



UNIVERSIDADE ESTADUAL DO OESTE DO
PARANÁ CENTRO DE CIÊNCIAS EXATAS E
TECNOLÓGICAS COLEGIADO DE CIÊNCIA DA
COMPUTAÇÃO

Relatório do Trabalho Prático – Redes de Computadores

Título: Trabalho Prático NTP

Aluno: Gabriel Santos da Silva

Data: 12/03/2025

1. INTRODUÇÃO

Esse é um relatório sobre o Trabalho Prático realizado na matéria de Redes de Computadores, o Trabalho Prático visa conceitos sobre NTP ao qual realizamos a criação de um cliente e servidor NTP capaz de se conectar com clientes e servidores oficiais.

De acordo com ("NTP.br", [s.d.]), NTP significa Network Time Protocol ou Protocolo de Tempo para Redes. É o padrão que permite a sincronização dos relógios dos dispositivos de uma rede como servidores, estações de trabalho, roteadores e outros equipamentos a partir de referências de tempo confiáveis. Além do protocolo de comunicação em si, o NTP define uma série de algoritmos utilizados para consultar os servidores, calcular a diferença de tempo e estimar um erro, escolher as melhores referências e ajustar o relógio local.

Para esse trabalho prático, utilizamos as especificações dadas pelo professor da disciplina, é necessário a implementação de um cliente NTP que deve ser capaz de solicitar e receber o tempo de servidores NTP compatíveis (incluindo servidores públicos, como pool.ntp.org). Um servidor NTP que deve ser capaz de responder a requisições de clientes NTP, incluindo servidores e clientes oficiais, fornecendo o tempo correto. Além de opcionalmente implementar autenticação usando HMAC-SHA256 com suporte para chaves pré-compartilhadas.

2. OBJETIVOS

Esse trabalho prático teve como objetivo implementar um cliente e um servidor compatíveis com o protocolo Network Time Protocol (NTP). O sistema implementado deverá ser capaz de se comunicar com clientes e servidores NTP oficiais, respeitando o protocolo NTP na versão 3 ou superior. O foco está na construção manual das funcionalidades do NTP, sem o uso de bibliotecas

específicas para NTP, mas com a possibilidade de usar bibliotecas de criptografia para autenticação.

3. MATERIAL UTILIZADO

O trabalho foi realizado no computador pessoal do autor do trabalho. Sendo utilizado uma máquina desktop, com as seguintes configurações, processador Intel Core i7-7700k, 16 GB de memória RAM e possuindo um SSD de 285 GB. A rede utilizada foi a internet cabeada residencial conectada a rede global. O sistema operacional utilizado é o windows 11, ao qual todos os comandos foram realizados no terminal. A IDE utilizada para o desenvolvimento é a Visual Code Studio.

4. IMPLEMENTAÇÃO, DESAFIOS E TESTES

Para o início do trabalho, foi necessário primeiramente o entendimento do Protocolo NTP, visando estudar a especificação oficial do NTPv4 (RFC 5905), com foco no formato de pacote NTP e nas funções de cliente e servidor além de compreender como calcular o offset e o round-trip delay de forma correta.

O segundo passo foi a criação do cliente ao qual manualmente foi criado o pacote NTP com todos os campos necessários, inicialmente testes foram feitos com servidores oficiais, ao receber a resposta todo o cálculo necessário de diferença de tempo e estimar um erro foi feito e ao fim o horário correto é apresentado.

Após isso, foi a criação do servidor ao qual deu um trabalho um pouco maior, já que seu pacote deve possuir uma estrutura mais complexa de dados, inicialmente a resposta ao cliente implementado não apresentou grandes problemas, porém para clientes oficiais obteve uma maior resistência em conseguir montar o pacote corretamente.

Com o cliente e servidor conseguindo se comunicar corretamente entre si, foi implementado um sistema de criptografia e autenticação HMAC, ao qual se utiliza de uma chave compartilhada para verificar a autenticidade das mensagens recebidas.

Por fim, a realização dos testes foi a etapa que mais surgiram bugs já que como dito anteriormente o maior problema foi a comunicação entre clientes oficiais e o servidor implementado, o problema se dava sempre que uma requisição de um cliente oficial era requisitada ao servidor implementado, o servidor tentava enviar o pacote ao cliente, porém o campo 'Leap', 'Version' e 'Mode' permaneciam zerados o que acarretava em um problema de que o campo 'Mode' não poderia ser 0, já que para servidores NTP o padrão é 'Mode' deve ser igual a 4, para resolver isso, depois de muita pesquisa foi descoberto que o modo de inicialização do pacote deveria ser usando uma função

bytearray(48), essa função indica a criação de um array de bytes zerados, que posteriormente podem ser preenchidos com os valores desejados, dessa forma o pacote era devidamente preenchido e as informações necessárias eram corretamente lidas pelo cliente oficial. Outro problema que surgiu durante os testes é a autenticidade sempre dizendo que houve um erro: falha na autenticação do cliente, esse erro era devido ao hmac recebido ser diferente do hmac calculado, esse erro foi facilmente corrigido apenas ajustando a SECRET_KEY que é a chave compartilhada, após definir exatamente a mesma chave tanto para o cliente quanto para o servidor o problema foi corrigido.

5. EXEMPLOS DE TESTES E DESAFIOS

A fim de testar se o cliente implementado estava recebendo o horário correto do servidor implementado, o seguinte teste foi realizado, imagem 1.

Imagem 1 - Cliente implementado recebendo corretamente do servidor implementado.

```
C:\Users\Gabriel-PC2024\Desktop\trab redes>"C:/Program Files/Python313/python.exe" "c:/Users/Gabriel-PC2024/Desktop/trab redes/main.py"
Por favor digite o server: 127.0.0.1
O SERVER É: 127.0.0.1
Modo de criptografia Ativado...
Tempo correto (UTC): Wed Mar 12 15:02:22 2025
```

Observe que o endereço do servidor implementado é o localhost (127.0.0.1) e o modo de criptografia está ativado, o resultado é mostrado com 'Tempo correto (UTC): DIA-SEMANA MÊS DIA H:M:S ANO.'

A fim de testar se o cliente implementado estava recebendo o horário correto dos servidores oficiais, o seguinte teste foi realizado, imagem 2.

Imagem 2 - Cliente implementado recebendo corretamente de servidores oficiais.

```
C:\Users\Gabriel-PC2024\Desktop\trab redes>"C:/Program Files/Python313/python.exe" "c:/Users/Gabriel-PC2024/Desktop/trab redes/main.py"
Por favor digite o server:
O SERVER É: a.st1.ntp.br
Tempo correto (UTC): Wed Mar 12 15:06:34 2025
```

Observe que o endereço não foi informado pelo usuário, então um servidor oficial padrão (a.st1.ntp.br) foi designado e o resultado foi novamente o horário correto.

Um teste também realizado foi com clientes oficiais e o servidor implementado, ao qual podemos ver na imagem 3.

Imagem 3 - Cliente oficial recebendo corretamente do servidor implementado.

```
mint@mint:~/Downloads/TrabalhodeRedes-main$ sudo ntpdate -q 127.0.0.1
2025-03-12 18:44:27.639883 (+0000) +601258565.819919 +/- 601276605.576938 127.0.0.1 s1 no-leap
mint@mint:~/Downloads/TrabalhodeRedes-main$
```

Para esse teste foi utilizado o comando 'sudo ntpdate -q 127.0.0.1' (teste realizado para o sistema LINUX) ao qual ele retorna a data e o horário, mas não altera a hora do sistema. Como podemos observar o resultado retorna o horário correto para o momento do teste.

A imagem 4 a seguir contém a estrutura do pacote presente no servidor implementado, pacote esse que foi necessário sua criação com bytearray(48) para que fosse possível a conexão com clientes oficiais.

Imagem 4 - Criação do pacote de resposta do servidor implementado.

```
resposta_ntp = bytearray(48)
try:
    solicitacao_ntp = struct.unpack("!12I", dados)

    struct.pack_into(
        "!BBBBIIQIIIIII", #formato do pacote
        resposta_ntp, #bytearray onde será armazenado
        0, #offset inicial
        (0 << 6) | (4 << 3) | 4, # LI (0) | Versão (4) | Modo (4 - Server)
        1, #stratum (1 = Primary Server)
        0, #poll Interval
        1, #precision
        solicitacao_ntp[2], #root Delay
        solicitacao_ntp[3], #root Dispersion
        0, #reference ID (0 para servidores de referência)
        int(timestamp_atual), #reference TIMESTAMP
        fracao,
        solicitacao_ntp[8], #origin TIMESTAMP
        int(timestamp_atual), #receive Timestamp
        fracao,
        int(timestamp_atual), #transmit Timestamp
        fracao
    )

    return resposta_ntp
```

Percebe-se que alguns campos como root delay e root dispersion são preenchidos com dados vindos do pacote do cliente.

CONCLUSÕES

O trabalho teve como o principal objetivo apresentar conceitos sobre NTP utilizando da implementação de um cliente e servidor NTP completos capazes de se conectar entre si e com serviços oficiais. Ao final o trabalho cumpriu com seu objetivos, uma vez que condiz com conteúdos abordados em sala de aula e conseguiu exemplificar de forma prática utilizando das implementações.

Dessa forma o trabalho prático cumpre com seu objetivo e suas atividades propostas possibilitando aos alunos o entendimento do funcionamento do cliente NTP, servidor NTP e até noções de criptografia.

BIBLIOGRAFIA

COMER, D. E. (2016). Redes de Computadores e Internet. tradução: José Valdeni de Lima, Valter Roesler. – 6. ed. Editora Bookman, 2016.

KUROSE, Jim. ROSS, Keith. Computer Networking: A Top-Down Approach. 8 ed. Editora Pearson, 2020.

NTP.br. Disponível em: <<https://ntp.br/conteudo/ntp/>>.

