

UNIVERSIDADE FEDERAL DO PAMPA

Leandro Martins Dallanora

Orquestração Dinâmica da Telemetria de
Rede *In-band* em Planos de Dados
Programáveis

Alegrete
2022

Leandro Martins Dallanora

**Orquestração Dinâmica da Telemetria de Rede
In-band em Planos de Dados Programáveis**

Dissertação de Mestrado apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Marcelo Caggiani Luizelli

Alegrete
2022

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

D144o Dallanora, Leandro Martins

Orquestração dinâmica da telemetria de rede in-band em
planos de dados programáveis / Leandro Martins Dallanora.
75 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa,
MESTRADO EM ENGENHARIA ELÉTRICA, 2022.

"Orientação: Marcelo Caggiani Luizelli".

1. Orquestração de telemetria de rede in-band. 2.
Programabilidade de rede. 3. Monitoramento de rede. I. Título.

Leandro Martins Dallanora

**Orquestração Dinâmica da Telemetria de Rede
In-band em Planos de Dados Programáveis**

Dissertação de Mestrado apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Dissertação de Mestrado defendida e aprovada em 28 de outubro de 2022

Banca examinadora:

Prof. Dr. Marcelo Caggiani Luizelli

Orientador

Universidade Federal do Pampa - Campus Alegrete

Prof. Dr. Fabio Diniz Rossi

Instituto Federal Farroupilha - Campus Alegrete

Prof. Dr. Roberto Irajá da Costa Filho

Instituto Federal Sul-rio-grandense - Campus Charqueadas



Assinado eletronicamente por **MARCELO CAGGIANI LUIZELLI, PROFESSOR DO MAGISTERIO SUPERIOR**, em 03/11/2022, às 14:17, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **Roberto Irajá Tavares da Costa Filho, Usuário Externo**, em 03/11/2022, às 14:38, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **Fábio Diniz Rossi, Usuário Externo**, em 04/11/2022, às 23:28, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0975672** e o código CRC **5EEA6EFE**.

Dedico este trabalho à minha família, por todo amor, incentivo, apoio e compreensão.
Nada disso teria sentido se vocês não existissem na minha vida.

AGRADECIMENTOS

Nesses anos de mestrado, de muito estudo, esforço e empenho, gostaria de agradecer a algumas pessoas e instituições que me acompanharam e foram fundamentais para a realização de mais este sonho.

Primeiramente, minha sincera gratidão ao meu orientador **Prof. Dr. Marcelo Caggiani Luizelli**, pela confiança depositada em mim, pelo incentivo, disponibilidade e apoio que sempre demonstrou.

Aos **meus pais** e à **minha família**, por acreditarem em mim, pela paciência nos meus momentos mais difíceis, e pela compreensão, nas incontáveis horas que fiquei ausente para me dedicar a este trabalho que concluo com muito orgulho.

Agradeço aos meus amigos **Ariel de Castro, Jhonathan Silveira e Pablo Viegas**, e também à **Universidade Federal do Pampa** e ao **Instituto Federal Farroupilha**, por todo o suporte e incentivo para que eu chegasse até aqui.

Enfim, quero demonstrar o meu agradecimento, a todos aqueles que, de um modo ou de outro, tornaram possível a realização da presente dissertação.

A todos o meu sincero e profundo **Muito Obrigado!**

“As dificuldades são apenas coisas a serem superadas.”

Sir Ernest Shackleton

RESUMO

A telemetria de rede *in band* (INT) é um paradigma emergente de monitoramento de rede. Ao coletar itens de telemetria de baixo nível em tempo próximo do real, a INT pode aumentar substancialmente a visibilidade de toda a rede - permitindo, por exemplo, a identificação de comportamentos de rede de curta duração, como *micro-burst* e outras anomalias. Estudos recentes tem focado em (i) desenvolver mecanismos INT para aumentar a visibilidade em toda a rede; e (ii) projetar novas soluções de monitoramento. No entanto, pouco ainda foi feito para coletar eficientemente dados de telemetria da rede neste paradigma. Isso é particularmente desafiador, pois dependendo de quais itens de telemetria de rede são coletados, isso pode degradar a visibilidade de toda a rede em termos de consistência/*freshness*. Neste Trabalho propomos o *Dynamic Probing Planning for In-band Network Telemetry* (DyPro), como uma solução heurística capaz de definir ciclos de sondagem otimizados para coletar dados de telemetria *in-band* em infraestruturas de rede programáveis. Nosso modelo, diferentemente de outros esforços de pesquisa, visa satisfazer dinamicamente os requisitos das aplicações de monitoramento em termos de dados INT. Os resultados mostram que o DyPro pode superar as soluções de ponta em até 5x no que diz respeito à porcentagem de aplicações de monitoramento satisfeitas.

Palavras-chave: Monitoramento de Rede. Programabilidade de Rede. Telemetria de Rede *In-band*. Orquestração de Telemetria.

ABSTRACT

In-band Network Telemetry (INT) is an emerging network monitoring paradigm. By collecting low-level telemetry items in near-real-time, INT can substantially increase visibility across the entire network - allowing, for example, the identification of short-lived network behaviors such as micro-burst and other anomalies. Recent studies have focused on (i) developing INT mechanisms to increase network-wide visibility; and (ii) designing new monitoring solutions. However, little has been done to efficiently collect telemetry data from the network in this paradigm. This is particularly challenging as depending on which items of network telemetry are collected, this can degrade the visibility of the entire network in terms of consistency/freshness. In this Work we propose the Dynamic Probing Planning for In-band Network Telemetry (DyPro), as a heuristic solution capable of defining optimized polling cycles to collect in-band telemetry data in programmable network infrastructures. Our model, unlike other research efforts, aims to dynamically satisfy the requirements of monitoring applications in terms of INT data. Results show that DyPro can outperform high-end solutions by up to 5x in terms of percentage of satisfied monitoring applications.

Key-words: Network Monitoring. Network Programmability. In-band Network Telemetry. Telemetry Orchestration.

LISTA DE FIGURAS

Figura 1 – Arquitetura de rede SDN.	30
Figura 2 – Programando um dispositivo em P4.	32
Figura 3 – Telemetria de Rede <i>In-band</i> (INT).	34
Figura 4 – Exemplo de uma solução para o problema de planejamento de sondagem, ilustrando um instantâneo onde pacotes de sondagem (f_1, f_2, f_3) coletam dados de telemetria a partir de dispositivos de rede selecionados.	45
Figura 5 – Número de ciclos de sondagem para um aumento da capacidade disponível da sonda.	52
Figura 6 – Uso da capacidade da sonda.	53
Figura 7 – Número de ciclos de sondagem por link de rede (ou seja, link overhead).	54
Figura 8 – Número de aplicações de monitoramento satisfeitas.	54
Figura 9 – Comparação entre nossas abordagens.	60

LISTA DE TABELAS

Tabela 1 – Principais características analisadas nos trabalhos relacionados	43
Tabela 2 – Resumo dos resultados para cenários dinâmicos.	55
Tabela 3 – Análise de Sensibilidade - Parâmetros α e β com ρ em 0.5.	60
Tabela 4 – Resumo dos resultados para cenários dinâmicos (DyPro vs DyProACO).	62

LISTA DE SIGLAS

ACO *Ant Colony Optimization*

API *Application Programming Interface*

CVRP *Capacitated Vehicle Routing Problem*

DCN *Data Center Network*

DDoS *Distributed Denial of Service*

DFS *Depth-First Search*

DyPro *Dynamic Probing Planning for In-band Network Telemetry*

DyProACO *Dynamic Probing Planning for In-band Network Telemetry powered by Optimizing Ant Colony*

ER *Edge Randomization*

FPGA *Field-programmable Gate Array*

INT *In-band Network Telemetry*

INTO *In-band Network Telemetry Orchestration*

IoT *Internet of Things*

IPFIX *IP Flow Information eXport*

MILP *Mixed Integer Linear Programming*

MTU *Maximum Transmission Unit*

NFV *Network Function Virtualization*

NIC *Network Interface Card*

NOS *Networking Operating System*

ONF *Open Networking Foundation*

ONOS *Open Network Operating System*

OPP *Optimal Path Planning*

P4 *Programming Protocol-independent Packet Processors*

P²INT *Probe Planning for In-Band Network Telemetry*

PLI *Programação Linear Inteira*

QoE *Quality of Experience*

SARG *Simulated Annealing based Random Greedy*

SDN *Software defined networking*

SNMP *Simple Network Management Protocol*

SR *Segment Routing*

LISTA DE SÍMBOLOS

D Conjunto de dispositivos de encaminhamento

$G = (D, L)$ Infraestrutura de rede

L Conjunto de dispositivos interconectados

M Conjunto de aplicações de monitoramento

P Conjunto de ciclos de sondagem

$R_m^s \subseteq \mathcal{P}(R_m)$ Conjunto de dependências espaciais das aplicações de monitoramento $m \in M$

$S : V \rightarrow \mathbb{N}^+$ Tamanho dos dados de telemetria $v \in V$

$U : P \rightarrow \mathbb{N}^+$ Tamanho de um pacote pertencente a um pacote de sondagem P

V Conjunto de itens de telemetria disponíveis

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Organização	28
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	29
2.1	Fundamentação Teórica	29
2.1.1	Programabilidade em Infraestruturas de Rede	29
2.1.2	Monitoramento de Infraestruturas	32
2.1.2.1	Monitoramento ativo e passivo	32
2.1.2.2	Monitoramento baseado em telemetria de rede in-band	33
2.1.3	Heurísticas e Meta-heurísticas	35
2.1.3.1	Otimização por Colônia de Formigas	36
2.2	Trabalhos Relacionados	38
3	PLANEJAMENTO DE SONDAÇÃO DINÂMICA PARA A TELEMETRIA DE REDE IN-BAND	45
3.1	Visão Geral do Problema	45
3.2	Descrição e Notação do Modelo	46
3.3	Heurística Proposta	49
3.4	Avaliação	51
3.4.1	Carga de Trabalho	51
3.4.2	Baseline	51
3.4.3	Resultados	52
4	ALGORITMO META-HEURÍSTICO ACO	57
4.1	Construção das Soluções	57
4.2	Feromônios	58
4.3	Melhor Solução	59
4.4	Avaliação	59
4.4.1	Carga de Trabalho	59
4.4.2	Sensibilidade dos Parâmetros	59
4.4.3	Resultados	60
5	CONCLUSÃO	63
5.1	Publicações Científicas	63
	REFERÊNCIAS	65

APÊNDICES	69
APÊNDICE A – DYPRO: DYNAMIC PROBING PLANNING FOR IN-BAND NETWORK TELEMETRY	71
Índice	73

1 INTRODUÇÃO

Com a utilização cada vez mais frequente de tecnologias mediadas pela internet em nosso dia a dia, como por exemplo *e-commerce*, aplicações de mídias sociais, jogos digitais e aplicações impulsionadas pela Internet das Coisas (IoT), há um esforço constante por parte de operadores de redes para que tudo funcione corretamente (ZHU et al., 2015). Nesse sentido, o monitoramento da rede possibilita aos operadores controlar a qualidade e o congestionamento, detectar anomalias e planejar a capacidade dos *data centers* – apenas para mencionar alguns.

Através da utilização de protocolos tradicionais de monitoramento, como o *Simple Network Management Protocol* (SNMP), é possível a um operador de rede monitorar constantemente parâmetros como carga de processamento, utilização de memória e outros dados estatísticos do estado atual dos dispositivos de rede. Porém, apesar de fornecer dados sobre o estado de funcionamento dos serviços e equipamentos, protocolos como o SNMP apresentam limitada frequência e abrangência de coleta dos dados monitorados. Dessa forma, problemas e/ou anomalias são potencialmente identificadas de forma tardia.

Neste contexto, vem ganhando tração a Telemetria de Rede *in-band* (INT), como um mecanismo de monitoramento de rede destinado a melhorar o nível de visibilidade (ou seja, cobertura dos dispositivos da rede) e ajustar a granularidade desejada sobre a coleta dos dados (JEYAKUMAR et al., 2014; PAN et al., 2019). A INT consiste na coleta de estatísticas de monitoramento de rede de baixo nível diretamente do plano de dados, através do tráfego de produção da infraestrutura, onde as informações coletadas são transportadas em um pacote ao longo de seu caminho de roteamento e, em algum ponto da infraestrutura, são extraídas e transmitidas para uma aplicação de monitoramento para processamento e, eventual reação aos eventos observados. Este processo de coleta pode conter estados internos dos dispositivos de encaminhamento (por exemplo, níveis de ocupação da fila de ingresso/egresso, identificação do dispositivo) e métricas relacionadas ao desempenho da infraestrutura (por exemplo, tempo de processamento no plano de dados, tempo de propagação entre dispositivos). A Telemetria de Rede *in-band* só foi possível devido à crescente adoção de dispositivos de encaminhamento programáveis e linguagens específicas de domínio de alto nível, como P4 (BOSSHART et al., 2014), sendo utilizada na identificação de comportamentos de rede de curta duração (por exemplo, *micro-burst*) e anomalias de rede (por exemplo, roteamento/dispositivos mal configurados).

Porém, orquestrar a coleta dos dados de telemetria não é um problema trivial, pois dependendo de quais itens de telemetria são coletados, pode degradar a visibilidade da rede em termos de cobertura e atualização (MARQUES et al., 2019), ou ainda, dependendo de como os dados são coletados, pode afetar o desempenho das aplicações de monitoramento da rede (HOHEMBERGER et al., 2019). Observa-se nos últimos anos, um número crescente de esforços de pesquisa para a orquestração da coleta de dados INT em redes programáveis de maneira eficiente (CASTRO et al., 2021; CHEN et al., 2021;

BHAMARE et al., 2019; HOHEMBERGER et al., 2019; LIN et al., 2020; MARQUES et al., 2019; PAN et al., 2019; ZHANG; SU; TAN, 2021), porém pouco ainda tem sido feito para coletar eficientemente dados de telemetria de rede considerando os requisitos das aplicações de monitoramento.

Desta forma, este Trabalho tem por objetivo propor uma solução heurística capaz de definir ciclos de sondagem otimizados para coletar dados de telemetria *in-band* em infraestruturas de rede programáveis. Ao contrário das soluções existentes (por exemplo, (LIU et al., 2018; PAN et al., 2019; CASTRO et al., 2020; CASTRO et al., 2021)), nossa solução visa satisfazer dinamicamente os requisitos das aplicações de monitoramento em termos de dados INT.

1.1 Organização

Este trabalho está organizado da seguinte forma. No capítulo 2, apresentamos a fundamentação teórica sobre programabilidade e monitoramento de infraestruturas de redes, e uma visão geral da literatura relacionada. No capítulo 3, formalizamos o planejamento de sondagem dinâmica e apresentamos nossa abordagem heurística para resolvê-lo. No capítulo 4 implementamos uma meta-heurística afim de melhorar o processo de construção de nossa solução inicial. Por fim, no capítulo 5, conclui-se este trabalho.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Neste capítulo examinamos os principais conceitos, no escopo desta dissertação, referente à programabilidade de infraestruturas de rede, ao monitoramento baseado em telemetria *in-band* em planos de dados programáveis e às heurísticas e meta-heurísticas. Na sequência, revisamos os estudos mais proeminentes relacionados ao problema de orquestração da coleta de dados de telemetria *in-band*.

2.1 Fundamentação Teórica

2.1.1 Programabilidade em Infraestruturas de Rede

A Internet possui um papel fundamental em nossa sociedade, provendo meios de comunicação em diversas áreas do conhecimento humano. Atualmente uma parcela significativa da população mundial tem acesso à internet, e diversas iniciativas de inclusão digital vem se desenvolvendo em todo o mundo para expandir seu alcance. Mas apesar da crescente expansão em sua utilização, vemos um cenário bastante complexo ao observarmos a infraestrutura necessária para que tudo funcione corretamente, com uma ampla gama de dispositivos intermediários (por exemplo, roteadores, *switches* e *firewalls*) executando softwares de controle complexos, com código proprietário, tornando-a pouco flexível e dificultando inovações no núcleo das infraestruturas. Para mitigar este problema, surge a programabilidade de redes como uma alternativa, através do paradigma das Redes Definidas por *Software* ou *Software Defined Networking* (SDN).

Rede Definida por *Software* (SDN), é um paradigma emergente que nos permite contornar o problema das limitações das infraestruturas de rede atuais. Por meio deste paradigma podemos separar o plano de controle (que determina a forma de encaminhamento de cada fluxo) do plano de dados (que encaminha o tráfego conforme as decisões do plano de controle) (FEAMSTER; REXFORD; ZEGURA, 2014).

Segundo Kreutz et al. (2015), o paradigma SDN baseia-se em quatro pilares: (i) desacoplar as funções de rede, separando os planos de controle e de dados; (ii) tomar decisões de encaminhamento de pacotes com base no fluxo ao invés do destino; (iii) atribuir o controle a uma entidade externa logicamente centralizada na rede, com o objetivo de prover a abstração da rede (controlador SDN ou NOS); (iv) programabilidade da rede por meio do controlador e suas aplicações. Nesse sentido, SDN fornecem gerenciamento centralizado, flexibilidade e programabilidade.

Kreutz et al. (2015) discutem ainda as vantagens desta tecnologia, como a visibilidade expandida do estado da rede através da implementação de controladores, possibilitando a identificação mais apurada de eventuais problemas na rede; programabilidade da rede; comutação inteligente de pacotes controlados por software, independente do hardware; dentre outras.

Para desacoplar o plano de controle do plano de dados, é necessário que haja uma

interface de programação de aplicação (API) bem definida para que o plano de controle possa instruir o plano de dados na maneira como ele deverá dar encaminhamento aos pacotes. Para o propósito do SDN, esta interface deve ser capaz de abstrair a implementação do plano de dados de um dispositivo de encaminhamento (como por exemplo, um *switch*) independente do fabricante.

Atualmente é possível escolher não somente o sistema operacional de rede ou controlador SDN, como o *Open Network Operating System* (ONOS), RYU e NOX/POX, mas também o protocolo que será utilizado na comunicação entre o controlador e os dispositivos do plano de dados (HOHEMBERGER, 2022). Uma API bastante proeminente é o *OpenFlow* (FEAMSTER; REXFORD; ZEGURA, 2014), uma interface aberta que permite um controle mais flexível da rede.

Nos últimos anos SDN tem ganhado cada vez mais atenção da indústria de dispositivos de encaminhamento, implementando o suporte ao OpenFlow. Grandes empresas no ramo da tecnologia da informação e telecomunicações no mundo, como por exemplo operadoras, fabricantes de equipamentos e empresas de serviços financeiros, tem apoiado a indústria SDN como a *Open Networking Foundation* (ONF) e a iniciativa *OpenDayLight*.

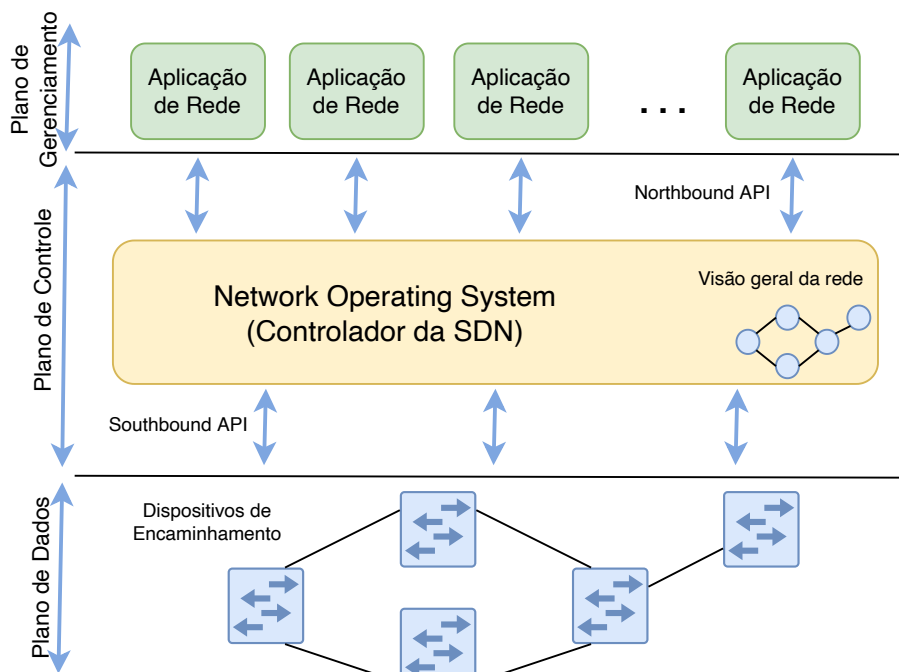


Figura 1 – Arquitetura de rede SDN.

A Figura 1 ilustra a arquitetura SDN, sendo dividida principalmente em plano de controle e plano de dados. Na parte superior, o plano de controle faz uso de uma interface *Northbound* para interagir com as aplicações, estando logicamente centralizado na infraestrutura permitindo que um operador possa instruir como os dados da rede devem ser gerenciados. O plano de controle é composto pelas aplicações que utilizam os

serviços da infraestrutura SDN e de um sistema operacional de rede (como por exemplo, POX ou RYU) que recebe as solicitações das aplicações do plano de controle e traduz em diretivas *OpenFlow*. Na parte inferior da figura, o plano de dados faz uso de uma interface *Southbound* para se comunicar com o plano de controle, e é composto pelos dispositivos de encaminhamento como *switches*, roteadores e outros, com a finalidade de processar e encaminhar os pacotes de dados. Entre os dois planos, o *OpenFlow*, considerado o primeiro padrão do SDN, permanece como a interface *Southbound* mais comum (FEAMSTER; REXFORD; ZEGURA, 2014), repassando ao plano de dados as regras comunicadas pelo plano de controle, possibilitando que um dispositivo da rede com suporte ao *OpenFlow* possa se comportar como um roteador, um *switch* ou um *firewall* por exemplo.

O *OpenFlow* em sua versão atual, suporta 42 campos de cabeçalhos de protocolo, ou seja, ações do *OpenFlow* que podem ser executadas em pacotes, tendo um crescimento bastante significativo se comparado aos 12 campos inicialmente suportados. Porém, ele ainda não oferece flexibilidade suficiente para se adicionar novos cabeçalhos e definir novas ações (FERNANDO et al., 2018).

Nesse sentido, a linguagem P4 (acrônimo para *Programming Protocol-independent Packet Processors*) (BOSSHART et al., 2014) possibilita uma maior flexibilidade e controle a um programador, ao permitir definir novos cabeçalhos e ações, diferentemente do *OpenFlow*, aumentando a programabilidade das Redes Definidas por Software. P4 descreve como um pacote deve ser processado pelo plano de dados de um dispositivo programável de encaminhamento, que pode ser implementado em *switches* via *hardware* ou *software*, placas de rede, roteadores ou dispositivos programáveis do tipo *Field Programmable Gate Array* (FPGA). Inicialmente foi projetada para atuar em *switches* programáveis, no entanto, seu escopo acabou sendo estendido para cobrir uma ampla variedade de dispositivos.

Segundo Bosshart et al. (2014), com a proposta do P4, busca-se a evolução do modelo por meio de três objetivos principais: (i) Reconfigurabilidade no campo: os programadores devem ser capazes de mudar a forma como os switches processam os pacotes depois de implementados; (ii) Independência do protocolo: os comutadores não devem estar vinculados a nenhum protocolo de rede específico; (iii) Independência do alvo: os programadores devem ser capazes de descrever a funcionalidade de processamento de pacotes independentemente das especificidades do hardware subjacente;

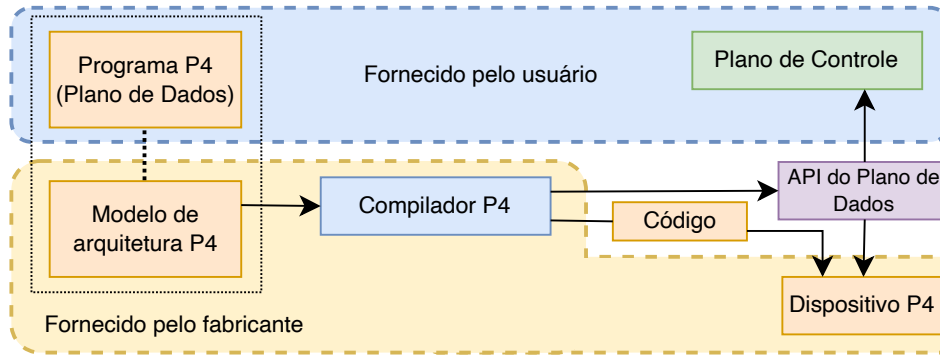


Figura 2 – Programando um dispositivo em P4.

A Figura 2 ilustra resumidamente o processo necessário para programar um dispositivo usando P4. A partir de um modelo de arquitetura e um compilador P4, programadores escrevem código P4 para uma arquitetura específica, que define um conjunto de componentes programáveis no dispositivo, bem como suas interfaces externas do plano de dados. A compilação de um conjunto de programas P4 produz dois artefatos: (i) uma configuração do plano de dados que implementa a lógica de encaminhamento descrita no programa de entrada e (ii) uma API para gerenciar o estado dos elementos do plano de dados à partir do plano de controle (The P4 Language Consortium, 2018). De posse destes artefatos um programador pode, por exemplo, programar o *switch* para ser um *switch top-of-rack*, um *firewall*, ou um *load-balancer*, podendo ainda adicionar recursos para executar diagnósticos automáticos e novos algoritmos de controle de congestionamento (KIM et al., 2015).

2.1.2 Monitoramento de Infraestruturas

O monitoramento de infraestruturas de rede é essencial para que os operadores possam realizar o gerenciamento de maneira oportuna. Obter dados precisos e atualizados sobre o estado da rede tornam possíveis as tarefas de monitoramento, como engenharia de tráfego e detecção de intrusão. Ao mesmo tempo, ferramentas de monitoramento precisam gerar dados sobre a rede próximo do tempo real sem prejudicar seu funcionamento.

2.1.2.1 Monitoramento ativo e passivo

Atualmente existem diversas soluções de monitoramento disponíveis a operadores de redes, com funções diversas como ajustar as cargas dos segmentos, identificar perdas de transmissão e verificar parâmetros de recursos, como consumo de energia. Outras funções importantes fornecidas pelas ferramentas de monitoramento de rede incluem indicadores de tempo de atividade/inatividade que enviam alertas por *e-mail* e texto, além da capacidade de personalizar os limites de desempenho nos quais os alertas são gerados. Integração SNMP, descoberta automática de topologia e mapeamento de rede são outras funções comuns.

Dois dos métodos mais comuns no monitoramento de redes são o monitoramento ativo e passivo. Algumas soluções se especializam em um ou outro, enquanto alguns fornecedores fornecem monitores com a capacidade de fazer as duas coisas.

No monitoramento ativo são inseridos pacotes especializados, chamados de *probes* ou sondas, na rede de forma a percorrer um determinado caminho com a finalidade de extrair métricas de desempenho para deduzir o estado geral da rede. Como o tráfego de teste imita o fluxo do tráfego de produção, o monitoramento ativo é ideal para fornecer uma visão *online* do desempenho de ponta a ponta de um serviço em relação a latência (ou atraso), *jitter* (ou variação de atraso) ou perda de pacotes. *Ping* e *Traceroute* são exemplos de soluções que se utilizam deste método. Porém no método ativo temos pelo menos dois desafios: (i) o caminho da sonda pode não ser exatamente o mesmo que o fluxo de tráfego que se deseja analisar, desse modo os resultados podem não refletir o esperado; (ii) adequar a quantidade de sondas necessárias para obter os resultados esperados sem que se interfira no desempenho ou na alocação excessiva de recursos.

O monitoramento passivo envolve a captura e análise do tráfego da rede, ou estatísticas de tráfego, em um ponto específico da rede, sem no entanto interferir no fluxo de pacotes ou na sua performance. O monitoramento passivo pode ser nada mais do que a coleta periódica de estatísticas de porta, como números de transmissão e recepção de *bytes* e pacotes. Mais tipicamente, também envolve a captura de parte ou todo o tráfego que flui através de uma porta para análise detalhada, *offline*, de dados como protocolos de sinalização, uso de aplicativos ou principais consumidores de largura de banda. Este método é ideal para análise detalhada de tráfego e protocolo, especialmente em um cenário "pós-evento", como análise de causa. Também é adequado para obter visibilidade em profundidade da qualidade da experiência dos clientes (QoE). *NetFlow*, *sFlow* e *IPFIX* se utilizam deste método. Porém como uma quantidade significativa de dados para análise são coletados, o desafio está em conseguir restringir a coleta sem sacrificar a precisão do monitoramento.

Devido à sua simplicidade de implantação, estes métodos de monitoramento de redes são amplamente utilizados no campo da gestão de redes, porém carecem de uma solução mais completa para as redes complexas atuais.

2.1.2.2 Monitoramento baseado em telemetria de rede in-band

O monitoramento de rede de forma coordenada e eficiente é imprescindível para garantir a confiabilidade da comunicação. Porém os métodos tradicionais de monitoramento são pouco eficazes, principalmente no que diz respeito à identificação de comportamentos de rede de curta duração. Em razão disso, a Telemetria de Rede *In-band* (INT) tem ganhado cada vez mais espaço como uma abordagem avançada de monitoramento de rede, capaz de suprir lacunas dos métodos tradicionais.

Desenvolvida à partir dos recentes avanços em hardware de dispositivos progra-

máveis (CHOLE et al., 2017) e das linguagens em alto nível para a descrição, consulta e gerenciamento desses dispositivos (por exemplo, a linguagem P4 (BOSSHART et al., 2014)) a telemetria de rede *in-band* possibilita melhorar a visibilidade de eventos de rede com granularidade fina (JEYAKUMAR et al., 2014; PAN et al., 2019). Exemplos de casos de uso bem sucedidos baseados na INT incluem a identificação de comportamentos de rede de curta duração (por exemplo, *micro-burst*) e anomalias de rede (por exemplo, roteamento/dispositivos mal configurados).

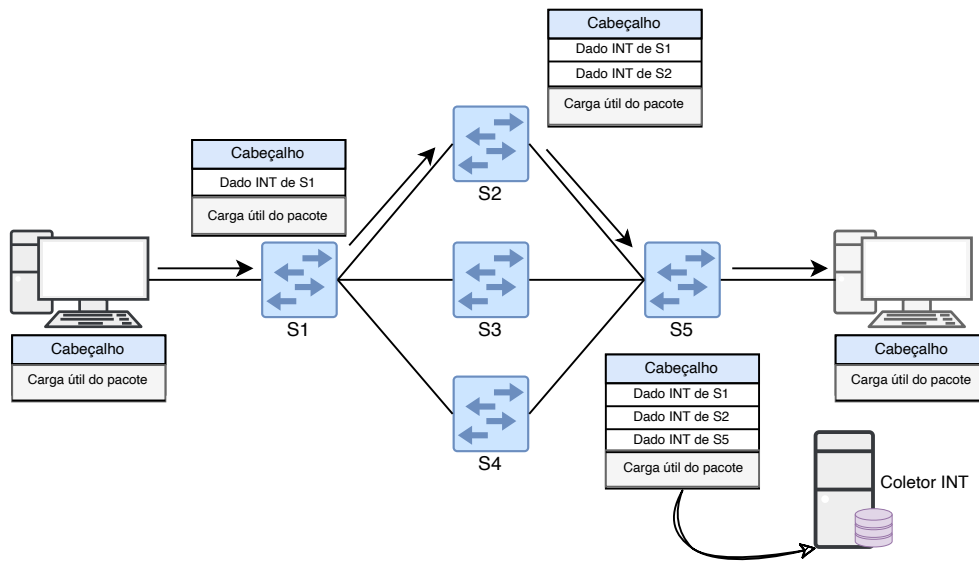


Figura 3 – Telemetria de Rede *In-band* (INT).

A INT consiste na coleta de estatísticas de monitoramento de rede de baixo nível diretamente do plano de dados, através do tráfego de produção da infraestrutura. A Figura 3 ilustra este processo, onde um pacote percorre um caminho do *switch* S1 até o *switch* S5 coletando informações de telemetria ao longo de seu caminho de roteamento, até que em um determinado ponto da infraestrutura essas informações são extraídas e transmitidas para uma aplicação de monitoramento para processamento e, eventual reação aos eventos observados. Este processo de coleta pode conter estados internos dos dispositivos de encaminhamento (por exemplo, níveis de ocupação da fila de ingresso/egresso, identificação do dispositivo) e métricas relacionadas ao desempenho da infraestrutura (por exemplo, tempo de processamento no plano de dados, tempo de propagação entre dispositivos). É importante salientar que para este processo, é possível utilizar além do tráfego de produção, pacotes especializados para realizar a coleta das informações de telemetria.

Desse modo, através do monitoramento utilizando a telemetria de rede *in-band*, um operador consegue obter uma visibilidade melhorada de toda a rede, através da coleta otimizada de informações de telemetria, possibilitando a detecção e reação imediata a eventos de curta duração (por exemplo, *micro-burst*) ou outros eventos anômalos, que eram até então inatingíveis com o monitoramento tradicional.

2.1.3 Heurísticas e Meta-heurísticas

O termo “heurístico” está relacionado com a resolução de problemas reais dispondo somente dos conhecimentos disponíveis. Esse termo provém de uma palavra grega com um significado próximo ao conceito de encontrar, sendo supostamente vinculada à expressão eureka de Arquimedes ao descobrir seu famoso princípio (MELIÁN et al., 2003).

Soluções heurísticas não garantem a otimalidade do problema ao qual elas são aplicadas, porém podem fornecer uma solução que se encontra, pelo menos, próxima da ótima (COLIN, 2007; HILLIER; LIERBERMAN, 2013). Problemas de otimização combinatória de grande porte normalmente são difíceis de se encontrar uma solução, desta forma os métodos heurísticos podem ser utilizados, pois proporcionam uma boa relação entre os resultados gerados e o tempo computacional necessário para gerar esses resultados.

Já as meta-heurísticas, são estratégias inteligentes para projetar ou melhorar procedimentos heurísticos muito generalistas. Por serem mais genéricas, elas utilizam a busca por uma solução dentro de seu espaço de pesquisa e também ao redor de sua vizinhança. É importante salientar que, devido ao conhecimento histórico dos resultados anteriormente adquiridos, as meta-heurísticas podem usar estratégias para escapar de paradas prematuras dentro de ótimos locais, explorando assim partes importantes do espaço de busca (BIRATTARI et al., 2001).

Segundo Melián et al. (2003), as meta-heurísticas podem ser classificadas em:

1. **Meta-heurísticas de relaxação:** são procedimentos de resolução de problemas que utilizam flexibilizações do modelo original (ou seja, modelos com modificações que tornam o problema mais fácil de resolver), cuja solução fornece a solução para o problema original;
2. **Meta-heurísticas construtivas:** baseiam-se em procedimentos que tratam da obtenção de uma solução a partir da análise e seleção paulatina dos componentes que a formam;
3. **Meta-heurísticas de busca:** denominam-se dessa forma qualquer método que percorra espaços de busca, compostos por soluções, levando em conta fundamentalmente, em cada passo, a vizinhança da solução obtida na iteração anterior;
4. **Meta-heurísticas evolutivas:** enfocam os métodos baseados em conjuntos de soluções que evoluem sobre o espaço de soluções;
5. **Meta-heurísticas híbridas:** são meta-heurísticas intermediárias em relação aos quatro tipos anteriores ou combinação de métodos.

Além disso, por compartilharem de uma série de características comuns, podemos dividir ainda as meta-heurísticas quanto à forma de exploração do espaço de busca por soluções, em duas categorias: busca local e busca populacional (SUCUPIRA, 2007):

1. A busca local é realizada por meio de deslocamentos aplicados sobre a solução atual, buscando uma solução de melhor qualidade em torno de sua vizinhança. Podemos citar como exemplos de meta-heurísticas de busca local: busca tabu, simulated annealing, método de pesquisa em vizinhança variável e busca reativa.
2. As buscas populacionais reúnem um conjunto de soluções de boa qualidade e as combinam de diferentes maneiras. O objetivo é extrair bons atributos de cada solução e, com essas informações, buscar uma solução ainda melhor. Podemos citar como exemplos de meta-heurísticas de busca populacional: algoritmos genéticos, colônia de formigas, nuvem de partículas e algoritmos meméticos.

2.1.3.1 Otimização por Colônia de Formigas

A meta-heurística construtiva *Ant Colony Optimization* (ACO), ou otimização por colônia de formigas, proposta por Dorigo e Stützle (2004), é conhecida por solucionar problemas relacionados a caminhos, inspirada na capacidade das formigas encontrarem o menor caminho entre uma fonte de alimento e sua colônia.

Em uma colônia de formigas, o principal aspecto para se entender é a liberação e captação de feromônio. Esse hormônio é liberado pelas formigas que saem do formigueiro à procura de comida nos caminhos em que passam, atraindo umas às outras e, consequentemente, fortalecendo rotas e criando caminhos mais eficazes entre o formigueiro e a fonte de comida. Entretanto, na natureza, estes feromônios deixados pelas formigas em suas rotas evaporam com o tempo, ou seja, na medida que uma determinada rota deixa de ser trafegada, este caminho deixa de atrair outras formigas (AKHTAR, 2019; ZHOU, 2009).

O uso do feromônio favorece caminhos mais curtos, fazendo-os ter uma maior probabilidade de serem escolhidos em relação aos demais. Em contrapartida, quanto mais longo um caminho, mais a trilha de feromônio irá evaporar conforme as formigas o atravessam. O atributo da distância entre duas localidades tem relação inversa para a escolha da formiga. Já o atributo do feromônio tem relação direta com a escolha de um caminho pela formiga, fazendo com que algumas trilhas tenham maior concentração do feromônio e assim, maior probabilidade de ser escolhida (AKHTAR, 2019).

Dorigo e Stützle (2004) propõem um algoritmo de otimização baseado no comportamento de formigas, como descrito acima, possuindo pelo menos oito variações conhecidas, tendo sido a primeira versão denominada *Ant System*, utilizada também como base para as variações seguintes.

O algoritmo *Ant System* funciona da seguinte forma, dado um grafo qualquer onde se pretenda obter os caminhos mais curtos passando por cada um dos nós uma única vez,

iniciamos colocando uma formiga em cada um dos n nós presentes do grafo, e definimos uma quantidade pequena de feromônio em cada um dos arcos. As formigas irão escolher, nó a nó, o caminho a ser tomado segundo uma regra de escolha de ação probabilística, chamada de regra de proporcionalidade aleatória, para decidir qual nó visitar em seguida.

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \quad \text{se } j \in N_i^k,$$

Onde N_i^k é a vizinhança viável da formiga k a partir do nó i . O valor τ_{ij} é a quantidade de feromônio depositada no arco que vai do nó i (atual) ao nó j , e $\eta_{ij} = 1/d_{ij}$ a informação heurística, que neste caso, leva em consideração a distância entre os nós (d_{ij}).

Os parâmetros α e β definem a influência tanto do feromônio quanto da informação heurística presentes. Se $\alpha = 0$, os nós mais próximos terão maior probabilidade de serem selecionados, correspondendo dessa forma a um algoritmo guloso clássico-estocástico (com vários pontos de partida, pois as formigas são inicialmente distribuídas aleatoriamente pelos nós). Se $\beta = 0$, apenas a amplificação do feromônio está em ação, ou seja, apenas o feromônio é usado, sem nenhum viés heurístico. Segundo Dorigo, isso geralmente leva a resultados ruins, e para valores de $\alpha > 1$ leva ao rápido surgimento de uma situação de estagnação, ou seja, uma situação em que todas as formigas seguem o mesmo caminho e constróem o mesmo passeio, o que, em geral, é fortemente abaixo do ideal (DORIGO, 1992; DORIGO; MANIEZZO; COLORNI, 1996).

Podemos notar que as formigas nunca escolhem um caminho de forma totalmente aleatória, no início da execução a regra de probabilidade dará maior relevância a caminhos mais curtos e ao fim a probabilidade penderá para caminhos com maiores acúmulos de feromônios, levando a uma estagnação, que é geralmente o critério de parada do algoritmo, ou uma quantidade limite de iterações.

Depois que todas as formigas construíram seus passeios, as trilhas de feromônios são atualizadas. Isso é feito primeiro diminuindo o valor do feromônio em todos os arcos por um fator constante e , em seguida, adicionando feromônio nos arcos que as formigas cruzaram em seus passeios. O cálculo de evaporação de feromônios é dado por:

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \quad \forall (i, j) \in L,$$

Sendo $0 < \rho \leq 1$ a taxa de evaporação de feromônios possibilitando que trajetos ruins possam ser esquecidos pelas formigas.

Após a evaporação, todas as formigas depositam feromônio nos arcos que cruzaram em seu passeio, assim, se múltiplas formigas andarem pelo mesmo arco, então mais feromônio será depositado nele:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad \forall (i, j) \in L,$$

Onde $\Delta\tau_{ij}^k$ é a quantidade de feromônio depositado pela formiga k nos arcos visitados. Definido da seguinte forma:

$$\Delta\tau_{ij} = \begin{cases} 1/C^k, & \text{se o arco } (i, j) \in T^k \\ 0, & \text{caso contrário;} \end{cases}$$

Onde C^k é o comprimento do passeio T^k construído pela formiga k , calculado como a soma dos comprimentos dos arcos pertencentes a T^k . A quantidade de feromônio depositado leva em consideração a distância total percorrida pela formiga, de forma que arcos utilizados para obter caminhos menores sejam mais atraentes.

O cálculo de probabilidades e depósito de feromônios são as características principais do algoritmo *Ant System* para resolver problemas.

2.2 Trabalhos Relacionados

Nesta seção, apresentamos uma breve revisão dos estudos mais proeminentes, relacionado ao monitoramento de infraestruturas de rede utilizando a INT.

Marques et al. (2019) com o trabalho descrito como *In-band Network Telemetry Orchestration* (INTO), apresentam os primeiros passos para a orquestração da telemetria de rede *in-band*. São propostas duas heurísticas para a orquestração de telemetria no plano de dados, a saber Concentrado e Balanceado, que atribuem fluxos de rede a dispositivos de encaminhamento, os quais irão coletar informações de telemetria destes dispositivos. Enquanto a estratégia Concentrado irá agregar os dados de telemetria em um número restrito de fluxos, a estratégia Balanceado fará a distribuição igualmente dos dados de telemetria pelos fluxos de rede disponíveis. Os resultados deste estudo demonstram que as heurísticas produzem soluções próximas ao ótimo para qualquer rede em uma escala de tempo aceitável, em relação à estratégia clássica com redução considerável da degradação do tráfego de rede e economia de recursos alocados nos *switches*.

Pan et al. (2019) apresentam o INT-path, um *framework* que utiliza a técnica dividir para conquistar, desacoplando a solução em um mecanismo de roteamento e uma política de geração de caminhos. Baseado no mecanismo de roteamento, projetaram duas políticas de planejamento de caminhos para gerar múltiplos caminhos INT não sobrepostos que cobrem toda a rede. A primeira é baseada em *Depth-First Search* (DFS), que é simples, mas eficiente em termos de tempo. A segunda é um algoritmo baseado nos circuitos de Euler que pode gerar de forma ótima caminhos INT não sobrepostos com um número mínimo de caminhos. Uma característica do INT-path é que ele codifica o estado do tráfego em toda a rede em uma série de "imagens *bitmap*", o que permite utilizar téc-

nicas avançadas, como o reconhecimento de padrões, para automatizar o monitoramento e a solução de problemas da rede.

Hohemberger et al. (2019) fizeram a primeira tentativa da coleta de itens de telemetria de forma coordenada em tempo real. Eles projetaram um modelo baseado em aprendizado de máquina e formalizaram o problema da coleta para satisfazer os requisitos espaciais e temporais, ou seja, consideram que os itens devem ser coletados de dispositivos específicos e em uma determinada taxa, respectivamente, para alimentar adequadamente os aplicativos de aprendizado de máquina com o objetivo de detectar anomalias (por exemplo, *Distributed Denial of Service* (DDoS)).

Geng et al. (2019) propõem o SIMON, um sistema de medição preciso e escalável para redes DCN que reconstrói as principais variáveis de estado da rede, como tempos de enfileiramento de pacotes em *switches*, utilizações de *links* e composições de filas e *links* a nível de fluxo, coletando dados inteiramente nas placas de interface de rede (NIC) de transmissão e recepção. Uma malha de sondas reconstrói redes DCN cobrindo os caminhos por pacote ou por fluxo. A reconstrução é baseada em tomografia de rede que utiliza um algoritmo de inferência LASSO (TIBSHIRANI, 1996) para alimentar oportunamente uma rede neural multicamadas fornecendo alta precisão para detectar diferentes problemas. No entanto, SIMON limita-se a operar apenas em redes DCN onde tem pleno conhecimento da topologia.

Bhamare et al. (2019) introduziram o IntOpt, uma estrutura híbrida de monitoramento *in-band* e sondagem ativa para monitoramento da virtualização da função de rede (NFV). Através de uma metaheurística *Simulated Annealing based Random Greedy* (SARG), tem por objetivo minimizar o número de caminhos necessários para cobrir todas as tarefas de telemetria, com isso diminuindo a sobrecarga da rede. Basicamente o algoritmo funciona assim, primeiro o SARG inicializa um fluxo de dados de telemetria e atribui a ele qualquer cadeia aleatória. Em seguida, um novo elo aleatório é adicionado. Se o novo elo tiver requisitos de telemetria semelhantes ou menores do que o elo já atribuído ao fluxo de dados de telemetria dado, o fluxo aceita o novo elo. Finalmente, o SARG adiciona repetidamente um novo elo até que o fluxo de dados de telemetria exceda o limite. Segundo os autores, os resultados do estudo mostram que SARG pode reduzir a sobrecarga de monitorização e o atraso total da telemetria.

Castro et al. (2020) demonstram com a solução Patcher, os primeiros esforços para a coordenação dos ciclos de sondagem INT tolerante a falhas, reconstruindo e consertando ciclos de sondagem que tenham sido afetados por nós que apresentem falhas. Baseando-se no uso de *probes*, em caso de falha em um dispositivo da rede, a solução busca um rearranjo dos fluxos que realizam a coleta. Caso não seja possível o arranjo, é criado um novo fluxo utilizando-se o menor caminho possível. A solução proposta leva em consideração critérios de cobertura, capacidade de transporte dos fluxos, capacidade do ciclo e conectividade.

Basat et al. (2020) propuseram o PINT, uma estrutura de telemetria de rede

in-band que limita a quantidade de informações adicionadas a cada pacote. Em sua abordagem, o PINT considera que para a maioria das aplicações não são necessários todos os dados de telemetria que a INT coleta, sendo assim, ele espalha as informações sobre vários pacotes para minimizar a sobrecarga por pacote, considerando que a maioria das aplicações não requer informações de telemetria perfeitas e permite a codificação de dados de consulta solicitados por pacote tão baixo quanto um *bit* em vários pacotes. Um mecanismo de consulta decide então em um plano de execução, a probabilidade de executar cada conjunto de consulta em pacotes ao longo de seu caminho. No entanto, o PINT não cobre necessariamente toda a topologia. Além disso, ele fornece uma visão apenas agregada do estado da rede, por exemplo, fluxos de rastreamento precisam de significativamente menos pacotes, mas pequenos fluxos podem consistir em um único pacote, o que impede a divisão dos dados de telemetria da consulta.

Lin et al. (2020) com o Netview, fornecem uma nova estrutura de telemetria de rede para redes de *data centers*. NetView é um sistema de telemetria que suporta diferentes frequências de telemetria, monitorando cada dispositivo usando sondas ativas. Tecnicamente, o NetView divide cada sonda em uma pilha de encaminhamento e uma pilha de telemetria, que são respectivamente responsáveis pelo encaminhamento flexível e pelo monitoramento do estado da rede, para alcançar maior cobertura e visibilidade. Além disso, uma série de algoritmos de geração de sondas e algoritmos de atualização reduzem muito o número de sondas, proporcionando uma elevada escalabilidade.

Song et al. (2020) com o INT-filter, buscam diminuir o *overhead* de dados, gerados pelo processo de coleta das informações de telemetria, entre os planos de controle e de dados através do *southbound*. Para isso, foram propostos mecanismos de predição, que se baseiam em métodos de regressão linear e não linear, de acordo com o histórico da janela de dados. Sendo assim, INT-filter pretende reduzir a coleta de itens de telemetria desnecessários e contribuir para o decremento da quantidade de itens coletados.

Castro et al. (2021) apresentam o *Probe Planning for In-Band Network Telemetry* (P²INT), uma solução que utiliza uma abordagem heurística de Randomização de Aresta (ER) adaptada, para coordenar como os pacotes de sondagem são gerados e roteados, buscando visitar todos os *links* e coletar todos os dados INT. O problema foi formulado como um modelo de programação linear inteira, consistindo na generalização de dois problemas de otimização, sendo eles o Problema de Roteamento em Arcos Capacitados e o Problema de Empacotamento.

Chen et al. (2021) abordam o problema da orquestração com o emprego de *Segment Routing* (SR) e INT, já que ambos utilizam o cabeçalho do fluxo de produção para armazenamento temporário de dados. A partir deste estudo foi formulado um modelo baseado em programação linear inteira para tentar resolver este problema, com a proposta de um algoritmo recursivo baseado em caminhos ranqueados, capaz de resolver em tempo polinomial.

Zhang et al. (2021) propõem um detectorINT como um sistema automatizado e rápido de detecção de anomalias de rede, baseado em um algoritmo de Aprendizado Profundo. Primeiro é construído um protótipo de telemetria baseado na INT, permitindo um monitoramento de granulometria fina através da aquisição de estados de dispositivos *hop-by-hop*. Depois, utilizam um algoritmo *Generative Adversarial Active Learning* (GAAL) que pode conduzir a aprendizagem ativa e treinar geradores/discriminadores através do processo adversário, para detectar anomalias sem depender excessivamente da intervenção humana. Além disso, realizam o pré-processamento de dados com filtragem de baixa passagem para eliminar os efeitos transitórios do trânsito para a detecção de anomalias mais persistentes.

Simsek, Ergenc e Onur (2021) propõem o GPINT, uma INT heurística baseada no algoritmo de particionamento gráfico de Kernighan-Lin (KERNIGHAN; LIN, 1970), para encontrar caminhos equilibrados para encaminhar informações de telemetria *in-band* buscando satisfazer três requisitos: (i) minimizar a sobrecarga de controle, (ii) garantir o frescor das informações de telemetria, e (iii) minimizar a redundância. Os resultados do estudo foram favoráveis, se comparados ao método de Euler e ao método de monitoramento tradicional usando SNMP.

Pan et al. (2021), assim como no trabalho de (BASAT et al., 2020), propõem uma versão probabilística da INT, que pode fornecer visibilidade de rede similar à INT com um custo geral menor. Para isto, utilizam uma abordagem de sondagem ativa, que combina um *hashing* global e um algoritmo de amostragem aleatória com um reservatório, para amostrar uniformemente os dados de telemetria dos dispositivos de rede. Esta abordagem faz com que os pacotes carreguem os dados de telemetria dos dispositivos da rede no caminho com igual probabilidade.

Zhang, Su e Tan (2021) projetaram um algoritmo guloso para a construção da solução inicial, com o objetivo de orquestrar as tarefas de telemetria de rede *in-band*, e melhoraram esta solução com um algoritmo genético de classificação não dominado (NSGA-II), no qual a operação de mutação é melhorada de acordo com as características do problema INTO.

Podemos observar que diferentes esforços tem sido empregados para a realização da telemetria de rede, tornando-se possíveis alternativas ao monitoramento de rede tradicional. Entretanto isso só foi possível devido à crescente adoção de dispositivos de encaminhamento programáveis e de linguagens específicas de domínio de alto nível, como P4. E apesar da telemetria de rede contribuir significativamente para suprir as lacunas dos métodos tradicionais de monitoramento, ela não resolve trivialmente o problema.

Deste modo diversos estudos tem focado em diferentes aspectos da telemetria de rede, como podemos observar resumidamente na Tabela 1. Um exemplo, é a sobrecarga gerada na infraestrutura, uma desvantagem fundamental da INT (BASAT et al., 2020; BHAMARE et al., 2019; MARQUES et al., 2019; SIMSEK; ERGENC; ONUR, 2021;

SONG et al., 2020) pois a telemetria de rede *in-band* pode gerar uma grande quantidade de dados de telemetria, aumentando a carga de coleta, armazenamento e análise dos dados no servidor. Na INT, o comprimento total de dados de telemetria de um pacote que precisa ser transportado é proporcional ao número de pontos de telemetria. Assumindo que um *switch* irá adicionar dezenas de *bytes* de dados de telemetria a cada pacote, esses dados de rastreamento acumulados podem até mesmo exceder o tamanho do pacote original. A quantidade de dados de telemetria gerados pelo plano de dados está relacionada à quantidade de metadados de telemetria, tráfego, escala de rede, etc.

Desta forma alguns trabalhos desenvolveram mecanismos que podem auxiliar na coleta de dados, como mecanismos de predição (SONG et al., 2020), heurísticas especializadas (CHEN et al., 2021; MARQUES et al., 2019), otimização probabilística (BASAT et al., 2020; PAN et al., 2021), aprendizado de máquina (HOHEMBERGER et al., 2019; ZHANG et al., 2021), para citarmos alguns.

A orquestração da coleta de dados também é um dos pontos chave para diminuir a sobrecarga na infraestrutura, deste modo diversos esforços (CASTRO et al., 2021; CHEN et al., 2021; BHAMARE et al., 2019; HOHEMBERGER et al., 2019; LIN et al., 2020; MARQUES et al., 2019; PAN et al., 2019; ZHANG; SU; TAN, 2021) tem focado na realização da orquestração das tarefas de telemetria de redes *in-band* de forma eficiente.

Trabalhos como o de Castro et al. (2020) demonstram os primeiros esforços para a coordenação dos ciclos de sondagem INT tolerante a falhas, reconstruindo ciclos de sondagem que tenham sido afetados por nós defeituosos.

Podemos observar ainda, que os trabalhos elencados nesta seção concentram-se na realização da telemetria de rede através de pacotes de sondagens ativas baseadas na INT (CASTRO et al., 2020; CASTRO et al., 2021; BASAT et al., 2020; BHAMARE et al., 2019; GENG et al., 2019; LIN et al., 2020; PAN et al., 2019; SIMSEK; ERGENC; ONUR, 2021; SONG et al., 2020; PAN et al., 2021), na incorporação de dados INT em pacotes de rede de produção (CHEN et al., 2021; HOHEMBERGER et al., 2019; MARQUES et al., 2019) ou ainda alternando a abordagem quando necessário (ZHANG et al., 2021).

Tabela 1 – Principais características analisadas nos trabalhos relacionados

Trabalho	Sondagem Ativa	Fluxo de Produção	Característica
Marques et al. (2019)		X	Heurísticas especializadas
Hohemberger et al. (2019)		X	Aprendizado de máquina
Pan et al. (2019)	X		Imagens bitmap
Geng et al. (2019)	X		Rede neural
Bhamare et al. (2019)	X		Otimização da sondagem
Castro et al. (2020)	X		INT tolerante a falhas
Basat et al. (2020)	X		Otimização probabilística
Lin et al. (2020)	X		Otimização da sondagem
Song et al. (2020)	X		Mecanismo de previsão
Castro et al. (2021)	X		Otimização da sondagem
Chen et al. (2021)		X	Heurísticas especializadas
Zhang et al. (2021)	X	X	Detecção de anomalias
Simsek, Ergenc e Onur (2021)	X		Particionamento gráfico
Pan et al. (2021)	X		Otimização probabilística
Zhang, Su e Tan (2021)	X		Otimização da sondagem

Em nossa abordagem, propomos uma solução heurística e meta-heurística capaz de definir ciclos de sondagem ativa otimizados para coletar dados de telemetria de rede *in-band* em infraestruturas de rede programáveis. Diferente das soluções elencadas, nossa abordagem visa satisfazer dinamicamente os requisitos das aplicações de monitoramento em termos de dados INT.

Conclui-se que a telemetria de rede *in-band*, por ser uma abordagem recente, ainda possui diversos desafios a serem enfrentados, havendo muito espaço para pesquisas e inovação.

3 PLANEJAMENTO DE SONDAGEM DINÂMICA PARA A TELEMETRIA DE REDE IN-BAND

Neste capítulo apresentamos o problema de Planejamento de Sondagem Dinâmica para Telemetria de Rede *In-band* ou *Dynamic Probing Planning for In-band Network Telemetry* (DyPro). Primeiramente, apresentamos uma visão geral do problema. Em seguida modelamos e definimos formalmente o problema e suas restrições. Na sequência apresentamos nossa abordagem heurística e a avaliamos em relação a outras duas abordagens de orquestração de última geração.

3.1 Visão Geral do Problema

O problema DyPro consiste em definir ciclos de sondagem otimizados para coletar dados INT. Ao contrário das soluções existentes (por exemplo, (LIU et al., 2018; PAN et al., 2019; CASTRO et al., 2020; CASTRO et al., 2021)), DyPro visa satisfazer dinamicamente os requisitos das aplicações de monitoramento em termos de dados INT. Como podemos observar, esse problema de otimização está longe de ser resolvido de maneira trivial, uma vez que os pacotes de sondagem são limitados pelo espaço (por exemplo, de acordo com o (MTU) do link ou encapsulamento de dados) e, portanto, não é viável coletar todos os dados de telemetria exigidos pelas aplicações de monitoramento usando um único pacote de sondagem.

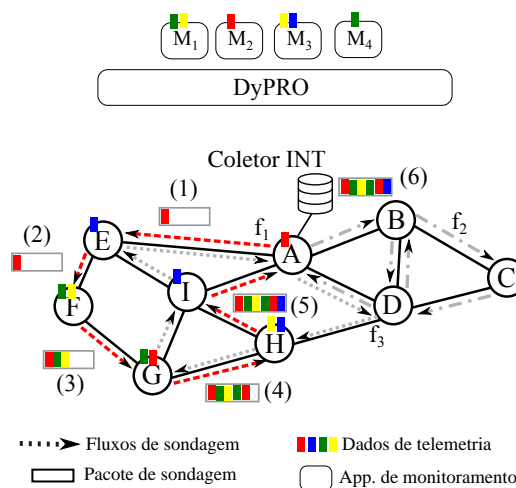


Figura 4 – Exemplo de uma solução para o problema de planejamento de sondagem, ilustrando um instantâneo onde pacotes de sondagem (f_1 , f_2 , f_3) coletam dados de telemetria a partir de dispositivos de rede selecionados.

Como exemplo, considere o cenário da Figura 4 que representa uma infraestrutura de rede com nove dispositivos programáveis (ou seja, $A-I$). Para simplificar, assumimos que (i) todos os itens de telemetria tem o mesmo tamanho e representam os estados internos do plano de dados (por exemplo, ocupação de fila), que serão utilizados posteriormente por aplicações de monitoramento especializadas (HOHEMBERGER et al.,

2019); e (ii) todos os pacotes de sondagem tem a mesma capacidade para transportar dados de telemetria (no exemplo, no máximo seis dados de telemetria). No topo da rede, existe um conjunto de aplicações de monitoramento especializadas $\{M_1, M_2, \dots, M_4\}$. Por exemplo, M_1 poderia prever e identificar o congestionamento da rede e, portanto, requer a coleta do número de fluxos de rede ativos, ocupação de filas e tempo de processamento diretamente do plano de dados.

Um conjunto de três ciclos de sondagem ativa (ou seja, f_1 , f_2 , e f_3) estão em vigor nesta infraestrutura, e são responsáveis pela contínua (i) coleta de dados de telemetria e (ii) verificação de conectividade da rede. Podemos observar que todos os links da rede são cobertos por pelo menos um ciclo de sondagem.

Existem algumas alternativas para orquestrar a aquisição de dados INT de acordo com as restrições acima mencionadas. Uma solução ingênua, consiste em coletar avidamente todos os dados INT da rede, levando em consideração a capacidade existente do ciclo de sondagem (por exemplo, (MARQUES et al., 2019)). Sempre que a capacidade de um ciclo estiver totalmente esgotada, o pacote é instruído a retornar ao coletor da INT. De fato, esta abordagem pode ser vista como uma generalização do problema conhecido como *Capacitated Vehicle Routing Problem* (CVRP) (RALPHS et al., 2003). Embora represente uma solução válida, dados INT não são coletados de acordo com os requisitos de monitoramento. Portanto, dois ou mais dados INT do mesmo dispositivo poderiam ser coletados em duas unidades de tempo diferentes, levando a uma fraca correlação entre eles.

A fim de resolver este problema, formalizamos um modelo de Programação Linear Inteira (PLI) que é capaz de coletar todos os dados INT requeridos pelas aplicações de monitoramento, satisfazendo ao mesmo tempo os requisitos existentes. Adotamos para isso, uma versão revisada e extendida do modelo proposto por (CASTRO et al., 2021).

3.2 Descrição e Notação do Modelo

Para o modelo de otimização proposto, consideramos como entrada uma infraestrutura física de rede $G = (D, L)$, um conjunto de itens de telemetria V , e um conjunto de aplicações de monitoramento M . O conjunto D na rede G representa os dispositivos de encaminhamento programáveis $D = \{1, \dots, |D|\}$, enquanto que o conjunto L representa os links que interligam pares de dispositivos $(d_1, d_2) \in (D \times D)$. Cada dispositivo de encaminhamento $d \in D$ pode incorporar um subconjunto de itens $V_d \subseteq V$ em um pacote de sondagem. Cada item de telemetria $v \in V$ tem o seu tamanho definido pela função $S : V \rightarrow \mathbb{N}^+$. Por sua vez, a aplicação de monitoramento $m \in M$ requer um subconjunto de itens de telemetria $R_m \subseteq V$ para operar.

Estes itens de telemetria podem ter dependências espaciais. Dizemos que dois ou mais itens de telemetria tem dependência espacial se tiverem de ser coletados juntos a partir do mesmo dispositivo de encaminhamento $d \in D$. Dependências espaciais

são representadas pelo conjunto de conjuntos $R_m^s \subseteq \mathcal{P}(R_m)$. Por exemplo, a aplicação de monitoramento M_1 poderia requisitar os subconjuntos $\{1, 3\}$ e $\{4, 6\}$, ou seja $R_1^s = \{\{1, 3\}, \{4, 6\}\}$. Em outras palavras, o modelo assegura que o subconjunto de dados INT $\{1, 3\}$ são coletados atômicamente por um determinado pacote de sondagem.

Para simplificar, consideramos que há no máximo $|P|$ ciclos de sondagem (isto é, $P = \{1, 2, \dots, |P|\}$) que são capazes de coletar itens de telemetria de dispositivos de encaminhamento D . Pacotes num ciclo de sondagem $p \in P$ tem espaço disponível limitado para incorporar itens de telemetria, definidos pela função $U : P \rightarrow \mathbb{N}^+$. Os ciclos de sondagem P são encaminhados dentro da infraestrutura de rede G , ou seja, o pacote é gerado à partir de um dispositivo de origem, roteado através de um subconjunto de dispositivos, e retorna à sua origem. Um ciclo $p \in P$ pode visitar um dispositivo de encaminhamento e não necessariamente coletar o conjunto de itens de telemetria associados. Denominamos a origem (dispositivo de começo/término) de cada ciclo $p \in P$ como um dispositivo de encaminhamento fixo $o \in D$ e, portanto, o nosso modelo está restrito a cenários de ciclos de sondagem de fonte única. Observe, contudo, que esta definição é facilmente estendida ao cenário de fonte múltipla.

O problema de otimização DyPro procura uma solução viável que satisfaça os requisitos das aplicações de monitoramento, ao mesmo tempo que minimiza o número de ciclos de sondagem existentes na infraestrutura de rede. A saída do modelo é dada por um conjunto de variáveis $\{Z, X, Y, S\}$. As variáveis de $Z = \{z_{p,v,i}, \forall p \in P, v \in V, i \in D\}$ indicam que um dispositivo de encaminhamento i incorpora um item de telemetria v dentro de um pacote de sondagem do ciclo p . As variáveis de $X = \{x_{p,i,j}, \forall p \in P, (i, j) \in L\}$ indicam que o link de rede $(i, j) \in L$ é utilizado para encaminhar o ciclo de sondagem $p \in P$. Por sua vez, a variável $Y = \{y_p, \forall p \in P\}$ é utilizada para acompanhar os ciclos de sondagem utilizados pela solução. Por último, as variáveis de $S = \{s_{m,d,P}^b, \forall m \in M, d \in D, P \in R_m^s\}$ são utilizadas para acompanhar as dependências espaciais satisfeitas pelo modelo. A seguir, descrevemos a formulação de programação linear inteira mista para o problema.

$$\text{Minimizar} \quad \sum_{p=1}^P y_p \quad (1)$$

Sujeito a:

$$z_{p,v,i} \leq \sum_{j \in D} x_{p,j,i} \quad \forall p \in P, i \in D, v \in V_i \quad (2)$$

$$z_{p,v,i} + x_{p,i,j} \leq 2 \cdot y_p \quad \forall p \in P, (i, j) \in L, v \in V_i \quad (3)$$

$$\sum_{j \in D} x_{p,i,j} - \sum_{j \in D} x_{p,j,i} = 0 \quad \forall p \in P, i \in D \quad (4)$$

$$\sum_{i \in S} \sum_{j \in S} x_{p,i,j} \leq |S| - 1 \quad \forall p \in P, S \subseteq \{D - o\}, |S| \geq 2 \quad (5)$$

$$\sum_{p \in P} x_{p,i,j} + x_{p,j,i} \geq 1 \quad \forall (i, j) \in L \quad (6)$$

$$\sum_{i \in D} \sum_{v \in V_i} z_{p,v,i} \cdot S(v) + \sum_{i \in D} \sum_{j \in D} x_{p,i,j} \leq U(p) \quad \forall p \in P \quad (7)$$

$$s_{m,i,p,V^*} = \sum_{v \in V^*} z_{p,v,i} \quad \forall m \in M, V^* \in R_m^s, i \in D, p \in P \quad (8)$$

$$\frac{s_{m,i,p,V^*}}{|V^*|} = 1 \quad \forall m \in M, v \in V^* \in R_m^s, i \in D \quad (9)$$

$$z_{p,v,i} \in \{0, 1\} \quad \forall p \in P, v \in V_i, i \in D \quad (10)$$

$$s_{m,d,p} \in \mathbb{N}^+ \quad \forall m \in M, d \in D, p \in P \in R_m^s \quad (11)$$

$$y_p \in \{0, 1\} \quad \forall p \in P \quad (12)$$

$$x_{p,i,j} \geq 0 \quad \forall p \in P, v \in V_i, i \in D \quad (13)$$

Os seguintes conjuntos de restrições definem uma solução viável para o problema DyPro. Primeiro, o conjunto de restrições (2) assegura que se o item de telemetria v for coletado do dispositivo de encaminhamento i , então deverá haver uma sonda sendo roteada através de i . O conjunto de restrições (3) contabiliza o número de ciclos de sondagem em uso. Ele define um ciclo p como ativo sempre que as variáveis $z_{p,v,i} = 1$ ou $x_{p,i,v} = 1$. Então, os conjuntos de restrições (4) e (5) asseguram que os ciclos sejam formados adequadamente. O conjunto de restrições (4) assegura a conservação do fluxo - ou seja, gera ciclos de sondagem sem ramificação ou autolops. Por sua vez, o conjunto de restrições (5) é a conhecida restrição de eliminação de sub-rota, assegurando que os ciclos gerados estejam fortemente conectados (DANTZIG; FULKERSON; JOHNSON, 1954). Então, o conjunto de restrições (6) assegura que um ciclo de sondagem cobre pelo menos uma direção do link. O conjunto de restrição (7) garante que a capacidade disponível não é violada nem pelos itens de telemetria coletados nem pelos links de rede que estão sendo cobertos. Os conjuntos de restrições (8) e (9) verificam se uma dependência espacial é ou não satisfeita. O conjunto de restrições (8) conta o número de itens de telemetria coletados

em um dispositivo d , enquanto o conjunto de restrições (9) verifica se a dependência é ou não satisfeita. Para simplificar, indexamos a dependência espacial como V^* no conjunto de variáveis s_m, d, V^* . Por último, os conjuntos de restrições (10)–(13) definem os domínios das variáveis de saída. Dado um espaço de solução viável definido pelas Equações (2)–(13), DyPro tem como objetivo minimizar o número de ciclos de sondagem usando a Equação (1).

Caso as aplicações de monitoramento M alterem os requisitos existentes (ou seja, o conjunto $R_m^s \subseteq \mathcal{P}(R_m)$), a nossa solução tenta manter o plano de orquestração existente tão semelhante como o original. Para isso, adicionamos as Equações (11)–(13) para calcular as diferenças entre os valores originais e os novos atribuídos às variáveis do modelo. As variáveis originais são representadas com o sobrescrito \bullet , enquanto a diferença entre elas e as novas é atribuída a uma variável representada pelo sobrescrito \oplus . Além disso, substituímos a função objetiva (1) pela Equação (14), pois o objetivo é minimizar as alterações na solução existente. Devido a limitações de espaço, omitimos como a subtração modular é feita no modelo de programação linear inteira mista.

$$z_{p,v,i}^{\oplus} = |z_{p,v,i} - z_{p,v,i}^{\bullet}| \quad \forall p \in P, v \in V_i, i \in D \quad (14)$$

$$x_{p,i,j}^{\oplus} = |z_{p,i,j} - z_{p,i,j}^{\bullet}| \quad \forall p \in P, (i,j) \in L \quad (15)$$

$$y_p^{\oplus} = |z_p - z_p^{\bullet}| \quad \forall p \in P \quad (16)$$

$$\text{Minimize} \quad \sum_{p \in P} \sum_{v \in V} \sum_{i \in D} z_{p,v,i}^{\oplus} + \sum_{p \in P} \sum_{(i,j) \in L} x_{p,i,j}^{\oplus} + \sum_{p \in P} y_p^{\oplus} \quad (17)$$

3.3 Heurística Proposta

Para enfrentar o problema acima de forma eficiente e fornecer uma solução de qualidade, propomos o DyPro (Algoritmo 1).

Ele maximiza o número de dependências espaciais R_m^{s-1} com um número mínimo de links e alterações na atribuição de itens a um conjunto de ciclos existente \mathcal{C}_ϕ^0 . Primeiro, uma solução viável (Algoritmo 2) ϕ^0 é processada para um conjunto inicial de aplicações de monitoramento M^0 , que gera um conjunto de ciclos \mathcal{C}_ϕ^0 (linha 1). Para um novo conjunto de dependências $\forall R_{mi}^{s-1} \in M^1$, ele faz o seguinte: (i) verifica se existem dependências de M^1 já satisfeitas na solução anterior ϕ^0 (linhas 2-8); (ii) remove dependências espaciais que não fazem parte do conjunto de dependências de M^1 (linhas 9-13); (iii) remove duplicatas - ou seja, dependências espaciais satisfeitas mais de uma vez por diferentes ciclos (linhas 14-20). Depois, para as dependências espaciais restantes, tenta encaixá-las nos ciclos

Algorithm 1 Visão geral do procedimento DyPro.

Input: Novo conjunto de aplicações de monitoramento M^1 e dependências espaciais $R_m^{s\ 1}$

- 1: $\phi^1(M^0, R_m^{s\ 0}) \leftarrow$ conjunto inicial viável de ciclos \mathcal{C}_ϕ^0 for $R_m^{s\ 0} \in M^0$
- 2: **for** $M_i^1 \in M^1$ **do**
- 3: **for** $R_j^{s\ 1} \in R_{mi}^{s\ 1}$ **do**
- 4: **if** $R_j^{s\ 1}$ já está satisfeito por um ciclo $c \in \mathcal{C}_\phi^0$ **then**
- 5: $R_{mi}^{s\ 1} \leftarrow R_{mi}^{s\ 1} - R_j^{s\ 1}$
- 6: **end if**
- 7: **end for**
- 8: **end for**
- 9: **for** $\forall c \in \mathcal{C}_\phi^0$ **do**
- 10: **if** c contém uma dependência espacial antiga – isto é, $R_j^{s\ 0} \notin R_i^{s\ 1}, \forall i \in M^1$ **then**
- 11: Remove os itens associados de c
- 12: **end if**
- 13: **end for**
- 14: **for** $M_i^1 \in M^1$ **do**
- 15: **for** $R_j^{s\ 1} \in R_{mi}^{s\ 1}$ **do**
- 16: **if** $R_j^{s\ 1}$ já foi satisfeita por mais de um ciclo $c \in \mathcal{C}_\phi^0$ **then**
- 17: $R_{mi}^{s\ 1} \leftarrow R_{mi}^{s\ 1} - R_j^{s\ 1}$
- 18: **end if**
- 19: **end for**
- 20: **end for**
- 21: **for** dependências espaciais não coletadas $\forall R_{mi}^{s\ 1} \in M^1$ **do**
- 22: **if** $c \in \mathcal{C}_\phi^0(p)$ tem espaço disponível U e c contém $d \in D$ associado com $R_{mi}^{s\ 1}$ **then**
- 23: $\mathcal{C}_\phi^0(p) \leftarrow$ dependência espacial atual $R_{mi}^{s\ 1}$
- 24: **break**
- 25: **end if**
- 26: **end for**
- 27: $\phi^1(M^1, R_m^{s\ 1}) \leftarrow$ novas atribuições de $R_m^{s\ 1}$ para ciclos existentes \mathcal{C}_ϕ^0
- 28: **if** Existem dependências espaciais não coletadas $R_{mi}^{s\ 1} \in R_m^{s\ 1}$ **then**
- 29: $\mathcal{C}_\phi^1 \leftarrow \mathcal{C}_\phi^1 \cup \mathcal{C}_\phi^0$ (cria novos ciclos – ou seja, \mathcal{C}_ϕ^1)
- 30: $\phi^1(M^1, R_m^{s\ 1}) \leftarrow \phi^1(M^1, R_m^{s\ 1}) \cup \mathcal{C}_\phi^1$ (adiciona os novos ciclos \mathcal{C}_ϕ^1 na solução existente)
- 31: **end if**
- 32: Tente realocar dependências espaciais $\forall R_j^{s\ 1} \in R_m^{s\ 1}$ em um ciclo coletando mais dependências espaciais
- 33: **return** $\phi^1(M^1, R_m^{s\ 1})$

existentes \mathcal{C}_ϕ^0 , salva essas novas atribuições (linhas 21-27), e cria novos ciclos, se necessário (linhas 28-31). Finalmente, nós iteramos sobre o conjunto de ciclos modificados \mathcal{C}_ϕ^1 e tentamos realocar as dependências para reduzir o número de ciclos (linha 32). Então, o algoritmo retorna a solução (linha 33).

O Algoritmo 2 resume a geração de ciclos para a solução inicial da seguinte forma. Nós começamos iterando sobre um conjunto inicial de aplicações de monitoramento M^0 e as suas dependências $R_m^{s\ 0}$, definindo como origem o um dispositivo associado a uma dependência espacial $R_{mi}^{s\ 0} \in R_m^{s\ 0}$ (linhas 3-5).

Então iteramos sobre as requisições e as inserimos no ciclo atual, desde que haja espaço disponível (linhas 6-10), definindo como próximo nó (linha 9), à partir de sua vizinhança atual, preferencialmente o primeiro nó que possua dependências ainda não

satisfeitas ou o primeiro da lista de vizinhos factíveis. Retornando a seguir, ao ponto de partida o à partir do último nó visitado d_{last} utilizando o caminho mais curto (linha 11). Finalmente, atualizamos e retornamos a solução (linhas 12-14).

Algorithm 2 Gerando um conjunto de ciclos para dependências espaciais.

Input: Conjunto inicial de aplicações de monitoramento M^0 e dependências espaciais $R_m^{s\ 0}$

- 1: $\phi^0(M^0, R_m^{s\ 0}) \leftarrow \emptyset$
- 2: $\mathcal{C}_\phi^0 \leftarrow \emptyset$
- 3: **while** M^0 não for satisfeito $R_m^{s\ 0}$ **do**
- 4: $\mathcal{C}_i \leftarrow$ novo ciclo
- 5: $\mathcal{C}_i(p) \leftarrow \mathcal{C}_i(p) \cup o$
- 6: **for** $R_{mi}^{s\ 0} \in R_m^{s\ 0}$ e \mathcal{C}_i tem capacidade disponível U para coletar todos os itens em $R_{mi}^{s\ 0}$ **do**
- 7: $\mathcal{C}_i(p) \leftarrow \mathcal{C}_i(p) \cup d \in D$ associado com $R_{mi}^{s\ 0}$
- 8: $U(p) \leftarrow U(p) -$ peso dos itens em $R_{mi}^{s\ 0}$
- 9: $\mathcal{C}_i(p) \leftarrow d \in D$ (selecione um vizinho $d \in D$)
- 10: **end for**
- 11: $\mathcal{C}_i(p) \leftarrow \mathcal{C}_i(p) \cup \text{shortestPath}(d_{last}, o)$
- 12: $\phi^0(M^0, R_m^{s\ 0}) \leftarrow \phi^0(M^0, R_m^{s\ 0}) \cup \mathcal{C}_i(p)$
- 13: **end while**
- 14: **return** $\phi^0(M^0, R_m^{s\ 0})$

3.4 Avaliação

3.4.1 Carga de Trabalho

DyPro foi implementado utilizando a linguagem Java, a fim de obtermos soluções viáveis para o problema de otimização. Todos os experimentos foram realizados em uma máquina com processador AMD Threadripper 2920X e 80 GB de RAM, rodando o sistema operacional Ubuntu 20.04. Foram consideradas diferentes instâncias de rede, geradas com Brite (Medina et al., 2001), seguindo o modelo Albert e Barabási (2000). Utilizamos infraestruturas com 50 dispositivos de encaminhamento e, em média, 200 links. Além das instâncias de infraestrutura, havia um conjunto M de aplicações de monitoramento. Variamos a quantidade de espaço disponível para incorporar itens de telemetria em pacotes de sondagem (ou seja, $U(p)$) de 100 a 1500 Bytes. E assumimos que os dispositivos de encaminhamento tem de 2 a 8 itens de telemetria possíveis para coletar, variando o seu tamanho $S(v)$ uniformemente de 2 a 20 Bytes (PAN et al., 2019). Cada aplicação de monitoramento M requisita no máximo 4 dependências espaciais, cada uma com pelo menos 2 itens (ou seja, switch ID + métrica de monitoramento) e no máximo 4. Cada experimento foi repetido 30 vezes, tendo sido suficiente neste caso para garantir um alto nível de confiança, de 95% ou superior.

3.4.2 Baseline

Comparamos DyPro com duas abordagens de orquestração de última geração: (i) o *Optimal Path Planning* (OPP) (PAN et al., 2019) e o (ii) P²INT (CASTRO et al.,

2021).

Lembrando que no trabalho de Pan et al. (2019) foi apresentado um framework que utiliza a técnica dividir para conquistar, desacoplando a solução em um mecanismo de roteamento e uma política de geração de caminhos. Baseado no mecanismo de roteamento, projetaram duas políticas de planejamento de caminhos para gerar múltiplos caminhos INT não sobrepostos que cobrem toda a rede. A primeira é baseada em *Depth-First Search* (DFS), que é simples, mas eficiente em termos de tempo. A segunda é um algoritmo baseado nos circuitos de Euler que pode gerar de forma ótima caminhos INT não sobrepostos com um número mínimo de caminhos. Uma característica interessante no trabalho de Pan et al. (2019), é que ele codifica o estado do tráfego em toda a rede em uma série de "imagens bitmap", o que permite utilizar técnicas avançadas, como o reconhecimento de padrões, para automatizar o monitoramento e a solução de problemas da rede.

Enquanto no trabalho de Castro et al. (2021) apresentam uma solução que utiliza uma abordagem heurística de Randomização de Aresta (ER) adaptada, para coordenar como os pacotes de sondagem são gerados e roteados, buscando visitar todos os links e coletar todos os dados INT. O problema foi formulado como um modelo de programação linear inteira, consistindo na generalização de dois problemas de otimização, sendo eles o Problema de Roteamento em Arcos Capacitados e o Problema de Empacotamento.

3.4.3 Resultados

Primeiro, nós analisamos a qualidade da abordagem proposta avaliando o número de ciclos de sondagem gerados pelas soluções consideradas.

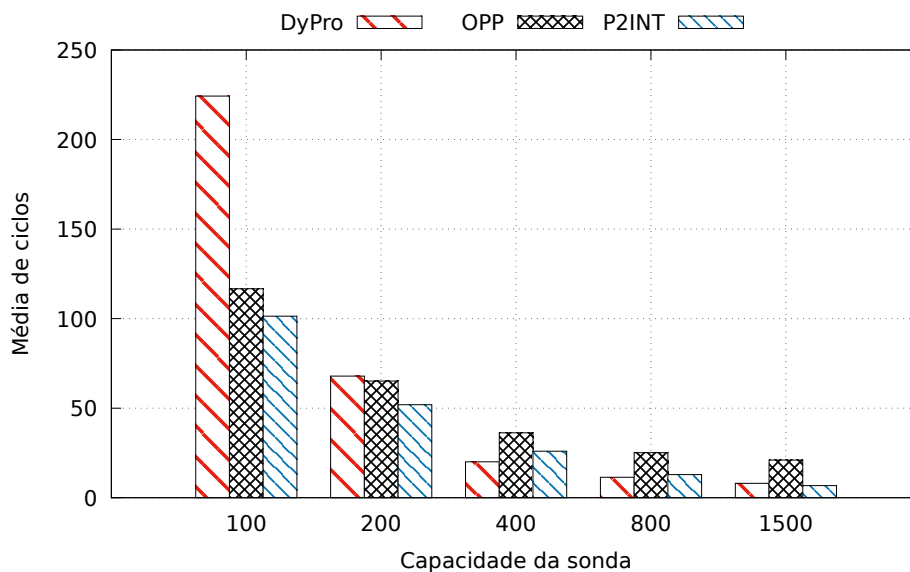


Figura 5 – Número de ciclos de sondagem para um aumento da capacidade disponível da sonda.

A Figura 5 ilustra a quantidade de ciclos de sondagem gerados para uma capacidade crescente (ou seja, de 100 a 1500 Bytes). À medida que a capacidade da sonda aumenta, observamos uma redução no número de ciclos de sondagem sendo gerados. Isso acontece principalmente porque há mais espaço para acomodar dados INT em um único pacote de sondagem.

A redução no número de ciclos de sondagem pode chegar a 30x no caso onde o DyPro os gera. Por exemplo, DyPro gera menos de 10 ciclos de 1500 Bytes, enquanto o número aumenta para quase 230 ciclos quando a capacidade de sondagem é definida para 100 Bytes. Além disso, observamos também que o número de ciclos de sondagem gerados pelo DyPro é, em geral, igual ou inferior ao seu concorrente.

De fato, para ciclos de sondagem maiores, DyPro se aproxima do número mínimo de ciclos dado por P²INT. Este, por sua vez, é o efeito de não coletar todos os dados INT da infraestrutura, mas aqueles pertencentes aos requisitos das aplicações de monitoramento.

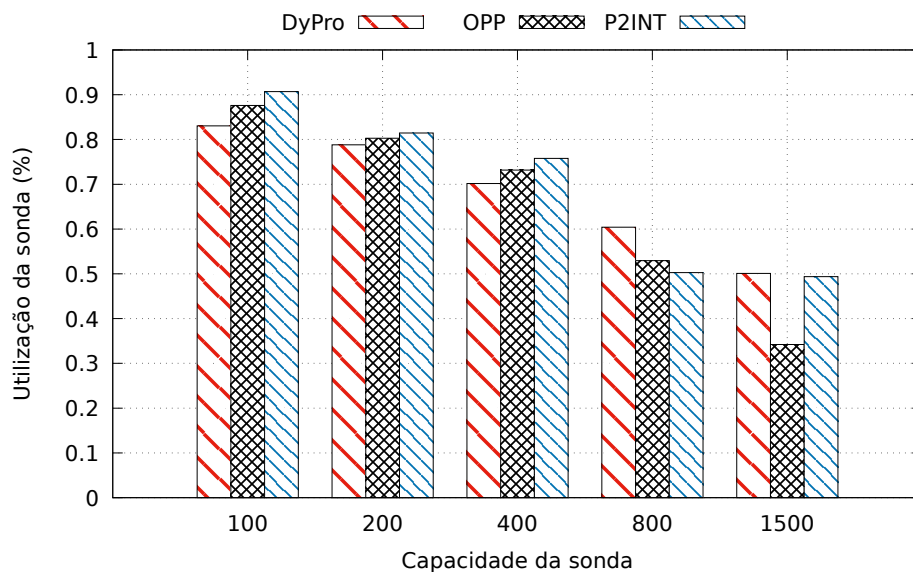


Figura 6 – Uso da capacidade da sonda.

Em seguida, avaliamos a média de uso da capacidade de sondagem pelos ciclos gerados. A Figura 6 ilustra a utilização média da capacidade para um tamanho crescente de pacotes de sondagem. Em geral, quanto mais recursos disponíveis nos pacotes de sondagem, menor é a utilização média de recursos.

Em média, pacotes de 100 Bytes tem mais de 80% (por exemplo, DyPro) de sua capacidade sendo usada, no entanto, quando a capacidade do pacote é aumentada para 1500 Bytes, essa utilização diminui para mais de 50% no caso do DyPro. No caso do OPP, essa redução no uso da sonda decai para 34%. Isso acontece devido aos problemas de fragmentação ao coletar dados INT. Como os dados INT tem tamanhos arbitrários, dependendo do conjunto de dados INT coletados por um ciclo de sondagem, isso pode levar a um estado da solução em que nenhum outro dado INT pode ser coletado.

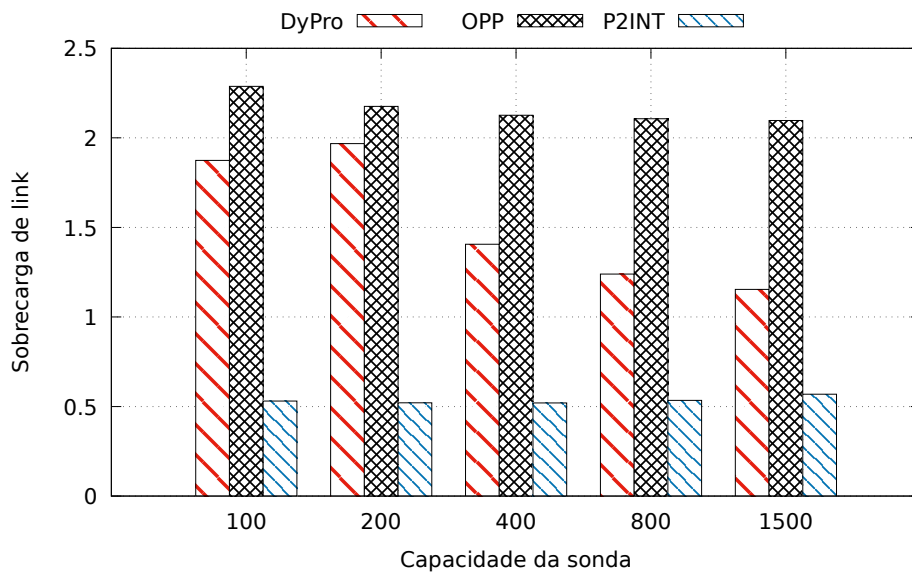


Figura 7 – Número de ciclos de sondagem por link de rede (ou seja, link overhead).

Em seguida, avaliamos a cobertura dos links de rede pelas soluções existentes. A Figura 7 mostra a cobertura dos links de rede, como a média de ciclos de sondagem por link de rede. Valores mais altos indicam que os links de rede estão sendo cobertos demais (ou seja, várias vezes), representando um desperdício de recursos, enquanto valores mais baixos indicam que um link tem cobertura abaixo da média. Chamamos essa métrica de sobrecarga de link. Como podemos observar, DyPro tende a manter esse valor próximo de um, à medida que a capacidade de pacotes de sondagem aumenta, enquanto as outras estratégias produzem soluções com links de rede começando cobertos por até 2x mais do que o necessário (por exemplo, OPP).

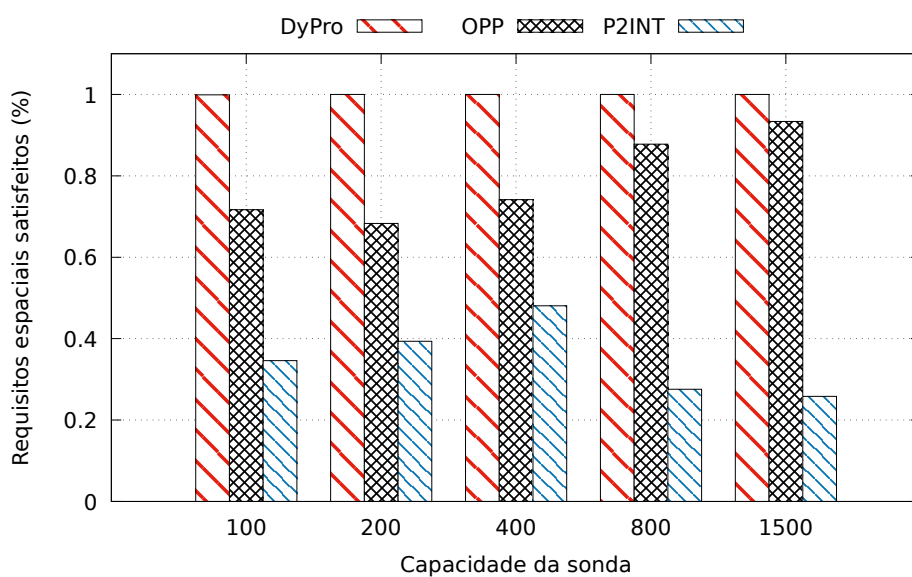


Figura 8 – Número de aplicações de monitoramento satisfeitas.

Em seguida, avaliamos o número de requisitos das aplicações de monitoramento que são atendidos por nossa abordagem. A Figura 8 ilustra o número médio de requisitos satisfeitos para um tamanho crescente de pacotes de sondagem. Podemos observar que DyPro pode satisfazer todos os requisitos das aplicações de monitoramento, enquanto o OPP aumenta a porcentagem de requisitos satisfeitos à medida que a capacidade de sondagem aumenta.

Este é o resultado da coleta de dados INT de forma arbitrária e inconsciente, das dependências existentes entre eles. DyPro, em vez disso, concentra-se desde o início em satisfazer a quantidade máxima de dependências espaciais selecionando os dados INT corretos dos dispositivos de encaminhamento.

Por último, analisamos o DyPro em relação ao número de alterações feitas na infraestrutura de rede para acomodar as alterações nos requisitos de monitoramento. Observe que essas alterações são traduzidas em reconfiguração da rede e, portanto, quanto menos alterações, melhor.

Para esses experimentos, consideramos as mesmas configurações dos experimentos anteriores. A partir da solução inicial fornecida, nós incrementamos novas solicitações de monitoramento (de 1 a 10) para quantificar como o DyPro alteraria a solução existente para atender aos novos requisitos. Devido a restrições de espaço, focamos nossa análise no número de links de rede que foram reconfigurados/alterados para atender a novas solicitações.

Tabela 2 – Resumo dos resultados para cenários dinâmicos.

# novas requisições	# alterações DyPro	# alterações OPP	# razão OPP/DyPro
1	7.73	109.29	14.1X
2	18.07	222.76	12.3X
3	29.07	345.53	11.8X
4	43.68	471.60	10.7X
5	57.77	587.92	10.1X
6	75.82	695.92	9.1X
7	96.07	812.95	8.4X
8	121.16	926.07	7.6X
9	154.76	1032.97	6.6X
10	187.64	1153.72	6.1X

A Tabela 2 resume os resultados obtidos pelo DyPro quando comparados ao OPP. A tabela mostra a média de alterações de link na solução existente para atender aos novos requisitos. Primeiro, observamos que os ganhos (ou seja, quarta coluna) são maiores ao adicionar o primeiro conjunto de requisitos de monitoramento. Ou seja, DyPro pode satisfazer os novos requisitos com 12-14X menos alterações do que o OPP ao adicionar 1-2 novos requisitos de monitoramento.

Em resumo, DyPro tenta sempre acomodar novas solicitações em ciclos de sondagem existentes trocando itens de telemetria antigos pelos novos.

4 ALGORITMO META-HEURÍSTICO ACO

Visando obter uma solução inicial de melhor qualidade, e conseqüentemente melhoria global em nossa abordagem, implementamos um algoritmo meta-heurístico para o problema proposto neste Trabalho, baseado em *Ant Colony Optimization* (ACO) (DORIGO; STÜTZLE, 2004), tendo sido aplicado nos últimos anos em diversos problemas reais de grande complexidade, tais como: problemas de roteamento, aprendizado de máquina, agendamento, entre outros (DORIGO; STÜTZLE, 2004).

Nosso algoritmo inicia-se instanciando uma população de formigas com a mesma quantidade de nós presente na infraestrutura, e suas respectivas soluções por meio do algoritmo ACO. A seguir a matriz de feromônios é atualizada com base nos caminhos únicos desenvolvidos na solução de cada uma das formigas, finalizando cada iteração com a atualização da melhor solução global encontrada até o momento (S_{best}). O algoritmo segue até que o número máximo de iterações tenha sido atingido ou caso a solução global fique estagnada por muito tempo, retornando então a melhor solução encontrada (S_{best}). Nas subseções seguintes iremos analisar resumidamente as etapas do Algoritmo 3.

Algorithm 3 Visão resumida da abordagem heurística baseada em ACO

Input: Conjunto inicial de aplicações de monitoramento M^0 e dependências espaciais $R_m^{s\ 0}$

- 1: $C_\phi^0 \leftarrow \emptyset$
- 2: $S_{best} \leftarrow \emptyset$
- 3: $\phi^0(M^0, R_m^{s\ 0}) \leftarrow \emptyset$
- 4: *inicializaDados()* ▷ parâmetros, feromônios e formigas
- 5: **while** critérios de parada não forem satisfeitos **do**
- 6: **for** cada formiga **do**
- 7: **while** M^0 não for satisfeito $R_m^{s\ 0}$ **do**
- 8: $C_i \leftarrow$ novo ciclo
- 9: $C_i(p) \leftarrow C_i(p) \cup o$
- 10: **for** $R_{mi}^{s\ 0} \in R_m^{s\ 0}$ e C_i tem cap. disp. U para coletar todos os itens em $R_{mi}^{s\ 0}$ **do**
- 11: $C_i(p) \leftarrow C_i(p) \cup d \in D$ associado com $R_{mi}^{s\ 0}$
- 12: $U(p) \leftarrow U(p) -$ peso dos itens em $R_{mi}^{s\ 0}$
- 13: $C_i(p) \leftarrow$ selecionaProxNo()
- 14: **end for**
- 15: $C_i(p) \leftarrow C_i(p) \cup shortestPath(d_{last}, o)$
- 16: $\phi^0(M^0, R_m^{s\ 0}) \leftarrow \phi^0(M^0, R_m^{s\ 0}) \cup C_i(p)$
- 17: **end while**
- 18: **end for**
- 19: *atualizaFeromonios()*
- 20: *atualizaSolucaoGlobal()*
- 21: **end while**
- 22: **return** S_{best}

4.1 Construção das Soluções

Na variação ACO denominada *Ant System* utilizada pelo nosso algoritmo 3, cada formiga representa uma solução com ciclos de sondagem que iniciam em um determinado

nó, definido para cada formiga k no momento da sua criação, e vão construindo suas rotas consecutivamente ao coletar todas as dependências espaciais possíveis, das aplicações de monitoramento.

Assim, podemos resumir a geração de cada solução para uma formiga k , da seguinte forma. Começamos iterando sobre um conjunto inicial de aplicações de monitoramento M^0 e as suas dependências R_m^s , definindo como origem o um dispositivo pré-determinado para a formiga k (linhas 7-9). Então iteramos sobre as requisições e as inserimos no ciclo atual, desde que haja espaço disponível (linhas 10-14) e retornamos ao ponto de partida o à partir do último nó visitado d_{last} utilizando o caminho mais curto (linha 15).

Em cada passo da construção, a formiga k aplica a fórmula probabilística (4.1) para definir o próximo nó a ser visitado j com base no atual i (linha 13), onde N_{lst}^k são os nós vizinhos ainda não visitados, da formiga k à partir do nó i . η_{ij} representa a informação heurística do arco ij , sendo a soma de todas as dependências espaciais ainda não satisfeitas do nó j . E τ_{ij} representa a quantidade de feromônio do arco ij , sendo os parâmetros α e β responsáveis, respectivamente, pela influência dos feromônios e da informação heurística. Utilizando um mecanismo de roleta, o arco com maior valor probabilístico (p_{ij}^k) possui maior chance de ser escolhido. Esse mecanismo é repetido até que todas as dependências espaciais possíveis tenham sido satisfeitas, repetindo esse processo k vezes, gerando uma população de k soluções. Se não for possível utilizar a fórmula probabilística, não havendo vizinhos não visitados, um nó vizinho qualquer será selecionado aleatoriamente até que todas as requisições possíveis tenham sido satisfeitas.

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_{lst}^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \quad \text{se } j \in N_{lst}^k \quad (4.1)$$

4.2 Feromônios

Após a construção de cada uma das soluções, a matriz de feromônios τ é atualizada com base nos arcos que cada formiga utilizou. Então primeiro todos os feromônios são evaporados segundo a Equação (4.2), onde ρ é o coeficiente de evaporação.

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad (4.2)$$

E na sequência é depositado o somatório dos feromônios de cada formiga que visitaram o arco ij . Completando nossa fórmula de evaporação e depósito de feromônios, temos a Equação (4.3).

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4.3)$$

As contribuições de cada formiga k são calculadas por meio do inverso do custo de sua rota única, ou seja, todas as rotas da formiga k são resumidas em uma única rota

contendo todos os nós que ela visitou, sem duplicatas. Como vemos na equação (4.4).

$$\begin{cases} \Delta\tau_{ij}^k = Q/T_k \\ \Delta\tau_{ij}^k = 0 \text{ em caso contrário, onde } Q \text{ é uma constante} \end{cases} \quad (4.4)$$

4.3 Melhor Solução

Depois de obter as soluções de cada uma das formigas, ao final de cada iteração, é realizada uma seleção entre as soluções obtidas na iteração atual afim de escolher a melhor (linha 20).

Primeiramente são selecionadas as soluções que possuam o maior número de requisições satisfeitas. Então, destas, é realizada uma nova seleção, obtendo aquelas com a melhor taxa de utilização da sonda, somando-se a utilização da sonda em cada ciclo da solução e dividindo pelo número de ciclos. Se nesta etapa tivermos mais de uma solução com o mesmo resultado, é selecionada a primeira da lista. Por fim, temos a melhor solução da iteração atual, que será comparada com a melhor solução global até o momento (S_{best}) para atualização.

4.4 Avaliação

4.4.1 Carga de Trabalho

Para avaliar nossa abordagem com a implementação do algoritmo *Ant Colony Optimization* (DyProACO), utilizamos o mesmo ambiente descrito na seção 3.4.1 deste documento.

4.4.2 Sensibilidade dos Parâmetros

O resultado de nossa análise de sensibilidade referente a três parâmetros ACO é apresentado neste tópico, sendo eles, de influência de feromônios (α), de informação heurística (β), e o coeficiente de evaporação (ρ). Durante os testes de sensibilidade, mantivemos fixos os seguintes parâmetros: a quantidade de formigas (50), de feromônio inicial (1) e a quantidade máxima de iterações (1000).

Analisamos os parâmetros α e β , variando ambos de maneira uniforme de 0.1 a 0.9, e o coeficiente de evaporação em nossos testes para 0.5 e 0.98. Na tabela 3, podemos analisar a média dos resultados obtidos da variação de α e β , com o melhor coeficiente de evaporação encontrado em nossos testes, de 0.5.

Desta forma, à partir destes resultados, ajustamos os parâmetros α e β para 0.3 e ρ para 0.5 em todos os nossos experimentos.

Tabela 3 – Análise de Sensibilidade - Parâmetros α e β com ρ em 0.5.

Parâmetros α e β	Média de Ciclos	Uso da Sonda	Requisitos Satisfeitos	Sobrecarga de Dispositivos	Sobrecarga de Link
0.1	33,822	78,60%	99%	1,864	1,788
0.2	33,858	78,60%	99%	1,834	1,790
0.3	32,912	78,80%	99%	1,754	1,776
0.4	33,812	78,60%	99%	1,752	1,786
0.5	33,832	78,60%	99%	1,800	1,790
0.6	33,784	78,40%	99%	1,750	1,794
0.7	33,918	78,20%	99%	1,810	1,796
0.8	33,800	78,60%	98%	1,804	1,782
0.9	33,974	78,40%	99%	1,758	1,798

4.4.3 Resultados

Analisamos abaixo a qualidade de nossa abordagem com a implementação do algoritmo *Ant Colony Optimization* (DyProACO), e comparamos com nossa abordagem inicial (DyPro).

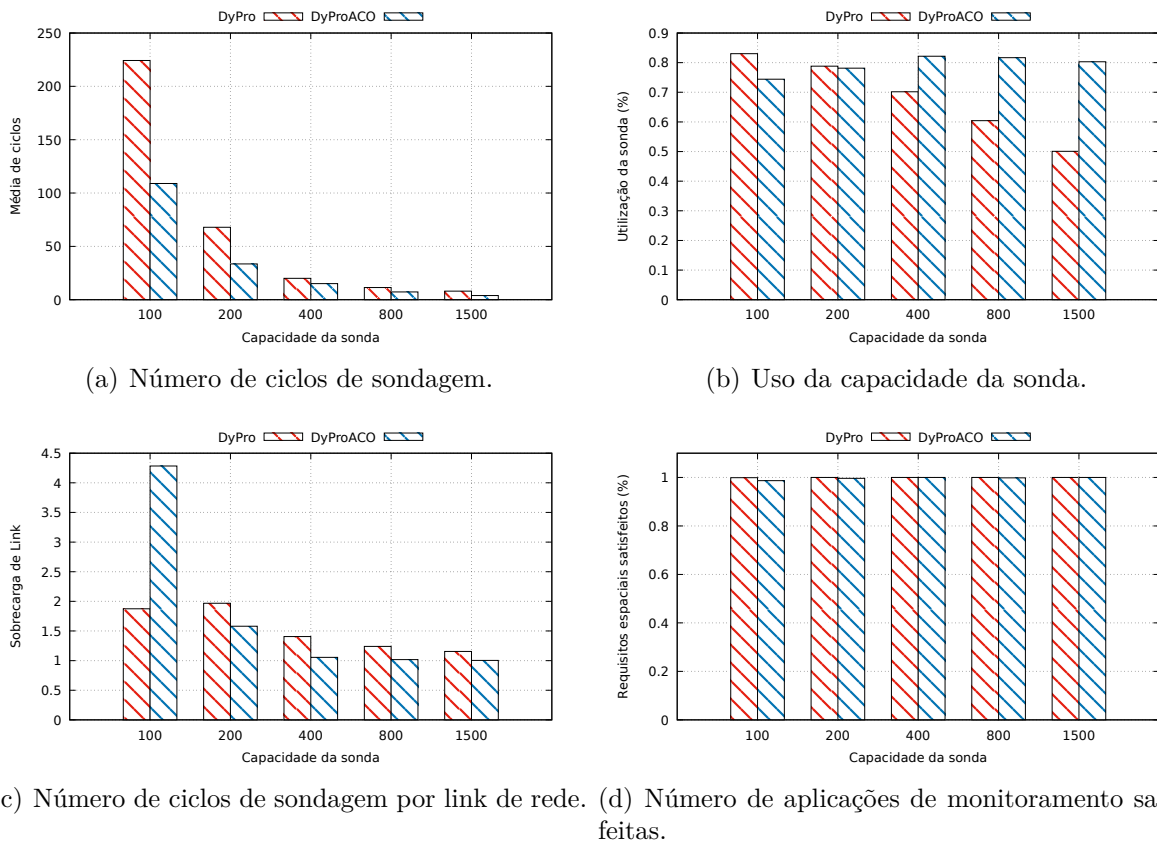


Figura 9 – Comparação entre nossas abordagens.

Observamos na Figura 9(a) que ao aumentarmos a capacidade da sonda, há uma redução no número de ciclos de sondagem sendo gerados, pois há mais espaço para acomodar dados INT em um único pacote de sondagem. Observamos também que o DyProACO

consegue obter melhores resultados que o DyPro, ao gerar ciclos de sondagem otimizados para a coleta de dados INT, construindo até 50% menos ciclos. Isso é reflexo de uma melhor utilização média de recursos, ao acomodar de maneira otimizada os dados INT na sonda, conforme demonstrado na Figura 9(b), onde o DyProACO obtém índices de aproveitamento iguais ou superiores a 80% quando definimos nossa capacidade de sondagem acima de 200 Bytes.

Em seguida, ao avaliarmos a cobertura dos links de rede na Figura 9(c), vemos que DyProACO necessitou cobrir até 4X mais do que o necessário quando definimos nossa capacidade para 100 Bytes, porém, se aproximando rapidamente do ponto ideal ao aumentarmos a capacidade de sondagem. Também avaliamos o número de requisitos das aplicações de monitoramento que são atendidos por nossas abordagens, na Figura 9(d), demonstrando que ambas conseguem satisfazer quase todos os requisitos das aplicações de monitoramento, buscando sempre satisfazer a quantidade máxima de dependências espaciais selecionando os dados INT corretos dos dispositivos de encaminhamento.

Por último, analisamos nossas abordagens em relação ao número de alterações feitas na infraestrutura de rede para acomodar as alterações nos requisitos de monitoramento. Para esses experimentos, consideramos as mesmas configurações dos experimentos anteriores, onde à partir da solução inicial fornecida, nós incrementamos novas solicitações de monitoramento (de 1 a 10) para quantificar como o DyProACO alteraria a solução existente para atender aos novos requisitos. Focamos nossa análise no número de links de rede que foram reconfigurados/alterados para atender a novas solicitações.

Na Tabela 4 tem-se um resumo dos resultados obtidos pelo DyProACO quando comparados ao DyPro. A tabela mostra a média de alterações de link na solução existente para atender aos novos requisitos, lembrando que essas alterações são traduzidas em reconfiguração da rede e, portanto, quanto menos alterações, melhor. Fica evidente que o DyProACO consegue satisfazer os novos requisitos, em média, com até 7X menos alterações que o DyPro, tentando sempre acomodar novas solicitações em ciclos de sondagem existentes trocando itens de telemetria antigos pelos novos. Assim, em resumo, observamos que ao utilizar o algoritmo baseado em *Ant Colony Optimization* (ACO) em conjunto com o DyPro, para gerar uma solução inicial de melhor qualidade, nos permitiu obter resultados satisfatórios em comparação ao DyPro sem esta implementação.

Tabela 4 – Resumo dos resultados para cenários dinâmicos (DyPro vs DyProACO).

# novas requisições	# alterações DyPro	# alterações DyProACO	# razão DyPro/DyProACO
1	7.73	1.23	6.2X
2	18.07	2.70	6.6X
3	29.07	4.36	6.6X
4	43.68	6.13	7.1X
5	57.77	8.16	7.0X
6	75.82	10.40	7.2X
7	96.07	13.06	7.3X
8	121.16	16.10	7.5X
9	154.76	19.33	8.0X
10	187.64	23.00	8.0X

5 CONCLUSÃO

O monitoramento de rede de forma coordenada e eficiente é imprescindível para garantir a confiabilidade da comunicação, sendo necessário criar meios para orquestrar a coleta de dados de telemetria de maneira eficiente, pois dependendo de quais itens de telemetria são coletados podemos degradar a visibilidade da rede, em termos de cobertura e atualização (MARQUES et al., 2019), ou ainda afetar o desempenho das aplicações de monitoramento da rede (HOHEMBERGER et al., 2019).

Neste trabalho, em um primeiro momento formalizamos o DyPro como um planejamento de sondagem dinâmico para INT, por meio de um modelo MILP, e introduzimos um procedimento heurístico capaz de encontrar soluções viáveis. Nossa proposta, DyPro, se concentra em satisfazer as dependências de telemetria de acordo com os requisitos das aplicações de monitoramento e, conforme mostram os resultados, supera as abordagens de última geração nesse sentido em até 5x.

Em um segundo momento implementamos um algoritmo baseado em *Ant Colony Optimization* (ACO), com o objetivo de construir uma solução inicial de melhor qualidade para nosso procedimento dinâmico. Desta forma, demonstramos que foi possível reduzir em até 7x a necessidade de alterações ao tentar acomodar novas solicitações em ciclos de sondagem existentes trocando itens de telemetria antigos pelos novos.

Para trabalhos futuros, sugere-se testar outras versões do algoritmo ACO para construir a solução inicial, e analisar a possibilidade de implementar este também nos cenários dinâmicos.

5.1 Publicações Científicas

L. M. Dallanora, A. G. Castro, R. I. T. d. C. Filho, F. D. Rossi, A. F. Lorenzon and M. C. Luizelli, "DyPro: Dynamic Probing Planning for In-Band Network Telemetry," 2022 IEEE Symposium on Computers and Communications (ISCC), 2022, pp. 1-6, doi: 10.1109/ISCC55528.2022.9912881.

REFERÊNCIAS

- AKHTAR, A. Evolution of ant colony optimization algorithm - A brief literature review. **CoRR**, abs/1908.08007, 2019. Disponível em: <<http://arxiv.org/abs/1908.08007>>. Citado na página 36.
- ALBERT, R.; BARABÁSI, A.-L. Topology of evolving networks: Local events and universality. **Physical Review Letters**, American Physical Society, v. 85, p. 5234 – 5237, Dec 2000. Citado na página 51.
- BASAT, R. B. et al. PINT: Probabilistic In-band Network Telemetry. v. 4, 2020. Disponível em: <<http://arxiv.org/abs/2007.03731>>. Citado 4 vezes nas páginas 39, 41, 42 e 43.
- BHAMARE, D. et al. IntOpt: In-Band Network Telemetry Optimization for NFV Service Chain Monitoring. **IEEE International Conference on Communications**, IEEE, v. 2019-May, p. 0–6, 2019. ISSN 15503607. Citado 6 vezes nas páginas 27, 28, 39, 41, 42 e 43.
- BIRATTARI, M. et al. **Classification of Metaheuristics and Design of Experiments for the Analysis of Components**. 2001. Citado na página 35.
- BOSSHART, P. et al. P4: Programming protocol-independent packet processors. **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 44, n. 3, p. 87–95, jul. 2014. ISSN 0146-4833. Disponível em: <<https://doi.org/10.1145/2656877.2656890>>. Citado 3 vezes nas páginas 27, 31 e 34.
- CASTRO, A. G. et al. Patcher: Towards fault - tolerant probing planning for in-band network telemetry. In: **2020 IEEE Latin-American Conference on Communications (LATINCOM)**. [S.l.: s.n.], 2020. p. 1–6. Citado 5 vezes nas páginas 28, 39, 42, 43 e 45.
- CASTRO, A. G. et al. Near-Optimal Probing Planning for In-Band Network Telemetry. **IEEE Communications Letters**, v. 7798, n. c, p. 1–5, 2021. ISSN 15582558. Citado 8 vezes nas páginas 27, 28, 40, 42, 43, 45, 46 e 52.
- CHEN, B. et al. Orchestrating segment routing with int for low-overhead and adaptive network monitoring. 2021. Citado 5 vezes nas páginas 27, 28, 40, 42 e 43.
- CHOLE, S. et al. drmt: Disaggregated programmable switching. In: **ACM SIGCOMM 17**. New York, NY, USA: ACM, 2017. p. 1–14. ISBN 978-1-4503-4653-5. Citado na página 34.
- COLIN, E. **Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas**. [S.l.]: LTC, 2007. v. 1. Citado na página 35.
- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. **Journal of the Operations Research Society of America**, v. 2, n. 4, p. 393–410, 1954. Citado na página 48.
- DORIGO, M. Optimization, learning and natural algorithms[in italian]. **PhD thesis. Dipartimentodi Elettronica, Politecnico di Milano, Milan.**, 1992. Citado na página 37.

- DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: Optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics—Part B**, p. 29 – 41, 1996. Citado na página 37.
- DORIGO, M.; STÜTZLE, T. **Ant Colony Optimization**. [S.l.]: The MIT Press, 2004. v. 1. Citado 2 vezes nas páginas 36 e 57.
- FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The road to SDN: An intellectual history of programmable networks. **Computer Communication Review**, v. 44, n. 2, p. 87–98, 2014. ISSN 19435819. Citado 3 vezes nas páginas 29, 30 e 31.
- FERNANDO, L. et al. Capítulo 4 Introdução à Linguagem P4-Teoria e Prática. **Sbrc 2018**, 2018. Disponível em: <<https://www.netronome.com/products/smartnic/overview/>> Citado na página 31.
- GENG, Y. et al. Simon: A simple and scalable method for sensing, inference and measurement in data center networks. **Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2019**, p. 549–564, 2019. Citado 3 vezes nas páginas 39, 42 e 43.
- HILLIER, F.; LIEBERMAN, G. **Introdução à pesquisa operacional**. [S.l.]: AMGH, 2013. v. 9. Citado na página 35.
- HOHEMBERGER, R. Orquestração Escalável da Coleta de Dados de Telemetria In-band em Planos de Dados Programáveis. 2022. Citado na página 30.
- HOHEMBERGER, R. et al. Orchestrating in-band data plane telemetry with machine learning. **IEEE Communications Letters**, v. 23, n. 12, p. 2247–2251, Dec 2019. ISSN 2373-7891. Citado 7 vezes nas páginas 27, 28, 39, 42, 43, 46 e 63.
- JEYAKUMAR, V. et al. Millions of little minions: Using packets for low latency network programming and visibility. In: **Proceedings of the ACM SIGCOMM '14**. New York, NY, USA: ACM, 2014. p. 3–14. ISBN 978-1-4503-2836-4. Citado 2 vezes nas páginas 27 e 34.
- KERNIGHAN, B. W.; LIN, S. An efficient heuristic procedure for partitioning graphs. **The Bell System Technical Journal**, v. 49, n. 2, p. 291–307, 1970. Citado na página 41.
- KIM, C. et al. In-band Network Telemetry via Programmable Dataplanes. **Sosr**, p. 2–3, 2015. Disponível em: <<https://nkatta.github.io/papers/int-demo.pdf>>. Citado na página 32.
- KREUTZ, D. et al. Software-defined networking: A comprehensive survey. **Proceedings of the IEEE**, v. 103, n. 1, p. 14–76, 2015. Citado na página 29.
- LIN, Y. et al. NetView: Towards on-demand network-wide telemetry in the data center. **Computer Networks**, Elsevier B.V., v. 180, n. February, 2020. ISSN 13891286. Citado 5 vezes nas páginas 27, 28, 40, 42 e 43.
- LIU, Z. et al. Netvision: Towards network telemetry as a service. In: **2018 IEEE 26th International Conference on Network Protocols (ICNP)**. [S.l.: s.n.], 2018. p. 247–248. ISSN 1092-1648. Citado 2 vezes nas páginas 28 e 45.

- MARQUES, J. A. et al. An optimization-based approach for efficient network monitoring using in-band network telemetry. **Journal of Internet Services and Applications**, v. 10, n. 1, p. 16, Jun 2019. Citado 8 vezes nas páginas 27, 28, 38, 41, 42, 43, 46 e 63.
- Medina, A. et al. Brite: an approach to universal topology generation. In: **IEEE MASCOTS 2001**. [S.l.: s.n.], 2001. Citado na página 51.
- MELIÁN, B. et al. Metaheuristics: A global view. **Revista Iberoamericana de Inteligencia Artificial**, v. 19, p. 7–28, 2003. ISSN 1137-3601. Disponível em: <<http://www.aepia.org/revista>>. Citado na página 35.
- PAN, J. et al. Orchestrating probabilistic in-band network telemetry for network monitoring. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 441–446. ISBN 9781665409506. Citado 3 vezes nas páginas 41, 42 e 43.
- PAN, T. et al. INT-path: Towards Optimal Path Planning for In-band Network-Wide Telemetry. In: **IEEE INFOCOM 2019 - IEEE Conference on Computer Communications**. IEEE, 2019. v. 2019-April, p. 487–495. ISBN 978-1-7281-0515-4. ISSN 0743166X. Disponível em: <<https://ieeexplore.ieee.org/document/8737529/>>. Citado 9 vezes nas páginas 27, 28, 34, 38, 42, 43, 45, 51 e 52.
- RALPHS, T. K. et al. On the capacitated vehicle routing problem. **Mathematical Programming**, v. 94, n. 2, p. 343–359, 2003. Disponível em: <<https://doi.org/10.1007/s10107-002-0323-0>>. Citado na página 46.
- SIMSEK, G.; ERGENC, D.; ONUR, E. Efficient Network Monitoring via In-band Telemetry. p. 1–6, 2021. Citado 3 vezes nas páginas 41, 42 e 43.
- SONG, E. et al. Int-filter: Mitigating data collection overhead for high-resolution in-band network telemetry. In: **GLOBECOM 2020 - 2020 IEEE Global Communications Conference**. [S.l.: s.n.], 2020. p. 1–6. Citado 4 vezes nas páginas 40, 41, 42 e 43.
- SUCUPIRA, I. R. Um estudo empírico de hiper-heurísticas. 2007. Citado na página 36.
- The P4 Language Consortium. P4 16 Language Specification v1.2.1. p. 129, 2018. ISSN 0279-1072. Citado na página 32.
- TIBSHIRANI, R. Regression Shrinkage and Selection Via the Lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, v. 58, n. 1, p. 267–288, 1996. Citado na página 39.
- ZHANG, Y. et al. Automating rapid network anomaly detection with in-band network telemetry. **IEEE Networking Letters**, Institute of Electrical and Electronics Engineers (IEEE), v. 4, p. 39–42, 11 2021. Citado 3 vezes nas páginas 41, 42 e 43.
- ZHANG, Z.; SU, W.; TAN, L. In-band network telemetry task orchestration based on multi-objective optimization. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 354–357. ISBN 9784885523328. Citado 5 vezes nas páginas 27, 28, 41, 42 e 43.
- ZHOU, Y. Runtime analysis of an ant colony optimization algorithm for tsp instances. **Evolutionary Computation, IEEE Transactions on**, v. 13, p. 1083 – 1092, 11 2009. Citado na página 36.

ZHU, Y. et al. Packet-level telemetry in large datacenter networks. In: **Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication**. [S.l.: s.n.], 2015. p. 479–491. Citado na página 27.

Apêndices

APÊNDICE A – DYPRO: DYNAMIC PROBING PLANNING FOR IN-BAND NETWORK TELEMETRY

Título: DyPro: Dynamic Probing Planning for In-Band Network Telemetry

Autores: Leandro M. Dallanora, Ariel G. Castro, Roberto I. T. da Costa Filho, Fabio D. Rossi, Arthur F. Lorenzon, Marcelo C. Luizelli.

Conferência: IEEE Symposium on Computers and Communications (ISCC).

Abstract: *In-band Network Telemetry (INT) is a novel net-work monitoring mechanism that improves fine-grained net-work visibility. Despite the increasing research efforts towards the orchestration of INT data acquisition, little has yet been done to efficiently collect telemetry data from the network considering monitoring applications requirements. In this paper, we introduce DyPro – a dynamic probing planning for INT. In particular, DyPro ensures that telemetry dependencies are always satisfied by monitoring application requirements. We theoretically formalize it as a Mixed-Integer Linear Programming (MILP) optimization model and propose a heuristic procedure to efficiently solve it. Results show that DyPro can outperform state-of-the-art solutions by up to 5x regarding the percentage of monitoring applications satisfied.*

Link: <<https://ieeexplore.ieee.org/document/9912881>>

ÍNDICE

ACO, 36, 57, 61, 63
API, 30, 32

CVRP, 46

DCN, 39
DDoS, 39
DFS, 38, 52
DyPro, 13, 45, 47–49, 51, 53–56, 60, 61,
63
DyProACO, 59–61

ER, 40, 52

FPGA, 31

INT, 13, 27, 28, 33, 34, 38–43, 45, 46, 52,
53, 55, 60, 61, 63
INTO, 38, 41
IoT, 27
IPFIX, 33

MILP, 63
MTU, 45

NFV, 39
NIC, 39
NOS, 29

ONF, 30
ONOS, 30
OPP, 51, 53–55

P4, 27, 31, 32, 34, 41
P²INT, 40, 51, 53
PLI, 46

QoE, 33

SARG, 39
SDN, 29–31
SNMP, 27, 32, 41
SR, 40