



Universidade Federal do Rio Grande do Norte - UFRN
Centro de Ensino Superior do Seridó - CERES
Departamento de Computação e Tecnologia - DCT

Curso: Bacharelado em Sistemas de Informação
Disciplina: DCT2102 – Redes de Computadores
Professor: João Borges
Data: 11 de novembro de 2024

Atividade 1.1 – Dupla

Projeto e Implementação de Protocolo de Aplicação

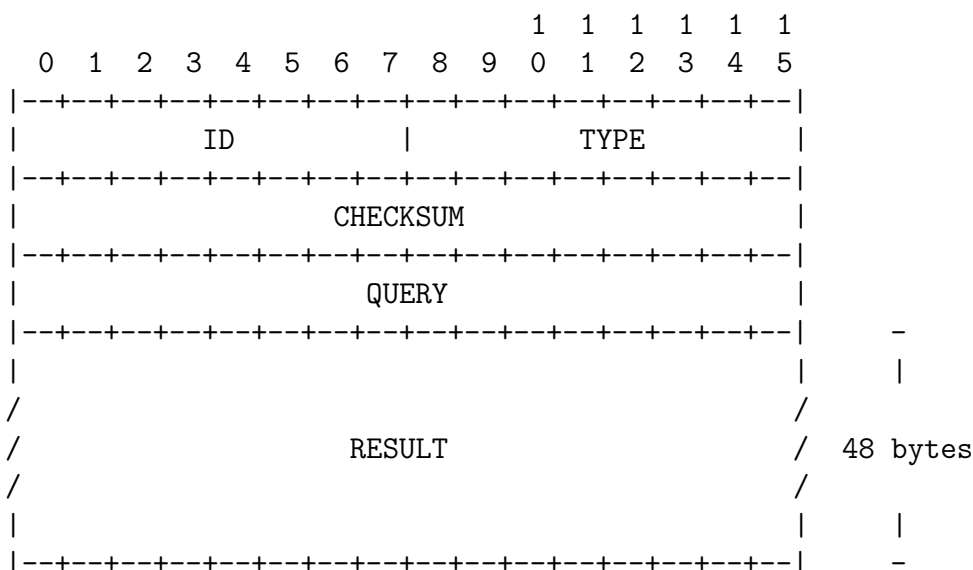
ATENÇÃO 1: Só serão aceitos trabalhos em **Dupla** ou **Individual**, mais do que isso invalidará o trabalho;

ATENÇÃO 2: Não serão permitidos plágios entre os grupos, sendo punidos, ambos os grupos que tiverem seus trabalhos iguais, com nota 0 (zero).

Descrição da Atividade Esta atividade consiste nas atividades de projeto e implementação de um cliente e um servidor que se comunicam seguindo o protocolo de aplicação **Network Server Information Protocol (NSIP)**.

Este protocolo permite que um cliente possa ter acesso a informações sobre o *status* e funcionamento de um determinado servidor remoto. Abaixo seguem os requisitos do protocolo:

1. O protocolo segue o modelo de comunicação cliente/servidor (requisição/resposta)
2. O protocolo deve operar sobre o protocolo de transporte UDP
3. O servidor escuta conexões na porta UDP 2102
4. O protocolo NSIP trafega pacotes com o seguinte formato:



5. O mesmo pacote é utilizado tanto para requisição quanto resposta

6. Descrição dos campos do pacote:

- ID (8-bits) – Utilizado para identificar a consulta, deve ser utilizado o mesmo valor para requisição e resposta
- TYPE (8-bits) - Tipo do pacote, define se o pacote é uma requisição (request) ou uma resposta (reply):
 - **0** - NSIP_REQ – Requisição
 - **1** - NSIP_REP – Resposta
 - **2** - NSIP_ERR – Erro
- CHECKSUM (16-bits) – Campo verificador de erros, é calculado conforme descrito abaixo no item 9
- QUERY (16-bits) – Corresponde ao tipo da consulta, isto é:
 - Se o tipo (campo TYPE) for uma requisição, significa qual a informação que o cliente deseja consultar no servidor
 - Se o tipo for uma resposta, indica o significado do valor retornado pelo servidor no campo RESULT
- RESULT (48-bytes)
 - Se o tipo (campo TYPE) for uma requisição:
 - * O campo RESULT deverá ir em branco
 - Se o tipo (campo TYPE) for uma resposta:
 - * O campo RESULT corresponde ao resultado enviado pelo servidor para a consulta (QUERY) realizada

7. O servidor NSIP deverá responder aos seguintes tipos de consultas (campo QUERY):

- SYS_PROCNUM [0x0] - Número de processos em execução no servidor
- SYS_BOOTIME [0x1] - Tempo em que o servidor está ligado (formatar adequadamente)
- CPU_COUNT [0x2] - Número de CPUs do servidor
- CPU_PERCT [0x3] - Porcentagem de utilização do servidor
- CPU_STATS [0x4] - Estatísticas da CPU (número de trocas de contexto e interrupções)
- MEM_TOTAL [0x5] - Memória total do servidor
- MEM_FREE [0x6] - Memória disponível no servidor
- MEM_PERCT [0x7] - Porcentagem de uso da memória no servidor
- DISK_PARTS [0x8] - Lista de partições dos discos do servidor
- DISK_USAGE [0x9] - Lista de utilização das partições
- NET_IFACES [0x10] - Lista de interfaces de rede do servidor
- NET_IPS [0x11] - Lista de IPs das interfaces de rede
- NET_MACS [0x12] - Lista de MACs das interfaces de rede
- NET_TXBYTES [0x13] - Quantidade de bytes enviados
- NET_RXBYTES [0x14] - Quantidade de bytes recebidos

- NET_TXPACKS [0x15] - Quantidade de pacotes enviados
 - NET_RXPACKS [0x16] - Quantidade de pacotes recebidos
 - NET_TCPCONS [0x17] - Quantidade de portas TCP ouvindo conexões
 - NET_TCPLIST [0x18] - Lista das portas TCP ouvindo conexões
 - NET_UDPCONS [0x19] - Quantidade de portas UDP ouvindo conexões
 - NET_UDPLIST [0x20] - Lista das portas UDP ouvindo conexões
8. Para obter as informações que o servidor deverá responder nas consultas, poderá ser utilizado o pacote Python `psutil`¹.
9. Os resultados que contém listas como resposta, como as portas abertas (TCPLIST ou UDPLIST), etc, deverão conter os seus valores separados por vírgula (,) sem conter espaços entre eles.
10. Para o cálculo do campo CHECKSUM:
- Este campo corresponde ao somatório de todos os 54 bytes do pacote enviado, limitado em 16-bits, isto é, operado em módulo de 2^{16}
 - Equação para cálculo do checksum

$$checksum = \left(\sum_{i=0}^{53} B_i \right) \bmod 2^{16} \quad (1)$$

- Para computar o checksum, o campo CHECKSUM deverá ser 0 (zero).
 - Ambos cliente e servidor devem verificar a integridade dos pacotes que receberam a partir da verificação do checksum.
11. O arquivo contendo o cabeçalho dos pacotes NSIP, bem como funções úteis, estará disponível no SIGAA com o nome `nsip.py`, e deverá ser utilizado (incluído) nas implementações do protocolo.

OBS.: para incluir o código-fonte disponibilizado com a classe, constantes e funções NSIP nos códigos `cliente.py` e `servidor.py`, é preciso fazer o seguinte:

```
from nsip import *
```

Alguns exemplos de como utilizar a classe NSIP, suas funções e constantes, estão disponíveis ao final do arquivo `nsip.py`.

Os códigos-fonte das implementações do cliente e do servidor NSIP deverão ser enviados por um dos integrantes da dupla pelo SIGAA, até a data estabelecida da tarefa cadastrada no sistema.

¹<https://psutil.readthedocs.io/en/latest/>