UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Rafael de Souza Vicentin

IMPLEMENTAÇÃO DO PROTOCOLO DE TEMPO PRECISO (PTP) EM UMA MÁQUINA LINUX

Florianópolis

2020

Rafael de Souza Vicentin

IMPLEMENTAÇÃO DO PROTOCOLO DE TEMPO PRECISO (PTP) EM UMA MÁQUINA LINUX

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Engenharia Elétrica para a obtenção do Grau de Bacharel em Engenharia Elétrica.
Orientador: Prof. Aguinaldo Silveira e Silva, Ph.D.

Florianópolis

2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Vicentin, Rafael de Souza
Implementação do Protocolo de Tempo Preciso (PTP) em uma máquina Linux / Rafael de Souza Vicentin ; orientador, Aguinaldo Silveira e Silva, 2020.
51 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Elétrica, Florianópolis, 2020.

Inclui referências.

1. Engenharia Elétrica. 2. PTP. 3. IEEE 1588. 4. Sincronização de Tempo. I. Silveira e Silva, Aguinaldo. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Elétrica. III. Título.

Rafael de Souza Vicentin

IMPLEMENTAÇÃO DO PROTOCOLO DE TEMPO PRECISO (PTP) EM UMA MÁQUINA LINUX

Este Trabalho foi julgado adequado como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica e aprovado, em sua forma final, pela Banca Examinadora

Florianópolis, 27 de fevereiro de 2020.

Prof. Renato Lucas Pacheco, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica, em exercício

Banca Examinadora:

Prof. Aguinaldo Silveira e Silva, Ph.D.
Orientador
Universidade Federal de Santa Catarina

Prof. Ildemar Cassana Decker, D.Sc.
Universidade Federal de Santa Catarina

Eng. Marcelo Neujahr Agostini, D.Eng.
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Agradeço primeiramente à minha noiva Carolina, sem a ajuda dela eu não teria encontrado forças para concluir este trabalho, e ao meu pai e à minha mãe, que me deram apoio emocional e financeiro essenciais para a realização dessa graduação.

Agradeço também aos meus amigos que conheci neste curso, que me proporcionaram momentos inesquecíveis durante a graduação, aos funcionários do departamento de Engenharia Elétrica, sempre atentos e dispostos à auxiliar os alunos, e aos meus colegas de trabalho da REIVAX, em especial ao Ismael, que me ajudaram com o desenvolvimento deste trabalho.

Por fim, mas não menos importante, agradeço aos professores do curso, por todos os seus ensinamentos que levarei para toda a vida, e em especial ao professor Aguinaldo, por sua paciência com minhas dificuldades neste trabalho.

RESUMO

Este trabalho se trata de uma implementação do Protocolo de Tempo Preciso (PTP) em uma máquina Linux usando um controlador de interface de rede I210. Uma explicação detalhada do funcionamento do PTP, além de uma breve explicação de redes de computador e de funcionamento de sistemas operacionais é apresentada. Alguns dos testes feitos para comprovar a eficácia da implementação do sistema de sincronização de tempo estão expostos e descritos ao fim deste trabalho.
Palavras-chave: PTP. IEEE 1588. Sincronização de Tempo.

ABSTRACT

This work is about an implementation of Precision Time Protocol (PTP) in a Linux machine using a I210 network interface controller. A detailed explanation of the operation of PTP, besides a brief explanation of computer networks and the operation of operating systems is presented. Some of the tests done to evaluate the efficiency of the implementation of the time synchronization system are described at the end of this work.

Keywords: PTP. IEEE 1588. Time Synchronization.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

1 INTRODUÇÃO

Neste trabalho foi feito uma apresentação do *Precision Time Protocol*, bem como uma introdução aos conceitos de sincronia de tempo. Espera-se que o leitor já tenha um conhecimento básico em programação e informática ao ler este trabalho, pois os conceitos serão utilizados, mas não serão explicados. Ao fim, foi apresentado uma solução do problema da implementação do PTP, além de testes para comprovar a eficácia da implementação.

1.1 MOTIVAÇÃO

O conceito de sequência de eventos (ou SoE, do inglês *Sequence of Events*) define que todos os eventos de um certo sistema acontecem seguindo uma determinada sequência. Para analisar a falha de um sistema complexo de computadores (como um sistema de controle de geradores elétricos), este conceito é relevante para descobrir qual equipamento iniciou o comando que causou a falha, economizando esforço e horas de trabalho para determinar a causa.

Para que seja possível determinar a exata ordem dos acontecimentos em um sistema, é necessário que todos os aparelhos estejam sincronizados temporalmente, o que é um desafio para os controladores mais atuais, com tempo de sincronização cada vez mais rápidos - podendo chegar na ordem de nanossegundos. Desta forma, este trabalho torna-se um estudo relevante, pois existe a necessidade de haver uma sincronização rápida entre os equipamentos na rede dos controladores.

A REIVAX Automação e Controle, empresa fundada em 1987 para suprir a necessidade de controladores modernos para máquinas síncronas, e que hoje é uma das líderes mundiais na área de tecnologia para geradores de energia, necessitava de uma sincronia de tempo melhor entre seus controladores, medidores e atuadores para melhorar a confiança da sequência de eventos dos seus produtos.

O Protocolo de Tempo Preciso (PTP, do inglês *Precision Time Protocol*) é uma tecnologia recente que consegue atingir uma sincronização da ordem de nanossegundos, funcionando com o mesmo tipo de rede usada pelos controladores da REIVAX; sendo assim, foi a tecnologia escolhida para solucionar o problema de determinar a sequência de eventos nos produtos da empresa.

1.2 OBJETIVOS

O objetivo deste trabalho é implementar o *Precision Time Protocol* em um computador instalado com placa do Controlador de Interface de Rede I210, com uma precisão mínima de 1 µs. O I210 será futuramente usado nos Controladores de Sistemas de Energia da REIVAX, e assim esta implementação será reutilizada nos futuros produtos da empresa.

1.3 *PRECISION TIME PROTOCOL*

O *Precision Time Protocol*, norma do *Institute of Electrical and Electronics Engineers* (IEEE) de número 1588, foi um esforço coletivo envolvendo várias entidades para obter-se uma sincronização com precisão superior à de microssegundos. Sua primeira versão, de 2002, descreve seu modelo padrão de funcionamento e unidades básicas; a segunda versão, de 2008, apresenta novas funcionalidades, alguns padrões exigidos pela norma (que foram revisados e aprimorados), mas teve como consequência a incompatibilidade entre as duas versões mencionadas. (IEEE, 2008)

Apesar de existirem diferentes soluções envolvendo alta precisão na sincronia de relógios, elas possuem suas problemáticas; o sistema de GPS, por exemplo, consegue um resultado muito bom, mas necessita de receptores em cada relógio, tornando-se caro e impraticável para aplicações dentro de áreas fechadas (sem vista direta para o céu).

1.3.1 Relógios

Antes de discutir acerca do funcionamento do protocolo, é importante entender o que é o relógio (seu elemento básico). Para o PTP, um relógio é qualquer dispositivo capaz de receber, enviar e processar mensagens PTP e que consiga medir a passagem do tempo desde a época definida para o sistema.

Para receber e transmitir mensagens, é necessário que exista ao menos uma interface de rede (porta), que pode estar em alguns estados, sendo os três principais: *Master* (Mestre), responsável por gerar os sinais de sincronização para os outros relógios de sua região; *Slave* (Escravo), que recebe os sinais de sincronização e faz requerimentos adicionais para atualizar seu relógio interno; e *Passive* (Passivo), ocor-

rendo quando a porta está "desativada", porém continua a receber mensagens administrativas (como as de anúncio) para verificar se é necessário reativá-la.

Os tipos de relógios previstos pela norma são, resumidamente, os seguintes: (ARNOLD, 2018c)

- *Boundary Clock* (Relógio de Fronteira): Relógio com múltiplas portas, com uma delas em modo escravo ou mestre, e as outras em modo mestre ou passivo. Ele trabalha como interface entre diferentes regiões da rede.

- *Grandmaster Clock* (Relógio Grão-Mestre): Relógio responsável por prover a referência de tempo para todos os outros relógios da rede. Deve estar conectado diretamente a uma fonte externa precisa de tempo, como um relógio atômico, um receptor de GPS ou GLONASS (*Globalnaya Navigatsionnaya Sputnikovaya Sistema*, sistema de satélites russo que equivale ao GPS).

- *Ordinary clock* (Relógio Comum): Relógio com apenas uma interface de rede, sendo normalmente um dos aparelhos no qual se deseja receber a referência precisa de tempo. Portanto, costuma estar com sua porta em modo escravo, podendo ser alterada para modo mestre caso não detecte nenhum mestre melhor na sua região.

- *Transparent Clock* (Relógio Transparente): Introduzido com a segunda versão da norma, sua função é corrigir as estampas de tempo nos pacotes que passam por ele para reduzir a não-simetria dos atrasos do canal. Este relógio não possui a necessidade de atualizar seu relógio interno. Normalmente se trata de um *switch* compatível com PTP.

1.3.2 Rede

Uma das maiores vantagens do PTP é a de trafegar sobre redes Ethernet, que é a mesma rede utilizada para trafegar outros pacotes de dados (como medições de sensores, por exemplo). Sendo assim, não há nenhuma necessidade de haver uma rede exclusiva para fazer a sincronização temporal entre seus dispositivos.

Os pacotes PTP usualmente trafegam sobre a segunda camada do modelo OSI (modelo que divide as redes de computadores em 7

camadas abstratas para facilitar a interoperabilidade de diferentes protocolos de comunicação), que é a camada de enlace de dados (*data link layer*), mas também é previsto na norma o uso da camada 3, a camada de rede (*network layer*).

O PTP normalmente trabalhe em *multicast*, onde o relógio mestre envia um pacote que é distribuído pela rede para todos os escravos, porém tabmém é previsto na normal o funcionamento no modo *unicast*, onde o relógio mestre envia um pacote em específico para cada relógio escravo.

1.3.3 Funcionamento

O princípio mais importante por trás do PTP é o de *timestamps*, ou estampas de tempo, que são como carimbos que marcam os pacotes sempre que eles chegam ou saem de uma porta PTP. Devido à esse mecanismo, o PTP consegue uma precisão tão alta, chegando a nanossegundos de diferença entre dois relógios.

Todas as mensagens deste protocolo contém um mesmo cabeçalho com informações relevantes, como a identidade de quem está enviando, o tipo de mensagem que está sendo enviada e a versão do PTP.

Quando qualquer dispositivo é iniciado, o algoritmo *Best Master Clock* (Melhor Relógio Mestre, algoritmo responsável por escolher qual relógio será a fonte de tempo para cada dispositivo) demanda que tal dispositivo envie uma mensagem de *Announce*, que é uma mensagem contendo dados do relógio como sua qualidade (que depende da precisão do seu oscilador e da sua fonte externa de tempo), sua prioridade, e a qual domínio ele pertence. De acordo com os valores das mensagens *Announce* recebidas por um relógio, as portas deste podem transicionar para o estado *Master* (quando não encontra nenhum relógio na região com uma referência de tempo melhor), *Slave* (quando há uma referência melhor de tempo e o relógio não está sendo sincronizado) ou *Passive* (caso não consiga adotar nenhum dos outros estados). As mensagens de *Announce* são periodicamente reenviadas, mesmo sem haver qualquer alteração física na rede, para garantir que todos os dispositivos estejam sempre conectados ao melhor mestre disponível. As mensagens de *Announce* também são responsáveis por mapear a rota que um pacote deve fazer para chegar no destino, resolvendo assim problemas de redundância ou alteração brusca na topologia da rede.

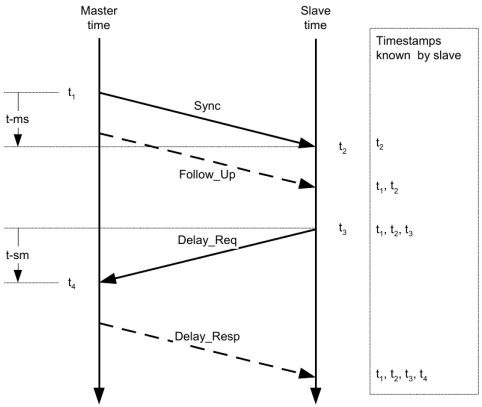

Figura 1 – Troca de mensagens de sincronização.

Todo o trabalho de sincronização do PTP é feito de acordo com a troca de mensagens mostrada na Figura 1. O *master clock* envia uma mensagem de *Sync*, que serve para sinalizar o início da troca de mensagens de sincronia. A mensagem de *Follow_ Up* é apenas presente em relógios de dois passos, e carrega com ela a estampa de tempo mais precisa referente ao momento em que o pacote *Sync* saiu do relógio mestre, t_1. Em relógios de um passo, a estampa t_1 está contida na própria mensagem de *Sync*. Assim que o relógio escravo recebe a mensagem de *Sync* ele grava uma estampa de tempo, t_2, e em posse dessa estampa e de t_1 ele envia uma mensagem *Delay_ Req* para o mestre; este tem a mesma composição da mensagem *Follow_ Up*, e grava a estampa de tempo da saída desta mensagem, t_3. Por fim, o mestre recebe a mensagem de *Delay_ Req* e grava a estampa de tempo de sua chegada, t_4, em seguida a enviando de volta para o escravo com a mensagem *Delay_ Resp*, que difere do *Delay_ Req* por haver um campo extra com a identidade do relógio que requisitou o atraso; isso se torna necessário pois o relógio mestre pode estar sincronizando vários relógios escravos ao mesmo tempo e cada um deles tem um trajeto de rede diferente, o que resulta em diferentes atrasos. Ao fim dessa troca, o escravo tem todas as estampas de tempo necessárias para calcular o atraso médio do caminho, e o faz utilizando da Equação 1.1. (ARNOLD, 2018d)

$$\frac{-t_1 + t_2 + t_3 - t_4}{2} \tag{1.1}$$

É importante notar que a Equação 1.1 não assume assimetria no caminho dos pacotes, nem *jitter* nos relógios transparentes; portanto, qualquer assimetria vai causar um erro na sincronização dos relógios. Assimetrias no caminho podem ser causadas por tempo de espera longo dentro de roteadores e *switches*, fazendo assim com que um pacote demore mais durante o caminho de ida do que o seu caminho de retorno (ou vice-versa), podendo ser eliminados ao se utilizar elementos compatíveis com PTP, que irão corrigir a assimetria adicionando os tempos de espera aos pacotes.

O sistema pode operar tanto com dois passos (*Two-Step*) ou um passo (*One-Step*). No caso de dois passos, o funcionamento é exatamente como o descrito acima. Para o funcionamento de um passo, a mensagem *Sync* deve conter uma estampa de tempo do horário atual do relógio e também deve receber uma estampa ao passar pela porta, anulando assim a necessidade da mensagem de *Follow_ Up*. Todos os relógios transparentes, relógios de fronteira e relógios grão-mestres devem permitir ser programados para funcionar tanto com um passo ou dois passos, enquanto os relógios escravos não necessitam dessa programação pois a mensagem de *Sync* virá marcada com qual modo está operando. (ARNOLD, 2018b)

Existe também o modo *Peer-to-Peer*, onde todos os relógios periodicamente calculam o atraso do caminho na rede entre ele e os outros relógios conectados a ele. Por conta disso, basta que o relógio grão-mestre envie as mensagens de *Sync* e *Follow_ Up*, e que os relógios adicionem esse atraso do caminho na rede antes de corrigir seu relógio e repassar a mensagem de *Sync* ou *Follow_ Up* para frente. (ARNOLD, 2018a)

O modo como o cálculo do atraso do caminho entre dois relógios é feito está demonstrado na Figura 2. É muito similar à sincronização do modo *End-to-End*, onde o relógio que necessita saber o atraso começa enviando uma mensagem de *Pdelay_ Req*, com TTL=1 (*Time to Live* mínimo, ou seja, o pacote vai chegar apenas nos equipamentos conectados diretamente ao relógio), contendo apenas uma estampa de tempo, e grava em seu registrador interno uma estampa do horário em que esse pacote saiu, t_1. Ao receber esta mensagem, o outro relógio grava uma estampa de tempo da chegada dela, t_2, e envia esta estampa na mensagem *Pdelay_ Resp*, também com TTL=1. O outro relógio também grava uma estampa de tempo quando a mensagem *Pdelay_ Resp* sai de sua porta, t_3, e a envia com a mensagem *Pdelay_ Resp_ Follow_ Up*. Ao receber a mensagem *Pdelay_ Resp* o relógio grava a estampa de tempo de sua chegada, t_4, e ao receber a mensagem *Pdelay_ Resp_ Follow_ Up*

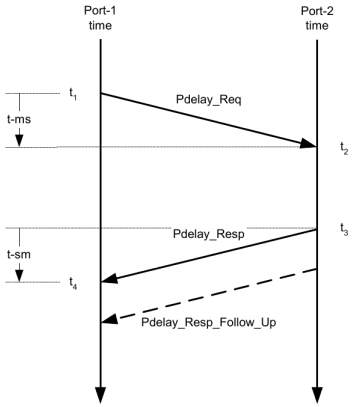

Figura 2 – Comunicação entre dois relógios adjacentes para calcular o atraso entre eles.

ele sabe todas as quatro estampas de tempo necessárias para calcular seu atraso em relação ao outro utilizando da Equação 1.1.

O modo *Peer-to-Peer* torna-se vantajoso porque se o relógio conhece seu atraso em relação aos relógios conectados à ele, e este atraso é recalculado frequentemente, isso faz com que o sistema venha a convergir rapidamente, e alterações de rotas (como por exemplo um *switch* sendo desconectado) não afetam significativamente a sincronização do sistema. Porém, para um bom funcionamento, o modo *Peer-to-Peer* necessita que todos os elementos da rede sejam capazes de fazer o cálculo de atraso, o que torna esta solução mais cara ou inviável.

1.3.4 Diferenças Entre PTP e Outras Tecnologias de Sincronização de Tempo

Um protocolo de sincronia muito difundido é o *Network Time Protocol* (NTP, "Protocolo de Tempo sobre Rede"), que atualmente já vem como padrão em todos os computadores. O NTP é um protocolo que distribui o tempo provido de diversas fontes confiáveis (relógios atômicos, por exemplo) para todos os dispositivos conectados à internet. Sua primeira versão oficial saiu em 1985, mas já estava em funcionamento desde antes desta data.

Seu funcionamento consiste em dividir os dispositivos em vários estratos, que são definidos de acordo com a distância (em nós) até uma dessas fontes registradas; assim, os relógios atômicos (que são usados como a base de tempo para o mundo todo) ficam no Estrato 0, os servidores diretamente conectados a esses relógios no Estrato 1, os computadores conectados a estes servidores no Estrato 2, e assim por diante. Cada dispositivo recebe sua referência de tempo do estrato localizado acima, e a passa para o estrato logo abaixo. Dispositivos de um estrato se conectam a múltiplos dispositivos do estrato acima para receber a referência de tempo, e do seu mesmo estrato para verificar a "saúde"dos dados recebidos. (MILLS, 2018)

Devido à sua estrutura feita com o propósito de manter uma sincronização razoável entre vários dispositivos conectados à internet (ou seja, distâncias muito longas), sua precisão não consegue ultrapassar a de 50 µs, sendo normalmente na ordem de 1 ms devido às latências da rede. O PTP, quando implementado puramente em software (ou seja, sem nenhum tipo de hardware que faça estampas de tempo) acaba sendo similar ao NTP.

Outra tecnologia de sincronização muito utilizada (principalmente em sistemas industriais) é o IRIG *timecode* (código de tempo IRIG). O IRIG *timecode* leva o nome do grupo responsável por sua criação, o *Inter Range Instrumentation Group* (pertencente ao exército dos Estados Unidos da América), que em 1960 teve seu padrão original aceito, e posteriormente revisado para se adequar às tecnologias novas.

O IRIG *timecode* é um conjunto de padrões de transmissão de informação de tempo, cada um diferenciado por uma letra e três números, sendo que a letra define a taxa de *bits* que a informação é enviada e a numeração define a forma do sinal. O mais comum é o IRIG-B (presente na maioria dos relógios GPS disponíveis no mercado), que tem uma taxa de 100 *bits* por segundo.

Portanto, realizando uma rápida comparação entre as tecnologias de sincronização, o NTP é a opção mais barata, porém com menor precisão; o IRIG-B possui preço e precisão razoáveis, porém necessitando de uma rede própria para distribuir seus sinais de sincronia, e o PTP possui maior precisão em troca do maior custo de implementação.

1.3.5 *PTP Profiles*

O PTP é um protocolo bastante flexível, com vários parâmetros para configuração viável e adequada, como o intervalo entre mensagens de anúncio, que é especificado como podendo ser qualquer potência de 2 dentro do intervalo 2^{-128} à 2^{127} (sendo este valor em segundos). Isso torna-se vantajoso pois permite bastante flexibilidade para várias implementações, porém pode ser prejudicial ao garantir que não haverá compatibilidade entre todos os equipamentos que implementam a norma, e por causa disso surgem os perfis (*profiles*) do PTP.

Os perfis servem para unificar equipamentos com um mesmo propósito, garantindo que todos operem dentro dos requisitos da aplicação. Como sincronia de tempo é indispensável para várias indústrias, surgiram diversos perfis para cada tipo de aplicação comum do protocolo. A norma em si já vem com o denominado "Perfil Padrão"(*Default Profile*), que não possui muito uso prático, porém é utilizado para testar equipamentos e servir de referência para outros perfis e aplicações.

Um perfil deve determinar as opções do algoritmo de escolha do melhor relógio mestre, as configurações de gestão, mecanismos de cálculo de atraso do caminho, os limites das configurações de comunicação (como intervalo entre mensagens de anúncio), além de determinar quais mecanismos de transporte, tipos de relógios, e opções são obrigatórios, permitidos ou proibidos. Além disso, é possível estender as opções utilizando do mecanismo TLV (*Type, Length, Value*, inglês para tipo, tamanho, valor, que é uma forma versátil de se adicionar dados customizados nas mensagens de um protocolo), ou especificando um novo algoritmo de escolha do melhor relógio mestre, mecanismo de gestão, ou mapeamento da camada de transporte.

Um exemplo destes perfis, relevante para o tema deste trabalho, é o *PTP Power Profile*, ou Perfil para Sistemas de Energia, que é utilizado em sistemas de geração e transmissão de energia elétrica. Sua necessidade vem do uso de sincrofasores, que são o resultado de medidas sincronizadas de medidores distribuídos espacialmente. O objetivo destes sincrofasores é obter o máximo de informações possíveis sobre o funcionamento e, principalmente, possíveis falhas e surtos dos sistemas de geração e transmissão de energia elétrica. Para ilustrar a necessidade do PTP neste tipo de medição, um erro de 0,01 radianos gera um Erro Total de Vetor (TVE, do inglês *Total Vector Error*) de 1% no sincrofasor, e para um sistema que opera à 50 Hz 0,01 radianos corresponde a aproximadamente 31 µs. Como é recomendável que a fonte de tempo para um medidor tenha uma precisão no mínimo 10

vezes menor que este valor, é necessário uma sincronia de 3 µs, que não é praticável com NTP. (IEEE, 2011)

2 IMPLEMENTAÇÃO

2.1 INTRODUÇÃO

Os detalhes técnicos da implementação, como as especificações dos componentes utilizados e a forma como eles foram montados para a realização deste trabalho, são descritos de forma que o projeto possa ser reproduzido.

O objetivo inicial deste trabalho era realizar uma implementação do PTP diretamente no controlador da REIVAX. Contudo, o sistema operacional em tempo real (RTOS, do inglês *Real Time Operating System*) do controlador não suporta relógios em hardware ou estampas para PTP, acarretando em um prazo de resolução do problema maior que o prazo limite do projeto. No entanto, todo o estudo sobre a norma, redes e *drivers* também é válido para a implementação no controlador, e também há a possibilidade de reutilizar o código deste projeto.

Considerando todos esses fatores, a implementação foi realizada em um sistema operacional Linux, que já possui APIs para utilizar relógios em hardware para PTP e para utilizar estampas de tempo, o que facilita e acelera a implementação. Foi utilizada a versão *lowlatency* do *kernel* do Linux, pois os tempos menores para processamento das tarefas do sistema auxiliam a obter resultados mais próximos de um RTOS.

Ainda para facilitar o trabalho e não ter de "reinventar a roda", foi utilizado um código pronto para a implementação do PTP como base, o PTPd. Ele foi escolhido por possuir todas as funções necessárias, utilizando da licença BSD, que é uma licença pouco restritiva e portanto facilita que aplicações comerciais usem códigos baseados nela.

2.2 IMPLEMENTAÇÃO DO PTP

Para realizar a implementação do PTP em um equipamento qualquer, é necessário:

1. Verificar o requisito de sincronia (1 µs, 100 ns...); se este for na ordem de até milissegundos, é possível atendê-lo com NTP por um custo muito mais acessível e confiabilidade semelhante. Quanto

menor for o requisito, maior será o gasto em equipamentos e hardware;

2. Montar o aparelho e testá-lo contra um relógio adequado para verificar a sua sincronia. Se o valor desta não estiver abaixo do requisito, o hardware ou o software deve ser melhorado.

3. Conectar o aparelho na rede em que ele deve operar e verificar a sua sincronia. Se o valor desta não estiver abaixo do requisito, a estrutura deve ser revista para mudar alguns dos equipamentos intermediários por equivalentes que sejam compatíveis com PTP, ou removê-los; adicionalmente, o hardware ou o software do aparelho deve ser melhorado.

2.3 SOBRE O I210

O componente mais importante deste trabalho é o controlador de interface de rede (NIC, do inglês *Network Interface Controller*). O NIC é a placa responsável por realizar a comunicação com outros equipamentos, via cabo RJ45 (cabo utilizado em redes *ethernet*), e portanto é a placa que irá processar as mensagens de PTP enviadas e recebidas pelo sistema.

O I210 foi escolhido por se tratar de um NIC que já possui registradores específicos para PTP e é suportado pela On-Time, empresa responsável pelo sistema operacional que roda nos controladores da REIVAX, facilitando o trabalho para implementar o PTP no controlador. Além disso, ele é barato e frequentemente utilizado, o que facilita a resolução de problemas e reduz o tempo utilizado no desenvolvimento. (INTEL CORPORATION, 2018)

2.4 SOBRE O PTPD

O PTPd é um *daemon* (um tipo de programa que roda em segundo plano no computador, realizando sua função sem ser notado pelo usuário) de código aberto (*open-source*) que implementa o PTP para relógios ordinários em sistemas baseados em Unix, podendo alcançar e manter uma precisão superior à microssegundos, até mesmo sem um hardware dedicado. Sua primeira versão implementava a primeira versão do PTP, IEEE 1588-2002, e sua segunda versão implementa a atual versão do PTP, IEEE 1588-2008.

Tabela 1 – Configurações PTP

Opção	Valor PTPd	Perfil Padrão PTP
Intervalo de Anúncio	2 s	1, 2, 4, 8 ou 16 s
Tempo para *Timeout*	6 anúncios	2 à 10 anúncios
Intervalo de Sincronização	1 s	0.5, 1 ou 2 s
Intervalo de Req. de Atraso	1 s	1, 2, 4, 8, 16 ou 32 s

Os valores padrão das configurações do PTPd estão demonstrados na Tabela 1, comparados aos valores permitidos pelo perfil padrão do PTP. O Intervalo de Anúncio é o tempo entre duas mensagens de anúncio. O Tempo para *Timeout* é o tempo sem receber mensagens de anúncio válidas, medido em número de mensagens de anúncio, para que um relógio mude seu estado para mestre. O Intervalo de Sincronização é o tempo entre duas mensagens de sincronização. Por fim, o Intervalo de Requisição de Atraso é o tempo entre duas mensagens de requisição de atraso (utilizadas no modo *peer-to-peer*). Todos os tempos entre mensagens são expressos (e configurados) em potências de 2 ($2^{-}1$ = 0.5 s ,2^0 = 1 s, 2^1 = 2 s, etc).

O programa PTPd foi escolhido por fazer uma implementação completa do protocolo, incluindo a máquina de estados, correção do horário interno do relógio e filtragem de *jitter*. Outro fator importante para a escolha é o uso da licença BSD, que permite implementação com fins comerciais e alteração sem complicações, ideal para a prototipagem de uma função em um produto novo.

2.5 SOBRE O RT430

O relógio mestre utilizado foi o RT430, de fabricação da Reason Tecnologia. O RT430 é um relógio referenciado aos satélites GPS e GLONASS capaz de ser a fonte de vários sinais de sincronização temporal em diversos formatos e protocolos, como IRIG-B e PTP. Possui uma precisão de tempo média de 50 nanossegundos para IRIG-B, e melhor que 100 nanossegundos para PTP, e suporta os perfis *Power Utility Automation* e *Power Profile* do PTP. (REASON TECNOLOGIA, 2018)

Por estes motivos, ele foi o relógio escolhido para verificar o funcionamento da implementação do PTP no I210.

2.6 MONTANDO O *SETUP*

Para realizar os testes, foi montado um computador com configuração mínima necessária: uma placa mãe, um pente de memória RAM, um processador e uma memória *flash* (os valores de memória e a velocidade do processador foram equivalentes aos do controlador da REIVAX que iria receber a funcionalidade de PTP). Então, neste computador foi adicionado o NIC I210, instalado o sistema operacional Xubuntu 16.04 (com a versão de latência baixa, para reduzir a degradação da sincronia devido à camada do sistema operacional) em sua memória e por fim instalado na máquina o PTPd. Este computador então foi conectado à um relógio mestre (primeiramente outro computador rodando o PTPd no modo *master* para testar se esta montagem funcionaria, e então o RT430 para os testes reais) e se executou o PTPd, salvando os valores de sincronia em um *log* para posterior verificação dos resultados. (JARC, 2018)

3 TESTES

3.1 INTRODUÇÃO

O objetivo dos testes realizados consiste em verificar a sincronia obtida para alguns casos que simulam o uso real da aplicação, utilizando uma disposição de componentes de forma a simular a realização física do projeto.

Os testes consistem em conectar o computador com a placa do I210 a um relógio mestre e depois iniciar o PTPd para sincronizar ambos. O modo *End-to-End* foi utilizado com dois passos e configurações padrão do PTPd (disponíveis na Tabela 1), pois estas representam de maneira adequada um cenário indesejado (pior caso), requerendo maior tráfego na rede e não tendo correção no atraso aleatório causado pelo *switch*.

Para analisar a precisão e estabilidade do I210 foi utilizado o próprio PTPd, que possui a opção de aumentar a verbosidade para mostrar com detalhe todas as variáveis internas do programa (como a diferença de tempo entre mestre e escravo e desvio padrão dela). A partir destes dados, gráficos da evolução do *offset* (desvio, neste caso de tempo) no tempo foram construídos utilizando a ferramenta gnuplot, que é um programa de linha de comando de código aberto com a funcionalidade de gerar gráficos a partir de arquivos de dados.

3.2 TESTE COM MÁQUINA VIRTUAL

No primeiro teste, foi utilizado outro computador rodando o PTPd para servir como relógio mestre, e devido à limitação deste computador (apenas com sistema operacional Windows), o PTPd precisou ser executado através de uma máquina virtual. As duas máquinas foram interconectadas usando um *switch* comum (sem suporte para PTP), tanto para se comunicarem quanto para atualizar o mestre via internet, como mostrado na Figura 3. Este primeiro teste foi feito para verificar o funcionamento do PTPd junto ao NIC I210.

O comando utilizado para o mestre foi:

sudo ./src/ptpd2 -C -V -M -i enp0s2

E, para o computador com o I210 (escravo):

sudo ./src/ptpd2 -C -V -s -i enp2s0

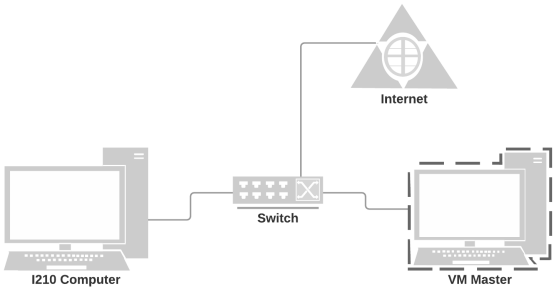

Figura 3 – Estrutura de teste utilizando como relógio-mestre uma máquina virtual

Onde **sudo ./src/ptpd2** é o comando para executar o programa (que está no caminho indicado) em modo administrador, **-C -V** são utilizados para que o programa funcione em primeiro plano com a mais alta verbosidade (ou seja, mostre todas as informações disponíveis sobre seu funcionamento), **-M** ou **-s** definem se o programa é executado em modo estritamente mestre ou escravo, e **-i** define qual interface de rede será utilizada (**enp0s2** para o mestre e **enp2s0** para o I210, sendo esta numeração definida pelo sistema operacional de acordo com a posição do NIC na placa mãe). Existem mais opções possíveis para o programa, mas as opções padrão servem para os testes pois estão de acordo com o perfil padrão do PTP.

O gráfico da Figura 4 corresponde ao *offset* da máquina contendo o I210 em relação ao mestre, rodando na máquina virtual do outro computador. A curva mais clara corresponde à média do último meio minuto; a mais escura, a média da última hora; e em preto, a média do teste. Neste gráfico e em todos os outros, os valores foram todos convertidos para serem apenas positivos e as escalas de tempo foram convertidas para logarítmicas, devido às discrepâncias nas magnitudes dos dados; desta forma, torna-se factível a comparação dos dados. Como era esperado, a sincronia se manteve em um valor relativamente alto (a média está em 298 µs) devido ao fato do mestre estar sendo executado a partir de uma máquina virtual, que está sujeita a todos os atrasos de entrega e recebimento de pacotes da hospedeira, e não haver nenhuma fonte de tempo precisa alimentando o mestre.

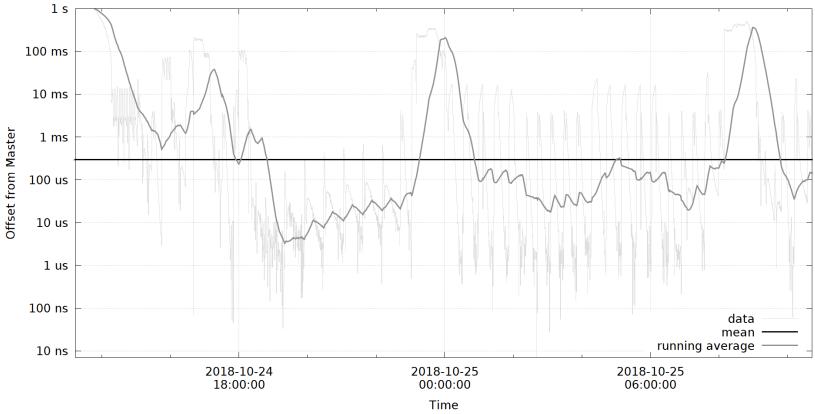

Figura 4 – *Offset* para mestre em máquina virtual

O gráfico da Figura 5 corresponde ao desvio padrão do *offset* da máquina contendo o I210 em relação ao mestre. A curva mais clara corresponde ao desvio padrão do último meio minuto; a mais escura, o desvio padrão médio da última hora; e em preto, a média do teste. Este gráfico complementa o gráfico da Figura 4, mostrando que nenhuma estabilidade foi alcançada.

O último gráfico deste teste, mostrado na Figura 6, mostra o histograma do desvio padrão da média do último meio minuto do *offset* da máquina contendo o I210 em relação ao mestre. Este gráfico ajuda a verificar a estabilidade geral do sistema durante o teste, mostrando por exemplo que mais de 20% do tempo o desvio padrão ficou dentro do intervalo de 1 ms à 10 ms.

Os resultados do teste ficaram fora dos requisitos da implementação, mas isto era esperado devido à má qualidade do relógio mestre, e portanto o teste foi um sucesso, pois verificou-se o funcionamento do PTPd na máquina com o I210. Com estes resultados foi possível prosseguir para os próximos testes utilizando de um relógio GPS como mestre.

3.3 TESTES COM O RT430 COM *SWITCH*

Nos testes que se seguiram, a configuração utilizada é exibida na Figura 7, com o RT430 recebendo sinal de GPS usando sua antena, e o

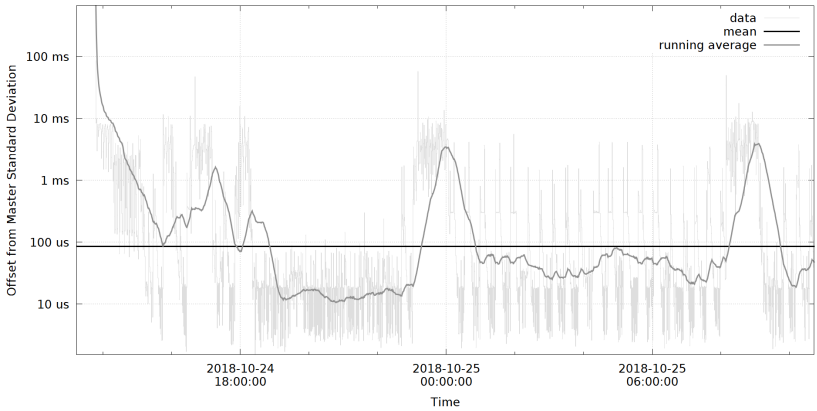

Figura 5 – Desvio padrão do *offset* para mestre em máquina virtual

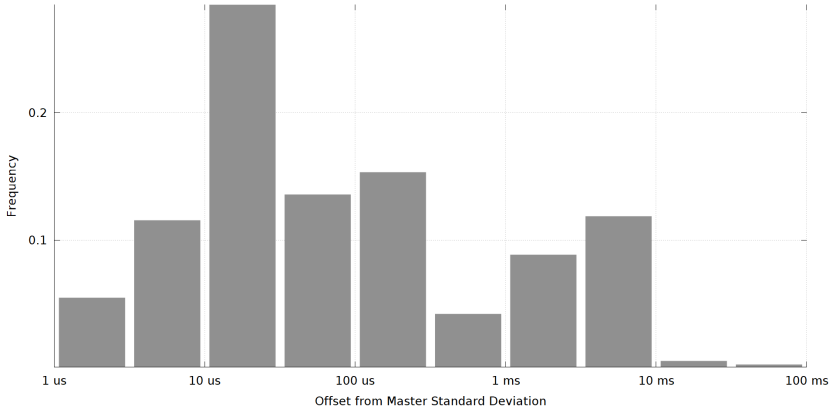

Figura 6 – Histograma do desvio padrão do *offset* para mestre em máquina virtual

computador do I210 conectado ao RT430 usando um switch gerenciável (porém sem suporte para PTP nem prioridade para pacotes PTP). O relógio RT430 foi configurado via Interface Web para ser compatível com as configurações padrão do PTPd, e a máquina contendo o I210 foi inicializada exatamente como no teste anterior.

Como o fator mais importante para a sincronia é o atraso nos pacotes, e o tráfego na rede é o principal causador de tais atrasos, foram realizados três testes em diferentes dias da semana para verificar o desempenho do sistema em condições adversas.

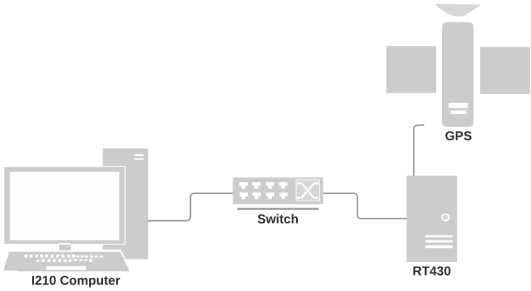

Figura 7 – Estrutura dos testes utilizando como relógio-mestre o RT430

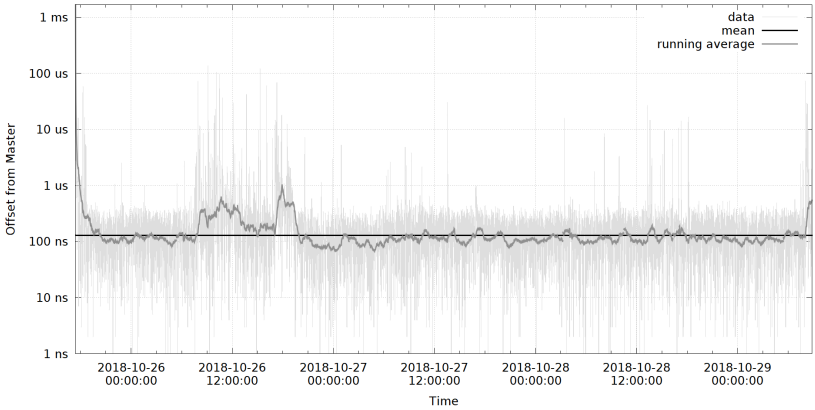

Figura 8 – *Offset* para mestre RT430, primeiro teste

Estes testes foram feitos sob a mesma estrutura, e por isso serão analisados e considerados como um todo. Todos os gráficos foram montados exatamente como os do primeiro teste.

Em qualquer um dos gráficos, é possível notar a influência de um relógio mestre adequado na sincronização entre os relógios, pois todas

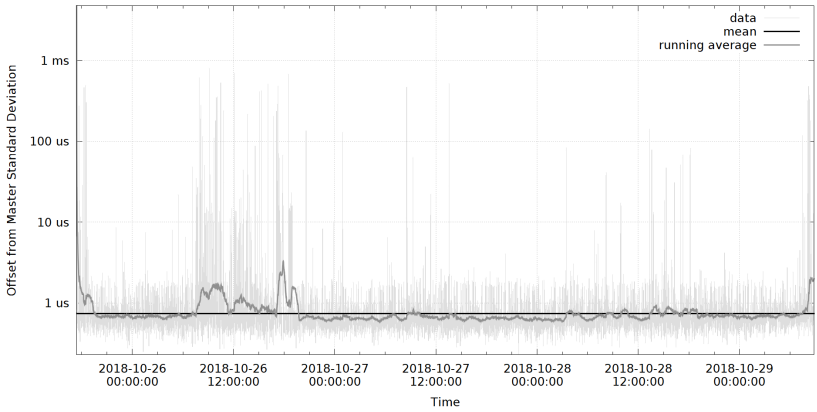

Figura 9 – Desvio padrão do *offset* para mestre RT430, primeiro teste

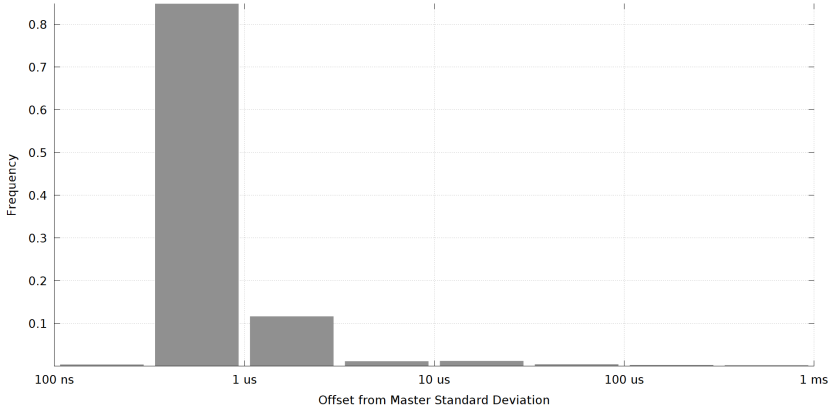

Figura 10 – Histograma do desvio padrão do *offset* para mestre RT430, primeiro teste

as médias apontam valores na ordem de 10^3 vezes abaixo das médias do primeiro teste, e as variações não apresentam a mesma característica oscilatória.

O primeiro teste apresentou ótimos resultados, com uma média de 129 ns de *offset* (Figura 8) e desvio padrão de 741 ns (Figura 9); além disso, o histograma mostra que durante mais de 80% do tempo,

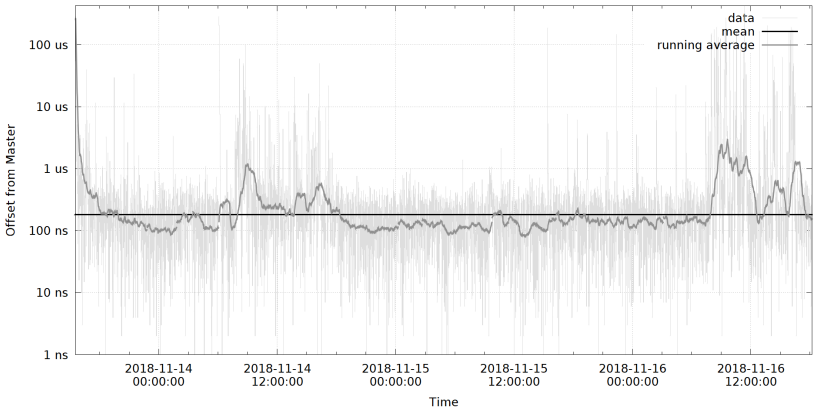

Figura 11 – *Offset* para mestre RT430, segundo teste

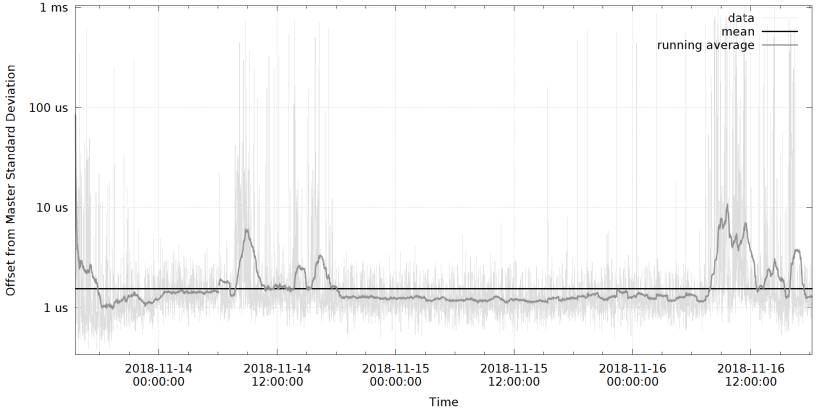

Figura 12 – Desvio padrão do *offset* para mestre RT430, segundo teste

o desvio padrão se manteve abaixo de 1 µs (Figura 10). Com este teste, já é possível reconhecer o padrão ocorrido nos testes com essa estrutura, onde durante os dias úteis nos horários em torno das 12h houveram picos de tráfego no *switch*, que comprometeram a qualidade da sincronização. Este mesmo *switch* sendo usado no teste supre toda a área de testes da empresa; portanto, durante os horários normais de

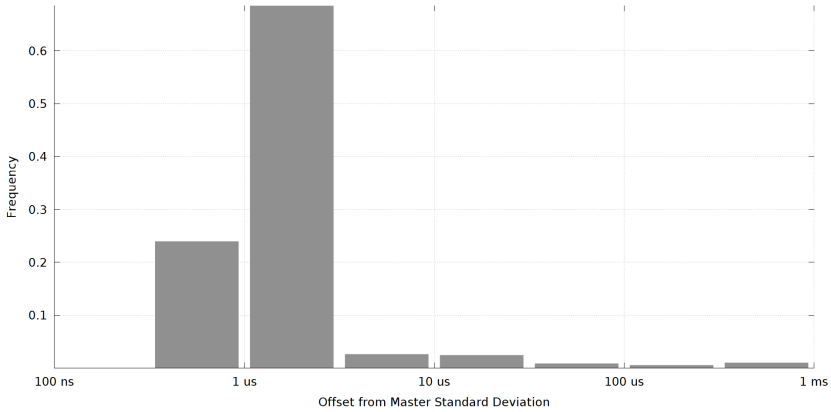

Figura 13 – Histograma do desvio padrão do *offset* para mestre RT430, segundo teste

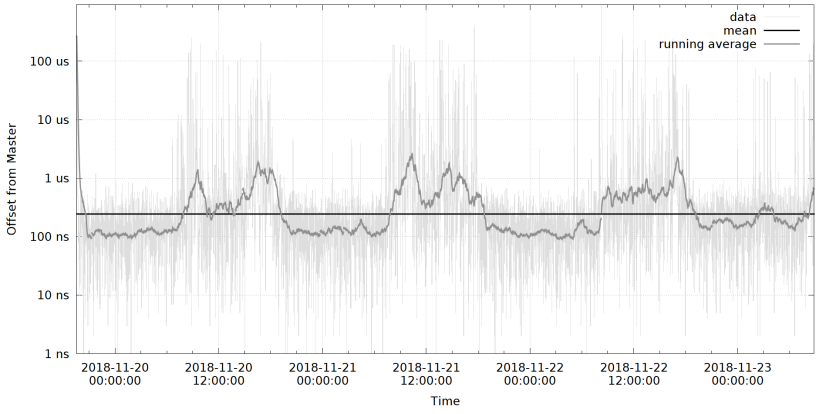

Figura 14 – *Offset* para mestre RT430, terceiro teste

trabalho ele fica congestionado, e isso acaba causando assimetria no trajeto dos pacotes PTP.

No segundo teste, o efeito do congestionamento da rede pode ser notado; suas médias estão acima das médias do primeiro teste (183 ns de *offset* segundo a Figura 11 e desvio padrão de 1.541 µs segundo a Figura 12). Em torno das 12h dos dias 14 e 16 é possível notar a influência

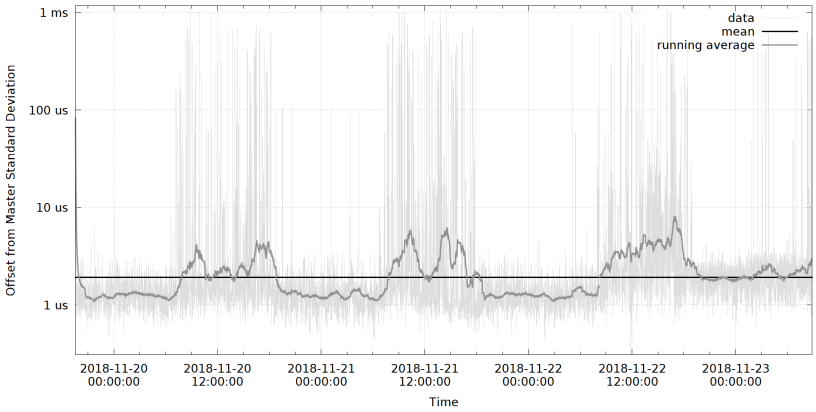

Figura 15 – Desvio padrão do *offset* para mestre RT430, terceiro teste

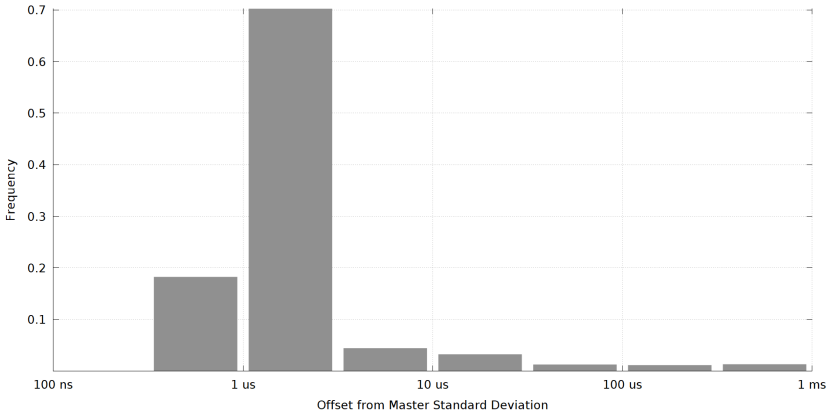

Figura 16 – Histograma do desvio padrão do *offset* para mestre RT430, terceiro teste

do congestionamento do *switch* na sincronização, e o histograma da Figura 13 demonstra também que o desempenho piorou devido à esta adversidade.

O terceiro teste confirmou a necessidade de transicionar para um *switch* compatível com PTP, pois o congestionamento do *switch* próximo das 12h dos dias úteis teve um forte impacto negativo neste

teste. As médias apresentam valores ainda mais distorcidos do que as médias do segundo teste, sendo 245 ns de *offset* segundo a Figura 14 e desvio padrão de 1.917 µs de acordo com a Figura 15. O histograma da Figura 16 demonstra uma característica prejudicial do congestionamento na rede: os valores de desvio padrão próximos e acima de 10 µs tornam-se significativos.

3.4 TESTE COM O RT430 SEM *SWITCH*

Para o último teste, o *switch* foi removido e a máquina contendo o I210 foi conectada diretamente ao relógio RT430, conforme mostrado na Figura 17. A motivação era isolar o teste do problema de congestionamento do *switch*, para então verificar a máxima capacidade do relógio GPS RT430 e do PTPd na máquina com o I210. Isso foi feito apenas por não se dispor de um *switch* PTP, pois este simularia mais fielmente um melhor caso para a sincronização, além de permitir testes com outros modos do PTP.

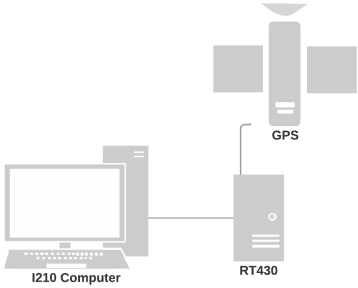

Figura 17 – Estrutura do teste utilizando como relógio-mestre o RT430, sem *switch*

Ao comparar o gráfico da Figura 18 com os outros gráficos dos testes com o RT430, fica evidente que o maior impedimento na sincronização era o congestionamento do tráfego de dados no *switch*. A média do *offset* neste teste ficou em 111 ns, muito próximos do limite de 100 ns de sincronização do RT430. Além disso, a variação durante todos os dias de testes foi quase nula, com alguns raros pontos ultrapassando 1 µs (com um pico sobressalente de mais de 10 ms, potencialmente devido a uma atualização do RT430).

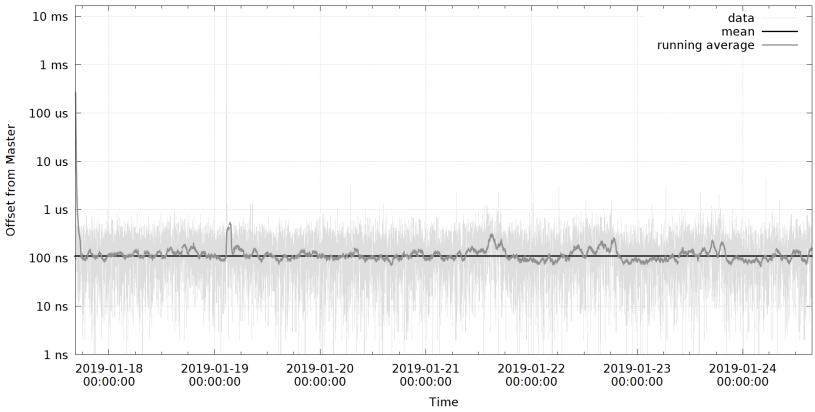

Figura 18 – *Offset* para mestre RT430, teste sem *switch*

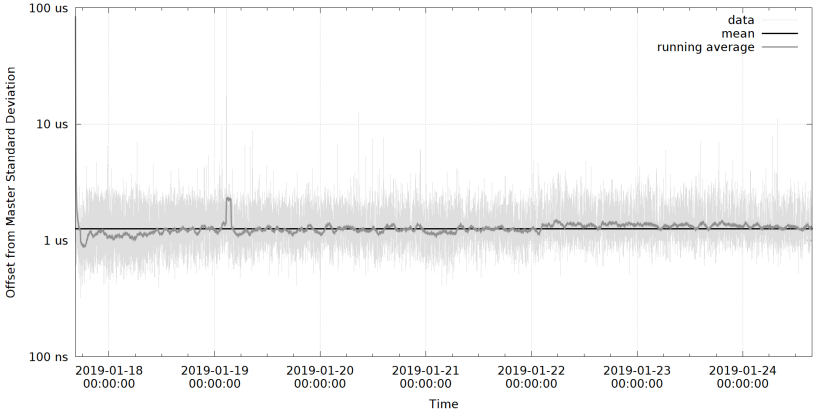

Figura 19 – Desvio padrão do *offset* para mestre RT430, teste sem *switch*

O desvio padrão mostrado no gráfico da Figura 19 apresenta resultado inferior ao resultado da Figura 9 (referente ao primeiro teste com o RT430 e o *switch*). Porém, a média deste teste de *switch* apresenta valores significativamente menores do que a média do outro, e considerando estes dois valores em conjunto pode-se notar que este teste teve uma melhor performance que os outros.

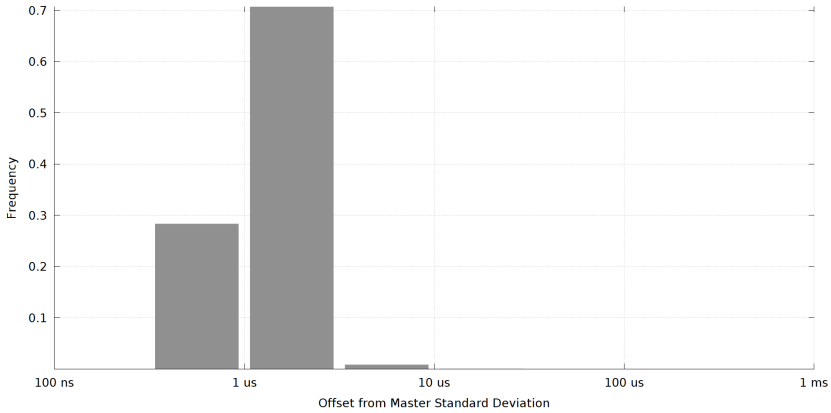

Figura 20 – Histograma do desvio padrão do *offset* para mestre RT430, teste sem *switch*

O histograma da Figura 20 mostra que praticamente 100% do tempo o desvio padrão se manteve em torno de 1 µs. Porém, a maioria se manteve logo acima deste valor, o que não é o ideal. Ainda assim, o resultado foi satisfatório, considerando que a média do teste se manteve em 111 ns.

É importante salientar que o desempenho deste teste aparentar ter sido pior que o do primeiro teste da configuração anterior não foi esperado. Isso pode ter ocorrido devido ao *jitter* do *switch* não-PTP ter compensado os pacotes de forma a dar melhores resultados. Como os valores das médias do último meio minuto mostram que há bem menos picos neste último teste, e os outros dois testes com o *switch* não-PTP tiveram resultados diferentes do primeiro, é seguro assumir que os resultados do primeiro teste com o *switch* não-PTP foram uma anomalia.

4 CONCLUSÕES E TRABALHOS FUTUROS

Através deste trabalho, foi possível demonstrar a implementação de uma sincronização utilizando o *Precision Time Protocol*. O projeto foi desenvolvido com sucesso, utilizando referências bibliográficas e modelos já implementados para estudar e adquirir domínio sobre o uso de tal protocolo.

Conforme os testes realizados, o objetivo de alcançar uma sincronização menor que 1 µs foi atingido. Alguns desvios foram observados; o uso de um *switch* não compatível com o Protocolo de Tempo Preciso foi responsável por gerar um atraso aleatório e demonstrar tal atraso como um erro nos dados finais obtidos. Ademais, há uma necessidade de aprimorar a estabilidade da sincronização, pois os resultados exibiram um desvio padrão acima do limite de 1 µs por um tempo considerável, apesar da média exibir um resultado abaixo deste valor por praticamente todo o tempo de execução dos testes.

Além dos problemas de projeto, a rede utilizada para executar os testes apresentava congestionamento em horários comerciais de trabalho, dificultando a sincronia entre os dispositivos; nota-se um maior atraso (causado por tal congestionamento) ao comparar estes resultados com testes realizados fora de horário comercial.

No caso de sincronizar o controlador da REIVAX, os resultados dos testes atenderam os requisitos para uma rede similar à que ele estará inserido; desta forma, este trabalho será utilizado como referência para a futura implementação do Protocolo de Tempo Preciso no controlador, o que suprirá a necessidade de manter precisão na sequência de eventos do mesmo.

Para aprimoramento e trabalhos futuros, uma implementação do programa PTPd junto com o NIC I210 em um RTOS deve ser o suficiente para apresentar melhorias na qualidade da sincronização (objetivo inicial deste trabalho). Utilizar um switch PTP para realizar os testes também seria desejável, pois este permitiria utilizar outras configurações estabelecidas pela norma.

REFERÊNCIAS

ARNOLD, D. *End-to-End Versus Peer-to-Peer*. Abril 2018. <http://blog.meinbergglobal.com/2013/09/19/end-end-versus-peer-peer/>. Acessado em 2018-04-23.

ARNOLD, D. *One-step or Two-step?* Abril 2018. <https://blog.meinbergglobal.com/2013/10/28/one-step-two-step/>. Acessado em 2018-04-23.

ARNOLD, D. *What Are All Of These IEEE 1588 Clock Types?* Abril 2018. <https://blog.meinbergglobal.com/2013/10/21/ieee-1588-clock-types/>. Acessado em 2018-04-23.

ARNOLD, D. *Why is IEEE 1588 so accurate?* Abril 2018. <http://blog.meinbergglobal.com/2013/09/14/ieee-1588-accurate/>. Acessado em 2018-04-23.

FERENCZ, B. *Hardware Assisted IEEE 1588 Clock Synchronization Under Linux*. Tese (Doutorado) — Budapest University of Technology and Economics, Budapest, 2013.

IEEE. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. *IEEE Std. 1588-2008*, 2008.

IEEE. IEEE standard for synchrophasor measurements for power system. *IEEE Std. C37.118.1-2011*, 2011.

INTEL CORPORATION. *Intel Ethernet Controller I210 Datasheet*. 3.3. ed. [S.l.], 2018.

JARC, A. *Testing various PTP setups (Part I)*. Maio 2018. <https://blog.meinbergglobal.com/2015/12/18/1588-ptp-test-i/>. Acessado em 18/05/2018.

MILLS, D. L. *How NTP Works*. Novembro 2018. <https://www.eecis.udel.edu/ mills/ntp/html/warp.html>. Acessado em 08/11/2018.

REASON TECNOLOGIA. *Manual Técnico RT430/434*. ver08b. Florianópolis, SC, Brasil, 2018.