Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Comunicações por Computador

Trabalho Prático 2

Ano Letivo de 2024/2025

Network Monitoring System

Flávia Alexandra da Silva Araújo (A96587)
Joshua David Amaral Moreira (A105684)
Miguel Torres Carvalho (A95485)

28 de novembro de 2024

CC

Resumo

«O resumo tem como objectivo descrever de forma sucinta o trabalho realizado. Deverá conter uma pequena introdução, seguida por uma breve descrição do trabalho realizado e terminando com uma indicação sumária do seu estado final.»

Área de Aplicação: «Identificação da Área de trabalho.»

Palavras-Chave: «Conjunto de palavras-chave que permitirão referenciar domínios de conhecimento, tecnologias, estratégias, etc., directa ou indirectamente referidos no relatório.»

Índice

Índice de Figuras

Índice de Tabelas

Índice de *Snippets*

1 Arquitetura da Solução

Dúvida: o diagrama da arquitetura da solução deve ser mais geral ou mais específico à solução (python) desenvolvida?

2 Especificações dos Protocolos Aplicacionais

2.1 *AlertFlow*

2.1.1 Formato de Mensagem

2.1.2 Diagrama de Sequência

- Normal

2.2 *NetTask*

O protocolo *NetTask* é essencial para a funcionalidade harmonizada do ***Network Monitoring System***, sendo este usado para a maioria das comunicações entre o servidor e os agentes, tais como, a primeira conexão de um agente ao servidor, o envio de tarefas pelo servidor, o envio de resultados de tarefas pelos agentes, e a terminação de conexões nos dois sentidos.

Deste modo, o protocolo *NetTask* foi desenvolvido para ser robusto e adaptável a condições adversas de rede, garantindo a entrega fiável e integral de mensagens, sobretudo em rotas deterioradas, com perdas ou duplicação de pacotes, latências elevadas e taxas de débito variáveis.

Para combater tais adversidades, o protocolo aplicacional *NetTask* responsabiliza-se pelas funcionalidades que serão exploradas no subcapítulo Descrição de Funcionalidades.

Nos subcapítulos seguintes, serão detalhadas as especificações do protocolo, nomeadamente o formato do cabeçalho, descrição de funcionalidades e diagramas de sequência que ilustram o comportamento do protocolo em situações normais e adversas.

2.2.1 Formato de Cabeçalho e Descrição de Campos

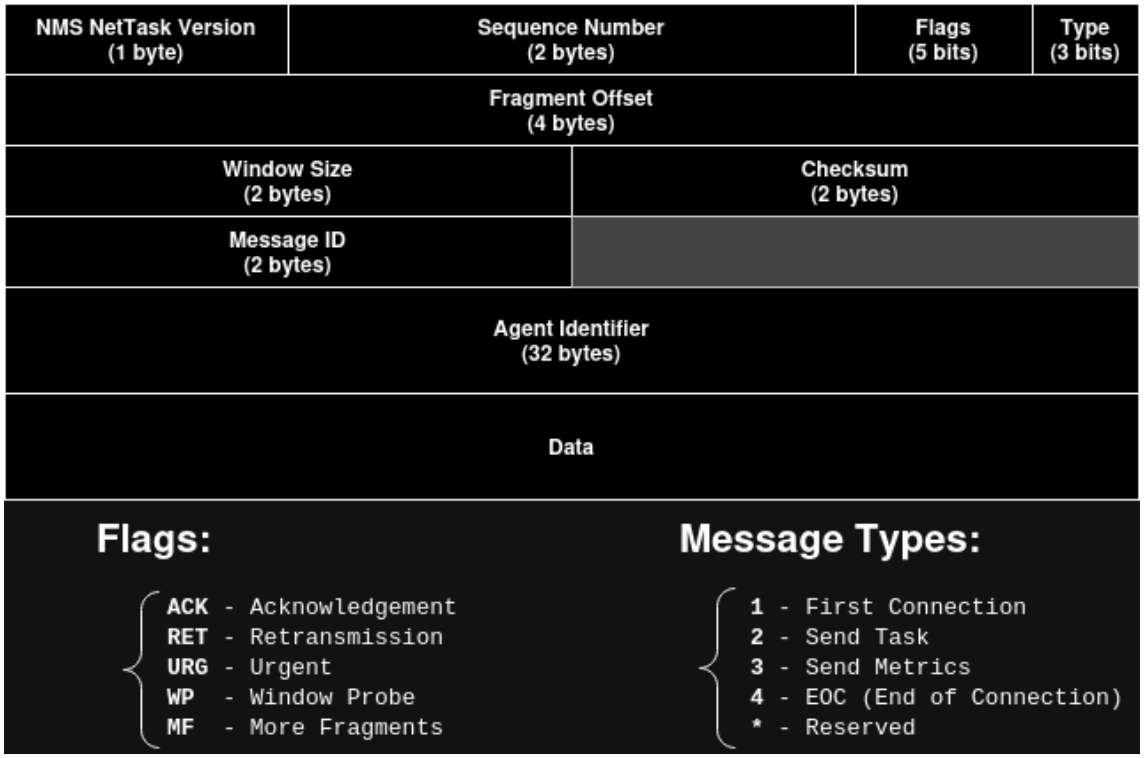

Figura 2.1: Formato do cabeçalho do protocolo *NetTask*

- ***NMS NetTask Version*** (1 *byte*): versão do protocolo, para a compatibilidade de versões entre o servidor e os agentes;
- ***Sequence Number*** (2 *bytes*): número de sequência da mensagem, para a ordenação de pacotes, deteção de pacotes duplicados e identificação de *acknowledgements*;
- ***Flags*** (5 *bits*): *flags* de controlo:

  ACK (1º *bit*): *Acknowledgement*, utilizado para confirmar a receção de pacotes;

  RET (2º *bit*): *Retransmission*, indica que o pacote é uma retransmissão;

  URG (3º *bit*): *Urgent*, indica que a mensagem é urgente;

  WP (4º *bit*): *Window Probe*, utilizado para controlo de fluxo;

  MF (5º *bit*): *More Fragments*, para (des)fragmentação de pacotes.

- ***Type*** (3 *bits*): tipo da mensagem:
  - ***0 - Undefined*** : mensagem indefinida, utilizada para testes ou quando ne-

nhum tipo de mensagem é aplicável, por exemplo no envio de *window probes*;

  - ***1 - First Connection*** : primeira conexão de um agente ao servidor;
  - ***2 - Send Task*** : envio de tarefas pelo servidor;
  - ***3 - Send Metrics*** : envio de resultados de tarefas pelos agentes;
  - ***4 - EOC (End of Connection)*** : terminação de conexões nos dois sentidos;
  - **** - Reserved*** : reservado para futuras extensões. (códigos de 5 a 7);

- ***Window Size*** (2 *bytes*): indica o tamanho da janela de receção, para controlo de fluxo;
- ***Checksum*** (2 *bytes*): soma de verificação da mensagem, para deteção de erros;
- ***Message Identifier*** (2 *bytes*): identificador da mensagem, utilizado para a desfragmentação e ordenação de pacotes;
- ***Agent Identifier*** (32 *bytes*): identificador do agente, podendo este ser do recetor ou do emissor da mensagem;
- ***Data/Payload*** (n *bytes*): carga útil da mensagem com tamanho variável, contendo a informação a ser transmitida nas mensagens do tipo *Send Task* e *Send Metrics*.

2.2.2 Descrição de Funcionalidades

Fragmentação de Pacotes

Retransmissão de Pacotes Perdidos

Deteção de Erros

Ordenação de Pacotes

Deteção e Manuseamento de Pacotes Duplicados

Controlo de Fluxo

Conpatibilidade de Versões

2.2.3 Diagramas de Sequência

- First Connection

- End of Connection
- Retransmission
- Desfragmentação e Ordenação de pacotes
- Controlo de Fluxo
- Deteção de Erros
- Deteção e Manuseamento de Pacotes Duplicados

3 Implementação

3.1 Estrutura do Projeto?

3.2 Parâmetros dos Executáveis

3.3 Ficheiro de Configuração

3.4 Bibliotecas Utilizadas

Dúvida: deve-se referir todas as bibliotecas utilizadas no desenvolvimento?
por exemplo: *socket*, *os*, *sys*, *threading*, etc Ou apenas as bibliotecas mais relevantes para a solução? *ping3*, *iperf*, etc

3.5 Detalhes Técnicos? Maybe

4 Testes e Resultados

Para além da testagem manual contínua ao longo do desenvolvimento, foram desenvolvidos testes unitários para as classes e funções mais críticas do sistema. Estes testes foram desenvolvidos com recurso à biblioteca *unittest* do Python.

5 Conclusões e Trabalho Futuro

Referências

[1] Connolly, T., & Begg, C. (2015). Database Systems: Example (6th ed.). Pearson Education. London, UK.

Anexos

[I] Exemplo de Anexo