Precision Time Protocol

O que é

- O PTP é um protocolo utilizado para sincronizar *clocks* em uma rede.
  - Ele é capaz de alcançar precisão de picosegundos, às vezes até nanosegundos.
- O PTP foi definido inicialmente no IEEE1588-2002, e em 2008 foi definido o chamado PTP versão 2 no IEEE1588-2008
- Ele foi desenvolvido para preencher o vazio que os dois protocolos dominantes até então não preenchiam (NTP e GPS). Para sistemas que necessitam de precisão maior que o NTP apresenta, e para sistemas que não podem se dar ao luxo de ter um receptor GPS em cada nodo da rede.

Protocolo

- Hierarquia mestre-escravo;
- Quatro tipos de relógio: *grandmaster*, *ordinary*, v2(*transparent)*, *boundary*;
  - *Grandmaster* é o *clock* “raíz”, ele que vai transmitir informações de sincronização para os clocks no seu segmento de rede.
  - *Ordinary clocks* são clocks que ou são escravos ou são mestres em um segmento de rede, estes têm apenas uma porta conexão.
  - *Boundary clocks* são clocks que tem várias conexões e estes podem sincronizar outros segmentos de rede.
  - *Transparent clock* é um dispositivo que mede o tempo que uma mensagem PTP levou para transitar e disponibiliza essa informação para *clocks* recebendo a mensagem.

Protocolo

Tipos de mensagens:

- Mensagens de evento (*Timestamped*)
  - SYNC
  - DELAY_REQ
  - PDELAY_REQ (V2)
  - PDELAY_RES (V2)
- Mensagens gerais (*Not timestamped*)
  - FOLLOW_UP
  - DELAY_RESP
  - PDELAY_RESP_FOLLOW_UP (V2)
  - ANNOUNCE (V2)
  - SIGNALING(V2)
  - MANAGEMENT

Protocolo

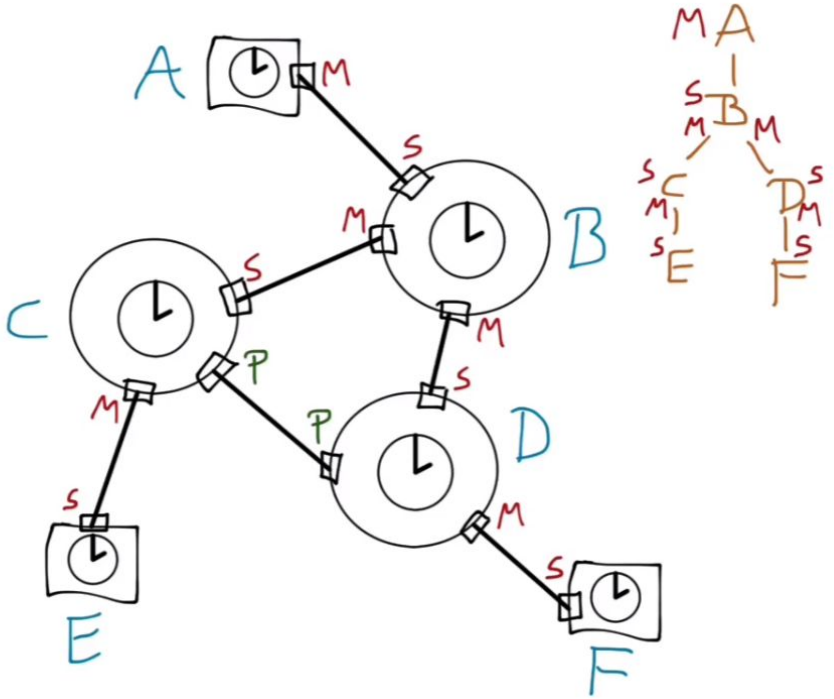
Tipos de mensagens:

- *SYNC*, *FOLLOW_UP*, *DELAY_REQ* e *DELAY_RESP* são utilizadas pelos *boundary e ordinary clocks* para comunicar informações relacionadas ao tempo e sincronizar os *clocks* da rede.
  - *One step clock:* O timestamp do mestre vai na mensagem de SYN
  - *Two step clock:* O timestamp do mestre é enviado na mensage FOLLOW_UP
- *PDELAY_REQ*, *PDELAY_RES* e *PDELAY_RES_FOLLOW_UP* são usados por *transparent* clocks (Não existem na versão 1)
- *SIGNALING e ANNOUNCE* estão presentes também na versão 2 do PTP.
  - SIGNALING para comunicações na rede que não são dependentes da precisão do tempo.
  - ANNOUNCE serve auxiliar na construção da hierarquia e seleção do *Grandmaster clock*
- *MANAGEMENT é* utilizada para monitorar e configurar um sistema PTP.

Hierarquia mestre-escravo

- Relógios podem ter portas nos estados mestre, escravo e passivo;
- O relógio grandmaster possui portas no estado mestre;
- Seus vizinhos terão as portas conectadas a ele no estado escravo, as suas outras portas estarão no estado mestre;
- Os vizinhos dos vizinhos terão as portas conectadas a ele no estado escravo, as suas outras portas estarão no estado mestre, e assim por diante;
- Algumas portas não são usadas para evitar ciclos na rede e estarão no estado passivo;

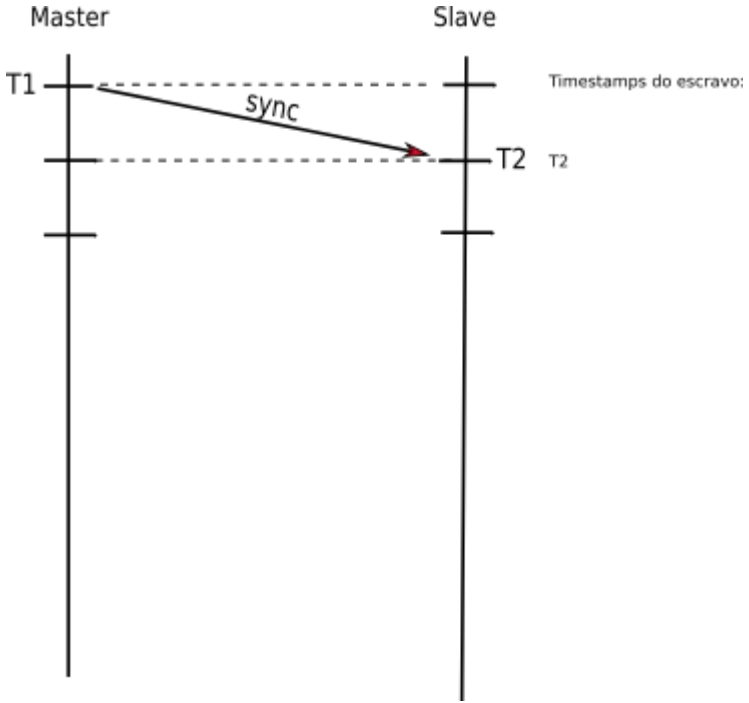
Hierarquia mestre-escravo



Diferenças em relação ao NTP

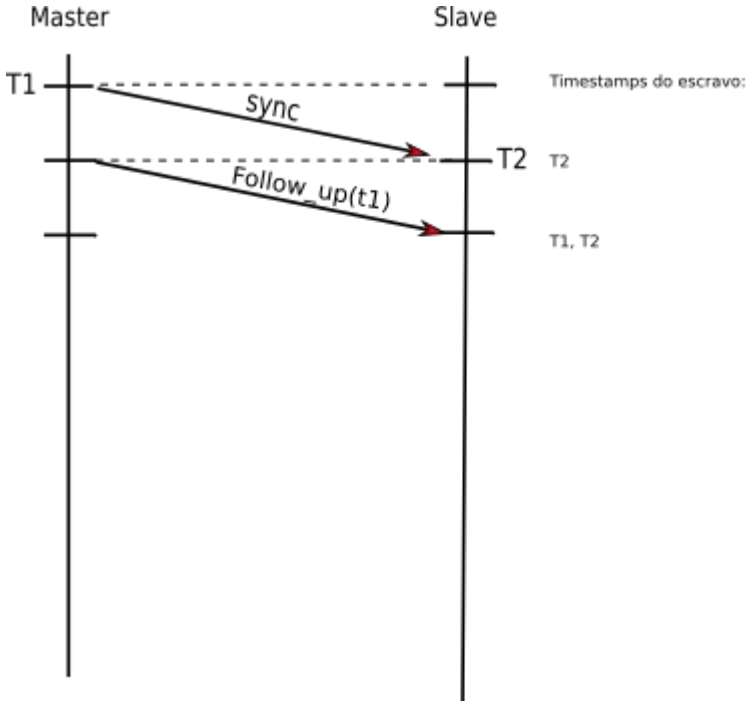
- O principal na hora de se comparar é a precisão. O NTP é preciso a casa do milisegundo, já o PTP consegue atingir precisão na casa dos nanosegundos.
- O motivo disso é que o NTP depende primariamente de algoritmos em *software* para processar o tempo. Já o PTP apesar de ainda necessitar de algoritmos em *software*, ele utiliza de *hardware* para realizar o *timestamping*.
- Outro ponto de diferença, é que o NTP pode ser implementado na rede Ethernet padrão, já o PTP necessita de uma infraestrutura que siga o padrão IEEE 1588

Funcionamento (Two step clock)



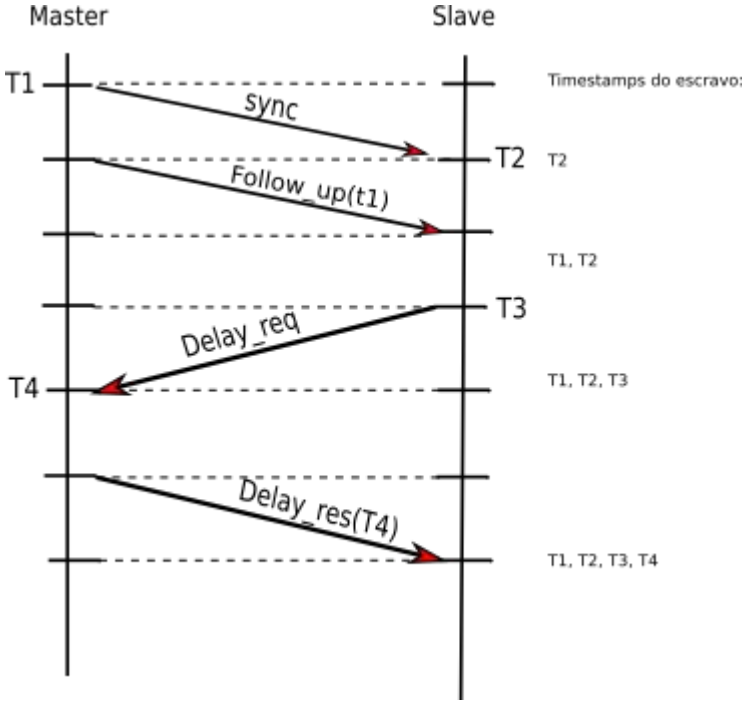
- Para sincronizar o escravo *o* mestre manda uma mensagem de *sync*, e ao enviar essa mensagem, ele guarda localmente o *timestamp* no qual a mensagem foi enviada.
- Ao receber a mensagem de sincronização, o slave então guarda localmente o seu próprio timestamp.

Funcionamento (Two step clock)



- O master então prossegue enviando ao slave uma mensagem *FOLLOW_UP* contendo o timestamp t1.
- Ao receber a mensagem de *FOLLOW_UP* o slave agora conhece o t1 e o t2 (que é o seu próprio timestamp).

Funcionamento (Two step clock)



- Agora o escravo manda ao mestre uma mensagem *DELAY_REQ,* e ao enviar, ele guarda seu timestamp t3.
- O mestre *ao* receber guarda em si o seu próprio *timestamp* t4.
- Depois *mestre* envia ao escravo uma mensagem de *DELAY_RES* com o t4.

Funcionamento (Two step clock)

Calculando o atraso (considerando que o atraso entre o mestre e o escravo é o mesmo do escravo para o mestre):

$$PD = \frac{(T2 - T1) + (T4 - T3)}{2}$$

$$OFFSET\,(DIFERENÇA\ DO\ ESCRAVO\ PRO\ MESTRE) = (T2 - T1) - PD$$

$$Clock_{novo} = Clock_{velho} - OFFSET$$

Problemas do Mundo Real

- Variação no atraso de rede.
  - A variação da rede (transmissão mestre-escravo escravo-mestre) pode causar problemas na precisão do PTP
- Influência da arquitetura sob a qual o PTP está rodando.
  - Algo que é ignorado mas pode ser impactante no PTP é as operações da arquitetura na qual o PTP roda.
  - Ex: Não encontramos PTP em SOs comuns pois, além de outras coisas, a precisão que seria obtida seria algo próximo ao NTP, devido ao não determinismo de operações do SO.
- Variação dos clocks
  - Outro problema também é os clocks terem frequências diferentes.