INSTITUTO FEDERAL DE SANTA CATARINA

MÁRIO ALLAN LEHMKUHL DE ABREU

Análise das implementações em hardware do protocolo de sincronização IEEE 1588-2008(v2)

São José - SC

Dezembro/2019

RESUMO

Sincronização é um problema clássico de sistemas distribuídos que vem recebendo cada vez mais atenção devido a expansão de aplicações no campo de internet da coisas (Internet of Things (IoT)), cidades inteligentes e outras aplicações, que fazem uso intensivo das redes de comunicação para sensoriamento e controle. Cada cenário possui seus requisitos de sincronização e protocolos dedicados para suprir essa necessidade. Um dos mais utilizados é o *Network Time Protocol* (NTP), o qual começou sendo utilizado para sincronizar horários locais e é utilizado por vários sistemas de uso geral, como por exemplo, celulares e computadores. Entretanto, em algumas aplicações, como exemplo, medições no sistema elétrico, é necessário sincronizar relógios de equipamentos que estão há vários quilômetros de distância com uma precisão de nanosegundos, o que limita o uso do NTP, uma vez que esse possui precisão na casa dos milissegundos. Nesse contexto, vem se tornando popular o uso do PTPv2 (*Precision Time Protocol*), especificado pelo padrão Institute of Electrical and Electronic Engineers (IEEE) 1588-2008(v2), que utiliza uma rede de pacotes para fornecer sincronização de frequência, fase e tempo. A precisão dessa sincronização chega a nanosegundos no caso da rede seguir as orientações do padrão IEEE 1588-2008(v2). Para alcançar essa precisão, é necessário que os pacotes do protocolo de sincronização sejam analisados assim que entram no equipamento, em geral, a implementação ocorre no hardware entre a camada física e a subcamada MAC da pilha de comunicação. Isso permite eliminar atrasos causados pelos drivers e do próprio sistema operacional. Sendo assim, esse trabalho propõe a avaliação de dois *IP Cores*, um de código aberto e outro comercial, que implementam o PTPv2. Para isso será criado um cenário para sincronização de relógio entre duas placas de desenvolvimento compostas por FPGA e interface de rede, uma placa será o mestre e a outra o escravo. A medição do tempo de sincronismo será analisada a partir de um osciloscópio conectado em ambas as placas que vai verificar se o sincronismo foi executado com sucesso. Ao final, será comparado os resultados de cada versão para concluir qual entregou o melhor resultado.

Palavras-chave: Protocolo de sincronização de tempo. Precision Time Protocol (PTP) IEEE-1588-2008(v2). Sincronização de tempo em Hardware. Field Programable Gate Array (FPGA).

LISTA DE ILUSTRAÇÕES

Figura 1 – Funcionamento de trigger de botão. 15
Figura 2 – Trigger com um timestamp para 12:00 PM. 16
Figura 3 – FTE ajustada para criar um trigger às 12:00 PM. 16
Figura 4 – Representação do clock. 16
Figura 5 – Sample clock ajustando o instante em que as amostras são reunidas de um sinal. . . . 17
Figura 6 – Relógios derivados de um relógio de referência. 17
Figura 7 – Representação da configuração entre mestre e escravo. 17
Figura 8 – Aquisição assíncrona. 18
Figura 9 – Aquisição síncrona. 18
Figura 10 – Relógio impreciso e instável de 10 MHz. 19
Figura 11 – Relógio preciso e instável de 10 MHz. 20
Figura 12 – Relógio impreciso e estável e de 10 MHz. 20
Figura 13 – Relógio preciso e estável de 10 MHz. 21
Figura 14 – Domínio de relógios separado. 21
Figura 15 – Desvio do relógio. 22
Figura 16 – Domínios comuns do relógio. 22
Figura 17 – Inclinação do relógio. 23
Figura 18 – Oscilação do relógio. 23
Figura 19 – Composição do sinal IRIG. 25
Figura 20 – Forma de transmissão do sinal elétrico no IRIG-B. 25
Figura 21 – Diagrama dos pontos de atraso não determinados. 29
Figura 22 – Diagrama de auxílio de hardware do protocolo PTP. 30
Figura 23 – Relação entre medições de retardo ponto a ponto e ponto a ponto. 31
Figura 24 – Exemplo de topologia de rede PTP. 32
Figura 25 – Exemplo de hierarquia de relógio PTP. 32
Figura 26 – Diagrama de tempo das mensagens do protocolo PTP. 33
Figura 27 – Exemplo do calculo das mensagens do protocolo PTP. 34
Figura 28 – IP Core da Intel do PTPv2 com foco no bloco da sincronização do relógio. 35
Figura 29 – IP Core do HA1588 do PTPv2 com foco no bloco da sincronização do relógio. 36
Figura 30 – FPGA Intel DE2-115 37
Figura 31 – Osciloscópio 37
Figura 32 – Cenário básico para executar os IP Cores do PTPv2 para demonstrar o sincronismo de tempo entre os FPGAs mestre e escravo. 38

LISTA DE TABELAS

Tabela 1 – Formatos do padrão IRIG. 24
Tabela 2 – Resumo dos protocolos de tempo IRIG-B, NTP e PTP. 34
Tabela 3 – Cronograma das atividades previstas 40

LISTA DE ABREVIATURAS E SIGLAS

IoT Internet of Things 1
IEEE Institute of Electrical and Electronic Engineers 1
PTP Precision Time Protocol 1
FPGA Field Programable Gate Array 1
HDL Hardware Description Language 11
IP' Propriedade Intelectual 11
MAC Controle de Acesso ao Meio 11
FTE Future time event 16
IED Intelligent Electronic Device 23
IEC International Electrotechnical Commission 23
IRIG Inter-Range Instrumentation Group 24
RCC Range Commanders Council 24
GPS Global Positioning System 24
PPS Pulso Por Segundo 24
BCD Binary Coded Decimal 25
PMU Phasor Measurement Units 25
NTP Network Time Protocol 26
MU Merging Unit 26
ONS Operador Nacional do Sistema Elétrico 26
GOOSE Generic Object Oriented Substation Event 27
SNTP Simple Network Time Protocol 27
IP Internet Protocol 27
TC Transparent Clock 28
CPU Central Processing Unit 29
TCP Transmission Control Protocol 29
OC Ordinary Clock 30
BC Boundary Clock 30
E2E End-To-End 30
P2P Peer-To-Peer 30
HMI Human Machine Interfaces 31
RTT Round Trip Time 32
IFSC Instituto Federal de Santa Catarina 35
SJ São José 35
EPROM Erasable Programmable Read-Only Memory 36
DUT Design Under Test 38
TCC Trabalho de Conclusão de Curso 40

SUMÁRIO

1 INTRODUÇÃO . . . 11
1.1 Objetivo geral . . . 13
1.2 Objetivos específicos . . . 13
1.3 Motivação . . . 13
1.4 Organização do texto . . . 13

2 FUNDAMENTAÇÃO TEÓRICA . . . 15
2.1 Conceitos básicos de tempo e sincronização . . . 15
2.2 Protocolos de sincronização de tempo . . . 23
2.2.1 IRIG-B . . . 24
2.2.2 NTP . . . 26
2.2.3 PTP . . . 28
2.2.3.1 Terminologias e dispositivos PTP . . . 30
2.2.3.2 Funcionamento do Protocolo PTP . . . 32
2.2.4 Resumo dos protocolos sincronização de tempo IRIG-B, NTP e PTP . . . 34
2.3 Projetos do PTPv2 implementados em IP Core comercial e de código aberto . . . 35
2.3.1 Projeto Intel do IP Core PTPv2 . . . 35
2.3.2 Projeto HA1588 do IP Core PTPv2 hospedado no OpenCores . . . 35
2.4 Cenário proposto para execução dos IP Cores do PTPv2 . . . 36
2.4.1 FPGA . . . 36
2.4.2 Osciloscópio . . . 37
2.4.3 Cabo de rede ethernet . . . 37
2.4.4 Cenário . . . 38

3 PROPOSTA . . . 39
3.1 Metodologia . . . 39
3.1.1 Compreensão do protocolo PTPv2 . . . 39
3.1.2 Projeto Intel do PTPTv2 . . . 39
3.1.3 Projeto HA1588 do PTPTv2 hospedado no OpenCores . . . 39
3.1.4 Execução dos IP cores no cenário básico dos FPGAs mestre e escravo . . . 39
3.1.5 Validação e testes . . . 39
3.2 Cronograma . . . 40

REFERÊNCIAS . . . 41

1 INTRODUÇÃO

Em uma rede de comunicação industrial, como por exemplo no ramo de energia ou telecomunicações, o tempo é um componente essencial utilizado principalmente na sincronização precisa dos relógios dos dispositivos de uma rede distribuída, normalmente de controle e medição de falhas. No sistema elétrico, por exemplo, esses dispositivos são distribuídos em vários pontos de uma região, onde ambos estão há vários quilômetros uns dos outros, no qual o status e alertas de uma informação deve ser informado o mais rápido possível, preferencialmente em tempo real entre todos os pontos. A sincronização dos relógios desses dispositivos, interligados por uma rede distribuída de cabeamento *ethernet* e coaxial, ocorre a partir de protocolos de sincronização de tempo. Nesses ambientes devido o tamanho dos seus sistemas, é necessário um alto valor de precisão no sincronismo de tempo, exigindo valores em nanossegundos. Já em redes comerciais e domésticas, pelo fato de não depender de alta precisão, o tempo de sincronismo dos relógios entre a comunicação de computadores e servidores atingem valores em milissegundos, sendo nesse contexto, considerados aceitáveis.

Uma métrica fundamental da precisão do relógio é a frequência. Na teoria, um relógio sempre levará a mesma quantidade de ciclos para representar um segundo. Porém na prática, os relógios localizados nos elementos de rede não são perfeitos. Sua precisão varia de acordo com os padrões de fabricação e temperatura. Assim com o uso contínuo desses equipamentos, seus relógios começam a perder o sincronismo. Essa alteração ocorre devido há mudança de frequência. Portanto, para garantir a sincronização dos relógios, um protocolo de sincronização de tempo deve funcionar para manter o tempo e a frequência estáveis por longos períodos (NEWS, 2014).

Nesse contexto o PTP (Protocolo de Precisão de Tempo) especificado no seu último padrão, o IEEE 1588-2008(v2), também conhecido como PTPv2, é a mais recente tecnologia de temporização baseada no padrão *ethernet*. pelo modelo de comunicação mestre e escravo, ele registra mensagens de data e hora com precisão a partir da camada física (Suporte por *hardware*), aproveitando os circuitos de *hardware* (QI; WANG; YANG, 2011). Projetado para aplicações críticas de automação industrial, fornece o mais alto nível de precisão de frequência, fase e hora do dia. Entregando uma precisão na faixa de nanossegundos, o PTP elimina os problemas de latência e atraso, a partir de uma técnica onde o mestre e o escravo se conversam para eliminar um atraso medido entre os dois nós, reduzindo os efeitos da latência da rede (SAMBERG, 2017), (PEDRETTI et al., 2019). É bastante utilizado em redes de controle e medição industrial, pois os mecanismos necessários podem ser implementados com o uso mínimo de hardware (KYRIAKAKIS; SPARSO; SCHOEBERL, 2018).

Para que o PTP alcance a precisão de nanosegundos, é necessário que os pacotes do protocolo de sincronização sejam analisados assim que entram no equipamento, em geral, a implementação ocorre no *hardware* entre a camada física e a subcamada Controle de Acesso ao Meio (MAC) da pilha de comunicação. Isso permite eliminar atrasos causados pelos *drivers* e do próprio sistema operacional. Dessa forma, para dispor do máximo de precisão do PTP, é possível utilizar *chipsets* dedicados que implementam o padrão ou utilizar blocos descritos em Hardware Description Language (HDL), também conhecidos como Propriedade Intelectual (IP') Cores, dentro de FPGAs, que possibilitem a análise das mensagens com a menor latência possível. Dentre as duas alternativas disponíveis, no uso de IP' Cores do PTPv2 em FPGAs, são encontrados na internet implementações em versão comercial pelo projeto da *Intel* e de código aberto pelo projeto HA1588 hospedado no *OpenCores*. Oferecendo análise e execução desses projetos, contribuindo no estudo e desenvolvimento prático das características desse protocolo (ALTERA,

2016), (OPENCORES, 1999), (LUIZ et al., 2011).

Assim, este trabalho propõe a avaliação de dois IP' cores que implementam o padrão PTPv2 (IEEE 1588-2008), um comercial e outro de código aberto. A implementação comercial é disponibilizado pela empresa *Intel* e a de código aberto é disponibilizado no projeto HA1588 no site *OpenCores*. Para essa avaliação será criado um cenário para sincronização de relógio entre duas placas de desenvolvimento compostas por FPGA e interface de rede, uma placa será o mestre e a outra o escravo. A medição do tempo de sincronismo será analisada a partir de um osciloscópio conectado em ambas as placas que vai verificar se o sincronismo foi executado com sucesso. Ao final, será comparado os resultados de cada versão para concluir qual entregou o melhor resultado.

1.1 Objetivo geral

Analisar e executar o protocolo PTP, especificado no seu ultimo padrão IEEE 1588-2008(v2), mediante ao projeto da empresa *Intel* e do projeto HA1588 hospedado no *OpenCores*. Para demonstrar a sincronização do relógio com uma precisão em nanossegundos, essas implementações são executadas no cenário básico constituído por duas placas de desenvolvimento FPGA, mestre e escravo. A comunicação entre as placas acontece ao longo de um cabo de rede *ethernet*. No final, é comparado os resultados de cada versão para concluir quem entrega o melhor resultado.

1.2 Objetivos específicos

- Compreender o protocolo de sincronização de tempo PTP baseado no padrão IEEE 1588-2008(v2).
- Analisar e executar o IP' Core (HDL) do PTPv2 do projeto da *Intel*.
- Analisar e executar o IP' Core (HDL) do PTPv2 do projeto HA1588 hospedado na *OpenCores*.
- Compreender a configuração do processador embutido Nios II, usado para implementar a função do algoritmo de trajetória de movimento do PTP.
- Executar os IP' Cores (HDL) do PTPv2 dos projetos da *Intel* e do HA1588, executados no cenário básico constituído por duas placas de desenvolvimento FPGAs mestre e escravo. A comunicação entre as placas ocorre a partir de um cabo de rede *ethernet*, no qual seus tempos são medidos por um osciloscópio.
- Comparar os resultados do projeto da *Intel* com a do projeto do HA1588 hospedado no *OpenCores*, para concluir quem entrega o melhor resultado.

1.3 Motivação

A motivação deste trabalho desenvolve-se perante ao fato de entender como ocorre a sincronização de tempo dos relógios entre equipamentos que depende de alta precisão, como é o caso por exemplo de equipamentos industrias de medição e controle de falhas usados nos setores de energia e telecomunicação. Esses dispositivos são muito sensíveis na questão do sincronismo de tempo, devido ao tamanho da infraestrutura desses sistemas. A distribuição desses dispositivos ocorre em vários pontos de uma região, o qual ambos estão há vários quilômetros uns dos outros, sendo que o status e alertas de uma informação deve ser informado o mais rápido possível, preferencialmente em tempo real entre todos os pontos.

1.4 Organização do texto

O texto está organizado da seguinte forma: No Capítulo 2 é apresentado a fundamentação teórica onde serão discutidos conceitos e ferramentas que serão necessários para a implementação desse trabalho. No Capítulo 3 é apresentada a proposta, que descreve como vai ser executada a metodologia para o desenvolvimento do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Conceitos básicos de tempo e sincronização

Em qualquer sistema, mas principalmente em aplicações de teste, controle e design o tempo é um item indispensável que fornece funções de medição, sendo limitado por variáveis como a distância. Com as tecnologias de tempo e sincronização é possível coordenar, cronometrar ou correlacionar eventos no tempo, relacionando os sinais entre si. Por meio da aquisição de dados, esses eventos podem ser várias amostras ou amostras entre vários sistemas. Em um sistema sincronizado, os componentes mais básicos são o *clock* (Relógio) que fornece o tempo no qual as amostras são geradas para os múltiplos sinais que precisam ser sincronizados. E os *triggers* (Gatilhos) que fornecem um ponto de partida para os sinais sincronizados (INSTRUMENTS, 2019b). A partir desses conceitos iniciais, para entender melhor o tempo e a sincronização, a seguir são mostrados os principais termos básicos envolvidos nesse cenário.

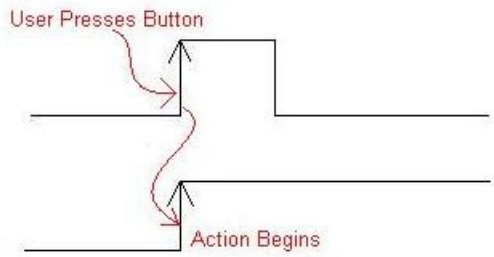
Event (Evento)

É um acontecimento de algum significado. No contexto de tempo e sincronização, os conceitos *event* e *trigger* são quase a mesma coisa. Para entender melhor sobre um evento, imagine um botão que liga algum tipo de equipamento, o acontecimento de apertar o botão é um evento que avisa o equipamento para iniciar (INSTRUMENTS, 2019b).

Trigger (Gatilho)

É um evento de *software* ou *hardware* que aciona um acontecimento, como começar ou parar uma aquisição. A Figura 1 ilustra como funciona um *trigger* de botão que inicia uma ação (INSTRUMENTS, 2019b).

Figura 1 – Funcionamento de trigger de botão.

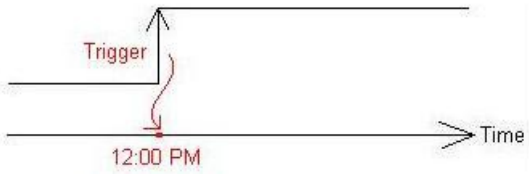


FONTE: (INSTRUMENTS, 2019b).

Timestamp (Registro de tempo)

É um evento correlacionado em um determinado tempo. Esse evento pode ser uma amostra de dados ou um sinal de *trigger*. A Figura 2 exibe um exemplo de como é um *trigger* com um *timestamp* para 12:00 PM (INSTRUMENTS, 2019b).

Figura 2 – Trigger com um timestamp para 12:00 PM.

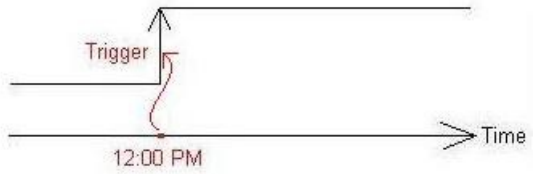


FONTE: (INSTRUMENTS, 2019b).

Future time event (FTE) (Evento de horário futuro)

É um evento criado por um usuário, como por exemplo um *trigger*, para ser executado em um determinado horário. A Figura 3 mostra um exemplo de um FTE ajustada para criar um *trigger* às 12:00 PM (INSTRUMENTS, 2019b).

Figura 3 – FTE ajustada para criar um trigger às 12:00 PM.

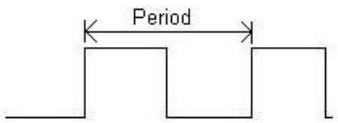


FONTE: (INSTRUMENTS, 2019b).

Clock (Relógio)

É uma sequência de eventos periódicos com espaçamentos iguais. A Figura 4 exibe um exemplo de como é representado um *clock* (INSTRUMENTS, 2019b).

Figura 4 – Representação do clock.



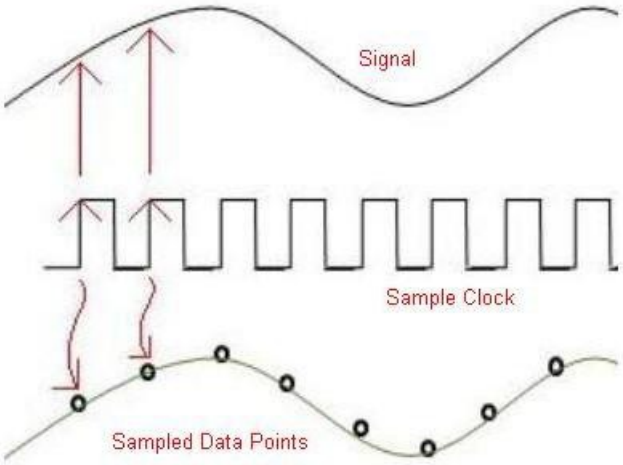
FONTE: (INSTRUMENTS, 2019b).

Nesse contexto, o relógio se caracteriza em dois tipos, o de amostra e o de referência.

- **Sample Clock (Relógio de amostra)**

  Em um equipamento de aquisição de dados, o *Sample Clock* controla o tempo da remoção de amostras desse dispositivo. A Figura 5 ilustra um exemplo de um *Sample Clock* ajustando o instante em que as amostras são reunidas de um sinal (INSTRUMENTS, 2019a), (INSTRUMENTS, 2019b).

Figura 5 – Sample clock ajustando o instante em que as amostras são reunidas de um sinal.

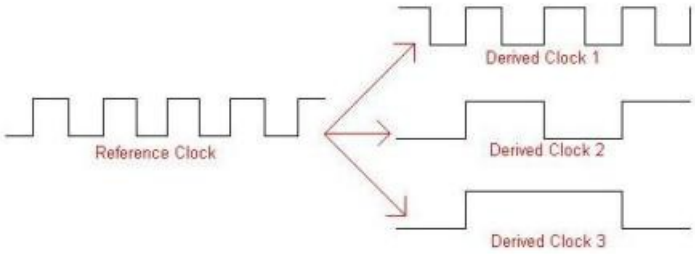


FONTE: (INSTRUMENTS, 2019b).

- **Reference Clock (Relógio de referência)**

  É um sinal de relógio escolhido como referência em que os relógios de outros sistemas derivam seus próprios sinais de relógio a partir dele. Os sinais desses relógios podem ser iguais, mais lentos ou mais rápidos que o relógio de referência, dependendo do valor que é multiplicado ou dividido em relação a ele. A Figura 6 exemplifica relógios derivados de um relógio de referência (INSTRUMENTS, 2019b).

Figura 6 – Relógios derivados de um relógio de referência.



FONTE: (INSTRUMENTS, 2019b).

Dispositivos Mestre e Escravo

Na criação de sistemas de medição sincronizados, é trivial escolher um equipamento como mestre e um ou mais equipamentos como escravos. O equipamento principal é quem cria um sinal ou sinais utilizados para controlar todos os equipamentos de medição no sistema. Os equipamentos *slaves* recebem os sinais de controle do equipamento mestre. A Figura 7 exibe como é representado a configuração entre mestre e escravo (INSTRUMENTS, 2019b).

Figura 7 – Representação da configuração entre mestre e escravo.

FONTE: (INSTRUMENTS, 2019b).

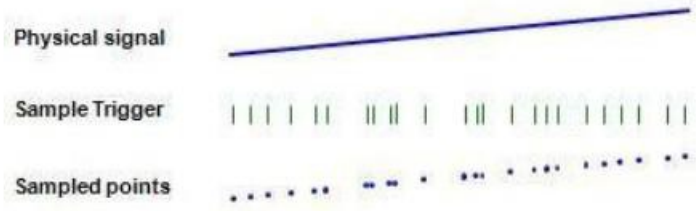
Operações de sincronização

As operações de sincronização se caracterizam em dois tipos, em operações assíncronas e síncronas.

- **Operações assíncronas**

  São acontecimentos que ocorrem sem sincronização com o relógio de referência ou cronômetro, isto é, ocorrem em qualquer momento. A Figura 8 exemplifica uma aquisição assíncrona, onde amostras são executadas a qualquer momento e dessa forma os pontos de dados não são relacionados ao tempo. Como exemplos de operações assíncronas, temos a transferência de qualquer tipo de dados em que o *handshake* é usado e a amostragem sob demanda (INSTRUMENTS, 2019b).

Figura 8 – Aquisição assíncrona.

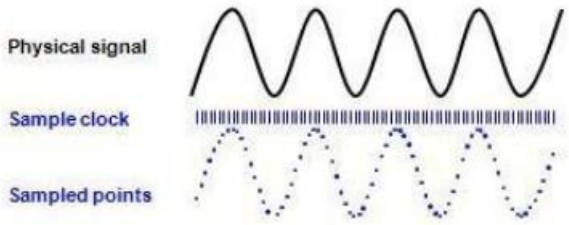


FONTE: (INSTRUMENTS, 2019b).

- **Operações síncronas**

  É quando em um relógio periódico e compartilhado, os dados são criados ou medidos a um valor constante. A Figura 9 exibe um exemplo de uma aquisição síncrona (INSTRUMENTS, 2019b).

Figura 9 – Aquisição síncrona.



FONTE: (INSTRUMENTS, 2019b).

Fontes de erro de temporização

É a descrição de como o tempo é medido ou fornecido. Exemplificando o conceito, é quando se mede um pulso perfeito de valor especificado de 1 ms, mas o resultado da leitura real é de 1,005 ms. A diferença do valor de 0,005 ms é o erro de temporização dessa medida. O erro de temporização também é utilizado na leitura de frequência. Por exemplo, em um relógio de amostra é especificado com frequência de 1 MHz, porém na leitura da frequência real desse relógio, encontra-se o valor de 1,002 MHz. Nesse caso, o valor da diferença entre as frequências de 0,002 MHz representa o erro de temporização.

O relógio trabalha com um determinado valor de precisão no qual essa característica tem o objetivo de informar se a frequência especificada corresponde com a frequência real do relógio. A precisão

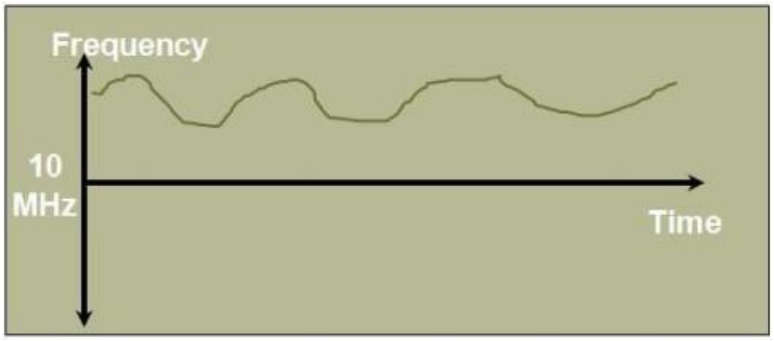
do relógio é informada em partes por milhão (ppm). Sendo assim, como exemplo, a precisão de um relógio com 5 ppm informa que a frequência média do relógio pode ser desativada em 5 Hz para cada 1 MHz do seu valor especificado. Dessa forma, com uma precisão de 5 ppm, um relógio de 1 MHz pode ter um erro de frequência de até 5 Hz. Já um relógio com a mesma precisão, mas com frequência de 2 MHz, pode ter um erro de frequência de até 10 Hz.

Outra característica do relógio é sobre a sua estabilidade que informa como sua frequência reage ao efeito de flutuação. Os motivos que tende a provocar a flutuação da frequência são as alterações de temperatura, o tempo de execução, a tensão de alimentação, a vibração, o choque e a carga capacitiva que o relógio deve acionar. No entanto a temperatura é a principal causa que interferi na estabilidade do oscilador de cristal (INSTRUMENTS, 2019b). Compreendendo a importância da precisão e da estabilidade de um relógio, a seguir são descritos quatro exemplos de tipos de relógios relacionados ao contexto de precisão, imprecisão, estável e instável:

- **Relógio impreciso e instável**

  Em um relógio de 10 MHz a frequência varia ao longo do tempo como exibido na Figura 10. O resultado obtido mostra que este relógio não é preciso pois o valor médio da frequência é maior que 10 MHz, e nem estável devido a flutuação da frequência, isto é, o valor da frequência se altera (INSTRUMENTS, 2019b).

Figura 10 – Relógio impreciso e instável de 10 MHz.

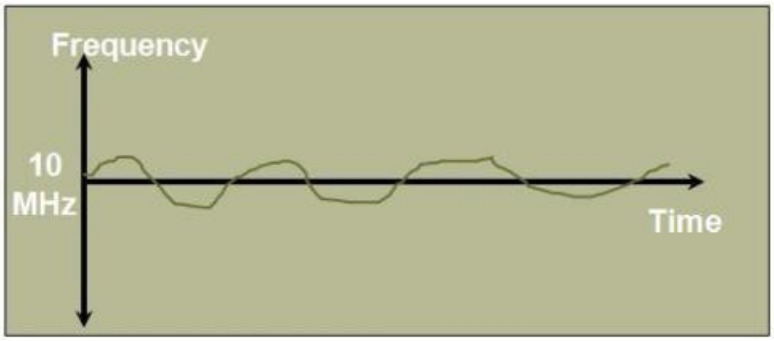


FONTE: (INSTRUMENTS, 2019b).

- **Relógio preciso e instável**

  A Figura 11 exibe a alteração do valor da frequência de um relógio de 10 MHz ao longo do tempo. O resultado desse caso mostra que o valor médio da frequência é muito próximo a 10 MHz, tornando esse relógio preciso. Porém, devido a frequência sofrer variação e não ficar em um valor constante, este relógio não é considerado estável (INSTRUMENTS, 2019b).

Figura 11 – Relógio preciso e instável de 10 MHz.

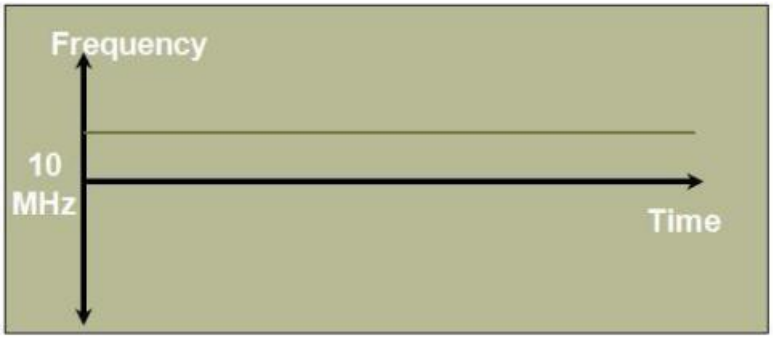


FONTE: (INSTRUMENTS, 2019b).

- **Relógio impreciso e estável**

  Novamente os valores da frequência de um relógio de 10 MHz são mostrados ao longo do tempo como exemplificado na Figura 12. Nota-se que o relógio não é preciso pois o valor médio da frequência é superior a 10 MHz. No entanto ele é estável porque a taxa da frequência não muda (INSTRUMENTS, 2019b).

Figura 12 – Relógio impreciso e estável e de 10 MHz.

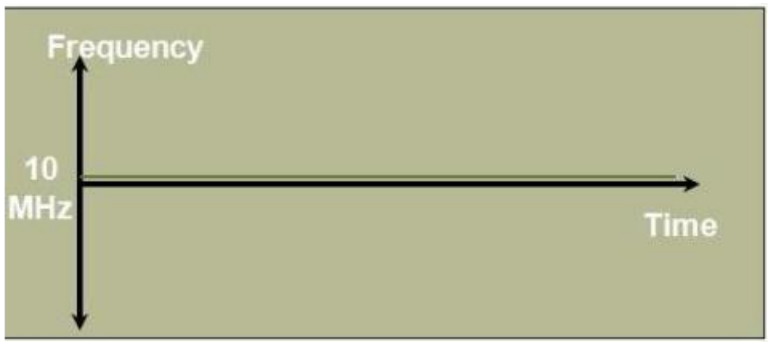


FONTE: (INSTRUMENTS, 2019b).

- **Relógio preciso e estável**

  A Figura 13 ilustra o último exemplo de um relógio de 10 MHz, no qual é mostrado os valores da frequência ao longo do tempo. Nesse caso o relógio é preciso pois seu valor médio de frequência é muito próximo de 10 MHz e também estável porque a frequência não varia, sendo assim, não sofre nenhuma flutuação (INSTRUMENTS, 2019b).

Figura 13 – Relógio preciso e estável de 10 MHz.

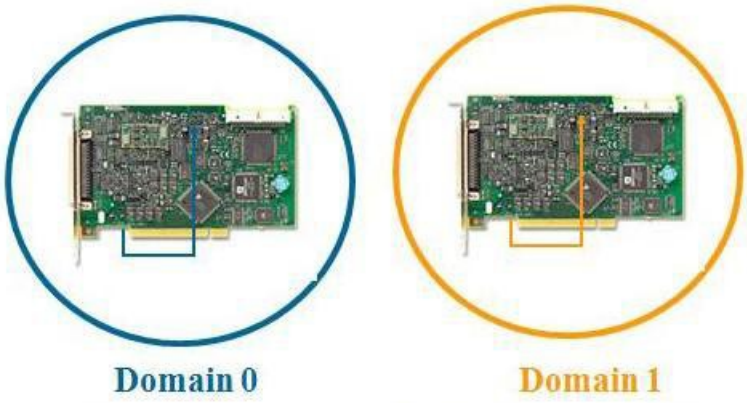


FONTE: (INSTRUMENTS, 2019b).

Relógios com domínios separados

Relógios com domínios separados, nada mais é do que relógios que tenham a fonte de seu tempo independente um do outro, ou seja, obtém o tempo a partir do seu oscilador de cristal, como mostrado na Figura 14. Entretanto em um cenário que necessita de sincronização precisa, mesmo que os relógios sejam idênticos e iniciados exatamente ao mesmo tempo, devido aos fatores de fabricação de seus osciladores de cristais, com o tempo, eles começam a apresentar pequenas variações na frequência, que em longo prazo, acarretam grandes diferenças e podem fazer com que os relógios se afastem um do outro. Esse problema se denomina de desvio do relógio (INSTRUMENTS, 2019b).

Figura 14 – Domínio de relógios separado.



FONTE: (INSTRUMENTS, 2019b).

Desvio do relógio

O Desvio do relógio (Drift) acontece quando dois equipamentos obtém dados com diferentes valores de amostragem. Mesmo configurando os equipamentos com o mesmo valor de amostragem, por exemplo 300 MHz, devido as maneiras de fabricação do oscilador de cristal, cada equipamento trabalha com uma taxa diferente de amostragem, não sendo idênticas e assim provocando o desvio do relógio. A Figura 15 mostra como ocorre o desvio de relógio (INSTRUMENTS, 2019b).

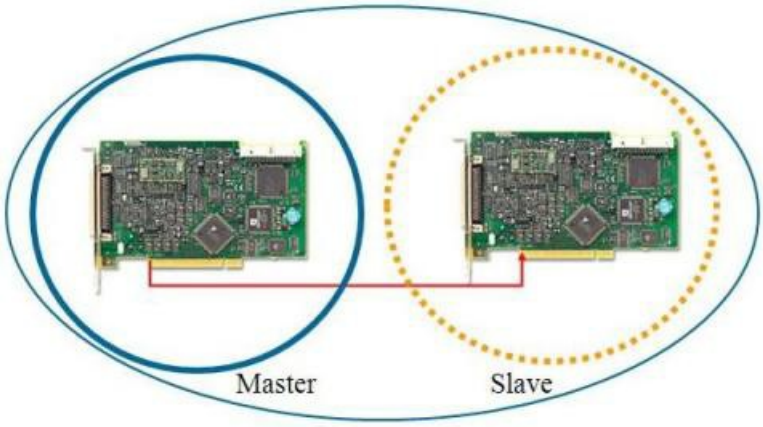
Figura 15 – Desvio do relógio.

FONTE: (INSTRUMENTS, 2019b).

Relógios com domínios comuns

É a utilização de um componente comum entre todos os outros componentes de um sistema, escolhido como o relógio de referência denominado de mestre, para sincronização dos relógios dos demais componentes, denominados de escravos. Por via dessa solução o desvio do relógio é eliminado. A Figura 16 exemplifica o uso de relógios com domínios em comum (INSTRUMENTS, 2019b).

Figura 16 – Domínios comuns do relógio.

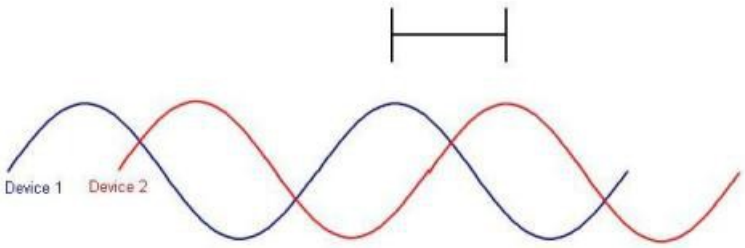


FONTE: (INSTRUMENTS, 2019b).

Inclinação do relógio

Em circuitos síncronos ocorre um fenômeno chamado inclinação do relógio, onde em diferentes dispositivos *slaves*, os sinais do relógio ou do *trigger* chegam em momentos diferentes. Vários motivos podem provocar esse fenômeno como por exemplo, o comprimento da interconexão do fio, as variações de temperatura, a variação nos dispositivos intermediários, o acoplamento capacitivo, as imperfeições do material e diferenças na capacitância de entrada nas entradas de relógio dos equipamentos que usam o relógio. Para que um circuito funcione adequadamente, com o aumento do valor do relógio, mínimas varições do tempo são toleradas. A Figura 17 exibe um exemplo do fenômeno inclinação do relógio (INSTRUMENTS, 2019b).

Figura 17 – Inclinação do relógio.



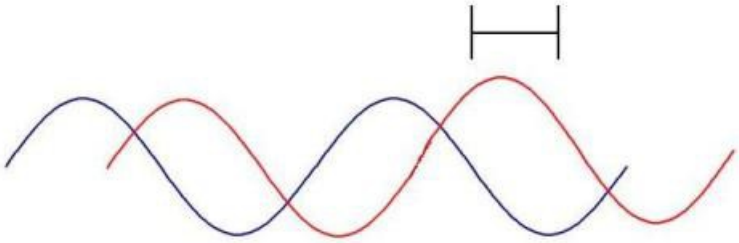
FONTE: (INSTRUMENTS, 2019b).

Para resolver a inclinação do relógio, uma solução encontrada, é por artifício de um sistema de linhas de sinal de relógio do tipo estrela, no qual todos os caminhos do sinal do relógio possuem o mesmo comprimento. Dessa forma a distribuição dos sinais, a partir dessa linhas, encontram seus destinos ao mesmo tempo, resolvendo o fenômeno da inclinação (INSTRUMENTS, 2019b).

Oscilação do relógio

A oscilação do relógio é o desvio do tempo ideal de um evento, calculado pelo cruzamento zero de um evento ou sinal de referência. A oscilação é normalmente resultado de fatores como diafonia, interferência eletromagnética, saídas de comutação simultâneas e outros sinais de interferência que geralmente acontecem. Como ao longo do tempo acontece variação do jitter[1], os valores e a quantidade de jitter podem mudar de uma estimativa visual, a uma medição da oscilação baseada no desvio padrão[2] ao longo do tempo. A Figura 18 mostra um exemplo de oscilação de relógio (INSTRUMENTS, 2019b).

Figura 18 – Oscilação do relógio.



FONTE: (INSTRUMENTS, 2019b).

2.2 Protocolos de sincronização de tempo

Muitos Intelligent Electronic Device (IED) (Dispositivos eletrônicos inteligentes) instalados em usinas e subestações de energia e de telecomunicações modernos executam aplicativos distribuídos que dependem de um tempo comum e preciso entre os IEDs. Alguns exemplos de aplicações que dependem de tempo preciso e sincronizado com taxas de atualização de micro a nanossegundos são sincrofasores, valores amostrados (International Electrotechnical Commission (IEC) 61850[3]-9-2[4]) e localização de falhas nas ondas de deslocamento. Para a comunicação e repasse do sinal de relógio entre os dispositivos, normalmente

[1] Jitter – É um desvio ou deslocamento de algum aspecto dos pulsos de um sinal digital (PINHEIRO, 2004).
[2] Desvio padrão – É uma medida que indica o quanto o conjunto de dados é uniforme (NOVAES, 2019).
[3] IEC 61850 – Norma responsável pela simplificação e potencialização dos controles nos sistemas elétricos de potência (VENTURELLI, 2017).
[4] IEC 61850-9-2 – Norma de Barramento de processos (JUNIOR et al., 2011).

nesse contexto utiliza-se o cabo coaxial. Entretanto outro padrão que começou a ser utilizado, é o padrão de cabo Ethernet, no qual antigamente era apenas utilizado na comunicação entre computadores. Porém com passar dos anos devido a implantação da sua infraestrutura em larga escala e também pelo conceito da industria 4.0, está se tornando a tecnologia de comunicação dominante para IEDs em subestações de energia, telecomunicações e ambientes industriais. No entanto por ser uma tecnologia baseada em pacotes, no início, o uso do padrão *ethernet* em subestações, se tornou um desafio para se obter alta confiabilidade, determinismo e disponibilidade (WATT et al., 2015). A partir desse cenário, as atuais soluções baseadas em protocolos de sincronismo temporal de alta precisão que se destacam e são implementadas entre os IEDs nos setores elétricos e de telecomunicações são:

- IRIG-B.
- NTP.
- PTP.

2.2.1 IRIG-B

Normalmente nas subestações de energia e de telecomunicações, o protocolo de sincronização de tempo mais utilizado é o IRIG-B, no qual utiliza um cabo coaxial dedicado para enviar os sinais de relógio aos IEDs, permitindo uma grande precisão temporal, alcançando valores de 100 nanossegundos a 1 microssegundo, desde que o sistema seja corretamente implementado. (WATT et al., 2015). Desenvolvido pelo Inter-Range Instrumentation Group (IRIG), grupo responsável pela criação de normas para o Range Commanders Council (RCC) das forças armadas dos Estados Unidos. O uso da primeira versão foi em 1960 e a versão mais recente no momento é o IRIG Standard 200-04, lançada em setembro de 2004. O IRIG-B usa o modelo mestre-escravo para a comunicação de informações de tempo entre os IEDs (LUIZ et al., 2011).

O IRIG é um protocolo de transmissão serial de tempo baseado em uma portadora com frequência fixada no Pulso Por Segundo (PPS)[5] e que envia informação por pulsos de largura variável, sendo que cada lado de subida representa um determinado instante no tempo. A norma IRIG definiu seis formatos representados por letras de acordo com taxa de atualização da informação como mostrado na Tabela 1 (ULIANA; ZIMATH, 2001).

Tabela 1 – Formatos do padrão IRIG.

Formato	**Taxa de atualização da informação**
IRIG-A	100 milissegundos
IRIG-B	1 segundos
IRIG-D	1 hora
IRIG-E	10 segundos
IRIG-G	10 milissegundos
IRIG-G	1 minuto

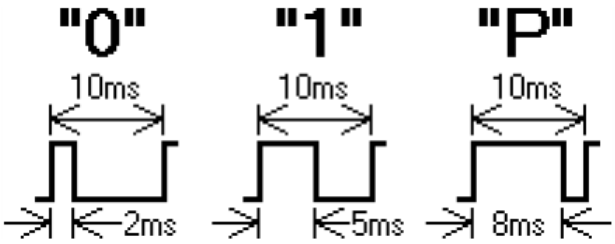
FONTE: (ULIANA; ZIMATH, 2001).

Dentre esses vários formatos, o fator que tornou o“IRIG-B” o formato mais difundido, foi devido a sua taxa de atualização de 1 segundo, no qual é obtida facilmente pelo recurso do sinal PPS gerado pelos receptores de Global Positioning System (GPS). O IRIG-B define um *frame* de dados com 100 bits

[5] PPS – Um pulso de sincronização muito preciso que não possui informações de hora ou data (INGRAM; SMELLIE, 2019).

de informação. Dessa forma a composição do sinal de relógio é feita por repetidos *frame* de 100 bits, onde esses bits podem ter três valores distintos, 0 (Zero), 1 (Um) e P como ilustrado na Figura 19 (ULIANA; ZIMATH, 2001).

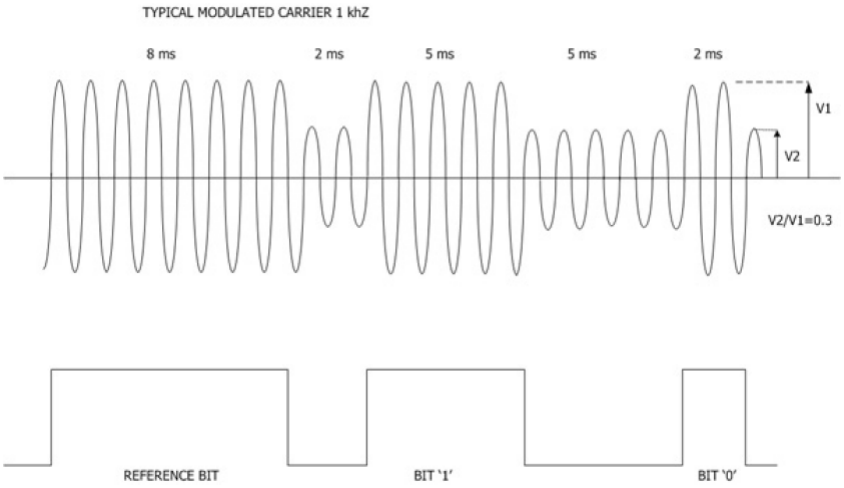
Figura 19 – Composição do sinal IRIG.



FONTE: (ULIANA; ZIMATH, 2001).

O uso do valor "P" serve como separador de bytes de dados, determinando o começo do *frame* de dados, além da posição do PPS dentro do *frame*. O IRIG-B transmite informações codificadas em Binary Coded Decimal (BCD) que possuem a informação de dia, hora, minuto e segundos, onde esses *frame* são transmitidos por sinais elétricos ou ópticos, como exemplificado na Figura 20 (LUIZ et al., 2011).

Figura 20 – Forma de transmissão do sinal elétrico no IRIG-B.



FONTE: (LUIZ et al., 2011).

Com a característica de cada letra, também podem ocorrer combinações de tipo de modulação de frequência (pulso, amplitude modulada sobre uma senoidal, Manchester), frequência da portadora e a presença de campos opcionais. Mesmo com a padronização destas características, ainda falta uma definição de outras características importantes como níveis elétricos, impedâncias e formatos de conectores. Com a chegada das unidades de medição fasorial Phasor Measurement Units (PMU), a composição do frame foi atualizado pela norma IEEE 1344[6] adicionando informação de ano, fuso horário, horário de verão e qualidade do tempo transmitido, ficando o frame dependendo de duas normas (LUIZ et al., 2011).

Dessa forma o uso do IRIG na sincronização de tempo alcança um valor de precisão na casa de a 100 ns a 1 μs, porém é preciso um maior detalhamento do projeto para obter essa taxa de precisão.

[6] IEEE 1344 – Padrão que define parâmetros para sincrofasores para sistemas de potência (MARTIN, 1995).

Entretanto também devem ser analisados as características da arquitetura como a topologia, onde vários equipamentos são ligados a um relógio, o uso de distribuidores e transceivers, isolação elétrica entre outros. Um ponto importante aos custos do projeto, é que deve-se considerar a rede IRIG exclusiva para sincronização do tempo do relógio, como também a rede de comunicação de dados (LUIZ et al., 2011).

2.2.2 NTP

O Network Time Protocol (NTP), criado pelo professor David L. Mills da Universidade de Delaware em 1980, é utilizado para sincronizar os relógios entre computadores em uma rede de comunicação a partir do bastante difundido padrão *ethernet*. Assim quase todos os computadores conectados à internet executando *windows* ou *linux* são fornecidos com sincronização de hora NTP, onde se tornou o protocolo de sincronização de tempo mais usado no mundo. Pela facilidade de usar o NTP para definir a hora local, dispositivos de comunicação de subestação de energia e de telecomunicações também começaram a usar esse protocolo, onde nesses equipamentos, as precisões da sincronização de milissegundos são aceitáveis. O NTP usa o modelo cliente-servidor para a comunicação de informações de tempo entre os dispositivos. A função de um servidor é fornecer tempo preciso aos seus clientes. Os clientes individuais executam um pequeno programa como uma tarefa em segundo plano que consulta periodicamente o servidor para obter informações precisas de tempo. A taxa de atualização dessas consultas geralmente ocorre a cada 15 minutos para manter a precisão da sincronização da rede (WATT et al., 2015). Outros detalhes do NTP são:

- Transmite a partir da rede *ethernet* o sinal de tempo de relógio. Dessa forma, possui funções de medição do tempo de envio e recebimento das mensagens, no qual permite a compensação do atraso causado pelo cabo de rede ou fibra, *switches* (Comutadores), conversores óptico-elétrico e demais equipamentos de rede.
- Ajusta o tempo do relógio a ser sincronizado de forma monotônica[7] a partir de um algoritmo, que não copia simplesmente para o relógio o tempo recebido na mensagem NTP, mas também adianta ou atrasa o tempo até que a diferença diminua. No entanto isso faz com que ocorra um atraso na sincronização dos tempos dos diferentes relógios.
- Medição do desvio do relógio (*Drift*) a ser sincronizado e a utilização desta informação para estabilizar o tempo quando acontece a perda da sincronização com o relógio de referência.

Os cuidados e complexidades do NTP surgiram com a necessidade em se ter a estabilidade do relógio em situações onde ocorria a perda do sinal da fonte de tempo, pois o funcionamento do protocolo se baseia na sincronização do relógio de um computador ao outro e não no envio preciso de informações de tempo. Dessa forma a primeira versão da norma IEC 61850 escolheu o NTP como forma de sincronização de IEDs. No entanto devido a sua forma de sincronização, acabou gerando confusão quanto ao seu entendimento. Ao se optar pelo uso do NTP, foi reutilizado a estrutura de comunicação que já existia para a comunicação entre IEDs, não sendo previsto as aplicações que estavam surgindo e que não foram abordadas pela primeira versão da IEC 61850, como por exemplo a tecnologia de medição fasorial. No capítulo 9-2 da norma IEC 61850 que relata da transmissão de valores amostrados pela rede Merging Unit (MU), é abordado a sincronização com um sinal de PPS, reconhecendo que o NTP não atende a necessidade de tempo preciso necessário para esta aplicação. Mesmo sabendo dessa informação, o NTP seguiu em uso para a sincronização de IEDs em subestações de energia e de telecomunicações com o objetivo de sincronizá-los dentro de uma precisão informada pelos procedimentos de rede do Operador

[7] Forma monotônica – Não tem “pulos” no tempo, para não haver descontinuidades (LUIZ et al., 2011).

Nacional do Sistema Elétrico (ONS) de até 1 ms. Em laboratório, testes mostraram que é possível obter esse valor de precisão, porém condições especificas devem ser consideradas:

- Uso de um relógio preciso de boa qualidade, pois há relógios em que o formato da mensagem NTP obedece a norma, no entanto a precisão não alcança o valor de 1 milissegundo, apenas taxas a partir de 10 a 100 milissegundos.
- Utilização de uma rede onde os *switches* tenham baixo trafego de informação, principalmente de mensagens Generic Object Oriented Substation Event (GOOSE)[8], pois essas mensagens tem preferência sobre outras que trafegam pela rede. Caso uma mensagem GOOSE chegue após uma mensagem NTP, esta é transmitida antes, passando na frente do NTP e causando atrasos com taxas de milissegundos.
- Correto processamento do cliente NTP nos IEDs, pois há alguns deles que interpretam as mensagens NTP, contudo não conseguem copiar corretamente a mensagem para seu relógio interno, ocasionando erros também da ordem de milissegundos.

Mesmo que todos os equipamentos informem que estão devidamente sincronizados, mesmo possuindo o acumulo das origens de erros citadas, é típico encontrar instalações que apresentam diferenças de tempo em diferentes equipamentos na ordem de vários milissegundos. Outro ponto importante é que na maioria dos atuais IEDs não é implementado o NTP, mas sim o Simple Network Time Protocol (SNTP), uma versão simplificada do NTP, no qual não faz uso de alguns algoritmos e que após grandes períodos de tempo não guarda o estado, tendo como prejuízo baixa precisão tempo. Como resultado, baseado em escolhas incorretas, percebe-se grandes dificuldades na análise de eventos e nas decisões sobre as arquiteturas de sincronização. De tal forma que adotam arquiteturas de sincronização não suportadas corretamente pelos IEDs. Por exemplo, pensando na redundância é comum a instalação de relógios NTP com duas portas NTP em uma rede, contudo, são poucos os IEDs que permitem a configuração de dois endereços diferentes de relógios. Com isso, há instalações em que um relógio com duas portas *ethernet* são configuradas para um mesmo Internet Protocol (IP), no qual se ocorrer a falta da conexão em uma das portas, imediatamente a mensagem será respondida pela outra. Entretanto, essa solução não é comum do NTP e pode ocasionar erros quando uma mensagem é respondida por mais de uma porta ao mesmo tempo. A experiência adquirida com o uso do NTP em subestações de energia e de telecomunicações revela as seguintes necessidades por parte dessas indústrias (LUIZ et al., 2011):

- A necessidade de sincronização de IEDs em subestações é algo essencial, não somente pelo requisito dos procedimentos de rede, mas também porque é necessário para operar-se o sistema. Necessitando ser confiável e monitorável, pois na ocorrência de alguma falha, o problema deve ser corrigido rapidamente.
- O uso do mesmo cabeamento de comunicação IEC 61850 é visto como uma redução de custo muito grande, não somente pela não necessidade de se criar uma rede de sincronismo dedicada, mas também pela redução do custo de engenharia envolvido, como a capacidade de saída dos relógios, distintos protocolos e configurações específicas em relógios e clientes.
- Nos IEDs, a necessidade de uma maior confiabilidade da sincronização a partir da monitoração do estado dos relógios quanto da possibilidade de redundância.

[8] GOOSE – São mensagens do tipo multicast que carregam informações entre os IEDs. São responsáveis apenas pelo tráfego de mensagens que informam sobre a atuação de qualquer proteção ou sinal digital (SOUZA, 2012).

- Necessidade de uma maior precisão para sincronização de novas aplicações como PMUs, MUs e localizadores de defeitos por ondas viajantes.

Pelos procedimentos citados acima, nota-se que eles não são atendidas nem pelo NTP (principalmente pela sua baixa precisão) e nem pelo IRIG-B (devido à necessidade de uma rede dedicada de sincronismo). Contudo esses pontos podem ser claramente atendidas por um protocolo implementado em 2002, de uso comprovado em diversas indústrias, principalmente as de energia e de telecomunicações, no qual já está em uso no mercado, PTP (LUIZ et al., 2011).

2.2.3 PTP

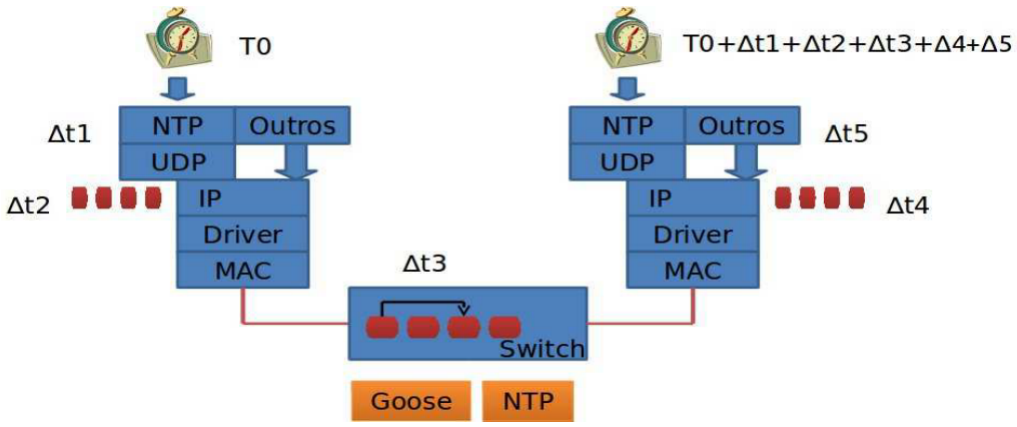
O protocolo PTP (Protocolo de tempo de precisão) surgiu por causa da necessidade da indústria de automação em obter dados sincronizados com uma elevada precisão. Sua primeira versão foi publicada pela IEEE em 2002, especificado no padrão IEEE 1588-2002, que distribui o tempo com uma precisão de microssegundos por via do padrão *ethernet* (Cabo *ethernet* ou fibra), que como já citado anteriormente está se tornando a tecnologia dominante para a comunicação entre os IED. Desta forma torna-se extremamente flexível e pode ser usado para uma variedade de aplicações de sincronização de tempo. Porém para alcançar a precisão especificadas pelo PTP, é importante entender como à assimetria da rede, as variações do atraso da mensagem e a topologia da rede afetam a exatidão e a precisão (WATT et al., 2015). O PTP é baseado no modelo de troca de informações mestre e escravo, em que mensagens são trocadas para determinar o atraso relativo ao tráfego na rede entre o relógio do dispositivo mestre em relação ao relógio do dispositivo escravo e compensar o *offset* (Deslocamento).

O PTP suporta vários relógios com capacidade de mestre, mas eles votam entre si para escolher um único relógio mestre, descrito como o ”grão-mestre”. Se o grão-mestre falhar ou sofrer um problema de desempenho, qualquer outro relógio com capacidade para mestre na rede passará a ser o grão-mestre se tiver uma melhor precisão. O tempo dessa troca varia, de acordo com as configurações de PTP, onde geralmente é inferior a 5 segundos (INGRAM; SMELLIE, 2019).

Em 2008 foi publicada a versão 2 do PTP, especificado no padrão IEEE 1588-2008(v2), também conhecido como PTP(v2), tendo como novidade a introdução de um tipo especial de comutador *ethernet* (*switch*) chamado de ”Transparent Clock (TC)”, que mede o tempo de permanência das mensagens sincronizadas de tal forma que o tráfego de rede PTP não precisa ser priorizado em relação a outro tráfego, simplificando o design da rede. Outra novidade é o suporte por *hardware* para transmissão de *timestamp* (Registro de data e hora), isto é, o registro do tempo é feito a partir da camada física, sendo desta forma, imune aos atrasos de processamento nos relógios, *switches* e IEDs, permitindo alcançar uma precisão em nanossegundos. Contudo esse valor só é atingido se todos os elementos do sistema implementados suportarem amostragem de tempo de hardware (LUIZ et al., 2011). Essa nova versão teve como objetivo atender as necessidades de outras aplicações, principalmente as de energia, com o uso de um protocolo padronizado. As duas versões são compatíveis e se conversam entre si, porém este trabalho será focado no PTP(v2) pois é o mais implementado no mercado (WATT et al., 2015), (INGRAM; SMELLIE, 2019).

O principal diferencial do PTP em relação ao NTP é que ele oferece suporte por *hardware* (IEEE 1588-2008(v2)) para transmissão de *timestamp*. No entanto o PTP também transmite o *timestamp* por *software*, porém essa implementação tem como desvantagem a adição de pontos de imprecisões não considerados no cálculo do atraso, afetando o resultado de precisão e não alcançando valores em nanossegundos. A Figura 21 mostra os pontos que acrescentam atrasos não determinados no caso do NTP e do PTP sem suporte em hardware, sendo assim, não podendo ser compensados de forma a zerar o atraso (LUIZ et al., 2011).

Figura 21 – Diagrama dos pontos de atraso não determinados.



FONTE: (LUIZ et al., 2011).

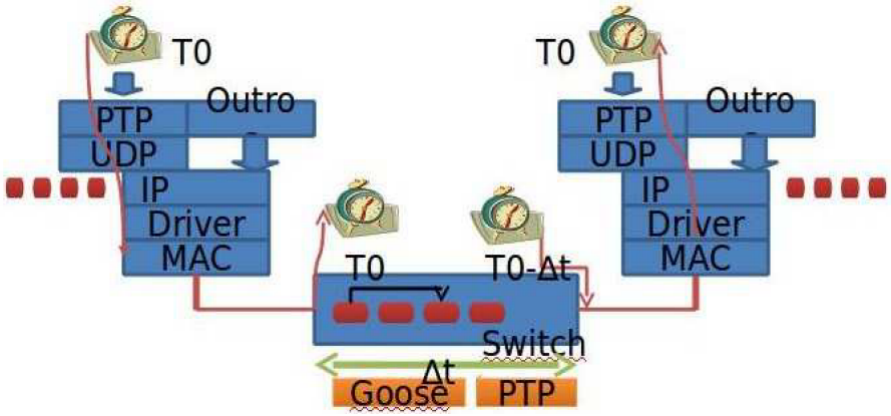
Conforme as descrições abaixo, os atrasos Δt1, Δt2, Δt3, Δt4 e Δt5 possuem período variável:

- Δt1 e Δt5: As causas destes atrasos ocorrem porque o processo do NTP/PTP processados pela Central Processing Unit (CPU), concorre com outros processos que têm maior prioridade, por exemplo, rotinas de proteção em um relé de proteção.
- Δt2 e Δt4: Os motivos destes atrasos acontecem pois na fila de envio do protocolo Transmission Control Protocol (TCP)/IP, muitos tipos de mensagens são armazenadas e posteriormente repassadas aos seus destinos. Porém dentre essas mensagens nem todas são do tipo NTP. Dessa forma o tempo de espera até o seu envio gera atrasos. Somado a isso, outros atrasos são provocados pela disputa de disponibilidade de tempo de processamento da CPU, pois outras rotinas de comunicação com prioridades mais baixas também são executadas.
- Δt3: O principal causador de atrasos, contudo, é provocado pelo *switch*, devido ao tempo de armazenamento e envio de mensagens. Sendo assim, mensagens do tipo GOOSE que têm maior prioridade e podem passar na frente de outros tipos de mensagens e caso aconteça uma sequência dessas mensagens, provocam consideráveis retardos no tempo (LUIZ et al., 2011).

Já no cenário do PTP com suporte a *hardware* (IEEE 1588-2008(v2)), o alcance da precisão do tempo atinge taxas em nanossegundos, porém como já comentado é necessário que todos os elementos de rede (relógio mestre, *switch* e o relógio escravo) possuam essa configuração. O suporte por hardware consiste na captura e geração de *timestamp* diretamente na interface de rede. A Figura 22 mostra o funcionamento do protocolo PTP quando executado por hardware:

- Quando o relógio envia o pacote com o *timestamp* T0, o mesmo só é adicionada ao pacote no momento em que a interface de rede o envia, evitando assim os atrasos ocorridos pela espera de processos mais prioritários e das camadas de protocolos de comunicação.
- No *switch*, é registrado a variação do tempo de chegada até o tempo de saída (Δt0), sendo esse tempo informado ou compensado no pacote que sai do *switch*.
- Por fim, no relógio escravo, o *hardware* registra o momento em que o pacote chega à interface de rede, que depois fornece para o processo PTP quando ele necessitar (LUIZ et al., 2011).

Figura 22 – Diagrama de auxílio de hardware do protocolo PTP.



FONTE: (LUIZ et al., 2011).

2.2.3.1 Terminologias e dispositivos PTP

Na topologia de rede implementada com o PTP são definidos cinco tipos de dispositivos conhecidos como:

- Relógio Comum (Ordinary Clock (OC)).
- Relógio de Limite (Boundary Clock (BC)).
- Relógio Transparente de Ponta a Ponta (TC End-To-End (E2E)).
- Relógio Transparente de Ponto a Ponto (TC Peer-To-Peer (P2P)).
- Nó de gerenciamento (Management node).

A seguir são descritos suas características:

Relógio Comum (OC)

É um dispositivo que se comunica na rede por intermédio de uma única porta PTP (Interface de rede *ethernet*) e que pode ter a função de fornecer o tempo ou de sincronizar o seu tempo. Quando um relógio comum fornece a hora ele é chamado de relógio mestre ou relógio grão-mestre, onde serve o tempo para toda a rede PTP. Oferece a melhor fonte de tempo para todos os outros dispositivos da rede e geralmente possui um receptor GPS (ou outro sistema de satélite) incorporado. Agora quando um relógio comum sincroniza a sua hora, ele é chamado de relógio escravo, no qual sincroniza o seu tempo com outro relógio, como o relógio mestre ou relógio de limite (WATT et al., 2015).

Relógio de Limite (BC)

É um dispositivo de rede com múltiplas portas *ethernet* (*Switch* com suporte a PTP) que sincroniza o seu horário por meio de um relógio mestre conectado em uma de suas portas e fornece o tempo sincronizado nas demais portas que podem estar conectadas com IEDs escravos ou relógios transparentes. Dessa maneira a porta conectada com o relógio mestre é descrita como uma porta escravo e as demais portas são descritas como portas mestre. Os relógios *boundary* iniciam e terminam a distribuição de tempo, no qual essa função é normalmente incorporada aos componentes de rede compatíveis com PTP, como *switches*, pontes e roteadores. Os relógios *boundary* são geralmente usados para aumentar a disponibilidade de uma rede PTP atendendo a solicitações de relógios escravos que, de outra forma, seriam atendidos

pelo relógio mestre. Isso torna possível o suporte a um grande número de IEDs escravos na rede (WATT et al., 2015).

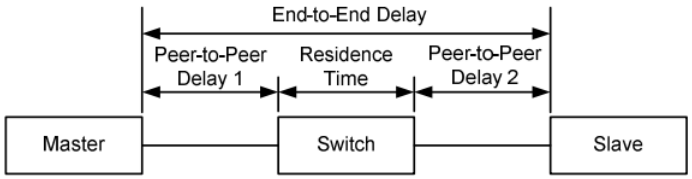
Relógio Transparente de Ponta a Ponta (TC E2E)

É um dispositivo de rede com varias portas *ethernet* (*Switch* com suporte a PTP) que mede o tempo que uma mensagem PTP passa dentro do dispositivo à medida que é roteada da porta de entrada para a porta de saída. Em seguida adiciona essas informações a um campo de correção na mensagem com o objetivo de eliminar qualquer variação nos atrasos e assimetrias nas mensagens que o dispositivo possa introduzir na transferência de mensagens PTP. A função do TC E2E é normalmente realizada por *switches* compatíveis com PTP (WATT et al., 2015).

Relógio Transparente de Ponto a Ponto (TC P2P)

É um dispositivo de rede multi-portas *ethernet* (*Switch* com suporte a PTP) que mede o atraso do link de cada porta e adiciona essas informações e o tempo de permanência nas mensagens PTP que atravessam o dispositivo. Como no TC E2E, o TC P2P também elimina as variações de assimetria e atraso de pacotes no dispositivo, além de permitir o seu dimensionamento, porque os dispositivos *slaves* não precisam enviar solicitações ao relógio mestre para medir o atraso de E2E. Com isso, cada dispositivo calcula o atraso do seu ponto, isto é, o atraso do P2P. A Figura 23 exibe a relação entre as medições de atraso de E2E e de P2P. A medição de atraso P2P é recomendada para redes com caminhos redundantes, pois na mensagem PTP sempre vai estar registrado o atraso real ocorrido na rede, independentemente do trajeto seguido. Com a medição de atraso E2E, o atraso calculado para um caminho pode ser usado para a medição de *offset* para uma mensagem que seguiu um caminho diferente com um atraso diferente (WATT et al., 2015).

Figura 23 – Relação entre medições de retardo ponto a ponto e ponto a ponto.



FONTE: (WATT et al., 2015).

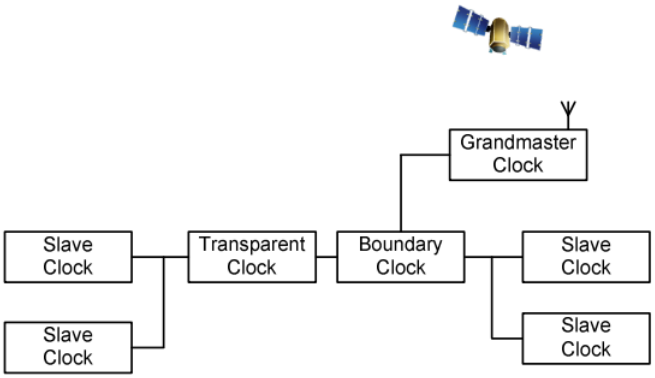
Nó de gerenciamento (Management node)

É um dispositivo conectado à rede, também conhecido como Human Machine Interfaces (HMI), que geralmente é um computador, utilizado apenas para monitorar e ajustar os dispositivos PTP. Dessa forma o modelo de sincronismo mestre e escravo não é aplicado nesse tipo de dispositivo. Além dele, nessa classificação também encontrasse o *switch* não transparente, que é um dispositivo sem suporte à PTP e não leva em consideração o tempo de permanência do tráfego passando por ele (WATT et al., 2015).

Topologia da rede PTP

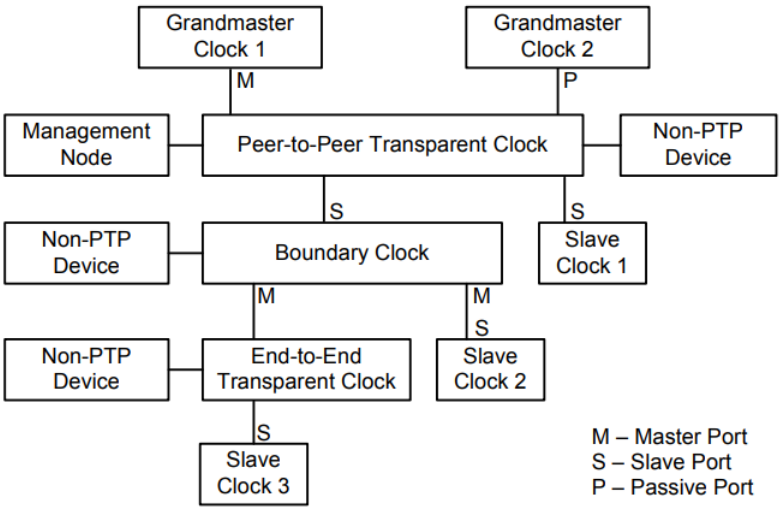
A partir desses dispositivos é implementado a topologia de rede PTP como mostrado na Figura 24, no qual as características de cada componente foram apresentadas anteriormente. Já a Figura 25 exibe a topologia da rede PTP de forma hierárquica e com mais detalhes ilustrando como as portas dos dispositivos com suporte e não ao PTP são configuradas no modelo mestre e escravo. Também é mostrado a utilização do uso de dois relógios mestres, mas após verificado quem é o mais preciso, apenas um é eleito como relógio mestre, enquanto o outro fica como segunda opção e ficando com sua porta de modo passiva (WATT et al., 2015).

Figura 24 – Exemplo de topologia de rede PTP.



FONTE: (WATT et al., 2015).

Figura 25 – Exemplo de hierarquia de relógio PTP.



FONTE: (WATT et al., 2015).

2.2.3.2 Funcionamento do Protocolo PTP

Para sincronizar o relógio escravo com o do mestre é necessário ajustar a frequência dos relógios e compensar o *offset* (Deslocamneto) entre eles. Para ajustar a frequência, são enviadas continuamente mensagens de *Sync* (Sincronizar), conforme mostrado na Figura 26. Se o relógio mestre possuir suporte por *hardware*, então o envio do *timestamp* t1 ocorre na própria mensagem *Sync* (Operação de uma etapa), porém se não possuir, então é enviada uma segunda mensagem *Follow_up* (Acompanhamento) contendo o *timestamp* t1 (Operação de duas etapas) . O escravo recebe a mensagem *Sync* e guarda o *timestamp* t2 do momento da chegada. A frequência do relógio do escravo é ajustada de forma que o intervalo entre o envio das sucessivas mensagens *Sync* sejam iguais ao intervalo entre suas recepções, ou seja:

$$t1^{k+1} - t1^{k} = t2^{k+1} - t2^{k}$$

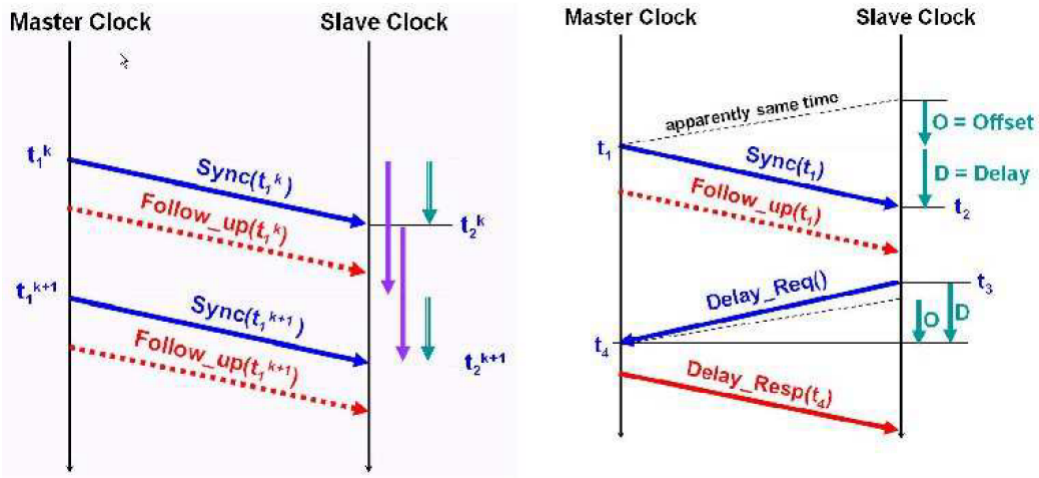
Para o ajuste do *offset* entre os relógios escravos e mestre, é necessário se medir o tempo de atraso do tráfego de mensagens na rede (Round Trip Time (RTT)[9]). Para tanto, o escravo envia mensagens de

[9] RTT – O tempo para enviar um pacote de requisição de eco e o tempo para tê-lo de volta (BRUSCATO et al., 2005).

Delay_Req (Pedido de atraso) para o mestre, guardando o momento de envio t3. O mestre responde com a mensagem Delay_resp (Resposta de atraso), contendo o *timestamp* t4 do momento da recepção da mensagem Delay_Req pelo mestre. Esse processo também é ilustrado na Figura 26 (LUIZ et al., 2011). Considerando que os tempos de envio e resposta são simétricos, então se pode calcular o caminho de atraso (Path_Delay) e o *offset* como:

$$\text{Path_Delay} = [\ (t2\text{-}t1) + (t4\text{-}t3)\ ] / 2$$
$$\text{Offset} = [\ (t2\text{-}t1) - \text{Path_Delay}\ ]$$

Figura 26 – Diagrama de tempo das mensagens do protocolo PTP.



FONTE: (LUIZ et al., 2011).

Operação de uma etapa e duas etapas

O PTP IEEE 1588-2008(v2) com o suporte para hardware *ethernet* permite modificar uma mensagem PTP em tempo real, atualizando o *timestamp* exato à medida que é transmitido. Esse modo de operação não necessita do uso de mensagens de *Follow_up*. Por esse motivo é chamado de operação de "uma etapa". Um relógio mestre de uma etapa transmite o *timestamp* exato na mensagem de *Sync*. O mesmo ocorre nos TCs que fornecem estimativas de atraso de rede no campo ”correção” da mensagem de *Sync* e não na mensagem de *Follow_up*. Isso diminui o número de mensagens enviadas na rede, contudo, esse tipo de processo necessita do uso de um hardware *ethernet* mais sofisticado.

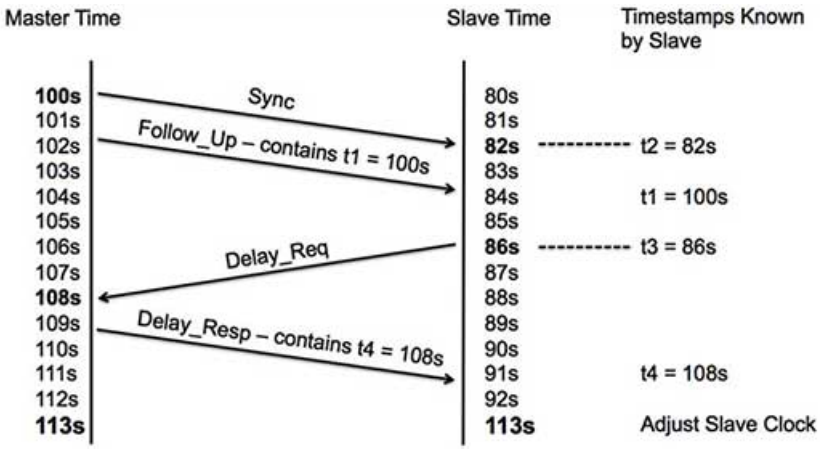
Já no PTP IEEE 1588-2002, por não ter suporte por hardware *ethernet*, não permite à alterção de uma mensagem PTP em tempo real. Perante a esse fato, esse modo de operação necessita do uso de mensagens de *Follow_up* para o enviar o *timestamp* para o relógio escravo. Por esse motivo é chamado de operação de "duas etapas". Dessa forma o mesmo ocorre nos TCs, que adicionam sua estimativa de atrasos na rede mediante ao campo ”correção” da mensagem de *Follow_up*.

Os sistemas PTP podem incluir uma combinação de relógios *masters* de uma e duas etapas e uma combinação de TC de uma e duas etapas. Nesse caso os relógios *slaves* levam em consideração as informações de correção inseridas diretamente nas mensagens de sincronização por TC de uma etapa e as informações atualizadas enviadas nas mensagens de *Follow_up* por TC de duas etapas (INGRAM; SMELLIE, 2019).

Exemplo da sincronização do PTP

A Figura 27 exibe um exemplo do funcionamento da sincronização do PTP, no qual o tempo do relógio mestre inicia com o valor de 100 segundos e o do relógio escravo com o valor de 80 segundos. É executado a sincronização do tempo do PTP pelo método da troca de mensagens de *Sync*, *Follow_up* (Se for necessário), Delay_Req e Delay_Resp. Ao final do processo com o ajuste da frequência dos relógios, a medição do tempo de atraso das mensagens e a compensação do *offset*, o valor do relógio escravo se ajusta com o valor de 113 segundos do relógio mestre (HBM, 2019).

Figura 27 – Exemplo do calculo das mensagens do protocolo PTP.



FONTE: (HBM, 2019).

2.2.4 Resumo dos protocolos sincronização de tempo IRIG-B, NTP e PTP

A Tabela 2 ilustra o resumo das principais características dos protocolos sincronização de tempo IRIG-B, NTP e PTP descritos nas seções anteriores.

Tabela 2 – Resumo dos protocolos de tempo IRIG-B, NTP e PTP.

Métodos de distribuição de tempo	IRIG-B	NTP	PTP IEEE 1588-2008(v2)
Camada física	Cabo Coaxial e Óptico	Cabo Ethernet e Óptico	Cabo Ethernet e Óptico
Modelo	Master (Mestre) e Slave (Escravo)	Cliente e Servidor	Master (Mestre) e lave (Escravo)
Precisão de sincronização	100 ns a 1 μs	1 a 100 ms	100 ns a 1 μs
Compensação de latência	Sim, usando o comprimento do cabo como entrada do usuário	Sim	Sim
Intervalo de atualização	Uma vez por segundo, pulso por segundo (pps)	Minutos	Configurável
Requisitos de hardware	Hardware especial necessário no mestre e escravo	Somente mestre	Suporte de hardware necessário para alta precisão
Custo relativo	Médio (cabeamento IRIG-B)	Baixo (Software)	Médio a alto (Hardware com suporte PTP)

FONTE: (WATT et al., 2015).

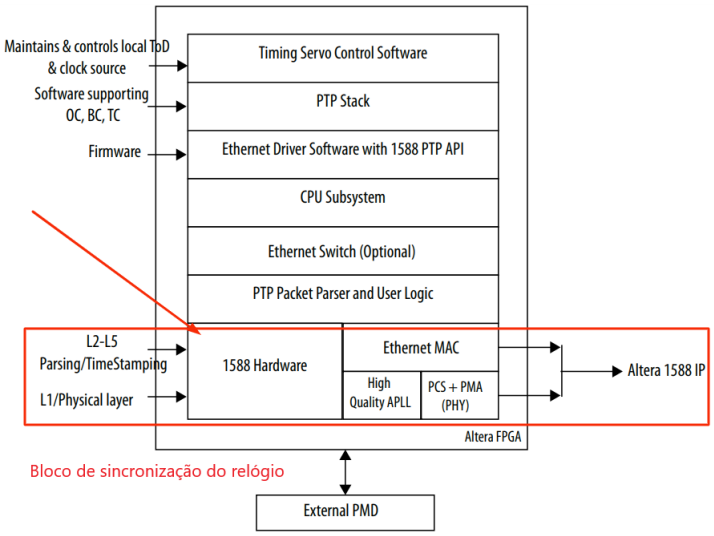
2.3 Projetos do PTPv2 implementados em IP Core comercial e de código aberto

A empresa *Intel* e o autor HA1588 da comunidade de código aberto *OpenCores* projetaram versões do PTP IEEE 1588-2008(v2), implementados em IP Cores (HDL) para a plataforma de desenvolvimento FPGA. O uso do FPGA que é um dispositivo lógico programável, possibilita o uso do suporte por hardware para demostrar os resultados de sincronização de tempo em nanossegundos desse protocolo. Nesse trabalho o foco da análise e execução do IP Core comercial e de código aberto do PTPv2 será apenas em relação ao bloco de sincronização de relógio como descrito nos tópicos seguintes.

2.3.1 Projeto Intel do IP Core PTPv2

A empresa *Intel* implementou um IP' Core (HDL) do PTP IEEE 1588-2008(v2). Sendo assim ela é descrita como uma versão comercial. Nesse primeiro contato não se obteve muitas informações detalhadas desse IP Core, apenas que o foco desse projeto será centralizado no bloco de sincronização de relógio como exibido na Figura 28. Um detalhe relevante, é que por ser tratar de uma versão comercial, à execução desse projeto necessitará do uso de uma licença comercial da *Intel*. Contudo, no Instituto Federal de Santa Catarina (IFSC) Campus São José (SJ) onde esta sendo realizado este trabalho, essa licença está disponível para uso (ALTERA, 2016).

Figura 28 – IP Core da Intel do PTPv2 com foco no bloco da sincronização do relógio.

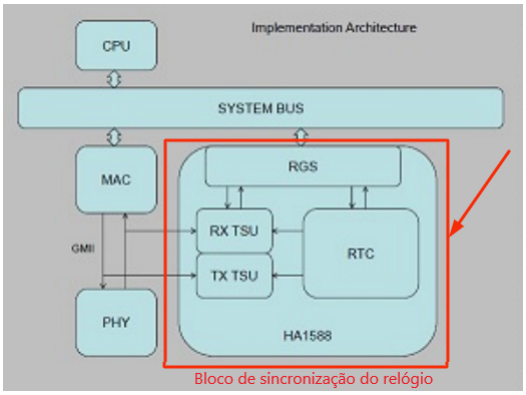


FONTE: (ALTERA, 2016).

2.3.2 Projeto HA1588 do IP Core PTPv2 hospedado no OpenCores

O projeto HA1588 da comunidade mais importante para o desenvolvimento de núcleos de IP', *OpenCores*, é implementado em um IP Core HDL do PTP IEEE 1588-2008(v2). Desa forma ela é chamada de versão de código aberto. Como mencionado na versão comercial, no primeiro contato desse IP' Core, também não se obteve muitas informações específicas. Simplesmente que a base desse projeto será apenas no bloco de sincronização de relógio como exibido na Figura 29. Por ser uma versão de software livre, não é preciso à utilização de nenhuma licença comercial para sua implementação *OpenCores*.

Figura 29 – IP Core do HA1588 do PTPv2 com foco no bloco da sincronização do relógio.



FONTE: (OPENCORES, 2012).

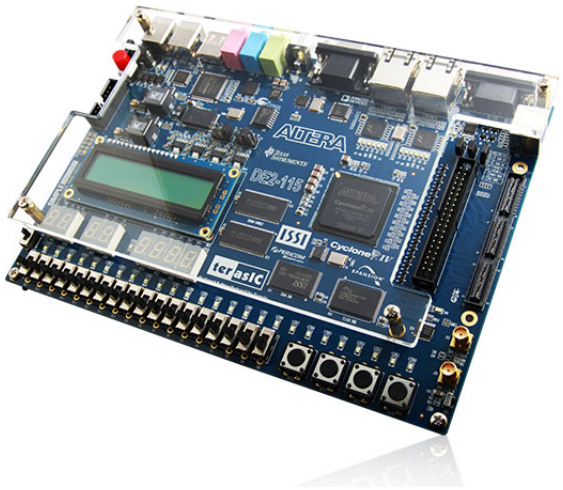
2.4 Cenário proposto para execução dos IP Cores do PTPv2

Aposé as análises e execuções dos IP' Cores (HDL) do PTPv2 da *Intel* e do HA1588 hospedado no *OpenCores*, ambos os projetos são executados em um cenário básico composto por duas placas de desenvolvimentos FPGAs, um osciloscópio e um cabo de rede *ethernet*.

2.4.1 FPGA

É um dispositivo lógico programável, isto é, é um chip composto por um grande número de chaves programáveis, que podem ser configurados para simular o comportamento de qualquer outro tipo de circuito. Um único FPGA pode simular um simples processador, além de outros circuitos de apoio, como o controlador de vídeo, uma interface serial e assim por diante. Os modelos recentes incluem inclusive uma pequena quantidade de memória RAM e circuitos de apoio, de forma que você pode ter um sistema completo usando apenas um chip FPGA previamente programado. A memória é composta por um chip de memória Erasable Programmable Read-Only Memory (EPROM) (Ou memória flash) com um software carregado. A placa de circuito possui trilhas e conectores, como também uma fonte de energia ou bateria. O FPGA também possui o processador Nios II, que oferece flexibilidade sem precedentes para as necessidades de processamento de aplicativos em tempo real, entre outras operações (INTEL, 2019). Nesse trabalho o FPGA utilizado para a implementação do PTP é o da fabricante *Intel* de modelo DE2-115, como exemplificado na Figura 30 (MORIMOTO, 2017).

Figura 30 – FPGA Intel DE2-115



FONTE: (TERASIC, 2019).

2.4.2 Osciloscópio

É um equipamento fundamental utilizado por profissionais da área de elétrica e eletrônica, pois permite visualizar a forma de onda de um sinal elétrico e principalmente analisar determinados parâmetros. Como exemplo temos à frequência, amplitude, tensão média, tensão eficaz, tensão pico a pico e muitos outros parâmetros. Todos os osciloscópios como mostrado na Figura 31 possuem normalmente dois canais de entrada, que são os locais por onde são feitas as leituras dos sinais. Na conexão entre o circuito analisado e o osciloscópio, pontas de provas são conectadas nos canais do osciloscópio. (ELETRICA, 2019).

Figura 31 – Osciloscópio

FONTE: (ELETRICA, 2019).

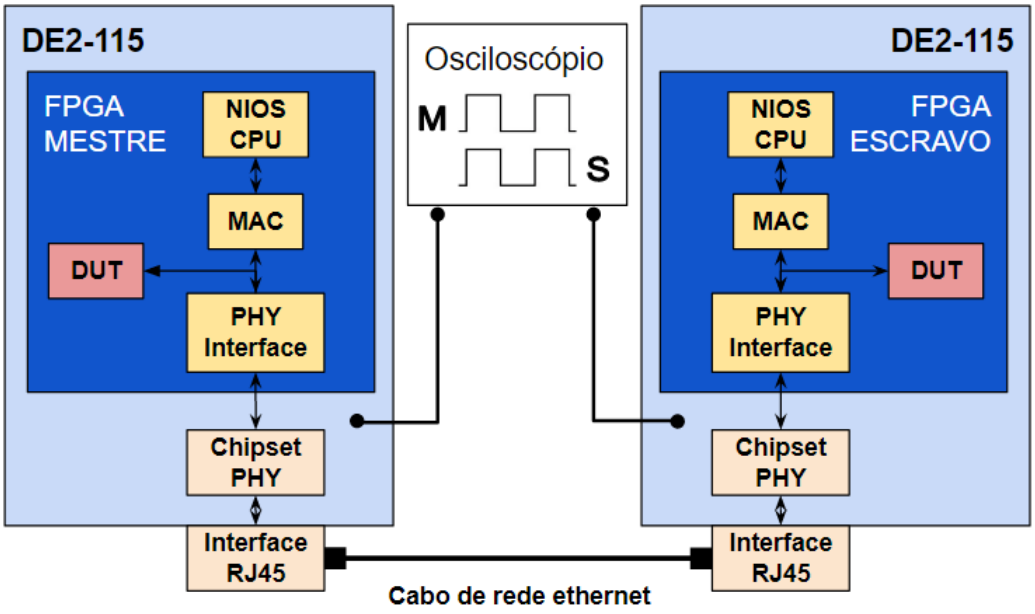
2.4.3 Cabo de rede ethernet

É um cabo de par trançado utilizados em rede *ethernet* (IEEE 802.3), para conectar dispositivos que possuem esse padrão como computadores, tvs, câmeras, FPGA entre outros. A interface de rede desses equipamentos é chamado de adaptador de rede *ethernet*. Nenhum dispositivo pode se conectar a uma rede sem um deles, pois é ele que fornece o endereço físico do computador na rede e converte os dados do dispositivo em um sinal adequado para a transmissão ao longo do cabo. O papel de um cabo de rede *ethernet* é conectar os adaptadores de rede desses dispositivos entre si ou na rede *ethernet*. (PTCOMPUTADOR, 2019).

2.4.4 Cenário

Depois de analisar e executar os IP' Cores (HDL) do PTPv2 da *Intel* e do HA1588 hospedado no *OpenCores*, cada um será executado no projeto de teste (Design Under Test (DUT)) implementado entre a camada física e a sub camada MAC do circuito FPGA, no qual está englobado no cenário básico constituído por duas placas de desenvolvimento FPGAs, onde uma vai estar configurada como mestre e a outra como escravo. Ambas conectadas por um cabo de rede *ethernet* para demonstrar a sincronização de tempo do relógio do FPGA escravo com a do relógio do FPGA mestre, esperando alcançar uma taxa de precisão em nanossegundos. A medição do tempo de sincronismo será analisada a partir de um osciloscópio conectado em ambas as placas que vai verificar se o sincronismo foi executado com sucesso. O cenário descrito é ilustrado na Figura 32. Ao final, será comparado os resultados de cada versão para concluir qual entregou o melhor resultado.

Figura 32 – Cenário básico para executar os IP Cores do PTPv2 para demonstrar o sincronismo de tempo entre os FPGAs mestre e escravo.



FONTE: PRÓPRIO AUTOR.

3 PROPOSTA

3.1 Metodologia

Para realização dos objetivos propostos de maneira que se tenha uma execução organizada, este trabalho é dividido em cinco etapas exibidos nos tópicos seguintes.

3.1.1 Compreensão do protocolo PTPv2

Nessa etapa é executado um estudo detalhado sobre o funcionamento do bloco de sincronização de relógio do protocolo PTPv2 com suporte a hardware, para se obter todo o conhecimento necessário para que seja possível sua análise e execução dos IP' Cores comercial e de códio aberto no FPGA.

3.1.2 Projeto Intel do PTPTv2

Nesse ponto é analisado e executado o IP (HDL) Core comercial do PTPv2 projetado pela *Intel*. A execução desse projeto requer o uso de uma licença paga da *Intel* para o processamento dessa implementação, sendo assim essa implementação requer custos financeiros. No entanto, nesse trabalho não será gasto nenhum valor na licença, pois o IFSC Campus SJ já possui essa licença.

3.1.3 Projeto HA1588 do PTPTv2 hospedado no OpenCores

Nessa etapa é analisado e executado o IP Core (HDL) de código aberto do PTPv2 a partir do projeto HA1588 disponibilizado no site da comunidade de código aberto *OpenCores*. Esse projeto é produzido em um ambiente de software livre, sendo assim, não necessitando de nenhuma licença e tendo custo zero para sua implementação.

3.1.4 Execução dos IP cores no cenário básico dos FPGAs mestre e escravo

Após as análises e execuções dos IP Cores comercial e de código aberto, cada um será executado no projeto de teste (DUT) implementado entre a camada física e a sub camada MAC do circuito FPGA, no qual está englobado no cenário básico constituído por duas placas de desenvolvimento FPGAs, configuradas em mestre e escravo. Ambas conectadas por um cabo de rede *ethernet* para demonstrar a sincronização de tempo do relógio do FPGA escravo com a do relógio do FPGA mestre, esperando alcançar uma taxa de precisão em nanossegundos. A medição do tempo de sincronismo será analisada a partir de um osciloscópio conectado em ambas as placas que vai verificar se o sincronismo foi executado com sucesso.

3.1.5 Validação e testes

Nessa última etapa é feita a validação e testes dos IP' Cores da *Intel* e do HA1588 hospedado na *OpenCores*, configurados no cenário básico a partir dos FPGAs mestre e escravo. Tendo como objetivo verificar quais dos IP' Cores obtém o melhor resultado na sincronização de tempo do relógio do FPGA escravo com a do relógio do FPGA mestre. Esperando-se obter taxas de atualização de sincronismo com valores em nanossegundos. Ao final dessa etapa, será possível saber qual cenário foi o melhor projetado para a sincronização do tempo, o da *Intel* ou o do HA1588 hospedado no *OpenCores*.

3.2 Cronograma

A Tabela 3 apresenta o cronograma das atividades a serem desenvolvidas ao longo dos meses para finalização do Trabalho de Conclusão de Curso (TCC).

Tabela 3 – Cronograma das atividades previstas

Atividades	Mês					
	Fev	Mar	Abr	Mai	Jun	Jul
A1	√					
A2	√	√				
A3		√	√			
A4			√	√		
A5				√	√	
A6					√	
A7	√	√	√	√	√	√
A8						√

- A1: Estudo detalhado do protocolo PTPv2.
- A2: Análise e execução do IP Core (HDL) do PTPv2 do projeto da *Intel*.
- A3: Análise e execução do IP Core (HDL) do PTPv2 do projeto HA1588 hospedado na *OpenCores*.
- A4: Execução dos dois IP Cores no cenário básico dos FPGAs mestre e escravo.
- A5: Testes práticos de ambas as versões no cenário básico dos FPGAs mestre e escravo.
- A6: Avaliação dos resultados.
- A7: Escrita do TCC.
- A8: Defesa do TCC.

REFERÊNCIAS

ALTERA. *Altera 1588 System Solution*. 2016. Disponível em: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an739.pdf>. Acesso em: 20 nov 2019. Citado 2 vezes nas páginas 12 e 35.

BRUSCATO, A. C. et al. *Medição de Desempenho de Redes WAN: Medindo Performance*. 2005. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialwan/pagina_4.asp>. Acesso em: 27 out 2019. Citado na página 32.

ELETRICA, M. da. *O que é osciloscópio e para que serve?* 2019. Disponível em: <https://www.mundodaeletrica.com.br/o-que-e-osciloscopio-e-para-que-serve/>. Acesso em: 03 dez 2019. Citado na página 37.

HBM. *Protocolo PTP na Aquisição de Dados e Ensaios*. 2019. Disponível em: <https://www.hbm.com/pt/5143/protocolo-ptp-na-aquisicao-de-dados-e-ensaios/>. Acesso em: 17 nov 2019. Citado na página 34.

INGRAM, D.; SMELLIE, B. *White Paper on Implementing PTP in Substations*. 2019. Disponível em: <http://digitalsubstation.com/en/2016/11/08/white-paper-on-implementing-ptp-in-substations/>. Acesso em: 22 out 2019. Citado 3 vezes nas páginas 24, 28 e 33.

INSTRUMENTS, N. *Sample Clock*. 2019. Disponível em: <http://www.ni.com/documentation/en/ni-daqmx/latest/mxcncpts/sampclock/>. Acesso em: 05 dez 2019. Citado na página 16.

INSTRUMENTS, N. *Synchronization Basics*. 2019. Disponível em: <https://www.ni.com/pt-br/innovations/white-papers/12/synchronization-basics.html>. Acesso em: 05 nov 2019. Citado 9 vezes nas páginas 15, 16, 17, 18, 19, 20, 21, 22 e 23.

INTEL. *PROCESSADORES NIOS II*. 2019. Disponível em: <https://www.intel.com.br/content/www/br/pt/products/programmable/processor/nios-ii.html>. Acesso em: 02 dez 2019. Citado na página 36.

JUNIOR, P. S. P. et al. *IEC 61850-9-2 Avaliação e testes de um barramento de processos*. 2011. Disponível em: <https://www.osetoreletrico.com.br/iec-61850-9-2-avaliacao-e-testes-de-um-barramento-de-processos/>. Acesso em: 19 nov 2019. Citado na página 23.

KYRIAKAKIS, E.; SPARSO, J.; SCHOEBERL, M. Hardware assisted clock synchronization with the ieee 1588-2008 precision time protocol. In: . Poitiers, France: [s.n.], 2018. p. 51 – 60. Clock Synchronization;FPGA implementations;Hardware assist;IEEE 1588-2008;Precise time protocols;Wcet analysis;. Disponível em: <http://dx.doi.org/10.1145/3273905.3273920>. Citado na página 11.

LUIZ, S. et al. Sincronização temporal de equipamentos usando rede ethernet - análise do estado da arte. In: . [s.n.], 2011. Disponível em: <https://www.researchgate.net/publication/270566616_SINCRONIZACAO_TEMPORAL_DE_EQUIPAMENTOS_USANDO_REDE_ETHERNET_-_ANALISE_DO_ESTADO_DA_ARTE>. Citado 9 vezes nas páginas 12, 24, 25, 26, 27, 28, 29, 30 e 33.

MARTIN, K. *IEEE 1344-1995 - IEEE Standard for Synchrophasors for Power Systems*. 1995. Disponível em: <https://standards.ieee.org/standard/1344-1995.html>. Acesso em: 18 nov 2019. Citado na página 25.

MORIMOTO, C. E. *FPGA*. 2017. Disponível em: <https://www.hardware.com.br/termos/fpga>. Acesso em: 02 dez 2019. Citado na página 36.

NEWS, R. W. *IEEE 1588v2A Look at Time and Frequency Synchronization*. 2014. Disponível em: <https://www.rcrwireless.com/20140513/wireless/ieee-1588v2>. Acesso em: 15 out 2019. Citado na página 11.

NOVAES, J. C. *Desvio Padrão: Aprenda a Calcular!* 2019. Disponível em: <https://matematicabasica.net/desvio-padrao/>. Acesso em: 05 dez 2019. Citado na página 23.

OPENCORES. *The reference community for Free and Open Source gateware IP cores*. 1999. Disponível em: <https://opencores.org/>. Acesso em: 24 nov 2019. Citado na página 12.

OPENCORES. *Núcleo IP IEEE 1588 assistido por hardware*. 2012. Disponível em: <https://opencores.org/projects/ha1588>. Acesso em: 20 nov 2019. Citado na página 36.

PEDRETTI, D. et al. Nanoseconds timing system based on ieee 1588 fpga implementation. *IEEE Transactions on Nuclear Science*, v. 66, n. 7, p. 1151 – 1158, 2019. ISSN 00189499. European organization for nuclear researches;Eye diagrams;Field-programmable gate array implementations;Front end electronics;Hardware implementations;Local area networks (LAN);Precision time protocols;Timing systems;. Disponível em: <http://dx.doi.org/10.1109/TNS.2019.2906045>. Citado na página 11.

PINHEIRO, J. M. S. *Dispersão, Jitter e Latência*. 2004. Disponível em: <https://www.projetoderedes.com.br/artigos/artigo_dispersao_jitter_latencia.php>. Acesso em: 05 dez 2019. Citado na página 23.

PTCOMPUTADOR. *O que o cabo Ethernet Do*. 2019. Disponível em: <http://ptcomputador.com/Networking/ethernet/65741.html>. Acesso em: 03 dez 2019. Citado na página 37.

QI, M.; WANG, X.; YANG, Z. Design and implementation of ieee1588 time synchronization messages timestamping based on fpga. In: . [s.n.], 2011. p. 1566 – 1570. Clock Synchronization;Design and implementations;Hardware circuits;Hardware solutions;IEEE1588;Media Independent Interface (MII);Precise time protocols;Time synchronization;. Disponível em: <http://dx.doi.org/10.1109/DRPT.2011.5994146>. Citado na página 11.

SAMBERG, L. *IEEE-1588 v2 (PTP)*. 2017. Disponível em: <https://wiki.mef.net/pages/viewpage.action?pageId=29230774>. Acesso em: 15 out 2019. Citado na página 11.

SOUZA, R. *Redes IEC-61850 – Estudo de Protocolo e Exemplo de Aplicação (Parte I)*. 2012. Disponível em: <https://www.automacaoindustrial.info/redes-iec-61850-estudo-de-protocolo-e-exemplo-de-aplicacao/>. Acesso em: 26 out 2019. Citado na página 27.

TERASIC. *Altera DE2-115 Development and Education Board*. 2019. Disponível em: <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=502>. Acesso em: 18 out 2019. Citado na página 37.

ULIANA, P. B.; ZIMATH, S. L. *Técnicas de sincronização temporal de equipamentos digitais*. 2001. Disponível em: <http://www.mfap.com.br/pesquisa/arquivos/20081218164225-gpc-029.pdf>. Acesso em: 26 out 2019. Citado 2 vezes nas páginas 24 e 25.

VENTURELLI, M. *Automação Elétrica com IEC 61850*. 2017. Disponível em: <https://www.automacaoindustrial.info/automacao-eletrica-com-iec-61850/>. Acesso em: 19 nov 2019. Citado na página 23.

WATT, S. T. et al. Understanding and applying precision time protocol. In: . Jeddah, Saudi arabia: [s.n.], 2015. Critical component;IEEE 1588;Intelligent electronic device;Power system applications;Power system networks;Precise time synchronizations;Precision time protocols;Substation Automation Systems;. Disponível em: <http://dx.doi.org/10.1109/SASG.2015.7449285>. Citado 7 vezes nas páginas 24, 26, 28, 30, 31, 32 e 35.