhes de programação em redes;
- Definir aspectos de polimorfismo para que o *Middleware* possa suportar várias pilhas de protocolos específicos de arquiteturas de Internet do Futuro distintas;
- Fornecer funcionalidades para a criação de aplicações para arquiteturas de Internet do Futuro;
- Formular interfaces para suportar instâncias de arquiteturas de Internet do Futuro que se propuserem a ser adaptadas ao PolimorFI.

1.4 METODOLOGIA

A fim de atestar a viabilidade da proposta, será implementada uma prova de conceito por meio de uma prototipação e realizada uma avaliação experimental preliminar do *Middleware* PolimorFI. O método utilizado toma como base a proposta apresentada por JAIN (1991). A referida metodologia indica as etapas necessárias para

planejar, executar, coletar dados e analisar os resultados em projetos de sistemas computacionais.

1.5 ESTRUTURA DA DISSERTAÇÃO

Na seção de introdução foram apresentados o contexto do problema, motivação e justificativa, o objetivo, a metodologia utilizada e os trabalhos relacionados à pesquisa. O restante da dissertação está estruturado da seguinte maneira:

- **Seção 2:** Apresenta os principais aspectos teóricos, fundamentação deste trabalho.
 - **Seção 3:** Expõe o *Middleware* PolimorFI, a visão geral da proposta, apresenta os requisitos e a arquitetura do *Middleware*. Por fim, são discutidos os trabalhos relacionados.
 - **Seção 4:** Demonstra a implementação da prova de conceito e pormenoriza o experimento.
 - **Seção 5:** Explana a avaliação experimental do protótipo do *Middleware*. No final, os resultados são analisados e são tiradas as conclusões.
 - **Seção 6:** Compreende as conclusões sobre o desenvolvimento do trabalho, aponta as contribuições e sugestões para trabalhos futuros.
-

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os fundamentos teóricos que respaldaram o desenvolvimento da presente dissertação. Inicialmente, será apresentada uma visão geral sobre arquiteturas de Internet do Futuro, em seguida será apresentada a RINA, por conseguinte, elucidados conceitos sobre: sistemas distribuídos, *Middleware*, paradigma de comunicação Inter-processos e o modelo de comunicação *publish / subscribe*.

1.6 ARQUITETURAS PARA INTERNET DO FUTURO

O conceito de *design clean slate* da Internet do Futuro foi idealizado como uma preocupação mundial, reunindo pesquisadores da academia, indústria e governo para unir esforços na construção de uma infraestrutura mais adequada para enfrentar os atuais desafios da Internet e as necessidades futuras (CASTA; JAIN; HAUSI, 2013).

Por outro lado, a rede do futuro não necessariamente se chamará de Internet (MONTEIRO, 2018). Inclusive existem propostas de abordagens pluralistas aproveitando os conceitos de virtualização e redes definidas por *software* (SDN), onde possivelmente haverá arquiteturas distintas coexistindo numa infraestrutura comum de suporte sendo a Internet atual, provavelmente, uma delas (FERNANDES et al., 2011).

O fato da Internet ter sido um sucesso não significa que as pesquisas em redes estejam maduras. O cenário atual, com relação às arquiteturas de Internet do Futuro, mostra que existem várias arquiteturas de rede com protótipos em diferentes estágios de implementação, diferentes arquiteturas com objetivos de *design* específicos e paradigmas de comunicação. Assim, não é possível identificar uma arquitetura de rede única que surja como uma resposta e resolva todas as limitações atuais da Internet (SIRIS et al., 2012), (BOUK et al., 2017).

Desta forma, a busca por uma arquitetura de rede única capaz de unificar os requisitos de comunicação desconectados, que pode suportar uma nova Internet, é uma corrida possível, no entanto, esta pesquisa não parece ser a melhor abordagem.

Durante a última década, órgãos de financiamento de pesquisa alocaram dinheiro para financiar projetos de Internet do Futuro, em especial *clean slate*, que poderiam reduzir a complexidade da rede ao redesenhar a arquitetura do protocolo de rede, questionando alguns de seus princípios chave (ICIN, 2019). Todavia, poucas iniciativas realmente conseguiram questionar o modelo central e os princípios subjacentes dos atuais protocolos da Internet. Das iniciativas que fizeram isso, o RINA é provavelmente a solução mais simples e mais geral (AMDOUNI et al., 2016).

Na seção seguinte, será apresentada e descrita a arquitetura de Internet do Futuro inicialmente suportada pelo PolimorFI, a RINA.

1.7 UMA VISÃO GERAL SOBRE RINA

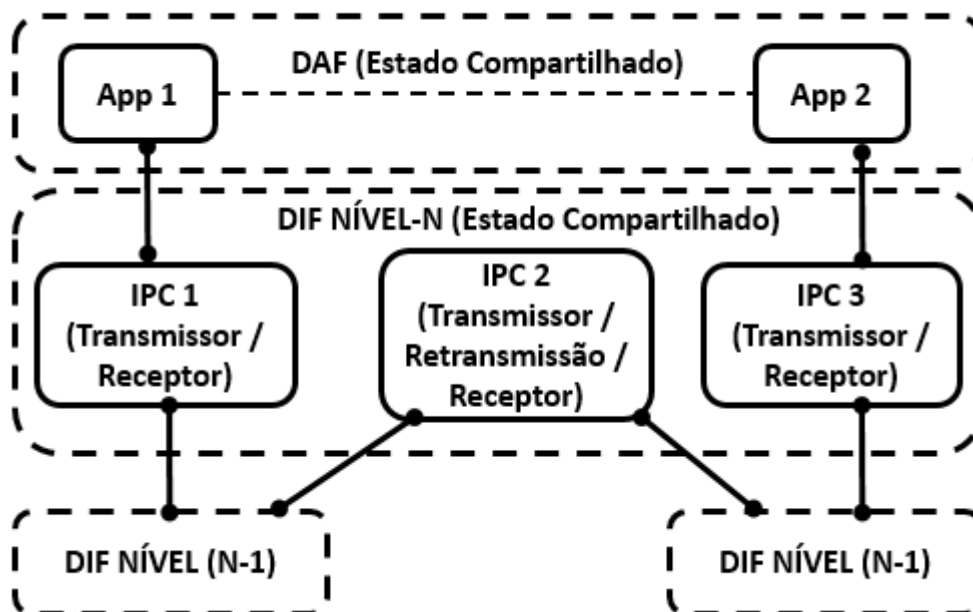
A arquitetura inter-redes recursiva (RINA) é uma proposta de arquitetura de Internet do Futuro *clean slate* originada por DAY (2008), ela parte do princípio de que as camadas de rede nada mais são do que recursos de comunicação entre processos (IPCs) - e somente IPC (VRIJDERS et al., 2014) – distribuídos, aplicados recursivamente em diferentes escopos e políticas de acordo com as necessidades dos usuários.

A comunicação Inter-processos (IPC) refere-se ao conjunto de métodos que permitem a troca de dados entre os processos. Quando dois processos são remotos, conectados por meio de uma rede, o IPC é realizado através do *socket* e da pilha de protocolo da rede implementada nos nós finais (GALLO et al., 2014).

A partir desses preceitos, a RINA intenciona resolver vários problemas presentes na Internet atual, como por exemplo a mobilidade de aplicações e gerenciamento de tráfego eficiente.

Na RINA, cada camada é chamada genericamente de *Distributed Application Facility* (DAF), como mostrado na Figura 2, sendo que camadas inferiores são DAFs especializados, denominados de *Distributed IPC Facilities* (DIFs) cuja aplicação realiza apenas a comunicação entre processos (TEYMOORI et al., 2016).

Figura 2 - RINA: visão geral

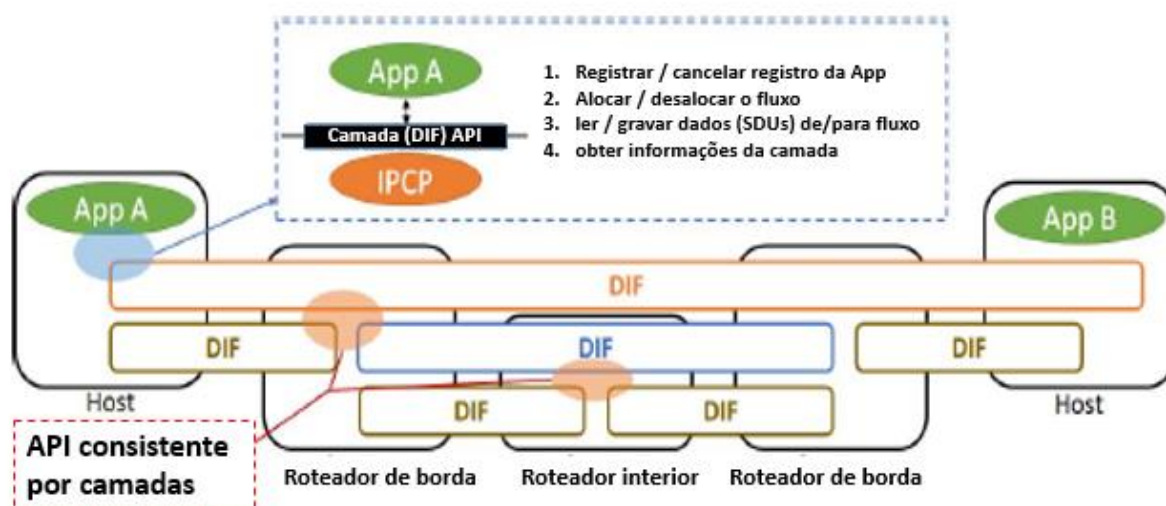


Fonte: Adaptada de WANG; MATTA; AKHTAR (2016)

A rede fornece os meios pelos quais os processos em sistemas de computadores separados se comunicam, generalizando o modelo de comunicações locais entre processos. Um DIF é uma estrutura de organização, agrupando processos de aplicações que fornecem serviços de IPC. Um DIF pode ser visto como o que geralmente é chamado de “camada” (STAESSENS, 2015).

De acordo com essa visão, a rede não é um conjunto de diferentes funções, mas uma única camada de IPC distribuído que se repete em diferentes escopos - ou seja, fornecendo as mesmas funções / mecanismos - que podem ser sintonizados com políticas diferentes para operar em diferentes faixas do espaço de desempenho (por exemplo, capacidade, atraso, perda). Conforme ilustrado na Figura 3.

Figura 3 - Estrutura da arquitetura RINA



Fonte: Adaptada de GAIXAS et al. (2017)

Cada IPC é responsável pelo roteamento e também pelo gerenciamento. O roteamento proposto na RINA sugere que cada IPC tenha uma preocupação maior com o próximo salto do que com o caminho inteiro proporcionando com isso o que os idealizadores da proposta chamam de roteamento recursivo.

O RINA abre as portas para um grau de automação sem precedentes no gerenciamento de rede, ela permite alterações sofisticadas de configuração em várias camadas da rede de uma só vez, minimizando o risco de causar paralisação do serviço (CROTTY; PUENTE, 2016).

1.7.1 Protocolos RINA

Cada camada é composta por apenas dois protocolos programáveis: um executa funções de transferência de dados e controle de transferência de dados, enquanto o outro executa funções de gerenciamento de camadas (WANG; MATTA; AKHTAR, 2016). Os referidos protocolos são especificados abaixo:

➤ **Common Distributed Application Protocol (CDAP)**

O CDAP é usado comunicando aplicações RINA para trocar quaisquer dados específicos de aplicações estruturadas necessárias para coordenar DIFs. É o protocolo de aplicação utilizado pelos Processos IPC para troca de estado (Processos IPC são processos de aplicação). Uma entidade CDAP (*Management Application Entity*) coleta

informações de outros pares no DIF em resposta a eventos (por exemplo, falhas) na rede, periodicamente ou como uma questão de política imposta por seus usuários. Alguns desses eventos podem envolver todos os membros do DIF ou ter um escopo mais limitado.

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

➤ **Error and Flow Control Protocol (EFCP)**

EFCP é o protocolo de transferência de dados necessário para manter uma instância de IPC dentro de um DIF. As funções deste protocolo garantem confiabilidade e controle de fluxo conforme necessário. O projeto original do EFCP é modelado a partir do protocolo de transmissão Delta-t de Richard Watson (DAY, 2008) e inclui um Protocolo de Transporte de Dados (DTP) e um Protocolo de Controle de Transporte de Dados (DTCP).

Na RINA, o protocolo de transferência de dados EFCP utiliza os mesmos mecanismos nas DIFs como, por exemplo, controles de fluxo e de congestionamento em conjunto com políticas para tratar os problemas relacionados ao congestionamento na rede. Políticas como as de alocação, reordenação e marcação de mensagens são características na RINA que podem ser tratadas de maneiras similares ou diferenciadas nas DIFs.

1.7.2 Protótipos da RINA

Vários trabalhos foram feitos com o objetivo de validar os princípios da arquitetura RINA, desde a publicação do livro que a originou, intitulado “*Patterns in Network Architecture: A Return to Fundamentals*” (DAY, 2008).

Pode-se citar, como exemplos de ferramentas desenvolvidas para avançar nas especificações da RINA, os projetos abaixo:

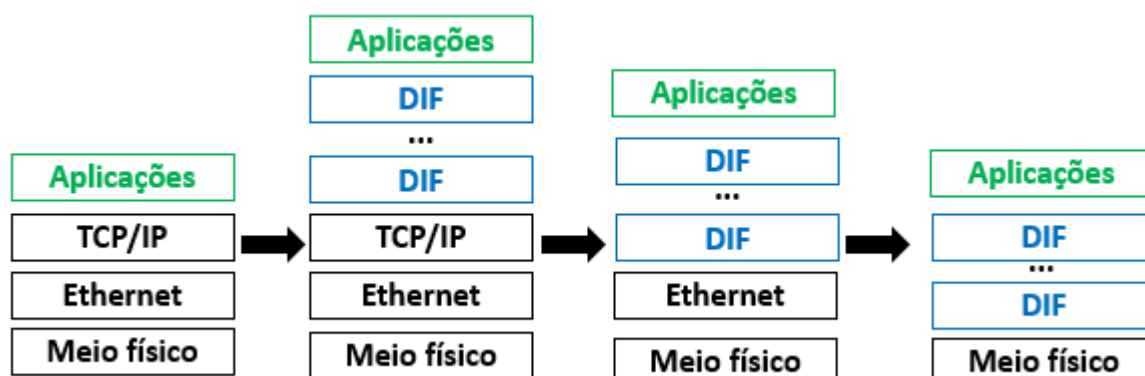
- **RUMBA** (VRIJIDERS et al., 2018) – um framework de experimentação *open source* para a instalação e *bootstrapping* de redes recursivas em vários *testbeds*;
- **ARCFIRE** (MEER et al., 2017) – projeto que tem como objetivo mostrar os benefícios e a viabilidade da RINA através de implementações experimentais em larga escala;
- **IRATI** (MAFFIONE et al., 2016) – projeto que teve como objetivo construir um protótipo que execute em espaço de kernel com a finalidade de testar os conceitos da RINA em uma implementação com qualidade para ser utilizada como espelho de uma implementação comercial;
- **ProtoRINA** (WANG et al., 2014) – protótipo da RINA que executa em espaço de usuário. Ele foi desenvolvido na Universidade de Boston e seu objetivo principal é validar a arquitetura RINA e explorar a programabilidade da arquitetura.

1.8 COMPARAÇÃO DE ASPECTOS CHAVE: RINA VS INTERNET ATUAL

A RINA parte do pressuposto que a substituição da pilha TCP/IP pode ser realizada de maneira gradativa, em outros termos, RINA não objetiva modificar a Internet atual, portanto objetiva adaptar-se e interoperar com a Internet atual, de modo que novos recursos e aplicações sejam construídos na RINA (DAY, 2018).

Por meio da sua estrutura recursiva, a RINA permite que DIF's sejam implementadas em uma camada acima do IP, abaixo ou até mesmo ao lado (DAY, 2018). A Figura 4 ilustra um simples exemplo de adoção que se fundamenta nesta elasticidade das DIF's, isto é, na característica de poder implementá-las na camada que se deseja. Um primeiro passo dessa estratégia é implantar as DIF's acima da camada TCP/IP e, a partir disto, mover a estrutura para camadas mais abaixo até atingir o cenário desejado, que é a completa adoção

Figura 4 - Exemplo de adoção da RINA com a Internet atual



Fonte: Adaptada de DAY (2018)

O modelo representado na Figura 4 corrobora com a ideia preconizada por DAY (2018) de que a Internet não pode ser corrigida, mas funcionará melhor com o RINA. Portanto, esta arquitetura pode conviver uniformemente com a Internet atual e ser desenvolvida gradativamente.

Considerando isso, foi realizada uma análise comparativa entre ambas as arquiteturas de rede, ponderando alguns aspectos chave mencionados na Tabela 1.

Tabela 1 - Aspectos chaves das arquiteturas de Internet RINA e TCP/IP

ID	ASPECTO CHAVE
01	Paradigma de Comunicação
02	Recurso Principal
03	Identificação e localização
04	Identificação de entidades
05	Endereçamento
06	Pilha de Protocolo
07	Roteamento

Fonte: Elaborada pelo autor

ID 01 - Paradigma que governou o *design* de cada projeto:

- RINA adota uma abordagem centrada no processo com base no IPC.
- O TCP/IP é centralizado no host

ID 02 - Principal característica das propostas:

- RINA se concentra no fornecimento de IPC recursivo, em camadas e personalizado;
- O TCP/IP visa fornecer comunicação de ponta a ponta.

ID 03 - Acoplamento de identificação e localização na arquitetura:

- No TCP/IP, os endereços IP são acoplados,
- Na RINA as abordagens de desacoplamento são empregadas.

ID 04 - Identificação de entidades:

- A RINA acomoda serviços, processos de aplicações, DAF/DIF, portas e IDs de terminais de conexão;
- A Internet atual emprega IDs de sistemas autônomos, nomes de domínio, FQDN (Nome de Domínio Totalmente Qualificado), portas de camada de transporte e endereços PoA (Point of Attachment) (IPv4 / IPv6)

ID 05 - Endereçamento de entidades:

- A RINA possui localizadores para Nó, PoA e endereço IPCP⁴, que é único em um determinado DIF;
- O TCP/IP emprega uma abordagem de nomenclatura dupla, onde representa os IDs e os localizadores

ID 06 - Protocolo de empilhamento:

- RINA adota o empilhamento flexível, no qual novas camadas podem ser adicionadas sob demanda.
- TCP/IP adota um empilhamento fixo.

Em particular, o RINA emprega o mesmo modelo para todas as camadas, permitindo sua personalização dependendo da necessidade.

ID 07 - Roteamento:

⁴ IPCP trata-se de um Nó interno de uma DIF.

- O roteamento RINA é baseado em endereços IPCP topológicos que são dependentes de localização, mas independentes de rota;
- TCP/IP emprega prefixos de rede e endereços de PoA.

A fim de facilitar o entendimento, a análise comparativa foi consolidada abaixo na Tabela 2.

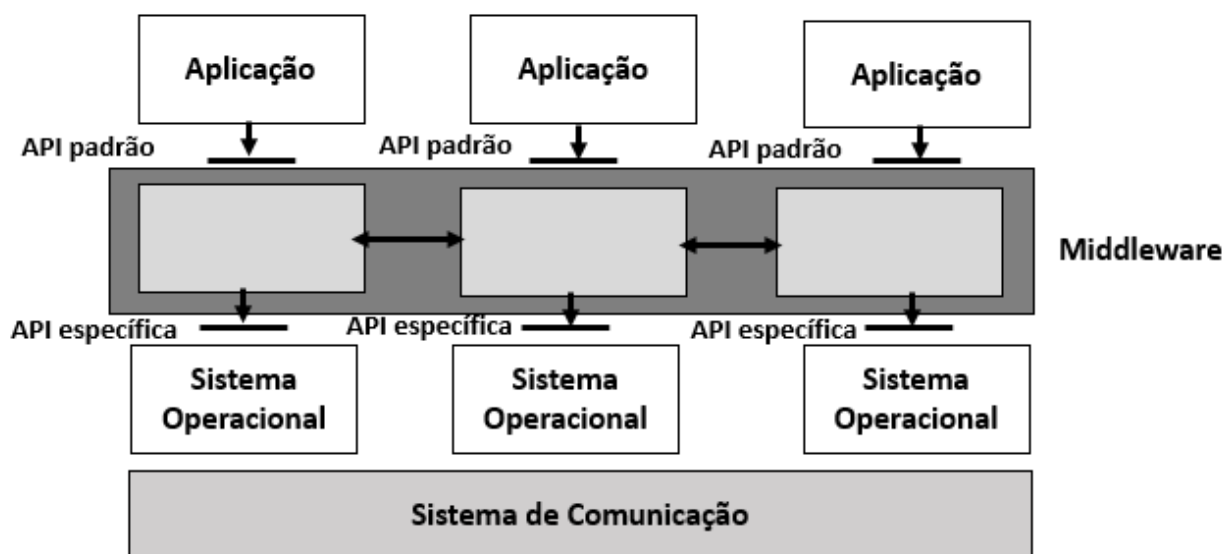
Tabela 2 - Consolidação da Análise comparativa RINA vs TCP/IP

ASPECTO CHAVE	RINA	TCP/IP
Paradigma de comunicação	Centrado a processo (baseado em IPC)	Centrado a host
Recurso principal	Cada camada fornece IPC em um determinado escopo	Entrega de pacotes fim a fim
Identificação e localização	Desacoplado	Acoplado
Identificação	Serviços, processo de aplicação. DAF/DIF, portas, endpoints de conexão, e QoS.	Sistemas autônomos, nomes de domínio, FQDN, portas, camada de transporte, endereços PoA (IPv4 / IPv6)
Endereçamnto	Node, PoA, IPCP-address (exclusivo em um determinado DIF)	Endereços PoA, camada de transporte, portas
Pilha de protocolo	Recursivo, flexível	Fixo
Roteamento	Endereço IPCP topológico	Prefixos de rede e endereços de PoA

Fonte: elaborada pelo autor.

1.9 MIDDLEWARE

Conforme COULORIS et al. (2011), *Middleware* é uma camada de *software* posicionada entre o Sistema Operacional (S.O.) e as aplicações distribuídas, conforme ilustrado na Figura 5. Nesse contexto, as aplicações distribuídas são componentes lógicos localizados em dispositivos interligados que se comunicam e coordenam suas ações apenas por mensagens.

Figura 5 - Arquitetura de *Middleware*

Fonte: Adaptada de SOUSA (2017)

O *middleware* tem a tarefa de fornecer uma abstração de programação para o desenvolvimento de sistemas distribuídos e, através de camadas lógicas, abstrair a heterogeneidade da infraestrutura subjacente para promover a interoperabilidade e a portabilidade (SOUSA, 2017, RAZZAQUE et al., 2016).

De acordo com WILCOX et al. (2015), os sistemas distribuídos são difíceis de implementar corretamente, pois precisam lidar com simultaneidade e falhas: os nós podem travar em pontos arbitrários e as redes podem reordenar, descartar ou duplicar pacotes.

O *middleware* também tem fundamental importância no controle de processos em computadores diferentes, oferecendo suporte à localização e nomeação de recursos distribuídos controlando a consistência dos dados distribuídos SOUSA (2017). Existem diversas plataformas de *middleware* e estas podem ser classificadas segundo suas primitivas de interação (VALENTE, 2011). Ressalta-se, ainda, a existência de plataformas híbridas que, são soluções com suporte a recursos não exclusivos de uma única arquitetura (COULOURIS et al., 2011).

As principais categorias de *middleware* conforme suas primitivas de comunicação são: *Middleware* Procedural, *Middleware* Transacional, *Middleware* Orientado a Objetos (MOO), e *Middleware* Orientado a Mensagens (MOM) (SOUSA, 2017). Esta dissertação

irá delimitar a explanação ao MOM, categoria que corresponde à arquitetura de *Middleware* proposta.

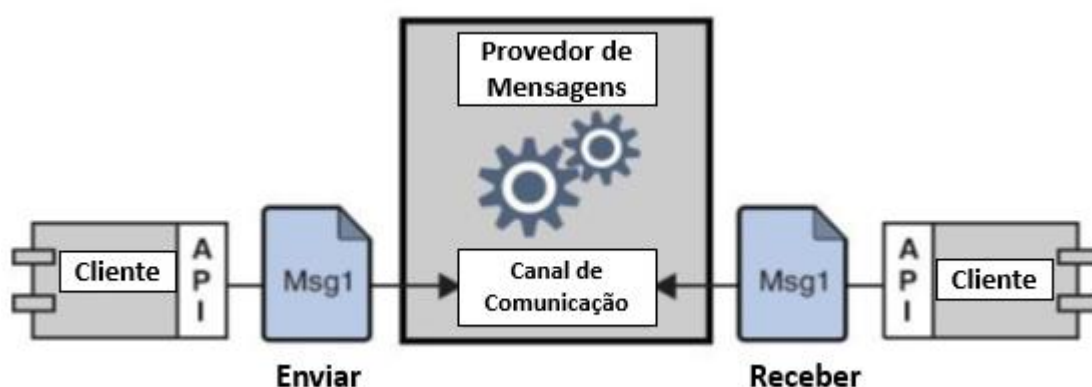
1.9.1 *Middleware Orientado a Mensagens*

As tecnologias de rede agora permitem um alto grau de conectividade em um grande número de computadores, aplicações e usuários. Nesses ambientes, é importante fornecer comunicações assíncronas para a classe de sistemas distribuídos que operam de maneira fracamente acoplada e autônoma e que exigem imunidade operacional contra falhas de rede.

Esse requisito foi preenchido por vários produtos de *Middleware* que são caracterizados como *Middleware Orientado a Mensagens* (MOM), enfileiramento de mensagens (do inglês *message queuing*) ou *publish/subscribe* (ORACLE9I, [s.d.]).

O *Middleware Orientado a Mensagens* (MOM) provê comunicação entre aplicações por meio da troca de mensagens. Na estrutura mostrada na Figura 6, as mensagens enviadas pelo *client* são organizadas em filas onde aguardam até serem encaminhadas para seus respectivos destinos.

Figura 6 - Arquitetura de *Middleware Orientado a Mensagens*



Fonte: Adaptada de SOUSA (2017)

Comumente, essas mensagens, contidas nas filas, são classificadas em uma ordem específica. Neste sentido, elas seguem o padrão *First-In, First-Out* (FIFO), onde a primeira mensagem da fila é também a primeira mensagem a ser encaminhada ao destino final (SOUSA, 2017).

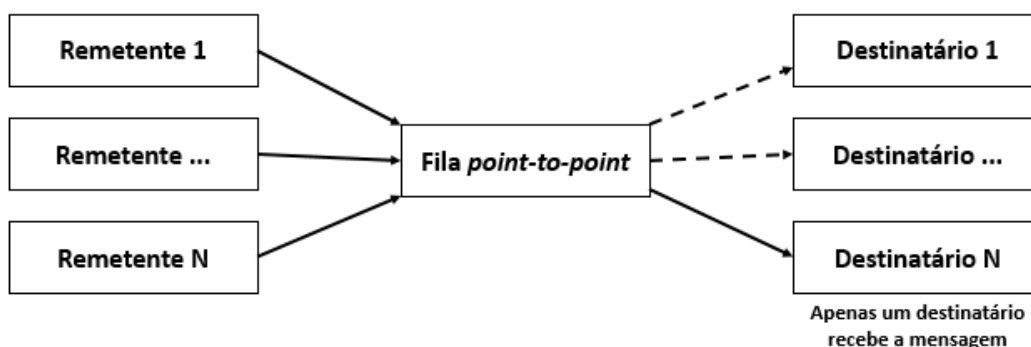
De acordo com SILVEIRA et al. (2011), a comunicação realizada pelo MOM, em um ambiente distribuído, pode ocorrer de modo síncrono ou assíncrono. No método síncrono, os elementos, permanecem bloqueados até finalizarem o processo de comunicação. Por outro lado, na comunicação assíncrona, o elemento requerente fica disponível logo após realizar uma solicitação, não sendo necessário aguardar um retorno dessa requisição (SOUSA, 2017).

1.9.2 Modelos de Mensagens

O Middleware Orientado a Mensagens dispõe de dois modelos de mensagens, conforme segue abaixo:

- 1) **Point-to-point:** Neste modelo, as mensagens são enviadas através de uma fila para um componente específico do sistema, ou seja, a comunicação ocorre exclusivamente entre remetente e destinatário, como representado na Figura 7.

Figura 7 - Modelo de Mensagem Point-to-Point



Fonte: Adaptada de SOUSA (2017)

- 2) **Publish/subscriber:** o modelo *publish/subscriber* aplica o conceito de produtor e consumidor de mensagens, onde um único produtor pode enviar uma mensagem para um ou vários consumidores. Os produtores publicam mensagens em um tópico ou canal (ambos filas) e os consumidores se inscrevem em determinado tópico ou canal conforme interesse, conforme ilustra a Figura 8.

Figura 8 - Modelo de Mensagem *Publish/Subscriber*

Fonte: Adaptada de EUGSTER et al. (2003)

Publish/subscribe é um paradigma de interação que permite a liberdade no sistema distribuído pelo desacoplamento de entidades de comunicação em termos de tempo, espaço e sincronização (KRISHNAMACHARI; WRIGHT, 2017).

O paradigma de interação *publish/subscribe* fornece aos processos *subscribers* a capacidade de expressar seu interesse em um tópico ou um padrão de eventos, a fim de serem notificados de qualquer evento, posteriormente, gerado por um *subscriber*, que corresponda aos interesses registrados (EUGSTER et al., 2003).

Em outras palavras, os *publishers* publicam informações em um barramento de *software* (um gerenciador de tópicos, por exemplo) e os clientes assinam as informações que desejam receber desse barramento, conforme apresentado na Figura 8.

As aplicações que se comunicam por meio do modelo *publish/subscribe* exigem que as aplicações de envio (*publishers*) publiquem mensagens sem especificar explicitamente os destinatários ou ter conhecimento dos destinatários pretendidos. Da mesma forma, as aplicações de recebimento (*subscribers*) devem receber apenas as mensagens nas quais o assinante registrou interesse.

1.10 CONSIDERAÇÕES FINAIS

Nesta seção foram apresentados os conceitos de arquiteturas de Internet do Futuro, a abordagem *clean slate* e a abordagem pluralista de Internet do Futuro onde, supostamente haverá arquiteturas de rede distintas coexistindo. Foi apresentada a RINA, suas características, protocolos de comunicação e alguns protótipos desenvolvidos dessa arquitetura. Em seguida, foi realizada uma comparação de aspectos chave entre RINA e a Internet atual, onde foi possível observar as especificidade de cada arquitetura.

Posteriormente, foi abordado o conceito de *Middleware* no contexto geral, suas funcionalidades e a importância em ambientes distribuídos. Também foram citadas as principais categorias de *Middleware* conforme suas primitivas de comunicação e explanado o *Middleware* Orientado a Mensagens (MOM), categoria que corresponde à proposta de arquitetura desta dissertação. Também foi apresentado os modelos de mensagens (*Point-to-point* e *Publish/subscribe*) que o MOM dispõe.

3 O *MIDDLEWARE* POLIMORFI

Nesta seção será exposto o *Middleware* PolimorFI, que é um MOM construído para o contexto de arquiteturas de Internet do Futuro. Serão apresentados a visão geral da proposta, os requisitos, assim como a arquitetura. No final, serão apresentados os trabalhos relacionados com a proposta desta dissertação.

3.1 VISÃO GERAL DA PROPOSTA

O cenário contemporâneo referente às arquiteturas de Internet do Futuro, mostra que há diferentes propostas de arquiteturas de rede com protótipos em diferentes estágios de implementação, cada qual possuindo objetivos peculiares de projeto e paradigmas de comunicação distintos. Nesse sentido, dificultando a identificação de uma única arquitetura de rede como a resposta para todos os entraves enfrentados pelas aplicações emergentes, bem como atender todos os requisitos das novas aplicações para a Internet do Futuro.

Deste modo, uma solução para resolver este problema é um sistema de *Middleware* que suporte o acoplamento de instâncias de novas arquiteturas de rede e que permita o desenvolvimento de aplicações que façam uso dos recursos ofertados por essas novas arquiteturas de rede.

O *Middleware* proposto nesta dissertação tem como finalidade tornar menos complexa e mais produtiva a construção de aplicações distribuídas para a chamada Internet do futuro, fornecendo abstração de modo a tornar transparente os detalhes de programação em redes ao desenvolvedor da aplicação.

O nível de abstração refere-se a como o desenvolvedor da aplicação visualiza o sistema (por exemplo, nível de Nó / dispositivo individual, nível do sistema). Por outro lado, o paradigma de programação (por exemplo, *publish/subscribe*) lida com o modelo

para desenvolvimento ou programação de aplicações ou serviços (RAZZAQUE et al., 2016).

De acordo com LEON et al. (2017), uma infinidade de aplicações distribuídas inovadoras foi desenvolvida nos últimos anos, cada um com requisitos de comunicação específicos. Isso levou à evolução dos cenários de rede, sobretudo das infraestruturas de rede de transporte, que precisam progredir com os novos requisitos.

Nesse sentido, o intuito do *Middleware* aqui proposto é permitir a construção e interação de várias aplicações com pouco esforço de codificação – especificando apenas as particularidades de cada aplicação para melhor usufruir dos recursos ofertados pela arquitetura de rede acoplada.

Tal proposta de *Middleware* é inédita na literatura e, para sua construção, requer o estudo e domínio das áreas de sistemas distribuídos e arquiteturas de Internet do Futuro, assim como conhecimento da realidade aplicada destas áreas na indústria de *software* atual. Também são fundamentais conhecimentos na área de engenharia de *software* e padrões de Projetos.

Com o uso do PolimorFI, os desenvolvedores de aplicações podem construir aplicações e testar a comunicação de *hosts*, processos e conteúdos, desta maneira fomentando e otimizando as pesquisas em interoperação de aplicações em redes distintas na Internet do futuro.

3.1.1 Requisitos

Com a finalidade de prover simplicidade no processo de desenvolvimento e criação de aplicações e serviços para a Internet do Futuro, o PolimorFI visa oferecer recursos que tornem transparente, ao desenvolvedor de novas aplicações, detalhes de programação em redes.

Os requisitos do PolimorFI foram determinados tomando como base os estudos dissertados em SOUSA (2017), VALENTE (2011), (RAZZAQUE et al., 2016). Nessa perspectiva, o PolimorFI possui os seguintes requisitos:

- **Homogeneidade:** Todos os componentes serão desenvolvidos na mesma tecnologia, por motivo de praticidade.
-

- **Retrocompatibilidade:** Nessa etapa do PolimorFI, um dos seus requisitos é ser retrocompatível com o a Internet atual, ou seja, permite, também, que aplicações utilizem *socket* TCP / IP para realizar a comunicação entre processos.
- **Transporte de Mensagens:** Os processos irão transportar mensagens utilizando paradigmas específicos para a arquitetura de rede determinada.
- **Serialização de Mensagens:** Os objetos serão convertidos em *bytes* para serem transmitidos através de *socket*.
- **Heterogeneidade:** A heterogeneidade de rede é transparente ao usuário da aplicação. Ou seja, ao executar a aplicação, esta pode utilizar a arquitetura/protocolos RINA ou TCP/IP, sem que isto influencie os resultados das operações providas pelas propostas. A questão essencial da interoperabilidade em sistemas distribuídos está se tornando ainda mais presente na Internet do Futuro, onde aplicações complexas serão compostas de sistemas extremamente heterogêneos (GEORGANTAS et al., 2013).
- **Ferramental para o Desenvolvimento:** Assim como em qualquer plataforma que possibilite a criação de aplicações, a propiciação de ferramentas que auxiliem nesse processo de construção é algo extremamente relevante. Portanto, a disponibilização desse ferramental trata-se de um requisito desejável para *Middleware* para a Internet do futuro.

3.1.2 Arquitetura

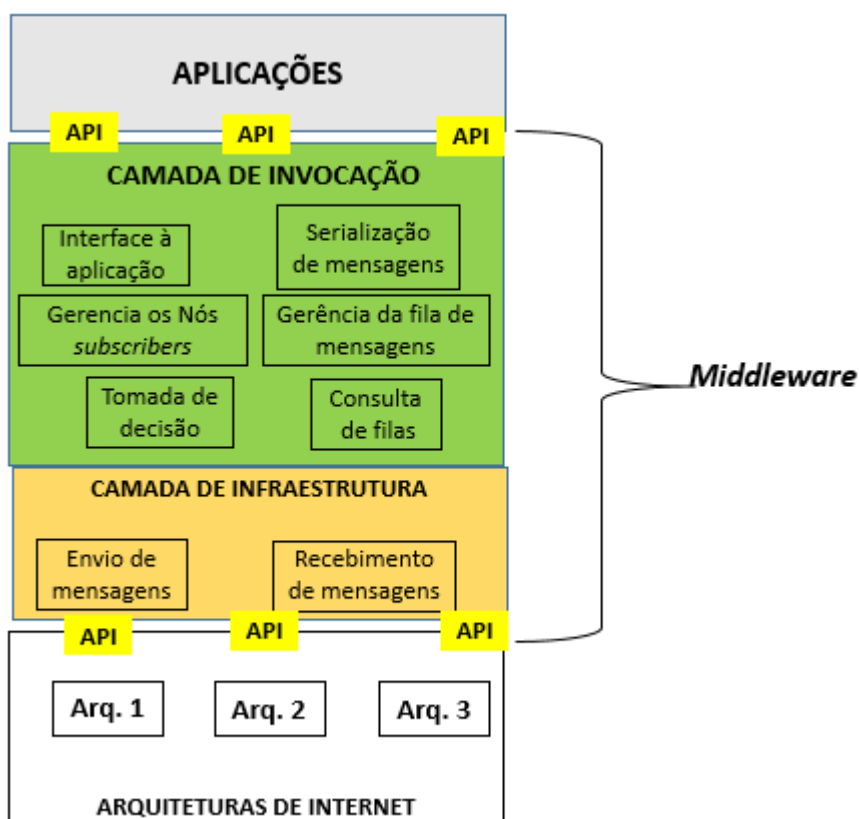
A arquitetura do PolimorFI segue alguns padrões de projetos, muito utilizados na engenharia de *software*. Esta prática provê um reuso de técnicas promissoras e bem documentadas, facilitando, portanto, a compreensão e a manutenção da plataforma construída.

O *Middleware* PolimorFI utiliza como base o padrão de projeto utilizado por SOUSA (2017), conhecido também como *Publish/subscribe*, haja vista que, em vários cenários, um ou múltiplos dispositivos estarão interessados em determinados tópicos de assuntos. Além disso, conforme disposto em RAZZAQUE et al. (2016), *Middlewares* Orientados a Mensagens, tipicamente, utilizam o padrão *Publish/subscribe*, pois essa abordagem de *design* aponta requisitos não funcionais, como confiabilidade, disponibilidade, desempenho em tempo real, escalabilidade e segurança.

A sua arquitetura é dividida em duas partes (vide Figura 9). A primeira, denominada de camada de infraestrutura, trata do gerenciamento das conexões de rede.

A segunda é chamada de camada de invocação, que tem a função de prover a transparência de acesso, transformando invocações escritas como locais em invocações remotas.

Figura 9 - Arquitetura do *Middleware* PolimorFI



Fonte: Elaborada pelo autor

3.1.2.1 Camada de Infraestrutura

Essa camada é responsável pela confiabilidade da transferência dos dados e controle de intercomunicação, ela provê a comunicação de rede entre processos. Dessa forma, todo o envio ou recebimento de mensagens (que são tratadas como sequências de bytes) são realizados por essa camada, usando um protocolo de rede pré-estabelecido.

Portanto, a camada provê, ao resto do *Middleware* e também à aplicação, transparência quanto à comunicação entre diferentes processos do sistema distribuído. Além do estabelecimento da conexão entre os processos que fazem os papéis de cliente e servidor, essa camada também é responsável por algumas configurações, como o

estabelecimento de *timeouts* e número de tentativas para se enviar informações, podendo fazer o tratamento de alguns erros antes de repassá-los à camada de invocação.

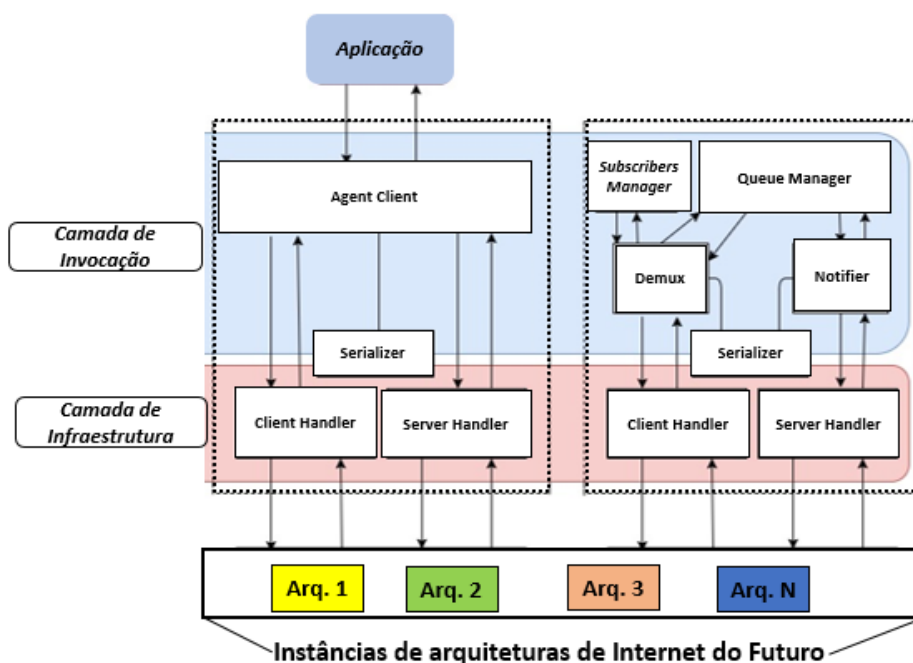
3.1.2.2 Camada de Invocação

A camada de invocação existe para garantir a transparência de acesso à interação das aplicações, onde se encontram os processos transmissor e o receptor. Dessa maneira, não será necessário que o desenvolvedor da aplicação se preocupe se o processo é remoto ou local.

3.1.2.3 Componentes

A Figura 10 retrata a arquitetura do PolimorFI com todos os componentes.

Figura 10 - Arquitetura do PolimorFI e todos os seus componentes



Fonte: Elaborada pelo autor

Em seguida, tem-se a explanação da atuação de cada componente que integra a arquitetura do PolimorFI:

- **Agent Client:** Este componente fornece uma interface à aplicação para fazer chamadas remotas ao servidor de mensageria.

- **Serializer:** Este componente é responsável por serializar e/ou deserializar as mensagens que serão trocadas. Isto é, transformar as mensagens no formato aceito pela arquitetura de rede. O componente *Client Agent* utiliza o *Serializer* para transformar o conteúdo em sequência de bytes, tipo de informação que trafega na rede. De certo modo, o *Serializer* pode atuar nas duas camadas do PolimorFI (infraestrutura e invocação), pois ele é independente de outro componente.
- **Client Handler:** este componente é responsável por gerenciar a comunicação pela rede dentro da aplicação do cliente. O *Client Handler* estabelece uma conexão por meio do protocolo de comunicação da arquitetura de rede utilizada, envia o pedido e espera pela resposta, trazida pelo *Server Handler*.
- **Server Handler:** O *Server Handler* recebe mensagens e as encaminha para a aplicação.
- **Queue Manager:** Faz a gerência da fila de mensagens.
- **Subscribers Manager:** Realiza a gerência dos nós.
- **Demux:** Esse componente recebe as mensagens e decide quais ações tomar.
- **Notifier:** O Notifier consulta as filas e envia as mensagens para os destinatários.

3.1.2.4 Aspectos de Polimorfismo

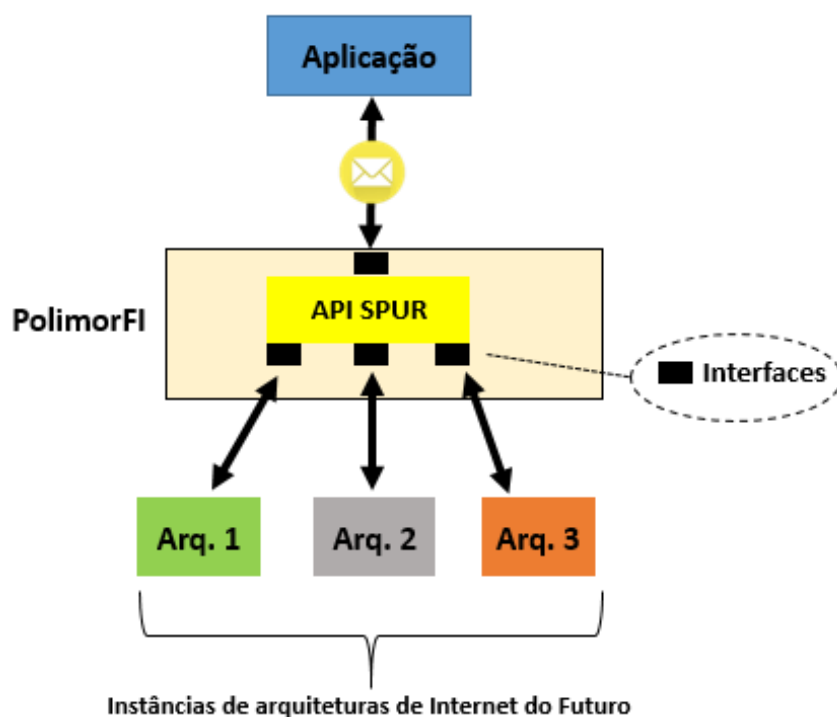
GRZECH et al. (2012) mencionam que redes orientadas a conteúdo e conteúdo orientado a aplicações suportam processos orientados a processos e serviços. Por conseguinte, os autores complementam que uma combinação desses modelos é possível e atrativa nas perspectivas de pesquisa, aplicação e negócios.

O *Middleware* proposto, ou simplesmente PolimorFI, segue o conceito polimórfico, abordado como pluralista por FERNANDES et al. (2011), e parte da premissa que a sua arquitetura poderá suportar várias pilhas de protocolos específicos de arquiteturas de

Internet do Futuro, que abordam paradigmas de rede distintos, de modo que as aplicações com características específicas para esses paradigmas sejam desenvolvidas e testadas utilizando a sua arquitetura.

O PolimorFI deve possibilitar que novas aplicações sejam criadas utilizando uma *Application Programming Interface* (API) padrão, denominada de SPUR⁵ – na perspectiva de múltiplos paradigmas e arquiteturas de rede e o próprio PolimorFI se encarrega de fazer todas as adaptações necessárias para o uso em qualquer arquitetura de Internet do Futuro. A Figura 11 expõe como uma aplicação distribuída pode acessar diferentes arquiteturas de rede por meio do PolimorFI fazendo uso da API SPUR e suas interfaces.

Figura 11 - Aplicação acessando diferentes arquiteturas de rede por meio do PolimorFI



Fonte: elaborada pelo autor

3.1.2.5 A API SPUR

Como toda especificação, a API SPUR não é um “produto” em si, mas sim um conjunto de interfaces que devem ser implementadas para toda instância de arquitetura de Internet do Futuro que se propõe a ser adaptada ao PolimorFI. Ela contém dois tipos de interfaces: a) interface de aplicação – que são utilizadas pelas aplicações para

⁵ SPUR trata-se de uma derivação do termo “*PIURalistic*”, haja vista a perspectiva de suportar múltiplos paradigmas e arquiteturas de rede.

solicitar serviços do *Middleware*; e b) interface de conexão, que são utilizadas pelas instâncias de arquiteturas de Internet do Futuro para que o PolimorFI utilize os protocolos específicos de cada arquitetura para fazer o transporte de mensagens para os processos/hosts.

O propósito dessa API é permitir que uma aplicação acesse diretamente protocolos específicos das arquiteturas de Internet do Futuro que serão acopladas ao *Middleware* PolimorFI, o que significa que esta API receberá mensagens particulares de cada paradigma de rede e possibilitará fluxo de comunicação entre transmissores e receptores através da rede.

Os protocolos de cada arquitetura de rede seriam adicionados a um “catálogo” de protocolos suportados por cada arquitetura de Internet do Futuro. Isso permitirá que várias aplicações abram *sockets* para seus protocolos específicos e o PolimorFI possa controlar qual *socket* transmite mensagens / pacotes específicos.

Para isso, o PolimorFI deverá possuir um mecanismo para reconhecer os datagramas de interesse da aplicação - podendo ignorar todos os outros – e encaminhar para o *socket* que possui os protocolos pertencentes à arquitetura de Internet do Futuro que será utilizada pela aplicação distribuída.

Num primeiro momento, os esforços de implementação foram concentrados numa prova de conceito para acoplar uma instância da RINA. A API adotada para o experimento compõe a própria instância da RINA utilizada. Nesse caso, o PolimorFI fará parte de uma DAF e a API possibilitará a comunicação entre uma DAF e uma DIF.

A seção 3.2 a seguir apresenta os trabalhos relacionados com esta dissertação e, subsequente, a seção 4 detalha a prova de conceito e discorre detalhes do experimento.

3.2 TRABALHOS RELACIONADOS

A Internet do Futuro é concebida como um ambiente mundial que conecta uma vasta quantidade de recursos heterogêneos e autônomos, quais sejam, “coisas”, serviços e conteúdos, que interagem entre si em qualquer lugar e a qualquer momento (CAPORUSCIO; GHEZZI, 2015). Nesse universo, várias pesquisas têm sido impulsionadas.

Por conseguinte, esta dissertação relaciona diversos trabalhos que visam progredir com o tema Internet do Futuro, tendo em vista atender a demanda emergente e futura das aplicações conectadas à rede. Assim, foi realizada uma análise sistemática de estudos que foram desenvolvidos durante a última década, apreciando assuntos de relevância e correlação direta ao tema proposto.

Tais pesquisas foram divididas em dois subconjuntos: **(I) Ambientes para desenvolvimento de aplicações na Internet atual** – que compreende trabalhos que almejam atender as demandas das aplicações emergentes, assim como fomentar o surgimento de novas aplicações, com soluções de *Middleware* voltadas para novos paradigmas de comunicação e IoT, todavia utilizando a arquitetura da Internet atual; e **(II) Aplicações em novas arquiteturas de rede** – que engloba pesquisas com características plenamente revolucionárias, as quais acreditam que apenas substituindo a arquitetura da Internet atual irá atender satisfatoriamente as aplicações conectadas à rede. Os trabalhos estão descritos abaixo:

➤ **Ambientes para desenvolvimento de aplicações na Internet atual:**

SANTHI (2010) propõe uma nova arquitetura de comunicação (*framework*), com aspectos orientados a serviços, que permite a integração de características de várias dimensões, em particular segurança e mobilidade, e explica que problemas com os modelos atuais de sistemas distribuídos impedem que eles sejam utilizáveis para uma arquitetura de rede de última geração. O autor afirma que a falta de interoperabilidade entre vários *middlewares* existentes causam ilhas de implementações incompatíveis. Assim, desenvolve a visão de uma abordagem *top down*, do ponto de vista dos desenvolvedores de aplicações. De tal perspectiva, a rede não é apenas conectividade especificada por uma tupla de pontos finais. Por fim, o autor traz a afirmação que uma rede é uma coleção de serviços distribuídos que estão disponíveis para as aplicações.

SOUSA (2017) apresenta o *xPresumo*, uma arquitetura de *Middleware* Orientado a Mensagens (MOM) para o contexto de Internet das Coisas (IoT). O projeto foi desenvolvido utilizando a API JMS, tendo como primitiva de comunicação o modelo *Publish-Subscribe*. O autor relata que o fato de optar pela primitiva *Publish-Subscribe* está relacionado à realidade de IoT, onde os eventos são gerados em grande quantidade. Desse modo, a interação entre coisas inteligentes torna-se mais flexível e otimizada.

(ISSARNY et al., 2011) discutem a respeito do *design* de *middleware* orientado a serviços, portanto, investigando a descrição, a descoberta, o acesso e a composição de serviços na Internet do futuro. Para os autores, a abordagem evolucionária da Internet atual desafia o paradigma. Eles complementam que a Internet do futuro pode ser definida como a união e cooperação da Internet de Conteúdo, Internet de Serviços e Internet das Coisas, apoiada por uma base de infraestrutura de rede em expansão.

PEREZ; LABRADOR; BARBEAU (2010) apresentam o G-Sense, um *Framework* híbrido que usa uma arquitetura cliente-servidor em um domínio de servidor e uma arquitetura ponto-a-ponto entre os servidores para escalar o sistema para implantações em todo o mundo. Os autores descrevem o *framework* de arquitetura e o desenvolvimento do protótipo do G-Sense, que demonstra que a proposta facilita o fornecimento de aplicações emergentes da Internet, como detecção participativa, serviços baseados em localização e detecção centrada no ser humano. Enfim, mostram a importância dessas novas aplicações e a nova onda de tráfego massivo que elas irão gerar.

TERZIYAN; ZHOVTOBRYUKH; KATASONOV, (2009) propuseram o *Middleware* PROFI como uma solução para interconectar os futuros elementos da rede de Internet usando linguagens semânticas. O PROFI se baseia no paradigma de redes de informação, para conectar abordagens multidisciplinares unificando o tratamento de todos os tipos de atores de comunicação (dispositivos, nós da rede, objetos de informação, aplicações e serviços, etc.), usando a arquitetura declarativa de sobreposição de rede para componentes de rede programáveis, fornecendo flexibilidade, abertura e capacidade de gerenciamento.

ZHOU et al. (2016) discorre sobre um *Framework* de aplicação para redes de sensores sem fio e atores, denominada de WSAN, utilizando uma abordagem SDN e os métodos relevantes para a aplicação da abordagem SDN, com o objetivo de melhorar a eficiência e a escalabilidade da rede. Os detalhes do *Framework* incluem uma estrutura de três camadas, as entidades do sistema relevantes, a pilha de protocolos aprimorada, e os tipos de mensagem programáveis para comunicação cooperativa e execução de tarefas entre nós WSAN.

➤ **Aplicações em novas arquiteturas de rede:**

Como a Internet do Futuro está sendo concebida para melhor suportar novas aplicações e serviços, é natural que os requisitos sejam provenientes de futuras aplicações, nesse sentido, CASTA; JAIN; HAUSI (2013) trazem uma abordagem intrigante sobre que tipo de aplicações pode ser esperado na Internet do Futuro. Os autores analisam algumas possíveis futuras aplicações tentando entender os principais recursos que a Internet do Futuro deve oferecer suporte para atender aos principais requisitos desafiadores. Eles argumentam que uma questão relacionada à implantação de novos serviços e aplicações é muitas vezes considerada impraticável. Isso se deve ao alto custo, em termos de dinheiro e tempo, associado à instalação de novo hardware. Por fim, eles comparam os requisitos com a pesquisa existente sobre a Internet do Futuro, para ver como eles se combinam, e concluem que, de fato, a evolução do *hardware* e o *design* desempenham um papel muito importante na sua criação.

Para CAPORUSCIO; GHEZZI (2015), as aplicações possivelmente surgirão dinamicamente como agregação oportunista de recursos disponíveis em um determinado momento e poderão se autoadaptar de acordo com a dinâmica do ambiente. Nesse contexto, os engenheiros devem receber abstrações de modelagem e programação adequadas para desenvolver aplicações capazes de se beneficiar da Internet do Futuro, sendo ao mesmo tempo fluída e confiável.

MARTIN; WIPPEL (2013) apresentam o NEMA (*Netlet-based Node Architecture*), um *Framework* de tempo de execução que permite que os nós se conectem simultaneamente a arquiteturas diferentes de Internet do Futuro. Essas redes podem ser baseadas em diferentes famílias de protocolos, usando protocolos de rede especializados. As aplicações acessam esses protocolos por meio de uma API que abstrai os detalhes da rede: em vez de fornecer endereços e protocolos de rede, as aplicações especificam nomes exclusivos globais como URIs e requisitos no serviço de comunicação solicitado para iniciar a comunicação com o conteúdo, serviço ou host associado ao nome.

Os autores avaliam os conceitos e interfaces da estrutura para determinar o conjunto mínimo de invariantes necessárias para operar famílias de protocolos que diferem em suas abstrações e paradigmas básicos.

LI (2018) aponta desafios básicos e requisitos da Rede para o ano de 2030, onde aborda novos conceitos de comunicações do futuro citando como necessidade um novo protocolo de rede – *New IP* – facilitando novos serviços de comunicação da Internet.

Discute novos paradigmas de comunicação para aplicações emergentes e ressalta grande impacto na área das telecomunicações referente aos desafios prescritos pelas aplicações holográficas.

Como proposta de arquitetura de Internet do Futuro para suportar novas aplicações e serviços, várias pesquisas mencionaram a RINA como solução.

GRASA et al. (2017) explicam que a renomeação de rede no mundo IP é um procedimento complicado e caro que deve ser cuidadosamente planejado e executado para evitar problemas de integridade de conexão de transporte. A origem da maioria desses problemas, segundo os autores, está na falta de uma arquitetura completa de nomes e endereçamento no conjunto de protocolos TCP/IP. Em seguida, os autores analisam como esses problemas afetam a renomeação em redes baseadas no RINA. A análise teórica apoiada pelos resultados da experimentação indica que a renomeação nas redes do RINA não é apenas contínua (pode ser feita sem afetar os fluxos existentes), mas também não requer mecanismos especiais.

Uma abordagem interessante é apresentada por LEON GAIXAS et al. (2017), onde os autores estudam os benefícios que a substituição do TCP/IP pela RINA pode trazer para DCNs comerciais, com foco no encaminhamento e escalabilidade de roteamento. Eles avaliam quantitativamente os benefícios que as soluções RINA podem gerar em relação às baseadas em TCP/IP e destacam como, implantando a RINA, as soluções de um roteamento topológico podem melhorar ainda mais a eficiência da rede. Propõem, ainda, uma política de encaminhamento de regra e exceção adaptada às características de várias variantes DCN, permitindo decisões de encaminhamento rápido com informações meramente dos vizinhos.

(SMALL, 2012) apresenta aspectos de segurança da RINA em comparação à estrutura TCP/IP. O autor explana que a RINA tem uma pilha de protocolo mais curta do que a arquitetura atual da Internet e dependem, em vez disso, da separação de mecanismo de política e implantação recursiva para alcançar redes de grande escala. Devido a essa menor pilha de protocolos, menos mecanismos de rede, segurança ou outros, devem ser necessários para proteger as redes RINA.

Perante o exposto, esta dissertação mescla os objetivos dos trabalhos citados acima, de ambos os subconjuntos, dado que pretende fomentar e facilitar a construção de novas aplicações por meio de um *Middleware* polimórfico, o qual possibilitará que as

aplicações façam uso de recursos oferecidos por novas arquiteturas de Internet e pela Internet atual, com paradigmas de comunicação específicos de cada proposta de rede. Diferentemente da proposta de MARTIN; WIPPEL (2013), o PolimorFI propõe um conjunto de interfaces (API SPUR) para aturem, de forma específica, com as aplicações e com as instâncias de arquiteturas de Internet do futuro.

No entanto, em princípio, o *Middleware* apresentado irá possibilitar experimentos com um protótipo da arquitetura RINA, proposta de rede que vem sendo desenvolvida com a finalidade de resolver as limitações atuais da pilha TCP/IP.

4 PROVA DE CONCEITO

Esta seção expõe a prova de conceito do protótipo do *Middleware* PolirmorFI, iniciando com a exposição do cenário de uso, em seguida são demonstrados o cenário de experimento, a arquitetura do PolimorFI, os diagramas de sequência, assim como os detalhes da implementação. Por fim, será realizada a discussão dos resultados obtidos e relatadas as conclusões do laboratório experimental.

Para compor o protótipo inicial, foi selecionada a proposta de arquitetura de Internet do futuro RINA, que utiliza o paradigma de comunicação IPC. A escolha deu-se pelo fato de que RINA é uma proposta simples, contudo uma das mais inovadoras do mundo (TEYMOORI et al., 2016).

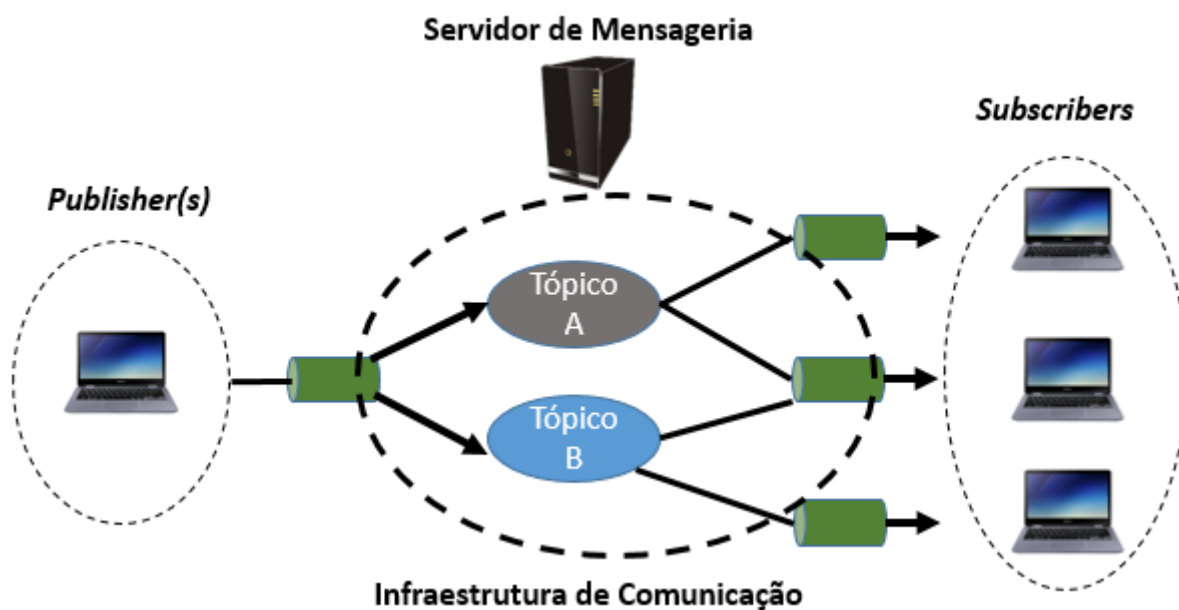
Para implementar o arquétipo inicial do PolimorFI serão utilizados padrões de projetos apresentados por VÖLTER; KIRCHER; ZDUN (2007) pois estes padrões permitem modularizar o sistema em componentes que tratam de aspectos comuns à maioria dos sistemas de *Middleware*.

As seções seguintes irão pormenorizar os detalhes da prova de conceito.

4.1 CENÁRIO DE USO

Um cenário que pode representar o uso do *Middleware* proposto nesta dissertação, pode ser exemplificado através de uma aplicação onde exista um ou vários nós remotos produzindo informações (como por exemplo, novas versões de aplicações), e outros nós remotos que estão interessados nessas informações para executar determinadas ações (a exemplo, realizar a atualização das aplicações). Conforme representa a Figura 12.

Figura 12 - Cenário de uso: Produção e consumo de informações no modelo *publish/subscribe*

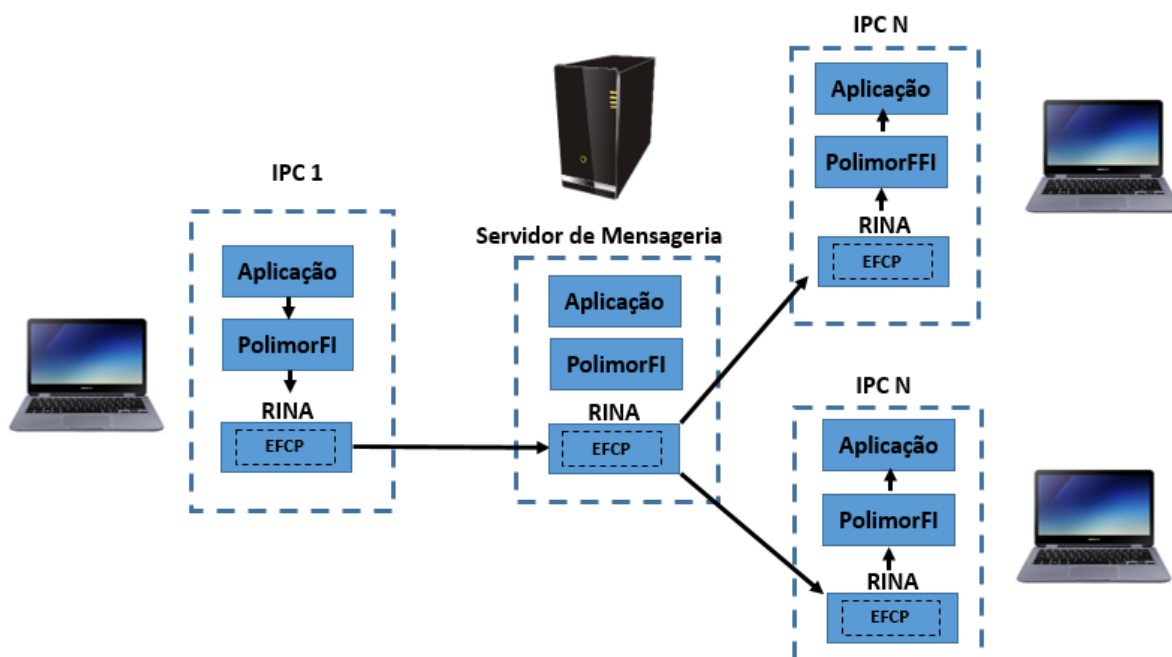


Fonte: elaborada pelo autor

A Figura 12 apresenta também a infraestrutura de comunicação onde ocorre o transporte de mensagens, a gerência das assinaturas e fila de tópicos. De um lado temos o(s) “*publisher(s)*” (publicadores) e do outro lado temos os “*subscribers*” (assinantes). Os assinantes (*subscribers*) fazem assinaturas (*subscriptions*) de tópicos desejados.

A Figura 13 apresenta o cenário que pode representar o uso do PolimorFI utilizado a arquitetura de rede RINA com o seu protocolo EFCP para transferência de dados.

Figura 13 - Cenário de Uso com RINA



Fonte: elaborada pelo autor

4.2 CENÁRIO DE EXPERIMENTO

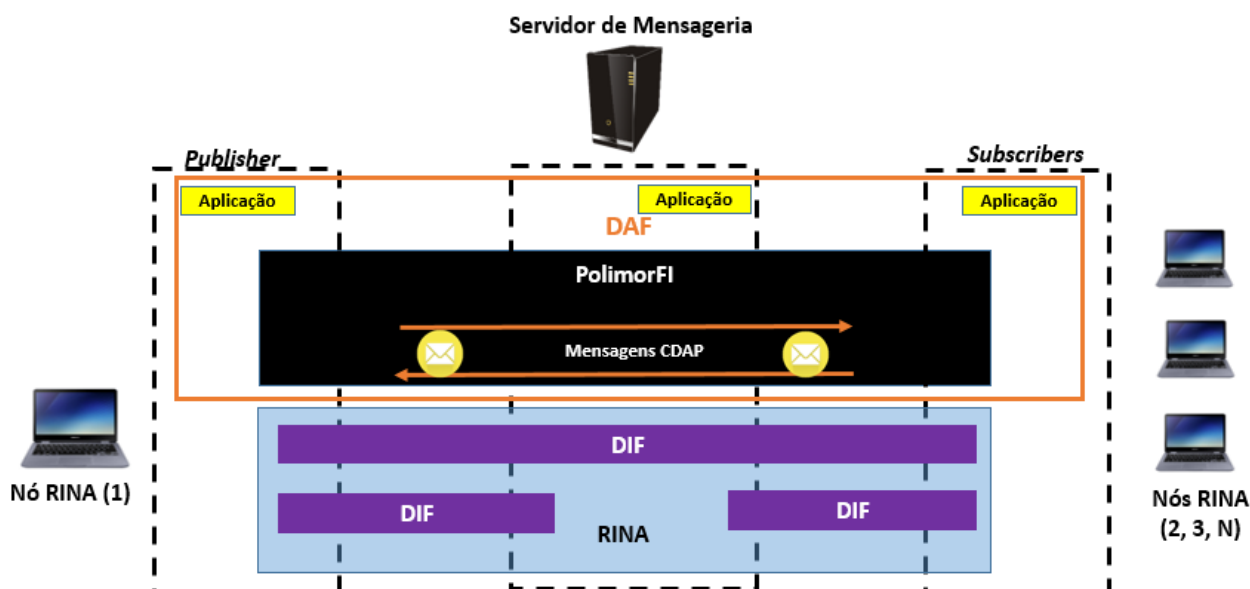
De acordo com a *OPUS SOFTWARE*, as aplicações mais populares são aquelas voltadas para troca de mensagens e comunicação, sendo assim, o PolimorFI enfoca, nesse primeiro momento, em possibilitar que as aplicações a serem desenvolvidas explorem os recursos e benefícios que a RINA oferece em termos de comunicação por seus protocolos.

A prova de conceito utilizará o ProtoRINA como instância da RINA no PolimorFI como uma abstração para uma aplicação distribuída. Para realizar o experimento, foi necessário integrar os pacotes de *software* do ProtoRINA para um funcionamento correto.

Os testes dos componentes do PolimorFI foram realizados de forma gradual, sendo implementada primeiramente a camada de infraestrutura e, em seguida, a camada de invocação do PolimorFI. Nesse sentido, foi construída uma aplicação – seguindo as considerações preconizadas em DAY et al. [s.d.] – cuja operação dar-se-á pela troca de mensagens CDAP entre processos RINA, também representados por *Publisher* e *Subscribers*, de modo que o primeiro faz o papel de servidor (que publica as mensagens)

e o outros fazem o papel de clientes (que assinam as mensagens), conforme mostrado na Figura 14.

Figura 14 - Cenário de experimento



Fonte: Elaborada pelo autor

O CDAP permite que aplicações distribuídas lidem com comunicações em um nível de objeto, em vez de forçar as aplicações a lidarem explicitamente com as operações de serialização e entrada/saída.

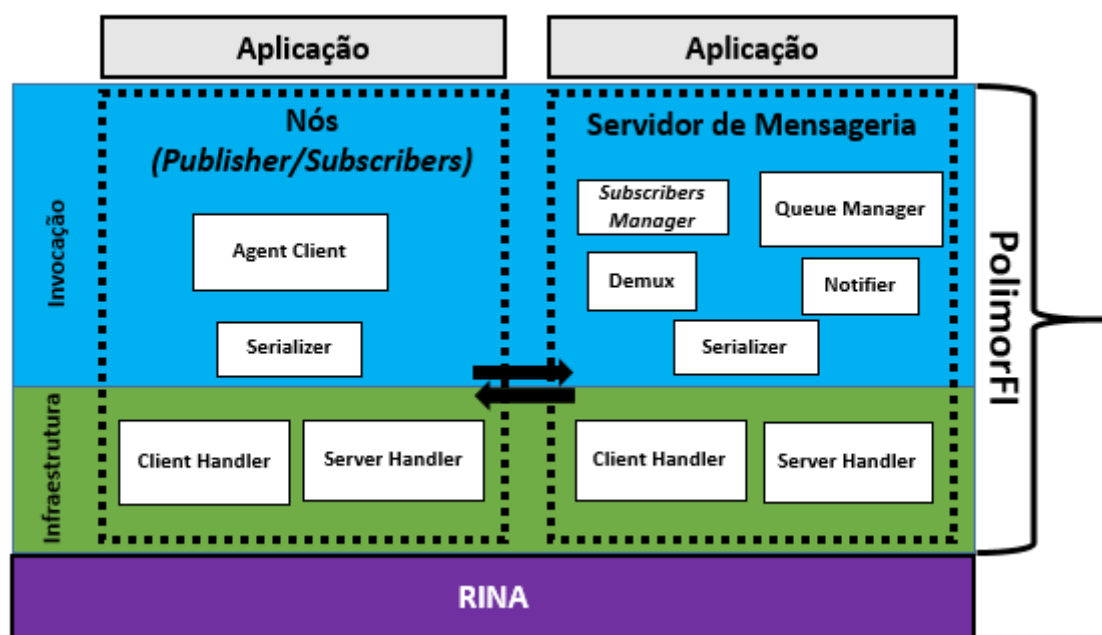
Neste cenário, as mensagens são publicadas pelo Nó transmissor e vão sendo transferidas passando pelos componentes do *Middleware* até chegar nos nós receptores. Nesta implementação, um Nó RINA representa um *host* no qual residem processos de aplicações e processos de IPC. O mecanismo de comunicação utilizado para estabelecer conexão entre os nós foi o *TCP shim layer package*. Ele é responsável por emular a conectividade física usando conexões TCP para a DIF.

Uma vez validada a camada de infraestrutura, que consistiu em testes de comunicação (estabelecimento de conexão via *TCP shim layer* e envio de mensagens por meio dos componentes *Handlers*), os demais componentes do PolimorFI foram implementados, possibilitando, portanto, um teste geral de toda a sua arquitetura.

4.3 ARQUITETURA DO POLIMORFI

A Figura 15 apresenta a localização de cada componente nas camadas do *Middleware* PolimorFI. Nela é possível observar as camadas de invocação e infraestrutura, bem como a instância da arquitetura de Internet do Futuro utilizada para a prova de conceito, a RINA.

Figura 15 - Arquitetura do PolimorFI com a instância da RINA.



Fonte: Elaborada pelo autor

Como mencionado anteriormente, para a parte de invocação, etapa em que o PolimorFI se comunica diretamente com a aplicação, é garantida a transparência de acesso, tornando invocação local semelhante à invocação remota. Para a camada de infraestrutura, serão gerenciadas as conexões de rede.

Neste procedimento, inicialmente, faz-se necessário executar o processo de aplicação *Publisher*, em seguida iniciar o *Subscribe*, deste modo, ambos estarão aptos a interoperar e a enviar e receber mensagens.

Cada componente do PolimorFI tem uma responsabilidade intrínseca, sendo assim, cada ação é discriminada conforme abaixo:

- i. *Agent Client*: Fornecer uma interface à aplicação para fazer chamadas remotas ao Servidor de Mensageria;

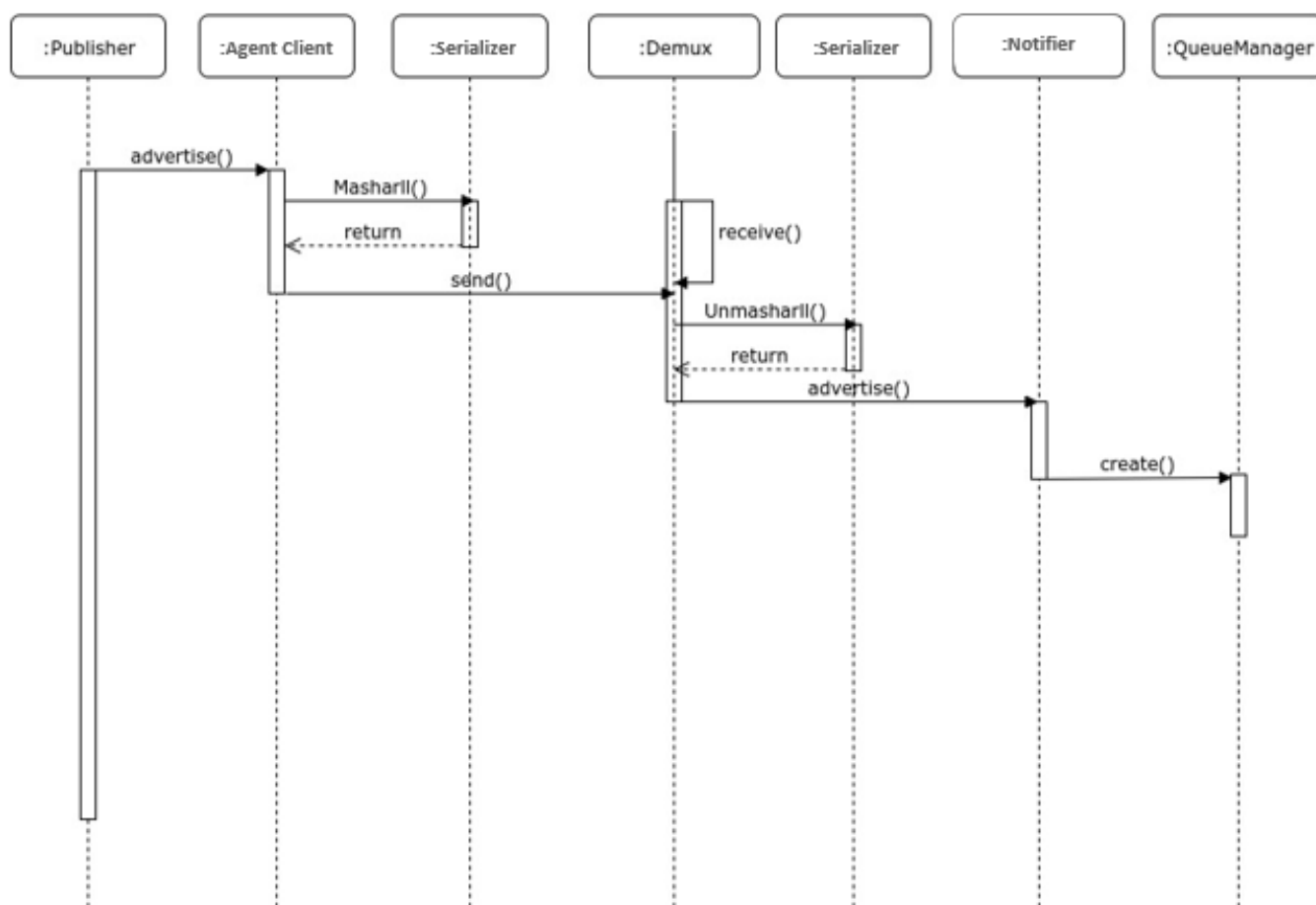
- ii. *Serializer*: Serializar e deserializar as mensagens;
- iii. *Client Handler*: Gerenciar conexões de rede e enviar e receber mensagens;
- iv. *Server Handler*: Receber mensagens e encaminhar as mensagens para a aplicação;
- v. *Queue Manager*: Gerenciar filas de mensagens;
- vi. *Subscribers Manager*: Gerenciar os nós subscribers;
- vii. *Demux*: Receber as mensagens e decidir quais ações tomar;
- viii. *Notifier*: Consultar as filas e enviar as mensagens para os destinatários;
- ix. *Client Handler*: Gerenciar conexões de rede e enviar e receber mensagens;
- x. *Server Handler*: Receber mensagens e encaminhar as mensagens para a aplicação.

4.4 DIAGRAMAS DE SEQUÊNCIA

O *Middleware* PolimorFI provê uma infraestrutura para que processos de aplicações se comuniquem e troquem informações. Isto posto, por via do conjunto de diagramas de sequência (Figura 16, Figura 17 e Figura 18) é possível compreender a interoperação dos componentes, tanto do lado do processo servidor (que faz o *Publish*) como do lado do processo cliente (que faz *Subscribe*), desde o anúncio do tópico, publicação e assinatura.

Uma vez gerada a mensagem, é enviada uma solicitação para publicar a mensagem no tópico. Na Figura 16 pode-se observar o anúncio do tópico pelo processo servidor (*Publisher*) e a interação dos componentes: Agent client; serializer; Demux; Notifier e Queue Manager. Após o anúncio, a mensagem é serializada, logo em seguida, o Agent Client a envia para o Demux que recebe a mensagem, envia para deserialização e a remete ao Notifier. Depois desse processo, a mensagem é enviada para o nó remoto.

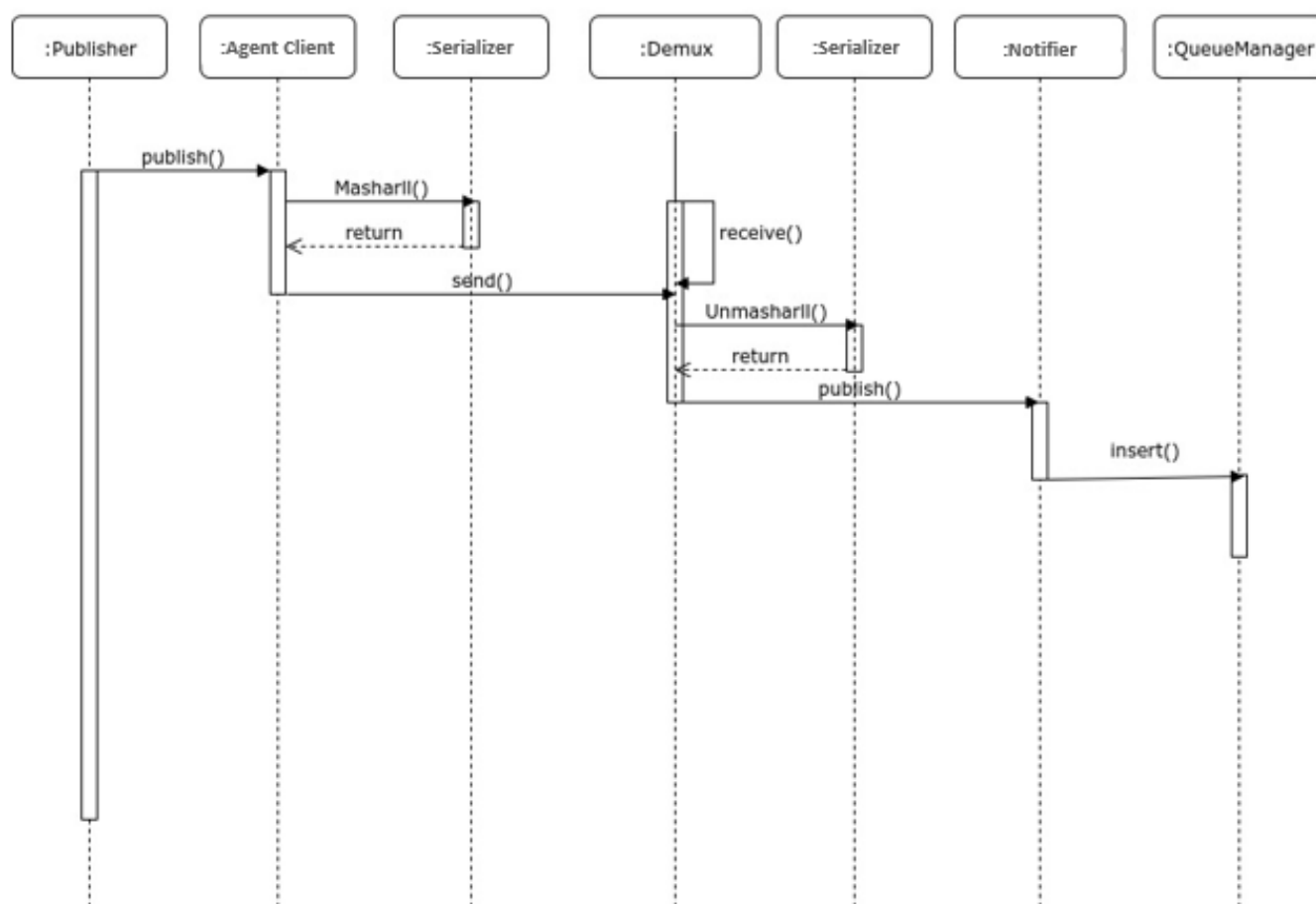
Figura 16 - Diagrama de sequência - anunciar tópico



Fonte: elaborada pelo autor

Na Figura 17, é possível reparar o processo de publicação da mensagem e todo seu trajeto pelos componentes, sendo ela serializada e deserializada, até chegar no Queue Manager, componente que faz a gerência da fila de mensagens. Para receber a mensagem publicada no tópico, o *Subscriber* realiza uma espécie de inscrição no tópico, o que faz com que ocorra a conexão entre eles. Assim, um tópico pode ter várias inscrições, contudo cada inscrição pertence a somente um tópico.

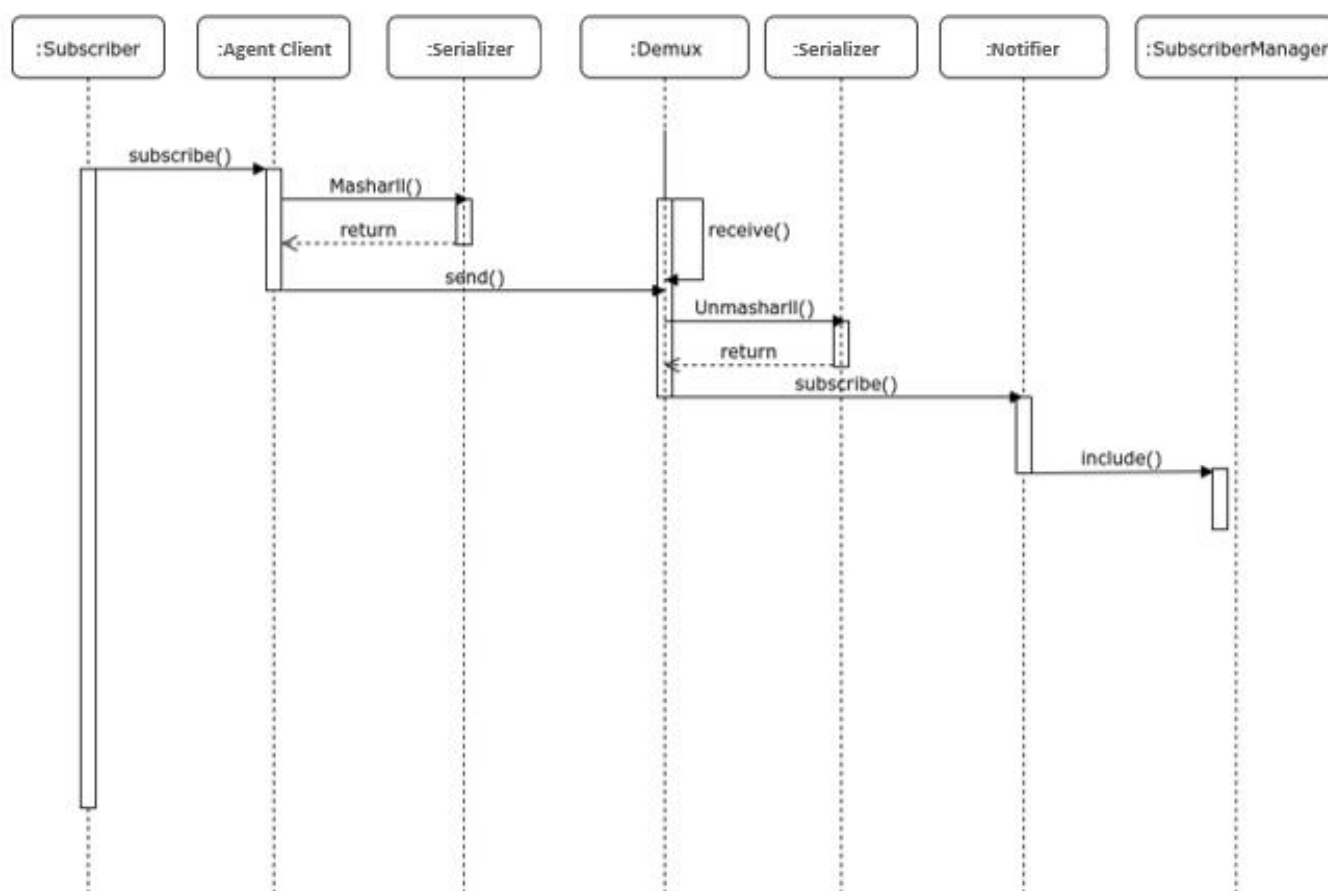
Figura 17 - Diagrama de sequência - publicar mensagem (*Publish*)



Fonte: elaborada pelo autor

Por fim, na Figura 18, percebe-se o processo de assinatura de tópico pelo cliente (*Subscriber*), o qual é gerenciado pelo componente Subscribers Manager. Nesse fluxo, o nó assina o tópico e recebe mensagens sobre ele. O Agent Client fornece a interface com o nó assinante do tópico, a mensagem passa pelo processo de serialização, e é enviada ao Demux que faz o recebimento. No caso de novos nós assinarem no mesmo tópico não receberão a mensagem (publicada anteriormente), a menos que seja uma publicação retida no PolimorFI.

Vale ressaltar que a adição de um novo *Publisher* ou de um *Subscriber* não afeta a funcionalidade do sistema, característica que é desejável ao lidar com as mudanças frequentes do ambiente móvel, por exemplo.

Figura 18 - Diagrama de sequência - assinar tópico (*Subscribe*)

Fonte: elaborada pelo autor

4.5 IMPLEMENTAÇÃO

Esta seção expõe os detalhes da implementação da prova de conceito, tais como a instância da RINA escolhida para realização do experimento, bem como os pacotes de *software* basilares para utilizar os módulos dos componentes desta arquitetura no PolimorFI, e a tecnologia utilizada para implementação do PolimorFI.

4.5.1 Instância da RINA

Nesta prova de conceito, será utilizada como instância RINA elementos do ProtoRINA na versão 2.0⁶. ProtoRINA é um protótipo da RINA que é utilizado, também, como ferramenta de ensino por educadores em disciplinas de redes e sistemas distribuídos, onde visa permitir o *design* de novos protocolos e aplicações (DAY et al., [s.d.]). Ele serve não apenas como um protótipo que demonstra as vantagens do RINA,

⁶ <http://csr.bu.edu/rina/protorina/2.0/>

mas também como uma ferramenta experimental de rede que permite aos usuários programar diferentes políticas usando seus mecanismos internos (WANG et al., 2014).

O ProtoRINA 2.0 permite o gerenciamento de rede baseado numa abordagem de Redes de Transporte Virtual (VTNs) que oferece serviço de comunicação com apoio explícito de Qualidade de Serviço (QoS) para aplicações através de fluxos de transporte, isto é, que envolve todos os mecanismos (por exemplo, endereçamento, roteamento, controle de erros e fluxo, alocação de recursos) necessários para atender aos requisitos de aplicações (WANG; MATTA, 2017).

Assim sendo, para realizar o experimento de executar uma aplicação distribuída no PolimorFI de modo que esta utilize como arquitetura de rede a RINA, foi necessário acoplar pacotes de *software* do ProtoRINA. A Tabela 3 **Erro! Fonte de referência não encontrada.** apresenta os pacotes de *software* do ProtoRINA que são fundamentais para utilizar os módulos dos componentes da arquitetura RINA no PolimorFI para este experimento. Os pacotes completos bem como os procedimentos de instalação do protótipo podem ser examinados no endereço <https://github.com/ProtoRINA/users/wiki>.

Tabela 3 - Pacotes de *Software* do ProtoRINA acoplados ao PolimorFI

PACOTES DE <i>SOFTWARE</i>	PROPÓSITO
<i>Message Package</i>	Este pacote suporta a implementação do formato de mensagens do Protocolo EFCP, ou seja, DTP, DTCP e CDAP.
<i>RINA Node package</i>	Este pacote possibilita a implementação do processo do nó RINA, no qual residem os processos de aplicações e os processos de IPC.
<i>TCP Shim Layer</i>	Este pacote é responsável por “emular” a conectividade física usando conexões TCP para as DIFs de nível 0. Cada processo IPC DIF-0 escuta uma porta TCP conhecida e dois tipos de fluxos (Gerenciamento e Transferência de Dados) entre cada par de processos IPC são mapeados em uma única conexão TCP.
<i>CDAP Message Generator</i>	Este pacote é definido em <code>rina.message/CDAPMessageGenerator.java</code> para gerar Mensagens CDAP que são serializadas, pelo componente <i>Serializer</i> , antes de serem publicadas.
<i>RINA object package</i>	Este pacote contém todos os objetos usados na arquitetura RINA.

Fonte: Elaborada pelo autor

Nesta prova de conceito, os empenhos têm sido efetivados no sentido de criar um ambiente de “emulação” abrigando componentes do ProtoRINA no PolimorFI para que as aplicações a serem desenvolvidas utilizem os recursos dessa arquitetura de rede, inicialmente. Deste modo, o PolimorFI será um DAF implementado em cima de um DIF do ProtoRINA.

A ideia é que, após validar experimentos tomando como ponto inicial a RINA, instâncias de outras propostas de arquiteturas de Internet do Futuro sejam incorporadas ao PolimorFI.

4.5.2 Implementação do PolimorFI

O arquétipo do PolimorFI foi desenvolvido utilizando a linguagem de programação Java, devido à sua particularidade multiplataforma e ser voltada para atender as necessidades do desenvolvimento de aplicações em ambientes distribuídos e heterogêneos. Portanto, Java concede um modo mais prático de desenvolver, implementar e validar os conceitos do PolimorFI.

Outro motivo para escolha dessa linguagem de programação, Java, é que a arquitetura de Internet do Futuro estudada nesta dissertação, a RINA, possui protótipos em diferentes estágios de implementação nesta linguagem (POUZIN SOCIETY, [s.d.]) e em linguagem semelhante, como é o caso de C/C++ (DAY, 2018). O que corrobora com o intuito de implementar componentes RINA no PolimorFI para realização de testes de aplicações.

Como auxílio para desenvolver e implantar aplicações, foi acoplado ao PolimorFI a biblioteca *Java SE Development Kit*⁷, que disponibiliza ferramentas para desenvolvimento, depuração e monitoramento de aplicações. Toda a implementação foi realizada utilizando a IDE Eclipse⁸, versão *Indigo Service Release 2 Build id: 20120216-1857*.

⁷ <https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html>

⁸ <http://www.eclipse.org/webtools>

4.6 CONCLUSÕES

Pode-se concluir que a atual implementação do arquétipo do PolimorFI funciona perfeitamente, executando a aplicação e realizando *publish* e *subscribe* de mensagens CDAP entre os processos transmissor e receptores, conforme o esperado.

Todos os requisitos foram implementados e as funções análogas foram empacotadas e abstraídas do desenvolvimento.

A prova de conceito conclui que o PolimorFI facilita o desenvolvimento de aplicações para arquiteturas de Internet do Futuro, haja vista que possibilitou o desenvolvimento de uma aplicação teste que utilizou-se dos protocolos da RINA para transportar mensagens entre processos transmissor e receptores, de modo que o foco do desenvolvimento ocorreu no propósito da aplicação. A sua arquitetura faz uso de tecnologias correlacionadas com engenharia de *software*, sistemas de comunicação e linguagem de programação.

As evidências demonstram que é possível o desenvolvedor construir novas aplicações para diferentes arquiteturas de rede, de forma transparente. Ainda, sem a preocupação com o fato de ser distribuída ou não, e problemas atrelados a essa característica, como problemas de rede ou localização, é algo bastante desejado. O PolimorFI objetiva tratar de problemas que não estão absolutamente ligados à aplicação.

O PolimorFI fornece suporte a modelo de mensagens *Publish/Subscriber* de modo a maximizar seu desempenho. Além disso, provê uma abstração da arquitetura de rede para que o desenvolvedor de aplicações não precise empenhar esforço com a comunicação de forma distribuída das suas aplicações e processos.

Demonstrou-se ainda que, dada a facilidade no desenvolvimento de aplicações com o PolimorFI, os desenvolvedores podem explorar os recursos da arquitetura, desenvolvendo e implantando aplicações sem muita complexidade para funcionarem em novas redes.

É importante frisar que, embora a arquitetura de Internet do Futuro observada nesta dissertação seja a RINA - que se baseia no paradigma de comunicação Inter-processos, tem-se o propósito venturo de manter esse trabalho extensível para experimentar outras iniciativas, como por exemplo, redes centradas em informações (ICN).

Diante do exposto, os testes realizados com o PolimorFI foram considerados satisfatórios na prova de conceito por atender os requisitos determinados na subseção 3.1.1. Tais requisitos, citados abaixo, foram atendidos conforme a seguir:

- **Homogeneidade** - O polimorFI tratou da homogeneidade de tecnologia, uma vez que a sua arquitetura foi desenvolvida utilizando a mesma tecnologia: Java.
 - **Retrocompatibilidade** - o PolimorFI mostrou-se ser retrocompatível com a Internet atual, haja vista a utilização de *socket TCP (API Shim TCP Layer)* para estabelecer conexões TCP para as DIFs.
 - **Transporte de Mensagens** - O transporte de mensagens foi realizado utilizando um paradigma de comunicação específico da RINA, IPC, por meio de uma aplicação teste usando o modelo *publish/subscribe*.
 - **Serialização de Mensagens** - As mensagens foram serializadas por meio do componente *Serializer*, na qual os objetos foram convertidos em *bytes* para serem transferidos.
 - **Heterogeneidade** – A instância da arquitetura de Internet do Futuro foi transparente para a execução da aplicação.
 - **Ferramental para o Desenvolvimento** – Para atendimento deste requisito, foi tomada a biblioteca *Java SE Development Kit* que forneceu as ferramentas para a escrita da aplicação.
-

5 AVALIAÇÃO EXPERIMENTAL

Nesta seção será detalhada a avaliação de desempenho do serviço de envio e recebimento de mensagens da prova de conceito com a utilização do *Middleware* PolimorFI. Será analisado o tempo de resposta necessário para que o PolimorFI entregue as mensagens a seus respectivos destinatários para diferentes números de assinantes e de mensagens enviadas.

5.1 DEFINIÇÃO DOS EXPERIMENTOS

5.1.1 *Objetivo*

O objetivo deste estudo é avaliar o sistema utilizado na prova de conceito com o uso do PolimorFI em termos do tempo de resposta para a entrega de mensagens a seus respectivos destinatários e analisar o impacto causado em relação ao aumento do número de nós *Subscribers*.

5.1.2 *Serviço*

O serviço avaliado será o envio e recebimento de mensagens entre os nós, chamados de *Publisher* e *Subscribers*. O *publisher* publica mensagens de um determinado tópico que são consumidas por um determinado número de nós (*Subscribers*) que manifestaram interesse em mensagens de tal tópico.

5.1.3 *Métrica*

A métrica avaliada será a latência. Essa métrica de resposta representa o tempo médio gasto pelo *Middleware* para entrega de uma mensagem, ou seja, quanto tempo a mensagem leva desde o seu envio até chegar a resposta, em milissegundos (ms).

Esses tempos são relevantes para aplicações que operam conforme o princípio de requisição, como por exemplo aplicações baseadas em chamadas de procedimentos remotos (RPC - *Remote Procedure Call*).

Além disso, para cada quantidade de mensagens publicadas, outras métricas serão consideradas para análise, quais sejam:

- Tempo total no processo de envio de mensagens;
- Tempo médio no processo de envio de mensagens.

5.1.4 Parâmetros de Carga de Trabalho

Os parâmetros que podem sofrer variações chamamos de fatores de desempenho e, portanto, estes podem impactar diretamente no desempenho do sistema. Essas variações que podem ocorrer são classificadas em níveis.

Têm-se como boa prática na definição dos fatores a serem utilizados, começar com o mínimo de fatores e poucos níveis em cada, de modo a ir crescendo os níveis conforme necessidade (SOUZA, 2017). Nesse sentido, é interessante dar prioridade aos parâmetros que mais podem influenciar no desempenho. No *Middleware PolimorFI*, os fatores e níveis podem ser observados na Tabela 4:

Tabela 4 - Fatores para os experimentos

FATORES	NÍVEIS
Número de nós (<i>Subscribers</i>)	1, 5, 10, 15, 20
Número de mensagens enviadas	100, 500, 1000

Fonte: elaborada pelo autor

A seleção da carga de trabalho (quantitativo da Tabela 4) foi assentada com o intuito de se aproximar de um ambiente compatível com o mundo real, considerando aspectos relevantes quanto ao desempenho do PolimorFI (transação), no que corresponde à quantidade de nós determinados para o experimento.

De acordo com ARTEIRO (2009), o componente básico da carga de trabalho, no caso de um sistema de transporte de mensagens, são as mensagens enviadas pelos “produtores” para serem encaminhadas aos “consumidores” em um intervalo de tempo. Na avaliação experimental do PolimorFI, o tempo entre mensagens enviadas, ou intervalo de requisição, trata-se de um parâmetro fixo de 10 milissegundos (ms), formando a configuração do cenário a ser avaliado, o que reflete um contexto de

aplicações de IoT, por exemplo, o qual as mensagens são transmitidas em um curto intervalo de tempo (RAZZAQUE et al., 2016).

5.1.5 *Preparações dos Experimentos*

Uma vez definidos os fatores e níveis, faz-se necessário elaborar como os experimentos serão realizados. O *hardware* utilizado como servidor do PolimorFI foi uma Máquina com processador Intel(R) Core(TM) i5-3337U CPU @ 1.80GHz, com 6 GB de memória RAM 1600 MHz DDR3 (4 GB + 2 GB) e Sistema Operacional (SO) Ubuntu xenial Versão 16.04.3 LTS. Para representar os nós (*Publisher* e *Subscribers*), foram utilizadas máquinas virtuais (*Virtual Machine* – VM) com 512 MB de memória RAM e processador com duas unidades de processamento independentes (núcleo) e sistema operacional Ubuntu.

5.1.6 *Técnica de Avaliação*

A técnica de avaliação usada nesta dissertação foi a **medição**, e está de acordo com a proposta apresentada por JAIN (1991), pois esta permite testar a solução desenvolvida em uma situação muito próxima da aplicação real.

5.2 AMBIENTE DE MEDIÇÕES

A atividade de medição é uma parte chave para todas as tarefas da área de avaliação de desempenho, formando a base para a composição de um modelo capaz de representar a carga real submetida ao sistema. Para esse processo, é necessário utilizar monitores para coletar as informações de desempenho do sistema. Monitores são ferramentas utilizadas para observar as atividades de um sistema, e em geral coletam informações estatísticas, analisam os dados e apresentam os resultados (JAIN, 1991).

O *Middleware* PolimorFI fora submetido a uma carga de trabalho, conforme fatores mencionados na Tabela 4, a qual ocorreu a “publicação” de um número considerável de mensagens que foram “assinadas” por um número determinado de nós, isso em um curto período de tempo (10 ms). As mensagens foram enviadas e, nesse processo, foi medido o tempo que a mensagem leva desde o seu envio até chegar a resposta. Para calcular a métrica (latência), e se obter a precisão em milissegundos, foi utilizado o método *Timestamp* com a classe `java.util.Date`.

O experimento de medição foi conduzido com base no disposto por ARTEIRO (2009), de forma que os *subscribers* foram iniciados antes dos *publishers*, para evitar que mensagens sejam acumuladas na fila.

Nesse universo, foram realizados testes individuais para cada quantidade de nós que fazem *subscribers*. O *Timestamp* foi inserido no *publisher* e, em seguida, analisado nos *subscribers*. Após a coleta dos dados, foi necessário realizar uma análise exploratória dos dados, tais quais: Tempo Total, Tempo Médio, Desvio Padrão e o Intervalo de Confiança.

5.3 APRESENTAÇÃO DOS RESULTADOS

Foi executada uma bateria de testes com a finalidade de realizar uma análise de desempenho **inicial** do PolimorFI em relação ao tempo gasto para *publish* e *subscribes* de mensagens. A bateria consistiu em testes individuais para cada quantidade de nós que fazem *subscribes*. Foram realizados experimentos baseando-se no impacto de desempenho causado pela publicação de 1000, 500 e 100 mensagens, repetindo trinta vezes cada cenário.

A Tabela 5 apresenta valores dos experimentos realizados na publicação e assinaturas de mensagens. Nela, podem ser observados os resultados dos tempos médio e total; dos intervalos de confiança; e desvio padrão. Os resultados para mais e para menos do erro em relação à média das trinta amostras foram alcançados com um nível de confiança de 95%.

Tabela 5 - Tempo de envio de mensagens (ms)

QUANTIDADE MENSAGENS = 100					
QNT. DE <i>SUBSCRIBERS</i>	1	5	10	15	20
TEMPO MÉDIO (ms)	1,78	8,63	81,9	434	508
TEMPO TOTAL (ms)	0,34	1,53	2,96	31,6	4,01
DESVIO PADRÃO (ms)	0,95	0,95	0,95	0,95	0,95
NÍVEL DE CONFIANÇA	0,5	0,5	0,5	0,5	0,5
NÍVEL DE SIGNIFICÂNCIA	0,04	0,19	0,36	3,89	0,49
TOLERÂNCIA/ERRO	1,82	8,82	82,2	437	509
INTERVALO DE CONFIANÇA (ms)	1,74	8,44	81,5	430	508
	1,78	8,63	81,9	434	508
QUANTIDADE DE MENSAGENS = 500					
QNT. DE <i>SUBSCRIBERS</i>	1	5	10	15	20
TEMPO MÉDIO (ms)	2,99	14,95	126	346	593

TEMPO TOTAL (ms)	89,6	449	3770	10390	17784
DESVIO PADRÃO (ms)	0,70	4,36	7,15	6,37	4,36
NÍVEL DE CONFIANÇA	0,95	0,95	0,95	0,95	0,95
NÍVEL DE SIGNIFICÂNCIA	0,5	0,5	0,5	0,5	0,5
TOLERÂNCIA/ERRO	0,09	0,54	0,88	0,78	0,54
INTERVALO DE CONFIANÇA (ms)	3,07	15,5	126,6	347	593
	2,90	14,4	124,8	346	592
QUANTIDADE DE MENSAGENS = 1000					
QNT. DE SUBSCRIBERS	1	5	10	15	20
TEMPO MÉDIO (ms)	14,8	66,1	348	729	1235
TEMPO TOTAL (ms)	444	1984	10431	21855	37050
DESVIO PADRÃO (ms)	3,6	22,8	52,1	31,4	17,2
NÍVEL DE CONFIANÇA	0,95	0,95	0,95	0,95	0,95
NÍVEL DE SIGNIFICÂNCIA	0,5	0,5	0,5	0,5	0,5
TOLERÂNCIA/ERRO	0,45	2,80	6,42	3,86	2,12
INTERVALO DE CONFIANÇA (ms)	15,3	68,9	354	732	1237
	14,4	63	341	725	1233

Fonte: elaborada pelo autor

Ao analisar a Tabela 5, nota-se que, conforme esperado, o PolimorFI apresentou um melhor desempenho – menor latência – quando enviado o menor número de mensagens (carga de mensagens igual a 100). Enquanto que, quando enviada a carga de 500 mensagens, segundo maior número de carga de mensagens submetidas, o desempenho caiu em comparação à primeira carga, de modo que o tempo médio de envio de mensagens cresceu cerca de 5%.

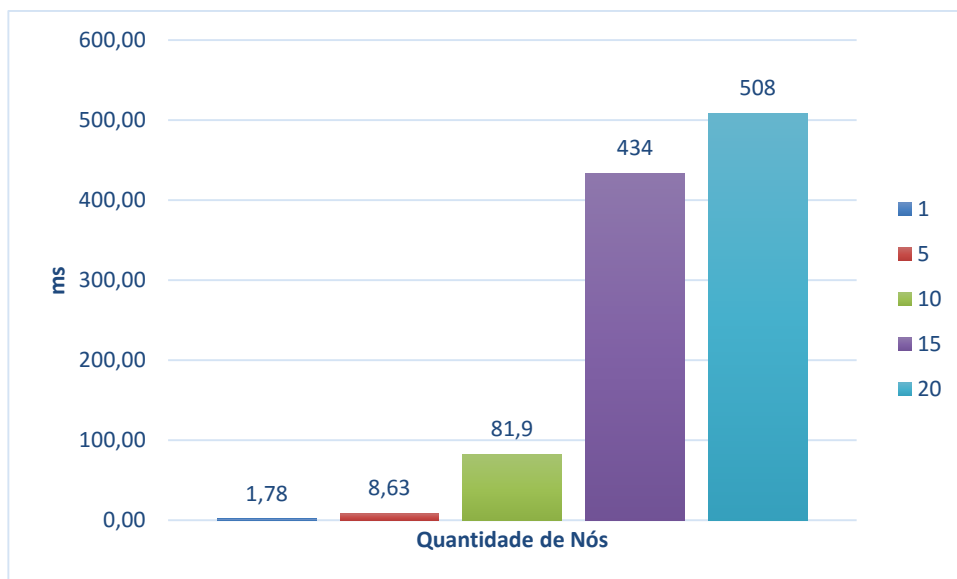
Por conseguinte, ao analisar a maior carga de mensagens (carga de mensagens igual a 1000), a latência continua crescendo exponencialmente, sendo esse crescimento equivalente à 131% em relação à carga de 100 mensagens e, 121%, em relação à carga de 500 mensagens, ocasionando assim uma latência maior na entrega da mensagem, levando o sistema a acumular requisições nas filas aguardando pelo recurso.

Ainda na Tabela 5, nota-se que o desvio padrão se mantém em nível consideravelmente baixo, indicando um desempenho satisfatório em relação à média do tempo decorrido no envio das mensagens.

A fim de melhorar o entendimento, os gráficos abaixo (Figura 19, Figura 20 e Figura 21) mostram o comportamento da latência quando se varia a quantidade de nós assinantes e da carga de mensagens enviadas. Nelas é possível visualizar os

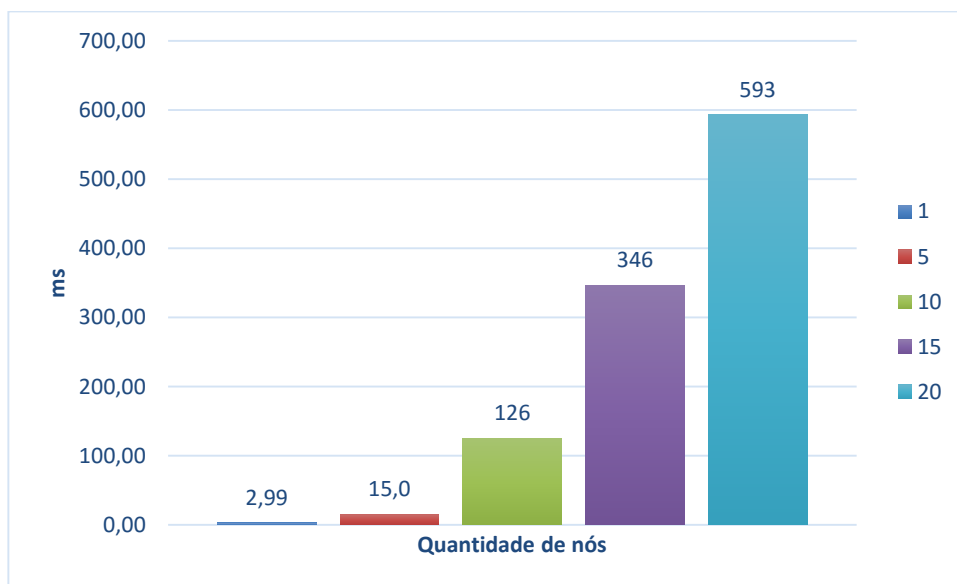
tempos médios para assinaturas de mensagens (em ms) pelos conjuntos de nós com a carga de 100; 500; e 1000 mensagens, respectivamente.

Figura 19 - Latência X Número de nós (100 mensagens)

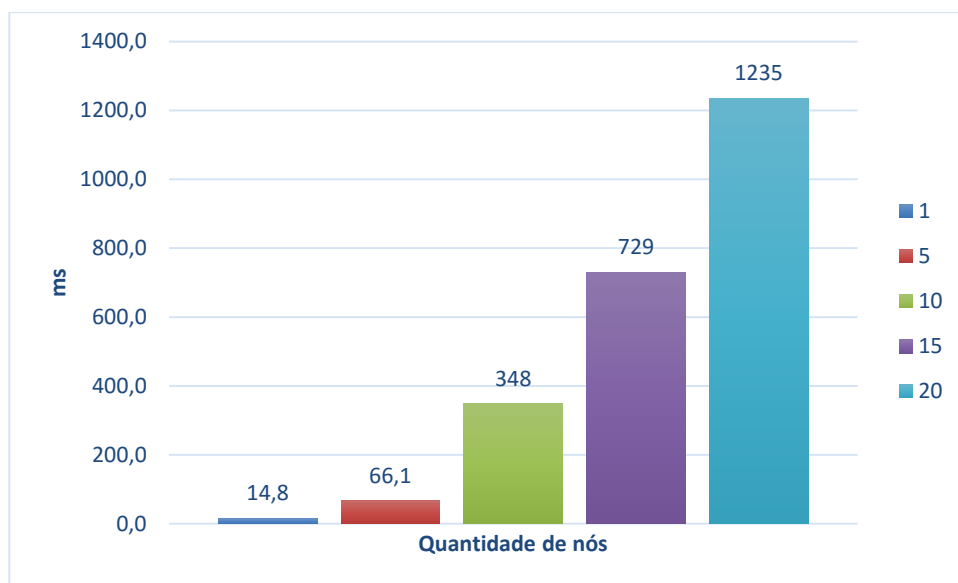


Fonte: elaborada pelo autor

Figura 20 - Latência X Número de nós (500 mensagens)

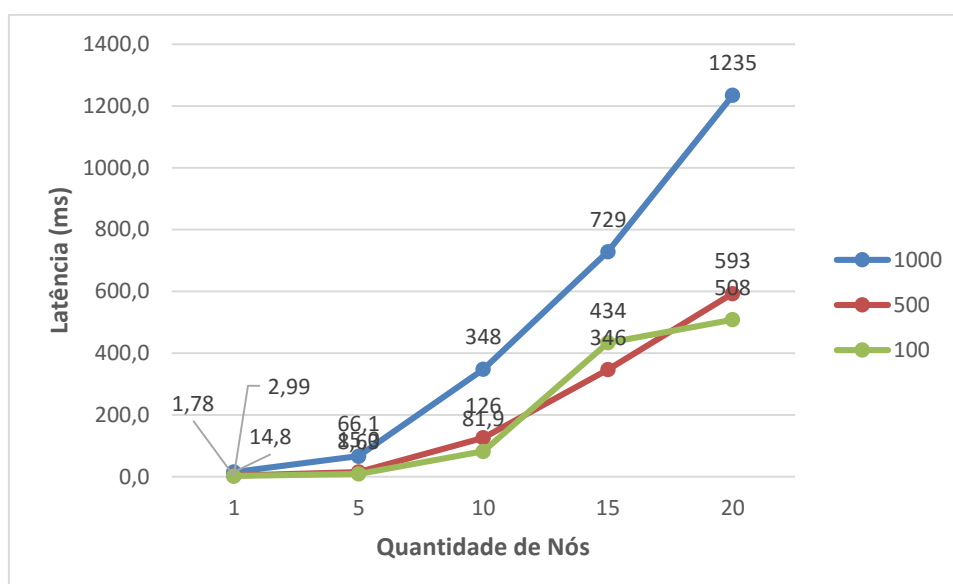


Fonte: elaborada pelo autor

Figura 21 - Latência X Número de nós (1000 mensagens)

Fonte: elaborada pelo autor

A Figura 22 traz um comparativo geral onde observa-se um crescimento gradual entre as cargas de mensagens e a quantidade de nós assinantes, contudo percebe-se uma expansão elevada da latência a partir de 10 nós quando enviado o quantitativo de 1000 mensagens.

Figura 22 - Comparativo geral

Fonte: elaborada pelo autor

Ainda na Figura 22, nota-se que a linearidade de crescimento da latência se mantém de forma progressiva para as demais cargas, porém identifica-se que, no ponto

de 15 nós, sucede uma concorrência entre as cargas de 100 e 500 mensagens e a maior quantidade de mensagens (500) levou menos tempo para serem entregues referente à menor quantidade (100).

Considerando a latência como o tempo de processamento das mensagens, o esperado é que, de fato, ela cresça com o número de assinantes para as mensagens, no entanto é observado que ocorre um certo impacto no perfil de crescimento da latência no envio de 1000 mensagens em relação à 500, dado que se elevou 116% ao comparar o percentual entre a carga de 500 e 100 mensagens, que se mostrou 5%.

5.4 CONSIDERAÇÕES FINAIS

Esta seção apresentou a avaliação experimental realizada com o *Middleware* PolimorFI. O objetivo foi avaliar o desempenho da solução em uma situação real. A metodologia de avaliação seguiu uma sequência de etapas, todas elas descritas e explicadas no decorrer da seção.

Pelos dados apresentados nos gráficos (Figura 19, Figura 20 e Figura 21) da seção anterior, nota-se que os tempos médios são estatisticamente diferentes para cada quantidade de mensagens enviadas. Por meio do gráfico apresentado na Figura 22, pode-se concluir que no envio de 1000 mensagens leva o maior tempo em comparação aos demais conjuntos. Percebe-se também que o tempo de latência necessário para envio e recebimento das mensagens cresce exponencialmente conforme a quantidade de nós aumenta, isto é, ocorre um certo impacto. Conforme o número de mensagens diminui, esse impacto tende a ser minimizado.

Inicialmente, com o aumento da carga submetida ao *Middleware* PolimorFI, a tendência é que a performance se eleve proporcionalmente, ou seja, a latência estabilize a partir de uma quantidade elevada de mensagens transportadas, aspecto esse (alta demanda de interoperação entre aplicações) que representa o cenário da Internet do Futuro onde caracteriza-se um grande volume de intercomunicação entre diferentes dispositivos com requisições intrínsecas. Depois de atingir um certo nível de carga de mensagens, acredita-se que a latência passará a crescer numa proporção menor que a carga submetida

Constata-se que, para uma mesma quantidade de nós, tem-se um tempo aumentado de latência para quantidades maiores de mensagens. Neste sentido, diferentes estratégias podem ser tomadas a fim de minimizar a latência necessária para o conjunto de nós, visando eficiência do *Middleware*. Uma delas é implementar paralelismo e observar o impacto causado. Outra abordagem para analisar esse fenômeno é aglutinar em uma mensagem uma quantidade menor de informação a ser enviada e observar se irá manter a mesma latência com um número maior de mensagens.

Para os cenários testados, não houve perda de pacotes de mensagens. Isto é, nesse primeiro experimento, o *Middleware* mostrou ser viável para aplicações distribuídas que não são de tempo real.

Face ao exposto, estes primeiros resultados serviram para apresentar informações pertinentes às funcionalidades do protótipo. Em um segundo momento, estes resultados poderão servir como parâmetros de análise comparativa com outras arquiteturas de *Middleware*, como por exemplo para IoT, que demandam uma carga elevada de troca de mensagens entre dispositivos distintos.

6 CONCLUSÕES E TRABALHOS FUTUROS

A última seção deste trabalho apresenta as considerações finais, os resultados obtidos relacionados com os objetivos iniciais da pesquisa, as possibilidades para realização de trabalhos futuros e a conclusão final do trabalho.

6.1 CONSIDERAÇÕES FINAIS

Novas propostas de arquiteturas de redes voltadas para a Internet do Futuro surgiram com o objetivo de ser uma alternativa ao modelo convencional da Internet. Haja vista as limitações apresentadas na estrutura da Internet atual, as aplicações que surgiram ao longo das décadas foram sendo adaptados à rede de forma deficiente, nesse sentido, as pesquisas com novas arquiteturas de redes têm avançado com a finalidade de atender demandas emergentes das aplicações.

Partindo desse pressuposto, esta dissertação apresentou o PolimorFI, um *Middleware* Orientado a Mensagens para suportar o desenvolvimento de aplicações para arquiteturas de Internet do futuro. A proposta apresenta, primeiramente, experimentos realizados com RINA, seguida de uma avaliação experimental do *Middleware*.

A prova de conceito implementada foi capaz de validar os componentes e as funcionalidades do PolimorFI. O protótipo apresentou êxito ao prover os requisitos, conforme mostrado no subitem 4.6. Além disso, a arquitetura do PolimorFI foi constituída nos padrões de projetos para o desenvolvimento de *Middleware*.

Face a este trabalho, os pesquisadores envolvidos com o tema Internet do Futuro terão mais informações para analisar e comparar os mecanismos de comunicação de redes com as outras abordagens *clean slate*. Através dos levantamentos feitos nessa pesquisa, considera-se que os esforços em Internet do Futuro precisam evoluir também para a camada de aplicações, a fim de alavancar a projeção de futuras aplicações de Internet sobre novas arquiteturas de rede. Embora os resultados alcançados tenham sido satisfatórios, há outras possibilidades em aberto para melhorar e ampliar o trabalho atual.

6.2 CONTRIBUIÇÕES

A principal contribuição deste trabalho é a construção do PolimorFI, um *Middleware* Orientado a Mensagens capaz de possibilitar o desenvolvimento de aplicações específicas para rodarem em cima de arquiteturas de Internet do Futuro, tendo como ponto de partida uma prova de conceito com arquitetura baseada no paradigma IPC, a RINA.

A arquitetura do PolimorFI pode ser tomada também como um metamodelo, de modo a descrever componentes que podem ser construídos utilizando-se outros padrões de projetos ou modelos arquiteturais, auxiliando, portanto, no desenvolvimento de novos módulos para aplicações peculiares.

Outra contribuição importante é ter a arquitetura do PolimorFI como modelo de desenvolvimento para novas soluções de *middleware* voltadas ao contexto de Internet do Futuro.

6.3 TRABALHOS FUTUROS

A seguir são apresentadas algumas possibilidades de trabalhos futuros:

- Experimentar o desenvolvimento de aplicações voltadas para outros paradigmas de rede, como por exemplo ICN;
 - Estender a prova de conceito abarcando a acoplação de outras instâncias de arquiteturas de Internet do Futuro, com paradigmas distintos, no *Middleware*;
 - Realizar um estudo quantitativo dos resultados apresentados pelo uso do PolimorFI;
 - Identificar o comportamento do tráfego de mensagens ao longo do tempo, permitindo identificar situações de pico de utilização;
 - Desenvolver recursos que possibilitem o gerenciamento de eventos (falhas/sucesso) no envio de mensagens entre as aplicações;
 - Realizar pesquisa de campo com desenvolvedores de aplicações para coletar informações referentes à facilidade do desenvolvimento por meio do PolimorFI.
-

REFERÊNCIAS

ALBERTI, A. M. et al. Performance Evaluation of NovaGenesis Information-Centric Network. **2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)**, 2017.

AMBROSIN, M. et al. Security and Privacy Analysis of NSF Future Internet Architectures. p. 1–21, 2016.

AMDOUNI, I. et al. Exploring the flexibility of network access control in the recursive InterNetwork Architecture. **Proceedings - Asia-Pacific Conference on Communications, APCC 2016**, p. 559–566, 2016.

ARTEIRO, R. D. **JMSCAPACITY - Um ToolKit para auxiliar no planejamento de capacidade de Middleware Orientado a Mensagem**. [s.l.] Dissertação de Mestrado. Universidade Federal de Pernambuco, 2009.

BOUK, S. H. et al. IEEE Access Special Section Editorial: Future Networks: Architectures, Protocols, and Applications. **IEEE Access**, v. 5, p. 27831–27835, 2017.

BRAUN, S.; MONTI, M. CCN & TCP co-existence in the future Internet: Should CCN be compatible to TCP ? **IFIP/IEEE International Symposium on Integrated Network Management (IM)**, p. 1109–1115, 2013.

BRAUN, T. et al. Service-centric networking. **IEEE International Conference on Communications**, 2011.

CANALTECH. **Mercado de aplicativos para celular deve movimentar US\$ 139 bi em 2021**. Disponível em: <<https://canaltech.com.br/apps/mercado-de-aplicativos-para-celular-deve-movimentar-us-139-bi-em-2021-91938/>>.

CAPORUSCIO, M.; GHEZZI, C. Engineering Future Internet applications: The Prime approach. **Journal of Systems and Software**, v. 106, p. 9–27, 2015.

CASTA, L.; JAIN, P.; HAUSI, A M. The Future of Internet Applications : A Survey of Future Internet Projects. **The Future of Internet Applications: A Survey of Future**

Internet Projects, v. 91, p. 1–51, 2013.

COULORIS, G. et al. **Distributed Systems: Concepts and Design**. Fifth Edition ed. [s.l.: s.n.].

CROTTY, M.; PUENTE, M. A. Simplifying Multi-Layer Network Management With Rina : Analysis of a Multi-Tenant Data Center Network. **TNC16 Networking Conference**, p. 1–15, 2016.

DAY, J. et al. **ProtoRINA Project**. Disponível em: <<https://github.com/ProtoRINA/users/wiki>>.

DAY, J. **Patterns in Network Architecture: A Return to Fundamentals**. 1 edition ed. [s.l.] Prentice Hall, 2008.

DAY, J. **Look Before You Leap**. 2018. Disponível em: <https://www.itu.int/en/ITU-T/Workshops-and-Seminars/201810/Documents/John_Day_Presentation.pdf>

EUGSTER, P. T. et al. The many faces of publish/subscribe. **ACM Computing Surveys**, v. 35, n. 2, p. 114–131, 2003.

FERNANDES, N. C. et al. Virtual networks: Isolation, performance, and trends. **Annales des Telecommunications/Annals of Telecommunications**, v. 66, n. 5–6, p. 339–355, 2011.

FISHER, D. A look behind the future internet architectures efforts. **ACM SIGCOMM Computer Communication Review**, v. 44, n. 3, p. 45–49, 2014.

GAIXAS, S. L. et al. Assuring QoS guarantees for heterogeneous services in RINA networks with ΔQ . **Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom**, p. 584–589, 2017.

GALLO, M. et al. NaNET : Socket API and Protocol Stack for Process-to-Content Network Communication. **Proceedings of the 1st international conference on Information-centric networking**, p. 185–186, 2014.

GEORGANTAS, N. et al. Service-oriented distributed applications in the future internet: The case for interaction paradigm interoperability. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 8135 LNCS, p. 134–148, 2013.

GRASA, E. et al. Seamless network renumbering in RINA: Automate address changes without breaking flows! **EuCNC 2017 - European Conference on Networks and Communications**, 2017.

GRZECH, A. et al. Applications of the future internet engineering project. **Proceedings - 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2012**, p. 635–642, 2012.

ICIN. **6th International Workshop on the Recursive InterNetwork Architecture (RINA 2019)**. Disponível em: <<https://www.icin-conference.org/RINA.php>>.

ISSARNY, V. et al. Service-oriented middleware for the Future Internet: State of the art and research directions. **Journal of Internet Services and Applications**, v. 2, n. 1, p. 23–45, 2011.

ITT, I. DI I. E T.-. **Ubiquitous Internet**. Disponível em: <https://www.iit.cnr.it/en/research_fields/ubiquitous_internet>.

JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. [s.l: s.n.].

JÄPPINEN, P.; GUARNERI, R.; CORREIA, L. M. An applications perspective into the Future Internet. **Journal of Network and Computer Applications**, v. 36, n. 1, p. 249–254, 2013.

KRISHNAMACHARI, B.; WRIGHT, K. The Publish-Process-Subscribe Paradigm for the Internet of Things. **USC ANRG Technical Report**, v. 91, n. July, p. 399–404, 2017.

LEON GAIXAS, S. et al. Scalable topological forwarding and routing policies in RINA-enabled programmable data centers. **Transactions on Emerging Telecommunications Technologies**, v. 28, n. 12, p. 1–23, 2017.

LEON, S. et al. Guaranteeing QoS requirements in long-haul RINA networks. **International Conference on Transparent Optical Networks**, p. 2–5, 2017.

LI, B. et al. Future internet services and applications. **IEEE Network**, v. 24, n. 4, p. 4–5, 2010.

LI, R. **Network 2030: Market Drivers and Prospects**. The 1st ITU Workshop on Network 2030. **Anais...**New York: 2018. Disponível em: <<https://www.itu.int/en/ITU-T/Workshops->

and-Seminars/201810/Documents/Richard_Li_Presentation.pdf>

MAFFIONE, V. et al. A software development kit to exploit RINA programmability. **2016 IEEE International Conference on Communications, ICC 2016**, p. 1–7, 2016.

MARTIN, D.; WIPPEL, H. Evaluating a framework for different networking paradigms. **Proceedings - Conference on Local Computer Networks, LCN**, p. 288–291, 2013.

MEER, S. VAN DER et al. Building the Future Internet through FIRE. In: **Building the Future Internet through FIRE**. [s.l.] Rivers Publishers, 2017. p. 1–794.

MONTEIRO, JOSÉ A. S. **Tópicos Especiais em Redes de Computadores 1**. Disponível em: <<http://www.cin.ufpe.br/~suruagy/cursos/AR/welcome.html>>.

OPUS SOFTWARE. **A Nova Geração de Aplicativos Móveis.pdf**. Disponível em: <<https://www.opus-software.com.br/a-nova-geracao-de-aplicativos-moveis/>>.

ORACLE9I. **Using the Publish-Subscribe Model for Applications**. Disponível em: <http://docs.oracle.com/cd/B10501_01/appdev.920/a96590/adg15pub.htm>.

PAN, J.; PAUL, S.; JAIN, R. FIArch-survey.pdf. n. July, p. 26–36, 2011a.

PAN, J.; PAUL, S.; JAIN, R. A survey of the research on future internet architectures. **Communications Magazine, IEEE**, v. 49, n. 7, p. 26–36, 2011b.

PEREZ, A. J.; LABRADOR, M. A.; BARBEAU, S. J. G-Sense: A scalable architecture for global sensing and monitoring. **IEEE Network**, v. 24, n. 4, p. 57–64, 2010.

POUZIN SOCIETY. **protoRINA**. Disponível em: <<http://pouzinsociety.org/implementations/protorina>>.

RAZZAQUE, M. A. et al. Middleware for internet of things: A survey. **IEEE Internet of Things Journal**, v. 3, n. 1, p. 70–95, 2016.

REXFORD, J.; DOVROLIS, C. Future Internet architecture: Clean-Slate Versus Evolutionary Research. **Communications of the ACM**, v. 53, n. 9, p. 36, 2010.

RITCHEY DIANE. **The Unstoppable Convergence Between Physical and Cybersecurity | 2018-04-01 | Security Magazine**. Disponível em: <<https://www.securitymagazine.com/articles/88847-the-unstoppable-convergence->

between-physical-and-cybersecurity>.

RNP. **Internet do Futuro**. Disponível em: <<https://memoria.rnp.br/pd/if/>>.

SANTHI, R. K. R. **A Service Based Approach For Future Internet Architectures**. [s.l.] Tese de Doutorado. University of Agder, 2010.

SILVEIRA, P. et al. **Introdução À Arquitetura De Design De Software. Uma Introdução À Plataforma Java**. [s.l.] 9th December 2011, 2011. v. 1

SIRIS, V. A. et al. Information-Centric Networking (ICN) architectures for integration of satellites into the Future Internet. **2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)**, p. 1–6, 2012.

SMALL, J. **Patterns in Network Security: an Analysis of Architectural Complexity in Securing Recursive Inter-Network Architecture Networks**. [s.l.] Dissertação de Mestrado. University of Massachusetts, 2012.

SMALL, J.; DAY, J.; CHITKUSHEV, L. Threat Analysis of Recursive Inter-Network Architecture Distributed Inter-process Communication Facilities. **Technical Report. Boston University Technical Report**, p. 1–9, 2011.

SOUSA, W. F. **XPRESUMO – Um Middleware Orientado à Mensagens para Internet das Coisas**. [s.l.] Dissertação de Mestrado. Universidade Federal de Pernambuco, 2017.

STAESSENS, D. Open Call Deliverable OCB-DS2 . 1 Final Report on Experimentation & Results Table of Contents. 2015.

TERZIYAN, V.; ZHOVTOBRYUKH, D.; KATASONOV, A. Proactive future internet: Smart semantic middleware for overlay architecture. **Proceedings of the 5th International Conference on Networking and Services, ICNS 2009**, p. 149–154, 2009.

TEYMOORI, P. et al. Congestion control in the recursive InterNetworking Architecture (RINA). **2016 IEEE International Conference on Communications, ICC 2016**, p. 1–7, 2016.

VALENTE, B. A. L. **Um Middleware Para a Internet Das Coisas**. [s.l.: s.n.].

VÖLTER, M.; KIRCHER, M.; ZDUN, U. **Remoting patterns : foundations of enterprise, Internet and realtime distributed object middleware**. 1. ed. [s.l.] Wiley, 2007.

VRIJDERS, S. et al. Experimental evaluation of a Recursive InterNetwork Architecture prototype. **2014 IEEE Global Communications Conference, GLOBECOM 2014**, p. 2017–2022, 2014.

VRIJDERS, S. et al. Rumba: A python framework for automating large-scale Recursive Internet Experiments on GENI and FIRE+. **INFOCOM 2018 - IEEE Conference on Computer Communications Workshops**, p. 324–329, 2018.

WANG, Y. et al. Introducing protorinA: A prototype for programming recursive-Networking policies. **Computer Communication Review**, v. 44, n. 3, p. 129–131, 2014.

WANG, Y.; MATTA, I. **ProtoRINA 2.0**, 2017. Disponível em: <<http://csr.bu.edu/rina/protorina/2.0/>>

WANG, Y.; MATTA, I.; AKHTAR, N. Application-driven network management with ProtoRINA. **Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium**, p. 873–877, 2016.

WILCOX, J. R. et al. Verdi: a framework for implementing and formally verifying distributed systems. **Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation - PLDI 2015**, p. 357–368, 2015.

ZHOU, J. et al. SDN-Based Application Framework for Wireless Sensor and Actor Networks. **IEEE Access**, v. 4, p. 1583–1594, 2016.