

Redes 2012/1

Protocolo Proprietário de Comunicação

Profº Orientador Paulo César Furlanetto Marques

Augusto Weiand
Deividi Schumacher Velho

Linguagem

Utilizamos a linguagem PHP, com orientação a Objeto para facilitar o Debug e desenvolvimento mais "limpo".

A linguagem deriva da "C", sendo assim a utilização do PHP nos traz funções e métodos muito parecidos conseguindo ter uma liberdade igual, mesmo se tratando de uma linguagem Web.

Protocolo

Para o desenvolvimento procuramos levar o modelo fornecido a risca assim como a idéia do trabalho.

Nosso protocolo se desenvolve com a mesma técnica utilizada pela placa de rede de quaisquer computador, definimos um tamanho máximo para o pacote gerado pela aplicação, o qual é exatamente o mesmo que a placa de rede utiliza para transmitir os arquivos dentro da rede.

Protocolo - Metodologia

Para demonstrar a metodologia vou descrever antes o modelo proposto pelo Profº Paulo César, que se baseia sucintamente no descrito abaixo.

Início de cabeçalho = SOH 0x01

Byte de Sincronia = SYN 0x16

Endereço de Origem = 4 bytes

Endereço de Destino = 4 bytes

Tamanho da Mensagem = 2 bytes - 00 até FFFF - tamanho máximo 64 kB

Número de sequência = 2 bytes

Tipo de Mensagem = 1 byte

Checksum cabeçalho = 1 byte calculado

Início da Mensagem = STX 0x02

Mensagem = Máximo 64kb

Finalizador1 = ETX 0x03 (indica o final da mensagem)

Finalizador2 = EOT 0x04 (indica o final da transmissão)

Checksum = 1 byte calculado

Protocolo - Metodologia

Para criar tal protocolo utilizamos da seguinte lógica: Iniciamos o sistema lendo os dados passados pelo usuário, obrigatórios e de formatação rígida (exemplo, endereço de IP, somente um arquivo por vez, sinalizador da mensagem). Após o usuário setar tais parâmetros e submeter o formulário, iniciamos a leitura do arquivo enviado, colocando seu conteúdo disposto em uma matriz com duas posições, sendo a primeira o conteúdo em bytes e a segunda o tamanho do texto armazenado na primeira posição em inteiro, ressaltando que temos uma constante que define o tamanho máximo para a leitura de cada "pacote" armazenado em cada nova posição da matriz.

Protocolo - Metodologia

Após a leitura do arquivo e seu conteúdo quebrado em "pacotes" armazenados na matriz, iniciamos a conversão dos endereços IP passados. Eles foram convertidos em caracteres com a função "chr" do PHP, para podermos escrever eles no arquivo final da forma correta. Com a mesma metodologia escrevemos as demais informações.

Após todas estas conversões e elas armazenadas em variáveis, temos uma outra função para criar os dois checksums (conforme parâmetros pré-estabelecidos).

Com todos estes dados criamos um laço repetitivo que calcula os checksums do pacote atual e os escreve no arquivo do pacote correspondente.

Protocolo - Geração do Pacote

Assim concluímos a(s) geração(ões) do(s) pacote(s) necessário(s). O sistema gera automaticamente o número de pacotes e grava os dados necessários nele.

Protocolo - Reconstrução do Arquivo

A reconstrução do arquivo foi um processo bem simples, pois como sabemos todas as posições onde os dados estão criamos um laço repetitivo condicional, o qual passa todos os arquivos que o usuário enviou e armazena os dados em um array, o qual vai se completando conforme os arquivos vão sendo lidos. Nestas leituras comparamos os checksuns de cada pacote, escrevendo na tela se Correto ou Não.

Ao final da leitura e gavação dos dados do pacote lido no array realizamos a gravação do arquivo no disco, bastando trocar a extensão para a original para o arquivo voltar ao seu estado original.

Protocolo - Descoberta

Conforme fomos realizando os testes descobrimos que o formato que checksums são calculados apresentam um grave problema de integridade. Caso as posições sejam trocadas por alguém ou por algo o checksums não alteram seu valor, por utilizarem a soma do valor dos caracteres e não um hash do estilo CRC.

Protocolo - Aplicações Futuras

Com o andamento deste projeto conseguimos perceber que ele pode ser utilizado em diversos locais, como criptografia de dados, troca de informações entre outros, inclusive nossa experiência com valores hexadecimais e operações com arquivos.