

# Implementação do protocolo telnet com Compressed Network Communication

Glisbel Aponte<sup>1</sup>, Lucas Anderson Ladislau Aguiar<sup>2</sup>, Vinicius<sup>3</sup>

<sup>1</sup>Ciência da Computação – Universidade Federal de Roraima (UFRR)  
CEP – 69310-000 – Boa Vista– RR – Brazil

<sup>2</sup>Departamento de Ciência da Computação

{glisap.n, laanderson183, noleto.vinicius.a}@gmail.com

**Abstract.** *Compressed network communication is a technique used to reduce the size of data transmitted over a network by compressing it before sending it, which can save bandwidth and transmission time. This method involves using algorithms to compress the data, such as gzip or deflate.*

**Resumo.** *A comunicação de rede compactada é uma técnica usada para reduzir o tamanho dos dados transmitidos em uma rede, compactando-os antes de enviá-los, podendo economizar a largura de banda e o tempo de transmissão. Esse método envolve o uso de algoritmos para compactar os dados, como gzip ou deflate.*

## 1. Introdução

No mundo de hoje, as redes de comunicação tornaram-se essenciais em quase todos os aspectos da vida humana. No entanto, essas redes enfrentam desafios significativos quando se trata de transferência de dados devido à sua largura de banda limitada. Essa limitação é causada por inúmeros fatores, como interferência de sinal, congestionamento da rede, distância entre nós de comunicação, entre outros.

Para superar esses desafios, a comunicação em rede comprimida surgiu como uma solução eficaz. A comunicação de rede compactada é um método de reduzir a quantidade de dados transmitidos por uma rede sem comprometer a integridade das informações. Esse método envolve compactar os dados antes de transmiti-los pela rede e descompactá-los na extremidade receptora. Conforme (DA Huffman, 1952).

## 2. Método Proposto

O estudo foi desenvolvido a partir da implementação de um protocolo telnet via tcp, foi feita a tentativa da implementação da comunicação compactada via zlib do C.

## 3. Avaliação Experimental

### 3.1. Objetivo do Compressed Network Communication

O objetivo da compactação é reduzir a quantidade de dados de espaço ocupados. Nos cenários de comunicação, uma versão compactada de dados usará menos largura de banda para transmitir os mesmos dados que a forma não compactada usaria.

### **3.2. Entrada e saída através de um soquete TCP**

O TCP é uma comunicação feita de forma unilateral, que se executa com processos, como: Gestão do controle de fluxo; Conhecimento dos pacotes do processo que chegam na fase final; Divisão dos dados em pacotes; Envio de pacotes.

Na entrada e saída através do soquete TCP em uma linguagem de programação, é necessário o estabelecimento de uma conexão com o cliente e o servidor. O cliente faz o envio de dados para o servidor, enquanto o servidor recebe os dados e processa, enviando uma resposta ao cliente. .

### **3.3 Telnet**

Telnet é um protocolo de rede utilizado para estabelecer uma conexão remota entre computadores em uma rede. Ele permite que o usuário acesse recursos e serviços remotamente, como um servidor de correio eletrônico, servidor de arquivos ou qualquer outro dispositivo que esteja conectado à rede.

Através do Telnet, é possível enviar e receber informações em tempo real, possibilitando o gerenciamento remoto de dispositivos e sistemas. No entanto, o Telnet tem sido substituído por protocolos mais seguros, como o SSH (Secure Shell), devido a suas vulnerabilidades de segurança.

### **3.4 Programa cliente**

Um cliente de telnet permite a um usuário se conectar a um servidor remoto que suporta o protocolo telnet. Geralmente, o usuário precisa especificar o endereço IP ou o nome de domínio do servidor para se conectar a ele. Após a conexão bem-sucedida, o usuário pode enviar comandos e receber respostas do servidor.

### **3.5 Programa do servidor**

Um servidor de telnet permite que os clientes se conectem a ele. Geralmente instalado em um servidor remoto e se executa em segundo plano, aguardando as conexões dos clientes. Quando um cliente se conecta a ele, o servidor autentica o usuário e inicia um canal de comunicação bidirecional entre o cliente e o servidor. O servidor suporta uma variedade de comandos que o cliente pode enviar para execução no servidor. O servidor envia as respostas para os comandos executados pelo cliente, como um fluxo de saída.

### 3.6 A opção --log

Inserindo a opção “--log=filename” na execução do cliente possibilita que o usuário especifique um arquivo de log, basta substituir o filename pelo nome desejado, para registrar mensagens de envio e recebimento de dados mostrando o tamanho dos dados que estão circulando. Isso pode ser útil para monitorar o progresso da compactação e identificar quaisquer problemas que possam ocorrer durante o processo.

Alguns exemplos de mensagens que podem ser registradas em um arquivo de log incluem o número de nós e relações processadas, a quantidade de memória utilizada e quaisquer erros ou avisos que ocorram durante a compactação. Em geral, a opção --log é recomendada para usuários que depurar eventuais problemas.

Além do log temos também outras flags como “- -port” que indica a porta em que a comunicação estará sendo feita assim como o “- -host”

### 3.7 Comunicação comprimida/compactada

Para comprimir os dados foi utilizada a biblioteca zlib, ou mais especificamente as funções compress() e uncompress(). Elas permitiram a compressão e descompressão do buffer enviado a elas, respectivamente.

Essa biblioteca faz o uso do algoritmo deflate para compressão de dados, é basicamente um algoritmo de compactação de dados sem perdas que combina o algoritmo LZ77 (Lempel-Ziv 77) e a codificação de Huffman. O LZ77 é responsável por identificar padrões repetitivos nos dados e substituí-los por referências a esses padrões, enquanto a codificação de Huffman é usada para representar os símbolos de maneira eficiente.

A compactação é útil porque reduz os recursos necessários para armazenar e transmitir dados. O processo de compactação e descompactação consome recursos de computação. A compactação de dados sofre de uma compensação de complexidade de espaço-tempo. Por exemplo, os esquemas de compactação de vídeo podem exigir hardware caro para descompactar o vídeo rápido o suficiente para ser visualizado sem compactação, e a opção de descompactar totalmente o vídeo antes de visualizá-lo pode ser inconveniente ou exigir espaço de armazenamento adicional. O projeto de um esquema de compressão de dados envolve uma compensação entre vários fatores, incluindo o grau de compressão.

No nosso trabalho a compactação está funcionando somente do cliente para o servidor, onde o cliente manda a mensagem compactada, o servidor descompacta a mensagem e a executa enviando a mensagem de retorno ao cliente.

Nossa maior dificuldade foi a implementação da compactação do lado do servidor, pois estamos utilizando a função system() que é responsável por executar um comando no shell. Entretanto, não conseguimos armazenar o valor de saída dessa função de forma correta para que só então pudéssemos fazer a compressão dos dados para o envio de volta ao cliente.

Para ativar a compressão é preciso inicializar tanto o servidor quanto o cliente com a flag “- -compress”.

## 4. Conclusão

A compressão de dados apresenta uma solução eficiente para aumentar a velocidade da transmissão de informações em uma rede. Esta técnica permite reduzir o tamanho dos pacotes de dados, tornando a transmissão mais rápida e diminuindo o tráfego na rede. Além disso, também contribui para a economia de espaço de armazenamento e redução de custos. Sendo assim, uma estratégia importante para melhorar as operações e a eficácia de uma rede em diversos contextos, incluindo ambientes corporativos, acadêmicos e de negócios.

Os maiores problemas enfrentados que podem ser contemplados em trabalhos futuros são a comunicação compactada nas duas vias de comunicação de cliente servidor e a implementação de pipes para o funcionamento dos processos shell do servidor.

## 5. Referências

Huffman D. A. (1952) “A method for the construction of minimum-redundancy codes”, vol. 40, no. 9.

Chen, Euccas. **Understanding zlib**. Disponível em: <https://www.euccas.me/zlib/>. Acesso em: 27 de junho de 2023

Fitzgibbons, Laura. Telnet. Disponível em: <https://www.techtarget.com/searchnetworking/definition/Telnet>. Acesso em: 27 de junho de 2023