UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

IZADORA DOURADO BERTI

Estudo de Aplicabilidade de Redes Programáveis de Baixo Custo para Transmissão de Dados em Links de Baixa Capacidade

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Weverton Cordeiro
Co-orientador: Prof. André Scheibe

Porto Alegre
2021

AGRADECIMENTOS

Quero agradecer especialmente ao meu co-orientador André Scheibe por todo auxílio na realização desse projeto e cujas palavras permitiram eu não desistir no caminho, ao orientador Weverton Cordeiro pela ideia de monografia e ter me acompanhado desde início de 2021, minha mãe Elilia Dourado pelo apoio moral e todo o incentivo, meu pai Saul Berti por toda a estrutura que me proporcionou durante a graduação e minha irmã Ana Shaista que exigiu estar nos meus agradecimentos.

Grata também a todos os professores com quem já tive a honra de ser aluna e que através dos seus ensinamentos permitiram que hoje eu pudesse hoje estar concluindo este trabalho; e aos amigos, pela compreensão das ausências e humor flutuante.

RESUMO

O foco de estudo do trabalho de conclusão de curso é fazer uma análise sobre a aplicabilidade de redes programáveis de baixo custo com capacidade de transmissão de dados muito baixa para fazer transmissão de dados em redes comunitárias e em uso em ambientes hostis à transmissão de dados sem fio (por exemplo, florestas). Redes programáveisseparam o plano de controle do plano de dados, o que torna o controle da rede diretamente programável e moldado de acordo com a abstração e aplicações desejadas para o determinado serviço de rede. O enfoque será em cima do protocolo LTP (Lightweight Tunnel Protocol) que, ao comprimir o cabeçalho do pacote de dados, busca aumento no desempenho em um linkde transmissão, utilizando-se de uma arquitetura LPWAN de baixo custo.

Por ser um protocolo novo, suas limitações ainda não são plenamente conhecidas, então são criadas hipóteses sobre suas potencialidades: sobre como aumentar escalabilidade, mudanças na topologia, como lida com perda de pacotes e se downloads da internet são realizados com sucesso se há perda de pacotes. Os experimentos mostraram que é um protocolo viável para ser implementado em redes comunitárias, demonstrando uma vantagem sobre uma estrutura standart sem uso de protocolo LTP, além de ser apto a lidar com um percentual de perdas de até 5%, um valor aceitável considerando seu meio wireless. Percebeu-se também que em relação à aumentar a escalabilidade, a proposta de aumentar a quantidade de bytes do cabeçalho não se mostrou viável por estourar o registrador de combinações de dispositivo e *TAG***,** mas poderia em novo trabalho uma alternativa para aumentar a escalabilidade utilizando outro tipo de controle diferente de um registrador.

Palavras-chave: LTP (Lightweight Tunnel Protocol), Low Power Wide Area Networks (LPWAN), SDN, Rede definida por software, P4, Redes comunitárias, Lightweight Tunnel Protocol (LTP), Planos de dados programáveis

ABSTRACT

This project studies the applicability of low-cost programmable networks with transmission capacity very low data to transmit data on community networks and in use in hostile environments to wireless data transmission (for example, forests). Networks programmables separate the control plane from the data plane, which makes the control of the directly programmable network and molded according to abstraction and applications desired for the given network service. The focus will be on the LTP protocol (Lightweight tunnel protocol) which, when compressing the data packet header, searches performance increase in a transmission link, using an architecture Low cost LPWAN.

As it is a new protocol, its limitations are not yet applicable, so hypotheses are raised about its potential: on how to increase scalability, topology changes, how it handles packet loss and whether internet downloads are performed successfully if there is packet loss. The differentiated experiments that is a viable protocol to be implemented in community networks, demonstrating a advantage over a standard structure without using LTP protocol, in addition to being able to handle with a percentage of losses of up to 5%, an acceptable value considering its environment wireless. It was also noticed that in relation to increasing scalability, the proposal of increasing the number of bytes of the header was not viable because it burst the device device logger and TAG, but could in new work a alternative to increase scalability using a control type other than a register.

Keywords: LTP (Lightweight Tunnel Protocol), Low Power Wide Area Networks (LPWAN), Programmable Forwarding Planes, P4, Community Networks, Lightweight Tunnel Protocol (LTP)

LISTA DE ABREVIATURAS E SIGLAS

LTP	Lightweight Tunnel Protocol
LPWAN	Low Power Wide Area Networks
BMv2	Behavioral Model version 2
P4	Programming Protocol-Independent Packet Processors
TCP	Transmission Control Protocol
UDP	User da*TAG*ram protocol
SDN	Software defined networking

LISTA DE FIGURAS

Figura 2.1. comparativo LPWAN com outras redes 16
Figura 2.2: Diferença entre rede tradicional e SDN, onde plano de controle se separa do plano de dados. 17
Figura 2.3: Representação do parser numa rede hipotética. 20
Figura 2.6: Exemplo de criação do cabeçalho Ethernet, seguido pelo uso da definição do cabeçalho, e organização dos campos de endereço destino, origem e ethertype. 20
Figura 2.3: Representação de uma máquina de estados hipotética dos estados definidos pelo programador sendo processador pelo parser. 21
Figura 2.5: Representação do fluxo de pacotes na arquitetura PISA. Fonte: Almeida, L. 2020. 22
Figura 2.6: Compilando programas na linguagem P4. 22
Figura 3.1: arquitetura conceitual de LPWAN programável. 26
Figura 3.2: cabeçalho do pacote LTP. 26
Fonte: Scheibe, André, et al., 2021. 26
Figura 3.3 Handshake do LTP. Fonte: Scheibe, André, et al., 2021. 28
Figura 3.4 Funcionamento do LTP: cenário usando dispositivo programável LPWAN e *LTP* em rede sem fio. 29
Figura 3.5: Log demonstrando funcionamento do protocolo enviando pacotes. Print do terminal com log aberto. 30
Figura 3.6: no print do log do controlador, pacote sendo enviado à porta 1 30
Figura 4.1: topologia considerada para cenários 1 a 5, sendo: hosts os h1-h10, switches os s1, s2 e s3, e s4 atua como hub. 33
Figura 4.2: Representação do cenário 6 com 2 dispositivos low-end programáveis e 5 hosts por switch.
Figura 5.1: plot os três fluxos do cenário 5, protocolo TCPs 36
Figura 5.2: plot os três fluxos do cenário 5, protocolo UDP 37
Figura 5.3: plot dos cinco fluxos do cenário 6, protocolo TCP. 37
Figura 5.4: plot dos quatro fluxos do cenário 1 (carga útil de 128 bytes), protocolo TCP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps) 38
Figura 5.5: plot dos quatro fluxos do cenário 1 (carga útil de 128 bytes), protocolo UDP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps). 38
Figura 5.6: plot dos quatro fluxos do cenário 2 (carga útil de 512 bytes), protocolo TCP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps). 39

Figura 5.7: plot dos quatro fluxos do cenário 2 (carga útil de 512 bytes), protocolo UDP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps).. 39
Figura 5.8: cenário 1 (carga útil de 128 bytes), protocolo UDP, link de 256 kbps. 39
Figura 5.9: cenário 2 (carga útil de 512 bytes), protocolo UDP, link de 256 kbps comparando o desempenho com LTP e STD. A proporção está reduzindo, comparando com a figura anterior. 40
Figura 5.10: cenário 3 (carga útil de 1024 bytes), protocolo UDP, link de 256 kbps comparando o desempenho com LTP e STD. 40
Figura 5.11: cenário 4 (carga útil limitada ao protocolo MTU), protocolo UDP, link de 256 kbps comparando o desempenho com LTP e STD.....40
Figura 5.12: cenário 7. 41
Figura 5.13: medindo as taxas de transferências em tempo real.. 41
Figura 5.14: cenário 8, download simultâneo por 8 hosts por 256 kbps. 43
Figura 5.15: cenário 9 com payload de 512 bytes.. 44
Figura 5.16: cenário 9 com payload de 512 bytes.. 45
Figura 5.17: cenário 9 com payload de 1024 bytes.. 45
Figura 5.18: cenário 9 com payload de MTU bytes. 45
Figura 5.19: cenário 9 com TCP e payload de 128 bytes. 46
Figura 5.20: cenário 9 com TCP e payload de 512 bytes. 46
Figura 5.21: cenário 9 com TCP e payload de 1024 bytes.. 46
Figura 5.22: cenário 9 com TCP e payload de MTU bytes 47
Figura 5.23: cenário 10 e compara download LTP e STD com 1% de perda de pacotes.. 47
Figura 5.24: cenário 10 e compara download LTP e STD com 2% de perda de pacotes. 48
Figura 5.25: cenário 10 e compara download LTP e STD com 5% de perda de pacotes. 48

SUMÁRIO

AGRADECIMENTOS 3
1 INTRODUÇÃO 10
1.1 Definição do Problema 10
1.2 Objetivos 11
1.3 Organização do Trabalho 12
2 FUNDAMENTOS 13
2.1 Redes Comunitárias 13
2.2 Trabalhos Relacionados 14
2.3 Rede LPWAN 16
2.4 SDN 17
2.5 P4 18
2.5.1 Arquitetura PISA 20
2.5.2 Mininet 23
3 LIGHTWEIGHT TUNNEL PROTOCOL 25
3.1 Formato do header LTP 26
3.2 LTP em relação às portas dos switches 29
4 CENÁRIOS 32
4.1 Cenários Iniciais 32
4.2 Hipóteses de cenários de teste 34
4.2.1 Mudança na topologia 35
4.2.2 Perda de pacotes 35
4.2.3 Download com perda de pacotes 36
4.2.4 Maior escalabilidade 36
5 RESULTADOS 38
5.1 Resultados cenários iniciais 38
5.2 Resultados Cenários Novos 45
5.2.1 Resultado Mudança na topologia 45
5.2.2 Resultado Perda de pacotes 46
5.2.3 Resultado download com perda de pacotes 49
6 CONSIDERAÇÕES FINAIS 52
REFERÊNCIAS 54

1 INTRODUÇÃO

A inclusão digital é o processo de democratização do acesso às tecnologias da informação, permitindo a todos a inserção no mundo digital. Inclusão digital é também simplificar a rotina diária, maximizar o tempo e as potencialidades, com a finalidade de melhoria das condições sociais, visto que inclusão digital está relacionada com a inclusão social. Assim sendo, redes comunitárias são um veículo de transformação que aumenta a participação de todos os membros da comunidade, e são estruturadas para serem abertas, livres e respeitar a neutralidade da rede, sendo uma propriedade coletiva da comunidade (Wanzinack et al, 2013).

1.1 Definição do Problema

As alternativas existentes para solucionar o problema de exclusão digital, são, na maioria das vezes, de implantação e manutenção com custos muito elevados o que as torna inviáveis para pequenas comunidades, como a implantação da comunicação via satélite. Diante disso, inúmeras possibilidades estão sendo desenvolvidas no mundo, visando baixo custo para essa conectividade. Em Nepal, por exemplo, em parceria com o Instituto Nepal Wireless Project, que são doze escolas, dois centros de saúde e um hospital comunitário em funcionamento; na Índia, em Tilônia, com o projeto Wireless for Communities (W4C) em parceria com a Digital Empowerment Foundation; na África, com treinamento para capacitar interessados que se especializam em escalar redes comunitárias, implantando em lugares remotos como Kibera, uma comunidade carente que fica no Quênia (Brown, 2017).

Mas todas soluções estão em seus momentos iniciais e possuem suas dificuldades, principalmente em relação à falta de apoio por parte dos reguladores e formuladores de políticas, altas taxas de importação e taxas alfandegárias sobre equipamentos de telecomunicações e dispositivos de usuários, altas taxas regulatórias sobre compra e uso de equipamentos sem fio isentos de licença, longos períodos de espera e custos para obter as permissões e licenças para implantar e operar tais redes (Brown, 2017).

Uma possível solução para conectar redes comunitárias e, desse modo, dimensionar o acesso de informações a comunidades globais, é se utilizando de dispositivos programáveis de baixo custo. O artigo Programmable Low-End Networks: Powering Internet Connectivity

for the Other Three Billion concebeu o procolo LTP (Lightweight Tunnel Protocol), que propõe reduzir a sobrecarga de comunicação entre link e o cabeçalho de camada de pacote, enviando apenas um cabeçalho contendo um indentificador de TAG que se traduz em dados que os dispositivos podem usar para reconstruir os cabeçalhos originais, invés de cabeçalhos IP e Ethernet, em conjunto com a utilização de uma antena Ronoth LoStich como LPWAN Transceiver de baixo custo. Simulações já realizadas com ele mostram um grande potencial, funcionando com poucos hosts e enviando diferentes tamanhos de payload em diferentes links de transmissão, mostrando capacidade de ganho de desempenho ao reduzir bytes do cabeçalho.

Sendo um protocolo novo, criado em 2021, suas limitações ainda não são plenamente conhecidas e são necessários mais estudos em cima dele para testar sua viabilidade. Busca-se responder várias questões ainda não respondidas sobre sua aplicabilidade, como:

• LTP é um procolo apto a lidar com cenários em que acontece perda de pacotes nos links de tranmissão? Até que percentual de perda ainda é viável?

• Como aumentar a escalabilidade dele? Isso é possível de que modo?

• Em cenários já testados, apenas 1 host fazia download. Qual é o funcionamento do protcolo se mais hosts realizam download simultaneamente? E qual a quantidade máxima desses hosts? E funcionaria com perda de pacotes?

1.2 Objetivos

O foco do projeto é analisar sobre a aplicabilidade de redes programáveis de baixo custo para transmissão de dados em links de baixa capacidade. Logo, destacam-se alguns objetivos para a realização dessa tarefa:

• Estudo compreensivo do uso de tecnologias de baixo custo para levar internet para regiões carentes.

• Enumerar todas possíveis configurações de cenários, executá-las e fazer uma análise crítica dos resultados.

• Discussão filosófica, teórica e estudo acima desses resultados.

• Potencialidades e limitações da solução. Analisar se funcionalidade está adequada e a melhor aplicabilidade.

1.3 Organização do Trabalho

O presente trabalho de graduação está dividido da seguinte maneira. O Capítulo 2 apresenta os fundamentos do projeto, uma breve explicação sobre rede LPWAN, SDN e linguagem P4. Em seguida, é explicado o funcionamento do protocolo LTP no capítulo 3. O Capítulo 4 contém os cenários já existentes e a proposta de cenários novos, cujos resultados serão experimentados e discutidos no capítulo 5. Por fim, o capítulo 6 contém a conclusão.

2 FUNDAMENTOS

2.1 Redes Comunitárias

Em 2015 as Nações Unidas estabeleceram 17 objetivos globais – os Objetivos de Desenvolvimento Sustentável (ODS) – com o objetivo de alcançar coisas extraordinárias nos próximos 15 anos, incluindo o combate às injustiças e desigualdades, acabar com as mudanças climáticas, vencer a discriminação, trazer energia sustentável e garantir que ninguém passe fome. E o ODS 9 enfoca o papel importante que a infraestrutura e a conectividade desempenham na conexão dos lugares menos conectados do planeta, pois a internet é um facilitador para os objetivos se todos puderem acessá-la e se beneficiar dela. No entanto, cerca de quatro bilhões de pessoas ainda não têm acesso a internet e uma das questões centrais na agenda internacional da Internet e da Governança da Internet é conectar o próximo bilhão. McKinsey & Company (2014) identificaram quatro barreiras principais para a adoção da Internet: (1) Incentivos para ficar online; (2) Baixa renda e acessibilidade; (3) Capacidade do usuário; e (4) Infraestrutura. Isso é agravado pelo fato de que a falta de acesso à Internet é um fator-chave da desigualdade (ISOC 2017) (Brown, 2017).

A dificuldade de operação, manutenção da estrutura e elevado custo da infraestrutura e equipamentos encarecem e dificultam o acesso para os usuários. A exemplo, um gigabyte de dados na África do Sul custa sete vezes mais que no Egito e é três vezes mais cara do que no Quênia. Inúmeros esforços, públicos e privados, estão sendo realizados para viabilizar o acesso à internet a população desconectada, como o Projeto Loon (mantido pelo Google) (LOON, 2013), o 4Afrika (Microsoft) (4AFRICA, 2013), Internet.org (Facebook) (INTERNET.ORG, 2013), Alliance for Affordable Internet (A4AI) (A4AI, 2008) e o Internet para Todos (do Governo Federal) (MCTIC, 2018) almejam levar conectividade, à custo acessível, para longe dos grandes centros populacionais, já atendidos pelos meios convencionais de acesso (Scheibe, André, et al., 2019).

Uma das maneiras de fornecer esse acesso é por meio de redes comunitárias. As redes comunitárias são uma forma complementar - em vários setores, economias e tecnologias - de fornecer conectividade. Eles oferecem uma maneira para que qualquer pessoa, em qualquer lugar, possa se conectar à Internet, desde que tenha as ferramentas, as parcerias e o suporte

certos. Ao capacitar as pessoas em vilarejos carentes em todo o mundo para que se conectem a si mesmas e às suas comunidades - as redes comunitárias fornecem acesso onde as redes tradicionais ou comerciais não alcançam ou atendem, ou a áreas onde pode não ser economicamente viável operar. Eles oferecem uma alternativa complementar às redes de telecomunicações comerciais tradicionais. As redes comunitárias também são uma forma de desenvolver negócios futuros, criando comunidades "digitalmente experientes", ávidas por mais conteúdo local e serviços adicionais. Frequentemente, essas redes não são super hightech. Eles servem a um propósito orientado pela comunidade local para conectar-se dentro e da vila ou comunidade "para fora" (Brown, 2017).

Essas iniciativas locais de redes comunitárias são, em resumo, implementadas com intuito de fornecer conexão utilizando equipamentos de fácil operação, aquisição e manutenção de custo baixo e que permitam comunicação de dados à longas distâncias. Segundo (SONG, 2019), pequenas redes comunitárias organizadas em cooperativas possuem inúmeras vantagens como baixar os custos dos consumidores, os valores pagos são reinvestidos na própria rede, além do investimento local promover habilidades também locais, como técnicos para manutenção e operação (Scheibe, André, et al., 2019).

2.2 Trabalhos Relacionados

As principais redes comunitárias já existentes se utilizam de tecnologias como Cognitive Radio (CR) e TV White Spaces (TVWS). CR é um rádio adaptável que aumenta a eficiência do espectro por meio do ajuste em tempo real dos recursos de rádio, assim se utilizando de frequências sem fio disponíveis em sua vizinhança para evitar interferência e permitir comunicações simultâneas. CR e TVWS geralmente exigem licenciamento para operação e equipamentos que podem não ser acessíveis para algumas comunidades **(**Scheibe, André, et al., 2019).

O TVWS, por sua vez, refere-se aos canais de TV não utilizados para o espectro VHF e UHF. No passado, os canais não utilizados eram colocados entre os ativos para proteger contra interferência de transmissão, mas hoje podem fornecer links de comunicação de banda larga (e acesso à Internet) enquanto operam sem afetar os canais de TV. Assim sendo, regiões em desenvolvimento e rurais têm sido um caso de uso chave para o TVWS. Apesar de suas potencialidades, Dentre os projetos que utilizam a tecnologia TVWS para comunicação de

dados, tem o 4Africa (4AFRICA, 2013), uma iniciativa da Microsoft que em parceria com empresas e universidades da África, entre outros serviços, fornece internet acessível para as pessoas desconectadas. Dois exemplos de sucesso desta parceria são os projetos com a Spectra Wireless (SPECTRA, 2015) em Gana, que no lançamento do serviço já atendia cerca de 3500 estudantes e com a Mawingu Networks (MAWINGU, 2013) que já atende mais de 27000 usuários. Além de Gana, Malawi e Zâmbia também já recebem sinal oriundo da comunição de dados utilizando TVWS. As principais características positivas do uso da tecnologia no local são a fácil manutenção e a possibilidade de utilizar comunicação através de TVWS mesmo em distâncias de 15 km sem visada. No mesmo continente, a TENET (TENET, 2000) utiliza TVWS e fibra optica para interligar as universidades e instituições de ensino pública da África do Sul, não fornecendo comunicação comercialmente para usuários domésticos.

Analisando, entretanto, o uso da tecnologia TVWS, (VÁZQUEZ, 2017) destaca que, mesmo sendo uma ótima alternativa para criação de redes comunitárias sem fio, pois trabalha muito bem com obstáculos físicos comuns em longas distâncias, existe, além do risco de espectros não homologados voltarem a ser utilizados e a comunicação para internet ser interrompida, também a falta de equipamentos homologados e a custo acessível. Segundo a matéria, nos Estados Unidos existem apenas 800 equipamentos compatíveis com a tecnologia TVWS e, devido a falta de investimento nesta área, os equipamentos homologados custam cerca de mil dólares.

Então LPWAN pode se tornar uma tecnologia interessante para implantar redes comunitárias, seja substituindo ou complementando CR e TVWS, principalmente por causa de seu baixo custo e fácil implantação. Tem havido investigações sobre o uso de Rede Definida por Software (SDN) para gerenciar LPWAN e, em geral, eles se preocupam em fornecer uma interface aberta para o gerenciamento dos dispositivos com recursos restritos que compõem a rede. Apesar de ter havido pesquisas sobre a programação do plano de encaminhamento no contexto de LPWA para agregação de pacotes usando desagregação em IoT, ainda não foi abordado na literatura antecipar a programabilidade do plano para LPWAN, como o protocolo LTP se propõe.

2.3 Rede LPWAN

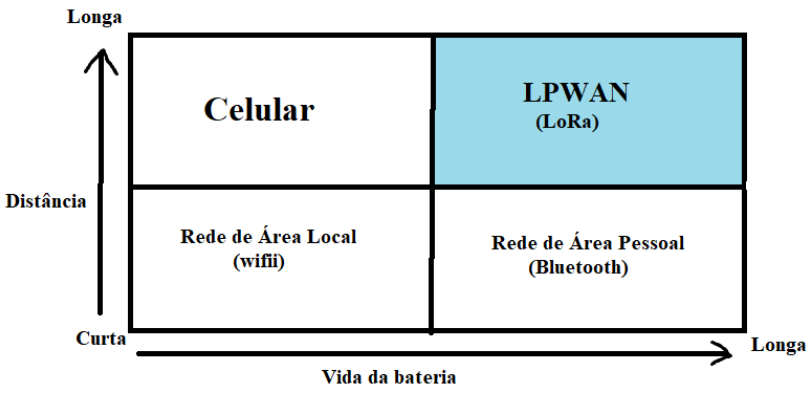
A rede LPWAN (*Low Power Wide Area Network*) é com frequência empregada em IoT (Internet of Things ou Internet das coisas) quando deseja-se transmitir, em distâncias relativamente largas, poucos dados, assegurando uma vida útil maior para as baterias a serem implementadas durante os processos de comunicação e aplicação, tendo surgido para conectar milhões unidades de valor agregado menor, com redes de cobertura melhor, custo de conexão menor e, desta forma, ampliando as possibilidades de soluções de IoT.

As características principais do LPWAN são sua capacidade de pouco consumo de energia, o que permite os dispositivos durarem até dez anos com apenas uma carga; transferência de dados otimizada que suporta blocos de dados pequenos e intermitentes; baixo custo unitário do dispositivo; fácil instalação da rede; otimizado para baixo rendimento, longa ou curta distância; e penetração e cobertura interna suficiente (MACNICA, 2004).

A rede LPWAN utiliza topologia em estrela – eliminando a implementação de um complicado protocolo de roteamento de malha sem fio, conhecido também por rede Mesh – ou seja, os dispositivos que são integrados à tal rede são conectados diretamente ao ponto de acesso, como na figura 1, e consiste em vários cabos que unem cada dispositivo a um ponto central (MELO, 2017).

As redes celulares, por exemplo, são excelentes para cobrir grandes áreas geográficas e apoiar a mobilidade, projetadas e otimizadas para fornecer rendimento e segurança (AZUTON, 2019), no entanto, seus custos, complexidade e consumo de energia são inerentes a tecnologia, inviabilizando projetos em muitas das aplicações. Portanto estas características não se traduzem bem nas aplicações IoT com necessidade de baixos custos e larga escala, sendo necessária uma solução diferente para cumprir esse objetivo.

Figura 2.1. comparativo LPWAN com outras redes



Fonte: da autora

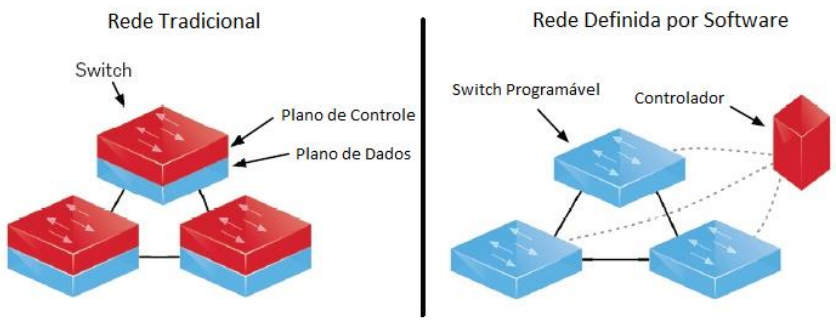
2.4 SDN

O projeto original do LTP usa uma antena LPWAN para transmitir internet e Rede Definida por Software (SDN) para gerenciar o encaminhamento de fluxo. LPWAN é uma tecnologia interessante para implantar redes comunitárias por causa de seu baixo custo e fácil implantação, além de alcançar distâncias grandes e usar pouca bateria. As redes definidas por software (SDN) separam o plano de controle do plano de dados com um controlador, essa estratégia permite a programabilidade do plano de controle da rede dos serviços de rede.

O SDN pode ser descrito como uma abordagem de rede que permite que os operadores de rede configurem, rastreiem, alterem e controlem programaticamente a operação da rede por meio de interfaces abertas, como o protocolo OpenFlow. O SDN transforma a operação, gerenciamento e configuração das infraestruturas de rede. A visão do SDN é baseada na separação do plano de dados do plano de controle. O SDN propõe concentrar a inteligência da rede em um único componente da rede, distinguindo o mecanismo de encaminhamento de pacotes de dados (plano de dados) do processo de routing (plano de controle), como visto na Fig.2.3 (Haji, Saad, et al, 2021).

O plano de controle é o responsável pelas informações da rede, ou seja, ele define as próximas rotas do fluxo de pacotes na rede, que podem executar, por exemplo, um protocolo de roteamento ou algo do gênero. O plano de dados é responsável pela transmissão dos pacotes pela rede, ou seja, apenas encaminha os pacotes ao seu destino, que é determinado pelo plano de controle (Lobato, Figueiredo, Alves, 2006).

Figura 2.2: Diferença entre rede tradicional e SDN, onde plano de controle se separa do plano de dados.



Fonte: PRAJAPATI, Arpita (2018)

Além disso, com a divisão dos planos, pode-se atingir altas taxas de pacotes retransmitidos, pois a partir do momento em que o fluxo é definido, os pacotes são simplesmente transformados pelo plano de dados, onde pode haver elementos especializado para esta função. “Com a retirada do plano de controle dos dispositivos, pode-se modelar e moldar dinamicamente a rede, de acordo com a necessidade do gestor, permitindo que a gerência da rede seja feita de forma muito mais fácil e eficiente.” (TELECO, 2017).

O paradigma de rede definida por software (SDN) torna possível separar o hardware e os programas executados nele. Essa estratégia promove a operação da rede porque melhora a programabilidade do plano de controle da rede. Recentemente, a programabilidade também foi estendida ao plano de dados. Para alcançar a programabilidade no hardware que reside no plano de dados, a linguagem P4 (Bosshart et al. 2014) foi criada para permitir que os administradores de rede definam o comportamento dos dispositivos intermediários. Um dos benefícios é que novos protocolos de rede totalmente personalizáveis podem ser facilmente criados e implementados sem depender da indústria para adicionar novos recursos ao comportamento do plano de dados (Parizotto, Castanheira, Schaeffer-Filho, 2019).

2.5 P4

Logo após surgirem as SDNs e protocolo Openflow em 2008, abriu-se oportunidades para realizar a programação o plano de controle de um dispositivo de rede. Por depender de uma flexibilização maior dos fabricantes, sobretudo na fabricação dos chips existentes nos dispositivos, programação no plano de dados sempre foi mais difícil (Almeida, 2020).

Em 2014, pesquisadores publicaram um artigo no qual foi proposto uma nova forma de programar o plano de dados de um dispositivo de rede. Este artigo foi entitulado como: "*P4: Programming protocol-independent packet processors*", (Almeida, 2020). A linguagem P4 descreve como um pacote deve ser processado pelo plano de dados de um dispositivo programável de encaminhamento, que pode ser implementado em switches por hardware ou software, placas de rede, roteadores ou dispositivos programáveis do tipo FPGA. Inicialmente, a linguagem P4 foi desenhada para atuar em switches programáveis, no entanto, seu escopo acabou sendo estendido para cobrir uma ampla variedade de dispositivos, que no escopo da linguagem, passaram a ser chamados de "alvos" (ou targets). A P4 é desenhada especificamente para descrever o plano de dados do alvo e também define a interface de comunicação com o plano de controle. No entanto, a linguagem P4 não pode ser usada para descrever as funções do

plano de controle de um alvo. Por esse motivo, sempre que se refere a programar um alvo, entende-se programar o plano de dados (Parizotto, Castanheira, Schaeffer-Filho, 2019).

Reconfigurar e programar a rede habilita a criação de novos protocolos e novas formas de encaminhar os dados, que poderão ser testadas e implementadas rapidamente, pois não dependem mais de modificações no firmware do switch ou do roteador. Além disso, o controlador pode ter o software atualizado e alterado sem haver perda de pacotes em uma rede em produção. Além disso, devido ao controle global, seja a falha em um link ou em um nó da rede, a rede se reorganizará e convergirá mais rapidamente para uma nova ótima configuração, devido a utilização de algoritmos centralizados ao invés de algoritmos distribuídos (TELECO, 2017). Para isso, é importante desenvolver técnicas abrangentes que habilitem a reconfiguração da rede ocorrer de forma mais rápida e sem corromper propriedades básicas de operação (Parizzoto, 2020) Alguns trabalhos recentes propõem a utilização de P4 para configurar funções virtualizadas no próprio plano de dados (Hancock and van der Merwe 2016, Zhang et al. 2017), entretanto essas propostas pecam por dois fatores, um deles sendo em escalabilidade, devido ao uso excessivo de tabelas de controle e primitivas de recirculação de pacotes, atrasando o processamento e encaminhamento dos pacotes; (Parizotto, Castanheira, Schaeffer-Filho, 2019) e o outro fator sendo o não fornecimento de isolamento necessário para as funções (Dimitropoulos et al. 2018). Com isso, fica evidente que são necessárias abstrações e estratégias que permitam que os administradores de rede possam implantar novas funcionalidades nos seus dispositivos programáveis, sem que isso impacte negativamente no desempenho das funções de rede (Parizotto, Castanheira, Schaeffer-Filho, 2019).

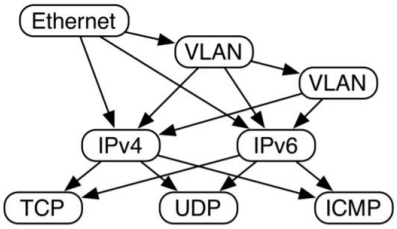
P4, sendo uma linguagem de alto nível, fornece comandos simplificados para descrever o caminho dos da*TAG*ramas, podendo trabalhar em conjunto com protocolos de controle SDN (Bosshart et al. 2014). Com a proposta do P4, busca-se a evolução do modelo por meio da reconfigurabilidade no campo onde os programadores devem ser capazes de mudar a forma como os switches processam os pacotes depois de implementados (Camera e Zanetti, 2020), independência do protocolo e a do alvo, pois os comutadores não devem estar vinculados a nenhum protocolo de rede específico e os programadores devem ser capazes de descrever a funcionalidade de processamento de pacotes independentemente das especificidades do hardware subjacente (Kreutz, Mansilha, Miers, 2019).

2.5.1 Arquitetura PISA

Em P4, foi definida uma arquitetura de switch programável chamada de PISA: Protocol-Independent Switch Architecture. A arquitetura PISA está dividida em 3 partes: Parser, Pipeline match-action e Deparser (Almeida, 2020). A seguir, os componentes principais do P4 associados ao modelo PISA:

- Parser programável: tem por função identificar, em uma cadeia (stream) de bits, os cabeçalhos que se encontram presentes no pacote. Ou seja, consegue especificar o formato para processamento e identificar os protocolos dentro no pacote. A definição de parser especifica como identificar (reconhecer) cabeçalhos ou sequências de cabeçalho válidos nos pacotes. Os estados definidos pelo programador descrevem como o pacote será processado pelo parser (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014)

Figura 2.3: Representação do parser numa rede hipotética.



Fonte: GIBB, G. et al.

- *Headers*: A definição do cabeçalho descreve a sequência e estrutura de uma série de campos de bits definidos pelo programador. Inclui especificações de largura, restrições de tipos e valores (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014)

Figura 2.6: Exemplo de criação do cabeçalho Ethernet, seguido pelo uso da definição do cabeçalho, e organização dos campos de endereço destino, origem e ethertype.

```
typedef bit<48> macAddr_t;
typedef bit<32> ip4Addr_t;

header ethernet_t {
    macAddr_t dstAddr;
    macAddr_t srcAddr;
    bit<16>   etherType;
}
```

```
header Ethernet_h {
    bit<48> dstAddr;
    bit<48> srcAddr;
    bit<16> etherType;
}
```

```
header Tcp_h { ... }
header Udp_h { ... }
struct Parsed_headers {
Ethernet_h ethernet;
Ip_h       ip;
Tcp_h      tcp;
Udp_h      udp;
}
```

Fonte: Almeida, L. 2020.

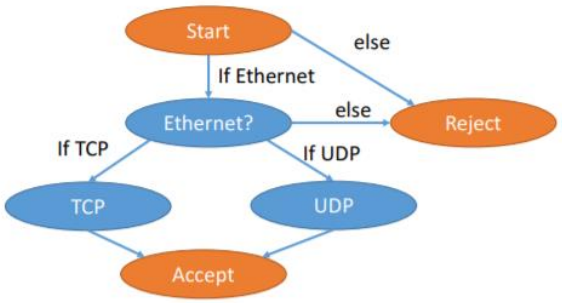
- *Match-Action*: nessa etapa faz-se um match, ou seja, a correspondência entre um campo e um valor, e executa-se uma ação de acordo com as entradas que se encontram associadas a esses matches nas tabelas; (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014).

- *Tables*: As tabelas *match-action* são o mecanismo para realizar o processamento de pacotes, nas quais são associadas ações (actions) aos pacotes de acordo com o matching realizado nos parsers (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014).

- *Actions*: P4 suporta construção de ações complexas construídas usando primitivas simples e independentes de protocolo. Em geral, o parser identifica os headers presentes em cada pacote ingressando no dispositivo. Cada tabela match-action realiza uma busca (lookup) em um subconjunto de campos de cabeçalho e aplica as ações correspondentes ao primeiro *matching* encontrado na tabela (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014).

Em resumo, o parser é máquina de estados finita, na qual o programador define (deixa explícito) quais cabeçalhos de rede devem ser reconhecidos pelo dispositivo. No Pipeline o programador define as tabelas contendo regras de correspondência (*match*) e as respectivas ações (*actions*) em um algoritmo. No Deparser o programador define como o pacote deve ser remontado para ser encaminhado na rede (Almeida, 2020).

Figura 2.3: Representação de uma máquina de estados hipotética dos estados definidos pelo programador sendo processador pelo parser.

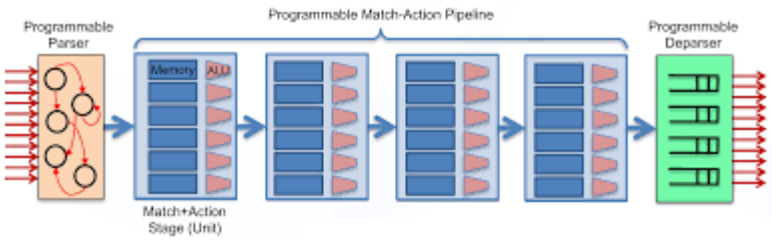


Fonte: Sudonull company

A Figura 3.6 mostra de forma diagramática o conceito do envio de dados dentro do modelo P4/PISA. Como consta no material de (Kreutz, Mansilha, Miers, 2019): primeiramente, os pacotes são manipulados pelo parser. Este reconhece e retira campos do header e, assim,

define os protocolos suportados pelo comutador. Os campos extraídos são passados para as tabelas de match-action, divididas entre entrada (ingress) e saída (egress). Embora ambas possam modificar o header, a ingress match-action determina a(s) porta(s) de egress e a fila na qual o pacote é colocado. Com base nesse processamento, o pacote pode ser encaminhado, replicado (para multicast, span ou até ao plano de controle), descartado ou mesmo acionar o controle de fluxo. A egress *match-action* executa modificações por instância no header, por exemplo, as cópias multicast. As tabelas de ação (contadores, policers e assim por diante) podem ser associadas a um fluxo para rastrear o estado frame-to-frame, bem como informações de metadados que ofertam detalhes entre os estágios percorridos.

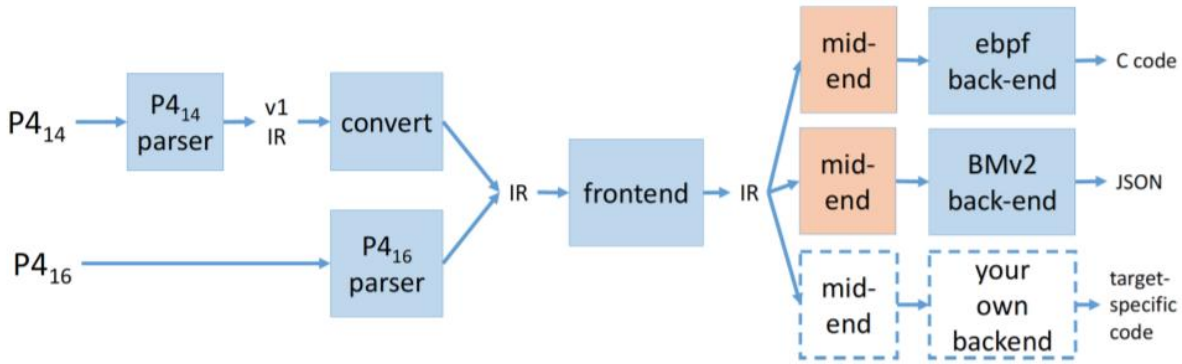
Figura 2.5: Representação do fluxo de pacotes na arquitetura PISA.



Fonte: Almeida, L. 2020.

Fluxo dos pacotes na arquitetura PISA: O parser identifica os cabeçalhos presentes em cada pacote ingressando no dispositivo. Realiza uma busca (lookup) em cada tabela *match-action* por um subconjunto de campos de cabeçalho e aplica as ações correspondentes ao primeiro *matching* encontrado na tabela (Kreutz, Mansilha, Miers, 2019).

Figura 2.6: Compilando programas na linguagem P4.



Fonte: Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014

Dentre as plataformas usadas para testes e simulação para criar provas de conceito está o BMv2, que é a implementação do "*Behavioral Model*" da arquitetura PISA, e esse por sua vez implementa um modelo simples de switch no backend específico do compilador: p4c-bm (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014).

Transforma-se o código P4 em uma representação JSON que pode ser consumida pelo software switch; esta representação dirá ao BMv2 quais tabelas inicializar e como configurar o parser (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014).

Assim que é compilado o programa P4, o backend será o responsável pela programação ou configuração do plano de dados apropriado (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014). Por exemplo, no caso do BMv2, ocorrerá a transformação do programa P4 em um descritivo JSON que é a saída do backend. Esse JSON é carregado então no alvo (BMv2) e inicializa as estruturas de dados que refletem o comportamento de encaminhamento desejado (Material de *tutorial P4.org: Lab 2: P4 Runtime*). Por sua vez, o arquivo JSO será a entrada para o switch BMv2 que na entrada, é responsável por fazer a segmentação dos pacotes, que segue a orientação das especificações no arquivo de configuração JSON, e populará as tabelas do pipeline do switch PISA de referência. Este processo todo é específico do BMv2 (Garcia, Villaça, Riberiro, Martins, Verdi e Marcondes, 2014).

2.5.2 Mininet

O mininet, sendo um emulador de rede, é capaz de criar redes com switches, controladores, servidores, links virtuais, aplicativos, uma infraestrutura de rede virtual em grande escala que permite customização e compartilhamento com outros usuários, e implementação em hardware real, em um único computador rodando em um único núcleo físico (kernel Linux), com sua própria linha de comando *CLI (Command Line Interface*) e API. Código ou máquina virtual (*Virtual Machine*-VM), pode ser nativo ou na nuvem.

Mininet é uma ferramenta útil de desenvolvimento (Silvério, Campista e Costa, 2016), aprendizagem e pesquisa, ideal para experimentos com OpenFlow e SDN. A principal limitação da rede simulada no Mininet é que a largura de banda disponível e a capacidade da CPU não podem exceder a capacidade disponível e a largura de banda no servidor onde o Mininet está instalado, e, também, não executa aplicações que não sejam compatíveis com o Linux.

Como a emulação possui suas limitações (Camera e Zanetti, 2019), é necessário determinar um padrão de procedimento fiel a uma rede P4 real. Para tal, existe a ajuda do BMv2 (*Behavioral Model version* 2), que é o compilador do software switch nativo do P4 e estabelece

o comportamento dos switches P4 dentro da estrutura sendo emulada, assim reproduzindo o plano de dados programável. Também tem O framework P4Runtime que desempenha as funções do plano de controle, gerenciando os elementos do plano de dados que foram definidos em P4.

3 LIGHTWEIGHT TUNNEL PROTOCOL

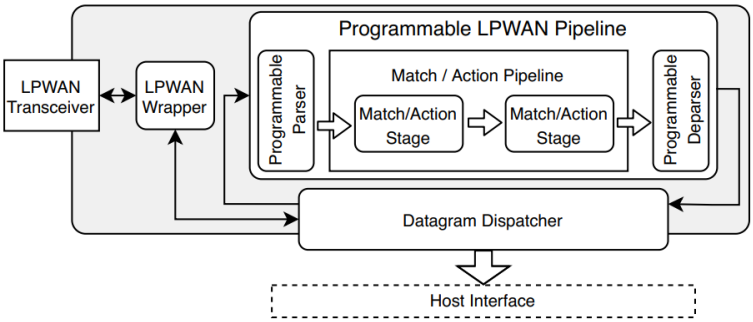
Com base especialmente no artigo intitulado Programmable Low-End Networks: Powering Internet Connectivity for the Other Three Billion, é proposta uma arquitetura conceitual programável e um novo e inovador protocolo de comunicação, Lightweight Tunnel Protocol (LTP), que maximiza o goodput em dispositivos de rede low-end com taxa de transmissão limitada, com bom desempenho para fornecer conectividade com a internet em coexistência com protocolos TCP/IP tradicionais, com ganhos de taxa de transferência de até 23%, possibilitando dispositivos LPWAN serem mais facilmente integrados às redes sem a necessidade de dispositivos especializados, como gateways, fornecendo vários benefícios como eficiência a baixo custo, segurança, baixo consumo de energia, baixa complexidade que mantém uma infraestrutura simples e escalável (Scheibe, André, et al., 2021).

LTP busca explorar o poder da programabilidade LPWAN para projetar um protocolo simples, mas funcional, para redes de baixo custo – promovendo a inclusão digital – e está fora de seu escopo projetar um protocolo superior às soluções de ponta para a otimização da transmissão de dados (como a família de soluções baseadas em *ROHC – Robust Header Compression*) e deixa este aspecto, bem como segurança e confiabilidade para LTP, para pesquisas futuras em redes programáveis de baixo custo para que se tornem uma opção viável de conexão remota. E é o que o atual projeto de monografia está interessado, estudar a aplicabilidade do protocolo e testar melhores soluções e cenários com sua utilização.

LTP é um protocolo que foi concebido para que reduza a sobrecarga de comunicação entre link e o cabeçalho de camada de pacote, em vez de enviar cabeçalhos Ethernet e IP para cada pacote de um fluxo TCP/IP, pode-se enviar um cabeçalho contendo apenas um identificador de *TAG*, que se traduz em dados que os dispositivos podem usar para reconstruir os cabeçalhos originais, contando com a noção de túneis virtuais entre pares de dispositivos. Por este motivo, seus pacotes se concentram em comunicação de salto único, assim como Ethernet e suporta qualquer protocolo de camada superior. Para mantê-lo simples, assume-se que recursos como retransmissão de pacotes e soma de verificação já são tratados por esses protocolos. Neste contexto, LTP é semelhante a qualquer protocolo L2, fornecendo comunicação de dados sobre um canal virtual entre pares de dispositivos.

Para programação no plano de dados, o protocolo LTP, como todos programas P4, é observada uma ordem de operações. No parser será extraído os dados dos pacotes, é processado pelo ingress e egress e, então os cabeçalhos serão recomentados pelo deparser. Um tipo de informação diferente é derivado de cada estado do parser.

Figura 3.1: arquitetura conceitual de LPWAN programável.



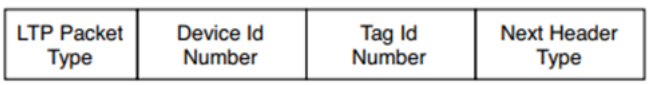
Fonte: Scheibe, André, et al., 2021.

Considerando a arquitetura da fig 4.1, um desenvolvedor pode escrever um programa P4 que redefine a semântica de análise, o processamento de pacotes do pipeline LPWAN e usar o compilador do P4 para gerar o programa/API responsável pelo comportamento do dispositivo. Utilizando-se dessa arquitetura conceitual, foi projetado e implementado o LTP, um protocolo de rede para dispotivos low-end.

No projeto de (Scheibe et al, 2021), é utilizado o Ronoth LoStich como LPWAN Transceiver, um programa python personalizado que interage com LoStick para programar, ler e gravar dados, e a chave de software bmv2 para emular um Pipeline LPWAN programável.

3.1 Formato do header LTP

Figura 3.2: cabeçalho do pacote LTP.



Fonte: Scheibe, André, et al., 2021.

O cabeçalho do pacote LTP, dentro do código, contém tamanho de 4 bytes distribuídos em 4 campos de um byte cada sendo, o preâmbulo, definido como Ltp_cpu ou ltp_def (identifica o tipo de cabeçalho LTP); a identificação do dispositivo Dev_ID, o identificador sequencial da TAG criada naquele dispositivo de origem, TAG_ID, e protocolo transporte, que possui a mesma informação do campo de protocolo de transporte que conta no cabeçalho IPv4 original. O Preâmbulo ou LTP Packet Type possui valores FE e FD (hexadecimais) que não são utilizados por prefixos de MAC válidos. E possui o seguinte formato:

```
header TAG_t {
        bit<8> preambulo;
        devId_t dev_id;
        TAGId_t TAG_id;
        bit<8> proto;
}
```

Para diferenciar pacotes TCP/IP de pacotes LTP, foi criado o campo LTP packet type, que contém algum valor que parser vai reconhecer como LTP invés de quadro Ethernet, usando comando lookahead() do P4, e esse valor não pode colidir com valores válidos de prefixo do endereço MAC. Com esse comando, o dispositivo consegue diferenciar um pacote com uma instrução (LTP CPU) onde encaminha para o controlador para separar a instrução do resto do pacote que será capaz de ser reconhecida no dispositvo destino quando processada ou se é apenas um pacote com dados que deve ser tratado. Além disso, também é necessário um campo para diferenciar túneis LTP de pares de hosts diferentes, e para isso tem o campo identificador de dispositivo Devide ID.

A combinação do número do dispositivo que originou o tráfego, DEV ID (Device Identification), e um número de evento sequencial *TAG* (representando cada combinação fonte/destino) garantem que este identificador não será repetido (e não foi usado antes). Uma vez que o fluxo é identificado com uma *TAG* única, será possível controlar todos dispositivos se o processamento desta mudança de rede já foi realizado.

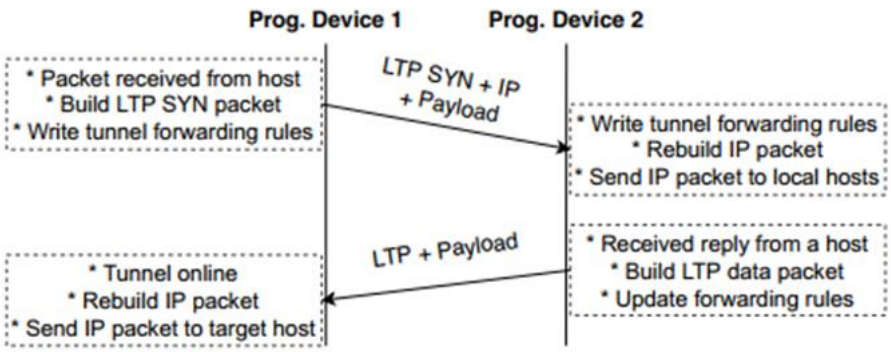
Devido a limitada capacidade de transmissão dos dispositivos e também do link de dados, não é possível criar uma conexão segura entre os controladores. Sendo assim, o sincronismo entre os controladores precisa ocorrer utilizando o link de dados. Para cada nova conexão estabelecida, o dispositivo de origem cria uma nova *TAG* sequencial junto com seu identificador de dispositivo e precisa propagar esta regra para todos os demais dispositivos.

Deste modo, foi necessário criar um mecanismo que permitisse cada dispositivo identificar o recebimento do registro de criação de uma nova regra, de um novo túnel LTP.

Assim o que ocorre é um esquema de handshake no protocolo LTP para hosts que ainda não trocaram dados, sendo criado um túvel LTP antes que qualquer envio de dados possa ocorrer, pois na primeira vez que o pacote de um determinado host chega, o cabeçalho completo com *TAG*, ethernet e ipv4 também é enviado ao switch para saber da substituição que ocorrerá, e a partir do próximo pacote é enviado apenas *TAG* e o payload de dados, assim proporcionando melhores van*TAG*ens em um ambiente onde link de transmissão é baixo, transmitindo o máximo com menor custo possível.

Ao receber um pacote TCP/IP, um identificador *TAG* é gerado e é enviado um LTP SYN conténdo um identificador do dispositivo Dev id, sendo SYN os pacotes de sincronismo que são o primeiro passo para iniciar qualquer conexão numa rede TCP/IP. O destino cria as tabelas e regras do pacote TCP/IP recebido, extraindo o cabeçalho LTP e Dev id e encapsula o cabeçalho IP e endereços de hosts origem e destino, enviando para o controlador. Então é reenviado o novo pacote para host destino.

Figura 3.3 Handshake do LTP.

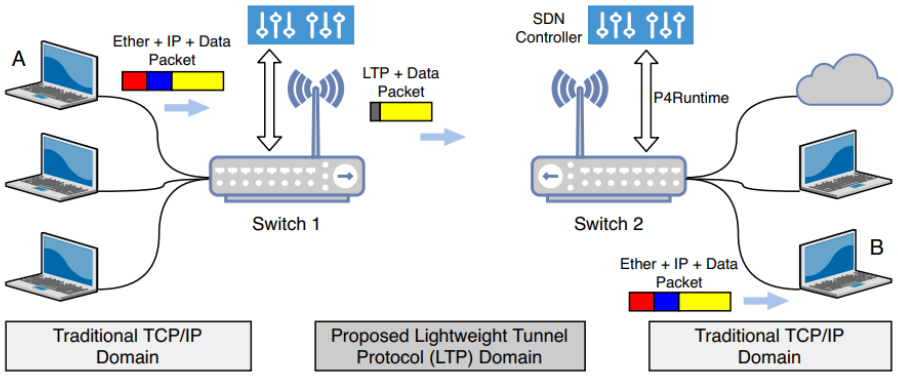


Fonte: Scheibe, André, et al., 2021.

Quando um host origem A envia um pacote ao host destino B com LTP, através de dispositivo programável LPWAN, os dispositivos no plano de dados dependerão somente de uma tabela de fluxos. Essas tabelas de fluxo atuam examinando o cabeçalho dos pacotes recebidos bit a bit, lendo informações como IP de origem e destino, portas de origem e destino, protocolo, dentre outros, e comparando-os com os valores presentes na tabela. Quando os valores encontrados forem compatíveis, ações podem ser tomadas para aquele fluxo, com estas geralmente sendo encaminhá-lo por um caminho específico ou descartá-lo. Outras ações incluem mandá-lo para a próxima tabela de fluxos ou, caso seja um fluxo novo, enviá-lo ao controlador para descobrir que ação deve ser tomada para casos

similares. Quando chega na tabela match+action para examinar, tratar e direcionar os fluxos recebidos, o endereço IP de A e B é traduzido em uma *TAG* que identifica o par – caso o cabeçalho *TAG* ainda não exista, ela é construída após análise dos cabeçalhos ethernet e IP, que também vai conter o payload e é enviada via antena LPWAN. Após receber o pacote LTP, o dispositivo analisa a *TAG*, acha o par correspondente de endereços IP dos hosts A e B nas tabelas de match+action e reconstrói o cabeçalho ethernet e IP, reenviando o pacote TCP/IP para host B.

Figura 3.4 Funcionamento do LTP: cenário usando dispositivo programável LPWAN e *LTP* em rede sem fio.



Fonte: Scheibe, André, et al., 2021.

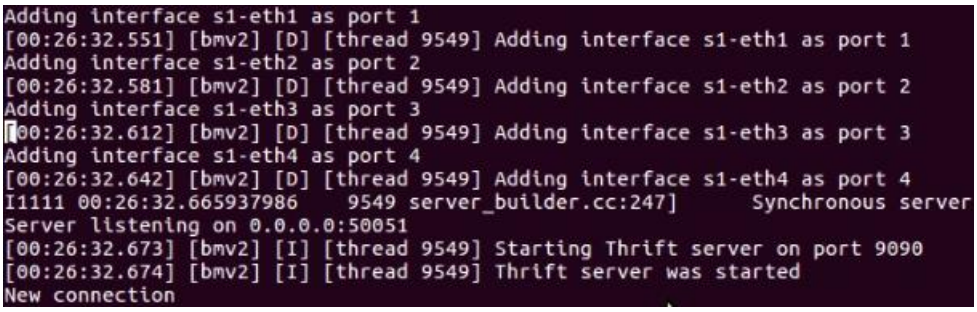
O protocolo LTP foi contruído com linguagem P4, simulado com mininet e switch de software bmv2. O controle SDN foi escrito em python2.7 com scapy para parsing do pacote. E iperf3 para gerar os fluxos nos switchs e hosts.

3.2 LTP em relação às portas dos switches

Considerando a topologia da figura 5.1, o caminho do pacote é: sai do host A (s1), chegou no switch, como não tem a tradução ainda no primeiro envio, é enviado para a cpu (controlador), onde são criados registros nas tabelas (o controlador do switch cria a regra de tradução, tanto para envio, quanto para retorno) e o pacote é devolvido na porta ingress do switch novamente como se pacote tivesse sido enviado de novo pelo host. Agora, com as regras criadas, é possível encontrar um registro e realizar match+action, realizando a tradução dos cabeçalhos pela TAG criada. No primeiro envio ao switch de destino, chega o cabeçalho traduzido e o cabeçalho ipv4/ethernet para que este possa identificar que está recebendo um

novo registro e possa encaminha-lo para ser tratado pelo seu controlador e também realizar os registros nas suas tabelas.

Figura 3.5: Log demonstrando funcionamento do protocolo enviando pacotes. Print do terminal com log aberto.



Fonte: da autora.

Pacote 0 => pacote recebido na porta 1 (antena) => é o primeiro pacote do ping oriundo do S1. Como não existe regra salva, preciso enviar para CPU.

Figura 3.6: no print do log do controlador, pacote sendo enviado à porta 1

```
Pipeline 'egress':start
Pipeline 'egress': end
Deparser 'deparser': start
Updating checksum 'cksum'
Deparsing header 'tag'
Deparsing header 'ethernet'
Deparsing header 'ipv4'
Deparser 'deparser': end
Transmitting packet of size 102 out of port 1
```

Fonte: da autora.

Pacote 1 => O pacote que voltou do controlador (CPU) após criar a regra provisória pois não se sabe se o destino está atrás deste switch. É replicado para todas as portas.

Figura 3.6: a porta 1, que é a da antena e comunica com os demais switches setada como egress, e transmite o pacote que recebeu

```
Pipeline 'ingress': end
Egress port is 1
Pipeline 'egress':start
```

Fonte: da autora.

Pacote 2 => resposta recebida na porta 2, correspondente ao pacote 1 (onde está o destino). Como a resposta, agora destino é conhecido e pode salvar as regras definitivas, então é enviado para CPU. No trecho printado da figura 3.8, tem um exemplo de tradução selecionado: Dispositivo 1, *TAG* 1, na porta 1 de saída.

Figura 3.7: Após o envio da porta 1 na porta 2, o pacote 2 é basicamente uma réplica do pacote 0, faz o parser, a chave é 0002 pois a porta é 2 e faz o lookahead().

```
Transmitting packet of size 102 out of port 1
Processing packet received on port 2
Parser 'parser': start
Parser 'parser': entering state 'start'
Parser state 'start': key is 0002
Bytes parsed: 0
Parser 'parser': entering state 'parse_ethernet_tag'
```

Fonte: da autora.

Figura 3.8: O pacote 2 já faz a tradução.

```
hdr.ethernet.ethsrcAddr : EXACT    080000000111
hdr.ipv4.srcAddr        : EXACT    0a000101
hdr.ipv4.dstAddr        : EXACT    0a000201
action_entry: MyIngress.tag_build: 1.1.1.
```

Fonte: da autora.

Pacote 3 => Novo pacote oriundo da antena (réplica do pacote 0), Como não tem regra salva, o comportamento é o mesmo do pacote 0, e precisa replicar em todas as portas. Pacote 4 => Nova resposta da porta 2 (onde está o destino), correspondente ao pacote 3. Como a regra definitiva ainda não foi salva, é enviado novamente para CPU. Neste caso, a regra foi salva.

Pacote 5 => O pacote 4 que foi enviado para CPU, volta da CPU após as regras serem salvas. Agora possui registro match+action na tabela e é possível enviar corretamente ao destino. Transmitido na porta 1 (antena) para ser recebido no destino. Este é o primeiro pacote de retorno que é visualizado no switch 1.

Pacote 6=> Oriundo da antena. Como agora as regras estão salvas, nao é mais necessário transmitir para CPU nem para todas as portas (replicas). Então envia apenas para porta destino salva anteriormente.

4 CENÁRIOS

A subseção 4.1 retrata os cenários já existentes no repositório do protocolo LTP, nela foram reproduzidos novamente todos cenários para ver seu comportamento e, a partir dele, serem criadas as ideias de modificações para novos cenários. Como objetivo do projeto do protocolo LTP é levar internet para comunidades de baixíssima renda, os testes a princípio foram limitados, com um host fazendo um download. Para melhor entender sobre a aplicabilidade e limitações desse protocolo, é gerada a construção de novos cenários de teste, identificando as condições de contorno e o trade off na sua utilização. Realizando alterações na estrutura, incluindo variáveis como perda de pacotes, mudando topologia e combinações dentro de cada dispositivo, com o objetivo de analisar o impacto que cada mudança provoca.

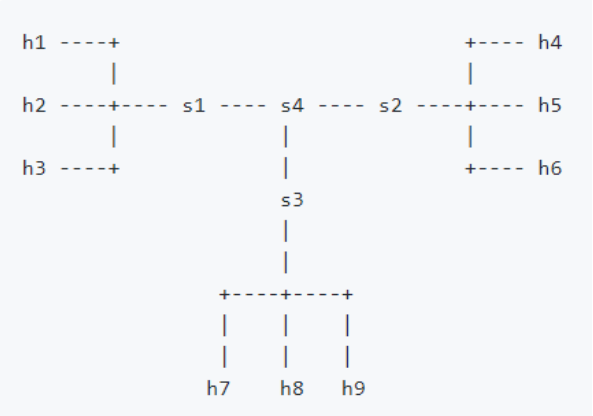
A partir da subseção 4.2 é minha contribuição para o projeto do protocolo LTP, aonda novos limites foram conhecidos, e com isso, novos estudos em cima podem ser gerados, com o objetivo de implementar o protocolo LTP em redes comunitárias, e com isso diminuir o problema da exclusão digital.

4.1 Cenários Iniciais

Tirados do repositório com o código-fonte do artigo "*Programmable Low-End Networks: Powering Internet Connectivity for the Other Three Billion*", apresentado ao Simpósio Internacional IFIP/IEEE (IM 2021), esses cenários possuem como objetivo testar a aplicabilidade do protocolo LTP, especialmente desenvolvidos para comparar sua perfomance com uma rede sem esse protocolo chamada rede standard (STD).

A figura 4.1 é a topologia adotada como parâmetro padrão para maioria dos cenários, tanto antigos, como novos. São os cenários 1, 2, 3, 4, 5 e 9. Nela, o switch s4 atua como hub, ou seja, tudo que recebe numa porta, replica em outra. Exemplo, tudo que switch s1 transmitir, precisa ser recebido em todos os outros switches. q está saindo de S1, todos têm que escutar.

Figura 4.1: topologia considerada para cenários 1 a 5, sendo: hosts os h1-h10, switches os s1, s2 e s3, e s4 atua como hub.

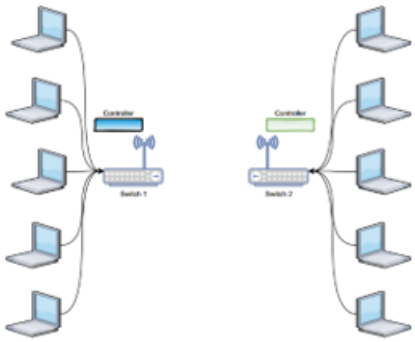


Fonte: extraído de <https://github.com/andrescheibe/ProgrammableLowEndNetworks/>

- Cenário 1 - Compara a estrutura utilizando o protocolo LTP e a estrutura STD com a variação dos enlaces em 64 kbps, 128 kbps, 256 kbps e 512 kbps e variando o protocolo entre UDP e TCP, considerando uma carga útil de 128 bytes.
- Cenário 2 - Comparar estrutura utilizando o protocolo LTP e estrutura STD com a variação dos enlaces em 64 kbps, 128 kbps, 256 kbps e 512 kbps e protocolo variando entre UDP e TCP, considerando uma carga útil de 512 bytes.
- Cenário 3 - Comparar estrutura usando o protocolo LTP e estrutura STD com a variação dos enlaces em 64 kbps, 128 kbps, 256 kbps e 512 kbps e protocolo variando entre UDP e TCP, considerando uma carga útil de 1024 bytes.
- Cenário 4 - Comparar estrutura utilizando o protocolo LTP e estrutura STD com a variação dos enlaces em 64 kbps, 128 kbps, 256 kbps e 512 kbps e protocolo variando entre UDP e TCP, considerando uma carga útil limitada a MTU.
- Cenário 5 - Analisa a estrutura com o protocolo LTP usando links de 256 kbps, pacotes UDP entre 2 hosts de cada switch. h1-> h4, h7-> h2, h5-> h8
- Cenário 6 - Analisa a estrutura com o protocolo LTP utilizando links de 256 kbps, pacotes UDP em uma rede com 2 switches e 5 estações cada.

h1 ----+ +---- h6
h2 ----+ +---- h7
h3 ----+---- s1 ---- s4 ---- s2 ----+---- h8
h4 ----+ +---- h9
h5 ----+ +---- h10

Figura 4.2: Representação do cenário 6 com 2 dispositivos low-end programáveis e 5 hosts por switch.



Fonte: Scheibe, André, et al., 2021.

- Cenário 7 - Análise do download de um arquivo da internet por um host conectado a um switch (s2) que está conectado a um segundo switch (s1) com conexão à internet. A conexão entre os dois switches usa o protocolo LTP e um link de 256 kbps.

h1 ----+---- s1 ---- s2 ----+---- internet

Figura 4.3: Representação do cenário 7 com 2 dispositivos low-end programáveis, 1 host e acesso à internet.

Fonte: Scheibe, André, et al., 2021.

4.2 Hipóteses de cenários de teste

Os cenários seguintes foram criados com objetivo de delimitar o funcionamento do protocolo LTP em cenários adversos e comuns na vida real. Busca-se responder as seguintes questões: LTP é um procolo apto a lidar com cenários em que acontece perda de pacotes nos links de tranmissão? Até que percentual de perda ainda é viável? Como aumentar a escalabilidade dele? Isso é possível de que modo? Qual é o funcionamento do protcolo se

mais que 1 host realiza download simultaneamente? E qual a quantidade máxima desses hosts? E funcionaria com perda de pacotes?

Realizando alterações no tamanho de pacote, diferentes tamanhos de *TAG*, quantidade de bytes device id, topologia e combinações dentro de cada dispositivo, a análise sobre o impacto que cada mudança provoca levará a um aprofundamento sobre o protocolo LTP.

4.2.1 Mudança na topologia

Considerando a topologia a seguir, com maior quantidade de hosts para analisar implicações no desempenho:

```
h1 ----+
h2 ----+
h3 ----+
h4 ----+
h5 ----+----- s2 ---- s1 ----+---- internet
h6 ----+
h7 ----+
h8 ----+
```

- Cenário 8: Análise do download de um arquivo da internet por 8 hosts simultaneamente, conectados a um switch (s2) que está conectado a um segundo switch (s1) com conexão à internet. A conexão entre os dois switches usa o protocolo LTP e um link de 256 kbps.

4.2.2 Perda de pacotes

Setando no mininet as condições do link, é possível simular os cenários em situações de perda de pacotes, o que é comum na vida real. Fazendo a inclusão de um campo de percentual de perda no arquivo *json* permite traçar as potenciais consequências. Por exemplo, se houver 10% dos pacotes em um determinado link, protocolo LTP ainda realiza uma taxa boa de transmissão? Até que ponto de perda *TAG* ainda consegue gerar transmissão, para saber o limite de perda para manter comunicação? Essa configuração de cenário é realizada no arquivo json, onde lê os arquivos do protocolo e realiza as funções para parsear links, host e switches.

- Cenário 9: Compara a estrutura utilizando o protocolo LTP e a estrutura STD com um link de 256 kbps com taxa de perda de pacotes de 1% e 5%, variando o protocolo entre UDP e TCP, considerando uma carga útil de 128 bytes, 256 bytes, 512 bytes, 1024 bytes e carga útil limitada a MTU.

Considerando que, para simular um ambiente wireless, onde qualquer envio é propagado para todos os dispositivos que estiverem no alcance, utilizamos um switch em condição de HUB (switch 4, figura 4.1) que replica em todas as demais portas um pacote recebido. Deste modo, considerando um percentual de perda entre o switch de origem e o switch hub (s4) e entre este e o switch destino, podemos afirmar que o percentual de perda configurado será cumulativo.

4.2.3 Download com perda de pacotes

- Cenário 10: Análise do download de um arquivo da internet, com uma taxa de perda de pacotes de 1% e 5%, por um host conectado a um switch (s2) que está conectado a um segundo switch (s1) com conexão à internet. A conexão entre os dois switches usa o protocolo LTP e um link de 256 kbps. Busca-se identificar com até qual percentual de perda a estrutura faz download.

4.2.4 Maior escalabilidade

Aumentando a quantidade de bytes do cabeçalho para ver que impacto causa. Considerando que 256 bits para *TAG* e 256 bits para dispostivos é a limitação da proposta inicial por usar 8 bits (1 byte) de tamanho para estas variáveis, aumentar o tamanho do campo iria garantir maior escalabilidade.

O pior caso envolve aumentar pra 2 bytes no identificador de dispositvo Dev_Id e 2 bytes no identificador *TAG* TAG_id. E então refazer os testes com determinado link para comparar usando tamanho de *TAG* diferente. Com esse aumento no cabeçalho, a escalabilidade se torna maior, por existir mais combinações entre dispositivos. Propõe-se descobrir se escalabilidade e ganho estão relacionados. Exemplo: numa ponta existe 10 máquinas e cada uma acessa 10 sites diferentes, só aí tem 100 combinações de 256. E então responder a pergunta: é útil incrementar o device id, se é necessário ter uma rede com mais de 256 switches distribuídos? Pensando em combinações de link, pode ser interessante. Para isso, é necessário

realizar a mudança do cabeçalho dentro de cada controlador. E analisar os dados do tráfego realizados pelo iperf. Logo, temos:

- Cenário 11: Compara a estrutura utilizando o protocolo LTP modificado para 2 bytes no identificador de dispositvo Dev_Id e 2 bytes no identificador *TAG TAG*_id, e a estrutura STD, com um link de 256 kbps, variando o protocolo entre UDP e TCP, considerando uma carga útil de 128 bytes, 256 bytes, 512 bytes, 1024 bytes e carga útil limitada a MTU.

Entranto essa possibilidade de cenário foi descartada por mostrar-se inviável, pois o registrador de combinações estourou, travando a máquina virtual e impossibilitando qualquer troca de pacotes ser feita. Com 1 byte (8 bits, que possibilitam 256 valores) para identificador de dispositivo e 1 byte para para identificador TAG, o registrador possibilita 65.536 combinações; aumentando para 2 bytes cada um dos identificadores, esse registrador precisa ser capaz de apontar até 4.294.967.296 valores, o que é muito alto. Então foi testado também aumentar apenas o identificador TAG para 2 bytes, e ainda assim não foi possível concluir a simulação com êxito e chegou-se a conclusão que a distribuição do header atual do LTP é a mais otimizada para o que se propõe. Uma alternativa para aumentar a escalabilidade é utilizar outro tipo de controle diferente de um registrador para armazenar o status de cada conexão, visto que o crescimento se mostrou inviável.

5 RESULTADOS

Para testar a viabilidade do LTP, foram simulados todos cenários propostos. Os resultados são extraídos dos logs dos controladores, mas também são gerados arquivos que podem ser importados no wireshark.

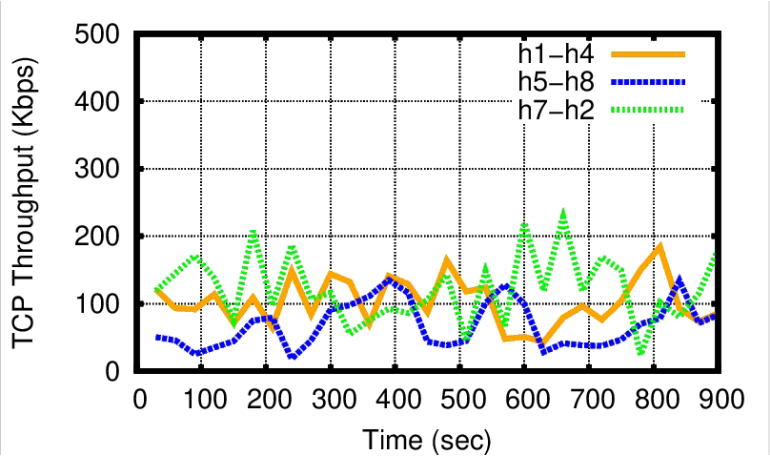
5.1 Resultados cenários iniciais

Com a mesma máquina e VM utilizados na construção do LTP, os cenários iniciais foram reproduzidos novamente e, a seguir, os resultados que foram alcançados.

Nos três gráficos seguintes, o primeiro com TCP e o segundo com UDP, é dado um compartilhamento justo do link de 256 kbps com o LTP e existem flutuações menores para o UDP, que busca se manter numa constante de transmissão. Existem 3 fluxos simultâneos acontecendo e rodando, mostrando a viabilidade do cenário.

Em TCP tem mecanismo que diminui as transmissões quando link satura para melhor se adaptar à rede, por isso no gráfico tem um throughput crescente do link que abaixa após alcançar o limite.

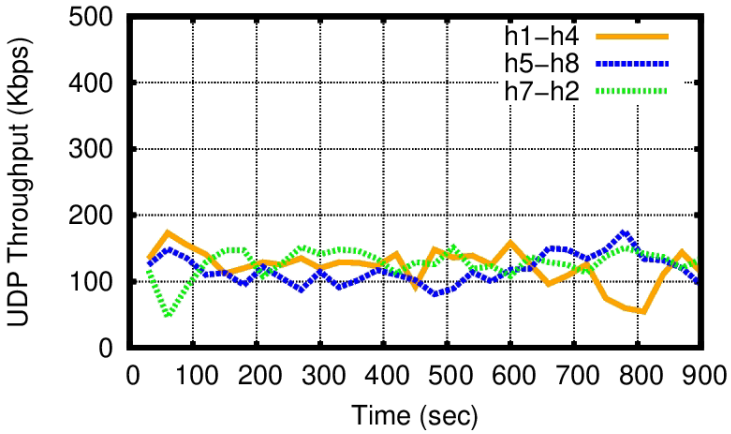
Figura 5.1: plot os três fluxos do cenário 5, protocolo TCP.
h1-> h4, h7-> h2, h5-> h8.



Fonte: da autora.

Figura 5.2: plot os três fluxos do cenário 5, protocolo UDP.

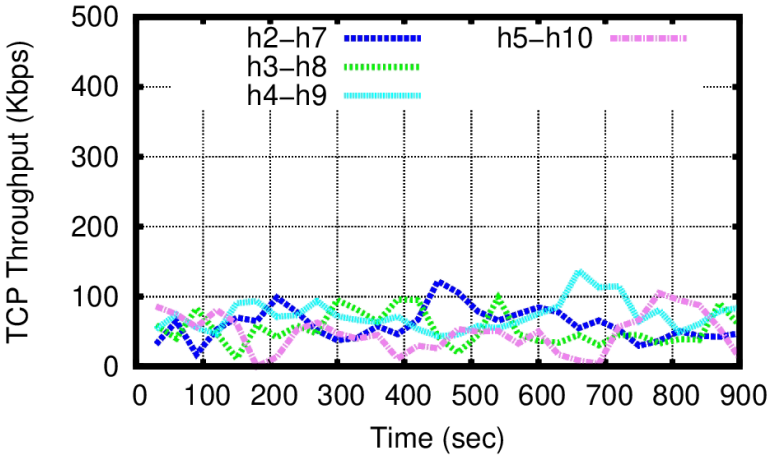
h1-> h4, h7-> h2, h5-> h8



Fonte: da autora.

Figura 5.3: plot dos cinco fluxos do cenário 6, protocolo TCP.

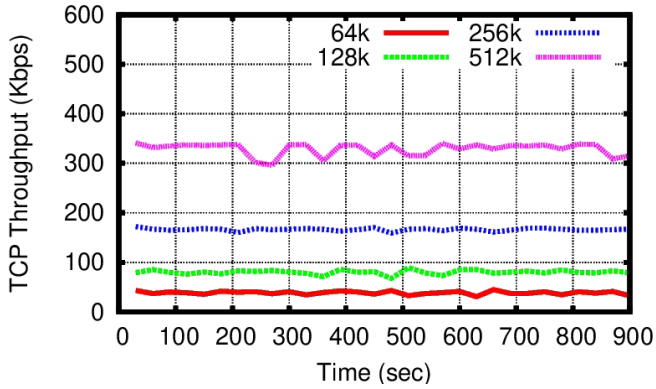
h1-> h6, h2-> h7, h3-> h8, h4-> h9, h5-> h10



Fonte: da autora.

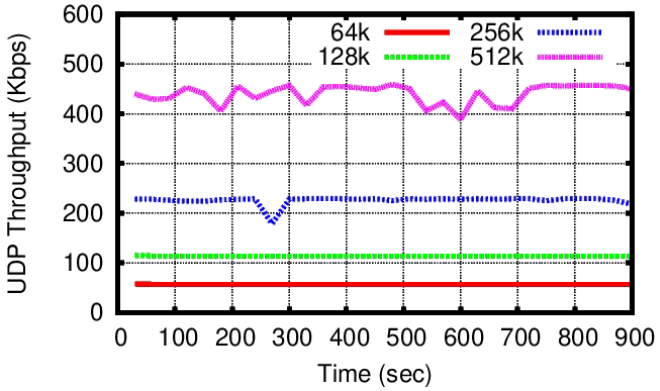
Nos 4 gráficos seguintes, alterando a velocidade do link de transmissão e utilizando MTU, é simulado a perfomance do LTP com variedade de restrições técnicas comuns aos dispositivos LPWAN, como links de banda estreita e pequenas cargas úteis, e nos resultados, cada fluxo obtém uma perfomance próximo à capacidade nominal do link, com um fluxo estável entre os pares de hosts:

Figura 5.4: plot dos quatro fluxos do cenário 1 (carga útil de 128 bytes), protocolo TCP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps)



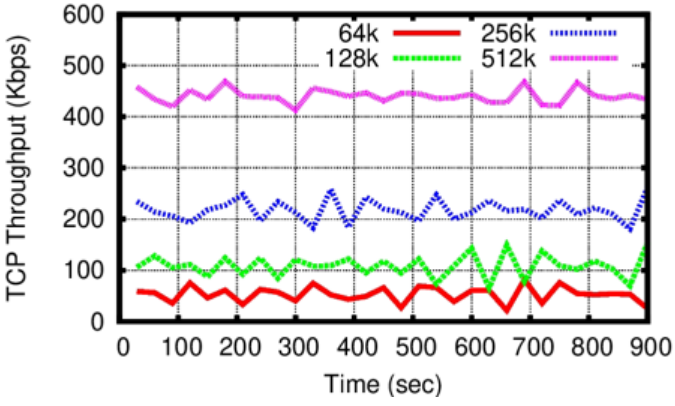
Fonte: da autora.

Figura 5.5: plot dos quatro fluxos do cenário 1 (carga útil de 128 bytes), protocolo UDP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps)



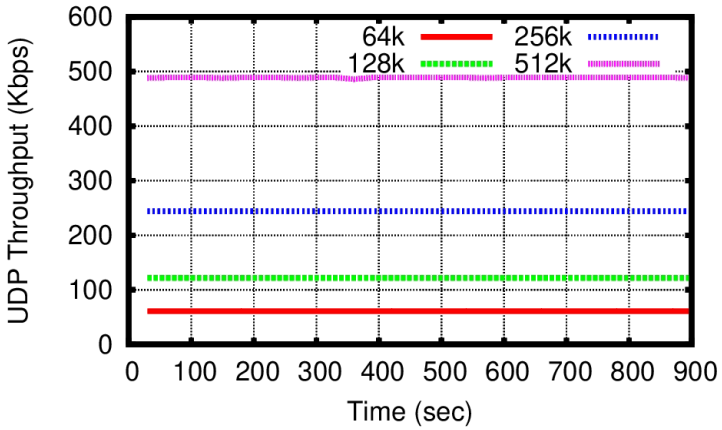
Fonte: da autora.

Figura 5.6: plot dos quatro fluxos do cenário 2 (carga útil de 512 bytes), protocolo TCP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps).



Fonte: da autora.

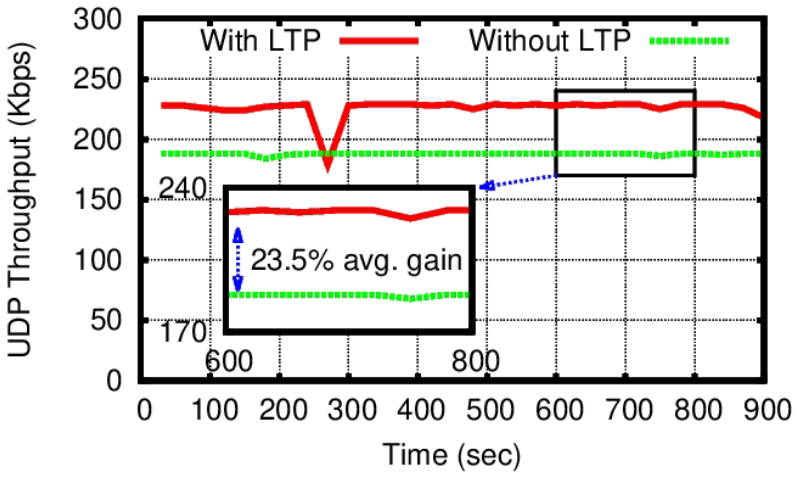
Figura 5.7: plot dos quatro fluxos do cenário 2 (carga útil de 512 bytes), protocolo UDP, alterando as quatro velocidades de link (64 kbps, 128 kbps, 256 kbps e 512 kbps).



Fonte: da autora.

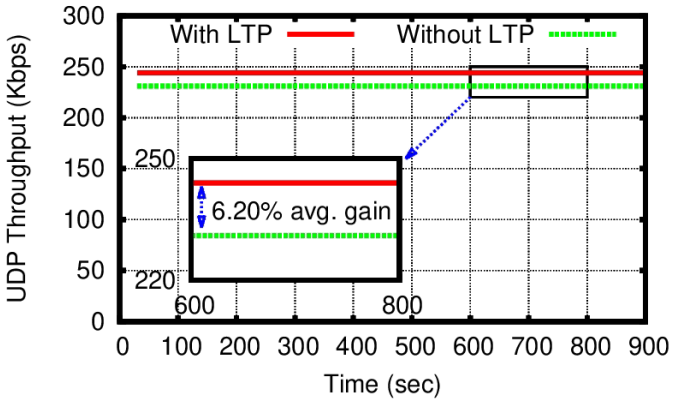
Nos 4 gráficos a seguir, compara-se a diferença de ganho onde tem a mesma velocidade de link e apenas um payload maior, existe ganho em qualquer um dos cenários, mas quanto maior é o payload, menor é o ganho por causa da proporção entre a redução e o tamanho do tamanho do pacote total. No gráfico 5.8, é comparado o desempenho com LTP e STD. Reduzindo 30 bytes num pacote de 128 bytes, tem um ganho de 23.5%. Em seguida, nos gráficos 5.9 e 5.10, a proporção continua reduzindo. Então, na figura 5.11, nota-se que reduzindo 30 bytes num pacote de 1500 (tamanho MTU), há uma redução de 2%, que é bem menor do que no caso da figura 5.8 com uma carga útil de apenas 128 bytes, principalmente.

Figura 5.8: cenário 1 (carga útil de 128 bytes), protocolo UDP, link de 256 kbps.



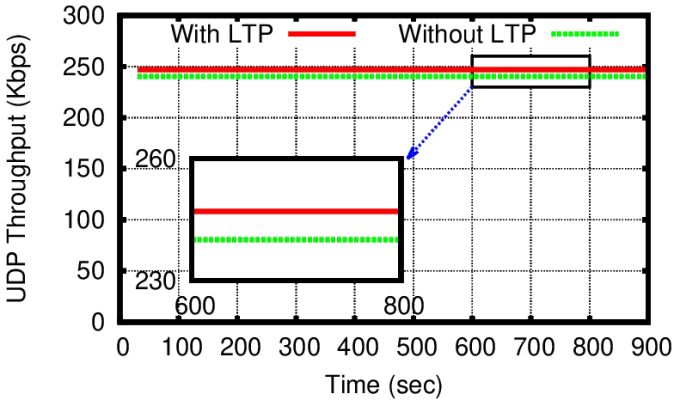
Fonte: da autora.

Figura 5.9: cenário 2 (carga útil de 512 bytes), protocolo UDP, link de 256 kbps comparando o desempenho com LTP e STD. A proporção está reduzindo, comparando com a figura anterior.



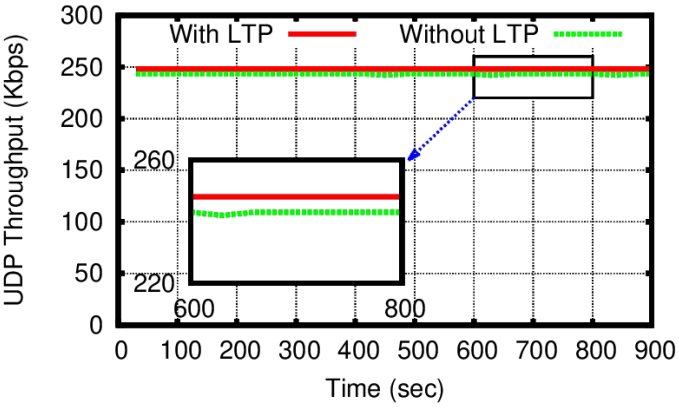
Fonte: da autora.

Figura 5.10: cenário 3 (carga útil de 1024 bytes), protocolo UDP, link de 256 kbps comparando o desempenho com LTP e STD.



Fonte: da autora.

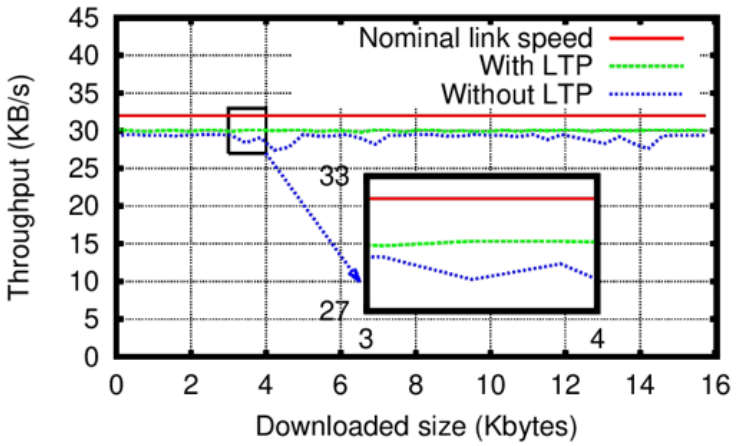
Figura 5.11: cenário 4 (carga útil limitada ao protocolo MTU), protocolo UDP, link de 256 kbps comparando o desempenho com LTP e STD.



Fonte: da autora.

O gráfico da figura 5.12 representa o cenário 7, onde ocorre download do arquivo da internet, com link de 256 kbps, usando a mesma topologia do cenário 6 porém um dos switches é um roteador que fornece acesso a internet e usa wget para recuperar um arquivo de 16 MB de um servidor HTTP na internet (DNS também passou pelo link sobre LTP), comparando o desempenho com LTP e STD. Percebe-se que LTP tem a transmissão mais alta do que sem LTP.

Figura 5.12: cenário 7



Fonte: da autora.

Figura 5.13: medindo as taxas de transferências em tempo real.

```
root@p4:~/ProgrammableLowEndNetworks/ltp-proto# rm results/scenario7/LTP-WGET-25
6k.txt
root@p4:~/ProgrammableLowEndNetworks/ltp-proto# tail -f results/scenario7/LTP-WG
ET-256k.txt
--2021-11-14 04:05:48--  http://sbrc2010.inf.ufrgs.br/anais/data/pdf/minicursos.
pdf
Resolving sbrc2010.inf.ufrgs.br (sbrc2010.inf.ufrgs.br)... 143.54.85.10
Connecting to sbrc2010.inf.ufrgs.br (sbrc2010.inf.ufrgs.br)|143.54.85.10|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 16444021 (16M) [application/pdf]
Saving to: ‘minicursos.pdf.2’

     0K .............. .............. ..............  2% 22.0K 11m53s
   384K .............. .............. ..............  4% 20.4K 12m3s
   768K .............. .............. ..............  7% 20.6K 11m50s
  1152K .............. .............. ..............  9% 21.3K 11m29s
  1536K .............. .............. .............. 11% 21.0K 11m12s
  1920K .............. .............. .............. 14% 21.2K 10m53s
  2304K .............. .............. .............. 16% 21.2K 10m34s
  2688K .............. .............. .............. 19% 20.7K 10m17s
  3072K .............. .............. .............. 21% 20.8K 10m0s
  3456K .............. .............. .............. 23% 21.3K 9m41s
  3840K .............. .............. .............. 26% 20.1K 9m25s
  4224K .............. .............. .............. 28% 21.1K 9m6s
  4608K .............. .............. .............. 31% 21.1K 8m47s
  4992K .............. .............. .............. 33% 20.6K 8m30s
```

Fonte: da autora.

Como pode-se notar pela tabela com valores médios retirados de medições dos cenários iniciais, disponível no artigo Managing Programmable Low-End Wireless Networks through Distributed SDN Controllers, LTP é melhor do que UDP/IP porque tem uma média Avg maior, considerando o intervalo de confiança CI de 99%, com 23,44% de ganho na comparação entre as duas médias, no caso #1, por exemplo. Conforme aumenta payload, o ganho reduz. Nas tabelas 1 e 2, constam dados dos resultados dos experimentos considerando várias velocidades de link, tamanhos de payload e nível de confiabilidade de 0.

#	Link speed (kpbs)	Payload size (bytes)	LTP					UDP/IP					Gain %
			Avg	SD	CI	Avg-CI	Avg+CI	Avg	SD	CI	Avg-CI	Avg+CI	
1	64	128	57.1	0.15	0.08	57.03	57.18	46.3	3.05	1.54	44.73	47.80	23.44
2	64	512	61.1	0.09	0.05	61.01	61.10	56.5	2.75	1.39	55.11	57.88	8.07
3	64	1024	61.7	0.39	0.20	61.52	61.91	59.0	2.69	1.36	57.66	60.37	4.57
4	64	1448	61.9	0.44	0.22	61.72	62.17	59.8	2.58	1.30	58.46	61.06	3.66
5	128	128	114.0	0.41	0.21	113.76	114.17	93.1	2.58	1.30	91.83	94.42	22.38
6	128	512	121.9	0.55	0.28	121.62	122.18	113.0	4.95	2.49	110.51	115.49	7.88
7	128	1024	123.0	0.79	0.40	122.60	123.40	118.0	4.04	2.03	116.00	120.06	4.21
8	128	1448	124.0	0.00	0.00	124.00	124.00	119.3	4.28	2.15	117.11	121.42	3.97
9	256	128	227.8	3.04	1.53	226.27	229.33	184.4	3.82	1.92	182.48	186.32	23.54
10	256	512	243.8	0.91	0.46	243.37	244.29	229.6	4.73	2.38	227.22	231.98	6.20
11	256	1024	246.1	2.32	1.17	244.90	247.23	237.6	5.70	2.87	234.73	240.47	3.56
12	256	1448	246.1	3.55	1.78	244.32	247.88	238.6	5.95	2.99	235.64	241.63	3.13
13	512	128	447.3	7.52	3.79	443.55	451.12	364.7	4.19	2.11	362.59	366.81	22.66
14	512	512	487.8	1.52	0.77	487.00	488.53	453.4	7.89	3.97	449.43	457.37	7.58
15	512	1024	493.6	2.01	1.01	492.62	494.64	472.9	7.78	3.91	468.99	476.81	4.38
16	512	1448	495.2	1.21	0.61	494.56	495.77	480.6	4.51	2.27	478.36	482.91	3.02

Tabela 1: Estatísticas para cada uma das variações com UDP

Fonte: Scheibe, André, et al., 2021.

No caso com TCP, o ganho e a média são maiores utilizando LTP, mas o desvio padrão é grande, impostos pelos controles do protocolo TCP/IP. Sendo o melhor caso de uso com LTP é para um link de transmissão de 64kbps, com payload de 128 bytes usando TCP. Com payload maior, maior a chance de saturar o link e os mecanismo de TCP agirem e se tornando mais difícil perceber a melhora com LTP, mesmo com nível de confiança de 99% não é possível afirmar que a solução com LTP possui os melhores resultados, mas para pacotes pequenos, sim.

#	Link speed (kpbs)	Payload size (bytes)	LTP					TCP/IP					Gain %
			Avg	SD	CI	Avg-CI	Avg+CI	Avg	SD	CI	Avg-CI	Avg+CI	
1	64	128	38.3	4.01	2.02	36.27	40.30	30.0	3.18	1.60	28.41	31.60	27.59
2	64	512	58.4	22.21	11.18	47.21	69.56	49.0	12.28	6.18	42.83	55.19	19.13
3	64	1024	57.2	16.05	8.08	49.14	65.30	55.0	15.55	7.83	47.12	62.78	4.13
4	64	1448	59.1	26.03	13.10	46.02	72.21	57.1	22.89	11.52	45.61	68.66	3.47
5	128	128	80.3	3.15	1.59	78.71	81.88	65.1	1.34	0.68	64.44	65.80	23.30
6	128	512	108.2	21.61	10.87	97.28	119.03	101.4	21.23	10.68	90.76	112.12	6.62
7	128	1024	115.7	28.01	14.09	101.62	129.81	111.0	20.90	10.52	100.50	121.54	4.23
8	128	1448	117.9	25.95	13.06	104.88	130.99	113.6	37.86	19.05	94.56	132.66	3.81
9	256	128	165.9	2.78	1.40	164.50	167.30	130.9	1.30	0.65	130.25	131.55	26.74
10	256	512	218.0	21.13	10.64	207.33	228.60	201.0	14.91	7.50	193.53	208.54	8.42
11	256	1024	231.8	19.09	9.61	222.16	241.37	221.7	19.84	9.99	211.68	231.65	4.56
12	256	1448	235.7	39.16	19.71	215.99	255.41	227.1	32.42	16.31	210.79	243.41	3.79
13	512	128	305.1	29.36	14.78	290.32	319.88	258.0	13.21	6.65	251.39	264.68	18.24
14	512	512	441.2	17.32	8.71	432.49	449.91	406.5	22.68	11.41	395.12	417.95	8.53
15	512	1024	466.9	17.44	8.78	458.16	475.71	446.6	11.76	5.92	440.68	452.52	4.55
16	512	1448	473.4	24.67	12.42	460.98	485.82	460.0	33.80	17.01	442.95	476.98	2.92

Tabela 2: Estatísticas para cada uma das variações com TCP.

Fonte: Scheibe, André, et al., 2021.

5.2 Resultados Cenários Novos

5.2.1 Resultado Mudança na topologia

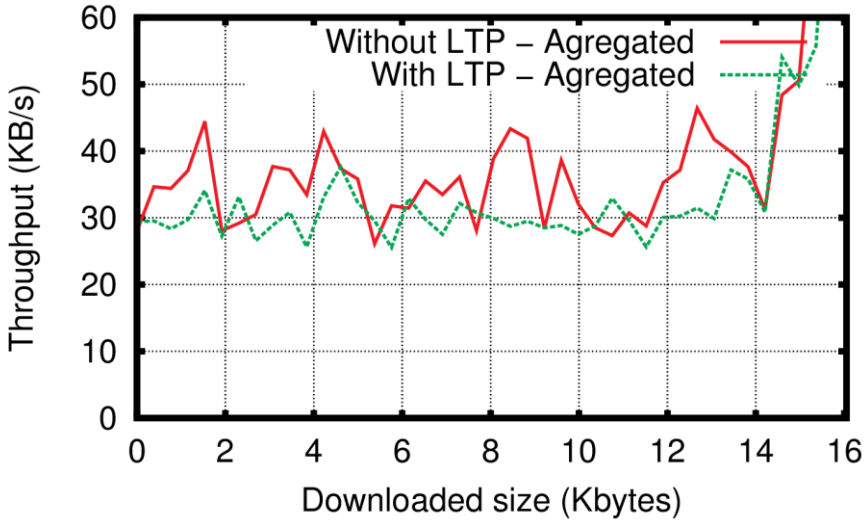
Considerando a topologia a seguir, com maior quantidade de hosts para analisar implicações no desempenho

h1 ----+
h2 ----+
h3 ----+
h4 ----+
h5 ----+----- s2 ---- s1 ----+---- internet
h6 ----+
h7 ----+
h8 ----+

- Cenário 8: Análise do download de um arquivo da internet por 8 hosts simultaneamente, conectados a um switch (s2) que está conectado a um segundo switch (s1) com conexão à internet. A conexão entre os dois switches usa o protocolo LTP e um link de 256 kbps.

Pelo resultado simulado na figura 5.14, não é possível afirmar que com oito estações simultâneas realizando download temos resultado melhor com LTP contra sem LTP, pois sem LTP houve um desempenho semelhante mas com uma vantagem pouco maior.

Figura 5.14: cenário 8, download simultâneo por 8 hosts por 256 kbps



Fonte: da autora.

Para realizar esse cenário, foi necessário alterar os controladores para acrescentar a verificação do mac de origem, ao invés de apenas comparar ip_src e ip_dst. A razão disso vem de, por ser DHCP, o primeiro pacote sempre será de 0.0.0.0 para 255.255.255.255 para encontrar o servidor DHCP, assim a única diferenciação entre um host buscando IP do outro host é o MAC da origem.

Anteriormente, apenas tinha sido realizado o teste com um cenário de download por apenas 1 host, no caso, a hipótese era testar com mais hosts fazendo download simultâneo. Foram testados com mais hosts e protocolo LTP não funciona com mais de 9 hosts, considerando as configurações atuais, para isso seria necessária mudanças complexas, envolvendo alterar o programa que faz o parser do arquivo, pois só lia uma posição dos parser dos link. Então cenário com 8 hosts foi criado para demonstrar sua viabilidade e limite atual de hosts fazendo downloads simultâneos.

5.2.2 Resultado Perda de pacotes

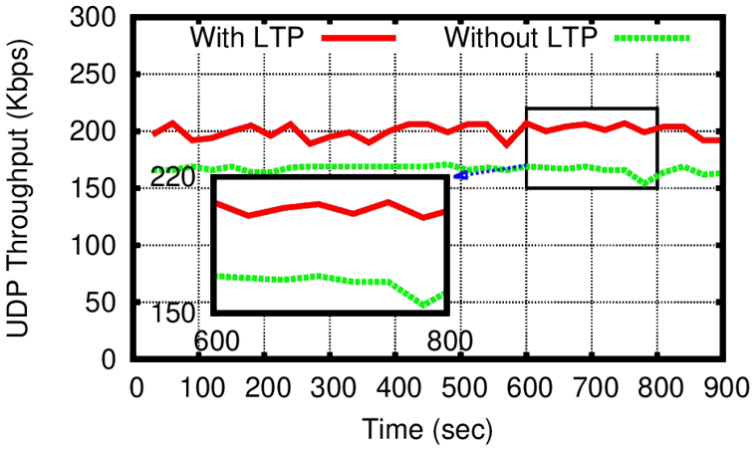
Neste cenário, busca-se descobrir sobre a aplicabilidade do protocolo LTP em uma situação envolvendo perda de pacotes, dado que ainda não foi testado. E, também, o limite do percentual de perda tolerado para se manter funcionando de forma adequada e com maiores vantagens de uso do que utilizando uma estrutura standart sem LTP.

- Cenário 9: Compara a estrutura utilizando o protocolo LTP e a estrutura STD com um link de 256 kbps com taxa de perda de pacotes variando para 10%, variando o protocolo entre UDP e TCP, considerando uma carga útil de 128 bytes, 512 bytes, 1024 bytes e carga útil limitada a MTU.

Observação: análise das simulações são inviáveis com perdas de 10% ou maiores com TCP, pois reduz a velocidade devido ao controle de congestionamento do protocolo que provoca a taxa de transferência a cair a zero muitas vezes. Além de existir uma perda acumulativa, por ocorrer entre diferentes switchs e somar no percentual total final.

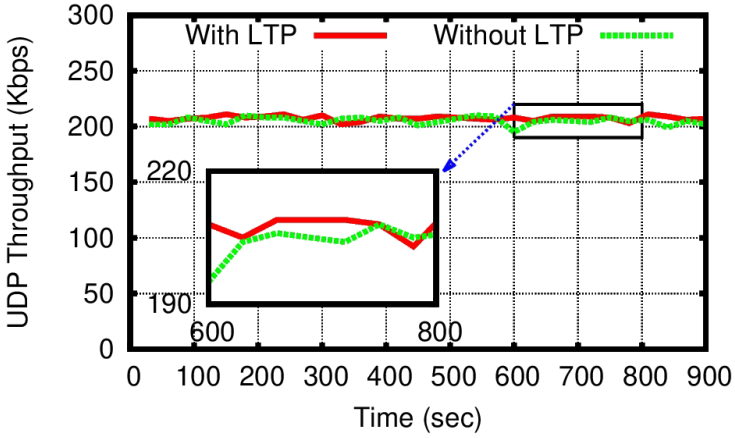
Nos quatro gráficos a seguir, é comparado o desempenho no cenário 9 entre LTP e STD no protocolo UDP com variações no payload, considerando perda de 10%.

Figura 5.15: cenário 9 com payload de 128 bytes.



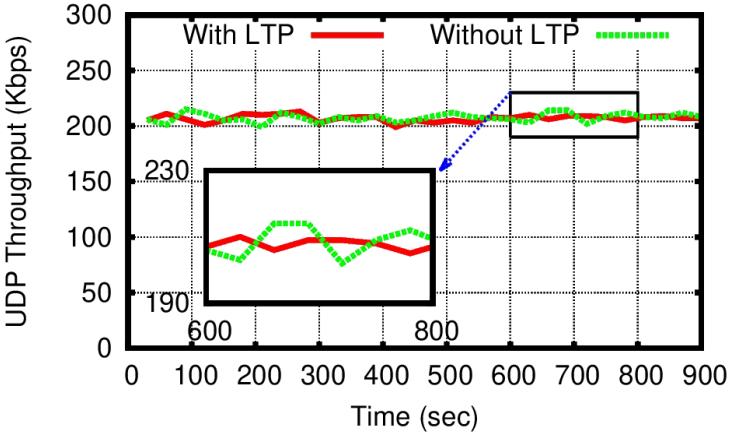
Fonte: da autora.

Figura 5.16: cenário 9 com payload de 512 bytes.



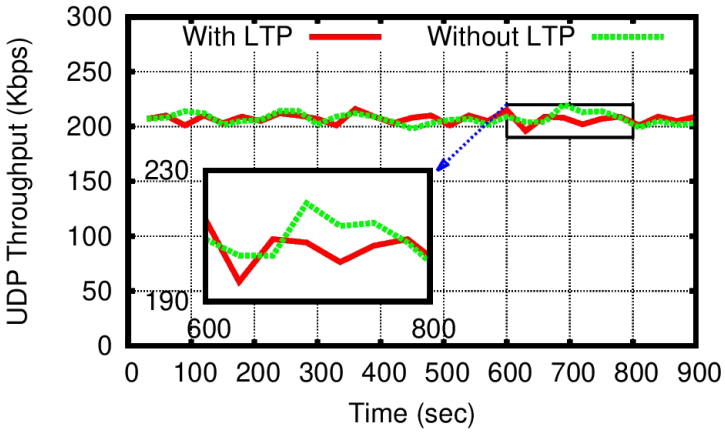
Fonte: da autora.

Figura 5.17: cenário 9 com payload de 1024 bytes.



Fonte: da autora.

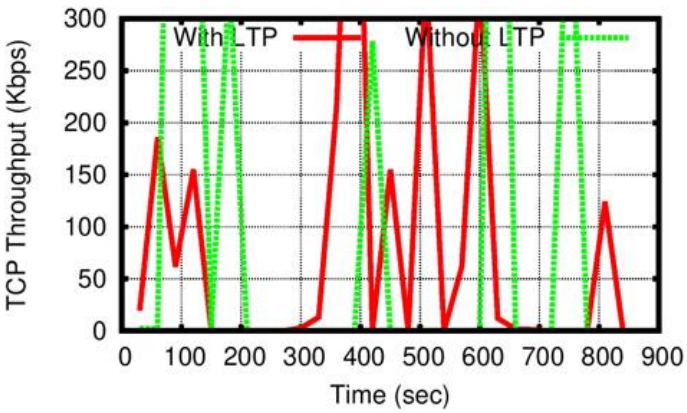
Figura 5.18: cenário 9 com payload de MTU bytes.



Fonte: da autora.

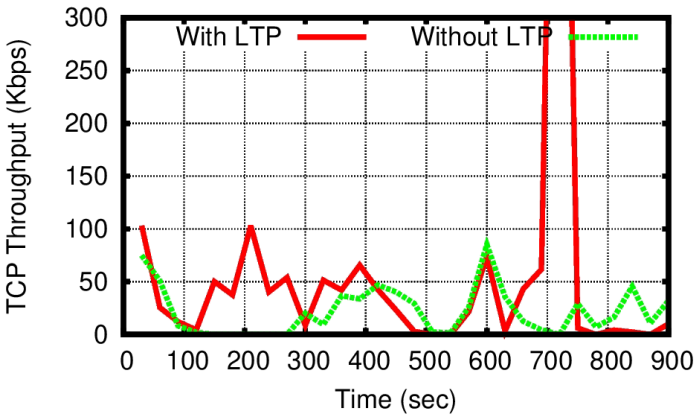
Nos quatro gráficos a seguir, é comparado o desempenho no cenário 9 entre LTP e STD no protocolo TCP com variações no payload, considerando perda de 10%.

Figura 5.19: cenário 9 com TCP e payload de 128 bytes.



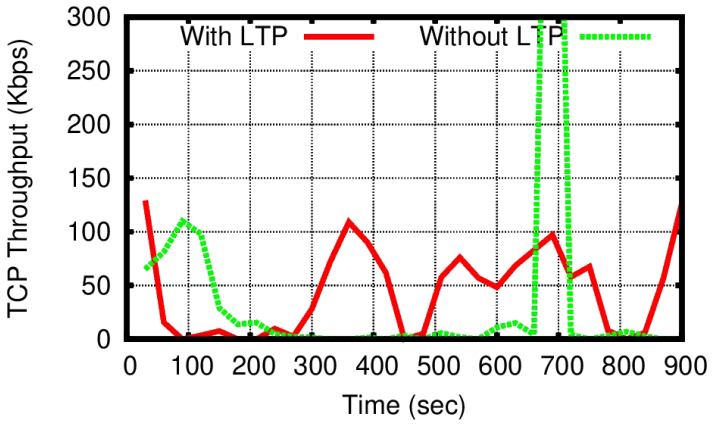
Fonte: da autora.

Figura 5.20: cenário 9 com TCP e payload de 512 bytes.



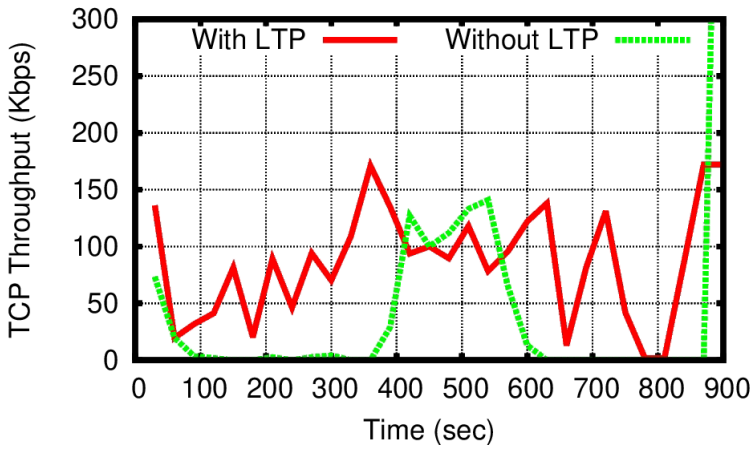
Fonte: da autora.

Figura 5.21: cenário 9 com TCP e payload de 1024 bytes.



Fonte: da autora.

Figura 5.22: cenário 9 com TCP e payload de MTU bytes.

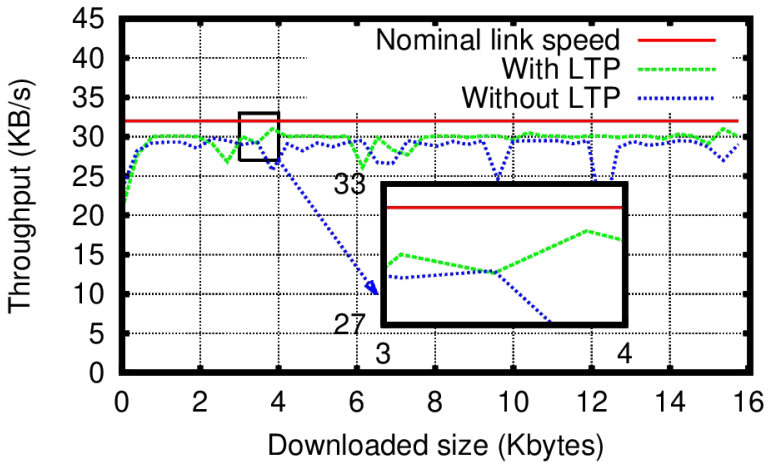


Fonte: da autora.

5.2.3 Resultado download com perda de pacotes

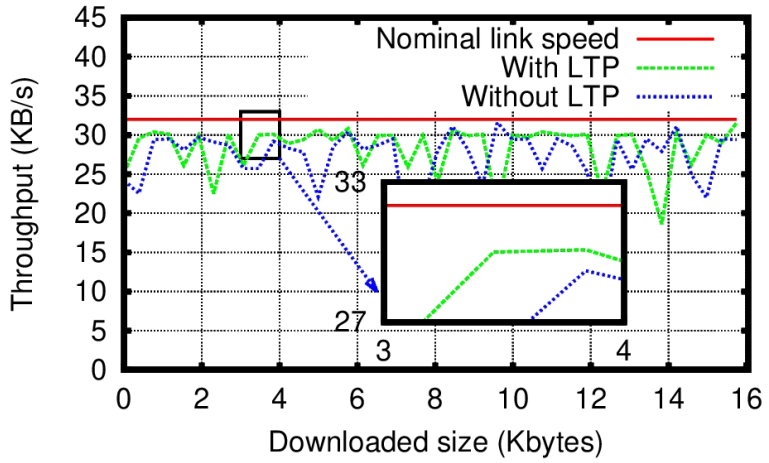
- Cenário 10: Análise do download de um arquivo da internet por um host conectado a um switch (s2) que está conectado a um segundo switch (s1) com conexão à internet. A conexão entre os dois switches usa o protocolo LTP e um link de 256 kbps. Comparando transferência de arquivo pela internet com perda de 10, 25, 35 e 50% de perda.

Figura 5.23: cenário 10 e compara download LTP e STD com 1% de perda de pacotes.



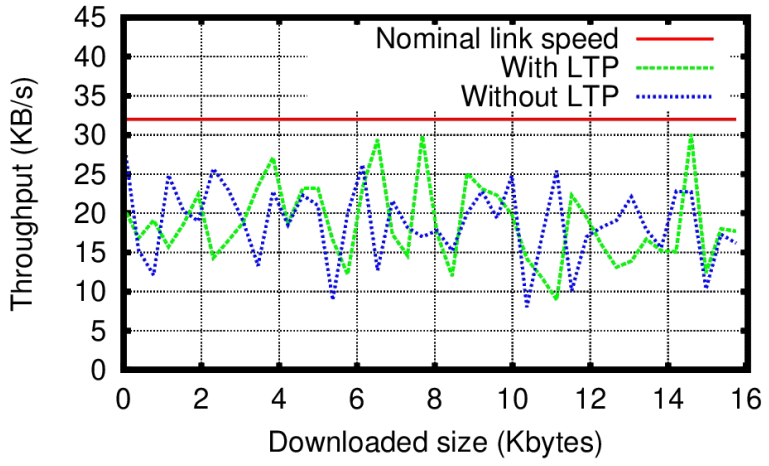
Fonte: da autora.

Figura 5.24: cenário 10 e compara download LTP e STD com 2% de perda de pacotes.



Fonte: da autora.

Figura 5.25: cenário 10 e compara download LTP e STD com 5% de perda de pacotes.



Fonte: da autora.

Os resultados demonstraram que protocolo LTP é apto a lidar com perda de pacotes e realizar com sucesso downloads considerando um percentual de perda de até 5%, e com um valor maior que esse, LTP já não é um protocolo ideal de uso por existir muita retransmissão, o que é um valor aceitável para uso, considerando seu meio wireless, conexão onde é natural ter perda de pacotes. Não existem regras definidas quanto a porcentagem de “perda aceitável”, por depender do serviço utilizado e em qual ambiente em que ocorre, porém uma boa prática é considerar uma perda de pacote a cada mil. Numa rede *fast Ethernet*, podem ocorrer perdas de até 10% sem impactar perceptivelmente na performance, enquanto que em *VOIP*, 2% já afeta (Wuemura, 2012).

6 CONSIDERAÇÕES FINAIS

Utilizando-se da linguagem de programação P4 para programabilidade do plano de dados, na primeira parte do escopo do trabalho de graduação, o foco foi se familiarizar melhor com a linguagem e rodar experimentos. Foram apresentados os conceitos básicos relacionados aos conteúdos abordados nessa monografia. Observações relacionadas ao plano de dados e ao plano de controle, assim como o funcionamento da arquitetura e ambiente de trabalho foram objetos desta seção. Então foram descritas as configurações do protocolo LTP, juntamente com as ferramentas e procedimentos metodológicos utilizados. O objetivo desta seção foi ilustrar o ambiente utilizado para aferir a proposta.

Então foram apresentados os cenários, tanto os já existentes quanto os novos, e os resultados encontrados, com o objetivo de avaliar o comportamento nos cenários propostos em relação aos pré-existentes. O principal objetivo desta seção foi apresentar os resultados da avaliação de encaminhamento dos pacotes, variando diversos fatores estruturais do protocolo, como aumentar escalabilidade e hosts, utilização com perda de pacotes e suas limitações.

Com base nos resultados é visível a aplicabilidade de redes programáveis de baixo custo com capacidade de transmissão de dados muito baixa para fazer transmissão de dados em redes comunitárias e em uso em ambientes hostis à transmissão de dados sem fio com protocolo LTP. Ao comprimir o cabeçalho do pacote de dados, aumenta o desempenho em um link de transmissão utilizando-se de uma arquitetura LPWAN de baixo custo.

Em dado compartilhamento do link de transmissão de 256 kbps entre vários hosts com o LTP, fluxos simultâneos rodam dividindo o link de forma justa, mostrando a viabilidade do protocolo, ocorrendo flutuações menores para o UDP, que busca se manter numa constante de transmissão, diferentemente de TCP onde ocorre controle de congestionamento para se adaptar à rede com throughput crescente do link que abaixa após alcançar o limite. Também, alterando uma variedade de restrições técnicas comuns aos dispositivos LPWAN, como links de banda estreita e pequenas cargas úteis, os resultados em cada fluxo obtiveram uma perfomance próximo à capacidade nominal do link, com um fluxo estável entre os pares de hosts. Além disso, notou-se que possui um ganho maior proporcionalmente a quão menor é o payload, pela proporção entre a redução e o tamanho do tamanho do pacote total. Em relação a download da internet, os resultados mostraram que LTP tem a transmissão mais alta do que sem LTP.

Os resultados dos cenários propostos mostraram que com vários hosts realizando downloads simultâneos, LTP não melhora no desempenho. Em relação à aumentar a escalabilidade, a proposta de aumentar a quantidade de bytes do cabeçalho não se mostrou viável por estourar o registrador de combinações de dispositivo e *TAG,* mas poderia em novo trabalho uma alternativa para aumentar a escalabilidade utilizando outro tipo de controle diferente de um registrador para armazenar o status de cada conexão. Também foi simulado a aplicabilidade do protocolo considerando perda de pacotes, e com 5% de perda, LTP já não é um protocolo ideal de uso por existir muita retransmissão, o que é um valor aceitável para uso, considerando seu meio wireless, conexão onde é natural ter perda de pacotes.

As análises dos cenários já existentes anteriormente, junto com os novos cenários propostos, contribuíram para um maior aprofundamento sobre o protocolo LTP, tendo sido exploradas as potencialidades e as limitações de diferentes conjunturas. Os objetivos do projeto estão alinhados ao desenvolvimento de um protótipo de baixo custo para conectar comunidades carentes em regiões remotas à internet, colaborando assim para resolver o problema de exclusão digital em regiões em desenvolvimento. Ou seja, os resultados apontam que é um protocolo viável para diversos cenários e apto a lidar com perda de pacotes.

REFERÊNCIAS

SCHEIBE, André et al. Managing Programmable Low-End Wireless Networks through Distributed SDN Controllers, 2021.

United Nations, “The promotion, protection and enjoyment of human rights on the internet,” United Nations Digital Library, vol. A/HRC/32/L.20, no. 32, pp. 1–4, June 2016. [Online]. Available: https://digitallibrary.un.org/record/845728

Internet.org, “Internet.org website,” 2020, available: <https://info.internet.org/en/>.

F4HDK. Build a Long-Distance Data Network Using HamRadio. 2019. <https://spectrum.ieee.org/geek-life/hands-on/build-a-longdistance-data-network-using-ham-radio>.

MAWINGU. Mawingu Networks. 2013. <https://www.wirelesswhitespace.org/wp-content/uploads/2017/06/TV-White-Space-for-Internet-Access-in-the-Developing-World_v5.pdf>.

Loon, “Project Loon website,” 2020, available: https://loon.co/.

A4AI, “A4AI Affordability Report 2018 [Online],” 2018, available: https://a4ai.org/affordability-report/report/2018/.

IETF GAIA Research Group, “IETF GAIA Research Group website,” 2020, available: https://datatracker.ietf.org/rg/gaia/about/.

4AFRICA**.** 4Africa Microsoft. 2013. <https://www.microsoft.com/africa/4afrika/>.

Crowcroft, Jon, Adam Wolisz, and Arjuna Sathiaseelan. "Towards an affordable internet access for everyone: The quest for enabling universal service commitment (dagstuhl seminar 14471)." Dagstuhl Reports. Vol. 4. No. 11. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

Braem, Bart, et al. "A case for research with and on community networks." (2013): 68-73.

Bosshart, Pat, et al. "P4: Programming protocol-independent packet processors." ACM SIGCOMM Computer Communication Review 44.3 (2014): 87-95.

Garcia, Luis Fernando Uria, et al. "Introdução à Linguagem P4-Teoria e Prática." Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos (2018).

DE GRADUAÇÃO, PROPOSTA DE TRABALHO. Uso de lofta para comunicacao em uma rede de sensores. Diss. UNIVERSIDADE FEDERAL DE PERNAMBUCO, 2019.

Wanzinack, Clóvis, Izabela Pichinin Bertola, and Marcos Claudio Signorelli. "INCLUSÃO DIGITAL DE IDOSOS NO LITORAL PARANAENSE: UMA PROPOSTA INTERDISCIPLINAR." Divers@! 6.1 (2013).

Lobato, Figueiredo, Alves, O que é inclusão digital e para que serve? - Todo Estudo 2006. Disponível em: https://www.gta.ufrj.br/grad/13_1/sdn/definicao.html

de Souza, Bruno Carlos, George Anthony Necyk, and Fábio Frezatti. "Ciclo de vida das organizações e a contabilidade gerencial." Enfoque: Reflexão Contábil 27.1 (2008): 09-22.

Kreutz, Mansilha, Miers, Mini-curso Escola Regional de Rede de Computadores, 2019.

Mafioletti, Diego Rossi. Crops–Uma Proposta de Comutador Programável de Código Aberto para Prototipação de Redes. Diss. Master's thesis, 2016.

J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," IEEE personal communications, vol. 6, no. 4, pp. 13–18, 1999.

U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 855–873, 2017.

Birnfeld, Karine, et al. "P4 Switch Code Data Flow Analysis: Towards Stronger Verification of Forwarding Plane Software." NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2020.

Sudonull Company, https://sudonull.com/post/25284-P4-programming-language-Factor-company-blog

Ritik Patel, https://ritikpatel17.medium.com/summer-internship-dynamic-load-balancing-using-sdn-42d95d6709ba, 2020.

Blog hardMob, usuário Wuamura, link: http://labcisco.blogspot.com/2013/05/interpretacao-dos-resultados-do-ping.html

Belli, Luca, et al. "Community networks: the Internet by the people, for the people." (2017).

Heimerl, Kurtis, et al. "Expanding rural cellular networks with virtual coverage." 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13). 2013.

L.-E. Jonsson, G. Pelletier, and K. Sandlund, “The robust header compression (rohc) framework,” Internet Requests for Comments, RFC Editor, RFC 4995, July 2007.

Prajapati A, Sakadasariya A, Patel J. Software defined network: Future of networking. In 2018 2nd International Conference on Inventive Systems and Control (ICISC). 2018;1351-1354.

Haji, S. H., Zeebaree, S. R., Saeed, R. H., Ameen, S. Y., Shukur, H. M., Omar, N., ... & Yasin, H. M (2021). Comparison of software defined networking with traditional networking. Asian Journal of Research in Computer Science, 1-18.